

UNIVERSIDAD DE CONCEPCIÓN

FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA



Profesor Patrocinante:

DSc. Pablo E. Aqueveque N.

Informe de Memoria de Título
para optar al título de:

Ingeniero Civil Biomédico

Diseño de Teclado Computacional Capacitivo para Personas con Limitaciones Motoras

UNIVERSIDAD DE CONCEPCIÓN
Facultad de Ingeniería
Departamento de Ingeniería Eléctrica

Profesor Patrocinante:
DSc. Pablo E. Aqueveque N.

Diseño de Teclado Computacional Capacitivo para Personas con Limitaciones Motoras

Marco Alejandro Ulloa Chandía

Informe de Memoria de Título
para optar al Título de

Ingeniero Civil Biomédico

Marzo 2016

Resumen.

En este trabajo se presenta lo que son los sensores capacitivos de proximidad y las utilidades que han tenido sobre todo en el campo biomédico. Además se muestran las necesidades que presentan los pacientes con dificultad motora (la cual es producida principalmente por lesiones medulares), la importancia de estimularles en la rehabilitación y ayudarles en el uso de la tecnología.

Por esta razón en este trabajo se mencionan los pasos seguidos para conseguir un modelo funcional de un teclado capacitivo, el cual se activa con el tacto y tiene como finalidad ayudar a las personas con dificultad motora a utilizar de manera más fácil un computador.

El teclado consta de una placa en PCB sobre la cual se crearon seis electrodos con los que se detecta el tacto de los dedos a través de una lámina de plástico. Debido a las características del paciente objetivo el tamaño de los electrodos es de 20 centímetros cuadrados.

El dispositivo se creó para funcionar de tres formas diferentes: la primera es emular las teclas direccionales de un teclado convencional junto con su tecla “*Enter*”, la segunda es controlar el funcionamiento del “*mouse*” donde se pueden hacer desplazamientos en cuatro direcciones y realizar “*clics*” con el botón izquierdo del mismo, finalmente en la tercera forma de funcionar se puede escribir tocando cada botón por un determinado tiempo hasta que aparezca la letra del abecedario que se desee, además están presentes las teclas borrar, “*Enter*” y espacio. El teclado posee tres LED que se encargan de indicar que funcionalidad está activa.

Para crear las funcionalidades descritas se implementó un programa en Arduino, el que se probó en una tarjeta de desarrollo modelo Leonardo. En esta tarjeta se usaron librerías que permitieron simular un “*mouse*” y un teclado además de otra librería que le permitió actuar como sensor capacitivo.

Una vez creado el dispositivo, se realizaron pruebas en personas sanas a las cuales se les pidió evaluar el desempeño del teclado. Como resultado un 80% de las personas pudo realizar todo lo que deseaba en el computador, mientras que un 80% de ellos lo considera útil al punto de recomendarlo para este tipo de pacientes. El 10% reconoció dificultades en su uso principalmente por ser más engorroso que un teclado común.



“Cada cosa tiene su precio, ya sea material o inmaterial. Si crees que algo es gratis se debe a dos casos: el primero es que alguien lo pagó por ti, el segundo es que aún no sabes lo que te ha costado.”

Agradecimientos.

Primeramente quiero agradecer a Dios por permitirme llegar a este punto, he de reconocer que ciertamente él es fiel a sus promesas.

Deseo agradecer a mi familia por hacer posible el trayecto que me llevó a realizar este trabajo. Comenzaré nombrando a las tres mujeres más importantes hasta este momento en mi vida: Mi madre, Virginia, quien ha creído en mí y me ha apoyado en todo momento, jamás olvidaré el esfuerzo que has hecho para que yo tenga la posibilidad de estudiar; A mi tía Marisa, por estar siempre presente y pendiente de mí con el consejo oportuno y un apoyo incondicional, y mi madrina Nieves, la cual me ha brindado su experiencia y me ha tenido constantemente en sus oraciones. A mi “tata”, Manuel, por estar presente y ayudarnos siempre; mis hermanos Ismael y Sebastián, sin los cuales la realización de mi memoria habría sido muy distinta, muchas gracias por esos “favores” y la motivación que me entregaron. Siempre es una alegría compartir con ustedes.

Quiero dedicar también unas palabras de agradecimiento a Don Ramiro, quien hizo gestiones no sólo una, sino cuatro veces con la finalidad de ayudarme en mi desarrollo académico. Gracias por ayudarme desde mi comienzo en la universidad y por hacerse presente incluso hasta en la realización de esta memoria.

Agradezco también a quienes con su ayuda hicieron posible mi avance en el desarrollo de esta memoria. Mi profesor guía, Pablo, además de Javier, Don Alejandro y Byron. Los cuáles con su disposición, consejo y ayuda me permitieron alcanzar esas pequeñas metas que uno se propone antes de alcanzar el objetivo final.

A mis mejores amigos en la vida universitaria: Natalie, Christopher y Leonardo. Con ustedes pude crecer como persona, entender otros puntos de vista y abrirme a nuevas realidades. El tiempo nos ha hecho madurar, pasar altos y bajos, realizar proyectos y tener distintas vivencias. Tras todo aquello debo decir que me alegra haber recorrido este camino con ustedes. Gracias por todo y espero continuemos siendo grandes amigos.

Finalmente, a mis compañeros, Valerie, Laura, Mariana, Ingrid, Juan, Britam, María Elena y Daniela. Con los cuales pasamos muchas jornadas de estudio, realizando distintos trabajos y compartiendo el tiempo libre. Les agradezco su ayuda, paciencia y apoyo, con ustedes aprendí que no es necesario ser bueno en todo, y que si se camina solo se puede llegar más rápido, pero si se camina acompañado se puede llegar más lejos.

Tabla de Contenidos.

LISTA DE TABLAS	VIII
LISTA DE FIGURAS	IX
NOMENCLATURA	XI
CAPÍTULO 1. INTRODUCCIÓN.	1
1.1. INTRODUCCIÓN GENERAL	1
1.2. OBJETIVOS	3
1.2.1 <i>Objetivo General.</i>	3
1.2.2 <i>Objetivos Específicos.</i>	3
1.3. ALCANCES Y LIMITACIONES.	3
1.4. METODOLOGÍA.	3
1.5. TEMARIO.....	4
CAPÍTULO 2. REVISIÓN BIBLIOGRÁFICA.	5
2.1. INTRODUCCIÓN.	5
2.2. PRINCIPIOS DE FUNCIONAMIENTO DE SENSORES CAPACITIVOS.	5
2.3. AVANCES EN EL USO BIOMÉDICO DE LOS SENSORES CAPACITIVOS.....	6
2.4. MÉDULA ESPINAL	8
2.5. LESIÓN MEDULAR Y DIFICULTAD MOTORA.....	10
2.6. USOS DE SENSORES CAPACITIVOS EN REHABILITACIÓN DE PACIENTES CON DIFICULTAD MOTORA.....	12
2.7. DISCUSIÓN.	14
CAPÍTULO 3. OPCIONES DE DISEÑO.	15
3.1. INTRODUCCIÓN.	15
3.2. ESQUEMA GENERAL.	15
3.3. DETERMINACIÓN DE FUNCIONES Y CANTIDAD DE ELECTRODOS.	16
CAPÍTULO 4. HARDWARE DEL TECLADO.	23
4.1. INTRODUCCIÓN.	23
4.2. SELECCIÓN DEL MICRO-CONTROLADOR (MCU).	23
4.3. MICRO-CONTROLADOR (MCU) ARDUINO LEONARDO COMO SENSOR CAPACITIVO.	24
4.4. MODELO DEL TECLADO.	26
CAPÍTULO 5. SOFTWARE DEL TECLADO.	29
5.1. INTRODUCCIÓN.	29
5.2. IMPLEMENTACIÓN DE LAS FUNCIONES DEL TECLADO.....	29

5.3.	DETECCIÓN DEL TACTO CON LOS ELECTRODOS	33
5.3.1	<i>Funciones Necesarias</i>	33
5.3.2	<i>Determinación de Umbral para Toques Directos en los Electrodo</i> s	34
CAPÍTULO 6. MODIFICACIONES PARA LOGRAR MODELO FINAL.		37
6.1.	INTRODUCCIÓN	37
6.2.	PRUEBAS CON SEIS ELECTRODOS.....	37
6.3.	PRUEBAS Y MODIFICACIONES EN LA PLACA PCB	41
6.4.	MODELAMIENTO CUBIERTA DEL TECLADO.....	46
CAPÍTULO 7. CONCLUSIONES.		48
7.1.	CONCLUSIONES.....	48
7.2.	TRABAJO FUTURO.....	49
BIBLIOGRAFÍA.....		50
ANEXO A.	VISTAS MODELO DE CUBIERTA DEL TECLADO.....	53
ANEXO B.	GUÍA RÁPIDA DE USUARIO.....	56
ANEXO C.	CUESTIONARIO DE APROBACIÓN.....	58
ANEXO D.	CÓDIGO DEL TECLADO	59



Lista de Tablas.

Tabla 2.1 Escala de clasificación de la lesión medular. 11



Lista de Figuras.

Figura 2.1: Detección de proximidad a través de un electrodo. Adaptado de [3].	6
Figura 2.2: Aplicación de sensores capacitivos para tacto en brazo robótico. Adaptado de [4].	7
Figura 2.3: Anatomía transversal de la médula espinal	9
Figura 2.4: Juego creado para Rehabilitación de pacientes con dificultad motora. Adaptado de [19].	13
Figura 2.5: Teclados de Concepto.	13
Figura 3.1: Esquema General del Hardware a Diseñar.	16
Figura 3.2: Diagrama de flujo con la metodología utilizada para definir opciones y n° de teclas necesarias.	17
Figura 3.3: Funciones propuestas en el primer modelo para la implementación del teclado.	17
Figura 3.4: Funciones propuestas en el segundo modelo para la implementación del teclado.	19
Figura 3.5: Funciones propuestas en el tercer modelo para la implementación del teclado.	21
Figura 4.1: Imagen de MCU utilizado (Arduino Leonardo).	23
Figura 4.2: Esquema de circuito necesario para librería CapacitiveSensor. Adaptado de [24].	24
Figura 4.3: Representación del circuito para medir la capacitancia a través del MCU.	26
Figura 4.4: Esquema en “Eagle” de PCB con la primer opción de posición definitiva de los electrodos.	27
Figura 4.5: Esquema en “Eagle” de PCB con la segunda opción de posición definitiva de los electrodos.	28
Figura 5.1: Prueba de los modos en botones.	29
Figura 5.2: Diagrama de flujo de los pasos que sigue el modo 3 del teclado.	32
Figura 5.3: Diagrama de flujo de los pasos que sigue el programa completo del teclado.	33
Figura 5.4: Prueba de tacto con un electrodo.	35
Figura 5.5: Gráfico de datos medidos de un solo electrodo.	36
Figura 6.1: Prueba de tacto con seis electrodos.	37
Figura 6.2: Gráfico datos medición inicial de los seis electrodos.	38
Figura 6.3: Gráfico datos segunda medición de los seis electrodos.	39
Figura 6.4: Imagen de la placa PCB de los electrodos.	40
Figura 6.5: Medición en placa PCB cuando dos electrodos son tocados a través de plástico.	42

Figura 6.6: Medición en placa PCB con umbrales para cada electrodo..... 43

Figura 6.7: Medición en placa PCB con placa conectada a tierra..... 44

Figura 6.8: Vista modelo final del teclado..... 45

Figura 6.9: Modelo cubierta del teclado (vista ensamble) 47

Figura A.1: Vista tapa de la cubierta del teclado..... 53

Figura A.2: Vista base de la cubierta del teclado..... 54

Figura A.3: Vista lateral de base y tapa de la cubierta del teclado..... 54

Figura A.4: Vista ensamble base y tapa de la cubierta del teclado..... 55



Abreviaciones.

LED	: Light Emitting Diodes.
MCU	: Micro-controlador.
GND	: Ground o Tierra.
PCB	: Tarjeta de Circuito Impreso (Printed Circuit Board).

Unidades de Medida.

Ω	: ohm.
$K\Omega$: Kilo ohm.
V	: Volt.
ms	: milisegundo.



Capítulo 1. Introducción.

1.1. Introducción General.

Los sensores de proximidad tienen como función detectar objetos o señales que se encuentran cerca de ellos, dentro de los más utilizados están los sensores inductivos, capacitivos, fotoeléctricos e infrarrojos. Para el caso de los sensores capacitivos, estos utilizan un capacitor o condensador que es un dispositivo electrónico conformado por dos placas metálicas llamadas electrodos que almacenan energía eléctrica. En su construcción un capacitor puede tener los electrodos separados por aire o por un material aislante, también llamado dieléctrico [1]. El funcionamiento de los sensores capacitivos de proximidad se basa en un cambio de estado, el que consiste en una variación de su campo eléctrico. Estos sensores pueden detectar metales o no metales midiendo el cambio en la capacitancia, que depende a su vez del tamaño del objeto, masa, distancia a la que se encuentra y su constante dieléctrica [1].

En la actualidad el uso de sensores capacitivos tiene aplicaciones como la detección de niveles de llenado en recipientes, presencia de líquido en cajas, proximidad de objetos, inspección de cajas, medición de variables ambientales (temperatura y humedad relativa), entre otras. De esta forma estos sensores tienen usos en empresas alimenticias, de transporte, seguridad y tecnologías de uso cotidiano, etc. Estando presentes en áreas como las industrias, investigación, robótica y medicina. En este último ámbito, ha sido posible su utilización gracias a que el cuerpo humano, al estar constituido en gran parte por agua, posee una constante dieléctrica alta. Algunas de las aplicaciones que se pueden encontrar son: Dispositivos implantables para la medición de presión sanguínea [2], sensores de presión externos o en tecnología “wearable” para control de movimiento. De este modo se ayuda a personas, tanto para adquirir información necesaria para posteriores cirugías a las que deberán someterse, como en rehabilitación después de sufrir accidentes. Dentro de este último grupo existen personas que han sufrido daño cerebral o lesiones medulares completas o parciales que conducen, según el nivel del daño, a tetraplejia donde existe una parálisis que afecta a los brazos y piernas (incluyendo el tronco en los casos más graves), o paraplejia, en cuyo caso hay una parálisis afectando sólo a las extremidades inferiores (pudiendo haber algún grado de compromiso del tronco). Otro caso son las enfermedades musculoesqueléticas conducentes a temblores involuntarios o fatiga muscular. Todo esto lleva a limitaciones motoras que influyen negativamente en la forma que la persona tiene de relacionarse con el entorno, aún cuando en

muchas oportunidades sus capacidades mentales no se ven afectadas. Lo anterior lleva a la mayoría de los pacientes a un estado de depresión, inconformidad y agresividad.

Para mitigar y prevenir lo anterior se han enfocado los esfuerzos en rehabilitar la coordinación, fuerza o movilidad perdida, con el fin de recuperar al máximo estas habilidades. En este sentido, el uso de tecnología *“touch”* ha ido reemplazando otros métodos tradicionales al dar opciones más estimulantes a los pacientes y requerir un menor esfuerzo por parte de ellos, sin embargo, esta recuperación no siempre es total, lo que puede derivar en una sensación de aislamiento por parte del paciente. Aquí los computadores pueden ser excelentes medios para conectarse con el mundo, pero esta posibilidad se ve truncada si el paciente no puede utilizar un teclado o un *“mouse”* convencional.

En este trabajo se propone el diseño de un teclado táctil en base a sensores capacitivos que permita usar de una forma más fácil un computador para personas que presenten debilidad muscular o falta de coordinación fina. De este modo, se pretende apoyar en la rehabilitación y mejorar la calidad de vida de estos individuos, estimulando la realización de movimientos coordinados y hacer más accesible la interacción con el mundo a través del uso de un computador.



1.2. Objetivos.

1.2.1 Objetivo General.

Desarrollar, diseñar y fabricar un modelo de teclado capacitivo, con el fin de conectarlo a un computador, para ser utilizado por personas con limitaciones motoras.

1.2.2 Objetivos Específicos.

- Estudiar los sensores capacitivos “*touch*” y su aplicación en rehabilitación.
- Desarrollar un modelo de teclado capacitivo.
- Implementar opciones (o interpretaciones) de teclado que sean amigables con el usuario, por ejemplo, subir, bajar, presión de la tecla Enter. De forma que se le facilite la experiencia de uso del computador respecto al método que actualmente utiliza.
- Evaluar la funcionalidad del dispositivo desarrollado.

1.3. Alcances y Limitaciones.

En este trabajo se expone el diseño y construcción de un teclado capacitivo, abarcando además, el proceso de detección de contacto y selección de las opciones que se activarán al tocar una tecla. Lo anterior tiene por finalidad mejorar la experiencia de usuario de personas con limitaciones motoras al interactuar con un computador.

1.4. Metodología.

Primero se definen las posibles opciones de diseño del teclado (cantidad de botones, posicionamiento de ellos) junto con las posibles funciones que se puedan implementar, es decir, decidir qué debería hacer o interpretar el computador cuando se toca una tecla. Luego, se trabaja en el desarrollo de un código en Arduino (extensión de C) que permita implementar las opciones que anteriormente fueron escogidas. Posterior a esto, se explica el proceso de la detección de un “toque” por medio del uso de sensores capacitivos, completando lo necesario para la implementación del

teclado. El micro-controlador (MCU) seleccionado para realizar esta memoria fue el “Arduino Leonardo”.

En primera instancia se pretendía evaluar el desempeño del instrumento en pacientes con dificultades motoras, sin embargo, dadas dificultades presentadas en los servicios de salud no fue posible hacerlo, por lo que se evalúa en personas sanas. No obstante el fin último del autor es que se evalúe el teclado en los pacientes objetivo para lo que el dispositivo fue creado.

1.5. Temario.

- En el capítulo dos se presenta la revisión bibliográfica hecha respecto a los sensores capacitivos y su uso en la biomedicina, junto con la potencialidad en aplicaciones en pacientes con dificultad motora leve y moderada.
- En el capítulo tres se presentan las opciones de diseño que se contemplaron para el teclado, tanto para el número de teclas con el que este contaría como con las funciones que se elegirían para ser implementadas.
- En el capítulo cuatro se presentan el hardware que compone el teclado, considerando el MCU seleccionado (y las razones de ello), las partes que compondrían el circuito y dos opciones de implementación para la distribución de los electrodos.
- En el capítulo cinco se presenta el software desarrollado, se explica cómo se programaron los modos elegidos en el capítulo uno viendo las funciones utilizadas, cómo funciona el programa desde el punto de vista lógico y cómo se detectó el toque en un electrodo de cobre.
- En el capítulo seis se presentan las modificaciones que fueron necesarias para llegar a un teclado funcional con la cantidad de electrodos decidida y sin errores de activación.
- En el capítulo siete se presentan las conclusiones del trabajo expuesto, además de los trabajos futuros que el autor considera necesario realizar en el desarrollo del dispositivo.

Capítulo 2. Revisión Bibliográfica.

2.1. Introducción.

En esta sección se presenta la revisión bibliográfica del tema en el que se trabajó. Esto abarca el uso en medicina de los sensores capacitivos como detectores de contacto, así como técnicas y usos que se le ha dado a la tecnología “*touch*” en la rehabilitación de personas con daño cerebral, medular o musculoesqueléticas que deriven en una disminución considerable de la motricidad fina de las manos, la cual es necesaria para manipular un teclado de computador convencional.

2.2. Principios de Funcionamiento de Sensores Capacitivos.

Los sensores de proximidad capacitivos utilizan un campo eléctrico para polarizar el material que se les acerca. Este campo eléctrico provoca un movimiento de las cargas dentro de un material dieléctrico, producto de la acumulación de estas cargas el sensor puede detectar la proximidad de un objeto. Para el caso de los capacitores abiertos (que son utilizados en los sensores de proximidad), se establecen líneas de campo entre sus electrodos, las cuales polarizan el material a detectar [1].

Existen sensores de proximidad de uno o múltiples electrodos, para el caso de un electrodo, cuando se aproxima un material dieléctrico, se crea un campo eléctrico entre el electrodo cargado y el material polarizado, el cuál se asume está conectado a una tierra virtual, es así como ocurre una corriente de fuga a través del material, al ser detectada esta variación el sensor determina la proximidad del objeto. Para el caso de los electrodos múltiples, un electrodo se encuentra cargado y el otro (o los otros) se conectan a tierra, al acercarse un material dieléctrico las líneas de campo de los electrodos se ven perturbadas, pues ahora el electrodo cargado tiene otra salida a tierra, siendo la variación en la corriente que llega por el electrodo conectado a tierra la que se detecta para medir la proximidad del objeto [3]. En la Figura 2.1 se puede apreciar un ejemplo de detección a través de un electrodo simple y uno doble.

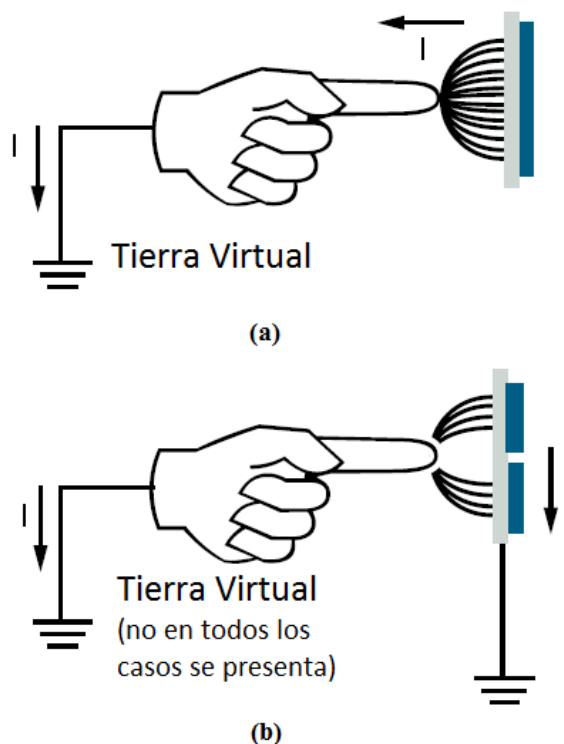


Figura 2.1: Detección de proximidad a través de un electrodo. Adaptado de [3].

(a) Detección con un electrodo simple; (b) Detección con un electrodo doble; Se observa la interacción de la mano con las líneas de campo y como ésta ofrece una vía de escape para una corriente de fuga.

Los materiales pueden polarizarse en distinta magnitud, dependiendo de su constante dieléctrica, entre más alta es la constante del material, más fácil se polariza siendo más fácil detectarlo por el sensor. La capacidad para detectar objetos puede verse afectada por el tamaño del sensor, un sensor grande posee un mayor rango de sensibilidad pero, por otro lado, es más susceptible al ruido, interferencia y campos eléctricos presentes en las cercanías, por ejemplo, de otros sensores capacitivos [1].

2.3. Avances en el Uso Biomédico de los Sensores Capacitivos.

En el área del uso de sensores capacitivos en medicina, los esfuerzos y avances se han enfocado mayoritariamente en ellos como sensores externos al cuerpo más que sensores implantables. No obstante se han realizado algunas investigaciones en esto último para la medición de variables fisiológicas como la presión sanguínea, de modo que se puedan obtener datos a distancia sin la necesidad de que el paciente esté conectado a algún aparato [2].

Los sensores capacitivos externos se utilizan principalmente como medio de control para el movimiento o fuerza realizada por extremidades del cuerpo, utilizándolos principalmente como sensores de proximidad y tacto. Como sensores de tacto el mayor uso está en el desarrollo de la robótica (permitiéndole a los robots interactuar con otras personas y su ambiente) [4], esto lo han logrado creando zonas del armazón que perciben el tacto y la presión, permitiéndoles controlar la fuerza ejercida. En la Figura 2.2 se muestra una imagen de la distribución de sensores capacitivos en la superficie de un brazo robótico para darle a este la capacidad de detectar cercanías de superficies o cuando realiza contacto con algún objeto conductivo.

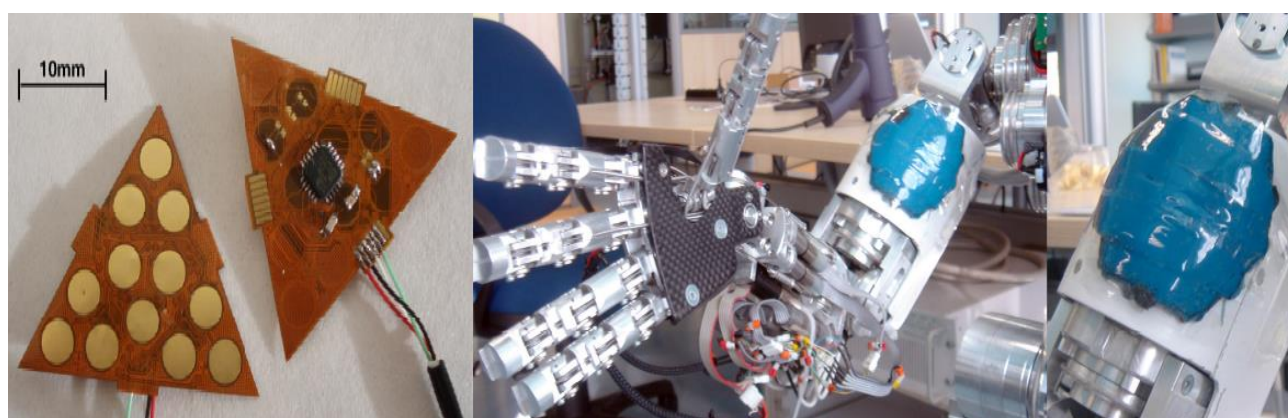


Figura 2.2: Aplicación de sensores capacitivos para tacto en brazo robótico. Adaptado de [4].

A la izquierda se muestra los sensores y cómo estos fueron distribuidos. A la derecha están puestos en el brazo robótico.

Para el caso del control de movimiento en humanos, se han usado estos sensores incorporados en tecnología “wearable” para controlar y monitorear la extensión y flexión de un brazo [5], además de la fuerza realizada. Para lograrlo el sensor mide la presión del cuerpo y la distribución de presión sobre la tela en la que se encuentra. Esto presentaría utilidad en la prevención de lesiones en el codo.

Como sensores de tacto, en medicina se han realizado aplicaciones para rehabilitación principalmente. Por ejemplo, como sensor de presión entre el pie y el calzado para un control post operatorio de pierna [6], de manera que se puede analizar la fuerza de los pasos del sujeto, y de este modo, darle indicaciones para una mejoría satisfactoria.

2.4. Médula Espinal.

La médula espinal es la estructura anatómica por la cual el encéfalo se comunica con el resto del cuerpo, se encuentra protegida por la columna vertebral y limita superiormente con el encéfalo por medio del bulbo raquídeo mientras que a nivel inferior lo hace en forma de cono a nivel de la primera o segunda vértebra lumbar. Su aspecto externo es blanquecino por estar cubierta de fibras nerviosas mielinizadas, mide aproximadamente 45 cm en la mayoría de los individuos y de ella se originan 31 pares nerviosos espinales, los cuales comunican sensorial y motoramente el cuerpo con el cerebro dependiendo si se conectan de forma posterior o anterior [7]. Cabe mencionar que la médula no es sólo un canal de comunicación entre el sistema nervioso central y el resto del cuerpo, la médula también es capaz de organizar respuestas por si sola a través de reflejos, los cuales sirven para evaluar los distintos segmentos medulares [8].

Cubriendo la médula espinal están las meninges, las cuales son membranas cuya función es proteger, dar soporte y mantener el ambiente adecuado para que la médula pueda funcionar con normalidad, las meninges están compuestas por la duramadre que es una membrana fuerte, densa y fibrosa, la aracnoides que es una membrana delgada e impermeable en donde está el espacio subaracnoideo el cual está lleno de líquido cefalorraquídeo que se encarga de brindar un ambiente óptimo a la médula, y la piamadre que es una capa única y delgada que está unida íntimamente a la médula espinal, su función es más bien vascular.

Estructuralmente, si se hace un corte transversal a la médula espinal se puede apreciar en la zona central una región con forma de H denominada sustancia gris, mientras que la región periférica de aspecto blanquecino se llama sustancia blanca (Figura 2.3). Las prolongaciones posteriores se llaman astas posteriores, mientras que las prolongaciones anteriores se llaman astas anteriores, estas últimas son más anchas y redondeadas [7].

Funcionalmente las astas posteriores son sensoriales (también se conocen como raíz dorsal) y están formadas por neuronas de axón corto, recibiendo las señales y sensaciones de las distintas partes del organismo que llegan a los ganglios espinales que son acumulaciones de neuronas que se encuentran de forma cercana y paralela a la médula. Mientras que las astas anteriores (también conocidas como raíz ventral) tienen funciones principalmente motoras, están compuestas por neuronas de gran tamaño, de forma estrellada y axón largo, estos van a controlar el movimiento y acción de distintas zonas del organismo ya sea por acciones reflejas de la médula o por un mandato enviado desde el sistema nervioso central [8].

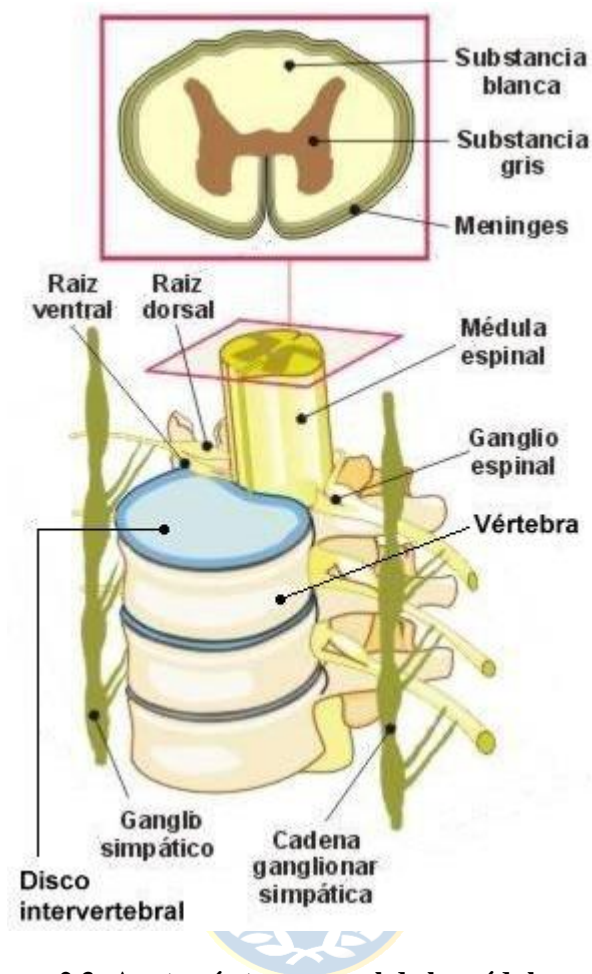


Figura 2.3: Anatomía transversal de la médula espinal

La sustancia blanca por otro lado se compone principalmente por fibras nerviosas mielínicas, estas fibras están encargadas de unir los segmentos medulares entre sí, y la médula espinal con el encéfalo. Existen tres tipos de organización de estas fibras en la médula espinal, la organización segmentaria que se encarga de las actividades reflejas coordinadas por la médula, la organización intersegmentaria que también se encarga de funciones reflejas pero conductuales como huir ante un estímulo de dolor y una organización suprasegmentaria en donde las actividades medulares son coordinadas por los centros nerviosos superiores [7].

Una lesión medular puede ocasionar la pérdida de la sensibilidad general y/o parálisis de los movimientos voluntarios en las zonas del cuerpo que se encuentren inervados por los nervios espinales que se originan de los segmentos bajo la lesión [8].

2.5. Lesión Medular y Dificultad Motora.

Una de las principales causas de dificultad motora adquirida durante la vida en la población adulta se encuentra las lesiones medulares. Estas se pueden definir como cualquier anomalía, ya sea por conmoción, contusión, laceración, compresión o sección, cuyo origen sea traumático o no traumático y que tenga como resultado un compromiso de la médula espinal, pudiendo ocasionar un mal funcionamiento de las funciones neurológicas que se encuentren debajo de la lesión en la columna, las funciones comprometidas pueden ser motoras, sensitivas y/o autonómicas [9].

Las secuelas de una lesión medular varían dependiendo de su ubicación, y nivel de compromiso de la medula, de este modo, existen dos formas de clasificar las lesiones medulares. La primera es por el nivel al que ocurre, la clasificación se basa en el último nivel de la médula que conserva todas sus funciones, así pueden existir pacientes con compromiso cervical, dorsal o lumbar, esto permite definir qué zonas del cuerpo o extremidades pueden verse comprometidas por el daño [10], entre más alta es la lesión mayor es el compromiso funcional. Los signos de la lesión medular deben evaluarse según las distintas sensibilidades ya sea táctil, dolorosa, vibratoria o térmica, además del control autonómico que son los esfínteres y la temperatura corporal. Dentro de los síntomas se encuentran espasmos en las extremidades, atrofia, debilidad flácida, arreflexia que es ausencia de reflejos ante estímulos y anestesia general o localizada, lo que conlleva a un mal control y percepción del movimiento [11].

Otra forma de clasificación es por el grado de daño bajo la lesión, el daño puede ser completo, es decir, no se presenta ninguna función neurológica bajo la lesión (no hay sensibilidad, control motor ni autonómico) o incompleto, en donde se conserva una o más funciones neurológicas de manera parcial o completa, este grado de conservación puede presentarse en un hemisferio o en ambos a estos pacientes se les denomina tetraparésicos o paraparésicos dependiendo si la lesión involucra a las extremidades superiores e inferiores o solamente a estas últimas [10]. La escala más usada internacionalmente para medir el grado de compromiso de la médula producto de una lesión es proveniente de la Asociación Americana de Lesión Medular (ASIA por sus siglas en inglés), la cual establece cinco grados diferentes de acuerdo a la ausencia o preservación de las funciones motoras y sensitivas [12], la condición de los distintos grados se muestra en la tabla 2.1.

Internacionalmente en las investigaciones de lesiones medulares se registran lesiones a nivel cervical de un 34 al 53% de los casos, las lesiones a nivel torácico ocupan entre un 43 al 48% y las lumbares son de un 10 a 14% de casos [13] [14]. Por otra parte, según esta escala existe un mayor

número de casos de grado A alcanzando entre un 48 a 58% de los casos, mientras que las lesiones de grado B y C alcanzan una proporción del 16 al 44% de los casos, quedando un segmento menor de casos para los grados D y E [15] [16]. Es justamente dentro de los casos que presentan un grado de compromiso C o menor donde es vital la rehabilitación, pues en ese nivel existe cierto grado de sensibilidad y fuerza en la zona bajo la lesión aunque la capacidad motora es muy débil.

Tabla 1.1 Escala de clasificación de la lesión medular.

Grado	Característica
Lesión completa A	Ausencia de función motora y sensitiva que se extiende hasta los segmentos sacros S4-S5
Lesión incompleta B	Preservación de la función sensitiva por debajo del nivel neurológico de la lesión, que se extiende hasta los segmentos sacros S4-S5 y con ausencia de función motora.
Lesión incompleta C	Preservación de la función motora por debajo del nivel neurológico, y más de la mitad de los músculos llave por debajo del nivel neurológico tienen un balance muscular menor de 3.
Lesión incompleta D	Preservación de la función motora por debajo del nivel neurológico, y más de la mitad de los músculos llave por debajo del nivel neurológico tienen un balance muscular de 3 o más.
Normal E	Las funciones sensitiva y motora son normales.

Un aspecto importante en los pacientes que sufren lesiones medulares son las características sociodemográficas que poseen. Por ejemplo, la relación hombre/mujer es 4:1 siendo los hombres un 66 a 92% del total de casos a nivel internacional. Por otra parte, las lesiones medulares son sufridas en su mayoría por adultos jóvenes con edades que oscilan entre los 16 y 35 años [13] [15]. Dentro de las causas más frecuentes de lesión medular están los traumatismos con un 60% de los casos en países desarrollados [9] cuyas causas son en su mayoría heridas por armas blancas o de fuego (del 20 al 60% de los casos) o accidentes de tránsito (del 12 al 63%) [17]. Estas causales hacen que las personas tengan fuertes repercusiones no solo físicas, sino que psicológicas y sociales, puesto a que en su mayoría son individuos jóvenes e independientes. Esto puede llevar a casos de depresión, negación, aislamiento y abuso de drogas por parte de los pacientes quienes ven un cambio rotundo

en su vida diaria, es por esto que se buscan acciones y herramientas que promuevan la autonomía personal e inclusión, fomentando la aceptación plena de la persona.

2.6. Usos de Sensores Capacitivos en Rehabilitación de Pacientes con Dificultad Motora.

En el caso de la rehabilitación de personas con daño cerebral o medular parcial, la tecnología “*touch*” es bastante utilizada para los ejercicios que se deben realizar en la recuperación de la movilidad [18], combatiendo de esta forma los problemas emocionales que enfrenta la persona producto de los cambios vividos por el accidente sufrido.

Dentro de las problemáticas a las que se enfrentan tanto los especialistas como los pacientes, y sus familiares, es que luego de sufrir un accidente limitante como los anteriormente mencionados, se crean diferencias en las expectativas de recuperación, las cuales se hacen más grandes con el tiempo [19]. La impaciencia por parte del paciente lo lleva a desmotivarse por lo repetitivo de los ejercicios y los pocos avances que nota en su recuperación versus los que el espera tener. Esto provoca que no cumpla con las metas impuestas por el especialista que lo trata, pues el paciente deja de realizar los ejercicios en el mediano a largo plazo [20], haciéndose necesaria una logística diferente para tratarlos, de modo que las diferencias en expectativas no sean tan grandes entre los pacientes y los profesionales que están a cargo de su terapia.

En base a lo anterior es que en la rehabilitación de personas con daño cerebral, medular parcial, o problemas musculo-esqueléticos que resulten en dificultad para realizar movimientos coordinados en distinto grado, se utilizan sensores capacitivos a través de pantallas “*touch*” para realizar estimulación mediante actividades didácticas que llamen la atención del paciente. Esto se realiza para evitar que los pacientes con estos problemas caigan fácilmente en depresión y abandonen los ejercicios dados por los especialistas por considerarlos demasiado repetitivos y poco estimulantes para ellos. Es por esto que los ejercicios son “disfrazados” con juegos u otras actividades interactivas que motiven al paciente a participar día a día [18], en la Figura 2.4 se muestra un juego elaborado específicamente para la rehabilitación de personas con discapacidad motora producto de un accidente. Esto ha llevado a mejores resultados de coordinación al comparar a pacientes que han utilizado métodos más tradicionales como terapia [21].



Figura 2.4: Juego creado para Rehabilitación de pacientes con dificultad motora. Adaptado de [19].

A la izquierda se ve una imagen del juego. A la derecha un paciente lo está utilizando en su rehabilitación.

Para acompañar la rehabilitación e inclusión actualmente existen hardware como los teclados de conceptos, estos están diseñados para personas con discapacidad motora o cognitiva y busca ayudarles a comunicarse con los demás, aprender y poder utilizar un computador. Por ejemplo, los teclados “*Intellikeys USB Keyboard*” y “*Helpikeys*”, que se muestran en la Figura 2.5, están diseñados con este propósito. Son táctiles y tienen variadas funciones como las letras del alfabeto, los números, flechas direccionales, control del “*mouse*” etc. sin embargo, su costo es elevado (actualmente bordean los \$600.000 pesos cada uno), no siendo asequible para la mayoría de las personas, más bien son comprados por instituciones que se dedican a la rehabilitación contando con unos pocos ejemplares, los que son destinados principalmente a los niños y sólo por el tiempo que tienen de terapia, no pudiendo usarlos en su vida diaria.



Figura 2.5: Teclados de Concepto.

(a) “*IntelliKeys USB Keyboard*”; (b) “*Helpikeys*”

2.7. **Discusión.**

La rehabilitación es prioritaria cuando los pacientes han sufrido una lesión que les impida tener motricidad fina, con el propósito de que las secuelas sean reducidas en la medida de lo posible. Sin embargo, existen casos donde el grado de control del movimiento es muy pobre incluso con terapia. A pesar de esto, muchos de estos pacientes conservan sus facultades mentales, pero debido a las limitaciones corporales con las que quedan no pueden relacionarse demasiado con el mundo exterior.

Se explica cómo es que en estos pacientes se generan sensaciones de depresión y agresividad producto de las frustraciones que viven debido a que la recuperación no cumple con sus expectativas en el tiempo, lo cual se traduce en un menor grado de recuperación a causa del abandono de los ejercicios.

Las aplicaciones en dispositivos “*touch*” ayudan a los pacientes adultos a interactuar de forma más fluida con dispositivos electrónicos lo cual es muy positivo para su autoestima y rehabilitación. No obstante, esta interacción está limitada al programa o “juego” desarrollado para los ejercicios que deben hacer en su terapia, y no así para darle otros usos al computador como estudiar, navegar, incluso buscar otros juegos, etc.

Los pacientes con discapacidad motora leve (ya sea temporal o permanente) no pueden hacer uso de un “*mouse*” o un computador convencional, pues carecen de la motricidad fina necesaria o de la capacidad para presionar botones. Es por esta razón que la creación de un teclado especialmente diseñado para reaccionar al tacto, que no posea “teclas” pequeñas, y a su vez no sea muy grande, resultaría útil para estimular la realización de movimientos coordinados e incrementar las posibilidades de interacción del paciente con el computador, si bien esta opción existe en el mercado, su público objetivo son niños y el costo de adquisición es elevado, por lo que se puede realizar una opción más enfocada al público adulto y/o de menor costo volviéndolo una opción alcanzable para un mayor número de personas y no solo instituciones.

Capítulo 3. Opciones de Diseño.

3.1. Introducción.

En este capítulo se presenta las distintas opciones de diseño funcional que se consideraron para la creación del dispositivo. La idea es poder elegir un modelo que sea funcional y sencillo a la vez. Se evalúan tanto cantidad de teclas con las que se contarán como la funcionalidad que estas tendrán.

3.2. Esquema General.

El circuito a implementar debe cumplir con ser confiable y sencillo de utilizar, es por esto que se utilizó la menor cantidad de piezas posibles. En el diagrama de bloques de la Figura 3.1 se muestra el esquema general del circuito, consistente en un número determinado de placas que actúan como electrodos, estos están conectados a una unidad controladora que mide la capacitancia de cada electrodo e interpreta la acción que se quiera realizar en el computador, finalmente el MCU se conecta a un computador realizando la acción. Cabe señalar que al conectarse el teclado con el computador vía USB, el MCU no necesita tener independencia energética (es decir, ser alimentado por una batería), ya que puede obtener energía de dicha conexión.

Las etapas del proceso son las siguientes: La diferencia en la capacitancia que experimenta al acercarse un objeto dieléctrico al teclado es capturada por los electrodos, esto debe hacerse mediante sensores (aunque más adelante se decidió que el MCU actuara como sensor capacitivo), luego, el MCU decide cual es la acción que se desea realizar dependiendo del electrodo que haya sido tocado, enviando una orden al computador vía USB, la que es interpretada por este último como una orden ejercida por un teclado convencional. El MCU utilizado fue un Arduino Leonardo, el que fue programado con la capacidad de emular un teclado o un “*mouse*” al conectarse a un computador. Lo anterior hace que no sea necesario implementar una interfaz en el computador que interprete los datos enviados por el MCU, sólo se necesita programarlo adecuadamente. Se hablará más en detalle cuando se explique la implementación de las funciones del teclado.

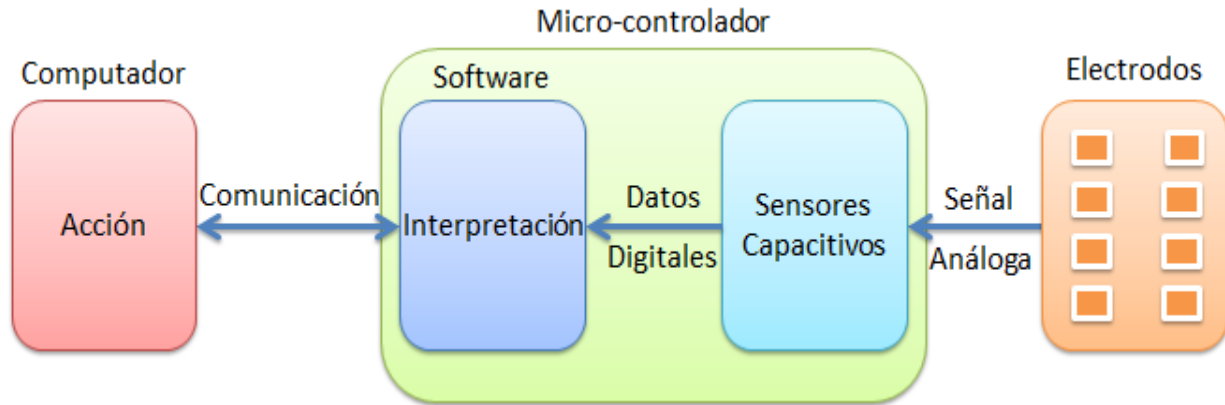


Figura 3.1: Esquema General del Hardware a Diseñar.

3.3. Determinación de Funciones y Cantidad de Electroodos.

Los factores que influyeron en la cantidad de electroodos posibles fueron los siguientes: Por un lado el teclado debía cumplir con ejecutar al menos las operaciones básicas que pueden hacerse con un teclado convencional, de manera que sea suficiente para que el usuario pueda interactuar con el computador. Por otra parte, al estar orientado para personas con dificultad motora, no debe poseer demasiadas teclas, ya que eso incrementaría el tamaño significativamente, pues, las teclas no pueden ser pequeñas por tratarse de pacientes con dificultad para realizar movimientos finos. Estos dos factores son limitantes entre sí, pues a menor cantidad de teclas menor cantidad de opciones disponibles para implementar.

El protocolo a seguir para determinar las funciones y cantidad de electroodos se basó en los siguientes pasos. Primero se propuso una cantidad de teclas a implementar eligiendo entre las del teclado del computador. Luego se pidió a otros compañeros que usasen sus computadores sólo con las teclas que habían sido seleccionadas, posteriormente se recogían sus impresiones respecto a lo que se podía mejorar o cambiar según las acciones que pudieron realizar y cuales no pudieron o se les dificultó más, de manera que se fueron escogiendo las teclas más usadas y descartando las menos usadas. En la Figura 3.2 se observa la metodología utilizada para elegir cuales teclas eran consideradas necesarias para la implementación del teclado.

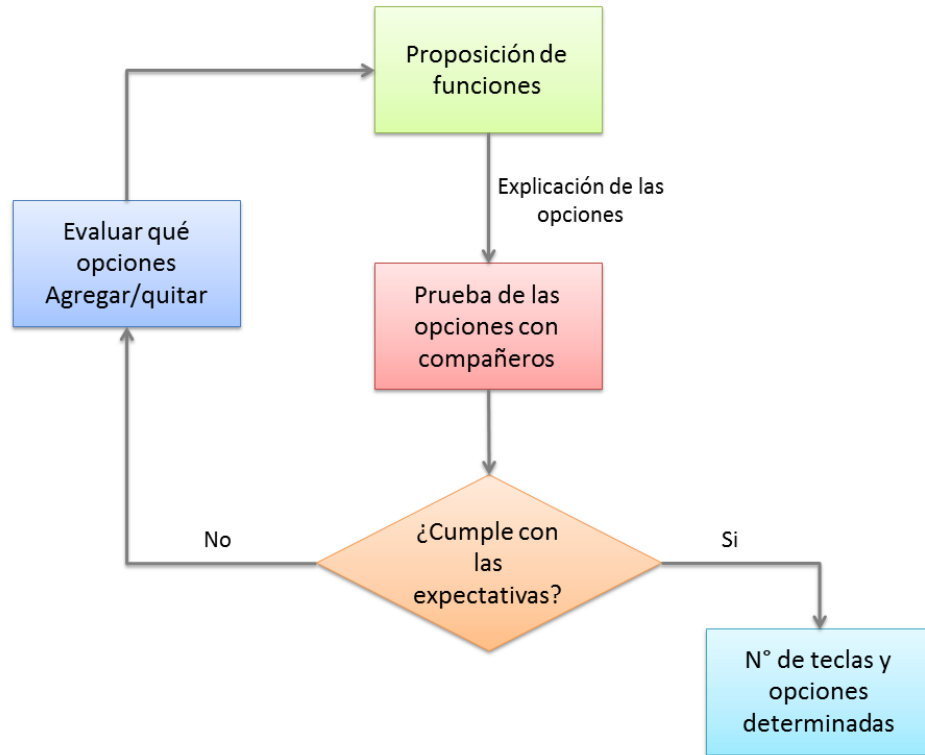


Figura 3.2: Diagrama de flujo con la metodología utilizada para definir opciones y n° de teclas necesarias.

El primer modelo propuesto constó de siete teclas (que posteriormente serían electrodos), En la Figura 3.3 se muestra una idea preconcebida de las teclas con su función.

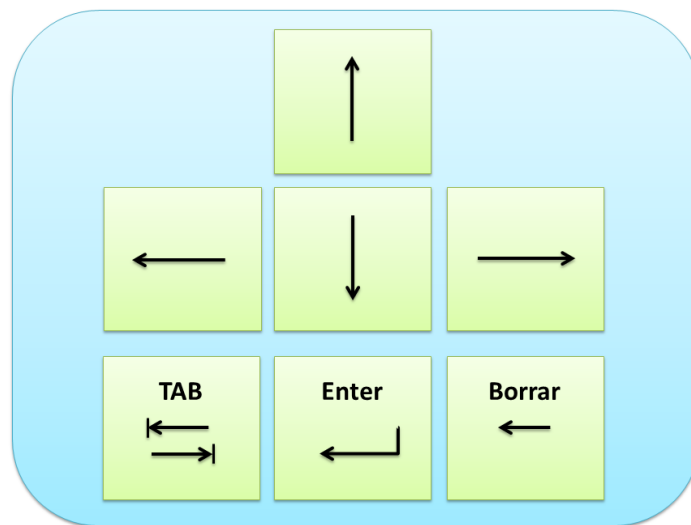


Figura 3.3: Funciones propuestas en el primer modelo para la implementación del teclado.

Las funciones elegidas fueron: las cuatro flechas direccionales, la tecla “*Enter*”, la tecla Tab y la tecla Borrar, de modo que los usuarios podían moverse en el escritorio del computador y seleccionar algún icono para ingresar en algún archivo o programa, además, al ingresar a internet existía la posibilidad de moverse dentro de los marcadores pre-establecidos en el navegador para cargar las páginas, también se podía volver atrás en las páginas de internet y estando en ellas, el usuario podía subir, bajar y seleccionar elementos o links. Sin embargo, la retroalimentación indicó que era necesaria la implementación del abecedario, de manera que se pudiese escribir con el teclado, además mencionaron que, si bien, con la tecla Tab podían acceder a los links, resultaba muy tedioso tener que presionar repetidas veces la tecla hasta seleccionar lo que deseaban, y que no podían cerrar los programas una vez que los abrían. De esta manera, el modelo de teclado cumpliría con una cantidad de electrodos reducida, sin embargo, la funcionalidad sería reducida y poco cómoda.

Considerando las impresiones recogidas del primer modelo propuesto, se pensó en utilizar la opción que traen los computadores de teclado en pantalla para poder escribir (con lo cual se podría acceder a todas las teclas del computador), pero para ello sería necesario el uso del “*mouse*” por parte del usuario, ya que la selección de las teclas en pantalla se hace a través del puntero del ordenador. De esta forma, el nuevo modelo propuesto constaba de 11 teclas, donde 9 de ellas tenían una doble función dependiendo de lo que se deseaba hacer. Las dos teclas restantes tendrían funciones fijas: mientras una se encargaría de desplegar el teclado en pantalla del computador (con un solo toque el teclado aparece en la pantalla del computador) la otra tecla cumpliría la función de cambiar el periférico simulado (de “*mouse*” a teclado y viceversa). Para una mejor comprensión en la Figura 3.4 se muestra la idea del teclado.

Las 9 teclas restantes tendrían sus funciones condicionadas a la tecla que controla el teclado en pantalla, pues tienen funciones que alternan de “*mouse*” a teclado. Mientras no esté activado el teclado en pantalla, cuatro de ellas se encargan de simular las flechas direccionales del teclado (flecha arriba, flecha abajo, flecha izquierda y flecha derecha), otras cuatro teclas adyacentes a las primeras simularían las teclas Borrar, Espacio, Tab y ALT+F4 lo que permite cerrar el programa que esté abierto en la pantalla. Por último, está la tecla “*Enter*”. Permitiendo de esta manera las mismas opciones que en el modelo anterior pero esta vez dando la opción de cerrar los programas abiertos.

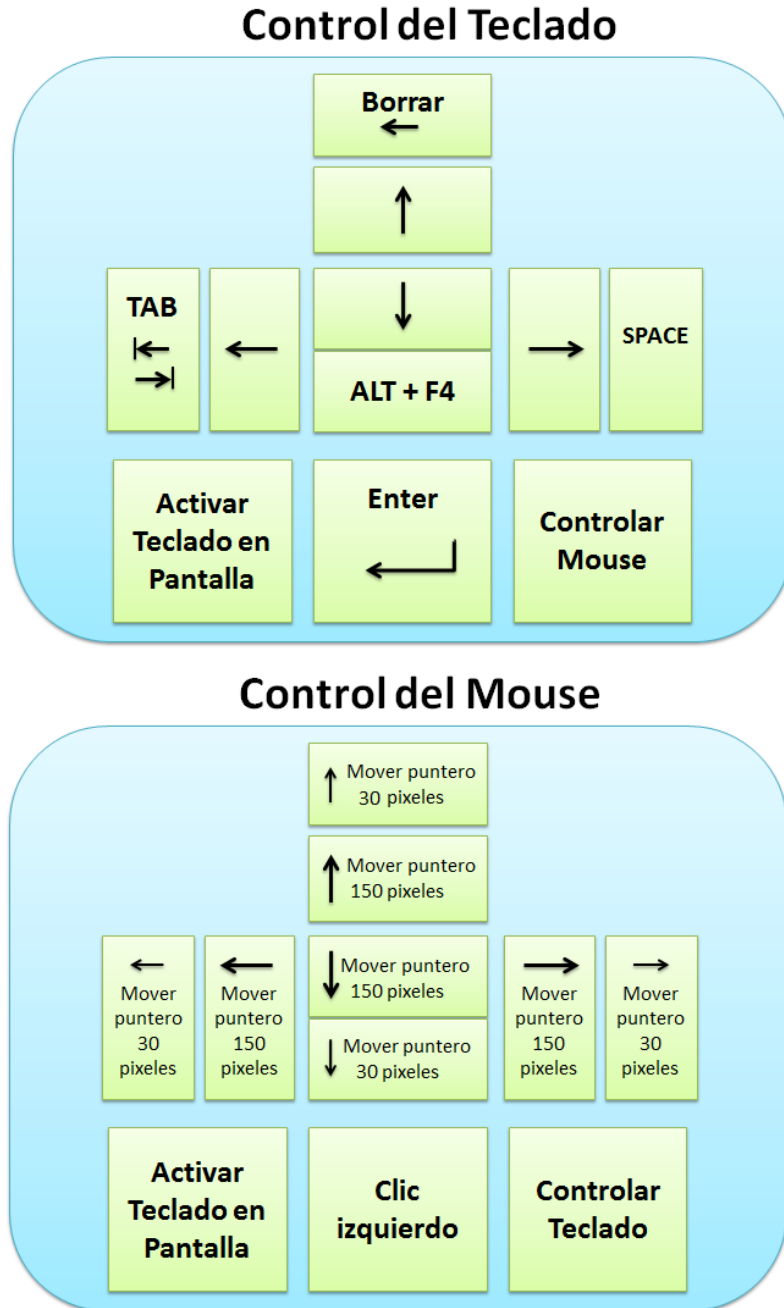


Figura 3.4: Funciones propuestas en el segundo modelo para la implementación del teclado.

Arriba están las funciones que cumplirían los electrodos controlando el teclado; Abajo las funciones que cumplirían los electrodos controlando el “*mouse*”.

Si se activa el teclado en pantalla se pasaría a controlar el “*mouse*” del computador con el teclado, en un comienzo se pensó en que el “*mouse*” debería moverse en espacios fijos, de esta manera las teclas que antes simulaban las flechas direccionales, ahora se encargarían de mover el

puntero del “*mouse*” un número determinado de píxeles por toque en las mismas direcciones que tenían asignadas las teclas para las flechas del teclado (150 píxeles aproximadamente por cada toque, esta cantidad fue escogida de forma arbitraria, aunque se pensaba ajustar a un valor luego de hacer pruebas), de manera que con unos pocos toques se llegase de un lado al otro de la pantalla, mientras que las teclas adyacentes se encargarían de mover igualmente el puntero, pero en un rango menor (30 píxeles), así se podría llegar a las zonas que se encuentran dentro de un rango menor. Finalmente, la que antes simulaba la tecla “*Enter*”, ahora simularía el clic izquierdo del “*mouse*”, permitiendo seleccionar los distintos objetos de la pantalla (entre ellas las letras del teclado desplegado). La finalidad de esto era poder desplazarse de forma rápida dentro del teclado en pantalla y así poder escribir.

El resultado obtenido al preguntarle a compañeros luego que usasen sus computadores con estas condiciones fue que habían mejorado su experiencia en el acceso a las opciones y cerrar ventanas, sin embargo, para pacientes que tienen dificultades motoras para realizar movimientos finos, el tener muchos botones (y hacerlos adyacentes) consideraron que podría resultar molesto, ya que incrementa la probabilidad de tocar un botón por accidente, además, mencionaron que el desplazamiento del “*mouse*” les parecía demasiado “brusco” siendo mejor darle un movimiento más fluido. De esta manera, el segundo modelo mejoró en términos de facilidad con el usuario, pero usa más teclas de las que se consideran necesarias y al observar el modelo se consideraron demasiado cercanas.

Finalmente se decidió hacer un modelo de teclado con seis teclas y tres “modos”, donde cinco de las teclas simularían el teclado o el “*mouse*” y la sexta se encargaría de cambiar el “modo” al ser tocada, en esta ocasión se incorporó el modo de escritura en el teclado, de manera que se pueda escribir sin tener necesariamente el teclado en pantalla activado, en la Figura 3.5 se muestra de mejor manera el modelo propuesto.

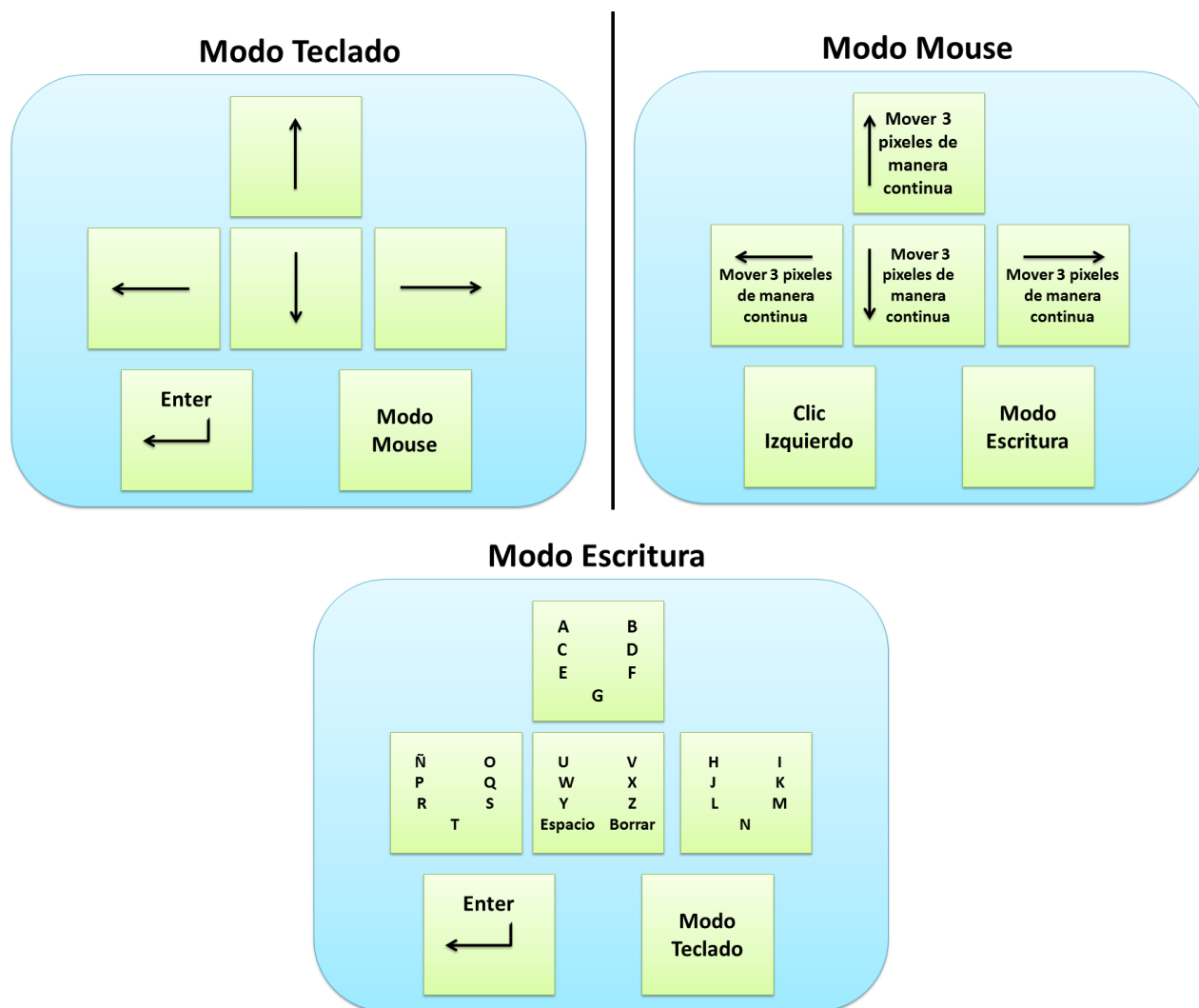


Figura 3.5: Funciones propuestas en el tercer modelo para la implementación del teclado.

A la izquierda las funciones de los electrodos controlando el teclado, a la derecha las funciones de los electrodos controlando el “*mouse*”, abajo las funciones de los electrodos en el modo de escritura.

El primer modo consiste en las cuatro flechas direccionales del teclado más la tecla Enter, y está pensado para que el usuario pueda seleccionar iconos del escritorio, y moverse dentro de una ventana o página de internet.

El segundo modo toma el control del “*mouse*” del computador, siendo las teclas direccionales responsables ahora de mover el puntero del “*mouse*” (en las mismas direcciones a las que apuntaban en el modo anterior), la tecla Enter ahora simula el clic izquierdo del “*mouse*”. En esta ocasión el movimiento del “*mouse*” se hace en un rango menor dándole un movimiento más fluido al puntero en la pantalla, el movimiento del puntero se basa en la siguiente idea: mientras la

tecla esté siendo tocada, el puntero se mueve en esa dirección a una velocidad determinada. Esto hace posible seleccionar iconos y opciones que antes no eran posibles o resultaba difícil acertarles, de manera que se pueden abrir y cerrar ventanas, además de seleccionar opciones dentro de las aplicaciones sólo con el “*mouse*”.

El tercer modo es el encargado de la escritura, este se basa en la forma de escribir de algunos modelos de teléfonos celulares, en donde al escribir un mensaje se presiona una tecla, el teléfono deja un espacio de tiempo para que vuelva a ser presionada, si se presiona dentro de ese lapso la letra que aparecía es borrada y remplazada por la siguiente en el alfabeto, si no se presiona la tecla dentro del lapso de tiempo, la letra que aparece se deja fija y se avanza al siguiente espacio para escribir una nueva letra. Si bien el principio es el mismo, por razones prácticas se prefirió no hacer que la tecla sea presionada muchas veces para cambiar de letra, sino que mientras la tecla es tocada la letra mostrada avanza y cuando se deja de tocar, se fija la letra, pasando al siguiente espacio. Para realizar esto, se distribuyó el abecedario en las cuatro teclas direccionales, dejando además las teclas Espacio y Borrar añadidas a esta lista, de manera que se puedan escribir oraciones o borrar en caso de cometer algún error. La tecla que en el modo anterior era el clic del “*mouse*”, vuelve a ser la tecla “*Enter*” en este modo.

Como consecuencia de los tres modos, se puede seleccionar archivos, navegar en internet, escribir y buscar aplicaciones en el computador (siendo posible buscar el teclado en pantalla si eso le parece más cómodo al usuario). Este modelo propuesto fue el que tuvo mejor aceptación al momento de pedir a compañeros que probasen utilizarlo, pudiendo realizar distintas tareas en el computador con un número limitado de teclas.

Capítulo 4. Hardware del Teclado.

4.1. Introducción.

En este capítulo se presentan los componentes que tiene el circuito para la implementación del teclado capacitivo, entre ellas la elección del MCU y las ventajas que posee comparado con otras opciones, además se mencionan los elementos necesarios para la detección táctil y la disposición de los electrodos (cuya cantidad fue elegida en el capítulo anterior).

4.2. Selección del Micro-Controlador (MCU).

Para la realización de este trabajo se utilizó un MCU Arduino modelo Leonardo (Figura 4.1), que es un MCU basado en el modelo ATmega32u4, pero a diferencia de otros MCUs basados en este modelo, el ATmega32u4 del Arduino Leonardo tiene incorporado en la comunicación USB la capacidad de emular un periférico (eliminando la necesidad de un procesador secundario para realizar esta operación), lo que tiene como consecuencia que este MCU puede aparecer conectado para el ordenador como un “mouse” o un teclado [22].

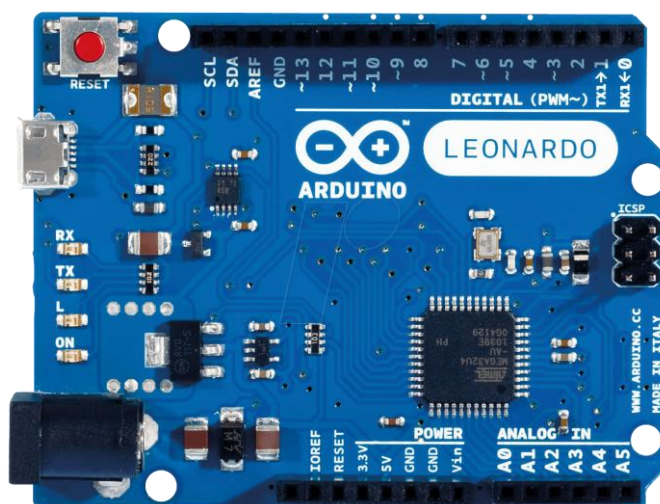


Figura 4.1: Imagen de MCU utilizado (Arduino Leonardo).

El Arduino Leonardo tiene 20 pines digitales de entrada/salida, de los cuales 7 pueden usarse para salidas de tipo PWM y 12 como entradas analógicas. Para los fines del proyecto se utilizaron 12 pines para los electrodos y 4 pines que se conectaron a LEDs para poder tener una retroalimentación visual de lo que estaba haciendo el teclado. Además, para este MCU están disponibles las librerías Keyboard.h y Mouse.h que permiten al computador reconocer al MCU como los respectivos periféricos cuando se conecta por USB [23].

4.3. Micro-Controlador (MCU) Arduino Leonardo como Sensor Capacitivo.

Para realizar la medición de proximidad y así detectar cuando se toca un electrodo, se utilizó el mismo MCU como un sensor capacitivo. Para ello se descargó la librería “CapacitiveSensor” de la página de Arduino. Esta librería da la opción de transformar un par de pines del MCU en un sensor capacitivo, permitiendo detectar la capacitancia eléctrica del cuerpo humano [24]. Los materiales necesarios para que el sensor funcione son: un medio de alto valor resistivo, y un trozo de papel aluminio (o lámina de cobre en este caso) conectados por el extremo de un cable. En la Figura 4.2 se observa un esquema de la construcción del circuito, indicando la capacitancia detectada.

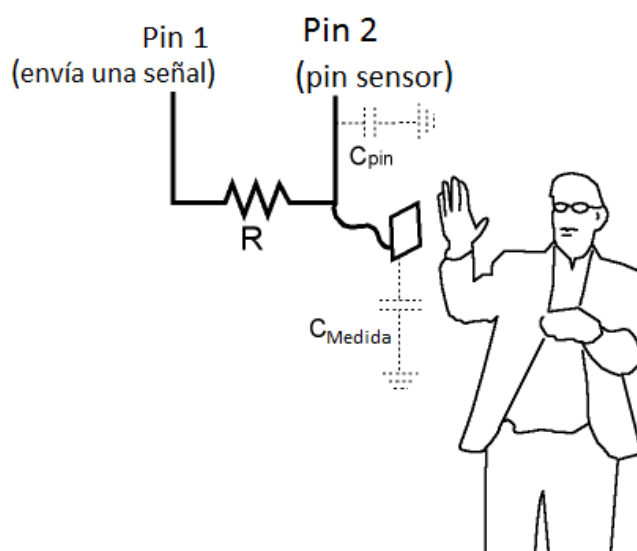


Figura 4.2: Esquema de circuito necesario para librería CapacitiveSensor. Adaptado de [24].

R, Resistencia que puede variar desde 1 a 50 mega ohm. Cmedida, Capacitancia que se medirá en el Pin 2. Cpin, capacitor optativo para obtener lecturas más estables, puede variar de 10 a 400 picofaradios.

De acuerdo a la descripción de la librería, el MCU puede llegar a detectar la capacitancia humana a través de más de un cuarto de pulgada de plástico, madera o cerámica, pudiendo ubicar al electrodo en alguna zona oculta o protegida.

El funcionamiento del MCU como sensor capacitivo se basa en el siguiente método: La librería cambia dos pines del MCU (los que serán llamados pin1 y pin2), uno de ellos será el encargado de enviar una señal, la cual será esperada por el otro pin, es decir, cuando el pin1 cambia de estado el MCU espera a que el pin2 reciba esa señal y cambie al mismo estado que el pin1 le ha enviado. El MCU cuenta el tiempo que toma realizar esta operación y lo va guardando en una variable que va incrementando, Luego el MCU recibe el valor final de esta variable, que es con la cual trabaja.

Cuando el pin1 cambia de estado, con el tiempo cambia el estado del pin2. El retraso entre el cambio del pin1 y la respuesta del pin2 es determinado por una constante de tiempo RC definida por $R * C$, donde R es el valor de la resistencia y C es la capacitancia presente entre ambos pines por el electrodo, además de cualquier otra capacitancia (por ejemplo, la capacitancia producto de la interacción del cuerpo humano) [25]. Si se adiciona un pequeño condensador (entre 20 a 400 pF) en paralelo con la capacitancia del cuerpo, las lecturas detectadas se vuelven más estables, lo que es altamente deseable para evitar errores en las lecturas [24].

Es así como los elementos del circuito del teclado para los electrodos son 6 circuitos RC (uno para cada tecla), donde en cada uno se utiliza una resistencia de 4.7 mega ohm y un condensador de 10 picofaradios. Esto se conecta a dos pines del MCU, que actúan como sensor capacitivo. En la Figura 4.3 observa una representación del circuito para los seis electrodos.

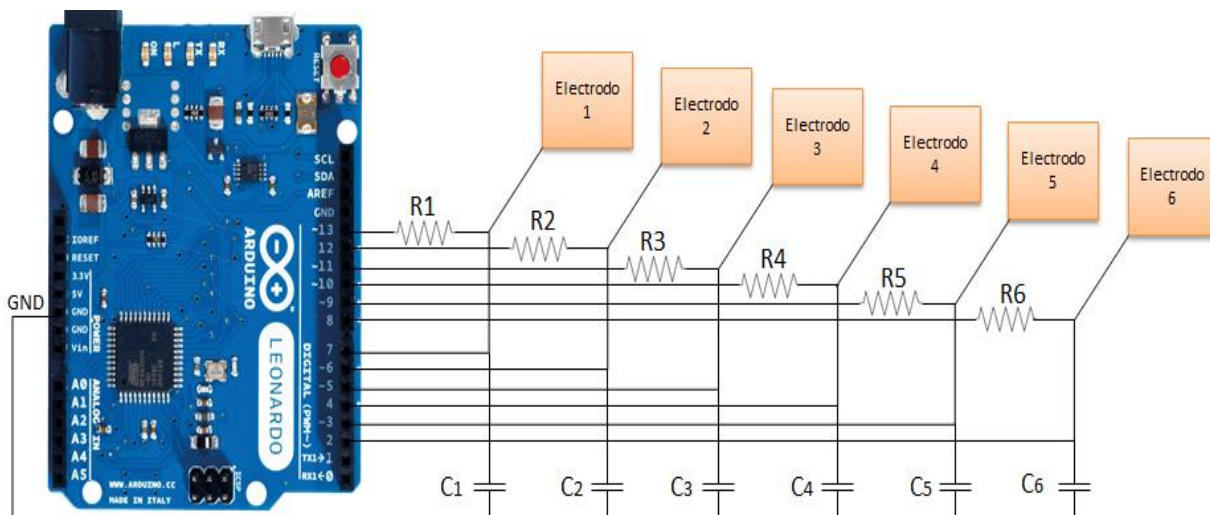


Figura 4.3: Representación del circuito para medir la capacitancia a través del MCU.

$R1 = R2 = R3 = R4 = R5 = R6 = 4.7$ mega ohm; $C1 = C2 = C3 = C4 = C5 = C6 = 10$ picofaradios.

4.4. Modelo del Teclado.

Una vez decidido el número de teclas (o electrodos) que tendría el teclado se propusieron dos diseños preliminares de hardware para implementarlo, la diferencia entre estos diseños radica solamente en la organización de los electrodos.

En el primer modelo los electrodos están organizados siguiendo una disposición más intuitiva para las funciones de cada sensor, es decir, los electrodos encargados de las flechas direccionales están posicionados en la misma orientación de sus flechas, con el electrodo que representa la tecla Enter al centro. Mientras que el electrodo para el cambio de modo se encuentra ubicado en una esquina. En la Figura 4.4 se muestra el esquemático de “Eagle” con la disposición de los electrodos en el teclado, junto con la salida que tendrían estos electrodos para conectarse a los otros componentes del Hardware. La desventaja de este modelo es que al querer presionar los electrodos que estén más alejados al usuario se debe extender la mano sobre los electrodos más cercanos lo que aumenta el ruido en su detección, pudiendo activarse electrodos que no sean tocados.

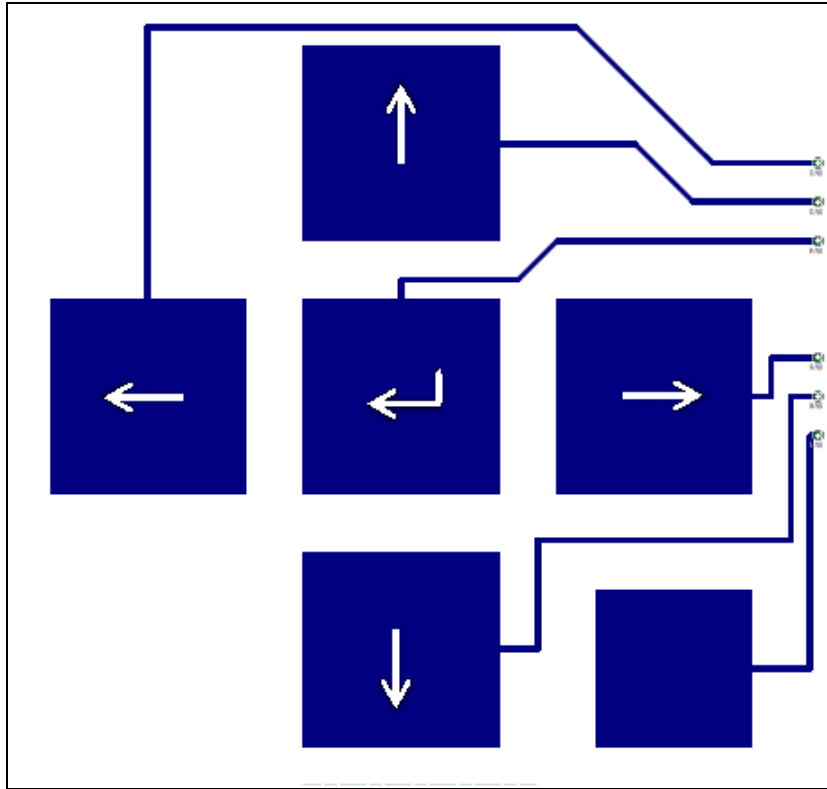


Figura 4.4: Esquema en “Eagle” de PCB con la primera opción de posición definitiva de los electrodos.

En el segundo modelo propuesto, los electrodos están posicionados de manera que se intenta que ningún electrodo se interponga en la ruta que seguiría la mano para poder activarlos, así los electrodos que representan las direcciones izquierda y derecha quedan en esas mismas ubicaciones, sin embargo los que representan las direcciones de arriba y abajo quedan a la misma altura desde el punto de vista del usuario sólo separados por el electrodo que representa la tecla Enter. En la Figura 4.5 se aprecia de mejor manera la disposición de los distintos electrodos en el esquemático de “Eagle”, junto con la salida que tendrían los electrodos para conectarse al resto del circuito. La ventaja de este modelo es que se disminuye el ruido al tocar un electrodo, la desventaja es que la posición de los electrodos deja de ser tan intuitiva para su uso, además de incrementar el ancho del teclado.

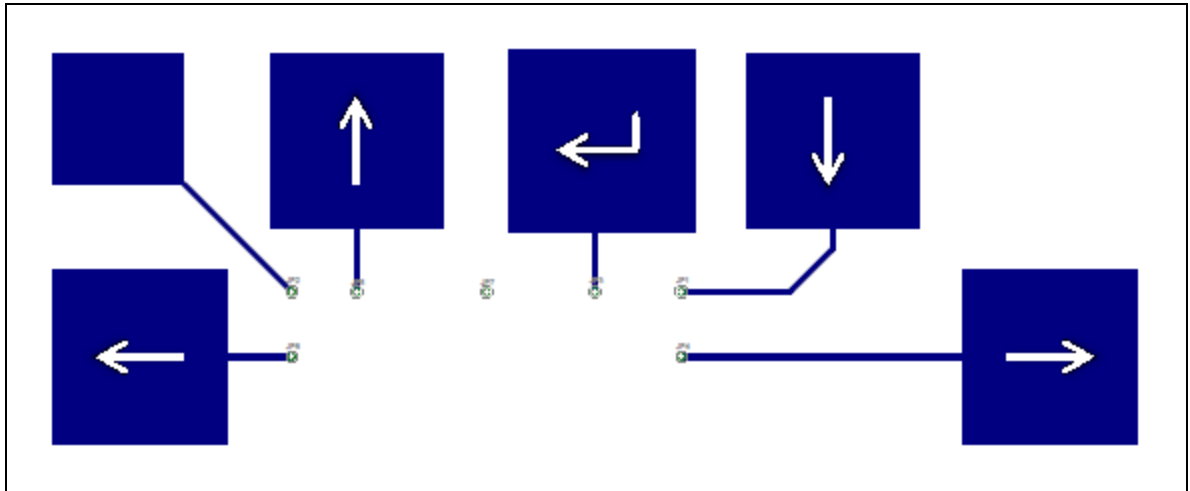


Figura 4.5: Esquema en “Eagle” de PCB con la segunda opción de posición definitiva de los electrodos.

Finalmente se optó por el primer modelo para la distribución de las teclas, pues se prefirió favorecer que el teclado fuese intuitivo para el usuario.



Capítulo 5. Software del Teclado.

5.1. Introducción.

En este capítulo se presentan los pasos que se siguieron para lograr el programa completo del teclado encargado de ejecutar los tres modos elegidos previamente, el capítulo finaliza con las funciones necesarias para hacer que el MCU funcione como sensor capacitivo realizando pruebas en un electrodo.

5.2. Implementación de las Funciones del Teclado.

Para hacer las pruebas de códigos, primero se implementó un circuito provisorio de botones conectados al MCU, donde cada botón simbolizó un futuro electrodo (Figura 5.1), esto también fue de utilidad al momento de probar los modelos expuestos en los capítulos anteriores con otras personas.

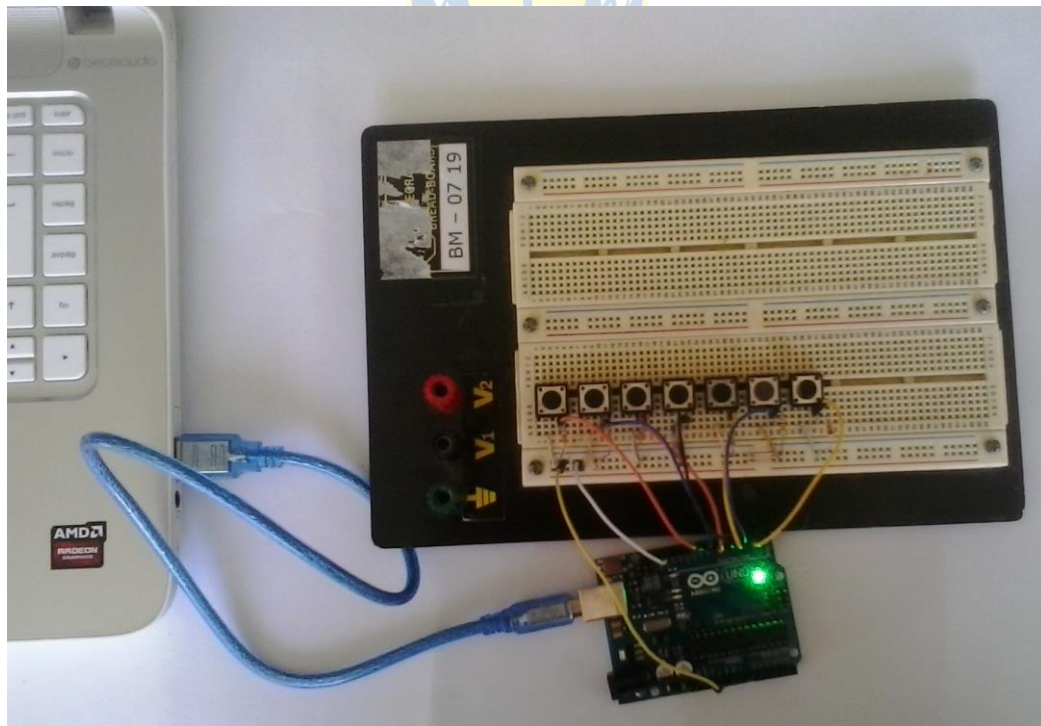


Figura 5.1: Prueba de los modos en botones.

Lo primero que se intentó realizar fue simular las teclas del primer modo, es decir, la tecla Enter y las cuatro flechas direccionales. Para esto se usaron algunas funciones de la librería Keyboard.h que permite al MCU emular un teclado al ser conectado vía USB con un computador, dentro de la página de información de la librería existe una lista de definiciones para las teclas que se quieren simular [23], en la librería existe la función “Keyboard.press()” que recibe estas definiciones y luego envía la señal correspondiente al computador quien interpretaría esta señal como que en el teclado se está presionando la tecla correspondiente a la definición.

Algo a tener en consideración es que a menos que luego de la función “Keyboard.press()” se indique la función “Keyboard.releaseAll()” en el código del programa, el MCU continuará enviando la señal como si la tecla continuara siendo presionada (entrando en un “loop” infinito), esta última función indica que se ha terminado de enviar todo lo que ha sido mandado con anterioridad, sin importar si fue sólo una tecla o varias de ellas. Así, el primer modo estaba completo, dejando un lapso de tiempo entre la presión de una tecla y otra, ya que si la tecla se mantenía apretada el computador interpretaba lo mismo, es decir, que la respectiva tecla del teclado continuaba siendo presionada.

Para el segundo modo, se incluyó la librería “Mouse.h” donde se utilizó la función “Mouse.move” que se encarga de mover el puntero tanto en el eje X como el eje Y de forma simultánea un determinado número de pixeles que el usuario decide. La forma de hacerlo fue medir si los botones direccionales estaban siendo presionados, si había alguno que estaba siendo presionado, se establecía un rango de pixeles en el que se movía el puntero en la dirección correspondiente a la representada por el botón, luego se esperaban algunos milisegundos y se volvía a sensar si los botones estaban siendo presionados repitiendo el procedimiento anterior. Para el caso de que fuese presionado más de un botón, el movimiento resultante es una combinación de las direcciones que están siendo presionadas, si las direcciones activadas son opuestas no hay movimiento del puntero. En el caso del botón del clic, se usaron las funciones “Mouse.press” y “Mouse.release” la primera ingresa la señal al computador de que se está presionando el clic izquierdo del “mouse” y la segunda indica que el clic del “mouse” ha terminado de ser presionado (evitando entrar en un “loop” infinito como el caso anteriormente descrito).

Finalmente la programación del tercer modo requirió el uso de las funciones “Keyboard.print()”, “Keyboard.press()” y “Keyboard.releaseAll()”, la primera permite enviar un mensaje completo, el que es interpretado por el computador como si se hubiera escrito por teclado, de esta forma se puede escribir, por ejemplo, tan sólo una letra o toda una oración, las otras dos

funciones permiten ingresar alguna tecla especial del teclado e indicar que el mensaje ya fue enviado o la tecla terminó de ser presionada.

Para lograr realizar el abecedario se distribuyeron siete letras por cada botón de dirección, para avanzar se crearon dos variables, una que diría si el botón está siendo apretado y otro que guardaría el estado anterior del botón por ciclo del MCU, la variable que analiza si el botón está siendo presionado varía su valor entre el 0 y el 6, asignándole a cada número una letra, si el botón es presionado, su valor cambia con respecto al que tenía en el ciclo pasado del MCU, de este modo se sigue la siguiente secuencia: se borra un espacio, se escribe la nueva letra y se espera. Si se continúa presionando la tecla, la última letra es borrada y se escribe la siguiente, así se hará hasta que se deje de mantener apretado el botón o se terminen todas las opciones disponibles en ese botón, entonces vuelve a comenzar desde la primera letra nuevamente. Esto mismo se hace para los cuatro botones direccionales, teniendo el último además la tecla Espacio y Borrar entre sus opciones. En la Figura 5.2 se puede ver un diagrama de flujo que explica el funcionamiento del modo 3.

A fin de unir los tres modos, se realizó una máquina de estado con el sexto botón, al ser presionado, las variables que recogen el estado de los botones adquieren un nuevo significado, pudiendo enviar señales de “mouse” o de teclado según sea conveniente, para esto se creó un contador que fuese de cero a dos, e incrementase cada vez que el sexto botón fuese presionado, de manera que dependiendo del valor de esta variable, el MCU realiza una interpretación diferente de lo que se buscaba realizar al presionar alguno de los otros cinco botones. En la Figura 5.3 se muestra un diagrama de flujo que ejemplifica la lógica del código completo anteriormente explicado.

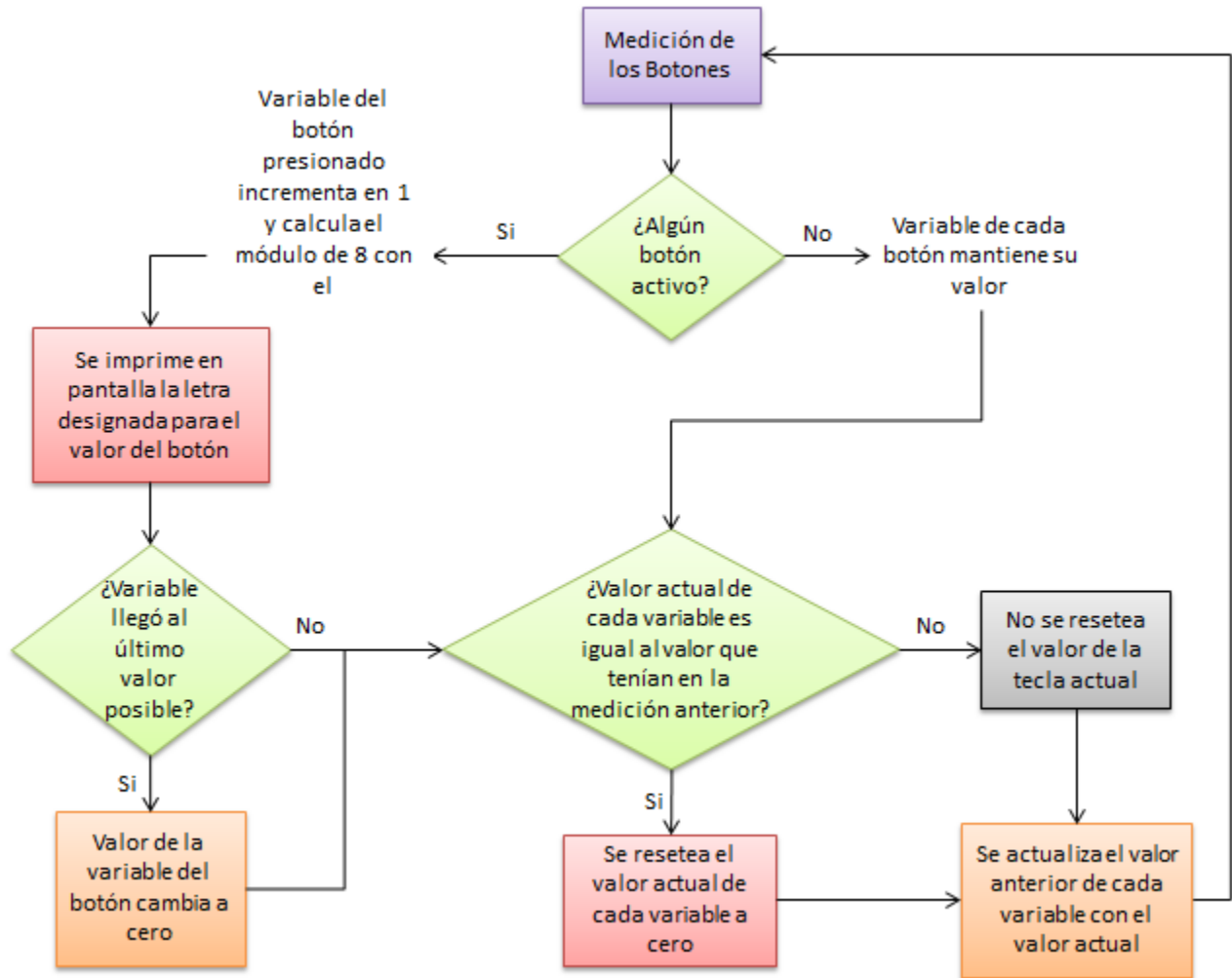


Figura 5.2: Diagrama de flujo de los pasos que sigue el modo 3 del teclado.

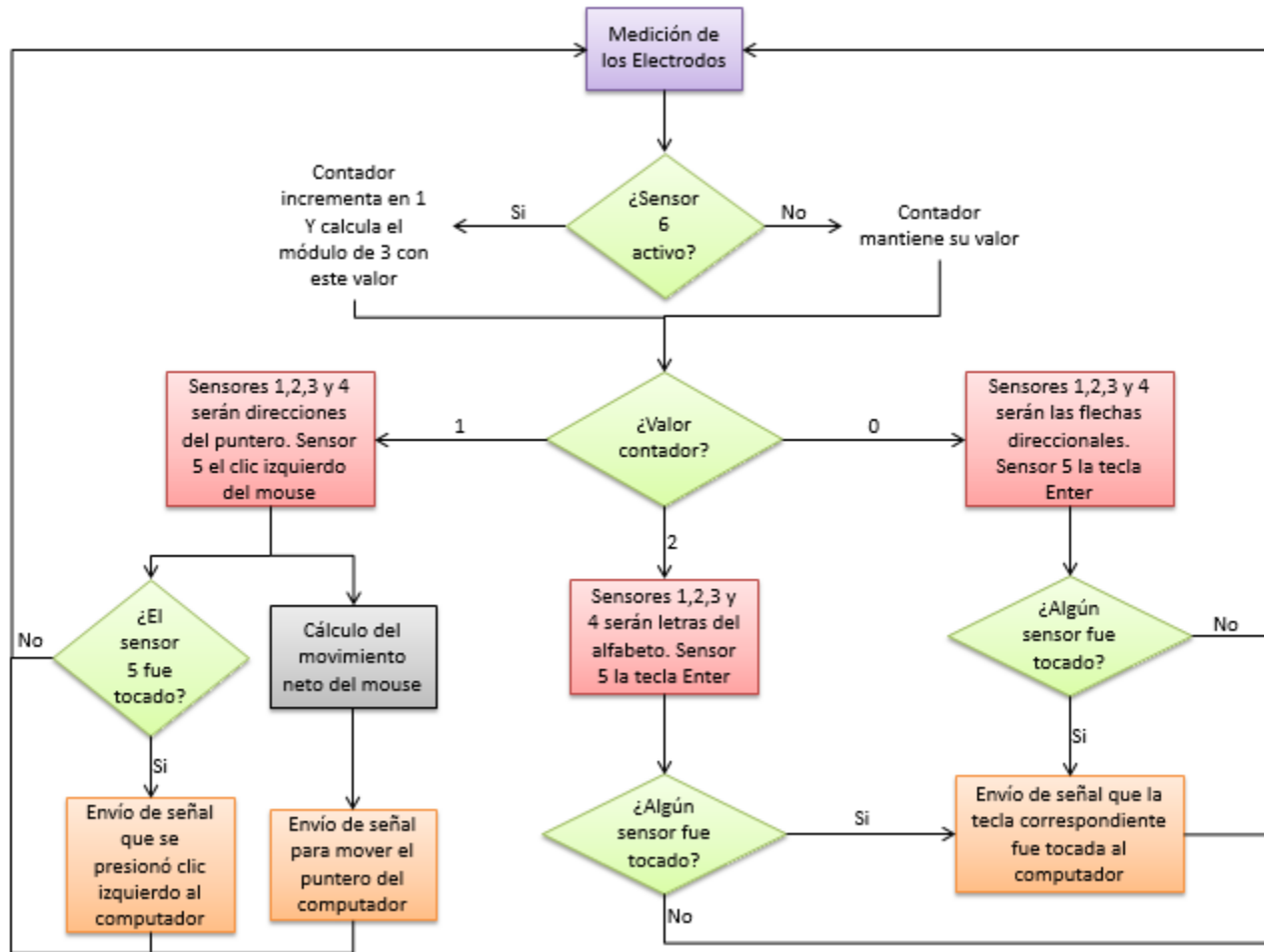


Figura 5.3: Diagrama de flujo de los pasos que sigue el programa completo del teclado.

Para un mejor entendimiento se debe considerar que los electrodos están enumerados del 1 al 6, siendo el electrodo 6 el encargado de cambiar el modo del teclado.

5.3. Detección del Tacto con los Electrodos.

5.3.1 Funciones Necesarias.

Para utilizar el MCU como sensor capacitivo se debía agregar la librería “CapacitiveSensor”. Las funciones que se utilizaron dentro de la librería para poder medir la capacitancia fueron “CapacitiveSensor(pin1,pin2)” la cual se encargaba de definir cuáles serían los pines que por un lado enviaría la señal y por otro mediría el tiempo de respuesta para así determinar el cambio de capacitancia en el electrodo. La función “capacitiveSensor(número de muestras)” toma la cantidad de muestras pedidas y luego retorna un entero que contiene el promedio del tiempo que tardó el cambio de estado, siendo un indicador del cambio de la capacidad. Este número es la suma de las

restas de cada muestra de la capacitancia medida menos un valor de base, que es asumido bajo la condición de no uso. Este valor de base es recalibrado a intervalos fijos, este intervalo esta prefijado en 20 segundos, aunque puede ser cambiado o anular esta re-calibración, esta condición hace sumamente importante una calibración antes del uso.

5.3.2 Determinación de Umbral para Toques Directos en los Electrodo.

Debido a que el MCU recibe la capacitancia como una serie de valores, se optó por pedir que en cada loop una cantidad de 30 muestras, las que luego fueron promediadas y observadas en el computador mediante la comunicación serial del MCU, la razón para promediar este número de muestras fue para evitar que la aparición de ruido ocasional afectara los resultados de las lecturas. Para comenzar se realizaron pruebas para medir sólo un electrodo, en donde cada vez que se tocara el electrodo (que consistía en un cuadrado de una lámina de cobre) debía encenderse un LED (Figura 5.4). El MCU mostró tener un rendimiento rápido para las necesidades que se tenían, tardando en obtener todas las muestras en un rango de los 10 a 15 milisegundos. La librería entrega la libertad de tener un solo pin que se encarga de enviar la señal de cambio y, por otro lado, tener muchos pines actuando como sensores conectados él.

Al realizar la prueba con dos electrodos, se notó una diferencia en el tiempo de respuesta mucho más lento, tardando entre 400 a 500 milisegundos en obtener todas las muestras para los dos electrodos, sin embargo, al darle a cada pin sensor su propio pin que entregara una señal de cambio, el tiempo de espera para tener todas las muestras volvió a estar en la magnitud de tiempo anterior, experimentando un incremento en el tiempo alrededor de 5 milisegundos.

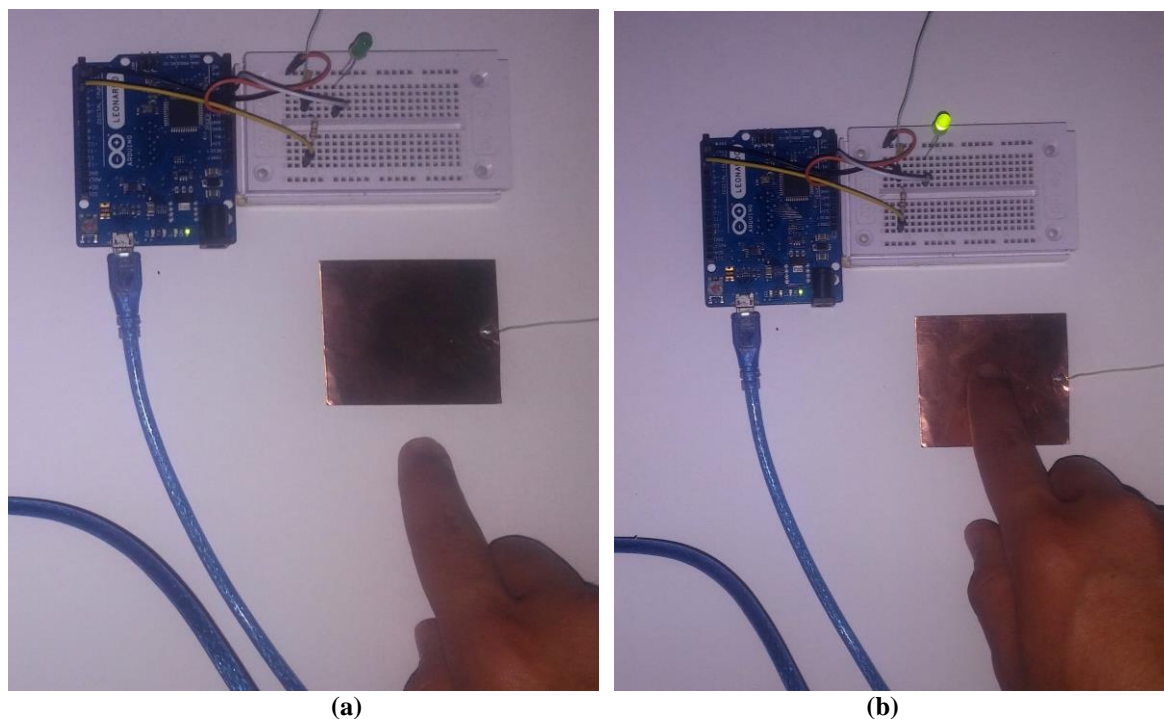


Figura 5.4: Prueba de tacto con un electrodo.

(a) Antes de tocar el electrodo; (b) después de tocar el electrodo.

Al comienzo se realizaron pruebas tocando directamente una lámina de cobre, en dónde debía encenderse un LED si se detectaba el contacto de la mano con el electrodo, esto se hizo de manera sucesiva hasta que se definió un umbral con un valor tal que sólo se activase el LED al sobrepasarlo, aunque los valores en reposo y cuando se tocaba el electrodo continuaban variando dentro de un cierto rango, este era bastante menor al umbral elegido, en la Figura 5.5 se observan los datos graficados antes y después de tocar el electrodo.

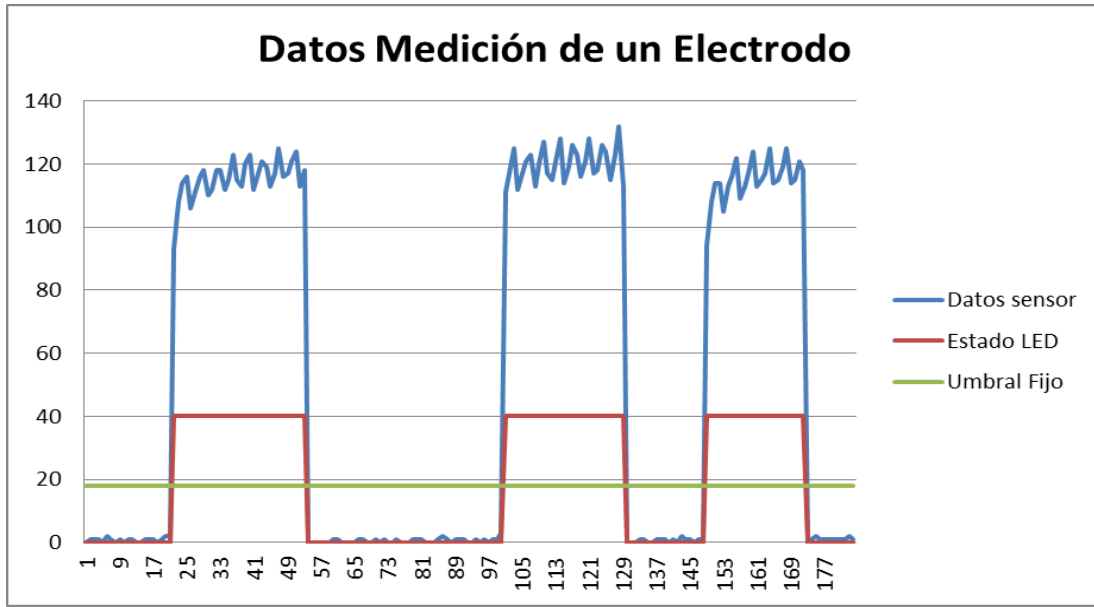


Figura 5.5: Gráfico de datos medidos de un solo electrodo.

El electrodo fue tocado entre los datos 20 y 55 la primera vez, luego entre los datos 100 y 130, finalmente entre los datos 150 y 175. El valor del LED marca 40 cuando está encendido y 0 cuando está apagado.



Capítulo 6. Modificaciones para Lograr Modelo Final.

6.1. Introducción.

Este capítulo inicia con la creación de seis electrodos de prueba con los que se comienza a probar el programa hecho en el capítulo anterior, se debieron realizar varios ajustes por efecto del ruido causado por los mismos electrodos, luego se prueban las modificaciones en una placa PCB pasando finalmente a pruebas sobre el circuito terminado.

6.2. Pruebas con Seis Electrodos.

Una vez que funcionaron las pruebas sobre un solo sensor se pasó a probar con seis electrodos (los que serían necesarios según el modelo elegido), cada uno con su propio par de pines en el MCU uno que envía la señal de cambio y otro que la recibe evitando de esta manera incrementar demasiado el tiempo que tarda en llegar las mediciones al MCU, en esta etapa se probó un programa inicial que solo encendía LEDs y luego se probaron los programas con los tres modos desarrollados anteriormente. En la Figura 6.1 se observan los electrodos hechos con cinta de cobre como electrodos pegados en un trozo de cartón piedra para mantenerlos fijos, además de seis LEDs que indican cuando se detecta el tacto en alguno de ellos.

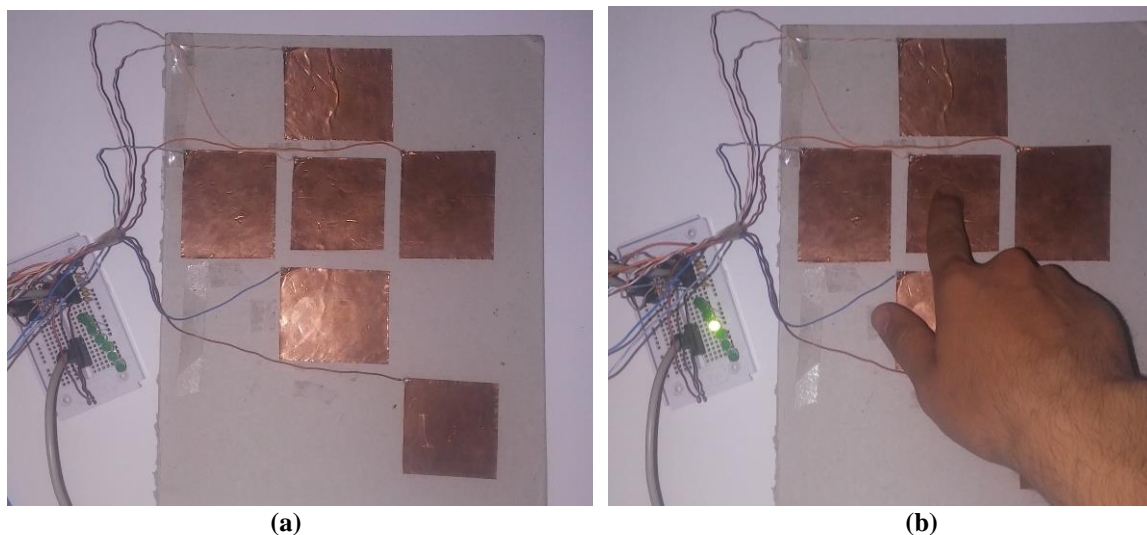


Figura 6.1: Prueba de tacto con seis electrodos.

(a) Antes de tocar algún electrodo; (b) mientras se está tocando el electrodo central.

Las pruebas realizadas arrojaron otras problemáticas, la primera de ellas fue que en algunas ocasiones al conectar el modelo al computador los valores devueltos al MCU en un comienzo eran muy altos, estando por sobre el umbral todo el tiempo, sin embargo una vez tocados los electrodos, estos volvían a su rango de valores normales, esto trajo como inconveniente que al conectar el modelo con las funcionalidades del teclado al computador este se salía de control estando así hasta que se tocaran todos los electrodos, es por esta razón que se optó por poner como condición en el programa que fuesen tocados todos los electrodos antes de que iniciasen los modos del teclado. En la Figura 6.2 se muestra un gráfico con los valores iniciales devueltos al tocar todos los electrodos una primera y segunda vez. La situación anterior se debía a perturbaciones al momento de la auto-calibración del sensor, al existir perturbaciones que afectaban la calibración producía una respuesta exagerada en el MCU hasta que se sensaba la capacitancia de la mano.

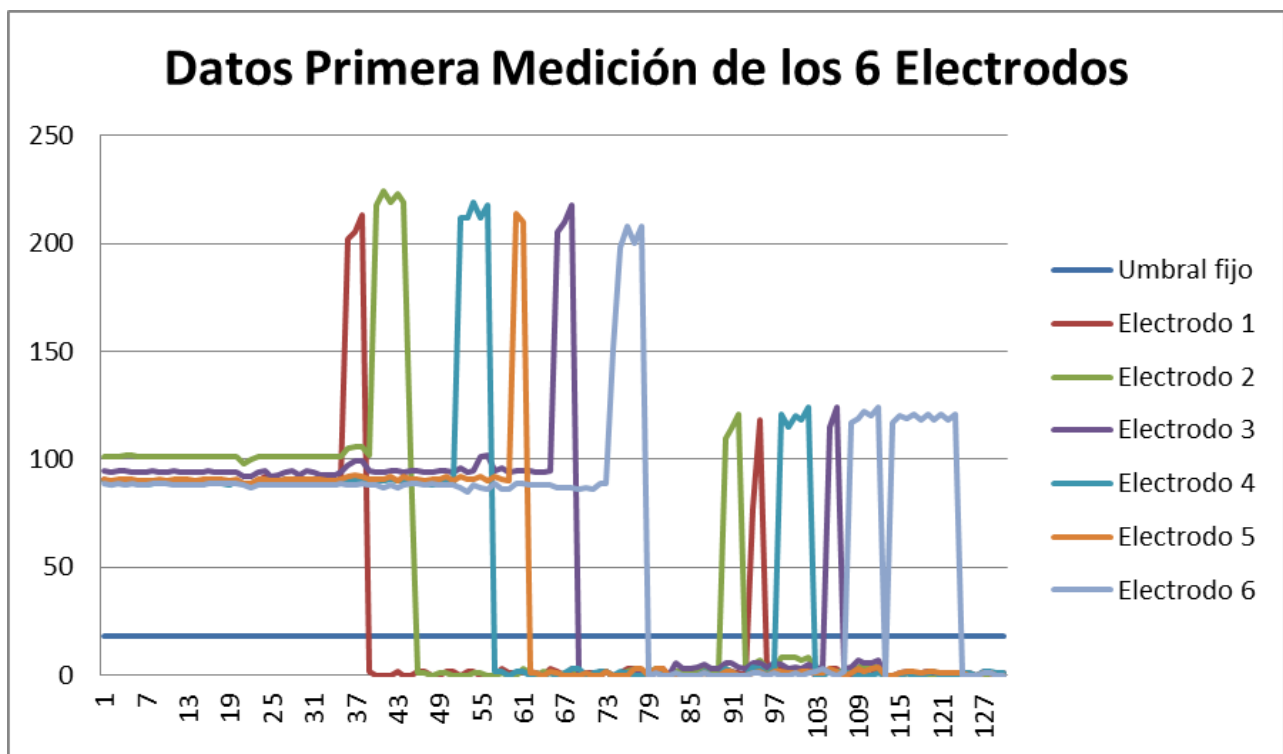


Figura 6.2: Gráfico datos medición inicial de los seis electrodos.

Los electrodos fueron tocados por primera vez entre los datos 35 y 79, luego entre los datos 90 al 125. Se puede observar el cambio que hay en los valores devueltos después de ser tocados por primera vez.

El segundo problema que ocurría sucedía cuando se tocaba un electrodo por varios segundos, la capacitancia del o los electrodos adyacentes se veía afectada por ele electrodo activado,

lo que producía un cambio en los valores devueltos en un electrodo que no se quería activar, ocurriendo en ocasiones que pasaba el umbral de activación. En la Figura 6.3 se observa una segunda medición de los seis electrodos en donde se mantiene tocado uno de ellos por 10 segundos lo que provoca que otro electrodo se accione al verse afectado por la capacitancia.

La situación anterior supuso un problema, pues al probar el programa con las funciones del teclado, cuando se tocaba un electrodo por diez segundos o más, su variación en la capacitancia influenciaba en la de los demás que aún no habían sido tocados moviendo los valores entregados por el sensor de algunos por momentos sobre el umbral de activación. Como consecuencia el teclado realizaba una serie de operaciones sin que el usuario lo ejecutara, como escribir letras inconexas con saltos de línea, mover el “*mouse*” y seleccionar o borrar elementos en un texto. Para solucionar se decidió cambiar el umbral fijo por un umbral dinámico de manera que aunque se viesen afectados por la capacitancia de un electrodo vecino, esta variación no sería suficiente para activar el electrodo de forma involuntaria.

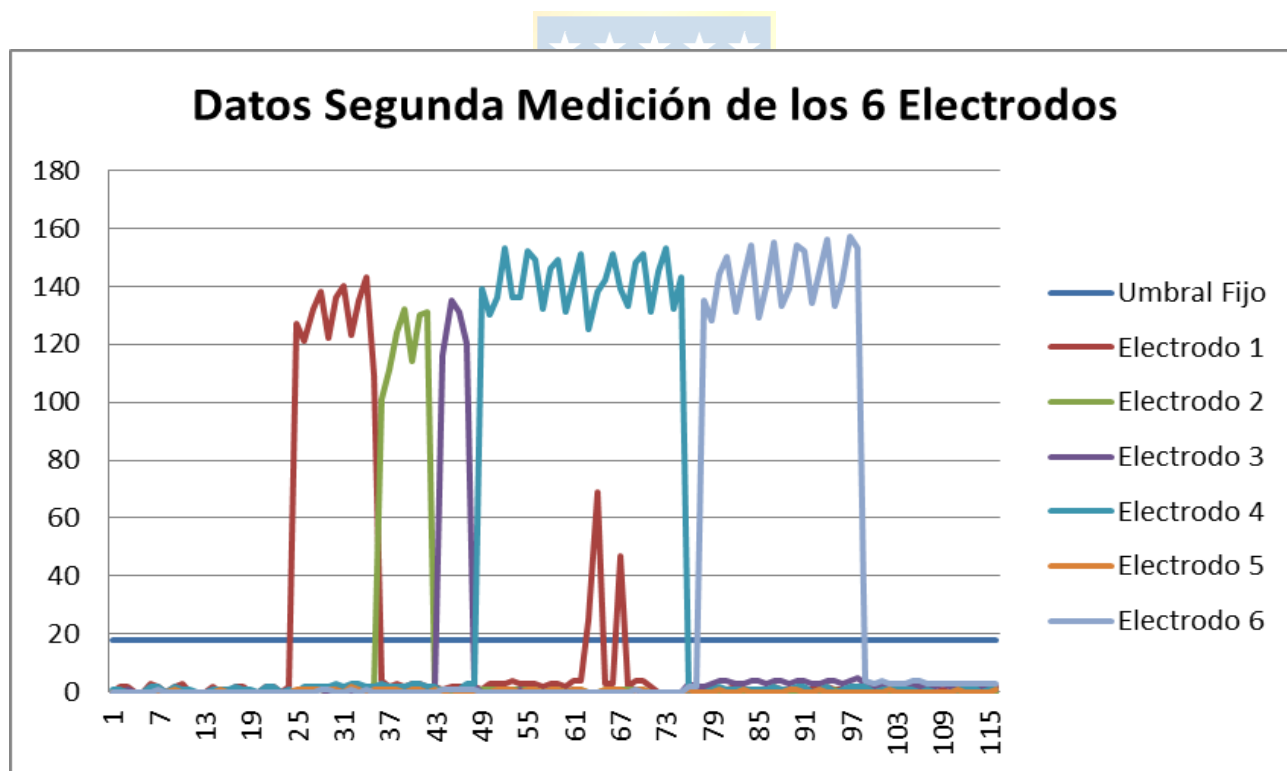


Figura 6.3: Gráfico datos segunda medición de los seis electrodos.

Debido a que los electrodos variaban de estado desde sobre a bajo el umbral antes de ser tocados por primera vez, esta condición se cumple sólo cuando todos los electrodos han sido tocados

por unos segundos (realizando esto a modo de calibración) para que los valores devueltos en la medición de la capacitancia estén en un rango adecuado y no varíe demasiado el valor devuelto de un electrodo cuando el electrodo adyacente sea tocado.

Dentro de la solución se estableció que el umbral sería igual a un porcentaje del valor más alto registrado por los electodos, para efectos prácticos se fijó en un 50% del valor más alto, de manera que una vez han sido tocados, el valor del umbral sería actualizado si es que un electrodo era mayor al doble del umbral actual, lo que se consiguió utilizando esta variación es que el teclado no activaba funciones en el tiempo al no ser utilizado y respondía adecuadamente a los toques que se le daban, pudiendo cumplir con todos los requerimientos en términos de funcionalidad.

Finalmente, se imprimió el circuito en una placa PCB para realizar las pruebas en la placa definitiva, en la Figura 6.4 se puede ver la placa y el circuito que se encuentra en un “*protoboard*” mientras se realizan las últimas pruebas de tacto y pasarlo a una placa definitiva.

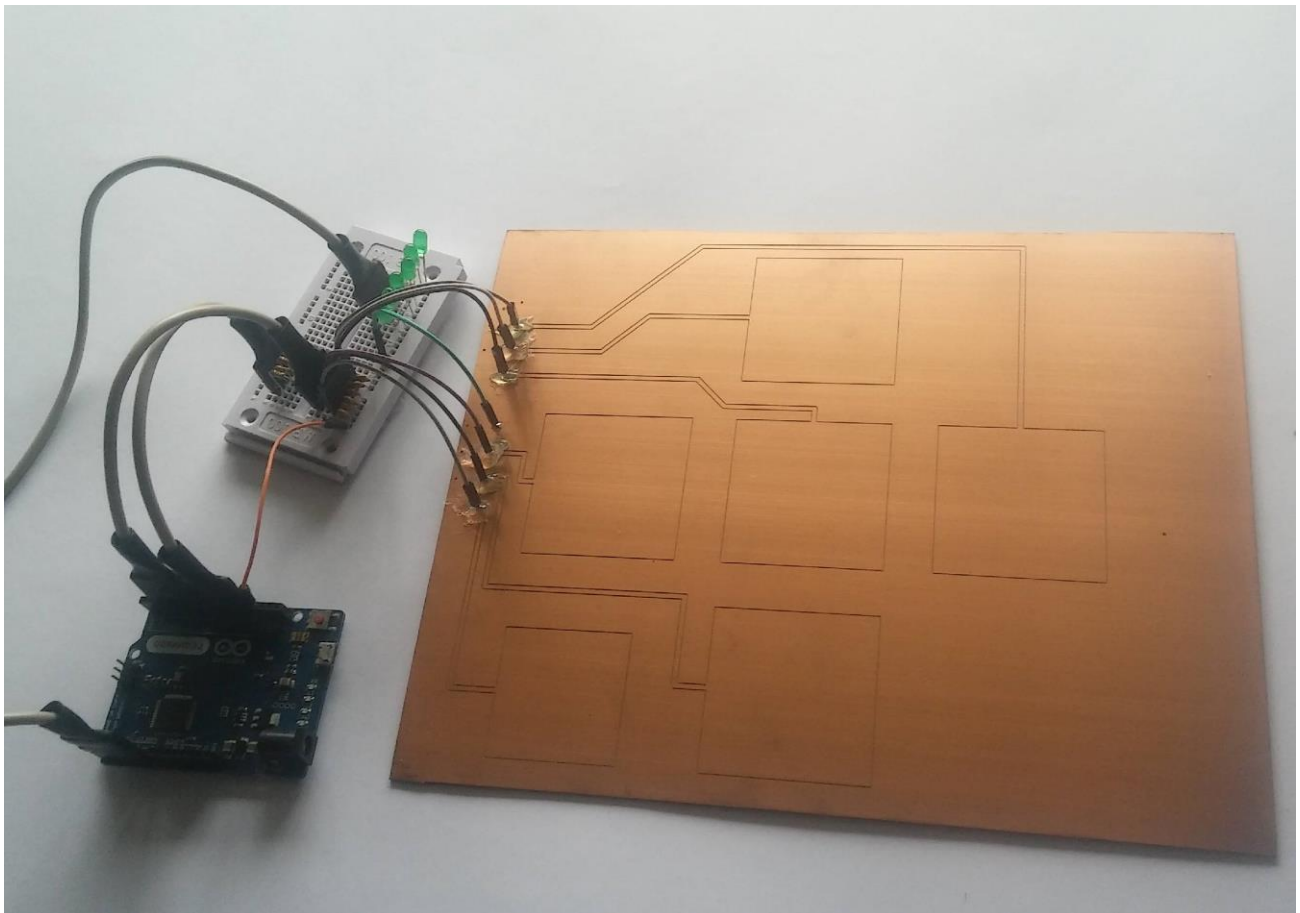


Figura 6.4: Imagen de la placa PCB de los electodos.

6.3. Pruebas y Modificaciones en la Placa PCB.

Si bien el programa funcionaba adecuadamente con la configuración anterior (tocando los electrodos directamente), la idea es que los electrodos estén ocultos a la vista, por lo que se hace necesario cubrirlos con algún material, es por esto que se decidió cubrir los electrodos con una lámina de plástico, el cual permite detectar la capacitancia a través de él, pero impide tener un contacto directo con los electrodos.

El inconveniente que se encontró con esto fue que al no tocar directamente los electrodos para activarlos, la diferencia de la capacitancia que se produce es mucho menor al que se obtenía en el caso anterior, cuando se tocaban directamente las placas metálicas. Por lo que fue necesario aumentar la sensibilidad del sistema, para esto se incrementó la resistencia del circuito a 10 mega ohm de manera que pudiese detectar los cambios de capacitancia a una distancia mayor (por aumentar su sensibilidad) [24]. Como resultado se obtuvo un incremento en la sensibilidad de cada electrodo, pero juntamente con ello, se hicieron más sensibles al ruido ambiental, a la proximidad de la mano (sin que se tocara el espacio que les correspondía) y sobre todo al cambio en la capacitancia que experimentaban sus electrodos vecinos, además, el tiempo por medición completa de los electrodos aumentó de 30 a 200 milisegundos, aunque luego este tiempo se logró reducir a 100 milisegundos eliminando algunos procesos innecesarios dentro del código del programa. En la Figura 6.5 se observa graficada esta situación al tocar solo dos electrodos de la placa a través de una lámina plástica, esta lámina tenía un espesor de 0.5 milímetros, como resultado se vio que otros electrodos se volvieron a ver influenciados por la capacitancia del electrodo activado el cual tampoco tenía un incremento significativo en los valores devueltos comparado con los otros electrodos.

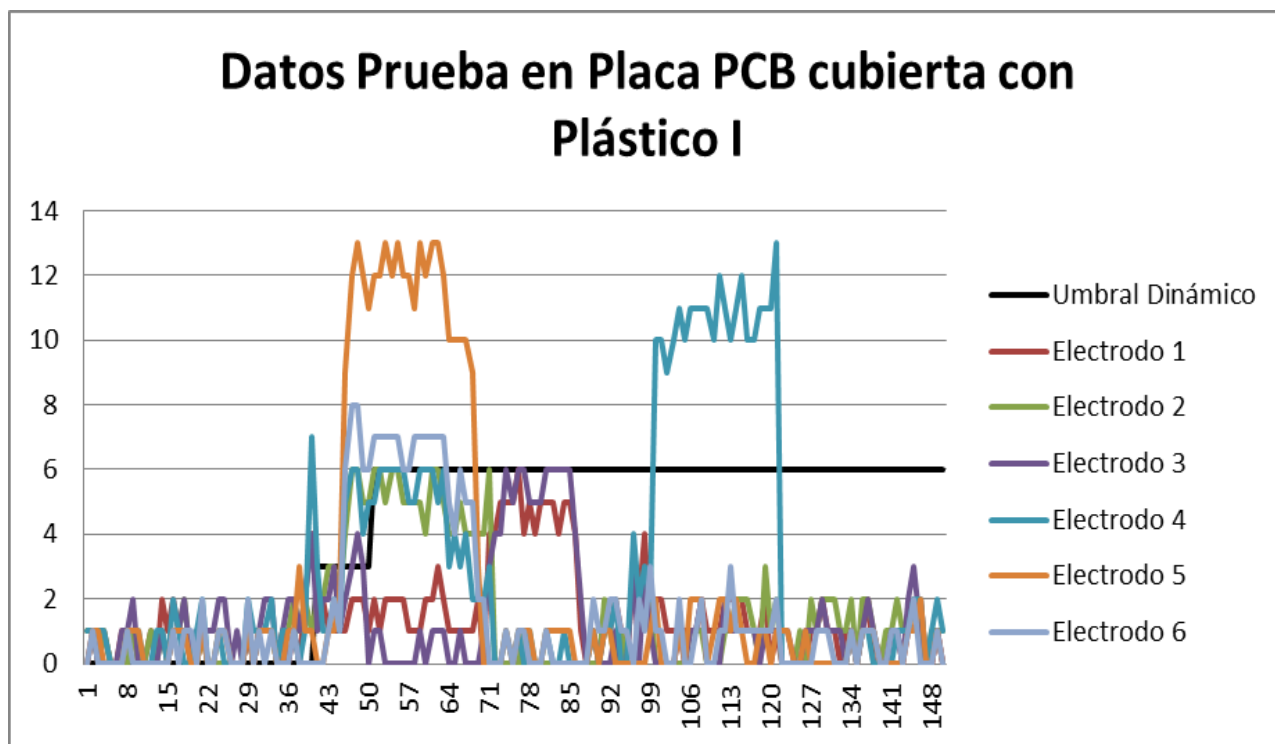


Figura 6.5: Medición en placa PCB cuando dos electrodos son tocados a través de plástico.

Los electrodos activados son el 4 y el 5, si bien se distingue claramente en su gráfica el cambio al ser tocados, los valores devueltos por los otros electrodos hacen difícil establecer un umbral único para todos.

Para solucionar esto, se intentó ajustar nuevamente el umbral a un nuevo porcentaje del valor más alto devuelto por los electrodos, pero se encontró que las variaciones no eran igual en todos los electrodos, quedando algunos (una vez tocados) en un rango sobre el umbral, otros siempre se movían bajo el umbral aunque fuesen tocados y la mayoría contenía al umbral dentro del rango de valores devueltos cuando son activados (ya que mientras son activados también existe una variación de los valores devueltos). Por la razón anterior se decidió hacer un arreglo de umbrales, de forma que cada electrodo tenga su propio valor umbral sobre el cual moverse, donde cada umbral sería calculado como el 60% del valor más alto detectado por su respectivo electrodo. Así, una vez “calibrados” todos los sensores (después del primer toque), cada uno tendría un rango propio sobre el cual moverse, el cual es actualizado si es que el valor de su electrodo incrementa con el tiempo. Además, para incrementar el cambio en los valores devueltos cuando un electrodo se activa se volvió a incrementar la resistencia del circuito de 10 a 20 mega ohm, esto sin embargo no trajo un aumento significativo en el tiempo que tomaba la adquisición de los datos de los electrodos por parte de MCU, tardando alrededor de 110 a 120 milisegundos en hacer una medición de todos los

electrodos. En la Figura 6.6 se observa como incrementaron los valores devueltos al activar un electrodo y como cada uno ahora posee su propio umbral sobre el cual moverse.

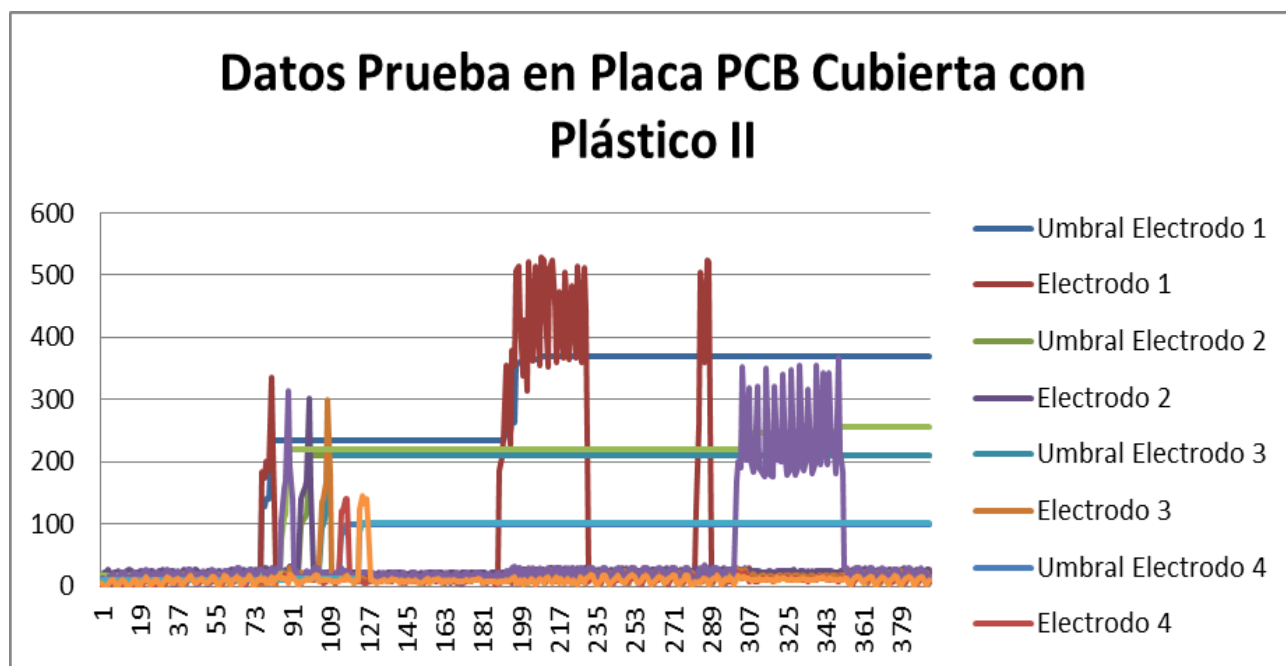
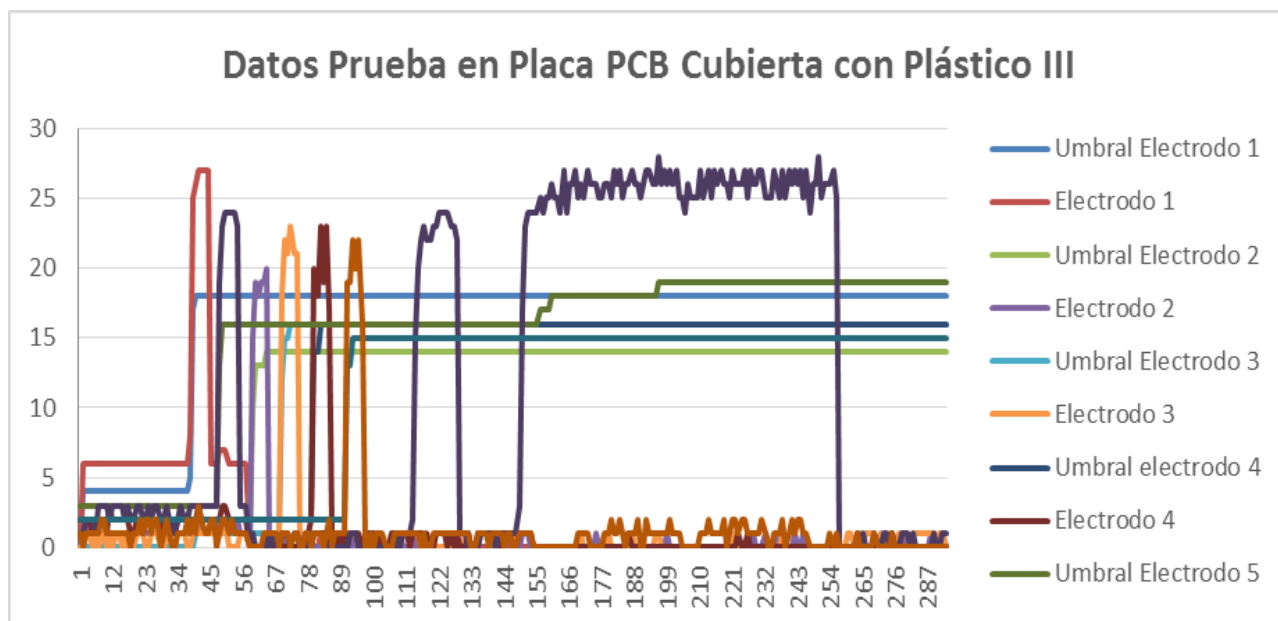
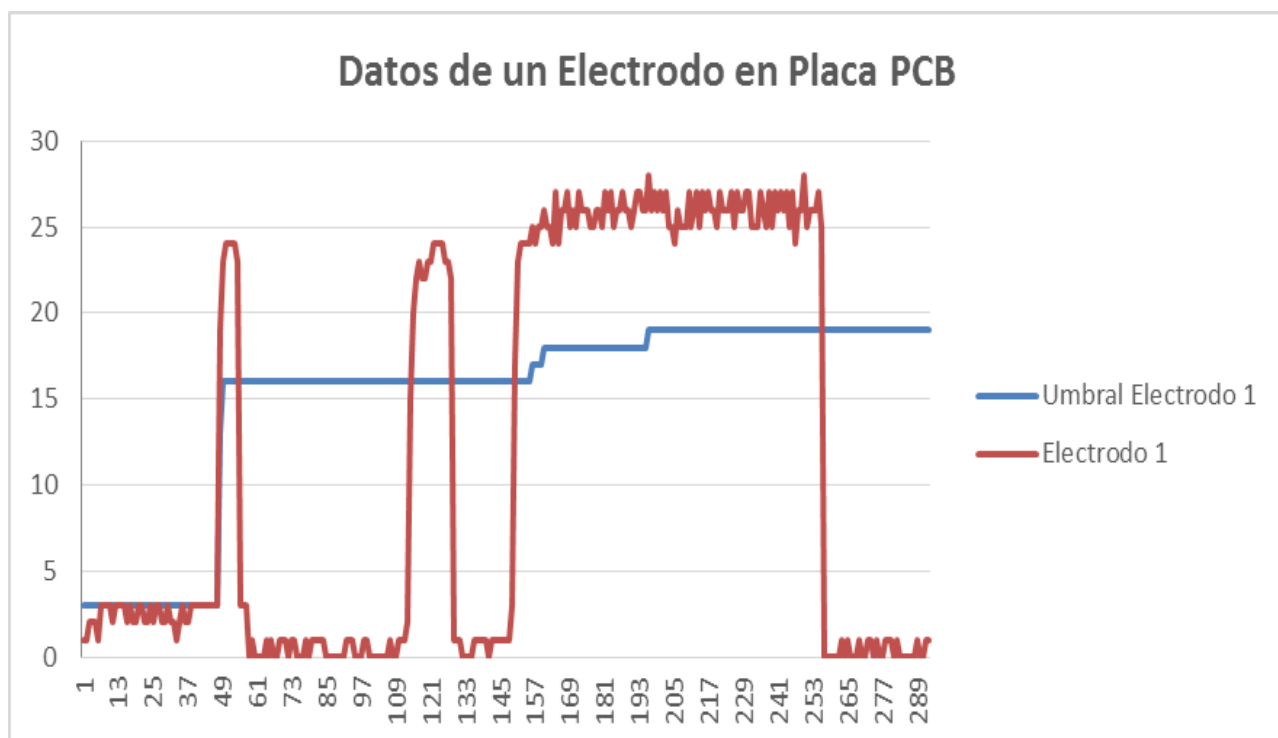


Figura 6.6: Medición en placa PCB con umbrales para cada electrodo.

Estas modificaciones eliminaron el problema de activar un electrodo vecino de forma involuntaria, sin embargo al activar un electrodo a través del plástico este se hacía demasiado sensible a las variaciones ambientales, lo que en la práctica se traducía en que el electrodo subía y bajaba del umbral de activación volviendo el funcionamiento del teclado intermitente lo que resultaba desagradable para el usuario (aunque no activase otra acción diferente a la deseada). Para solucionar esto se intentó bajar el umbral de cada electrodo, sin embargo se incurría en la posibilidad que las variaciones fuesen lo bastante grandes para que se repitiera el mal funcionamiento. Finalmente se probó conectar el resto de la placa de PCB a la tierra del MCU para tener sus variaciones como referencia, ya que al ser también una superficie de cobre podía verse afectada por las variaciones en la capacitancia (y afectar a los electrodos al mismo tiempo). En la Figura 6.7 se observa la medición una vez hecha la conexión del resto de la placa a tierra.



(a)



(b)

Figura 6.7: Medición en placa PCB con placa conectada a tierra.

(a) Se observan todos los electrodos con su respectivo umbral dinámico; (b) Se aprecia de mejor forma los valores entregados por el electrodo 1 y cómo evoluciona su propio umbral.

Con la última modificación el teclado pudo ser usado sin inconvenientes ni activaciones de electrodos no deseadas. La prueba de mayor duración fue de 45 minutos en donde se utilizó el

teclado a intervalos, es decir, primero se usó por diez minutos, luego se dejó de utilizar por 10 minutos, a continuación se usó por 7 minutos más y luego se dejó de usar por 8 minutos, finalmente se usó por 10 minutos concluyendo de este modo la prueba. La finalidad de esta prueba era ver si el teclado experimentaba fallos en el tiempo (por ejemplo que se descalibrara) ya sea por la utilización en el tiempo como si se dejaba de usar por unos momentos, estas pruebas fueron hechas en un notebook utilizando solo la batería como fuente de alimentación. De esta forma el teclado mostró un buen comportamiento en el tiempo evitando activar los electrodos involuntariamente.

El modelo final del teclado cuenta con cuatro LEDs los cuales indican cuando el teclado ha sido calibrado y está listo para usarse, y si está en el modo de las teclas direccionales, control del “mouse” o en escritura respectivamente y su cubierta de plástico consiste en un mica semi-transparente con el fin de que se puedan ver unas imágenes impresas en papel de la función de cada electrodo a través de ella. En la 6.8 se ve una imagen del modelo con el cuál se probó el teclado con distintas personas.

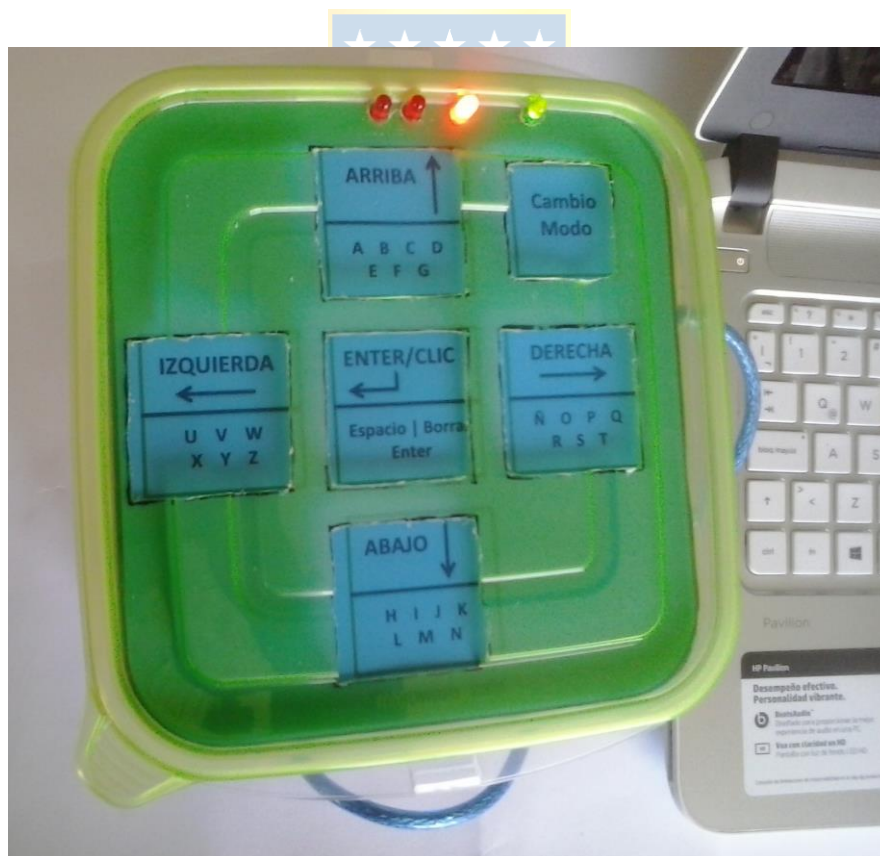


Figura 6.8: Vista modelo final del teclado.

Una vez terminados los ajustes del teclado se realizaron pruebas de funcionamiento. Si bien idealmente se deseaba probar el dispositivo con un paciente real, la falta de disponibilidad de uno

llevó a realizar pruebas con 10 personas sanas, estas personas fueron adultos jóvenes con edades comprendidas entre 20 y 30 años de edad, donde según la bibliografía están dentro del rango etario en donde más lesiones medulares se sufren.

La forma de probar el funcionamiento del teclado fue solicitarles a las personas que usasen el teclado por media hora mientras realizaban sus actividades cotidianas en el computador, y luego respondiesen un cuestionario de aprobación. Los resultados arrojados por el cuestionario fueron que el 80% prefería el modelo con la disposición de los electrodos como se desarrolló finalmente versus la variación presentada en el capítulo 3. Por otra parte el 80% dijo haber podido realizar todo lo que quería hacer en el computador, sin embargo el 10% indicó que se les hizo dificultoso el uso del teclado, dentro de los argumentos fue que no estaban acostumbrados al teclado y resultaba más lento que el método al que están acostumbrados, no obstante al ponerse en el lugar de un paciente con tetraparesia un 70% lo ocuparía y un 80% lo recomendaría para el uso cotidiano. Dentro de las mejoras que se pudieran hacer la mayoría mencionó el agregar los números o la opción de hacer mayúsculas.

Dentro de las impresiones recogidas por las personas es que si bien entendían el funcionamiento del teclado una vez explicado y que les era posible realizar las actividades que deseaban en el computador hasta cierto punto, consideraban necesaria una guía de uso teniendo en cuenta a las personas que no contasen con la explicación del funcionamiento, es por ello que se elaboró una guía, la que fue anexada a este trabajo.

6.4. Modelamiento Cubierta del Teclado.

Una vez finalizado el dispositivo se realizó un modelamiento de una cubierta para el teclado con la finalidad de entregar una opción más orientada al usuario en cuanto al diseño final se refiere. Este diseño consiste en una base donde van los elementos del circuito y una tapa que permite delimitar de mejor forma el área de cada botón, lo que tiene por finalidad ayudar al usuario a evitar accionar un electrodo involuntariamente. Aunque no se logró crear la cubierta por razones de tiempo, el acrílico o a través de una impresora 3D son opciones factibles para la fabricación. Cabe mencionar que las dimensiones del modelo presentado están pensadas para el teclado diseñado, es decir, de 18.5 cm de ancho y largo, y 5 cm de alto. En la Figura 6.9 se observa una vista del ensamble entre la tapa y la base de la cubierta, otras vistas del modelo serán anexadas a este trabajo.

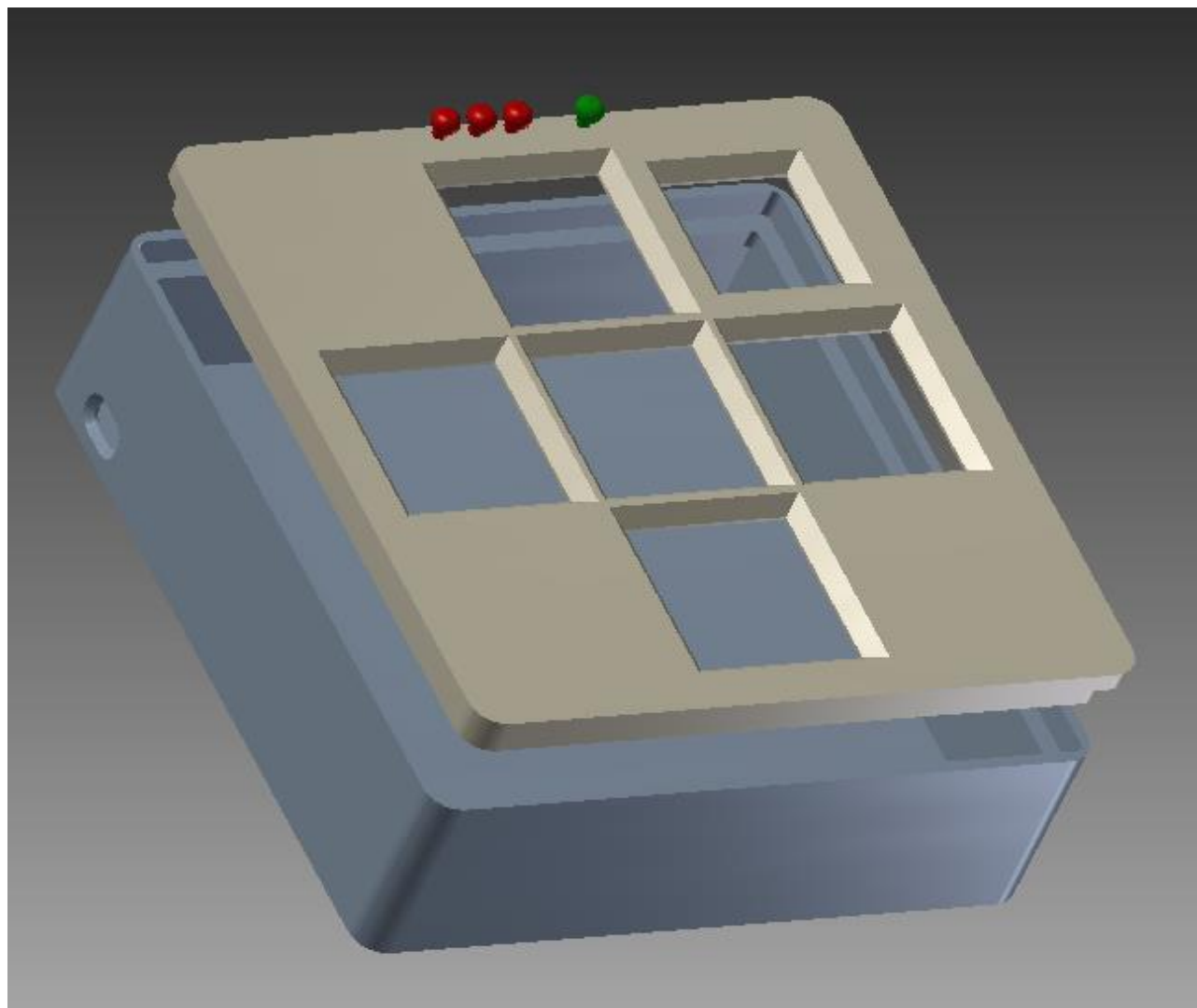


Figura 6.9: Modelo cubierta del teclado (vista ensamble)

Capítulo 7. Conclusiones.

7.1. Conclusiones.

Se ha diseñado un modelo de teclado con las funciones necesarias para poder utilizar un computador bajo el requerimiento de no necesitar presión o movimientos finos, ya que no utilizaría botones, sino electrodos de cobre, con un tamaño aproximado de 20 cm cuadrados por placa.

Se logró implementar un MCU como sensor capacitivo permitiendo detectar el tacto a través de plástico, como resultado se pudieron controlar las funciones que se deseaban, obteniendo muy pocas ocasiones alguna activación de un sensor de forma involuntaria.

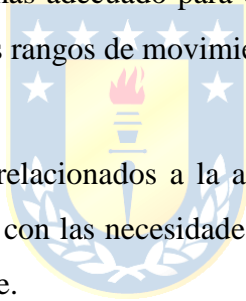
Se ha generado el programa necesario para que el computador reconozca la tecla que se quiere activar o el movimiento que se desea realizar con el puntero, esto fue probado en un comienzo con botones en vez de usar los sensores capacitivos dando buenos resultados en su funcionamiento, más tarde se implementó con el MCU como sensor capacitivo teniendo un correcto funcionamiento tras algunas mejoras en el programa para la detección del tacto con los electrodos.

En definitiva el desarrollo de este tipo de dispositivos es posible, realizando en esta ocasión un modelo sencillo, no obstante los mismos pasos se pueden seguir para realizar otros modelos de teclado más complejos con un mayor número de funcionalidades.

7.2. Trabajo Futuro.

Para los trabajos futuros, teniendo como finalidad mejorar el dispositivo se recomienda considerar al menos estos puntos:

- Realizar pruebas en pacientes con dificultad motora de distinto grado de manera que se puedan obtener sus impresiones del dispositivo, adaptándolo a las necesidades que ellos manifiesten.
- Analizar y optimizar el código del programa de manera que al incluir nuevas funcionalidades o modos no afecten el desempeño del dispositivo.
- Implementar un diseño más adecuado para el usuario final, además de examinar las dimensiones en base a los rangos de movimiento que tengan estas personas.
- Incluir a profesionales relacionados a la atención del usuario objetivo, de manera que estos puedan aportar con las necesidades que debe cumplir el dispositivo para la rehabilitación del paciente.



Bibliografía

- [1] “*Sensores Capacitivos Principios de Funcionamiento*”, SKILLTEN. E-learning Tecnológico. 14, Sep. 2013.
- [2] S. Chatzandroulis, D. Tsoukalas, and P. Neukomm. “A Miniature Pressure System with a Capacitive Sensor and a Passive Telemetry Link for Use in Implantable Applications,” *Journal of Microelectromechanical System*, vol. 9, no. 1, March 2000.
- [3] Bryce Osoinach, System and Applications Engineer. “*Proximity Capacitive Sensor Technology for “touch” Sensing Applications*” Freescale Semiconductor, Inc, 2008.
- [4] M. Maggiali, G. Cannata, P. Maiolino, G. Metta, M. Randazzo, and G. Sandini. “*Embedded Distributed Capacitive Tactile Sensor*” *Mechatronics 2008*, June 23 – 25, University of Limerick, Ireland.
- [5] J. Meyer, P. Lukowicz, and G. Tröster, “*Textile Pressure Sensor for Muscle Activity and Motion Detection*,” *Wearable Comput. Lab.*, ETH Zurich, Zurich, 2006. *IEEE*.
- [6] T. Salpavaara, J. Verho and J. Leikkala. “*Capacitive Insole Sensor for Hip Surgery Rehabilitation*,” *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on Date Jan. 30 2008-Feb.2008*, *IEEE*.
- [7] “*Curso de Neuroanatomía*”, Escuela de Medicina Pontificia Universidad Católica de Chile. Recuperado de: <http://escuela.med.puc.cl/paginas/departamentos/anatomia/cursoenlinea/down/medula.pdf> en Feb. 2016.
- [8] Dr. Matamala. “*Médula Espinal*”, Universidad de la Frontera. Recuperado de: <http://www.med.ufro.cl/Recursos/neuroanatomia/archivos/pdf/apunte4.pdf> en Feb. 2016.
- [9] Claudia Patricia Henao Lema, Julio Ernesto Pérez Parra. “*Lesiones medulares y discapacidad: revisión bibliográfica*”, Año 10 - Vol. 10 N° 2 - CHÍA, Colombia; Agost. 2010, p 157-172.

- [10] “*Clasificación de la Asociación Americana de Lesión Medular*”, Plataforma de comunicación de la Fundación del Hospital Nacional de Parapléjicos. Recuperado de: http://www.infomedula.org/index.php?option=com_content&view=article&id=1006%3Aclasificacion-de-la-asociacion-americana-de-lesion-medular-&catid=35%3Ahospital&lang=es en Feb. 2016.
- [11] Fenichel GM. “*Paraplejia y cuadriplejia*”. Neurología pediátrica clínica. Un enfoque por signos y síntomas (pp 252-272). 6th ed. (2005). Philadelphia.
- [12] Montoto A, Ferreiro ME, Rodríguez A, Sánchez I, Ferrero A, Aguilar JJ, Climent JM, Conejero JA, Flórez MT, Peña A, Zambudio P. “*Lesión medular*”, Manual SERMEF de Rehabilitación y Medicina Física. Médica Panamericana, Madrid; 2006. p. 505-519
- [13] Franceschini M, Di Clemente B, Rampello A, Nora M, Spizzichino L. “*Longitudinal outcome 6 years after spinal cord injury*”, Spinal Cord 2003; 41: 280-5.
- [14] Bender JE, Hernández E, Prida M, Araujo F, Zamora F. “*Caracterización clínica de pacientes con lesión medular traumática*”, Rev Mex Neuroci 2002; p. 135-42.
- [15] Sipski ML, Jackson AB, Gómez-Marín O, Estores I, Stein A. “*Effects of gender on neurologic and functional recovery after spinal cord injury*”, Arch Phys Med Rehabil 2004.
- [16] Middleton JW, Lim K, Taylor L, Soden R, Rutkowski S. “*Patterns of morbidity and rehospitalisation following spinal cord injury*”, Spinal Cord 2004.
- [17] Lugo LH, Salinas F, García HI. “*Out-patient rehabilitation programme for spinal cord injured patients: Evaluation of the results on motor FIM score*”, Disability and Rehabilitation 2007; p. 11-12.
- [18] R. Lloréns, M. Alcañiz, M. Navarro, J Ferri, and E. Noé. “*Self-awareness rehabilitation through a multi-touch virtual game board after acquired brain injury*” Inst. Interuniversitario de Investig. en, Bioingeniería y Tecnol. Orientada al Ser Humano, Univ. Politec. de Valencia, Valencia, Spain. *IEEE*.
- [19] M. Sherer, T. Hart, T. G. Nick, J. Whyte, R. N. Thompson, and S. A. Yablon, “*Early impaired self-Awareness after traumatic brain injury*,” Arch Phys Med Rehabil, vol. 84, pp.

- 168-76, Feb 2003.
- [20] M. Sherer, P. Bergloff, E. Levin, W. M. High, Jr., K. E. Oden, and T. G. Nick, "*Impaired awareness and employment outcome after traumatic brain injury*," *The Journal of head trauma rehabilitation*, vol. 13, pp. 52-61, Oct 1998.
- [21] T. Markow, N. Ramakrishnan, K. Huang, T. Starner, M. Eicholtz, S. Garrett, H. Profita, A. Scarlata, C. Schooler, A. Tarun, and D. Backus, "*Mobile Music Touch: Vibration Stimulus in Hand Rehabilitation*," *Coll. of Comput., Georgia Inst. of Technol., Atlanta, GA, USA, 2010. IEEE.*
- [22] Luis Thayer Ojeda. "*Arduino Leonardo*", Ingeniería MCI Ltda. (Olimex Chile). Recuperado de: <http://arduino.cl/arduino-leonardo/>. En Dic. 2015.
- [23] "*Mouse and Keyboard libraries*", Arduino Website. Arduino 2015. Recuperado de: <https://www.arduino.cc/en/Reference/MouseKeyboard>. En Dic. 2015.
- [24] Paul Badger, "*Capacitive Sensing Library*", Arduino Website. Arduino 2015. Recuperado de: <http://playground.arduino.cc/Main/CapacitiveSensor?from=Main.CapSense>. En Dic. 2015.
- [25] Antonio José Salazar Gómez. "*Circuitos de primer orden RL y RC*", Universidad de los Andes. Recuperado de: http://www.prof.uniandes.edu.co/~ant-sala/cursos/FDC/Contenidos/08_Circuitos_de_Primer_Orden_RL_y_RC.pdf en Feb. 2015.

Anexo A. Vistas Modelo de Cubierta del Teclado.

A continuación se presentan algunas vistas desde distintas perspectivas del modelo para la cubierta del teclado capacitivo.

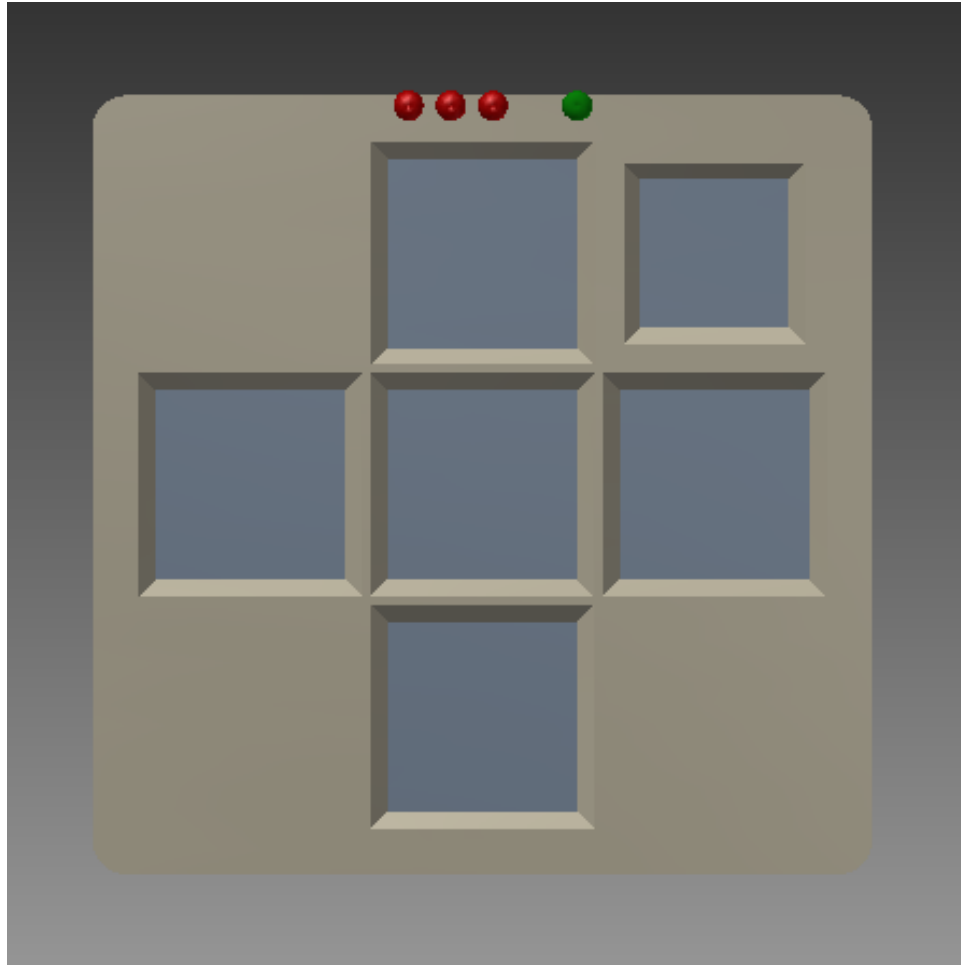


Figura A.1: Vista tapa de la cubierta del teclado.

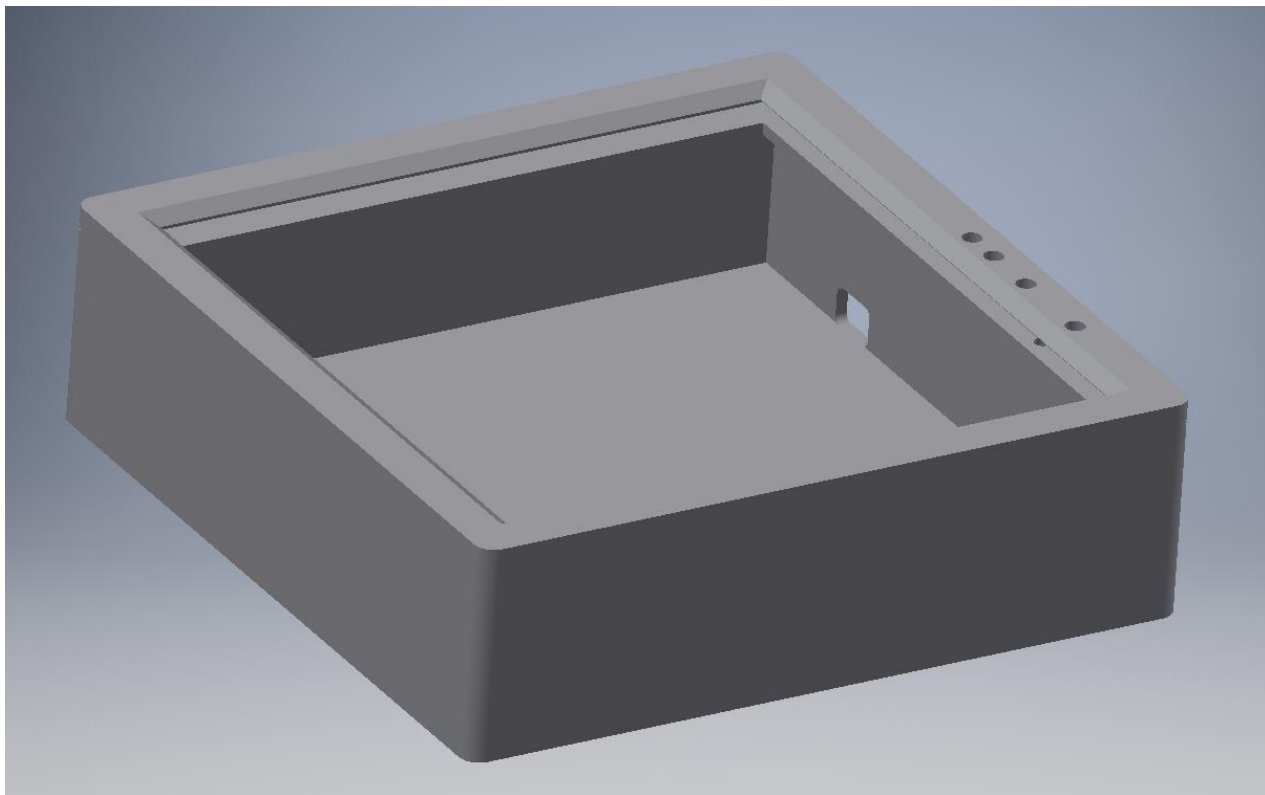


Figura A.2: Vista base de la cubierta del teclado.

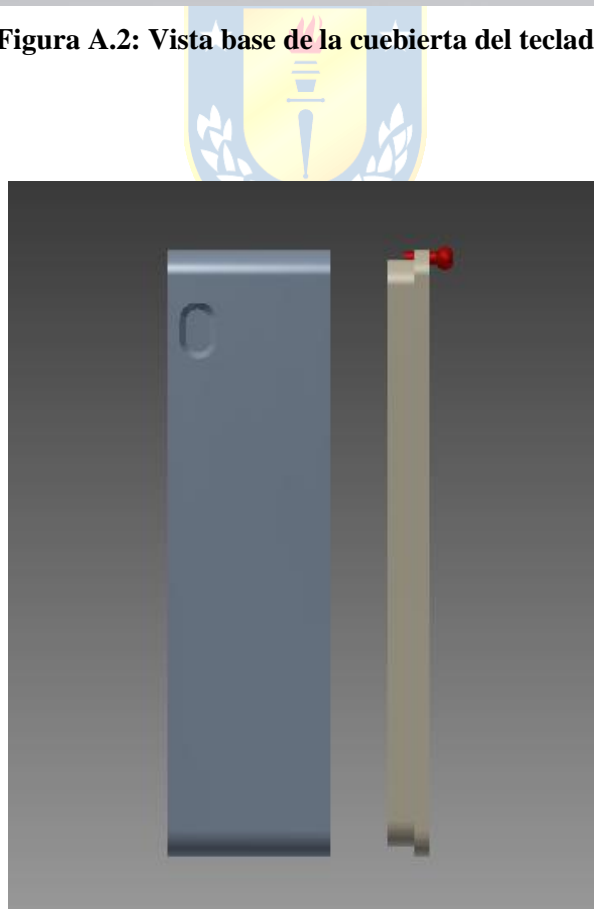


Figura A.3: Vista lateral de base y tapa de la cubierta del teclado.

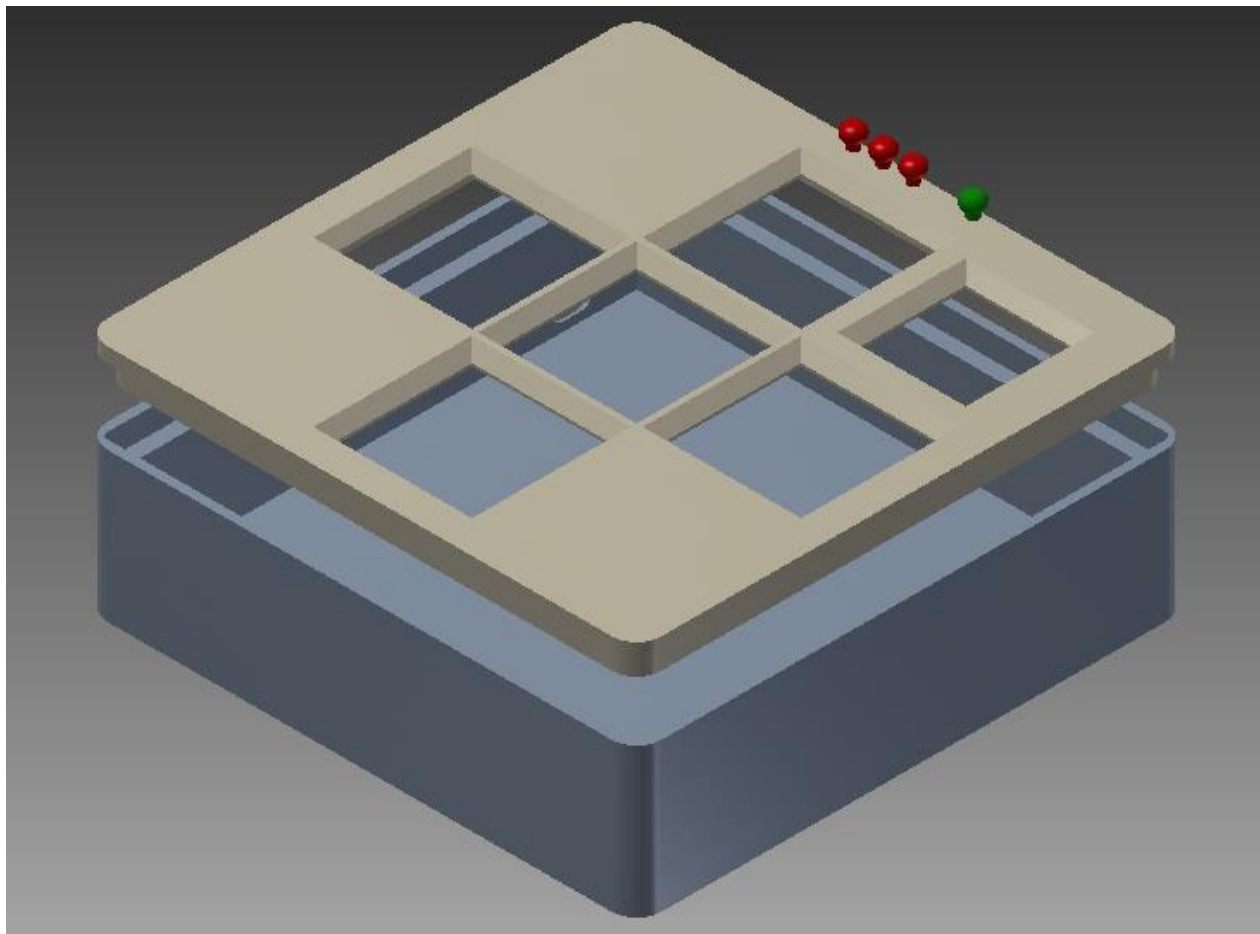
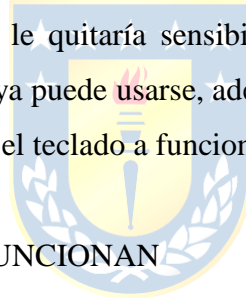


Figura A.4: Vista ensamble base y tapa de la cubierta del teclado.

Anexo B. Guía Rápida de Usuario.

Esta guía rápida tiene como objetivo ayudar al usuario a entender el funcionamiento del teclado de forma que su experiencia de uso sea satisfactoria. El teclado consta de seis “teclas” las cuales se activan con el tacto y están orientadas de forma intuitiva para generar la idea de las direcciones arriba, abajo, derecha e izquierda junto con una tecla central. Además el teclado cuenta con cuatro LEDs uno verde y tres rojos, encargados de indicar cuando se ha hecho la calibración y qué modalidad está en uso respectivamente.

Lo primero que se debe hacer con el teclado es calibrarlo, para esto se debe conectar a un computador y acto seguido tocar un par de segundos cada electrodo, se debe tener en cuenta como se utilizará el teclado, es decir, si el usuario usara solo un dedo para activar las teclas en la calibración de be usar sólo un dedo para tocar las teclas (si usará más dedos no hay problema con tocar con un dedo las teclas en la calibración, pero no se debe calibrar el teclado con más dedos o la palma si es que no se usarán, ya que le quitaría sensibilidad). Una vez calibrado el teclado se encenderá el LED verde indicando que ya puede usarse, además se encenderá uno de los LEDs rojos que significará en qué modo comenzará el teclado a funcionar.



MODOS DEL TECLADO Y COMO FUNCIONAN

El teclado cuenta con tres modos de operación, el modo activo puede cambiarse por uno de los otros dos cuando el usuario lo desee, los tres modos son:

- A) Modo uno o flechas direccionales: En este modo las teclas imitan las cuatro flechas direccionales de un teclado convencional en la misma dirección en la que aparecen en la superficie del dispositivo, la tecla central simula la tecla “Enter”, este modo tiene por finalidad por ejemplo moverse en los iconos del escritorio y entrar a algún programa, o subir y bajar por páginas de internet, el primer LED rojo de derecha a izquierda indica cuando está activado este modo.
- B) Modo dos o control de “mouse”: En este modo el teclado toma el control del puntero del computador, las cuatro teclas que antes eran las flechas direccionales de un teclado ahora

son las direcciones en las que se puede mover el puntero en la pantalla, para mover el puntero solo se debe mantener el toque sobre la dirección deseada y el puntero se moverá hasta que se deje de tocar la tecla, la tecla del centro esta vez funciona como el clic izquierdo del “*mouse*”. Este modo está indicado como activo por el LED rojo del centro.

- C) Modo tres o modo de escritura: En este modo el teclado puede escribir las letras del abecedario, para esto cada tecla tiene en su mitad inferior escrito cuales son las letras que le corresponden, para escribir se debe mantener tocada la tecla que contenga la letra deseada hasta que esta aparezca en la pantalla, la tecla central en este caso simula tres teclas: la “Barra Espaciadora”, la tecla de “Borrar” y la tecla “Enter” y su activación es la misma que para las otras teclas, se debe mantener el tacto sobre la tecla hasta que suceda lo deseado. Una recomendación es que si se desean realizar caracteres especiales o que no están en el teclado es activar el teclado en pantalla del computador y tomando el control del puntero del computador escribir o seleccionar los caracteres que no se encuentren en el teclado, para ello vaya a “Buscar” en su computador y escriba “teclado en pantalla”, con eso le aparecerá el programa que despliega dicho teclado el cual tiene las mismas opciones de un teclado convencional.



La tecla de la esquina superior derecha sólo sirve para cambiar de modo, para hacerlo se debe tocar una vez y el teclado pasará al siguiente modo lo que será indicado por los LEDs rojos (no es necesario mantener el tacto sobre este electrodo para cambiar de modo).

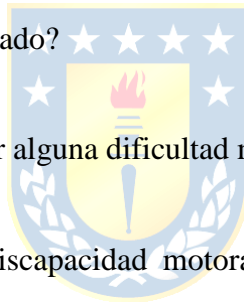
Un funcionamiento erróneo puede deberse a una des-calibración del teclado, en caso de que esto suceda desconecte el teclado y vuelva a conectarlo al computador, esto reiniciará el dispositivo siendo necesario volver a calibrarlo volviendo a su correcto funcionamiento.

Anexo C. Cuestionario de Aprobación.

El siguiente cuestionario tuvo por finalidad evaluar el desempeño del teclado por personas sanas las cuales dieron sus impresiones luego de utilizar el teclado, sus respuestas fueron recogidas en el desarrollo de este trabajo siendo un aporte en el modelo resultante.

CUESTIONARIO

- 1-. Tras presentárseles las dos formas de distribución de los electrodos ¿Cuál distribución le resulta más sencilla de utilizar?
- 2-. Una vez utilizado el teclado. ¿Logró realizar todo lo que quería en el computador?
- 3-. ¿Considera dificultoso el uso del teclado?
- 4-. ¿Utilizaría el teclado en caso de tener alguna dificultad motora de consideración?
- 5-. Si conociera a una persona con discapacidad motora. ¿Le recomendaría el uso del teclado presentado?
- 6-. ¿Qué funcionalidad faltó a su juicio y le gustaría que se implementase?



Anexo D. Código del Teclado

Este es el código implementado en el dispositivo desarrollado.

```

#include <CapacitiveSensor.h>
#include "Mouse.h"
#include "Keyboard.h"

char arriba = KEY_UP_ARROW;
char abajo = KEY_DOWN_ARROW;
char izquierda = KEY_LEFT_ARROW;
char derecha = KEY_RIGHT_ARROW;
char enter = KEY_RETURN;

CapacitiveSensor  capacitor_arriba = CapacitiveSensor(8,2);
CapacitiveSensor  capacitor_abajo = CapacitiveSensor(13,7);
CapacitiveSensor  capacitor_derecha = CapacitiveSensor(12,6);
CapacitiveSensor  capacitor_izquierda = CapacitiveSensor(9,3);
CapacitiveSensor  capacitor_enter_clic = CapacitiveSensor(11,5);
CapacitiveSensor  capacitor__cambio = CapacitiveSensor(10,4);

int touch_arriba = 0;
int touch_abajo = 0;
int touch_derecha = 0;
int touch_izquierda = 0;
int touch_enter_clic = 0;
int touch_cambio = 0;
int touch_anterior_cambio = 0;

const int led1 = A0;
const int led2 = A1;
const int led3 = A2;
const int led4 = A3;

int estado_led1 = LOW;
int estado_led2 = LOW;
int estado_led3 = LOW;
int estado_led4 = LOW;
int estado_led5 = LOW;
int estado_led6 = LOW;

const int led = 13;
int estado_led = LOW;

```

```

int letras1 = 0;
int letras2 = 0;
int letras3 = 0;
int letras4 = 0;
int letras5 = 0;
int letras1_anterior = 0;
int letras2_anterior = 0;
int letras3_anterior = 0;
int letras4_anterior = 0;
int letras5_anterior = 0;

```

```

int calibracion = 0;
int teclado_mouse = 0;
int rango_mouse = 11;
int umbral[6] = {0,0,0,0,0,0};
int iniciar = 0;

```

```

int espera1 = 0;
int espera_tecla = 100;
int espera_letra = 500;

```

```

void setup() {

```

```

    pinMode(led1,OUTPUT);
    pinMode(led2,OUTPUT);
    pinMode(led3,OUTPUT);
    pinMode(led4,OUTPUT);

```



```

    capacitor_arriba.set_CS_AutocaL_Millis(0xFFFFFFFF);
    capacitor_abajo.set_CS_AutocaL_Millis(0xFFFFFFFF);
    capacitor_derecha.set_CS_AutocaL_Millis(0xFFFFFFFF);
    capacitor_izquierda.set_CS_AutocaL_Millis(0xFFFFFFFF);
    capacitor_enter_clic.set_CS_AutocaL_Millis(0xFFFFFFFF);
    capacitor__cambio.set_CS_AutocaL_Millis(0xFFFFFFFF);

```

```

    pinMode(led, OUTPUT);

```

```

    Mouse.begin();
    Keyboard.begin();
}

```

```

void loop() {

```

```

    int presion_tecla = 0;
    long estado_actual[6];
    int i;

```

```

estado_actual[0] = capacitor_arriba.capacitiveSensor(30)/15;
estado_actual[1] = capacitor_abajo.capacitiveSensor(30)/15;
estado_actual[2] = capacitor_derecha.capacitiveSensor(30)/15;
estado_actual[3] = capacitor_izquierda.capacitiveSensor(30)/15;
estado_actual[4] = capacitor_enter_clic.capacitiveSensor(30)/15;
estado_actual[5] = capacitor__cambio.capacitiveSensor(30)/15;

```

```

for(i = 0 ; i < 6 ; i++)
{
  if(umbral[i] < estado_actual[i]*0.5)
    umbral[i] = estado_actual[i]*0.5;
}

```

```

if(estado_actual[0] > umbral[0])
{
  touch_arriba = 1;
}
else
{
  touch_arriba = 0;
}

```

```

if(estado_actual[1] > umbral[1])
{
  touch_abajo = 1;
}
else
{
  touch_abajo = 0;
}

```



```

if(estado_actual[2] > umbral[2])
{
  touch_derecha = 1;
}
else
{
  touch_derecha = 0;
}

```

```

if(estado_actual[3] > umbral[3])
{
  touch_izquierda = 1;
}
else
{
  touch_izquierda = 0;
}

```



```

}

if(estados_actuales[4] > umbral[4])
{
  touch_enter_clic = 1;
}
else
{
  touch_enter_clic = 0;
}

if(estados_actuales[5] > umbral[5])
{
  touch_cambio = 1;
}
else
{
  touch_cambio = 0;
}

if(iniciar < 100)
{
  iniciar = iniciar + 1;
}
if(iniciar == 100)
{
  if(touch_arriba == 0 && touch_abajo == 0 && touch_derecha == 0 && touch_izquierda == 0
  && touch_enter_clic == 0 && touch_cambio == 0)
  {
    calibracion = 1;
    digitalWrite(led1, HIGH);
  }
}

if(calibracion == 1)
{
  if(touch_cambio && !touch_anterior_cambio)
  {
    teclado_mouse = teclado_mouse + 1;
    teclado_mouse = teclado_mouse%3;
  }

  //////////// TECLADO ////////////

  if(teclado_mouse == 0)
  {
    digitalWrite(led2, HIGH);
  }
}

```



```

digitalWrite(led3, LOW);
digitalWrite(led4, LOW);
if(touch_arriba)
{
  Keyboard.press(arriba);
  presion_tecla = 1;
}

if(touch_abajo)
{
  Keyboard.press(abajo);
  presion_tecla = 1;
}

if(touch_derecha)
{
  Keyboard.press(derecha);
  presion_tecla = 1;
}

if(touch_izquierda)
{
  Keyboard.press(izquierda);
  presion_tecla = 1;
}

if(touch_enter_clic)
{
  Keyboard.press(enter);
  presion_tecla = 1;
}

if(presion_tecla == 1)
{
  Keyboard.releaseAll();
  delay(espera_tecla);
  presion_tecla = 0;
}
delay(espera_tecla);
}

//////////////////////////////////  MOUSE  ////////////////////////////////////

if(teclado_mouse == 1)
{
  digitalWrite(led2, LOW);
  digitalWrite(led3, HIGH);
  digitalWrite(led4, LOW);

```



```
int Distanciax = (touch_derecha - touch_izquierda) * rango_mouse;
int Distanciay = (touch_abajo - touch_arriba) * rango_mouse;
```

```
if((Distanciax != 0) || (Distanciay != 0))
{
  Mouse.move(Distanciax, Distanciay, 0);
}
```

```
if(touch_enter_clic)
{
```

```
  if(!Mouse.isPressed(MOUSE_LEFT))
  {
    Mouse.press(MOUSE_LEFT);
  }
}
```

```
else
{
```

```
  if(Mouse.isPressed(MOUSE_LEFT))
  {
    Mouse.release(MOUSE_LEFT);
  }
}
```



```
////////// ESCRITURA //////////
```

```
if(teclado_mouse == 2)
{
  digitalWrite(led2, LOW);
  digitalWrite(led3, LOW);
  digitalWrite(led4, HIGH);
```

```
//***** LETRAS DE LA A LA G *****
```

```
if(touch_arriba)
{
  presion_tecla = 1;
  letras1 = letras1 + 1;
```

```
if(letras1 == 1)
{
  Keyboard.print("a");
}
else if(letras1 == 2)
{
```

```

Keyboard.press(KEY_BACKSPACE);
Keyboard.releaseAll();
Keyboard.print("b");
}
else if(letras1 == 3)
{
Keyboard.press(KEY_BACKSPACE);
Keyboard.releaseAll();
Keyboard.print("c");
}
else if(letras1 == 4)
{
Keyboard.press(KEY_BACKSPACE);
Keyboard.releaseAll();
Keyboard.print("d");
}
else if(letras1 == 5)
{
Keyboard.press(KEY_BACKSPACE);
Keyboard.releaseAll();
Keyboard.print("e");
}
else if(letras1 == 6)
{
Keyboard.press(KEY_BACKSPACE);
Keyboard.releaseAll();
Keyboard.print("f");
}
else if(letras1 == 7)
{
Keyboard.press(KEY_BACKSPACE);
Keyboard.releaseAll();
Keyboard.print("g");
}
else if(letras1 > 7)
{
Keyboard.press(KEY_BACKSPACE);
Keyboard.releaseAll();
Keyboard.print("a");
letras1 = 1;
}
}

//***** LETRAS DE LA H A LA N *****
if(touch_abajo)
{
presion_tecla = 1;
letras2 = letras2 + 1;

```



```
if(letras2 == 1)
{
    Keyboard.print("h");
}
else if(letras2 == 2)
{
    Keyboard.press(KEY_BACKSPACE);
    Keyboard.releaseAll();
    Keyboard.print("i");
}
else if(letras2 == 3)
{
    Keyboard.press(KEY_BACKSPACE);
    Keyboard.releaseAll();
    Keyboard.print("j");
}
else if(letras2 == 4)
{
    Keyboard.press(KEY_BACKSPACE);
    Keyboard.releaseAll();
    Keyboard.print("k");
}
else if(letras2 == 5)
{
    Keyboard.press(KEY_BACKSPACE);
    Keyboard.releaseAll();
    Keyboard.print("l");
}
else if(letras2 == 6)
{
    Keyboard.press(KEY_BACKSPACE);
    Keyboard.releaseAll();
    Keyboard.print("m");
}
else if(letras2 == 7)
{
    Keyboard.press(KEY_BACKSPACE);
    Keyboard.releaseAll();
    Keyboard.print("n");
}
else if(letras2 > 7)
{
    Keyboard.press(KEY_BACKSPACE);
    Keyboard.releaseAll();
    Keyboard.print("h");
    letras2 = 1;
}
}
```



```

}

// ***** LETRAS DE LA NN A LA T *****
if(touch_derecha)
{
    presion_tecla = 1;
    letras3 = letras3 + 1;

    if(letras3 == 1)
    {
        Keyboard.print(";");
    }
    else if(letras3 == 2)
    {
        Keyboard.press(KEY_BACKSPACE);
        Keyboard.releaseAll();
        Keyboard.print("o");
    }
    else if(letras3 == 3)
    {
        Keyboard.press(KEY_BACKSPACE);
        Keyboard.releaseAll();
        Keyboard.print("p");
    }
    else if(letras3 == 4)
    {
        Keyboard.press(KEY_BACKSPACE);
        Keyboard.releaseAll();
        Keyboard.print("q");
    }
    else if(letras3 == 5)
    {
        Keyboard.press(KEY_BACKSPACE);
        Keyboard.releaseAll();
        Keyboard.print("r");
    }
    else if(letras3 == 6)
    {
        Keyboard.press(KEY_BACKSPACE);
        Keyboard.releaseAll();
        Keyboard.print("s");
    }
    else if(letras3 == 7)
    {
        Keyboard.press(KEY_BACKSPACE);
        Keyboard.releaseAll();
        Keyboard.print("t");
    }
}

```



```

else if(letras3 > 7)
{
  Keyboard.press(KEY_BACKSPACE);
  Keyboard.releaseAll();
  Keyboard.print(";");
  letras3 = 1;
}
}

// ***** LETRAS DE LA U A LA Z *****
if(touch_izquierda)
{
  presion_tecla = 1;
  letras4 = letras4 + 1;

  if(letras4 == 1)
  {
    Keyboard.print("u");
  }
  else if(letras4 == 2)
  {
    Keyboard.press(KEY_BACKSPACE);
    Keyboard.releaseAll();
    Keyboard.print("v");
  }
  else if(letras4 == 3)
  {
    Keyboard.press(KEY_BACKSPACE);
    Keyboard.releaseAll();
    Keyboard.print("w");
  }
  else if(letras4 == 4)
  {
    Keyboard.press(KEY_BACKSPACE);
    Keyboard.releaseAll();
    Keyboard.print("x");
  }
  else if(letras4 == 5)
  {
    Keyboard.press(KEY_BACKSPACE);
    Keyboard.releaseAll();
    Keyboard.print("y");
  }
  else if(letras4 == 6)
  {
    Keyboard.press(KEY_BACKSPACE);
    Keyboard.releaseAll();
    Keyboard.print("z");
  }
}

```



```

}
else if(letras4 > 5)
{
  Keyboard.press(KEY_BACKSPACE);
  Keyboard.releaseAll();
  Keyboard.print("u");
  letras4 = 1;
}
}

```

```

if(touch_enter_clic)
{
  presion_tecla = 1;
  letras5 = letras5 + 1;

```

```

if(letras5 == 1)
{
  Keyboard.print(" ");
}
else if(letras5 == 2)
{
  Keyboard.press(KEY_BACKSPACE);
  Keyboard.releaseAll();
  Keyboard.print(char(124));
}
else if(letras5 == 3)
{
  Keyboard.press(KEY_BACKSPACE);
  Keyboard.releaseAll();
  Keyboard.print(char(63));
}
else if(letras5 > 3)
{
  Keyboard.press(KEY_BACKSPACE);
  Keyboard.releaseAll();
  Keyboard.print(" ");
  letras5 = 1;
}
}

```



```

if(presion_tecla == 1)
{
  Keyboard.releaseAll();
  delay(espera_letra);
  presion_tecla = 0;
}

```

```

if(letras1 == letras1_anterior)

```



```

    letras1 = 0;

    if(letras2 == letras2_anterior)
        letras2 = 0;

    if(letras3 == letras3_anterior)
        letras3 = 0;

    if(letras4 == letras4_anterior)
        letras4 = 0;

    if(letras5 == letras5_anterior)
    {
        if(letras5 == 2)
        {
            Keyboard.press(KEY_BACKSPACE);
            Keyboard.releaseAll();
            Keyboard.press(KEY_BACKSPACE);
            Keyboard.releaseAll();
        }
        if(letras5 == 3)
        {
            Keyboard.press(KEY_BACKSPACE);
            Keyboard.releaseAll();
            Keyboard.press(enter);
            Keyboard.releaseAll();
        }
        letras5 = 0;
    }

    letras1_anterior = letras1;
    letras2_anterior = letras2;
    letras3_anterior = letras3;
    letras4_anterior = letras4;
    letras5_anterior = letras5;
}
}
touch_anterior_cambio = touch_cambio;
}

```

