



Universidad de Valladolid



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS DE
TELECOMUNICACIÓN

SISTEMA DE AYUDA A LA
REHABILITACIÓN DE PACIENTES
CON DISFUNCIÓN EN
EXTREMIDADES INFERIORES

CURSO 2018-2019

AUTORA: ISABEL JUNQUERA GODOY
TUTOR: DR. ALONSO ALONSO ALONSO

ÍNDICE

1. Introducción
 - 1.1. Justificación
 - 1.2. Estado del arte
 - 1.3. Antecedentes
 - 1.4. Objetivos
 - 1.5. Metodología

2. Planteamiento estructura
 - 2.1. Posibles soluciones
 - 2.2. Solución escogida

3. Hardware
 - 3.1. Arduino
 - 3.2. Potenciómetro
 - 3.3. Estructura impresión 3D
 - 3.4. Recubrimiento
 - 3.5. Módulo Bluetooth
 - 3.6. Diseño final

4. Software
 - 4.1. Blynk
 - 4.2. Aplicación
 - 4.3. Código

5. Protocolo de pruebas y resultados

6. Conclusiones
 - 6.1. Posibles líneas futuras
 - 6.1.1. Futuro de las ortesis
 - 6.1.2. Mejoras en el sistema actual

RESUMEN

El siguiente Trabajo de Fin de Grado tiene como objetivo el desarrollo de un sistema de ayuda a la rehabilitación de pacientes con afección en las extremidades inferiores. Concretamente se aplicará a disfunciones en la articulación de la rodilla. Se diseñará una órtesis sensorizada para el seguimiento de los ejercicios de rehabilitación junto con una aplicación en el teléfono móvil como guía para el paciente durante la ejecución de los ejercicios.

Una órtesis es un dispositivo externo aplicado al cuerpo para modificar los aspectos funcionales y/o estructurales del esqueleto o del sistema neuromuscular. Se caracterizan por ser de gran utilidad como parte de la rehabilitación en diversas enfermedades. Las órtesis se pueden clasificar en activas, las cuales ayudan a la realización del movimiento de rehabilitación, y en pasivas, cuyo objetivo es totalmente opuesto, impiden la realización del movimiento.

ABSTRACT

The following Final Degree Project aims to develop an aid system for the rehabilitation of patients with lower limb condition. Specifically, it will be applied to dysfunctions in the knee joint. A sensorized orthosis will be designed to monitor rehabilitation exercises together with an application on the mobile phone as a guide for the patient during the execution of the exercises.

An orthosis is an external device applied to the body to modify the functional and structural aspects of the skeleton or neuromuscular system. It is characterized by being very useful as part of rehabilitation in various diseases. The orthoses can be classified as active, which help in the realization of the rehabilitation movement, and in passive ones, whose objective is totally opposite, prevent the realization of the movement.

Estudio anatómico de la extremidad inferior

Para poder comprender el comportamiento del movimiento de la rodilla se realizará un primer estudio biomecánico de dicha articulación:

La rodilla está formada por la unión de dos huesos importantes, el fémur y la tibia. También dispone de un pequeño hueso llamado rótula que está localizada en la parte inferior y anterior del fémur. Esta articulación puede realizar los movimientos de flexión y extensión tal y como se ve en la

siguiente imagen (Figura 1). Está rodeada por ligamentos que le dan estabilidad y músculos que hacen posible el movimiento de la extremidad. [1]

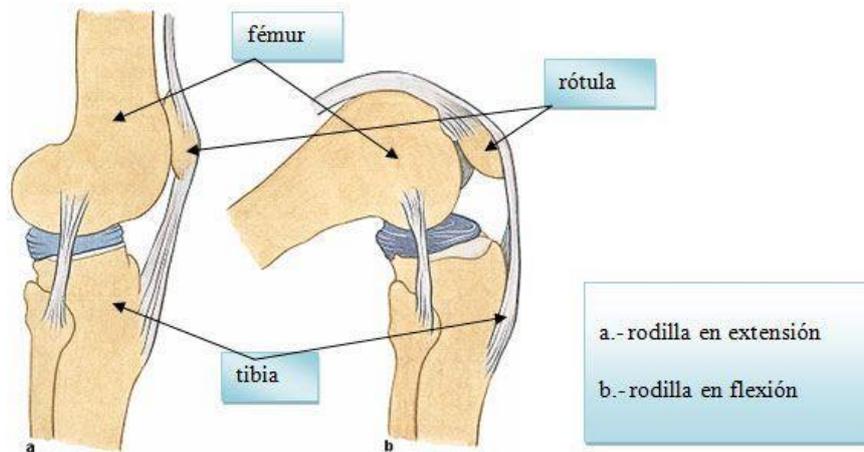


Figura 1. Rodilla en flexión y extensión.

Componentes óseos de la anatomía de la rodilla [2]:

- Epífisis distal del fémur: Es la parte inferior del fémur. Formada por dos cóndilos femorales con forma redondeada. Entre ambos cóndilos se encuentra la escotadura intercondílea que los separa por la parte de atrás. A ambos lados de los cóndilos existen unos relieves óseos denominados epicóndilos.
- Rótula: Se encuentra situada en la parte anterior de la rodilla, por delante de la tróclea femoral.
- Epífisis proximal de la tibia: Se corresponde a la parte superior de la tibia. Esta es aplanada y recibe el nombre de meseta tibial. Esta soporta el peso del cuerpo, el cual se transmite también hacia el pie. Tiene dos cavidades llamadas glenoidas las cuales albergan los cóndilos del fémur.

Tejidos blandos de la anatomía de la rodilla [2]:

- Cápsula articular: Es un recubrimiento fibroso que envuelve a la rodilla y forma un espacio cerrado. En su interior se une con los meniscos y conecta a la tibia a través de los ligamentos coronarios.
- Membrana sinovial: Recubre a la cápsula articular desde el fémur hasta la unión de los meniscos.
- Brusas: Sacos llenos de líquido que funcionan como amortiguador entre el tendón y el hueso.
- Meniscos: Son anillos de fibrocartílago para ayudar a la correcta congruencia entre las dos superficies tan irregulares del fémur y de la tibia.

- Ligamentos: estructuras que proporcionan estabilidad a la rodilla e impedimento de movimientos extremos.
- Músculos: El cuádriceps es el músculo principal, y se encarga de la función de extensión de la rodilla. Los isquiotibiales situados en la parte posterior del muslo se encargan del movimiento de flexión.

La rodilla es una articulación muy compleja que sufre tanto de manera cotidiana con el desgaste, como en el deporte. Por ello es frecuente que se lesione. La ortesis sensorizada propuesta en este documento está indicada para pacientes postquirúrgicos que presentan lesiones en los tejidos blandos situados en la rodilla. No está indicada para pacientes con lesiones neuronales ya que el dispositivo es pasivo, es decir, el paciente debe realizar el movimiento de rehabilitación. Para este segundo grupo de personas, las órtesis más recomendadas son las activas, ya que ayudan a la realización del movimiento [2].

Movimiento articular de la rodilla [3]:

La articulación tibio-femoral permite dos movimientos:

- Rotación: este movimiento es muy limitado y solo puede realizarse en posición de flexión.
- Flexión-extensión: Partiendo de una posición de reposo con la tibia y el fémur totalmente alineados correspondería a los 0° y la flexión total de la rodilla alcanzaría a una media de 140° tal y como se ve en la siguiente imagen (Figura 2).

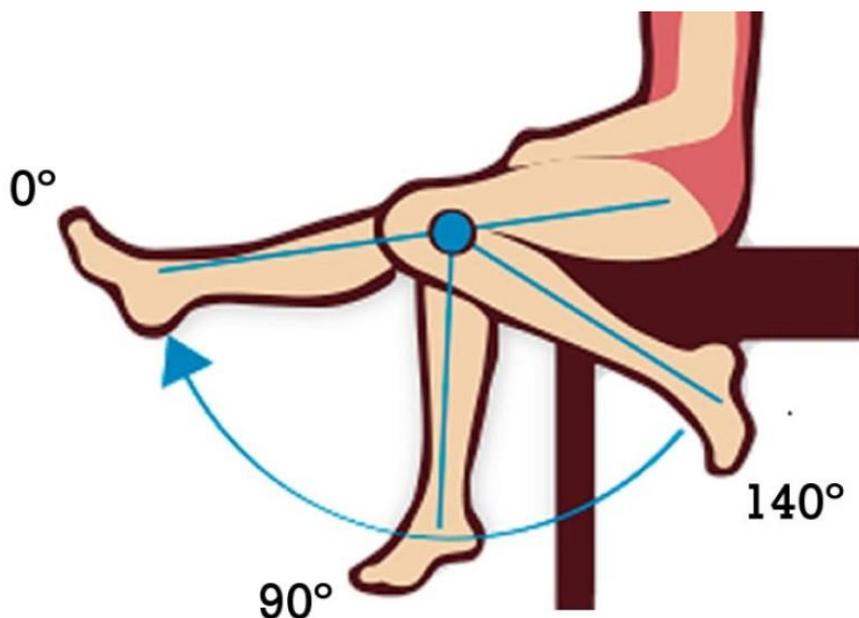


Figura 2. Articulación rodilla.

Ejercicio de rehabilitación

Según la afección que se quiera tratar, los ejercicios de rehabilitación necesarios cambian, así como las metas físicas a las que deben llegar los pacientes a la hora de hacer el movimiento.

Se ha diseñado esta órtesis con el objetivo de que el paciente pueda saber de forma sencilla si está realizando de forma correcta la rehabilitación. En este caso concreto, se desarrollan cinco ejercicios distintos. El usuario puede elegir dentro de la aplicación el ejercicio que quiere realizar. Todos ellos están destinados a fortalecer el cuádriceps y conseguir un movimiento articular completo.

El ejercicio de rehabilitación se puede modificar de manera muy sencilla cambiando el software del dispositivo.

Diseño de la órtesis sensorizada

Para poder realizar un buen diseño de la órtesis es necesario tener un objetivo claro del funcionamiento que se quiere conseguir del dispositivo. Conforme se haya reflexionado sobre el propósito final se desarrollará un mecanismo que cumpla con las expectativas impuestas. En este caso se desarrolla un diseño de ortesis sensorizada con el objetivo de realizar un seguimiento de la rehabilitación de pacientes de manera remota. Los pacientes deben ser capaces de realizar todos los ejercicios de rehabilitación de manera correcta, y que la órtesis sea capaz de recoger el movimiento con cierta precisión.

Al ser un instrumento cuya finalidad es el uso doméstico, debe ser fácil de usar, ligero, cómodo, ajustable a cada estructura fisiológica de cada paciente concreto y, además, económico.

Para encontrar el material idóneo para el diseño del mecanismo hay que tener en cuenta las características mencionadas anteriormente. En la impresión 3D existen una gran cantidad de materiales con los que trabajar (ácido poliláctico, poliestireno de alto impacto, tereftalato de polietileno...). Es necesario comparar las características de estos materiales y escoger uno que se ajuste a las necesidades del dispositivo.

El material propuesto para la fabricación de la órtesis es el ácido poliláctico (PLA) ya que es muy tenaz, duro y rígido, con resistencia a la abrasión y a componentes químicos. Es resistente al impacto y se puede recubrir fácilmente con láminas metálicas en caso de que se quiera dar más resistencia a la órtesis. Cumple con nuestros requerimientos de

material ligero y económico. Es uno de los filamentos más usados en la impresión 3D.

Las piezas de la órtesis se diseñaron a ordenador para su posterior impresión en 3D utilizando la impresora modelo BQ Prusa i3 Hephestos disponible en la Escuela Técnica Superior de Ingenieros de Telecomunicación.

Para poder realizar el registro del movimiento de la rodilla, meta principal del diseño de esta órtesis, se necesita un dispositivo que pueda "leer" el movimiento de extensión y flexión realizado por el paciente durante la rehabilitación. Por ello, una vez impresa la estructura principal, se realiza la sensorización de la pieza a partir de un potenciómetro. La función de dicho potenciómetro es registrar la posición angular en la que se encuentra la rodilla, por ello este sensor se coloca en la zona de la articulación de la pieza 3D. El potenciómetro está controlado por una placa desarrollada por Arduino llamada Arduino Mega 2560.

Esta placa electrónica está formada por un microcontrolador y varios pines digitales y analógicos que funcionan como "puertas" de entrada y salida de los datos.

Cada medio segundo la placa de Arduino calcula la diferencia de voltaje que pasa por el potenciómetro y procesa esa información asignando una posición angular. Dicho parámetro se lo manda, mediante conexión WiFi a una aplicación en el teléfono móvil. Esta app presenta los datos recogidos de manera gráfica para que el usuario pueda seguir su ejercicio y comprobar que realiza la rehabilitación de manera correcta. La aplicación se desarrolla en la plataforma para el diseño de aplicaciones llamada Blynk. La velocidad de medida de los ángulos de la rodilla está escogida en función de las limitaciones que presenta la aplicación de Blynk que se verán más adelante.

Palabras clave

Arduino, biomecánica, Blynk, potenciómetro, exoesqueleto, impresión 3D, ortesis, rehabilitación, sensorización.

1. INTRODUCCIÓN

1.1. Justificación

Una parte importante de la población presenta lesiones articulares que necesitan tratamiento continuado que se debe realizar en clínicas de rehabilitación. Sin embargo, muchas personas, por diversos motivos, no se pueden desplazar hasta la clínica a diario para realizar dichos ejercicios. De ahí surge la idea de este proyecto. Crear un dispositivo de uso doméstico que pueda monitorizar los ejercicios de rehabilitación de los pacientes y ayudarles a ejecutarlos de manera correcta. De esta manera los médicos podrían realizar un seguimiento continuo de sus pacientes sin saturar las clínicas de rehabilitación. Gracias a una ortesis impresa en 3D diseñada con las características fisiológicas específicas del usuario, hace que se consiga un adecuado y personalizado proceso de rehabilitación. La aplicación ayuda a corregir el movimiento durante el ejercicio indicando pautas y consejos que los pacientes puedan seguir.

1.2. Estado del Arte

Según la Norma UNE 11-909-90/1, una ortesis u órtesis, según definición de la Organización Internacional de Normalización (ISO), es un apoyo u otro dispositivo externo (aparato) aplicado al cuerpo para modificar los aspectos funcionales o estructurales del sistema neuromusculoesquelético, con la finalidad de mantener, mejorar o recuperar la función. Además, está en contacto permanente con el cuerpo humano, diferenciándola de los demás productos de apoyo. La palabra ortesis deriva del griego "ortho" que significa recto, enderezado o correcto. Es un término global que incluye dispositivos tanto dinámicos como estáticos.

El término se usa para denominar aparatos o dispositivos, férulas, ayudas técnicas y soportes usados por los pacientes, prescritos en especialidades médicas como: ortopedia y fisioterapia (medicina física y rehabilitación), y en algunas terapias como: fisioterapia y terapia ocupacional que corrigen o facilitan la ejecución de una acción, actividad o desplazamiento, procurando ahorro de energía y mayor seguridad. Sirven para sostener, alinear o corregir deformidades y para mejorar la función del aparato locomotor.

Es necesario diferenciar entre prótesis y ortesis. Las prótesis tanto para los miembros inferiores como superiores, son habitualmente necesarias en caso de amputaciones, al sustituir, parcial o totalmente al miembro. En cambio, la función de las ortesis es reemplazar o reforzar las funciones del miembro.

Las ortesis de miembro inferior están diseñadas para modificar las características estructurales y funcionales del sistema neuromuscular y del sistema esquelético de los miembros inferiores; Dichos dispositivos se pueden diseñar a medida, es decir, se fabrican para adaptarse a los requisitos funcionales específicos del usuario, o bien pueden ser prefabricados para cumplir con requisitos funcionales genéricos; Estos últimos suelen ser ajustables, y de esta manera se puedan adaptar al usuario individual.

Las ortesis pueden ser dinámicas, si disponen de componentes articulares que permiten definir una serie de rangos de movilidad, estáticas, si carecen de movilidad y en caso de que incorporen tanto partes dinámicas como estáticas se denominen ortesis mixtas o híbridas. Estos dispositivos ortopédicos tienen una serie de objetivos generales:

1. Estabilización de articulaciones paralizadas o débiles.
2. Función de soporte para articulaciones lesionadas o dañadas.
3. Control de movimientos anormales o espásticos.
4. Descarga de los segmentos distales.

La clasificación de las ortesis se puede realizar según diferentes criterios como puede ser según la zona anatómica involucrada (lumbar, cervical, extremidades, etc.), según el tipo de restricción que proporcionan a la articulación (rígida, flexible, semirrígida), según los objetivos de la ortesis (soportar, inmovilizar, corregir, etc.), según la patología o afección que trata (esguinces, fracturas, patologías nerviosas...). No obstante, una clasificación que resulta de utilidad en la práctica clínica habitual es la que utiliza el criterio terapéutico basándose en las funciones que realizan las ortesis, distinguiendo las funciones principales, primarias o directas que son aquellas que se dirigen a conseguir los objetivos terapéuticos más importantes. Según las funciones principales que realizan las ortesis se diferencian los siguientes tipos [4]:

- Ortosis funcionales o dinámicas: Permiten mejorar la función de la marcha.
- Ortosis posturales o pasivas: Destinadas a la prevención/corrección de deformidades.
- Ortosis correctoras: Actúan por aplicación progresiva de fuerzas, dentro de un sistema equilibrado de fuerzas en tres puntos, aprovechando las propiedades viscoelásticas de las partes blandas, la capacidad de remodelación del hueso cortical y la capacidad de crecimiento óseo de los núcleos epifisarios.
- Ortosis inmovilizadoras: Inmovilizan el segmento afectado, permitiendo precozmente la bipedestación, la marcha y otras actividades de la vida diaria.

- Ortesis de descarga: Permiten la descarga del peso corporal sobre el miembro inferior, transfiriendo la carga desde el encaje proximal hasta el estribo distal.
- Ortesis mixtas: Son una combinación de los tipos básicos.

Las ortesis se pueden fabricar de diferentes tipos de materiales. Estos se eligen en función de múltiples factores que hay que tener en cuenta, como son: el objetivo del artefacto ortésico terapéutico que desea el profesional médico; las funciones que realizará la ortesis en el miembro afectado, ya sea tanto de apoyo como estructural o de contacto; las características propias del material como son la resistencia, la densidad, la durabilidad o coste económico, etc.

Los materiales pueden ser metálicos y no metálicos.

- Las ortesis fabricadas a partir de elementos metálicos son duras, pesadas, maleables, buenas conductoras, tanto de calor como de electricidad. Sin embargo, estos metales empleados suelen ser vulnerables a la corrosión y al desgaste. Las ortesis metálicas más comunes son fabricadas a partir de aluminio, de acero y de algunas aleaciones de magnesio y titanio. Las principales desventajas que presentan son su elevado peso y una mayor dificultad a la hora de adaptarse a la fisiología del cuerpo, a diferencia de lo que ocurre con los termoplásticos.

- Las ortesis diseñadas con materiales no metálicos, como plásticos o poliméricos, son estables frente a la corrosión y el envejecimiento, y además son más ligeros y maleables que las metálicas por lo que pueden adaptarse mejor a la forma del cuerpo sobre la que se aplican. Son de fácil fabricación, lo que permite una reducción considerable de costes de fabricación. También se obtienen acabados más estéticos [4].

La prescripción de una ortesis es parte del proceso de rehabilitación para mejorar la capacidad funcional del usuario. Consta de tres fases principales:

- 1) Fase I: Identificación de los problemas, la patología y la discapacidad subyacente y se establece el pronóstico.
- 2) Fase II: Establecimiento del plan terapéutico global que incluye todos los tratamientos terapéuticos que se van a emplear (tratamiento farmacológico, infiltraciones, cirugías, terapias físicas, ortesis, productos de ayuda,...) y la instrucción del paciente.
- 3) Fase III: seguimiento de la evolución del paciente para evaluar el resultado funcional, y así observar la mejora producida en la capacidad funcional y estructural del paciente o en la calidad de vida de este. Es por este motivo que la prescripción de una ortesis debe ser un trabajo colaborativo entre paciente, médico prescriptor, técnico

ortesta y terapeutas que participen en el tratamiento (fisioterapeuta, terapeuta ocupacional, podólogo), si bien esta colaboración multidisciplinar no siempre es posible [4].

Posibles efectos secundarios:

- Lesiones cutáneas: eritema, erosiones, dermatitis por contacto o por acúmulo de sudor.
- Problemas derivados de una mala o incorrecta adaptación: úlceras por presión, dolor con el uso, lesiones por fricción de tejidos blandos, rozaduras, restricción respiratoria, molestias digestivas, trastornos circulatorios, etc.
- Problemas derivados de una inmovilización prolongada como debilidad muscular, atrofia cutánea, retracción muscular o contracturas, etc.
- Aumento del gasto energético del paciente debido al uso de la órtesis.
- Rechazo psicológico.

1.3. Antecedentes

La necesidad de ortesis y prótesis ha existido siempre y cuando la humanidad haya tenido que lidiar con la pérdida de la utilidad de las extremidades a través de una lesión o disfunción. Los simples bastones para caminar deben ciertamente haber sido usados como bastones básicos desde los primeros días de la humanidad y, de hecho, como lo demostró la losa de la estela egipcia mencionada anteriormente, se ha demostrado que las ayudas para caminar rudimentarias se han utilizado para gran parte de la historia humana registrada.

El arte de la ortesis se remonta al estudio de la fabricación práctica de férulas y refuerzos. Las ortesis y prótesis primitivas eran generalmente dispositivos simples contruidos con materiales a mano, generalmente de hueso, madera, cuero y en el tiempo metales. Actualmente se dispone de una amplia variedad de ortesis específicas para cada articulación. En específico, las ortesis de rodilla con mayor similitud a la ortesis que de diseña en este proyecto son las siguientes, ya que se utilizan para limitación del movimiento articular y para la corrección y estabilización de la rodilla. Esta lista de ortesis pertenece al catálogo de Medi, una compañía multinacional dedicada a la fabricación de ayudas médicas. Como esta compañía existen muchas más que se dedican al diseño y fabricación de ortesis [5].



Figura 3. Ortesis diseñadas por el grupo Medi

1.4. Objetivos

Al comienzo del proyecto se definieron una serie de objetivos que debían cumplirse para poder desarrollar un dispositivo funcional y completo. A continuación, se muestra la lista de objetivos propuestos y alcanzados en el desarrollo de este trabajo.

- 1) Realizar una revisión bibliográfica de las ortesis existentes destinadas a la rehabilitación del miembro inferior.
- 2) Consultar con profesionales sanitarios y especialistas en rehabilitación el proyecto que se quiere realizar y obtener su asesoramiento clínico para poder realizar un diseño óptimo que ayude a la rehabilitación.
- 3) Elegir el diseño de la estructura de la ortesis.
- 4) Seleccionar los componentes hardware destinados a la recolección, procesamiento y envío de datos.
- 5) Elegir plataforma de desarrollo de la aplicación móvil.
- 6) Crear la estructura de la ortesis gracias a la impresora 3D.
- 7) Crear el programa software en el entorno de Arduino.
- 8) Crear interfaz de la aplicación.
- 9) Realizar protocolo de pruebas para un óptimo funcionamiento entre la aplicación y los dispositivos hardware escogidos. Corregir los fallos o disfunciones encontradas. Dentro de este protocolo de pruebas, el dispositivo diseñado debe conseguir:

- a. Registrar el movimiento de la rodilla gracias a la ortesis sensorizada.
 - b. Procesar los datos recogidos y enviarlos a la aplicación móvil.
 - c. La aplicación móvil debe mostrar los resultados y ayudar al paciente a través de mensajes para realizar un óptimo ejercicio de rehabilitación.
 - d. El proyecto desarrollado también debe ser capaz de llevar un registro de los ejercicios realizados para que el profesional rehabilitador pueda hacer una monitorización del progreso de la afección.
- 10) Probar el dispositivo final tanto en sujetos de control como en pacientes.
- 11) Realizar una encuesta de satisfacción del producto.
- 12) Documentar todo el procedimiento y recopilarlo en una memoria.

1.5. Metodología

El proyecto comienza con una revisión bibliográfica sobre las ortesis existentes. Una vez escogida la ortesis que se quiere diseñar y el objetivo que debe cumplir se contacta con los médicos Rehabilitadores del Servicio de Rehabilitación de Hospital Universitario Río Hortega y del Centro Regional de Medicina Deportiva y se realiza una presentación en Powerpoint para mostrarles el objetivo y la finalidad de la órtesis, así como su diseño y funcionalidad. Tras la presentación se los médicos aportan su punto de vista clínico y las posibles funcionalidades a desarrollar para su aplicabilidad en la práctica clínica sobre el dispositivo a desarrollar.

A continuación, se desarrolla el dispositivo hardware y software según las especificaciones previamente acordadas con los médicos rehabilitadores. Una vez corregidas las deficiencias y obstáculos encontrados durante el desarrollo se realizan las pruebas en diferentes sujetos para comprobar su funcionamiento. Finalmente se realiza una recopilación de los resultados obtenidos, se plantean las conclusiones finales y se redacta la memoria del trabajo.

2. PLANTEAMIENTO Y ESTRUCTURA

A lo largo de este proyecto se han ido desarrollando diversas soluciones para conseguir un dispositivo con un funcionamiento óptimo y así poder alcanzar los objetivos planteados inicialmente.

2.1. Solución inicial

Inicialmente, se propuso el uso de una placa de Arduino Uno como el procesador del programa hardware, junto con una conexión bluetooth entre el dispositivo hardware y la aplicación móvil. Además, se seleccionó un sensor que determinaba la posición angular de la rodilla que consistía en un encoder rotativo. A continuación, se expondrán dichos dispositivos hardware que han sido probados y descartados puesto que no se alcanzaba el objetivo deseado de conseguir un funcionamiento óptimo del dispositivo.

- **Placa Arduino Uno:**

Es una placa basada en un microcontrolador. Una de las principales características de las placas desarrolladas por Arduino es que son de código abierto y ésta, en concreto, está basada en el microchip ATmega328P. La placa contiene varios conjuntos de pines de entrada y salida tanto digitales como analógicos que pueden conectarse a diferentes circuitos tal y como aparece en la siguiente figura (Figura 4). Esta placa es una opción muy buena para personas principiantes en proyectos de Arduino, ya que es muy robusta frente a fallos [6].

El problema principal encontrado en esta placa ha sido la memoria SRAM (static random access memory): Se trata de las variables locales y datos parciales. La parte de la memoria en la que el sketch crea y manipula las variables durante la ejecución del programa. Arduino Uno consta solo de 2K bytes de memoria SRAM. Si se ejecuta el programa desarrollado en este proyecto, consume más del 70% de la memoria SRAM del Arduino Uno. Ha sido posible comprobar el fallo de la ejecución de dicha placa cuando el programa superaba el 66% de memoria SRAM ocupada. En caso de que el programa no supere este 66% la ejecución se desenvuelve sin ningún problema realizando todas las funciones programadas. En caso contrario, la ejecución se queda bloqueada en la función de Setup. Por ello ha sido necesario un cambio de placa a una con mayor memoria SRAM. Se ha escogido la placa Arduino Mega2560 ya que contiene cuatro veces más de memoria SRAM que Arduino Uno, es decir, 8Kbytes.

Para proyectos con programas menos pesados Arduino Uno sigue siendo una opción recomendable, sobre todo para desarrolladores primerizos.

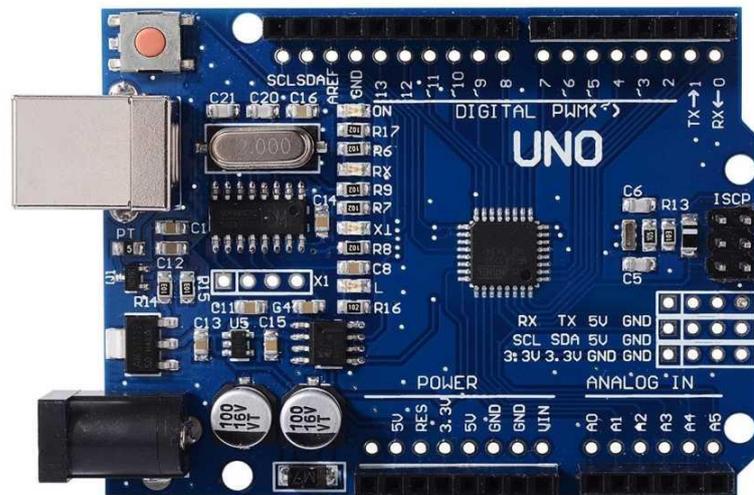


Figura 4. Placa Arduino Uno

- **Módulo Bluetooth HC-06:**

Consiste en un pequeño módulo cuya función es convertir los datos que le llegan a través de Bluetooth hacia el puerto serie y viceversa. Dicho módulo sólo puede trabajar en modo esclavo, es decir, sólo puede conectarse a un dispositivo maestro. En este caso el dispositivo maestro sería el dispositivo móvil en el que esté descargada la aplicación. Su función era realizar la conexión entre la placa de Arduino y la aplicación móvil. Este módulo Bluetooth puede conseguir de forma sencilla la transmisión inalámbrica de datos en serie. Trabaja a una frecuencia de funcionamiento de 2,4 GHz. Adopta el estándar Bluetooth 2.0 + EDR (Enhanced Data Rate), cuya tasa de transmisión llega a los 3Mbps [7] [8]. Este dispositivo no cumplía con los requisitos necesarios para una buena comunicación entre la placa y la aplicación por lo siguiente:

- 1) El tamaño del paquete de datos que se envía es en ocasiones demasiado grande para que el módulo lo pueda transmitir, por lo que daba problemas de estabilidad en la conexión, es decir estaba constantemente conectándose y desconectándose.
- 2) No proporcionaba conexión a internet, por lo que la placa no podía enviar el email con los datos del ejercicio al médico rehabilitador.

Por ello se sustituyó por el módulo WiFi ESP8266 esp-01, el cual subsana las carencias del módulo bluetooth HC-06. En las siguientes imágenes se puede ver una foto del módulo HC-06 junto con la conexión que se realizaría a una placa de Arduino Uno (Figuras 6 y 7).



Figura 5. Módulo HC-06

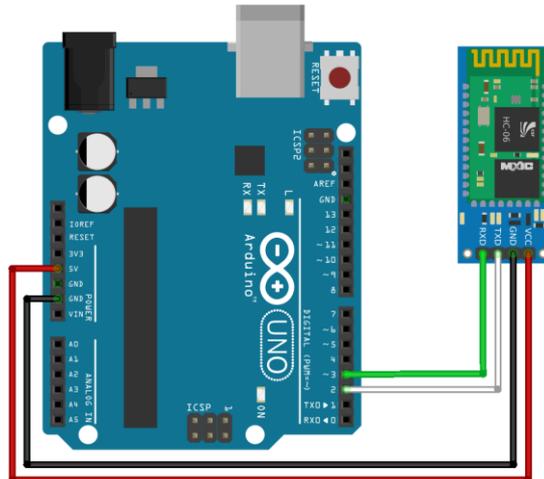


Figura 6. Conexión HC-06 con placa Arduino Uno

- **Encoder rotativo:**

Un encoder rotativo consiste en un dispositivo que permite determinar la velocidad angular y la posición de un giro. Tiene un aspecto exterior muy parecido a un potenciómetro, sin embargo, su electrónica y su comportamiento es totalmente dispar. Específicamente, este encoder se basa en la generación de pulsos digitales durante la rotación hasta un determinado ángulo. Por cada vuelta dicho encoder genera 256 pulsos. Este número puede variar dependiendo del encoder utilizado.

Dentro del encoder se encuentran dos escobillas que deslizan sobre una pista metálica ranurada. Junto con las escobillas se encuentra una bola metálica cuya función es cerrar el contacto entre las dos escobillas actuando (Figura 7), así como un pulsador [9].



Figura 7. Encoder rotativo

Dichos encoders, también llamados encoders de cuadratura, debido a que disponen de dos canales (Canal A y B) y, a la salida de estos canales, obtenemos dos señales cuadradas desplazadas entre sí 90° eléctricos como se puede observar en la siguiente gráfica (Figura 8). De forma gráfica se podría representar de la siguiente manera.

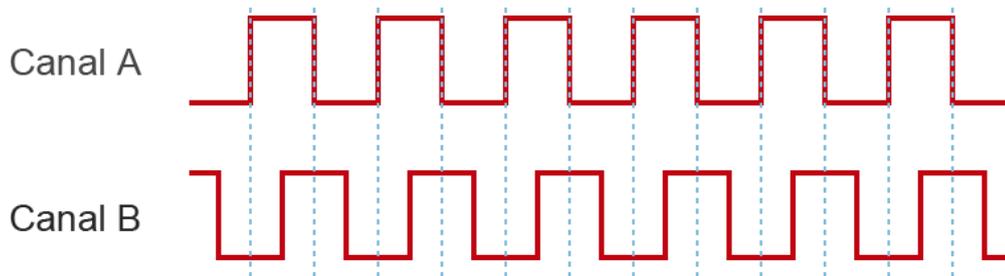


Figura 8. Señales en canales A y B

Así como este tipo de encoders permiten determinar la posición y la velocidad, también permiten identificar el sentido de giro. Como ejemplo práctico, se observan las siguientes gráficas, en las que se toma como referencia los flancos de subida y de bajada del canal A. La primera se realiza el giro en sentido CW (Figura 9) y la segunda en sentido CCW (Figura 10) produciéndose los eventos t_0 , t_1 ... Se puede observar que cuando se realiza el giro con sentido CW, en los eventos tomados como referencia, las señales son totalmente inversas, en cambio con sentido CCW las señales son idénticas. De esta manera se consigue realizar la distinción del sentido del giro [9].

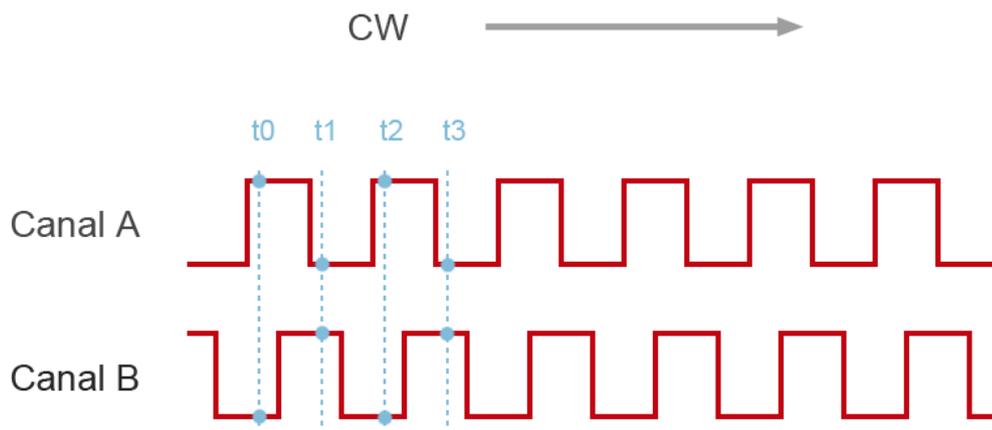


Figura 9. Eventos producidos en sentido de giro CW

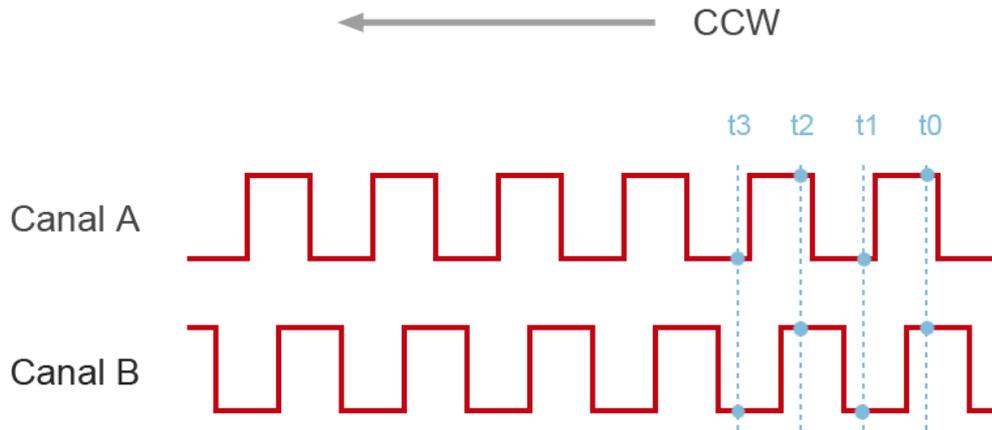


Figura 10. Eventos producidos en sentido de giro CCW

El problema con estos encoders es que trabajan de forma incremental, es decir que al inicializarlo siempre toma la posición en la que se encuentra como cero de referencia, y no guarda la posición en la que se encontraba una vez desconectado. Por lo que, para poder registrar los ángulos de la rodilla, el dispositivo usado como sensor debe saber en todo momento cuál es su posición actual, y no depender de la posición en la que se ha encendido. Por ello se sustituyó este encoder rotativo de cuadratura por un potenciómetro lineal. A continuación, se representa un diagrama de cómo se realizaría la conexión entre el encoder y la placa de Arduino Uno (Figura 11).

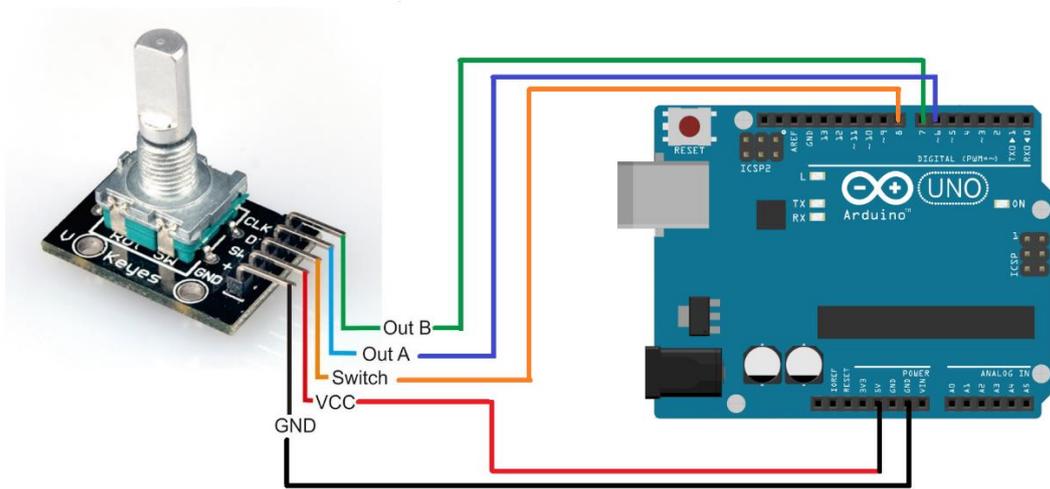


Figura 11. Conexión entre encoder rotativo y placa de Arduino Uno

2.2. Solución escogida

Finalmente, tras descartar los anteriores dispositivos por los problemas comentados anteriormente, se consigue el siguiente diseño final:

Hardware:

- **Placa Arduino Mega2560** como procesador principal del programa, en él se ejecuta el sketch con todas las funciones programadas en el entorno de Arduino y se procesan los datos recogidos por el potenciómetro. Estos datos los transforma y los manda hacia el módulo WiFi ESP8266 esp-01 para su posterior transmisión a la aplicación móvil.
- **Módulo WiFi ESP8266 esp-01** como medio de conexión entre el hardware y la aplicación. Encargado de transmitir los datos enviados desde la placa de Arduino hacia la aplicación Blynk.
- **Potenciómetro lineal de 10kohms** como medidor de la posición angular de la rodilla. Realiza la función de sensor registrando los movimientos realizados durante el ejercicio y transmitiéndolos a la placa de Arduino.
- **Impresión 3D** como estructura de la ortesis. Consiste en el cuerpo rígido, fabricado a partir del material PLA, que rodea la articulación y donde se encuentra acoplado el potenciómetro.

Software:

- **Aplicación Blynk** como plataforma de desarrollo de la aplicación. Aquí se desarrolla la interfaz de la aplicación, junto con el servidor y las librerías asociadas.
- **Entorno de programación de Arduino** donde se realiza el desarrollo del sketch, procesado por la placa de Arduino.

Ejecución del dispositivo: Todos los apartados que se describen posteriormente tienen un papel fundamental en resultado final del proyecto. El procedimiento que debe realizarse para conseguir un correcto uso del dispositivo es el siguiente:

- 1) Enchufar la caja con los componentes hardware a la red eléctrica y encender sus dos botones acoplados.
- 2) Conectar la ortesis a la caja.

- 3) Abrir la aplicación Blynk y realizar la conexión entre el Hardware de la caja y de la aplicación a través de la conexión WiFi. En caso de que sea infructuosa esta conexión reiniciar la caja apagando y encendiendo los botones.
- 4) Una vez asegurada la conexión, asegurar la ortesis a la pierna del paciente.
- 5) Seleccionar uno de los ejercicios de rehabilitación disponibles en la aplicación.
- 6) Realizar dicho ejercicio tantas veces como repeticiones aparezcan en la aplicación.
- 7) Finalmente, una vez terminados todos los ejercicios de rehabilitación, desenchufar la caja de la red eléctrica y desconectar la ortesis de la caja.

En los siguientes capítulos 3 y 4 se describirán los componentes Hardware comentados anteriormente junto con el desarrollo del software final para una comprensión más profunda y exhaustiva del funcionamiento del sistema.

3. HARDWARE

3.1. Arduino Mega 2560

Los datos recogidos por el potenciómetro colocado en la ortesis se transmiten al Arduino Mega 2560. Ésta es una placa electrónica basada en el microcontrolador ATmega2560. Como se puede observar en la siguiente figura (Figura 12), se compone principalmente de 54 pines digitales de entrada y salida, 16 pines analógicos de entrada, 4 UARTs (puertos serie hardware) y un botón de reset. También consta de un oscilador que trabaja a una frecuencia de funcionamiento de 16MHz, una conexión de alimentación, un conector ICSP y una conexión de USB tipo B [10].

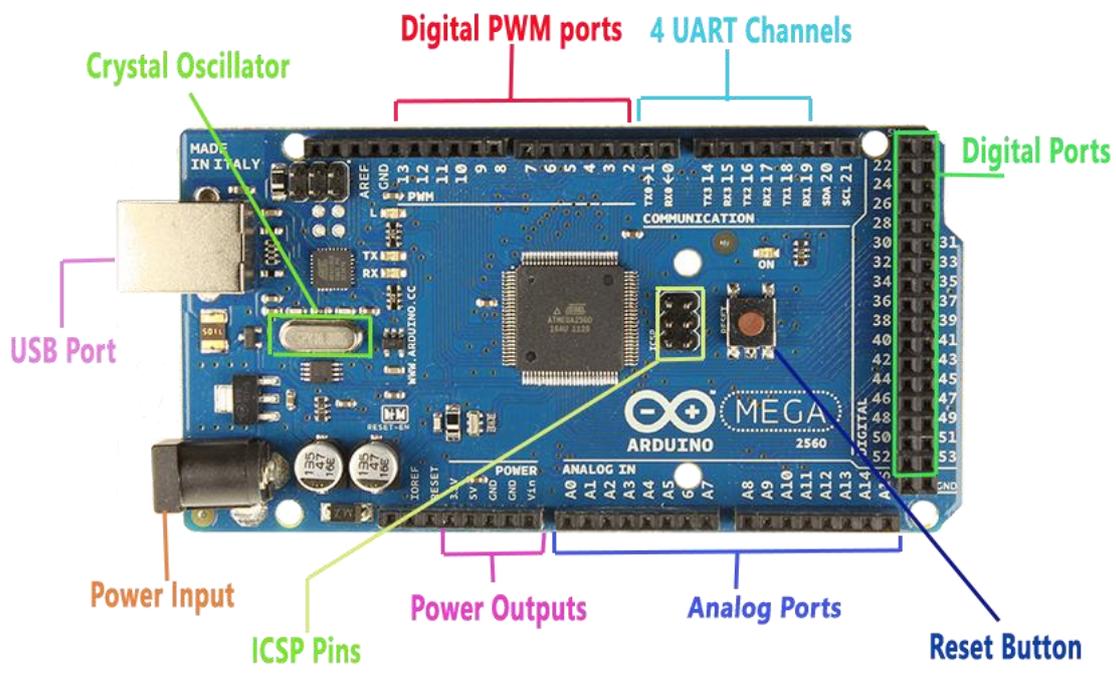


Figura 12. Placa Arduino Mega 2560

ESPECIFICACIONES TÉCNICAS

Microcontrolador	ATmega2560
Tensión de trabajo	5V
Tensión de entrada recomendada	7-12V
Tensión de entrada límite	6-20V
Pines digitales	54
Pines de entrada analógicos	16
Memoria Flash	256KBytes
SRAM	8KBytes
EEPROM	4KBytes
Velocidad de reloj	16MHz
Largo	101.52mm
Ancho	53.3mm
Peso	37g

Para la programación de la placa se realiza a través del software de Arduino (IDE), utilizando la conexión USB tipo B de la que dispone conectada al ordenador para poder cargar el sketch. El programa cargado dentro del Arduino se indica en el capítulo 4.3.

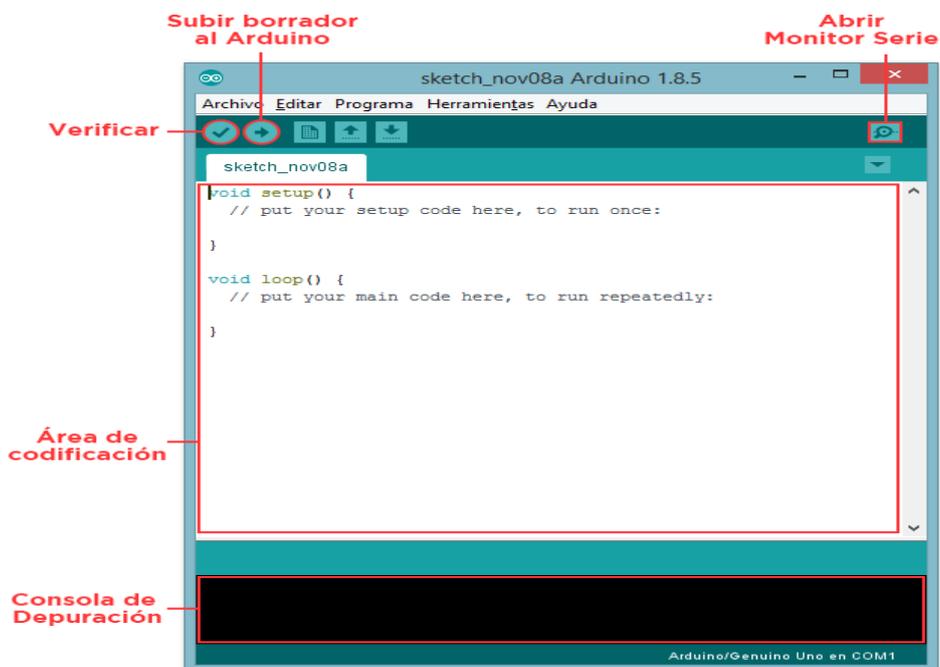


Figura 13. Interfaz de programación de Arduino

ALIMENTACIÓN

El Mega 2560 puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

La alimentación externa (no USB) puede venir de un adaptador de CA a CC o de una batería. El adaptador se puede conectar al enchufe de 2,1 mm de centro-positivo en la clavija de alimentación de la placa. Los cables desde una batería pueden ser insertados en GND y en el pin Vin del conector de alimentación.

La tarjeta puede funcionar con un suministro externo de 6 a 20 voltios. Si se alimenta con menos de 7 V, sin embargo, el pin de 5V puede suministrar menos de cinco voltios y la placa se puede volver inestable. Si se utiliza más de 12 V, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios [10].

Los pines de alimentación son los siguientes:

- Vin. La tensión de entrada a la placa cuando se utiliza una fuente de alimentación externa.
- 5V. Este pin es una salida de 5 V regulada por el regulador de la placa. La placa puede ser alimentada desde el conector de alimentación de CC (7 - 12 V), a través del conector USB (5 V), o por el pin VIN de la placa (7-12V).
- 3V3. Un suministro de 3,3 voltios generado por el regulador de la placa. El consumo de corriente máximo es de 50 mA.
- GND. Son los pines de tierra.
- IOREF. Este pin en la placa proporciona la referencia de tensión con la que opera el microcontrolador.

En este proyecto, la placa se alimenta a través de la conexión USB conectada a un cargador de móvil con adaptador a USB como el que se puede ver en la siguiente imagen (Figura 14). El circuito del cargador de móvil se encarga de transformar los 220V que llegan del enchufe de pared al que esté conectado la caja a los 5V necesarios para alimentar el Arduino Mega 2560.

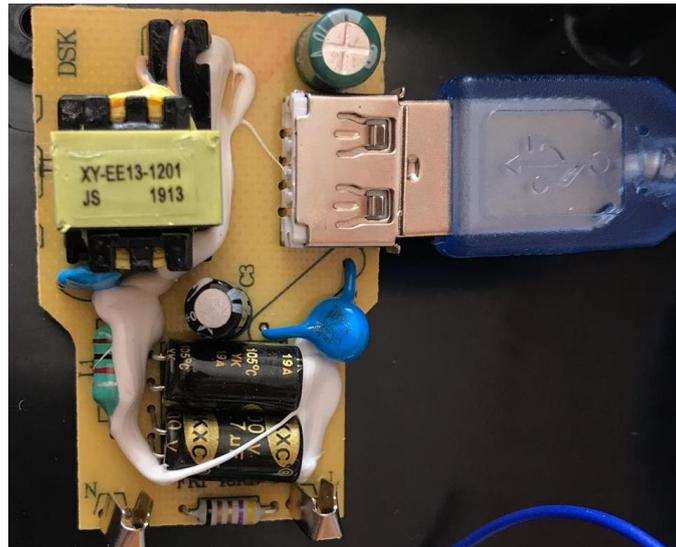


Figura 14. Regulador de voltaje 220V-5V

La placa Mega 2560 tiene un polyfusible reajutable destinado a proteger a los puertos USB de su ordenador de cortocircuitos y sobrecorrientes que puedan producirse. A pesar de que la mayoría de los ordenadores tienen su propia protección interna, el polyfusible proporciona una capa adicional de protección [10].

MEMORIA

Para realizar con éxito el dispositivo, la memoria de la placa es un factor determinante ya que el programa que se necesita ejecutar realiza un uso intensivo de la memoria SRAM. La memoria SRAM (static random access memory): Se encarga de almacenar variables locales y datos parciales. Es la zona de memoria donde el sketch crea y manipula las variables cuando se ejecuta. La razón principal por la que se cambió de la placa Arduino Uno a la Arduino Mega 2560 es debido a que la Mega 2560 tiene 8Kbytes de memoria SRAM mientras que la Arduino Uno sólo dispone de 2Kbytes. El microcontrolador ATmega2560 consta de 256Kbytes de memoria flash (encargada de almacenar el código del programa), 4Kbytes de EEPROM (su función es mantener datos después de un reset de la placa), y, como ya se ha comentado, 8Kbytes de SRAM. En comparación con otras placas desarrolladas por Arduino, se puede observar en la siguiente tabla que el microcontrolador del que dispone la Arduino Mega 2560 dispone de la mayor capacidad de memoria [10].

Arduino	Procesador	Memoria Flash	SRAM	EEPROM
UNO, UNO Ethernet, Menta, Boarduino	ATmega328	32KBytes	2KBytes	1KByte
Leonardo, Micro, Flora, Teensy...	ATmega32U4	32KBytes	2.5KBytes	1KByte
Mega, MegaADK	ATmega2560	256KBytes	8KBytes	4Kbytes

CONEXIONES

En la siguiente figura (Figura 15) se puede observar el diagrama de conexiones que hay a la placa para recibir los datos recogidos por el potenciómetro, y transmitir dichos datos procesados a la aplicación móvil a través del módulo WiFi ESP8266 esp-01.

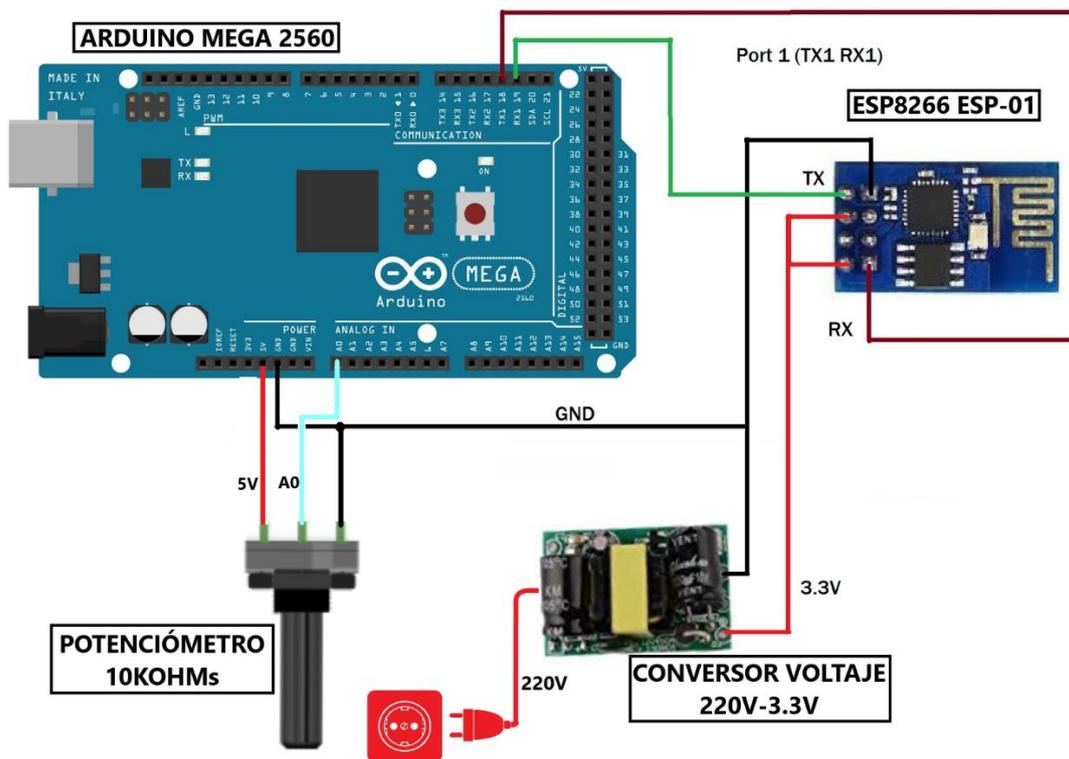


Figura 15. Conexiones a la placa de Arduino Mega 2560

3.2. Potenciómetro

Un potenciómetro es una resistencia variable ajustable manualmente con 3 terminales. Dos terminales están conectados a ambos extremos de un elemento resistivo, y el tercer terminal se conecta a un contacto deslizante, llamado cursor, que se mueve sobre el elemento resistivo. La posición del cursor determina el voltaje de salida del potenciómetro. El potenciómetro funciona esencialmente como un divisor de voltaje variable. El elemento resistivo puede verse como dos resistencias en serie (resistencia del potenciómetro), donde la posición del cursor determina la relación de resistencia entre la primera y la segunda resistencia, tal y como se observa en el siguiente circuito (Figura 16) [11].

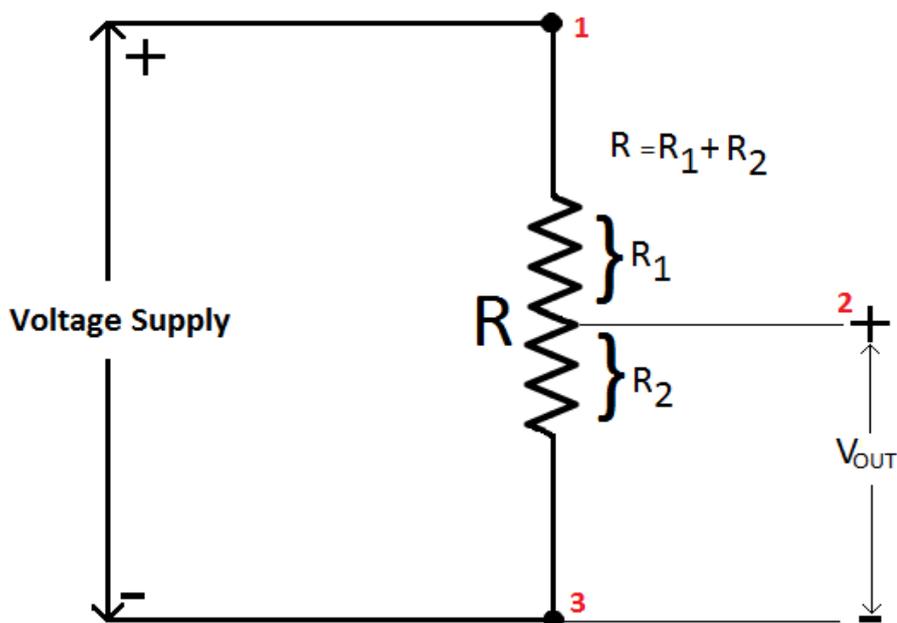


Figura 16. Circuito de un potenciómetro

Se utilizan diferentes materiales para construir potenciómetros, incluyendo composición de carbono, cermet, bobinado, plástico conductor o película metálica. La forma más común de potenciómetro es el rotatorio de una sola vuelta como el de la siguiente imagen (Figura 17). En este proyecto se utiliza un potenciómetro rotatorio de 10Kohms y su función es recoger la posición angular de la rodilla durante la realización del ejercicio de rehabilitación. Una de las características más importantes de este tipo de potenciómetros es su larga duración gracias a un lento deterioro por el uso.

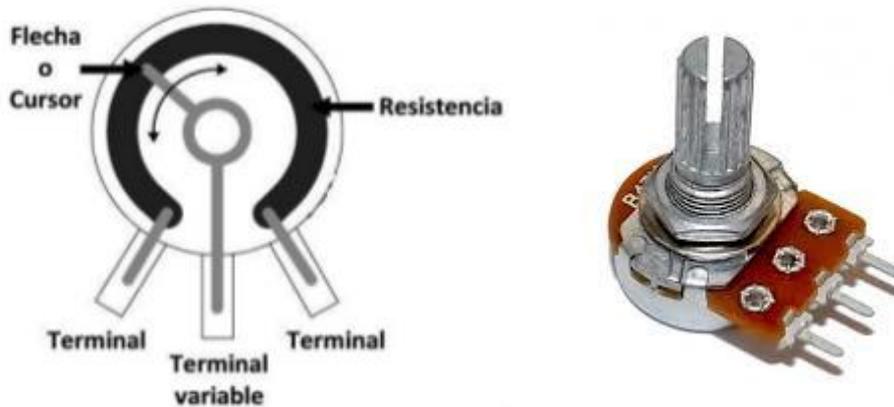


Figura 17. Potenciómetro rotatorio

Este potenciómetro realiza una variación lineal, como se señala en el siguiente gráfico (Figura 18). Es decir, la resistencia es directamente proporcional al ángulo de giro. Esta propiedad es fundamental para luego poder hacer la transformación del voltaje en la salida del potenciómetro a la posición angular correcta del cursor [11].

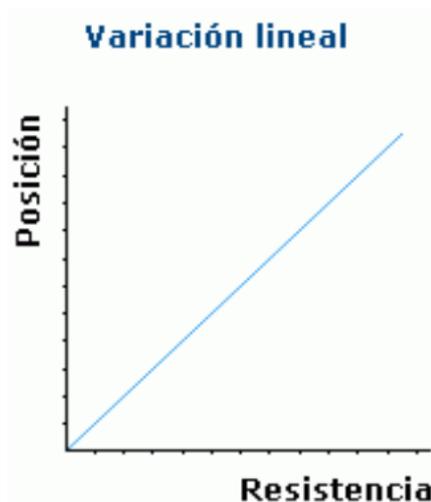


Figura 18. Variación del potenciómetro rotatorio usado en el proyecto

ACOPLAMIENTO A LA ESTRUCTURA DE IMPRESIÓN 3D

Para poder ajustar el potenciómetro a la pieza articular 3D de la ortesis se realiza un agujero en la pieza con las dimensiones del potenciómetro. Tras esto se realiza un corte transversal en el eje del potenciómetro tal y como aparecen en las siguientes figuras (Figuras 19 y 20).



Figura 19. Engranaje impreso en 3D



Figura 20. Potenciómetro rotatorio

Una vez acoplado el potenciómetro a la pieza 3D se suelda una punta metálica insertada en la ranura previamente realizada con el propósito de ensamblar el eje del potenciómetro a la parte interior de la pieza articular 3D y el cuerpo del potenciómetro a la parte exterior, tal y como aparece en la siguiente figura (Figura 21). Una vez conseguida esta unión, el potenciómetro girará a la vez que se articula la ortesis.

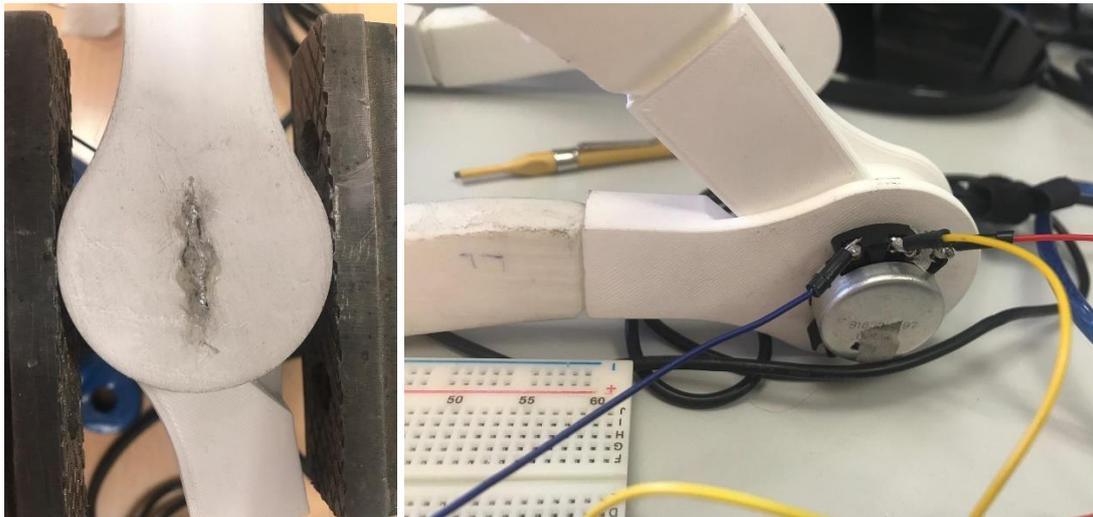


Figura 21. Acoplamiento del potenciómetro a la ortesis impresa

3.3. Estructura impresión 3D

Para poder crear una ortesis que cumpla con los objetivos de un dispositivo de uso doméstico, (es decir que fácil de usar, ligero, cómodo, ajustable a la estructura fisiológica de cada paciente concreto y económico), y que además sea posible sensorizarlo para poder controlar los ejercicios de rehabilitación, se llega a la conclusión que la impresión en 3D del cuerpo de la ortesis es la mejor solución.

Las herramientas que se necesitan para poder crear las piezas que compondrán la ortesis son las siguientes: Tinkercad, impresora 3D y Ultimaker Cura 4.1.0.

- **Tinkercad:**

Tinkercad es un software gratuito online creado por Autodesk, una de las empresas líderes en software de diseño 3D. Su interfaz (Figura 22) es muy intuitiva y contiene herramientas básicas fáciles de emplear y muy completas. Como se puede observar en la siguiente figura, la interfaz consta a la derecha de un panel de herramientas que sirven para crear y modificar las piezas necesarias para el prototipo [12].

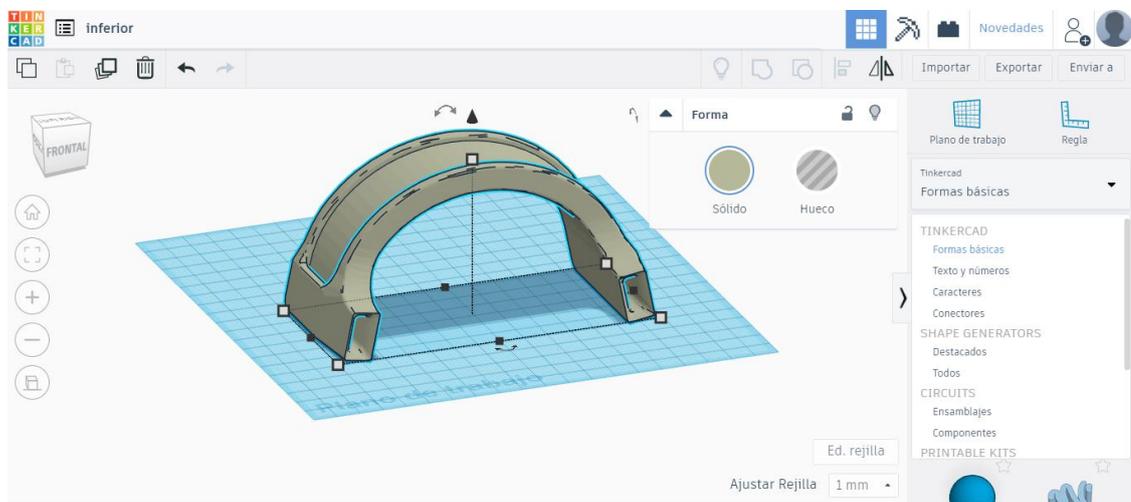


Figura 22. Interfaz de Tinkercad

Las piezas que componen la ortesis del proyecto han sido sacadas de Thingiverse, una página web que dispone de diseños de piezas en 3D gratuitos. Han sido adaptadas y modificadas según los criterios que se necesitaban, utilizando la plataforma de Tinkercad. [26] Una vez terminado el rediseño de las piezas se exportan los archivos .STL y .OBJ que serán necesarios para su posterior laminación en el programa Ultimaker Cura.

- **Impresora 3D:**

Las piezas se realizan con la impresora cedida por el departamento de Teoría de la Señal y Comunicación e Ingeniería Telemática. Esta impresora es un modelo BQ Prusa i3 Hephestos (Figura 23) que emplea la tecnología FFF (Fused Filament Fabrication) o Modelado por Deposición Fundida [13].



Figura 23. Impresora 3D BQ Prusa i3 Hephestos

El material que se ha empleado es un termoplástico denominado PLA (ácido poliláctico) formado por moléculas de ácido láctico y obtenido a partir de almidón de productos vegetales. Este material es bastante económico y biodegradable. Se suministra en forma de un filamento cilíndrico, y como aparece en la siguiente imagen (Figura 24), enrollado en forma de bobina. El extrusor de la impresora se calienta hasta una temperatura de 200° para fundir el filamento, dando lugar a un flujo de material fundido con grosor mucho menor.



Figura 24. Bobinas de PLA

- **Ultimaker Cura 4.1.0:**

Cura es un software de laminación para procesar los archivos de diseño en 3D y así generar cada capa que será fabricada por la impresora 3D. Una vez compilado el archivo se exporta en formato .GCode a una tarjeta SD. Dicha tarjeta se introduce en la impresora y tras la selección del archivo, comenzará su impresión capa por capa. Admite los formatos STL /OBJ / DAE y AMF [14].

Esta herramienta de laminación ha sido desarrollada por Ultimaker y es gratuita y de código abierto. Es necesario configurarla con las características de la impresora que se utilizará para imprimir la pieza. También dispone de otras opciones para controlar la impresión, como por ejemplo la velocidad de impresión, la cantidad de relleno de la pieza y los soportes de la pieza. En función de las características escogidas el tiempo estimado de impresión varía.

En la siguiente ilustración (Figura 25) se encuentra la interfaz del programa Ultimaker Cura 4.1.0. donde se puede ver a la derecha de la imagen las diferentes opciones de laminado de las que dispone.

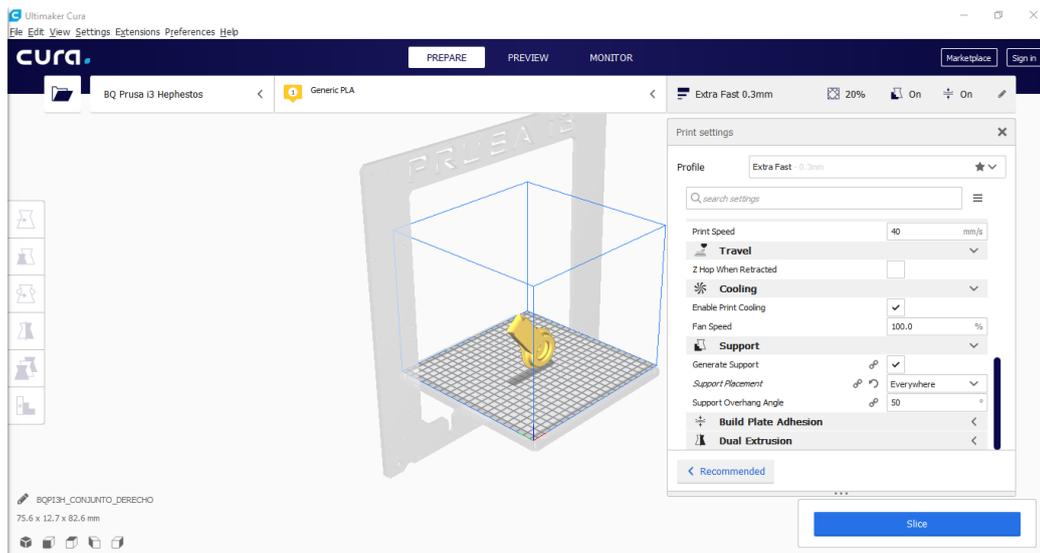


Figura 25. Interfaz Ultimaker Cura

Las siguientes imágenes muestran las diferentes piezas fabricadas por la impresora 3D junto con su ensamblaje final. Para su correcta impresión ha sido necesario realizar una calibración previa de la impresora. Para pegar todas las piezas se ha utilizado una pistola de termofusible. Los planos en 3D de las piezas se encuentran en el apartado de ANEXOS.



Figura 26. Soporte inferior derecho



Figura 27. Soporte inferior izquierdo



Figura 28. Soporte superior izquierdo



Figura 29. Soporte superior derecho



Figura 30. Soporte superior



Figura 31. Soporte inferior

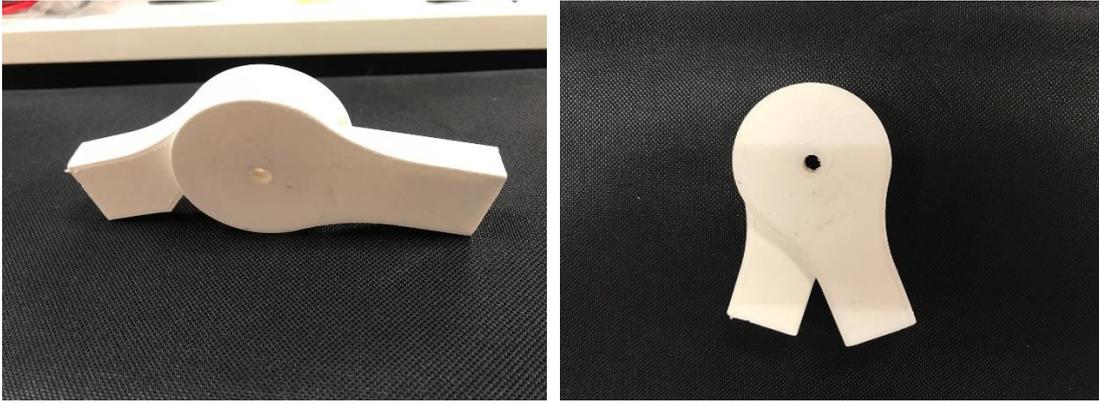


Figura 32. Engranaje derecho

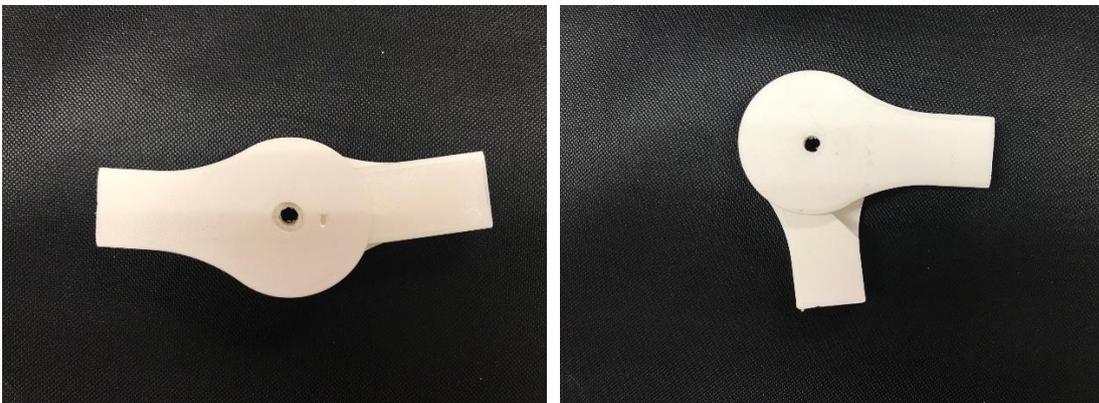


Figura 33. Engranaje izquierdo

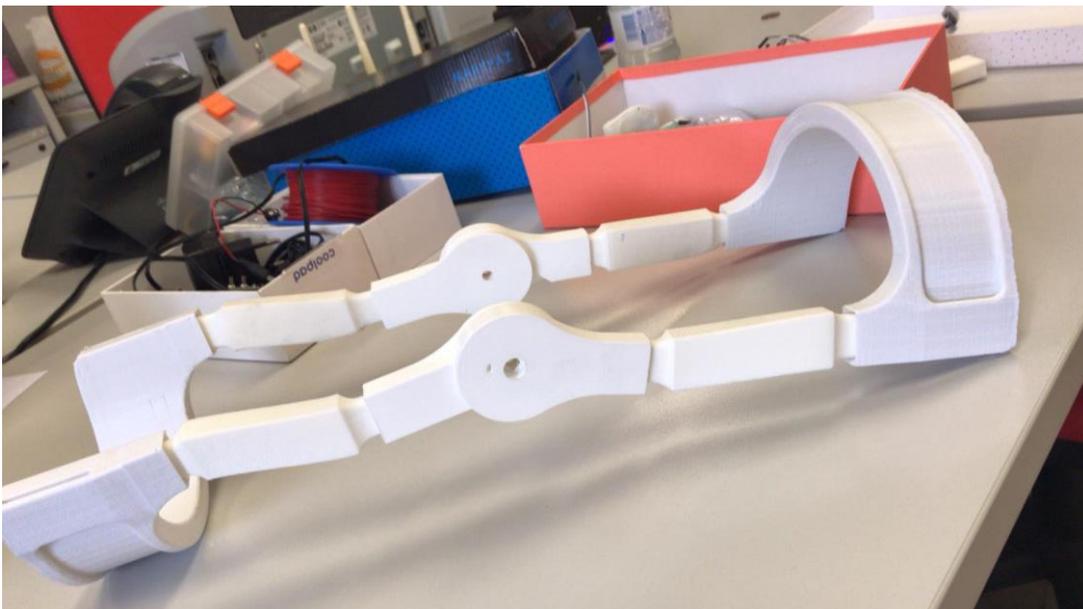


Figura 34. Conexión final de las piezas de la ortesis

3.4. Recubrimiento

Uno de los efectos indeseados en el uso de ortesis son rozaduras y lesiones en la piel debido a un uso continuado de ésta. Para evitar este efecto secundario se recubre el interior de la ortesis impresa en 3D con un material acolchado y suave al tacto, haciendo que aumente la comodidad del usuario al usar el aparato. Como se puede ver en las siguientes imágenes se ha colocado este material en los puntos con más riesgo a rozadura (Figura 35).



Figura 35. Recubrimiento de neopreno de la ortesis

3.5. Módulo WIFI

Para realizar la conexión inalámbrica entre la placa Mega 2560 y la aplicación desarrollada en la plataforma Blynk se utiliza un módulo WiFi denominado ESP8266 ESP-01. Es el encargado de dirigir los paquetes de datos desde la aplicación al hardware y viceversa, consiguiendo así una buena comunicación por ambas partes.

ESP8266 es el nombre del microcontrolador diseñado por la empresa Espressif Systems. El ESP8266 se diseñó con el propósito de crear una solución autónoma de redes WiFi para poder dotar de conexión a internet a placas basadas en microcontroladores (como la Arduino Mega 2560), además de ser capaz de ejecutar aplicaciones de forma independiente. Para simplificar la programación y uso de este procesador se desarrollan placas prefabricadas específicas. Esto da lugar a varias versiones de ESP8266 (Ej: ESP-01, ESP-02...) en las que todas con el mismo procesador. Las diferencias que se encuentran entre estas placas son el número de pines GPIO y la forma de exponerlos, las dimensiones de la placa, la cantidad de memoria flash y otras consideraciones relativas a su fabricación. Pero, desde una perspectiva de programación, todas son iguales. [27]

PROCESADOR:

El System on a Chip (SoC) ESP8266EX (Figura 36) usa un procesador Tensilica Xtensa L106. Esto es un microcontrolador de 32 bit con instrucciones de 16 bit. El procesador tiene una frecuencia de funcionamiento de 80 MHz por defecto, aunque puede llegar a hasta los 160 MHz. Con respect a la memoria, consta de 80kB de DRAM (Data RAM), y 35kB IRAM (Instruction RAM) [15].

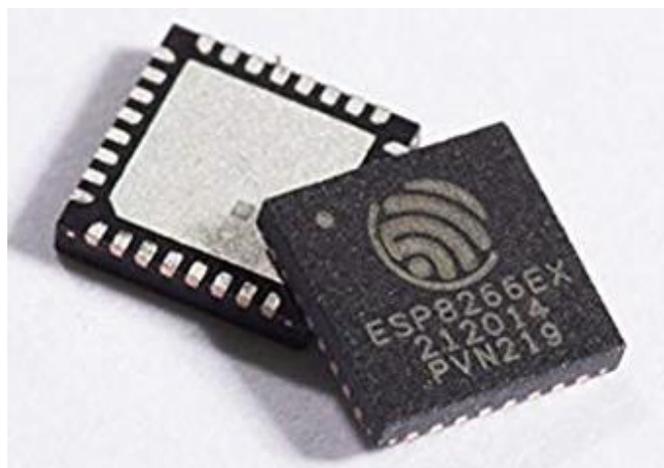


Figura 36. Procesador ESP8266EX

Placa ESP-01:

En este proyecto se utiliza la placa prefabricada ESP-01. Es una de las placas más populares que integran el procesador ESP8266 [16]. Sus características principales son:

- ESP8266 es el microcontrolador del módulo ESP-01.
- 8 pines donde se conecta la alimentación, sensores y transmisión de programa. 2 de ellos son pines GPIO (GPIO0 y GPIO2)
- BG25Q80A es la memoria flash donde residen los programas o sketches. El procesador ESP8266 no dispone de este tipo de memoria y por eso es un chip aparte.
- LEDs que informan de si está alimentado y de la transmisión de datos (Tx y Rx).
- Tipo de antena: impresa
- Dimensiones de la placa: 14,3 x 24,8 x 3 mm
- Distancia entre pines: 2,54 mm

En las siguientes imágenes (Figuras 37 y 38) se muestran las partes fundamentales de la placa ESP-01 junto con sus dimensiones.

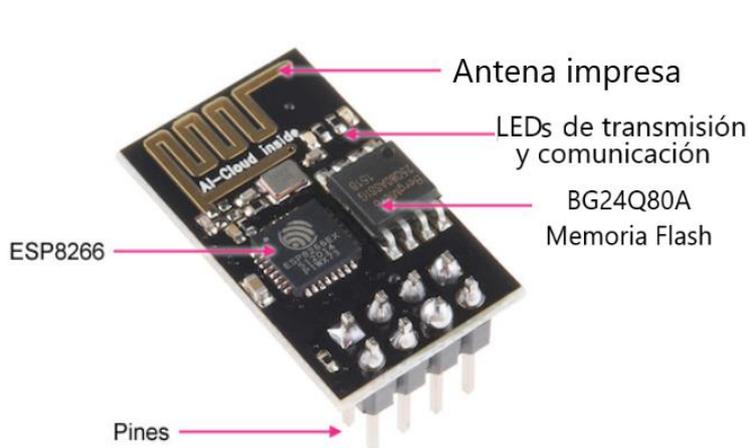


Figura 37. Partes fundamentales de la placa ESP-01

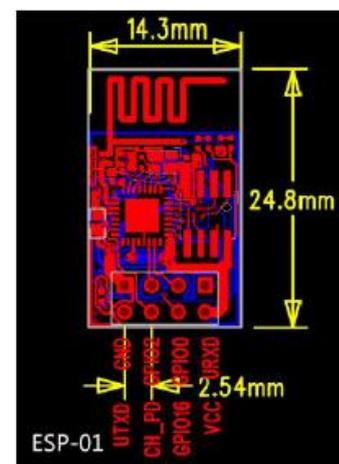


Figura 38. Dimensiones de la placa ESP-01

En la siguiente imagen (Figura 39) se muestran los diferentes pines que conforman la placa seguida de una descripción del funcionamiento de cada uno de ellos [17].

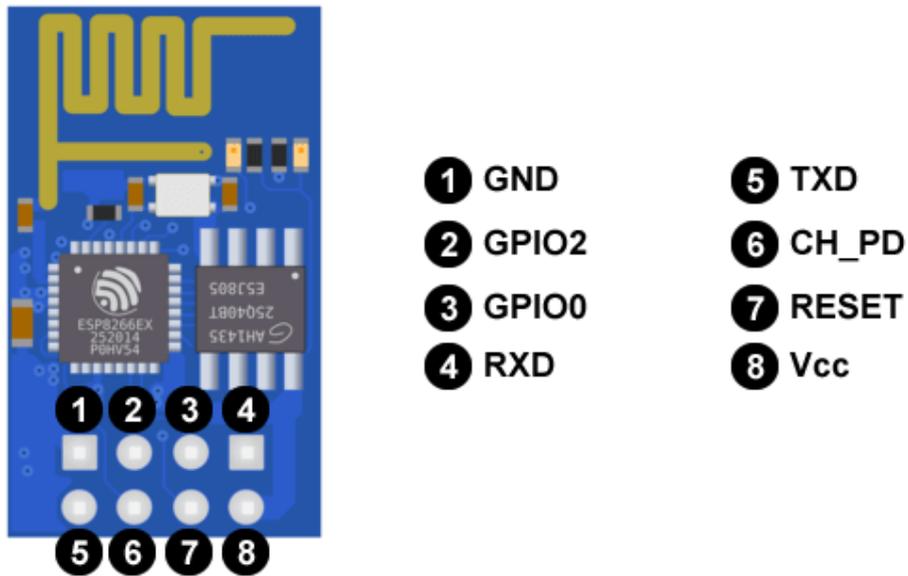


Figura 39. Pines de la placa ESP-01

- 1) GND es la toma de tierra.
- 2) GPIO2 es una entrada salida de propósito general. Es el pin digital número 2.
- 3) GPIO0 es una entrada salida de propósito general. Es el pin digital número 0.
- 4) RXD es el pin por donde se van a recibir los datos del puerto serie. Trabaja a 3,3 V. También se puede utilizar como pin digital GPIO: sería el número 3.
- 5) TXD es el pin por donde se van a transmitir los datos del puerto serie. Trabaja a 3,3 V. También se puede utilizar como pin digital GPIO: sería el número 1.
- 6) CH_PD pin para apagar y encender el ESP-01: si lo ponemos a 0 V (LOW) se apaga, y a 3,3 V (HIGH) se enciende.
- 7) RESET pin para resetear el ESP-01: si lo ponemos a 0 V (LOW) se resetea.
- 8) Vcc es por donde alimentamos el ESP-01. Funciona a 3,3 V y admite un máximo de 3,6 V. La corriente suministrada debe ser mayor que 200 mA.

ALIMENTACIÓN

Para alimentar el módulo ESP-01 se realiza a través de una alimentación externa. No se puede alimentar a través de la placa de Arduino debido a que no suministra la suficiente corriente al módulo, por lo que si se intentara realizar alguna transmisión o recepción de datos el módulo no respondería correctamente. En el proyecto se ha utilizado el convertor de voltaje de 220V a 3,3V que se puede observar a continuación (Figura 40):

Entrada:

Rango de voltaje: 85-265V.

Se conecta a la red eléctrica (220V)



Salida:

Voltaje: 3,3V (1A)

Alimenta al módulo WiFi

Figura 40. Conversor de voltaje (220V-5V)

CONEXIÓN:

El módulo WiFi se conecta a la placa de Arduino Mega 2560 a través del puerto serie 1. A continuación un diagrama de las conexiones entre las dos placas (Figura 41).

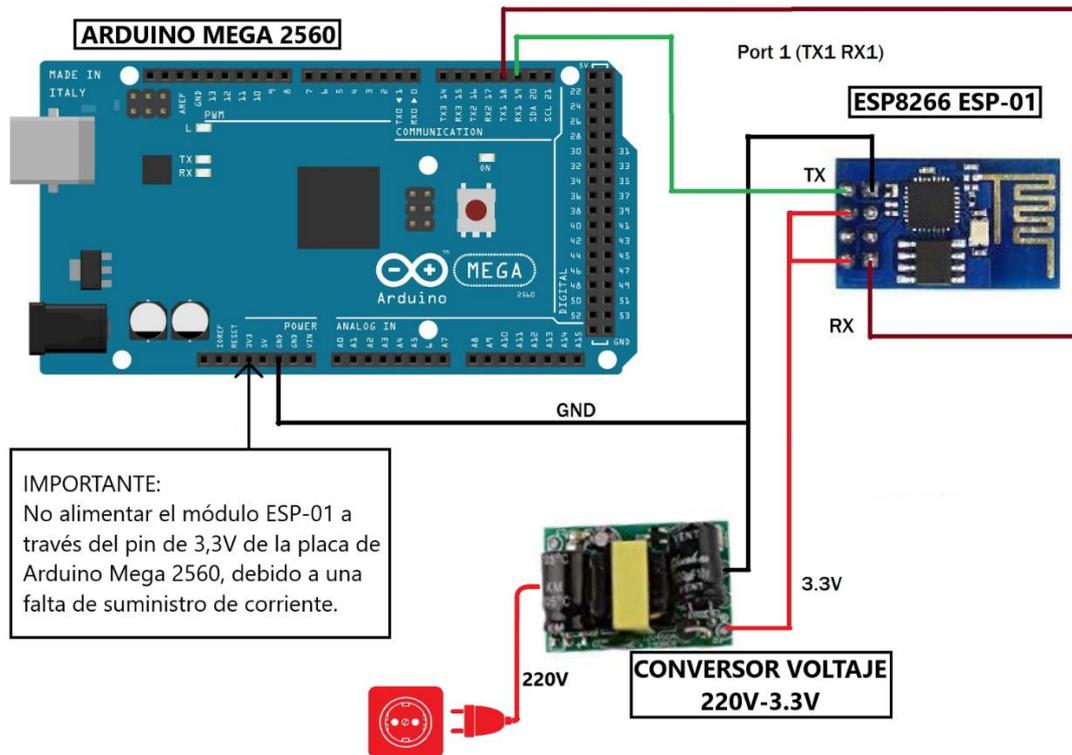


Figura 41. Conexiones a la placa Arduino Mega 2560

PROGRAMACIÓN:

Para la programación del módulo, se realiza a partir del sketch cargado en la Arduino Mega, indicando el tipo de conexión que se realiza (WiFi), a través de que puerto está conectado el módulo (Serie1) y las credenciales necesarias para conectarse a una red WiFi (SSID y contraseña). El resto del intercambio de datos se realiza de forma transparente al usuario. Para que la conexión se realice con éxito es necesaria una red WiFi a la que se pueda conectar a través de su SSID y su contraseña, ya sea a partir de un router o una zona WiFi creada por un terminal móvil. El mensaje que indica por el puerto serie una correcta conexión a la red es el siguiente (Figura 42):

del potenciómetro al que irá conectado. A continuación, el esquema final con todos los dispositivos hardware:

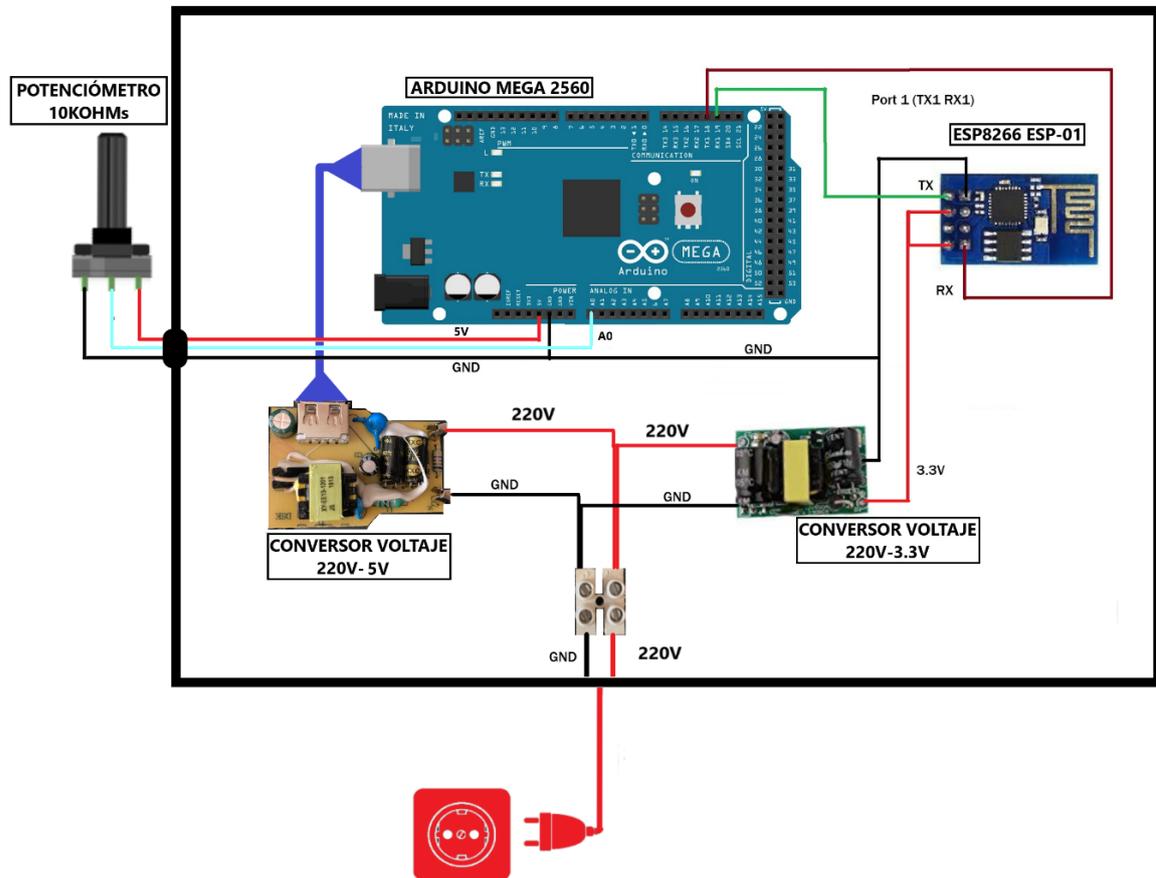


Figura 43. Conexiones finales dentro de la caja

La disposición final de la caja es la siguiente (Figuras 44 y 45):

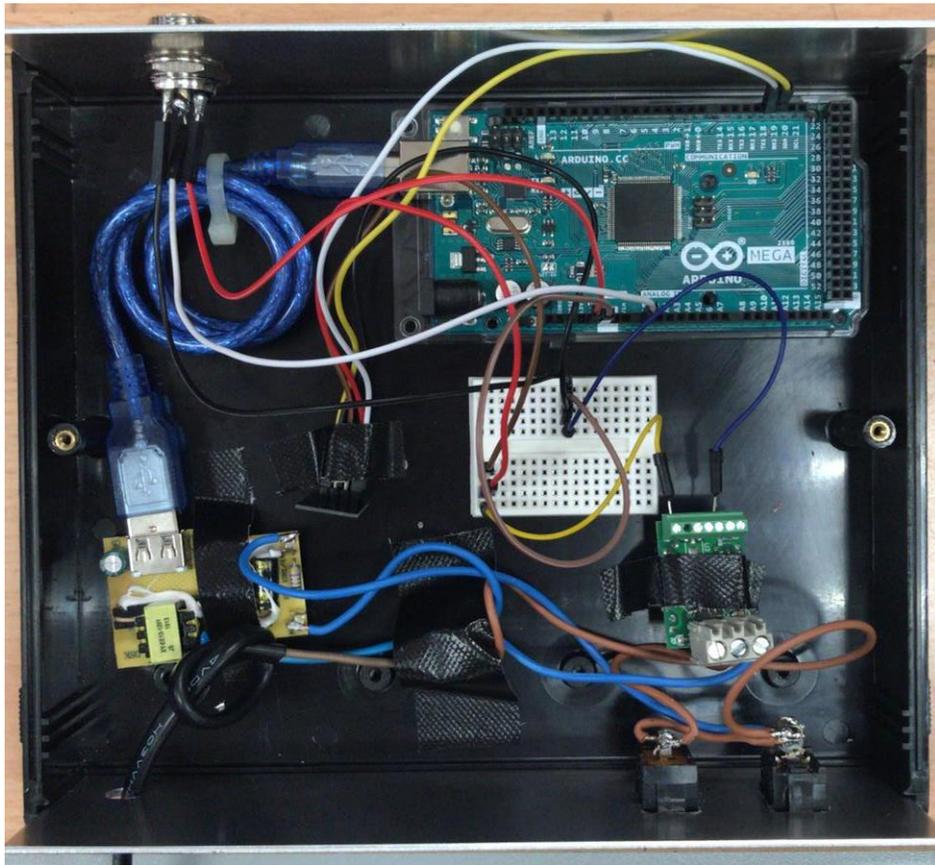


Figura 44. Disposición de los dispositivos Hardware dentro de la caja.



Figura 45. Laterales exteriores de la caja.

4. SOFTWARE

Una parte fundamental de este proyecto es la aplicación móvil que recibe los datos de los sensores insertados en el dispositivo hardware. Es la encargada de presentar la información de manera amigable al usuario gracias a una interfaz gráfica intuitiva y fácil de manejar. La aplicación se crea mediante la plataforma Blynk, disponible tanto para iOS como para Android. A continuación, se explicará en qué consiste la plataforma Blynk, cómo funciona la aplicación y el código involucrado para su correcto funcionamiento. Cabe destacar que toda la programación se ha realizado usando lenguaje C/C++ en el entorno de programación de Arduino.

4.1. Blynk

Este proyecto se ha realizado con miras hacia un futuro en el que todos los dispositivos estén conectados entre sí, es decir, la implantación del Internet de las Cosas (IoT). Por ello se ha escogido el entorno Blynk donde se desarrollará esta aplicación.

Blynk consiste en una plataforma de código abierto disponible para Android y iOS, específicamente diseñada para controlar proyectos de IoT desarrollados en plataformas como Raspberry Pi o Arduino, como es en este caso [18].

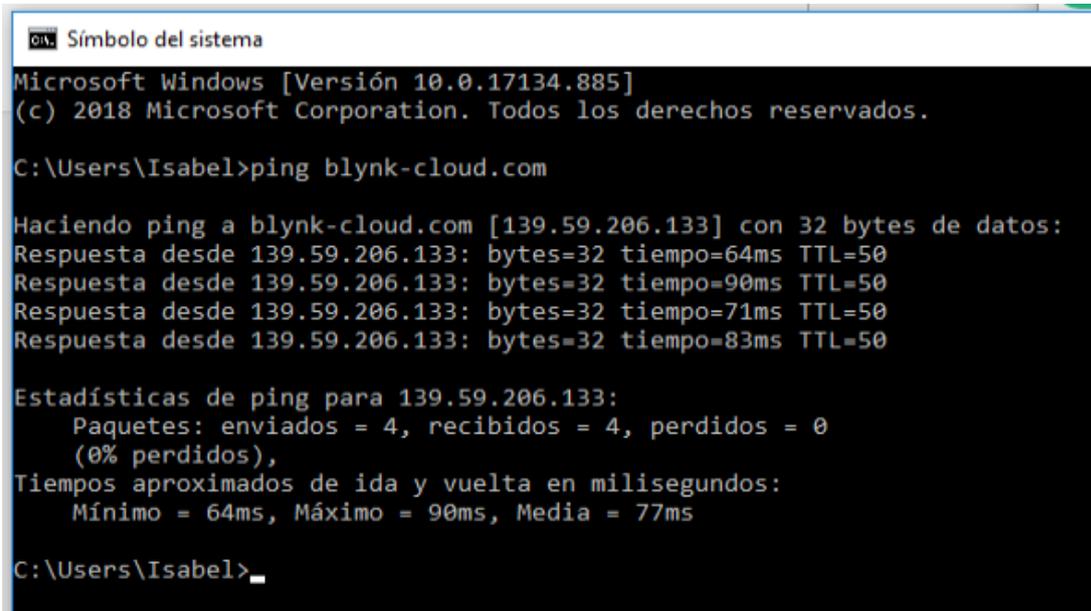
El software desarrollado en este proyecto se apoya en dos principales pilares.

El primero es el servidor, el cual se encarga de todas las comunicaciones entre el dispositivo móvil y el hardware. Existen dos formas de trabajar con los servidores:

Una de ellas es creando un servidor local esto tiene sus ventajas, como por ejemplo el aumento de la seguridad de la información ya que sólo el creador del servidor local conoce los datos de este servidor y además se pueden añadir políticas de seguridad. También tiene una mayor estabilidad ya que no depende del estado de ningún servidor online externo (partido) en caso de saturación de muchos usuarios conectados a la vez y se tiene total control del servidor. Y la tercera ventaja es menor latencia ya que cuanto más cerca se esté del servidor más rápido se transmiten los datos [19]. El problema encontrado con el servidor local es que todos los datos se almacenan en el ordenador en el que el servidor se esté ejecutando. Debido a una mayor facilidad de implementación y

que además no sería necesario el uso de ningún ordenador para la implementación del servidor local se usará la siguiente opción alternativa: El servidor online de Blynk.

En el proyecto la aplicación estará conectada al servidor online llamado www.blynk-cloud.com y cuya dirección IP se averigua haciendo un PING desde el ordenador (Figura 46).



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.17134.885]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Isabel>ping blynk-cloud.com

Haciendo ping a blynk-cloud.com [139.59.206.133] con 32 bytes de datos:
Respuesta desde 139.59.206.133: bytes=32 tiempo=64ms TTL=50
Respuesta desde 139.59.206.133: bytes=32 tiempo=90ms TTL=50
Respuesta desde 139.59.206.133: bytes=32 tiempo=71ms TTL=50
Respuesta desde 139.59.206.133: bytes=32 tiempo=83ms TTL=50

Estadísticas de ping para 139.59.206.133:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
        (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 64ms, Máximo = 90ms, Media = 77ms

C:\Users\Isabel>
```

Figura 46. CMD del ordenador en la que se realiza un PING al servidor Blynk

El dispositivo móvil en el que esté descargado la aplicación debe tener conexión a internet, ya sea a través de WiFi, datos móviles o conexión Ethernet, para poder conectarse al servidor. Hay una gran variedad de opciones a la hora de conectar el hardware con la aplicación, ya sea a través de Bluetooth, BLE, GSM, Ethernet...[19] En este caso se utilizará el módulo WiFi llamado ESP8266 ESP-01 explicado en el apartado 3.5. (Figura 47)

Para realizar la conexión con la aplicación Blynk se utiliza la función:

Blynk.begin(auth, wifi, ssid, pass); a la que le pasamos los datos necesarios para que se puedan conectar, es decir el nombre de la red WIFI (SSID) y su contraseña (PASS), además del código de autenticación de la aplicación Blynk (AUTH) y el método de conexión (WIFI).

El código de autenticación de la aplicación es un código alfanumérico proporcionado por la aplicación de Blynk, Figura 48. Es específico para cada proyecto, y un dato fundamental para poder hacer la conexión entre la aplicación y el hardware.

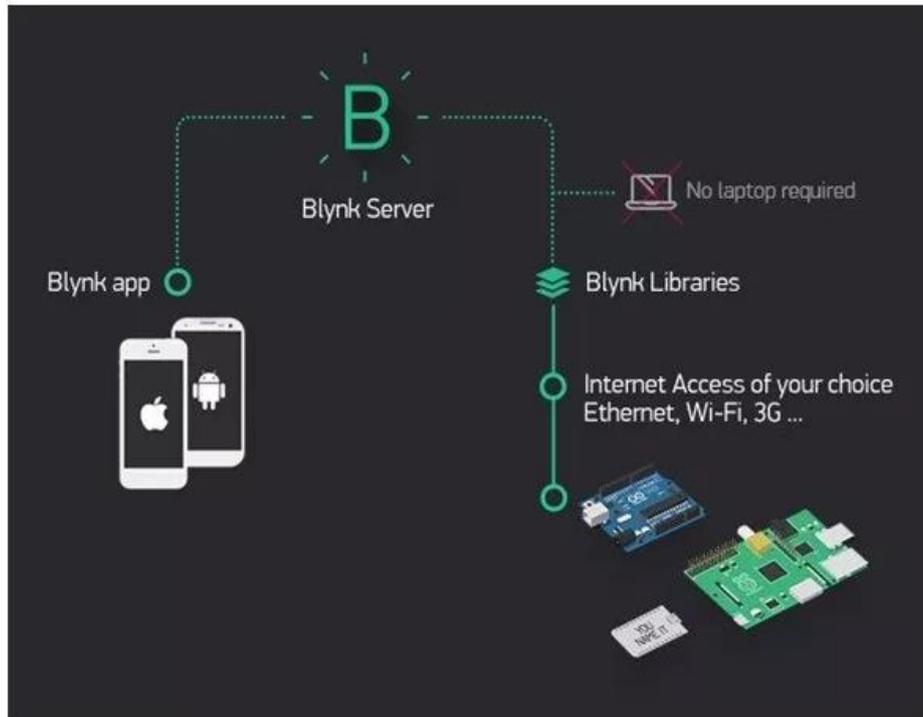


Figura 47. Funcionamiento de la plataforma Blynk.

El segundo pilar del proyecto es la aplicación Blynk, la cual permite crear la interfaz gráfica con los controles necesarios del proyecto. Gracias a esta herramienta se diseña la interfaz gráfica de la aplicación con la que se pueden controlar los dispositivos conectados al Arduino; como encender o apagar leds, subir o bajar el volumen de un zumbador, etc. Otra característica que presenta la herramienta es la posibilidad de mostrar en la interfaz de la aplicación la información recogida del hardware conectado a la placa (sensores, botones...). Para crear esta aplicación se debe elegir la placa hardware junto con el tipo de conexión. En este proyecto se escogería Arduino Mega 2560 junto con la conexión WiFi, tal y como se observa en la siguiente imagen (Figura 46).



Figura 48. Ajustes del dispositivo en la aplicación Blynk.

Para la creación de la interfaz se realiza a través de una serie de Widgets (o módulos del sistema) disponibles en la aplicación Blynk. Cada widget tiene una funcionalidad concreta con una programación asociada a cada uno de ellos. Se puede diferenciar varios tipos de widgets:

- Controladores, se utilizan para enviar comandos que controlen el hardware tales como botones, potenciómetros, Joystick, temporizadores, etc.
- Pantallas que sirven para visualizar datos recogidos de sensores u otras fuentes como pantallas LCD, gráficas, display de valores, LEDs, etc.
- Notificaciones del sistema por medio de twitter, e-mail y pop-ups.
- Interacción con la interfaz usando menús, tablas, introducir texto, números u horas, posición geográfica, etc.
- Sensores del Smartphone tales como el acelerómetro o la posición GPS.

Para controlar el número de widgets que introduces en la aplicación, cada widget tiene asociado un consumo de energía, y cuando se crea la aplicación obtienes una energía determinada, que se va consumiendo en función de los widgets que introduzcas. Una vez consumida toda la energía de la que se dispone no se pueden introducir más widgets. La aplicación tiene la posibilidad de comprar más energía a través de la PlayStore o AppStore (Figura 49) [19].

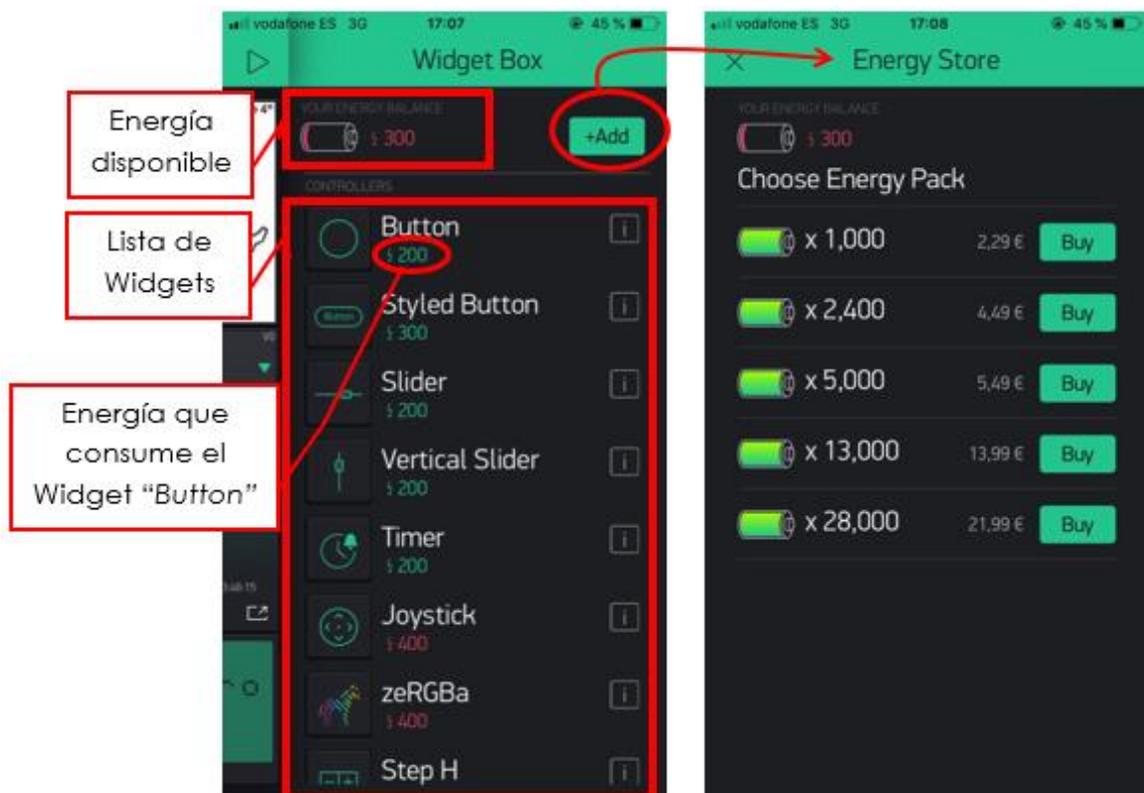


Figura 49. Interfaz de Blynk con la lista de Widgets

A la hora de hacer la recepción o transmisión de los datos entre los widgets, que controlan o muestran los datos, y el hardware se debe contar con un canal de comunicación. En el caso concreto de la placa de Arduino dispone de una serie de pines analógicos y digitales que pueden ser controlados por la interfaz de Blynk, es decir que todos los elementos Hardware conectados a dichos pines pueden dialogar con el sistema software desarrollado. Además de estos pines GPIO físicos, Blynk cuenta con una serie de pines virtuales tal y como se puede observar en la siguiente figura 50. Su principal funcionalidad es usarse como un canal de envío de datos. Estos datos se envían desde el dispositivo móvil, se procesan en el microcontrolador de la placa de Arduino y se vuelven a enviar hacia la aplicación. No existen físicamente. Estos pines llevan asociadas unas funciones de escritura y lectura definidas dentro de la librería Blynk [19].

Para la escritura en los pines virtuales → `Blynk.virtualWrite(vPin,valor)`

Para la lectura en los pines virtuales → `BLYNK_WRITE(vPin)`

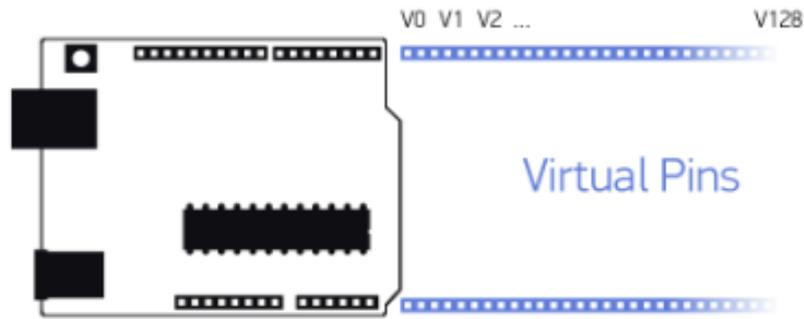


Figura 50. Representación de los pines físicos y pines virtuales

Ventajas e inconvenientes encontrados a la hora del uso de la plataforma para el desarrollo de aplicaciones Blynk

- Facilidad del desarrollo de la interfaz de la aplicación a partir del uso de widgets, ya que las funcionalidades principales de estos ya están implementadas y simplemente es necesario comprender las funciones prediseñadas por los desarrolladores de Blynk y usarlas de forma correcta para un óptimo funcionamiento de la aplicación. Aunque esto también es un arma de doble filo, ya que, que los widgets ya tengan diseñadas las funciones, limita bastante las posibles modificaciones que se quisieran llevar a cabo ya que cada proyecto tiene unas necesidades concretas de funcionamiento de la aplicación y que no han sido consideradas por los desarrolladores. Un claro ejemplo de esto es el widget Superchart que consiste en una gráfica para mostrar datos. A la hora de crear los gráficos en tiempo real, el periodo de escritura de los datos en esta es de un segundo, y no tiene la posibilidad de aumentar o disminuir este tiempo. Para subsanar esto, se puede contactar con los desarrolladores para comentarles las posibles mejoras de los widgets, y ellos podrían poner a disposición de los usuarios una actualización de la aplicación Blynk con los widgets mejorados.
- Una ventaja muy grande es el uso de los pines virtuales para la comunicación de los datos en vez del uso de los pines físicos, dejando así más pines físicos para controlar mayor número de sensores u otros dispositivos hardware.
- La posibilidad de trabajar tanto en servidor local como con el servidor online de Blynk aumenta las posibilidades de diferentes proyectos en función de las necesidades específicas de cada uno.

- Blynk soporta una gran variedad de placas Hardware de código abierto, así como diferentes tipos de conexión entre el hardware y la aplicación.
- Un inconveniente es que no existe como tal un manual de Blynk en el que te expliquen detalladamente como funciona todo. Y al ser algo tan específico y poco conocido cuesta bastante encontrar información sobre el tema fuera de la página web oficial. Aunque tiene un foro de dudas y preguntas en el que se puede pedir ayuda para los proyectos, en el que los participantes de este foro responden con bastante rapidez. Sin embargo, aún con la ayuda recibida por los foros, no implica que el problema pueda resolverse.

Blynk es una herramienta con gran potencial, pero por falta de información detallada o por tener una comunidad de usuarios tan pequeña, se desperdicia mucho de este. Al ser una plataforma tan novedosa hay bastantes mejoras que podrían realizarse, como la posibilidad de modificar los widgets de una manera más autónoma sobre todo para personas con mayor experiencia en desarrollo software. Como plataforma gratuita para desarrolladores de aplicaciones principiantes es una apuesta bastante segura ya que es un entorno amigable e intuitivo.

4.2. Aplicación

Como anteriormente se ha comentado la aplicación constará de una serie de Widgets para controlar el hardware y mostrar los datos recogidos por los sensores conectados a este. La aplicación se puede descargar tanto en sistema operativo Android como en iOS.

Como se puede observar en la siguiente figura (Figura 51), la aplicación tiene una interfaz gráfica amigable al usuario y muy intuitiva. Se pueden diferenciar los componentes de la aplicación según su finalidad y como en conjunto interactúan entre sí para dar cierta coherencia a la aplicación. A continuación, se describirán los diferentes módulos que conforman la aplicación junto con su funcionamiento.



Figura 51. Interfaz final de la aplicación

- **Widget “BUTTON”:**

Consiste en un botón de parada y arranque de los ejercicios de la rehabilitación. El canal de datos por el que se transmite la información del widget es el pin virtual V3. En función de en qué posición se encuentre transmitirá un 0 o un 1. El modo del botón que se elige es “SWITCH” para que una vez presionado el botón se mantenga en ese modo hasta que se vuelva a presionar. El otro modo del botón, “PUSH”, cambia el modo del botón mientras se mantenga presionado, pero una vez se deje de pulsar, vuelve al estado anterior. Cuando se presiona el modo “play” del botón, habilita el temporizador de la función que lee los datos recogidos del potenciómetro y los transforma en una posición angular. Para parar la lectura de datos y salir del ejercicio se debe pulsar el modo “stop” del botón y automáticamente deshabilita el temporizador mencionado anteriormente (Figura 52).

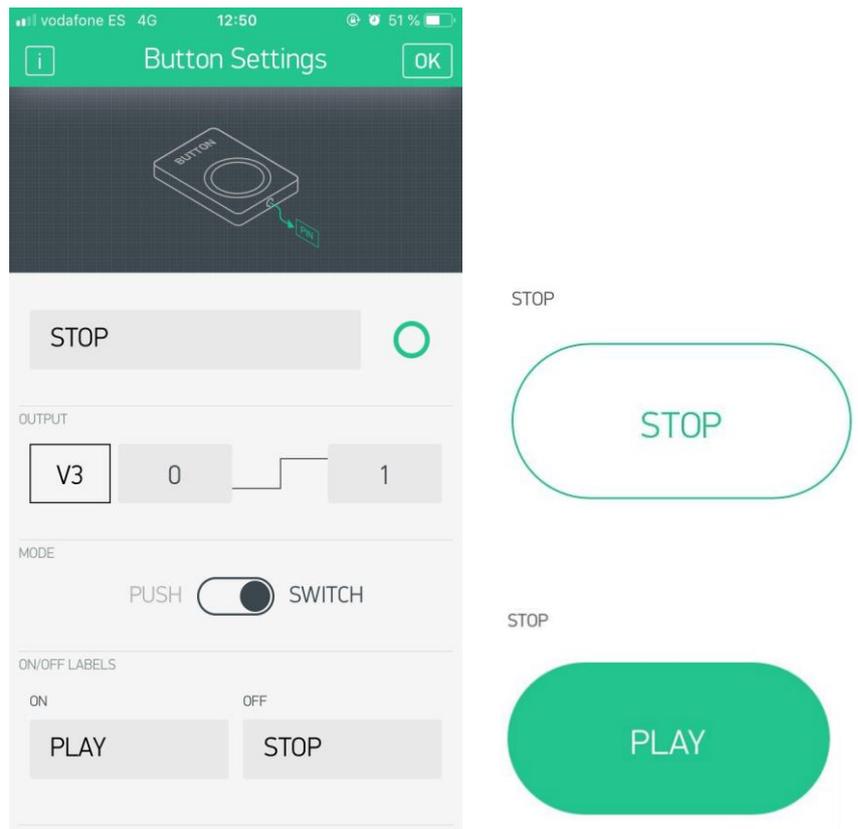


Figura 52. Funcionamiento y ajustes del widget Button

- **Widget "MENU":**

Se trata de un menú de selección de la lista de ejercicios de rehabilitación programados dentro de la aplicación. La información recogida a través de este menú se transmite por el pin virtual V0. Cada ejercicio de rehabilitación es diferente y tiene sus metas asociadas, por lo que en función de cuál se escoja, se realizarán unas comprobaciones u otras, llamando en el código del programa a sus correspondientes funciones, para saber si se está realizando correctamente el ejercicio. La información transmitida según la elección del usuario es un número del 1 al 5 donde cada número se corresponde con un ejercicio concreto. El widget "Image" del que se hablará a continuación depende de la selección que se realice en este menú (Figura 53).

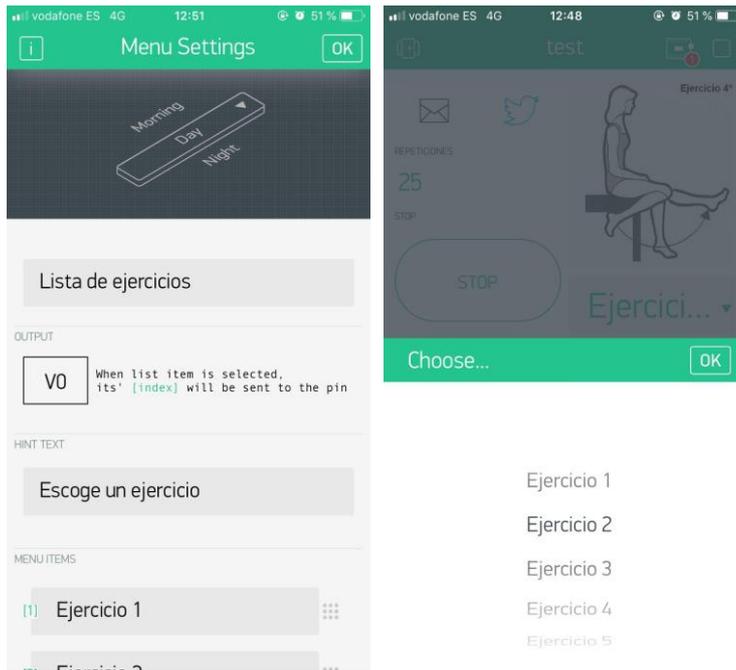


Figura 53. Funcionamiento y ajustes del widget Menu

- **Widget “IMAGE”:**

Este módulo consiste en una galería de imágenes en la que se muestran cómo se debe hacer los ejercicios de rehabilitación. En función del ejercicio escogido en el menú la imagen mostrada se corresponderá con dicho ejercicio. La entrada de datos al módulo se realiza a través del pin virtual V1, y los datos admitidos son números del 1 al 5 y a cada número le corresponde una URL que redirige a la imagen del ejercicio.

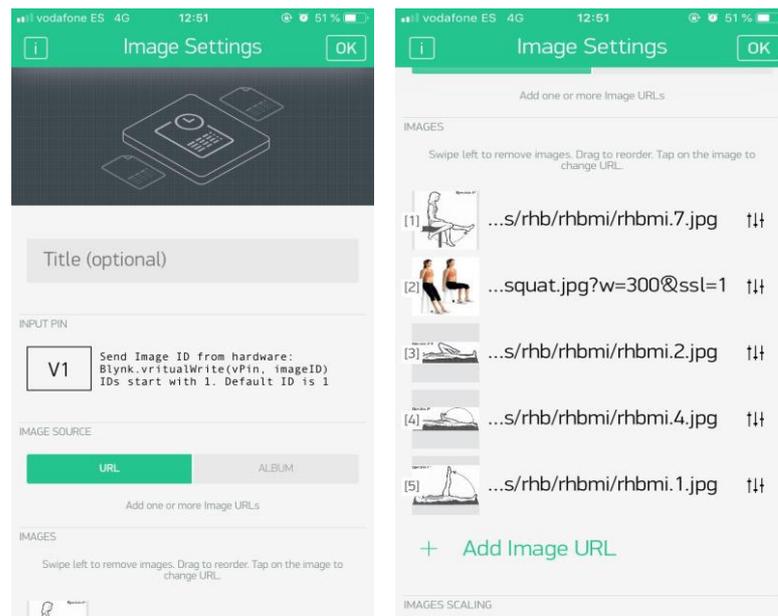


Figura 54. Funcionamiento y ajustes del widget Image

- **Widget “SUPERCHART”:**

Se trata de una gráfica que representa la posición angular de la articulación de la rodilla en tiempo real durante la realización del ejercicio. Se ejecuta cuando el botón comentado anteriormente se encuentra en modo “play”. La gráfica representa en el eje vertical la posición angular de 0 grados a 140. El eje horizontal es el eje de tiempos. El modo de gráfica “live” representan los datos recogidos en tiempo real. También se pueden ver los gráficos realizados en los últimos 15 o 30 minutos o en el último día, en los modos de gráfica “15m”, “30m” o “1d” respectivamente ubicados en la esquina inferior izquierda. La transmisión de las posiciones angulares se realizan por el pin virtual V5. En los ajustes de la gráfica se podrían visualizar más de un vector de datos, aunque en este caso no aplique ya que sólo es necesario observar el movimiento angular de la articulación. Tiene la opción de usar la gráfica en pantalla completa del terminal móvil accediendo al icono de la flecha en la esquina inferior derecha de la gráfica. El problema principal encontrado con este widget ha sido el intentar mostrar los datos en la gráfica a una velocidad mayor de un dato por segundo. Como ha sido comentado anteriormente ha sido uno de los inconvenientes más graves encontrados a lo largo del trabajo y que no ha sido posible subsanarlo.

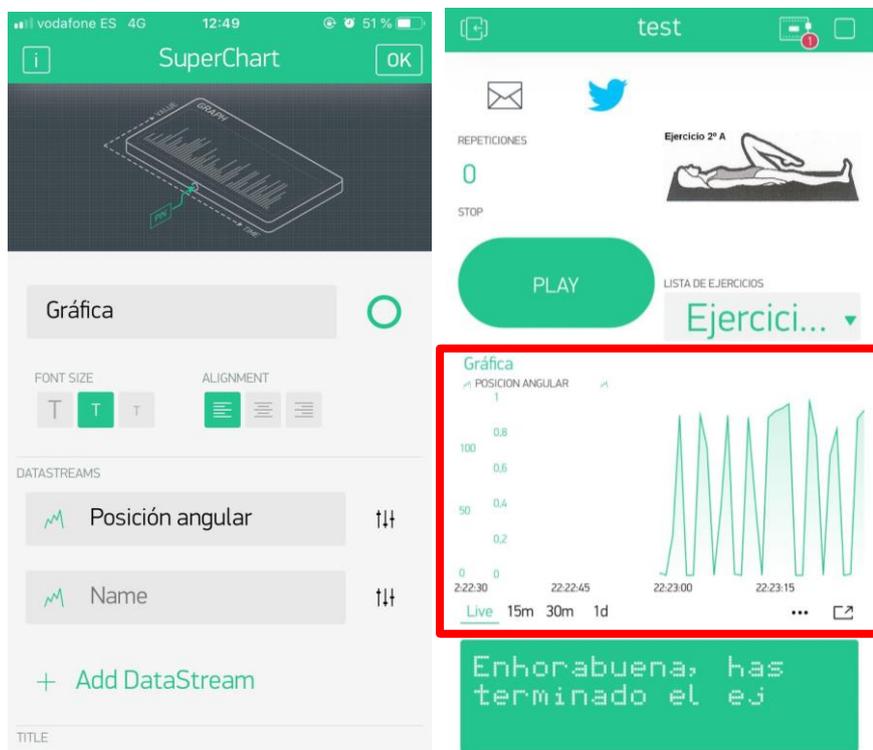


Figura 55. Funcionamiento y ajustes del widget SuperChart

- **Widget LCD:**

Este módulo simula una pantalla LCD cuya función es mostrar mensajes al usuario con el objetivo de dar consejos para realizar los ejercicios de rehabilitación de manera correcta. Además ayuda al usuario a interactuar con la aplicación mostrándole mensajes para que pueda saber cuando empezar o terminar el ejercicio, o qué ejercicio es el que ha seleccionado. El canal de datos por el que se introducen las cadenas de caracteres es el pin virtual V2. Algunos de los mensajes que podemos encontrar en esta pantalla son:

- “Has elegido el ejercicio X”.
- “Empieza el ejercicio”.
- “Has terminado el ejercicio”.
- “Estira más la rodilla”.
- “Flexiona más la rodilla”.

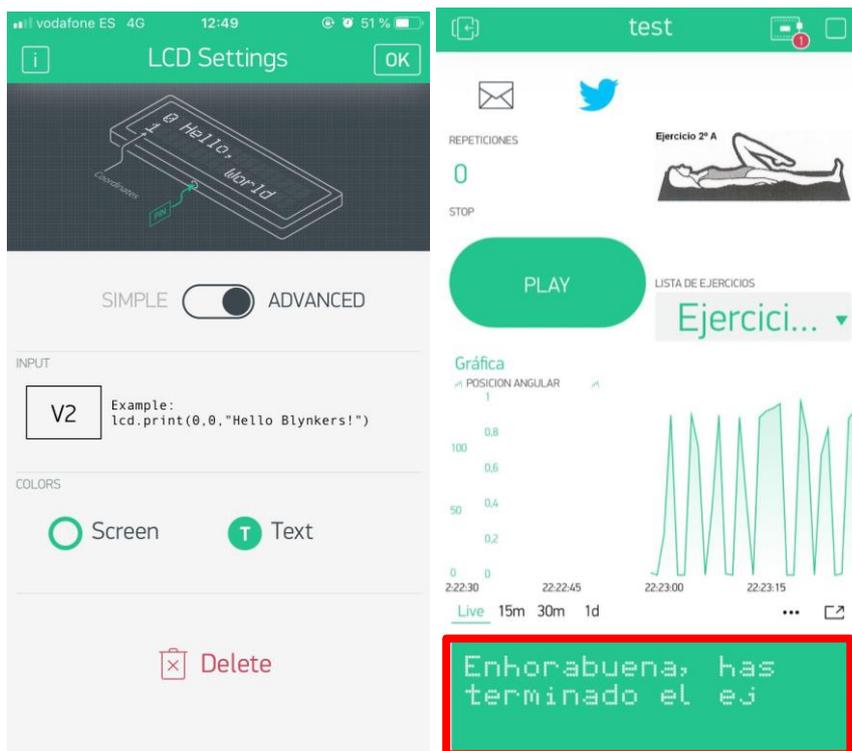


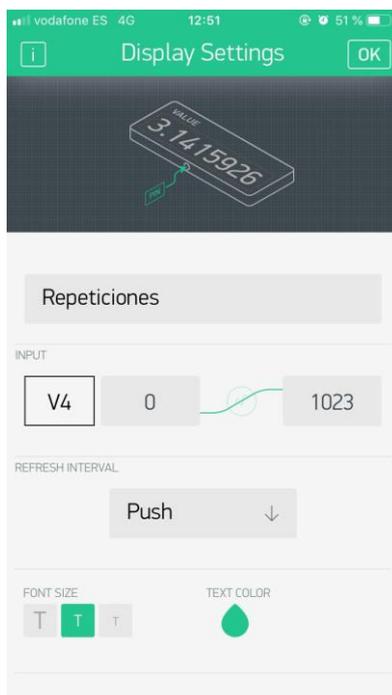
Figura 56. Funcionamiento y ajustes del widget SuperChart

- **Widget “DISPLAY”:**

Consiste en, como bien indica su nombre, un display que representa el numero de repeticiones que faltan por realizar del ejercicio de rehabilitación. Cada vez que se cumplen los objetivos del ejercicio disminuye una repetición. Cuando este número llegue a cero, que significa que se ha terminado el ejercicio, ocurrirán una serie de eventos en la aplicación:

- 1- El botón se modificará a modo “stop” y la aplicación dejará de leer la posición angular de la rodilla.
- 2- Aparecerá un mensaje en la pantalla LCD indicando que se ha terminado el ejercicio.
- 3- Se recopilarán todas las posiciones recogidas de la rodilla a lo largo del ejercicio y se enviará un correo electrónico al médico rehabilitador indicando el nombre del paciente que ha realizado el ejercicio, el nombre del ejercicio que ha realizado y las posiciones angulares recogidas.

Una vez terminada esa serie de eventos, el usuario podrá escoger otro ejercicio de rehabilitación. En caso de que se pare el ejercicio sin haber completado el número de repeticiones, el ejercicio se considerará incompleto y no se mandará el correo electrónico. El pin virtual por el que se indican el número de repeticiones es el V4. Se puede mandar un numero entre 0 y 1023, a pesar de que estos números se pueden modificar en los ajustes del widget.



REPETICIONES

25

Figura 57. Funcionamiento y ajustes del widget Display

- **Widget “EMAIL”:**

Su función es poder modificar de una manera sencilla el destinatario del email al que queremos mandar los datos del ejercicio completado por el usuario y el tipo de información que se quiere transmitir en el correo. En este caso se envía texto plano.

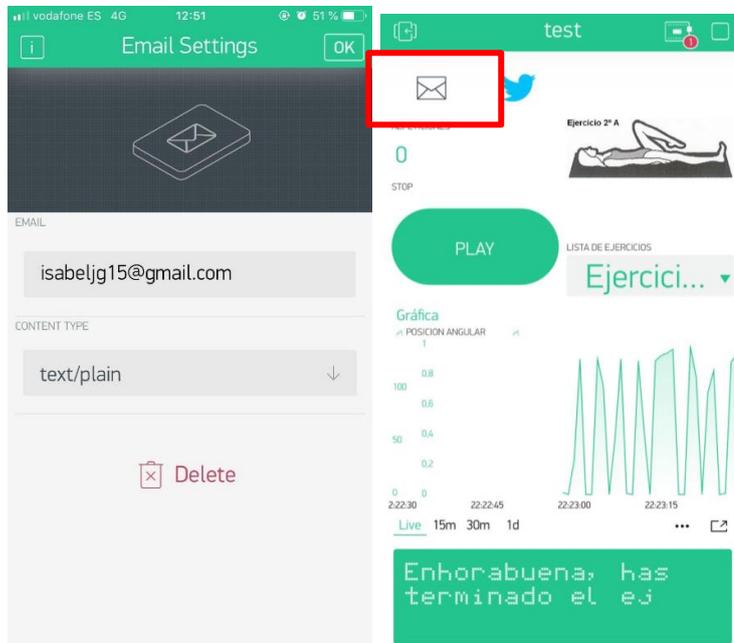


Figura 58. Funcionamiento y ajustes del widget Email

- **Widget “TWITTER”:**

Cambiar la configuración de la cuenta en la que se mandaría un *tweet* cuyo mensaje puede variar en función de lo que se quiera publicar. Podría funcionar como método de promocionarse en las redes sociales.

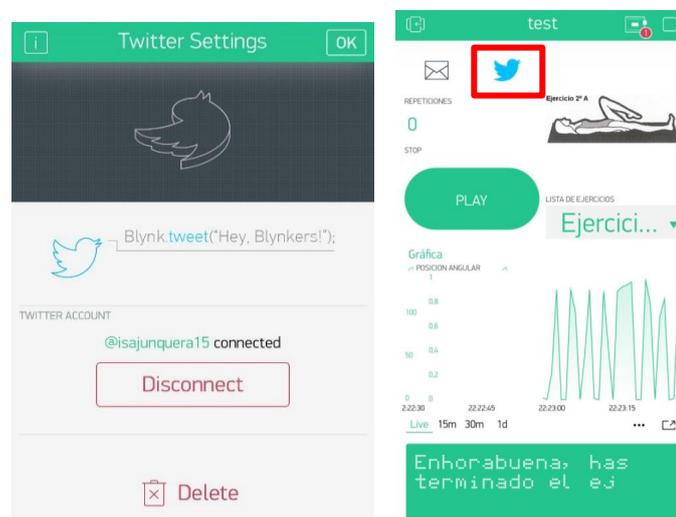


Figura 59. Funcionamiento y ajustes del widget Twitter

4.3. Código

```
#define BLYNK_PRINT Serial
#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>

// Auth Token o código de autenticación de Blynk App.
char auth[] = "7c454bcfd36340d7851fceb1d2161d6";
//credenciales de la red WIFI
char ssid[] = "iphone";
char pass[] = "123456789";
//Nombre del módulo ESP
#define EspSerial Serial1
//tasa de baudios a los que trabajará el módulo ESP
#define ESP8266_BAUD 9600
//inicialización del ESP
ESP8266 wifi(&EspSerial);

WidgetLCD lcd(V2); //asignación del pin virtual 2
SimpleTimer timer2; //temporizador
byte timerID; //ID del temporizador
byte repeticiones; // numero de repeticiones
byte ej; //numero de ejercicio escogido
byte contador=0; //contador de lectura de datos
int valorMax=0; //valor máximo recogido
int valorMin=140; //valor mínimo leído
int position; //posición del potenciómetro en tanto por ciento
String cadena = ""; //cadena que se enviará por email con los datos del
ejercicio

/*****FUNCIÓN QUE ESCRIBE EN LA
PANTALLA LED DE LA APP*****/
void imprime1(String cadena){
    lcd.clear();
    lcd.print(0, 0, cadena);
}
void imprime2(String cadena){
    lcd.print(0, 1, cadena);
}
/*****FUNCIÓN DE LECTURA DEL EJERCICIO
ESCOGIDO EN LA APP*****/
BLYNK_WRITE(V0)
{ ej = param.asInt(); //Lee el ejercicio escogido
  switch (param.asInt())
  {
    case 1://se elige el ejercicio 1
      Blynk.virtualWrite(V1, 1); //se cambia la imagen 1 en el widget
de IMAGE GALERY
      repeticiones=20;
      cadena = "";
      imprime1("Has elegido");
      imprime2("ejercicio 1");
      delay(2000);
      break;
    case 2://se elige el ejercicio 2
      Blynk.virtualWrite(V1, 2); //se cambia la imagen 2
      repeticiones=20;
      cadena = "";
      imprime1("Has elegido");
```

```

    imprime2("ejercicio 2");;
    delay(2000);
    break;
case 3://se elige el ejercicio 3
  Blynk.virtualWrite(V1, 3); //se cambia la imagen 3
  repeticiones=20;
  cadena = "";
  imprime1("Has elegido");
  imprime2("ejercicio 3");
  delay(2000);
  break;
case 4://se elige el ejercicio 4
  Blynk.virtualWrite(V1, 4); //se cambia la imagen 4
  repeticiones=20;
  cadena = "";
  imprime1("Has elegido");
  imprime2("ejercicio 4");
  delay(2000);
  break;
case 5://se elige el ejercicio 5
  Blynk.virtualWrite(V1, 5); //se cambia la imagen 5
  repeticiones=20;
  cadena = "";
  imprime1("Has elegido");
  imprime2("ejercicio 5");
  delay(2000);
  break;
default:
  imprime1("No se ha");
  imprime2("escogido nada");
}

}

/*****FUNCIÓN QUE LEE LOS DATOS DEL
POTENCIÓMETRO*****/
void leedatos()
{
  const int analogPin = A0;//variable que almacena el valor del
potenciometro
  int value = analogRead(A0);      // realizar la lectura analógica
raw
  position = round(map(value, 0, 1023, 0, 370)); // convertir a
porcentaje el potencimetro
  Blynk.virtualWrite(V5, position);//escribe en la aplicacion en el
widget SUPERCHART
  //Serial.print(position);
  valorMax = max(position, valorMax);//Actualiza el valor máximo
recogido
  valorMin = min(position, valorMin);//Actualiza el valor mínimo
recogido
  cadena=cadena+", "+position;
  switch (ej)//llama a el ejercicio correspondiente
  {
    case 1: ejercicio1();//comprobaciones del ej 1

      break;
    case 2: ejercicio2();//comprobaciones del ej 2

      break;
    case 3: ejercicio3();//comprobaciones del ej 3

```

```

        break;
    case 4: ejercicio4(); //comprobaciones del ej 4

        break;
    case 5: ejercicio5(); //comprobaciones del ej 5

        break;
    default:
        imprime1("Escoge un");
        imprime2("ejercicio ");
        delay(1000);
        break;
    }
}

/*****FUNCION QUE HACE LAS COMPROBACIONES DEL
EJERCICIO 1*****/
void ejercicio1() //Sentada flexiona y estira la rodilla al máximo
{
    contador++; //Aumenta el contador para la escritura en LCD

    if (valorMax < 130) //comprobacion para ver si NO flexiona la rodilla
    {
        if (contador >= 40) //si el contador llega a 40 que escriba en LCD
        {
            imprime1("Flexiona mas la");
            imprime2("rodilla");
            contador = 0; //Se restauran los valores
            valorMin = 140; //Se restauran los valores
            valorMax = 0; //Se restauran los valores
        }
    }
    else if (valorMin > 10) //comprobacion para ver si NO estira la
rodilla
    {
        if (contador >= 40) //si el contador llega a 40 que escriba en LCD
        {
            imprime1("Estira mas la");
            imprime2("rodilla");
            contador = 0;
            valorMin = 140;
            valorMax = 0;
        }
    }
}
else //si consigue cumplir los ángulos correctos
{
    if(valorMin <10 && valorMax > 130)
    {
        if(contador>=4)
        {
            imprime1("Sigue asi");
            contador=0;
        }

        valorMin = 140;
        valorMax = 0;
    }
}
}

```

```

    repeticiones--; //se disminuye una repeticion de la serie
    Blynk.virtualWrite(V4, repeticiones); //se imprime en la
aplicación en el widjet de DISPLAY
    if (repeticiones <= 0) //si termina con las repeticiones
    {
        imprime1("Enhorabuena, has");
        imprime2("terminado el ej"); //termina el ejercicio (imprime
lcd)
        Blynk.virtualWrite(V3, 1); //pone el boton a STOP y para de
leer datos
        timer2.disable(timerID); //deshabilita el temporizador que
llama a Leedatos
        Blynk.email("isabeljg15@gmail.com", "Subject",
"Ejercicio1:"+cadena);
    }
}
}
}
}

```

```

/*****FUNCION QUE HACE LAS COMPROBACIONES DEL
EJERCICIO 2*****/

```

```

void ejercicio2() //Sentadilla contra la pared
{
    contador++;

    if (valorMax < 90) //comprobabcion de flexión
    {
        if (contador >= 40)
        {
            imprime1("Flexiona mas la");
            imprime2("rodilla");
            contador = 0;
            valorMin = 140;
            valorMax = 0;
        }
    }
    else if (valorMin > 10) //comprobabción de estiramiento
    {
        if (contador >= 40)
        {
            imprime1("Estira mas la");
            imprime2("rodilla");
            contador = 0;
            valorMin = 140;
            valorMax = 0;
        }
    }
    else
    {
        if (valorMin < 10 && valorMax > 90) //comprobacion de movimiento
correcto
        {
            imprime1("Sigue asi");
            valorMin = 140;
            valorMax = 0;
            repeticiones--; //disminuye una repeticion
            Blynk.virtualWrite(V4, repeticiones); //escribe en el widget de
DISPLAY

```

```

    if (repeticiones <= 0)//termina el ejercicio
    {
        imprime1("Enhorabuena, has");
        imprime2("terminado el ej");
        Blynk.virtualWrite(V3, 1);//para de leer y pone la app en STOP
        Blynk.email("isabeljg15@gmail.com", "Subject",
"Ejercicio2:"+cadena);

    }
}
}
}
}

```

```

/*****FUNCION QUE HACE LAS COMPROBACIONES DEL
EJERCICIO 3*****/

```

```

void ejercicio3()//tumbada llevar la rodilla al pecho flexionando
{
    contador++;

    if (valorMax < 130)//comparativa de flexión
    {
        if (contador >= 40)
        {
            imprime1("Flexiona mas la");
            imprime2("rodilla");
            contador = 0;
            valorMin = 140;
            valorMax = 0;
        }
    }
    else if (valorMin > 10)//comparativa de estiramiento
    {
        if (contador >= 40)
        {
            imprime1("Estira mas la");
            imprime2("rodilla");
            contador = 0;
            valorMin = 140;
            valorMax = 0;
        }
    }
    else if (position > 130)//comparativa de la rodilla al pecho
    {
        if (contador >= 5)
        {
            imprime1("lleva la rodilla");
            imprime2("al pecho");
            contador = 0;
        }
    }
    else//se realiza correctamente el ejercicio
    {
        if(valorMin <10 && valorMax > 130)
        {
            lcd.clear();
            lcd.print(0, 0, "Sigue asi");
            valorMin = 140;

```

```

    valorMax = 0;
    repeticiones--; //disminuye una repetición
    Blynk.virtualWrite(V4, repeticiones); //escribe en el widget de
DISPLAY las repeticiones que quedan
    if (repeticiones <= 0) //termina el ejercicio
    {
        imprime1("Enhorabuena, has");
        imprime2("terminado el ej");
        Blynk.virtualWrite(V3, 1);
        Blynk.email("isabeljg15@gmail.com", "Subject",
"Ejercicio3:"+cadena);
        timer2.disable(timerID); //deshabilita el temporizador que
llama a Leedatos
    }
}
}
}
}
}

```

```

/*****FUNCION QUE HACE LAS COMPROBACIONES DEL
EJERCICIO 4*****/

```

```

void ejercicio4() //boca abajo flexionar y estirar
{
    contador++;

    if (valorMax < 130) //comprobación de flexión
    {
        if (contador >= 40)
        {
            imprime1("Flexiona mas la");
            imprime2("rodilla");
            contador = 0;
            valorMin = 140;
            valorMax = 0;
        }
    }
    else if (valorMin > 10) //comparación de estiramiento
    {
        if (contador >= 40)
        {
            imprime1("Estira mas la");
            imprime2("rodilla");
            contador = 0;
            valorMin = 140;
            valorMax = 0;
        }
    }
    else //realiza correctamente el ejercicio
    {
        if (valorMin < 10 && valorMax > 130)
        {
            imprime1("Sigue asi");
            valorMin = 140;
            valorMax = 0;
            repeticiones--;
            Blynk.virtualWrite(V4, repeticiones);
        }
    }
}

```

```

    if (repeticiones <= 0)
    {
        imprime1("Enhorabuena, has");
        imprime2("terminado el ej");
        Blynk.virtualWrite(V3, 1);
        Blynk.email("isabeljg15@gmail.com", "Subject",
"Ejercicio4:"+cadena);
        timer2.disable(timerID);//deshabilita el temporizador que
llama a Leedatos

    }
}
}

/*****FUNCION QUE HACE LAS COMPROBACIONES DEL
EJERCICIO 5*****/
void ejercicio5()//Levantar la pierna estirada
{
    contador++;

    if (valorMax > 20)//comprobar que no flexiona la rodilla
    {
        if (contador >= 40)
        {
            imprime1("Estira mas la");
            imprime2("rodilla");
            contador = 0;
            valorMin = 140;
            valorMax = 0;
        }
    }
    else if (valorMin > 10)//comparar que estira lo suficiente la
rodilla
    {
        if (contador >= 40)
        {
            imprime1("Estira mas la");
            imprime2("rodilla");
            contador = 0;
            valorMin = 140;
            valorMax = 0;
        }
    }
    else
    {
        if(valorMin <10 && valorMax <30)//realiza correctamente el
ejercicio
        {
            imprime1("Sigue asi");
            valorMin = 140;
            valorMax = 0;
            repeticiones--;//disminuye una repeticion
            Blynk.virtualWrite(V4, repeticiones);//imprime en DISPLAY las
repeticiones que quedan
            if (repeticiones <= 0)//termina el ejercicio
            {
                imprime1("Enhorabuena, has");
            }
        }
    }
}

```

```

        imprime2("terminado el ej");
        Blynk.virtualWrite(V3, 1);
        Blynk.email("isabeljg15@gmail.com", "Subject",
"Ejercicio5:"+cadena);
        timer2.disable(timerID); //deshabilita el temporizador que
llama a Leedatos

    }
}
}

BLYNK_WRITE(V3)
{ if (param.asInt() == 1) //El boton en la aplicación está a 1
{
    timer2.disable(timerID); //deshabilita el temporizador que llama a
Leedatos
    imprime1("Has salido del"); //Imprime en LCD
    imprime2("ejercicio");
    delay(3000);
}
else
{
    timer2.enable(timerID); //habilita el temporizador que llama a
leedatos
    imprime1("Empieza el"); //Imprime en LCD
    imprime2("ejercicio");
    delay(3000);
}
}
/*****FUNCIÓN DE
INICIALIZACIÓN*****/
void setup()
{

    //Serial.begin(9600); //Para que aparezca por el puerto serie del
ordenador los mensajes

    // Set ESP8266 baud rate
    EspSerial.begin(ESP8266_BAUD); //inicializa el módulo ESP8266
delay(10);

    Blynk.begin(auth, wifi, ssid, pass); //conexión con la aplicación

    //Serial.println("Waiting for connections...");
    timerID= timer2.setInterval(500L, leedatos); //inicializacion del
timer de leedatos cada 250ms
    Blynk.syncAll(); //sincroniza la aplicacion blynk todos los pines
virtuales

}
/*****FUNCIÓN DEL BUCLE
PRINCIPAL*****/
void loop()
{
    Blynk.run(); //ejecuta la aplicacion de blynk
    timer2.run(); //ejecuta el temporizador
}

```


5. PROTOCOLO DE PRUEBAS Y RESULTADOS

Una vez conseguido el funcionamiento óptimo tanto de la aplicación como de los componentes hardware involucrados en el proyecto se realiza el banco de pruebas y resultados. Estas pruebas se realizan tanto a sujetos de control como a pacientes con afecciones fisiológicas en la articulación de la rodilla. Hay que tener en cuenta que estas pruebas sólo pueden realizarse a sujetos que tengan un diámetro de cuádriceps y gemelo compatible con el tamaño de la ortesis impresa. Si la ortesis no se ajusta correctamente al usuario los resultados obtenidos podrían no resultar fiables. Se realiza un ejercicio de rehabilitación por cada usuario. La elección del ejercicio de rehabilitación queda a preferencia del usuario. A continuación, se mostrarán los datos básicos de cada sujeto cuidando la confidencialidad de la identidad de cada persona. El objetivo es ver si son capaces de interactuar con la aplicación y realizar el ejercicio de rehabilitación de una manera autónoma. Se recogerán los resultados enviados por email una vez concluido el ejercicio y se mostrarán de manera gráfica. También se realizará un cuestionario evaluando la experiencia personal de cada usuario como una retroalimentación para futuras mejoras del dispositivo.

Lista de ejercicios disponibles en la app:

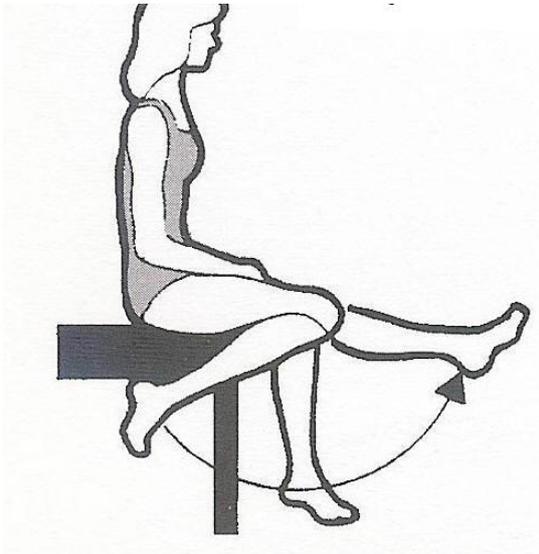


Figura 60. Ejercicio 1



Figura 61. Ejercicio 2

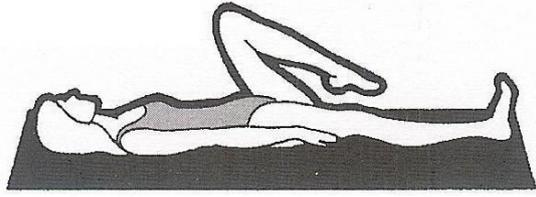


Figura 62. Ejercicio 3

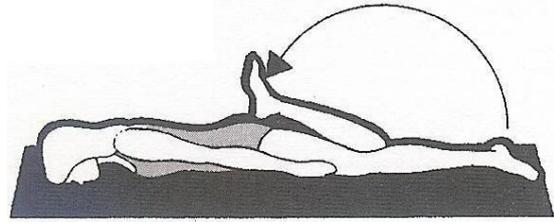


Figura 63. Ejercicio 4

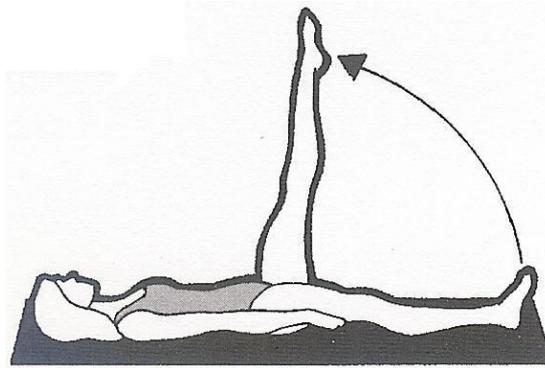
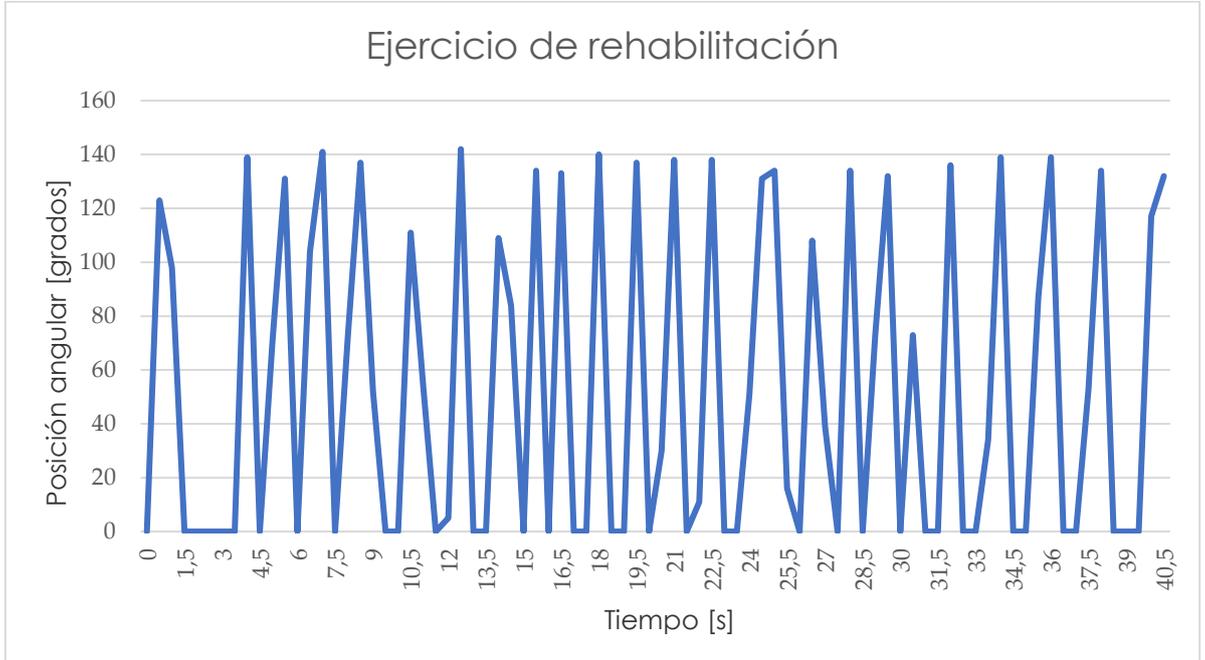


Figura 64. Ejercicio 5

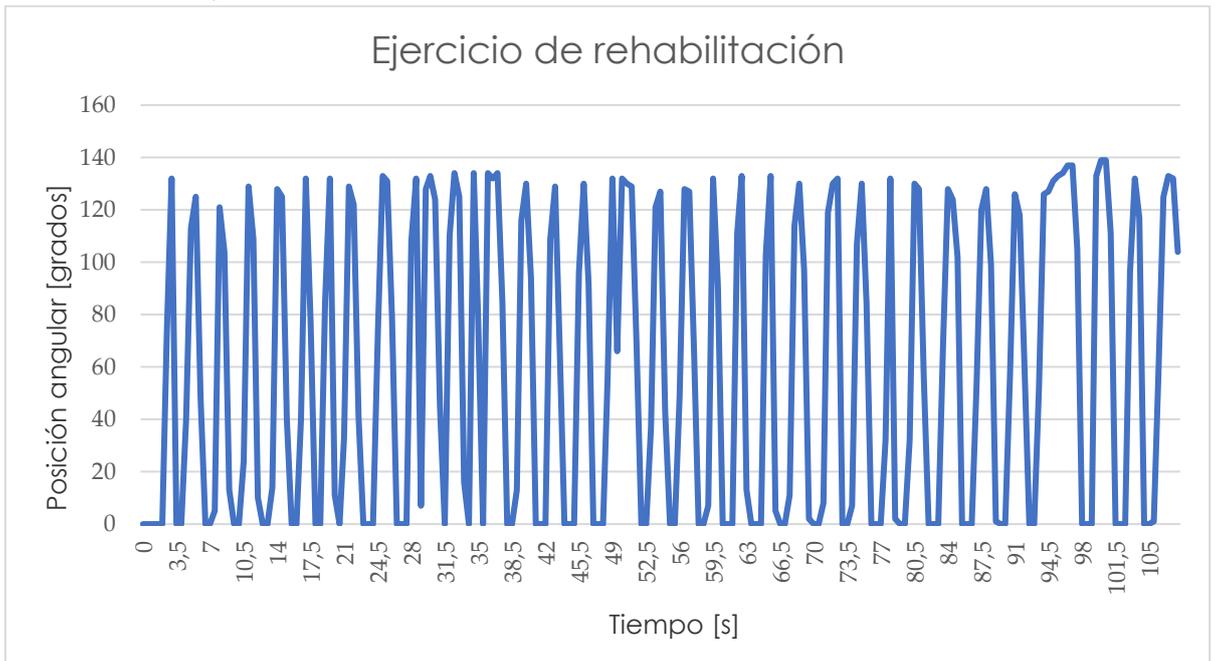
SUJETOS DE CONTROL

El grupo de sujetos de control está formado por un total de 6 personas de un rango de edades entre los 16 y los 52. No han tenido lesiones ni afecciones en la articulación de la rodilla previamente.

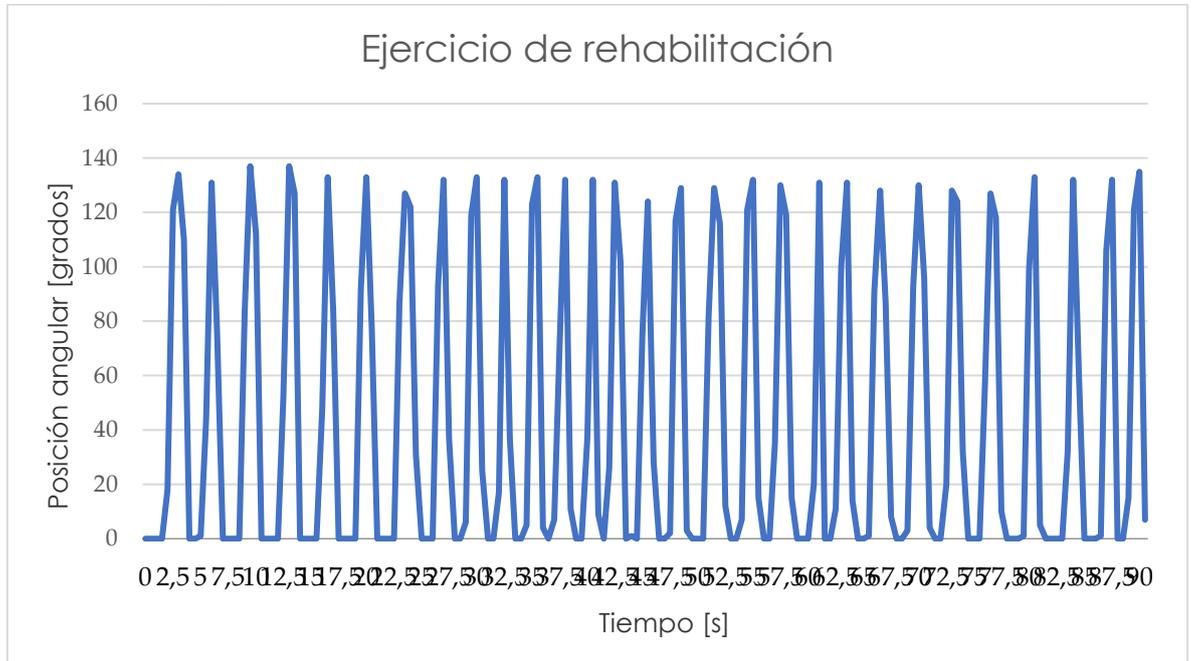
- SUJETO 1:
 Edad: 16
 Sexo: Mujer
 Ejercicio escogido: 4
 Gráfica del ejercicio realizado:



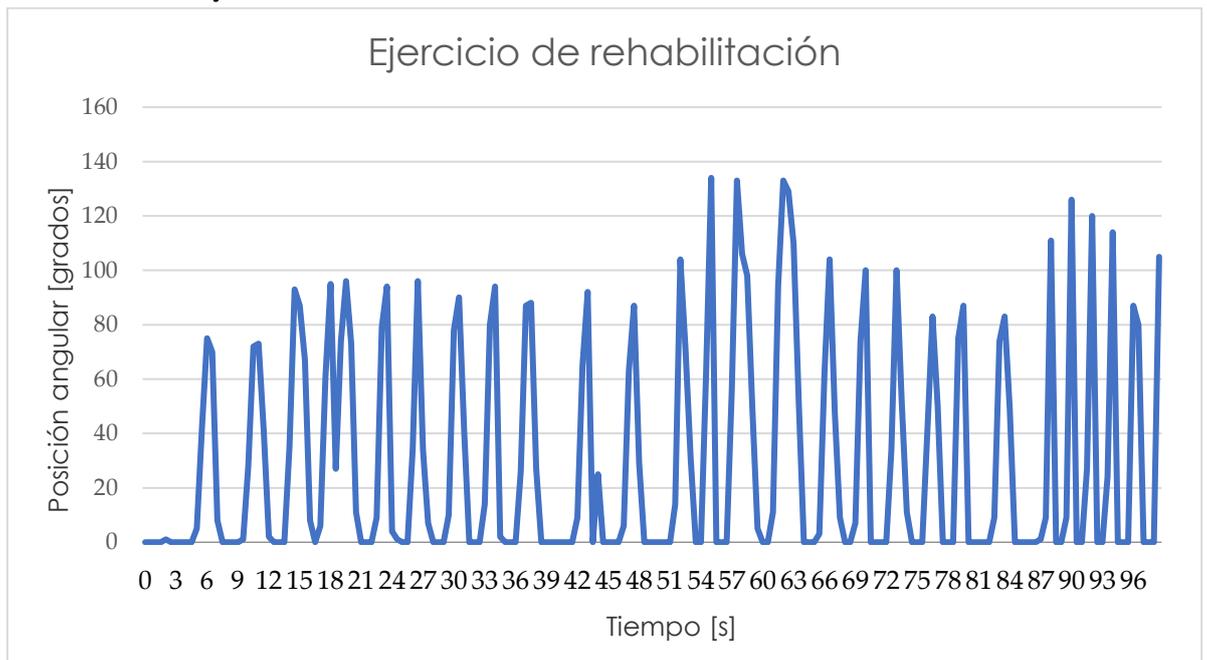
- SUJETO 2:
 Edad: 22
 Sexo: Mujer
 Ejercicio escogido: 3
 Gráfica del ejercicio realizado:



- SUJETO 3:
 Edad: 25
 Sexo: Mujer
 Ejercicio escogido: 3
 Gráfica del ejercicio realizado:



- SUJETO 4:
 Edad: 38
 Sexo: Hombre
 Ejercicio escogido: 2
 Gráfica del ejercicio realizado:



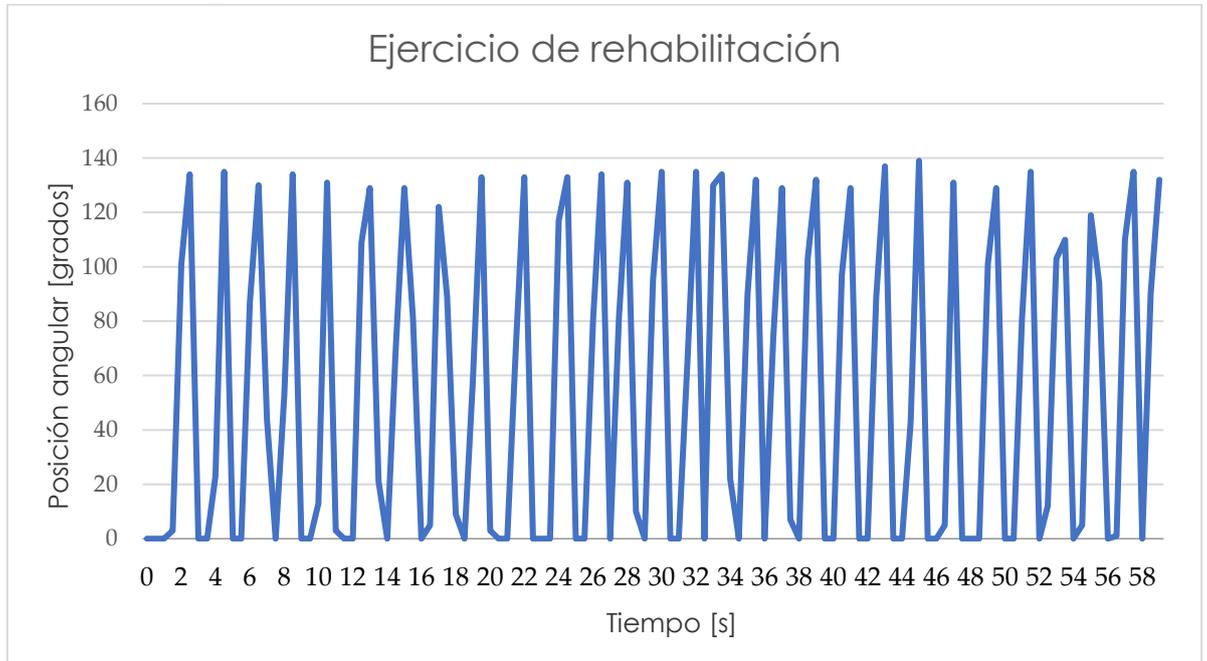
- SUJETO 5:

Edad: 51

Sexo: Mujer

Ejercicio escogido: 1

Gráfica del ejercicio realizado:



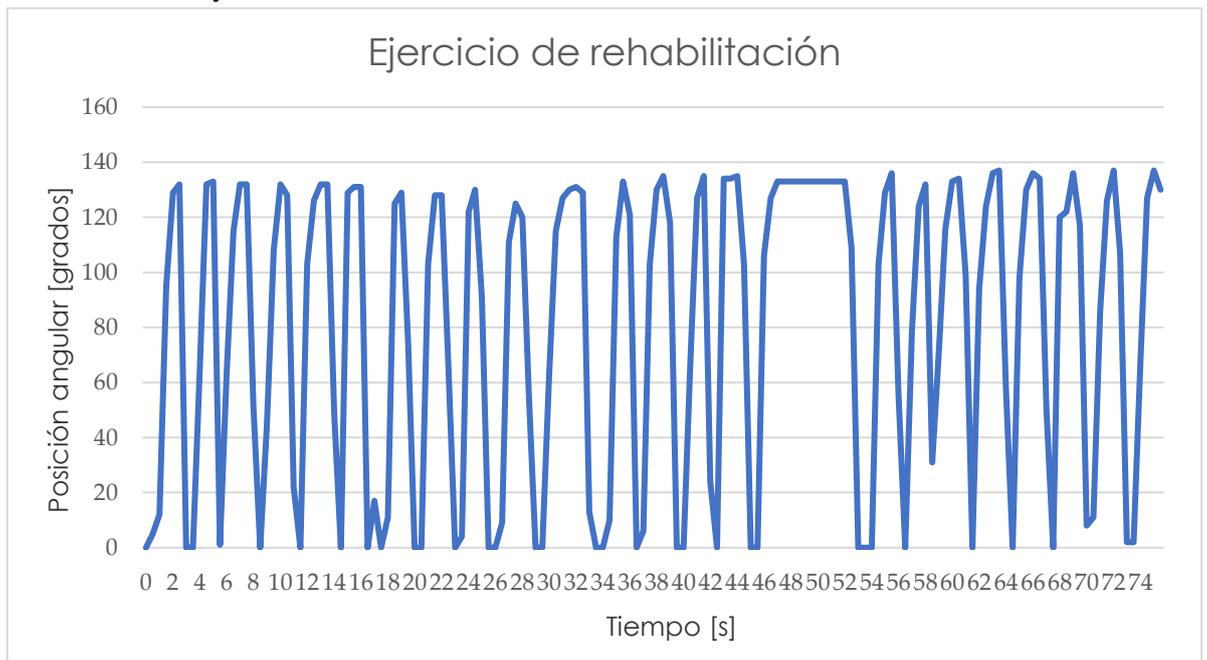
- SUJETO 6:

Edad: 52

Sexo: Hombre

Ejercicio escogido: 3

Gráfica del ejercicio realizado:



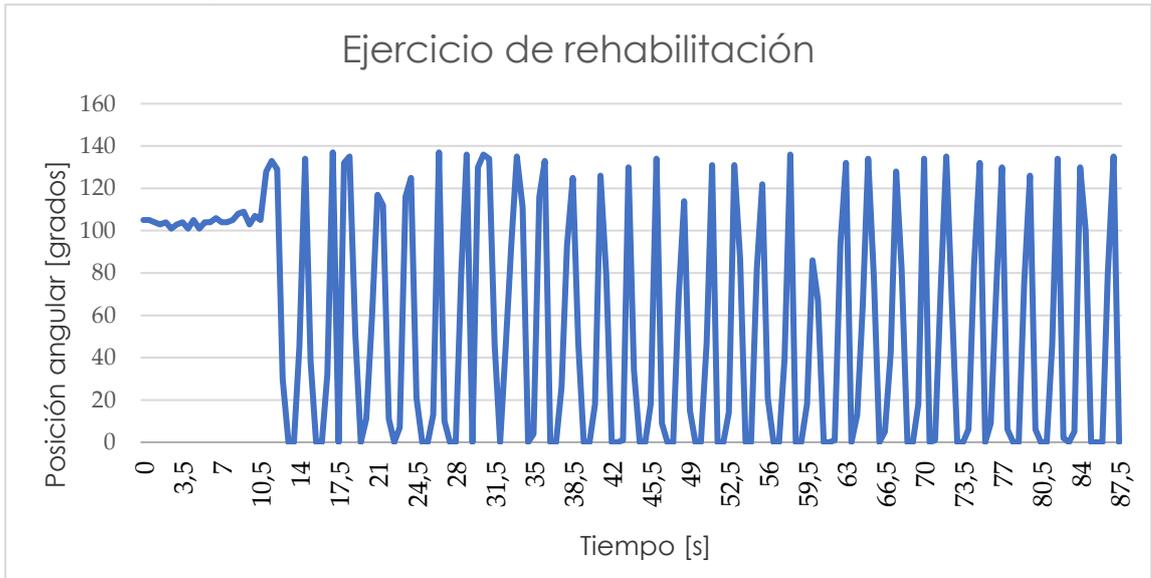
- SUJETO 2:
 Edad: 56
 Sexo: Hombre
 Afección: Calcificación en la rodilla
 Tiempo recibiendo rehabilitación: 7 días
 Ejercicio escogido: 1
 Gráfica del ejercicio realizado:



- SUJETO 3:
 Edad: 27
 Sexo: Mujer
 Afección: Rodilla laxa
 Tiempo recibiendo rehabilitación: 1 mes
 Ejercicio escogido: 2
 Gráfica del ejercicio realizado:



- SUJETO 4:
 Edad: 50
 Sexo: Mujer
 Afección: Rotura de menisco
 Tiempo recibiendo rehabilitación: 1 año
 Ejercicio escogido: 4
 Gráfica del ejercicio realizado:



- SUJETO 5:
 Edad: 22
 Sexo: Mujer
 Afección: Luxación de rodilla
 Tiempo recibiendo rehabilitación: 3 meses
 Ejercicio escogido: 2
 Gráfica del ejercicio realizado:



- SUJETO 6:
Edad: 39
Sexo: Hombre
Afección: rotura de menisco
Tiempo recibiendo rehabilitación: 4 meses
Ejercicio escogido:1
Gráfica del ejercicio realizado: No ha conseguido completar el ejercicio de rehabilitación debido a dolor articular

CUESTIONARIO DE SATISFACCIÓN DE LOS USUARIOS:

- ¿La aplicación resulta intuitiva y fácil de manejar?
Respuesta: "Si, aunque hay algunas repeticiones que no las consigue registrar correctamente."
Respuesta: "Si, estaría bien que en vez de un mensaje de texto para indicarte cómo estas haciendo el ejercicio, que la aplicación te lo diga en voz alta."
Respuesta: "El ejercicio y la gráfica son buenos guías para el ejercicio."
- ¿La ortesis es cómoda de colocar?
Respuesta: "Si, aunque el cable de conexión a la caja es un poco aparatoso."
Respuesta: "Si, es muy facil de poner."
Respuesta: "No hace ninguna rozadura en la piel."
Respuesta: "Es muy cómoda gracias a las cintas elásticas, aunque el cable es un poco rollo."
- ¿Ha tenido algún problema a la hora de realizar el ejercicio?
Respuesta: "Si, eran demasiadas repeticiones."
Respuesta: "No, el ejercicio ha sido sencillo de realizar."
Respuesta: "No, estaba muy bien explicado en la imagen de la aplicación."
- ¿Tiene algún otro comentario o propuesta de mejora para este sistema?
Respuesta: "Que sea una ortesis inalámbrica."
Respuesta: "Que incluya más ejercicios."
Respuesta: "Que se puedan introducir el numero de repeticiones."

6. CONCLUSIONES

En este proyecto se ha llevado a cabo el diseño de una ortesis sensorizada junto con una aplicación móvil donde el usuario puede ver el progreso del ejercicio de rehabilitación. Se ha procurado realizar un trabajo muy completo, cubriendo todos los objetivos propuestos al inicio de este desarrollo.

El dispositivo resultante cumple las expectativas que se propusieron que debía cumplir una ortesis de uso doméstico destinada a la rehabilitación autónoma. Gracias al material escogido para la impresión de la ortesis, se ha conseguido una estructura ligera, resistente y económica. Uno de los inconvenientes que presenta la impresión en 3D es que la estructura debe ser impresa específicamente para la fisiología concreta de un paciente ya que si no se ajusta debidamente a la pierna del paciente podría tener efectos negativos en su proceso de rehabilitación. Mirándolo por el lado positivo, esto hace que el tratamiento de la afección sea totalmente personalizado para el paciente concreto.

Con respecto a la aplicación desarrollada, se han encontrado varias limitaciones en algunas de las funcionalidades que se querían conseguir. El ejemplo más claro ha sido intentar representar la gráfica del movimiento de la rodilla con una mayor calidad temporal. Sin embargo, la plataforma Blynk para el desarrollo de aplicaciones móviles sigue siendo un recurso muy novedoso y con grandes posibilidades para realizar proyectos orientados al Internet de las cosas. Se ha conseguido una interfaz para el usuario muy intuitiva y completa. Esto es un objetivo fundamental, ya que son los pacientes quienes interactúan con la aplicación.

El proyecto está orientado a la rehabilitación de afecciones específicas de la rodilla realizando un conjunto concreto de ejercicios. Si se quisiera comercializar un dispositivo sensorizado destinado a la rehabilitación habría que reconsiderar aspectos del diseño de la ortesis y ciertos parámetros de la aplicación. Se deberían añadir sistemas en la articulación de la ortesis con el objetivo de que se puedan modificar el tipo de trabajo que se quiera realizar sobre la lesión (inmovilización, de descarga, correctora...).

Así pues, el diseño realizado es una propuesta que pretende cubrir la siguiente necesidad: dar a los pacientes la oportunidad de realizar la

rehabilitación de forma autónoma. Gracias a la sensorización incorporada la aplicación móvil puede dar consejos durante la realización del ejercicio para que los usuarios puedan saber si lo están haciendo de manera correcta. Además, la información de los ejercicios realizados pueden ser enviados al médico rehabilitador para que pueda hacer un seguimiento de la mejora del paciente y así monitorizar si los ejercicios se realizan de manera óptima. Teniendo en cuenta el tiempo y los recursos disponibles para este trabajo, los resultados obtenidos tanto de la aplicación como del dispositivo fabricado son muy satisfactorios. En los siguientes apartados se hablarán del futuro de las ortesis y de las posibles mejoras que se podrían incorporar a este sistema

6.1. Posibles líneas futuras

6.1.1. Futuro de las ortesis

LA IMPRESIÓN 3D Y LAS ORTESIS

Con la aparición de la impresión en 3D ha aparecido un auge del uso de esta tecnología para el diseño de las prótesis y ortesis. Las ortesis personalizadas se han relacionado con la reducción del riesgo y la recurrencia de lesiones a través de la individualización de las características de amortiguación y soporte. Las aplicaciones incluyen, pero no se limitan a: alivio del dolor, aumento de la amortiguación, corrección de la deformidad flexible y aumento de la estabilidad. Los métodos actuales de mejores prácticas utilizados para obtener una impresión tridimensional (3D) de un paciente para su uso en la fabricación de ortesis son yeso y fundición de cajas de espuma. Cada uno de estos métodos está sujeto a posibles errores del profesional y son métodos muy engorrosos de realizar. En los últimos años, ha surgido tecnología que permite el uso del escaneo 3D usando el diseño asistido por computadora (CAD) y la fabricación asistida por computadora (CAM)(Figura 65). Estudios recientes informan calificaciones positivas de confort y parámetros de marcha biomecánicos con el uso de ortesis fabricadas con impresión 3D en comparación con las ortesis fabricadas con medios tradicionales. La adopción masiva de la impresión 3D para la fabricación de ortesis requerirá un rendimiento comprobado, una buena relación calidad-precio. Actualmente, muchas herramientas de escaneo 3D, programas CAD y máquinas de sinterización láser son inaccesibles para muchos profesionales debido al alto costo de adquisición [20] [21].

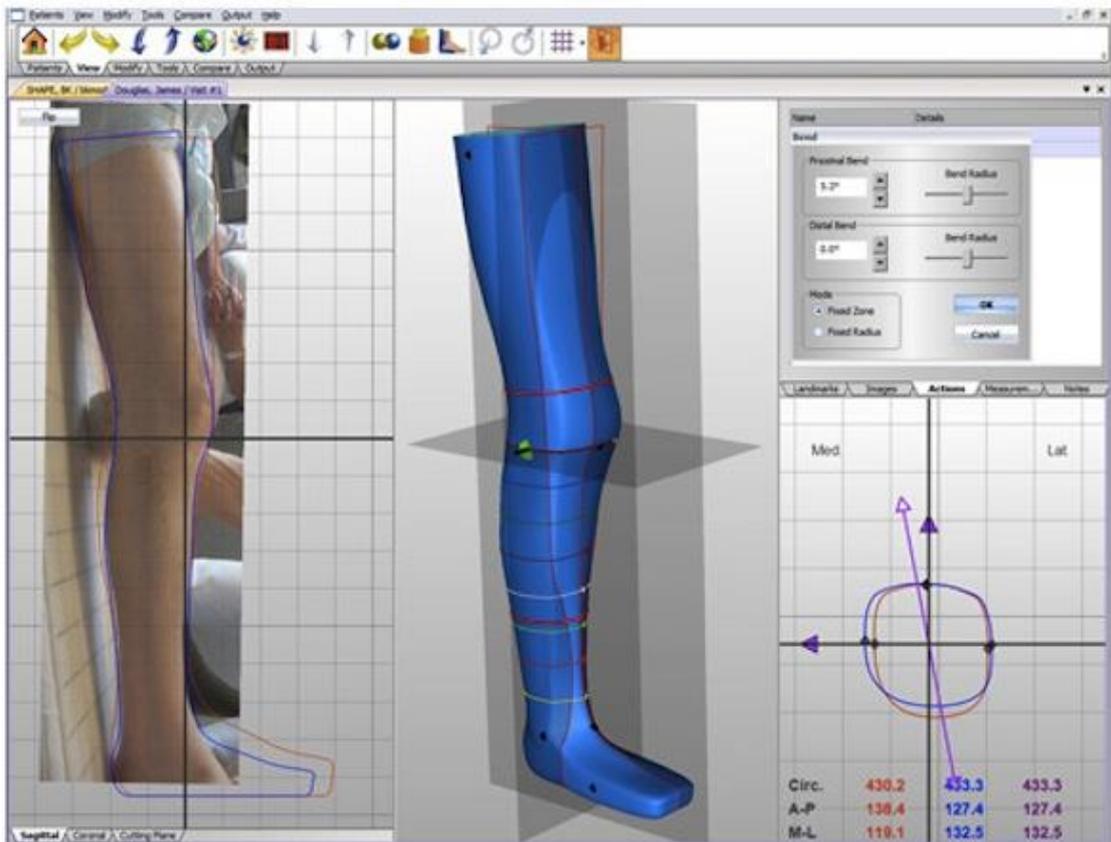


Figura 65. Escáner 3D usando CAM y CAD

APLICACIONES PARA LA REHABILITACIÓN EN EL HOGAR

Un área emergente de aplicación de tecnología portátil es el uso de sensores portátiles para facilitar la implementación de intervenciones de rehabilitación en el hogar [20]. Los sistemas que tienen como objetivo facilitar la implementación de programas de ejercicios de rehabilitación a menudo aprovechan la combinación de tecnología de detección y entornos de juegos interactivos o realidad virtual (VR). Por ejemplo, el Centro de Investigación de Ingeniería de Rehabilitación de la Universidad del Sur de California [21] se basa en los juegos de realidad virtual para abordar los desafíos de cumplimiento y motivación [22]. La tecnología de simulación de realidad virtual que utiliza dispositivos de interfaz especializados se ha aplicado para mejorar las habilidades motoras en sujetos sometidos a rehabilitación para abordar los déficits funcionales, como el alcance, la función de las manos y la marcha. Se ha propuesto que tales actividades basadas en realidad virtual se puedan realizar en el hogar a través de un enfoque de telerehabilitación para apoyar el mayor acceso de los pacientes a la rehabilitación y la programación de ejercicios preventivos. Cuando esto se pone en un contexto interactivo

basado en el juego, existe el potencial para mejorar el compromiso y la motivación necesarios para impulsar los cambios neuroplásticos que subyacen en el mantenimiento y la mejora del proceso motor. Sin embargo, los sistemas basados en el hogar deben ser asequibles y fáciles de implementar y mantener, a la vez que brindan la fidelidad interactiva necesaria para producir la actividad motora significativa necesaria para fomentar los objetivos de rehabilitación y promover la transferencia a actividades del mundo real.

Un ejemplo de tales sistemas es el sistema Valedo de Hocoma AG que se muestra en la siguiente imagen (Figura 66). El sistema Valedo es un dispositivo de entrenamiento médico para la espalda, que mejora el cumplimiento del paciente y le permite a uno lograr una mayor motivación mediante la Realimentación Aumentada en tiempo real basada en los movimientos del tronco. Transfiere los movimientos del tronco desde dos sensores inalámbricos a un entorno de juego motivador y guía al paciente a través de ejercicios diseñados específicamente para la terapia del dolor lumbar. Para facilitar el desafío al paciente y lograr un entrenamiento eficiente, los ejercicios se pueden ajustar de acuerdo con las necesidades específicas del paciente. Varios otros sistemas están actualmente en desarrollo. Por ejemplo, GE Healthcare está desarrollando un sistema inalámbrico de monitoreo médico que se espera permita recopilar datos fisiológicos y de movimiento, facilitando así las intervenciones de rehabilitación en el hogar. Otro ejemplo de tecnología de rehabilitación en el hogar es el Ejercitador de rehabilitación para accidentes cerebrovasculares desarrollado por Philips Research [23]. El Ejercitador de rehabilitación por accidente cerebrovascular entrena al paciente a través de una secuencia de ejercicios para el reentrenamiento motor, que son prescritos por el fisioterapeuta y se cargan en una unidad de paciente. Un sistema de sensor de inercia inalámbrico registra los movimientos del paciente, analiza los datos en busca de desviaciones de un objetivo de movimiento personal y proporciona retroalimentación al paciente y al terapeuta.

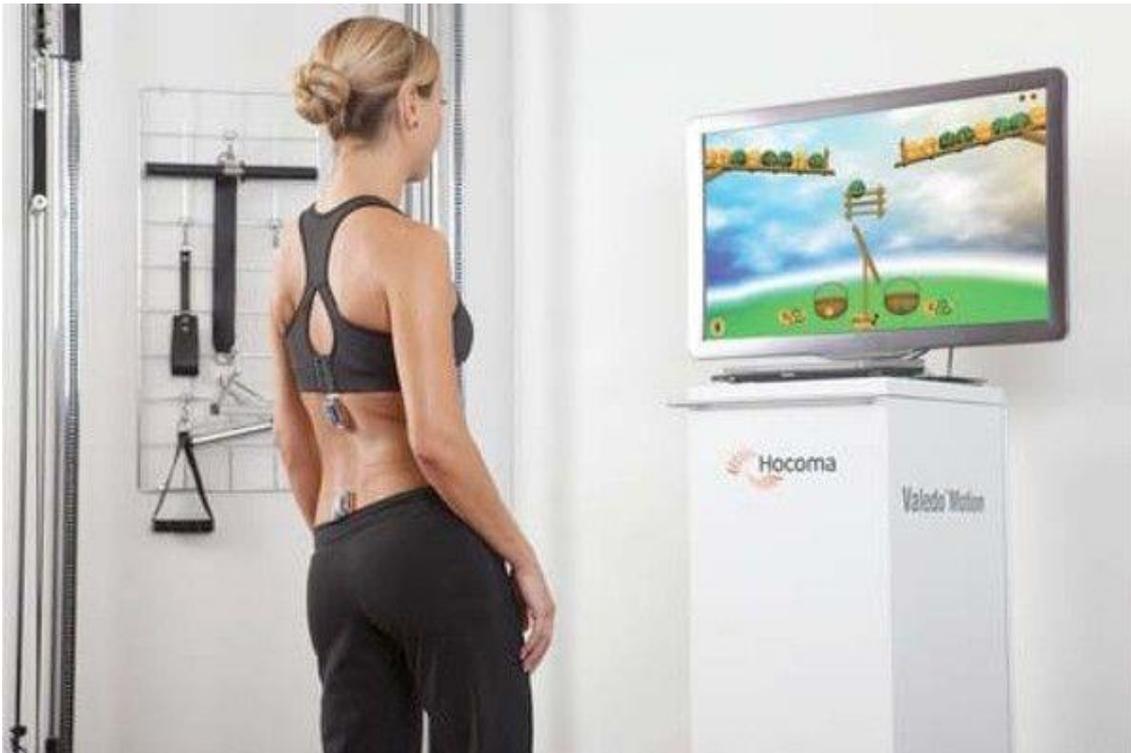


Figura 66. Sistema Valedo de Hocoma AG

Los grupos europeos han realizado grandes esfuerzos para desarrollar sistemas adecuados para las intervenciones en el hogar que dependen de la tecnología portátil. Un proyecto que formó parte de la iniciativa myHeart [24, 25] condujo al desarrollo de un sistema sensorial basado en prendas para facilitar las intervenciones de rehabilitación en el hogar. El sistema permite a los pacientes aumentar la cantidad de ejercicio motor que pueden realizar de forma independiente, proporcionándoles una retroalimentación en tiempo real basada en sensores portátiles integrados en la prenda a través de la extremidad superior y el tronco. Después de la fase de retroalimentación, los datos se almacenan en una ubicación central para su revisión y estadísticas. Las estaciones de trabajo se pueden instalar en casa o en el hospital para ayudar a los pacientes, independientemente de su ubicación. Otras dos iniciativas importantes en el campo incluyen los programas de investigación establecidos por los Centros TRIL y CLARITY en Dublín, Irlanda. El Centro TRIL reúne a la industria y la academia para realizar estudios de investigación en adultos mayores y examinar cómo la tecnología puede permitir la atención médica y social. El Centro CLARITY para tecnologías web de sensores se ocupa de la investigación del potencial de los datos del sensor para desempeñar un papel clave en la gestión de la salud personal. Los proyectos relevantes en los Centros TRIL y CLARITY incluyen:

Desarrollo y evaluación de un sistema remoto para evaluar la función cognitiva y mejorar el estado de alerta mental entre los adultos mayores en sus hogares.

Building Bridges, un programa de redes sociales que permite a las personas comunicarse con sus familias y con otros en la red sin experiencia previa en el uso de la computadora.

Tecnología para integrar la reconstrucción del día en línea, medidas psicométricas, evaluaciones ecológicas y marcadores biológicos en situaciones del mundo real.

Aplicaciones que monitorean el cumplimiento y brindan retroalimentación a los pacientes durante la realización del ejercicio de rehabilitación utilizando datos recopilados a través de sensores portátiles.

Otros proyectos realizados por grupos europeos que vale la pena mencionar son el proyecto TeleKat y el "Auxilium Vitae Volterra" en el Centro de Rehabilitación-Scuola Superiore Sant'Anna. El proyecto TeleKat (Universidad de Aalborg, Aalborg, Dinamarca) está aplicando la innovación dirigida por el usuario para desarrollar tecnología inalámbrica de teleasistencia domiciliar que permita a los pacientes con enfermedad pulmonar obstructiva crónica realizar un autocontrol de su estado y mantener actividades de rehabilitación en sus hogares. El proyecto de Tele-rehabilitación "Auxilium Vitae Volterra" en el Centro de Rehabilitación-Scuola Superiore Sant'Anna es un programa de rehabilitación cardíaca que aprovecha el uso de un sistema basado en sensores para monitorear remotamente a los pacientes en su hogar. El sistema incluye un cicloergómetro computarizado, un ECG de diagnóstico inalámbrico de 12 derivaciones, un sensor para la saturación de oxígeno en la sangre, un sistema de medición de presión arterial no invasivo y un sistema de videoconferencia de alto rendimiento.

6.1.2. Mejoras en el sistema actual

Con respecto al proyecto desarrollado en este trabajo se pueden realizar varias mejoras tanto en el diseño de la ortesis como en la aplicación. Además de añadir nuevas funcionalidades.

Para el diseño de la ortesis, la impresión en 3D sigue siendo la mejor opción, sin embargo, dada su estructura, podría incluirse sistemas de bloqueo de la articulación para que pueda funcionar tanto como ortesis sensorizada como una ortesis de inmovilización de la articulación. Se podría conseguir realizando un ajuste sencillo en los engranajes de la pieza impresa.

Otra de las mejoras que podrían implementarse en la ortesis es modificar el modo de sensorización. En vez de basado en el uso del potenciómetro como medidor angular de la rodilla, podrían implantarse dos acelerómetros, uno en la parte superior de la ortesis y otro en la parte inferior. De esta manera se obtendría de una manera más precisa la posición actual de la pierna junto con su movimiento angular.

Con respecto a la aplicación diseñada en la plataforma de Blynk, podrían implementarse más ejercicios de rehabilitación con nuevas comprobaciones, como el tiempo en el que debe completarse el ejercicio. También se podrían realizar estimadores de los ejercicios de rehabilitación, cuales se han realizado correctamente, cuales no se han podido terminar y cuales se han realizado de forma incorrecta.

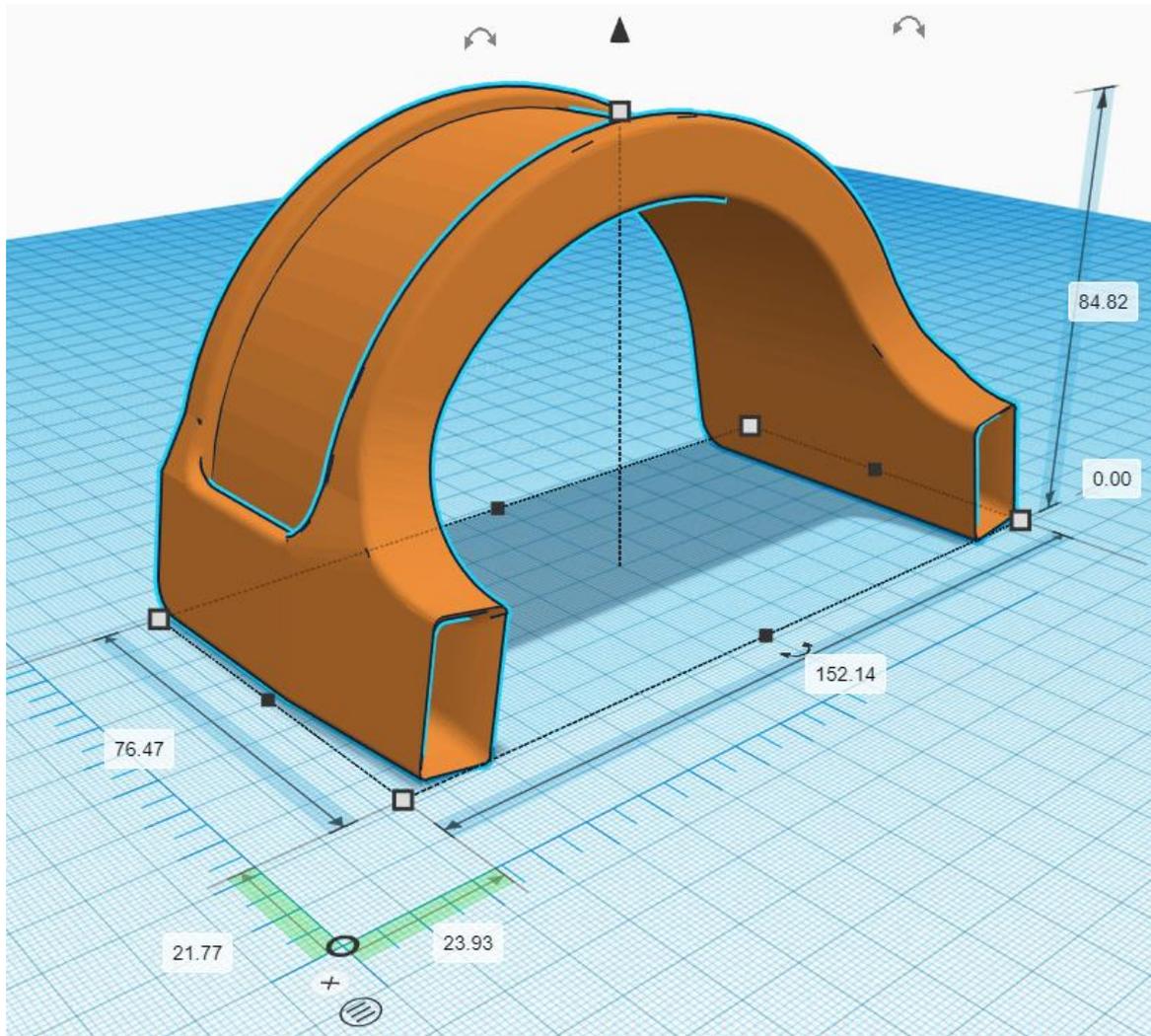
También se podría crear una base de datos para el profesional rehabilitador, en la que se recopile todos los datos de los pacientes, junto con una interfaz que muestre sus expedientes y su progreso en la rehabilitación.

Gracias a la recopilación de los datos recogidos durante todo el proceso de rehabilitación, se podrían aplicar técnicas de Big Data y Deep Learning para realizar un estudio sobre el progreso de las diferentes afecciones junto con los sistemas de rehabilitación empleados.

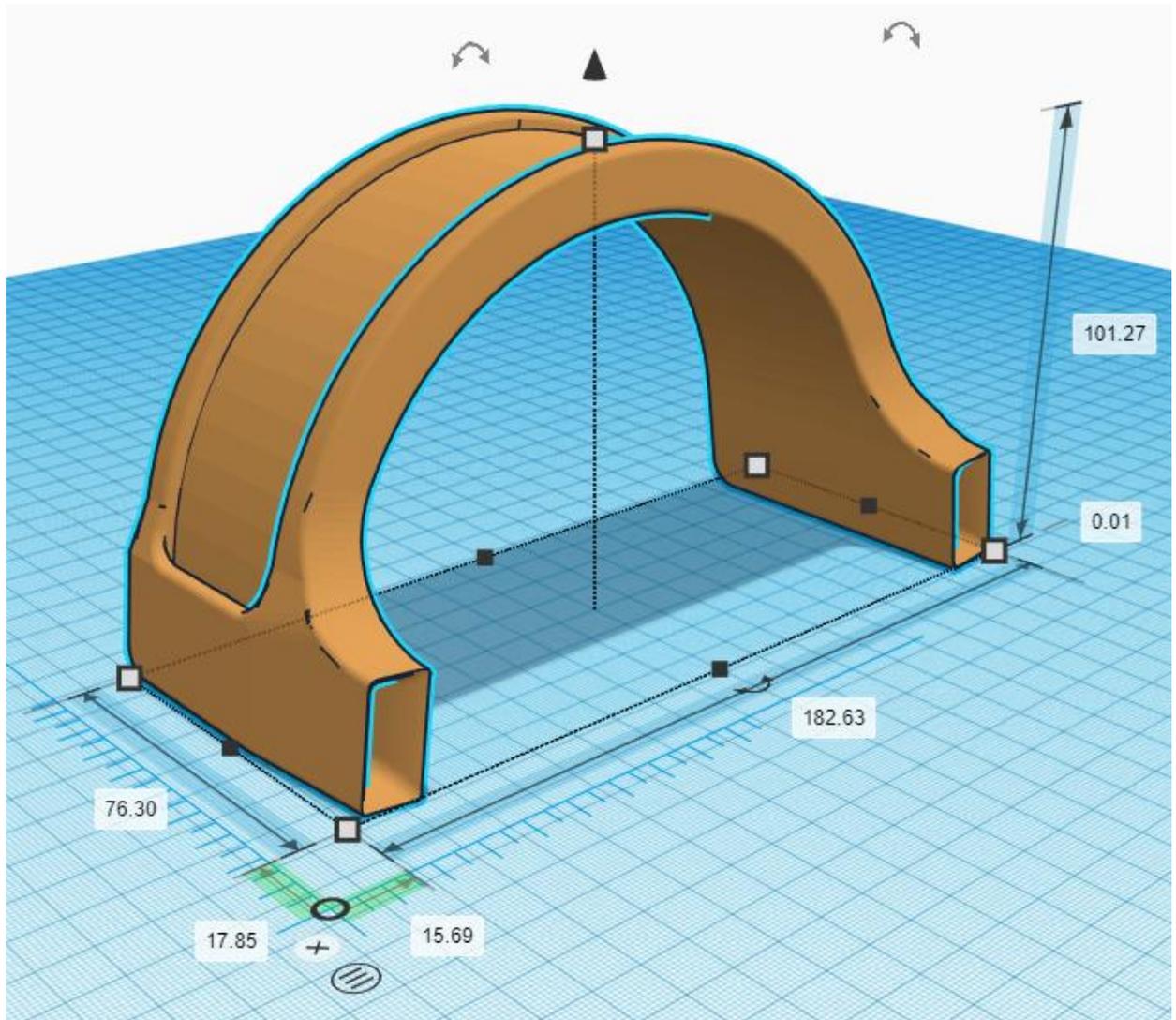
Finalmente concluir con que este sistema compuesto por ortesis sensorizada y aplicación para la rehabilitación se podría replicar para cualquier articulación que requiera terapia física de forma autónoma y que requiera un seguimiento del progreso de la afección.

7. ANEXOS

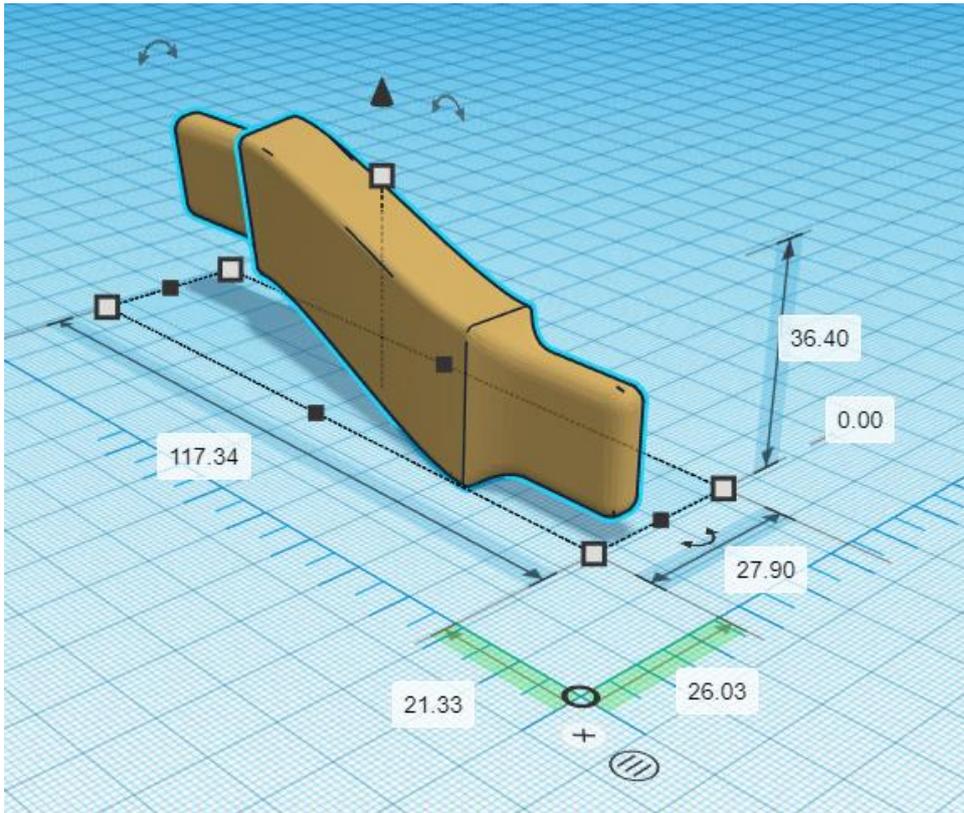
ARCO INFERIOR



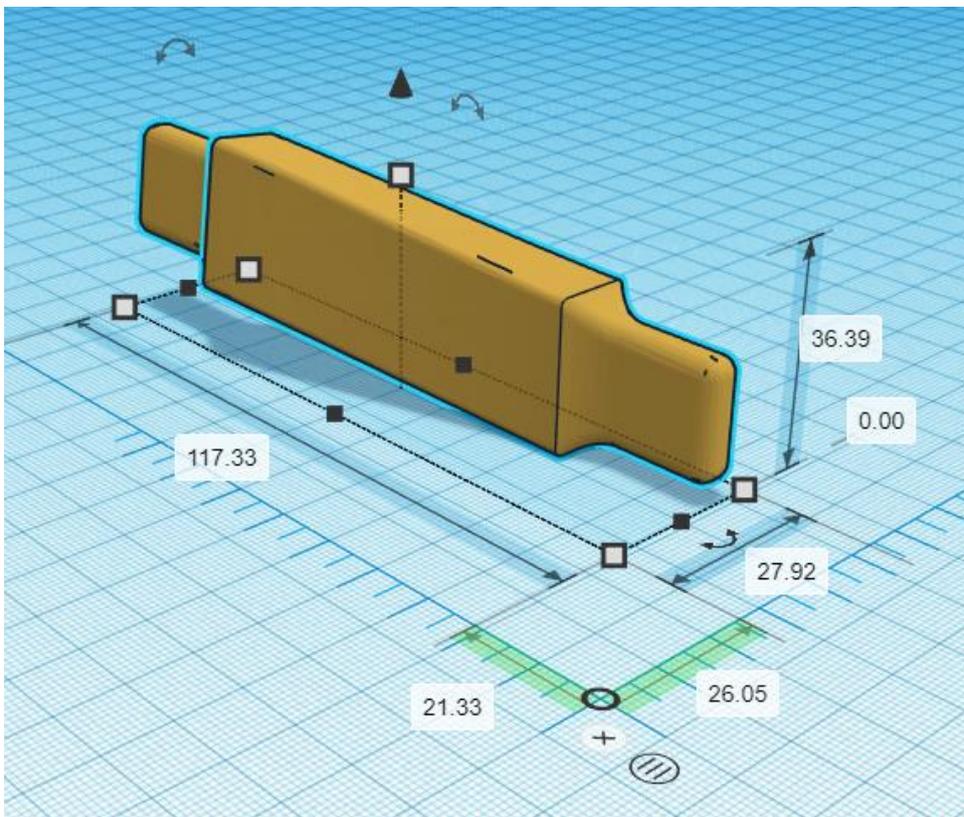
ARCO SUPERIOR



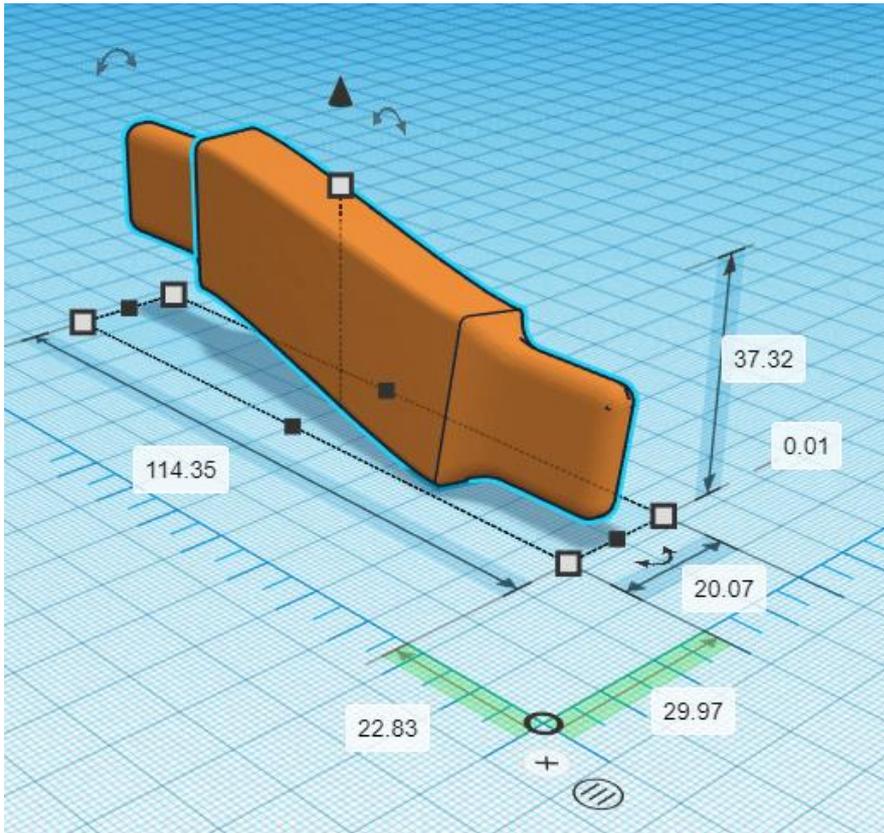
SUPERIOR LATERAL DERECHO



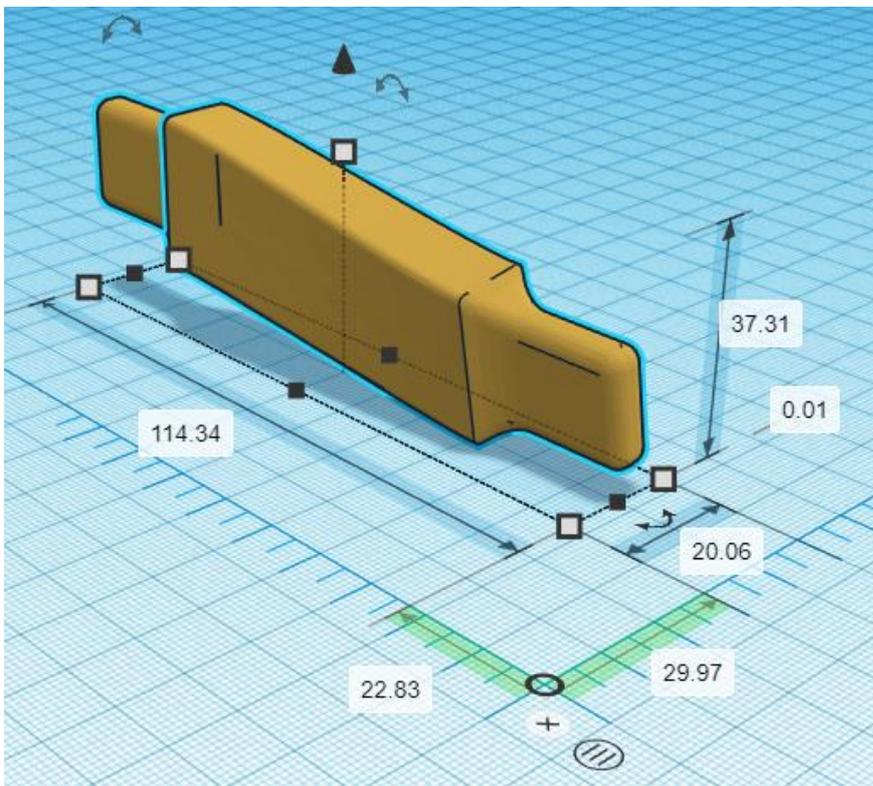
SUPERIOR LATERAL IZQUIERDO



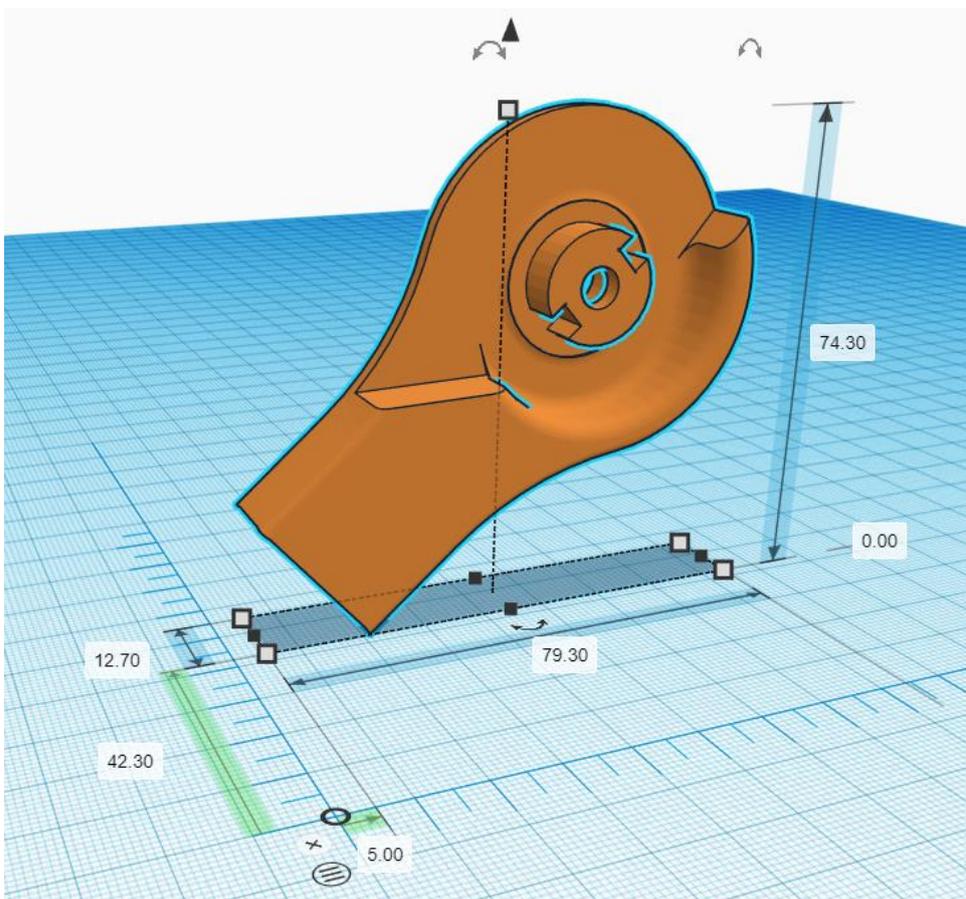
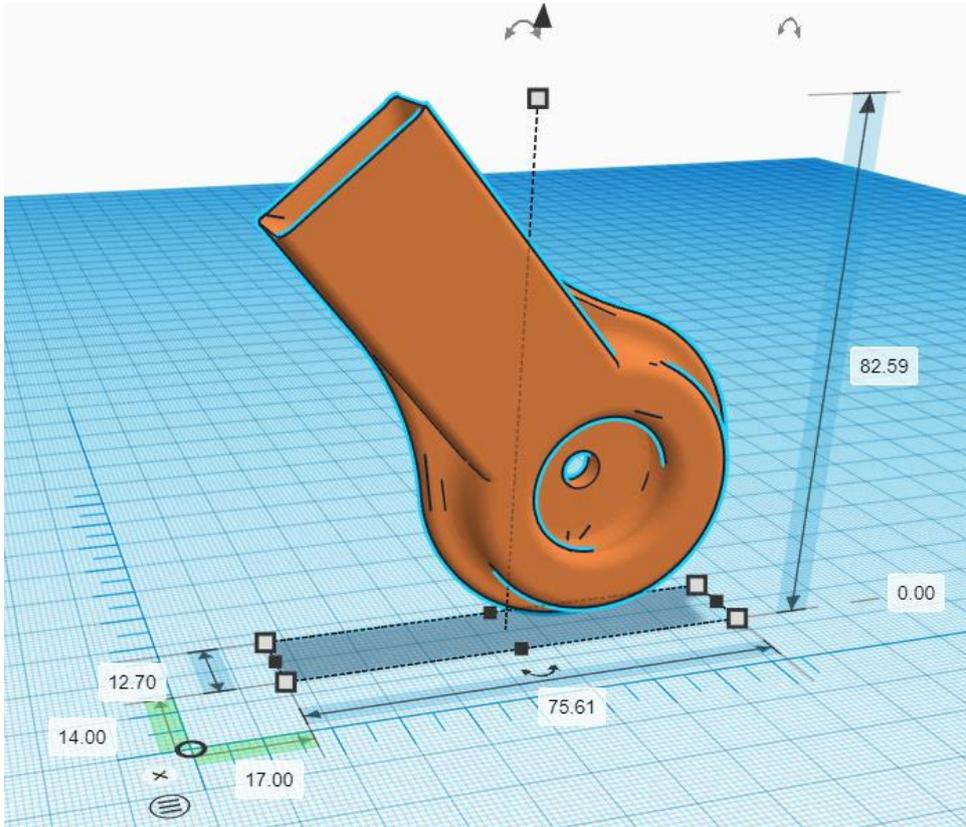
INFERIOR LATERAL DERECHO



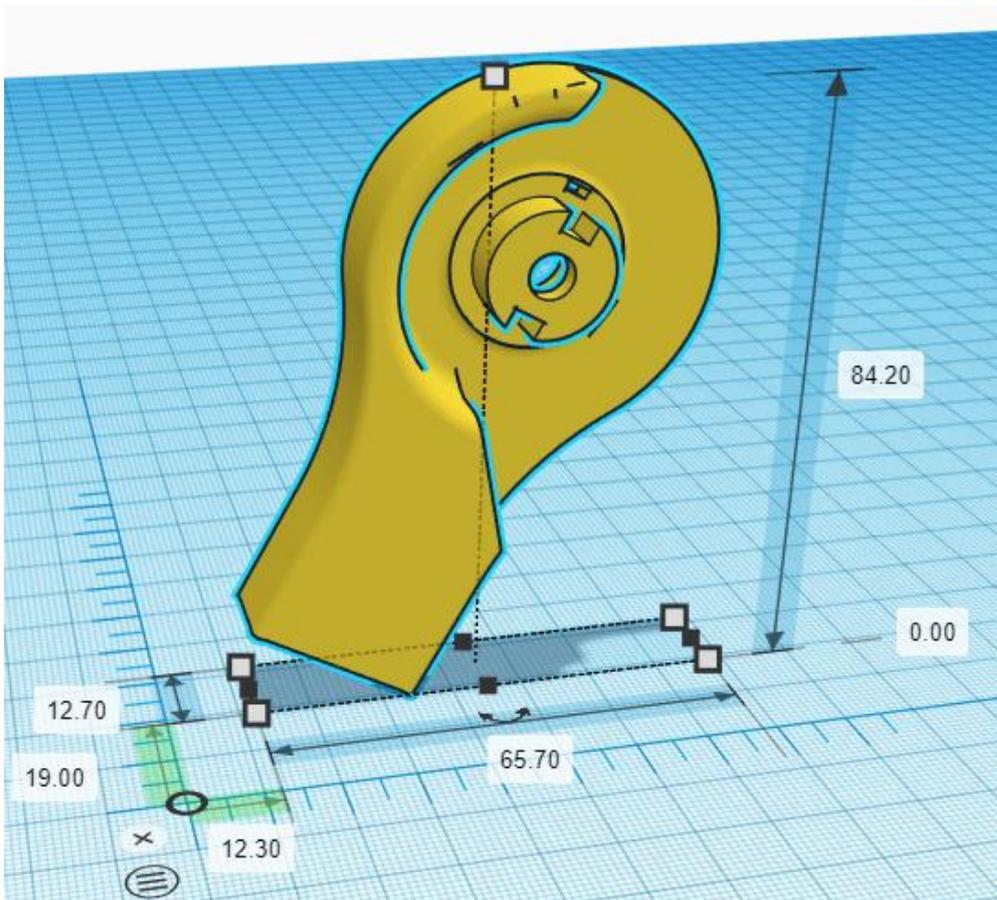
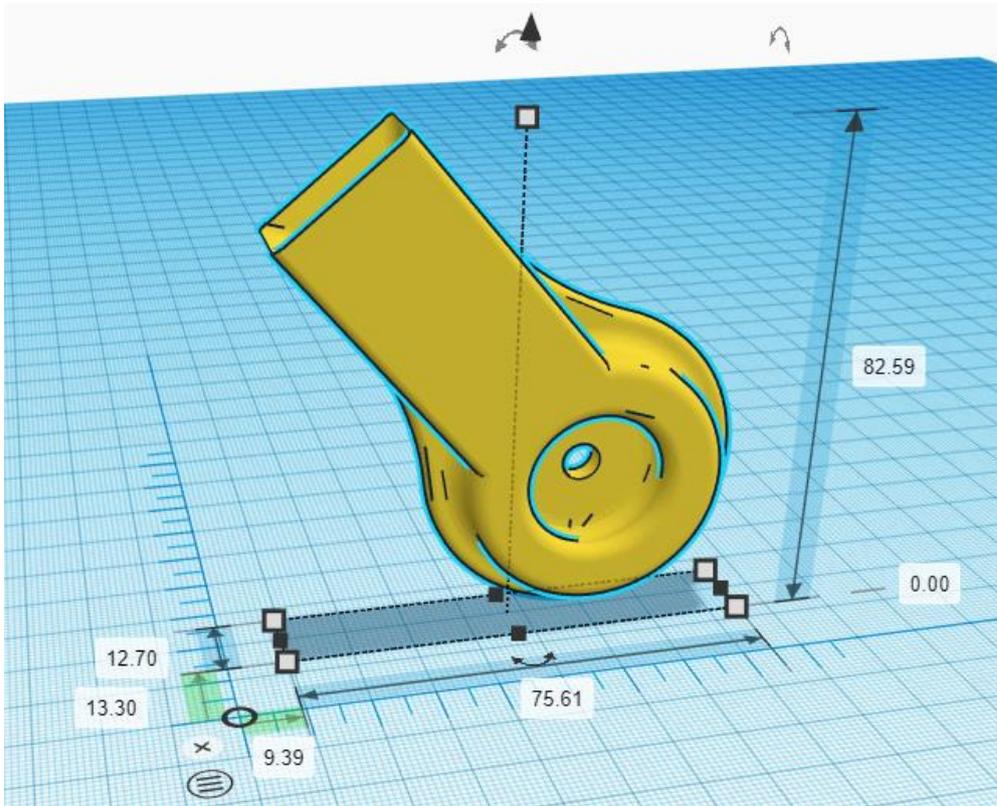
INFERIOR LATERAL IZQUIERDO



ENGRANAJE DERECHO



ENGRANAJE IZQUIERDO



8. BIBLIOGRAFÍA

- [1] Clínica de traumatología Martín Gómez, Granada
[<https://clinicamartingomez.es/anatomia-rodilla/>]
- [2] Traumatólogo Dr Vicente De La Varga, Málaga [<https://camde.es/anatomia-de-la-rodilla/>]
- [3] Biomechanics of the knee H. U. Dr. Peset de Valencia Sanjuan Cerveró R. Jiménez Honrado P. J. Gil Monzó E. R. Sánchez Rodríguez R. J. Fenollosa Gómez J.
- [4] Indicaciones de las ortesis en atención primaria, Francisco Vargas Negrín
- [5] Catálogo de Ortesis de rodilla de la empresa Medi
- [6] Web oficial de Arduino, especificaciones para Arduino Uno
[<https://store.arduino.cc/arduino-uno-rev3>]
- [7] Web oficial de Prometec [<https://www.prometec.net/bt-hc06/>]
- [8] Ingeniero industrial Luis Llamas, Web oficial, especificaciones para modelo HC-06
[<https://www.luisllamas.es/conectar-arduino-por-bluetooth-con-los-modulos-hc-05-o-hc-06/>]
- [9] Ing. Luis Llamas Web oficial, especificaciones para encoder rotativo
[<https://www.luisllamas.es/arduino-encoder-rotativo/>]
- [10] Web oficial de Arduino, especificaciones para Arduino Mega 2560
[<https://store.arduino.cc/mega-2560-r3>]
- [11] Web oficial ResistorGuide [<http://www.resistorguide.com/potentiometer/>]
- [12] Web oficial Tinkercad [<https://www.tinkercad.com/>]
- [13] Web oficial BQ [<https://www.bq.com/es/landing-mundo-3d>]
- [14] Web oficial Ultimaker [<https://ultimaker.com/software/ultimaker-cura>]
- [15] ESP8266EX Datasheet
- [16] ESP-01 WiFi Module Ai Thinker

[17] Página web [<https://programarfacil.com/podcast/como-configurar-esp01-wifi-esp8266/>]

[18] Web oficial Blynk [<https://blynk.io/>]

[19] Web oficial documentación de Blynk [<http://docs.blynk.cc/>]

[20] A review of wearable sensors and systems with application in rehabilitation, Shyamal Patel, Hyung Park, Paolo Bonato, Leighton Chan & Mary Rodgers, Journal of NeuroEngineering and Rehabilitation volume 9, Article number: 21 (2012)

[21] USC Rehabilitation Engineering Research Center [<http://www.isi.edu/research/lerc>]

[22] Rizzo A, Requejo P, Winstein CJ, Lange B, Ragusa G, Merians A, Patton J, Banerjee P, Aisen M: Virtual reality applications for addressing the needs of those aging with disability. Stud Health Technol Inform 2011, 163: 510-516.

[23] Lanfermann G, te Vrugt J, Timmermans A, Bongers E, Lambert N, van Acht V: Philips stroke rehabilitation exerciser. In Technical Aids for Rehabilitation-TAR 2007: January 25-26, 2007 2007. Technical Aids for Rehabilitation-TAR 2007-Technical University of Berlin; 2007.

[24] Giorgino T, Tormene P, Lorussi F, De Rossi D, Quaglini S: Sensor evaluation for wearable strain gauges in neurological rehabilitation. IEEE Trans Neural Syst Rehabil Eng 2009, 17: 409-415.

[25] Giorgino T, Tormene P, Maggioni G, Pistarini C, Quaglini S: Wireless support to poststroke rehabilitation: myheart's neurological rehabilitation concept. Information Technology in Biomedicine, IEEE Transactions on 2009, 13: 1012-1018.

[26] Diseño original de la ortesis en la página de Thingiverse [<https://cdn.thingiverse.com/site/js/thingiverse/build/makes-066f3a76bf.js>].

[27] Marco Schwartz, Internet of Things with ESP8266, 2016.

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer a mi tutor Alonso ya que sin él este proyecto seguiría siendo una mera idea. También al departamento de rehabilitación del Hospital Río Hortega y al Centro Regional de Rehabilitación de Medicina Deportiva de Valladolid. Agradecer especialmente a mi familia por el apoyo que me han dado durante el desarrollo del TFG y por ser mis "conejiillos de indias" para ver si funcionaba o no. Y finalmente a Lorena, Fran, Andrea, Vic, Javi, Azienchio y Laura, quienes, aunque no sean familia, como si lo fueran, por escucharme hablar incansablemente del trabajo y nunca dejaron de creer que lo conseguiría sacar adelante.