



POLITÉCNICA



INDUSTRIALES
ETSII | UPM

DISEÑO E IMPLEMENTACIÓN DE UNA INTERFAZ ELECTROMECAÁNICA PARA EL MANEJO DE SILLAS DE RUEDAS MOTORIZADAS

Trabajo de Fin de Grado
José Antonio Cué Palero

Tutores:

José Ignacio Serrano Moreno

Antonio Barrientos Cruz

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES

Febrero 2018



CENTRO DE AUTOMÁTICA Y ROBÓTICA



CSIC

CONSEJO SUPERIOR DE INVESTIGACIONES CIENTÍFICAS

*“Cuando lo hicisteis con uno de estos mis hermanos más pequeños,
conmigo lo hicisteis.”*

Mt 25,40

AGRADECIMIENTOS

En primer lugar, quería agradecer a mi tutor, José Ignacio Serrano, toda su labor y esfuerzo a lo largo de este trabajo. Desde el comienzo, me ha guiado, orientado, aconsejado y animado. Su comprensión de mis circunstancias personales y sus sabios consejos profesionales han sido muy provechosos para mí. Le agradezco también el haberme dado la oportunidad de realizar un TFG como este, pues personalmente, no entendería haber hecho uno distinto, que no hubiese tenido una aplicación tan directa sobre las personas como la tiene este proyecto.

También doy las gracias a Antonio Barrientos, por facilitarme el contacto y acceso a este proyecto, y por ser un profesor capaz de transmitir su ilusión y compromiso con lo que enseña, algo al alcance de muy pocos, pero que motiva mucho el aprendizaje.

Agradezco igualmente, a todas las personas del CAR de Arganda, que, de un modo u otro, han contribuido a este proyecto.

Finalmente, doy las gracias a mi familia por su apoyo, implicación y ánimo transmitido para conmigo. Con eso queda dicho todo.

Por último, me queda el agradecer a Dios el haber podido llevar a cabo este trabajo. A Él se lo ofrezco.

RESUMEN

El presente Trabajo de Fin de Grado (TFG) comprende el diseño y desarrollo de un dispositivo que se adapte al joystick de una silla de ruedas para desplazarlo, a partir de los datos recibidos vía Bluetooth, procedentes de un sensor que se coloca en la cabeza del usuario.

En la actualidad, existen muy numerosos casos de hipotonía cervical, que se manifiesta en una incorrecta posición de la cabeza como consecuencia de la falta de tono en los músculos del cuello. Son múltiples las causas que la originan, como ictus, enfermedades neurodegenerativas o parálisis cerebral, entre otras, que afectan a un elevado número de personas.

De hecho, la parálisis cerebral es la causa de discapacidad motórica más frecuente en niños y una de las más frecuentes también en adultos. El hecho de cursar con hipotonía cervical, al igual que para el resto de patologías, repercute negativamente en la salud del paciente.

Esto se debe a que una postura incorrecta de la cabeza acarrea cuantiosos problemas en el bienestar de una persona, y viceversa; cuando está bien posicionada, se mantiene un adecuado estado de salud.

La influencia de la postura cervical ha sido, y es, objeto de diversos estudios para conocer sus campos de afectación. Esos estudios aseguran que la posición de la cabeza repercute a nivel de la salud, en la respiración, el ritmo cardiaco, la deglución o el campo visual, y en aspectos más sociales, como en la comunicación o el confort.

Este proyecto trata de fomentar el movimiento de los músculos cervicales, mediante el control de una silla de ruedas eléctrica con la cabeza, para prevenir la aparición de problemas secundarios derivados de la falta de higiene postural.

Actualmente, ya existen distintas soluciones para mover una silla de ruedas con la cabeza, además de con otras partes del cuerpo como manos, pies, boca o barbilla. El inconveniente de estas soluciones es que presentan un precio elevado, por lo que es una de las cuestiones que se han de solventar en este trabajo.

La necesidad de este proyecto la detectó la confederación ASPACE junto con la empresa Werium Assistive Solutions, cuando trabajando con niños con parálisis se dieron cuenta de que no eran capaces de controlar sus propias sillas de ruedas, porque tenían afectadas las extremidades superiores. No así ocurría con el cuello, donde observaron que tenían mayor movilidad, viendo que ejercitarlo sería muy beneficioso para ellos.

Esta necesidad se trasladó al Consejo Superior de Investigaciones Científicas (CSIC) con el fin de hallar una solución adecuada.

Para ello, se establecieron unos objetivos que guiaran y definieran la construcción del dispositivo desde su concepción hasta su puesta en funcionamiento, de manera que:

- Sea capaz de recibir señales de Bluetooth procedentes de un sensor.
- Permita accionar unos motores en función de esas señales, para controlar el joystick de la silla de ruedas por medio de un dispositivo mecánico.
- Sea reconfigurable para poderlo adaptar a distintos niveles de movilidad del cuello.
- Implice al usuario como parte activa del sistema de rehabilitación mediante el movimiento de la silla de ruedas, viéndose esto como una actividad más lúdica que sanitaria.
- Sea portátil para poder llevarlo sobre la silla de ruedas.
- Tenga un coste lo más reducido posible.
- Pueda ser utilizado por personal no técnico para su configuración.
- Cumpla las especificaciones de seguridad expuestas en el real decreto 1591/2009, del 16 de octubre por el que se regulan los productos sanitarios, incluido en el Boletín Oficial del Estado número 268, del 6 de noviembre de 2009.

Con el objetivo de diseñar y desarrollar un dispositivo de manera eficiente, se ha establecido un proceso de cinco etapas: análisis de requisitos, diseño, desarrollo, integración e implementación y evaluación y conclusiones.

En la etapa de diseño se distinguen dos partes: el diseño de la estructura y el del sistema electrónico.

La etapa de integración e implementación transcurre, en realidad, en paralelo con la etapa de desarrollo, ya que se han realizado pruebas durante el montaje del sistema electrónico, en sus dos partes por separado, Bluetooth y motores, y una vez integrado el sistema.

Es por eso que se ha realizado un programa principal, que implementa la funcionalidad básica del dispositivo, y otros programas auxiliares, para realizar las pruebas de la parte del Bluetooth y de los motores.

En el programa principal, hay una parte inicial de variables que puede modificar el terapeuta o fisioterapeuta que use el dispositivo, con las que se puede elegir el ángulo mínimo y máximo que se van a considerar en el movimiento de la cabeza, el plano con el que se van a realizar los giros (frontal o transversal), y el nivel de dificultad, que

efectúa más o menos desplazamiento del joystick, según sea menor o mayor respectivamente.

Dicho programa incluye la lectura de los datos del sensor y el movimiento de los motores atendiendo a la configuración de las variables anteriores.

Las pruebas realizadas con los programas auxiliares corroboran el correcto funcionamiento de cada parte independientemente. Por su parte, las pruebas con el programa principal verifican la adecuada integración entre el Bluetooth y los motores, y confirman que se cumple la funcionalidad pensada para el dispositivo.

Esto es, cuando el ángulo recibido está por debajo del mínimo, el joystick retorna a su posición de equilibrio; no moviéndose en caso de ya estar en ella. Si está entre el mínimo y el máximo, se mueve proporcionalmente a su relación con el máximo. Y si está por encima del máximo, se mueve como si tuviese tal valor.

También se observa en dichas pruebas que, según se aumenta el nivel de dificultad, el desplazamiento efectivo es menor. Asimismo, se aprecia que al seleccionar el ángulo frontal o el transversal, el sistema ignora el valor dado al otro ángulo, tal y como se había planteado.

Como líneas futuras de investigación y desarrollo se han identificado algunas mejoras del dispositivo en aspectos como la adaptabilidad a otros tipos de joystick pudiendo expandir así su aplicabilidad. Por otro lado, sería interesante incluir una interfaz de usuario que facilitase la configuración del dispositivo. Igualmente sería conveniente aumentar la autonomía de uso, conectando el Arduino y los motores a la alimentación de la silla.

Otro aspecto que resultaría útil sería el guardado de los datos procedentes del sensor, para su análisis posterior. Algo más avanzado se encontraría en el desarrollo de un algoritmo adaptativo que se modificase automáticamente durante su ejecución, adecuándose al estado del usuario. Un último aspecto a mejorar, sería el incremento de la protección del sistema, tanto del cableado como del dispositivo de control.

Por último, se ha realizado un análisis de los impactos del proyecto a nivel sanitario, económico, social y ambiental. El principal impacto es a nivel sanitario, pues el proyecto constituye un sistema de entrenamiento cervical para la rehabilitación de esta zona, y la prevención de la degeneración muscular, que origine patologías graves derivadas de una postura incorrecta, consecuencia de la hipotonía cervical.

En cuanto nivel económico, el principal impacto está en la reducción del coste frente a otras alternativas, haciendo más abordable el acceso a estas soluciones a las familias y a los centros sanitarios que trabajan con personas que necesitan este tipo de productos.

El impacto social radica, principalmente, en el beneficio sanitario que se deriva de la aplicación del producto, influyendo positivamente tanto en el paciente como en su entorno.

Por su parte, el impacto ambiental tiene dos vertientes, por un lado, la posibilidad de reutilización múltiples veces del dispositivo, y por otro, un impacto algo más negativo, que es la utilización de pilas y baterías para la alimentación del Arduino y de los motores; aunque, el hecho de ser recargables para ambos, permite alargar su vida útil.

Por todo lo expuesto anteriormente, cabe concluir que, en este Trabajo de Fin de Grado, se ha desarrollado un dispositivo de bajo coste que permite controlar el joystick de una silla de ruedas motorizada, fomentando el movimiento cervical y previniendo al mismo tiempo, futuros problemas de salud.

Palabras clave: hipotonía cervical, parálisis cerebral, joystick, dispositivo, Bluetooth, motores, estructura.

Código UNESCO: 1203, 3204, 3304,3307, 3311, 3313

ÍNDICE

AGRADECIMIENTOS.....	5
RESUMEN.....	7
ÍNDICE.....	12
1. INTRODUCCIÓN.....	16
1.1. Motivación.....	16
1.2. Parálisis cerebral.....	22
1.3. Sistemas de movilidad para gente con discapacidad.....	25
1.4. Estado del arte.....	28
1.4.1. Control manual.....	29
1.4.2. Control con la cabeza.....	32
1.4.3. Control con el pie.....	33
1.4.4. VR2.....	34
2. OBJETIVOS.....	36
3. METODOLOGÍA.....	38
3.1. Análisis de requisitos.....	38
3.1.1. Marco legal.....	39
3.2. Diseño.....	44
3.2.1. Diseño del dispositivo.....	44
3.2.1.1. Componentes.....	46
3.2.1.2. Montaje.....	59
3.2.1.2.1. Montaje del circuito.....	59
3.2.1.2.2. Montaje de la estructura.....	63
3.2.2. Desarrollo del software.....	66
3.2.2.1. Programa principal.....	67

3.2.2.2. Programas auxiliares.....	70
4. DESCRIPCIÓN DEL DISPOSITIVO. ALTERNATIVAS CONSIDERADAS.....	72
4.1. Características generales.....	72
4.2. Variables programables.....	73
4.3. Ejemplos de funcionamiento.....	74
4.4. Alternativas.....	77
4.4.1. Diseño físico.....	77
4.4.2. Componentes.....	78
4.4.3. Software.....	86
5. PRUEBAS, RESULTADOS Y CONCLUSIONES.....	88
5.1. Pruebas y resultados.....	88
5.1.1. Pruebas del sistema electrónico.....	88
5.1.1.1. Pruebas del Bluetooth.....	88
5.1.1.2. Pruebas de los motores.....	90
5.1.1.3. Pruebas del conjunto integrado.....	92
5.1.2. Pruebas de simulación.....	94
5.2. Análisis de resultados y conclusiones.....	97
5.2.1. Análisis y conclusiones de las pruebas del Bluetooth.....	97
5.2.2. Análisis y conclusiones de las pruebas de los motores.....	98
5.2.3. Análisis y conclusiones de las pruebas del conjunto integrado.....	98
5.2.4. Análisis y conclusiones de las pruebas de simulación.....	99
6. APLICACIONES E IMPACTOS.....	101
6.1. Aplicaciones en medicina.....	101
6.2. Otras aplicaciones.....	102
6.3. Impactos.....	102
6.3.1. Impacto sanitario.....	103

6.3.2. Impacto económico.....	103
6.3.3. Impacto social.....	104
6.3.4. Impacto ambiental.....	105
7. LÍNEAS FUTURAS.....	107
8. PLANIFICACIÓN TEMPORAL Y PRESUPUESTO.....	111
8.1. Planificación temporal.....	111
8.2. Presupuesto.....	115
9. BIBLIOGRAFÍA.....	119
10. ÍNDICE DE FIGURAS.....	124
11. ÍNDICE DE TABLAS.....	126
ANEXO 1: Código programa principal.....	128
ANEXO 2: Código prueba Bluetooth.....	132
ANEXO 3: Código prueba motores.....	134

1.INTRODUCCIÓN

1.1. Motivación

La hipotonía cervical es la falta de tono muscular en la zona del cuello que se manifiesta en una incorrecta posición de la cabeza. Existen numerosas causas que originan hipotonía cervical. Algunas de ellas son traumatismos provocados por accidentes de tráfico, ictus, enfermedades neurodegenerativas, lesiones medulares o parálisis cerebral.

Por otro lado, la parálisis cerebral es una de las causas de discapacidad motórica más frecuente y la más frecuente en niños. Se da en una de cada 500 personas en España, registrándose un total de 120000 personas con parálisis cerebral. En el mundo, se estima que son 17 millones de personas. Actualmente, la parálisis cerebral no tiene cura, pero sí se puede trabajar en cuatro áreas para potenciar la mayor autonomía de la persona: la logopedia, fisioterapia, apoyos educativos y terapia ocupacional.

El trabajo que se desarrollará se centra en la rehabilitación física de las personas que sufren hipotonía cervical, ya que ésta puede derivar en problemas más serios si no se corrige o trata a tiempo. Para ello, se usará un sistema basado en el control de una silla de ruedas a partir de movimientos de la cabeza, obligando a articular los músculos cervicales con ese fin.

De esta forma, en este proyecto se le da a la silla de ruedas el valor añadido de ser un elemento de rehabilitación (no solo un producto de apoyo), jugando un papel fundamental en el bienestar de la persona, ya que no solo previene futuras complicaciones o enfermedades derivadas de una postura incorrecta, sino que al servir para fortalecer los músculos cervicales se consiguen numerosas mejoras en la calidad de vida.

A continuación, se va a exponer la influencia de una correcta posición de cabeza sobre la calidad de vida para poner de manifiesto la relevancia de este proyecto al contribuir a tener un adecuado tono en los músculos del cuello.

Posicionar adecuadamente la cabeza y ser capaz de mirar a los ojos es el primer paso para interactuar con otra persona y revela cómo se siente la persona. Cuando no se tiene esta habilidad, como en el caso de la parálisis cerebral, se pierde esta oportunidad.

La posición de la cabeza influye en: la respiración, el ritmo cardiaco, la deglución, la comunicación, el campo visual, aspectos sociales y confort, así como en el tono muscular y reflejos involuntarios. Se van a ir tratando cada uno de estos aspectos en lo que sigue.

- Respiración: El hecho de tener un adecuado alineamiento del cuello sitúa las vías respiratorias en una posición óptima para respirar. Frecuentemente, al darse una hipotonía cervical la cabeza se sitúa hacia abajo causando hipoventilación.
- Ritmo cardiaco: Como consecuencia de la hipoventilación antes nombrada, al haber menos nivel de oxígeno, se produce un aumento del ritmo cardiaco provocando fatiga.
- Deglución: La incapacidad de ingerir comida con seguridad tiene un impacto significativo en la nutrición y como consecuencia, en la salud, el crecimiento y la fatiga. Cuando la cabeza está hacia abajo existe una mayor dificultad para mantener la comida y la insalivación dentro de la boca. Además, la cavidad oral es clave para la ingestión de alimentos, y más aún para el intercambio de aire. En una deglución normal, el hueso hioide es elevado empujando hacia arriba a la laringe y permitiendo así a la epiglotis cubrir la tráquea para evitar la entrada de comida. Cuando se da una deficiente posición de cabeza y cuello esta función no se realiza adecuadamente produciendo un cubrimiento incompleto de la tráquea con el correspondiente peligro que ello conlleva.
- Comunicación: La gente con parálisis en muchas ocasiones se comunica fundamentalmente mediante el lenguaje no verbal, por lo que es primordial el poder percibir su lenguaje corporal y expresiones faciales. Por otro lado, si alguien se comunica bien mediante el lenguaje verbal, una posición erguida mejora la expansión pulmonar permitiendo una adecuada respiración y un buen volumen del habla, viéndose claramente afectado en el caso de hipotonía cervical.
- Campo visual: El campo visual de las personas con parálisis cerebral se ve limitado tanto si tienen la cabeza mirando al suelo, como si tienen el cuello hiperextendido mirando al techo, originando una actitud pasiva al haber muy poca interacción con el entorno.

- Aspectos sociales: Cuando una persona tiene la cabeza en posición vertical su aspecto mejora, recibiendo por ello comentarios positivos que contribuyen a su autoestima.
- Confort: A lo largo del tiempo, una posición estática causará un sobre-alargamiento de unos músculos y un acortamiento de los contrarios. Esto puede influir en la higiene y el vestirse. El sobre-alargamiento añade la dificultad de desarrollar fuerza y control muscular.
- Tono muscular y reflejos primitivos: Un tono excesivo y patrones espásticos de movimiento hacen que sea muy difícil controlar la posición sentada. Los reflejos primitivos son intrincados patrones de movimiento como respuesta a estímulos sensoriales en cabeza y cuello, originando cambios en el tono muscular por todo el cuerpo. Si no existe un adecuado soporte de cuello y cabeza puede facilitarse la activación de estos reflejos involuntarios. Si esto no se corrige, puede desencadenar una postura asimétrica crónica lo que puede llegar a causar escoliosis, inclinación pélvica y otros problemas de cadera.

Por tanto, si no se corrigen estos problemas de postura, fortaleciendo, como en este proyecto, los músculos cervicales, se facilitará la aparición de movimientos involuntarios y problemas secundarios derivados de la falta de higiene postural.

Control postural en parálisis cerebral

Para ahondar más en la cuestión de la importancia de la postura se llevó a cabo un estudio a cargo del departamento de psicología humana y del instituto de neurociencia de la universidad de Oregon (S. Saavedra, M. Woollacot, P. Donkelaar, 2010). El objetivo era determinar la influencia de la visión y del soporte troncal en la estabilidad de la cabeza durante el sentado estático de niños con distintos tipos y grados de afectación de parálisis cerebral; ya que estos presentan déficits en la posición de la cabeza incluso sentados, como ya demostró dicho equipo en investigaciones anteriores.

Para ello, se probaron tres tipos de restricciones posturales: soporte troncal, pélvico y sin soporte, y dos circunstancias de visión: ojos abiertos y ojos cerrados.

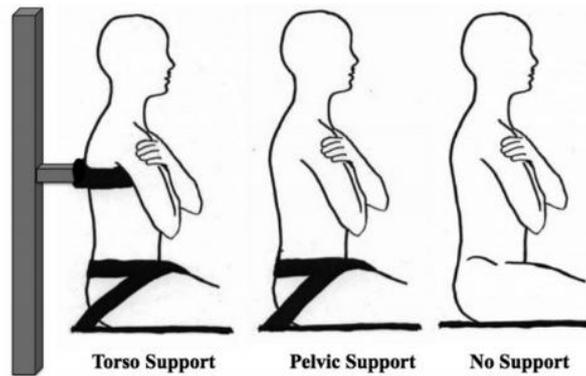


Figura 1.1: Tipos de restricciones posturales

En primer lugar, hay que decir que uno de los objetivos primordiales del control postural es estabilizar la cabeza en el espacio, pues esto es fundamental para la orientación y el equilibrio.

El estudio inició determinando las diferencias de comportamiento en función de las restricciones en el control postural que se iban a utilizar en los experimentos entre el plano frontal y el sagital. La conclusión fue que la estabilidad de la cabeza en el plano frontal es más sensible a las restricciones en la postura y en la visión, es decir, se ve más afectada que en el plano sagital.

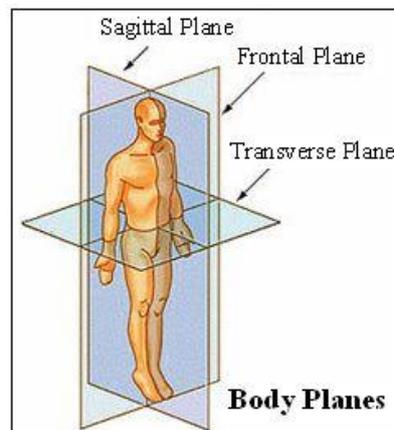


Figura 1.2: Planos del cuerpo humano

Se hizo un análisis adicional agrupando a los niños con parálisis por el tipo de trastorno del movimiento que presentaban, y se concluyó que el control troncal y la visión contribuyen de manera distinta en la estabilidad de la cabeza según el tipo de parálisis; lo que más tarde se comentará más detalladamente.

Soporte

Por su parte, sobre el efecto del apoyo los resultados indican que, en comparación con apoyo pélvico y sin apoyo, el apoyo del torso reducía todos los indicadores de balanceo (desplazamiento, ritmo y frecuencia) para todos los grupos y en ambos planos. Así se aprecia en la figura 1.3.

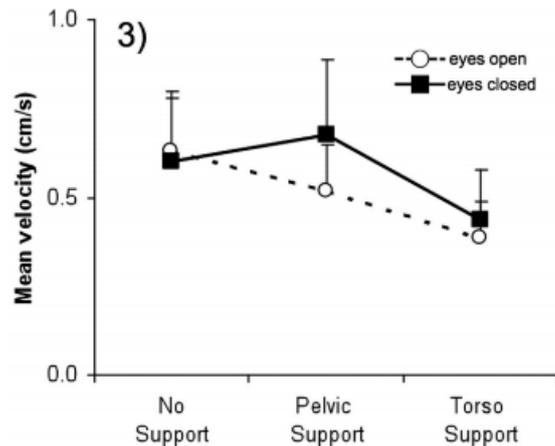


Figura 1.3: Gráfica de la velocidad media de balanceo frente al tipo de apoyo

Con soporte pélvico y ojos cerrados había un comportamiento similar de rigidez activa para adultos y niños mayores, pero no para niños más pequeños ni para ningún grupo con ojos abiertos. Esto indica que la visión era adecuada para estabilizar la cabeza en el plano frontal y que puede haber un desarrollo progresivo en la habilidad para usar la rigidez activa para controlar la estabilidad en el plano frontal; de donde se deduce que, en los niños más pequeños, no se ha desarrollado aún esta habilidad motora.

Para sin apoyo y con apoyo pélvico, los niños con parálisis presentaban más inestabilidad que el grupo de control de niños y adultos. Para soporte en el torso, se pensaba que no iba a haber diferencias, pero sí las había en el plano sagital, donde presentaban mayor amplitud y velocidad en el movimiento de cabeza. Esta diferencia entre el plano frontal y sagital sugiere que los mecanismos básicos de control de postura de la cabeza en el plano sagital pueden ser deficientes en personas con parálisis, y estas deficiencias se dan en los mecanismos centrales de control, no solo por desafíos biomecánicos o por control motor del tronco.

De lo anterior se deduce que los niños con parálisis cerebral tienen déficits en el control medio-lateral del tronco, lo que contribuye a un mayor balanceo.

Visión

Respecto al efecto de la visión, se observó que en el plano sagital ésta no afecta. Mientras que para el plano frontal dependía del apoyo y del grupo estudiado. Para los niños con desarrollo típico más pequeños y para los niños con parálisis atáxica la visión no afectaba para ningún tipo de apoyo. Resulta interesante que la falta de visión conducía a un menor movimiento en adultos sin apoyo y en niños mayores con desarrollo típico y apoyo del torso. Lo mismo se observa para niños con parálisis discinética, mientras que los de espasticidad tienen un incremento en el balanceo de la cabeza con ojos cerrados para todos los tipos de soporte.

El fenómeno de menor balanceo se atribuyó a la ausencia de la dificultad de mantener la vista fija, que se da cuando se tienen con los ojos abiertos.

En cuanto al incremento de movimiento en el caso de la parálisis espástica se adujo a la posibilidad de que hubiese déficits en el sistema de equilibrado postural o en el sistema vestibular para este tipo de parálisis.

Conclusiones

Todos estos resultados sugieren que los niños con parálisis cerebral pueden tener deficiencias en los mecanismos de estabilidad de la cabeza en el plano sagital y en el control del tronco, lo que también afecta a la estabilidad de la cabeza en el plano frontal. Además, la influencia de la visión sobre la estabilidad depende del tipo de parálisis.

Con esto, queda de manifiesto el papel fundamental de este proyecto a la hora de ejercitar los músculos cervicales mejorando esos mecanismos de control postural mencionados anteriormente y previniendo futuras complicaciones derivadas de una mala postura.

1.2. Parálisis cerebral

El sistema que se desarrolla en este proyecto se puede aplicar a cualquier persona en silla de ruedas que sufra hipotonía cervical severa, acompañada o no de trastorno de control motor del cuello, causada por cualquier patología (parálisis cerebral, ictus, traumatismo, enfermedad neurodegenerativa...). En este caso, se va a hablar de la parálisis cerebral, pues es con niños con esta patología con los que se van a realizar las pruebas del sistema.

La parálisis cerebral es un conjunto de trastornos que afecta a la capacidad de una persona para moverse, mantener el equilibrio y la postura, que no empeoran con el tiempo. (<https://medlineplus.gov/spanish/cerebralpalsy.html>)

- Etiología

La parálisis cerebral aparece durante los primeros años de vida hasta los cinco años como límite. Si bien la parálisis no es hereditaria, sí que puede existir una cierta predisposición genética lo que le da el nombre de parálisis cerebral congénita. Además existen otros factores de riesgo que pueden ocasionar su aparición, entre los que se encuentra: prematuridad, falta de peso al nacer, infecciones virales de la madre durante el embarazo, incompatibilidad sanguínea madre-bebé, falta de oxígeno prolongada durante el parto, infecciones del bebé...

Según el momento en que se produzca se habla de periodo prenatal, perinatal o postnatal.

En el periodo prenatal las causas se pueden deber a hipoxia (falta de oxígeno), rubéola durante el embarazo, exposición a rayos X o diabetes.

En el perinatal las principales causantes son desprendimiento de la placenta, también falta de oxígeno, o traumas.

Cuando se produce en el periodo postnatal, se habla de parálisis cerebral adquirida y sus causas son enfermedades infecciosas, meningitis, encefalitis, pero también accidentes automovilísticos, caídas graves o maltrato infantil.

- Tipología

Para clasificar la parálisis cerebral se emplean cuatro criterios: Tipo, tono, topografía y grado.

○ Tipo

- Espasticidad: Aumento exagerado del tono (hipertonía), acompañado de reflejos fuertes y descoordinados. Puede interferir con la actividad de caminar, el movimiento o el habla.
- Disquinesia: Presenta movimientos involuntarios especialmente cuando la persona intenta moverse. Según el tipo de movimiento a su vez puede ser:
 - Distonía: movimiento rápido, intermitente y espasmódico.
 - Atetosis: movimientos lentos e irregulares con fluctuación de hipertonía a hipotonía.
 - Corea: movimientos bruscos, arrítmicos, asimétricos y breves.
- Ataxia: Alteración del equilibrio caracterizada por la dificultad para andar y falta de motricidad fina (coordinación óculo-manual y óculo-pedal) y gruesa (habilidad para moverse armoniosa y equilibradamente).
- Mixto: En realidad, la parálisis cerebral suele presentar una combinación de los anteriores tipos.

○ Tono

Según el tono muscular puede ser:

- Isotónica: Tono normal.
- Hipotónica: Tono reducido.
- Hipertónica: Tono aumentado.
- Variable

○ Topografía

En función de la parte del cuerpo afectada se distingue:

- Hemiplejia: Parálisis que afecta a una de las dos mitades laterales del cuerpo.
- Diplejia: Parálisis que afecta a partes simétricas del cuerpo. La mitad inferior está más afectada que la superior.

- Cuadriplejia: Los cuatro miembros están paralizados.
- Paraplejia: Afecta a los miembros inferiores.
- Monoplejia: Afecta a un único miembro, sea superior o inferior.
- Triplejia: Se da cuando tres miembros se encuentran afectados.

- Grado

En cuanto al grado de afectación se diferencian tres niveles:

- Grave: Autonomía prácticamente nula.
- Moderado: Hay autonomía pero puede precisar la ayuda de un asistente.
- Leve: Presenta autonomía prácticamente total.

1.3. Sistemas de movilidad para gente con discapacidad

Existen diferentes soluciones de movilidad para gente con discapacidad en función de sus necesidades, hábitos o gustos.

- Silla de ruedas bimanual

Puede llevar dos motores, uno en cada rueda que facilita la propulsión. Cuenta con dos baterías y se maneja con un joystick. Tiene menor peso y autonomía que una silla de ruedas eléctrica convencional, pero por contra es más plegable y transportable.



Figura 1.4: Silla de ruedas bimanual con motores para propulsión

Las que no llevan motores necesitan de una persona que ejerza la fuerza y que puede ser bien la propia persona que va en la silla o bien un asistente.

Existe también la opción de incorporar un motor de ayuda al asistente, lo que disminuye el coste de la silla frente a una de ruedas de propulsión eléctrica, aunque necesita de la ayuda de dicho asistente.



Figura 1.5: Silla de ruedas convencional

- Moto eléctrica de tres o cuatro ruedas (scooter)

Es adecuada para gente que puede andar pero no tiene fuerza o resistencia adecuada para desplazarse según sus necesidades. Se controla con un manillar y dispone de reposabrazos. Es difícil de maniobrar en espacios reducidos, pero el precio es menor que el de una silla de ruedas eléctrica. Hay que tener en cuenta el caso de las enfermedades degenerativas, pues con el paso del tiempo sería complicado manejarla en el sentido del control postural, de la transferencia y conducción de la moto.



Figura 1.6: Scooter

- Segway adaptado

El segway está pensado para desplazarse de pie sobre él. Pero recientemente se han desarrollado adaptaciones para poder ir sentado, lo que lo convertiría en una versión moderna de las sillas de ruedas eléctricas. La dificultad está en que el usuario debe tener una buena movilidad del tronco para poder conducirlo inclinándose más o menos hacia delante, aunque los giros se realicen con un manillar. Pese a su cuidada estética y fácil movilidad incluso en espacios reducidos resulta bastante más caro que una silla de ruedas eléctrica.



Figura 1.7: Segway adaptado

- Handbike eléctrico

Es un sistema de ayuda que consiste en un manillar que se acopla en la parte delantera de la silla y que proporciona una mayor fuerza para sillas de ruedas manuales. El único inconveniente es la necesidad de desmontar el manillar a la hora de acercarse, por ejemplo, a una mesa, para lo que no todas las personas tienen la fuerza suficiente por sí solos.



Figura 1.8: Handbike eléctrico

- Sillas de ruedas deportivas

Están diseñadas para conseguir una eficiente transmisión de la fuerza propulsora. Se caracterizan por su maniobrabilidad y facilidad de giro. Son ligeras y robustas. Se usan por ejemplo en baloncesto en silla de ruedas o en paraatletismo para lo que cuentan con un sistema de pedaleo manual.



Figura 1.9: Silla de ruedas adaptada para jugar al baloncesto

1.4. Estado del arte

Para controlar la silla es necesario conocer las capacidades globales del usuario y la parte del cuerpo más adecuada para la conducción. Para evaluar este aspecto se toman en consideración distintos parámetros:

- Capacidad visual y perceptiva

Hay que tener en cuenta el campo y la agudeza visual para incluir más estímulos o adaptaciones si fuese necesario.

- Audición

En caso de algún defecto de audición es preciso valorar la necesidad de incluir espejos retrovisores o sustituir alarmas acústicas.

- Funciones cognitivas

Aunque pueden condicionar la conducción de la silla hay muchas personas con discapacidad intelectual que no presentan problemas de orientación espacial, con lo que la conducción de la silla se puede hacer de manera segura.

- Movimientos involuntarios

Hay que tenerlos en cuenta para inhibirlos con los controles posturales adecuados y que no sean usados para la controlar la conducción.

- Movimientos voluntarios

Serán los adecuados para dirigir la conducción de la silla. Es importante conocer la fuerza y la amplitud o recorrido de ese movimiento, pero también el grado de precisión con el que se realiza, así como la fatiga que experimenta el músculo tras un esfuerzo continuado, con el objetivo de poder ajustar y seleccionar lo mejor posible el tipo de mando de control y sus características.

- Sensibilidad superficial y profunda

Con el estudio de la sensibilidad se determina el nivel de tacto y de localización de las partes del cuerpo a la hora de realizar un movimiento.

En función de las condiciones previas se pueden distinguir tres tipos de control: con la cabeza, con la mano y con el pie.

Por la naturaleza del proyecto se va a trabajar sobre un control manual, por lo que será este el que se explique en primer lugar.

1.4.1. Control manual

Existen dos tipos de mandos en general: mandos proporcionales y no proporcionales. En los primeros se controla la dirección y la velocidad, de forma que cuanto más se desplace del punto central más rápido se moverá la silla. Por su parte, los no proporcionales solo tienen el control de la dirección, que únicamente puede ser delante, detrás, izquierda y derecha, mientras que velocidad y aceleración deben ser programadas. En ocasiones, se pueden crear varios perfiles determinados con la velocidad y aceleración que se elija.

Los joysticks tradicionales son mandos proporcionales de control manual. Tienen distintas formas y tamaños, pero en general siempre presentan: palanca de dirección y velocidad, indicador de carga de baterías, interruptor de encendido/apagado y controlador de velocidad. Se pueden colocar en distintas posiciones y algunos permiten modificar la postura, o incluyen interruptores para distintas funciones como luces o bocina, entre otras.



Figura 1.10: Joystick tradicional

El joystick tradicional consiste en una palanca acabada en forma de bola, cilindro o cono. Existen adaptadores en forma de T, bola o ventosa para facilitar su agarre.

Se coloca en un soporte abatible delante del reposabrazos, sobre un brazo articulado y abatible o sobre una bandeja igualmente abatible, siguiendo el criterio de más precisión y eficacia para el usuario.

Hay joysticks especiales entre los que cabe destacar:

- Joystick de dedo: se controla con un solo dedo.



Figura 1.11: Joystick de dedo

- Joysticks muy sensibles: pensados para personas que cuentan con precisión manual pero no así con fuerza. Éste además se puede controlar mediante distintas partes del cuerpo.



Figura 1.12: Joystick de alta sensibilidad

- Mini joystick sellado: que precisa de muy poca fuerza, apenas 10 gramos, y se puede controlar por barbilla, boca o mano.



Figura 1.13: Mini joystick sellado

- Mando controlado por un dedo: El dedo se introduce en el agujero y la silla se mueve hacia donde se desplaza éste.



Figura 1.14: Joystick de control por dedo

- Control por tableta táctil: Funciona como el Touchpad del ordenador, por donde el dedo se desplaza indicando de esta manera la dirección a seguir.

Existen otro tipo de joysticks que son no proporcionales:

- Joystick reforzado: tiene ocho direcciones (delante, detrás, izquierda, derecha y diagonales), funciona por microinterruptores.



Figura 1.15: Joystick no proporcional

- Mini joystick sensible no proporcional: Tiene cuatro direcciones definidas por microinterruptores. Puede controlarse por barbilla, boca o mano.



Figura 1.16: Mini joystick no proporcional

- Bandeja de pulsadores: Consta de cinco botones, cuatro de dirección y uno de encendido apagado. Se suele situar sobre una bandeja.



Figura 1.17: Bandeja de pulsadores

- Disco con pulsadores: Son cinco pulsadores mecánicos pequeños insertados en un disco, donde el botón central cambia el programa.

- Activación por un solo pulsador: Es necesaria una pantalla en el que las flechas de dirección se van iluminando secuencialmente, seleccionándose la dirección deseada cuando la flecha correspondiente está iluminada.

1.4.2. Control con la cabeza

Es necesario asegurar una buena posición para evitar daños cervicales al controlar la conducción con la cabeza.

Los controles proporcionales más habituales son:

- Controlado por mentón: cuentan con una palanca acabada en forma de bola o ventosa; además del interruptor de encendido.
- Control occipital: Parece un reposacabezas pero no permite el apoyo de la cabeza. Funciona según la fuerza que se haga con la cabeza, con la peculiaridad de que para ir hacia delante se presiona la cabeza hacia atrás, aunque para izquierda y derecha se presiona respectivamente a cada lado. Para conseguir la marcha atrás se requiere de programación y un pulsador adicional.
- Control de conducción Magitek: El sensor se coloca sobre la cabeza y el movimiento de ésta se traduce en desplazamientos de la silla. El inconveniente es que necesita un control preciso.

En los mandos no proporcionales se encuentran:

- Controlado por soplo/aspiración: necesitan muy poca fuerza de soplido para conseguir el desplazamiento. Consisten básicamente en una boquilla enganchada a un tubo sujeto al reposacabezas.



Figura 1.18: Ilustración de control por soplado

- Control de cabeza por sensores de proximidad: consiste en un reposacabezas que cuenta con tres sensores de proximidad (derecha, izquierda y posterior). De forma que inclinando la cabeza hacia atrás (sin necesidad de tocar el sensor) se va hacia delante, y hacia la izquierda o derecha según se incline la cabeza respectivamente. Se requiere un pulsador adicional para cambiar de dirección hacia atrás.



Figura 1.19: Sensores de proximidad en silla de ruedas

- Control por sopló/aspiración: puede presentar distintas configuraciones, por ejemplo: sopló fuerte es hacia delante, aspiración fuerte hacia atrás, sopló suave derecha y aspiración suave izquierda. Aunque esto puede variar según el modelo y el caso.
- Conducción por pulsadores de libre colocación: Es un sistema reposacabezas para soportar sensores bien de contacto, proximidad o presión. No es adecuado para gente hipertónica.



Figura 1.20: Ilustración de pulsadores de libre de colocación

Para estos tipos de mandos es preciso contar con unos adecuados soportes que suelen ir sujetos al reposacabezas, aunque existen también arneses pectorales para el mentón.

1.4.3. Control con el pie

Gran parte de los mandos anteriores se pueden controlar con el pie, siempre y cuando estén en la posición correcta y el usuario tenga suficiente habilidad. Lo más adecuado es tener el pie en el reposapiés y que se mueva para controlar la dirección, ya que si no se tiene esa posibilidad de reposo el movimiento se vuelve más inseguro.

Asimismo, existen pedales con los que se puede controlar no solo la dirección sino también la velocidad.

1.4.4. VR2

El joystick que se va a usar en este proyecto es el VR2. Es un mando proporcional manual de uso extendido y con distintas formas de la palanca. Tiene una tensión de entrada de 24 V y una tensión de salida de entre 16 y 28 V. La corriente durante la conducción es de 60, 70 o 90 A. Tiene una fuerza media nominal de 1,3 N a 10° de deflexión, siendo el ángulo mecánico de deflexión nominal de 18° desde el centro hacia delante/detrás e izquierda/derecha. Con lo que se ve que el par necesario para mover la palanca es bastante bajo.

Se le pueden añadir distintos módulos en función de las necesidades o gustos, como módulos de luz, de potencia o de actuadores.



Figura 1.21: Joystick VR2 utilizado en el proyecto

2. OBJETIVOS

El objetivo de este proyecto se puede resumir en diseñar e implementar un adaptador mecánico para el joystick de control de sillas de ruedas motorizadas que será controlado mediante un dispositivo electrónico a través de Bluetooth. Las señales de Bluetooth proceden de un sensor ya existente colocado en la cabeza del usuario y que da información sobre el movimiento cervical.

La finalidad de este sistema es fomentar el uso de los músculos cervicales mediante el movimiento de la silla para prevenir la hipotonía cervical que puede degenerar en problemas de salud más graves.

Asimismo, se busca que el sistema sea fácil de usar y económico, para hacerlo accesible a cualquier persona que pudiera necesitarlo.

Por tanto, el objetivo del presente trabajo de fin de grado trata de desarrollar y construir un dispositivo que:

- Sea capaz de recibir señales de Bluetooth procedentes de un sensor.
- Permita accionar unos motores en función de esas señales, para controlar el joystick de la silla de ruedas por medio de un dispositivo mecánico.
- Sea reconfigurable para poderlo adaptar a distintos niveles de movilidad del cuello.
- Implice al usuario como parte activa del sistema de rehabilitación mediante el movimiento de la silla de ruedas, viéndose esto como una actividad más lúdica que sanitaria.
- Sea portátil para poder llevarlo sobre la silla de ruedas.
- Tenga un coste lo más reducido posible.
- Pueda ser utilizado por personal no técnico para su configuración.
- Cumpla las especificaciones de seguridad expuestas en el real decreto 1591/2009, del 16 de octubre por el que se regulan los productos sanitarios, incluido en el Boletín Oficial del Estado número 268, del 6 de noviembre de 2009.

Para lograr estos objetivos se emplearán las herramientas, tanto a nivel de software como de material necesarias, que garanticen el cumplimiento de los requisitos planteados.

3. METODOLOGÍA

Para desarrollar cualquier proyecto es importante hacer una adecuada planificación del mismo detallando en cada etapa las tareas que son precisas abordar.

Por esa razón, se van a describir las etapas que conciernen a este proyecto:

- **Análisis de requisitos:** En primer lugar, es necesario analizar las circunstancias que deben cumplirse para lograr el objetivo principal. En este proyecto, el análisis se centra en el ámbito de la salud, social y económico.
- **Diseño:** Esta etapa se caracteriza por ser donde se deciden los materiales necesarios y la estructura del conjunto según los objetivos planteados.
- **Desarrollo:** En esta etapa se lleva a cabo el aprendizaje del modo de utilización de los distintos elementos y programas, así como la construcción del sistema. Comprende también la elaboración del programa de control del dispositivo.
- **Integración e implementación:** En esta fase se integra todo el sistema y se realizan las pruebas correspondientes para conocer el comportamiento del conjunto atendiendo a posibles errores para su corrección.
- **Evaluación y conclusiones:** Por último, se analizan los resultados de la implementación para comprobar si se han cumplido los requisitos, se elaboran las conclusiones y se detallan los aspectos a mejorar.

Si bien este proyecto abarca todas las etapas, desde el análisis hasta la evaluación, las dos últimas etapas no se tratan tan en profundidad como las precedentes, pero sí se realiza una pequeña implementación con sujetos de control para corroborar que se cumple con los objetivos establecidos.

Para una validación definitiva sería necesario contar con un experto que certificase el resultado de la aplicación del sistema en las personas a las que va destinado.

3.1. Análisis de requisitos

Según lo descrito anteriormente es necesario, en primer lugar, hacer un análisis de las necesidades y condiciones que debe satisfacer el proyecto.

En este caso, la necesidad fue detectada por la confederación ASPACE y la empresa Werium Assistive Solutions cuando trabajando con niños con parálisis cerebral en silla de ruedas se dieron cuenta de que muchos de ellos tenían también parálisis en las extremidades superiores, lo que les impedía controlar la silla de ruedas. Sin embargo, observaron que sí tenían movilidad en el cuello, y además el ejercitarlo resulta muy beneficioso para su salud, como está explicado en el apartado 1.1 de la Introducción.

También es imprescindible conocer el marco legal que aplica al proyecto para que pueda ser utilizado en el ámbito médico convenientemente.

3.1.1. Marco legal

Puesto que el dispositivo se va a utilizar con fines médicos la normativa que afecta al mismo se recoge en el Real Decreto 1591/2009, donde se define 'producto sanitario' como sigue:

"Cualquier instrumento, dispositivo, equipo, programa informático, material u otro artículo, utilizado solo o en combinación, incluidos los programas informáticos destinados por su fabricante a finalidades específicas de diagnóstico y/o terapia y que intervengan en su buen funcionamiento, destinado por el fabricante a ser utilizado en seres humanos con fines de:

- 1º) Diagnóstico, prevención, control, tratamiento o alivio de una enfermedad,
- 2º) diagnóstico, control, tratamiento, alivio o compensación de una lesión o de una deficiencia,
- 3º) investigación, sustitución o modificación de la anatomía o de un proceso fisiológico,
- 4º) regulación de la concepción,

y que no ejerza la acción principal que se desee obtener en el interior o en la superficie del cuerpo humano por medios farmacológicos, inmunológicos, ni metabólicos, pero a cuya función puedan contribuir tales medios."

En este caso el dispositivo se corresponde con la descripción de los puntos 1 y 2, ya que previene complicaciones futuras a causa de la hipotonía cervical y sirve como tratamiento para fortalecer esos músculos del cuello compensando la falta de tono a partir de su uso.

Ahondando en la cuestión, en el Anexo IX del Real Decreto se especifica en el punto 1.4 que se considera 'producto sanitario activo' como: "Cualquier producto sanitario cuyo funcionamiento dependa de una fuente de energía eléctrica o de cualquier fuente de energía distinta de la generada directamente por el cuerpo humano o por la gravedad, y que actúe mediante conversión de dicha energía. No se considerarán productos sanitarios activos los productos sanitarios destinados a transmitir, sin ninguna modificación significativa, energía, sustancias u otros elementos de un producto sanitario activo al paciente. Los programas informáticos autónomos se consideran un producto sanitario activo." Por esa descripción, el dispositivo que se desarrolla en este proyecto se puede considerar como producto sanitario activo.

Asimismo, en el artículo 1.5 de dicho Anexo se describe como ‘producto activo terapéutico’ a “Cualquier producto sanitario activo utilizado solo o en combinación con otros productos sanitarios, destinado a sostener, modificar, sustituir o restaurar funciones o estructuras biológicas en el contexto del tratamiento o alivio de una enfermedad, lesión o deficiencia.” Y tanto en cuanto el dispositivo se encarga de restaurar la capacidad motora del movimiento cervical así como de soporte, el dispositivo se engloba también en la categoría de producto activo terapéutico.

En el apartado III del Anexo, la clasificación, se recoge en el punto 1 lo que se entiende por ‘producto no invasivo’ siguiendo unas reglas, de las cuales se van a eliminar las que no atañen al presente caso, relativas a productos quirúrgicos:

“Regla 1.- Todos los productos no invasivos se incluirán en la clase I, salvo que sea aplicable alguna de las reglas siguientes.”

“Regla 2.- Todos los productos no invasivos destinados a la conducción o almacenamiento de sangre, fluidos o tejidos corporales, líquidos o gases destinados a una perfusión, administración o introducción en el cuerpo entrarán en la clase IIa:

Si pueden conectarse a un producto sanitario activo de la clase IIa o de una clase superior.

Si están destinados a ser utilizados para el almacenamiento o canalización de sangre u otros fluidos o para el almacenamiento de órganos, partes de órganos o tejidos corporales.

En todos los demás casos se incluirán en la clase I.”

“Regla 3.- Todos los productos no invasivos destinados a modificar la composición biológica o química de la sangre, de otros fluidos corporales o de otros líquidos destinados a introducirse en el cuerpo se incluirán en la clase IIb, salvo si el tratamiento consiste en filtración, centrifugación o intercambios de gases o de calor, en cuyo caso se incluirán en la clase IIa.”

“Regla 4.- Todos los productos no invasivos que entren en contacto con la piel lesionada:

Se clasificarán en la clase I si están destinados a ser utilizados como barrera mecánica, para la compresión o para la absorción de exudados.

Se clasificarán en la clase IIb si se destinan principalmente a utilizarse con heridas que hayan producido una ruptura de la dermis y solo puedan cicatrizar por segunda intención.

Se incluirán en la clase IIa en todos los demás casos, incluidos los productos destinados principalmente a actuar en el microentorno de una herida.”

“Regla 9.- Todos los productos terapéuticos activos destinados a administrar o intercambiar energía se incluirán en la clase IIa salvo si sus características son tales que puedan administrar energía al cuerpo humano o intercambiarla con el mismo de forma potencialmente peligrosa, teniendo en cuenta la naturaleza, la densidad y el punto de aplicación de la energía, en cuyo caso se incluirán en la clase IIb.

Todos los productos activos destinados a controlar el funcionamiento de los productos terapéuticos activos de la clase IIb o destinados a influir directamente en el funcionamiento de dichos productos se incluirán en la clase IIb.”

“Regla 10.- Todos los productos activos con fines de diagnóstico se incluirán en la clase IIa:

Si se destinan a suministrar energía que vaya a ser absorbida por el cuerpo humano, excluidos los productos cuya función sea la iluminación del organismo del paciente en el espectro visible.

Si se destinan a crear una imagen de la distribución in vivo de fármacos radiactivos.

Si se destinan a permitir un diagnóstico directo o la vigilancia de procesos fisiológicos vitales a no ser que se destinen específicamente a la vigilancia de parámetros fisiológicos vitales, cuando las variaciones de esos parámetros, por ejemplo, las variaciones en el funcionamiento cardíaco, la respiración, la actividad del SNC, puedan suponer un peligro inmediato para la vida del paciente, en cuyo caso se incluirán en la clase IIb.

Los productos activos destinados a emitir radiaciones ionizantes y que se destinen a la radiología con fines diagnósticos y terapéuticos, incluidos los productos para controlar o vigilar dichos productos, o que influyan directamente en el funcionamiento de los mismos, se incluirán en la clase IIb.”

“Regla 11.- Todos los productos activos destinados a administrar medicamentos, líquidos corporales u otras sustancias al organismo, o a extraerlos del mismo, se incluirán en la clase IIa, a no ser que ello se efectúe de forma potencialmente peligrosa, teniendo en cuenta la naturaleza de las sustancias, la parte del cuerpo de que se trate y el modo de aplicación, en cuyo caso se incluirán en la clase IIb.”

“Regla 12.- Todos los demás productos activos se incluirán en la clase I.”

Por todo lo anterior, se llega a la conclusión de que el dispositivo desarrollado en este trabajo es un producto no invasivo de la clase I. El programa desarrollado pertenece a

la misma clase que el producto según se explica en el apartado 2.3 del mencionado Anexo: “Los soportes informáticos que sirvan para manejar un producto o que tengan influencia en su utilización se incluirán automáticamente en la misma categoría”.

Por otro lado, en el Anexo I se explicitan los requisitos esenciales. En el apartado I de requisitos generales se detallan distintos aspectos que es necesario tener en cuenta a la hora de llevar a cabo el proyecto:

1. Los productos deberán diseñarse y fabricarse de forma tal que su utilización no comprometa el estado clínico o la seguridad de los pacientes ni la seguridad y la salud de los usuarios, y en su caso, de otras personas cuando se utilizan en las condiciones y con las finalidades previstas. Los posibles riesgos asociados a la finalidad prevista deberán ser aceptables en relación con el beneficio que proporcionen al paciente y compatibles con un nivel elevado de protección de la salud y de la seguridad.

Esto implicará:

la reducción, dentro de lo posible, del riesgo derivado de errores de utilización debidos a las características ergonómicas del producto y al entorno en el que está previsto utilizar el producto (diseño que tenga en cuenta la seguridad del paciente), y

tener en cuenta los conocimientos técnicos, la experiencia, la formación, el adiestramiento y, en su caso, las condiciones médicas y físicas de los usuarios previstos (diseño para usuarios no profesionales, profesionales, con discapacidad u otros).

2. Las soluciones adoptadas por el fabricante en el diseño y la construcción de los productos deberán ajustarse a los principios de integración de la seguridad teniendo en cuenta el estado generalmente reconocido de la técnica.

Al seleccionar las soluciones más adecuadas el fabricante aplicará los siguientes principios, en el orden que se indica:

Eliminar o reducir los riesgos en la medida de lo posible (seguridad inherente al diseño y a la fabricación).

Adoptar las oportunas medidas de protección incluso alarmas, en caso de que fuesen necesarias, frente a los riesgos que no puedan eliminarse.

Informar a los usuarios de los riesgos residuales debidos a la incompleta eficacia de las medidas de protección adoptadas.

3. Los productos deberán ofrecer las prestaciones que les haya atribuido el fabricante y estarán diseñados, fabricados y acondicionados de forma que puedan desempeñar una o varias de las funciones contempladas en las letras a) o b), del apartado 1, del artículo 2, y tal y como el fabricante las haya especificado.

4. Las características y prestaciones referidas en los apartados 1, 2 y 3 no deberán alterarse en un grado tal que se vean comprometidos el estado clínico y la seguridad de los pacientes ni en su caso, de terceros, mientras dure el período de validez previsto por el fabricante, cuando el producto se vea sometido a las situaciones que puedan derivarse de las condiciones normales de utilización.

5. Los productos deberán diseñarse, fabricarse y acondicionarse de forma tal que sus características y prestaciones, según su utilización prevista, no se vean alteradas durante el almacenamiento y transporte, teniendo en cuenta las instrucciones y datos facilitados por el fabricante.

6. Cualquier efecto secundario no deseado deberá constituir un riesgo aceptable en relación con las prestaciones atribuidas.

Lo que se puede resumir en los puntos más destacados en seguridad, limitación, reducción y aviso de riesgos y garantías de uso y fabricación.

De los puntos que siguen solo cabe mencionar relativo a este trabajo el apartado 12: "Requisitos para los productos sanitarios conectados a una fuente de energía o equipados con una fuente de energía.", puesto que el resto o no aplican o se pueden incluir en el resumen del párrafo anterior. De este punto, caben destacar los siguientes apartados, puesto que el dispositivo precisa de una fuente de energía:

12.1-"Los productos que lleven incorporados sistemas electrónicos programables deberán diseñarse de forma que se garanticen la repetibilidad, fiabilidad y eficacia de dichos sistemas, en consonancia con la utilización a que estén destinados. En caso de condiciones de primer defecto en el sistema, deberán preverse los medios para poder eliminar o reducir lo más posible los riesgos consiguientes."

12.1bis-"Para los productos que incorporen programas informáticos o que sean programas informáticos médicos, dichos programas informáticos deben ser validados con arreglo al estado actual de la técnica, teniendo en cuenta los principios de desarrollo del ciclo de vida, gestión de los riesgos, validación y verificación."

12.3-"Los productos conectados a una fuente de energía externa de la que dependa la seguridad de los pacientes deberán incluir un sistema de alarma que señale cualquier fallo de la fuente de energía."

12.6 Protección contra los riesgos eléctricos. –Los productos deberán diseñarse y fabricarse de forma que, cuando estén correctamente instalados y se utilicen normalmente o en condiciones de primer defecto, se eviten en la medida de lo posible los riesgos de choque eléctrico accidental.

12.7.1 Los productos deberán diseñarse y fabricarse de forma que el paciente y el usuario estén protegidos de los riesgos mecánicos relacionados, por ejemplo, con la resistencia, la estabilidad y las piezas móviles.

12.7.5 Las partes accesibles de los productos (con exclusión de las partes o de las zonas destinadas a proporcionar calor o a alcanzar determinadas temperaturas) y su entorno no deberán alcanzar temperaturas que puedan representar un peligro en condiciones de utilización normal.

Todos estos requisitos han de tenerse en cuenta en todas las fases de desarrollo del proyecto.

3.2. Diseño

En esta etapa se definen los objetivos a cumplir, siguiendo los requisitos anteriores, y las herramientas para conseguirlos.

Por tanto, abarca desde la estructura física, hasta la elección de los componentes necesarios y la programación del sistema.

3.2.1. Diseño del dispositivo

En lo que sigue, se va a explicar la elección de componentes para el dispositivo así como sus características.

La información recogida en la Introducción refleja que la forma más adecuada para mover la silla de ruedas y al tiempo ejercitar los músculos cervicales es mediante el control con la cabeza. Sin embargo, aunque existen diferentes diseños en el mercado

para el control con la cabeza, ninguno de estos es ajustable a los distintos niveles de movimiento con los que se pretende efectuar el ejercicio en función de las características de cada persona.

Esto justifica el hecho de contar con un sensor que recoja los movimientos de la cabeza. Para utilizar esos datos en el movimiento de la silla es menester contar con un dispositivo de control, que en este caso es una placa Arduino Uno, que cuenta con los pines de entrada y salida suficientes para lo que se va a utilizar.

Para controlar el movimiento de la silla debe haber algún elemento que mueva el joystick. Esa función la desempeñan dos motores paso a paso a partir de la información que le llega a la placa de control. Esos dos motores están situados de manera perpendicular uno respecto al otro consiguiendo el movimiento deseado en dos dimensiones.

Con el fin de transmitir ese movimiento se cuenta con una estructura con distintos niveles que se encaja sobre el joystick y se ajusta al soporte del mismo.

Por otro lado, la comunicación sensor-placa es vía Bluetooth. Para llevarla a cabo, hay que utilizar un conector Bluetooth conectado al Arduino, ya que no lleva integrado ningún elemento específico para ello.

Respecto a la alimentación, el sistema debe tener su propia fuente para que la silla de ruedas pueda desplazarse con libertad sin necesidad de contar con un cable que alimente la placa de control desde un ordenador. Para este fin se ha optado por una batería portátil recargable que se acople en la silla de ruedas.

En resumen, los elementos que se necesitan para el dispositivo son:

- Sensor.
- Placa Arduino Uno.
- Motores paso a paso.
- Driver.
- Conector Bluetooth.
- Fuente de alimentación portátil recargable.
- Estructura.

Una vez identificados los componentes necesarios se va a detallar el motivo de elección de cada uno y sus características principales.

3.2.1.1. Componentes

PLACA ARDUINO UNO

El propósito de este proyecto es hacer un dispositivo funcional de bajo coste, con lo que uno de los requisitos que más hay que tener en cuenta a la hora de elegir el dispositivo de control es el precio. Pero no solo ese, sino también la eficiencia y la adecuación al proyecto.

Estas características las reúne el Arduino Uno, que con un precio bastante asequible permite llevar a cabo una gran variedad de proyectos y tiene unas prestaciones suficientes para este trabajo.

El Arduino es un microcontrolador de placa simple con software y hardware liberados que permiten la libertad de acceso a cualquier usuario.

Nació en Estados Unidos en el año 2005 como una placa de desarrollo de bajo coste, que permite una gran variedad de entradas y salidas para conectar sensores, actuadores... de manera sencilla.

Existen en torno a veinte tipos distintos de Arduino, desde el Mega al Leonardo, pasando por el Uno, el Nano, Arduino Robot, Arduino Esplora, el Diecimila o el Micro. Todas estas se pueden clasificar en tres grandes grupos: clásicas (Mega, Uno, Leonardo), aplicadas al control remoto y robótica (Esplora y Robot) y las más nuevas enfocadas a dispositivos que se puedan incorporar en la ropa (LilyPad, Gemma). Asimismo, se les pueden añadir o conectar otras placas (*shields*) que les dan funcionalidades adicionales como wifi o Ethernet.

Para elegir el modelo, ahora los criterios a tener en cuenta ahora varían ligeramente respecto a los anteriores, aunque el precio sigue teniendo gran parte del peso. Otro aspecto es la posibilidad de usar pines para PWM (*Pulse Width Modulation*) con los que controlar los motores y que las salidas puedan proporcionar el par necesario para ello. Mas esto último no es gran inconveniente, ya que el joystick se mueve con una fuerza nominal de 1.3 N (que resulta muy asequible para cualquier motor controlado desde un Arduino), como se explica detalladamente en el punto 1.4.4 de la introducción. Un tercer criterio son las interrupciones, que pueden ser necesarias para el control de los motores.

En primer lugar, y atendiendo al primer criterio (el precio), se comprobó si el Arduino de más bajo precio cumplía con el resto de las prestaciones requeridas.

El Arduino Uno tiene 6 pines para PWM y 2 pines para interrupciones. Esto es más que suficiente para esta aplicación, que solo necesita de dos motores, y además su tamaño

reducido 8 x 5,5 x 2,5 cm lo hace apropiado para poderlo transportar en la silla de ruedas.

El Arduino Uno cuenta con la siguiente distribución de pines y conectores:

- Entradas/salidas digitales: Van del 0 al 13. La señal digital puede estar encendida o apagada. Los pines 2 y 3 son los que permiten las interrupciones.
- Salidas analógicas: Son los pines 3, 5, 6, 9, 10 y 11. Son los pines del PWM.
- Entradas analógicas: Pines A0, A1, A2, A3, A4 Y A5. Para señales procedentes de un sensor analógico. También se pueden utilizar como pines digitales.
- Pines de alimentación: GND (tierra), 5V (para dar 5V), 3.3V (este pin da 3.3V), Vin (voltaje de entrada), RESET, IOREF (para que la placa identifique la alimentación necesaria para las distintas shields).
- Pin AREF: que proporciona una tensión distinta a 5V por los pines digitales.
- Conector USB: sirve para alimentar la placa y cargar el programa.
- Conector de alimentación: sirve para alimentar a la placa.

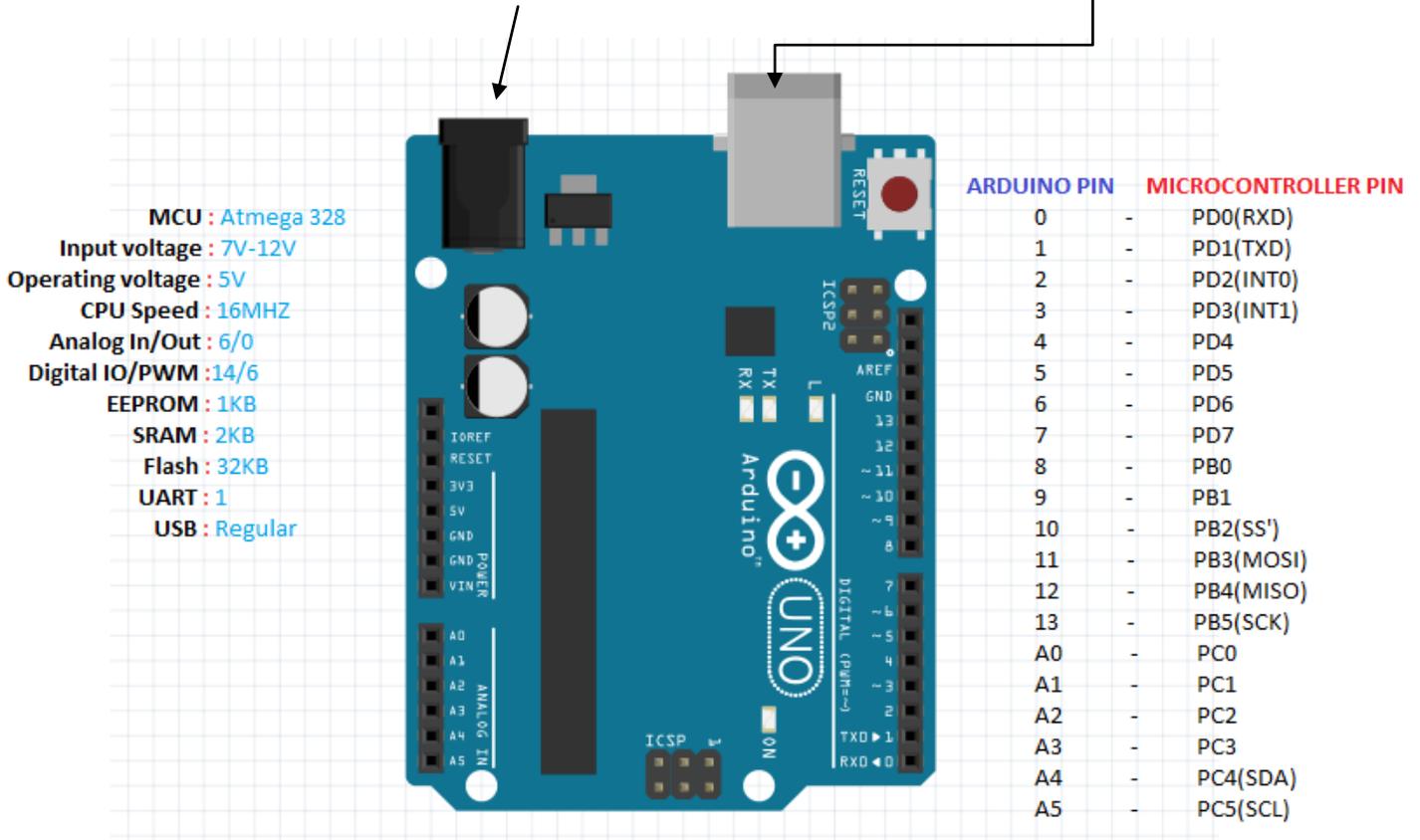


Figura 3.1: Esquema de pines del Arduino Uno



Figura 3.2: Imagen real del Arduino Uno

Microcontrolador	ATmega328
Voltaje de funcionamiento	5V
Alimentación	7-12 V
Voltaje máximo de entrada	20 V
Pines digitales E/S	14 (6 con salida PWM)
Pines de entrada analógica	6
Corriente DC por E/S Pin	40 mA
Corriente DC para el pin de 3.3V	50 mA
Memoria Flash	32 kB
SRAM	2 kB
EEPROM	1 kB
Velocidad del reloj	16 MHz

Tabla 3.1: Resumen de características Arduino Uno R3

MOTORES PASO A PASO

Un motor paso a paso es un dispositivo electromecánico que funciona transformando impulsos eléctricos en desplazamientos angulares, los llamados pasos. El paso es el ángulo que gira el motor en cada paso, de forma que: $\text{ángulo} = \frac{360^\circ}{\text{pasos}}$.

Los motores paso a paso se clasifican en unipolares y bipolares:

- Unipolares: Son los más sencillos de controlar ya que por sus bobinas la corriente siempre circula en el mismo sentido. Tienen un terminal común a la fuente de alimentación. Son de cuatro fases y tienen 5 o 6 cables en función de si tienen las cuatro bobinas un polo común o el común es separado para cada

una. Los motores unipolares de seis cables se pueden usar como bipolares dejando los comunes al aire.

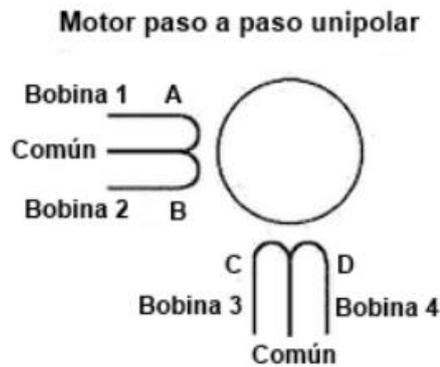


Figura 3.3: Esquema de motor paso a paso unipolar

A continuación se muestran las distintas secuencias de control de un motor paso a paso unipolar:

	Bobina 1	Bobina 2	Bobina 3	Bobina 4
Paso 1	1	0	0	0
Paso 2	0	1	0	0
Paso 3	0	0	1	0
Paso 4	0	0	0	1

Tabla 3.2: Secuencia simple o wave drive (solo una bobina activa)

	Bobina 1	Bobina 2	Bobina 3	Bobina 4
Paso 1	1	1	0	0
Paso 2	0	1	1	0
Paso 3	0	0	1	1
Paso 4	1	0	0	1

Tabla 3.3: Secuencia normal (la más habitual)

	Bobina 1	Bobina 2	Bobina 3	Bobina 4
Paso 1	1	0	0	0
Paso 2	1	1	0	0
Paso 3	0	1	0	0
Paso 4	0	1	1	0
Paso 5	0	0	1	0
Paso 6	0	0	1	1
Paso 7	0	0	0	1
Paso 8	1	0	0	1

Tabla 3.4: Secuencia de medio paso (giro más suave y preciso)

- Bipolares: Cuentan con cuatro cables de salida puesto que son de dos fases. Las bobinas están formadas por cuatro arrollamientos por los que la corriente va cambiando secuencialmente de sentido, lo que complica al mismo tiempo su control.

Motor paso a paso bipolar

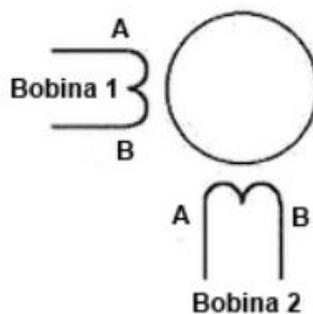


Figura 3.4: Esquema de motor paso a paso bipolar

La secuencia para controlar un motor paso a paso bipolar se recoge en la siguiente tabla:

	Bobina 1	Bobina 2	Bobina 3	Bobina 4
Paso 1	1	0	1	0
Paso 2	1	0	0	1
Paso 3	0	1	0	1
Paso 4	0	1	1	0

Tabla 3.5: Secuencia para excitar las bobinas de un motor paso a paso bipolar

Los motores bipolares proporcionan una mejor relación par-tamaño que los unipolares debido a que el par es proporcional a la intensidad de campo magnético generado por el devanado del estator. Cuantas más espiras o más corriente por cada fase, más se tienden a calentar los motores, y como la resistencia de los unipolares es el doble que la de los bipolares puesto que la sección es la mitad (suponiendo mismo tamaño para ambos), se calentarán más los unipolares, quedando en desventaja.

Siendo el precio un factor determinante para la elección del motor, el par que proporciona es también algo fundamental a tener en cuenta. Así pues, motor más adecuado que cumpliera con el requisito mínimo de fuerza para mover el joystick, que es de 1.3 N, según se consultó con los proveedores en su origen. El motor seleccionado fue un Nema 17, los mismos que se usan en las impresoras 3D y tienen una gran precisión.

La figura 3.5 muestra las características principales del tipo de motor escogido:



Description	Length	Mounted Rated Current	Mounted Holding Torque		Winding Ohms mH		Detent Torque		Rotor Inertia		Motor Weight	
			Nm Typ.	oz-in Typ.	±10% @ 20°C Typ.	mNm oz-in	g cm ² oz-in ²	kg	lbs			
(Stack)	*L* Max	Amps										
Single	39.8 mm (1.57 in)	2	0.48	68	1.04	2.2	15	2.1	57	0.31	0.28	0.62

Figura 3.5: Características del motor Nema 17

Como se puede comprobar, el par mayor que el par necesario para mover el joystick. Otro aspecto a destacar, y por el que se realizó esta elección es el paso angular, o sea, el ángulo por paso, que con un valor de 1.8° se considera suficiente para esta aplicación. Además, el motor incorpora una reductora interna de ratio de reducción 1/8.

DRIVER

El *driver* es un elemento indispensable para controlar motores paso a paso bipolares. Para su gobierno presenta un doble puente en H que permite variar la alimentación que se da a las dos bobinas con las que cuenta cada motor paso a paso. Su funcionamiento consiste en generar pulsos de distinta duración y valor, con los que se va calculando el valor medio de tensión de alimentación al motor, es lo que se conoce como señal *PWM (Pulse Width Modulation)* o señal de modulación de ancho de pulso. Esta señal es la que le llega a cada bobina según la secuencia establecida, como se puede observar en la tabla de la secuencia para controlar un motor bipolar.

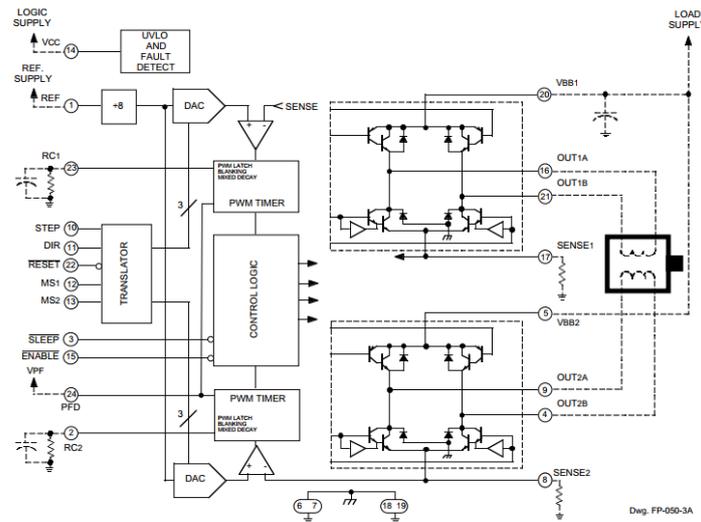


Figura 3.6: Esquema de los componentes del driver

En este caso se ha optado por el *driver* A3967. Consiste en un circuito integrado, con el chip A3967 del que recibe el nombre. El motivo de elección de este controlador es su sencillez para controlar los motores a pasos con pocas líneas de código para el *driver*. Su tamaño reducido y la posibilidad de usar varios a la vez (dos en este trabajo) conectados al Arduino lo hacen ideal para este proyecto. Además, ofrece la opción de realizar micropasos de hasta un octavo. Esto permite un movimiento más suave, mayor precisión y resolución y una resonancia más reducida.

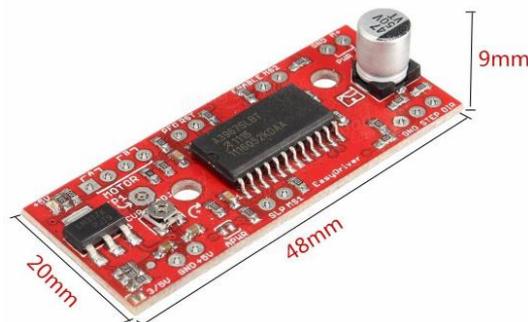


Figura 3.7: Imagen del EasyDriver A3967

Existen dos tipos de entradas al *driver*: las señales de control y las entradas de regulación.

Los elementos más destacados de las señales de regulación, dentro de las cuales se encuentran condensadores y resistencias, son:

- Condensador de entrada: Para evitar que se pierdan pasos, ya que en cada impulso que se le da al motor éste arranca y se detiene en tiempos muy cortos, pudiendo no contar alguno de los pasos indicados por el control.
- Resistencias en los terminales SENSE 1 y SENSE2: Para limitar la corriente que circula por las bobinas dado que cada motor tiene unas características distintas a las que el *driver* debe adaptarse por medio de estas resistencias.

De las señales de control para su manejo las más importantes son:

- DIR: Indica la dirección de rotación del motor.
- STEP: Es el puerto de entrada de los impulsos.
- GND: Es el pin de conexión a tierra.
- *Microstepping* MS1 y MS2: Son las señales que permiten la división del paso. Para operarlo en los distintos modos se configuran según la tabla 3.6:

MS1	MS2	Resolución
LOW	LOW	Paso completo
HIGH	LOW	Medio paso
LOW	HIGH	Cuarto de paso
HIGH	HIGH	Octavo de paso

Tabla 3.6: Modos de configuración para la división de los pasos

Por su parte, la tensión de entrada tiene un amplio rango, de 7 a 30 V, siendo mayor el par a altas velocidades cuanto mayor sea el voltaje. También tiene un modo *sleep* para poner el *driver* en estado de mínima energía mientras no se haga nada, y una función de *enable*, que permite activar o desactivar el *driver* a conveniencia.

CONECTOR BLUETOOTH

Dado que el Arduino no lleva función de Bluetooth es necesario utilizar un dispositivo ajeno que le proporcione esta función. La elección está condicionada por el mismo criterio principal que se usó para el resto de los componentes: algo que cumpla con los requerimientos de funcionamiento de manera suficiente con el mínimo precio. Así, el conector Bluetooth elegido es el HC-06, que es un módulo USB de Bluetooth versión 2.0, clase 2, de baja energía y mayor velocidad de transmisión de datos (EDR) que funciona a 2.4 GHz. Es un módulo ampliamente extendido para realizar proyectos con Arduino usando Bluetooth.

La versión de Bluetooth hace referencia a la generación del mismo, es decir, los avances y nuevos modelos que salen al mercado según progresa la tecnología.

La clase atiende a su potencia de transmisión. Existen tres clases: la clase 1, la 2 y la 3. La clase 1 presenta una potencia máxima permitida de 100 mW, que le permite tener un alcance aproximado de 100 metros. La clase 2 tiene una potencia máxima de 2,5 mW y un alcance de hasta 10 m. Finalmente, la clase 3, con una potencia máxima permitida de 1 mW tiene un alcance de no más de un metro. El módulo Bluetooth elegido, perteneciente a la clase 2 permite un alcance de hasta 10 metros.

En la siguiente tabla se recogen las características técnicas más importantes del conector:

Versión	Bluetooth 2.0 + EDR
Clase	2
Potencia de emisión	≤ 6 dBm
Tensión de alimentación máxima	3.6 a 6 V
Velocidad de datos	2 Mbps
Frecuencia de trabajo	2.4 GHz
Temperatura de trabajo permitida	-25 a 75°C
Dimensiones (Largo x Ancho)	40 mm x 17 mm
Consumo de corriente	30 a 40 mA
Antena	Antena PCB
Pines	VCC, GND, TXD, RXD

Tabla 3.7: Características del HC-06

Este módulo de Bluetooth actúa como esclavo, esto es, necesita de otro dispositivo con conexión Bluetooth que lo controle, que recibe el nombre de maestro, y solo puede estar conectado con ese. Por otro lado, un maestro puede conectarse con varios esclavos gestionando las transferencias de información.

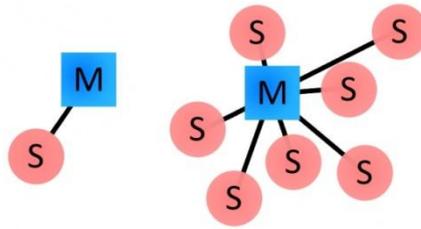


Figura 3.8: Representación esquemática de conexión maestro-esclavo

Sin embargo, para este proyecto no es necesario que el Bluetooth funcione como maestro que controle otros dispositivos, sino al revés; el módulo debe recibir señales del sensor, actuando como esclavo.

Existe un módulo, el HC-05 que puede funcionar como esclavo o maestro según se configure y permite utilizar más instrucciones que el HC-06. Aunque presenta el mismo precio que el HC-06 no se ha elegido porque se estarían desaprovechando funcionalidades del mismo y con el 06 es suficiente para esta aplicación.

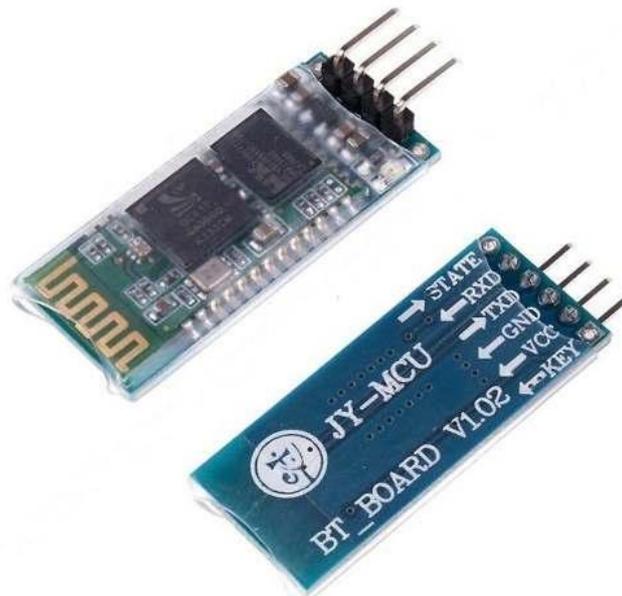


Figura 3.9: Imagen del módulo Bluetooth HC-06

ALIMENTACIÓN ARDUINO

Una fuente portátil recargable es un conjunto de pilas unidas en un paquete que puede ser recargado sucesivamente un elevado número de veces garantizando un suministro de corriente estable en el tiempo. Para ello cuenta con un terminal que permite conectarlo tanto a dispositivos directamente o a cargadores, si bien en ocasiones es necesario utilizar un adaptador.

La fuente de alimentación se decidió que fuese portátil para dotar a la silla de ruedas de libertad de movimiento al no tener que permanecer la placa de control enchufada a ninguna toma de corriente. Esa autonomía implica la necesidad de que la fuente pueda ser transportada en la misma silla de ruedas y garantice un tiempo adecuado de funcionamiento del sistema, por lo que no solo se buscaba un precio comedido sino también unas características, de tamaño y duración de la batería, apropiadas.

Con todo esto y tras mirar distintos catálogos y fuentes se determinó emplear una batería de ion litio de 7.4V con una capacidad nominal de 2.25 Ah, o lo que es lo mismo, 2250 mAh.

Las características más destacadas de la misma son, aparte de las ya nombradas:

- Temperatura de trabajo de 0 a 45°C en carga y de -20 a 60 en descarga, es decir, tiene un buen rango de temperaturas de funcionamiento.
- Peso: 100 gramos, lo cual no supone ningún inconveniente para poder ser transportado.
- Dimensiones: Altura: 18.6 mm; Anchura: 37.3 mm; Longitud: 70 mm. Con lo que es un tamaño adecuado para poder ser incorporado en la silla de ruedas.
- Ciclo de vida: Aguanta al menos 300 ciclos de carga garantizando una capacidad del 80% de la inicial.
- Almacenamiento: Se garantiza el 80% de la capacidad para un almacenamiento de 28 días a 25°C, y del 85 % para un almacenamiento de 7 días a 60°C.



Figura 3.10: Imagen de la batería portátil recargable

Otras características se recogen en la siguiente tabla:

Nominal voltage	7.4V
Nominal capacity	2250mAh (0.2C discharged from 4.2V to 2.75V each cell)
Minimum capacity	2150mAh (0.2C discharged from 4.2V to 2.75V each cell)
Charge voltage	4.20±0.05V<each cell>
Standard charge current	0.5C(1100mA)
Max charge current	1C (2200mAh, 10°C≥T≥0°C)
Charge conditions	CC / CV
Max continuous discharge current	1C (2200mA)
Max peak discharge current	1C (2200mA)
Discharge cut-off voltage	3.0V±0.075V<each cell>
Thermister	10K NTC((±5%)
Internal Impedance	≤200mohm (after charged)
Working temperature	Charge: 0 ~ +45°C
	Discharge: -20 ~ +60°C
Storage temperature	-20 ~ +45°C<recommend -0~35°C>
Relative humidity	65±20%
Weight	Approx 100g <for battery pack>
Battery pack dimension	Thickness: 18.6mm<±0.3mm>
	Width : 37.3mm<±0.5mm >
	Length : 70.0mm<±0.5mm >

Tabla 3.8: Características de la batería recargable

ALIMENTACIÓN MOTORES

Aunque en para probar los motores y el programa de Arduino se usó una fuente enchufable de 12 V y 1 A, había que decidir una forma de alimentar los motores que pudiese ser transportada con la silla de ruedas del mismo modo que sucedía con la alimentación del Arduino.

Tras sopesar entre baterías y pilas recargables se optó por la segunda opción por ser más económica, más ligera y al usar portapilas el peso podía distribuirse de mejor manera.

Los motores se alimentan a 12 V, por lo que eran necesarias 10 pilas de 1,2 V (que es la tensión habitual de las pilas recargables) y tres portapilas de cuatro unidades que se conectarían en serie para obtener el voltaje y corriente deseados.

Las pilas escogidas no requieren un cargador especial. Son pilas de NiMH, 1.2 V AA y 2900 mAh; esto es, en una aplicación que consuma un amperio tardan en descargarse 2,9 horas, es decir, dos horas y cincuenta y cuatro minutos; si consumiese dos amperios tardarían 1,45 horas en descargarse. Este valor es de los más altos del mercado, pues habitualmente tienen 2500 o 2700 mAh. Y en este caso ese incremento de capacidad no supuso un incremento de precio, pues en otra tienda vendían paquetes de cuatro pilas de 2500 mAh a mayor precio que las mismas unidades de 2900 en la tienda de electrónica elegida.



Figura 3.11: Imagen de las pilas elegidas

En el envoltorio de las pilas DigiVolt+ aparece indicado que se asegura un 80% de la carga después de un año aunque no hayan sido usadas, dando muestras de su gran capacidad gracias a la nueva tecnología que emplean en su fabricación. También se indica que pueden ser recargadas hasta mil veces exponiendo así su larga vida útil.

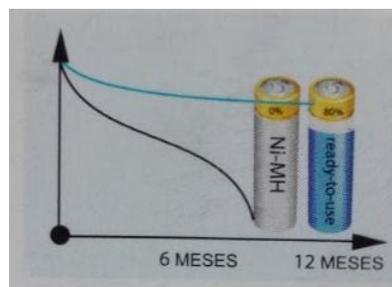


Figura 3.12: Gráfica comparativa de la duración de carga de las pilas tras un año sin usar: pilas elegidas (azul) frente a otras del mismo tipo (negro)

3.2.1.2. Montaje

El montaje se puede dividir en dos partes bien diferenciadas, por un lado el montaje eléctrico de las conexiones entre los *drivers*, el Bluetooth, el Arduino y los motores, y por otro lado el montaje de la estructura.

3.2.1.2.1. Montaje del circuito

El montaje del circuito se llevó a cabo usando una *protoboard* o placa de prototipado.

En ella se insertaron los dos *drivers* necesarios para el funcionamiento de los motores. Para conectarlos era necesario utilizar unos pinchos conectores que atravesaban los agujeros de los *drivers* y se clavaban en la *protoboard* creando el camino de conducción. Sin embargo, había problemas en los contactos ya que no todos los pines recibían alimentación, es decir, no todos estaban bien conectados. Por ello, hubo que soldar los pinchos conectores a los *drivers* para conseguir un contacto efectivo, pudiéndose de esta forma conectar con éxito a la *protoboard*.

Los *drivers* cuentan con cuatro pines para la conexión de cada motor paso a paso, correspondiendo los dos primeros a una bobina y los otros dos a la otra. Es importante no desconectar los motores mientras los *drivers* están conectados ya que en caso contrario pueden sufrir daños internos en el chip volviéndolos inservibles.

La alimentación la reciben por otros dos pines conectados a la línea de alimentación de la *protoboard*. A ella llegaban en las pruebas preliminares una tensión de 12 V y una corriente de 1 A procedentes de una fuente de alimentación conectada a la corriente y que mediante un conector hembra con dos cables (positivo y negativo) se conectan con la *protoboard*. Para el diseño final se optó por un conjunto de pilas recargables puestas en serie y conectadas a la placa del mismo modo.

El Arduino se conecta a cada uno de los *drivers* por dos pines, uno correspondiente al STEP y otro correspondiente al DIR. Hay un tercer pin de conexión común que es la tierra (GND); de forma que los *drivers* se conectan a la misma línea de la *protoboard*, a donde va la tierra del Arduino.

En la Figura 3.13 se aprecia la disposición de las conexiones reales para la parte de los motores.

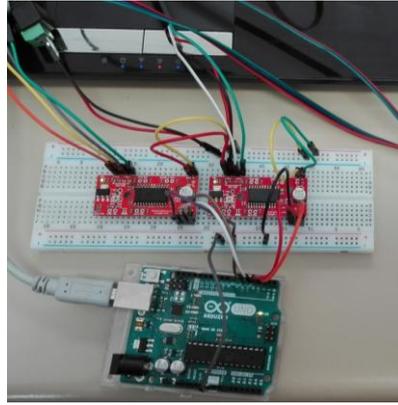


Figura 3.13: Sistema conectado a la protoboard

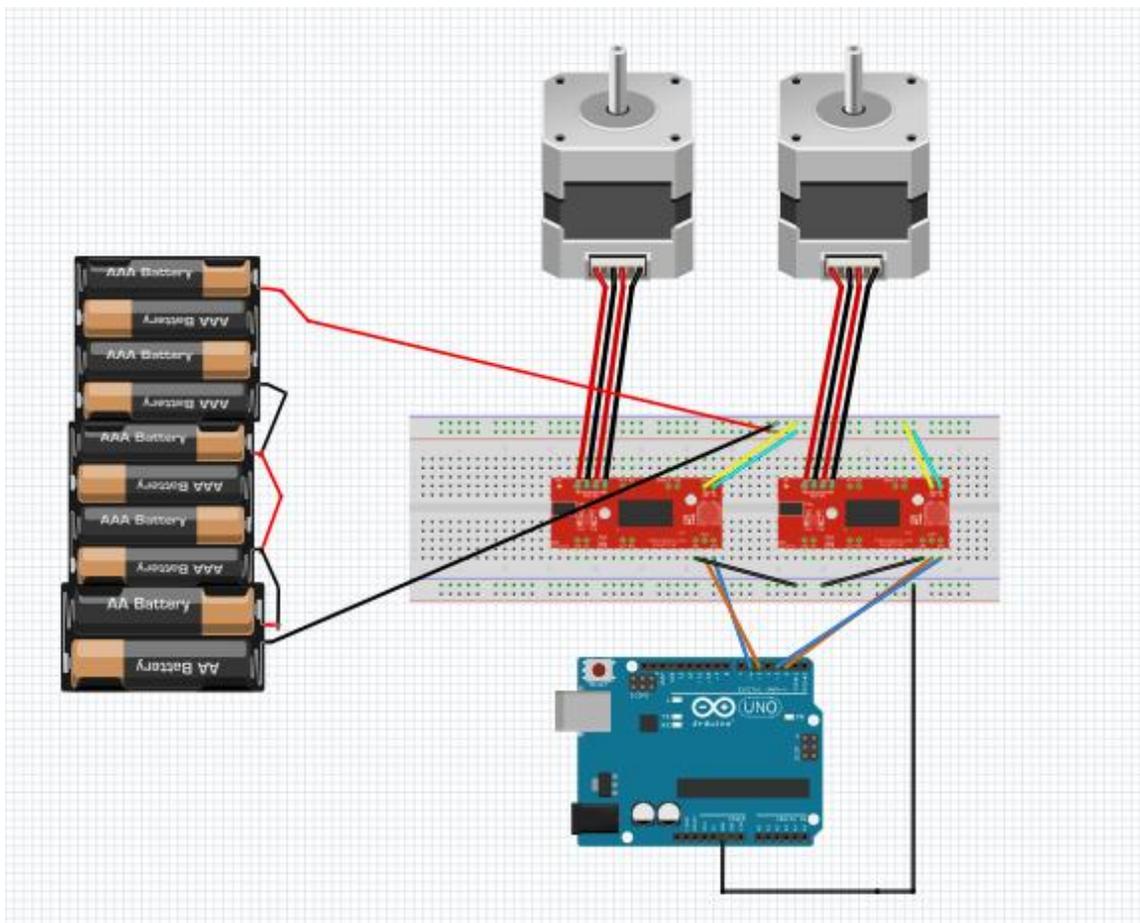


Figura 3.14: Esquema de conexiones para la parte de los motores

Por otro lado, para incluir la funcionalidad del Bluetooth se conecta directamente el módulo HC-06 a la placa de prototipado, por la parte de la izquierda, que es donde hay más espacio. A continuación, se conectan unas resistencias (como se aprecia en la imagen siguiente) para bajar la tensión del pin TX del Arduino de 5 a aproximadamente 3 V, ya que el módulo Bluetooth tiene un rango de tensión más estrecho y bajo que los

drivers, admitiendo hasta un máximo en torno a 6 V. Por ese mismo motivo, se debe alimentar desde el pin de 3,3 V del Arduino, conectándolo con el pin VCC del HC-06, y conectar el segundo pin de tierra del Arduino con la línea de la *protoboard* donde se inserta el cable de GND del Bluetooth.

El pin RX del Bluetooth se conecta al TX del Arduino (pin 0 en el Arduino Uno) y el pin TX se conecta al pin RX (pin 1 del Arduino Uno). Es importante desconectar ambos cables cuando se va a cargar el programa y conectarlos una vez cargado porque si no da problemas y no se puede subir a la placa.

Para controlar el Bluetooth desde otro dispositivo, en este caso para hacer las pruebas, desde el móvil, es necesario sincronizar ambos dispositivos. Para ello, se activa el Bluetooth del teléfono y se buscan los dispositivos disponibles, seleccionando el HC-06. Se solicita entonces una contraseña que, por defecto, es 1234 para el HC-06, y una vez introducida quedan ambos dispositivos vinculados. Cuando esto ocurre, el led del módulo Bluetooth que hasta entonces estaba parpadeando queda fijo.

En la Figura 3.15 se muestra el esquema de conexiones de la parte del Bluetooth exclusivamente.

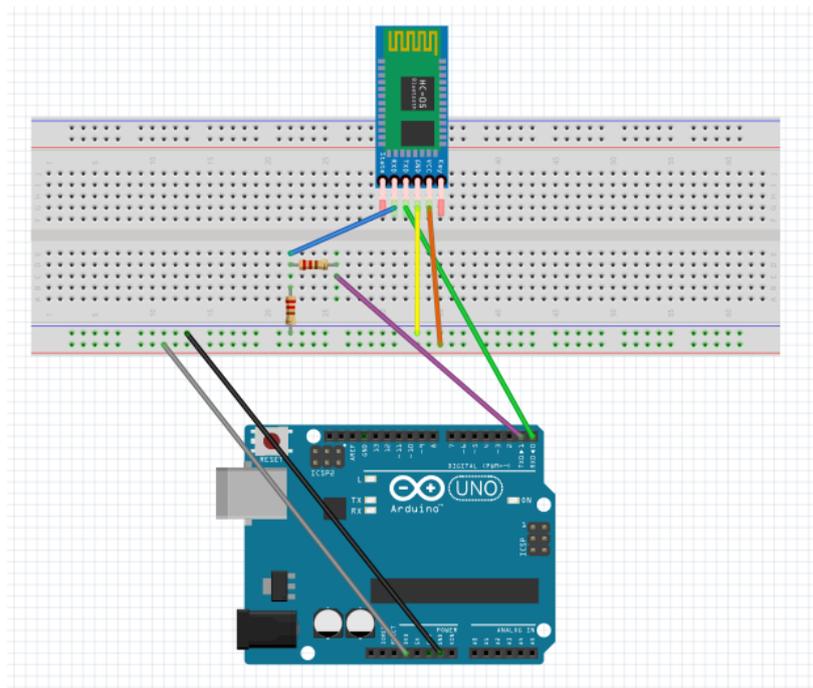


Figura 3.15: Esquema de conexión para el Bluetooth

Finalmente, el montaje del Bluetooth más los motores, se realizó según se ilustra en el esquema de la Figura 3.16 quedando como muestra la Figura 3.17.

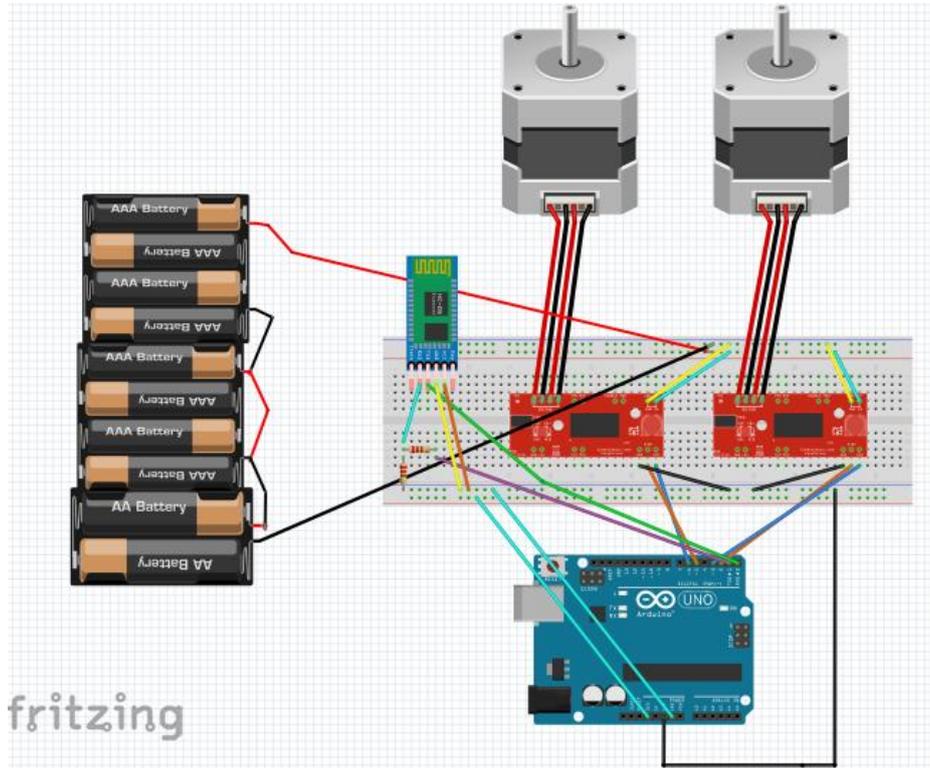


Figura 3.16: Esquema de conexiones completo

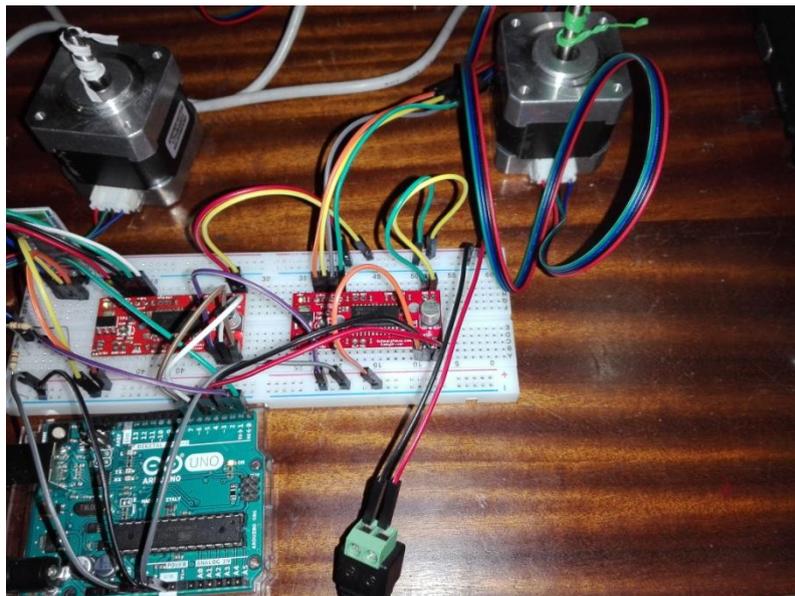


Figura 3.17: Imagen de las conexiones reales

3.2.1.2.2. Montaje de la estructura

La estructura decidió hacerse con impresora 3D en su mayoría. Las únicas partes que no se hicieron así fueron las varillas metálicas roscadas que unían las distintas partes y las tuercas que las sujetaban. Las piezas de la impresora 3D se diseñaron con FreeCad, un programa gratuito e intuitivo de usar puesto que combina características de otros programas de diseño asistido por ordenador.

La estructura está formada por distintas plataformas. La primera (Figura 3.18) es la pieza donde van a engranar los motores. Se encaja en el joystick por medio del agujero que tiene en su centro con la medida exacta para que no pase de cierto punto y resbale hacia abajo. Esta pieza cuenta en su cara inferior con un conjunto de cuatro cremalleras anchas dispuestas en forma de cruz en torno al agujero central. La anchura adicional que presentan respecto a una cremallera dentada habitual es para permitir el deslizamiento en una dirección mientras se mueve en la perpendicular, permitiendo así el desplazamiento simultáneo en dos direcciones sin usar otros elementos que complicarían el diseño.

Para hacer las cremalleras y los engranajes (Figura 3.19) fue necesario descargar una librería adicional de FreeCad, llamada *Gear*.



Figura 3.18 a: Vista de la superficie inferior de la primera plataforma donde engranan los motores

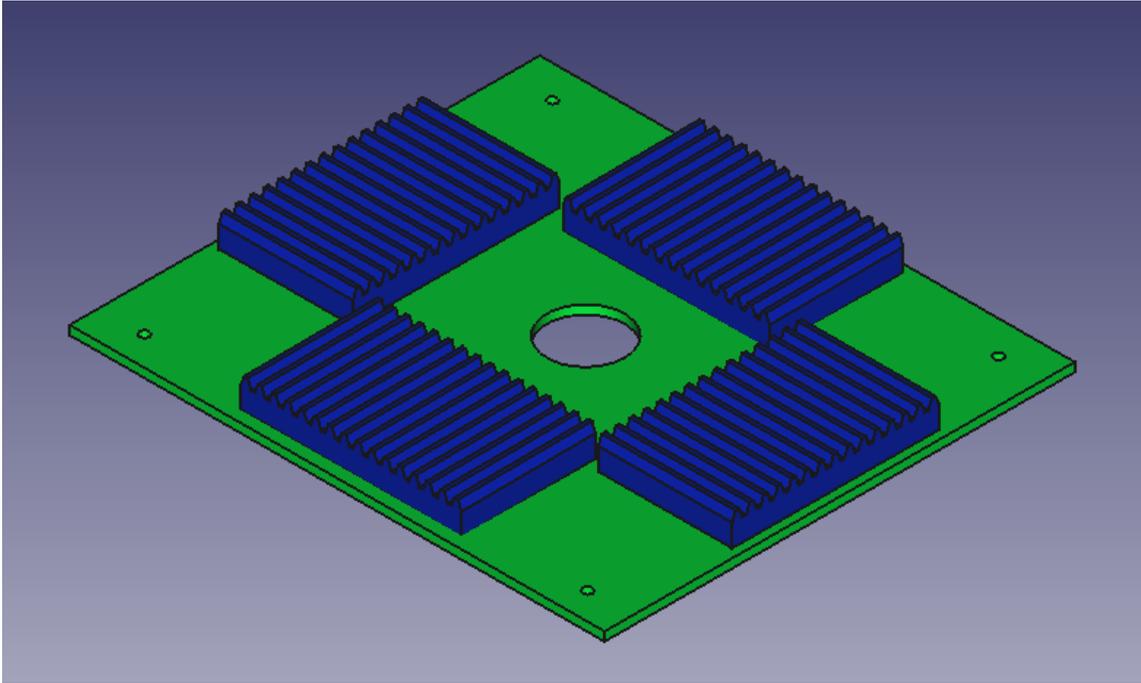


Figura 3.18 b: Diseño de la pieza en FreeCad

El hecho de estar engranados los dos motores a la vez consigue evitar la rotación de la pieza sobre el joystick y facilita el guiado de los engranajes por los dientes de las cremalleras. Los engranajes de los motores se sitúan en la mitad de las cremalleras; un motor a la derecha del joystick y otro delante de él dirigiendo su eje hacia el centro.

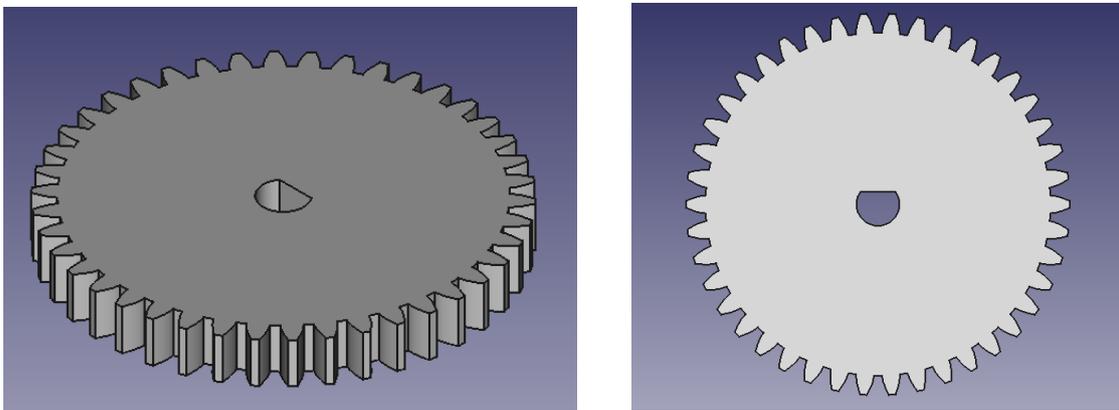


Figura 3.19: Vistas del modelo de los engranajes en FreeCad

Van apoyados sobre otra base (Figura 3.20) que con un gran agujero central permite su colocación encima de la superficie superior de la base del joystick. Para su correcto posicionamiento es necesario elevarlos ligeramente sobre la superficie mediante un elemento tipo lengüeta o cuña horizontal evitando que los engranajes rocen con la superficie inferior.

Con el fin de que la pieza tenga mayor rigidez se decidió pegar por debajo de esta pieza unos palos planos de madera haciendo una armadura plana en forma de cuadrícula.

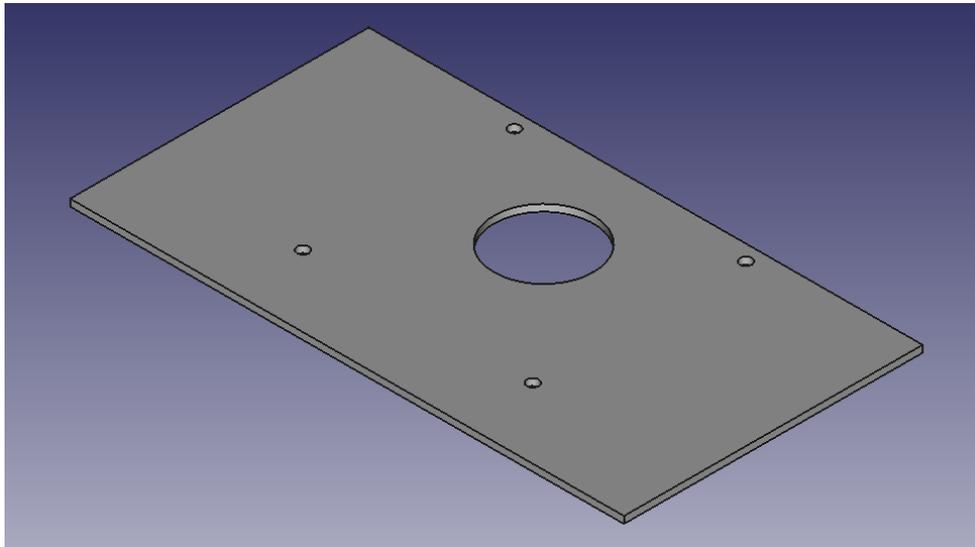


Figura 3.20: Base de apoyo de los motores

La base mencionada anteriormente cuenta con cuatro agujeros pasantes que permiten conectar con la estructura en la que van colocados los elementos de la parte electrónica.

Más abajo se encuentra una plataforma donde va la *protoboard* con todos los cables de conexión y los elementos que lleva conectados, es decir, *drivers* y módulo Bluetooth (Figura 3.21).

Finalmente, hay una última plataforma (Figura 3.21) donde se coloca la alimentación del Arduino y el Arduino, y que presenta una forma de caja abierta por arriba, evitando el vuelco de los materiales por los laterales. Esta plataforma, al igual que la anterior, cuenta con unos agujeros pasantes por donde van las varillas, roscándolas por debajo con tuercas. En esta última base las tuercas son tuercas ciegas y no pasantes.

Cada una de las plataformas va sujeta por las tuercas roscadas a las varillas por arriba y por abajo, de manera que asociadas a cada pieza se encuentran ocho roscas.

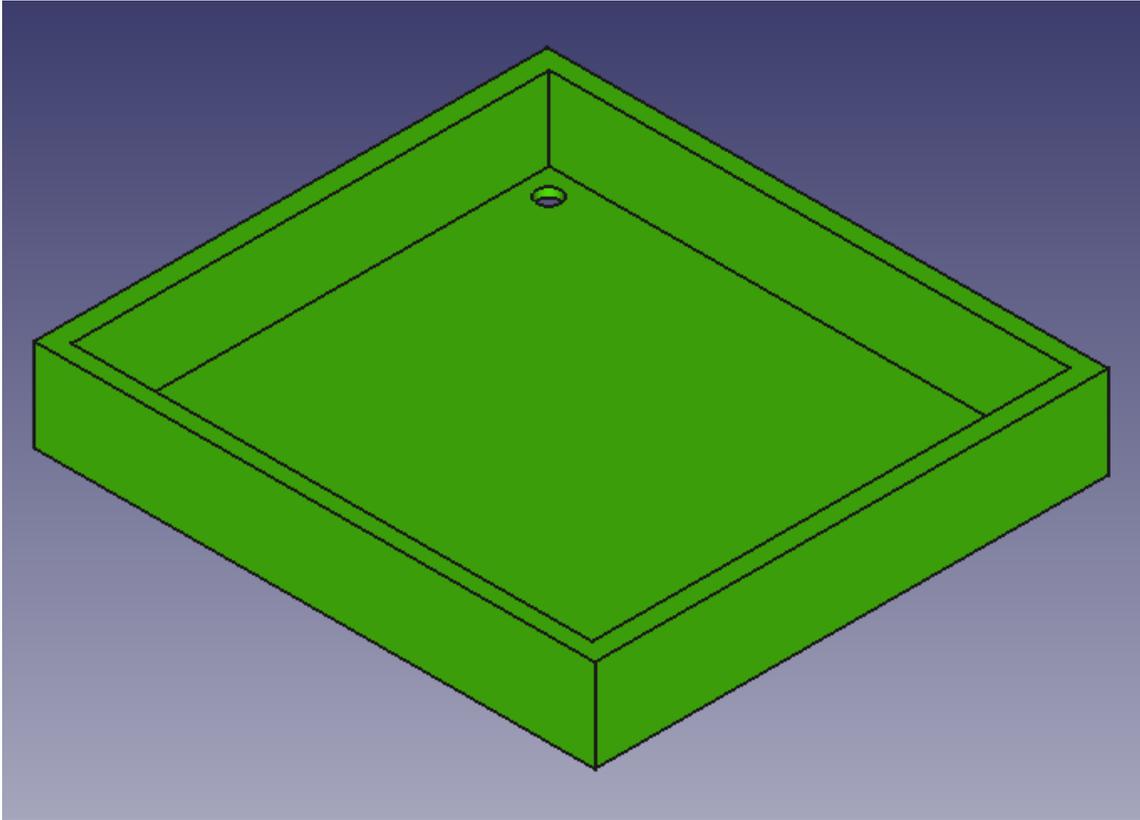


Figura 3.21: Plataforma donde se alojan los componentes electrónicos

3.2.2. Desarrollo del software

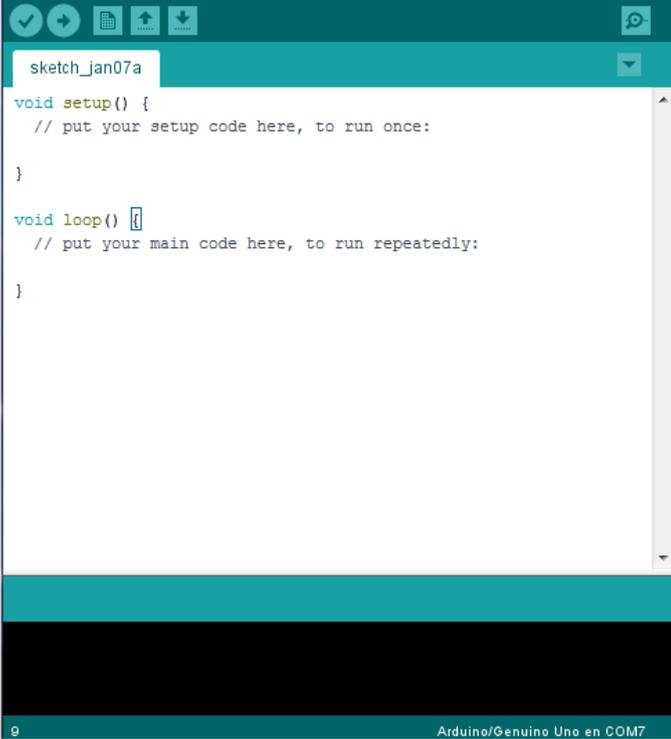
El programa que se ha utilizado ha sido el entorno de desarrollo integrado de Arduino (IDE de Arduino, de sus siglas en inglés). Se trata de un software sencillo, que permite programar fácilmente cualquier dispositivo automático y suficientemente potente para desarrollar diversos proyectos de diversa índole, como en el presente proyecto. En este caso, su función es recibir unos datos procedentes de un sensor vía Bluetooth y actuar sobre los motores.

El IDE de Arduino es un software libre y de código abierto, que cuenta con una comunidad online de desarrolladores y foros (página web Arduino.cc) de donde se puede obtener gratuitamente el programa y otras funciones relacionadas.

Para hacer el esquema de conexiones se ha utilizado Fritzing, un programa también de uso libre.

Respecto al lenguaje de programación, Arduino usa un lenguaje propio basado en Processing, C y C++. Todos los programas tienen una estructura común formada por tres partes: una parte inicial donde se incluyen las librerías que se van a usar, algunas

cabeceras y definiciones; el `void setup ()`, donde se inicializan variables, se indica si los pines son de salida o de entrada, o se incluyen algunas funcionalidades que solo se ejecutan una vez, y el `void loop ()` donde se programa el funcionamiento que se desee. La diferencia entre el `setup` y el `loop` es que el `setup` solo se ejecuta una vez, mientras que el `loop` es un bucle infinito que se está ejecutando continuamente.



```
sketch_jan07a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Arduino/Genuino Uno en COM7

Figura 3.22: Interfaz del IDE de Arduino

3.2.2.1. Programa principal

En primer lugar se incluye la librería `AccelStepper`, necesaria para el movimiento de los motores y se asignan los pines a cada uno de los dos motores que se va a emplear.

Después, se declaran las variables donde se van a guardar los valores en grados que se reciben del sensor, inicializándolas a 0. Se usa una variable auxiliar para asignar ángulo frontal o transversal según se haya seleccionado.

Para leer los datos del sensor se utiliza una cadena. A continuación, se declaran dos variables que se utilizarán para identificar el dato recibido, según se explica en la parte del `loop`. Tras estas, se declara una variable de tipo `float` para ajustar el movimiento de los motores a los límites de su recorrido físico, llamada "limit". No se usa un entero porque al incluirlo en la operación que calcula el valor que se le pasa a cada motor, se convertiría en un entero, pudiendo obtener valores indeseados. También, para

compensar el efecto de la reductora interna que llevan los motores, se incluye otra variable tipo *float* llamada “cte”, cuyo valor se calcula como se comenta en el código (ver Anexo 1).

Más abajo, se incluye una sección de las variables que el usuario puede modificar, con un comentario inicial y final que así lo identifica. Las primeras que se encuentran son el mínimo y el máximo de los ángulos del sensor. El mínimo se establece para ignorar movimientos involuntarios o temblores, con un valor de 5 grados. El máximo es un valor a partir del cual los motores se moverán al límite de su recorrido, obteniéndose desplazamientos proporcionales a éste para valores comprendidos entre el mínimo y el máximo. Al igual que antes, para poder hacer esta operación, el máximo debe ser de tipo *float* porque si no la división la considera como división de enteros, resultando 0 si el denominador es mayor que el numerador.

A continuación, se define una variable llamada “angulo_elegido”, de forma que si vale uno, se seleccionará el plano frontal y si vale dos, el transversal. Esto sirve para elegir el movimiento entre uno u otro plano con el que mover el joystick. No se hace referencia al plano sagital porque siempre va a estar presente.

Por último, se establece un nivel de dificultad por defecto (con valor igual a 1.0), con la posibilidad de cambiar su valor modificando el código si se desea, pudiendo tomar los valores: 1.1, 1.2 ó 1.3, de forma que al aumentar la dificultad disminuye el desplazamiento de los motores.

En el *setup*, se inicializa el monitor serial, esto es, la ventana en la que se puede ir comprobando el estado del programa y los valores que toman de las variables si se ha programado para tal objetivo. También se establece el valor de la máxima velocidad y aceleración para cada uno de los motores; por defecto, se consideran iguales para ambos.

Tras esto, se encuentra el *loop*. En él lo primero que se hace es comprobar si hay datos disponibles en el buffer serial, es decir, si llegan datos y, mientras esa condición se cumpla, se ejecuta el bucle continuamente. En él, se leen los caracteres que llegan por el puerto serial, y se van concatenando en una cadena de caracteres.

Para identificar el ángulo que se ha recibido se usan las dos variables anteriormente mencionadas: “cuenta_interna” y “cuenta”. Al inicio, valen 0; cuando coinciden, entonces es el ángulo sagital el que se ha recibido; cuando “cuenta_interna” es igual a “cuenta” más uno, es el ángulo frontal; y cuando “cuenta_interna” es igual a “cuenta” más dos, es el ángulo transversal.

Para que vaya pasando de uno a otro, después de cada comprobación se aumenta el valor de “cuenta_interna”, haciendo que coincida con la comprobación del ángulo que sigue, en la siguiente vez que se ejecuta el *loop*. Sin embargo, en la última

comprobación (cada comprobación es un *if*), se aumenta “cuenta” en dos unidades para que coincida con el valor que tiene “cuenta_interna” y pueda volver a comenzar las comprobaciones.

Después de las tres comprobaciones de los *if* y de incrementar la cuenta correspondiente, se asigna a la variable de cada ángulo el valor de la cadena previamente convertido a entero.

Una vez realizado esto, se comprueba si el dato en valor absoluto es mayor que el mínimo. En caso de que no lo sea, se asigna un cero al ángulo para que el joystick vuelva al origen, si no estaba en él. En caso de cumplirlo, se comprueba si dicho valor absoluto es mayor que el máximo. Si lo es, toma el valor del máximo en positivo o negativo según sea mayor o menor que cero el ángulo; si no lo es, se sigue la ejecución normal del programa.

Finalizada esta parte y a continuación de los tres *if*, se asigna el valor al ángulo sagital, con la siguiente fórmula:

$$\text{dato_rx} = \frac{\text{dato_rx} * \text{cte} * \text{limit}}{\text{MAX} * \text{nivel}},$$

Donde “dato_rx” es el valor del ángulo sagital, “limit” es el valor máximo de pasos que puede dar el motor sin salirse de la cremallera, “MAX” es el ángulo máximo y “nivel” es el nivel de dificultad.

Hecho esto, se llama a la función “mover()” y se inicializa la variable de tipo cadena.

En la función “mover()”, lo primero que se hace es comprobar si se ha elegido el ángulo frontal o el transversal. A continuación, se asigna el valor del ángulo frontal o transversal según el caso y con una fórmula equivalente a la anterior, a una variable llamada “angle” y se le pasa este valor a la función que asigna la distancia a moverse al motor del giro.

Después, si se cumple no se está en el caso de ángulo sagital, se da la orden para mover el motor de giro. En caso contrario, se asigna la distancia a moverse al motor de avance, y tras comprobar que el motor de giro ha terminado su movimiento, se da la orden para mover el motor de avance.

Esto se hace así por motivos de seguridad para evitar posibles colisiones frontales y para tener un mayor control de la silla, estableciendo, como se ha dicho antes, primero el giro y luego el avance.

Todos los mensajes que se muestran por pantalla tienen la finalidad de conocer y seguir la ejecución del programa en la fase de pruebas. En la aplicación real, estos mensajes carecen de importancia porque no se van a visualizar, dado que el Arduino no se conecta por USB al ordenador.

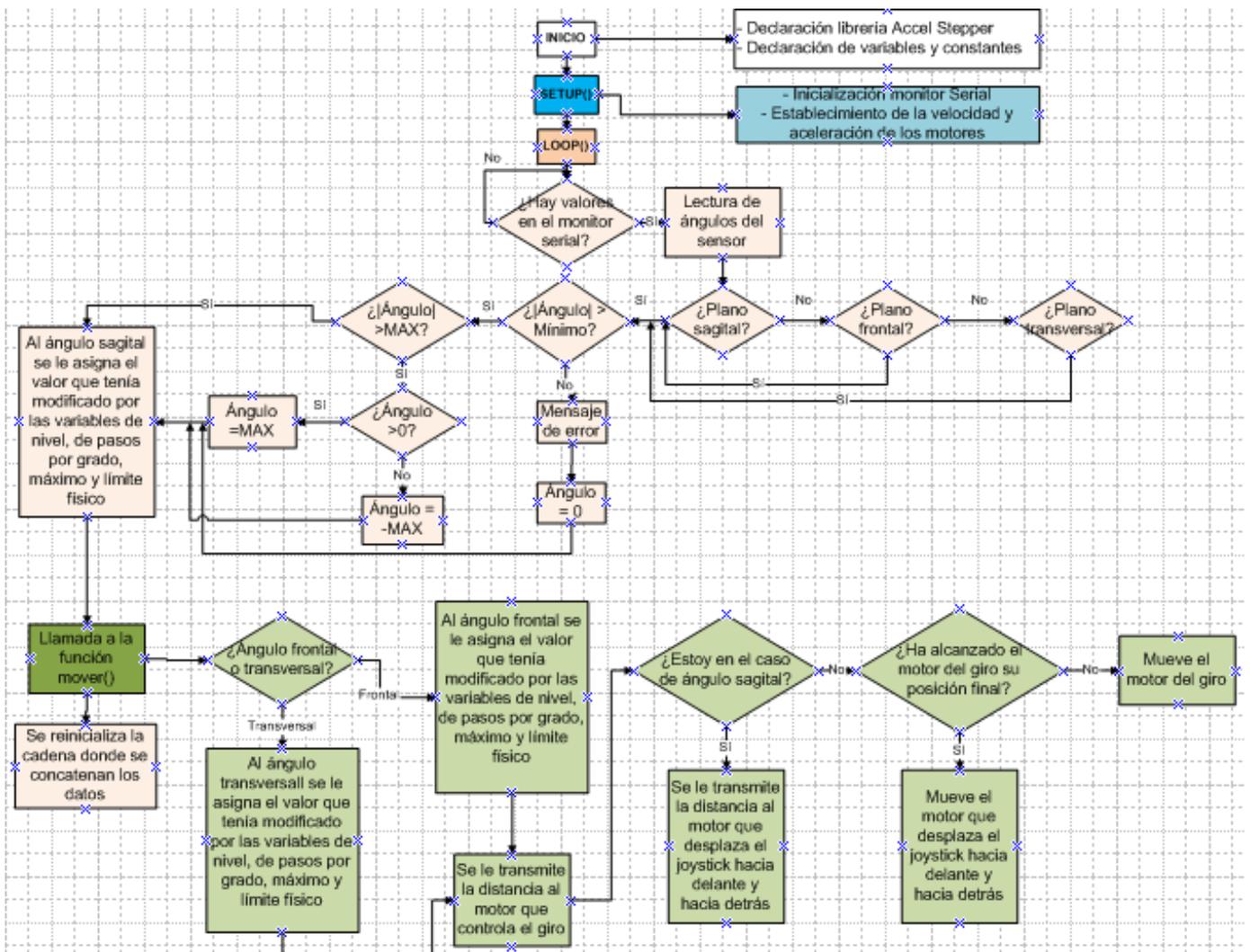


Figura 3.23: Flujoograma del proceso

3.2.2.2. Programas auxiliares

Como se ha indicado en el apartado de Montaje, se usó el software libre FreeCad, que es un programa de diseño asistido por ordenador gratuito. Desde los modelos generados por FreeCad se utilizó el software libre Cura V15-04.6 para generar los archivos en *gcode*, que son los que interpreta la impresora 3D.

Otros programas auxiliares que se han utilizado están asociados a la elaboración de la memoria, no a la construcción y diseño del dispositivo.

Para hacer el flujoograma del proceso se ha usado Microsoft Visio 2007. Por su parte, para hacer el esquema de conexiones se ha utilizado Fritzing, un software libre para el diseño de esquemas de conexión de componentes electrónicos.

4. DESCRIPCIÓN DEL DISPOSITIVO. ALTERNATIVAS CONSIDERADAS

En este capítulo se describe el dispositivo desarrollado en este proyecto: un dispositivo orientado al ámbito sanitario, que permite mover una silla de ruedas eléctrica con la cabeza y que está destinado a fortalecer la musculatura cervical con el fin de rehabilitarla o en su defecto mantenerla para prevenir posibles problemas de salud como consecuencia de la degeneración de esos músculos.

En este apartado se comentarán sus características generales del dispositivo así como las alternativas consideradas durante el desarrollo del proyecto.

4.1. Características generales

La misión principal del dispositivo consiste en mover el joystick de una silla de ruedas eléctrica mediante una plataforma en dos dimensiones accionada por unos motores paso a paso. Estos reciben la señal vía Bluetooth procedente de un sensor que se coloca en la cabeza del usuario.

Los motores tienen acoplados unos engranajes que actúan sobre una parte dentada que se encuentran en la cara inferior de dicha plataforma. Esa parte dentada es un conjunto de cuatro cremalleras individuales de gran anchura que permiten desplazarse a la plataforma en una dirección y en la perpendicular al mismo tiempo. Se sitúan en torno al agujero central de la pieza en forma de cruz. Dicho agujero central es donde se encaja el joystick.

El procesamiento de los datos que se reciben vía Bluetooth y el envío de los mismos (una vez transformados para poder ser ejecutados por los motores) lo lleva a cabo el Arduino Uno. Esta placa de control de bajo coste y pequeño tamaño se alimenta mediante una batería recargable de 7,4 V y 2250 mAh. Ambos, tanto el Arduino como la alimentación, se encuentran en un soporte de forma cuadrada y sujeto por unas varillas roscadas con tuercas.

Ese soporte es la parte inferior de una estructura con varias plataformas intermedias. Las dos primeras permiten la sujeción al joystick, la primera por la parte de arriba de la base del joystick y la segunda por abajo. En la primera es donde van situados los motores.

Hay otra plataforma entre la de abajo del joystick y el soporte del Arduino y la batería, que es donde van la *proto-board* en la que se realizan las conexiones junto con el módulo Bluetooth, los drivers y la alimentación de los motores.

La razón de usar alimentación no mediante cables sino mediante pilas y baterías es que el sistema ha de ser portátil; es decir, tiene que poder moverse la silla de ruedas libremente sin estar enchufada a ninguna alimentación.

Con este dispositivo se pretende lograr que las personas con hipotonía cervical (independientemente de su origen) puedan ejercitar el movimiento del cuello. Para ello, se han dispuesto unas variables programables que consiguen adecuar el sistema al usuario del dispositivo y que se explican en el siguiente apartado.

4.2. Variables programables

Dado que el dispositivo puede emplearse para más de una finalidad (recuperar, mantener o fortalecer la movilidad del cuello), es necesario que distintas características puedan ser fácilmente adaptadas. Por esta razón, en el programa desarrollado hay un conjunto de parámetros modificables que permiten cambiar las condiciones de funcionamiento de una forma rápida y sin tener conocimientos específicos de programación.

Un motivo adicional para incorporar estas variables es que de esta manera se puede utilizar el sistema para cualquier persona con hipotonía cervical independientemente de su grado de movilidad, puesto que estas variables permiten la versatilidad de condiciones.

Por lo tanto, los parámetros sobre los que se puede actuar son los siguientes:

- Ángulo de inclinación de la cabeza del que se van a recibir los datos: Esto es, seleccionar qué planos se van a utilizar para controlar el dispositivo. Puesto que no todos los sujetos tienen la misma capacidad de movimiento en todos los planos es necesario poder elegir entre el plano frontal y el transversal según el caso. El plano sagital siempre va a estar presente.
- Mínimo y máximo de los ángulos del sensor: Es necesario determinar a partir de qué ángulo se empiezan a mover los motores para que el sistema sea robusto ante posibles movimientos involuntarios. También es conveniente establecer un máximo a partir del cual se consigue el movimiento máximo de los motores, para evitar que se salgan los engranajes de su recorrido físico.
- Nivel de complejidad del movimiento: Selecciona un nivel u otro en función de si se quiere mayor o menor facilidad de movimiento. Actúa como un factor de reducción que minora los pasos que da el motor según se incremente el nivel de dificultad.

```
//---VARIABLES A MODIFICAR POR EL USUARIO---//  
  
//Para determinar el mínimo y el máximo de los ángulos del sensor  
#define MIN 5  
float MAX=20.0;  
  
//Para seleccionar el plano frontal '1', para el plano transversal '2'  
int plano=1;  
  
//Seleccionar el nivel de dificultad 1, 1.1, 1.2, 1.3 ó 1.4  
float factor=1;  
  
//---FIN DE VARIABLES A MODIFICAR POR EL USUARIO---//
```

Figura 4.1: Imagen del código donde se especifican las variables programables

4.3. Ejemplos de funcionamiento

A continuación se van a exponer dos ejemplos en los que se hace patente el funcionamiento del dispositivo ilustrando su utilidad y aplicaciones.

Es menester decir que son casos ficticios por lo que las indicaciones y datos dados no deben tomarse en cuenta para su aplicación en usuarios reales, sino que estos deben ser establecidos por un experto, preferiblemente un terapeuta ocupacional o fisioterapeuta, que lleve el seguimiento del paciente.

- Ejemplo 1: Sesión de entrenamiento

Un niño de diez años con parálisis cerebral de tipo atetosis, caracterizada por movimientos lentos e irregulares, acude a su primera sesión de entrenamiento para poder ir adquiriendo control del movimiento de la cabeza. Estas sesiones de entrenamiento de carácter semanal se las prescribió el terapeuta ocupacional que lleva su seguimiento como complemento a otras para otros miembros del cuerpo que también recibe.

Al niño, sus padres le comunican que va a jugar como otro día cualquiera con su terapeuta a quien llama “Tera”. Tera le dice que hoy va a conducir, consiguiendo despertar en el niño una gran satisfacción que le hace moverse brusca y repentinamente. Tras calmarle, el terapeuta le coloca el sensor en la cabeza y prepara el sistema para colocarlo sobre el joystick y anclarlo al soporte.

Previamente, ha establecido en el programa el mínimo y el máximo en 5 y 20 grados respectivamente, el nivel de dificultad 1 (puesto que está empezando) y el plano frontal en lugar del transversal porque es donde el niño tiene más movilidad. Se trata

de que al mover el cuello fortalezca esos músculos en los que tiene más movilidad y vaya adquiriendo tono en los músculos más hipotónicos. El terapeuta ha previsto una duración de una hora para la sesión.

Así pues, con todo preparado enciende el joystick y le dice al niño que mueva la cabeza para conducir la silla, mirando hacia donde quiere ir. El niño empieza inclinando la cabeza hacia el lado derecho y la silla empieza a girar, pero no consigue avanzar hacia delante. El terapeuta le indica que tiene que inclinar la cabeza hacia delante para moverse, enseñándole el movimiento que tiene que hacer. Con esfuerzo, el niño consigue mover la cabeza hacia delante y poco a poco va comprendiendo cómo conducir la silla.

Pasados quince minutos desde el inicio de la sesión y viendo que ya es capaz de controlar la silla, el terapeuta le hace parar y le propone un juego. Tiene que llegar hasta el final de la sala y dar la vuelta donde el terapeuta ha colocado un cono; y al llegar al principio lo mismo; cinco veces en un sentido y luego las mismas en el otro.

Viendo que es capaz de realizarlo bien, el terapeuta decide poner un cono en medio de los otros dos. Ahora él hace el recorrido sorteando los conos y el niño tiene que seguirle. De esta forma, el niño se encuentra más seguro que antes. De hecho, como le parece estar jugando al pilla-pilla una de las veces se inclina mucho hacia delante alcanzando al terapeuta que se sorprende del entusiasmo del niño.

Sin embargo, tras 45 minutos el niño empieza a mover menos la cabeza como consecuencia de la fatiga y le cuesta más desplazarse. Es entonces cuando el terapeuta decide dar por acabada la sesión felicitándole por lo bien que lo ha hecho y diciéndole que le espera la semana siguiente, a lo que el niño le responde con un enérgico “sí”.

Ha pasado una semana y el niño vuelve a la “clase de conducir” como dicen sus padres. Espera alegre a que su “Tera” le coloque el sensor y prepare el sistema. Esta vez, elige el siguiente nivel de dificultad, manteniendo los demás parámetros iguales.

Realiza el mismo circuito que la semana anterior, y el terapeuta se sorprende porque el niño no parece acusar el incremento de dificultad, ya que ahora tiene que desplazar más la cabeza para conseguir el mismo movimiento que antes.

Para el segundo ejercicio, le guía igualmente por entre los conos, y el niño vuelve a jugar al pilla-pilla con su terapeuta, mostrando el mismo entusiasmo que la semana previa.

Cuando llegan los 45 minutos de sesión le pregunta al niño si está cansado. Él le responde que no, que quiere más conducir. Pero el terapeuta decide dar un descanso de cinco minutos y mientras ajusta el nivel de dificultad al 1. Cuando vuelve a activar el

sistema, el niño nota que es más fácil entonces mover la silla y consigue aguantar la hora entera.

Tras varias semanas de entrenamiento progresivo, los padres le comentan al terapeuta que ahora su hijo tiene el cuerpo más erguido, y cuando le hablan es capaz de levantar más la cabeza, mirándoles de frente e interactuando más con ellos.

- Ejemplo 2: Recuperación de movilidad cervical

Una mujer de cuarenta y cinco años se encuentra en el metro y empieza a decir palabras raras y a marearse poniendo un gesto desfigurado y con la vista perdida. Antes de caerse un hombre que estaba a su lado la coge y la sienta en el suelo. Inmediatamente, acciona la parada de emergencia antes de que se pusiese en marcha de nuevo y llama a emergencias. Rápidamente la mujer es trasladada a un hospital. La mujer ha sufrido un ictus.

Tras varias semanas en el hospital la mujer vuelve a su casa pero con secuelas, aunque menores de lo que podría haber sido si no hubiese habido una rápida intervención. Tiene paralizado el lado izquierdo del cuerpo sobre todo el brazo. La movilidad en el cuello la conserva parcialmente, pudiendo girar más la cabeza hacia el lado derecho que hacia el izquierdo.

El médico le propone unas sesiones de rehabilitación tanto para el brazo como para el cuello, con la esperanza de que, como en otros casos médicos ha ocurrido, la persona pueda volver a valerse por sí misma, aunque después de mucho esfuerzo. El doctor afirma que está bastante convencido de que el cuello lo podrá volver a mover, si no como antes, sí más que ahora.

Así pues, la paciente inicia las sesiones de entrenamiento cervical. Para ello, le colocan un sensor en la cabeza y la hacen sentar en una silla de ruedas eléctrica con el dispositivo ya colocado en el joystick. El responsable ha seleccionado el plano transversal de movimiento por indicación del médico.

Le comenta que tiene que moverse por la sala sin chocar con las paredes. La mujer, algo asustada al principio, empieza a mover la cabeza hacia delante y hacia la derecha. Cuando le dice que gire a la izquierda se da cuenta de la diferencia de movilidad hacia un lado y hacia otro. Decide pues, rebajar el ángulo mínimo de movimiento para poder producir desplazamiento efectivo. Le dice que no se preocupe si ahora le cuesta moverse más hacia la izquierda porque en un tiempo podrá moverse sin problemas.

Tras tres cuartos de hora de ejercicios de desplazarse con la silla por la sala finaliza la sesión.

La siguiente sesión la recibe esa misma semana, para cumplir con los consejos del médico de dos veces por semana para la rehabilitación del cuello y los tres restantes para el brazo.

En esta ocasión, el rehabilitador ha subido el nivel de dificultad pero manteniendo el mínimo modificado de la sesión anterior. La mujer se da cuenta de que tiene que girar más la cabeza para desplazar la silla pero una gran fuerza de voluntad la hace capaz de ello. El terapeuta percibe que inclina un poco más la cabeza hacia la izquierda, y le hace en los ejercicios girar más en ese sentido.

A las tres semanas, ya ha experimentado todos los niveles de dificultad del dispositivo y el médico nota una leve mejoría, por lo que le recomienda otras tres semanas de sesiones.

Pasado este tiempo, con ejercicios cada vez más complicados y variando el nivel de dificultad, la paciente es capaz de alcanzar un mayor grado de inclinación de la cabeza hacia el lado que tenía semiparalizado.

4.4. Alternativas

En las fases iniciales del proyecto se sopesaron diferentes alternativas a la hora de elegir el diseño físico, los componentes y el software. A continuación, se detallan las alternativas concretas estudiadas para cada caso.

4.4.1. Diseño físico

La primera cuestión que se planteó era cómo mover el joystick usando unos motores, es decir de qué forma se iba a convertir el movimiento rotatorio de estos en lineal en dos dimensiones que era lo que se buscaba. Lo más adecuado era utilizar motores paso a paso, que permitían un preciso control del movimiento; pero había que elegir el mecanismo con el que transformar ese movimiento.

Una primera opción era utilizar barras de deslizamiento como en las impresoras 3D, en las que haciéndolas girar por medio de los motores la plataforma correspondiente se desplaza en un sentido o en otro, siendo los movimientos en los tres ejes X, Y y Z independientes y desacoplados. Para este caso, se obviaría el eje Z y se utilizarían sólo el eje X y el eje Y. El movimiento en un eje haría que el motor del otro se desplazase, de forma que si se activase el movimiento en este segundo eje ya se obtendrían las dos dimensiones buscadas. Esto queda ilustrado en la siguiente imagen. En algunas variantes de estas plataformas XY, se incluían correas para transmitir el movimiento.



Figura 4.2: Plataforma con movimiento en dos dimensiones

Esto era una buena alternativa pero presentaba un problema de espacio, pues el espacio para poner algo alrededor del joystick es relativamente pequeño.

Otra alternativa considerada respecto al diseño era la utilización de una plataforma que se encajase en el joystick y que fuese posible moverla en dos dimensiones por medio de unas varillas sujetas a las esquinas de dicha plataforma y enganchadas a una plataforma inferior que sirviese de base, accionados por los motores. El problema era cómo mover en una dirección y en la perpendicular a la vez, ya que al estar ancladas en la base habría un desplazamiento impedido.

El hacer las cabezas de las varillas que se engancharan a la base pasantes, facilitaba el movimiento en la dirección perpendicular pero no solucionaba el problema del todo ya que el desplazamiento en esa dirección estaría limitado. Otra opción consistía en usar unos muelles enganchados a la esquina de la plataforma reforzados interiormente con un trozo de alambre, aunque esto quedaría, incluso con ese refuerzo, inestable. Una posible solución era el empleo de rótulas, pero eso encarecería mucho el proyecto.

Por lo todo esto, finalmente se descartó también esta opción, aunque sirvió de idea preliminar a la solución final, explicada en el apartado de Metodología.

4.4.2. Componentes

DISPOSITIVO DE CONTROL

Las alternativas consideradas fueron: Arduino, Raspberry Pi y FPGA.

Las FPGA son circuitos integrados configurables que pueden ser reprogramados mediante un lenguaje descriptivo (VHDL) interconexionando los distintos bloques según se desee.

Ofrecen una gran versatilidad y capacidad de procesamiento a un precio relativamente bajo para su potencialidad, pero aún así demasiado elevado para este proyecto. Otro factor que llevó a rechazarla fue que en este proyecto no eran necesarias tan elevadas prestaciones.



Figura 4.3: Imagen de una FPGA modelo Spartan-3 de Xilinx

La verdadera elección fue entre Arduino y Raspberry Pi. Ya que eran más parecidos entre sí frente a la FPGA.

Las características del Arduino ya se describieron en el apartado de metodología.

Por su parte, la Raspberry Pi es un computador de placa simple y por tanto, más potente que el Arduino, sobre todo a nivel software.

La Raspberry Pi surgió como fundación en el 2009 en el Reino Unido con una idea parecida a la que dio origen a Arduino, acercar la informática a los niños.

Las primeras ventas masivas tuvieron fecha en 2012 con una fuerte demanda de las mismas. Existen distintos modelos de Raspberry, la Raspberry Pi 1 A, A+, B, B+, Raspberry Pi 2, Raspberry Pi 3; que formarían parte del grupo de las clásicas, y otra gama de placas de formato y precio reducido, pero con menores prestaciones: la Raspberry Pi Zero y Zero W.

Todas tienen entrada y salida de vídeo, salida de audio, almacenamiento integrado, y funcionan con sistema operativo GNU/Linux. Tienen una gran capacidad de computación, reflejada en la velocidad del reloj y en la memoria, y la Raspberry Pi 3 presenta la posibilidad de conexión vía Bluetooth.



Figura 4.4: Imagen de la Raspberry Pi 3

A continuación, se va a hacer una comparación entre las dos opciones, para determinar cuál resulta más adecuada al proyecto. En primer lugar, se hace una descripción de las características de cada uno en relación al otro, para finalmente tomar la decisión de qué dispositivo utilizar empleando el método de los factores ponderados.

- Precio

En cuanto al precio, la Raspberry Pi 3 más barata se encuentra por 39,90 €, en Media Markt. Por su parte, en el Arduino varían los precios según el modelo: desde los 20,75 € del Leonardo hasta los 34,89 € del Mega, pasando por los 19,89 € del Uno. A esto habría que añadirle el precio del módulo conector de Bluetooth, que oscila entre los 4 y los 10 euros.

- Información disponible

Ambos dispositivos cuentan con una comunidad online de usuarios, tutoriales, preguntas frecuentes y blogs y foros asociados, aunque es cierto que se encuentra más información de Arduino que de Raspberry, lo cual se puede comprobar por el número de resultados en el buscador de Google, habiendo casi 2,5 veces más resultados de Arduino que de Raspberry Pi.

- Prestaciones

- Conectividad Bluetooth

La Raspberry Pi 3 es el único modelo de Raspberry Pi que presenta conectividad Bluetooth integrado. Sin embargo, cualquiera de los modelos de Arduino necesita de un conector específico para tener la posibilidad del Bluetooth.

- Memoria

El microprocesador ATmega328 (muy frecuente en las placas) tiene una memoria flash de 32 kB, una SRAM de 2 kB y una EEPROM de 1 kB. Mientras que la Raspberry Pi 3 tiene una memoria SDRAM de 1 GB.

- Velocidad de reloj

El reloj del Arduino va a 16 MHz para todos los microcontroladores. Por su parte, en la Raspberry Pi 3 va a 1,2 GHz.

- Entradas y salidas

El ATmega328 tiene 14 entradas y salidas digitales, seis de ellas pueden controlar un PWM, y seis salidas analógicas, que permiten gran versatilidad e interacción con múltiples dispositivos (sensores, actuadores...). Posee además un puerto USB. La Raspberry Pi 3 tiene 4 puertos USB, entrada y salida de vídeo y salida de audio.

- Sistema operativo

El Arduino no tiene sistema operativo propio, usa lenguaje C; la Raspberry usa distribuciones de Linux como Raspbian, no tan conocido como el lenguaje de programación C.

- Consumo

La intensidad de corriente para el Arduino varía entre 40 y 90 mA, con un voltaje operativo de 5 V. Para la Raspberry Pi 3 son 800 mA a 5 V. Estos consumos son medios, variando en función de lo que demande la carga o actividad correspondiente.

De la descripción anterior se deduce que:

- El Arduino tiene un precio considerablemente más reducido que la Raspberry Pi 3.
- El Arduino está más respaldado por la información disponible que la Raspberry.
- La Raspberry Pi supera en prestaciones al Arduino, si bien resultan excesivas puesto que no se estaría utilizando todo el potencial de este mini ordenador, y el precio de más que se paga no se vería reflejado en este uso concreto.

- El Arduino tiene una gran variedad de entradas y salidas digitales y analógicas, adecuado para acoplar módulos externos o para la comunicación entre dispositivos (sensor-placa, placa-actuador...)
- El Arduino funciona en cualquier ordenador, mientras que la Raspberry Pi 3 precisa de Linux para funcionar.
- El consumo del Arduino es más reducido, de lo que se deduce que la duración de la fuente de energía que se emplease para alimentarlo sería mayor.

Para determinar qué dispositivo elegir se va a utilizar una matriz de decisión en la que se va a dar una puntuación de 0 a 5 a cada aspecto considerado según su ajuste al proyecto, con un coeficiente, en tanto por uno, que refleja su importancia en relación a los criterios dichos al comienzo:

- Precio → 0.4
- Información disponible → 0.12
- Conectividad Bluetooth → 0.12
- Memoria → 0.05
- Velocidad del reloj → 0.03
- Entradas y salidas → 0.12
- Sistema operativo → 0.04
- Consumo → 0.12

Criterios de influencia

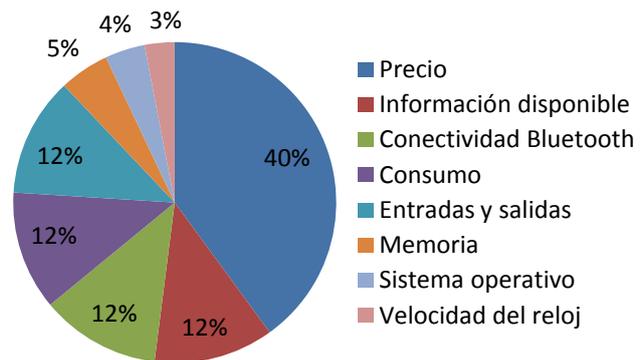


Figura 4.5: Criterios de influencia en la elección del dispositivo

	<u>Arduino</u>	<u>Raspberry Pi 3</u>
Precio	5	3
Información disponible	4	3
Prestaciones		
Conectividad Bluetooth	3	5
Memoria	3	4
Velocidad del reloj	2	4
Entradas y salidas	4	3
Sistema operativo	4	3
Consumo	4	2

Tabla 4.1: Factores ponderados para el Arduino y la Raspberry Pi 3

CONECTOR BLUETOOTH

Se consideró otro módulo de Bluetooth además del HC-06 finalmente elegido. El alternativo era el LM910-0630, que es un módulo USB de Bluetooth versión 4.1, de baja energía y mayor velocidad de transmisión de datos (EDR) que funciona entre los 2.4 y los 2.485 GHz.

La versión v4.0 surgió en 2010 e incluye el Bluetooth clásico, el de alta velocidad (basado en Wifi) y los protocolos de Bluetooth de bajo consumo. El módulo Bluetooth, pertenece a la clase 1, esto indica que tiene un alcance de hasta 110 metros.



Figura 4.7: Conector Bluetooth LM910-0630

En la Tabla 4.2 se recogen las características técnicas más importantes del conector:

Versión	Bluetooth 4.1 + EDR
Clase	1
Tensión de alimentación máxima	5 V
Velocidad de datos	3 Mbps
Frecuencia de trabajo	2.4 a 2.485 GHz
Temperatura de trabajo permitida	-20 a 75°C
Dimensiones (Largo x Ancho x Alto)	18 mm x 9.7 mm x 2.5 mm
Peso	0.81 g
Consumo de corriente (modo continuous Rx)	27 mA
Consumo de corriente (modo continuous Tx)	35 mA
Antena	Antena PCB

Tabla 4.2: Características del conector Bluetooth LM910-0630

Asimismo, tiene un factor de forma muy pequeño y un módulo de alto rendimiento que le permite estar conectado a 7 dispositivos simultáneamente.

A pesar de esto, la dificultad de conectarlo al Arduino o a la *protoboard* era mucho mayor que en el caso del HC-06 que se conecta directamente sobre la tabla de prototipado.

ALIMENTACIÓN ARDUINO

Para que el Arduino pudiese ser transportado junto con la silla de ruedas necesitaba recibir alimentación también portátil, es decir, in situ. Para ello, se barajaron varias alternativas de pilas y baterías para su utilización.

Una pila de 9 V era una opción válida para alimentar a un Arduino. Únicamente necesitaba de un portapilas y un conector jack, fácilmente disponibles. El problema es que proporcionan una corriente muy baja y no son recargables.

En su lugar se podrían utilizar cuatro pilas de seis voltios que proporcionan hasta dos amperios. Sin embargo, a largo plazo no resulta rentable porque una vez se agotan no se pueden recargar.

Por lo tanto, era necesario recurrir a pilas recargables. Las de 1,2 V AA se podrían usar en portapilas de 5 unidades para obtener 6V. Estas proporcionan menos carga que las convencionales, variando también según la composición, pero se pueden reutilizar.

Otra alternativa era usar dos baterías de litio recargables puestas en serie, que proporcionen 7,4 V que es un voltaje recomendado para alimentar Arduino. Son más caras que las anteriores, pero tienen buenas prestaciones.

Existe asimismo la opción de usar una batería USB, las mismas que se usan para cargar los móviles. Proporciona buena movilidad, son recargables, aunque son algo más caras que otras opciones y su intensidad máxima es inferior a 2 A.

ALIMENTACIÓN MOTORES

Las mismas alternativas que para la alimentación del Arduino fueron consideradas para los motores, con la salvedad de que había que proporcionar más tensión y corriente.

Existían pues dos grandes grupos de alternativas: baterías o pilas. Ambas habían de ser recargables para que fuesen rentables a largo plazo.

Aunque las baterías eran una opción más optimizada por duración y consumo, el hecho de tener que cargarlas con un cargador especial de muy alto precio las hizo ser descartadas por no cumplir con el objetivo de bajo coste que tiene el proyecto.

Entre las pilas había variedad de precios y de capacidad. Pero como el precio bajo era un requisito indispensable en el proyecto se descartaron las pilas que tenían un precio más alto. Aún así, dado que el precio variaba de unos establecimientos a otros se encontró una tienda de electrónica que vendía pilas recargables más baratas con más mAh que la competencia.

4.4.3. Software

El software empleado está determinado por el dispositivo de control elegido, de forma que si se hubiese optado por otro, el software también habría cambiado.

Como se explicó en el apartado de Metodología, el dispositivo seleccionado es un Arduino Uno, por lo que el software que corresponde es el IDE de Arduino.

Si se hubiese escogido una Raspberry habría que haber programado en Python, que es el lenguaje estándar de la Raspberry, o en Linux en su defecto usando las distribuciones e intérpretes que fuesen necesarios. Esta alternativa implicaría una mayor dificultad al tener que aprender a usar un lenguaje de programación, que aunque sencillo y potente, sería desconocido.

En el caso de utilizar una FPGA, el lenguaje empleado sería VHDL (Very High-Speed Description Language), que es un lenguaje de descripción de hardware muy potente porque cambia respecto al enfoque del lenguaje C, pues en este caso, se describe cómo está conectado el software, y qué elementos lo componen (puertas lógicas, decodificadores, multiplexores...). Surgió en los años 80 en el departamento de defensa de los Estados Unidos favoreciendo el desarrollo de la electrónica del momento. Un programa para escribir en VHDL sería Xilinx ISE, que permite programar descriptiva o esquemáticamente.

5. PRUEBAS, RESULTADOS Y CONCLUSIONES

Con el fin de comprobar el funcionamiento del dispositivo, se han realizado dos tipos de pruebas: pruebas del sistema electrónico y pruebas de simulación.

Posteriormente, se analizaron los resultados y se obtuvieron las conclusiones correspondientes.

5.1. Pruebas y resultados

Las pruebas del sistema electrónico tenían como objetivo corroborar que el comportamiento del sistema en cada una de sus partes era el deseado para poder seguir avanzando en la realización del proyecto. Esas partes se corresponden con el Bluetooth, los motores y el conjunto integrado de los motores con el Bluetooth.

Por su parte, las pruebas de simulación buscaban confirmar el correcto funcionamiento de la aplicación.

5.1.1. Pruebas del sistema electrónico

En primer lugar, se realizaron las pruebas de funcionamiento básico para validar el montaje electrónico, detectar posibles problemas y solucionarlos.

5.1.1.1. Pruebas del Bluetooth

Para realizar estas pruebas se utilizó una aplicación de Android llamada “BluControl”, cuya interfaz se muestra en la Figura 5.1.

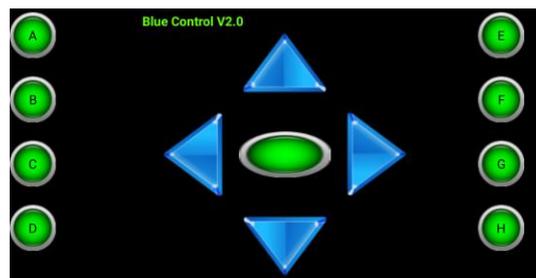


Figura 5.1: Interfaz de la aplicación BluControl

Con esta aplicación se hizo encender y apagar un led mediante un sencillo programa en Arduino.

Los resultados se recogen en la Tabla 5.1:

Tecla pulsada	Comportamiento esperado	Comportamiento observado
A	Se enciende el led y se mantiene en ese estado	Led encendido y sin cambiar
D	El led debe apagarse si está encendido, y empezar a parpadear cada 0,5 segundos durante dos segundos. Si está apagado, debe parpadear directamente hasta apagarse	Después de pulsar primero "A" y luego "D", se apaga y parpadea durante dos segundos. Si no se ha pulsado previamente A, simplemente parpadea ese tiempo.
Otra	No debe modificar su estado de encendido o apagado	No se observa ningún cambio

Tabla 5.1: Resultados de las pruebas con el Bluetooth

5.1.1.2. Pruebas de los motores

Para llevar a cabo las pruebas con los motores se usó un programa en Arduino (ver Anexo 3) en el que se les daba un determinado número de pasos por teclado desde el monitor serial y se comprobaba mediante una bandera que los motores habían hecho dicho movimiento utilizando varias funciones de la librería "AccelStepper".

Las funciones en las que se les asignaban los pasos ("stepper.moveTo(valor)") realizaban la operación de moverse considerando el valor introducido como una posición con coordenadas a la que ir, de manera que si primero se le daban 200 pasos y luego 250, la segunda vez solo iba a moverse 50, a menos que se le diese 0 para volver a su origen.

Para seguir el movimiento de rotación de los ejes se les colocó un alambre de color alrededor quedando el extremo en horizontal (ver Figura 5.4).

Las conexiones quedaron descritas en el apartado de Metodología, más concretamente en las Figuras 3.13 y 3.14.

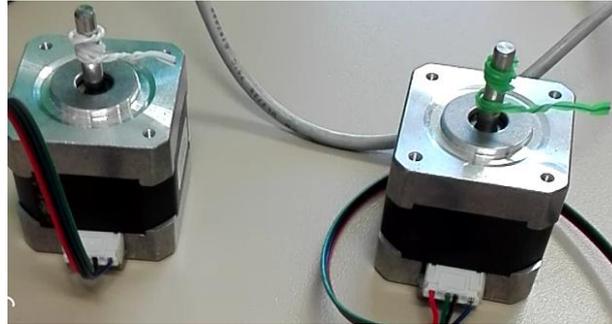


Figura 5.4: Motores durante las pruebas de funcionamiento

En la tabla 5.2 se recogen los resultados de las pruebas.

Secuencia de pasos	Comportamiento esperado	Comportamiento observado	Duración del movimiento (s)
0, 0	No deben moverse los motores	No hay desplazamiento alguno	0
200, 200	Los motores deben girar hasta completar una vuelta entera	Los motores se mueven los pasos indicados pero no dan una vuelta entera*	1,17
500, 500	Los motores deben moverse más tiempo que en el caso previo	La duración del movimiento es ligeramente superior respecto al comando anterior	1,89
0, 0	Los motores deben girar en sentido inverso al que han tenido antes y tener más duración este movimiento	Los motores giran en sentido opuesto durante más tiempo hasta volver a la posición inicial	1,92
1000, 1000	Los motores deben girar durante más tiempo que en el movimiento anterior y llevando sentido opuesto	Los motores tardan más en acabar su movimiento y llevan el mismo sentido que en las primeras secuencias de pasos	2,35
0, 0	Los motores deben quedar en su posición inicial	Los motores giran en sentido inverso quedando el alambre de referencia en la misma posición que al principio	2,28

Tabla 5.2: Resultados de las pruebas de los motores

La discrepancia observada puso de manifiesto la existencia de una reductora que llevan los motores internamente, y que experimentalmente se comprobó que era de $1/8$, ya que para dar una vuelta completa necesitaba 1600 pasos, cuando según el valor de grados por paso o paso angular (1,8 para el Nema 17) había de ser de 200.

5.1.1.3. Pruebas del conjunto integrado

Una vez conectados los motores y el Bluetooth en la protoboard según se ilustra en las Figuras 3.16 y 3.17 de la Metodología, se realizaron las pruebas de todo el sistema electrónico.

Para ello, se utilizó el programa de Arduino diseñado para la aplicación final (ver Anexo 1), con el fin de determinar su fiabilidad y comportamiento, sabiendo que las partes funcionaban correctamente por separado.

La prueba consistía en mandar ángulos vía Bluetooth y que los motores se desplazasen proporcionalmente al ángulo indicado en función de los valores que se diesen a las variables modificables del programa. El funcionamiento del programa queda recogido en el apartado 3.2.2.1 de la Metodología.

Los ángulos se enviaban por Bluetooth usando una aplicación para Android llamada “Arduino Stepper Motor”, cuya interfaz se muestra en la Figura 5.5.

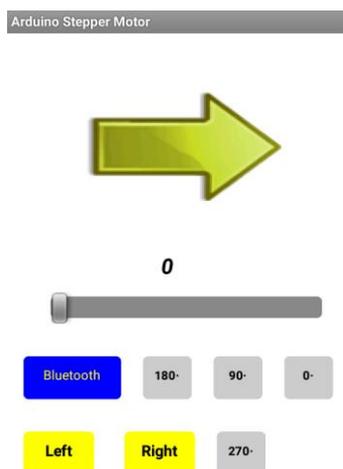


Figura 5.5: Interfaz de la aplicación Arduino Stepper Motor

Así pues, se mandaban los valores de los ángulos y se comprobaba el movimiento realizado usando las mismas referencias que en el caso de los motores y midiendo el ángulo, variando el nivel de dificultad para verificar que cuanto más aumenta este, menor es el desplazamiento de los motores. Ver Tablas 5.3a, 5.3b, 5.3c y 5.3d.

Ángulo mandado desde el origen (°)	Eje	Ángulo efectivo en X (°)	Ángulo efectivo en Y (°)
0	X	0	-
0	Y	-	0
4	X	0	-
4	Y	-	0
5	X	10,5	-
5	Y	-	10,5
10	X	21	-
10	Y	-	21
15	X	31,5	-
15	Y	-	31,5
20	X	42	-
20	Y	-	42
25	X	42	-
25	Y	-	42

Tabla 5.3a: Primer nivel de dificultad

Ángulo mandado desde el origen (°)	Eje	Ángulo efectivo en X (°)	Ángulo efectivo en Y (°)
0	X	0	-
0	Y	-	0
4	X	0	-
4	Y	-	0
5	X	9,6	-
5	Y	-	9,6
10	X	19,1	-
10	Y	-	19,1
15	X	28,6	-
15	Y	-	28,6
20	X	38,2	-
20	Y	-	38,2
25	X	38,2	-
25	Y	-	38,2

Tabla 5.3b: Segundo nivel de dificultad

Ángulo mandado desde el origen (°)	Eje	Ángulo efectivo en X (°)	Ángulo efectivo en Y (°)
0	X	0	-
0	Y	-	0
4	X	0	-
4	Y	-	0
5	X	8,8	-
5	Y	-	8,8
10	X	17,5	-
10	Y	-	17,5
15	X	26,3	-
15	Y	-	26,3
20	X	35	-
20	Y	-	35
25	X	35	-
25	Y	-	35

Tabla 5.3c: Tercer nivel de dificultad

Ángulo mandado desde el origen (°)	Eje	Ángulo efectivo en X (°)	Ángulo efectivo en Y (°)
0	X	0	-
0	Y	-	0
4	X	0	-
4	Y	-	0
5	X	8	-
5	Y	-	8
10	X	16,2	-
10	Y	-	16,2
15	X	24,2	-
15	Y	-	24,2
20	X	32,3	-
20	Y	-	32,3
25	X	32,3	-
25	Y	-	32,3

Tabla 5.3d: Cuarto nivel de dificultad

Tras cada dato, se devolvía a los motores a su posición inicial pulsando “0” para cada eje, y observando que avanzaban en sentido opuesto al que habían llevado.

5.1.2. Pruebas de simulación

Las pruebas de simulación, muy parecidas en concepto a las descritas en el apartado anterior, se realizaron para comprobar el desplazamiento de la plataforma que va encajada sobre el joystick cuando se le daban unos determinados ángulos a los motores, de modo equivalente a la función que desempeñaría el sensor en la aplicación final.

Para ello, fue utilizada la misma aplicación que en el caso anterior. Se situaron los motores en su posición final y se midió el desplazamiento lineal en cada eje y el tiempo que tardaba en hacer el movimiento según el nivel de dificultad seleccionado para los distintos casos de ángulos dados; como se ve reflejado en las tablas siguientes.

Todos los valores se dan desde la posición de reposo, salvo el último, que se da desde la posición alcanzada con los 22°. Si se diese desde cualquier otra posición avanzaría lo avanzado hasta esa posición en sentido opuesto.

Por su parte, el tiempo se ha obtenido midiendo con un cronómetro el movimiento de los motores, repitiendo entre tres y cinco veces cada medida, y tomando el valor medio, dada la desviación de los resultados por ser un proceso manual.

Ángulo introducido (º)	Eje	Dirección de desplazamiento	Desplazamiento (cm)	Tiempo en realizar el movimiento (s)
4	X	Horizontal derecha	0	0
	Y	Hacia delante	0	0
5	X	Horizontal derecha	0,65	0,38
	Y	Hacia delante	0,65	0,38
10	X	Horizontal derecha	1,30	0,79
	Y	Hacia delante	1,30	0,79
15	X	Horizontal derecha	2,00	1,08
	Y	Hacia delante	2,00	1,08
20	X	Horizontal derecha	2,60	1,21
	Y	Hacia delante	2,60	1,21
22	X	Horizontal derecha	2,60	1,21
	Y	Hacia delante	2,60	1,21
0	X	Horizontal izquierda	2,60	1,21
	Y	Hacia detrás	2,60	1,21

Tabla 5.4a: Datos de desplazamiento de la plataforma para el primer nivel de dificultad

Ángulo introducido (°)	Eje	Dirección de desplazamiento	Desplazamiento (cm)	Tiempo en realizar el movimiento (s)
4	X	Horizontal derecha	0	0
	Y	Hacia delante	0	0
5	X	Horizontal derecha	0,60	0,35
	Y	Hacia delante	0,60	0,35
10	X	Horizontal derecha	1,20	0,73
	Y	Hacia delante	1,20	0,73
15	X	Horizontal derecha	1,85	0,98
	Y	Hacia delante	1,85	0,98
20	X	Horizontal derecha	2,40	1,10
	Y	Hacia delante	2,40	1,10
22	X	Horizontal derecha	2,40	1,10
	Y	Hacia delante	2,40	1,10
0	X	Horizontal izquierda	2,40	1,10
	Y	Hacia detrás	2,40	1,10

Tabla 5.4b: Datos de desplazamiento de la plataforma para el segundo nivel de dificultad

Ángulo introducido (°)	Eje	Dirección de desplazamiento	Desplazamiento (cm)	Tiempo en realizar el movimiento (s)
4	X	Horizontal derecha	0	0
	Y	Hacia delante	0	0
5	X	Horizontal derecha	0,55	0,32
	Y	Hacia delante	0,55	0,32
10	X	Horizontal derecha	1,10	0,66
	Y	Hacia delante	1,10	0,66
15	X	Horizontal derecha	1,70	0,90
	Y	Hacia delante	1,70	0,90
20	X	Horizontal derecha	2,20	1,05
	Y	Hacia delante	2,20	1,05
22	X	Horizontal derecha	2,20	1,05
	Y	Hacia delante	2,20	1,05
0	X	Horizontal izquierda	2,20	1,05
	Y	Hacia detrás	2,20	1,05

Tabla 5.4c: Datos de desplazamiento de la plataforma para el tercer nivel de dificultad

Ángulo introducido (°)	Eje	Dirección de desplazamiento	Desplazamiento (cm)	Tiempo en realizar el movimiento (s)
4	X	Horizontal derecha	0	0
	Y	Hacia delante	0	0
5	X	Horizontal derecha	0,50	0,30
	Y	Hacia delante	0,50	0,30
10	X	Horizontal derecha	1,00	0,61
	Y	Hacia delante	1,00	0,61
15	X	Horizontal derecha	1,55	0,85
	Y	Hacia delante	1,55	0,85
20	X	Horizontal derecha	2,00	1,00
	Y	Hacia delante	2,00	1,00
22	X	Horizontal derecha	2,00	1,00
	Y	Hacia delante	2,00	1,00
0	X	Horizontal izquierda	2,00	1,00
	Y	Hacia detrás	2,00	1,00

Tabla 5.4d: Datos de desplazamiento de la plataforma para el cuarto nivel de dificultad

Duales resultados se obtienen si se dan ángulos negativos; esto es, se desplazan lo mismo pero en sentido contrario.

En la gráfica siguiente se muestran los resultados de los desplazamientos en conjunto para todos los niveles de dificultad. Se observa que conforme el nivel aumenta, el desplazamiento disminuye de manera proporcional para todos los ángulos.

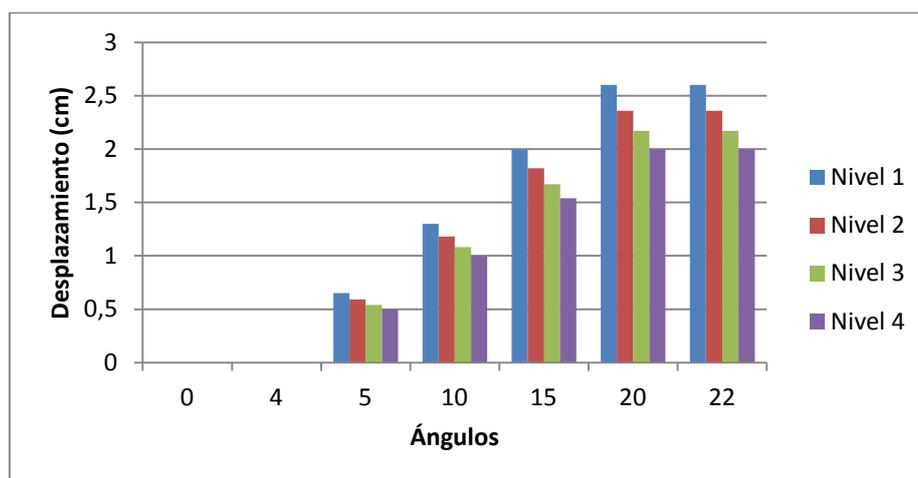


Figura 5.6: Desplazamientos de la plataforma para cada ángulo dado variando el nivel de dificultad

En la Figura 5.6, se representa el tiempo medio de movimiento de los motores para cada nivel de dificultad y cada grado elegido, obviando el cero y ángulos inferiores al mínimo (5°).

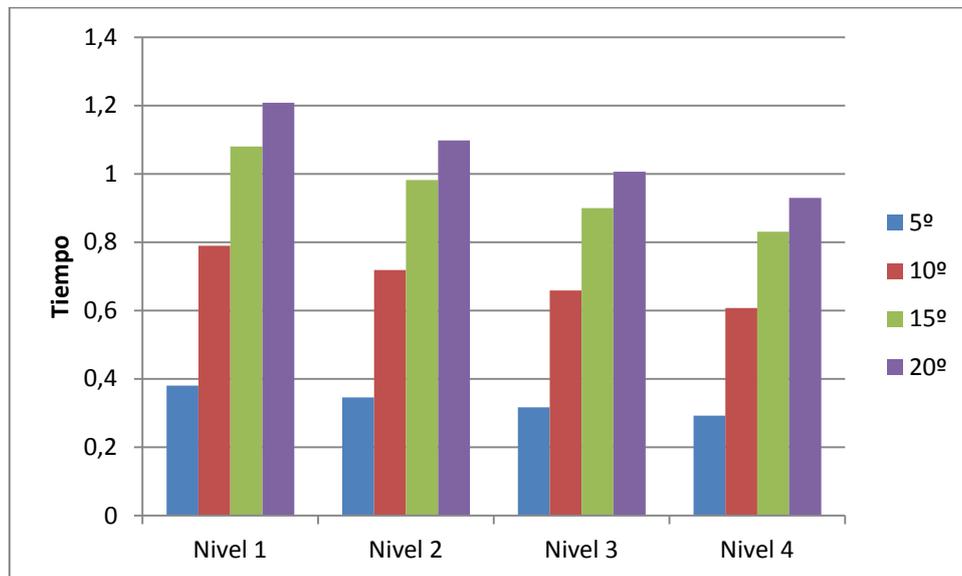


Figura 5.7: Tiempo desplazamiento de los motores según nivel de dificultad y ángulo

5.2. Análisis de resultados y conclusiones

En esta sección se analizan los resultados de las distintas pruebas para, posteriormente, sacar las conclusiones de ellos derivadas.

5.2.1. Análisis y conclusiones de las pruebas del Bluetooth

En la fase preliminar, el comportamiento del módulo HC-06 se corresponde con el esperado, de lo que se deduce que el sistema está bien conectado y funciona correctamente; por lo que puede ser integrado con el resto del sistema electrónico.

5.2.2. Análisis y conclusiones de las pruebas de los motores

Respecto al movimiento esperado de los motores cabe destacar el hecho de que los doscientos pasos no se correspondiesen con una vuelta. Esto ayudó a corregir el factor de reducción mediante el uso de una constante. Cualitativamente, el resto de los valores cumplían lo predecible. Cuantitativamente, solo hacía falta añadir dicha constante para compensar la reducción y conseguir que se moviese exactamente lo deseado.

Respecto al tiempo de movimiento, se observa una pequeña variabilidad en los datos, cuando siempre deberían salir los mismos valores para el movimiento en un sentido y en otro; esto se debe a contabilizar el tiempo manualmente. No obstante, el margen de variación es pequeño por lo que dichos valores se pueden considerar válidos.

Siendo así, hay un fenómeno que resulta aparentemente ilógico. Los valores para 500 pasos no son 2,5 veces mayores que para los de 200 (ver Tabla 5.2). Esto se explica porque los motores tienen una cierta aceleración, de manera que al dar más pasos tienen más tiempo para incrementar su velocidad tardando proporcionalmente menos de lo que deberían. Igualmente ocurre entre 500 y 1000 pasos.

5.2.3. Análisis y conclusiones de las pruebas del conjunto integrado

Las pruebas del Bluetooth y los motores juntos sugieren que el sistema funciona como se esperaba. Es decir, el desplazamiento en ambos ejes para el mismo valor era prácticamente el mismo. Si se daba un valor desde el origen por debajo del mínimo, los motores no se movían. Sin embargo, cuando se daba cero o un valor por debajo del mínimo desde una posición distinta al origen, los motores regresaban a su origen. El desplazamiento es proporcional al ángulo dado, y va incrementándose con este, hasta que por encima del máximo toma este valor desplazándose lo mismo, tal y como implementaba el programa y queda recogido en las tablas 5.3.

También se aprecia que con cada nivel de dificultad el ángulo desplazado va disminuyendo.

Finalmente, es necesario comentar que la discrepancia entre el ángulo enviado y el ángulo efectivo desplazado se debe a la inclusión de un factor que controla que el desplazamiento de los engranajes no se salga del límite físico de la cremallera, como se explica en el apartado 3.2.2.1 de la Metodología.

5.2.4. Análisis y conclusiones de las pruebas de simulación

Los resultados de las pruebas de simulación recogidos en las Tablas 5.4 hacen llegar a conclusiones muy parecidas a las del caso precedente y que corroboran el correcto funcionamiento del sistema:

Por debajo del mínimo, la plataforma no se desplaza si se encuentran en su posición de equilibrio. Si no lo está, vuelve al origen.

Para los mismos ángulos, el desplazamiento viene a ser el mismo en cada eje.

Según se incrementa el ángulo, la plataforma se desplaza más, proporcionalmente. No así ocurre con el tiempo, donde, por ejemplo, los valores para ángulo 20 no son el doble que para ángulo 10, por entrar en juego la aceleración.

Las gráficas de las Figuras 5.6 y 5.7 corroboran la disminución lineal del desplazamiento y del tiempo, como consecuencia del primero, según se incrementa el nivel de dificultad establecido por código.

6. APLICACIONES E IMPACTOS

En este apartado se va a realizar un análisis de las principales aplicaciones e impactos del presente proyecto en los múltiples ámbitos de influencia.

6.1. Aplicaciones en medicina

Como ya se indicó en apartados anteriores, este proyecto surge de la conveniencia detectada por la fundación ASPACE y la empresa Werium Assistive Solutions de que los niños con parálisis cerebral ejerciten los músculos cervicales, de forma que puedan controlar la silla de ruedas sin necesidad de usar las extremidades superiores, que en muchos casos se encuentran inoperantes.

Esta aplicación concreta constituye al mismo tiempo un entrenamiento que sirve de rehabilitación y prevención de futuras patologías derivadas de la incorrecta postura que suelen presentar, como pueden ser dificultad para respirar y deglutir, aumento del ritmo cardíaco o alteración del tono muscular normal.

Sin embargo, aunque la idea original surgió a raíz de la observación de niños con parálisis, se puede aplicar igualmente a adultos, con iguales efectos beneficiosos que los mencionados anteriormente, frutos de un mismo entrenamiento continuo y periódico en presencia de un médico o terapeuta.

Por otra parte, pensando en la rehabilitación cervical, surgen aplicaciones para otras patologías y enfermedades que cursen con hipotonía cervical. Así pues, una persona que ha sufrido un ictus y ha perdido movilidad de la cabeza puede utilizar el dispositivo para, de forma progresiva, ir recuperando el tono muscular que le permita volver a mover el cuello.

Igualmente, los músculos del cuello pueden ver mermada su capacidad de movimiento por distintas causas como enfermedades neurodegenerativas, accidentes de tráfico o lesiones medulares. Por lo que el empleo de este sistema ayudaría a mejorar su situación, bien ralentizando la degeneración o bien recuperando la movilidad.

Asimismo, se puede usar como indicador del progreso del paciente en cuanto a capacidad de mover el cuello, ya que el sistema cuenta con distintos niveles configurables que permiten mover la silla haciendo menos o más esfuerzo en función de la movilidad que estos presenten.

6.2. Otras aplicaciones

El sistema diseñado puede utilizarse también para otras aplicaciones no exclusivamente médicas como videojuegos, sillas de limpieza, teleoperación.

Estas otras aplicaciones se pueden clasificar en dos tipos: directamente derivadas del uso principal del sistema o no relacionadas con el propósito fundamental de éste.

En el primer tipo, un ejemplo de aplicación sería el uso de este sistema en sillas o carros automáticos de limpieza. De forma que una persona con discapacidad, principalmente parálisis, pudiese integrarse en el mercado laboral haciendo este tipo de tareas recorriendo una superficie para su limpieza mediante el movimiento de la cabeza. Si bien es verdad que necesitaría de la adaptación del asiento del carro para poder operar desde una posición cómoda y segura. Por ello, surge la alternativa de en lugar de modificar el carro (asiento y joystick), añadir simplemente a la silla el sistema de limpieza, mopa, escoba o semejante.

En el segundo tipo, más alejado del objetivo básico del proyecto, se pueden encontrar algunas aplicaciones como videojuegos para gente con parálisis. Esto sería de la siguiente manera: el sistema se coloca sobre el joystick de juego y del mismo modo que con la silla, el usuario maneja el joystick, controlando el juego, que debería estar adaptado para él y podría tener fines educativos o terapéuticos.

Una distinta aplicación sería la teleoperación. Aunque ya existen métodos avanzados de teleoperación podría utilizarse para mover objetos con ruedas a distancia usando la cabeza o incluso otra parte del cuerpo si el sensor captador se coloca en otro lugar, por ejemplo la mano.

Siguiendo en esta línea, podrían hallarse de nuevo fines terapéuticos, pues si el sensor se sitúa en la mano o en el pie, se puede ver la movilidad de las articulaciones mediante el control remoto de un sistema con ruedas, por ejemplo una plataforma de pequeño tamaño, información que puede resultar útil en el caso de lesiones y problemas de huesos o articulaciones.

6.3. Impactos

A continuación, se procede a realizar un análisis de los impactos más relevantes del proyecto en su conjunto en función de los grupos de interés implicados y de la aplicación del sistema.

6.3.1. Impacto sanitario

El principal impacto que puede apreciarse es en el ámbito sanitario, como consecuencia directa de la aplicación del dispositivo.

Este impacto radica en la mejora de la movilidad cervical de los pacientes evitando al mismo tiempo una degeneración muscular que derive en problemas graves de salud, como los detallados en la Introducción de este trabajo.

Actualmente, el tratamiento para la parálisis consiste en recuperar las capacidades que sean posibles; en caso de no poder, habilitar nuevas y mantener las capacidades existentes. Este proyecto cumple con eso, centrándose sobre todo en recuperar y mantener, mediante el ejercicio continuo del cuello que implica el control de la silla.

Esto último pone de manifiesto otro impacto relevante y a la vez un rasgo diferenciador de los tratamientos habituales: el uso de la silla de ruedas eléctrica como elemento de rehabilitación. La silla de ruedas eléctrica no está pensada sino como un producto de apoyo (al igual que la manual); por eso, al darle ese valor añadido se aprovechan mejor los recursos con los que puede contar una familia para con hijos con parálisis, como se explica en el siguiente apartado.

Adicionalmente, hay otro impacto a destacar que afecta a los terapeutas y médicos que tratan con parálisis cerebral. Mediante el empleo del sistema se facilita el control del entrenamiento y los ejercicios, sin necesidad de tener conocimientos especializados de programación, y se permite llevar una observación más cuantificable del progreso en la movilidad del paciente por medio de los distintos niveles de control programados.

6.3.2. Impacto económico

Este aspecto alude principalmente a la disminución del coste que supone frente a otras alternativas semejantes.

El sistema desde su diseño está concebido para ser algo que sea económico. Ahí es donde reside el impacto. Otros productos sirven también para mover una silla de ruedas con la cabeza, pero son mucho más caros por la tecnología, la producción y la normativa a cumplir que implican. Este, por el contrario, es un dispositivo sencillo, fácil de usar, adaptable a una silla de ruedas convencional, y barato. Esto hace que las familias no tengan que invertir un gran capital en productos de rehabilitación para sus hijos, o en sillas de ruedas especialmente adaptadas.

De igual modo ocurre para los centros médicos donde se trabaja con estas personas. El adquirir este dispositivo frente a otro elemento de rehabilitación es mucho más ventajoso, porque, por un precio mucho menor, se pueden obtener incluso varios, facilitando así el llevar a más personas los beneficios de este sistema.

6.3.3. Impacto social

El impacto social se deriva directamente del beneficio sanitario de la aplicación del producto. Éste es otro de los impactos relevantes del proyecto y uno de los que más se puede llegar a apreciar por lo que se detalla en lo que sigue.

La rehabilitación cervical no sólo tiene por objetivo recuperar la movilidad del cuello o ralentizar su degeneración; también persigue fortalecer dichos músculos para conseguir una mejora postural que repercuta positivamente en el paciente y su entorno.

Al conseguir un adecuado tono cervical, la cabeza no permanece caída ni el tronco hiperextendido mirando al techo. Con esto se logra una mejor respiración, mejor deglución, menor esfuerzo cardíaco y en conclusión, mejor estado de salud.

El tener la cabeza erguida facilita la relación con el entorno, mejora la comunicación corporal, visual y la emisión de voz fomentando las relaciones sociales. Asimismo, da un aspecto más agradable que puede repercutir en que la persona reciba comentarios positivos mejorando su autoestima y bienestar.

Además, el hecho de que el usuario del dispositivo sea capaz de desplazar por sí mismo la silla de ruedas, esto es, que sea capaz de “conducir su silla”, hace que se sienta más autónomo, incrementa también su autoestima y favorece su independencia.

A su vez, esto genera en su entorno un ambiente de optimismo y alegría al ver a su familiar o amigo en un estado emocional equilibrado, lo que promueve una actitud positiva de la que a su vez se beneficiará el paciente.

Otro impacto social es que los responsables del acompañamiento y supervisión de gente con parálisis pueden contar con un elemento adicional para realizar los tratamientos facilitando su desempeño profesional.

6.3.4. Impacto ambiental

El impacto ambiental del sistema pasa por considerar la vida del producto en su conjunto para ver dónde es necesario hacer más hincapié para mejorar.

Tanto el dispositivo de control como la estructura son reutilizables múltiples veces (el arduino es reconfigurable) y adaptables a cualquier silla de ruedas eléctrica convencional.

La parte que puede presentar un impacto más negativo es la alimentación del arduino y de los motores, pero ésta se va a realizar con pilas o baterías recargables que prolongan su vida útil frente a las no recargables.

7. LÍNEAS FUTURAS

En este apartado se exponen las posibles líneas futuras de investigación y de mejora aplicables a este proyecto.

Alguna de esas mejoras o líneas de investigación se han identificado en el desarrollo de alguna de las etapas del proyecto. Otras han surgido de la reflexión para mejorar y hacer avanzar este trabajo.

En lo que sigue, se explican dichas líneas de investigación y posibles mejoras para este proyecto.

- Adaptación a otros joysticks

Se trataría de hacer un elemento que se pudiese encajar en otros tipos de joystick para poder ampliar la aplicabilidad del dispositivo. El principio de funcionamiento sería el mismo y el diseño muy parecido. Se trataría de una estructura adaptable en la que la parte que se agarra al joystick fuese retráctil para poder abarcar distintos tamaños. Se podría dividir la estructura por la mitad con unas uniones elásticas para ajustarse al tamaño del joystick. Las ruedas dentadas accionadas por los motores engranarían por la parte de abajo, igual que en el caso del proyecto.

- Interfaz móvil

Con el fin de poder configurar el dispositivo de forma más intuitiva sin tener que modificar el código se podría desarrollar una interfaz gráfica sencilla. Esta podría tener como parámetros: la selección del plano de donde se quieren recibir los ángulos, el nivel de dificultad o entrenamiento, el ángulo máximo y mínimo de funcionamiento y una funcionalidad para activar o desactivar el sistema desde la interfaz para evitar posibles accidentes. Esto se podría realizar también mediante una aplicación para móvil que se comunicase por Bluetooth con el dispositivo.

- Aumento de la autonomía

Se podría aumentar la autonomía del sistema conectándolo a la batería de la propia silla. Bastaría con la introducción adaptador para poder alimentar con el voltaje y corriente adecuado al Arduino y a los motores. De esta forma se podría usar el sistema más tiempo y de manera más continuada.

- Guardado de datos del sensor

Una funcionalidad adicional sería el guardado de los datos procedentes del sensor. Esto podría hacerse mediante código guardando los ángulos de cada movimiento efectivo en un archivo. Con esta información los terapeutas podrían analizar posteriormente y de manera objetiva la capacidad de movimiento del cuello de los usuarios, cómo varía a lo largo de la sesión y el progreso de la movilidad.

- Algoritmo adaptativo

El algoritmo adaptativo consistiría en un programa que pudiese modificarse automáticamente durante su ejecución para que en función de la dificultad que encontrase el usuario para mover la silla, se modificasen los umbrales que permitiesen el movimiento. Sería parecido al programa del proyecto, solo que el nivel de dificultad se seleccionaría según el rango (máximo y mínimo) de los ángulos de movimiento de la cabeza que la persona fuese capaz de hacer en un periodo de tiempo determinado. De esta forma cuanto más fatiga tuviese menor sería el nivel de dificultad y, por tanto, más bajos los niveles para producir el movimiento.

- Diseño de carcasas de protección

Dado que se trata de un prototipo el arduino y el resto de elementos quedan sin excesiva protección aunque se encuentren dentro de la estructura. Por ello, sería recomendable cubrirlos por la parte superior con algún tipo de carcasa para evitar posibles daños. Dichas protecciones se podrían diseñar usando FreeCad e imprimiéndolas con impresoras 3D.

- Mejora del cableado

Se podría reducir la presencia de los cables realizando soldaduras permanentes, evitando así malas conexiones. Asimismo, se deberían aislar con recubrimientos plásticos y sujetar, para reducir peligros tanto físicos como materiales.

- Mejoras en el programa

Se puede optimizar el diseño del programa haciéndolo más modular, es decir, con más funciones, pero más sencillas que facilitasen su interpretación y mejora o ampliación.

8. PLANIFICACIÓN TEMPORAL Y PRESUPUESTO

En este capítulo se explicará la planificación temporal del proyecto, detallando las horas dedicadas a cada tarea y el correspondiente diagrama de Gantt con la distribución temporal de dichas tareas.

Asimismo, se especifica el presupuesto del proyecto con los costes asociados a su realización.

8.1. Planificación temporal

El proyecto comenzó en noviembre de 2016 con la selección y asignación del mismo, pero no comenzó de facto hasta febrero de 2017; finalizando en febrero de 2018 con la defensa ante tribunal.

El motivo del retardo entre la asignación y el comienzo fue la carga lectiva y la premura por poder aprobar todas las asignaturas del grado en dicho curso para poder acceder al máster en el año siguiente. Por ello, se decidió en aquel momento como fecha de presentación el mes de febrero de 2018. Puesto que de esa forma se podían simultanear ambas cosas pero priorizando las asignaturas.

Desde entonces se ha trabajado de forma más o menos continua en el proyecto, intensificando el trabajo en periodos de vacaciones (verano y Navidad 2017) y disminuyéndolo en los periodos de examen pero manteniéndolo, es decir, se le dedicaban menos horas pero siempre se hacía alguna tarea del proyecto.

En primer lugar, se llevó a cabo la documentación del tema, empezando por las características del joystick, y siguiendo por la lectura de múltiples textos informativos y estudios acerca de la hipotonía cervical, la falta de control motor y los problemas que producen para profundizar en el impacto y las ventajas del proyecto. También se recopiló y analizó información sobre la parálisis cerebral y el estado del arte. Paralelamente, se iban realizando los primeros apartados de la memoria.

A continuación, se buscaron y seleccionaron los materiales atendiendo a características y vigilando que cumpliesen con el objetivo de bajo coste.

Entonces se comenzó el diseño de las piezas de la estructura, previo estudio de las opciones de movimiento en 2D. Al mismo tiempo, se continuaba la redacción de la memoria.

Una vez diseñadas las piezas que constituyen la estructura, se procedió a comenzar con la programación, primero programando los motores y probándolos, para después

programar el Bluetooth y probarlo. Tras esto, se realizó la integración de ambas partes, realizando pruebas preliminares.

Cuando el programa estaba acabado, se llevó a cabo el montaje y una fase de pruebas para conocer el comportamiento del sistema.

Finalmente, se concluyó la memoria para posteriormente ser revisada y entregada.

NOMBRE DE TAREA	FECHA DE INICIO	FECHA DE FIN	DURACIÓN	HORAS	DEPENDENCIAS
TRABAJO DE FIN DE GRADO	13/11/2016	09/02/2018	454	378	
0. Asignación	13/11/2016	13/11/2016	0		
1. Análisis inicial	06/02/2017	02/04/2017	56	40	
1.1. Primer contacto con el tutor	06/02/2017	06/02/2017	0	-	
1.2. Primera reunión informativa	15/02/2017	15/02/2017	0	2	1.1
1.3. Estudio de la hipotonía cervical	16/02/2017	01/03/2017	14	14	1.2
1.4. Estudio de la parálisis cerebral	20/02/2017	10/03/2017	19	12	1.2
1.5. Estudio de los tipos de joystick y sistemas de movilidad	22/02/2017	30/03/2017	37	8	1.2
1.6. Estudio de plataformas de movimiento en 2D	31/03/2017	02/04/2017	3	4	1.2
2. Elección y compra de los materiales	09/04/2017	06/05/2017	28	14	
2.1. Búsqueda del material necesario	09/04/2017	12/04/2017	4	8	1.2, 1.5
2.2. Elección de los componentes	12/04/2017	18/04/2017	7	4	2.1
2.3. Recogida del material	03/05/2017	03/05/2017	0	-	2.2
2.4. Familiarización con los componentes	04/05/2017	06/05/2017	3	2	2.3
3. Diseño de la estructura	28/07/2017	25/11/2017	121	45	
3.1. Alternativas de diseño y boceto inicial	28/07/2017	31/07/2017	4	6	1.5
3.2. Familiarización con FreeCad	02/08/2017	03/08/2017	2	2	
3.3. Diseño definitivo estructura	31/07/2017	01/08/2017	2	4	3.1, 3.2
3.4. Diseño en FreeCad	04/08/2017	24/08/2017	21	25	3.3
3.5. Revisión	15/10/2017	19/10/2017	5	8	3.3, 3.4
3.6. Impresión en 3D	22/11/2017	24/11/2017	3	-	3.5
4. Programación con Arduino	22/09/2017	04/01/2018	105	96	
4.1. Descarga de librerías y programa	22/09/2017	23/09/2017	2	2	
4.2. Programa para	01/10/2017	22/11/2017	53	44	4.1

mover los motores					
4.3. Programa para probar el Bluetooth	25/11/2017	30/11/2017	6	10	4.1
4.4. Programa integrado Bluetooth-motores	23/12/2017	30/12/2017	8	30	4.2, 4.3
4.5. Revisión y preparación para simulación	30/12/2017	04/01/2018	6	10	4.4
5. Montaje del sistema	01/10/2017	20/01/2018	112	15	
5.1. Montaje electrónico	01/10/2017	15/10/2017	15	10	4.4
5.2. Montaje de la estructura	17/01/2018	20/01/2018	4	5	3.6
6. Fase de pruebas	05/01/2018	26/01/2018	22	20	
6.1. Pruebas del sistema electrónico	05/01/2018	10/01/2018	6	8	4.5, 5.1
6.2. Pruebas de simulación	24/01/2018	26/01/2018	3	12	6.1, 5.2
7. Redacción de la memoria	10/03/2017	03/02/2018	331	150	
7.1. Introducción	10/03/2017	20/04/2017	42	42	1.2, 1.3, 1.4
7.2. Objetivos	10/07/2017	11/07/2017	2	2	1.2
7.3. Metodología	21/04/2017	20/05/2017	30	50	1.2, 2.2,3.3, 4.4
7.4. Descripción del dispositivo. Alternativas consideradas	12/07/2017	19/07/2017	8	20	1.2, 1.5, 2.1, 3.1
7.5. Pruebas, resultados y conclusiones	25/01/2018	30/01/2018	6	10	6.1, 6.2
7.6. Aplicaciones e impactos	29/12/2017	30/12/2017	2	4	1.2, 1.3, 1.4
7.7. Líneas futuras	02/01/2018	03/01/2018	2	2	1.2, 1.5, 2.1, 3.1, 4.5
7.8. Planificación temporal y presupuesto	25/01/2018	29/01/2018	5	10	1.2, 2.1, 3.3, 3.6, 5
7.9. Resumen	30/01/2018	31/01/2018	2	3	7.1, 7.2
7.10. Revisión y correcciones	07/01/2017	31/01/2018	25	5	
8. Entrega del TFG	05/02/2018	23/02/2018	18	0	
8.1. Entrega en Indusnet	05/02/2018	05/02/2018	0	0	
8.2. Entrega en secretaría	09/02/2018	09/02/2018	0	0	

Tabla 8.1: Estructura de descomposición del proyecto (EDP)

En la tabla se incluyen los días de inicio y de fin en el cómputo de días. Las horas están basadas en anotaciones propias que se hicieron para llevar el seguimiento del proyecto.

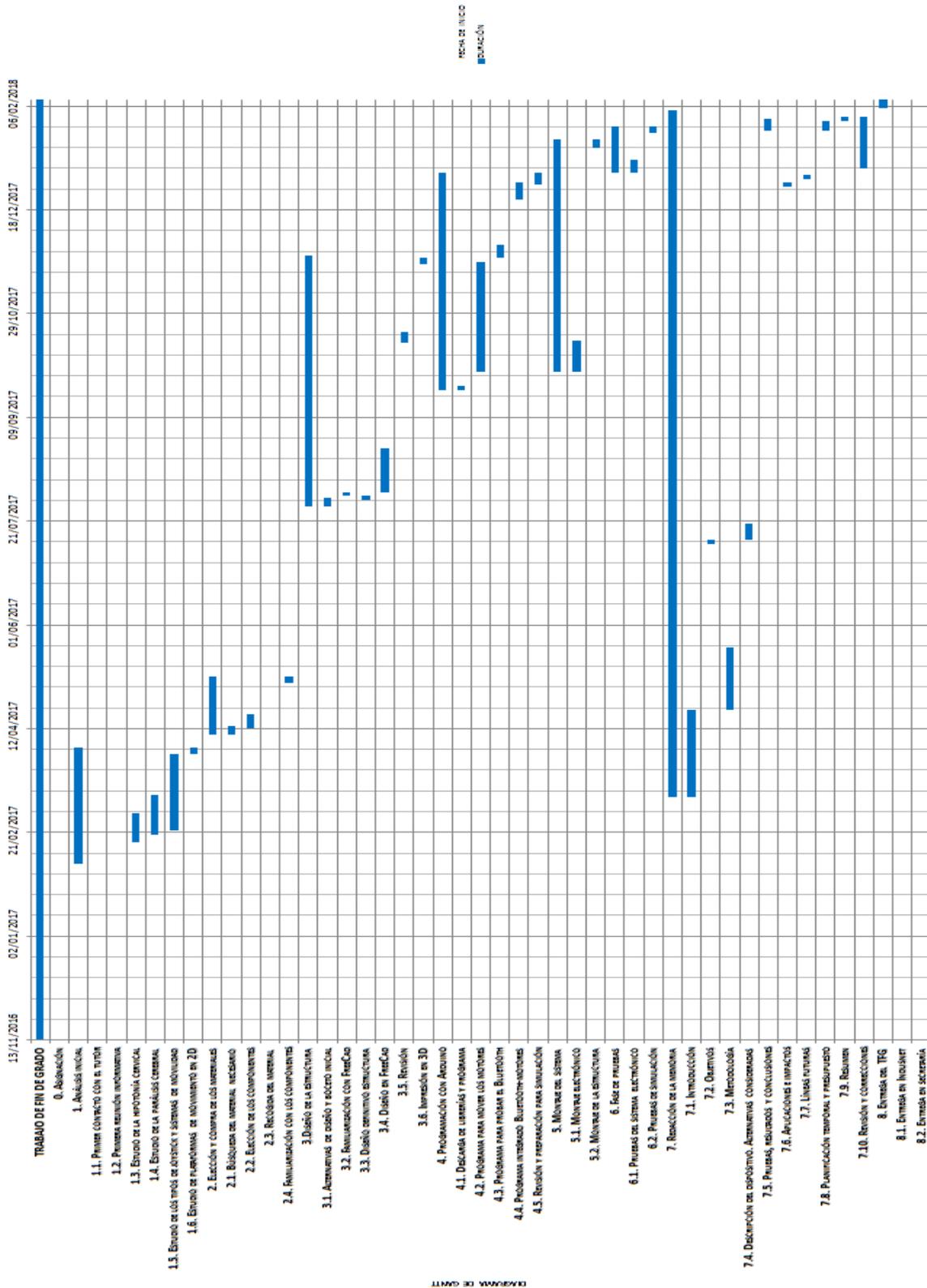


Figura 8.1: Diagrama de Gantt

8.2. Presupuesto

A continuación se van a detallar los costes materiales, de software y los derivados del trabajo personal.

En la tabla 8.2 aparecen los costes asociados al material, cuya elección se justificó en el apartado de Metodología.

Producto	Unidades	Precio/unidad	Precio total
Arduino Uno Rev 3	1	19,89 €	19,89 €
Cables de conexión cortos	1	1,85 €	1,85 €
Cables de conexión largos	1	3,10 €	3,1 €
Cargador pilas	1	8,00 €	8,00 €
Conector jack hembra	1	1,27 €	1,27 €
EasyDriver A3967	2	4,50 €	9,00 €
Filamento impresora 3D	300 g	19,90 €/kg	5,97 €
Fuente alimentación 12V 1ª	1	7,26 €	7,26 €
Fuente alimentación Arduino	1	28,21 €	28,21 €
Módulo Bluetooth HC-06	1	8,65 €	8,65 €
Nema 17	2	15,40 €	30,8 €
Palos de madera	8	0,016 €	0,13 €
Paquete de 4 pilas recargables	3	14,38 €	43,14 €
Paquete tuercas ciegas 20 ud	1	1,75 €	1,75 €
Paquete tuercas pasantes 20 ud	1	1,75 €	1,75 €
Pegamento para madera	1	3,90 €	3,90 €
Portapilas	3	0,50 €	1,5 €
Protoboard	1	6,12 €	6,12 €
Resistencia 470 ohm	1	0,25 €	0,25 €
Resistencia 680 ohm	1	0,25 €	0,25 €
Varilla metal roscada 1 m	1	0,35 €	0,35 €
TOTAL			183,14 €

Tabla 8.2: Costes asociados al material

También hay que considerar en los costes el software utilizado, excluyendo los programas gratuitos como Arduino y Fritzing. Los datos se exponen en la siguiente tabla:

Software	Unidades	Precio/unidad	Precio total
Microsoft Office 2007	1	19,90 €	19,90 €
Windows 7 Home Premium	1	19,99 €	19,99 €
Microsoft Visio 2007	1	125 €	125 €
TOTAL			164,89 €

Tabla 8.3: Costes asociados al software utilizado

Otro aspecto a considerar es la amortización de los elementos utilizados. Los elementos a amortizar son el ordenador y la impresora en su parte proporcional al uso, de ahí el factor de ponderación.

El método empleado para calcular la amortización es el método lineal tomando valor residual nulo, válido según el Plan General de Contabilidad. Para un equipo de proceso de información se establece una amortización entre a 4 y a 8 años, por lo que se han considerado 6 años como tiempo razonable.

Concepto	Tiempo de amortización	Precio compra	Cuota mensual	Duración uso en proyecto	Ponderación	Coste a imputar
Ordenador	72 meses	690 €	9,58 €	12 meses	0,75	86,25 €
Impresora 3D	48 meses	1200 €	25 €	1 mes	0,2	5 €
TOTAL						91,25 €

Tabla 8.4: Costes de amortización

En cuanto al coste de la electricidad, intervienen varios factores: la impresora 3D, el ordenador, la iluminación y la alimentación de los motores durante la fase de pruebas.

Concepto	Potencia	Horas	Energía	Precio kWh	Coste a imputar
Ordenador	50 W	357	17,85 kWh	0,117988 €	2,11 €
Impresora 3D	150 W	72	10,8 kWh	0,117988 €	1,274 €
Calentador impresora	40 W	72	2,88 kWh	0,117988 €	0,34 €
Alimentación motores	12 W	90	1,08 kWh	0,117988 €	0,127 €
Iluminación	20 W	170	3,4 kWh	0,117988 €	0,40 €
TOTAL					4,25 €

Tabla 8.5: Costes de electricidad

Finalmente, los costes de los recursos humanos se exponen en la siguiente tabla. Se ha considerado un salario de ingeniero de 20 €/h, y los honorarios del director del proyecto se han imputado a 50 €/h.

Concepto	Coste horario	Horas	Coste a imputar
Ingeniero	20 €/h	378	7560 €
Director de proyecto	50 €/h	70	3500 €
TOTAL			10060 €

Tabla 8.6: Coste de recursos humanos

En conjunto, todas las partidas del proyecto suman **10503,53 €**.

Concepto	Coste
Materiales	183,14 €
Software	164,89 €
Amortización	91,25 €
Consumo eléctrico	4,25 €
Recursos humanos	10060 €
TOTAL	10503,53 €

Tabla 8.7: Costes totales del proyecto

Si se considerasen exclusivamente los costes de material para una unidad en su estado final quedaría un precio total de **165,39 €**:

Producto	Unidades	Precio/unidad	Precio total
Arduino Uno Rev 3	1	19,89 €	19,89 €
Cables de conexión cortos	1	1,85 €	1,85 €
Cables de conexión largos	1	3,10 €	3,1 €
Cargador pilas	1	8,00 €	8,00 €
Conector jack hembra	1	1,27 €	1,27 €
EasyDriver A3967	2	4,50 €	9,00 €
Filamento impresora 3D	300 g	19,90 €/kg	5,97 €
Fuente alimentación Arduino	1	28,21 €	28,21 €
Módulo Bluetooth HC-06	1	8,65 €	8,65 €
Nema 17	2	15,40 €	30,8 €
Palos de madera	8	0,016 €	0,13 €
Pilas recargables	10	3,60 €	36,00 €
Pegamento para madera	1/2	3,90 €	1,95 €
Portapilas	3	0,50 €	1,5 €
Protoboard	1	6,12 €	6,12 €
Resistencia 470 ohm	1	0,25 €	0,25 €
Resistencia 680 ohm	1	0,25 €	0,25 €
Tuercas ciegas	4	0,0875 €	0,35 €
Tuercas pasantes	20	0,0875 €	1,75 €
Varilla metal roscada 1 m	1	0,35 €	0,35 €
TOTAL			165,39 €

Tabla 8.8: Costes de materia para una unidad

BIBLIOGRAFÍA

Control postural

Saavedra, S., Woollacott, M., van Donkelaar, P.

“Head stability during quiet sitting in children with cerebral palsy: effect of vision and trunk support”.

Exp Brain Res, 2010

Fitzsimmons, L.

“Positioning the head: strategies to improve head control and posture”.

Clinical corner, pp. 42-45, Directions Vol. 2014.1

Parálisis cerebral

<https://medlineplus.gov/spanish/cerebralpalsy.html>

<http://www.aspace.org>

<http://www.aspace.org/pagina/18/tipos-de-paralisis-cerebral#>

<http://www.msmanuals.com/es-es/hogar/enfermedades-cerebrales,-medulares-y-nerviosas/trastornos-del-movimiento/corea,-atetosis-y-hemibalismo>

<https://medlineplus.gov/spanish/ency/article/003297.htm>

<http://paracereb.blogspot.com.es/>

<https://neuropediatra.org/2015/03/04/tipos-de-paralisis-cerebral-infantil/>

<https://www.cerebralpalsy.org.au/what-is-cerebral-palsy/>

<https://www.cerebralpalsy.org.au/what-is-cerebral-palsy/types-of-cerebral-palsy/dyskinetic-cerebral-palsy/>

<https://aspace.org/algunos-datos>

<https://www.fundacionaindace.org/sobre-nosotros/algunas-cifras/>

Daños cervicales por accidentes de tráfico

<http://www.dgt.es/revista/archivo/pdf/num173-2005-Latigazo.pdf>

Joysticks y sistemas de movilidad

http://www.ceapat.es/ceapat_01/index.htm

http://www.ceapat.es/ceapat_01/cat_apo/catalogo/index.htm

<http://www.ceapat.es/interpresent3/groups/imsero/documents/binario/sillasderuedas.pdf>

<http://proyectosidesim.blogspot.com.es/>

http://eao.com/fileadmin/documents/PDFs/es/03_brochures/EAO_PB_09_Joysticks_ES.pdf

<http://www.cw-industrialgroup.com/Products/Mobility-Vehicle-Solutions/VR2.aspx>

Sistemas de movilidad para gente con discapacidad

<http://www.sunrisemedical.es/sillas-de-ruedas/quickie/sillas-de-ruedas-electricas#mostPopular>

<http://www.sunrisemedical.es/sillas-de-ruedas/rgk/sillas-deportivas>

<http://www.sunrisemedical.es/sillas-de-ruedas/quickie/sillas-de-ruedas-electricas>

<http://www.sunrisemedical.es/scooters/sterling/scooters-electricos>

http://www.invacare.ca/cgi-bin/imhqprd/inv_catalog/prod_cat.jsp?s=0&catOID=-536885238

<https://www.ortoweb.com/movilidad/sillas-de-ruedas-electricas>

Dispositivo de control

<https://www.raspberrypi.org/help/faqs/>

<http://www.tresdprinttech.com/pasos-configurar-bluetooth-raspberry-pi-3/>

<https://raspberryparatorpes.net/hardware/sobre-la-raspberry-pi-3/>

https://es.wikipedia.org/wiki/Raspberry_Pi

<https://www.profesionalreview.com/2016/03/10/los-mejores-usos-para-raspberry-pi/>

<http://www.prometec.net/indice-tutoriales>

<http://hacedores.com/arduino-o-raspberry-pi-cual-es-la-mejor-herramienta-para-ti/>

<https://www.arduino.cc/en/Guide/HomePage>

<https://www.luisllamas.es/que-es-arduino-que-modelo-comprar/>

<http://www.digikey.es/product-detail/es>

<http://www.prometec.net/consumos-arduino/>

<https://martingonzalezproyectos.wordpress.com/arduino-2/explicacion-pines-arduino-uno/>

<http://icircuit.net/arduino-boards-pin-mapping/141>

<http://www.fabricadigital.org>

<http://www.airspayce.com/mikem/arduino/AccelStepper/index.html>

Mecanismos bidimensionales-Tablas XY

<https://www.robotshop.com/letsmakerobots/xy-table>

<http://www.intellidrives.com/XY-tables-XYZ-Systems/Linear-Motor-XY-Robotic-Arm/Linear-Motor-XY-Robotic-Arms>

<https://openbuilds.com/builds/v-slot%E2%84%A2-x-y-stage-hidden-acme-rod-example.66/>

https://www.ijirset.com/upload/2015/september/21_Design.pdf

Driver

<http://www.norwegiancreations.com>

<http://teslabem.com/easydriver-motor-a-pasos-a3967.html>

https://es.aliexpress.com/item/1pcs-A3967-EasyDriver-Stepper-Motor-Driver-V44-for-arduino-development-board-3D-Printer-A3967-module/32725286649.html?src=google&albslr=225519109&isdl=y&aff_short_key=Un eMJZVf&source=%7Bifdyn:dyn%7D%7Bifpla:pla%7D%7Bifdbm:DBM&albch=DID%7D&src=google&albch=shopping&acnt=494-037-6276&isdl=y&albcpr=652736826&albag=32471627666&slnk=&trgt=61865531738&plac=&crea=es32725286649&netw=g&device=c&mtctp=&gclid=Cj0KCQiAnOzSBRDGARIsA L-mUB1xs5WeC_eYmfi9FPKn8-YYbnGafXaYAaDKVQCKy9zHtyO168JIZagaAvb-EALw_wcB

<https://www.sparkfun.com/datasheets/Robotics/A3967.pdf>
https://www.banggood.com/es/A3967-EasyDriver-Shield-Stepping-Stepper-Motor-Driver-Module-For-Arduino-p-1137049.html?utm_source=google&utm_medium=cpc_elc&utm_content=rubio&utm_campaign=ds-intustria-category&gclid=Cj0KCQiAnOzSBRDGARIsALmUB065hInmuGlzXp-at5wdDSV6sUjMiiJrFKwsAX0UW5xzbKjaQ00j_laAo1EEALw_wcB&cur_warehouse=CN
<http://www.schmalzhaus.com/EasyDriver/>

Bluetooth y motores

<https://electronicaymas.com/>
<https://www.youtube.com/watch?v=Ei43v93NEMQ>
<https://www.youtube.com/watch?v=YZiwhuKb8Ek>
<http://rogerbit.com/wprb/index.php/2015/08/06/control-de-motor-paso-a-paso-por-bluetooth-hc-06-ulm2003-28byj-48/>
<http://www.electroingenio.com/arduino/controlar-motor-dc-desde-un-telefono-movil/>
<http://www.electroingenio.com/arduino/motor-paso-a-paso-con-aplicacion-movil/>
<https://www.youtube.com/watch?v=OhnxU8xALtg>
https://www.pjrc.com/teensy/td_libs_AccelStepper.html
<https://www.youtube.com/watch?v=UD37HhbyocQ>
<http://www.electroensaimada.com/motor-paso-a-paso.html>
<https://www.youtube.com/watch?v=UD37HhbyocQ>
<https://www.youtube.com/watch?v=H8oEPMpSVkM>
<https://www.youtube.com/watch?v=D-vCXWX287c>
www.pblinear.com
<https://brainy-bits.com/blogs/tutorials/control-2-stepper-motors-using-the-arduino-serial-monitor>
<https://brainy-bits.com/blogs/tutorials/arduino-bluetooth>
<https://www.geekfactory.mx/tutoriales/bluetooth-hc-05-y-hc-06-tutorial-de-configuracion/>
<https://www.prometec.net/mbock-coche-hc06/>
<https://forum.arduino.cc/index.php?topic=163178.0>
http://www.naylampmechatronics.com/blog/12_Tutorial-B%C3%A1sico-de-Usode-del-M%C3%B3dulo-Bluetooth-H.html
<http://www.electronicoscaldas.com/modulos-rf/482-modulo-bluetooth-hc-06.html>
http://dinastiatecnologica.com/?s=BLUETTOTH&post_type=product

Presupuesto

https://www.software-shop.es/Microsoft-Office-2007-Professional-Plus_1?curr=EUR&gclid=Cj0KCQiAzMDTBRDDARIsABX4AWzEUT1Mr7lbrxqDxuOzQVADA6VQJycGvTVeeTeU0e_78SDzjFjBpWgaAj9WEALw_wcB
https://www.lizengo.es/microsoft/windows-7?gclid=Cj0KCQiAzMDTBRDDARIsABX4AWwKcNJyQCQIJBeT6FKFE-d3x-w1P-CAKW5OzAPunMJIVhAToOqFa9gaAmVYEALw_wcB

https://www.lucidchart.com/pages/landing/visio_2007?utm_source=google&utm_medium=cpc&utm_campaign=visio_2007_spain&gclid=Cj0KCQIAzMDTBRDDARIsABX4AWyqhLueum-OE1hVVKt9IFLYo4fiEBKAuwnL7Jq8HJxJDrA70GtCt2AaAui_EALw_wcB

<https://products.office.com/es-es/microsoft-visio-2007>

<https://www.ebay.es/itm/Visio-2007-Standard-Vollversion-englisch-neu-SKU-D86-02751/310897536409?hash=item4862efed99:g:jsgAAOSwZkJUSAWN>

<https://infoautonomos.eleconomista.es/consultas-a-la-comunidad/988/>

https://www.idealos.es/resultados.html?q=portatil+samsung+intel+core+i5&gclid=Cj0KCQIAzMDTBRDDARIsABX4AWyeymyzDGwQ2qU5nrs4Yy4b8Ks121KxKiobVLkE4PbAXXXMjhf_JAQaAkd-EALw_wcB

<https://www.idealos.es/precios/2898311/samsung-np300v5a-300v5a-s02.html>

<https://cuentica.com/asesoria/anos-y-porcentaje-de-amortizacion-para-sociedades/>

<http://www.leantricity.es/cuanta-energia-gasta-un-ordenador-aproximaciones/>

<https://www.indeed.es/salaries/Ingeniero/a-de-calidad-Salaries>

Códigos Unesco

http://upct.es/escueladoctorado/documentos/codigos_unesco_7809.pdf

9. ÍNDICE DE FIGURAS

Figura 1.1: Tipos de restricciones posturales.....	19
Figura 1.2: Planos del cuerpo humano.....	19
Figura 1.3: Gráfica de la velocidad media de balanceo frente al tipo de apoyo.....	20
Figura 1.4: Silla de ruedas bimanual con motores para propulsión.....	25
Figura 1.5: Silla de ruedas convencional.....	25
Figura 1.6: Scooter.....	26
Figura 1.7: Segway adaptado.....	26
Figura 1.8: Handbike eléctrico.....	27
Figura 1.9: Silla de ruedas adaptada para jugar al baloncesto.....	27
Figura 1.10: Joystick tradicional.....	29
Figura 1.11: Joystick de dedo.....	29
Figura 1.12: Joystick de alta sensibilidad.....	30
Figura 1.13: Mini joystick sellado.....	30
Figura 1.14: Joystick de control por dedo.....	30
Figura 1.15: Joystick no proporcional.....	31
Figura 1.16: Mini joystick no proporcional.....	31
Figura 1.17: Bandeja de pulsadores.....	31
Figura 1.18: Ilustración de control por soplado.....	32
Figura 1.19: Sensores de proximidad en silla de ruedas.....	33
Figura 1.20: Ilustración de pulsadores de libre de colocación.....	33
Figura 1.21: Joystick VR2 utilizado en el proyecto.....	34
Figura 3.1: Esquema de pines del Arduino Uno.....	47
Figura 3.2: Imagen real del Arduino Uno.....	48
Figura 3.3: Esquema de motor paso a paso unipolar.....	49
Figura 3.4: Esquema de motor paso a paso bipolar.....	50
Figura 3.5: Características del motor Nema 17.....	51
Figura 3.6: Esquema de los componentes del driver.....	52
Figura 3.7: Imagen del EasyDriver A3967.....	52
Figura 3.8: Representación esquemática de conexión maestro-esclavo.....	55
Figura 3.9: Imagen del módulo Bluetooth HC-06.....	55
Figura 3.10: Imagen de la batería portátil recargable.....	56
Figura 3.11: Imagen de las pilas elegidas.....	58
Figura 3.12: Gráfica comparativa de la duración de carga de las pilas tras un año sin usar: pilas elegidas (azul) frente a otras del mismo tipo (negro).....	58
Figura 3.13: Sistema conectado a la protoboard.....	60
Figura 3.14: Esquema de conexiones para la parte de los motores.....	60
Figura 3.15: Esquema de conexión para el Bluetooth.....	61
Figura 3.16: Esquema de conexiones completo.....	62
Figura 3.17: Imagen de las conexiones reales.....	62

Figura 3.18 a: Vista de la superficie inferior de la primera plataforma donde engranan los motores.....	63
Figura 3.18 b: Diseño de la pieza en FreeCad.....	64
Figura 3.19: Vistas del modelo de los engranajes en FreeCad.....	64
Figura 3.20: Base de apoyo de los motores.....	65
Figura 3.21: Plataforma donde se alojan los componentes electrónicos.....	66
Figura 3.22: Interfaz del IDE de Arduino.....	67
Figura 3.23: Flujograma del proceso.....	70
Figura 4.1: Imagen del código donde se especifican las variables programables.....	74
Figura 4.2: Plataforma con movimiento en dos dimensiones.....	78
Figura 4.3: Imagen de una FPGA modelo Spartan-3 de Xilinx.....	79
Figura 4.4: Imagen de la Raspberry Pi 3.....	80
Figura 4.5: Criterios de influencia en la elección del dispositivo.....	82
Figura 4.6: Driver L298N.....	83
Figura 4.7: Conector Bluetooth LM910-0630.....	84
Figura 5.1: Interfaz de la aplicación BluControl.....	88
Figura 5.2: Esquema de conexiones del Bluetooth.....	89
Figura 5.3: Esquema de las conexiones reales.....	89
Figura 5.4: Motores durante las pruebas de funcionamiento.....	91
Figura 5.5: Interfaz de la aplicación Arduino Stepper Motor.....	92
Figura 5.6: Desplazamientos de la plataforma para cada ángulo dado variando el nivel de dificultad.....	96
Figura 5.7: Tiempo desplazamiento de los motores según nivel de dificultad y ángulo.....	97
Figura 8.1: Diagrama de Gantt.....	114

10. ÍNDICE DE TABLAS

Tabla 3.1: Resumen de características Arduino Uno R3.....	48
Tabla 3.2: Secuencia simple o wave drive (solo una bobina activa).....	49
Tabla 3.3: Secuencia normal (la más habitual).....	49
Tabla 3.4: Secuencia de medio paso (giro más suave y preciso).....	50
Tabla 3.5: Secuencia para excitar las bobinas de un motor paso a paso bipolar.....	50
Tabla 3.6: Modos de configuración para la división de los pasos.....	53
Tabla 3.7: Características del HC-06.....	54
Tabla 3.8: Características de la batería recargable.....	57
Tabla 4.1: Factores ponderados para el Arduino y la Raspberry Pi 3.....	82
Tabla 4.2: Características del conector Bluetooth LM910-0630.....	84
Tabla 5.1: Resultados de las pruebas con el Bluetooth.....	90
Tabla 5.2: Resultados de las pruebas de los motores.....	91
Tabla 5.3a: Primer nivel de dificultad.....	93
Tabla 5.3b: Segundo nivel de dificultad.....	93
Tabla 5.3c: Tercer nivel de dificultad.....	93
Tabla 5.3d: Cuarto nivel de dificultad.....	93
Tabla 5.4a: Datos de desplazamiento de la plataforma para el primer nivel de dificultad.....	94
Tabla 5.4b: Datos de desplazamiento de la plataforma para el segundo nivel de dificultad.....	95
Tabla 5.4c: Datos de desplazamiento de la plataforma para el tercer nivel de dificultad.....	95
Tabla 5.4d: Datos de desplazamiento de la plataforma para el cuarto nivel de dificultad.....	96
Tabla 8.1: Estructura de descomposición del proyecto (EDP).....	113
Tabla 8.2: Costes asociados al material.....	115
Tabla 8.3: Costes asociados al software utilizado.....	115
Tabla 8.4: Costes de amortización.....	116
Tabla 8.5: Costes de electricidad.....	116
Tabla 8.6: Coste de recursos humanos.....	116
Tabla 8.7: Costes totales del proyecto.....	117
Tabla 8.8: Costes de materia para una unidad.....	117

ANEXO 1: CÓDIGO PROGRAMA PRINCIPAL

```

#include "AccelStepper.h"
// Library created by Mike McCauley at http://www.airspayce.com/mikem/arduino/AccelStepper/
// AccelStepper Setup
AccelStepper stepperX(1, 2, 3);
AccelStepper stepperZ(1, 5, 6); // 1 = Easy Driver interface
// UNO Pin 5 connected to STEP pin of Easy Driver
// UNO Pin 6 connected to DIR pin of Easy Driver

///ángulos del sensor: dato_rx --> ángulo sagital, dato_rx2 --> frontal, dato_rx3 --> transversal
int dato_rx=0; // valor recibido en grados
int dato_rx2=0; // valor recibido en grados
int dato_rx3=0; // valor recibido en grados
int angle; //variable auxiliar a la que se asigna dato_rx2 o dato_rx3 según angulo_elegido

String leeCadena; // Almacena la cadena de datos recibida

int cuenta_interna=0; //variable que se compara con cuenta para determinar cuál es el ángulo recibido
int cuenta=0; //variable que se incrementa en dos cada tres valores
int correcto1=1; //vale 1 si dato_rx es mayor que MIN en valor absoluto
int correcto2=1; //vale 1 si dato_rx2 es mayor que MIN en valor absoluto
int correcto3=1; //vale 1 si dato_rx3 es mayor que MIN en valor absoluto
float limit=42.0; //variable para ajustar el movimiento de los motores
float cte = 4.44; ///este valor se obtiene de la siguiente forma, sabiendo que la reductora es de 1/8 y el paso angular de 1,8º:
//una vuelta son 360º. 360/1,8=200 pasos. Con la reductora, necesito dar 8*200=1600 pasos para completar una vuelta.
//1600/360=4.44 --> Debo multiplicar por 4.44 para compensar el efecto de la reductora

////////// VARIABLES A MODIFICAR POR EL USUARIO //////////

//Para determinar el mínimo y el máximo de los ángulos del sensor
#define MIN 5
float MAX=20.0;
//Para determinar el ángulo elegido--cuando vale 1 es el frontal, cuando vale dos es el transversal
int angulo_elegido=1;
//Para determinar el nivel de dificultad (1-1.2-1.3-1.4)
float nivel=1;

////////// FIN DE VARIABLES A MODIFICAR POR EL USUARIO //////////

void setup() {
  Serial.begin(9600); // Start the Serial monitor with speed of 9600 Bauds

  // Set Max Speed and Acceleration of each Steppers
  stepperX.setMaxSpeed(500.0); // Set Max Speed of X axis
  stepperX.setAcceleration(500.0); // Acceleration of X axis
  stepperZ.setMaxSpeed(500.0); // Set Max Speed of Y axis slower for rotation
  stepperZ.setAcceleration(500.0); // Acceleration of Y axis

```

```

}

void loop() {
  while (Serial.available()) { // Leer el valor enviado por el Puerto serial
    delay(2);
    char c = Serial.read(); // Lee los caracteres
    leeCadena += c; //Concatena los caracteres
  }
  if (leeCadena.length()>0){
    if (cuenta_interna == cuenta) { //Comprobación para ángulo sagital
      cuenta_interna=cuenta+1; //Incremento cuenta_interna para luego entrar en ángulo frontal
      dato_rx = leeCadena.toInt();
      Serial.print("dato_rx ");
      Serial.println(dato_rx);
      if (abs(dato_rx)>=MIN) { //¿El dato es mayor que el mínimo?
        Serial.println("dato_rx correcto");
        if (abs(dato_rx)>MAX) { //¿El dato en valor absoluto es mayor que el máximo?
          if (dato_rx>0) { //Si el dato es positivo y mayor que el máximo, le doy el valor del máximo
            dato_rx=MAX;}
          else dato_rx=-MAX; //Si el dato es negativo y mayor que el máximo en valor absoluto, le doy el
valor de -máximo
            Serial.println("mayor que MAX");
          }
        else {Serial.println("dato entre mínimo y máximo ");}
      }
    }
    else {
      Serial.println("dato_rx incorrecto");
      dato_rx=0; //Si no es mayor que el mínimo, le doy valor cero para que no se mueva si está en
el origen o vuelva a este en otro caso
    }
  }
  else if (cuenta_interna==cuenta+1) { //Comprobación para ángulo frontal
    cuenta_interna=cuenta+2; //Incremento cuenta_interna para luego entrar en ángulo transversal
    dato_rx2 = leeCadena.toInt();
    Serial.print("dato_rx2 ");
    Serial.println(dato_rx2);
    if (abs(dato_rx2)>=MIN) { //¿El dato es mayor que el mínimo?
      Serial.println("dato_rx2 correcto ");
      if (abs(dato_rx2)>MAX) { //¿El dato en valor absoluto es mayor que el máximo?
        if (dato_rx2>0) { //Si el dato es positivo y mayor que el máximo, le doy el valor del máximo
          dato_rx2=MAX;}
        else dato_rx2=-MAX; //Si el dato es negativo y mayor que el máximo en valor absoluto, le doy el
valor de -máximo
          Serial.print("dato mayor que MAX ");
          Serial.println(dato_rx2);
        }
      else {Serial.println("dato entre mínimo y máximo ");}
    }
  }
  else {
    Serial.println("dato_rx2 incorrecto");
    dato_rx2=0; //Si no es mayor que el mínimo, le doy valor cero para que no se mueva si está en
el origen o vuelva a este en otro caso
  }
}
  else if (cuenta_interna==cuenta+2){ //Comprobación para ángulo transversal

```

```

cuenta=cuenta+2;          //Incremento cuenta_interna para luego entrar en ángulo sagital
dato_rx3 = leeCadena.toInt();
Serial.print("dato_rx3 ");
Serial.println(dato_rx3);
if (abs(dato_rx3)>=MIN) {    //¿El dato es mayor que el mínimo?
  Serial.println("dato_rx3 correcto ");
  if (abs(dato_rx3)>MAX) {    //¿El dato en valor absoluto es mayor que el máximo?
    if (dato_rx3>0) {        //Si el dato es positivo y mayor que el máximo, le doy el valor del máximo
      dato_rx3=MAX;}
    else dato_rx3=-MAX;      //Si el dato es negativo y mayor que el máximo en valor absoluto, le doy el
valor de -máximo
    Serial.println("mayor que MAX");
  }
  else {Serial.println("dato entre mínimo y máximo ");}
}
else {
  Serial.println("dato_rx3 incorrecto");
  dato_rx3=0;              //Si no es mayor que el mínimo, le doy valor cero para que no se mueva si está en
el origen o vuelva a este en otro caso
}
}

dato_rx = (dato_rx*cte*limit/(MAX*nivel)); //asigno el valor al ángulo sagital considerando el valor actual, la
razón respecto al máximo, el límite físico y el nivel
Serial.print("cuenta ");
Serial.println(cuenta);
Serial.print("cuenta interna ");
Serial.println(cuenta_interna);
}
mover();
leeCadena="";
}

void mover() {
  if (angulo_elegido==1){    //Si ángulo frontal
    angle = (dato_rx2*cte*limit/(MAX*nivel)); //asigno el valor del ángulo frontal convertido a la variable que le
da la distancia a los motores
    stepperZ.moveTo(angle); //doy esa distancia al motor responsable del giro
  } else {                  //Si ángulo transversal
    angle = (dato_rx3*cte*limit/(MAX*nivel)); //asigno el valor del ángulo transversal convertido a la variable
que le da la distancia a los motores
    stepperZ.moveTo(angle); //doy esa distancia al motor responsable del giro
  }
  if (cuenta_interna!=cuenta+1){ //Si no estoy en ángulo sagital (la cuenta ya se ha incrementado, por
eso no es cuenta_interna=cuenta)
    if (stepperZ.distanceToGo()!=0) { //Si la distancia que le queda por moverse es distinta de cero, muevo el
motor del giro
      stepperZ.run();}
    }
  else {stepperX.moveTo(dato_rx); //Si estoy en el ángulo sagital, asigno la distancia a moverse al motor
de avance
    if (stepperZ.distanceToGo()==0) { //Si ha acabado de moverse el motor del giro
      stepperX.run(); //muevo el motor de avance
    }
  }
}
}
}

```


ANEXO 2: CÓDIGO PRUEBA BLUETOOTH

```
void setup() {
  Serial.begin(9600);
  pinMode(8, OUTPUT);
}

void loop() {

  if(Serial.available(>0)
  {
    char data= Serial.read();           // leer los datos recibidos del módulo Bluetooth
    switch(data)
    {
      case 'a': digitalWrite(8, HIGH);break; // enciendo led al pulsar a
      case 'd': {digitalWrite(8, LOW);      // cuando pulso d, se apaga el led si estaba encendido y parpadea
                  delay(200);
                  digitalWrite(8,HIGH);
                  delay(500);
                  digitalWrite(8, LOW);
                  delay(500);
                  digitalWrite(8,HIGH);
                  delay(500);
                  digitalWrite(8, LOW);
                  delay(500);
                  break;}

      default : break;
    }

    Serial.println(data);
  }
  delay(50);
}
```


ANEXO 3: CÓDIGO PRUEBA MOTORES

```
/* Arduino Multiple Stepper Control Using The Serial Monitor
Created by Yvan / https://Brainy-Bits.com
This code is in the public domain...
You can: copy it, use it, modify it, share it or just plain ignore it!
Thx!
*/

#include "AccelStepper.h"
// Library created by Mike McCauley at
http://www.airspayce.com/mikem/arduino/AccelStepper/

// AccelStepper Setup
AccelStepper stepperX(1, 2, 3); // 1 = Easy Driver interface
// UNO Pin 2 connected to STEP pin of Easy Driver
// UNO Pin 3 connected to DIR pin of Easy Driver

AccelStepper stepperZ(1, 5, 6); // 1 = Easy Driver interface
// UNO Pin 5 connected to STEP pin of Easy Driver
// UNO Pin 6 connected to DIR pin of Easy Driver

// Stepper Travel Variables
long TravelX; // Used to store the X value entered in the Serial Monitor
long TravelZ; // Used to store the Z value entered in the Serial Monitor

int move_finished=1; // Used to check if move is completed

void setup() {

  Serial.begin(9600); // Start the Serial monitor with speed of 9600 Bauds

  // Print out Instructions on the Serial Monitor at Start
  Serial.println("Enter Travel distance seperated by a comma: X,Z ");
  Serial.print("Enter Move Values Now: ");

  // Set Max Speed and Acceleration of each Steppers
  stepperX.setMaxSpeed(500.0); // Set Max Speed of X axis
  stepperX.setAcceleration(500.0); // Acceleration of X axis

  stepperZ.setMaxSpeed(500.0); // Set Max Speed of Y axis slower for rotation
  stepperZ.setAcceleration(500.0); // Acceleration of Y axis
```

```
}
```

```
void loop() {
```

```
while (Serial.available()>0) { // Check if values are available in the Serial Buffer
```

```
    move_finished=0; // Set variable for checking move of the Steppers
```

```
    TravelX= Serial.parseInt(); // Put First numeric value from buffer in TravelX variable
```

```
    Serial.print(TravelX);
```

```
    Serial.print(" X Travel , ");
```

```
    TravelZ= Serial.parseInt(); // Put Second numeric value from buffer in TravelZ variable
```

```
    Serial.print(TravelZ);
```

```
    Serial.println(" Y Travel ");
```

```
    stepperX.moveTo(TravelX); // Set new move position for X Stepper
```

```
    stepperZ.moveTo(TravelZ); // Set new move position for Z Stepper
```

```
    delay(1000); // Wait 1 seconds before moving the Steppers
```

```
    Serial.print("Moving Steppers into position...");
```

```
    }
```

```
// Check if the Steppers have reached desired position
```

```
if ((stepperX.distanceToGo() != 0) || (stepperZ.distanceToGo() !=0)) {
```

```
    stepperX.run(); // Move Stepper X into position
```

```
    stepperZ.run(); // Move Stepper Z into position
```

```
    }
```

```
// If move is completed display message on Serial Monitor
```

```
if ((move_finished == 0) && (stepperX.distanceToGo() == 0) && (stepperZ.distanceToGo() == 0)) {
```

```
    Serial.println("COMPLETED!");
```

```
    Serial.println("");
```

```
    Serial.println("Enter Next Move Values (0,0 for reset: "); // Get ready for new Serial monitor values
```

```
    move_finished=1; // Reset move variable
```

```
    }
```

```
}
```