

Control de sistema de realidad aumentada mediante gestos de los dedos.

Manuel Palacios Hurtado

Tutor: José Manuel Mossi García

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2016-17

Valencia, 24 de junio de 2017

Resumen

El humo generado por incendios en recintos cerrados dificulta la visibilidad del servicio de bomberos en las labores de rescate y evacuación. Ante el problema mencionado anteriormente, se busca solapar la visión térmica de una cámara infrarroja y la visión del ojo humano, ambas a través de unas gafas de realidad aumentada. Mediante una interfaz de usuario que interpreta el gesto de los dedos se interactuaría con la aplicación. Se podría controlar distintas funciones, como: la activación y desactivación de la cámara térmica, la comunicación inalámbrica con el resto del equipo del servicio de bomberos o la obtención de información sobre el entorno en cuanto a la concentración de gases tóxicos en el aire.

En este documento, se realiza un análisis sobre los componentes necesarios para desarrollar la interfaz. Además, se diseña un prototipo basado en la clasificación mediante el uso de redes neuronales de la señal recibida del acelerómetro al movimiento de los dedos. Se realiza un breve análisis de las tecnologías de comunicación inalámbrica de corta distancia, baja potencia y alta velocidad de transmisión, con el fin de comunicar el acelerómetro con el dispositivo central y visualizar la interfaz.

Finalmente, se expone una conclusión del trabajo y se proponen mejoras que permitan obtener una interfaz fiable y baja latencia para esta aplicación de realidad aumentada.

Abstract

Smoke generated by fire at buildings makes more difficult the visibility of firefighters and their labours to rescue or evacuate people. There is a solution for this problem and is overlapping infrared thermal vision of a camera and human eye vision, both through augmented reality glasses. By means of an interface that process finger movement, we can interact with the application and control different menus, like: activation and deactivation of the thermal camera, wireless communication with the rest of the fire service team or obtaining information about the environment in a concentration of toxic gases in the air.

In this document, an analysis has been made on the components needed to develop the interface. In addition, a prototype has been designed based on the classification using neural networks in response of the accelerometer to the movement of the fingers. It has been made a brief analysis of short distance wireless communication technologies, low-power and high-speed transmission, to communicate the accelerometer with the central device and display the interface.

Finally, as a conclusion it is proposed improvements to obtain a reliable interface and low latency for this application of augmented reality.

Resum

El fum generat per incendis en recintes tancats dificulta la visibilitat del servei de bombers i les tasques de rescat i evacuació. Davant aquest problema, es busca solapar la visió tèrmica de una càmera infraroja i la visió de l'ull humà, dues mitjançant ulleres de realitat augmentada. A través, de una interfície que interpreta els gestos dels dits podríem interactuar amb la aplicació. Es poden controlar diferents funcions, com: la activació i desactivació de la càmera tèrmica, la comunicació inalàmbrica amb la resta de l'equip del servei de bombers o la obtenció d'informació sobre l'entorn i la seua concentració de gasos tòxics en l'aire.

En aquest document, es realitza un anàlisi sobre els components necessaris per a desenvolupar la interfície. A més es dissenya un prototip basat en la classificació mitjançant l'ús de xarxes neuronals en resposta de l'acceleròmetre al moviment dels dits. Es realitza un breu anàlisi de les tecnologies de comunicació inalàmbrica de curta distància, baixa potència i alta velocitat de transmissió, que es podria utilitzar per a amb la fi de comunicar l'acceleròmetre amb el dispositiu central i visualitzat la interfície.

Finalment, es realitza una conclusió del treball i es proposen millores que permetran obtenir una interfície fiable i baixa latència per esta aplicació de realitat augmentada.

Índice

Capítulo 1.	Objetivo de este documento	2
Capítulo 2.	Metodología del trabajo	3
2.1	Gestión del proyecto.....	3
2.2	Distribución de tareas.....	3
2.3	Diagrama temporal.....	4
Capítulo 3.	Desarrollo y resultados.....	5
3.1	Realidad aumentada.	5
3.2	Interfaces de usuario para aplicaciones de realidad aumentada.	6
3.3	Investigación de soluciones al objetivo.....	7
3.3.1	Identificación por radiofrecuencia.....	7
3.3.2	Unidades de medición inercial.	9
3.4	Propuesta de diseño.....	9
3.4.1	GY-521.....	10
3.4.2	BlueDuino rev2.	11
3.5	Análisis de las señales.	11
3.5.1	Filtro complementario.....	12
3.5.2	Librería MPU6050.h	12
3.5.3	Módulo, Moving Average Filter y detección por umbral.....	13
3.5.4	Búsqueda de parámetros significativos de la señal.	15
3.6	Redes neuronales artificiales.	16
3.6.1	Tipos de redes neuronales artificiales	17
3.6.2	Aplicación con la señal del acelerómetro.....	20
3.7	Interfaz con Matlab.	24
3.8	Interfaz con GTK.	24
3.9	Comunicación inalámbrica.....	26
3.9.1	Bluetooth Low Energy.	26
3.9.2	WiFi.....	28
Capítulo 4.	Conclusiones y propuesta de trabajo futuro	30
Capítulo 5.	Bibliografía.....	31
Capítulo 6.	Agradecimientos.....	33

Capítulo 1. Objetivo de este documento

El principal objetivo de este documento es describir el control mediante los gestos de los dedos de una interfaz de usuario para una aplicación de realidad aumentada. Este proyecto está vinculado a la colaboración con el sistema de realidad aumentada desarrollado por varios alumnos y dirigido por el profesor José Manuel Mossi García.

Mediante la realidad aumentada se permite a los bomberos visualizar a través del humo la señal térmica infrarroja. Esto se consigue gracias a una cámara térmica y a las gafas EPSON Moverio BT 200, con el objetivo de facilitar las labores de rescate, evacuación y extinción de incendios en recintos cerrados

La necesidad de interactuar con las gafas y de añadir funcionalidades extra en forma de menú, conlleva el desarrollo y diseño de un prototipo de interfaz de usuario controlada por el movimiento de los dedos e integrada mediante acelerómetros en los guantes del agente. Son dos los movimientos que permiten navegar por la interfaz, el toque entre el dedo índice y el dedo pulgar será un movimiento de confirmación en el menú y el doble toque entre ambos será un movimiento de retroceso en el menú.

Es necesario crear una interfaz inmersiva, fácil de controlar y lo más ergonómica posible, esto supone una gran ventaja para el bombero ya que el entorno en el que trabaja se caracteriza por elevadas temperaturas y condiciones extremas de estrés, esto repercute en el esfuerzo físico del agente y en la interacción con el sistema desarrollado. Además, la interfaz debe tener una respuesta rápida y fiable para que se convierta en un complemento ágil para el desarrollo de su trabajo.



Figura 1. Prototipo de una interfaz creada por Joseph Juhnke. Representación virtual de una interfaz de usuario en la mano, que permite controlar un sistema de realidad aumentada para el cuerpo de bomberos.

Capítulo 2. Metodología del trabajo

2.1 Gestión del proyecto

La gestión del proyecto se ha basado en la elaboración de un informe de progreso actualizado mensualmente en el que se marcaban los objetivos, se proponían soluciones, se realizaba una pequeña descripción del trabajo realizado y finalmente se evaluaba y se resolvían dudas con el tutor.

2.2 Distribución de tareas

- Tarea 1.
 - Título: Investigación sobre las distintas soluciones para la interfaz.
 - Descripción: Búsqueda de información y estudio de las distintas posibilidades para la creación del interfaz. Análisis de la tecnología RFID, acelerómetros y giroscopios.
- Tarea 2.
 - Título: Investigación sobre el microcontrolador utilizado.
 - Descripción: Análisis del microcontrolador BlueDuino rev2 con conectividad Bluetooth.
- Tarea 3.
 - Título: Investigación sobre la conectividad entre el microcontrolador y la interfaz.
 - Descripción: Búsqueda de información sobre la conexión entre el microcontrolador y el dispositivo central. Uso de tecnología Bluetooth Low Energy y WiFi
- Tarea 4.
 - Título: Investigación sobre el procesado de la señal del acelerómetro.
 - Descripción: Búsqueda de librerías que estimen las señales de yaw, pitch y roll. Análisis de la detección de pulsación en la mano mediante la detección de un umbral. Filtrado sencillo e integración en el microcontrolador.
- Tarea 5.
 - Título: Investigación sobre redes neuronales y su uso para clasificar patrones.
 - Descripción: Análisis de la estructura y de los tipos de redes neuronales.
- Tarea 6.
 - Título: Creación de interfaz.
 - Descripción: Exportación de redes neuronales a código C y ejecución.

2.3 Diagrama temporal

Las tareas se han ido realizando durante todo el año y en el diagrama temporal de la Tabla 1 se puede observar el tiempo dedicado a cada una.

<i>Tareas</i>	<i>Octubre</i>	<i>Noviembre</i>	<i>Diciembre</i>	<i>Enero</i>	<i>Febrero</i>	<i>Marzo</i>	<i>Abril</i>	<i>Mayo</i>	<i>Junio</i>
<i>1</i>	■	■	■	□	□	□	□	□	□
<i>2</i>	■	■	■	□	□	□	□	□	□
<i>3</i>	■	■	■	□	□	□	□	□	□
<i>4</i>	□	□	□	■	■	■	□	□	□
<i>5</i>	□	□	□	□	□	□	■	■	■
<i>6</i>	□	□	□	□	□	□	□	■	■

Tabla 1. Diagrama temporal.

Capítulo 3. Desarrollo y resultados

3.1 Realidad aumentada.

La realidad aumentada es la combinación de información digital con datos del entorno en tiempo real. A diferencia de la realidad virtual en la cual se crea un ambiente totalmente artificial, la realidad aumentada superpone lo que el usuario puede percibir con nueva información del entorno, como podría ser, la visión térmica, nivel de concentración de gases tóxicos o posicionamiento indoor.

Una de las primeras aplicaciones de realidad aumentada fue la superposición de datos de jugadores de fútbol americano durante la retransmisión de partidos en televisión, de esta forma, el espectador podía observar las estadísticas del jugador junto con la señal del partido en tiempo real. Actualmente, las gafas o la tecnología de head-up display son los productos más conocidos de la realidad aumentada, tecnología que ha sido usada en otros muchos sectores como el sanitario, militar, salvamento, turismo o marketing. [29]

Las gafas están dominando el sector de la realidad aumentada y un indicador de ello, es el desarrollo y la producción de estos dispositivos por las empresas tecnológicas más importantes del mundo, algunos ejemplos son las HoloLens de Microsoft, las Google Glasses o las Epson Moverio.

La estructura de un sistema de gafas de realidad aumentada está compuesta por: el diseño, el sistema de visualización de la aplicación, la unidad de procesamiento, los sistemas de adquisiciones de datos y los módulos de conectividad.

El sistema de visualización de la aplicación es la parte más importante de su diseño, la combinación de la física óptica con materiales semitransparentes hace que se pueda superponer la información procedente de las gafas con lo que el usuario en ese momento está viendo. Como se puede ver en la Figura 2 el display proyecta la imagen que recibe de la unidad de procesamiento en un espejo y éste a su vez en un cristal con unas dimensiones específicas que marcan la distancia focal y, por lo tanto, dónde se reflejará.

La unidad de adquisición de datos permite registrar información del entorno y acondicionarlos para su visualización en las gafas. Son muchos los dispositivos comerciales que integran comunicación inalámbrica a través de Bluetooth Low Energy y WiFi en sus aplicaciones, además, se puede disponer de sistemas GPS para tener la posición en tiempo real del usuario. Uno de los periféricos que más se utiliza son las cámaras, que permiten grabar lo que en ese momento está viendo el usuario al mismo tiempo que le permite ver información adicional sobre la imagen real.

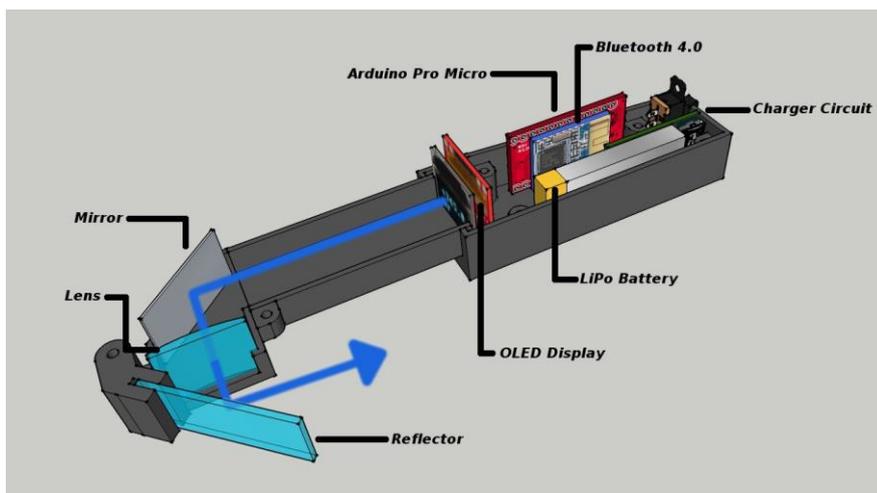


Figura 2. Esquema de unas gafas de realidad aumentada. [30]
 Diagrama de propagación de la imagen de un display hasta el cristal de unas gafas de realidad aumentada. Sistema diseñado por Alan Mauer.

En colaboración con el proyecto descrito en este documento, un grupo de varios alumnos dirigidos por el profesor José Manuel Mossi García han desarrollado un sistema de realidad aumentada que combina las gafas Epson Moverio BT-200 y la cámara térmica Seek Thermal, con el objetivo de poder visualizar en las gafas la señal térmica infrarroja y que el servicio de bomberos pueda desarrollar su trabajo en recintos cerrados con grandes cantidades de humo donde no hay suficiente visibilidad.

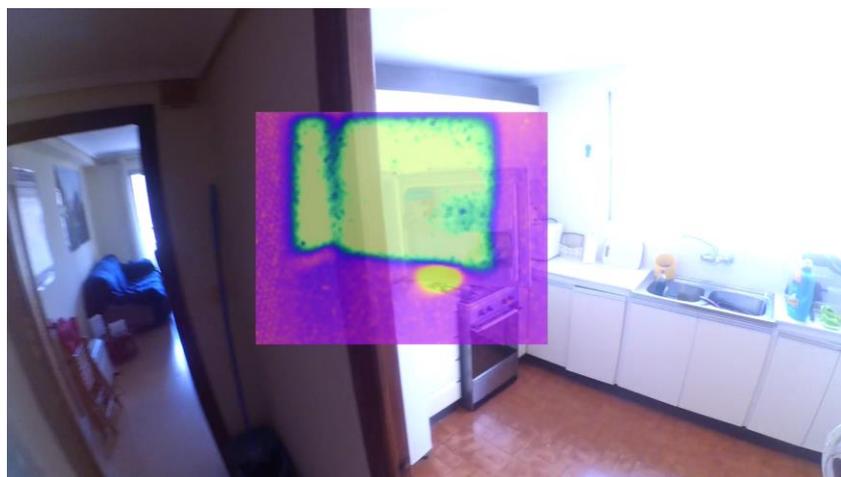


Figura 3. Visión térmica en las gafas de realidad aumentada.
 Superposición de la imagen térmica infrarroja y de la imagen de una cámara GoPro. En el recuadro central se puede apreciar la imagen térmica de la cámara Seek Thermal, en una cocina donde los colores fríos representan zonas de baja temperatura (congelador abierto) y los cálidos zonas de alta temperatura (vitrocerámica encendida).

3.2 Interfaces de usuario para aplicaciones de realidad aumentada.

El mercado de la realidad aumentada y virtual se encuentra en pleno auge, tanto por la revolución que supone para la sociedad como para las nuevas aplicaciones que se pueden desarrollar y que suponen una innovación. Los sectores industrial, sanitario y militar emplean la realidad aumentada para optimizar los procesos y obtener mejores resultados en tiempos menores.



Figura 4. Leap motion. Ejemplo de interfaz en tiempo real utilizando la tecnología leap motion.

Estimación de la posición, pero tienen el inconveniente de que se necesita visión directa entre las manos y la cámara, hecho que en nuestro entorno de aplicación no se cumple ya que en muchos incendios la densidad del humo es tan intensa, que la visibilidad entre las manos y la cabeza se ha reducido completamente.

Por otra parte, tenemos una tecnología, en la que necesitamos ponernos un guante que contiene marcadores y mediante una cámara externa se puede posicionar la mano en tiempo real la interfaz gracias a la relación que tiene con la imagen. [5][6] La mayoría de estos dispositivos, están basados en tracking óptico y, por lo tanto, necesitan de visibilidad directa con el objeto que contiene los marcadores.

Existen interfaces basados en tecnología flexible, como los sensores de galgas flexibles. Estos dispositivos son relativamente económicos y fáciles de manejar, pero los estudios de su uso a largo plazo indican que sus características eléctricas en función de la flexión varían y se necesita un recalibrado frecuente, lo cual desaconseja su uso en un entorno como el nuestro que exige una alta fiabilidad.



Figura 5. Hi5 glove. Ejemplo de aplicación de interfaz en tiempo real utilizando la tecnología de tracking óptico.

3.3 Investigación de soluciones al objetivo.

Tras el análisis del mercado y de las características de la interfaz que se va a desarrollar, se buscan distintas tecnologías que proporcionen una solución al objetivo, se analizan las ventajas e inconvenientes de cada una y se opta por desarrollar un prototipo con la opción que dote al sistema de las características de fiabilidad y de baja latencia.

3.3.1 Identificación por radiofrecuencia.

Una de las soluciones que se plantea es la implementación de la tecnología RFID (Radio Frequency Identification) utilizada en muchas aplicaciones en las que es necesario la identificación, como los sistemas de control y acceso a recintos cerrados.

La tecnología RFID es utilizada en varias bandas de frecuencias, se opera principalmente en las bandas de LF (Low Frequency), HF (High Frequency) y UHF (Ultra High Frequency) lo que marcará sus características de operación, es decir, a mayor frecuencia mayor interferencia con líquidos, metales u otras partículas, sin embargo, se consiguen velocidades de lectura mucho mayores. [8]

La banda de baja frecuencia o LF, se encuentra en el rango de 30kHz a 300kHz. Los dispositivos comerciales que podemos encontrar y que trabajan en esta banda tienen como frecuencia central 125kHz, esto permite que la lectura se pueda realizar en un rango de hasta 0.5 metros.

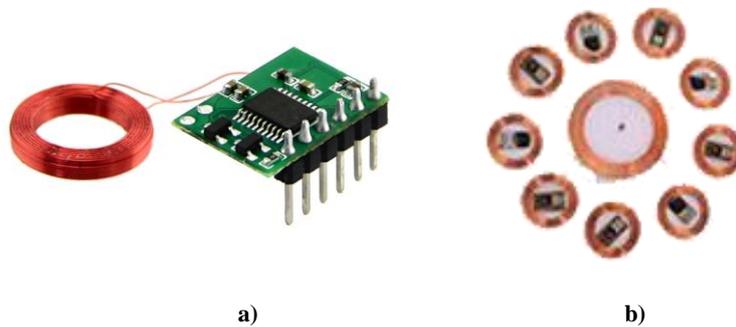


Figura 6. Dispositivos RFID a 125kHz. La figura a) representa un transmisor RFID a 125kHz de dimensiones reducidas y con sistema UART de comunicación. La figura b) se corresponde con un receptor en forma de pegatina y compatible con el transmisor.

Los dispositivos comerciales que podemos encontrar en la banda de alta frecuencia funcionan a 13.56MHz. La distancia máxima de identificación es de un metro, lo que permite utilizarlos en control de acceso, tracking o pagos con tarjeta.

Finalmente, tenemos los dispositivos que operan en la banda de UHF y SHF que permiten obtener la identificación de un dispositivo receptor hasta distancias de 10 metros, siendo los más utilizados en AVI (Automatic Vehicle Identification).

RFID FREQUENCY BAND / SPECTRUM ALLOCATIONS			
RFID FREQUENCY BAND	FREQUENCY BAND DESCRIPTION	TYPICAL RANGE	TYPICAL RFID APPLICATIONS
125-134.2 kHz 140-148.5 kHz	Low Frequency LF	Up to ~ 1/2 metre	These frequencies can be used globally without a license. Often used for vehicle identification. Sometimes referred to as LowFID.
13.553 - 13.567 MHz	High Frequency HF	Up to ~ 1 metre	These RFID frequencies are typically used for electronic ticketing, contactless payment, access control, garment tracking, etc
858 - 930 MHz	Ultra High Frequency UHF	1 to 10 metres	These RFID frequencies cannot be accessed globally and there are significant restrictions on their use. When they are used, it is often used for asset management, container tracking, baggage tracking, work in progress tracking, etc. and often in conjunction with Wi-Fi systems.
2.446 - 2.454GHz	Super High Frequency SHF	3 metres upwards	These RFID frequencies are used for long range tracking and with active tags, RFID and AVI (Automatic Vehicle Identification).

Tabla 2. Tabla resumen de las frecuencias a las que opera el RFID. [8] En esta tabla se puede observar el resumen de todas las frecuencias en las que operan los sistemas de identificación por radiofrecuencia y las aplicaciones más comunes.

La ventaja principal de esta tecnología es su reducido coste y su facilidad de implementación, ya que no es necesario la integración de algoritmos en microcontroladores para detectar el

identificador, a diferencia de los sistemas de medición inercial. Durante la investigación y el análisis de esta tecnología, se plantearon varios inconvenientes, entre ellos, se encuentra el tamaño, la distancia de detección y la respuesta con el entorno. Es difícil encontrar en los dispositivos comerciales actuales un tamaño adecuado de transmisor que fuese lo más cómodo y sencillo de integrar en el sistema del guante.

3.3.2 Unidades de medición inercial.

Los acelerómetros son instrumentos que están pensados para realizar medidas de aceleración o vibración. Algunos de estos sistemas incluyen giroscopios y magnetómetros que se usan como complemento a la aceleración y permiten obtener la posición exacta del dispositivo mediante algoritmos matemáticos.

Las unidades de medición inercial permiten obtener resultados de velocidad y orientación gracias a la combinación de acelerómetros, giroscopios y magnetómetros. La respuesta de las unidades se fundamenta en la microelectrónica de los MEMS (Micro-Electro-Mechanical Systems) que se caracterizan por unas dimensiones de decenas de micrómetros y que permiten medir la inercia de los dispositivos. Compuestos por una masa móvil y formando pequeñas estructuras similares a las de un condensador, se consigue que cualquier variación espacial cambie las distancias entre los metales que forman el condensador y se modifique la capacidad, obteniendo así una relación entre las variables de aceleración y capacidad. [14]

La gran variedad comercial y sus reducidas dimensiones lo hacen ideal para el objetivo que nos hemos planteado. El inconveniente en un primer análisis reside en encontrar un algoritmo que proporcione una medida fiable del posicionamiento espacial del dispositivo.

En la actualidad, empresas como X-Sens comercializan sistemas de posicionamiento 3D con muy poco error gracias a la tecnología XKF3, el elevado coste que suponen estos dispositivos hace que se plantee buscar sistemas de medición inercial más económicos como el MPU6050.

3.4 Propuesta de diseño.

Tras el análisis de las tecnologías que mejor se adaptan a los objetivos planteados en este documento, se opta por desarrollar un prototipo con el acelerómetro y giroscopio MPU6050 integrado en la placa GY-521 y conectado al microcontrolador BlueDuino rev2. Ambos dispositivos se integran en un guante para poder simular los resultados de una manera lo más parecida a la realidad. Se recubren todos los dispositivos con aislamiento termo retráctil para proteger las conexiones y ofrecer una mayor resistencia a golpes.

La conexión entre la placa GY-521 y el microcontrolador se realiza mediante el protocolo I2C, el cual permite conectar varios periféricos con un sistema central. A diferencia del protocolo de comunicación serie, el I2C es síncrono funcionando en nuestro caso a una frecuencia de 200kHz. Aunque no es el protocolo que mejor tasa ofrece, su versatilidad para comunicarse con varios dispositivos y el reducido número de pines, lo hace destacar sobre el conjunto de protocolos de comunicación de corta distancia. [25]

La comunicación entre la interfaz y el microcontrolador se realiza a través de un protocolo de comunicación serie, ya que permite transferir datos con el ordenador. La tasa elegida para la comunicación es de 9600 baudios y la conexión se realiza mediante un cable USB.

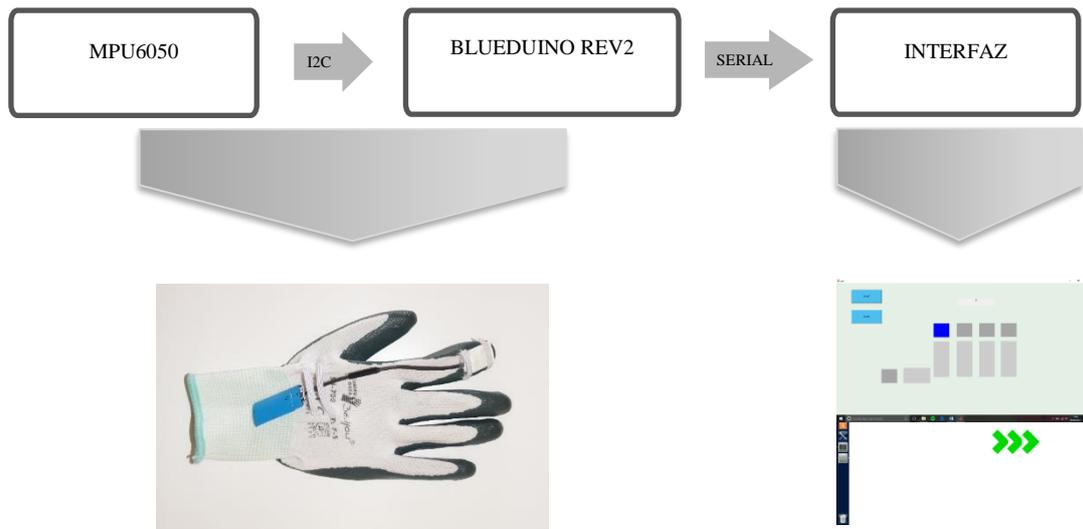


Figura 7. Diagrama de bloques del prototipo. Esta figura representa los bloques del sistema desarrollado junto con el tipo de comunicación que los interconecta. Incluye imágenes del guante que procesa los datos del MPU6050 y de la interfaz desarrollada en Matlab y C.

3.4.1 GY-521.

Se trata de un dispositivo que integra un acelerómetro y un giroscopio, gracias al circuito integrado MPU6050, el cual, se puede comunicar mediante el protocolo I2C. En el circuito integrado hay una unidad digital de procesamiento de movimiento que reduce la carga del microcontrolador. También ofrece la posibilidad de combinar varios acelerómetros gracias al pin AD0 que nos permite realizar una multiplexación entre varios acelerómetros.

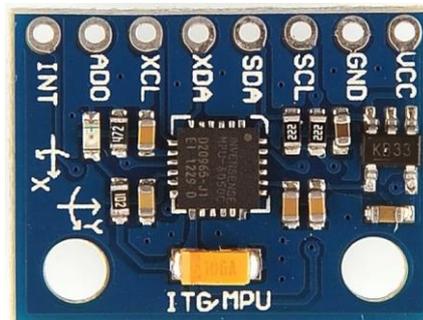


Figura 8. GY-521. Dispositivo integrado en el extremo del dedo índice y que mide la aceleración y velocidad angular gracias al circuito integrado MPU6050.

A continuación, se describen los pines utilizados para la conexión con el microcontrolador.

- SDA: Señal de datos del protocolo I2C.
- SCL: Señal de sincronismo del protocolo I2C
- VCC: Tensión de alimentación para el dispositivo, en este caso, es de 3.3V.
- GND: Tensión de referencia.
- AD0: Pin para multiplexar varios acelerómetros. Este pin se encuentra en la placa GY-521 y permite ser utilizado como señal de CS (Chip Select). Si el pin AD0 está conectado a masa la dirección del protocolo I2C será 0x68, mientras que si está conectado a 3.3V la dirección será 0x69. [26] La dirección del protocolo I2C se configura en el microcontrolador, por defecto, la librería MPU6050.h la tiene configurada a la dirección 0x68.

El circuito integrado MPU6050 permite configurar varios parámetros, entre ellos, la sensibilidad del acelerómetro o del giroscopio.

ACCELEROMETER OUTPUT		
AFS_SEL	FULL SCALE RANGE	LSB SENSITIVITY
0	±2g	16384 LSB/g
1	±4g	8192 LSB/g
2	±8g	4096 LSB/g
3	±16g	2048 LSB/g
Valores de 16bits en complemento a dos		

Tabla 3. Valores de sensibilidad del acelerómetro. [12][13]

GYROSCOPE OUTPUT		
FS_SEL	FULL SCALE RANGE	LSB SENSITIVITY
0	± 250 °/S	131 LSB/°/S
1	± 500 °/S	65.5 LSB/°/S
2	± 1000 °/S	32.8 LSB/°/S
3	± 2000 °/S	16.4 LSB/°/S
Valores de 16bits en complemento a dos		

Tabla 4. Valores de sensibilidad del giroscopio. [12][13]

3.4.2 BlueDuino rev2.

Este microcontrolador contiene el circuito integrado ATmega32U4. Necesita una alimentación de más de 3.3V y funciona a una frecuencia de 8MHz, además integra el módulo CC2540 de Texas Instruments que permite a la placa disponer de conectividad Bluetooth Low Energy.

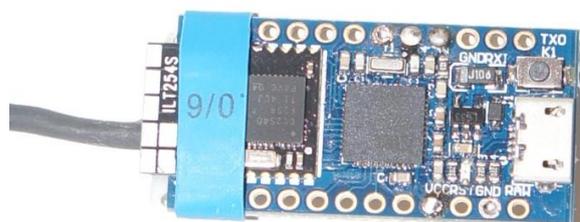


Figura 9. BlueDuino rev2. Microcontrolador utilizado para interpretar los valores de aceleración y velocidad angular del MPU6050.

Incluye un regulador de tensión para unos valores de entrada comprendidos entre los 4V y los 12V y un puerto USB para una comunicación serie con el dispositivo.

El circuito integrado CC2540 lleva integrado el firmware ZeroBeacon, el cual, permite utilizar comandos AT [22] para configurar el modo de comunicación, la velocidad de transmisión, la potencia o el nombre del dispositivo.



Figura 10. Orientación de los ejes del acelerómetro. Esta figura representa la posición del MPU6050 en el guante y la orientación de los ejes en los que se produce la medida.

3.5 Análisis de las señales.

Tras estudiar el hardware que se utiliza para crear el prototipo se procede al análisis de las señales de aceleración y de velocidad angular para así poder detectar el movimiento del dedo y, por lo tanto, reconocer así si se ha producido un toque o un doble toque entre los dedos índice y pulgar.

El primer objetivo es el de obtener el desplazamiento del dedo con el acelerómetro integrado en el extremo de éste, para ello, se analizan los distintos algoritmos que existen y su respuesta. El filtro complementario es uno de los algoritmos que permite obtener una orientación angular gracias a su integración en el microcontrolador.

3.5.1 Filtro complementario

El filtro complementario es sencillo de implementar en un microcontrolador y proporciona la posición angular gracias a la combinación de la señal del giroscopio y del acelerómetro.

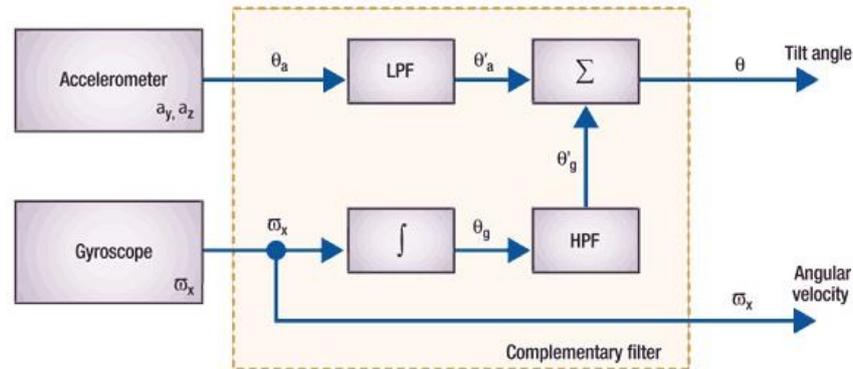


Figura 11. Esquema del filtro complementario. Diagrama de bloques del algoritmo del filtro complementario, donde obtenemos la velocidad angular y la orientación del dispositivo MPU6050.

La respuesta de éste muestra algunos errores debido a las señales del acelerómetro y del giroscopio, además de que se ven distorsionados por cualquier fuerza externa. El problema en el giroscopio es la deriva a lo largo del tiempo y el no retorno a cero producidos por la integración de la señal. [21]

La combinación de las derivas del acelerómetro y del giroscopio se solucionan en el filtro complementario mediante la integración y el filtrado de la velocidad angular y una compensación utilizando la señal filtrada del acelerómetro.

$$\text{angle} = 0.98 * (\text{angle} + \text{gyrData} * \text{dt}) + 0.02 * \text{accData}$$

El filtro ha sido usado para calcular la orientación y establecer un umbral que proporcione una respuesta los toques simple y doble. Surgen dos inconvenientes con respecto a la idea anterior, el primero es la aparición de falsos positivos causados por movimientos normales de la mano no asociados al gesto del toque y el segundo es la dificultad de detectar un doble toque debido a la integración de la señal del giroscopio, para ello, hay que tener en cuenta la dimensión temporal de la señal.

3.5.2 Librería MPU6050.h

Como se ha mencionado en el apartado anterior, el principal problema con el filtro complementario es la realización de movimientos fuera de la posición de equilibrio lo que genera error en el cálculo de la orientación.

Jeff Rowberg ha creado una librería donde podemos configurar el dispositivo MPU6050, hacer uso de la unidad digital de procesamiento de movimiento o DMP, obtener los valores de posición angular y calcular los valores de la aceleración eliminando la presencia del vector de gravedad.

Con el objetivo de hacer un uso optimizado de los recursos del microcontrolador para obtener las señales del dispositivo MPU6050, se ha integrado en el BlueDuino rev2 la librería MPU6050.h creada por Jeff Rowberg. [11]

Después de analizar la librería, se comprueban los parámetros básicos de las señales de aceleración y velocidad angular, así como, la frecuencia de muestreo, la sensibilidad y la tasa de transmisión.

PARÁMETRO	VALOR
Frecuencia de muestreo	67 Hz
Sensibilidad	8192 LSB/g
Velocidad de transmisión	9600 baudios

Tabla 5. Valores de los parámetros configurables de la librería MPU6050.

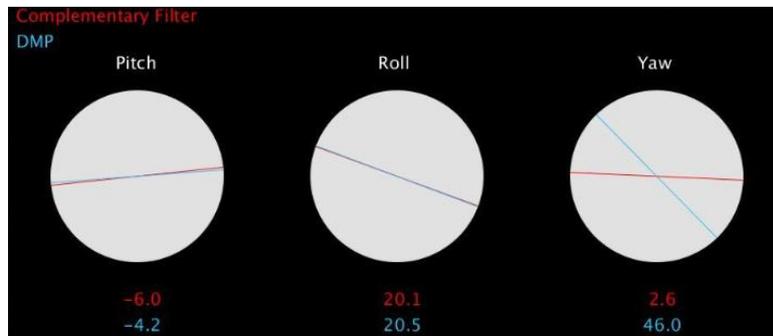


Figura 12. Comparativa entre el filtro complementario y la librería MPU6050.h. Interfaz en Processing para la comparación entre los valores de orientación angular calculados mediante el filtro complementario y la librería MPU6050.h.

Se consigue comparar la orientación calculada mediante la librería MPU6050.h y el filtro complementario. En base a la respuesta de ambos se decide utilizar la librería MPU6050.h, ya que se puede tener un control más exacto de los valores de posición angular y de la configuración del dispositivo. Tras varias pruebas con las señales de orientación angular, yaw, pitch y roll, no se encuentra forma de detectar el patrón de movimiento. Se opta por buscar una solución basada en el cálculo del módulo de las tres aceleraciones, la realización de un filtrado paso bajo y la clasificación del patrón mediante un umbral experimental.

3.5.3 Módulo, Moving Average Filter y detección por umbral

Para realizar la detección por umbral, se utiliza el módulo de la aceleración que se calcula con cada muestra que nos llegue del acelerómetro.

$$acc = \sqrt{accx^2 + accy^2 + accz^2}$$

Se filtra digitalmente la señal para reducir las variaciones de alta frecuencia, ya que sólo interesa conservar las componentes de baja frecuencia que son las que proporcionarán información sobre la presencia de movimiento. Se analizan distintos tipos de filtros y se opta por uno que se pueda integrar en el BlueDuino rev2, el Moving Average Filter.

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i + j]$$

El filtro de media o Moving Average Filter [1] es un filtro digital paso bajo que realiza la media de una ventana deslizante de muestras. La frecuencia de corte discreta está relacionada con el tamaño de la ventana que se utiliza para calcular la media y sigue la relación indicada abajo, donde M es el tamaño de la ventana.

$$f_d = \frac{1}{M}$$

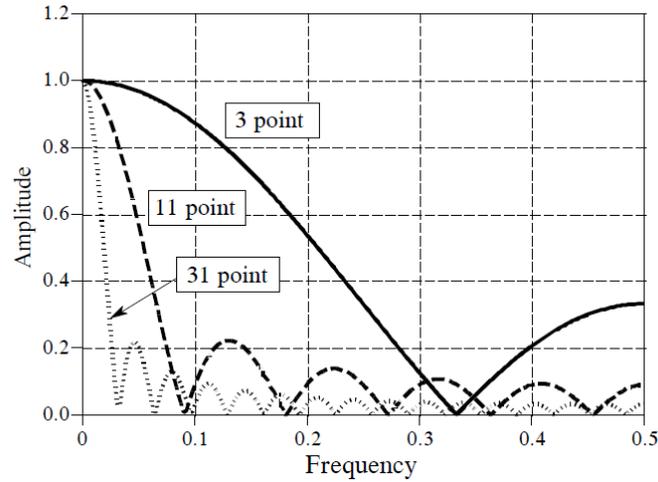


Figura 13. Respuesta en frecuencia Moving Average Filter.
 Función de transferencia de tres tamaños de ventana del filtro digital Moving Average Filter.

Implementando el filtro en el microcontrolador y se compara la señal de salida del filtro creado en éste y la generada por Matlab. Tras la comprobación, se usa la comunicación por el puerto serie y la representación en tiempo real en ventanas de 100 muestras para visualizar la señal y poder extraer conclusiones a los distintos movimientos.

De forma experimental se ha elegido una frecuencia de corte para que la señal de doble toque se pudiera distinguir sobre la de uno. Después de realizar varias pruebas con el movimiento del dedo, se establece un umbral experimental para comprobar que si la señal supera dicho umbral se envíe por el puerto serie el tipo de movimiento realizado.

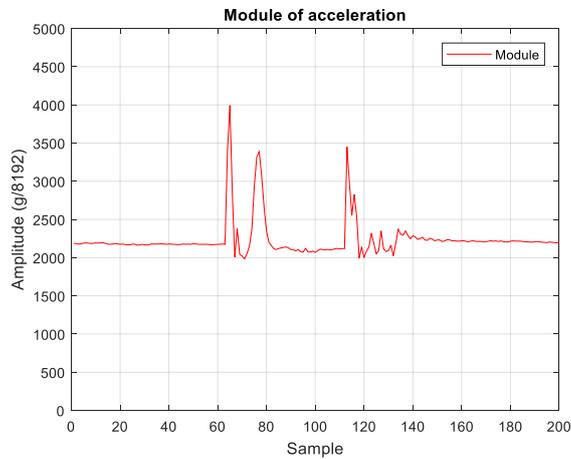


Figura 14. Módulo de la aceleración. Cálculo del módulo de la aceleración en el BlueDuino rev2.

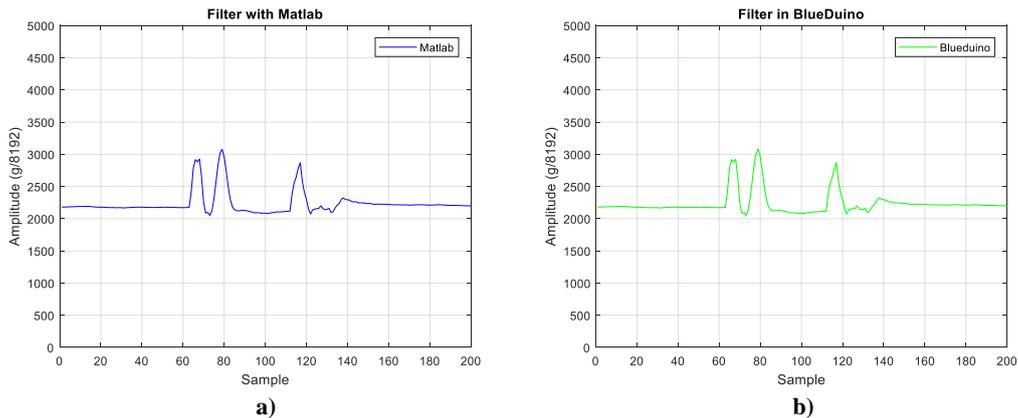


Figura 15. Respuesta del filtrado usando el BlueDuino y Matlab.

En la figura a) se puede observar el filtrado de la señal de la Figura 14 utilizando el filtro de media de Matlab, mientras que en la figura b) se puede observar la salida del filtro en el BlueDuino rev2.

Se encuentra el inconveniente de que siguen produciéndose falsos positivos después de realizar varias pruebas debido a movimientos aleatorios o golpes involuntarios. Se buscan parámetros que nos permitan diferenciar el toque y los movimientos aleatorios, por eso, se investiga sobre características de la señal tanto en el dominio temporal como en el de la frecuencia.

El valor de pico de la señal viene determinado por la fuerza y la rapidez con la que cada persona realice el movimiento entre el dedo índice y pulgar. Se busca información para generar una respuesta rápida y fiable y se analizan los sistemas basados en detección por umbral adaptativo o en detección de picos.

La media, el valor rms o información sobre parámetros en el espectro, son algunos de los atributos con los que se podría determinar el movimiento de la mano. En el siguiente capítulo se aborda este aspecto y se analiza las señales en el dominio de la frecuencia.

3.5.4 Búsqueda de parámetros significativos de la señal.

Para tener una visión más precisa sobre el tipo de movimiento que queremos detectar se analizan las señales detenidamente y se calculan los atributos que la hacen única. [2]

Se puede extraer información de tres atributos:

- Atributos en el tiempo: Las características en el tiempo extraen información sobre la amplitud de la señal, la duración de la actividad o la distancia entre máximos y mínimos.

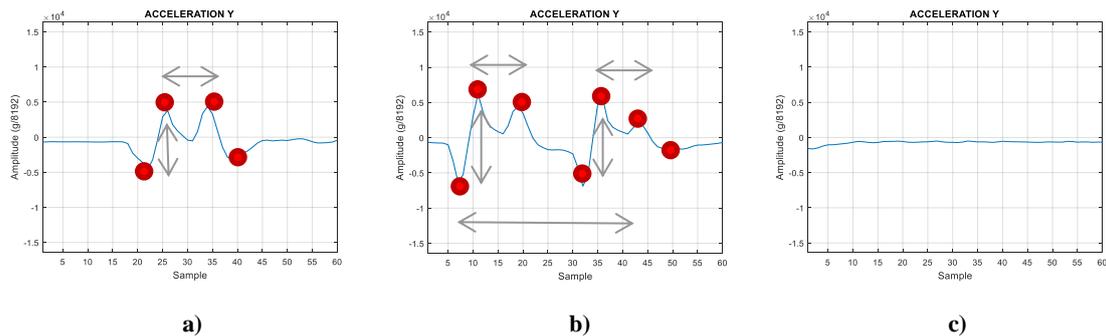


Figura 16. Representación temporal de los movimientos de toque, doble toque y nada. La figura a) se corresponde con la aceleración procedente del BlueDuino rev2 a un movimiento de toque entre el dedo índice y el dedo pulgar, mientras que la figura b) se corresponde al movimiento de doble toque. En la figura c) se puede observar la señal cuando no se produce ninguno de los movimientos mencionados anteriormente.

- Atributos en frecuencia: La representación de la señal en frecuencia nos permite buscar una componente espectral dominante que distinga las señales. Es sencillo diferenciar donde hay ausencia de movimiento y donde se produce un toque.

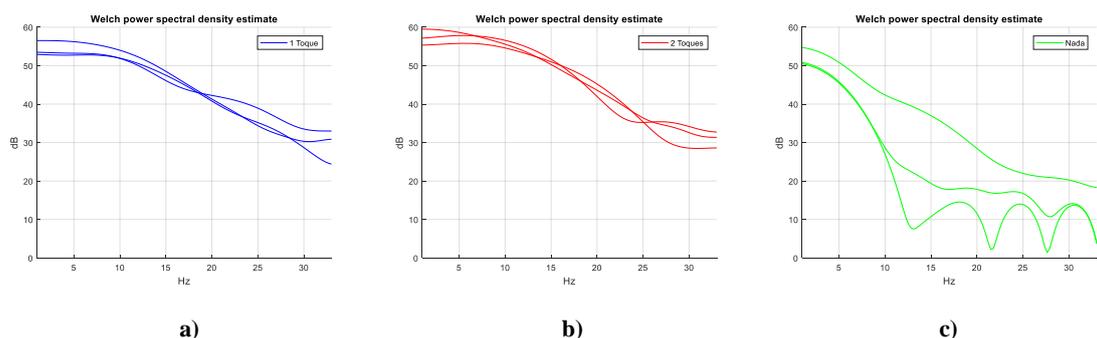


Figura 17. Representación en el dominio de la frecuencia de los movimientos de toque, doble toque y nada. La figura a) representa el espectro en frecuencia de la señal de un toque y la figura b) el de doble. El espectro en frecuencia de los movimientos de nada se encuentra en la figura c).

- Atributos derivados: Estas características hacen referencia a cálculos sobre parámetros de la señal, por ejemplo, valor medio, valor rms.

El cálculo de la estimación de la densidad espectral a través del método de Welch, la búsqueda de máximos y mínimos en cada ventana o el cálculo de los atributos derivados son procesos que requieren de una gran carga computacional, por lo que se busca en las redes neuronales reducir el procesado e introducir directamente la ventana con los valores de la señal.

3.6 Redes neuronales artificiales.

Las redes neuronales artificiales representan una parte del conjunto de sistemas de inteligencia artificial actuales. Se trata de sistemas formados por elementos simples e interconectados que procesan información y generan una respuesta dinámica.

La característica principal de las redes neuronales es la capacidad de aprender de la experiencia, por ello, estos sistemas reciben el nombre de redes neuronales, debido a su semejanza con la actividad del cerebral, en cuanto al aprendizaje heurístico. [15]

Existen varias aplicaciones en las que las redes neuronales se ven involucradas, predicción, filtrado o clasificación. En este proyecto se han utilizado las redes neuronales para la clasificación de patrones de movimiento de los dedos, en concreto, el desplazamiento del dedo índice al pulgar.

El uso de redes neuronales artificiales está vinculado a situaciones donde hay una gran dificultad de obtener una solución a algunos problemas. Actualmente existen varios tipos de redes neuronales que se caracterizan por su estructura y por sus algoritmos de entrenamiento. A continuación, se describen algunas de las más relevantes y que podrían ser útiles para conseguir el objetivo del proyecto.

3.6.1 Tipos de redes neuronales artificiales

3.6.1.1 Feedforward Neural Network

Las redes del tipo feedforward son redes con una estructura sencilla y que están formadas por una capa de entrada, una o más capas ocultas y una capa de salida. Las capas ocultas están formadas por unidades simples denominadas perceptrones que interconectan todas las capas creando una estructura simple y extrayendo los atributos más relevantes de la entrada. [27]

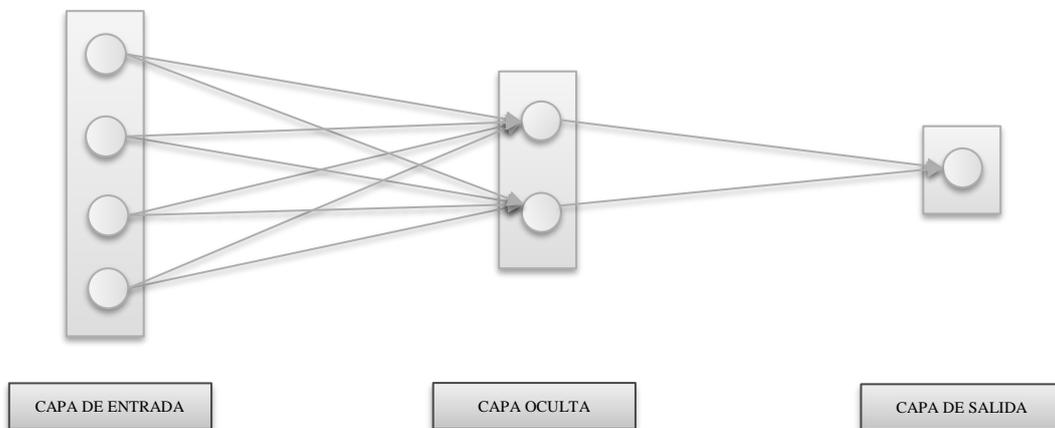


Figura 18. Esquema de la red neuronal multilayer. El esquema de esta figura representa una estructura sencilla de red neuronal formada por cuatro neuronas en la capa de entrada, dos en la capa oculta y una en la capa de salida.

El perceptrón permite interconectar todas las capas y generar una respuesta dinámica a una serie de entradas. Se compone principalmente de un proceso multiplicativo, un sumatorio y una función de transferencia que genera un valor de salida.

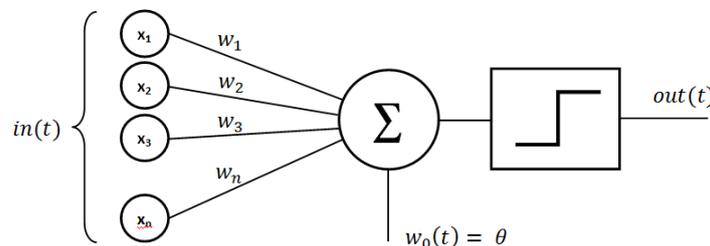


Figura 19. Estructura del perceptrón. El diagrama de la figura representa los bloques que forman un perceptrón. Se puede observar la multiplicación de la entrada por los pesos sinápticos, el valor de la constante, el sumatorio y la función de transferencia tipo escalón.

Como todos los tipos de redes neuronales, es necesario aplicar un algoritmo de aprendizaje que modifique los pesos y los valores constantes de los perceptrones. El entrenamiento de las redes multilayer está basado en el algoritmo de back-propagation que se concreta en tres fases:

- Initialization phase: Los parámetros de los perceptrones de la capa oculta son inicializados con valores aleatorios al principio del entrenamiento de la red neuronal.
- Forward phase: Se calcula el error entre el valor de salida deseable y el valor que genera la red.
- Backward phase: El error de la fase anterior se utiliza para modificar los pesos y los valores constantes de las capas ocultas.

El algoritmo de aprendizaje de back-propagation es simple de implementar y computacionalmente eficiente. Sin embargo, el inconveniente es la no convergencia y la poca agilidad de la red para encontrar una solución, sobre todo, en situaciones en las que es necesario un elevado número de neuronas.

Estas son algunas de las soluciones para mejorar el rendimiento de las redes.

- Reordenar de forma aleatoria las muestras de entrenamiento de la red.
- Comenzar por muestras que sean fáciles de procesar e ir aumentando progresivamente la dificultad.
- Procesar, si es posible, las muestras antes de que entren a la red, filtrando o acondicionando la señal.

Este tipo de redes tienen características ideales para ser utilizada en nuestro proyecto gracias a que disponen de una estructura sencilla y posibilitan su integración en un microcontrolador.

3.6.1.2 Time-delay Neural Networks.

El tiempo es una dimensión a tener en cuenta en el diseño de redes neuronales, existe un tipo de red que utiliza esta variable tanto de forma implícita como explícita. Un tipo de estructura de estas redes es el de añadir una estructura short-term memory en la entrada de una red neuronal estática, como es la multilayer perceptron, recibe el nombre de Tapped-Delay-Line Memory y consiste en un número p de bloques de retardo y $(p + 1)$ entradas como se puede observar en la Figura 20. Esta arquitectura de red ha servido de base para desarrollar la red que clasificará los patrones de movimiento que se han establecido en el objetivo.

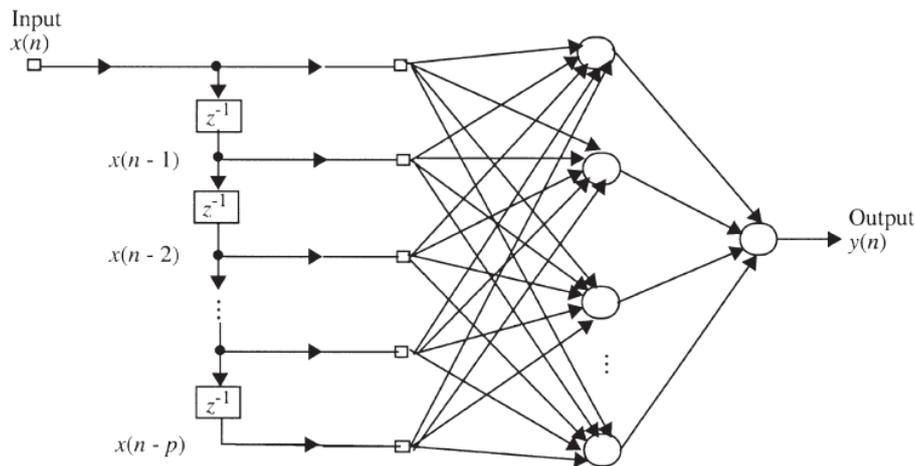


Figura 20. Tapped-Delay-Line Memory. Tipo de estructura de red neuronal donde la variable temporal está considerada de forma implícita en el entrenamiento.

3.6.1.3 Convolutional Neural Networks.

Las redes neuronales de convolución son unas de las más populares en las aplicaciones de inteligencia artificial debido a que están perfectamente diseñadas para reconocimiento de imágenes. Comparado con las redes del tipo feedforward, las redes de convolución tienen

menos conexiones y parámetros, por lo que suelen ser más fáciles de entrenar. [18] La estructura de este tipo de redes simplifica la búsqueda de patrones gracias a la subdivisión de la imagen en regiones.

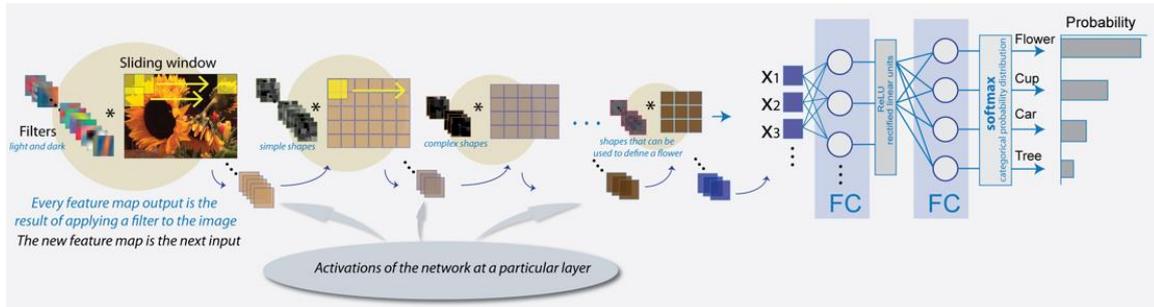


Figura 21. Diagrama de flujo de una imagen en una red de convolución. Estructura de red neuronal de convolución de Matlab, donde se representa el recorrido de una imagen por las capas que forman la red.

La arquitectura está formada por varias capas interconectadas y con una funcionalidad específica. [16]

- Capa de entrada: Define las dimensiones y el número de componentes de color de las imágenes que entrarán a la red neuronal. En esta capa se almacena el valor original de los píxeles de cada imagen.
- Capa de convolución: Se produce una interconexión entre las regiones de la imagen, se realiza un escaneo de toda la imagen y se extraen una serie de parámetros localizados. El entrenamiento en esta capa consiste en la multiplicación de cada región por un vector de pesos, es decir, se realiza un filtrado recorriendo toda la imagen.
- Capa ReLU: Es la siguiente a la etapa de convolución o filtrado y consiste en una función de transferencia no lineal.
- Capa Max-Average Pooling: Esta capa sucede a la ReLU y sirve para disminuir las conexiones de las siguientes.
- Capa de conexión: Combina todas las características aprendidas por las anteriores para finalmente clasificar los patrones.
- Capa de clasificación: Tiene como salida todos los tipos de clasificación que se han utilizado para entrenar la red.

En la actualidad, surgen nuevas aplicaciones para las redes neuronales gracias a la capacidad de síntesis y a la agilidad de entrenamiento que las caracterizan. El reconocimiento de voz es una de ellas lo que ha supuesto uno de los objetos de estudio para este proyecto.

3.6.1.4 Recurrent Neuronal Networks.

Las redes neuronales recurrentes tienen una característica que las hace distinguirse entre las demás que es la incorporación de la realimentación, ello les permite estar optimizadas para sistemas de procesamiento de secuencias. [17] En esta sección, se describen tres tipologías de redes recurrentes que podrían ser utilizadas para resolver el objetivo de este trabajo y que se diferencian por el tipo de feedback utilizado. [28]

- Input-Output Recurrent Model: Es un modelo formado varios bloques de retardo, una red estática del tipo multilayer y un elemento de realimentación que conecta la salida de la red con las entradas. Recibe también el nombre de NARX (nonlinear autoregressive with exogenous inputs).

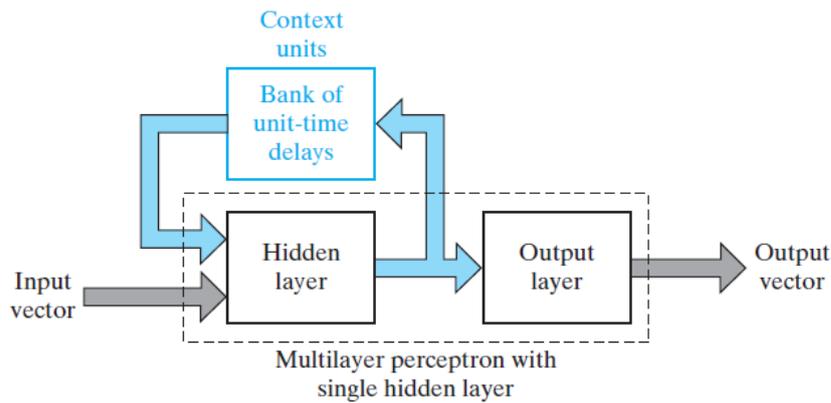


Figura 22. Simple Recurrent Model. Estructura de red recurrente simple con un elemento de realimentación en la capa oculta que la forma.

- Recurrent Multilayer Perceptrons: Esta estructura tiene una o más capas ocultas y cada una tiene un elemento de realimentación. Permite seleccionar de entre las capas de la red el origen y el destino de la realimentación.

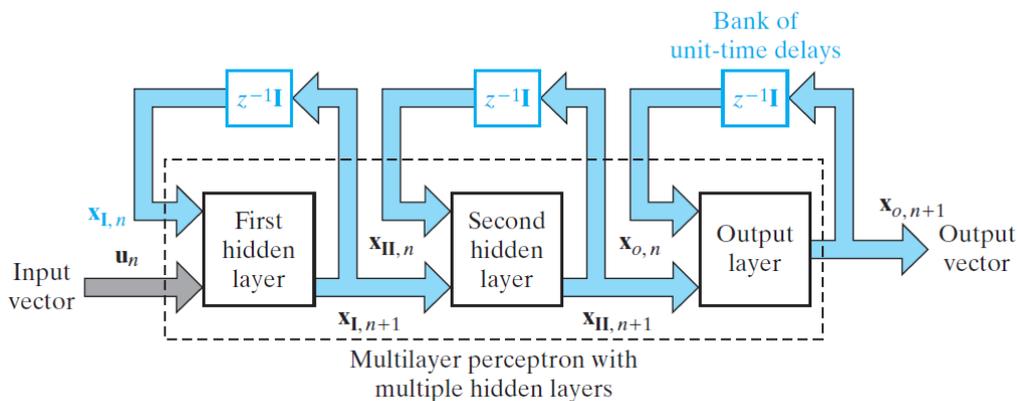


Figura 23. Recurrent multilayer perceptron. Estructura compleja de red recurrente formada por varias capas ocultas y elementos de realimentación en cada una.

3.6.2 Aplicación con la señal del acelerómetro

En esta sección del documento se propone una solución a la detección del toque entre el dedo índice y pulgar haciendo uso de las herramientas de diseño y entrenamiento de redes neuronales de Matlab.

El tipo de red que se usa en este proyecto es feedforward e incluye la variable temporal de forma implícita en la capa de entrada de la misma, lo que facilita la detección de patrones y agiliza el entrenamiento. Esta estructura es similar a las de Time-Delay ya que la variable temporal está presente en la entrada de ella, sin embargo, los bloques de retardo que se ha creado no forman parte del entrenamiento, simplificando así el tiempo de aprendizaje.

Se puede utilizar como valores de entrada para la red los atributos de la señal, es decir, introducir el valor medio, valor rms o parámetros característicos de la señal en el dominio de la frecuencia, como se mencionaba en anteriores capítulos, sin embargo, también podemos introducir la ventana temporal directamente. [3]

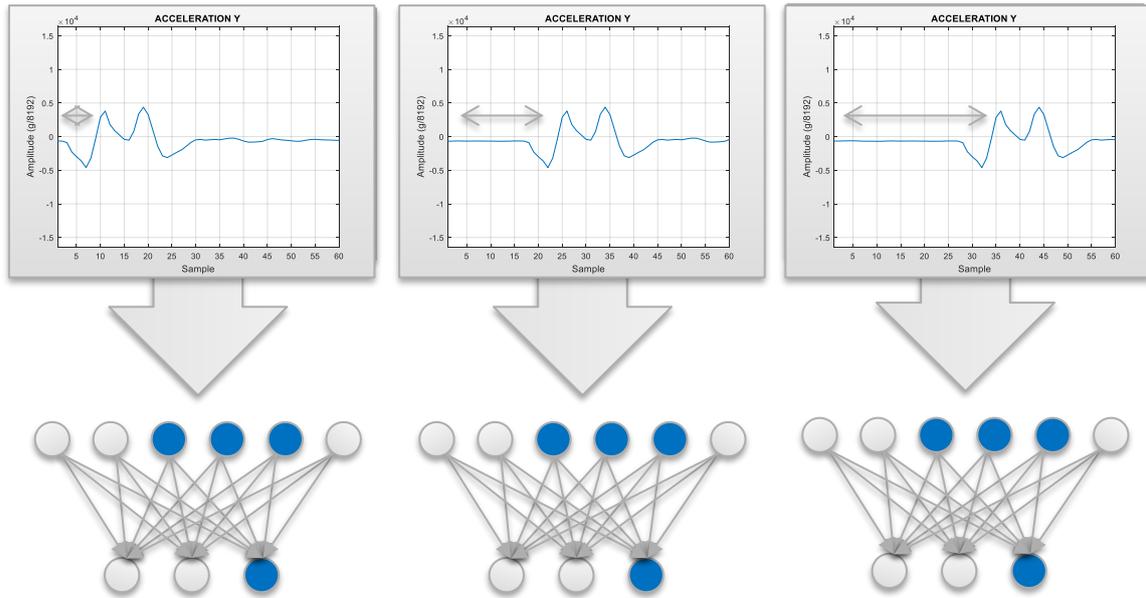


Figura 24. Esquema de la red neuronal. Esta figura representa un esquema del procesado en tiempo real de la red neuronal. En la parte superior se puede observar el desplazamiento temporal de una ventana que contiene la información de un toque simple entre el dedo índice y pulgar. En la parte inferior se encuentra la red con la capa oculta y la capa de salida. El valor de salida de la red es siempre constante ya que la información sobre el movimiento dentro de la ventana no varía.

El diseño de la red está basado en tres capas:

- Capa de entrada: Tiene un tamaño equivalente al elegido para la ventana temporal.
- Capa oculta: Se compone de 30 neuronas que interconectarán la capa de entrada y de salida.
- Capa de salida: Está formada por tres neuronas que se corresponden cada una con un tipo de movimiento.
 - o Neurona 1: Movimiento de un toque entre el dedo índice y el dedo pulgar.
 - o Neurona 2: Movimiento de doble toque entre el dedo índice y el dedo pulgar.
 - o Neurona 3: No se produce ningún movimiento de los anteriores.

En cuanto al tipo de algoritmo de entrenamiento, se ha elegido el Scaled Conjugate Gradient, ya que ofrece resultados positivos y en menos de un minuto.

La señal que entra a la red es la que se corresponde con la aceleración en el eje Y, ya que el vector de desplazamiento y aceleración se encuentran en el mismo plano, haciendo que las variaciones de amplitud sean elevadas.

La transmisión de una nueva muestra se produce cada 15ms. Hemos seleccionado experimentalmente una ventana temporal de 60 muestras que, en tiempo, equivale a 900ms de duración de señal.

El entrenamiento de la red ha consistido en registrar sets de 3.000 muestras de las aceleraciones en los tres ejes y generar ventanas de 60 muestras con la información de los movimientos. Mediante un GUI creado en Matlab, se puede marcar el inicio y el final del movimiento, desplazar la señal y asignar la activación de una de las neuronas de la capa de salida en función del tipo de movimiento que se esté clasificando.

A través de los números del teclado numérico interactuamos con la interfaz de Matlab.

- Numpad1: Marcamos el inicio del movimiento que se va a clasificar.
- Numpad3: Marcamos el final del movimiento que se va a clasificar.

- Numpad4: Desplazamos el vector una muestra hacia valores anteriores a la referencia.
- Numpad6: Desplazamos el vector una muestra hacia valores posteriores a la referencia.
- Numpad7: Desplazamos el vector 10 muestras hacia valores anteriores a la referencia.
- Numpad9: Desplazamos el vector 10 muestras hacia valores futuros a la referencia.
- Return: Una vez seleccionado los valores de inicio y final del movimiento, se crean los vectores de entrenamiento desplazando una ventana entre los márgenes que hemos marcado.

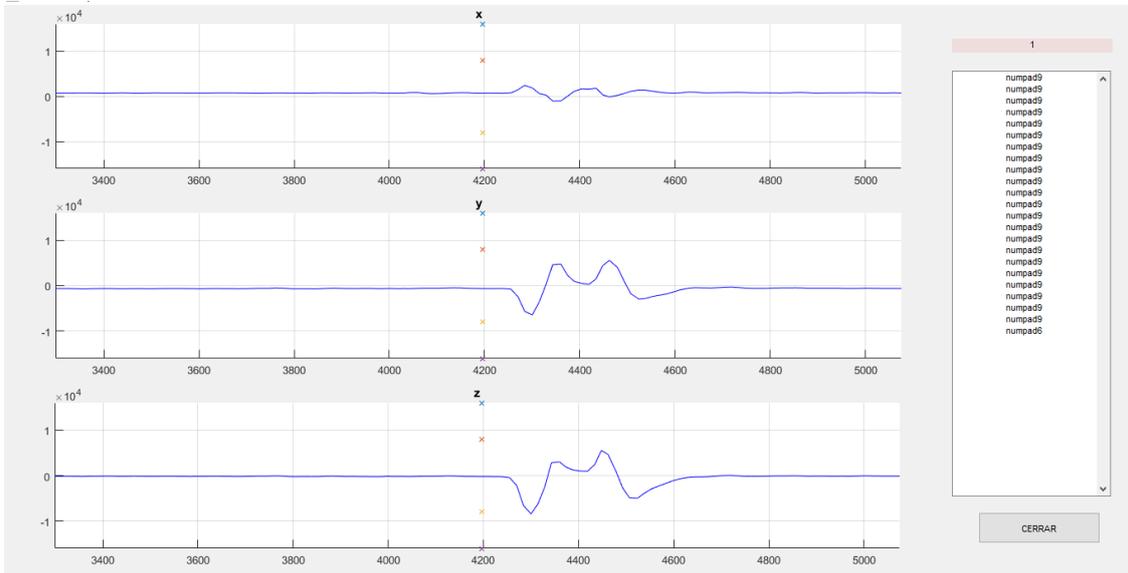


Figura 25. Interfaz para clasificar señales. GUI realizado en Matlab que permite recorrer los registros de entrenamiento y clasificarlos para generar los sets de entrenamiento de la red.

En la interfaz se pueden observar las gráficas de las tres aceleraciones junto con un eje en medio que hace de referencia para marcar el inicio y el final del movimiento. Además, a la derecha tenemos un registro de las teclas que hemos pulsado y en la parte superior de la lista se encuentra un contador que marcará el número de ventanas que tenemos para nuestra red.

Confusion Matrix

	1	2	3	
1	11565 13.0%	53 0.1%	210 0.2%	97.8% 2.2%
2	73 0.1%	12224 13.7%	101 0.1%	98.6% 1.4%
3	253 0.3%	51 0.1%	64391 72.4%	99.5% 0.5%
	97.3% 2.7%	99.2% 0.8%	99.5% 0.5%	99.2% 0.8%
	1	2	3	

Target Class

Figura 26. Matriz de confusión. Porcentajes sobre el error de interpretación de la red en la fase de entrenamiento. Esta figura muestra el número de vectores de entrenamiento que han sido utilizados para entrenar la red.

Las señales clasificadas se unen y son utilizadas para el entrenamiento de la red. Para comprobar que la red entrenada clasifica las señales correctamente, se registran vectores de 3.000 muestras y se van introduciendo ventanas de 60 muestras desplazadas de una en una.

Los datos correspondientes al número de señales introducidas y porcentaje de acierto entre las muestras de entrenamiento se muestran en la Figura 27.

Para entrenar la red mediante esta metodología, hay que tener en cuenta el tamaño de la ventana, el cual, dependerá de la duración del movimiento, en este caso, el movimiento más restrictivo es el del doble toque.

La primera red creada y entrenada tenía una ventana de 140 muestras, con el objetivo de tener un movimiento completo dentro de la ventana y que incluyera un periodo de reposo tanto

antes de la señal como después. Tras realizar los test para verificar que la red está entrenando correctamente, se observan dos inconvenientes:

- No se puede realizar ningún movimiento adicional durante 1 segundo posterior a la señal, ya que aún sigue procesando el movimiento.
- Hay un retraso de 0.5 segundos desde que se realiza el movimiento hasta que se muestra el resultado.

La disminución de la ventana reducirá los problemas de retardo, es por ello, que se cambia a una ventana de 60 muestras y se eliminan los periodos de reposo anterior y posterior. La información principal de cada movimiento se representa en la siguiente figura.

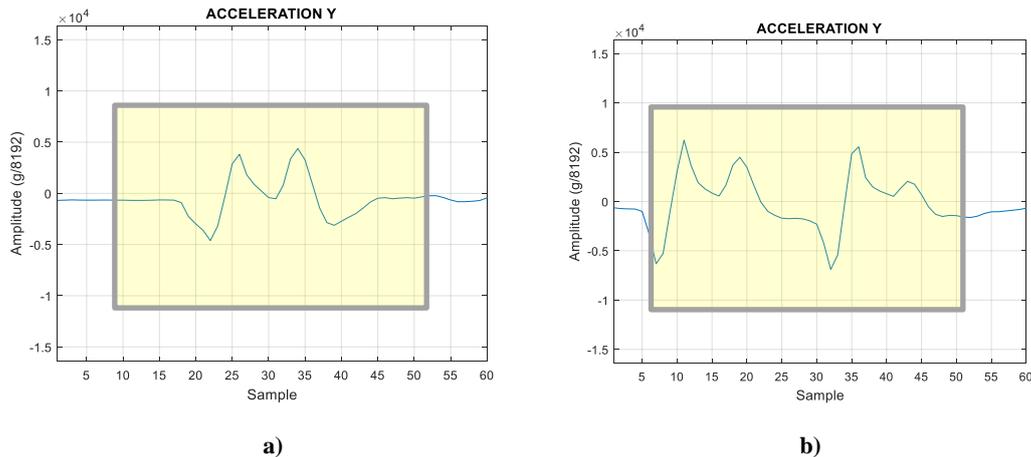


Figura 28. Información temporal sobre los movimientos de toque y doble toque. En esta figura se puede observar un recuadro amarillo encima de la señal de la aceleración representando así donde se encuentra la información sobre los movimientos dentro de la ventana. Figura a) movimiento de un toque. Figura b) movimiento de doble toque.

El entrenamiento sigue un proceso gradual, se comienza entrenando la red con movimientos sencillos y repetitivos y se realizan test parecidos. Cuando se ha obtenido una buena respuesta, se añaden sets de entrenamiento distintos, en los que la amplitud de la señal o la distancia temporal entre picos varía.

La presencia del vector de gravedad en los movimientos donde es paralelo al vector de aceleración en el eje Y, lleva a la realización de sets de entrenamiento en posición horizontal, ya que estas señales incluyen un pequeño offset resultado del valor de gravedad.

Es importante documentar todos los movimientos que se van a utilizar en el entrenamiento, el objetivo de tener un mayor control sobre los sets que generan más errores y modificarlos para evitar los falsos positivos. Se ha generado una tabla con distintos tipos de movimientos. A continuación, se realiza un resumen de su contenido.

- Movimiento de un toque entre el dedo índice y pulgar: Se registran movimientos de un toque fuerte, suave y normal a una sola persona, tanto en posición vertical como posición horizontal.
- Movimiento de dos toques entre el dedo índice y pulgar: Se realizan los mismos registros que en el anterior movimiento.
- Ausencia de movimiento o movimientos negativos: Podemos encontrar varios tipos, entre ellos que destaca el movimiento de reposo, el de poner y quitar el guante y los movimientos aleatorios de la mano.

En la siguiente tabla, se observa un ejemplo con la descripción de los movimientos y el nombre del archivo .mat que los contiene.

DESCRIPCIÓN DEL MOVIMIENTO DE NADA	.mat
Movimiento de desplazamiento de la mano sobre la mesa en posición vertical.	set3_1 set3_2 set3_3
Movimiento de poner y quitar el acelerómetro en posición vertical	set3_4 set3_5 set3_6
Movimiento relajado de la mano en posición vertical.	set3_7 set3_8 set3_9
Movimiento de pulsación continua entre el dedo índice y pulgar en posición vertical.	set3_10 set3_11 set3_12
Movimiento de retroceso del dedo índice y pulsación continua en posición vertical.	set3_13 set3_14 set3_15
Movimientos aleatorios bruscos en posición vertical.	set3_16 set3_17
Movimientos aleatorios suaves y de transición entre posición vertical y horizontal.	set3_18 set3_19 set3_20
Movimiento de ponerse el guante del prototipo en ambas posiciones.	set3_21 set3_22

Tabla 6. Ejemplo de tabla de clasificación. Resumen de los movimientos negativos o que no pertenecen al grupo de clasificación de movimientos de un toque o doble toque.

3.7 Interfaz con Matlab.

Para comprobar el funcionamiento de la red neuronal en tiempo real, se ha creado un interfaz en Matlab que procesa muestra a muestra los valores de aceleración y representa la respuesta en tiempo real de la red neuronal.

Está compuesta de los botones de iniciar y apagar y un esquema de la mano con un cuadrado que se iluminará. El color verde significa que no se ha realizado ningún movimiento, el de color azul indica que se ha realizado un toque y el rojo dos toques.

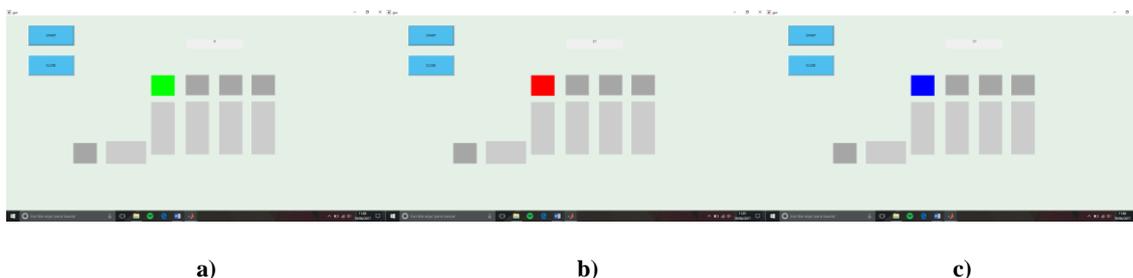


Figura 29. Interfaz en Matlab. En las tres figuras se puede apreciar una misma estructura de interfaz, pero con un cuadrado de distintos colores. La figura a) con el recuadro verde representa ningún movimiento, la figura b) representa el movimiento de doble toque y la figura c) el de un toque.

3.8 Interfaz con GTK.

Con el objetivo de diseñar una interfaz para que en un trabajo futuro pueda ser integrada en un dispositivo Raspberry Pi, se exporta la red neuronal entrenada a código C y se crea una interfaz

mediante el uso de las librerías GTK y Cairo que permiten visualizar la respuesta de la red neuronal en tiempo real.

Para convertir la red neuronal a código C hay que crear una función en Matlab utilizando las siguientes instrucciones. [19]

```
load("nn.mat")  
genFunction(net, nn, 'MatrixOnly', 'yes')
```

Una vez generada, la función tiene como argumento de entrada un vector de tamaño 60 muestras y como valor de salida genera un vector de 3 valores que hacen referencia a cada uno de los tipos de clasificación que hemos considerado en el entrenamiento.

A continuación, se utiliza la herramienta Matlab Coder que se encuentra en el programa Matlab y se siguen los pasos para generar un directorio que contendrá todas las librerías que son necesarias para ejecutar la red neuronal en código C.

Generamos en código C tres funciones con el objeto de obtener un programa que haga de interfaz para la aplicación de realidad aumentada, estas funciones son:

- Lectura del puerto serie.
- Generación de un vector de salida por la función de la red neuronal a partir de los valores que llegan por el puerto serie.
- Visualización del resultado de la red.

La lectura del puerto serie se ha realizado utilizando GIOChannel que permite integrar sus funciones en el programa principal sin necesidad de crear otro hilo de ejecución. [9]

Los archivos creados por Matlab Coder, incluyen: una función de inicialización, otra que clasifica el patrón y una tercera de finalización. La ventana que entra a la red sigue un desplazamiento muestra a muestra conforme a la llegada de valores de aceleración por el puerto serie. Dicha ventana es procesada por la función de la red neuronal y genera los vectores de clasificación con los tres tipos de movimientos que se marcaron en el entrenamiento.

Por último, para visualizar el resultado de la red neuronal se han creado dos flechas utilizando las librerías GTK y Cairo. Al realizar el movimiento de un toque con el guante se representará la flecha de la derecha en color verde, si se realiza doble toque se representará la flecha de la izquierda en color rojo y si no se realiza ningún movimiento, no se representará nada. [10]

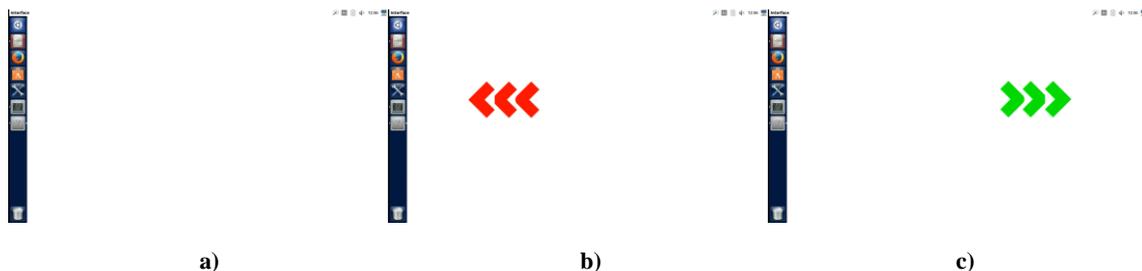


Figura 30. Interfaz en GTK. La figura a) representa ningún movimiento, la figura b) representa el movimiento de doble toque y la figura c) el de un toque.

Tenemos que compilar el programa principal y todos los archivos que se encuentran en el directorio, para ello nos situamos en el directorio, abrimos el terminal y escribimos el comando que se muestra a continuación. [20]

```
gcc `pkg-config --cflags gtk+-3.0` -o main -lm `pkg-config --libs gtk+-3.0`
```

Necesitamos ejecutar el programa principal con permisos de administrador para poder acceder al puerto serie. Es necesario activar el puerto serie en la máquina virtual.

```
sudo ./main
```

3.9 Comunicación inalámbrica.

3.9.1 Bluetooth Low Energy.

La comunicación Bluetooth es uno de los dos protocolos de radiocomunicaciones más populares por su alta tasa de transmisión de datos. En 2014, surge el Bluetooth Low Energy, con el objetivo de reducir el consumo, además, supone un cambio en el protocolo de Bluetooth y, por lo tanto, permite reducir costes y aumentar la sencillez en el desarrollo de aplicaciones basadas en esta tecnología. [7]

Comparado con otros protocolos de comunicación, el Bluetooth Low Energy es el que menos consume y el que mantiene una tasa óptima de transmisión de datos. Además, la posibilidad del modo de bajo consumo durante periodos de inactividad lo hacen perfecto para aplicaciones que no necesiten una transmisión de datos continua.

	BLUETOOTH CLASSIC	BLE	ZIGBEE	WIFI
IEEE STANDARD	802.15.1	802.15.1	802.15.4	802.11 (a, b, g, n)
FREQUENCY (GHZ)	2.4	2.4	0.868, 0.915, 2.4	2.4 and 5
MAXIMUM RAW BIT RATE (MBPS)	1-3	1	0.250	11 (b), 54 (g), 600 (n)
TYPICAL DATA THROUGHPUT (MBPS)	0.7-2.1	0.27	0.2	7 (b), 25 (g), 150 (n)
MAXIMUM (OUTDOOR) RANGE (METERS)	10 (class 2), 100 (class 1)	50	10-100	100-250
RELATIVE POWER CONSUMPTION	Medium	Very low	Very low	High
EXAMPLE BATTERY LIFE	Days	Months to years	Months to years	Hours

Tabla 7. Comparación entre protocolos de comunicación inalámbrica. [31]

A continuación, se analiza la capa física de la tecnología BLE. Opera en una banda de frecuencia no licenciada y es capaz de admitir cuarenta canales, de ellos, tres son para dispositivos que se encuentren en el modo de advertencia, mientras que el resto son utilizados para comunicaciones punto a punto.

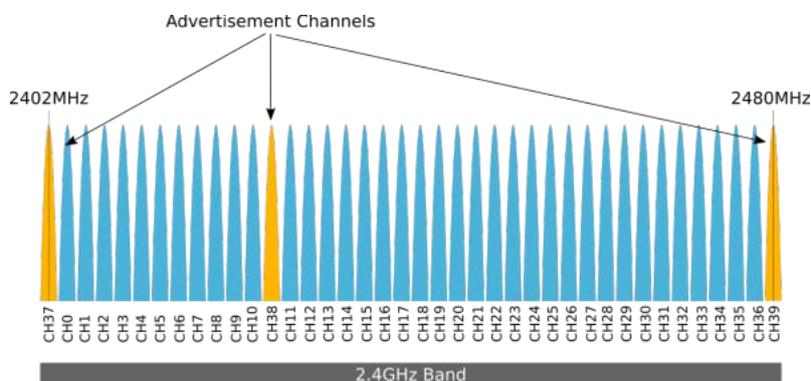


Figura 31. Canales utilizados en BLE. En esta figura se puede observar la división en canales de la banda que utiliza el BLE, en color naranja se representan los canales donde se producen las

advertencias y en color azul se representa los canales donde se pueden establecer las conexiones.

La capa de enlace de BLE, controla las conexiones, la transmisión y la recepción de datos, es lo que se denomina GAP o Generic Access Profile. Existen dos tipos de roles para el dispositivo Bluetooth, el de periférico, que se corresponde con dispositivos pequeños y de bajo consumo y el rol central, que dispone de mucha mayor carga de procesado que los anteriores.

Los estados en los que el dispositivo BLE se puede encontrar son:

- Standby: Es el modo de bajo consumo, no recibe ni emite paquetes.
- Advertencia: El dispositivo se encuentra en modo periférico. En este estado emite paquetes en los canales de advertencia con un tamaño de 31 bytes.
- Búsqueda: Es el estado en el que se reciben paquetes de advertencia.
- Iniciando: Es el estado en el que entra un dispositivo BLE cuando se encuentra en modo central y ha terminado de establecer una conexión. Escuchará paquetes de advertencia hasta que reciba el paquete deseado y se producirá la conexión y el intercambio de datos en los 37 canales restantes.

Después de que se realice la conexión entre el dispositivo periférico y el dispositivo central, se puede acceder a los los servicios y características GATT, lo que permite una mayor velocidad de transmisión con respecto a los paquetes de advertencia, aunque se ve restringido a una sola conexión. [23]

GATT es el acrónimo de Generic Attribute Profile, se define el intercambio de datos entre dos dispositivos BLE haciendo uso de servicios y de características. El protocolo ATT o Attribute Protocol unifica los datos relativos a la comunicación, permitiendo así la lectura y escritura de características.

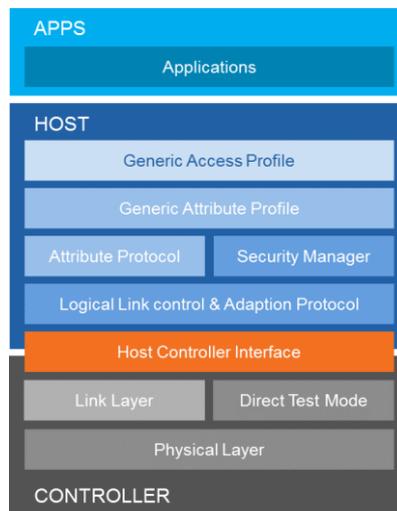


Figura 32. Capas del protocolo de comunicación BLE. En esta figura se puede observar el orden de las capas en un dispositivo BLE.

Un concepto importante para entender la comunicación BLE es la relación entre el servidor y cliente. El dispositivo periférico es conocido como el servidor GATT y el central es el cliente GATT que inicia la conexión y envía las peticiones.

El protocolo GATT está formado por:

- Perfiles: Están definidos por el fabricante del dispositivo BLE y puede tener varios servicios.

- Servicios: Están organizados en bloques que contienen características donde se almacena la información. Los servicios pueden contener una o más características y se diferencian a través de su UUID.
- Características: Almacenan datos sobre cada servicio, al igual que los servicios, cada característica tiene un UUID predefinido.

La placa BlueDuino rev2, nos permite una comunicación BLE gracias al circuito integrado CC2540, el cual lleva embebido el firmware ZeroBeacon y nos permite la comunicación mediante comandos AT para su configuración. Podemos modificar el intervalo de advertencia, el role en la conexión o la inhabilitación para reducir el consumo de potencia.

El firmware ZeroBeacon incluye un servicio con el UUID de 16 bits 0xFFFF0 y dos características que hacen referencia al canal de transmisión y al de recepción, en las direcciones 0xFFFF1 y 0xFFFF2. Además, integra el protocolo iBeacon que inventó Apple para compartir información entre dispositivos BLE de bajo coste y baja potencia. Es una opción que hemos desactivado del firmware.

El dispositivo periférico será el BlueDuino rev2 y el dispositivo central el ordenador donde se ejecute la interfaz. En el dispositivo central tendremos que configurar la lectura mediante Bluetooth con la librería Bluez y para poder interactuar con ella necesitaremos el uso de la librería Pexpect, mediante la cual podemos interactuar con Bluez a través del terminal.

Se han tenido varios problemas con el BlueDuino rev2 que se han ido resolviendo, como la actualización del firmware, debido a bugs en la comunicación. Por otro lado, la transmisión de datos tiene un retraso importante del cuál por ahora se desconoce el origen.

3.9.2 WiFi.

Los problemas en la configuración del Bluetooth en la placa BlueDuino rev2, hacen replantearse la necesidad de usar este dispositivo. Es necesario actualizar el firmware que incluye el dispositivo CC2540, ya que hay un error que no permite el uso de la UART transparente a través de los servicios UUID. Por otra parte, se ha intentado modificar el tiempo de advertencia del protocolo BLE y no es posible bajarlo de 100ms. Finalmente, tenemos el problema de comunicación en el dispositivo central, ya que por falta de documentación no ha sido posible crear una comunicación fiable que permita recibir datos sin retraso temporal. La única forma de establecer una conexión ha sido a través de la aplicación BlueZ ejecutada desde el terminal y de la librería Pexpect en el lenguaje Python lo que permite utilizar el terminal desde un programa.

La comunicación inalámbrica se puede realizar también a través de otros protocolos como el WiFi y las placas basadas en el circuito integrado ESP8266. De las cuales tenemos experiencia de que funcionan correctamente y permite una conexión sencilla entre el dispositivo periférico y el central. La placa ESP12F, es un módulo que incluye el microcontrolador, la conectividad WiFi y la antena, pero no incorpora ni conector de batería ni conector FTDI para convertir de USB a puerto serie. La programación de este dispositivo se puede realizar a través de un programador serie y del IDE de Arduino seleccionando la placa ESP8266 genérica. Sus dimensiones la hacen ideal para integrarla en guante y la conectividad WiFi está más estandarizada que el BLE, aunque el consumo sería mayor.



Figura 33. ESP12

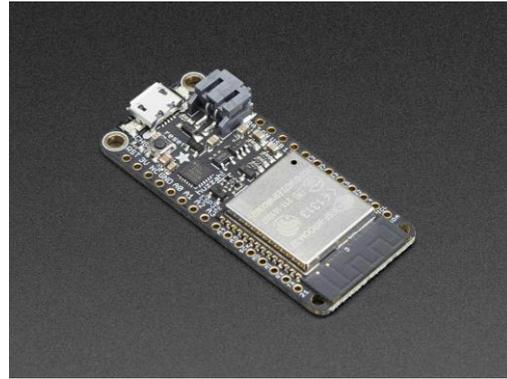


Figura 34. Adafruit HUZAH32

Una de las placas que mejores características podría aportar para este proyecto es la Adafruit HUZAH32, que integra conectividad inalámbrica WiFi, Bluetooth clásico y BLE, además posee un conector externo de batería que nos permite recargarlo por USB sin necesidad de desconectar la batería. Las dimensiones y las prestaciones del dispositivo junto con la documentación que proporciona Adafruit hacen que sea un buen microcontrolador en el desarrollo de un trabajo futuro.

Capítulo 4. Conclusiones y propuesta de trabajo futuro

Este trabajo desarrolla un prototipo para proporcionar una solución a la necesidad de interactuar con una aplicación de realidad aumentada a través del uso de acelerómetros. Se han investigado las formas de obtener el posicionamiento o desplazamiento a partir de la señal del acelerómetro directamente. Sin embargo, ante la falta de un resultado fiable se ha propuesto una solución que permite reconocer patrones de movimiento mediante el uso de redes neuronales. Se ha desarrollado una interfaz con un solo acelerómetro, pero en un trabajo futuro se extenderá a todos los dedos de la mano.

Se hacen varias propuestas de trabajo futuras correspondientes a cada una de las partes que integran este proyecto.

- Aumentar el número de muestras de entrenamiento y reducir al máximo el retardo de respuesta de la red para que la interpretación sea lo más rápida y fiable posible.
- Comunicar inalámbricamente el microcontrolador con el dispositivo central en el que se ejecuta la interfaz.
- Integrar la interfaz en una Raspberry Pi, dispositivo que controla la aplicación de realidad aumentada.
- Optimizar el código C de la interfaz lo que proporcionará más eficiencia.
- Integrar más acelerómetros en el guante para tener un mayor control del movimiento de los dedos.

Capítulo 5. Bibliografía

- [1] Analog Devices company. “The Scientist and Engineer's Guide to Digital Signal Processing”. CHP 15.
- [2] Anusha K. S., Mathew T. Mathews, Subha D. Puthankattil. “Classification of Normal and Epileptic EEG Signal using Time & Frequency Domain Features through Artificial Neural Network”. *2012 International Conference on Advances in Computing and Communications*.
- [3] Donald B. Malkoff. “A Neural Network for Real-Time Signal Processing”.
- [4] Matlab Corporation, “Signal Processing for Machine Learning in MATLAB”. <https://www.youtube.com/watch?v=VO0d6EuGpO0>. [Online] [Recuperado en 20/05/2017]
- [5] Phasespace Corporation. “X2E motion capture”. <http://phasespace.com/x2e-motion-capture/>. [Online] [Recuperado en 08/06/2017]
- [6] J. Martínez, A. García, M. Oliver, J.P. Molina, P. González. Identifying Virtual 3D Geometric Shapes with a Vibrotactile Glove. *IEEE Computer Society*, 2016
- [7] Argenox Corporation. “Introduction to Bluetooth Low Energy (BLE) v4.0”. <http://www.argenox.com/bluetooth-low-energy-ble-v4-0-development/library/introduction-to-bluetooth-low-energy-v4-0/>. [Online] [Recuperado en 20/06/2017]
- [8] Adrio Communications. “RFID Standards”. <http://www.radio-electronics.com/info/wireless/radio-frequency-identification-rfid/low-high-frequency-bands-frequencies.php>. [Online] [Recuperado en 15/05/2017]
- [9] John Coppens. “GTK+-GIOChannel”. <http://jcoppens.com/soft/howto/gtk/giochannel.en.php>. [Online] [Recuperado en 11/06/2017]
- [10] ZetCode web. “Cairo backends, gtk window”. <http://zetcode.com/gfx/cairo/cairobackends/>. [Online] [Recuperado en 11/06/2017]
- [11] Jeff Rowberg. “MPU6050 library”. <https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050> [Online] [Recuperado en 22/02/2017]
- [12] Invensense Corporation. “MPU-6000 and MPU-6050 Product Specification Revision 3.3”.
- [13] Invensense Corporation. “MPU-6000 and MPU-6050 Register Map and Descriptions Revision 4.2”.
- [14] Ruoyu Zhi. “A Drift Eliminated Attitude & Position Estimation Algorithm In 3D”.
- [15] M. Y. Rafiq, G. Bugmann, D. J. Easterbrook. “Neural network design for engineering applications”.

- [16] Matlab Corporation. "Specify Layers of Convolutional Neural Network". <https://es.mathworks.com/help/nnet/ug/layers-of-a-convolutional-neural-network.html>. [Online] [Recuperado en 27/05/2017]
- [17] Peter Tino, Lubica Benuskova, Alessandro Sperduti. "Artificial neural network models".
- [18] A. Krizhevsky, I Sutskever, G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Network".
- [19] Matlab Corporation. "Deploy Trained Neural Network Function". <https://es.mathworks.com/help/nnet/ug/deploy-neural-network-functions-and-objects.html>. [Online] [Recuperado en 06/05/2017]
- [20] The GNOME Project. "Getting Started with GTK+". <https://developer.gnome.org/gtk3/stable/gtk-getting-started.html>.
- [21] Pieter-Jan Van de Maele. "Reading a IMU Without Kalman: The Complementary Filter". <http://www.pieter-jan.com/node/11> [Online] [Recuperado en 12/02/2017]
- [22] April Brother Corporation "Firmware/ ZeroBeacon". <http://wiki.aprbrother.com/wiki/Firmware/ZeroBeacon> [Online] [Recuperado en 4/11/2016]
- [23] Kevin Townsend "Introduction to Bluetooth Low Energy". <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction>. [Online] [Recuperado en 25/06/2017]
- [24] RFWireless World. "Differences between Bluetooth and BLE". <http://www.rfwireless-world.com/Terminology/Bluetooth-vs-BLE.html>. [Online] [Recuperado en 12/06/2017]
- [25] Sparkfun Corporation. "I2C". <https://learn.sparkfun.com/tutorials/i2c>. [Online] [05/03/2017]
- [26] Arduino Corporation. "MPU-6050 Accelerometer + Gyro". <http://playground.arduino.cc/Main/MPU-6050#sketch>. [Online] [Recuperado en 28/11/2017]
- [27] Simon Haykin. "Feedforward neural networks: an introduction".
- [28] Simon Haykin. "Dynamically Driven Recurrent Networks".
- [29] Margaret Rouse. "Augmented reality definition". <http://whatis.techtarget.com/definition/augmented-reality-AR>. [Online] [Recuperado en 22/06/2017]
- [30] Alan Mauer. "Arduino Glasses a HMD for Multimeter". <https://hackaday.io/project/12211-arduino-glasses-a-hmd-for-multimeter>. [Online] [Recuperado en 01/10/2017]
- [31] Sparkfun Corporation. "Bluetooth Basics". <https://learn.sparkfun.com/tutorials/bluetooth-basics>. [Online] [Recuperado en 22/07/2017]

Capítulo 6. Agradecimientos.

El presente trabajo de fin de grado ha sido elaborado bajo la supervisión del profesor José Manuel Mossi García, a quien me gustaría expresar mi más profundo agradecimiento, por hacer posible la realización de este trabajo. Además, agradecer su tiempo, dedicación y apoyo para que el proyecto pudiera ser desarrollado de manera exitosa.

Quisiera agradecer también a mis padres, Rafael e Isabel, y a mi hermano, Santiago, por estar siempre apoyándome y ayudándome en todo lo que he necesitado y por ser una fuente de motivación constante durante mis estudios y mi vida personal. Gracias a mi abuela Isabel, una luchadora incansable que ha conseguido que no me rindiera nunca y me ha dado toda la energía necesaria para alcanzar los objetivos que me proponga.