



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

Curso Académico:

RESUMEN

En el siguiente trabajo se realiza el diseño e implementación de un robot caminante hexápodo el cual se podrá utilizar en un programa formativo dirigido al ciclo de educación secundaria. Dicho programa introduce herramientas básicas de programación y metodologías relacionadas con las nuevas tecnologías, con el objetivo de acercar a los alumnos al mundo de la ingeniería. El proyecto pretende ser una alternativa novedosa a los trabajos realizados en distintas asignaturas, como por ejemplo tecnología.

Para el diseño del robot se ha utilizado el software *Lego Digital Designer*, realizándose una maqueta con piezas Lego. Más tarde, para reducir costes y aumentar prestaciones, se optó por el diseño en INVENTOR de distintas piezas que se utilizarían como estructura. En la parte de control se ha optado por las placas *Arduino* que integran un microcontrolador, así como un entorno de desarrollo. De esta forma, al implementar un programa no muy complicado, se puede controlar los motores desde una aplicación móvil, creada con *AppInventor*, la cual se comunica con la placa mediante el módulo *bluetooth* que se ha conectado. A lo largo del proyecto se detallarán los distintos componentes utilizados, el diseño de cada una de las configuraciones, la programación del software y el coste del mismo.

Palabras clave: Robótica móvil, hexápodo, Arduino, Lego, programación, electrónica, servomotor, bluetooth, aplicación móvil, educación, robótica educativa.

RESUM

En el següent treball es realitza el disseny e implementació d'un robot caminant hexàpode el qual s'utilitzarà en un programa d'interacció amb les escoles. Aquest programa introdueix ferramentes bàsiques de programació i metodologies relacionades amb les noves tecnologies, amb l'objectiu d'apropar els alumnes al món de l'enginyeria. El projecte pretén ser una alternativa nova als treballs realitzats en diferents assignatures, com per exemple tecnologia.

Per al disseny del robot s'ha emprat el software *Lego Digital Designer*, fent-se una maqueta amb peces Lego. Més endavant, per reduir costos y augmentar prestacions, es va optar pel disseny en INVENTOR de diferents peces que s'utilitzarien com estructura. En la part de control es va optar per les plaques *Arduino* que integren un microcontrolador, així com un entorn de desenvolupament. D'aquesta forma, al implementar un programa no molt complicat, es pot fer el control dels motors des d'una aplicació mòbil, creada amb *AppInventor*, la qual es comunica amb la placa *Arduino* mitjançant el mòdul *bluetooth* que s'ha connectat. Al llarg del projecte es detallaran els diferents components utilitzats, el disseny de cadascuna de les configuracions, la configuració del software i el cost del mateix.

Paraules clau: Robòtica mòbil, hexàpode, Arduino, Lego, programació, electrònica, servomotor, bluetooth, Aplicació mòbil, educació, robòtica educativa.

ABSTRACT

In the next document it's performed the design and implementation of a hexapod wayfarer robot, which will be used in a program of interaction with schools. This program introduces basic programming tools and methodologies related to new technologies, with the aim of bring in to the world of engineering. The project aims to be an alternative to the work that is made in different subjects, such as technology.

For the design of the robot, it has been used the software Lego Digital Designer, making a model with Lego pieces. Later, to reduce costs and increase benefits, we chose INVENTOR to design different parts of the structure. In the control part it has chosen Arduino boards that integrate a microcontroller and a development environment. Thus, by implementing a not very complicated program, we can control the motors from a mobile application, created with AppInventor, which communicates with the board by the Bluetooth module. Throughout the project, we will explain the components used, the design of each configurations, the software program and the cost of the project.

Keywords: mobile robotics, hexapod, Arduino, Lego, programming, electronic, servomotor, bluetooth, mobile app, education, educational robotics.

INDICE

MEMORIA.....	
1. INTRODUCCIÓN	1
1.1 ROBÓTICA EDUCATIVA	1
1.2. OBJETIVOS DEL PROYECTO	2
2. CLASIFICACIÓN DE LOS ROBOTS SEGÚN SU ARQUITECTURA.....	2
2.1. POLIARTICULADOS.....	3
2.2. MÓVILES CON RUEDAS.....	3
2.3. ANDROIDES.....	5
2.4. ZOOMÓRFICOS	6
2.5. HÍBRIDOS	6
3. ANÁLISIS	6
3.1. REQUERIMIENTOS DE HARDWARE.....	6
3.2. REQUERIMIENTOS DE SOFTWARE.....	7
4. DISEÑO Y CORRECCIONES	7
4.1. LEGO DIGITAL DESIGNER	8
4.2. PROCEDIMIENTO	8
4.3. DESPLAZAMIENTO DE LAS PATAS.....	12
4.3. TRASMISIÓN DE LOS ESFUERZOS	13
4.4. LIMITACIONES LEGO	16
5. ELECTRÓNICA	17
5.1. ARDUINO	18
5.2. PLACA DE EXPANSIÓN	19
5.3. MOTORES	19
5.4. MÓDULO BLUETOOTH.....	21
5.5. OTROS COMPONENTES	23
5.6. CONEXIONES.....	23
6. PROGRAMACIÓN ARDUINO	24
6.1. LIBRERÍAS.....	25
6.2. PROGRAMA	26
7. APLICACIÓN MÓVIL	28
7.1.-APPINVENTOR.....	28

7.2. ASPECTO Y PROGRAMACIÓN	29
8. RESULTADOS.....	32
9. CONCLUSIONES	33
10. PROYECTO EDUCATIVO	33
11. MEJORAS	34
12. BIBLIOGRAFÍA	36
PRESUPUESTO	
1. PRESUPUESTO	39
1.1. PRESUPUESTO MATERIAL.....	39
1.2. COSTE PERSONAL	43
1.3. COSTE ELEMENTOS SOFTWARE Y HARDWARE	44
1.4. COSTE TOTAL DEL PROYECTO.....	45
PLANOS.....	
1.1. PLANO BASE INFERIOR ROBOT HEXÁPODO	47
1.2. PLANO BASE SUPERIOR ROBOT HEXÁPODO	48
ANEJO 1. PROGRAMA.....	49

INDICE DE FIGURAS

Figura 1. LEGO NXT en programas extraescolares	1
Figura 2. Robot manipulador.....	3
Figura 3. Robot Cartesiano.....	3
Figura 4. Configuración diferencial	3
Figura 5. Configuración <i>Ackerman</i>	4
Figura 6. Ejemplos de las distintas configuraciones.....	5
Figura 7. Robot Asimo	5
Figura 8. Robot cuadrúpedo que imita al caballo	6
Figura 9. Diseño 1 del robot hexápodo	9
Figura 10. Implementación del primer diseño	10
Figura 11. Diseño del segundo modelo	11
Figura 12. Implementación del diseño final	11
Figura 13. Fijación del engranaje.....	12
Figura 14. Distancias iniciales.....	13
Figura 15. Tren de engranajes.....	14
Figura 16. Esquema de un sistema electrónico.....	17
Figura 17. Placa <i>Arduino</i> NANO.....	18

Figura 18. Placa de expansión <i>funduino</i> nano 006v3.0.....	19
Figura 19. Estructura interna de un servomotor.....	20
Figura 20. Ejemplo PWM.....	21
Figura 21. Módulo HC-06	22
Figura 22. Esquema de conexiones	23
Figura 23. Compilador <i>Arduino</i>	24
Figura 24. Pantalla de diseño en App Inventor	28
Figura 25. Pantalla de programación App Inventor	29
Figura 26. Pantallas de la aplicación.....	30
Figura 27. Conexión <i>bluetooth</i>	30
Figura 28. Programación control mediante acelerómetro.....	31
Figura 29. Programación control mediante botones	32
Figura 30. Programa selección de pantalla	32

INDICE DE TABLAS

Tabla 1. Medidas piezas Lego.....	9
Tabla 2. Posición parte superior pata.....	13
Tabla 3. Relaciones de transmisión	15
Tabla 4. Características <i>Arduino</i> NANO	18
Tabla 5. Características PowerHD AR-3606HB	20
Tabla 6. Características FS90r.....	20
Tabla 7. Características HC-06.....	22
Tabla 8. Presupuesto material del TOTAL DEL PROYECTO	39
Tabla 9. Presupuesto material FINAL	41
Tabla 10. Coste personal	44
Tabla 11. Coste elementos de software y hardware.....	44
Tabla 12. Coste total del proyecto	45



MEMORIA

**DISEÑO E IMPLEMENTACIÓN DE UN ROBOT
CAMINANTE HEXAPODO BASADO EN LEGO Y
ARDUINO**

1. INTRODUCCIÓN

La robótica es una ciencia o rama de la tecnología, que estudia el diseño y construcción de máquinas capaces de realizar tareas que requieran inteligencia o imitar comportamientos del ser humano u otros seres vivos. El término robot apareció en 1921 en la obra teatral *Rosum's Universal Robots* del novelista y dramaturgo Karel Capek, para el cual la palabra *robot* significa servidumbre o trabajo forzado.

Debido a su fuerte relación con otras ciencias, como la electrónica o la automática, esta tecnología no se desarrolló hasta mediados del pasado siglo XX. A partir de este momento el mundo de la robótica fue expandiéndose en el ámbito industrial, sustituyendo aquellas tareas que resultaban más costosas, repetitivas, precisas o peligrosas para los humanos. El avance de estas ciencias nos ha llevado a la creación de máquinas programables con elevada velocidad de procesamiento, capaces de realizar tareas fuera del alcance del ser humano.

Actualmente la robótica es una de las ciencias que más se ha apoderado de nuestro entorno. No es extraño ver como los robots han ido ocupando distintas funciones en diferentes áreas como la medicina, la industria, el hogar, el entretenimiento e incluso en el ámbito militar, entre otras. El concepto de robótica ha evolucionado hacia los sistemas móviles autónomos, capaces de desenvolverse por sí mismos en entornos desconocidos y cambiantes.

1.1 ROBÓTICA EDUCATIVA

Este marco cambiante, provoca el nacimiento y desarrollo de otras aplicaciones como la robótica educativa. Este término se puede definir como el conjunto de actividades pedagógicas que apoyan y fortalecen áreas específicas del conocimiento, favoreciendo la comprensión de esta ciencia por el alumno mediante la construcción de un robot. Con esta enseñanza se busca la adaptación de los alumnos a los nuevos procesos de producción, donde la automatización y otras ramas tecnológicas son muy importantes. Mediante el estudio e implementación de comandos sencillos de programación, el alumno será capaz de provocar los movimientos del robot, siendo muy importante la experimentación y el autodescubrimiento. Con este medio se busca enriquecer la creatividad del alumno, su capacidad de observación y un aprendizaje de forma divertida.

Figura 1. LEGO NXT en programas extraescolares



Fuente: La térmica, Diputación de Málaga

En la actualidad son muchas las marcas que han apostado por esta forma educativa. LEGO, MULTIPLO o ROBO-ED son algunos ejemplos de empresas que han puesto a nuestra disposición piezas para crear la estructura del robot, así como una gran variedad de sensores y motores para la realización de diversas funciones.

En los últimos años hemos podido observar cómo se han realizado en España numerosas reformas educativas, las cuales han incluido nuevos contenidos sobre el control automático y la robótica en la Educación Secundaria Obligatoria, en concreto en el tercero y cuarto curso. Así

pues, según el REAL DECRETO 1631/2006, podemos observar como en la asignatura de tecnología se han introducido los bloques de control y robótica, aunque no se ha extendido esta modificación a todas las comunidades españolas.

También es importante subrayar la robótica educativa fuera del aula. Recientemente hemos vivido el auge de los cursos, actividades extraescolares e incluso competiciones o campamentos alrededor de toda España, cuyo principal objetivo es dar a conocer esta tecnología y servirse de la misma para potenciar el conocimiento técnico/científico de los alumnos. Algunas de las competiciones son: la *First Lego League*, la *ClauTIC League*, competición online multimarca e internacional completada con la *ClauTIC CUP*; la *Robolot* de Girona, etc. De esta forma, el concepto de robótica educativa está llegando a todas partes del mundo, siendo muy fuerte el empuje recibido desde Latinoamérica. En Valencia también hemos notado la expansión de este mundo, manifestada a través de actividades organizadas por la comunidad “DROIDE” y otras. Dentro de su programa se pueden encontrar talleres, cursos y campamentos dirigidos a un público infantil, así como talleres y cursos de programación e impresión 3D para adultos. Existen también otros grupos como por ejemplo, el grupo *Makers* de la universidad politécnica de Valencia, creado por estudiantes de la propia universidad, quienes organizan múltiples talleres y actividades. Recientemente han participado en concursos y eventos como por ejemplo la ORC (*Olympic Robotic Challenge*) o la “Primavera educativa”.

1.2. OBJETIVOS DEL PROYECTO

El diseño del robot caminante hexápodo puede complementar a un proyecto mayor, el cual se podría llevar a las aulas de los colegios. Este proyecto está diseñado para incluir otros modelos de robot, como por ejemplo un robot de configuración diferencial, algunos de estos modelos ya se pudieron ver en la pasada edición de la Primavera educativa en Valencia. En particular, este diseño pretende ser una alternativa a otros modos de locomoción, facilitando que el alumno experimente con las distintas resoluciones de un mismo problema.

De esta manera el proyecto pretende ser una alternativa didáctica interdisciplinaria que potencie el desarrollo de las habilidades y las competencias de los alumnos. Mediante la construcción y programación de los distintos modelos, los alumnos tendrán que afrontar diversos retos en grupo, aportando soluciones y aprendiendo de sus propias acciones y errores. Finalmente, se trata de aumentar la dificultad de los problemas a medida que los alumnos aumentan sus capacidades para afrontarlos, de manera que el aprendizaje sea dinámico y motivante para ellos, como alternativa a los métodos convencionales. El objetivo del proyecto será el diseño de un robot hexápodo, implementado con piezas Lego y *Arduino*, que se adapte a este programa de interacción con los alumnos. Para ello se diseñará una estructura, se escogerá una electrónica determinada y se realizará la programación correspondiente al microcontrolador y a la aplicación móvil.

2. CLASIFICACIÓN DE LOS ROBOTS SEGÚN SU ARQUITECTURA

El término robot engloba a un gran nombre de dispositivos y mecanismos, lo que hace que sea muy difícil establecer una clasificación. Para ello en el siguiente apartado nos basaremos en la

arquitectura del robot, es decir, su configuración general. De esta forma, encontramos los siguientes grupos: poliarticulados, móviles, androides, zoomórficos e híbridos.

2.1. POLIARTICULADOS

Bajo este grupo se encuentran robots de formas muy diversas y diferentes configuraciones, siendo una característica común en todos ellos el sedentarismo (aunque pueden ser guiados para efectuar movimientos limitados). Estos robots actúan en un espacio físico determinado, que suele ser limitado. Son ejemplos de robots con esta configuración los industriales, cartesianos y manipuladores.

Figura 3. Robot Cartesiano.



Fuente: Dotest S.L.

Figura 2. Robot manipulador



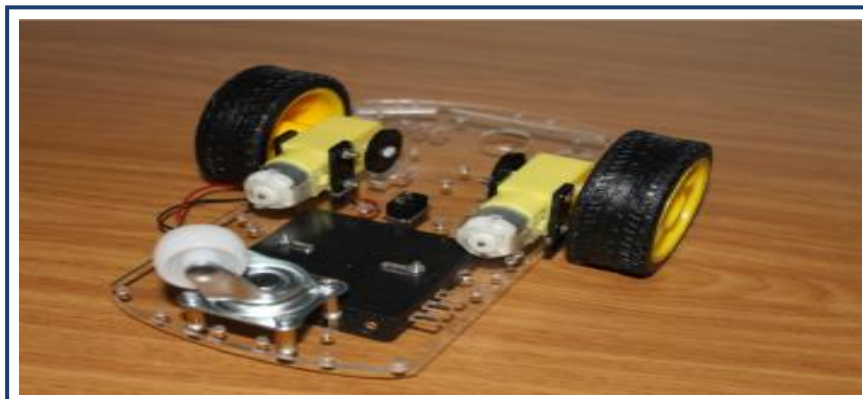
Fuente: Logismarket

2.2. MÓVILES CON RUEDAS

Este tipo de robots nos ofrece la solución más simple y eficiente para movernos por superficies suficientemente duras. Su principal inconveniente es que pueden patinar las ruedas. De esta manera podemos utilizar los robots para realizar movimientos relativamente fiables y alcanzar altas velocidades. Estos robots son de gran utilidad para funciones de transporte o exploración en terrenos abruptos o peligrosos para el ser humano. Existen distintas configuraciones:

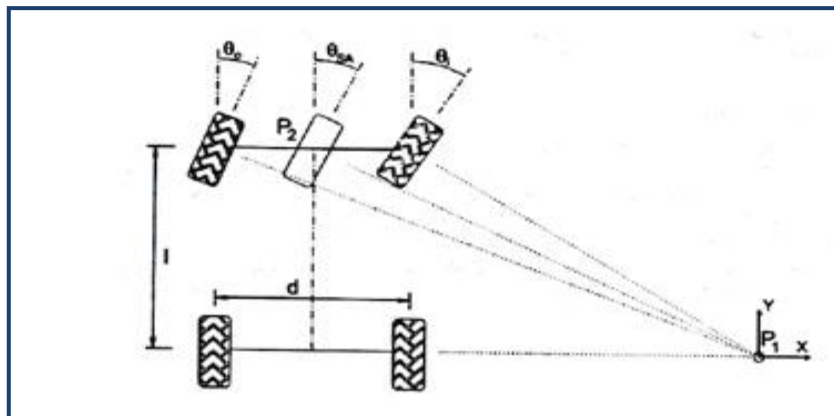
- ▶ **Configuración diferencial:** Se clasifican por tener dos ruedas de tracción y una tercera de apoyo que gira libremente (rueda loca). En función de la velocidad de cada una de las ruedas podemos implementar giros. Es un sistema barato pero difícil de controlar.

Figura 4. Configuración diferencial



- ▶ **Configuración oruga:** Sustitución de las ruedas por cadenas en los laterales. Se comporta como el modelo anterior siendo el equivalente a las ruedas los centros de las cadenas. Este modelo presenta mejor tracción y seguridad antiderrape.
- ▶ **Configuración triciclo:** En este caso disponemos también de dos ruedas traseras. La diferencia con la configuración diferencial consiste en que la rueda delantera es la encargada del movimiento, tanto para la velocidad como para la orientación. De esta forma podremos actuar sobre la velocidad de avance y la orientación de la rueda delantera. Este modelo consigue mejor adherencia pero puede no ser útil en terrenos irregulares.
- ▶ **Configuración Ackerman:** Configuración típica de los automóviles. Utiliza 4 ruedas de las cuales dos resultan de tracción y otras dos de dirección. En esta configuración la rueda interior gira un ángulo mayor a la exterior en las curvas para evitar deslizamientos. La prolongación de los ejes delanteros intersecta en la proyección del eje trasero.

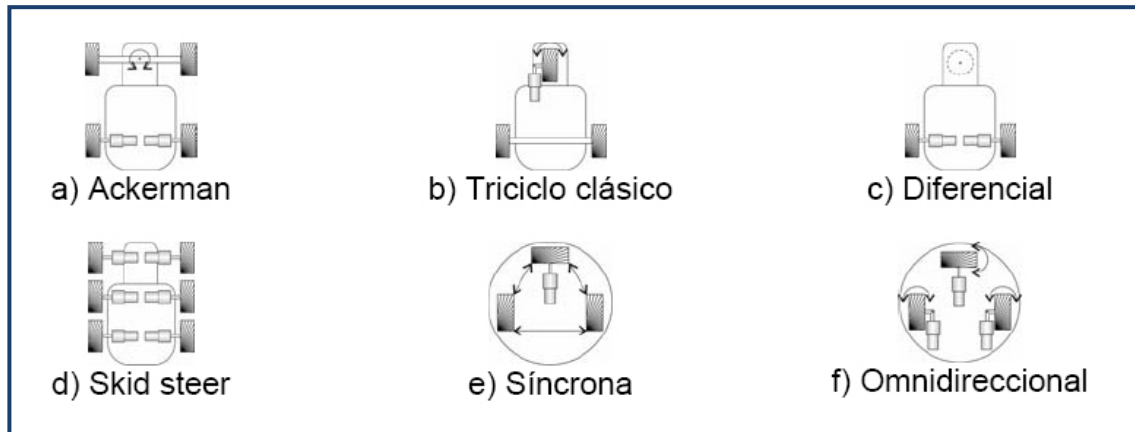
Figura 5. Configuración Ackerman



Fuente: Muchotrasto, tipos de plataformas

- ▶ **SkidSteer:** Esta configuración emana de la configuración diferencial. Los lados del vehículo actúan de forma simultánea añadiendo gran estabilidad y posibilitan el giro sobre su centro. La principal ventaja es que admite mucho peso sobre las ruedas.

Figura 6. Ejemplos de las distintas configuraciones

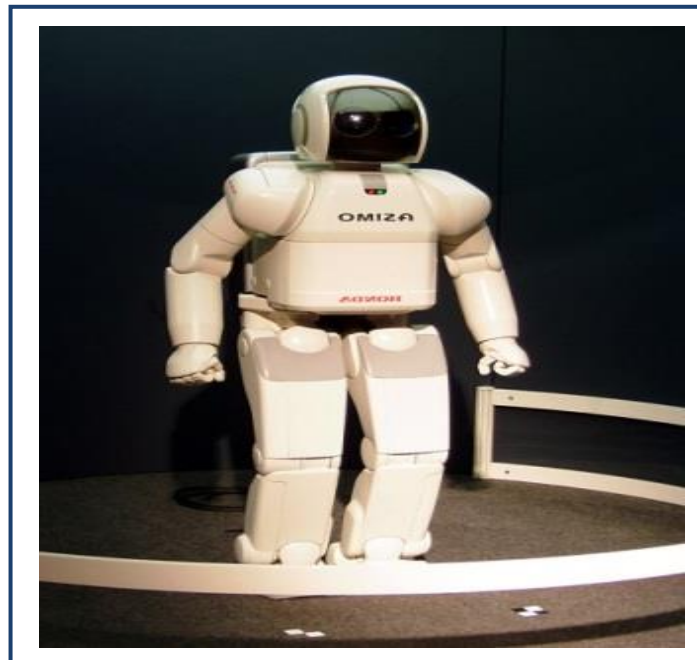


Fuente: Publicaciones Urbe

2.3. ANDROIDES

En este grupo encontramos a los robots que tratan de imitar los movimientos y la apariencia del ser humano. Existen diversos enfoques en este punto ya que se intenta avanzar en dos vías: imitar las expresiones humanas y conseguir un buen movimiento usando la locomoción bípeda. El principal problema que presentan estos robots es controlar de forma dinámica y coordinada el movimiento, mientras el robot mantiene el equilibrio. Uno de los modelos más avanzados es el robot Asimo de Honda, capaz de andar sobre sus dos pies así como de subir y bajar escaleras.

Figura 7. Robot Asimo



Fuente: fotogaleriasrtve

2.4. ZOOMÓRFICOS

Dentro de este grupo también podríamos incluir a los androides, constituyen un grupo que imita los movimientos de multitud de seres vivos. Existen dos subgrupos dentro de esta clase: los caminadores y los no caminadores. Los no caminadores han ido evolucionando en los últimos años dando modelos que imitan a los pájaros, peces, libélulas e incluso serpientes; siendo su principal característica que no utilizan patas para realizar el movimiento. Dentro de los caminadores tienen gran importancia los múltipodos, los cuales se han desarrollado por su gran versatilidad en terrenos muy accidentados. Un ejemplo puede ser el robot caballo desarrollado por el grupo *Boston Dynamics* cuya misión sería el transporte de un importante peso. Estos modelos cuentan con una gran estabilidad, siendo muy difícil que pierdan el equilibrio.

Figura 8. Robot cuadrúpedo que imita al caballo



Fuente: Actualidad RT

2.5. HÍBRIDOS

Dentro de este grupo encontramos aquellos de difícil clasificación ya que su estructura no corresponde a ninguna de las anteriores ya expuestas con total integridad. Existen algunos modelos que combinan algunos de los grupos como por ejemplo un segmento articulado y con ruedas, que entraría en el grupo de móviles y zoomórficos a la vez.

3. ANÁLISIS

El diseño de un robot comienza por el análisis de todos aquellos componentes que serán necesarios para su elaboración. Estos se pueden dividir en dos grupos en función de si son elementos de hardware o software.

3.1. REQUERIMIENTOS DE HARDWARE

El primer punto para desarrollar un robot móvil es la definición de la estructura que será más adecuada para su funcionamiento. Existen multitud de alternativas pero todas ellas dependen de la disposición de las patas y su movimiento. Además, el número de motores empleado también es un condicionante en el diseño del robot, cuanto mayor sea su número, más caro resultará el proyecto.

Una de estas alternativas podría ser el implementar el movimiento con servomotores, de manera que cada pata utilice dos o tres para su movimiento. Como se ha mencionado antes, esta opción se

descarta para este modelo ya que se necesitarían un total de 12 o 18 motores lo cual encarecería el precio final del robot (no olvidemos que este robot va dirigido a la comunidad escolar). De esta forma se ha optado por una configuración en la cual tan solo con dos motores de rotación continua se accionen todas las patas. La utilización de dos motores en lugar de uno es necesaria para que se puedan implementar giros ya que si un lado avanza más rápido que el otro, el robot girará. El motor se conectará a un eje el cual transmitirá el movimiento a las patas. Cada pata presentará dos movimientos, uno vertical y otro horizontal, imitando el movimiento de los hexápodos. La estructura tendrá que ser lo suficientemente rígida para sustentar el propio peso del robot e impedir el desplazamiento de los ejes ya que un pequeño movimiento de estos puede resultar fatal para la transmisión de los esfuerzos. Los materiales para la elaboración de la estructura serán piezas lego *TECHNIC*, cuyo ensamblaje se realizará con un programa de la propia compañía.

Una vez diseñada la parte motora, nos centramos en la parte de control. Como se ha establecido a la hora de realizar el proyecto, este se implementará con el hardware *Arduino*. Para facilitar las conexiones se utilizará una placa *shield* y la comunicación se establecerá mediante un módulo *bluetooth*. El dispositivo de control que servirá como "mando" será el propio teléfono móvil. Se ha seleccionado este método ya que en la actualidad todo el mundo dispone de esta tecnología e incluso los modelos básicos de *Smartphone* ya disponen de pantalla táctil, *bluetooth* y acelerómetro. Con esto se evita el tener que usar emisoras de radio o infrarrojos, muy utilizadas en el campo de los coches de radio control. La parte de hardware se completará con dos baterías de 9V encargadas de suministrar electricidad y de aquellos componentes electrónicos que sean necesarios.

3.2. REQUERIMIENTOS DE SOFTWARE

El software del robot ha de ser capaz de recibir señales desde el terminal móvil e interpretarlas para realizar los movimientos. De esta manera, serán necesarios dos programas para el control del robot.

El primer programa será el que incorpore la placa *Arduino*, el cual tendrá una serie de instrucciones para los motores, necesarias para controlar el movimiento. Este programa tendrá configurados todos los componentes electrónicos necesarios para el funcionamiento del robot.

El segundo programa será el que incorpore la aplicación móvil, el cual mandará caracteres por *bluetooth* en función de los botones que seleccionemos o la inclinación que le demos al teléfono. Se intentará que el control de la aplicación sea visual e intuitivo.

Estos programas consistirán en instrucciones sencillas, con el objetivo de que los alumnos los puedan elaborar sin que les resulte muy complicado.

Los programas utilizados para desarrollar el código serán de licencia gratuita con el objetivo de abaratar el coste del proyecto.

4. DISEÑO Y CORRECCIONES

Para el diseño del método de locomoción del robot se optó por un robot hexápodo como alternativa al movimiento con ruedas. Esta opción emplea un mecanismo de mayor complejidad, lo cual nos invita a un diseño más elaborado. Durante el diseño se llegó a múltiples alternativas, seleccionándose aquella que mejor movimiento y mayores prestaciones ofrecía.

4.1. LEGO DIGITAL DESIGNER

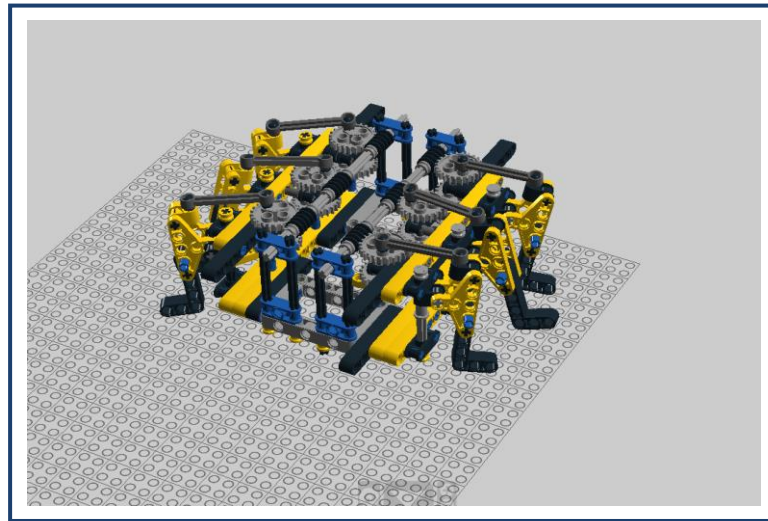
Para realizar la estructura del robot, se ha utilizado el programa de *Lego Digital Designer* o también conocido como LDD. Este es un software libre desarrollado por *The Lego Group*, disponible para *Mac* y *Windows*. Utilizando este programa se puede diseñar a través del PC cualquier construcción de LEGO. Cuenta con un amplio catálogo de piezas, siendo las más útiles para nuestro proyecto las correspondientes a la línea *technic*, y la mayoría están disponibles en varios colores. Su interfaz es sencilla e intuitiva, incluyendo elementos de líneas como *Mindstorms* o *Creator* y también ofrece diseños predeterminados sin acabar para poder personalizarse a gusto del usuario.

El programa dispone de varias herramientas para realizar ensamblajes de piezas, movimientos o visualizaciones. Mediante este programa podemos generar guías de construcción, de modo que se pueden hacer instrucciones automáticamente de cómo construir paso a paso las creaciones, incluso con la opción de guardarlas en formato HTML. También nos dice cuánto costaría crear la construcción en la realidad. Podemos guardar nuestras creaciones y compartirlas, enviarlas a la página oficial de LEGO y visualizar o descargar las creaciones de otras personas que las hayan subido previamente.

4.2. PROCEDIMIENTO

Tras analizar múltiples alternativas y diseños, se llegó a una opción viable y económica ya que se buscaba el movimiento con el menor número de motores, como se ha mencionado anteriormente. De esta forma cada pata dispone de dos movimientos, uno vertical y otro horizontal, implementados con un mecanismo biela manivela. Los mecanismos de las tres patas de cada uno de los lados están accionados a su vez por un eje de transmisión mediante el correspondiente tornillo sin fin por pata. Para transmitir el movimiento de rotación del motor al eje, se ha optado por un tren de engranajes. La plataforma de la parte inferior sería para albergar la electrónica. De esta forma conseguimos el siguiente diseño:

Figura 9. Diseño 1 del robot hexápodo



Fuente: Elaboración propia usando LDD

Como se ha mencionado antes, una de las facilidades que nos proporciona el software *Legó Digital Designer* es la guía de montaje. De esta forma, a través de una secuencia de instrucciones, se puede montar el robot de manera rápida y sencilla.

Tras realizar el montaje y con el objetivo de abaratar el coste del robot, se observó que mediante el diseño de una base estructural se reducían un importante número de piezas ya que sobre el mismo plano se realizaban numerosas uniones. Esta opción añade importantes mejoras al robot como por ejemplo rigidez, resultando necesaria su implementación. El diseño de esta pieza se realizó mediante INVENTOR, tomando las medidas y distancias de la propia maqueta a partir de las dimensiones de cada una de las piezas. Lego utiliza lo que ellos denominan *Legó Unit* para hacer referencia a las dimensiones principales de las piezas. En la siguiente tabla, se muestran los valores de las distintas dimensiones:

Tabla 1. Medidas piezas Lego

Elemento	Medida(mm)
Distancia entre agujeros	8
Diámetro de los agujeros	4,9
Diámetro de las cruces	4,8
Altura de las piezas	8

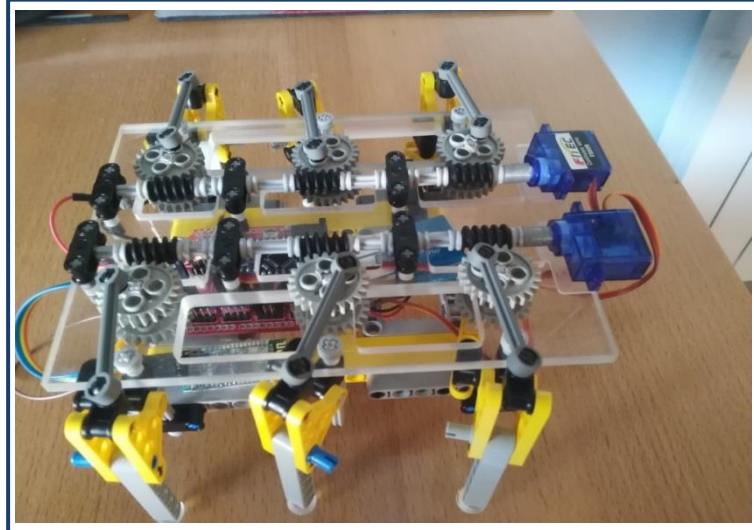
Fuente: Elaboración propia

De esta manera, para obtener una medida longitudinal por ejemplo, únicamente tenemos que multiplicar la cantidad de agujeros por la distancia que los separa. Así pues una barra con 15 agujeros tiene una longitud total de 120mm.

Para la producción de la base se pensó en distintos métodos como el fresado de metacrilato, el corte de una placa de contrachapado o la impresión 3D, descartándose esta última por los elevados costes ya que es una tecnología novedosa y la pieza era de grandes dimensiones. Así pues se seleccionó como material para su elaboración metacrilato, ya que dispone de elevada rigidez. Para eliminar peso se optó por hacer numerosos huecos de forma que no restaran rigidez al conjunto. Por último,

para completar el diseño, se requería de una pieza encargada de la transmisión del movimiento producido por el motor. Para ello se diseñó un acople entre el motor y el eje, para el cual se recurrió a la impresión 3D por ser de reducidas dimensiones y presentar un coste asumible. En la siguiente figura podemos observar el montaje:

Figura 10. Implementación del primer diseño



Fuente: Elaboración propia

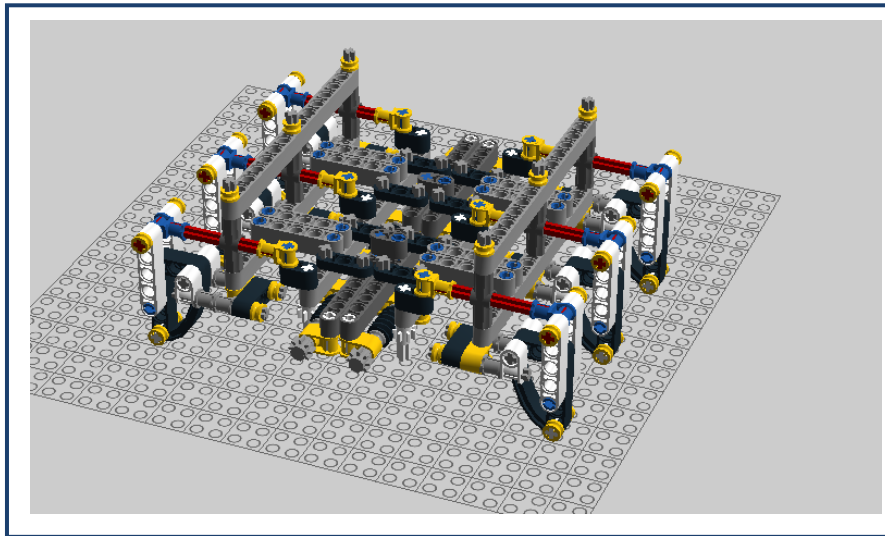
Este diseño presentaba una dificultad a la hora de reproducir el movimiento de retroceso del robot ya que debido al tipo de unión con el eje se favorecía el movimiento de avance por tener un mayor recorrido en esta dirección que en la otra. Además, el movimiento era lento ya que el tornillo sin fin accionaba un engranaje de elevado número de dientes y la velocidad del eje era directamente la del motor. Se modificaron varios aspectos del primer modelo para resolver estos problemas.

- ▶ Transmisión: se optó por implementar un tren de engranajes en lugar de la transmisión directa motor/eje. Con esta medida se mejoró la velocidad de avance.
- ▶ Emplazamiento de la electrónica en la parte superior. Los componentes electrónicos que inicialmente estaban en la parte inferior del robot, se pasaron a la superior, facilitando la accesibilidad.

El resultado final no fue satisfactorio porque no resolvía problema de avance/retroceso uniforme del robot, rechazándose esta configuración.

Finalmente se optó por cambiar la configuración de la estructura del robot, tanto de las patas como del sistema de transmisión. En este caso, para conseguir el movimiento horizontal se actúa en el eje central de la pata, obteniendo un movimiento fluido en las dos direcciones. De esta manera se volvió a realizar todo el proceso de diseño, pasando por las distintas partes antes mencionadas. La pieza encargada de dar rigidez a la estructura se realizaría otra vez en metacrilato dados los buenos resultados alcanzados en el anterior modelo. También se necesitaba cambiar el tipo de engranaje que conectaba con el tornillo sin fin ya que este era el principal motivo por el cual el movimiento era lento.

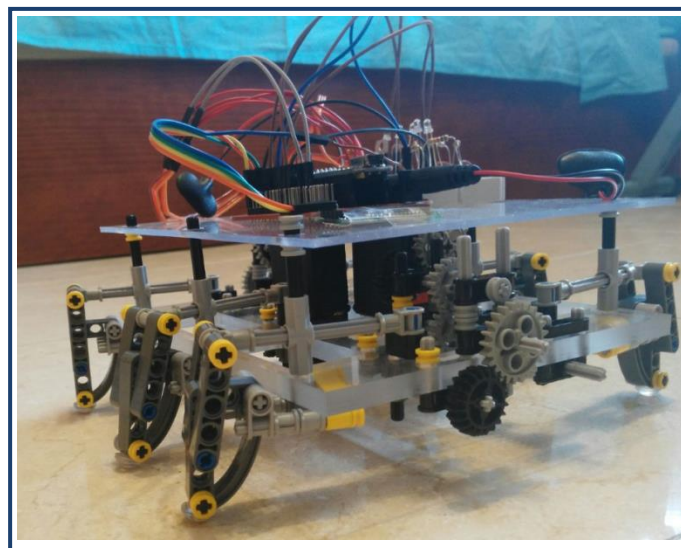
Figura 11. Diseño del segundo modelo



Fuente: Elaboración propia usando LDD

Este diseño proporciona mayor velocidad de movimiento al estar unido el tornillo sin fin a un engranaje de 8 dientes en lugar de uno de 24 y presentar un tren de engranajes más elaborado. También encontramos otras mejoras como es la disposición de la electrónica en la parte superior (más accesible) y la corrección del movimiento. La gran diferencia en el movimiento de la pata reside en que con una única unión a la parte giratoria conseguimos los dos movimientos. Cuando la rueda dentada gire 180° desde la posición de reposo, la pata alcanzara el mayor desplazamiento vertical ya que tirará de la barra que las une. Cuando esta gire entre 90° o 270° , tendremos los puntos de máximo desplazamiento horizontal, ya que la pieza que nos sirve como guía transmite su propio movimiento de giro al sistema conectado a la pata. Sin embargo, esta disposición obligó a cambiar los motores por unos de mayores prestaciones, ya que se producía un mayor rozamiento entre piezas y habíamos disminuido la fuerza de empuje de las patas al cambiar el tipo de engranaje conectado al tornillo sin fin. La implementación de este diseño queda de la siguiente forma:

Figura 12. Implementación del diseño final



Fuente: Elaboración propia

Finalmente se modificaron las dimensiones la base, ya que necesitábamos un mayor espacio para los nuevos motores. Esta modificación nos permite tener suficiente espacio para albergar el tren de engranajes más elaborado, conectado directamente al motor. Los motores se sitúan en la parte central, al contrario que en anterior modelo, para una mayor estabilidad del robot. Con todas estas modificaciones se consiguió elevar la velocidad de giro del eje, un mejor acople entre engranaje conductor y conducido y en definitiva, un movimiento mejorado.

Una de las partes más importantes a la hora de realizar el movimiento es el sistema de transmisión del movimiento de giro del motor al primer engranaje. El acople entre motor y engranaje se realizó con microtornillería, ya que esta opción permitía transmitir todo el esfuerzo de manera fiable e impidiendo que el engranaje se soltara. Para ello se lijó una de las caras del engranaje para hacerla plana, se realizaron cuatro agujeros utilizando brocas de 1,5mm y se pasaron los cuatro tornillos con sus respectivas tuercas, evitando posibles deformaciones en la pieza en el caso de haber usado solo dos tornillos. La pieza a la cual se fija el engranaje es uno de los diversos acoples que trae el motor. En la siguiente figura podemos observar el resultado:

Figura 13. Fijación del engranaje

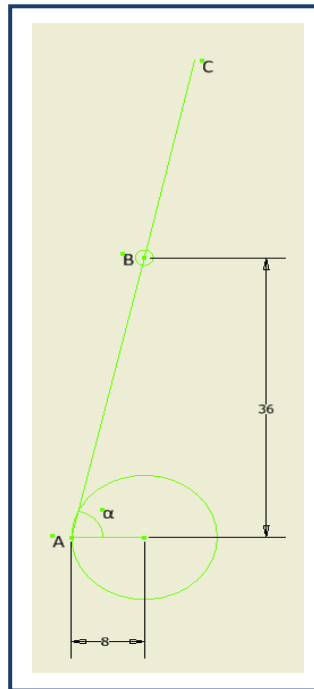


Fuente: Elaboración propia

4.3. DESPLAZAMIENTO DE LAS PATAS

Un aspecto importante en el diseño del robot es el movimiento que describen las patas. Sabiendo las distancias entre varios puntos podemos calcular cual es el desplazamiento que se produce en la cabeza de la pata, trasladándolo hasta la base. Como datos de partida, tenemos la dimensión de la barra que une los puntos A y C $L_{AC}=59\text{mm}$, siendo las distancias las que se muestran en la figura. De esta manera, podemos hallar el ángulo entre las barras (α), la distancia horizontal BC_x y la distancia vertical BC_y .

Figura 14. Distancias iniciales



Fuente: Elaboración propia

$\alpha = \text{tg}^{-1}\left(\frac{36}{8}\right) = 77,47^\circ$, este ángulo se mantiene para la barra BD por lo que podemos hallar la distancia vertical total $BD_y = 63 * \sin(77,47^\circ) = 57,6 \text{ mm}$, por lo que la distancia $CD_y=21,6 \text{ mm}$. Con el coseno podemos calcular la distancia $BD_x=12,8 \text{ mm}$, siendo $CD_x=4,8 \text{ mm}$.

De esta manera, según el ángulo que gire la barra AB tendremos los siguientes puntos a partir del punto B:

Tabla 2. Posición parte superior pata

β	X	Y
0	4,8	21,6
90	0	31
180	-4,8	21,6
270	0	15

Fuente: Elaboración propia

Debido a la arquitectura de la pata, el desplazamiento en el eje Y se traducirá en un movimiento en el eje Z, siendo este máximo para el ángulo de 270° y con un valor de 10mm. Debido al tipo de movimiento, entre los ángulos 0° y 180° se producirá un contacto constante con la superficie de apoyo.

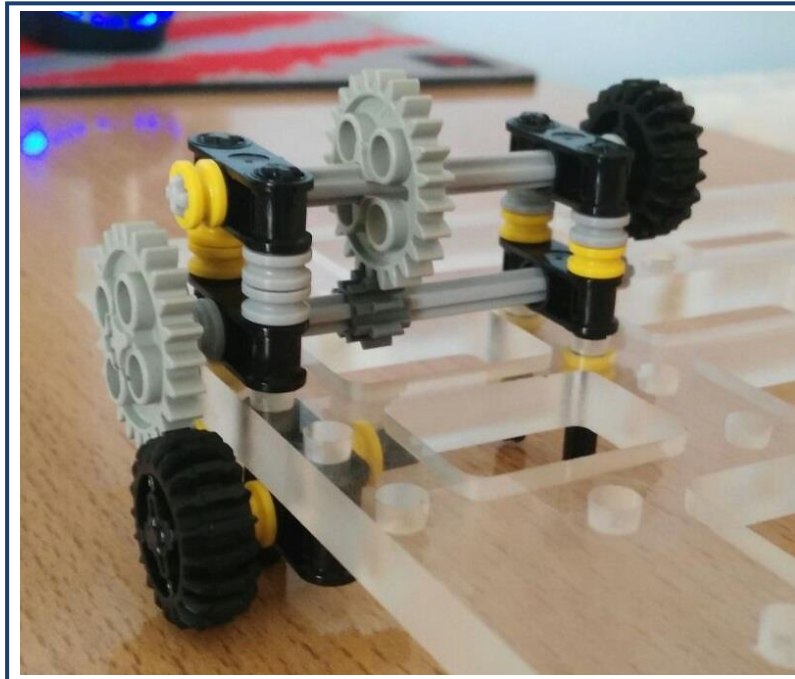
4.3. TRASMISIÓN DE LOS ESFUERZOS

Un aspecto importante a la hora de realizar el diseño de un objeto con movimiento, es la transmisión de los esfuerzos. Para este caso se ha seleccionado uno de los métodos más fiables y con menos pérdidas de potencia, los engranajes. Con estos, se pueden resolver los problemas en el movimiento ya que permiten aumentar la velocidad de giro del eje y llevar el movimiento a distintos planos.

La transmisión del movimiento del motor al eje se ha realizado mediante un tren de engranajes. Estos engranajes son de ejes paralelos, cilíndricos y de dientes rectos. Utilizando los engranajes adecuados se consigue aumentar la velocidad de rotación de los diferentes ejes, muy importante para la segunda fase de transmisión.

Normalmente se suelen utilizar ruedas dentadas dobles para optimizar el espacio pero en nuestro caso, LEGO no dispone de este tipo de piezas por lo que el tren se ha diseñado con los elementos disponibles de la siguiente forma:

Figura 15. Tren de engranajes



Fuente: Elaboración propia

De esta manera, según la siguiente ecuación, podemos calcular la velocidad de giro de la pata.

$$N_1 * Z_1 = N_2 * Z_2$$

Siendo N=Velocidad de giro y Z=Número de dientes. Aplicando la ecuación a las distintas etapas de transmisión obtenemos el valor de la velocidad en el primer eje (N4).

$$N_2 = N_1 * \frac{Z_1}{Z_2} = 75 * \frac{24}{20} = 90 \text{ RPM}$$

$$N_3 = N_2 * \frac{Z_3}{Z_4} = 90 * \frac{20}{12} = 150 \text{ RPM}$$

$$N_4 = N_3 * \frac{Z_5}{Z_6} = 150 * \frac{24}{20} = 180 \text{ RPM}$$

Siendo la velocidad del motor a 6V y sin carga de 75 RPM aproximadamente. Resaltar algo imprescindible para que se pueda producir la transmisión y es que ambos engranajes deben de tener el mismo módulo. Este parámetro queda definido como $M = \frac{D_p}{Z}$, y en nuestro caso tiene un valor de 1, ya que el diámetro primitivo del engranaje D_p coincide con el número de dientes.

Para la segunda etapa de transmisión, el objetivo consiste en trasladar el movimiento del primer eje (N4) a un segundo eje perpendicular al primero (N5). Esto se consigue mediante el sistema de rueda y tornillo sin fin, ya que LEGO no dispone de otras alternativas para esta necesidad. Este mecanismo es utilizado para transmitir grandes esfuerzos a costa de reducir considerablemente la velocidad de giro del engranaje al cual está conectado y hacer el movimiento irreversible. En este caso, la expresión por la que se rige este mecanismo es:

$$N_4 * e_1 = N_5 * Z_7$$

Donde e_1 son las entradas del tornillo sin fin (hélices simples). De esta forma:

$$N_5 = 180 * \frac{1}{8} = 22.5 \text{ RPM}$$

Respecto al primer diseño, significa una velocidad 5,41 veces superior ya que aunque el motor se mueve a 75 RPM en lugar de 100 RPM, conseguimos sucesivas multiplicaciones de la velocidad antes de conectar al primer eje con velocidad (N4) como se muestra en las formulas anteriormente citadas. Como se puede observar en la última fórmula, la velocidad de movimiento se vuelve significativamente menor en el eje conectado al tornillo sin fin ya que necesitamos que este realice una vuelta entera para mover un único diente, o lo que es lo mismo, la velocidad de rotación del tornillo sin fin es N veces la del engranaje, siendo N el número de dientes del engranaje.

Una ventaja presente en este tren de engranajes es que se pueden modificar los tamaños de las ruedas, en función de si queremos una relación que proporcione más velocidad o menos. Cabe destacar que estos aumentos de velocidad tienen como consecuencia la reducción de la fuerza que se aplica, cuanto más se aumente la velocidad, mayor reducción en la fuerza motora se estará consiguiendo. Los engranajes Z_1 , Z_2 , Z_5 y Z_6 siempre serán los mismos, variando únicamente Z_3 y Z_4 . Así pues, obtendremos las siguientes relaciones:

Tabla 3. Relaciones de transmisión

Z3	Z4	Relación de transmisión (N4/N1)
24	8	4,32
16	16	1,44
20	12	2,4

Fuente: Elaboración propia mediante Excel

Para el tren de engranajes se podrían conseguir otras relaciones de transmisión variando la distancia entre los ejes y utilizando engranajes con distintos dientes, pero esto dificulta el engrane con el motor, el cual debería estar moviéndose entre distintos puntos para conseguir transmitir bien los esfuerzos. Se han seleccionado tres juegos que proporcionan distinta relación de transmisión con la misma distancia entre ejes. Importante reseñar que los cálculos se han realizado con la tercera configuración de la tabla y despreciando rozamientos. En cuanto a transmisión del par motor, encontramos las siguientes fórmulas:

$P = T * W$, en donde $W = \frac{2*\pi*N}{60}$ y $T = Ft * r$, siendo (T) el valor del par de fuerza y (W) la velocidad angular de giro del engranaje. En una transmisión, la potencia suministrada entre ejes se mantiene constante, variando los valores de T y W según tengamos una transmisión reductora o multiplicadora ($W_2 < W_1$ o $W_2 > W_1$ respectivamente). De esta manera, en el eje tendremos el siguiente par:

$$T_{motor} = 6.7Kg * cm = \frac{6.7*9.8}{100} = 0.66 N * m$$

$$P_{motor} = 0.66 * \frac{2*\pi*75}{60} = 5.18W$$

Esta potencia se transmitirá, en un caso ideal (libre de rozamientos) hasta el primer eje, siendo el par fuerza de este:

$$P_{motor} = P_{eje}, 5.18 = T_4 * W_4$$

$$T_4 = \frac{5.18}{2.4*7.85} = 0.28 N * m$$

4.4. LIMITACIONES LEGO

Durante el proceso de diseño de la estructura del robot, se observaron diversas limitaciones en las piezas de Lego, las cuales se enumeran a continuación.

- ▶ En primer lugar, Lego dispone de un número limitado de piezas, lo que hace que algunas uniones sean difíciles de implementar. A la hora de realizar un diseño con numerosos movimientos, el número de piezas se reduciría notablemente si existieran otras configuraciones sobretodo en la parte de conexiones entre piezas.
- ▶ Otra desventaja aparece a la hora de ofrecer rigidez a la estructura. Las uniones de diversas barras horizontales se puede implementar mediante la conexión de estas con otras perpendiculares. Ahora bien, por el tipo de orificios y conectores, se crea un mecanismo el cual no tiene todos sus grados de libertad restringidos, pudiéndose modificar la distancia entre barras. Este problema se podría resolver con barras en forma de U, pero como no están disponibles, se necesita una barra en forma de L y otra longitudinal. Para grandes distancias entre barras no se puede realizar estas uniones por falta de un modelo adecuado de pieza en L, por lo que se tiene que recurrir a un gran número de piezas extra encargadas de dar rigidez. Recordemos que Lego basa sus diseños con un ladrillo NTX inteligente como “cerebro” del robot, el cual sirve de estructura.
- ▶ En nuestro caso, al tener una estructura con numerosos ejes, se precisaba de un gran número de piezas para situar estos según las exigencias del modelo. Estas piezas extra junto a las encargadas de la rigidez elevaban la cantidad total de piezas y elementos de conexión, por lo que se decidió realizar el diseño de la estructura como se ha comentado anteriormente con el objetivo de reducir un gran número de pieza, simplificando el diseño.
- ▶ Al diseñar un elemento externo para utilizar en una estructura, el usuario está totalmente condicionado a la hora de elegir las distancias entre las piezas, teniendo que ser estas las establecidas por Lego. Esto dificulta la tarea de diseño y provoca que las piezas tengan mayores dimensiones. Lo mismo ocurre a la hora de utilizar los servos, es complicado adaptar la estructura a estos ya que no siguen las medidas de Lego con lo que se acaba aumentando las dimensiones del diseño. El uso de motores Lego hubiera supuesto un aumento de coste muy considerable.

5. ELECTRÓNICA

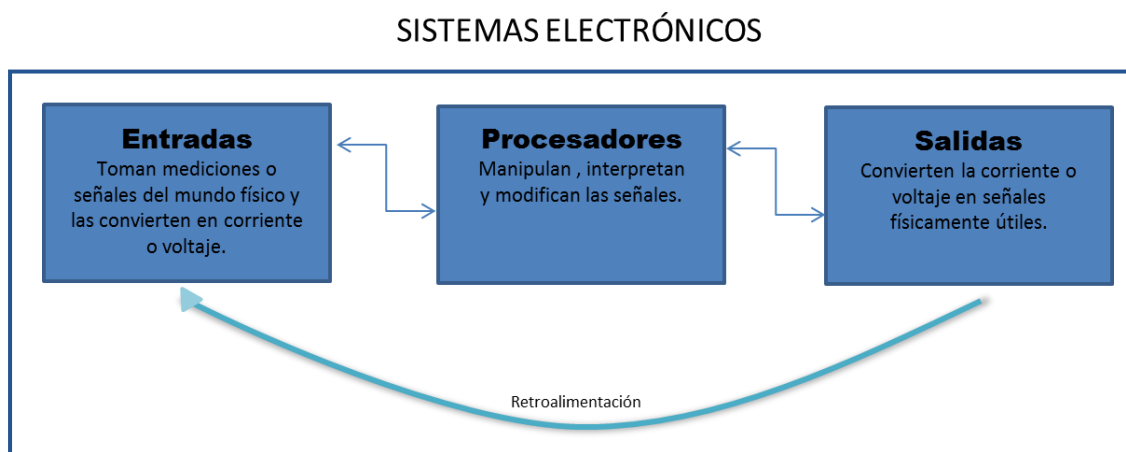
La electrónica es una de las partes más importantes del robot, ya que es esta la que podemos identificar como el cerebro y los sentidos que harán posible la interacción de nuestro robot con el entorno. Esta queda definida como rama de la física y especialización de la ingeniería, que estudia y emplea sistemas cuyo funcionamiento se basa en la conducción y el control del flujo de los electrones u otras partículas cargadas eléctricamente. El diseño y la construcción de circuitos electrónicos para resolver problemas prácticos forman parte de la electrónica y de los campos de la ingeniería electrónica, electromecánica y la informática en el diseño de software para su control.

Un sistema electrónico es un conjunto de circuitos que interactúan entre sí para obtener un resultado. Una forma de entender los sistemas electrónicos consiste en dividirlos en las siguientes partes:

- ▶ **Entradas o Inputs:** Sensores electrónicos o mecánicos que toman las señales (en forma de temperatura, presión, etc.) del mundo físico y las convierten en señales de corriente o voltaje.
- ▶ **Circuitos de procesamiento de señales:** Consisten en artefactos electrónicos ensamblados para manipular, interpretar y transformar las señales de voltaje y corriente provenientes de los sensores.
- ▶ **Salidas u Outputs:** Actuadores u otros dispositivos que convierten las señales de corriente o voltaje en señales físicamente útiles.

Básicamente se consigue en tres etapas: la primera (sensores), la segunda (circuito procesador) y la tercera (circuito actuador).

Figura 16. Esquema de un sistema electrónico



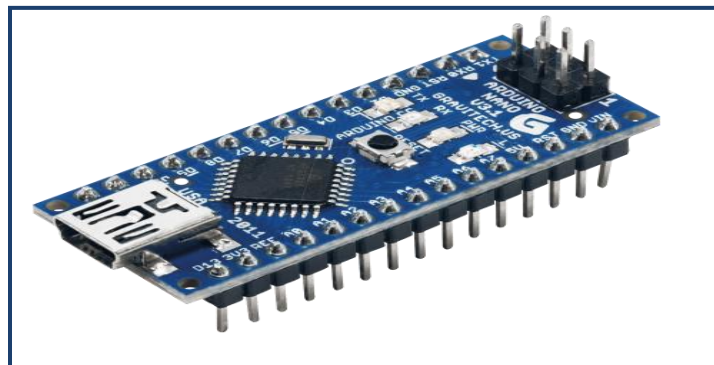
Fuente: Elaboración propia

Los diferentes elementos de los cuales se compone nuestro sistema electrónico son : placa *Arduino* nano, placa de expansión *funduino* nano 006 v3.0, módulo *bluetooth* HT-06, motores de rotación continua Power HD AR-3606HB, leds, regulador de tensión, resistencias, condensadores, una *protoboard* y adaptadores para pilas 6LR61 (9V) con salida *jack*.

5.1. ARDUINO

Para la parte de control de nuestro robot, se ha seleccionado una placa *Arduino NANO*. Estas placas se basan en un circuito impreso que incorporan un microcontrolador, normalmente *Amtel* (aunque podemos encontrar *ARM* o *Intel*), las cuales se han hecho famosas ya que *Arduino* es una compañía de hardware libre y a un precio asequible. Disponen de multitud de puertos digitales y analógicos para las distintas entradas/salidas y un entorno de desarrollo (*IDE*) que facilita su uso, por lo que han tenido una rápida difusión. El software está basado en un entorno *Processing* (lenguaje de programación y entorno de programación de código abierto) y un lenguaje basado en *Wiring*, encontrándose de forma gratuita en la página oficial de *Arduino*.

Figura 17. Placa *Arduino NANO*



Fuente: Electrotec

Para empezar a hacer uso de una de estas placas, solo es necesaria la conexión a un PC mediante USB e inicializar el software de la tarjeta una vez configurado. La configuración es sencilla ya que únicamente tenemos que seleccionar el tipo de placa que tenemos, su microcontrolador y el puerto establecido para la comunicación, en nuestro caso COM3. El usuario puede hacer uso de una gran cantidad de librerías que aportan distintas funciones, desde el control de distintos servomotores hasta leer SMS y mostrarlos por el monitor serie. Todo ello viene explicado en la propia página de la empresa y ofrecen una serie de tutoriales para explicar el funcionamiento de estas librerías.

La placa *Arduino NANO* es uno de los modelos más pequeños de este fabricante, el cual cubre las necesidades de nuestro proyecto. Las características más relevantes de este modelo son las siguientes:

Tabla 4. Características *Arduino NANO*

Características <i>Arduino NANO</i>	
Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)

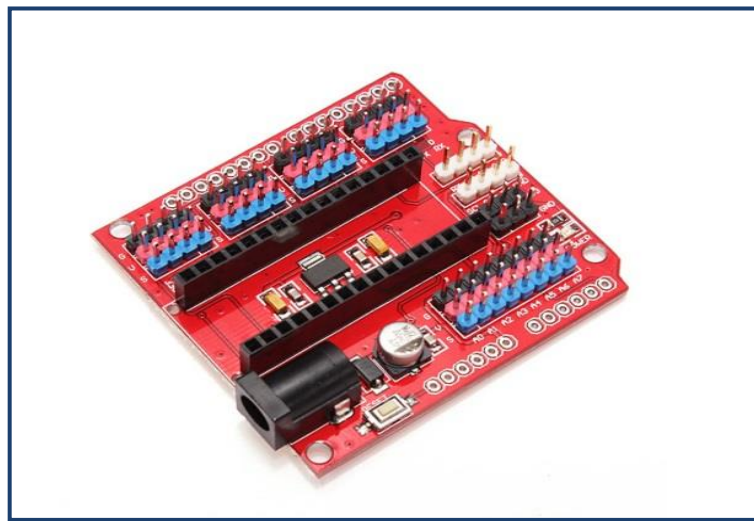
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz

Fuente: Elaboración propia a través de Arduino

5.2. PLACA DE EXPANSIÓN

Como placa de expansión, se ha elegido el modelo *funduino* nano 006 v3.0. Una placa de expansión o *shield* es una placa impresa que se puede conectar directamente a la placa de *Arduino* para aumentar sus capacidades. Este aumento de prestaciones puede girar en torno a distintas posibilidades como facilitar la conexión de diversos elementos u ofrecer una conexión vía *Ethernet* entre otras.

Figura 18. Placa de expansión *funduino* nano 006v3.0



Fuente: Banggood

Entre las prestaciones que nos ofrece esta placa encontramos la facilidad de conexión de los distintos elementos ya que contamos con 3 pines de conexión para cada uno (control, alimentación y puesta a tierra). De esta forma se evita el uso de una *protoboard* para algunas conexiones, añadiendo un elemento que nos permite la alimentación directa al sistema.

5.3. MOTORES

Hoy en día existe una gran variedad de motores según la aplicación a la cual están destinados. Los tipos más importantes utilizados en robótica suelen ser los servomotores, aunque también existen otras alternativas como por ejemplo los servos de rotación continua o los motores *brushless*, utilizados estos últimos en aerodelismo. Para este trabajo hemos considerado como mejor opción los servomotores de rotación continua, ya que con pocos podemos implementar el movimiento.

5.3.1. SERVOMOTOR DE ROTACIÓN CONTINUA

Los motores seleccionados para el movimiento son los Power HD AR-3606HB. Los servomotores de rotación continua son muy populares en el campo de la robótica, estos se caracterizan por la capacidad de realizar giros de 360° mientras que los servos estándar solo giran 180°. Estos pequeños

motores se alimentan a través de corriente continua y *Arduino* nos permite controlar la velocidad de giro con la librería servo.h.

El motor está compuesto por diversos elementos y presentando las siguientes características:

Tabla 5. Características PowerHD AR-3606HB

Características Power HD AR-3606HB	
Size	40.5 × 20.0 × 38.0 mm
Free-run current @ 6V:	200 mA
Speed @ 6V:	0.14 sec/60°
Stall torque @ 6V:	93 oz·in-6.7kg·cm
Speed @ 4.8V:	0.16 sec/60°
Stall torque @ 4.8V:	83 oz·in-6 kg·cm

Fuente: Elaboración propia a partir de Polulu

Estos motores se seleccionaron como alternativa a los de un principio ya que no disponían de suficiente fuerza para el segundo diseño. Como se puede observar en las tablas, los segundos motores ofrecen una fuerza 4 veces mayor, sin tener grandes repercusiones en la velocidad que pueden alcanzar. Además esto no es problema ya que se corrige con el tren de engranajes, pudiendo aumentar la velocidad del eje de transmisión hasta un valor de 4,32.

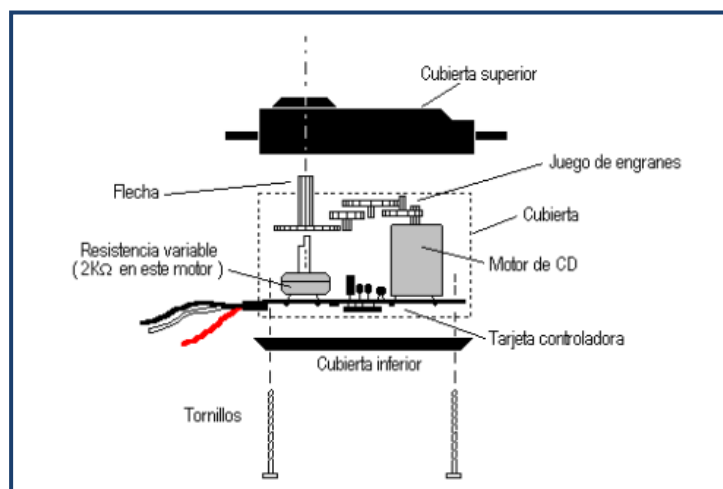
Tabla 6. Características FS90r

Características FS90R	
Size	23.2 × 12.5 × 22 mm
Free-run current @ 6V:	200 mA
Speed @ 6V:	0.07 sec/60° ²
Stall torque @ 6V:	21 oz·in-1.5 kg·cm
Speed @ 4.8V:	0.09 sec/60° ³
Stall torque @ 4.8V:	18 oz·in-1.3 kg·cm

Fuente: elaboración propia a partir de Polulu

Mientras que su esquema de construcción sería:

Figura 19. Estructura interna de un servomotor



Fuente: Slideshare

En este esquema podemos apreciar los 3 cables, dos de ellos para la alimentación y otro para la señal de control, estando este último conectado a la placa de expansión.

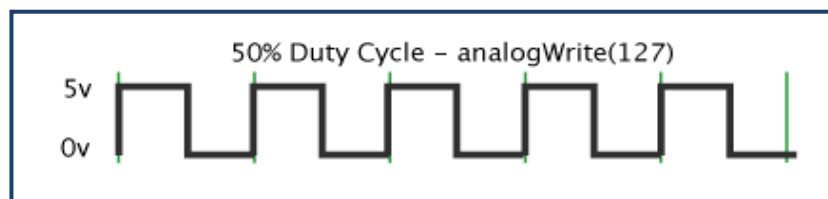
La gran diferencia de este tipo de motores con los servos de 180° de rotación se centra en dos elementos. En primer lugar, el juego de engranajes del cual disponen los servos de 180° presenta un tope en una de las ruedas, de manera que no es posible girar un ángulo mayor. En segundo lugar encontramos una gran diferencia en el potenciómetro. En los servos de rotación continua, únicamente tenemos que seleccionar el valor del potenciómetro girando este hasta la posición intermedia. Este valor corresponderá con el punto en el que el servo está en reposo, a partir de este punto, podremos girar en un sentido u otro dando diferentes valores a la función. En los servos normales, el potenciómetro está conectado con el tren de engranaje interno, de forma que este gira al girar el motor. De esta manera, en función de los grados que gire, la resistencia tendrá un valor u otro, correspondientes a la posición del motor.

5.3.2. PWM

La modulación por ancho de pulso, en inglés *pulse width modulation* (PWM), es una técnica que logra producir el efecto de una señal analógica sobre una carga, a partir de la variación de la frecuencia y ciclo de trabajo de una señal digital. El ciclo de trabajo hace referencia al tiempo que la señal está en estado activo, que es un determinado porcentaje de la duración total del ciclo. Al cambiar la señal digital de estado alto a bajo a una tasa lo suficientemente rápida y con un cierto ciclo de trabajo, la salida parecerá comportarse como una señal analógica constante cuanto esta está siendo aplicada a algún dispositivo. Así pues, obtendremos el máximo valor de voltaje si la señal permanece en alto todo el ciclo, la mitad de este valor si está en estado alto solo la mitad de un ciclo o apagado si permanece todo el ciclo en estado bajo.

En las placas *Arduino* se puede hacer uso de este tipo de modulación de forma sencilla con la función `analogwrite(valor)` e introduciendo un valor comprendido entre 0 y 255, o bien con la función `servo`. El ciclo de la señal dura aproximadamente 2 ms o lo que es lo mismo, tiene una frecuencia de 500Hz. El estado alto corresponde a un valor de 5V. En la siguiente figura podemos observar como conseguimos un valor de 2,5 Voltios haciendo que la señal este activa solo 1ms del total del ciclo.

Figura 20. Ejemplo PWM



Fuente: *Arduino*

5.4. MÓDULO BLUETOOTH

El módulo seleccionado para la comunicación con el teléfono móvil es el HC-06. Este módulo es muy popular para aplicaciones con microcontroladores PIC y *Arduino*. Se trata de un dispositivo relativamente económico y que habitualmente se vende en un formato que permite conectarlos a la placa *Arduino*. En la actualidad podemos encontrar este tipo de transmisión de datos en la mayoría de los dispositivos ya que permite la conexión de una forma sencilla, inalámbrica, segura y con elevada

velocidad de transmisión de datos. Estos dispositivos se caracterizan por una serie de parámetros, siendo estos:

Tabla 7. Características HC-06

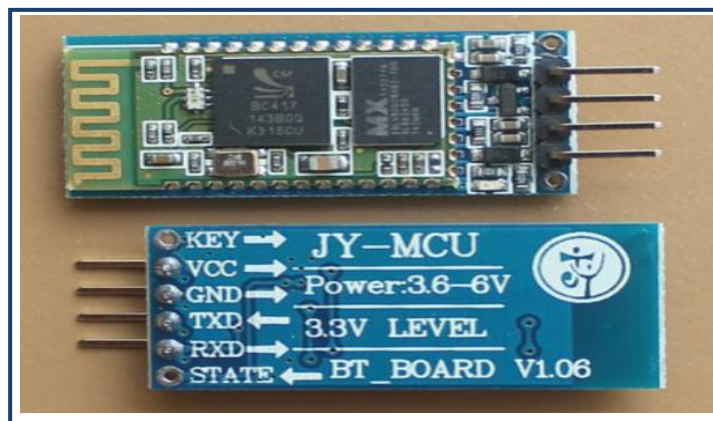
Características módulo <i>bluetooth</i> HC-06	
Compatible con el protocolo	Bluetooth V2.0.
Voltaje de alimentación:	3.3VDC – 6VDC.
Voltaje de operación:	3.3VDC.
Baudrate ajustable:	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.
Corriente de operación:	< 40 mA

Fuente: Elaboración propia a partir de Electronilab

En nuestro caso, se transmitirán cadenas de caracteres desde el móvil a la placa *Arduino* cuando se produzca una variación en el acelerómetro o cuando pulsemos un botón de nuestra aplicación. El módulo funcionará como esclavo, es decir, la conexión tendrá que ser realizada desde el teléfono al seleccionar la dirección. La conexión a la placa de expansión es cruzada, conectándose el terminal RX al TX y viceversa, mientras que los otros dos terminales serán para la alimentación del dispositivo. La velocidad de comunicación será de 9600 baudios, medida utilizada en telecomunicaciones para indicar el número de símbolos por segundo. Por último, mencionar la necesidad de adaptar el voltaje de salida de nuestro puerto TX al RXD del módulo *bluetooth*. Para ello aplicamos un sencillo divisor de tensiones que descienda el voltaje de 5V a 3.3V implementado con una resistencia de 1000Ω en paralelo con otras dos del mismo valor pero conectadas en serie. La fórmula del divisor de tensiones es:

$$V_{out} = V_{in} * \frac{R2}{R1 + R2}$$

Figura 21. Módulo HC-06



Fuente: Ebay

Siempre que se descargue un programa a la placa *Arduino*, los pines RX y TX deben estar desconectados. De no hacer esto aparecerá un error en el compilador que impedirá actualizar el programa.

5.5. OTROS COMPONENTES

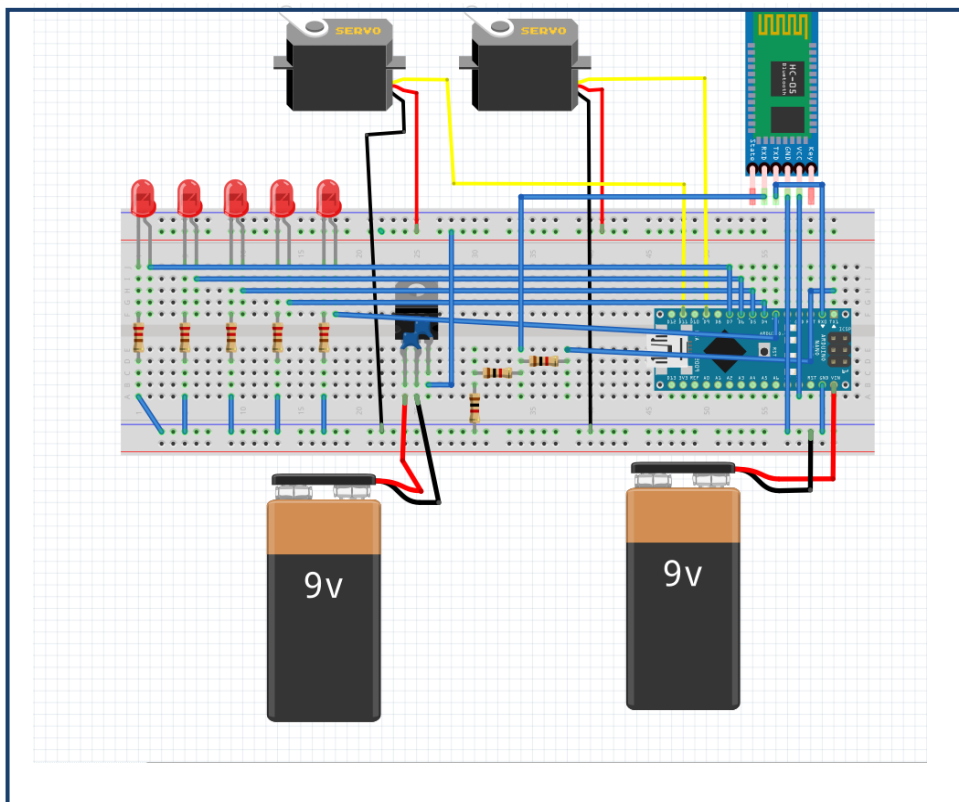
Como es habitual en el campo de la electrónica, es necesario utilizar un gran número de componentes sencillos. En este caso en concreto, se han utilizado distintos tipos de leds para hacer el trabajo más visual y atractivo de cara al público. También se ha precisado de una placa de conexiones (*protoboard*) para conectar los diferentes elementos a las resistencias que permiten la adaptación de la tensión de salida a las distintas necesidades.

Para la alimentación de los motores se ha utilizado un circuito integrado regulador de tensión positiva L7806. Este componente es muy común en fuentes de alimentación ya que nos permite rebajar la tensión de entrada desde los 30V hasta el voltaje que queramos a la salida, en este caso 6V. Para que los motores funcionen, se precisa también de dos condensadores que aseguren el filtrado de la fuente y una correcta regulación de la tensión, conectando uno a la entrada y otro a la salida. Los valores de estos condensadores son de $0.33\mu\text{F}$ y $0.1\mu\text{F}$, respectivamente. Con esto conseguimos mayores prestaciones por parte de los motores ya que la alimentación es más eficiente al quitarle carga a la placa *Arduino* y porque estos se alimentan a una tensión de 6V en lugar de 5V, valor máximo de salida de este tipo de placas.

5.6. CONEXIONES

Para las conexiones eléctricas, se ha utilizado una *protoboard* de 270 contactos. Las conexiones se han realizado siguiendo el siguiente esquema:

Figura 22. Esquema de conexiones



Fuente: Elaboración propia a partir de Fritzing

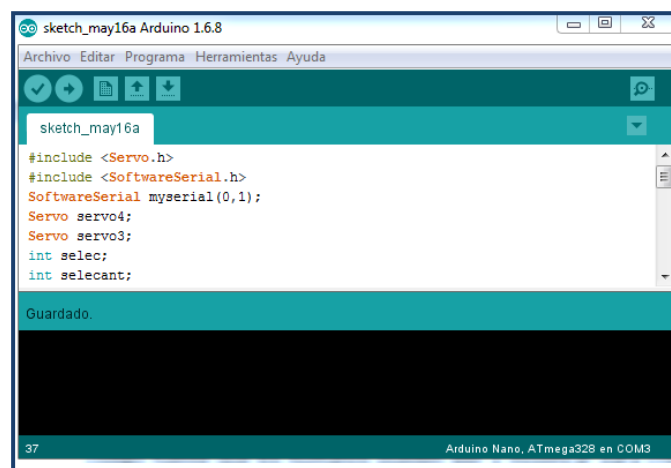
En la figura podemos observar la forma de conexión de los distintos elementos. La placa de expansión, no se pudo incluir en el esquema ya que el programa no tiene el modelo para su implementación. Los distintos leds se conectan a 5 pines digitales por el ánodo, estando el cátodo conectado a una resistencia de 220Ω, encargada de reducir la tensión a 3V, seguido de la puesta a tierra. En la figura encontramos un divisor de tensiones, para reducir la tensión del pin TX de 5V a 3,3V para el correcto funcionamiento del módulo *bluetooth*. Los motores están conectados a un pin digital, encargado del control del motor, la puesta a tierra y un pequeño circuito que permite la alimentación de estos con una fuente distinta a la de la placa *Arduino*. Por último, la alimentación de la placa no se produce como se muestra en la figura, si no que se produce conectando la pila directamente a la placa de expansión mediante una conexión de *Jack*. En la figura se puede observar como la toma a tierra es la misma para todos los componentes en el esquema ya que la línea de pines esta interconectada, y esta a su vez se conecta al propio puerto GND de la placa *Arduino*.

Este esquema ha sido desarrollado con la aplicación gratuita *Fritzing*, disponible en internet. Con esta aplicación se puede hacer uso de una gran cantidad de elementos para los esquemas de conexión como placas *Arduino*, *shields* o componentes eléctricos. También nos permite elaborar nuestros propios componentes y elegir la representación del dibujo.

6. PROGRAMACIÓN ARDUINO

Definimos un programa informático como el conjunto de instrucciones escritas que realizará un tipo de hardware, ejecutadas en un procesador central. El mismo programa tiene dos tipos de formato: uno ejecutable que la computadora utiliza para realizar las instrucciones y otro en código fuente que el programador puede leer y modificar para crear los distintos algoritmos. De esta manera llamaremos software a la agrupación de distintos programas informáticos con sus datos relacionados. Llamamos código fuente a las distintas instrucciones escritas en un lenguaje de programación, y que posteriormente será convertido a un archivo ejecutable por un compilador y más tarde ejecutado por la unidad central de procesamiento.

Figura 23. Compilador *Arduino*



Fuente: Elaboración propia

En nuestro caso, como se ha mencionado anteriormente, el entorno de programación es *Processing*. Este es un entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y

que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital. Dentro de este entorno de programación surge el lenguaje *Wiring*, caracterizado por ser un lenguaje de entradas y salidas, en el cual tenemos que definir los distintos elementos en función del flujo de información y luego hacer uso de ellos mediante una serie de instrucciones y librerías. Por último, *Arduino* pone a nuestra disposición un compilador propio en el cual podemos configurar fácilmente el tipo de placa y el puerto de comunicación con nuestro ordenador. La estructura de estos programas se mantendrá, estando organizada en tres partes: una primera de inicialización, donde introducimos las librerías y algunas funciones; una segunda etapa de configuración o *setup*, donde configuramos los pines de la tarjeta y una tercera de bucle, donde programamos el bucle que se repetirá de forma indefinida.

6.1. LIBRERÍAS

El entorno de programación se puede extender con el uso de bibliotecas, al igual que la mayoría de plataformas de programación. Las bibliotecas nos proporcionan algunas funcionalidades adicionales, como por ejemplo, manipular algún tipo de hardware. *Arduino* pone a nuestra disposición un gran número de estas, pudiendo nosotros incluir otras con la opción **Programa>Incluir librería>Añadir librería.ZIP** desde el compilador. En la página oficial de *Arduino* y en otros sitios de la red, podemos encontrar información y tutoriales para comprender su funcionamiento con programas sencillos. De esta forma, para la realización de nuestro programa se ha hecho uso de dos de estas librerías: *Servo.h* y *SoftwareSerial.h*.

- ▶ ***Servo.h***: Esta librería permite controlar los servomotores con una serie de comandos sencillos. Los servos pueden ser de rotación continua o no, sirviendo las mismas instrucciones para ambos, y pudiendo controlar un total de 12 servomotores distintos. Las distintas instrucciones que incluye son:
 - ***Servo***: Con esta función podemos nombrar a nuestro servo en la etapa de inicialización del programa.
 - ***servoname.attach(pin)***: Adjunta el servo anteriormente nombrado a un pin de la tarjeta.
 - ***servoname.write(0-180)***: En los servos de rotación continua, permite controlar la velocidad de giro. De esta manera el valor 0 corresponde a un giro anti horario a la máxima velocidad y 180 el giro en sentido horario a máxima velocidad, el valor 90 es el punto donde no existe movimiento. En los servomotores por el contrario, el valor entre 0 y 180 hace referencia al ángulo que girará el motor.
- ▶ ***SoftwareSerial.h***: Esta librería permite establecer una comunicación en serie en los pines 0 y 1, encargados de recibir y enviar la información. Esta comunicación se puede establecer gracias a una pieza integrada en el hardware llamada UART. También permite establecer otros pines de la tarjeta para esta función a unas velocidades de 115200 bps, aunque no se pueden recibir distintos datos simultáneamente. Las distintas instrucciones que se incluyen en esta biblioteca son:
 - ***SoftwareSerialname(RX,TX)***: Con esta función podemos nombrar a nuestro módulo *bluetooth* en la etapa de inicialización del programa. Los parámetros X e Y indican los pines que se encargarán de hacer la recepción y el envío de datos RX y TX respectivamente.

- *Name.begin(bps)*: Con este comando inicializamos el módulo *bluetooth* que hemos nombrado a una velocidad determinada en bps. Las velocidades que este tipo de comunicación soporta son: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 31250, 38400, 57600, y 115200.
- *Name.available()*: Esta función nos devuelve el número de bits disponibles para realizar la lectura desde el puerto serie. Es muy útil para comprobar el estado del módulo antes de ejecutar el programa. La función devuelve un número de bits, siempre mayor de 0 en caso de no producirse fallo.
- *Name.read()*: Mediante el uso de esta función podemos leer la información que llega al dispositivo *bluetooth* y almacenarla en una variable. Recordar que solo puede llegar información de un solo dispositivo al mismo tiempo.
- *Name.write()*: Mediante el uso de esta función podemos escribir la información que deseamos enviar al dispositivo conectado.
- *Name.listen()*: Ya que únicamente podemos recibir información de un dispositivo de forma simultánea, es necesario una función que seleccione dicho dispositivo.

Por último, mencionar algunas de las funciones no pertenecientes a las librerías pero utilizadas en el programa. Estas son: *PinMode(pin,modo)* y *DigitalWrite(pin,valor)*.

- ▶ *PinMode(pin,modo)*: Con esta función podemos configurar un pin a un valor determinado, dependiendo si es salida (output) o entrada (input).
- ▶ *DigitalWrite(pin,valor)*: Nos permite modificar el estado de un pin digital. Estos pines puede cambiar de estado apagado (LOW) a encendido (HIGH), entregando 0 ó 5V respectivamente.

6.2. PROGRAMA

Para la realización del programa se ha buscado una estructura no muy compleja para que resulte fácil de entender y reproducir, de cara a los alumnos. Así pues, dentro del bucle del programa se han establecido una serie de igualdades para diferenciar cada uno de los posibles casos. El programa completo se encuentra como anejo, elaborándose un desglose del mismo en este apartado.

En primer lugar se realiza la inicialización del programa. En esta parte se incluyen las distintas librerías a utilizar con el comando *#include*, se nombra tanto el módulo *bluetooth* como los motores y se crean algunas variables que serán utilizadas más adelante.

```
#include <Servo.h>
```

```
#include <SoftwareSerial.h>
```

```
SoftwareSerialmyserial(0,1);
```

```
Servo servo4;
```

```
Servo servo 3;
```

```
int selec;
```

```
int seλεκant;
```

La etapa *setup* sirve para configurar los distintos pines de la tarjeta a cada uno de los elementos. De esta manera, los pines 9 y 11 se encargarán del funcionamiento de los motores, mientras que los pines 3, 4, 5, 6 y 7 serán para los leds. En esta etapa también se configura el módulo *bluetooth* a una velocidad de 9600 bps que es a la que suelen tener por defecto.

```
void setup()
{
  servo3.attach(9);
  servo4.attach(11);
  myserial.begin(9600);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);
}
```

En la última etapa se implementa el bucle que se repetirá de forma infinita, una vez pasadas las dos etapas anteriores. En esta parte se ha intentado simplificar el programa, para hacerlo más entendible de cara al público. Como se puede observar, únicamente entra en funcionamiento si la orden *if(myserial.available(>0))* se cumple. Esto significa que el módulo está funcionando correctamente y está preparado para recibir n número de datos, produciéndose a continuación la lectura del carácter *que entra y almacenándolo en una variable llamada selec*.

```
void loop()
{
  if(myserial.available(>0)){
    selec=myserial.read();
  }
}
```

A continuación se entra en los bucles que dan las distintas órdenes a los motores y leds para el funcionamiento de nuestro robot. Así pues el primer paso es comprobar que ha habido un cambio en la orden para no detener el movimiento, ya que de no tener este paso se tendría que estar continuamente pulsando el botón. El siguiente paso sería la comparación del valor almacenado en la variable *selec* con las diferentes condiciones de cada bucle, ejecutándose aquel que cumpla la igualdad impuesta. En el siguiente código mostramos un caso, ya que todos presentan la misma pauta, variando el valor de las órdenes:

```
if (selec=='b'){
  servo3.write(0);
```

```
servo4.write(180);
```

```
digitalWrite(3,HIGH);
```

En la cual encontramos la comparación y las órdenes oportunas para el funcionamiento de los motores y de los leds. Finalmente, se actualiza la variable `selecant=selec` y se cierra el bucle principal. Las órdenes enviadas a los motores dependerán del movimiento que se quiera implementar. De esta forma jugando con la velocidad y sentido de giro del motor se conseguirán los giros u otros movimientos. Como es lógico, el sentido de giro de los motores ha de ser contrario entre si para avanzar en una dirección determinada ya que la disposición de estos es inversa.

7. APLICACIÓN MÓVIL

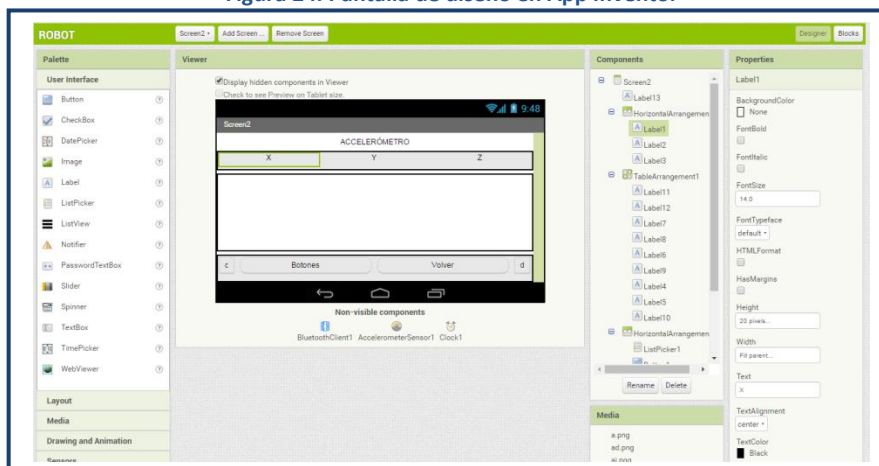
En la actualidad, los móviles se han convertido en herramientas fundamentales que necesitamos en el día a día, ya sea por ocio o por trabajo. Los distintos sistemas operativos móviles como *Android* o *iOS*, ponen a nuestra disposición la posibilidad de crear aplicaciones propias para ejecutarlas luego en nuestros terminales. Así pues, para la comunicación con el módulo *Arduino* se optó por una aplicación móvil, la cual mediante el uso de sensores como el acelerómetro o la pantalla táctil, enviara las instrucciones necesarias para el control del robot. Esta aplicación se ha desarrollado mediante el software *AppInventor*, disponible de forma gratuita en internet.

7.1.-APPINVENTOR

Este programa, creado por *Google*, aporta diferentes herramientas que permiten crear una aplicación compatible con el sistema operativo *Android*, perteneciente a la misma compañía. El sistema se basa en una programación gráfica que tiene dos partes: una zona donde se introducen aquellos elementos que más tarde se programarán y otra donde realizamos el programa con bloques predefinidos.

De esta manera, en la primera parte se encuentran múltiples funcionalidades como son la de introducción de distintos objetos de interacción con el usuario (botones, listas, textos, etc.), herramientas para organizarlos en la pantalla, sensores del propio teléfono, herramientas de conexión, etc. Con la combinación de todos estos definimos el aspecto que tendrán cada una de las pantallas de nuestra aplicación cuando se utilice.

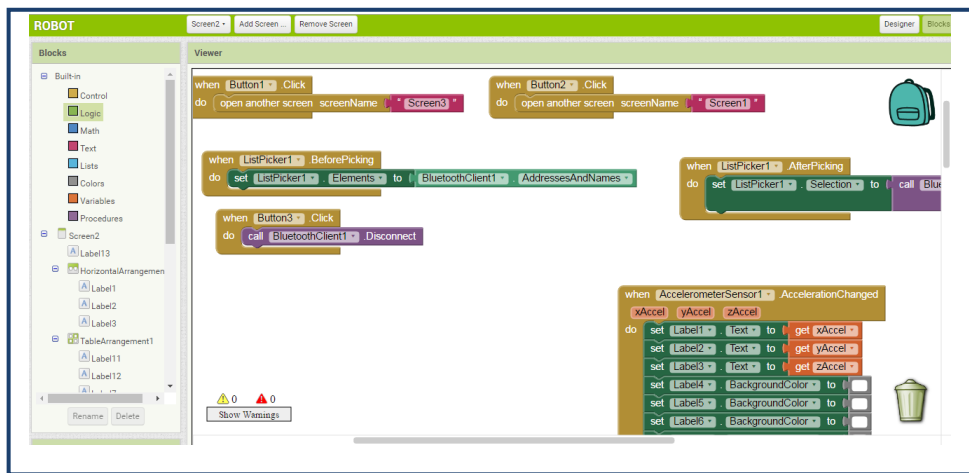
Figura 24. Pantalla de diseño en App Inventor



Fuente: Elaboración propia a partir de App Inventor

En la segunda parte se pasa a la programación de los objetos y herramientas introducidas. Para esta parte el programa hace uso de la librería *Open Blocks* de Java, la cual nos permite un lenguaje de programación visual basado en bloques. Cada elemento tiene sus funciones predefinidas que permiten al usuario el uso de estructuras de control, lógicas, matemáticas, modificación de textos, etc. Mediante la combinación de estos bloques a modo de puzle, se pueden crear programas complejos en menor tiempo que con otros lenguajes de texto. El único problema que se presenta es que las aplicaciones pueden resultar limitadas en algunos puntos, aunque para el desarrollo de la aplicación ha satisfecho las necesidades que se presentaban. App Inventor diferencia los distintos tipos de orden por colores para facilitar la interpretación de los comandos, como se puede apreciar en la siguiente imagen.

Figura 25. Pantalla de programación App Inventor

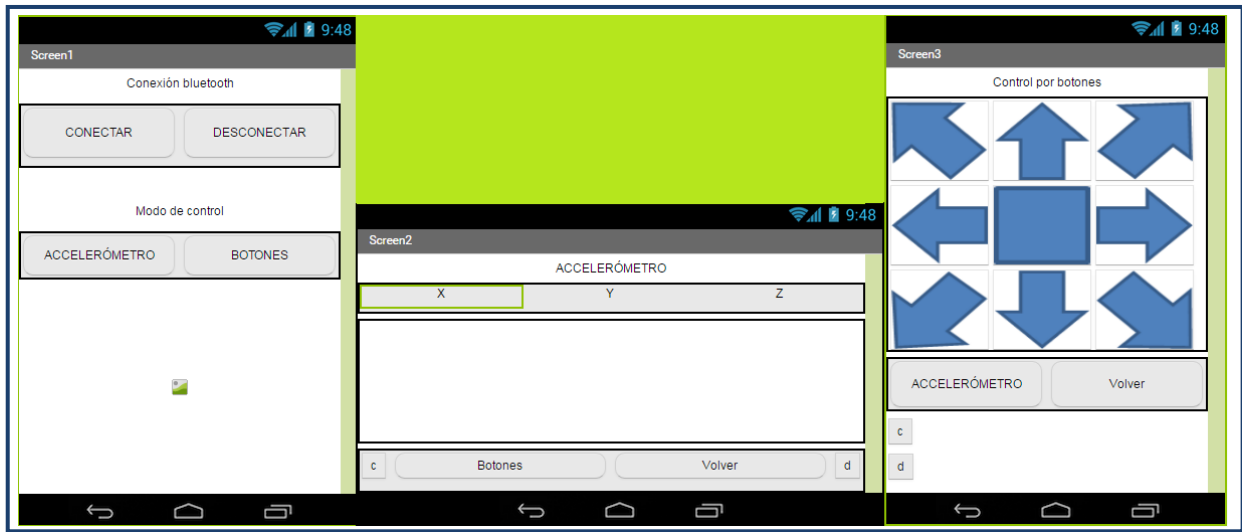


Fuente: Elaboración propia a partir de App Inventor

7.2. ASPECTO Y PROGRAMACIÓN

La aplicación desarrollada presenta 3 pantallas diferentes, una para seleccionar el modo de control del robot y las otras dos para el control mediante con distintos sensores. La conexión al módulo *bluetooth* se ha de realizar en cada una de las pantallas ya que si no puede dar problemas de conexión, después del uso, también será necesaria la desconexión. De esta manera, el aspecto de las diferentes pantallas es el siguiente:

Figura 26. Pantallas de la aplicación

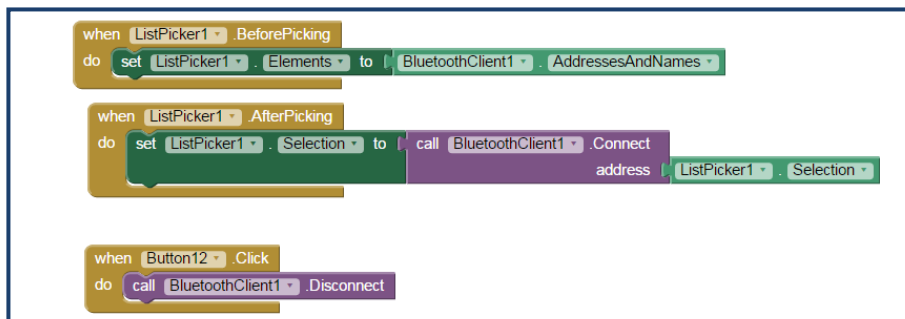


Fuente: Elaboración propia a partir de App Inventor

En cuanto a la programación, las tres pantallas presentan los mismos comandos para realizar la conexión mediante *bluetooth*, diferenciándose en el uso de los sensores. Por consiguiente, se detallará la programación de las partes diferenciadas.

- ▶ En primer lugar encontramos la programación correspondiente a la parte de conexión con el módulo *bluetooth*. Esta se implementa con los siguientes bloques:

Figura 27. Conexión *bluetooth*

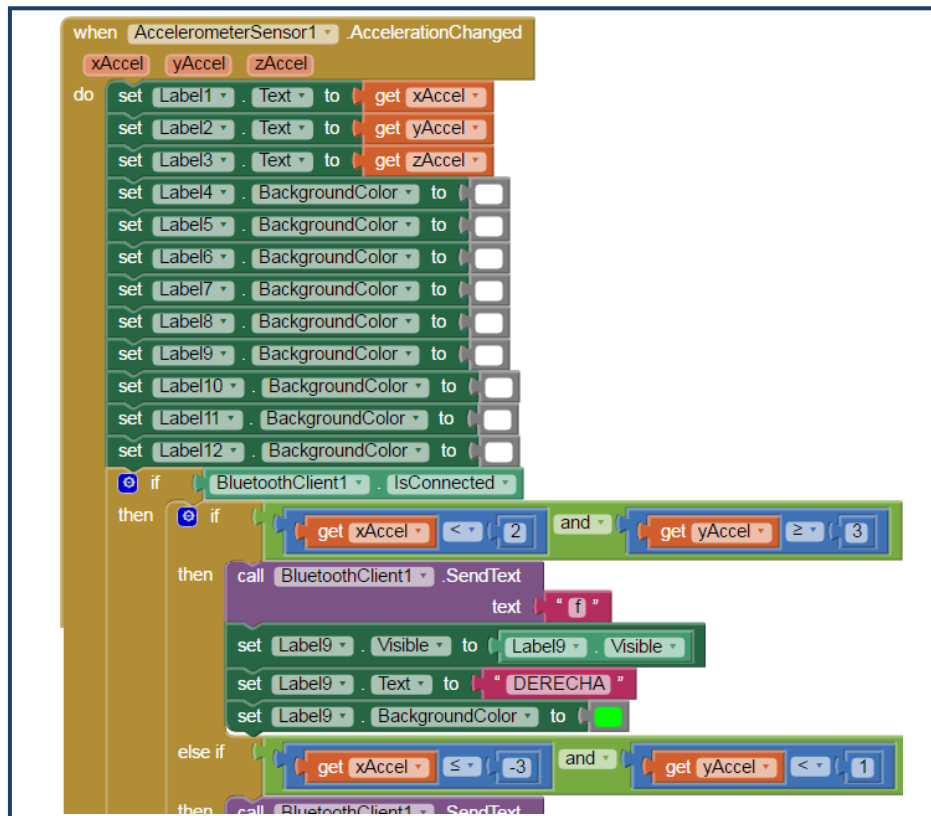


Fuente: Elaboración propia a partir de App Inventor

Antes que nada, debemos enlazar nuestro dispositivo con el módulo *bluetooth* mediante la introducción de la contraseña. Una vez realizado este paso, al pulsar el elemento **ListPicker1** de nuestra pantalla, aparecerá una lista con todos los aparatos enlazados a nuestro dispositivo móvil, en donde seleccionaremos el correspondiente para realizar la conexión (primer bloque). Al pulsar el elemento, **ListPicker1>Selection**, el programa llama al *bluetooth* y lo conecta a la dirección correspondiente (segundo bloque). Por último, se dispone de un botón para realizar la desconexión.

- ▶ Para la parte de control del robot mediante el acelerómetro del teléfono móvil, se encuentra la siguiente programación:

Figura 28. Programación control mediante acelerómetro

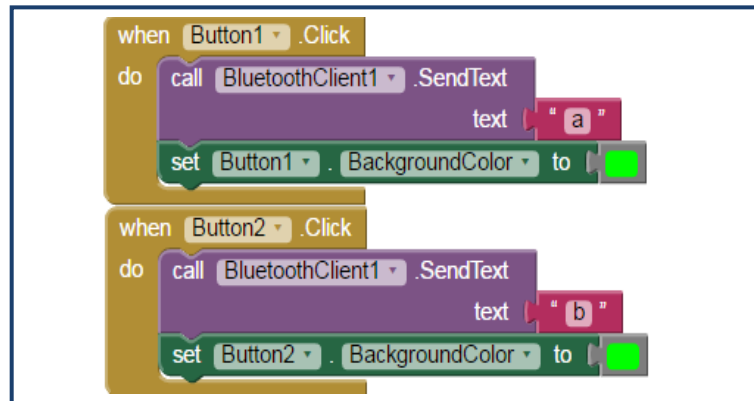


Fuente: Elaboración propia a partir de App Inventor

En esta parte, se puede observar como a cada cambio de aceleración se obtiene una respuesta del programa. En este tenemos en primer lugar 3 cuadros (**Label1**, **Label2** y **Label3**) que nos muestran la información del acelerómetro en todo momento, así como el cambio de color de los diferentes cuadrados. Estos hacen referencia a la matriz 3X3 en que hemos dividido el cuadrado grande de la **figura26>Screen2**. A continuación, si el *bluetooth* está conectado, pasamos a la parte de comparación de los distintos bucles. Cuando los valores de **xAccel** e **yAccel** cumplan con las inecuaciones, entraremos en el bucle de control correspondiente, enviando un carácter determinado, cambiando el texto del cuadro después de hacerlo visible y cambiando el fondo a color verde. De esta forma, cuando se esté produciendo un movimiento, veremos el cuadro correspondiente iluminado y con el texto de la operación.

- ▶ Para la parte de control mediante la pantalla táctil, se dispone de la siguiente programación:

Figura 29. Programación control mediante botones

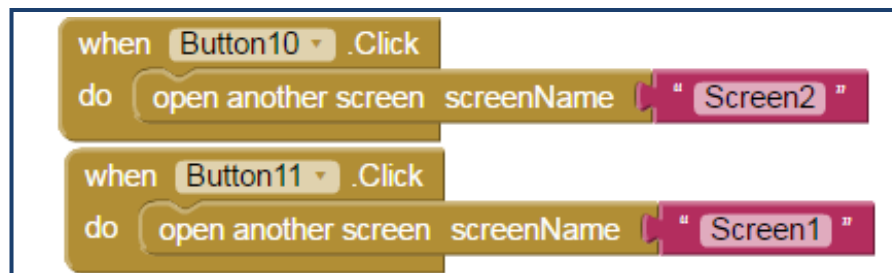


Fuente: Elaboración propia a partir de App Inventor

En esta parte podemos observar como el programa es mucho más sencillo. Cuando se pulse un botón, automáticamente se mandará el carácter correspondiente por *bluetooth*.

- ▶ Por último, se encuentran los comandos que nos permiten saltar de una pantalla a otra. Al pulsar el botón correspondiente, se envía una orden de control que salta a la pantalla seleccionada.

Figura 30. Programa selección de pantalla



Fuente: Elaboración propia a partir de App Inventor

8. RESULTADOS

Considerando que el objetivo del proyecto es el diseño de un robot hexápodo implementado con Lego y *Arduino* dirigido a los alumnos de educación secundaria para colaborar en el aprendizaje de diversas materias relacionadas con la tecnología, como mecanismos, transmisiones, electrónica, programación, etc., en mi opinión, el robot construido cumple ampliamente con estas expectativas, siendo un buen medio para adquirir de forma amena y motivante estos conocimientos.

Como en todo proyecto que supone diseño y construcción, han ido surgiendo problemas que han habido que solventar a base de ingenio y soluciones técnicas, lo que lo ha convertido en una experiencia enriquecedora y motivante, y este mismo es el objetivo que nos planteamos de cara a esos escolares que esperamos que aprendan los fundamentos de la mecánica y otras tecnologías de forma amena y divertida.

9. CONCLUSIONES

Después de realizar el trabajo se han obtenido una serie de conclusiones, las cuales se van a comentar a continuación.

En primer lugar, se ha podido observar como una alternativa distinta al movimiento normal mediante ruedas complica sustancialmente el trabajo. Para la realización de un pequeño robot con ruedas no se necesitan etapas de transmisión ya que la velocidad y fuerza de los motores es suficiente para su movimiento, además este se produce a elevada velocidad ya que solo se necesita conectar la rueda a la salida del motor. En el caso de las patas se necesita transmitir una elevada fuerza a una velocidad adecuada, con lo que intentar producir el movimiento sin usar gran número de motores se vuelve complicado. Para ello hemos necesitado un tren de engranajes y una transmisión a base de tornillos sin fin, estando limitados en cuestión de espacio y en la conexión del tren con el motor. Así mismo, se ha podido comprobar como las piezas de lego no son del todo ideales para este tipo de movimiento ya que se producen elevadas fricciones en los distintos ejes y es difícil restringir totalmente el movimiento de las piezas que sustentan el eje.

En segundo lugar, se ha observado la gran utilidad de las placas *Arduino*, así como su versatilidad. Estos microcontroladores nos ofrecen una gran variedad de funciones ya que prácticamente cualquier proceso es programable, desde trabajos de robótica o domótica hasta el control y medición de procesos. Estos elementos tienen gran cantidad de componentes compatibles que aumentan sus prestaciones de cara a la comunicación con otros dispositivos, sensores, motores u otros elementos. Mediante una pequeña inversión de dinero, podemos hacer uso de esta tecnología y además no se necesitan grandes conocimientos ya que actualmente la comunidad crea gran cantidad de tutoriales y explicaciones sobre sus distintas librerías o comandos de programación. También mencionar que todos los programas utilizados para la programación eran totalmente gratuitos, con lo que el usuario se puede olvidar de licencias que encarezcan el coste final del proyecto. Para la realización de un proyecto de estas características, no hay que gastar una gran cantidad de dinero y además, si se dispone de algún medio de impresión 3D propio, se reduce el coste total.

El diseño de la estructura ha resultado la etapa más compleja del trabajo ya que la estructura condiciona el ensamblaje de los distintos componentes y un pequeño cambio en esta repercute en todo el trabajo, así como en el movimiento del robot. Buscar una alternativa satisfactoria del movimiento ha sido todo un reto, el cual se ha completado con el diseño y programación de los distintos componentes.

El usar el sistema Lego, supone una ventaja importante por la facilidad a la hora de conseguir los componentes y la amplia gama de posibilidades, pero también supone un inconveniente cuando se trata de diseñar con nuevos elementos fuera del "sistema", como ha sido la placa *Arduino* así como los motores seleccionados, lo que nos ha obligado a buscar soluciones e implementarlas.

10. PROYECTO EDUCATIVO

Como se ha mencionado a lo largo del proyecto, este robot pretende ser un complemento más dentro de un programa educativo para las escuelas. La importancia de la robótica en los últimos años la ha hecho objetivo claro de gran variedad de talleres, cursos y otras actividades destinadas tanto a

la educación como al ocio en todo tipo de edades. No es extraño ver gran cantidad de agrupaciones apostando por esta ciencia.

De esta manera el proyecto puede consistir en un taller a realizar por los alumnos, donde formando grupos de 6 pueden montar el robot y hacer frente a los distintos problemas que se les planteen.

El coste del proyecto se ha intentado reducir lo máximo posible para que no signifique un gasto superior a 30€ por alumno, de manera que no suponga un coste demasiado elevado para las escuelas. Hay que tener en cuenta que una vez realizada la primera inversión, esta se puede amortizar a lo largo de años, simplemente incorporando piezas que se estropeen o aquellas que se extravíen. El proyecto se puede desarrollar como un taller dentro de la asignatura de tecnología, o puede implantarse como una actividad extraescolar del propio centro. El objetivo principal es que el alumno aprenda conceptos básicos de programación y del mundo de la tecnología. Estos conceptos pueden girar en torno a conocimientos eléctricos como la ley de ohm, mecánicos como la ley de engranajes o informáticos centrándose en la programación; reglas y funciones elementales que no tienen gran dificultad pero que sí son de mucha utilidad. El alumno aprende de una forma amena, tomando sus propias decisiones y aprendiendo de los errores, inmerso una actividad divertida que no se implementa con los métodos tradicionales de lápiz y papel.

También se pretende que el alumno comprenda las posibilidades que ofrece un objeto tan popular como es el móvil, que ha pasado de ser una herramienta de comunicación a un potente microordenador con todas las posibilidades que ello conlleva. Este, en lugar de ser una simple alternativa de ocio, como ocurre en el caso de la mayoría de jóvenes, puede ofrecernos otras funciones como es el caso del control de un robot. A través del proyecto se pretende introducir al alumno en un mundo en el cual, con sus propios esfuerzos, pueden conseguir grandes resultados, despertando su interés por el mundo de la ciencia y de la tecnología.

11. MEJORAS

Debido a que el proyecto consistía en el diseño de un robot caminante hexápodo, se pueden incluir mejoras destinadas tanto a la estética como al aumento de prestaciones. Como el robot está destinado a la educación, todos aquellos elementos perceptibles visual o acústicamente contribuirán a que aumente el interés de los alumnos hacia el mismo, mejorando la interacción de los alumnos con el robot. Estos elementos pueden ser pantallas LCD, matrices de leds, zumbadores, altavoces, etc., todos ellos compatibles con la placa *Arduino* montada. De la misma forma, otro elemento a considerar podría ser el sensor de ultrasonidos, el cual no presenta una dificultosa programación y puede resultar muy útil a la hora de detectar obstáculos.

En cuanto a estética, se podrían incluir carcasas que nos ofrecieran un recubrimiento de la parte electrónica. Mediante estos componentes, el alumno podría personalizar el propio robot haciendo uso de pinturas o pegatinas. Resaltar la importancia de que estos materiales deberían ser totalmente ligeros ya que añadir más peso repercute negativamente en el movimiento.

Otra mejora podría ser el uso de alguna batería que proporcionara la suficiente energía como para alimentar tanto a los motores como a la placa *Arduino*. Esta batería sería recargable con lo que eliminamos el problema del nivel de carga de las pilas y las continuas reposiciones de estas.

Por último, se podrían buscar alternativas a los materiales empleados, centrándose en elementos con menor peso y que ofrezcan las mismas prestaciones. También todas aquellas mejoras destinadas a la reducción de la fricción de los distintos componentes ayudarían a mejorar el movimiento de manera significativa, ya que esta es una de los principales problemas.

12. BIBLIOGRAFÍA

¿Qué es la robótica educativa? (n.d.). Consultado el 12 de mayo de 2016, página web de centros edukative: <http://www.edukative.es/que-es-la-robotica-educativa/>

Ayala, Ana (2014, Julio). La robotica educativa como metodología de aprendizaje . Consultado el 12 de mayo del 2016, página web de robótica 3.0: <http://www.educaciontrespuntocero.com/noticias/la-robotica-educativa-como-metodologia-de-aprendizaje/18904.html>

Cabrera Jiménez, Omar Lucio (1996, Diciembre). La Robótica Pedagógica. Soluciones Avanzadas No.40 [Versión electrónica]. Consultado el 15 de mayo de 2016, página web web.archive: https://web.archive.org/web/20070320212918/http://fodweb.net/robotica/roboteca/articulos/pdf/robotica_pedagogica.pdf

SaguaAlanguía, Yanet Maritza (n.d.). Clasificación de los Robots. Blogspot . Consultado el 15 de mayo de 2016. <http://roboticapuno.blogspot.com.es/2013/01/clasificacion-de-los-robots.html>

Historia de la Robótica. Blogspot(2007, Octubre). Consultado el 15 de mayo de 2016. <http://robotiica.blogspot.com.es/2007/10/historia-de-la-robotica.html>

Tren de engranajes (n.d.). Consultado el 5 de Junio de 2016. Página web cenice: http://concurso.cnice.mec.es/cnice2006/material107/mecanismos/mec_eng_tren.htm

Pulido, Antonio (2009, Marzo). El tornillo sinfín y la rueda dentada. Consultado el 5 de Junio de 2016. Página web IES VillalbaHervás: <https://aprendemostecnologia.org/2009/03/19/el-tornillo-sinfin-y-la-rueda-dentada/>

Rodriguez, Javier (2012, Marzo). Engranajes de dientes rectos. Blogspot. Consultado el 5 de Junio de 2016. <http://tecnoblogueando.blogspot.com.es/2012/03/engranajes-de-dientes-rectos.html>

¿Qué es la electrónica ? (n.d.). Consultado el 10 de junio de 2016. Página web mis respuestas.com: <http://www.misrespuestas.com/que-es-la-electronica.html>

Flores lopez, Victor (2012, Octubre). Sistema electrónico. Consultado el 10 de junio de 2016. Página web slideshare: <http://es.slideshare.net/Shatbell/sistema-electronico>

Introduction to Arduino (n.d.). Consultada el 5 de junio de 2016. Página web de Arduino: <https://www.arduino.cc/en/Guide/Introduction>

Hardware page Arduino Nano (n.d.). Consultada el 10 de junio de 2016. Página web de Arduino: <https://www.arduino.cc/en/Main/ArduinoBoardNano>

Pompa, Pablo (n.d.). Como utilizar los servos en nuestros proyectos. Consultada el 12 de junio de 2016. Página web de Superrobótica: <http://www.superrobotica.com/Servosrc.htm>

Ramirez, Yamid (n.d.). Servomotores. Consultada el 12 de junio de 2016. Página web de monografias.com: <http://www.monografias.com/trabajos60/servo-motores/servo-motores.shtml>

González, Victor (2003). Servomotores. Consultada el 13 de junio de 2016. Página web de Platea: http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/sistema/motores_servo.htm

Hirzel, Timothy (n.d.). PWMArduino. Consultada el 13 de junio de 2016. Página web de Arduino: <https://www.arduino.cc/en/Tutorial/PWM>

MÓDULO BLUETOOTH HC-06 (n.d.). Consultada el 13 de junio de 2016. Página web de Prometec: <http://www.prometec.net/bt-hc06/>

SoftwareSerial Library (n.d.). Consultada el 20 de junio de 2016. Página web de Arduino: <https://www.arduino.cc/en/Reference/SoftwareSerial>

Positive voltage regulators. Consultada el 17 de junio de 2016. Página web de Electrónicaembajadores: http://www.electronicaembajadores.com/datos/pdf1/sm/smci/l78xx_l78xxc.pdf

Conectar servomotores a una fuente externa (2012, Agosto). Consultada el 17 de junio de 2016. Foro oficial de Arduino: <http://forum.arduino.cc/index.php?topic=117379.15>

Servo Library (n.d.). Consultada el 22 de junio de 2016. Página oficial de Arduino: <https://www.arduino.cc/en/Reference/Servo>

Arduino + Processing: primeros pasos (n.d.). Consultada el 18 de junio de 2016. Página web DIYMakers: <http://diymakers.es/arduino-processing-primeros-pasos/>

Primeros pasos con App Inventor (n.d.). Consultada el 20 de junio de 2016. Página web sites.google: <https://sites.google.com/site/appinventormegusta/primeros-pasos>

The background is a vibrant blue gradient. A grid of light blue squares is visible, becoming more prominent towards the right. Overlaid on this are several flowing, wavy lines made of many thin, parallel lines, creating a sense of motion and depth. Two bright, starburst-like light effects are positioned on the lines. The word 'PRESUPUESTO' is written in a bold, dark blue, serif font across the middle of the image.

PRESUPUESTO

1. PRESUPUESTO

El coste del proyecto viene reflejado en el siguiente presupuesto. Se ha de tener en cuenta que este puede tener variaciones significativas ya que no es lo mismo la realización de un prototipo que la realización de diversos modelos. Algunos de estos proveedores aplican importantes descuentos si la cantidad que se compra es elevada y a la hora de realizar la estructura, no es lo mismo realizar un prototipo con fresadora que una gran cantidad de bases con métodos como el corte láser.

El presupuesto final responde a la suma de tres presupuestos que atienden a los siguientes criterios:

- ▶ Presupuesto de componentes materiales (PRESUPUESTO MATERIAL) Comprende tanto los componentes de Lego como electrónica, estructuras de bases y motores.
- ▶ Presupuesto de Coste de Personal (COSTE PERSONAL), donde se incluye coste del tiempo empleado tanto en el diseño como en la elaboración del proyecto.
- ▶ Presupuesto de Hardware y Software (COSTE DE ELEMENTOS DE SOFTWARE Y HARDWARE). Incluye los elementos de esta naturaleza empleados en el proyecto, esto principalmente supone el gasto en licencias y herramientas empleadas.

No se añadirán costes de materiales fungibles, como el papel de impresora, o herramientas de las cuales se disponía antes de la realización del proyecto. Tampoco se incluirán el precio de los envíos, ya que dependiendo del lugar donde se implemente el proyecto o de las cantidades, estos son variables.

1.1. PRESUPUESTO MATERIAL

Para entender este presupuesto de material hay que tener en cuenta que hasta llegar al modelo definitivo de robot se diseñó y construyó un modelo previo que finalmente fue desechado por los problemas de movilidad, pero que supuso un coste en el proyecto.

Hemos diferenciado el coste de las piezas empleadas para la totalidad del proyecto del coste de las piezas que realmente se han utilizado en el modelo final. Todos los componentes reflejan un coste sin IVA, incluyéndose este al total final con un valor de 21%.

Así pues, en la siguiente tabla podemos observar el nombre del componente, la familia a la que pertenece, el coste de cada unidad, el número de unidades, el coste total y el proveedor del material.

Tabla 8. Presupuesto material del TOTAL DEL PROYECTO

COMPONENTE	FAMILIA	COSTE UNIDAD	CANTIDAD	COSTE TOTAL	PROVEEDOR
Light Bluish Gray, Axle 3	Estructura	0,05 €	26	1,30 €	Electricbricks
Dark Bluish Gray, Gear 20 tooth	Estructura	0,40 €	6	2,40 €	Electricbricks
Dark Bluish Gray, Gear 16 tooth	Estructura	0,20 €	4	0,80 €	Electricbricks
Dark Bluish Gray, Gear 12 tooth	Estructura	0,15 €	2	0,30 €	Electricbricks
Light Bluish Gray, Gear 24 tooth	Estructura	0,25 €	22	5,50 €	Electricbricks
Yellow, Liftarm 1x3 thick	Estructura	0,19 €	6	1,14 €	Electricbricks
Yellow, Liftarm 1 x 5 Thin	Estructura	0,13 €	12	1,56 €	Electricbricks
Yellow, Liftarm Triangle Thin,	Estructura	0,20 €	12	2,40 €	Electricbricks

Diseño e Implementación de un Robot Caminante Hexápodo Basado en Lego y Arduino

Type II					
Yellow, Bush 1/2 Smooth	Estructura	0,07 €	84	5,88 €	Electricbricks
Dark Bluish Gray, Axle connector	Estructura	0,32 €	4	1,28 €	Electricbricks
Dark Bluish Gray, Liftarm 3 x 5 L-Shape with Quarter Ell	Estructura	0,19 €	12	2,28 €	Electricbricks
Black, Axle 2 Notched	Estructura	0,05 €	12	0,60 €	Electricbricks
black, Axle 4	Estructura	0,08 €	12	0,96 €	Electricbricks
Light Bluish Gray, Axle 5	Estructura	0,11 €	12	1,32 €	Electricbricks
Black, Axle 6	Estructura	0,11 €	6	0,66 €	Electricbricks
Light Bluish Gray, Axle 3	Estructura	0,08 €	38	3,04 €	Electricbricks
Black, Axle and Pin Connector	Estructura	0,08 €	22	1,76 €	Electricbricks
Light Bluish Gray, Angle Connector #2 - 180 degrees	Estructura	0,13 €	6	0,78 €	Electricbricks
Light Gray Technic, Connector with axle hole	Estructura	0,12 €	12	1,44 €	Electricbricks
Black, Pin with Friction Ridges Lengthwise and Towball	Estructura	0,05 €	12	0,60 €	Electricbricks
Light Bluish Gray, Axle 7	Estructura	0,06 €	10	0,60 €	Electricbricks
Light Bluish Gray, Liftarm 1 x 7 Thick	Estructura	0,12 €	6	0,72 €	Electricbricks
Blue, Axle Pin with Friction Ridges Lengthwise	Estructura	0,10 €	8	0,80 €	Electricbricks
Black, Liftarm 1 x 2 Thick with Pin Hole and Axle Hole	Estructura	0,16 €	6	0,96 €	Electricbricks
Light Bluish Gray, Bush	Estructura	0,11 €	18	1,98 €	Electricbricks
Blue, Pin Long with Friction Ridges Lengthwise	Estructura	0,11 €	6	0,66 €	Electricbricks
Light Bluish Gray, Axle 9	Estructura	0,20 €	6	1,20 €	Electricbricks
Black, Axle and Pin Connector Perpendicular 3L with Center Pin Hole	Estructura	0,12 €	18	2,16 €	Electricbricks
Black, Gear worm	Estructura	0,20 €	6	1,20 €	Electricbricks
Light Bluish Gray, Gear 8 tooth	Estructura	0,13 €	8	1,04 €	Electricbricks
Black, Pin with Friction Ridges Lengthwise WITH Center Slots	Estructura	0,02 €	42	0,84 €	Electricbricks
Yellow, Liftarm 1 x 15 Thick	Estructura	0,60 €	4	2,40 €	Electricbricks
Dark Bluish Gray, Liftarm 1 x 5 Thick	Estructura	0,12 €	12	1,44 €	Electricbricks
Light Bluish Gray, Liftarm 5 x 7 Open Center Frame Thick	Estructura	1,75 €	2	3,50 €	Electricbricks
Dark Bluish Gray, Link 1 x 6	Estructura	0,32 €	6	1,92 €	Electricbricks
Brocas Microtornillería	Fijación	4,32 €	1	4,32 €	Leroy Merlin
Microtornillería (1,6x12mm) tornillo y tuerca	Fijación	2,20 €	1	2,20 €	Leroy Merlin
Resina epoxy	Fijación	4,40 €	1	4,40 €	Ferretería
Placa metacrilato	Material	5,00 €	1	5,00 €	Leroy Merlin

Diseño e Implementación de un Robot Caminante Hexápodo Basado en Lego y Arduino

Tablón contrachapado 300x600mm	Material	2,50 €	1	2,50 €	Leroy Merlin
Base estructura metacrilato con fresado	Estructura	8 €	3	24,00 €	Particular
Shield Arduino Nano 3.0 Placa de Expansion	Electrónica	3,60 €	1	3,60 €	Electrrio
MODULO BLUETOOTH ARDUINO HC-06	Electrónica	8 €	1	8,00 €	Electrrio
Arduino NANO V3.0 ATMEGA328	Electrónica	9 €	1	9,00 €	Electrrio
Adaptador Jack 5.5 pila 9V conector	Electrónica	1 €	1	1,00 €	Electrrio
Condensador 330nF	Electrónica	0,08 €	1	0,08 €	Micropick
Condensador 100nF	Electrónica	0,08 €	1	0,08 €	Micropick
Conector macho alimentación	Electrónica	0,83 €	1	0,83 €	Micropick
Cable conexión sección 0,14 mm	Electrónica	1,17 €	1	1,17 €	Micropick
Regulador de tensión 7806	Electrónica	0,35 €	1	0,35 €	Micropick
Pila 9V	Electrónica	3 €	2	6,00 €	Micropick
Modulo board con 270 pines	Electrónica	4,50 €	1	4,50 €	Micropick
Cables macho hembra pack 10	Electrónica	1,10 €	1	1,10 €	Micropick
Led alta luminosidad amarillo	Electrónica	0,53 €	2	1,06 €	Micropick
Led alta luminosidad rojo	Electrónica	0,39 €	1	0,39 €	Micropick
Led alta luminosidad blanco	Electrónica	0,32 €	1	0,32 €	Micropick
Led alta luminosidad verde	Electrónica	0,64 €	1	0,64 €	Micropick
Resistencia 220Ω	Electrónica	0,02 €	5	0,11 €	Micropick
Resistencia 1000Ω	Electrónica	0,02 €	6	0,13 €	Micropick
Power HD AR-3606HB	Motores	7,50 €	2	15,00 €	Hobbyking
FS90R Micro Continuous Rotation Servo	Motores	7,50 €	2	15,00 €	Electan
TOTAL (Sin IVA)				168,20 €	
TOTAL (incluyendo IVA 21%)				203,52 €	

Fuente: Elaboración propia mediante Excel

El siguiente presupuesto muestra el coste de fabricación del modelo final del robot. Este únicamente incluye las piezas del modelo seleccionado, motores y electrónica. Al igual que el anterior, el precio de cada componente está especificado si IVA, incluyéndose este al final con un valor de 21%.

Tabla 9. Presupuesto material FINAL

COMPONENTE	FAMILIA	COSTE UNIDAD	CANTIDAD	COSTE TOTAL	PROVEEDOR
Light Bluish Gray, Gear worm	Estructura	0,20 €	6	1,20 €	Electricbricks
Dark Bluish Gray, Gear 24 tooth	Estructura	0,25 €	6	1,50 €	Electricbricks
Dark Bluish Gray, Gear 20 tooth	Estructura	0,40 €	6	2,40 €	Electricbricks

Diseño e Implementación de un Robot Caminante Hexápodo Basado en Lego y Arduino

Dark Bluish Gray, Gear 16 tooth	Estructura	0,20 €	4	0,80 €	Electricbricks
Dark Bluish Gray, Gear 12 tooth	Estructura	0,15 €	2	0,30 €	Electricbricks
Dark Bluish Gray, Gear 8 tooth	Estructura	0,13 €	8	1,04 €	Electricbricks
Dark Bluish Gray, Axle connector	Estructura	0,32 €	4	1,28 €	Electricbricks
Yellow, Bush 1/2 Smooth	Estructura	0,07 €	84	5,88 €	Electricbricks
Yellow, Liftarm 1x3 thick	Estructura	0,19 €	6	1,14 €	Electricbricks
Dark Bluish Gray, Liftarm 1x5 thick	Estructura	0,12 €	12	1,44 €	Electricbricks
Dark Bluish Gray, Liftarm 3 x 5 L-Shape with Quarter Ell	Estructura	0,19 €	12	2,28 €	Electricbricks
Black, Axle 2 notched	Estructura	0,10 €	6	0,60 €	Electricbricks
Light Gray, Axle 3	Estructura	0,08 €	38	3,04 €	Electricbricks
black, Axle 4	Estructura	0,08 €	12	0,96 €	Electricbricks
Light Bluish Gray, Axle 5	Estructura	0,11 €	12	1,32 €	Electricbricks
Black, Axle 6	Estructura	0,11 €	6	0,66 €	Electricbricks
Light Bluish Gray, Axle 7	Estructura	0,15 €	4	0,60 €	Electricbricks
Light Bluish Gray, Axle 9	Estructura	0,35 €	4	1,40 €	Electricbricks
Light Bluish Gray, Angle Connector #2 - 180 degrees	Estructura	0,13 €	6	0,78 €	Electricbricks
Black, Axle Joiner Perpendicular 3L with Center Hole	Estructura	0,12 €	18	2,16 €	Electricbricks
Light Bluish Gray, Axle Joiner Perpendicular	Estructura	0,08 €	22	1,76 €	Electricbricks
Light Bluish Gray, Bush	Estructura	0,11 €	18	1,98 €	Electricbricks
Light Gray Technic, Connector with axle hole	Estructura	0,12 €	12	1,44 €	Electricbricks
Black, Liftarm 1 x 2 Thick with Pin Hole and Axle Hole	Estructura	0,16 €	6	0,96 €	Electricbricks
Blue, Axle Pin WITH Friction Ridges Lengthwise On Shaft	Estructura	0,10 €	6	0,60 €	Electricbricks
Blue, Pin Long with Friction Ridges Lengthwise	Estructura	0,11 €	6	0,66 €	Electricbricks
Brocas Microtornillería	Fijación	4,32 €	1	4,32 €	Leroy Merlin
Microtornillería (1,6x12mm) tornillo y tuerca	Fijación	2,20 €	1	2,20 €	Leroy Merlin
Resina epoxy	Fijación	4,40 €	1	4,40 €	Ferretería
Placa metacrilato	Material	5,00 €	1	5,00 €	Leroy Merlin
Tablón contrachapado 300x600mm	Material	2,50 €	1	2,50 €	Leroy Merlin
Base estructura metacrilato con fresado	Estructura	8 €	3	24,00 €	Particular
Shield Arduino Nano 3.0	Electrónica	3,60 €	1	3,60 €	Electrrio

Diseño e Implementación de un Robot Caminante Hexápodo Basado en Lego y Arduino

Placa de Expansion					
MODULO BLUETOOTH ARDUINO HC-06	Electrónica	8 €	1	8,00 €	Electrrio
Arduino NANO V3.0 ATMEGA328	Electrónica	9 €	1	9,00 €	Electrrio
Adaptador Jack 5.5 pila 9V conector	Electrónica	1 €	1	1,00 €	Electrrio
Pila 9V	Electrónica	3 €	2	6,00 €	Electrrio
Condensador 330nF	Electrónica	0,08 €	1	0,08 €	Micropick
Condensador 100nF	Electrónica	0,08 €	1	0,08 €	Micropick
Conector macho alimentación	Electrónica	0,83 €	1	0,83 €	Micropick
Cable conexión sección 0,14 mm	Electrónica	1,17 €	1	1,17 €	Micropick
Regulador de tensión 7806	Electrónica	0,35 €	1	0,35 €	Micropick
Módulo board con 270 pines	Electrónica	4,55 €	1	4,55 €	Micropick
Cables macho hembra pack 10	Electrónica	1,10 €	1	1,10 €	Micropick
Led alta luminosidad amarillo	Electrónica	0,53 €	2	1,06 €	Micropick
Led alta luminosidad rojo	Electrónica	0,39 €	1	0,39 €	Micropick
Led alta luminosidad blanco	Electrónica	0,32 €	1	0,32 €	Micropick
Led alta luminosidad verde	Electrónica	0,64 €	1	0,64 €	Micropick
Resistencia 220Ω	Electrónica	0,02 €	5	0,11 €	Micropick
Power HD AR-3606HB	Motores	7,50 €	2	15,00 €	Hobbyking
Resistencia 1000Ω	Electrónica	0,02 €	3	0,06 €	Micropick
TOTAL (Sin IVA)				133,94 €	
TOTAL (incluyendo IVA 21%)				162,07 €	

Fuente: Elaboración propia utilizando Excel

1.2. COSTE PERSONAL

Para realizar el coste personal, se va a detallar quienes han sido las personas que han participado en el proyecto y las horas que le han dedicado. El coste por hora de cada integrante se ha definido de acuerdo a una aproximación, basada en lo que se está cobrando en la actualidad. En la tabla se muestra el nombre del trabajador, su categoría, el coste por hora de su trabajo, la cantidad de horas dedicadas al proyecto y finalmente el coste total.

En la tabla aparece Leopoldo Armesto Ángel, tutor del proyecto, al cual se le han realizado algunas preguntas en el caso de surgieran dudas.

Tabla 10. Coste personal

Nombre	Categoría	Euros/hora	Horas totales	Coste total
Francisco Esteve Villar del Saz	Ingeniero Junior	12 €	330	3.960 €
Leopoldo Armensto Ángel	Ingeniero Sénior	30 €	12	360 €
Total				4.320 €

Fuente: Elaboración propia mediante Excel.

1.3.COSTE ELEMENTOS SOFTWARE Y HARDWARE

El coste del material reflejado en este apartado, hace referencia a los diferentes equipos informáticos que se han utilizado para la realización del trabajo, las licencias de los programas informáticos utilizados y las herramientas de fabricación. En la tabla se mostrará el producto, su precio, el tiempo de amortización en meses, los meses de uso y el coste que ha significado para el proyecto. Más adelante, se hará una descripción de estos materiales .

Tabla 11. Coste elementos de software y hardware

Producto	Precio	Periodo de amortización	Uso del producto	Coste para el proyecto
Licencia Inventor	2.060 €	12	3	515,00 €
Licencia AUTOCAD	2.000 €	12	3	500,00 €
Ordenador Windows	800 €	48	4	66,67 €
Herramientas de trabajo	140 €	48	3	8,75 €
Huawei G7-L01	200 €	20	4	40,00 €
Total				1.130,42 €

Fuente: Elaboración propia mediante Excel.

1.3.1. ORDENADOR WINDOWS

El ordenador Windows utilizado para el desarrollo del proyecto presenta las siguientes características:

- ▶ Procesador AMD FX quad core – 3.8GHz
- ▶ 8 GB de memoria RAM DDR3

- ▶ Disco duro de 500 GB
- ▶ Pantalla LG 21"
- ▶ Windows 7 64-bits
- ▶ Tarjeta gráfica Nvidia GT 240 1GB DDR5

1.3.2. HERRAMIENTAS DE TRABAJO

Para la realización del proyecto se han utilizado un serie de herramientas de trabajo, habituales en cualquier domicilio. Estas son:

- ▶ Taladro eléctrico 500W
- ▶ Brocas de diversas medidas
- ▶ Sierra caladora eléctrica
- ▶ Juego de destornilladores
- ▶ Banco de trabajo
- ▶ Alicates
- ▶ Elementos de medición

1.3.3. Huawei G7-L01

Para comprobar y utilizar la aplicación móvil se ha utilizado este modelo de teléfono . Este modelo es tipo *Smartphone* de calidad media. Las características son las siguientes:

- ▶ Pantalla de 5,5"
- ▶ Sistema operativo *Android* 4.4 (KitKat)
- ▶ Procesador Quad Core a 1.2 GHz
- ▶ 2 GB de memoria RAM
- ▶ 16 GB de almacenamiento
- ▶ Bluetooth 4.0
- ▶ Batería Li-ion de 3000 mAh

1.4. COSTE TOTAL DEL PROYECTO

Por último, se va a mostrar el coste total que supondría el proyecto sumando el coste de material final, el coste de los elementos de software y hardware y el coste de personal.

Tabla 12. Coste total del proyecto

Descripción	Coste total
Coste material final (IVA incluido)	162,07 €
Coste de personal	4.320 €
Coste software y hardware	1.130,42 €
COSTE TOTAL PROYECTO	5.612,49 €

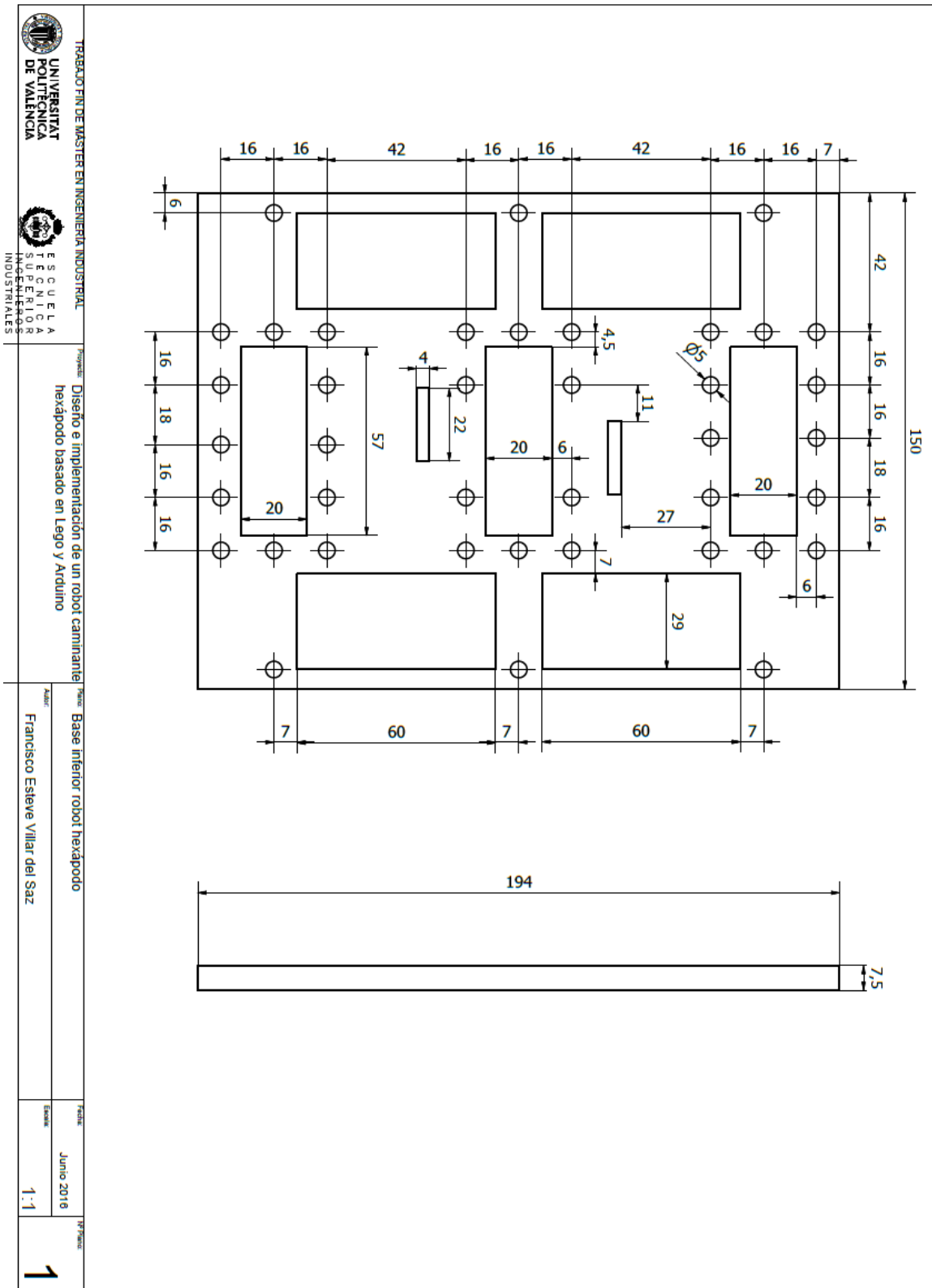
Fuente: Elaboración propia mediante Excel

Por lo tanto, si el proyecto se hubiera realizado en una empresa, habría supuesto un coste de **5.612,49€**.

PLANOS

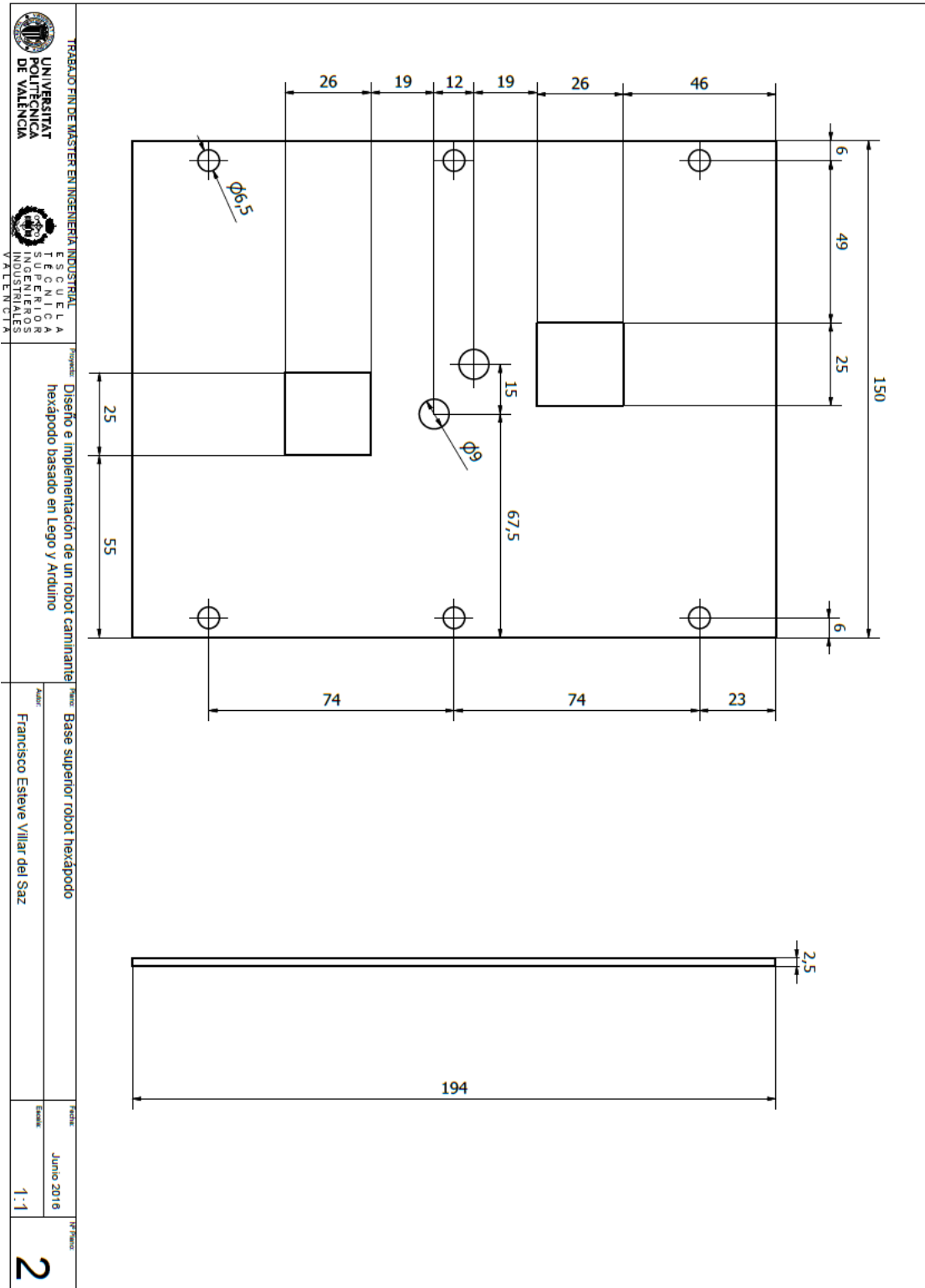


1.1. PLANO BASE INFERIOR ROBOT HEXÁPODO



Fuente: Elaboración propia mediante AutoCad

1.2. PLANO BASE SUPERIOR ROBOT HEXÁPODO



Fuente: Elaboración propia mediante AutoCad

ANEJO 1. PROGRAMA

```
#include <Servo.h>

#include <SoftwareSerial.h>

SoftwareSerialmyserial(0,1);

Servo servo4;

Servo servo3;

intselec;

intselecant;

void setup()

{

    servo3.attach(8);

    servo4.attach(11);

    myserial.begin(9600);

    pinMode(3,OUTPUT);

    pinMode(4,OUTPUT);

    pinMode(5,OUTPUT);

    pinMode(6,OUTPUT);

    pinMode(7,OUTPUT);

}

void loop()

{

    if(myserial.available(>0){

selec=myserial.read();

    }

    if (selecant!=selec){

digitalWrite(3,LOW);
```

```
digitalWrite(4,LOW);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
if (selec=='b'){
  servo3.write(180);
  servo4.write(180);
digitalWrite(3,HIGH);
}
if (selec=='a'){
  servo3.write(180);
  servo4.write(120);
digitalWrite(6,HIGH);
}
if (selec=='c'){
  servo3.write(120);
  servo4.write(180);
digitalWrite(5,HIGH);
}
if (selec=='d'){
  servo3.write(140);
  servo4.write(40);
digitalWrite(6,HIGH);
}
if (selec=='e'){
  servo3.write(90);
```

```
servo4.write(90);  
digitalWrite(4,HIGH);  
}  
if (selec=='f'){  
servo3.write(40);  
servo4.write(140);  
digitalWrite(5,HIGH);  
}  
if (selec=='g'){  
servo3.write(0);  
servo4.write(60);  
digitalWrite(6,HIGH);  
}  
if (selec=='h'){  
servo3.write(0);  
servo4.write(0);  
digitalWrite(7,HIGH);  
}  
if (selec=='i'){  
servo3.write(60);  
servo4.write(0);  
digitalWrite(5,HIGH);  
}  
selecant=selec;  
}  
}
```