



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

**DISEÑO, PROGRAMACIÓN Y CONSTRUCCIÓN DE UN ROBOT
HEXÁPODO BASADO EN ARDUINO Y CONTROLADO A TRAVÉS DEL
MÓDULO BLUETOOTH MEDIANTE UNA APLICACIÓN DE TELÉFONO
MÓVIL**

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

Álvaro Alberto Giner

TUTORIZADO POR

Leopoldo Armesto Ángel

CURSO ACADÉMICO: 2019/2020

Hoja en blanco dejada a propósito

Agradecimientos

A mi familia por todo su apoyo y ayuda.

A mis amigos por estar ahí aconsejando y dando siempre buenas ideas.

Hoja en blanco dejada a propósito

RESUMEN

En este proyecto se pretende realizar el diseño e implementación de un robot caminante hexápodo con 18 grados de libertad, el cual se podrá utilizar con fines didácticos o como prototipo de transporte a pequeña escala.

Para su diseño y construcción se utilizará software de diseño 3D, así como una impresora 3D. Además, su diseño será modular, facilitando el rápido reemplazo de piezas dañadas o para añadir futuras mejoras. Cabe añadir que el entorno de programación de dicho robot será el de Arduino, dándole la sencillez y accesibilidad que éste ofrece, junto con AppInventor para el desarrollo gratuito de la aplicación móvil con la cual podremos controlarlo.

El diseño de la electrónica estará diseñado en función de las necesidades de este robot.

PALABRAS CLAVE

Arduino, Impresión 3D, Robot Hexápodo, Bluetooth, Aplicación Móvil, Programación, Electrónica, Robótica Móvil, Batería Li-Po, Servomotor, AppInventor y Educación.

Hoja en blanco dejada a propósito

RESUM

En aquest projecte es pretén realitzar el disseny i implementació d'un robot caminant hexàpode amb 18 graus de llibertat, el qual es podrà utilitzar amb fins didàctics o com a prototip de transport a petita escala.

Per al seu disseny i construcció es farà servir programari de disseny 3D, així com una impressora 3D. A més, el seu disseny serà modular, facilitant el ràpid reemplaçament de peces danyades o per afegir futures millores. Cal afegir que l'entorn de programació d'aquest robot serà el d'Arduino, donant-li la senzillesa i accessibilitat que aquest ofereix, juntament amb AppInventor per al desenvolupament gratuït de l'aplicació mòbil amb la qual podrem controlar-lo.

El disseny de l'electrònica estarà dissenyat en funció de les necessitats d'aquest robot.

PARAULES CLAU

Arduino, Impressió 3D, Robot Hexàpode, Bluetooth, Aplicació Mòbil, Programació, Electrònica, Robòtica Mòbil, Bateria Li-Po, Servomotor, AppInventor i Educació.

Hoja en blanco dejada a propósito

ABSTRACT

This project aims to carry out the design and implementation of a hexapod walking robot with 18 degrees of freedom, which can be used for educational purposes or as a prototype of small-scale transport.

For its design and construction, 3D design software will be used, as well as a 3D printer. In addition, its design will be modular, facilitating the quick replacement of damaged parts or to add future improvements. It should be added that the programming environment of said robot will be that of Arduino, giving it the simplicity and accessibility that it offers, together with AppInventor for the free development of the mobile application with which we can control it.

The electronics design will be designed based on the needs of this robot.

KEYWORDS

Arduino, 3D Printing, Hexapod Robot, Bluetooth, Mobile Application, Programming, Electronics, Mobile Robotics, Li-Po Battery, Servomotor, AppInventor and Education.

Hoja en blanco dejada a propósito

ÍNDICE

MEMORIA	1
1.- OBJETO.....	3
2.- FACTORES A CONSIDERAR: NECESIDADES, LIMITACIONES Y CONDICIONANTES	3
2.1.- MOTIVACIONES PARA LA REALIZACIÓN DEL PROYECTO	3
2.2.- ESPECIFICACIONES DEL ENCARGO	4
2.3.- NORMATIVA.....	4
2.4.- ANÁLISIS DE HOJAS DE ESPECIFICACIONES.....	4
3.- IMPRESIÓN DE PIEZAS.....	10
4.- DISEÑO Y CORRECCIONES	12
4.1.- DESARROLLO DEL CUERPO.....	12
4.2.- ELECCIÓN Y DESCARTE DE LA DE ALIMENTACIÓN	16
5.- DESCRIPCIÓN DETALLADA DE LA SOLUCIÓN ADOPTADA	17
6.- JUSTIFICACIÓN DETALLADA DE LA SELECCIÓN Y DIMENSIONADO DE LOS COMPONENTES ELECTRÓNICOS.....	20
6.1.- ELECCIÓN DEL CONTROLADOR ARDUINO NANO.....	20
6.2.- ELECCIÓN DEL MÓDULO BLUETOOTH	20
6.3.- ELECCIÓN DE LOS SERVOMOTORES MG996R Y SG90.....	20
6.4.- ELECCIÓN DE LOS CONTROLADORES PCA9685.....	22
6.5.- ELECCIÓN DEL MÓDULO DE ULTRASONIDOS	23
6.6.- ELECCIÓN DE LA MATRIZ DE LEDS 8X8.....	23
6.7.- ELECCIÓN DE LOS DIODOS	24
6.8.- ELECCIÓN DE LA BATERÍA.....	24
7.- PROGRAMACIÓN.....	25
7.1.- PROGRAMACIÓN DEL ROBOT	25
7.2.- PROGRAMACIÓN DE LA APLICACIÓN MÓVIL Y USO DEL MÓDULO BLUETOOTH	34
8.- BIBLIOGRAFÍA	36
ANEXO A-1: MANUAL DEL USUARIO y MANTENIMIENTO	38
1.- DESCARGAR LA APLICACIÓN	40
2.- VINCULACIÓN DE DISPOSITIVOS.....	40
3.- CONTROLES Y FUNCIONALIDADES.....	42
4.- CARGA DE LA BATERÍA	45
5.- MANTENIMIENTO.....	45
ANEXO A-2: PROGRAMACIÓN DE LA APLICACIÓN MÓVIL	47
ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO	55
PLANOS.....	95

PLANO 2.....	97
PLANO 2.1	98
PLANO 2.1.1.1	99
PLANO 2.1.1.2	101
PLANO 2.1.2.1	103
PLANO 2.1.2.2	105
PLANO 2.1.3	107
PLANO 2.1.4	108
PLANO 2.1.5	109
PLANO 2.2	111
PLANO 2.2.1.1	112
PLANO 2.2.1.2	113
PLANO 2.2.1.3	114
PLANO 2.2.2	115
PLANO 2.2.3.1	116
PLANO 2.2.3.2	117
PLANO 2.2.4	118
PLANO 2.2.5.1	119
PLANO 2.2.5.2	120
PLANO 2.3	121
PLANO 2.3.1	122
PLANO 2.3.2	123
PLANO 2.3.3	125
PLANO 2.3.4	127
PLANO 2.4	128
PLIEGO DE CONDICIONES.....	129
1.- OBJETO.....	131
2.- CONDICIONES DE LOS MATERIALES.....	131
2.1.- CHASIS	131
2.2.- UNIDAD DE CONTROL	132
2.3.- SISTEMA DE MOVIMIENTO	132
3.- EJECUCIÓN Y MONTAJE.....	133
3.1.- SUBMONTAJE DEL CHASIS DE LA PARTE SUPERIOR.....	133
3.2.- SUBMONTAJE DEL CHASIS DE LA PARTE INFERIOR.....	133
3.3.- ENSAMBLAJE DEL CHASIS.....	134
3.4.- SUBMONTAJE DE LA BOCA.....	136

3.5.- SUBMONTAJE DE LAS PATAS.....	136
3.6.- ENSAMBLAJE FINAL.....	137
3.7.- MONTAJE DE LA ELECTRÓNICA.....	137
4.- CONTROL DE CALIDAD	138
5.- ENSAYOS DE RECEPCIÓN.....	138
6.- ENTREGA.....	139
PRESUPUESTO	141
TOTAL PARTES ESTRUCTURALES PLÁSTICAS	143
COSTE CHASIS.....	143
COSTE BOCA	144
COSTE PATAS.....	145
TOTAL PARTE ELECTRÓNICA.....	146
TOTAL PARTE DE FUERZA	147
TOTAL PARTE ELÉCTRICA.....	148
TOTAL OTROS	149
TOTAL MANO DE OBRA.....	150
TOTAL MEDIOS AUXILIARES	151
TOTAL DEL PRESUPUESTO.....	152

Hoja en blanco dejada a propósito

ÍNDICE DE FIGURAS

FIGURA 1: PINES DEL ARDUINO NANO UTILIZADOS. [1] (MODIFICACIÓN PROPIA PARA ESTE PROYECTO)	5
FIGURA 2: PLACA EXPANSORA DE ENTRADAS Y SALIDAS COMPATIBLE CON EL ARDUINO NANO. PINES ANALÓGICOS EN COLOR AZUL. PINES DIGITALES EN COLOR AMARILLO. CONEXIÓN DE 12 V EN VERDE.[2] (MODIFICACIÓN PROPIA PARA ESTE PROYECTO).....	6
FIGURA 3: IMPRESORA 3D MODELO CREALITY ENDER 3. [12]	10
FIGURA 4: COMPARACIÓN ENTRE LOS TIPOS DE EXTRUSIÓN QUE EXISTEN EN EL MERCADO. [13]	11
FIGURA 5: EJEMPLO DE EFECTO "WARPING". LA BASE SE HA DESPEGADO DE LA CAMA CALIENTE, DEFORMANDO ASÍ LA PIEZA. [14]	11
FIGURA 6: EJEMPLO DE EFECTO "CRACKING". SE HA PRODUCIDO UN CAMBIO DE TEMPERATURA BRUSCO ENTRE CAPAS PROVOCANDO UNA SEPARACIÓN DE ÉSTAS. [15]	11
FIGURA 7: VERSIÓN 1 DEL ROBOT HEXÁPODO	12
FIGURA 8: VERSIÓN 2 DEL ROBOT HEXÁPODO	13
FIGURA 9: VERSIÓN 3 DEL ROBOT HEXÁPODO	13
FIGURA 10: VERSIÓN 4 DEL ROBOT HEXÁPODO	14
FIGURA 11: VERSIÓN 5 DEL ROBOT HEXÁPODO	15
FIGURA 12: VERSIÓN 6 Y FINAL DEL ROBOT HEXÁPODO	16
FIGURA 13: DISEÑO DE LA SOLUCIÓN ADOPTADA	17
FIGURA 14: POSICIÓN ESTÁTICA DEL ROBOT DONDE LA FORMA DE LAS PATAS CON EL CUERPO FORMA UN ARCO.	18
FIGURA 15: ORGANIGRAMA DEL PROYECTO.	19
FIGURA 16: PUNTO CRÍTICO A PARTIR DEL CUAL SE CALCULA EL PAR MOTOR NECESARIO.	22
FIGURA 17: EJEMPLO DE MODULACIÓN DE ANCHO DE PULSO (PWM).....	26
FIGURA 18: CONTROLADOR PCA9685 UTILIZADO EN EL ROBOT HEXÁPODO. [16]	26
FIGURA 19: NUMERACIÓN DE LOS SERVOMOTORES DEL ROBOT HEXÁPODO.	28
FIGURA 20: CICLO REALIZADO POR LAS PATAS DEL ROBOT HEXÁPODO PARA MOVERSE HACIA ADELANTE.....	28
FIGURA 21: CICLO REALIZADO POR LAS PATAS DEL ROBOT HEXÁPODO PARA GIRAR HACIA LA DERECHA.	29
FIGURA 22: CICLO REALIZADO POR LAS PATAS DEL ROBOT HEXÁPODO PARA GIRAR A LA IZQUIERDA.	29
FIGURA 23: CICLO REALIZADO POR LAS PATAS DEL ROBOT HEXÁPODO PARA DESPLAZARSE HACIA ATRÁS.	30
FIGURA 24: NOMBRE Y SITUACIÓN DE LOS SERVOMOTORES DE LAS PATAS DEL ROBOT HEXÁPODO.	31
FIGURA 25: DIFERENTES ALTURAS DEL CUERPO DEL ROBOT HEXÁPODO. A LA IZQUIERDA POSICIÓN BAJA Y A LA DERECHA POSICIÓN ALTA.	31
FIGURA 26: COMPARACIÓN Y POSICIONAMIENTO DE LAS PATAS PLEGADAS DEL ROBOT HEXÁPODO. A LA IZQUIERDA PATAS ABIERTAS Y A LA DERECHA LAS PATAS PLEGADAS.	31
FIGURA 27: ROTACIÓN DEL ROBOT HEXÁPODO HACIA LA DERECHA.	32

FIGURA 28: ROTACIÓN DEL ROBOT HEXÁPODO HACIA LA IZQUIERDA.....	32
FIGURA 29: MÁQUINA DE ESTADOS UTILIZADA PARA LA PROGRAMACIÓN DEL SENSOR DE PROXIMIDAD Y LA MATRIZ LED.	33
FIGURA 30: DIFERENTES CONFIGURACIONES DE LA MATRIZ LED. RODEADO EN COLOR VERDE LA CONFIGURACIÓN UTILIZADA PARA EL MODO PATRULLA Y EN COLOR AZUL LA CONFIGURACIÓN UTILIZADA PARA DAR LA BIENVENIDA. [17].....	34
FIGURA 31: ICONO DE LA APLICACIÓN.	40
FIGURA 32: PANTALLA PARA LA VINCULACIÓN DE LA CONEXIÓN BLUETOOTH.	41
FIGURA 33: PANTALLA PARA LA ELECCIÓN DEL DISPOSITIVO A VINCULAR.	41
FIGURA 34: PANTALLA PRINCIPAL DE LA APLICACIÓN.....	42
FIGURA 35: VISTA EXPLOSIONADA DE LA PARTE DEL CHASIS SUPERIOR. A LA DERECHA CHASIS SUPERIOR TRASERO Y A LA IZQUIERDA EL DELANTERO.....	133
FIGURA 36: VISTA EXPLOSIONADA DE LA PARTE DEL CHASIS SUPERIOR. A LA DERECHA CHASIS SUPERIOR TRASERO, A LA IZQUIERDA EL DELANTERO.....	134
FIGURA 37: VISTA EXPLOSIONADA DEL CHASIS. DESCRIBIENDO LA FIGURA DE ARRIBA A ABAJO Y EN ORDEN, ENCONTRAMOS EL SOPORTE DE ULTRASONIDOS, CHASIS SUPERIOR, PILARES, CHASIS INFERIOR Y SOPORTE DE LA BATERÍA.	135
FIGURA 38: VISTA EXPLOSIONADA DE LA BOCA. DESCRIBIENDO LA FIGURA DE ARRIBA ABAJO Y EN ORDEN, ENCONTRAMOS EN ROJO LOS ENGRANAJES Y EN NEGRO LOS ESTABILIZADORES, EN EL SIGUIENTE NIVEL, EN NEGRO EL SOPORTE Y EN ROJO LAS MANDÍBULAS Y FINALMENTE EN EL ÚLTIMO NIVEL LOS BULONES.	136
FIGURA 39: VISTA EXPLOSIONADA DE UNA PATA. DESCRIBIÉNDOLA DE IZQUIERDA A DERECHA Y EN ORDEN, ENCONTRAMOS LA MUÑECA Y SU SERVOMOTOR, CODO Y SU SERVOMOTOR, UNIÓN HOMBRO-CODO Y FINALMENTE HOMBRO Y SU SERVOMOTOR.....	137
FIGURA 40: VISTA FINAL DEL ENSAMBLAJE DEL ROBOT HEXÁPODO.....	137

ÍNDICE DE TABLAS

TABLA 1: CARACTERÍSTICAS GENERALES DEL ARDUINO NANO. [1]	5
TABLA 2: CARACTERÍSTICAS GENERALES DE LA PLACA EXPANSORA DE ENTRADAS Y SALIDAS COMPATIBLE CON ARDUINO NANO. [2]	6
TABLA 3: CARACTERÍSTICAS GENERALES DEL MÓDULO BLUETOOTH HC-06. [3].....	7
TABLA 4: CARACTERÍSTICAS GENERALES DE LA MATRIZ LED 8X8 CON UN CONTROLADOR MAX-7219. [4]	7
TABLA 5: CARACTERÍSTICAS GENERALES DEL CONTROLADOR PCA9685. [5]	7
TABLA 6: CARACTERÍSTICAS GENERALES DEL SERVOMOTOR SG90. [6].....	8
TABLA 7: CARACTERÍSTICAS GENERALES DEL SERVOMOTOR MG996R. [7]	8
TABLA 8: CARACTERÍSTICAS GENERALES DEL MÓDULO DE ULTRASONIDOS HC-SR04. [8]	9
TABLA 9: CARACTERÍSTICAS GENERALES DE LA BATERÍA TIPO LI-PO 2S. [9]	9
TABLA 10: CARACTERÍSTICAS GENERALES DEL DIODO MUR420. [10]	9
TABLA 11: CARACTERÍSTICAS GENERALES DEL PLÁSTICO ABS IMPRESO EN 3D. [11].....	10
TABLA 12: MASAS DE LOS ELEMENTOS DEL ROBOT HEXÁPODO.	21
TABLA 13: DISTANCIA ENTRE EL CENTRO DE MASAS DE LOS ELEMENTOS Y EL PUNTO CRÍTICO.	22
TABLA 14: RELACIÓN ENTRE LA TENSIÓN EN LA BATERÍA Y LA TENSIÓN EN LOS CONTROLADORES PCA9685.	24
TABLA 15: CONSUMOS MÁXIMOS DE LOS COMPONENTES DEL ROBOT HEXÁPODO Y SU CONSUMO TOTAL.....	25
TABLA 16: GRADOS A LOS CUALES SE DEBEN SITUAR LOS SERVOMOTORES DE LOS HOMBROS DEL ROBOT HEXÁPODO PARA PLEGAR LAS PATAS.....	32
TABLA 17: RELACIÓN ENTRE LAS ACCIONES Y LOS CARACTERES ENVIADOS DESDE LA APLICACIÓN MÓVIL AL MÓDULO BLUETOOTH.....	35
TABLA 18: ICONOS, NOMBRE DE LOS BOTONES, COMANDOS DE VOZ Y SU FUNCIONALIDAD.	44
TABLA 19: ENSAYOS DE RECEPCIÓN.	139

Hoja en blanco dejada a propósito



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

DOCUMENTO NÚMERO 1: MEMORIA

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

Álvaro Alberto Giner

TUTORIZADO POR

Leopoldo Armesto Ángel

CURSO ACADÉMICO: 2019/2020

Hoja en blanco dejada a propósito

1.- OBJETO

El departamento de Ingeniería de Sistemas y Automática de la Universidad Politécnica de Valencia va a promover el diseño, programación y la construcción de un prototipo de robot hexápodo destinado principalmente al uso educativo, así como para el transporte a pequeña escala.

El robot hexápodo pretende simular e imitar el comportamiento de un insecto hexápodo. Consta de seis patas con 3 grados de libertad cada una los cuales le permiten una gran movilidad y variedad de movimientos. Cabe añadir que este robot estará dotado de unas pinzas en la parte delantera las cuales le permitirán interaccionar con objetos de su entorno, teniendo la suficiente fuerza para transportarlos al igual que poder transportar objetos en la parte superior.

Se controlará desde una aplicación móvil propia que se podrá descargar en el smartphone del usuario (únicamente para dispositivos Android cuyas versiones de éste sean superiores a la 4.0) con la posibilidad de hacer que el robot se mueva libremente sin colisionar con objetos de su entorno. Éste podrá ser controlado mediante la aplicación hasta un alcance máximo de 15 metros gracias a la conectividad Bluetooth. Además, contará con una batería recargable con una autonomía mínima de 7 minutos.

El promotor del proyecto, el departamento de Ingeniería de Sistemas y Automática de la Universidad Politécnica de Valencia, con C.I.F. A20917125 y con dirección Camí de Vera, CP 46022, Valencia, se trata de un departamento especializado en automatización, control y optimización de procesos, robótica y visión artificial entre otros. Con este proyecto pretende crear un prototipo de robot hexápodo cuya finalidad es educativa, permitiendo que los alumnos lo puedan reprogramar y a la vez poder aplicar sus conocimientos.

2.- FACTORES A CONSIDERAR: NECESIDADES, LIMITACIONES Y CONDICIONANTES

2.1.- MOTIVACIONES PARA LA REALIZACIÓN DEL PROYECTO

La motivación que ha fomentado que se desarrolle esta idea es que tras cursar el grado en Ingeniería Electrónica Industrial y Automática en la Universidad Politécnica de Valencia siento que para finalizar y reforzar lo aprendido en programación, diseño, electrónica y otras aptitudes no tan solo son necesarios aquellos espacios virtuales dónde se pueden simular los trabajos de los alumnos, sino que es necesario tener una herramienta física, en este caso un robot hexápodo, dónde el alumno pueda ver con sus propios ojos y manipular aquello que ha programado sin necesidad de una pantalla, enfrentándose así a posibles problemas, que posiblemente, simulando y por falta de tiempo en clase no se les haría frente, como pueden ser fricciones entre piezas, colisiones, efectos de la gravedad, inercias, deformaciones de piezas, desgastes etc.

Cabe destacar que con este proyecto se pretende que el alumno esté mucho más motivado a la hora de realizar sus proyectos aprendiendo de una forma mucho más divertida y amena, además de poder aprender y profundizar en conceptos del propio grado u otras áreas ajenas a éste como sería la impresión 3D.

2.2.- ESPECIFICACIONES DEL ENCARGO

El departamento de Ingeniería de Sistemas y Automática ha encargado el diseño, programación y construcción de un robot hexápodo.

En las reuniones con dicho departamento se ha manifestado que el robot debe poderse imprimir con una impresora 3D cuyo volumen de impresión no sea superior a 220x220x250 mm, que las piezas y componentes sean económicos y fáciles de remplazar en caso de rotura o defecto y finalmente debe contar con una batería cuya duración sea suficiente para que el alumno pueda probar aquello que ha programado y que ésta se pueda recargar a lo largo de una clase de laboratorio, alrededor de una hora. Además, debe poderse controlar a través de conexión Bluetooth mediante una aplicación móvil.

El plazo de entrega del proyecto no debe exceder el mes de julio del año 2020, si no se cumple tendrá una penalización del 8% del total del coste del proyecto a favor del promotor.

El coste del proyecto debe ser inferior a los 13500 €.

2.3.- NORMATIVA

Se redacta el presente proyecto en cumplimiento de la siguiente normativa:

- **Real Decreto 186/2016**, de 6 de mayo, por el que se regula la compatibilidad electromagnética de los equipos eléctricos y electrónicos. (BOE nº 113, de 10/05/2016).
De acuerdo con dicho Real Decreto, el dispositivo debe tener la capacidad de funcionar de manera satisfactoria en su entorno electromagnético sin introducir perturbaciones electromagnéticas intolerables para otros equipos en ese entorno.
- **UNE-EN 61558-1:2007**. Seguridad de los transformadores de potencia, fuentes de alimentación, bobinas de inductancia y productos análogos. Parte 1: Requisitos generales y ensayos. (IEC 61558-1:2005)
Esta norma reconoce el nivel de protección aceptado internacionalmente contra riesgos eléctricos, mecánicos y de incendio a causa los transformadores cuando funcionan en uso normal y según las instrucciones del fabricante.
- **REGLAMENTO (CE) nº1907/2006 DEL PARLAMENTO EUROPEO Y DEL CONSEJO** de 18 de diciembre de 2006 relativo al registro, la evaluación, la autorización y restricción de las sustancias y preparados químicos (REACH).
Según dicho reglamento, en la fabricación del producto se deben utilizar materiales no infecciosos ni alérgenos como por ejemplo el plástico ABS.

2.4.- ANÁLISIS DE HOJAS DE ESPECIFICACIONES

2.4.1.- ARDUINO NANO

La siguiente tabla de especificaciones (Tabla 1) corresponden al microprocesador Arduino Nano, cuyo microcontrolador es el ATmega328, que se ha utilizado para este proyecto.

Como podemos observar en la Tabla 1 la tensión máxima de alimentación es de 12 V a través del pin Vin, la velocidad del reloj es de 16 MHz, la masa del componente es de 7

g, el consumo propio del microprocesador es de 19 mA, además de un consumo máximo por cada pin de 40 mA.

Especificaciones Técnicas del Arduino Nano	
Microcontrolador	ATmega328
Tensión de entrada	7 V a 12 V
Tensión Operacional (Nivel lógico)	5 V
Pines Digitales	14 (6 de los cuales con salida PWM)
Pines Analógicos	8
Consumo de corriente máximo por Pin	40 mA (Valor máximo)
Velocidad del Reloj	16 MHz
Dimensiones	45x18 mm
Masa	7 g
Consumo del Microcontrolador a 8 MHz y Vcc a 5 V	19 mA (Valor máximo)

Tabla 1: Características generales del Arduino Nano. [1]

Para la realización de este proyecto se han utilizado 9 de los 14 pines digitales y 3 de los 8 pines analógicos, concretamente los marcados con una flecha roja en la Figura 1.

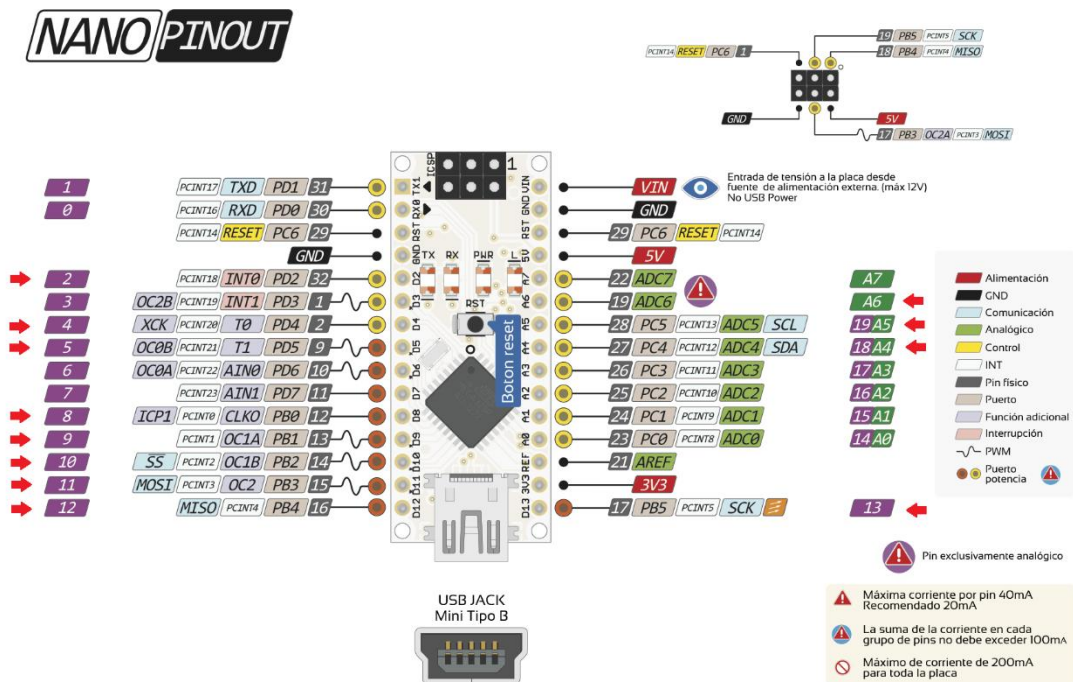


Figura 1: Pines del Arduino Nano utilizados. [1] (Modificación propia para este proyecto)

Cabe destacar que dicho microprocesador cuenta con una entrada mini USB la cual facilita la carga del programa en el robot.

2.4.2.- PLACA EXPANSORA DE ENTRADAS Y SALIDAS

Para poder realizar las conexiones pertinentes ha sido necesario la utilización de una placa de expansión de las entradas y salidas compatible con el Arduino Nano (Figura 2). Ésta añade en cada disposición de las patillas una conexión de 5 V y de GND para facilitar

la conexión directa con los sensores y servomotores, así como una conexión de alimentación máxima de 12 V entre otros.

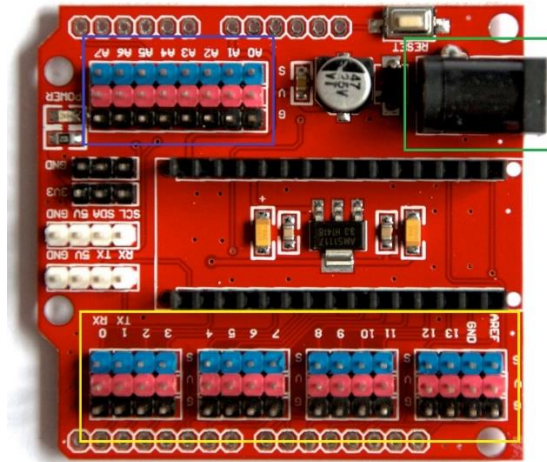


Figura 2: Placa expansora de entradas y salidas compatible con el Arduino Nano. Pines analógicos en color azul. Pines digitales en color amarillo. Conexión de 12 V en verde.[2] (Modificación propia para este proyecto)

Como se muestra en la siguiente tabla de características técnicas (Tabla 2), nos permite tener hasta 14 y 8 pines de entrada y salida digitales y analógicos respectivamente. Además, cuenta con unas dimensiones de 57.2x53.5 mm y una masa de 21 g.

Especificaciones Técnicas de la Placa Expansora	
Compatibilidad	Arduino Nano V2 y V3
Configuración	3P en pines VCD, GND y E/S tanto digitales como analógicas
Tensión de entrada	12 V (Valor máximo)
Dimensiones	57.2x53.5 mm
Masa	21 g
Número de pines Entrada/Salida Digitales	14
Número de pines Entrada/Salida Analógicos	8
Número de pines compatibles con control PWM	6
Número de pines con expansión I2C	5

Tabla 2: Características generales de la placa expansora de entradas y salidas compatible con Arduino Nano. [2]

2.4.3.- MÓDULO BLUETOOTH HC-06

Para realizar las comunicaciones entre el robot y el teléfono móvil del usuario, se utilizará un módulo Bluetooth, concretamente el modelo HC-06 (Tabla 3). Su consumo energético es de 40 mA y se puede alimentar entre los 3.3 V y los 6 V por lo que la tensión de 5 V proporcionada por la placa expansora será suficiente para su correcto funcionamiento. En este caso no aparece la masa del módulo, pero se ha podido pesar y es de 5 g.

Especificaciones Técnicas del Módulo Bluetooth HC-06	
Versión del Bluetooth	Bluetooth V2.0
Ancho de banda de frecuencias	2.40 GHz a 2.48 GHz, con banda ISM
Tipo de conexión de la alimentación	USB v1.1/2.0
Tensiones de funcionamiento	+3.3 V a +6 V
Corriente de funcionamiento	40 mA

Tabla 3: Características generales del módulo Bluetooth HC-06. [3]

2.4.4.- MATRIZ LED 8X8

La matriz de leds 8x8 está dotada de 64 leds de color rojo y un controlador tipo MAX-7219 (Tabla 4) encargado de controlar cada uno de los leds, de manera independiente.

Se estima que el consumo de cada led encendido es de 2 mA y del controlador cuando están todos los leds encendidos es de 330 mA. Además, el rango de tensión para operar correctamente está entre los 4 V y los 5.5 V, por lo tanto, se puede alimentar a 5 V.

Especificaciones Técnicas de la Matriz 8x8	
Tipo de controlador	MAX-7219
Tensión de alimentación funcional	+4.0 V a 5.5 V
Corriente de alimentación funcional del controlador (todos los segmentos encendidos)	330 mA
Número de leds	64
Consumo de corriente de cada LED	2 mA

Tabla 4: Características generales de la matriz LED 8x8 con un controlador MAX-7219. [4]

2.4.5.- CONTROLADORES PCA9685

Los controladores de los servomotores utilizados para controlar las patas son los PCA9685. Este tipo de controlador permite controlar 16 servos o leds mediante un control PWM. Puesto que tan solo cuenta con 16 pines de salida y tenemos 18 servomotores, es necesario el uso de 2 controladores.

Como se muestra en la siguiente tabla de características (Tabla 5) el rango de tensiones de alimentación se encuentra entre los 3 V y los 5 V para la placa, mientras que para la alimentación de los servos su tensión máxima es de 6 V, tensión a tener en cuenta a la hora de alimentar los servomotores.

Especificaciones Técnicas del Controlador PCA9685	
Tensión de alimentación de la placa	3 V a 5 V
Tensión de alimentación de los servos	6 V (Valor máximo)
Rango de frecuencia de salida	40 Hz a 1000 Hz
Tipo de control	PWM (Modulación de ancho de pulso)
Número de canales	16
Resolución por canal	12 bits
Interfaz	I2C
Dimensiones	62.3x25.3 mm

Tabla 5: Características generales del controlador PCA9685. [5]

2.4.6.- SERVOMOTOR SG90

El servomotor SG90 cuenta con una rotación máxima de 180°. Como se aprecia en la Tabla 6, su rango de alimentación va entre los 4 V y los 7.2 V. Además, cuenta con un par de 1.2 Kg·cm alimentado a 4.8 V. El consumo de este servomotor es variable, depende de la carga que tenga y el ángulo que se desee mover. A su vez, se controla a través de una señal PWM y sus engranajes son de plástico.

Especificaciones Técnicas del Servomotor SG90	
Tensión de alimentación	4 V a 7.2 V
Velocidad	0.12 segundos cada 60°
Par	1.2 Kg·cm (alimentado a 4.8 V)
Peso total	10.6 g
Dimensiones	22x11.5x27 mm
Giro máximo	180°
Compatibilidad	Arduino y microcontroladores que funcionen a 5 V
Material de los engranajes internos	Plástico
Control	PWM

Tabla 6: Características generales del servomotor SG90. [6]

2.4.7.- SERVOMOTOR MG996R

El servomotor MG996R cuenta con una rotación máxima de 180° y engranajes metálicos a diferencia del servo SG90. Como se aprecia en la Tabla 7, su rango de alimentación es inferior al del SG90, abarcando desde una tensión mínima de 4.8 V a una máxima de 7.2 V. Este servo cuenta con un par máximo de 11 Kg·cm y su consumo de corriente es variable según la carga y el ángulo al cual se desea mover, pudiendo llegar éste a un consumo máximo de 900 mA.

Especificaciones Técnicas del servomotor MG996R	
Tensión de alimentación	4.8 V a 7.2 V
Velocidad	0.17 segundos cada 60° (4.8 V), 0.14 segundos cada 60° (6 V)
Par	9.4 Kg·cm (4.8 V), 11 Kg·cm (6 V)
Peso total	55 g
Dimensiones	40.7x19.7x42.9 mm
Giro máximo	180°
Banda muerta	5 microsegundos
Consumo de corriente	500 mA a 900 mA (6 V)
Material de los engranajes internos	Metal
Control	PWM

Tabla 7: Características generales del servomotor MG996R. [7]

2.4.8.- SENSOR DE PROXIMIDAD O MÓDULO DE ULTRASONIDOS

Para este proyecto se ha empleado un módulo de ultrasonidos para la detección de obstáculos, concretamente el modelo HC-SR04. En la Tabla 8, podemos observar que su rango de medida se encuentra entre los 2 cm y los 400 cm. La tensión de alimentación es de 5 V y que su consumo energético es de 15 mA.

Especificaciones Técnicas del Sensor de Proximidad	
Tensión de alimentación	5 V
Corriente de trabajo	15 mA
Frecuencia de trabajo	40 KHz
Rango de medida	2 cm a 400 cm
Ángulo de medida	<15°, efectivo 30°
Dimensiones	45x20x15 mm
Salida	Digital
Pulso de disparo	10 microsegundos tipo TTL
Resolución	3 mm

Tabla 8: Características generales del módulo de ultrasonidos HC-SR04. [8]

2.4.9.- BATERÍA

La batería utilizada en el robot hexápodo es una batería tipo Li-Po de dos celdas (2S) cuya tensión nominal es de 7.4 V, su capacidad es de 2200 mAh y su capacidad de descarga continua es de 30C pudiendo tener una descarga de pico o de explosión de 60C (Tabla 9).

Especificaciones Técnicas de la Batería	
Capacidad	2200 mAh
Tipo de batería	Li-Po
Tasa de descarga continua	30C
Tasa de explosión	60C
Tensión	7.4 V (8.4 V cargada al máximo)
Número de celdas	2S (3.4 V nominales cada una)
Dimensiones	105x34x16 mm
Masa	116 g

Tabla 9: Características generales de la batería tipo Li-Po 2S. [9]

2.4.10.- DIODOS MUR420

La Tabla 10 corresponde con las características eléctricas de los diodos MUR420. En ella podemos observar que la corriente en polarización directa media ($I_{F(AV)}$) es de 4 A y la caída de tensión en polarización directa (V_F), en el caso más desfavorable, es de 0.71 V. Además, cabe destacar que soporta hasta 200 V DC.

Especificaciones Técnicas del Diodo MUR420	
Pico repetitivo de tensión en polarización inversa	200 V
Corriente media rectificadora en polarización directa ($I_{F(AV)}$)	4 A ($T_a=80\text{ }^\circ\text{C}$)
Tensión máxima instantánea en polarización directa (V_F)	0.71 V
Temperatura operativa de la unión del semiconductor	-65 °C a +175 °C

Tabla 10: Características generales del diodo MUR420. [10]

2.4.11.- PLÁSTICO ABS

La Tabla 11 corresponde a las propiedades mecánicas y térmicas del acrilonitrilo butadieno estireno (ABS) el cual se ha utilizado para la impresión y construcción del cuerpo del robot.

Cabe destacar sus características mecánicas puesto que posee la capacidad de alargarse hasta la rotura un 4.8% y de un alargamiento de ésta del 3.5% hasta la deformación siendo así suficiente a resistir los posibles esfuerzos que el robot pueda tener. Además, su temperatura de reblandecimiento es de 97 °C, una temperatura que en condiciones normales de uso nunca se alcanzará. Se debe tener en cuenta que la temperatura de fusión, importante factor para poder imprimir en 3D, es de 225 °C siendo esta temperatura fácilmente alcanzable por todas o la gran mayoría de impresoras 3D del mercado.

Especificaciones Técnicas del Acrilonitrilo Butadieno Estireno (ABS)	
Módulo de elasticidad a la tracción	1681.5 MPa
Esfuerzo de tracción a la deformación	39 MPa
Alargamiento a la deformación	3.5%
Alargamiento a la rotura	4.8%
Resistencia a la flexión	70.5 MPa
Módulo de flexión	2070 MPa
Temperatura de reblandecimiento	97 °C
Temperatura de fusión	225 °C a 245 °C

Tabla 11: Características generales del plástico ABS impreso en 3D. [11]

3.- IMPRESIÓN DE PIEZAS

Para la impresión 3D de este prototipo, se ha utilizado una impresora 3D, concretamente el modelo Creality Ender 3 (Figura 3).



Figura 3: Impresora 3D modelo Creality Ender 3. [12]

Esta impresora cuenta con un volumen de impresión de 220x220x250 mm, una extrusión tipo bowden (es decir, el extrusor empuja el filamento de ABS por el interior de un tubo de teflón hasta el hotend donde se funde (Figura 4)), además de contar con una cama caliente o base que alcanza los 90 °C necesarios para poder imprimir ABS.

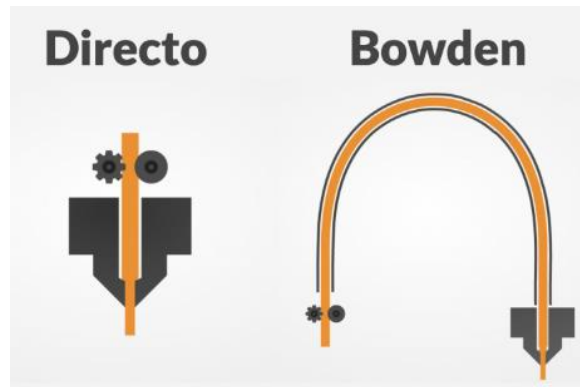


Figura 4: Comparación entre los tipos de extrusión que existen en el mercado. [13]

Durante la impresión de las piezas ha habido ciertas complicaciones como el conocido y más común efecto “warping” o pando (Figura 5) separando e incluso llegando a despegar en su totalidad la pieza de la cama caliente. Este efecto se suele producir por un cambio de temperatura brusco durante la impresión puesto que el ABS es un material que se suele retraer bastante y tiende a deformarse si no se encuentra en su temperatura ideal de impresión.



Figura 5: Ejemplo de efecto “warping”. La base se ha despegado de la cama caliente, deformando así la pieza. [14]

Por otro lado, también se han tenido problemas de “cracking” o de rotura (Figura 6) por los mismos motivos que el “warping”.



Figura 6: Ejemplo de efecto “cracking”. Se ha producido un cambio de temperatura brusco entre capas provocando una separación de éstas. [15]

Una posible solución es imprimir estas piezas en una habitación cerrada con la temperatura controlada o incluso, si cabe la posibilidad, utilizar una impresora cuyo volumen de impresión se encuentre encerrado y con la temperatura y la humedad controladas.

Las piezas del robot se han impreso con las siguientes características:

- Temperatura de impresión: 240 °C
- Temperatura de la cama caliente: Primera capa a 70 °C y el resto a 85 °C
- Velocidad de impresión: 60 mm/s
- Número de paredes: 3
- Número de capas inferiores: 3
- Número de capas superiores: 3
- Relleno y geometría: 20 % y con forma de rejilla
- Altura de capa: 0.2 mm
- Boquilla: 0.4 mm

El tiempo aproximado de impresión de todo el robot, sin tener en cuenta piezas fallidas, es de aproximadamente unas 100 horas.

4.- DISEÑO Y CORRECCIONES

A lo largo del proyecto se ha ido cambiando el diseño tanto a nivel estructural como a nivel de componentes electrónicos.

4.1.- DESARROLLO DEL CUERPO

El primer diseño de la estructura (Figura 7), estaba basado en un robot hexápodo cuya base era hexagonal y las patas estaban separadas con un ángulo aproximado de 60°. Las patas contaban con tan solo dos articulaciones accionadas por servomotores SG90 y este no contaba con sensor de ultrasonidos, tampoco matriz de leds ni pinzas móviles.

En su defecto, contaba con 5 leds, un zumbador y una pequeña pinza meramente estética.

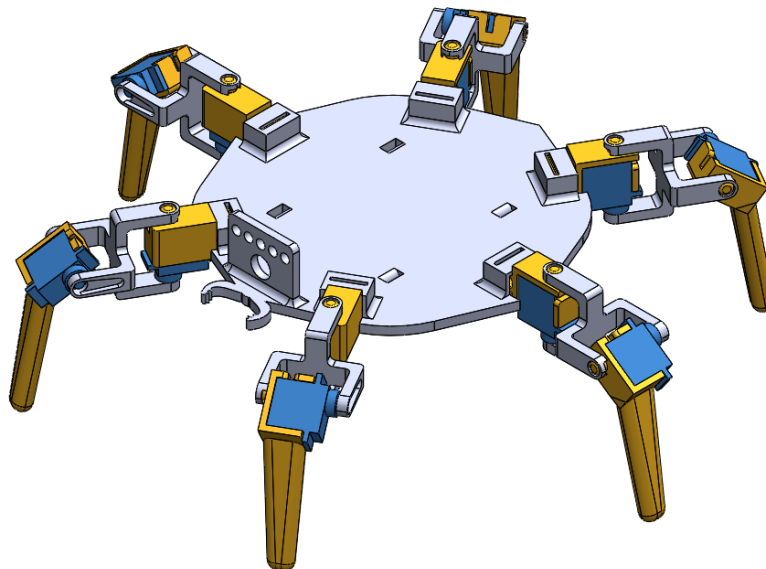


Figura 7: Versión 1 del robot hexápodo.

El segundo diseño (Figura 8) se basaba únicamente en la mejora de la movilidad de las patas, a éstas se les añadió una nueva articulación que le permitiría bajar y subir el cuerpo además del poder andar entre otras posibilidades.

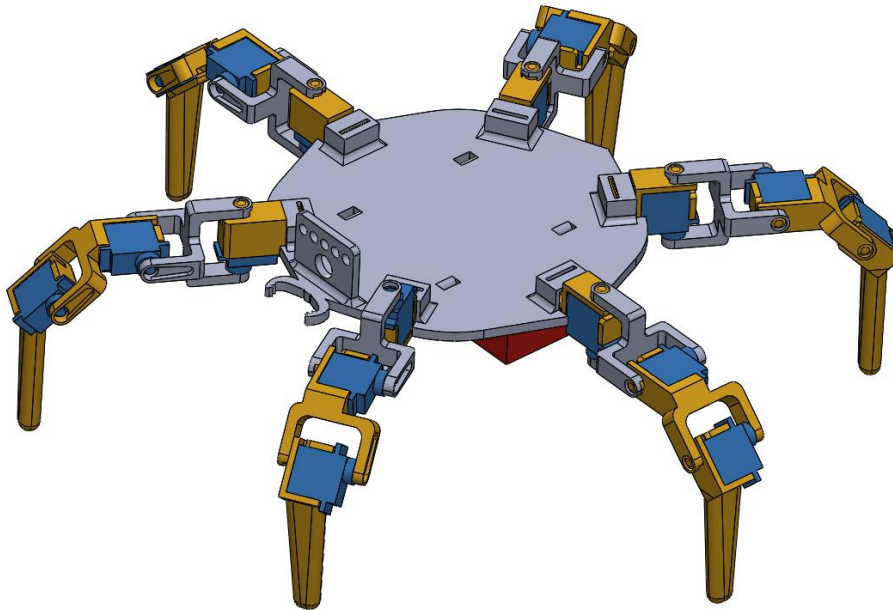


Figura 8: Versión 2 del robot hexápodo.

Ambos diseños se descartaron puesto que no cumplían las especificaciones del encargo además de carecer del suficiente espacio para albergar todos los componentes electrónicos necesarios para poder manejar los nuevos 6 servomotores.

Para solucionar las deficiencias de los diseños anteriores, se creó la versión 3 del robot (Figura 9), la cual cambiaba completamente la geometría del robot recolocando y orientando las patas en los laterales de éste siendo perpendiculares a la recta de simetría del robot y añadiendo un gran volumen interior para albergar la electrónica necesaria. Además, se le desarrolló una pequeña pinza móvil y un lugar para situar la matriz de leds, careciendo de sensor de obstáculos.

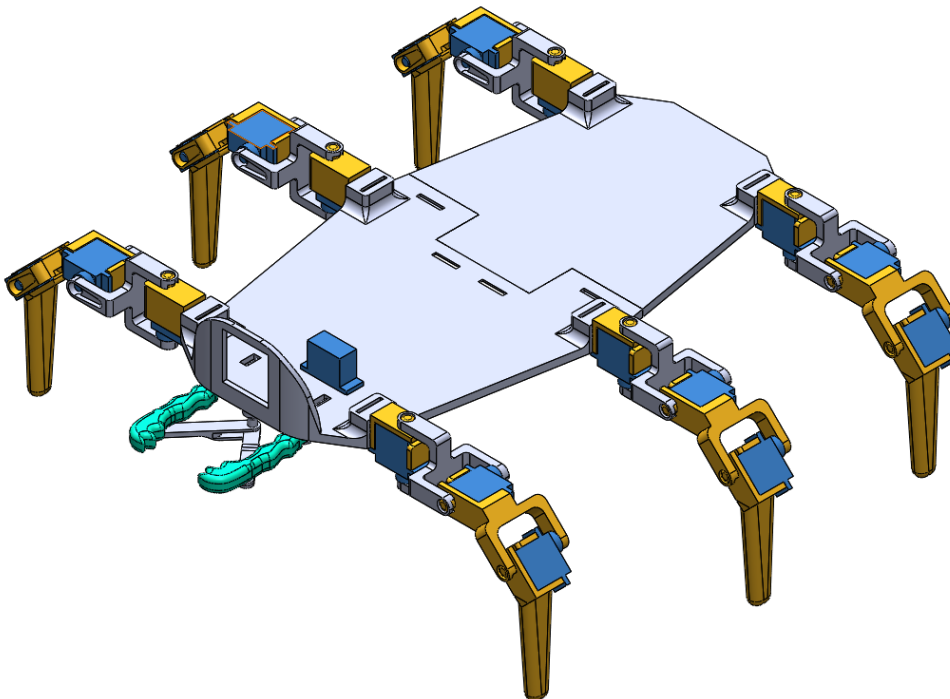


Figura 9: Versión 3 del robot hexápodo.

Con la versión 3 del hexápodo apareció un nuevo problema, la falta de par en los servomotores. Éste era debido a la gran distancia entre patas y peso del nuevo chasis además de los nuevos componentes electrónicos.

Para intentar solucionar este problema se desarrolló una versión del chasis anterior agujereado (Figura 10) el cual reducía el peso, pero no siendo éste suficiente.

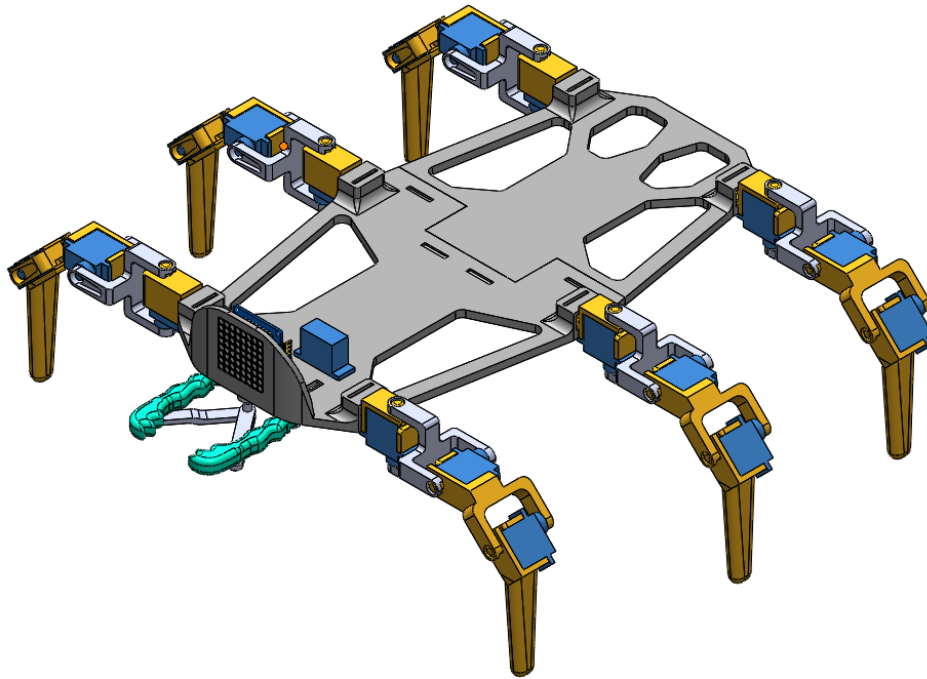


Figura 10: Versión 4 del robot hexápodo.

Tras estos problemas de falta de fuerza en los servomotores, se decidió cambiar el modelo del servo, pasando del SG90 con dientes de plástico y 1.8 Kg·cm de par a los servomotores MG996R con dientes metálicos y un par de 11 Kg·cm.

Este cambio repercutió en el tamaño de las patas, así como en la necesidad de reforzar la propia estructura para evitar el pandeo de ésta, además de proteger la electrónica de su interior puesto que se trata de un prototipo destinado principalmente a la educación.

Con todo esto se creó la versión 5 del robot (Figura 11). También, se decidió cambiar el tamaño de las pinzas para hacerlas funcionales y realizarle una mejora estética considerable. Además, en este modelo, se añadió un soporte en la parte superior del robot para el módulo de ultrasonidos.

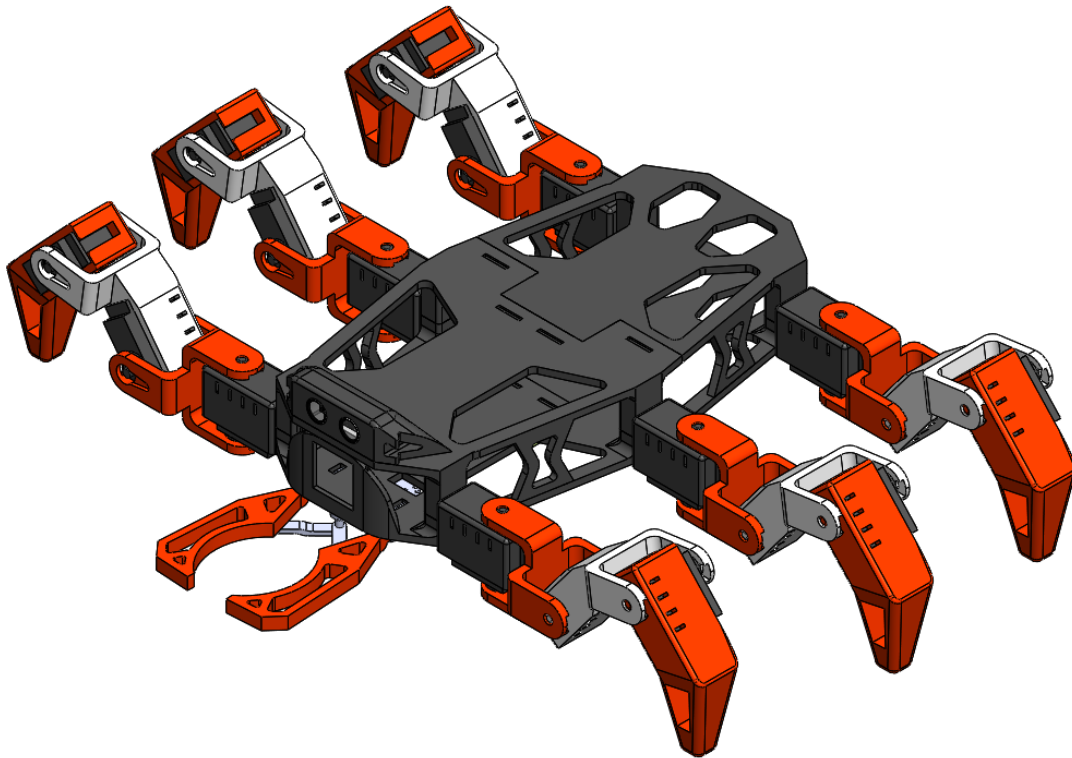


Figura 11: Versión 5 del robot hexápodo.

Finalmente, tras realizar diferentes pruebas con la versión 5 del robot y observar algunos fallos como podían ser la falta de rigidez del chasis, la estabilidad y el par, se decidió crear una versión definitiva de éste, la versión 6 (Figura 12). Ésta conseguía solucionar los problemas del modelo anterior además de añadirle dos agujeros en la parte delantera para dos leds blancos y un soporte para el interruptor de apagado y encendido en la parte trasera del chasis.

Se decidió reorientar las patas delanteras y traseras, dándole un ángulo de 30° respecto a las patas centrales mejorando así su estabilidad al andar. Se vació considerablemente el chasis superior. Se reforzó el chasis inferior reduciendo los tamaños de los agujeros. Se añadió un soporte inferior para sujetar y proteger la batería, así como un soporte para el interruptor de encendido en la parte trasera del chasis inferior como ya se ha comentado. Se redujo en 50 mm la distancia entre las patas derechas e izquierdas reduciendo así el par necesario.

Finalmente, para facilitar la impresión y el tiempo de ésta, se decidió separar en dos partes el chasis superior: los pilares, que unen el chasis inferior y superior, y el propio chasis superior.

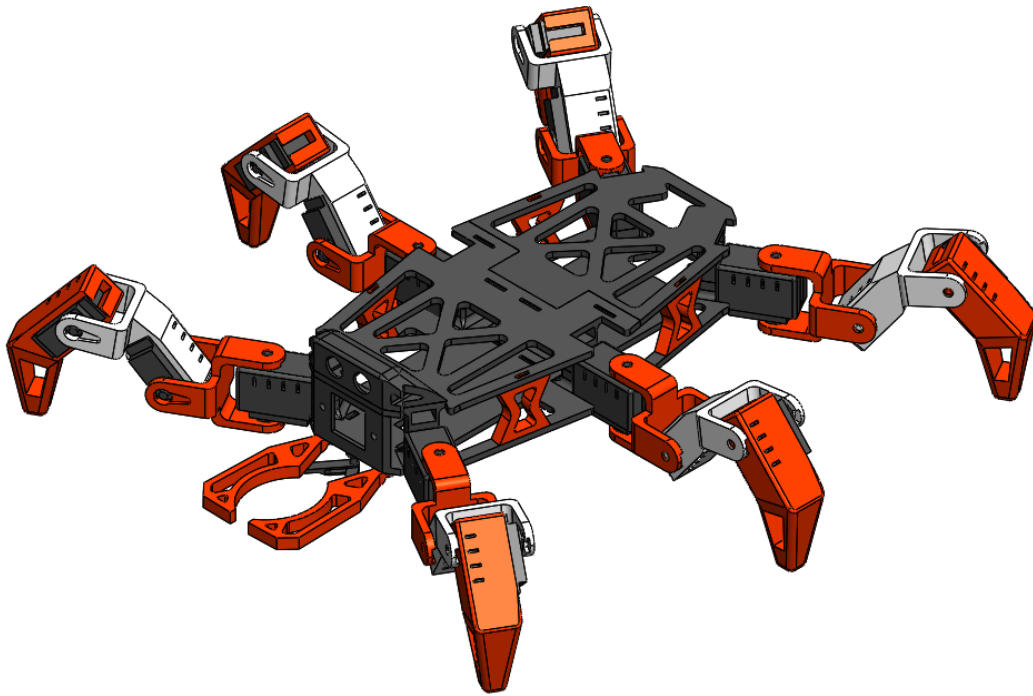


Figura 12: Versión 6 y final del robot hexápodo.

4.2.- ELECCIÓN Y DESCARTE DE LA DE ALIMENTACIÓN

Para la alimentación del robot se plantearon tres posibles soluciones:

En primer lugar, se pensó utilizar pilas alcalinas, concretamente dos porta-pilas de 4 pilas AA de 1.5 V consiguiendo una tensión total de 6 V los cuales, cada uno, alimentarían un controlador PCA y por consiguiente los servomotores de las patas. Además, se usaría otra batería de 5 V adicional para alimentar el microprocesador y el resto de los periféricos conectados directamente a éste.

Esta opción se descartó rápidamente puesto que el consumo de pilas era excesivo, la corriente que podían proporcionar las pilas era insuficiente para que los servomotores alcanzaran su par máximo, estando limitado éste por el par máximo de los SG90, además de todo el peso extra que suponía el contener 3 fuentes de alimentación distintas.

La segunda opción que se planteó fue conectar el robot directamente a la red eléctrica mediante una etapa AC/DC cuya salida fuera 5 V o 6 V. También se descartó pues se trata de un robot móvil y con este tipo de alimentación era necesario estar conectados a la red mediante cables, algo molesto e incluso peligroso.

La última opción y la escogida es la utilización de una batería tipo Li-Po de dos celdas cuya tensión nominal es de 7.4 V. Este tipo de baterías soluciona el problema relacionado con la corriente, pero añade el problema de que cuando está completamente cargada su tensión es de 8.4 V suponiendo un problema a la hora de alimentar los controladores PCA puesto que su tensión máxima según el fabricante es de 6 V. Esto ha obligado a diseñar una etapa intermedia entre los controladores y la batería.

Para poder adaptar la tensión se barajó entre dos opciones, el uso de un convertidor DC-DC tipo BUCK o el uso de diodos.

Finalmente, se eligió el uso de diodos puesto que son mucho más económicos además del hecho de poder crear 3 ramas de alimentación distintas, así como fijar la corriente máxima por rama a 4 A. Se ha creado una rama para alimentar el microprocesador y sus periféricos sin limitar la tensión de la batería y dos ramas idénticas que disminuyen la tensión para que en el peor de los casos la tensión en la entrada de los controladores PCA sea como mucho de 5.5 V aproximadamente.

5.- DESCRIPCIÓN DETALLADA DE LA SOLUCIÓN ADOPTADA

El diseño de la solución adoptada se muestra en la Figura 13. El prototipo final está separado en cuatro subsistemas como se puede observar en el organigrama de la Figura 15.

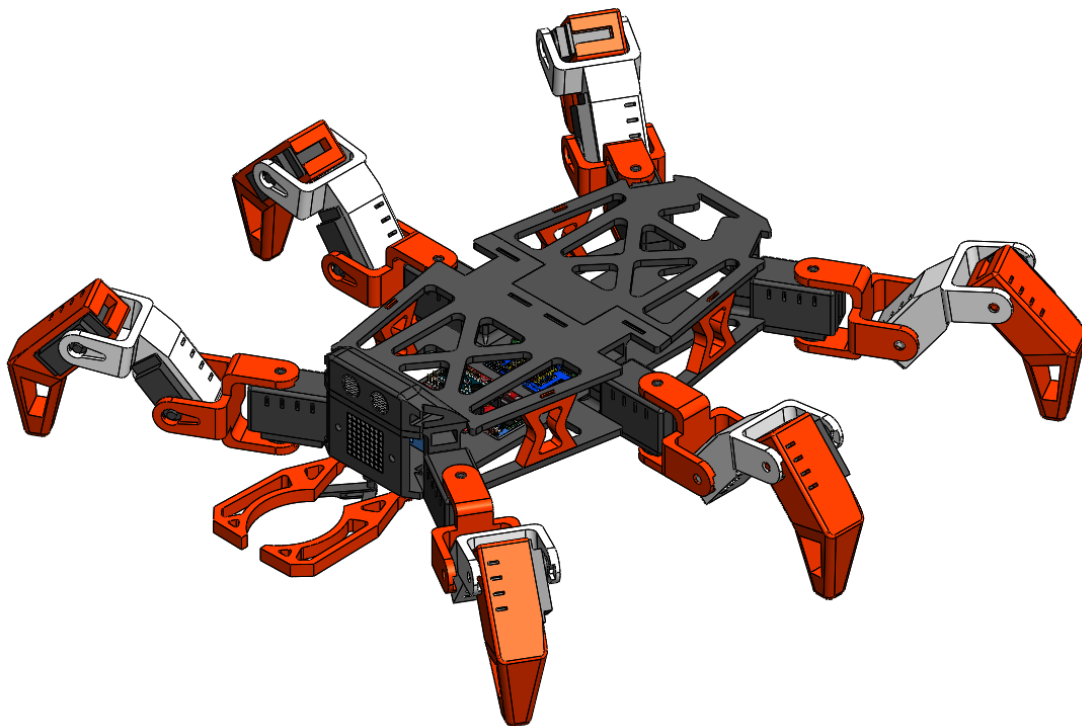


Figura 13: Diseño de la solución adoptada.

En primer lugar, encontramos el subsistema del chasis. Dentro de este subsistema se encuentran 5 partes importantes; chasis inferior, chasis superior (estos dos últimos se subdividen en parte delante y parte trasera), luego los pilares, el soporte de la batería y finalmente el soporte del módulo de ultrasonidos. Todas estas partes han sido impresas en 3D con plástico ABS con las características que aparecen en el punto 4 de este documento.

Su principal función es sostener y proteger toda la electrónica del robot además de permitir anclar y fijar las patas.

El chasis en sí está creado para que en su parte delantera frontal pueda albergar dos leds blancos y una matriz led de 8x8. Además, tanto en la parte superior delantera como en la inferior delantera, se encuentran unos anclajes para que el soporte del sensor de ultrasonidos y las pinzas queden fijadas en su posición permitiendo así su fácil cambio en caso de rotura de dichas piezas.

En la parte interior del chasis se encuentran atornillados o pegados el servomotor de las pinzas, el módulo bluetooth, el microprocesador, los dos controladores PCA9685 y la etapa para adecuar la tensión de alimentación. Por otro lado, en la parte media inferior, encontramos anclado el soporte de la batería. Además, en la parte trasera hay un orificio para poder anclar el interruptor de encendido y apagado del robot.

El siguiente subsistema por describir, es el de la estructura de las pinzas o de la boca. Este subsistema está constituido por diferentes piezas que en conjunto logran transmitir el movimiento del servomotor SG90 a las propias pinzas, abriéndolas o cerrándolas. Este subsistema en su totalidad está construido e impreso con plástico ABS y cuya utilidad principal es coger objetos para poder transportarlos.

El subsistema de las patas es uno de los más importantes, cada pata está compuesta por 4 piezas; hombro, unión hombro-codo, codo y muñeca. Dentro de las piezas del hombro, codo y muñeca encontramos un servomotor MG996R encargado de la movilidad del dispositivo. Se ha tenido en cuenta que la estructura del chasis y las patas cuando el robot está en la posición estática tenga forma de arco (Figura 14), siendo así mucho más resistente a su propio peso.

Mediante la coordinación de las 6 patas y de sus 18 articulaciones el dispositivo es capaz de moverse y realizar diferentes movimientos.

Al igual que el chasis y las pinzas, todas estas piezas han sido impresas en 3D con plástico ABS.

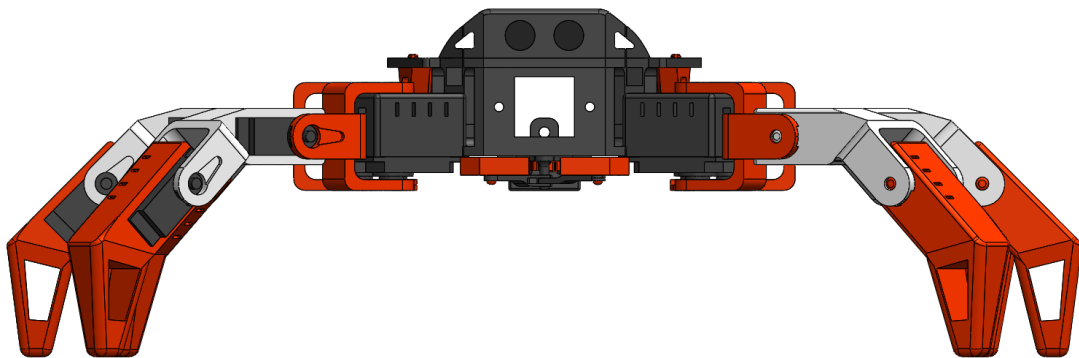


Figura 14: Posición estática del robot donde la forma de las patas con el cuerpo forma un arco.

Finalmente, el subsistema de la unidad de control, como ya se ha comentado anteriormente se encuentra en el interior del robot protegido por el chasis y las patas. Es el encargado de todo el control del dispositivo, está alimentado por una batería tipo Li-Po de dos celdas. La comunicación se realiza a través del módulo bluetooth el cual se comunica con el smartphone del usuario. Además, para la detección de posibles obstáculos cerca de él se utiliza un sensor o módulo de ultrasonidos.

Cabe destacar que el entorno de programación del robot se realiza en Arduino, dándole la sencillez y accesibilidad que éste ofrece.

DOCUMENTO NÚMERO 1: MEMORIA

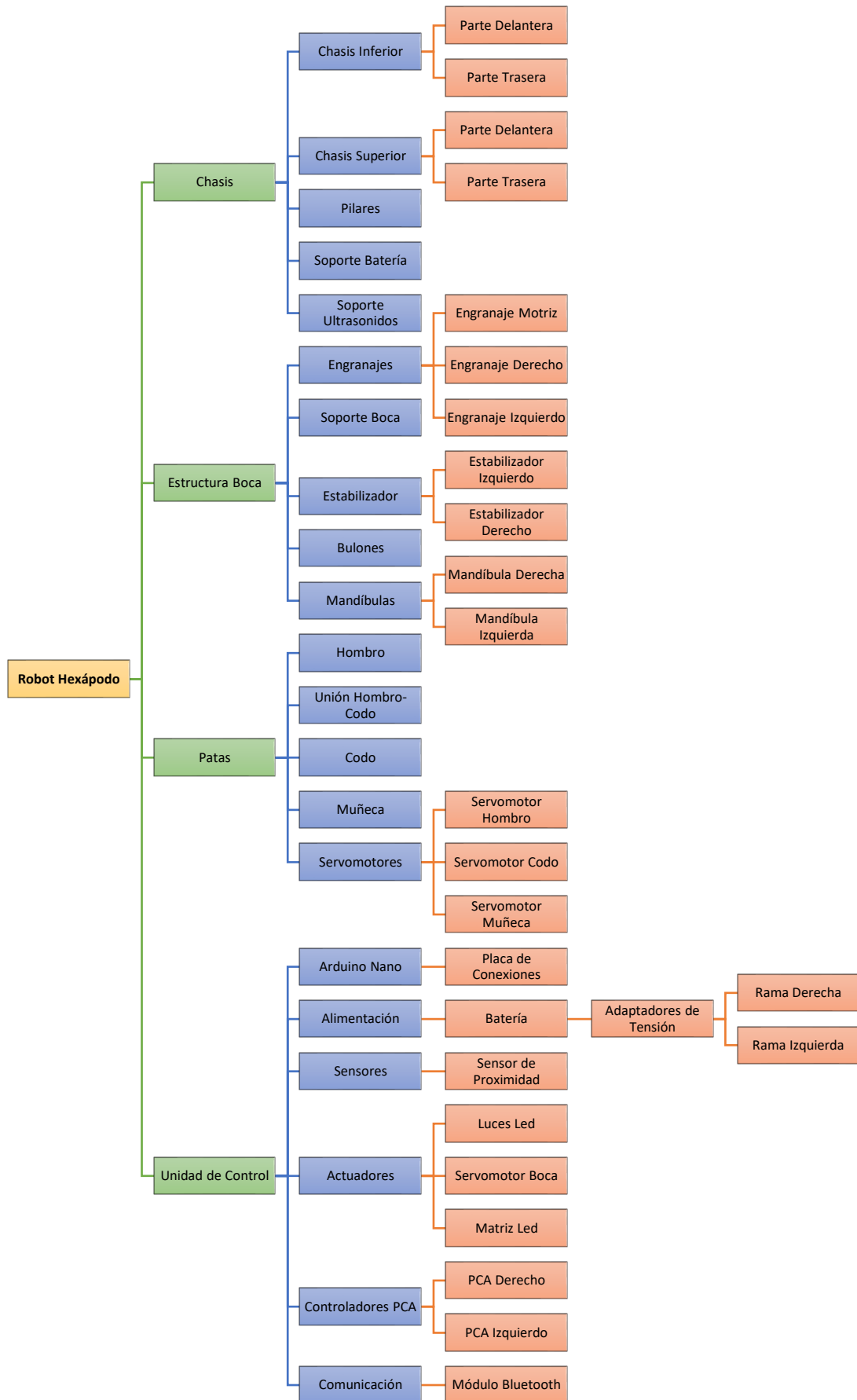


Figura 15: Organigrama del Proyecto.

6.- JUSTIFICACIÓN DETALLADA DE LA SELECCIÓN Y DIMENSIONADO DE LOS COMPONENTES ELECTRÓNICOS

Para la elección de todos los elementos mencionados a continuación siempre se ha tenido en cuenta el peor de los casos posibles para poder asegurar las prestaciones y funcionalidades del robot en todo momento.

6.1.- ELECCIÓN DEL CONTROLADOR ARDUINO NANO

Para la elección del microprocesador se ha optado por el modelo de Arduino Nano por diferentes razones:

Su tamaño, un factor importante a tener en cuenta a la hora de construir un robot móvil ya que el espacio interior de éstos suele ser reducido a diferencia de otras configuraciones de robots. Además, el Arduino Nano permite poder acoplarlo a una placa expansora de entradas y salidas que facilita las conexiones de los sensores y actuadores, ocupando un menor espacio que el propio Arduino Uno. El tamaño, teniendo en cuenta la placa expansora, es de 56x54 mm a diferencia del Arduino Uno que es de 80x55 mm.

Otro factor por considerar es el rango de tensiones de alimentación que nos permite tener la placa expansora de entradas y salidas puesto que es muy amplio y además que se adecua al tipo de alimentación elegido permitiendo ser alimentado hasta con 12 V DC, aunque para el proyecto la tensión máxima de alimentación sea de 8.4 V DC para cuando la batería esté completamente cargada.

Por último, pero no menos importante, es el número de entradas/salidas que contiene, puesto que este cuenta con 15 pines digitales y 8 analógicos y para la realización del proyecto hacen falta al menos 9 pines digitales y 3 analógicos.

6.2.- ELECCIÓN DEL MÓDULO BLUETOOTH

Se ha decidido elegir el módulo Bluetooth modelo HC-06 puesto que es un modelo muy económico, fácil de utilizar, con muy buenas prestaciones y que existe una gran comunidad de usuarios que lo utiliza, lo cual facilita la búsqueda de información sobre dicho módulo. Además de ser compatible con las tensiones de alimentación que ofrece el Arduino Nano.

El módulo Bluetooth se puede encontrar con un precio de 2.60 € aproximadamente, siendo este muy económico en comparación con otros modelos y fácil de sustituir y de volver a conectar en caso de ser dañado.

6.3.- ELECCIÓN DE LOS SERVOMOTORES MG996R Y SG90

Para la selección de estos servomotores se ha tenido que realizar una serie de cálculos para determinar el par que estos necesitan.

Para calcular el par motor necesario se ha procedido de la siguiente manera:

En primer lugar, ha sido necesario saber las masas de cada una de las piezas y de los elementos que componen el robot (Tabla 12).

COMPONENTES O ELEMENTOS	MASA (g)
Hombro	19
Unión Codo-Hombro	20
Codo	29
Muñeca	32
Servo MG996R	56
Chasis Inferior Delantero	65
Chasis Inferior Trasero	42
Chasis Superior Delantero	74
Chasis Superior Trasero	51
Mandíbula Derecha	18
Mandíbula Izquierda	18
Soporte Pinzas	5
Estabilizador Derecho	2
Estabilizador Izquierdo	2
Engranaje Derecho	2
Engranaje Izquierdo	2
Bulón (2 unidades)	1 (cada bulón)
Soporte Ultrasonidos	32
Sensor de Ultrasonidos	10
Soporte Batería	16
Batería	135
Arduino Nano + Placa Expansora	40
Controlador PCA (2 unidades)	10 (cada controlador)
Matriz Led 8x8	15
Módulo Bluetooth	10
Servomotor SG90 de las Pinzas	11
Placa PCB	5
Diodos (8 unidades)	5 (en total)
Cables y Conexiones	40 (aproximadamente)

Tabla 12: Masas de los elementos del robot hexápodo.

Para facilitar el cálculo se ha decidido compactar algunas masas, como puede ser la masa total del chasis que conjuntamente es de 232 g, la masa total de las pinzas que es de unos 51 g y la masa total de la electrónica y de la batería que suponen 285 g.

Como es sabido, para realizar cálculos de par sobre un punto es necesario saber a qué distancia se encuentran las masas a dicho punto (Tabla 13). Se ha considerado que el caso peor dónde hace falta más par motor es en la articulación del codo de las patas centrales del robot (Figura 16).

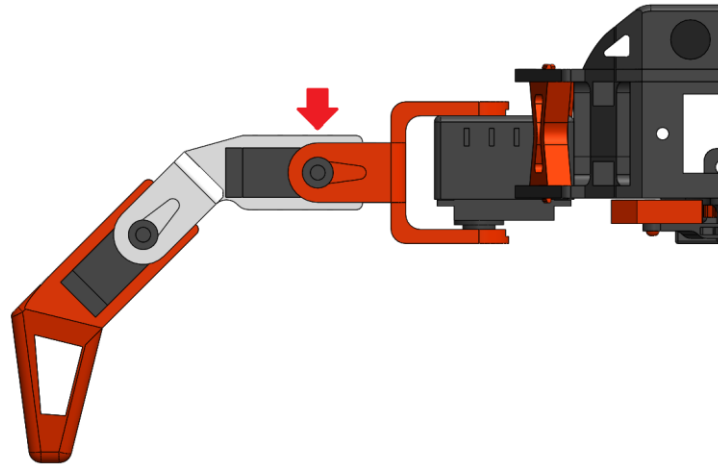


Figura 16: Punto crítico a partir del cual se calcula el par motor necesario.

COMPONENTES O ELEMENTOS	DISTANCIA (mm)
Chasis	165
Pinzas	165
Hombro	60
Servomotor del Hombro	60
Unión Hombro-Codo	40
Electrónica y Batería	165

Tabla 13: Distancia entre el centro de masas de los elementos y el punto crítico.

Una vez disponemos de todos los datos necesarios, calculamos el momento generado en el punto de la Figura 16. Para ello, realizamos el sumatorio de las masas de cada elemento multiplicado por la distancia al punto crítico. Además, puesto que el robot debe andar y realizar diferentes acciones, consideraremos el caso peor, el cual implica tener tres patas levantadas.

La masa de cada pata es de aproximadamente 268 g por lo tanto, la masa total de las tres patas será de 804 g. La distancia, considerando que las patas levantadas se sostienen ancladas sobre el chasis y aplicando el caso menos favorable, será de 165 mm situando esta masa en el centro de masas del robot.

Tras considerar todo esto, el par necesario asciende a 9.1 Kg·cm aproximadamente siendo éste inferior a los 11 Kg·cm que logra proporcionar el servomotor MG996R y por consecuencia lograr mover sin ningún problema el robot.

Por otro lado, para la elección del servomotor SG90 se buscaba tan solo que fuera ligero además de que pudiese adecuarse a la tensión y a la corriente proporcionada por la placa expansora. Con un par de 1.8 Kg·cm es suficiente para abrir y cerrar las pinzas además de poder sujetar algún objeto ligero como puede ser una lata de refresco.

6.4.- ELECCIÓN DE LOS CONTROLADORES PCA9685

Para la realización de este proyecto se necesita una o varias placas controladoras de servos por diferentes motivos.

La primera y la más obvia, al tener 18 servomotores que controlar no hay suficientes pines en la placa expansora además de no poder realizar un control PWM en todos ellos si los hubiera.

Otro inconveniente muy importante es que no es posible conectar este tipo de servos directamente a la placa expansora debido al alto consumo de corriente que éstos necesitan para funcionar y producir los 11 Kg-cm de par motor necesarios para el movimiento del robot. El consumo de cada servomotor, en caso de quedar bloqueado puede ascender hasta casi 1 A quemando por tanto la placa expansora y el Arduino Nano, además de posiblemente crear un incendio en el robot, algo totalmente indeseado.

Finalmente, se decidió optar por el uso de dos controladores PCA9685 puesto que con solo uno no hay suficientes pines, ya que cuenta con tan solo 16 de los 18 pines necesarios. Por otro lado, no se eligió una placa controladora con 32 pines por temas económicos, ya que una PCA9685 cuesta 1.82 € y la de 32 pines alrededor de diez veces más, 17.42 €.

Cabe destacar que también se eligieron dos placas debido a la cantidad de corriente necesaria que necesitan 18 servomotores moviéndose simultáneamente, y aunque no aparezca en ninguna hoja de características de éstos, al tratarse de una placa impresa, se consideró que podrían hacer un “cuello de botella” en cuanto a la corriente que se deja pasar y por tanto limitar en par motor a los servos.

6.5.- ELECCIÓN DEL MÓDULO DE ULTRASONIDOS

Se decidió seleccionar el módulo de ultrasonidos HC-SR04 por razones parecidas al módulo Bluetooth.

En primer lugar, destaca por sus buenas prestaciones a un muy bajo coste. Se puede encontrar por menos de un euro en el mercado, alrededor de los 70 céntimos.

Este sensor logra ofrecer un buen rango de medida, entre los 2 cm y los 400 cm con una resolución máxima de 3 mm, suficiente para la aplicación que se le desea dar. Además, es un sensor muy utilizado por la comunidad favoreciendo así tanto la búsqueda de información del propio sensor y la facilidad de encontrar diferentes funcionalidades o maneras de programarlo.

Cabe destacar que la tensión de alimentación para su correcto funcionamiento se adecua a la que puede ofrecer la placa extensora del Arduino Nano, 5 V, asimismo un bajo consumo de corriente de alrededor de los 15 mA.

6.6.- ELECCIÓN DE LA MATRIZ DE LEDS 8X8

Para la elección de esta matriz además de buscar la funcionalidad, ligereza y la compatibilidad con la placa extensora y el Arduino Nano, se buscó que principalmente fuera acorde con la estética del robot. Es importante que los leds sean de color rojo puesto que gran parte de la estructura del robot es roja.

Por otro lado, el factor económico también es importante ya que su coste actualmente en el mercado ronda los 90 céntimos.

Cabe destacar que al contener integrado el controlador MAX-7219 permite controlar con tan solo 3 pines digitales 64 leds pudiendo controlar individualmente cada uno de ellos.

6.7.- ELECCIÓN DE LOS DIODOS

Para adecuar la tensión de la batería a la entrada de la alimentación de los controladores PCA9685 se utilizaron ocho diodos separados en dos ramas, rama derecha y rama izquierda, cuatro diodos en serie en cada una.

Para seleccionar el modelo y la cantidad de diodos se tuvieron en cuenta diferentes aspectos como su disponibilidad en tiendas, así como la caída de tensión en ellos y la corriente máxima que logran soportar.

El modelo seleccionado es el MUR420 puesto que puede soportar una corriente en polarización directa media de hasta 4 A y su caída de tensión en el caso menos favorable, cuando el diodo está caliente, ronda los 0.71 V según el fabricante.

Por otra parte, siendo el caso más desfavorable el momento donde la batería está totalmente cargada, 8.4 V, la tensión máxima que tendría en la entrada de cada PCA sería de aproximadamente 5.5 V siendo ésta inferior a los 6 V de tensión máxima soportada por los controladores de los servos.

Ahora bien, a medida que la batería se va descargando, la tensión disminuye, así que la tensión mínima a la que la batería puede llegar a descargarse es de aproximadamente 7 V, puesto que si este umbral inferior es superado (4.2 V en la entrada de alimentación de los controladores), los servos trabajarán fuera de su rango de tensión y por tanto dejarán de funcionar correctamente.

Para avisar al usuario de que las baterías se encuentran descargadas se utiliza un pequeño módulo conocido como salva-baterías, el cual pitará alertando al usuario cuando la batería supere un umbral de tensión previamente definido. Éste se encuentra conectado al terminal de balanceo de las celdas de la propia batería.

A continuación, en la Tabla 14, se muestran las relaciones entre la tensión de la batería y la tensión que llega a la entrada de alimentación de los PCA9685 tras haber pasado por los diodos MUR420.

TENSIÓN EN LA BATERÍA	TENSIÓN EN LOS PCA9685
8.4 V (tensión máxima de la batería)	5.5 V
7.4 V (tensión nominal de la batería)	4.6 V
7 V	4.2 V (tensión mínima del rango de tensiones de los servos)
6.4 V (tensión mínima de la batería)	3.6 V (tensión fuera del rango de tensiones de los servos)

Tabla 14: Relación entre la tensión en la batería y la tensión en los controladores PCA9685.

6.8.- ELECCIÓN DE LA BATERÍA

Para la selección de la capacidad, el factor de descarga continua y duración se ha decidido optar por el caso más desfavorable, máximo consumo de corriente de todos los elementos del robot, así como su factor de simultaneidad igual a 1.

Por otro lado, entre todas las baterías tipo Li-Po del mercado se decidió seleccionar una de dos celdas (2S) puesto que tiene una tensión nominal de 7.4 V a diferencia de las de una celda (1S) con 3.7 V, siendo insuficiente para alimentar los componentes del robot o la de tres celdas (3S) con una tensión de 11.1 V, dificultando así la adecuación de

tensiones en los controladores PCA además del incremento tanto en el coste económico como en la masa de ésta.

En la siguiente tabla (Tabla 15) se muestran los consumos máximos de corriente de los distintos elementos del robot:

ELEMENTO	NÚMERO DE ELEMENTOS	CORRIENTE REQUERIDA POR UNIDAD (mA)	CONSUMO TOTAL (mA)
Arduino Nano	1	40	40
Leds de la Matriz	64	2	128
MAX-7219	1	330	330
Led blanco	2	2	4
Módulo Ultrasonidos	1	15	15
Servo SG90	1	250	250
Servo MG996R	18	900	16200
Módulo Bluetooth	1	40	40
Controlador PCA9685	2	20 (se estima)	40
		TOTAL:	17047

Tabla 15: Consumos máximos de los componentes del robot hexápodo y su consumo total.

Por lo tanto, eligiendo una batería con 2200 mAh de capacidad con una capacidad de 30C¹ de descarga continua logramos que se pueda utilizar el dispositivo, en su peor caso, 7 minutos y 45 segundos aproximadamente.

Puesto que en ningún caso se van a utilizar todos los elementos simultáneamente ni consumir la corriente máxima en todo momento, la duración de la batería será mucho más.

¹ En las baterías tipo Li-Po se suelen utilizar los términos C para hacer referencia a la capacidad de esa batería a otorgar constantemente una corriente. La letra C hace referencia a la capacidad de dicha batería y la capacidad de descarga continua, por ejemplo, 30C es el producto de la capacidad por el número. Por ejemplo, una batería con una capacidad de 2200 mAh con un marcado de 30C podría otorgar una corriente de 66000 mA o 66 A.

7.- PROGRAMACIÓN

Para la programación del robot hexápodo se ha tenido que programar tanto el propio robot como una aplicación móvil la cual utiliza la conexión Bluetooth para poder controlarlo.

7.1.- PROGRAMACIÓN DEL ROBOT

En este apartado se pretende explicar aquellos aspectos más importantes que han hecho posible la programación y el correcto funcionamiento del robot. Si se desea ver el código completo véase el “Anexo A-3: Programación del robot. Código Arduino” adjunto a este documento.

7.1.1.- PROGRAMACIÓN DE LOS CONTROLADORES PCA9685

En primer lugar, para poder realizar un control mediante la modulación PWM sobre los servomotores conectados a los controladores PCA9685, es necesario explicar en qué consiste dicha modulación.

La modulación por ancho de pulso o PWM de sus siglas en inglés “Pulse Width Modulation” consiste en una señal con una determinada frecuencia, en nuestro caso 50

Hz (20 ms de período), que consta de dos valores fijos: uno alto, que es la amplitud, y otro bajo, que es el valor nulo. Además, consta de otro parámetro conocido como ciclo de trabajo que hace referencia al porcentaje de tiempo que dicho pulso se encuentra en estado alto durante cada período. Variando dicho ciclo de trabajo, conseguimos que los servomotores puedan moverse entre los 0° y los 180° (Figura 17).

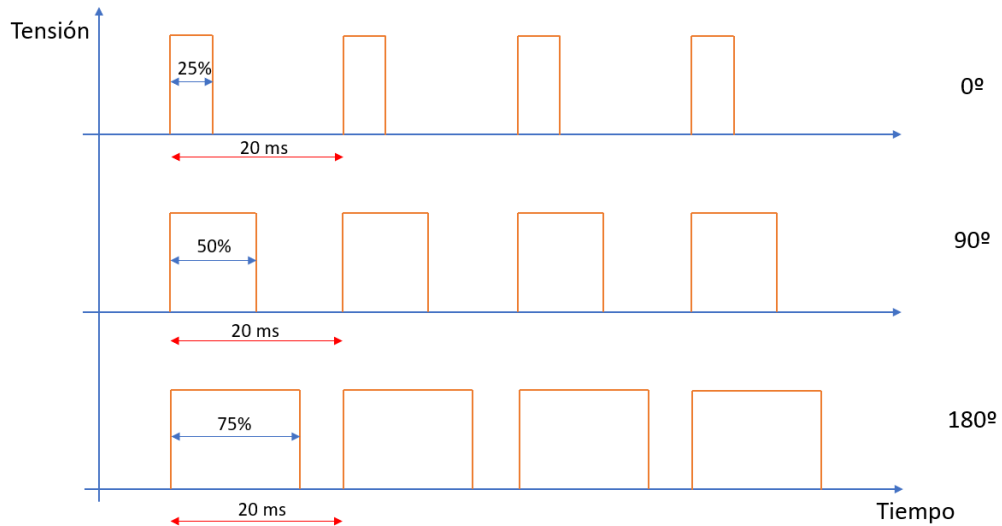


Figura 17: Ejemplo de modulación de ancho de pulso (PWM).

Para ello se han utilizado las siguientes líneas de código:

En primer lugar, es necesario incluir la librería [Adafruit_PWMServoDriver](#) la cual permite que el programa compile correctamente.

En segundo lugar, será necesario determinar la dirección de cada controlador PCA, puesto que tenemos dos controladores será necesario tener dos direcciones distintas. Para conseguir tener una dirección única para cada PCA será necesario soldar uno de los “jumpers” marcados con flechas rojas en la Figura 18. Estos “jumpers” nos permiten tener hasta un total de 7 PCA9685 conectados con distintas direcciones.

En nuestro caso se ha soldado el primer “jumper” contando desde la derecha.

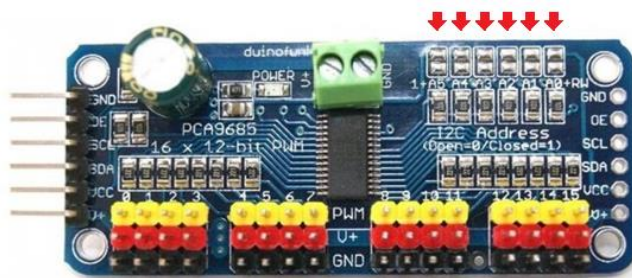


Figura 18: Controlador PCA9685 utilizado en el robot hexápodo. [16]

Tras soldar dicho “jumper” introducimos las siguientes líneas de código:

```
Adafruit_PWMServoDriver pwm1 = Adafruit_PWMServoDriver(0x40);
```

```
Adafruit_PWMServoDriver pwm2 = Adafruit_PWMServoDriver(0x41);
```

Las direcciones de los controladores están expresadas en forma hexadecimal y son 0x40 y 0x41, para las patas derechas (pwm1) y para las patas izquierdas (pwm2) respectivamente. Con esto conseguimos tener dos controladores independientes entre sí puesto que si no se realiza esta modificación las patas derechas e izquierdas se moverían de la misma forma y en vez de tener 18 servos independientes tendríamos tan solo 9.

A continuación, debemos definir el número de “ticks” mínimos y máximos correspondientes a la amplitud del pulso en la modulación PWM. Este número de “ticks” se ha obtenido experimentalmente y podemos concluir que 110 “ticks” corresponden a los 0° y los 500 “ticks” a los 180° aproximadamente.

También es necesario fijar la frecuencia de trabajo, 50 Hz, de los propios servos mediante las siguientes líneas de código:

```
pwm1.setPWMFreq(Frecuencia);  
pwm2.setPWMFreq(Frecuencia);
```

Tras realizar estos pasos, se han creado las siguientes funciones:

```
void writeServoD( int nServo, int posicion)  
{  
    int pos = map (posicion, 0, 180, anguloMinimo, anguloMaximo);  
    pwm1.setPWM(nServo, 0, pos);  
}  
void writeServoI( int nServo, int posicion)  
{  
    int pos = map (posicion, 0, 180, anguloMinimo, anguloMaximo);  
    pwm2.setPWM(nServo, 0, pos);  
}
```

Estas funciones hacen una relación entre grados y “ticks”, de manera que el programador introduce los grados de giro deseados en vez del número de “ticks” siendo así mucho más sencillo de programar el movimiento de las patas del robot además de indicarle el pin de la placa PCA9685 al que está conectado el servomotor que desea mover.

Tras realizar todas estas funciones, cuando se desee mover un servomotor, tan solo será necesario introducir el comando writeServoD o writeServoI según si se desea mover un servo de las patas derechas o de las patas izquierdas, así como indicando el número del servo (Figura 19) y el ángulo de giro deseado.

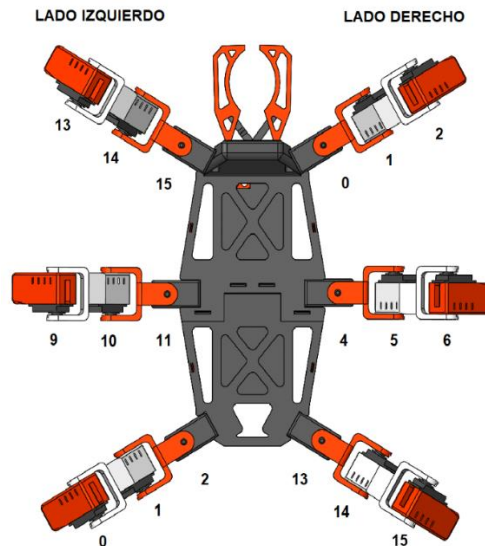


Figura 19: Numeración de los servomotores del robot hexápodo.

7.1.2- MOVIMIENTOS DEL ROBOT

El robot tiene diferentes movimientos para poder realizar su desplazamiento. Para poder explicarlos más fácilmente se harán uso de diferentes esquemas.

7.1.2.1- MOVIMIENTO HACIA ADELANTE

En primer lugar, el desplazamiento hacia adelante sigue el ciclo descrito en la Figura 20. Este ciclo consiste en ir adelantando una a una todas patas del robot. Esto permite que siempre haya 5 patas en contacto con el suelo dándole una total estabilidad además de asegurar que el par necesario para sostener el robot no sea superior al máximo que puedan ejercer los servos.



Figura 20: Ciclo realizado por las patas del robot hexápodo para moverse hacia adelante.

7.1.2.2.- GIRO HACIA LA DERECHA

Para girar a la derecha, se realiza el ciclo descrito en la Figura 21. En todo momento siempre hay 5 patas en contacto con el suelo.



Figura 21: Ciclo realizado por las patas del robot hexápodo para girar hacia la derecha.

7.1.2.3.- GIRO HACIA LA IZQUIERDA

Para girar a la izquierda, el proceso es inverso al ciclo de girar a la derecha. Lo podemos observar descrito en la Figura 22.

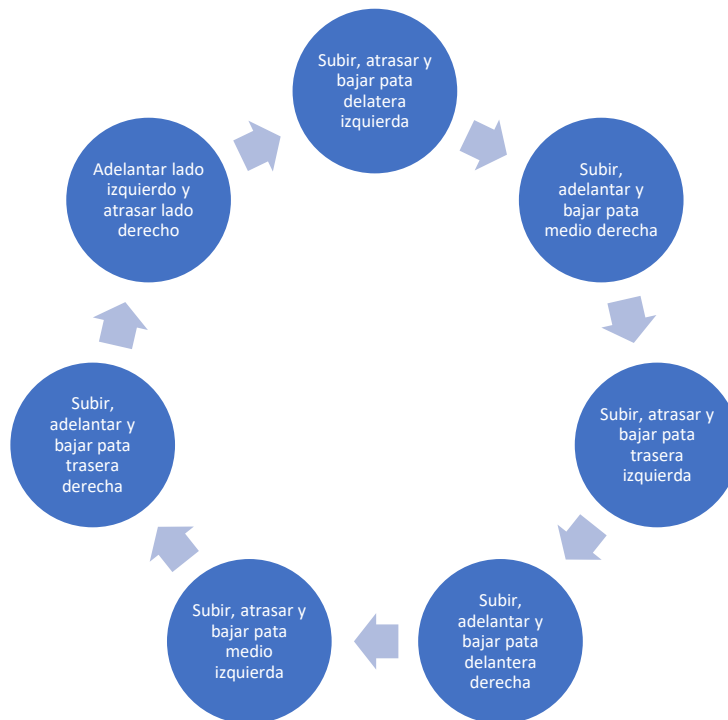


Figura 22: Ciclo realizado por las patas del robot hexápodo para girar a la izquierda.

Tanto el giro hacia la derecha como hacia la izquierda provocan una rotación sobre el propio robot, de manera que mientras gira hacia un lado éste no se desplaza ni hacia adelante ni hacia atrás, siendo esto una ventaja al utilizar el dispositivo en lugares con poco espacio.

7.1.2.4.- MOVIMIENTO HACIA ATRÁS

Para desplazar el hexápodo hacia atrás se utiliza el ciclo descrito en la Figura 23. Como se puede observar, dicho ciclo es opuesto al ciclo para desplazar el robot hacia adelante.

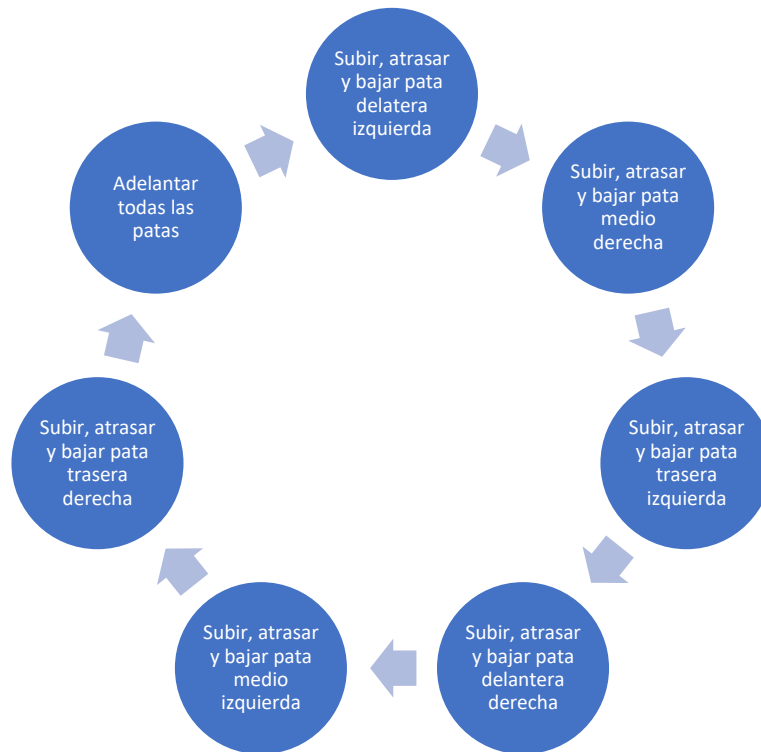


Figura 23: Ciclo realizado por las patas del robot hexápodo para desplazarse hacia atrás.

7.1.2.5.- CAMBIAR LA ALTURA DEL ROBOT

Para modificar la altura del robot, tan solo es necesario modificar los ángulos de todos los servomotores (Figura 24) situados en las articulaciones de muñeca y codo dejando inmóviles las articulaciones de los hombros de manera que, el robot logra pasar de una posición baja a una alta como se muestra en la Figura 25. Los ángulos de la muñeca pasan de los 45° a los 90° mientras que los servomotores del codo pasan de los 135° a los 70° . Por otro lado, si se desea que el robot baje su altura, se haría de la misma forma solo que de manera inversa.

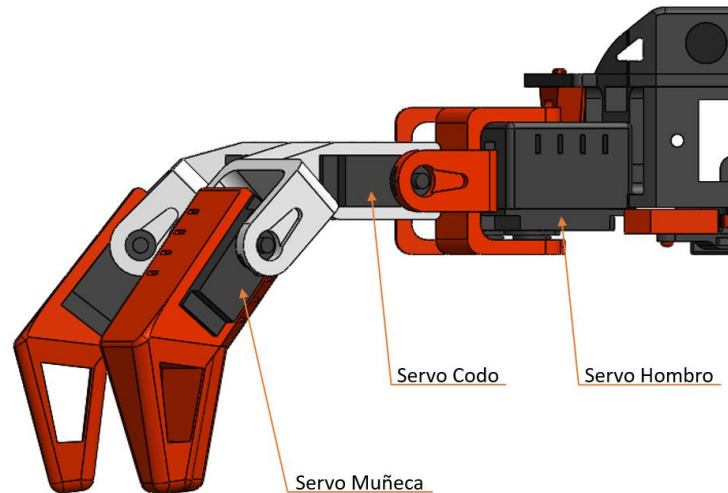


Figura 24: Nombre y situación de los servomotores de las patas del robot hexápodo.

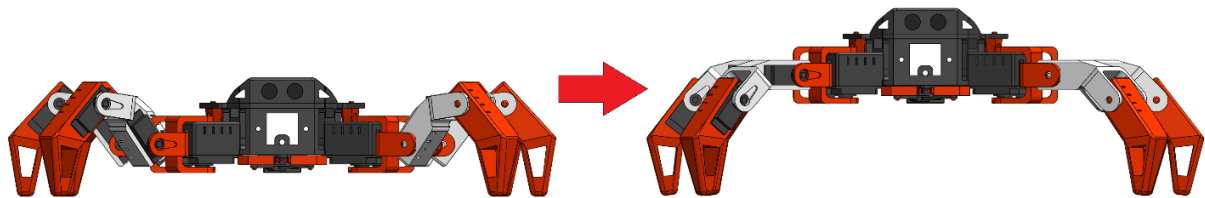


Figura 25: Diferentes alturas del cuerpo del robot hexápodo. A la izquierda posición baja y a la derecha posición alta.

7.1.2.6.- MOVIMIENTO PARA EL MODO PAQUETE

Otro movimiento con el que cuenta el robot es la capacidad de plegar sus patas para facilitar su transporte (Figura 26). Esto se consigue situando los servomotores de los hombros en los grados representados en la Tabla 16 y los servomotores de los codos y muñecas a 180° y 0° respectivamente:

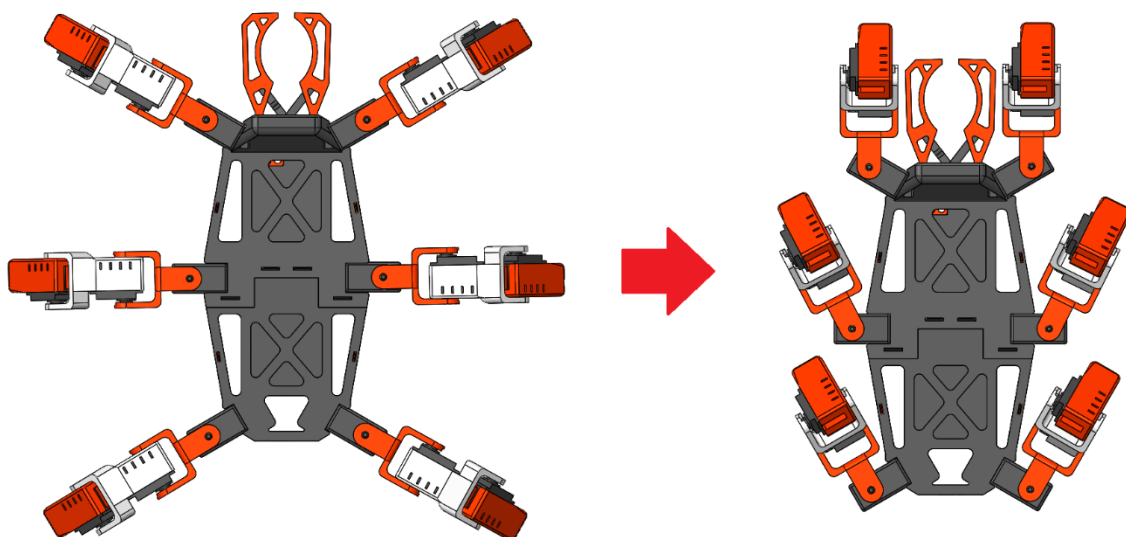


Figura 26: Comparación y posicionamiento de las patas plegadas del robot hexápodo. A la izquierda patas abiertas y a la derecha las patas plegadas.

Posicionamiento de los Servomotores para su Transporte	
Hombro pata delantera izquierda	150°
Hombro pata delantera derecha	30°
Hombro pata medio izquierda	130°
Hombro pata medio derecha	50°
Hombro pata trasera izquierda	180°
Hombro pata trasera derecha	0°

Tabla 16: Grados a los cuales se deben situar los servomotores de los hombros del robot hexápodo para plegar las patas.

7.1.2.7.- MOVIMIENTO DE ROTACIÓN HACIA LA DERECHA Y HACIA LA IZQUIERDA

Esto movimiento consiste en rotar hacia un lado (Figura 27 y Figura 28) el cuerpo del robot sin mover las patas de su posición inicial. Esto se consigue utilizando un bucle el cual va modificando automáticamente los grados de los hombros de todas las patas al mismo tiempo, llegando a la posición final y volviendo a la posición inicial de la cual partió.

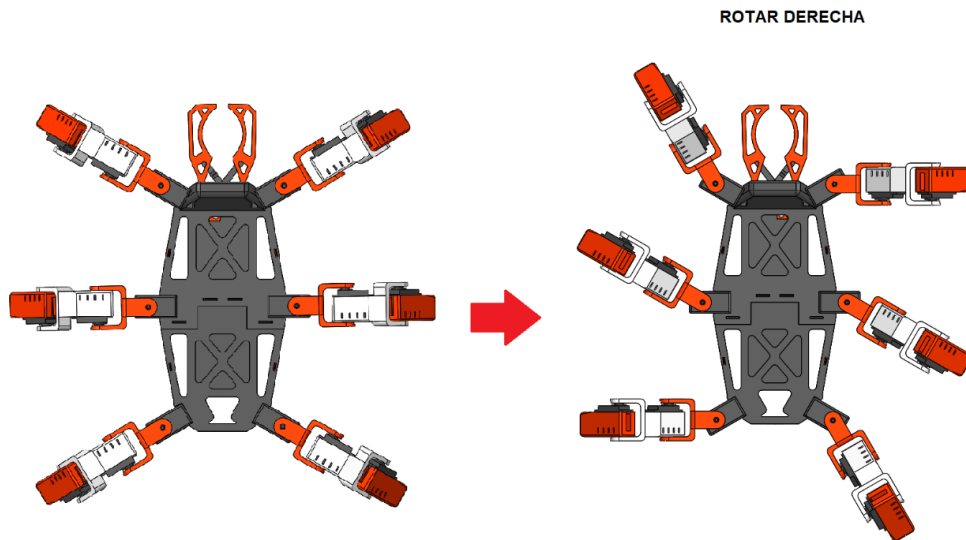


Figura 27: Rotación del robot hexápodo hacia la derecha.

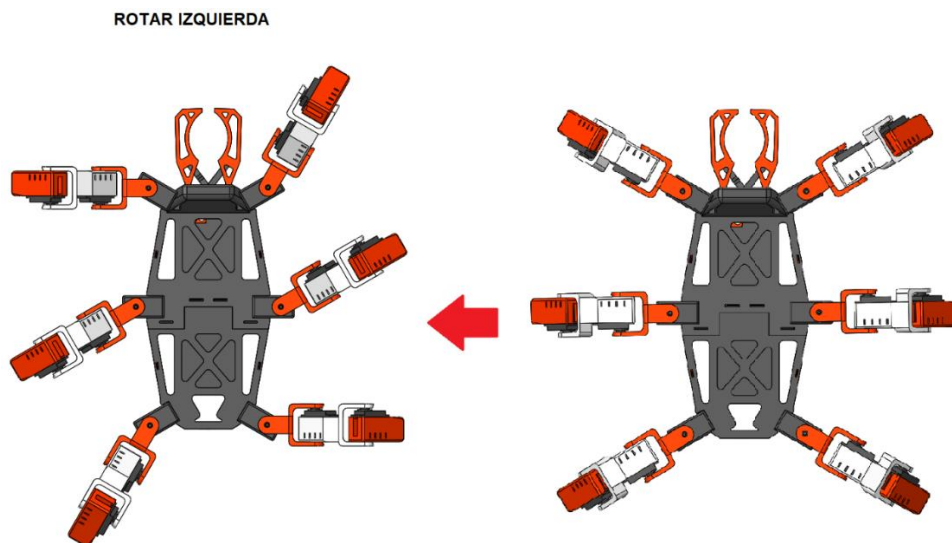


Figura 28: Rotación del robot hexápodo hacia la izquierda.

7.1.3.- PROGRAMACIÓN DEL MÓDULO DE ULTRASONIDOS Y DE LA MATRIZ LED

Para programar el módulo de ultrasonidos y la matriz led se ha creado una máquina de estados como se muestra en la Figura 29 la cual se repite tras cada medida si el modo patrulla del robot está activado.

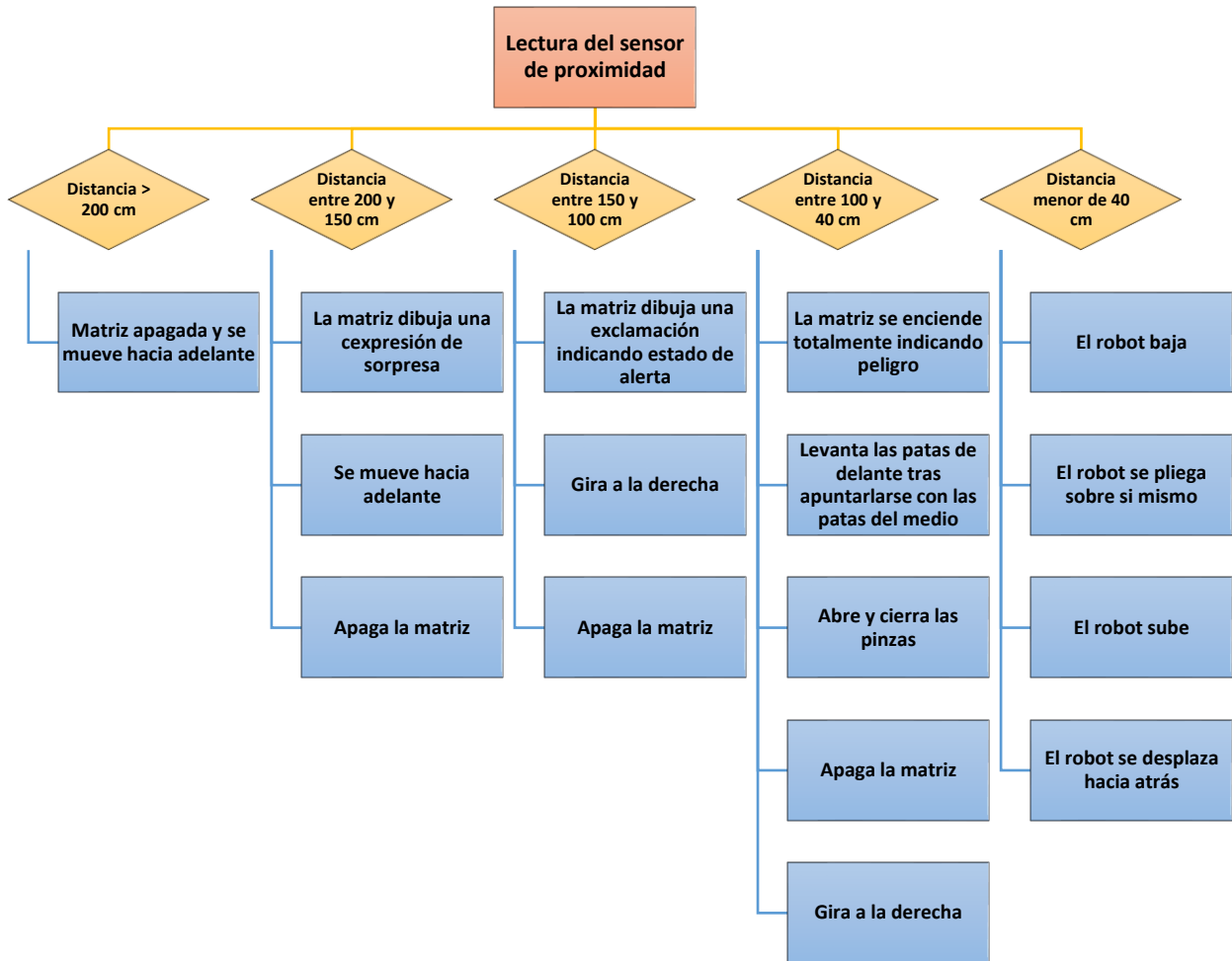


Figura 29: Máquina de estados utilizada para la programación del sensor de proximidad y la matriz led.

Por otro lado, los diferentes dibujos y efectos utilizados tanto en el modo patrulla como para saludar o simplemente dar la bienvenida al iniciar el robot se muestran en la Figura 30.

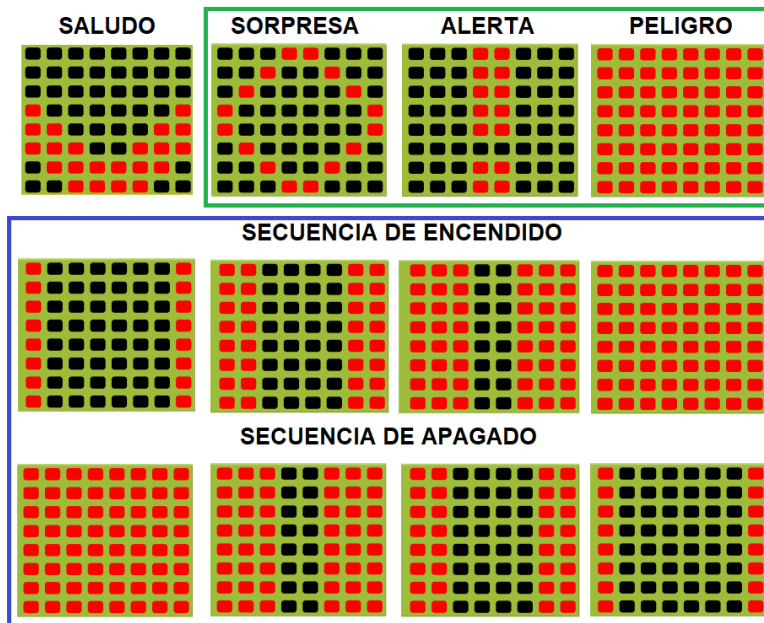


Figura 30: Diferentes configuraciones de la matriz led. Rodeado en color verde la configuración utilizada para el modo patrulla y en color azul la configuración utilizada para dar la bienvenida. [17]

7.2.- PROGRAMACIÓN DE LA APLICACIÓN MÓVIL Y USO DEL MÓDULO BLUETOOTH

Para la programación de la aplicación móvil se ha empleado AppInventor2 el cual permite programar dicha aplicación mediante bloques de una manera muy sencilla.

Si se desea ver toda la programación de la aplicación utilizada en este proyecto se recomienda ver el “Anexo A-2: Programación de la aplicación móvil” adjuntado en este documento, además de contar con el “Anexo A-1: Manual del Usuario y Mantenimiento” el cual da una visión más amplia de cómo funciona la aplicación móvil desde el punto de vista del usuario.

La aplicación móvil está programada para que cada acción que se desee que realice el robot envíe un carácter a través del módulo Bluetooth. En nuestro caso se han utilizado letras en minúsculas y alguna que otra en mayúscula como se aprecia en la Tabla 17.

Relación entre las Acciones y los Caracteres Enviados	
Adelante	w
Atrás	s
Izquierda	a
Derecha	d
Cerrar Pinzas	b
Abrir Pinzas	c
Encender Luces	e
Apagar Luces	f
Modo Patrulla	i
Quitar Modo Patrulla	j
Modo Paquete	k
Detener Modo Paquete	l
Ráfagas	m

DOCUMENTO NÚMERO 1: MEMORIA

Rotar Hacia la Derecha	p
Rotar Hacia la Izquierda	q
Parar	Q
Velocidad Lenta	r
Velocidad Media	t
Velocidad Rápida	x
Mantenimiento	y
Subir	n
Bajar	o
Saludar	M

Tabla 17: Relación entre las acciones y los caracteres enviados desde la aplicación móvil al módulo Bluetooth.

Cabe destacar que es importante no utilizar los caracteres que se encuentren en las siguientes frases: "+CONNECTING<<A8:9C:ED:2B:33:FC", "CONNECTED", "READY" o "DISCONNECTED". Esta serie de caracteres son enviados automáticamente por el módulo bluetooth al teléfono móvil o tableta para indicarle a éste su estado, de lo contrario, dichos caracteres se almacenarán en el "buffer" del módulo Bluetooth, realizando las acciones correspondientes con dicho carácter no finalizando hasta que estos hayan sido ejecutados y por tanto siendo el robot incontrolable hasta su finalización.

8.- BIBLIOGRAFÍA

- [1] P. por R. | | A.- Genuino, «▷ Arduino Nano Pinout sus Funciones, Características y Donde Comprar», *Descubrearduino.com*, dic. 26, 2018. <https://descubrearduino.com/arduino-nano-pinout/> (accedido jun. 10, 2020).
- [2] «Shield De Expansion Arduino Nano», *ZTRobotic*. <https://ztrobotic.com/product/shield-expansion-arduino-nano/> (accedido jun. 02, 2020).
- [3] «HC 06 Bluetooth module pinout, features & datasheet». <https://components101.com/wireless/hc-06-bluetooth-module-pinout-datasheet> (accedido jun. 02, 2020).
- [4] alldatasheet.com, «MAX7219 datasheet(2/16 Pages) MAXIM | Serially Interfaced, 8-Digit LED Display Drivers». <https://html.alldatasheet.com/html-pdf/227793/MAXIM/MAX7219/306/2/MAX7219.html> (accedido jun. 02, 2020).
- [5] «Modulo PCA9685 controlador servos 16 canales», *e-ika.com*. <https://www.e-ika.com/modulo-pca9685-controlador-servos-16-canales> (accedido jun. 10, 2020).
- [6] «Micro Servo SG90 1.8Kg/9g/0.12seg» *IBERBOTICS*, *IBERBOTICS*. <https://www.iberobotics.com/producto/micro-servo-towerpro-sg90-1-8kg9g0-12seg/> (accedido jun. 10, 2020).
- [7] «Servomotor MG996R», *Electronicos Caldas*. <https://www.electronicoscaldas.com/es/motores-y-servos/546-servo-motor-mg996r.html> (accedido jun. 10, 2020).
- [8] «Sensor ultrasonico HC-SR04», *Electronicos Caldas*. <https://www.electronicoscaldas.com/es/sonido-ultrasonido/324-sensor-modulo-ultrasonico-hc-sr04.html> (accedido jun. 10, 2020).
- [9] «HRB 7.4V 2200mAh 30C EC2 Lipo Batería para Juguetes de Avión Helicóptero RC Car Model Heli Plano Carro del Barco FPV Coches (7.4v 2200mah EC2): Amazon.es: Juguetes y juegos». https://www.amazon.es/gp/product/B07486HVH6/ref=ppx_yo_dt_b_asin_title_o04_s00?ie=UTF8&psc=1 (accedido jun. 10, 2020).
- [10] alldatasheet.com, «MUR460 datasheet(2/7 Pages) KERSEMI | Power Rectifiers». <https://html.alldatasheet.com/html-pdf/546119/KERSEMI/MUR460/592/2/MUR460.html> (accedido jun. 10, 2020).
- [11] «2310520.pdf». Accedido: jun. 10, 2020. [En línea]. Disponible en: <http://www.farnell.com/datasheets/2310520.pdf>.
- [12] «ender 3 - Búsqueda de Google». https://www.google.com/search?q=ender+3&sxsrf=ALeKk03NVs0BLXp112cCmiREMdca3AfCwA:1591822818025&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiw9_OMkvjAhWHnxQKHThHDK8Q_AUoA3oECAwQBQ&biw=958&bih=959 (accedido jun. 10, 2020).
- [13] «Guía para elegir el extrusor de mi impresora 3D», *Bitfab*, may 28, 2019. <https://bitfab.io/es/blog/extrusor-impresora-3d/> (accedido jun. 10, 2020).
- [14] «ARQUEADO», *Prusa3D - 3D Printers from Josef Průša*, ene. 04, 2019. <https://www.prusa3d.es/warping/> (accedido jun. 10, 2020).
- [15] «ABS 3D printing filament cracking - Rigidink», *MANUFACTUR3D*. <https://manufactur3dmag.com/3d-printing-filament-guide-abs-filament/abs-3d-printing-filament-cracking-rigidink/> (accedido jun. 10, 2020).
- [16] «pca9685 - Búsqueda de Google». https://www.google.com/search?q=pca9685&rlz=1C1GCEA_enES798ES798&sxsrf=ALeKk03WUsEX_L3TouPLO9_kMa9N7ZI7kQ:1591884207535&source=lnms&tbm=isch&sa=X&ved=2ahUKEWjmk9rl9vnpAhVr5OAKHX1zCroQ_AUoAXoECAwQAw&biw=958&bih=959#imgrc=tjRq3VhWUxPWvM (accedido jun. 11, 2020).
- [17] «Facilino». <https://roboticafacil.es/facilino/blockly/Facilino.html> (accedido jun. 11, 2020).

DOCUMENTO NÚMERO 1: MEMORIA

Hoja en blanco dejada a propósito



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

ANEXO A-1: MANUAL DEL USUARIO y MANTENIMIENTO

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

Álvaro Alberto Giner

TUTORIZADO POR

Leopoldo Armesto Ángel

CURSO ACADÉMICO: 2019/2020

Hoja en blanco dejada a propósito

INSTRUCCIONES

1.- DESCARGAR LA APLICACIÓN

En primer lugar, deseamos darle las gracias de todo corazón por haber adquirido nuestro producto.

Como ya sabrá el producto que ha adquirido funciona mediante una aplicación móvil. Puede descargarla escaneando el siguiente código QR o entrando en la siguiente página web. (Es necesario tener una versión de Android de su dispositivo inteligente superior a la 4.0) (NO DISPONIBLE PARA DISPOSITIVOS APPLE).



<http://ai2.appinventor.mit.edu/b/i60i>

Es posible que en algunos dispositivos inteligentes se le tenga que habilitar el “permitir la descarga de aplicaciones de terceros”.

2.- VINCULACIÓN DE DISPOSITIVOS

Una vez tenga descargada e instalada la aplicación deberá conectar el Bluetooth de su teléfono móvil o tableta.

Tras conectar el Bluetooth ya puede entrar en la aplicación pinchando en el siguiente icono (Figura 31) y con el nombre de “TFG de Alvaro”:



Figura 31: Icono de la aplicación.

Nada más entrar le aparecerá una pantalla de bienvenida y tras unos instantes le llevará automáticamente a la siguiente pantalla dónde le pedirá que vincule su dispositivo con el robot. Esto se consigue pinchando en el icono azul de la esquina superior izquierda (Figura 32).

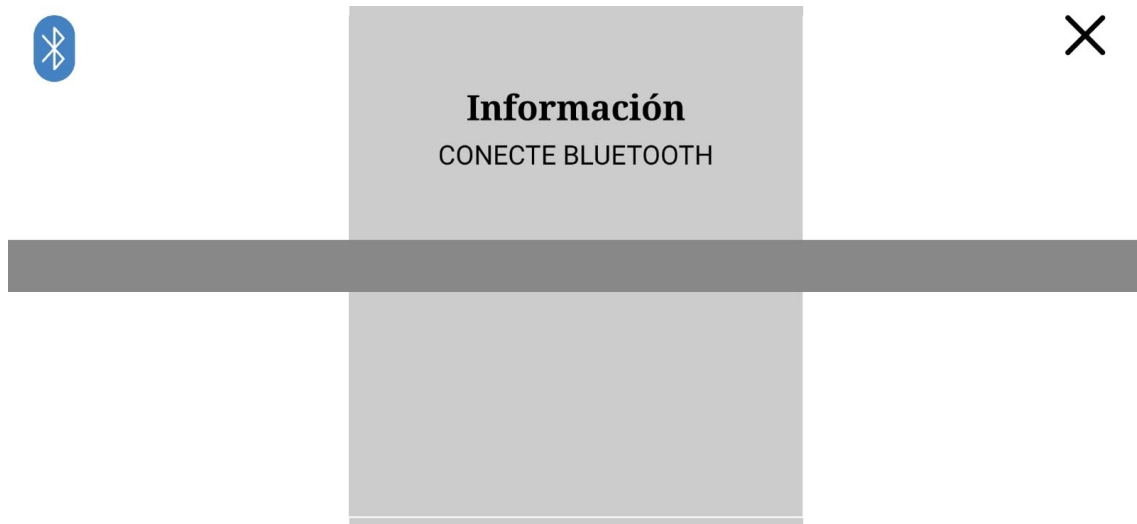


Figura 32: Pantalla para la vinculación de la conexión Bluetooth.

Una vez pinchado en el icono le aparecerá toda una lista de opciones con todos los dispositivos Bluetooth vinculados y guardados en la memoria del teléfono móvil o tableta.

Debe pinchar sobre el dispositivo con el nombre de “RobotAlvaro”. En caso de que dicho dispositivo no le apareciera en la lista, deberá salir de la aplicación e ir a los ajustes del Bluetooth de su dispositivo inteligente y vincularlo.

Deberá introducir “5430” como contraseña. Una vez realizados los pasos anteriores debería aparecerle el dispositivo en la lista (Figura 33).

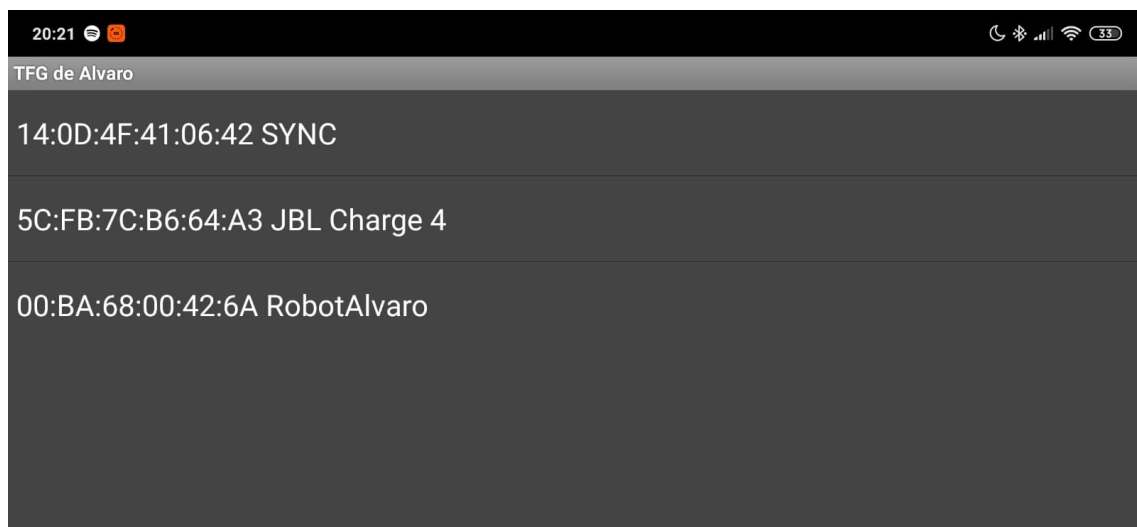


Figura 33: Pantalla para la elección del dispositivo a vincular.

3.- CONTROLES Y FUNCIONALIDADES

Tras vincular el robot con el teléfono móvil o tableta le aparecerá la pantalla siguiente (Figura 34):

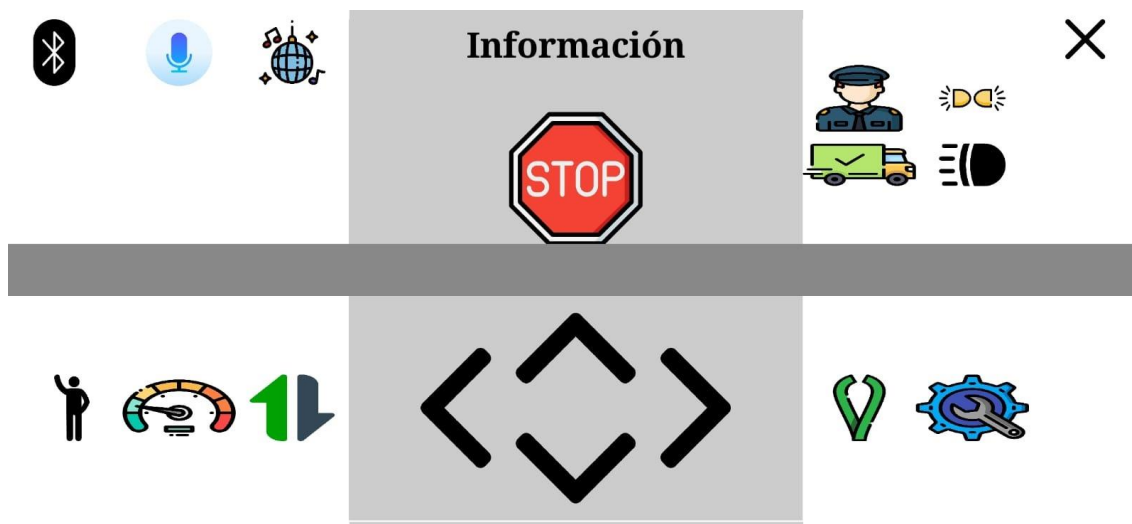













Figura 34: Pantalla principal de la aplicación.

En dicha pantalla podrá controlar el robot y salir de la aplicación.

A continuación, se pueden ver todos los botones, sus nombres, comandos de voz y funcionalidades de cada uno (Tabla 18).

ICONO	NOMBRE	COMANDO DE VOZ	FUNCIONALIDAD
	VINCULAR BLUETOOTH	----	Vincula la conexión Bluetooth entre su teléfono móvil o tableta y el propio robot.
	DESVINCULAR BLUETOOTH	----	Desvincula la conexión Bluetooth entre su teléfono móvil o tableta y el propio robot.
	COMANDOS DE VOZ	----	Permite dar órdenes al robot a través de la voz del usuario.
	HABILITAR MODO FLEXIBILIDAD	----	Habilita los botones de rotar a la derecha y el de rotar a la izquierda.
	DESHABILITAR MODO FLEXIBILIDAD	----	Deshabilita los botones de rotar a la derecha y el de rotar a la izquierda. Aparece tras pulsar el botón de Habilitar Modo Flexibilidad.

	ROTAR A DERECHAS	Rotar derecha	Permite rotar el robot hacia la derecha a baja altura.
	ROTAR A IZQUIERDAS	Rotar izquierda	Permite rotar el robot hacia la izquierda a baja altura.
	PARAR	Parar	Permite poner el robot en una posición neutral, situando el robot en una altura elevada y las patas en la posición de seguridad.
	MODO PATRULLA ACTIVADO	Activar modo patrulla	Permite poner el robot en modo automático. Éste patrullará automáticamente haciendo uso de sus sensores.
	MODO PATRULLA DESACTIVADO	Detener modo patrulla	Permite salir del modo patrulla, deteniendo el robot en la posición neutral. Este botón se habilita tras activar el modo patrulla.
	MODO TRANSPORTE ACTIVADO	Activar modo paquete	Permite plegar las patas para facilitar su transporte.
	MODO TRANSPORTE DESACTIVADO	Detener modo paquete	Permite salir del modo transporte, dejando el robot en su posición neutral a una altura baja.
	ENCENDER LUCES	Encender luces	Permite encender las luces delanteras del robot.
	APAGAR LUCES	Apagar luces	Permite apagar las luces delanteras del robot. Este botón se habilita tras pulsar el botón de encender las luces.
	BOTÓN SALIR	----	Permite salir de la aplicación. También desvincula automáticamente el bluetooth.
	SALUDAR	Saludar	Permite hacer un saludo. El robot levantará una de sus patas y la moverá de un lado a otro con una sonrisa dibujada en la matriz led.

	VELOCIDAD LENTA	Velocidad lenta	Permite cambiar la velocidad de los movimientos a la velocidad más lenta del robot. (POR DEFECTO)
	VELOCIDAD MEDIA	Velocidad media	Permite cambiar la velocidad de los movimientos a la velocidad media del robot
	VELOCIDAD RÁPIDA	Velocidad rápida	Permite cambiar la velocidad de los movimientos a la velocidad más rápida del robot. (RECOMENDABLE)
	SUBIR	Subir	Permite subir la altura del cuerpo del robot.
	BAJAR	Bajar	Permite bajar la altura del cuerpo del robot.
	GIRAR A LA IZQUIERDA	Girar izquierda	Permite girar a la izquierda.
	ADELANTE	Adelante	Permite andar hacia adelante.
	ATRÁS	Atrás	Permite andar hacia atrás.
	GIRAR A LA DERECHA	Girar derecha	Permite girar a la derecha.
	ABRIR PINZAS	Abrir pinzas	Permite abrir las pinzas. Tras pulsar el botón se quedan abiertas.
	CERRAR PINZAS	Cerrar pinzas	Permite cerrar las pinzas. Tras pulsar el botón se quedan cerradas. Este botón se habilita tras pulsar el botón de Abrir Pinzas.
	MANTENIMIENTO	Mantenimiento	Permite situar las pinzas de manera que éstas puedan ser sustituidas por unas nuevas u otras herramientas.

Tabla 18: Iconos, nombre de los botones, comandos de voz y su funcionalidad.

4.- CARGA DE LA BATERÍA

Este robot contiene una batería de tipo Li-Po con una capacidad de 2200 mAh.

Es importante que se sigan las instrucciones para la carga de la batería que se explican en este documento o por lo contrario puede dañar la batería o incluso crear un incendio.

Junto con el robot le debe venir el cargador. Los pasos para cargar la batería son los siguientes:

1^{er} PASO: Conecte el cargador a una toma de corriente.

2^o PASO: Seleccione el tipo de batería, en nuestro caso se trata de una batería tipo Li-Po.

3^{er} PASO: Seleccione el número de celdas de la batería, en este caso se trata de una batería 2S (7,4 V).

4^o PASO: Seleccione el modo de carga deseado, en este caso se debe seleccionar BALANCE CHARGE.

5^o PASO: Ajuste la corriente a la cual debe cargarse la batería. Recomendamos 1.6 A.

6^o PASO: Conecte la batería al cargador, ambos conectores, tanto el conector blanco como el amarillo. Solo hay una posición.

7^o PASO: Mantenga el botón START hasta que el cargador haga un pitido. Una vez hecho el pitido el cargador empezará a cargar la batería. Tras cargar la batería éste volverá a realizar una serie de pitidos indicando el fin de la carga.

8^o PASO: Desconectar la batería del cargador y desconectar el cargador de la toma de corriente.

5.- MANTENIMIENTO

Junto con el dispositivo se le adjunta un kit de recambios. Dicho kit contiene piezas para sustituir una pata al completo, así como piezas para la sustitución de las pinzas.

En caso de rotura de algún servomotor u otro componente electrónico será necesario sustituir dicho componente dañado por uno nuevo de características similares.

Cabe destacar que con la compra del dispositivo se adjuntan en la tarjeta de memoria los archivos .stl necesarios para poder imprimir las piezas de recambio que así fueran necesarias. Dicha tarjeta de memoria debe venir con el producto.

Nota: Si no se va a utilizar el dispositivo durante un largo período de tiempo, se recomienda cargar la batería en modo STORAGE.

ANEXO A-1: MANUAL DEL USUARIO

Hoja en blanco dejada a propósito



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

**ANEXO A-2:
PROGRAMACIÓN DE LA
APLICACIÓN MÓVIL**

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

Álvaro Alberto Giner

TUTORIZADO POR

Leopoldo Armesto Ángel

CURSO ACADÉMICO: 2019/2020

Hoja en blanco dejada a propósito

ANEXO A-2: PROGRAMACIÓN DE LA APLICACIÓN MÓVIL

```

when Conectar_Bluetooth .BeforePicking
do
  if Bluetooth .Available
  then set Conectar_Bluetooth .Elements to Bluetooth .AddressesAndNames

```

```

when Velocidad_Lenta .Click
do
  set Label1 .Text to "VELOCIDAD LENTA"
  set Velocidad_Lenta .Visible to false
  set Velocidad_Media .Visible to true
  set Velocidad_Alta .Visible to false
  call Bluetooth .SendText
  text "r"

```

```

when Velocidad_Media .Click
do
  set Label1 .Text to "VELOCIDAD MEDIA"
  set Velocidad_Lenta .Visible to false
  set Velocidad_Media .Visible to false
  set Velocidad_Alta .Visible to true
  call Bluetooth .SendText
  text "t"

```

```

when Velocidad_Alta .Click
do
  set Label1 .Text to "VELOCIDAD ALTA"
  set Velocidad_Lenta .Visible to true
  set Velocidad_Media .Visible to false
  set Velocidad_Alta .Visible to false
  call Bluetooth .SendText
  text "x"

```

```

when Subir_altura_base .Click
do
  set Label1 .Text to "SUBIENDO"
  set Subir_altura_base .Visible to false
  set Bajar_Altura_Base .Visible to true
  call Bluetooth .SendText
  text "n"

```

```

when Bajar_Altura_Base .Click
do
  set Label1 .Text to "BAJANDO"
  set Subir_altura_base .Visible to true
  set Bajar_Altura_Base .Visible to false
  call Bluetooth .SendText
  text "o"

```

```

when Abrir_Pinza .Click
do
  set Label1 .Text to "ABRIENDO PINZAS"
  set Abrir_Pinza .Visible to false
  set Cerrar_Pinza .Visible to true
  call Bluetooth .SendText
  text "c"

```

```

when Cerrar_Pinza .Click
do
  set Label1 .Text to "CERRANDO PINZAS"
  set Abrir_Pinza .Visible to true
  set Cerrar_Pinza .Visible to false
  call Bluetooth .SendText
  text "b"

```

```

when parar .Click
do
  set Label1 .Text to "PARADO"
  set Bajar_Altura_Base .Visible to true
  set Subir_altura_base .Visible to false
  call Bluetooth .SendText
  text "q"

```

```

when Desconectar_Bluetooth .Click
do
  call Bluetooth .Disconnect
  set Desconectar_Bluetooth .Visible to false
  set Conectar_Bluetooth .Visible to true

```

```

when Rafaga .Click
do
  if Luces_ON .Visible = false
  then
    set Label1 .Text to "RAFAGAS"
    call Bluetooth .SendText
    text "m"
    call Bluetooth .SendText
    text "e"
  else
    set Label1 .Text to "RAFAGAS"
    call Bluetooth .SendText
    text "m"

```

```

when Luces_ON .Click
do
  set Label1 .Text to "LUCES ON"
  set Luces_ON .Visible to false
  set Luces_OFF .Visible to true
  call Bluetooth .SendText
  text "e"

```

```

when saludar .Click
do
  set Label1 .Text to "SALUDANDO"
  call Bluetooth .SendText
  text "M"

```

```

when ADELANTE .Click
do
  set Label1 .Text to "MOVIENDO ADELANTE"
  call Bluetooth .SendText
  text "w"

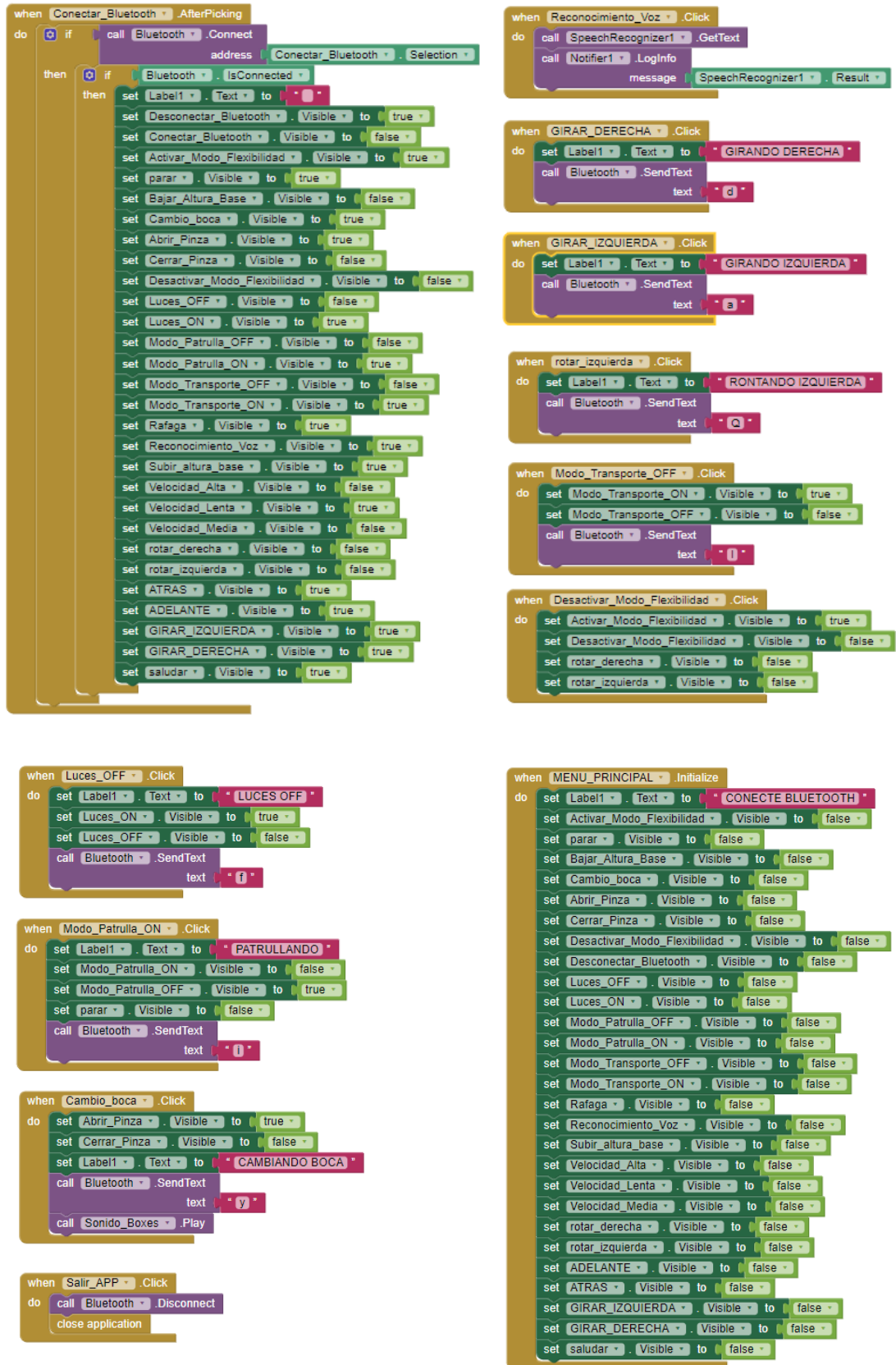
```

```

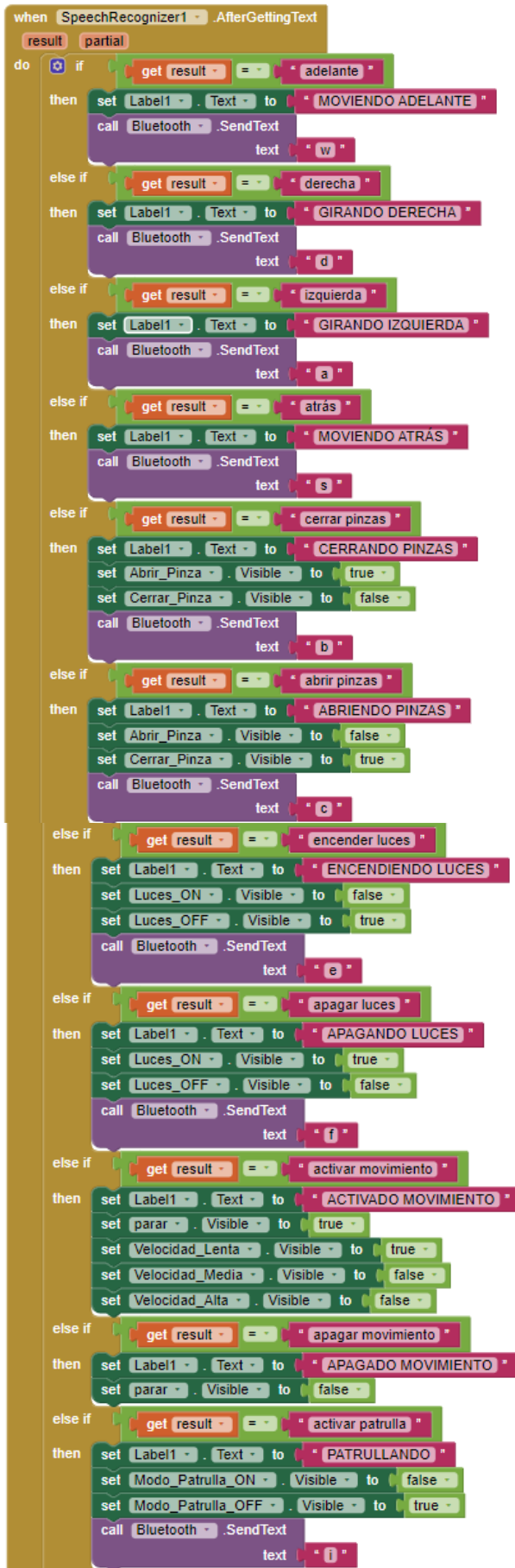
when ATRAS .Click
do
  set Label1 .Text to "MOVIENDO ATRÁS"
  call Bluetooth .SendText
  text "s"

```

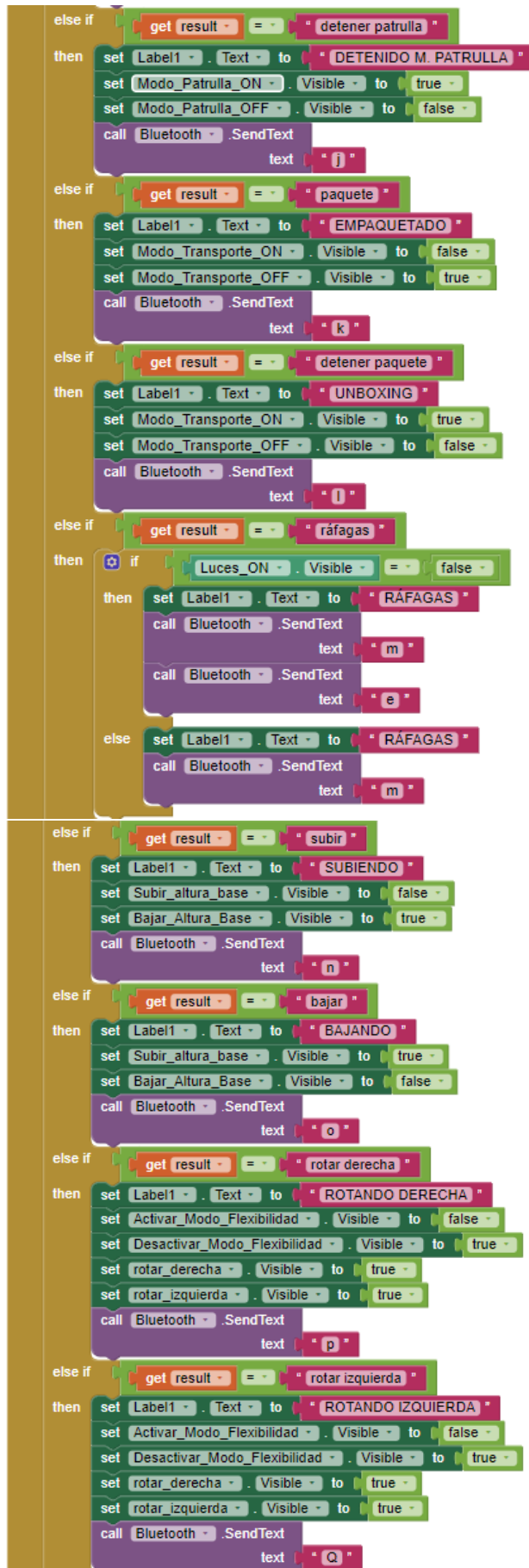
ANEXO A-2: PROGRAMACIÓN DE LA APLICACIÓN MÓVIL



ANEXO A-2: PROGRAMACIÓN DE LA APLICACIÓN MÓVIL



ANEXO A-2: PROGRAMACIÓN DE LA APLICACIÓN MÓVIL



ANEXO A-2: PROGRAMACIÓN DE LA APLICACIÓN MÓVIL

```
else if [get result] = "velocidad media"
then
  set [Label1] . Text to "VELOCIDAD MEDIA"
  set [Velocidad_Lenta] . Visible to false
  set [Velocidad_Media] . Visible to false
  set [Velocidad_Alta] . Visible to true
  call [Bluetooth] . SendText
  text "i"
else if [get result] = "velocidad rápida"
then
  set [Label1] . Text to "VELOCIDAD RÁPIDA"
  set [Velocidad_Lenta] . Visible to true
  set [Velocidad_Media] . Visible to false
  set [Velocidad_Alta] . Visible to false
  call [Bluetooth] . SendText
  text "x"
else if [get result] = "calibrar"
then
  set [Label1] . Text to "CALIBRANDO BOCA"
  call [Bluetooth] . SendText
  text "y"
  call [Sonido_Boxes] . Play
else if [get result] = "parado"
then
  set [Label1] . Text to "PARADO"
  set [Subir_altura_base] . Visible to false
  set [Bajar_Altura_Base] . Visible to true
  call [Bluetooth] . SendText
  text "q"
else if [get result] = "saludar"
then
  set [Label1] . Text to "SALUDANDO"
  call [Bluetooth] . SendText
  text "M"
else
  set [Label1] . Text to "REPITA PORFAVOR"
```

```
when [Modo_Transporte_ON] . Click
do
  set [Modo_Transporte_ON] . Visible to false
  set [Modo_Transporte_OFF] . Visible to true
  call [Bluetooth] . SendText
  text "X"
```

```
when [Activar_Modo_Flexibilidad] . Click
do
  set [Activar_Modo_Flexibilidad] . Visible to false
  set [Desactivar_Modo_Flexibilidad] . Visible to true
  set [rotar_derecha] . Visible to true
  set [rotar_izquierda] . Visible to true
```

```
when [rotar_derecha] . Click
do
  set [Label1] . Text to "ROTANDO DERECHA"
  call [Bluetooth] . SendText
  text "p"
```

```
when [Modo_Patrulla_OFF] . Click
do
  set [Label1] . Text to "PATRULLA DETENIDA"
  set [Modo_Patrulla_ON] . Visible to true
  set [Modo_Patrulla_OFF] . Visible to false
  set [parar] . Visible to true
  call [Bluetooth] . SendText
  text "f"
```

Hoja en blanco dejada a propósito



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

**ANEXO A-3:
PROGRAMACIÓN DEL
ROBOT. CÓDIGO ARDUINO**

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

Álvaro Alberto Giner

TUTORIZADO POR

Leopoldo Armesto Ángel

CURSO ACADÉMICO: 2019/2020

Hoja en blanco dejada a propósito

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
//-----LIBRERÍAS-----
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <Servo.h>
#include <SoftwareSerial.h>

//-----DEFINICIÓN DE LAS DIRECCIONES DE LOS CONTROLADORES PCA-----
Adafruit_PWMServoDriver pwm1 = Adafruit_PWMServoDriver(0x40); //para el control del
                                                                controlador PCA de las
                                                                patas derechas

Adafruit_PWMServoDriver pwm2 = Adafruit_PWMServoDriver(0x41); //para el control del
                                                                controlador PCA de las
                                                                patas izquierdas

//-----CONFIGURACIÓN DE LOS PINES DEL MÓDULO BLUETOOTH-----
SoftwareSerial Bluetooth((4), (5)); // el RX del bluetooth va al 5 (TX de la placa) y el
                                     TX del módulo al 4 (RX de la placa)
                                     CONTRASEÑA: 5430 NOMBRE: RobotAlvaro

//-----DEFINICIÓN DE LAS CONSTANTES DEL PROGRAMA-----
Servo servoBoca;
int t = 750; //tiempo de los delay's para andar
int i = 0;
int j = 0;
int vez = 0;
char ordenes;
int estadomodopatrulla = 0;

//-----DEFINICIÓN DE LOS PINES DEL SENSOR DE ULTRASONIDOS-----
const int trigPin = 11;
const int echoPin = 12;

//-----DEFINICIÓN DE LAS VARIABLES DEL SENSOR DE ULTRASONIDOS-----
long duration;
int distance;

//-----FUNCIONES-----
//función para inicializar los canales PWM de los controladores PCA
void beginServos()
{
  pwm1.begin();
  pwm2.begin();
  pwm1.setPWMPFreq(Frecuencia); // define la frecuencia de trabajo de los servos
  pwm2.setPWMPFreq(Frecuencia);
  servoBoca.attach(2); //Pin para el servomotor de las pinzas
}
//funciones para generar los pulsos para los controladores PCA
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
void writeServoD( int nServo, int posicion)    //para el lado derecho del robot
{
  int pos = map (posicion, 0, 180, anguloMinimo, anguloMaximo);
  pwm1.setPWM(nServo, 0, pos);
}
void writeServoI( int nServo, int posicion)    //para el lado izquierdo del robot
{
  int pos = map (posicion, 0, 180, anguloMinimo, anguloMaximo);
  pwm2.setPWM(nServo, 0, pos);
}

//Funciones para el control de la matriz led
void LEDMatrix_init(int cs, int din, int clk);
void writeRow(int cs, int din, int clk, int row, int data);
void maxAll (int cs, int din, int clk, int reg, int col);
void putByte (int din, int clk, int data);
void expression(int cs, int din, int clk, int col1, int col2, int col3, int col4, int col5, int col6, int col7,
int col8);

void LEDMatrix_init(int cs, int din, int clk) {
  maxAll(cs, din, clk, 11, 7);
  maxAll(cs, din, clk, 9, 0);
  maxAll(cs, din, clk, 12, 1);
  maxAll(cs, din, clk, 15, 0);
  int i = 0;
  for (i = 1; i <= 8; i++)
  {
    maxAll(cs, din, clk, i, 0);
  }
  maxAll(cs, din, clk, 10, 15);
}

void writeRow(int cs, int din, int clk, int row, int data)
{
  digitalWrite(cs, LOW);
  putByte(din, clk, row);
  putByte(din, clk, data);
  digitalWrite(cs, LOW);
  digitalWrite(cs, HIGH);
}

void maxAll (int cs, int din, int clk, int reg, int col)
{
  digitalWrite(cs, LOW);
  putByte(din, clk, reg);
  putByte(din, clk, col);
  digitalWrite(cs, LOW);
  digitalWrite(cs, HIGH);
}
```

```
}

```

```
void putByte (int din, int clk, int data)

```

```
{
  byte i = 8;
  byte mask;
  while (i > 0)
  {
    mask = 0x01 << (i - 1);
    digitalWrite(clk, LOW);
    if (data & mask)
    {
      digitalWrite(din, HIGH);
    }
    else
    {
      digitalWrite(din, LOW);
    }
    digitalWrite(clk, HIGH);
    --i;
  }
}
```

```
void expression(int cs, int din, int clk, int col1, int col2, int col3, int col4, int col5, int col6, int col7,
int col8)

```

```
{
  writeRow(cs, din, clk, 1, col1);
  writeRow(cs, din, clk, 2, col2);
  writeRow(cs, din, clk, 3, col3);
  writeRow(cs, din, clk, 4, col4);
  writeRow(cs, din, clk, 5, col5);
  writeRow(cs, din, clk, 6, col6);
  writeRow(cs, din, clk, 7, col7);
  writeRow(cs, din, clk, 8, col8);
}
```

```
//-----FUNCIONES DEL ROBOT-----
```

```
void abajo()

```

```
{
  //PATA IZQUIERDA DELANTE
  //Hombro
  writeServo(15, 90);
  delay(25);
  //Codo
  writeServo(14, 135);
  delay(25);
  //Muñeca
}
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
writeServo(13, 45);  
delay(25);  
  
//PATA DERECHA DELANTE  
//Hombro  
writeServoD(0, 90);  
delay(25);  
//Codo  
writeServoD(1, 135);  
delay(25);  
//Muñeca  
writeServoD(2, 45);  
delay(25);  
  
//PATA IZQUIERDA MEDIO  
//Hombro  
writeServo(11, 90);  
delay(25);  
//Codo  
writeServo(10, 135);  
delay(25);  
//Muñeca  
writeServo(9, 45);  
delay(25);  
  
//PATA DERECHA MEDIO  
//Hombro  
writeServoD(4, 90);  
delay(25);  
//Codo  
writeServoD(5, 135);  
delay(25);  
//Muñeca  
writeServoD(6, 45);  
delay(25);  
  
//PATA IZQUIERDA DETRÁS  
//Hombro  
writeServo(2, 90);  
delay(25);  
//Codo  
writeServo(1, 135);  
delay(25);  
//Muñeca  
writeServo(0, 45);  
delay(25);  
  
//PATA DERECHA DETRÁS
```

```
//Hombro
writeServoD(13, 90);
delay(25);
//Codo
writeServoD(14, 135);
delay(25);
//Muñeca
writeServoD(15, 45);
delay(25);
}

//-----

void arriba()
{
  //PATA IZQUIERDA DELANTE
  //Hombro
  writeServo(15, 90);
  delay(25);
  //Codo
  writeServo(14, 70);
  delay(25);
  //Muñeca
  writeServo(13, 90);
  delay(25);

  //PATA DERECHA DELANTE
  //Hombro
  writeServoD(0, 90);
  delay(25);
  //Codo
  writeServoD(1, 70);
  delay(25);
  //Muñeca
  writeServoD(2, 90);
  delay(25);

  //PATA IZQUIERDA MEDIO
  //Hombro
  writeServo(11, 90);
  delay(25);
  //Codo
  writeServo(10, 70);
  delay(25);
  //Muñeca
  writeServo(9, 90);
  delay(25);
}
```

```
//PATA DERECHA MEDIO
//Hombro
writeServoD(4, 90);
delay(25);
//Codo
writeServoD(5, 70);
delay(25);
//Muñeca
writeServoD(6, 90);
delay(25);

//PATA IZQUIERDA DETRÁS
//Hombro
writeServoI(2, 90);
delay(25);
//Codo
writeServoI(1, 70);
delay(25);
//Muñeca
writeServoI(0, 90);
delay(25);

//PATA DERECHA DETRÁS
//Hombro
writeServoD(13, 90);
delay(25);
//Codo
writeServoD(14, 70);
delay(25);
//Muñeca
writeServoD(15, 90);
delay(25);
}

//-----
void abrir_pinzas()
{
  servoBoca.write(75);
  delay(25);
}

//-----
void cerrar_pinzas()
{
  servoBoca.write(0);
  delay(25);
}
```



```
//-----  
void rotar_izquierda()  
{  
  //PATA IZQUIERDA DELANTE  
  //Codo  
  writeServo(14, 135);  
  delay(25);  
  //Muñeca  
  writeServo(13, 45);  
  delay(25);  
  
  //PATA DERECHA DELANTE  
  //Codo  
  writeServoD(1, 135);  
  delay(25);  
  //Muñeca  
  writeServoD(2, 45);  
  delay(25);  
  
  //PATA IZQUIERDA MEDIO  
  //Codo  
  writeServo(10, 135);  
  delay(25);  
  //Muñeca  
  writeServo(9, 45);  
  delay(25);  
  
  //PATA DERECHA MEDIO  
  //Codo  
  writeServoD(5, 135);  
  delay(25);  
  //Muñeca  
  writeServoD(6, 45);  
  delay(25);  
  
  //PATA IZQUIERDA DETRÁS  
  //Codo  
  writeServo(1, 135);  
  delay(25);  
  //Muñeca  
  writeServo(0, 45);  
  delay(25);  
  
  //PATA DERECHA DETRÁS  
  //Codo  
  writeServoD(14, 135);  
  delay(25);  
  //Muñeca
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```

writeServoD(15, 45);
delay(25);
j = 0;
for (i = 0; i <= 15; i++)
{
  j = 2 * i;
  writeServoI(15, 90 - j); //izquierda delante
  writeServoD(0, 90 - j); //derecha delante
  delay(25);
  writeServoI(11, 90 - j); //izquierda medio
  writeServoD(4, 90 - j); //derecha medio
  delay(25);
  writeServoI(2, 90 - j); //izquierda atrás
  writeServoD(13, 90 - j); //derecha atrás
  delay(25);
}
j = 0;
for (i = 0; i <= 15; i++)
{
  j = 2 * i;
  writeServoI(15, 60 + j); //izquierda delante
  writeServoD(0, 60 + j); //derecha delante
  delay(25);
  writeServoI(11, 60 + j); //izquierda medio
  writeServoD(4, 60 + j); //derecha medio
  delay(25);
  writeServoI(2, 60 + j); //izquierda atrás
  writeServoD(13, 60 + j); //derecha atrás
  delay(25);
}
}

//-----
void rotar_derecha()
{
  //PATA IZQUIERDA DELANTE
  //Codo
  writeServoI(14, 135);
  delay(25);
  //Muñeca
  writeServoI(13, 45);
  delay(25);

  //PATA DERECHA DELANTE
  //Codo
  writeServoD(1, 135);
  delay(25);
  //Muñeca

```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
writeServoD(2, 45);
delay(25);

//PATA IZQUIERDA MEDIO
//Codo
writeServo(10, 135);
delay(25);
//Muñeca
writeServo(9, 45);
delay(25);

//PATA DERECHA MEDIO
//Codo
writeServoD(5, 135);
delay(25);
//Muñeca
writeServoD(6, 45);
delay(25);

//PATA IZQUIERDA DETRÁS
//Codo
writeServo(1, 135);
delay(25);
//Muñeca
writeServo(0, 45);
delay(25);

//PATA DERECHA DETRÁS
//Codo
writeServoD(14, 135);
delay(25);
//Muñeca
writeServoD(15, 45);
delay(25);
j = 0;
for (i = 0; i < 15; i++)
{
  j = 2 * i;
  writeServo(15, 90 + j); //izquierda delante
  writeServoD(0, 90 + j); //derecha delante
  delay(25);
  writeServo(11, 90 + j); //izquierda medio
  writeServoD(4, 90 + j); //derecha medio
  delay(25);
  writeServo(2, 90 + j); //izquierda atrás
  writeServoD(13, 90 + j); //derecha atrás
  delay(25);
}
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
j = 0;
for (i = 0; i < 15; i++)
{
  j = 2 * i;
  writeServo(15, 120 - j); //izquierda delante
  writeServoD(0, 120 - j); //derecha delante
  delay(25);
  writeServo(11, 120 - j); //izquierda medio
  writeServoD(4, 120 - j); //derecha medio
  delay(25);
  writeServo(2, 120 - j); //izquierda atrás
  writeServoD(13, 120 - j); //derecha atrás
  delay(25);
}
}

//-----
void adelante()
{
  //PASO 1-----
  //-----SUBIMOS Y ADELANTAMOS PATA ID
  //PATA IZQUIERDA DELANTE
  //Hombro
  writeServo(15, 120);
  //Codo
  writeServo(14, 135);
  //Muñeca
  writeServo(13, 45);
  delay(25);
  delay(t);

  //-----BAJAMOS LA PATA ID ADELANTADA
  //PATA IZQUIERDA DELANTE
  //Hombro
  writeServo(15, 120);
  delay(25);
  //Codo
  writeServo(14, 70);
  delay(25);
  //Muñeca
  writeServo(13, 90);
  delay(25);
  delay(t);

  //PASO 2-----
  //-----SUBIMOS Y ADELANTAMOS PATA DM
  //PATA DERECHA MEDIO
  //Hombro
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
writeServoD(4, 60);  
//Codo  
writeServoD(5, 135);  
//Muñeca  
writeServoD(6, 45);  
delay(25);  
delay(t);  
  
//-----BAJAMOS DM ADELANTADA  
//PATA DERECHA MEDIO  
//Hombro  
writeServoD(4, 60);  
delay(25);  
//Codo  
writeServoD(5, 70);  
delay(25);  
//Muñeca  
writeServoD(6, 90);  
delay(25);  
delay(t);  
  
//PASO 3 -----  
//-----SUBIMOS Y ADELANTAMOS IA  
//PATA IZQUIERDA DETRÁS  
//Hombro  
writeServoI(2, 120);  
//Codo  
writeServoI(1, 135);  
//Muñeca  
writeServoI(0, 45);  
delay(25);  
delay(t);  
  
//-----BAJAMOS IA ADELANTADA  
//PATA IZQUIERDA DETRÁS  
//Hombro  
writeServoI(2, 120);  
delay(25);  
//Codo  
writeServoI(1, 70);  
delay(25);  
//Muñeca  
writeServoI(0, 90);  
delay(25);  
delay(t);  
  
//PASO 4 -----SUBIMOS Y ADELANTAMOS DD  
//PATA DERECHA DELANTE
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
//Hombro
writeServoD(0, 60);
//Codo
writeServoD(1, 135);
//Muñeca
writeServoD(2, 45);
delay(25);
delay(t);

//-----BAJAMOS DD ADELANTADA
//PATA DERECHA DELANTE
//Hombro
writeServoD(0, 60);
delay(25);
//Codo
writeServoD(1, 70);
delay(25);
//Muñeca
writeServoD(2, 90);
delay(25);
delay(t);

//PASO 5-----SUBIMOS Y ADELANTAMOS IM
//PATA IZQUIERDA MEDIO
//Hombro
writeServo(11, 120);
//Codo
writeServo(10, 135);
//Muñeca
writeServo(9, 45);
delay(25);
delay(t);

//-----BAJAMOS IM ADELANTADA
//PATA IZQUIERDA MEDIO
//Hombro
writeServo(11, 120);
delay(25);
//Codo
writeServo(10, 70);
delay(25);
//Muñeca
writeServo(9, 90);
delay(25);
delay(t);

//PASO 6-----SUBIMOS Y ADELANTAMOS DA
//PATA DERECHA DETRÁS
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
//Hombro
writeServoD(13, 60);
//Codo
writeServoD(14, 135);
//Muñeca
writeServoD(15, 45);
delay(25);
delay(t);

//-----BAJAMOS DA ADELANTADA
//PATA DERECHA DETRÁS
//Hombro
writeServoD(13, 60);
delay(25);
//Codo
writeServoD(14, 70);
delay(25);
//Muñeca
writeServoD(15, 90);
delay(25);
delay(t);

//PASO 7 -----ATRASAMOS TODO
j = 0;
for (i = 0; i <= 30; i++)
{
  j = 2 * i;
  writeServo(15, 120 - j); //izquierda delante
  writeServoD(0, 60 + j); //derecha delante
  delay(25);
  writeServo(11, 120 - j); //izquierda medio
  writeServoD(4, 60 + j); //derecha medio
  delay(25);
  writeServo(2, 120 - j); //izquierda atrás
  writeServoD(13, 60 + j); //derecha atrás
  delay(25);
}
delay(t);
}

//-----
void atras()
{
  //PASO 1-----
  //-----SUBIMOS Y ATRASAMOS PATA ID
  //PATA IZQUIERDA DELANTE
  //Hombro
  writeServo(15, 60);
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
//Codo
writeServol(14, 135);
//Muñeca
writeServol(13, 45);
delay(25);
delay(t);

//-----BAJAMOS LA PATA ID ATRASADA
//PATA IZQUIERDA DELANTE
//Hombro
writeServol(15, 60);
delay(25);
//Codo
writeServol(14, 70);
delay(25);
//Muñeca
writeServol(13, 90);
delay(25);
delay(t);

//PASO 2-----
//-----SUBIMOS Y ATRASAMOS PATA DM
//PATA DERECHA MEDIO
//Hombro
writeServoD(4, 120);
//Codo
writeServoD(5, 135);
//Muñeca
writeServoD(6, 45);
delay(25);
delay(t);

//-----BAJAMOS DM ADELANTADA
//PATA DERECHA MEDIO
//Hombro
writeServoD(4, 120);
delay(25);
//Codo
writeServoD(5, 70);
delay(25);
//Muñeca
writeServoD(6, 90);
delay(25);
delay(t);

//PASO 3 -----
//-----SUBIMOS Y ATRASAMOS IA
//PATA IZQUIERDA DETRÁS
```


ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
//Hombro
writeServol(2, 60);
//Codo
writeServol(1, 135);
//Muñeca
writeServol(0, 45);
delay(25);
delay(t);

//-----BAJAMOS IA ATRASAMOS
//PATA IZQUIERDA DETRÁS
//Hombro
writeServol(2, 60);
delay(25);
//Codo
writeServol(1, 70);
delay(25);
//Muñeca
writeServol(0, 90);
delay(25);
delay(t);

//PASO 4 -----SUBIMOS Y ATRASAMOS DD
//PATA DERECHA DELANTE
//Hombro
writeServoD(0, 120);
//Codo
writeServoD(1, 135);
//Muñeca
writeServoD(2, 45);
delay(25);
delay(t);

//-----BAJAMOS DD ATRASADA
//PATA DERECHA DELANTE
//Hombro
writeServoD(0, 120);
delay(25);
//Codo
writeServoD(1, 70);
delay(25);
//Muñeca
writeServoD(2, 90);
delay(25);
delay(t);

//PASO 5-----SUBIMOS Y ATRASAMOS IM
//PATA IZQUIERDA MEDIO
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
//Hombro
writeServo(11, 60);
//Codo
writeServo(10, 135);
//Muñeca
writeServo(9, 45);
delay(25);
delay(t);

//-----BAJAMOS IM ATRASADA
//PATA IZQUIERDA MEDIO
//Hombro
writeServo(11, 60);
delay(25);
//Codo
writeServo(10, 70);
delay(25);
//Muñeca
writeServo(9, 90);
delay(25);
delay(t);

//PASO 6-----SUBIMOS Y ATRASAMOS DA
//PATA DERECHA DETRÁS
//Hombro
writeServoD(13, 120);
//Codo
writeServoD(14, 135);
//Muñeca
writeServoD(15, 45);
delay(25);
delay(t);

//-----BAJAMOS DA ATRASADA
//PATA DERECHA DETRÁS
//Hombro
writeServoD(13, 120);
delay(25);
//Codo
writeServoD(14, 70);
delay(25);
//Muñeca
writeServoD(15, 90);
delay(25);
delay(t);

//PASO 7 -----ADELANTAMOS TODO
j = 0;
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```

for (i = 0; i <= 30; i++)
{
  j = 2 * i;
  writeServol(15, 60 + j);    //izquierda delante
  writeServoD(0, 120 - j);   //derecha delante
  delay(25);
  writeServol(11, 60 + j);   //izquierda medio
  writeServoD(4, 120 - j);   //derecha medio
  delay(25);
  writeServol(2, 60 + j);    //izquierda atrás
  writeServoD(13, 120 - j);  //derecha atrás
  delay(25);
}
delay(t);
}

//-----
void derecha()
{
  //PASO 1-----
  //-----SUBIMOS Y ADELANTAMOS PATA ID
  //PATA IZQUIERDA DELANTE
  //Hombro
  writeServol(15, 120);
  //Codo
  writeServol(14, 135);
  //Muñeca
  writeServol(13, 45);
  delay(25);
  delay(t);
  //-----BAJAMOS LA PATA ID ADELANTADA
  //PATA IZQUIERDA DELANTE
  //Hombro
  writeServol(15, 120);
  delay(25);
  //Codo
  writeServol(14, 70);
  delay(25);
  //Muñeca
  writeServol(13, 90);
  delay(25);
  delay(t);

  //PASO 2-----
  //-----SUBIMOS Y ADELANTAMOS PATA DM
  //PATA DERECHA MEDIO
  //Hombro
  writeServoD(4, 120);

```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
//Codo
writeServoD(5, 135);
//Muñeca
writeServoD(6, 45);
delay(25);
delay(t);

//-----BAJAMOS DM ADELANTADA
//PATA DERECHA MEDIO
//Hombro
writeServoD(4, 120);
delay(25);
//Codo
writeServoD(5, 70);
delay(25);
//Muñeca
writeServoD(6, 90);
delay(25);
delay(t);

//PASO 3 -----
//-----SUBIMOS Y ADELANTAMOS IA
//PATA IZQUIERDA DETRÁS
//Hombro
writeServoI(2, 120);
//Codo
writeServoI(1, 135);
//Muñeca
writeServoI(0, 45);
delay(25);
delay(t);

//-----BAJAMOS IA ADELANTADA
//PATA IZQUIERDA DETRÁS
//Hombro
writeServoI(2, 120);
delay(25);
//Codo
writeServoI(1, 70);
delay(25);
//Muñeca
writeServoI(0, 90);
delay(25);
delay(t);

//PASO 4 -----SUBIMOS Y ADELANTAMOS DD
//PATA DERECHA DELANTE
//Hombro
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
writeServoD(0, 120);
//Codo
writeServoD(1, 135);
//Muñeca
writeServoD(2, 45);
delay(25);
delay(t);

//-----BAJAMOS DD ADELANTADA
//PATA DERECHA DELANTE
//Hombro
writeServoD(0, 120);
delay(25);
//Codo
writeServoD(1, 70);
delay(25);
//Muñeca
writeServoD(2, 90);
delay(25);
delay(t);

//PASO 5-----SUBIMOS Y ADELANTAMOS IM
//PATA IZQUIERDA MEDIO
//Hombro
writeServoI(11, 120);
//Codo
writeServoI(10, 135);
//Muñeca
writeServoI(9, 45);
delay(25);
delay(t);

//-----BAJAMOS IM ADELANTADA
//PATA IZQUIERDA MEDIO
//Hombro
writeServoI(11, 120);
delay(25);
//Codo
writeServoI(10, 70);
delay(25);
//Muñeca
writeServoI(9, 90);
delay(25);
delay(t);

//PASO 6-----SUBIMOS Y ADELANTAMOS DA
//PATA DERECHA DETRÁS
//Hombro
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
writeServoD(13, 120);
//Codo
writeServoD(14, 135);
//Muñeca
writeServoD(15, 45);
delay(25);
delay(t);

//-----BAJAMOS DA ADELANTADA
//PATA DERECHA DETRÁS
//Hombro
writeServoD(13, 120);
delay(25);
//Codo
writeServoD(14, 70);
delay(25);
//Muñeca
writeServoD(15, 90);
delay(25);
delay(t);

//PASO 7 -----ATRASAMOS TODO
j = 0;
for (i = 0; i <= 30; i++)
{
  j = 2 * i;
  writeServoI(15, 120 - j); //izquierda delante
  writeServoD(0, 120 - j); //derecha delante
  delay(25);
  writeServoI(11, 120 - j); //izquierda medio
  writeServoD(4, 120 - j); //derecha medio
  delay(25);
  writeServoI(2, 120 - j); //izquierda atrás
  writeServoD(13, 120 - j); //derecha atrás
  delay(25);
}
delay(t);
}

//-----
void izquierda()
{
  //PASO 1-----
  //-----SUBIMOS Y ADELANTAMOS PATA ID
  //PATA IZQUIERDA DELANTE
  //Hombro
  writeServoI(15, 60);
  //Codo
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
writeServo(14, 135);
//Muñeca
writeServo(13, 45);
delay(25);
delay(t);

//-----BAJAMOS LA PATA ID ADELANTADA
//PATA IZQUIERDA DELANTE
//Hombro
writeServo(15, 60);
delay(25);
//Codo
writeServo(14, 70);
delay(25);
//Muñeca
writeServo(13, 90);
delay(25);
delay(t);

//PASO 2-----
//-----SUBIMOS Y ADELANTAMOS PATA DM
//PATA DERECHA MEDIO
//Hombro
writeServoD(4, 60);
//Codo
writeServoD(5, 135);
//Muñeca
writeServoD(6, 45);
delay(25);
delay(t);

//-----BAJAMOS DM ADELANTADA
//PATA DERECHA MEDIO
//Hombro
writeServoD(4, 60);
delay(25);
//Codo
writeServoD(5, 70);
delay(25);
//Muñeca
writeServoD(6, 90);
delay(25);
delay(t);

//PASO 3 -----
//-----SUBIMOS Y ADELANTAMOS IA
//PATA IZQUIERDA DETRÁS
//Hombro
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
writeServo(2, 60);
//Codo
writeServo(1, 135);
//Muñeca
writeServo(0, 45);
delay(25);
delay(t);

//-----BAJAMOS IA ADELANTADA
//PATA IZQUIERDA DETRÁS
//Hombro
writeServo(2, 60);
delay(25);
//Codo
writeServo(1, 70);
delay(25);
//Muñeca
writeServo(0, 90);
delay(25);
delay(t);

//PASO 4 -----SUBIMOS Y ADELANTAMOS DD
//PATA DERECHA DELANTE
//Hombro
writeServoD(0, 60);
//Codo
writeServoD(1, 135);
//Muñeca
writeServoD(2, 45);
delay(25);
delay(t);

//-----BAJAMOS DD ADELANTADA
//PATA DERECHA DELANTE
//Hombro
writeServoD(0, 60);
delay(25);
//Codo
writeServoD(1, 70);
delay(25);
//Muñeca
writeServoD(2, 90);
delay(25);
delay(t);

//PASO 5-----SUBIMOS Y ADELANTAMOS IM
//PATA IZQUIERDA MEDIO
//Hombro
```


ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
writeServo(11, 60);
//Codo
writeServo(10, 135);
//Muñeca
writeServo(9, 45);
delay(25);
delay(t);

//-----BAJAMOS IM ADELANTADA
//PATA IZQUIERDA MEDIO
//Hombro
writeServo(11, 60);
delay(25);
//Codo
writeServo(10, 70);
delay(25);
//Muñeca
writeServo(9, 90);
delay(25);
delay(t);

//PASO 6-----SUBIMOS Y ADELANTAMOS DA
//PATA DERECHA DETRÁS
//Hombro
writeServoD(13, 60);
//Codo
writeServoD(14, 135);
//Muñeca
writeServoD(15, 45);
delay(25);
delay(t);

//-----BAJAMOS DA ADELANTADA
//PATA DERECHA DETRÁS
//Hombro
writeServoD(13, 60);
delay(25);
//Codo
writeServoD(14, 70);
delay(25);
//Muñeca
writeServoD(15, 90);
delay(25);
delay(t);

//PASO 7 -----ATRASAMOS TODO
j = 0;
for (i = 0; i <= 30; i++)
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
{
  j = 2 * i;
  writeServo(15, 60 + j);    //izquierda delante
  writeServoD(0, 60 + j);   //derecha delante
  delay(25);
  writeServo(11, 60 + j);   //izquierda medio
  writeServoD(4, 60 + j);   //derecha medio
  delay(25);
  writeServo(2, 60 + j);    //izquierda atrás
  writeServoD(13, 60 + j);  //derecha atrás
  delay(25);
}
delay(t);
}
```

```
//-----
void modo_Paquete()
{
  arriba();
  delay(300);
  abajo();
  delay(300);

  //PASO 1----- NEUTRAL
  //PATA IZQUIERDA DELANTE
  //Hombro
  writeServo(15, 150);
  delay(25);
  //Codo
  writeServo(14, 180);
  delay(25);
  //Muñeca
  writeServo(13, 0);
  delay(25);

  //PATA DERECHA DELANTE
  //Hombro
  writeServoD(0, 30);
  delay(25);
  //Codo
  writeServoD(1, 180);
  delay(25);
  //Muñeca
  writeServoD(2, 0);
  delay(25);

  //PATA IZQUIERDA MEDIO
  //Hombro
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
writeServo(11, 130);
delay(25);
//Codo
writeServo(10, 180);
delay(25);
//Muñeca
writeServo(9, 0);
delay(25);

//PATA DERECHA MEDIO
//Hombro
writeServoD(4, 50);
delay(25);
//Codo
writeServoD(5, 180);
delay(25);
//Muñeca
writeServoD(6, 0);
delay(25);

//PATA IZQUIERDA DETRÁS
//Hombro
writeServo(2, 180);
delay(25);
//Codo
writeServo(1, 180);
delay(25);
//Muñeca
writeServo(0, 0);
delay(25);

//PATA DERECHA DETRÁS
//Hombro
writeServoD(13, 0);
delay(25);
//Codo
writeServoD(14, 180);
delay(25);
//Muñeca
writeServoD(15, 0);
delay(25);

}

//-----
void saludar()
{
  arriba();
}
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
delay(300);
//-----SONRIE
expression((9), (8), (10), 24, 126, 231, 195, 129, 0, 0, 0);
delay(75);

//---ADELANTAMOS PATA IZQUIERDA
//PATA IZQUIERDA DELANTE
//Hombro
writeServo(15, 120);
delay(25);
//Codo
writeServo(14, 135);
delay(25);
//Muñeca
writeServo(13, 45);
delay(250);

//-----BAJAMOS PATA IZQUIERDA DELANTERA
//PATA IZQUIERDA DELANTE
//Hombro
writeServo(15, 120);
delay(25);
//Codo
writeServo(14, 70);
delay(25);
//Muñeca
writeServo(13, 90);
delay(250);

//-----ADELANTAMOS Y SUBIMOS PATA DM
//PATA DERECHA MEDIO
//Hombro
writeServoD(4, 60);
delay(25);
//Codo
writeServoD(5, 135);
delay(25);
//Muñeca
writeServoD(6, 45);
delay(250);

//-----BAJAMOS PATA DM-----
//PATA DERECHA MEDIO
//Hombro
writeServoD(4, 60);
delay(25);
//Codo
writeServoD(5, 70);
```

```
delay(25);
//Muñeca
writeServoD(6, 90);
delay(250);

//-----SUBIMOS PATA DD
//PATA DERECHA DELANTE
//Hombro
writeServoD(0, 30);
//Codo
writeServoD(1, 180);
//Muñeca
writeServoD(2, 90);
delay(300);

//-----MUEVE LA PATA A UN LADO Y A OTRO
//PATA DERECHA DELANTE
//Hombro
writeServoD(0, 90);
//Codo
writeServoD(1, 180);
//Muñeca
writeServoD(2, 120);
delay(300);

//PATA DERECHA DELANTE
//Hombro
writeServoD(0, 30);
//Codo
writeServoD(1, 180);
//Muñeca
writeServoD(2, 120);
delay(300);

//PATA DERECHA DELANTE
//Hombro
writeServoD(0, 90);
//Codo
writeServoD(1, 180);
//Muñeca
writeServoD(2, 120);
delay(300);

//PATA DERECHA DELANTE
//Hombro
writeServoD(0, 30);
//Codo
writeServoD(1, 180);
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
//Muñeca
writeServoD(2, 120);
delay(300);

//PATA DERECHA DELANTE
//Hombro
writeServoD(0, 90);
//Codo
writeServoD(1, 180);
//Muñeca
writeServoD(2, 120);
delay(300);

//PATA DERECHA DELANTE
//Hombro
writeServoD(0, 30);
//Codo
writeServoD(1, 180);
//Muñeca
writeServoD(2, 120);
delay(300);

//PATA DERECHA DELANTE
//Hombro
writeServoD(0, 30);
delay(25);
//Codo
writeServoD(1, 70);
delay(25);
//Muñeca
writeServoD(2, 90);
delay(300);

expression((9), (8), (10), 0, 0, 0, 0, 0, 0, 0, 0);
delay(75);
}

//-----
void calibrado()
{
  servoBoca.write(0);
  delay(25);
}

//-----
void modo_Patrulla()
{
  lectura_ultrasonidos();
}
```

```

if (distance > 200)
{
  expression((9), (8), (10), 255, 255, 0, 0, 0, 0, 255, 255);
  delay(75);
  t = 250;
  adelante();
  expression((9), (8), (10), 0, 0, 0, 0, 0, 0, 0, 0);
  delay(75);
}
if ( (distance <= 200) and (distance > 150) )
{
  t = 250;
  //BOCA EN FORMA DE O
  expression((9), (8), (10), 24, 60, 102, 195, 195, 102, 60, 24);
  delay(75);
  adelante();
  expression((9), (8), (10), 0, 0, 0, 0, 0, 0, 0, 0);
  delay(75);
}
if ( (distance <= 150) and (distance > 100) )
{
  t = 250;
  //BOCA EN EXCLAMACIÓN
  expression((9), (8), (10), 24, 24, 0, 24, 24, 24, 24, 24);
  delay(75);
  derecha();
  expression((9), (8), (10), 0, 0, 0, 0, 0, 0, 0, 0);
  delay(75);
}
if ( (distance >= 40) and (distance <= 100) )
{
  t = 250;
  expression((9), (8), (10), 255, 255, 255, 255, 255, 255, 255, 255);
  delay(75);

  asustar();
  abrir_pinzas();

  //-----BAJAMOS PATAS DI Y DD
  //PATA IZQUIERDA DELANTE
  //Hombro
  writeServo(15, 120);
  delay(25);
  //Codo
  writeServo(14, 70);
  delay(25);
  //Muñeca

```

```

writeServo(13, 90);
delay(25);

//PATA DERECHA DELANTE
//Hombro
writeServoD(0, 60);
delay(25);
//Codo
writeServoD(1, 70);
delay(25);
//Muñeca
writeServoD(2, 90);
delay(25);
//-----
delay(250);
cerrar_pinzas();
delay(250);
abrir_pinzas();
delay(250);
cerrar_pinzas();
delay(250);
arriba();
delay(300);
derecha();
delay(300);
expression((9), (8), (10), 0, 0, 0, 0, 0, 0, 0, 0);
delay(75);
}
if (distance < 40)
{
t = 250;
arriba();
delay(300);
abajo();
delay(300);
//PASO 1----- NEUTRAL
//PATA IZQUIERDA DELANTE
//Hombro
writeServo(15, 150);
delay(25);
//Codo
writeServo(14, 180);
delay(25);
//Muñeca
writeServo(13, 0);
delay(25);

//PATA DERECHA DELANTE

```



```
//Hombro
writeServoD(0, 30);
delay(25);
//Codo
writeServoD(1, 180);
delay(25);
//Muñeca
writeServoD(2, 0);
delay(25);

//PATA IZQUIERDA MEDIO
//Hombro
writeServo(11, 130);
delay(25);
//Codo
writeServo(10, 180);
delay(25);
//Muñeca
writeServo(9, 0);
delay(25);

//PATA DERECHA MEDIO
//Hombro
writeServoD(4, 50);
delay(25);
//Codo
writeServoD(5, 180);
delay(25);
//Muñeca
writeServoD(6, 0);
delay(25);

//PATA IZQUIERDA DETRÁS
//Hombro
writeServo(2, 180);
delay(25);
//Codo
writeServo(1, 180);
delay(25);
//Muñeca
writeServo(0, 0);
delay(25);

//PATA DERECHA DETRÁS
//Hombro
writeServoD(13, 0);
delay(25);
//Codo
```

```

writeServoD(14, 180);
delay(25);
//Muñeca
writeServoD(15, 0);
delay(25);
delay(300);
abajo();
delay(300);
arriba();
delay(300);
atras();
expression((9), (8), (10), 0, 0, 0, 0, 0, 0, 0, 0);
delay(75);
}
}

```

```

//-----
void lectura_ultrasonidos()
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2;
}

```

```

//-----
void asustar()
{
//-----ADELANTAMOS Y SUBIMOS PATA MI
//PATA IZQUIERDA MEDIO
//Hombro
writeServo(11, 120);
//Codo
writeServo(10, 135);
//Muñeca
writeServo(9, 45);
delay(250);

//-----BAJAMOS PATA MI ADELANTADA
//PATA IZQUIERDA MEDIO
//Hombro
writeServo(11, 120);
//Codo
writeServo(10, 70);
//Muñeca

```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
writeServo(9, 90);
delay(250);

//-----ADELANTAMOS Y SUBIMOS PATA MD
//PATA DERECHA MEDIO
//Hombro
writeServoD(4, 60);
//Codo
writeServoD(5, 135);
//Muñeca
writeServoD(6, 45);
delay(250);

//-----BAJAMOS PATA MD ADELANTADA
//PATA DERECHA MEDIO
//Hombro
writeServoD(4, 60);
delay(25);
//Codo
writeServoD(5, 70);
delay(25);
//Muñeca
writeServoD(6, 90);
delay(250);

//-----LEVANTAMOS PATAS DI Y DD
//PATA IZQUIERDA DELANTE
//Hombro
writeServo(15, 120);
//Codo
writeServo(14, 180);
//Muñeca
writeServo(13, 120);
delay(25);

//PATA DERECHA DELANTE
//Hombro
writeServoD(0, 60);
//Codo
writeServoD(1, 180);
//Muñeca
writeServoD(2, 120);
delay(1000);
}

//-----SET UP-----
void setup()
{
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
Bluetooth.begin(9600);

beginServos();
//-----PINES PARA EL ULTRASONIDOS-----
pinMode(trigPin, OUTPUT); // TRIG COMO SALIDA
pinMode(echoPin, INPUT); // ECHO COMO ENTRADA

//-----PINES PARA LA MATRIZ LED-----
pinMode((9), OUTPUT);
pinMode((8), OUTPUT);
pinMode((10), OUTPUT);

maxAll((9), (8), (10), 11, 7);
maxAll((9), (8), (10), 9, 0);
maxAll((9), (8), (10), 12, 1);
maxAll((9), (8), (10), 15, 0);
int i = 0;
for (i = 1; i <= 8; i++)
  maxAll((9), (8), (10), i, 0);
maxAll((9), (8), (10), 10, 15);

//-----
servoBoca.write(0);
pinMode(13, OUTPUT); //LUCES LEDS

//-----BIENVENIDA-----
delay(250);
abajo();

//-----inicio secuencia bienvenida-----
expression((9), (8), (10), 129, 129, 129, 129, 129, 129, 129);
delay(100);
expression((9), (8), (10), 195, 195, 195, 195, 195, 195, 195);
delay(100);
expression((9), (8), (10), 231, 231, 231, 231, 231, 231, 231);
delay(100);
expression((9), (8), (10), 255, 255, 255, 255, 255, 255, 255);
delay(100);
expression((9), (8), (10), 231, 231, 231, 231, 231, 231, 231);
delay(100);
expression((9), (8), (10), 195, 195, 195, 195, 195, 195, 195);
delay(100);
expression((9), (8), (10), 129, 129, 129, 129, 129, 129, 129);
delay(100);
expression((9), (8), (10), 0, 0, 0, 0, 0, 0, 0);
delay(100);

expression((9), (8), (10), 129, 129, 129, 129, 129, 129, 129);
```

ANEXO A-3: PROGRAMACIÓN DEL ROBOT. CÓDIGO ARDUINO

```
delay(100);
expression((9), (8), (10), 195, 195, 195, 195, 195, 195, 195, 195);
delay(100);
expression((9), (8), (10), 231, 231, 231, 231, 231, 231, 231, 231);
delay(100);
expression((9), (8), (10), 255, 255, 255, 255, 255, 255, 255, 255);
delay(100);
expression((9), (8), (10), 231, 231, 231, 231, 231, 231, 231, 231);
delay(100);
expression((9), (8), (10), 195, 195, 195, 195, 195, 195, 195, 195);
delay(100);
expression((9), (8), (10), 129, 129, 129, 129, 129, 129, 129, 129);
delay(100);
expression((9), (8), (10), 0, 0, 0, 0, 0, 0, 0, 0);
delay(100);

expression((9), (8), (10), 129, 129, 129, 129, 129, 129, 129, 129);
delay(100);
expression((9), (8), (10), 195, 195, 195, 195, 195, 195, 195, 195);
delay(100);
expression((9), (8), (10), 231, 231, 231, 231, 231, 231, 231, 231);
delay(100);
expression((9), (8), (10), 255, 255, 255, 255, 255, 255, 255, 255);
delay(100);
expression((9), (8), (10), 231, 231, 231, 231, 231, 231, 231, 231);
delay(100);
expression((9), (8), (10), 195, 195, 195, 195, 195, 195, 195, 195);
delay(100);
expression((9), (8), (10), 129, 129, 129, 129, 129, 129, 129, 129);
delay(100);
expression((9), (8), (10), 0, 0, 0, 0, 0, 0, 0, 0);
delay(100);
}

//-----BUCLE PRINCIPAL-----
void loop()
{
  if (Bluetooth.available() > 0)
  {
    ordenes = Bluetooth.read();
    if ( ordenes == 'q')
    {
      arriba();

    }
    if (ordenes == 'w')
    {
      adelante();
    }
  }
}
```

```
}

if (ordenes == 'a')
{
  izquierda();
}

if (ordenes == 's')
{
  atras();
}

if (ordenes == 'd')
{
  derecha();
}

if (ordenes == 'b')
{
  cerrar_pinzas();
}

if (ordenes == 'c')
{
  abrir_pinzas();
}

if (ordenes == 'e')
{
  //ENCENDER LUCES
  digitalWrite(13, HIGH);
  delay(50);
}

if (ordenes == 'f')
{
  //APAGAR LUCES
  digitalWrite(13, LOW);
  delay(50);
}

if (ordenes == 'k')
{
  modo_Paquete();
}

if (ordenes == 'm')
```

```

{
  //RÁFAGAS
  digitalWrite(13, LOW);
  delay(50);
  digitalWrite(13, HIGH);
  delay(50);
  digitalWrite(13, LOW);
  delay(50);
  digitalWrite(13, HIGH);
  delay(50);
  digitalWrite(13, LOW);
  delay(50);
  digitalWrite(13, HIGH);
  delay(50);
  digitalWrite(13, LOW);
  delay(50);
}

if (ordenes == 'n')
{
  arriba();
}

if (ordenes == 'o')
{
  abajo();
}

if (ordenes == 'p')
{
  rotar_derecha();
}

if (ordenes == 'Q')
{
  rotar_izquierda();
}

if (ordenes == 'r')
{
  //VELOCIDAD LENTA
  t = 750;
}

if (ordenes == 't')
{
  //VELOCIDAD MEDIA
  t = 500;
}

```

```
}

if (ordenes == 'x')
{
  //VELOCIDAD ALTA
  t = 250;
}

if (ordenes == 'y')
{
  calibrado();
}

if ( ordenes == 'l')
{
  abajo();
}

if ( ordenes == 'M')
{
  saludar();
}

if ( ordenes == 'i')
{
  estadomodopatrulla = 1;
  while (estadomodopatrulla == 1)
  {
    ordenes = Bluetooth.read();
    modo_Patrulla();
    if ( ordenes == 'j')
    {
      estadomodopatrulla = 0;
      arriba();
    }
  }
}
}
}
```




UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

DOCUMENTO NÚMERO 2:

PLANOS

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

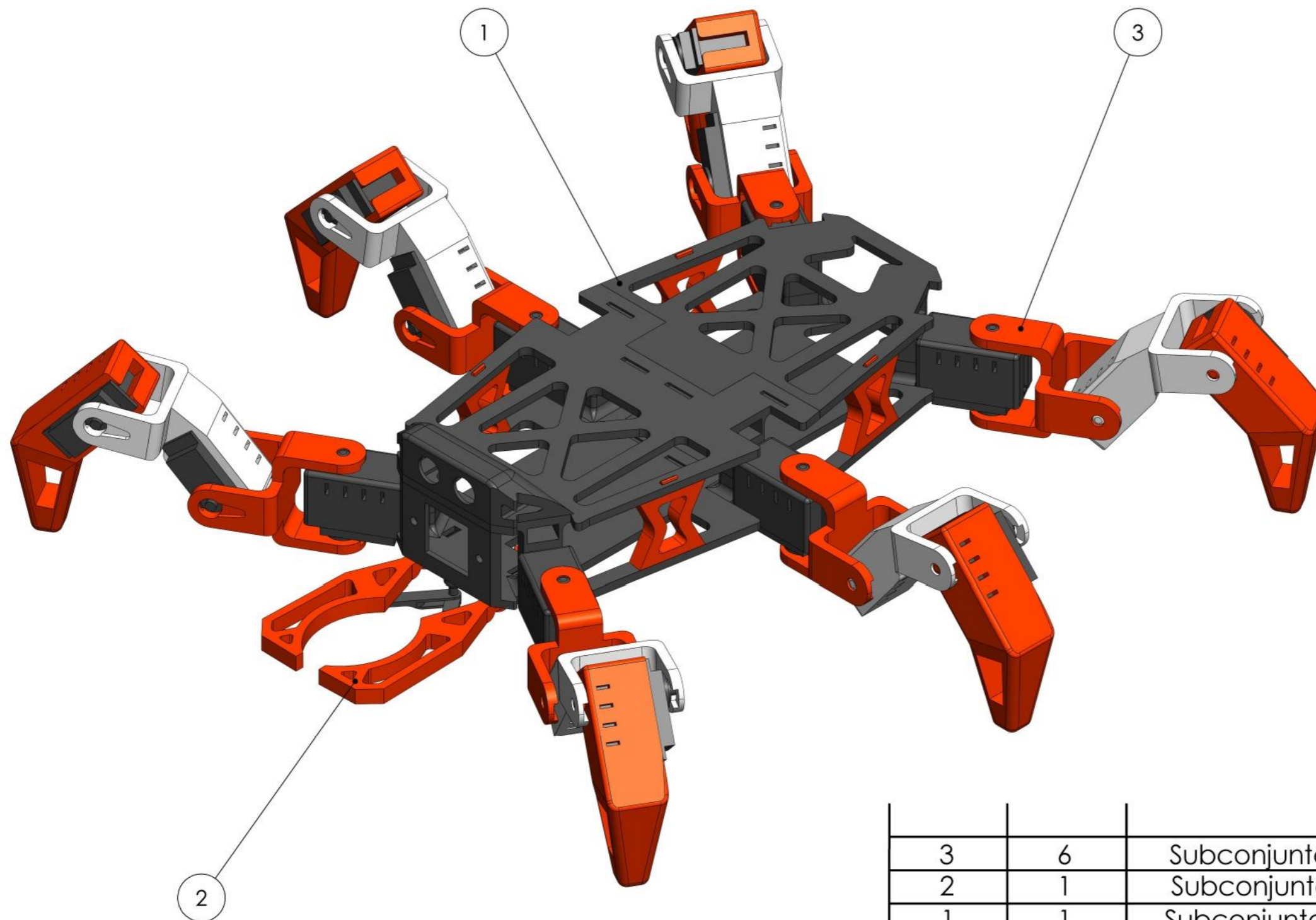
Álvaro Alberto Giner



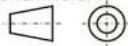
TUTORIZADO POR

Leopoldo Armesto Ángel

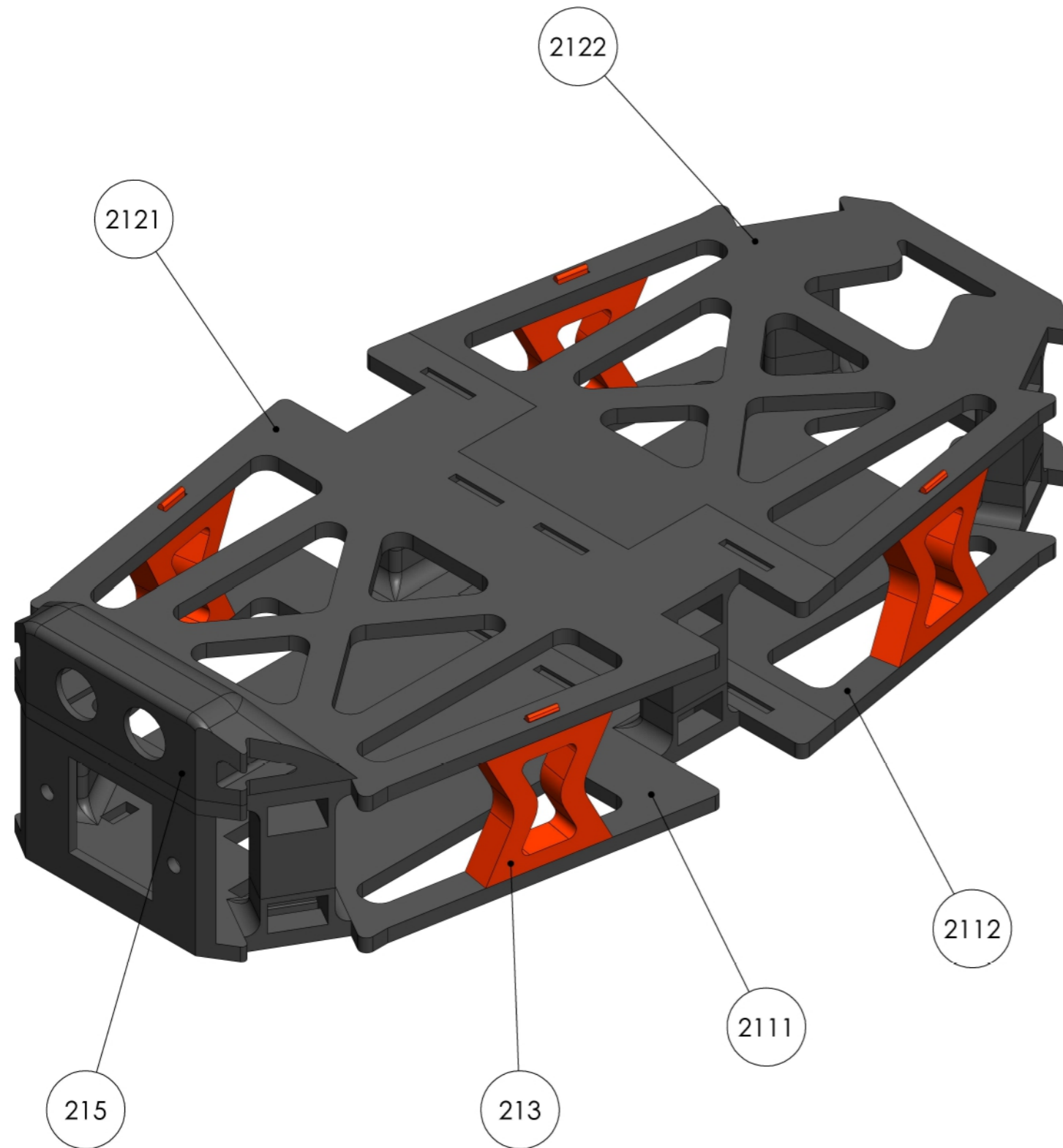
CURSO ACADÉMICO: 2019/2020

Hoja en blanco dejada a propósito



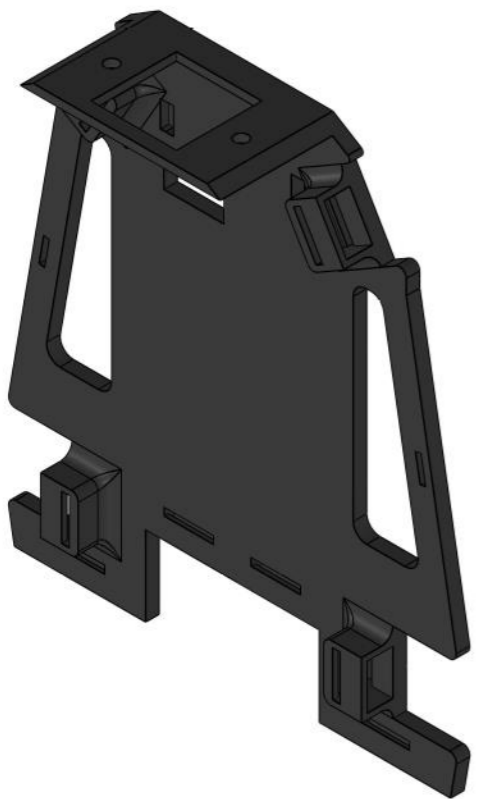
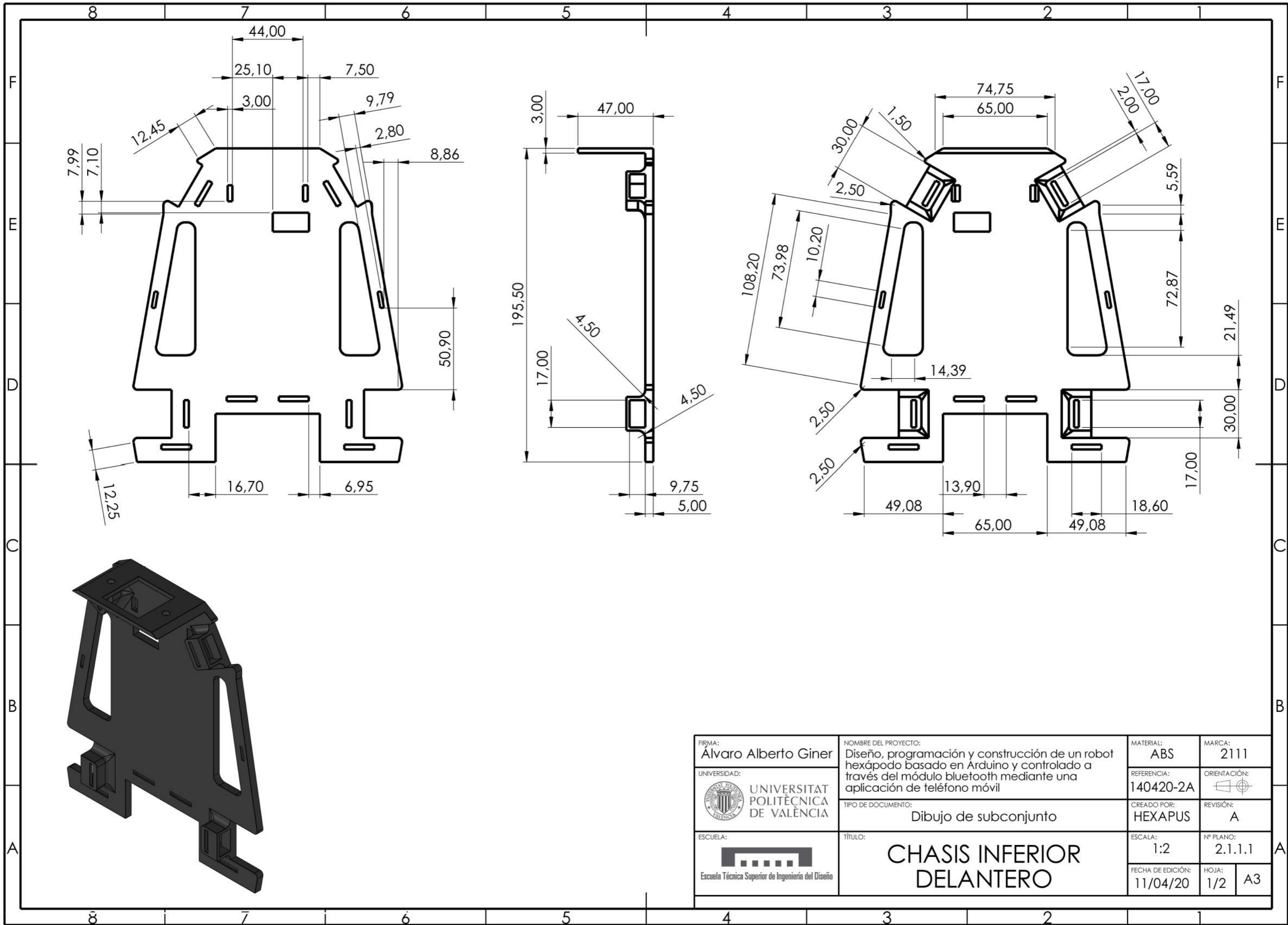
Marca	Nº Pieza	Designación	Norma	Material
	3	6	UNE-EN ISO 15015	ABS
	2	1	UNE-EN ISO 15015	ABS
	1	1	UNE-EN ISO 15015	ABS
FIRMA: Álvaro Alberto Giner		NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil		MATERIAL: ABS
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA		TIPO DE DOCUMENTO: VISTA GENERAL		REFERENCIA: 120420-9
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño		TÍTULO: ROBOT HEXÁPODO		ORIENTACIÓN: 
				CREADO POR: HEXAPUS
				REVISIÓN: A
				ESCALA: 1:2.5
				Nº PLANO: 2
				FECHA DE EDICIÓN: 12/04/20
				HOJA: 1/1
				A3

ESCALA ADICIONAL: 1:2

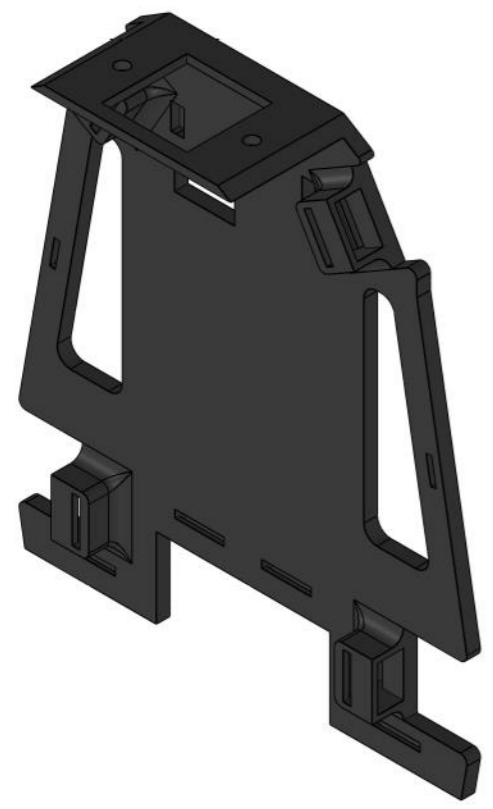
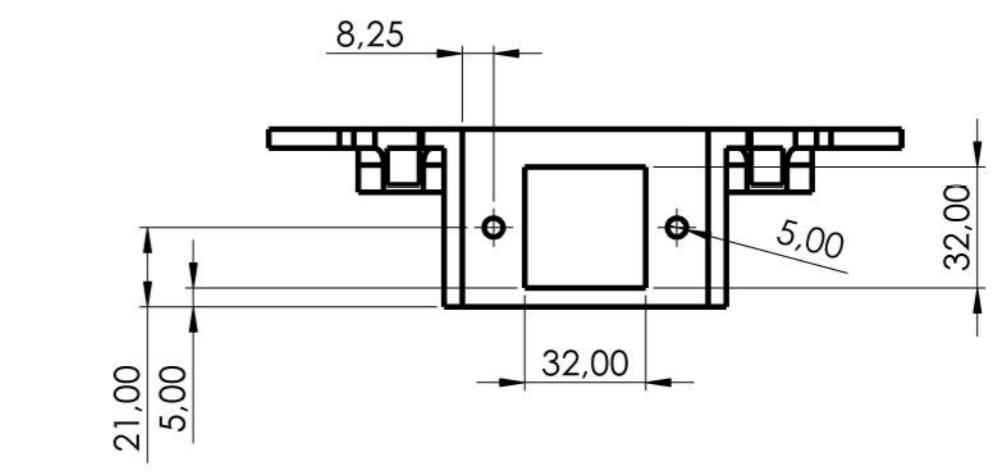
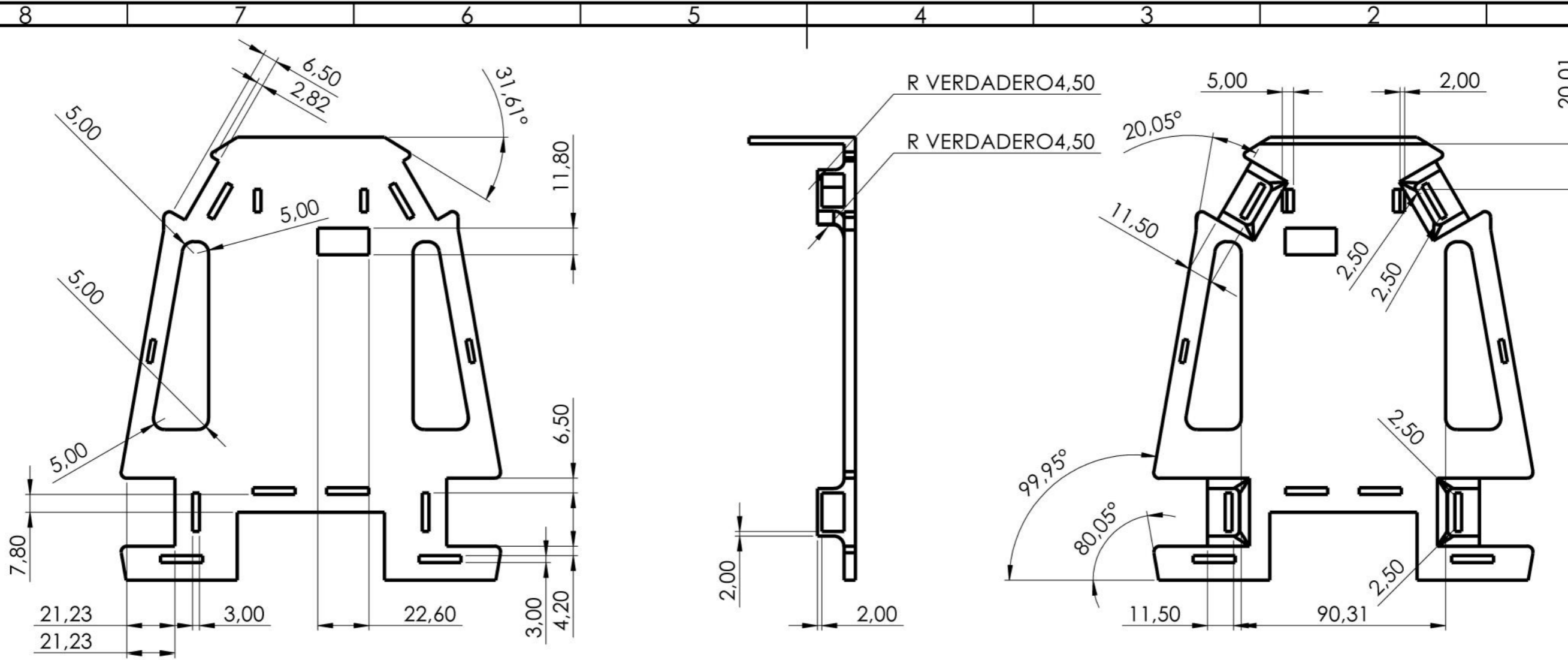


Marca	Nº Pieza	Designación	Norma	Material	
	215	1	Soporte Ultrasonidos	UNE-EN ISO 15015	ABS
	214	1	Soporte Batería	UNE-EN ISO 15015	ABS
	213	4	Pilares	UNE-EN ISO 15015	ABS
	2122	1	Chasis Superior Trasero	UNE-EN ISO 15015	ABS
	2121	1	Chasis Superior Delantero	UNE-EN ISO 15015	ABS
	2112	1	Chasis Inferior Trasero	UNE-EN ISO 15015	ABS
	2111	1	Chasis Inferior Delantero	UNE-EN ISO 15015	ABS

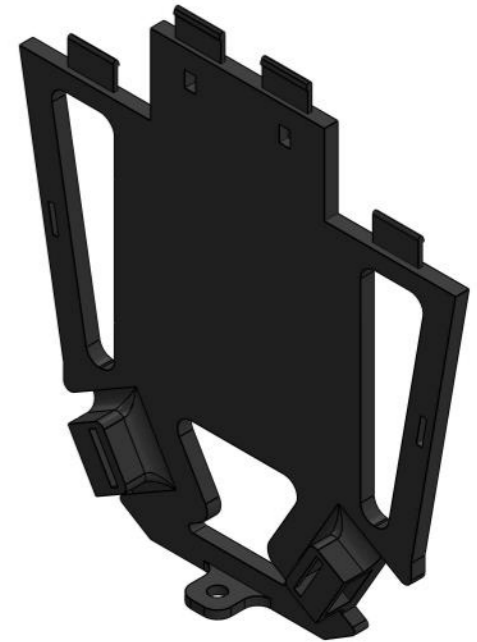
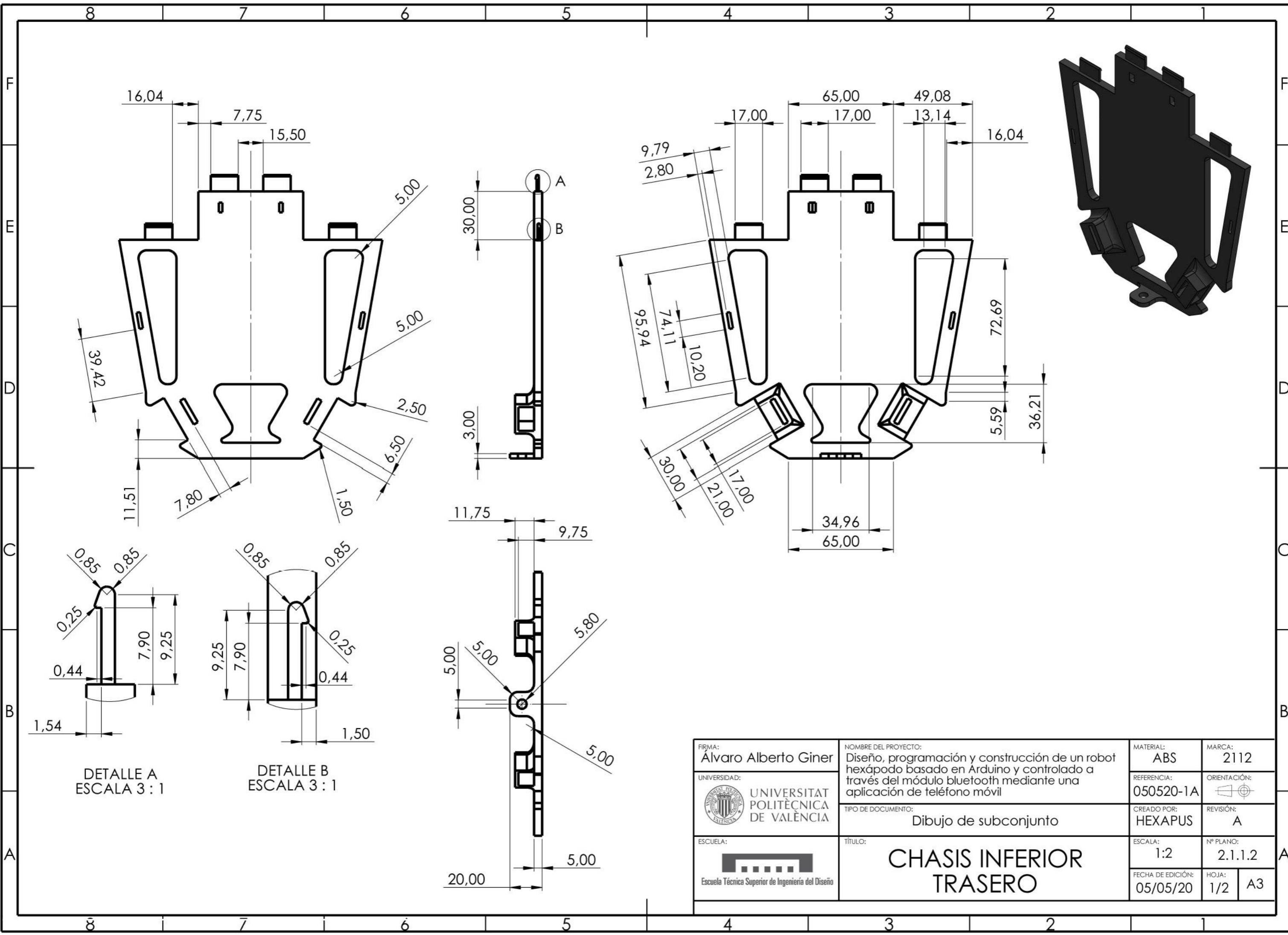
FIRMA: Álvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 1
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Vista General de subconjunto	REFERENCIA: 120420-7	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: CHASIS	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 1:1.5	Nº PLANO: 2.1
		FECHA DE EDICIÓN: 12/04/20	HOJA: 1/1
			A3



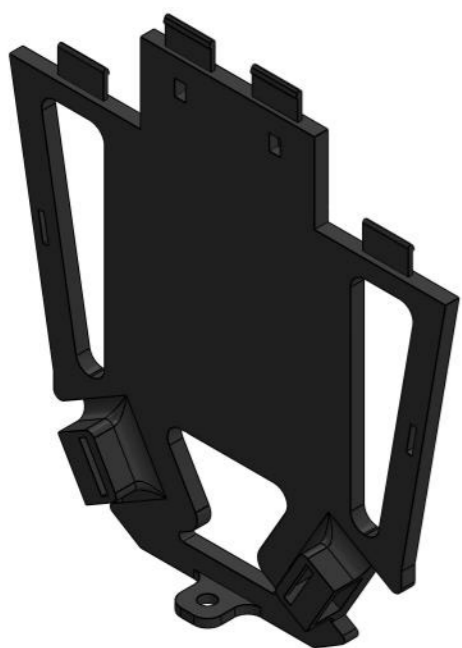
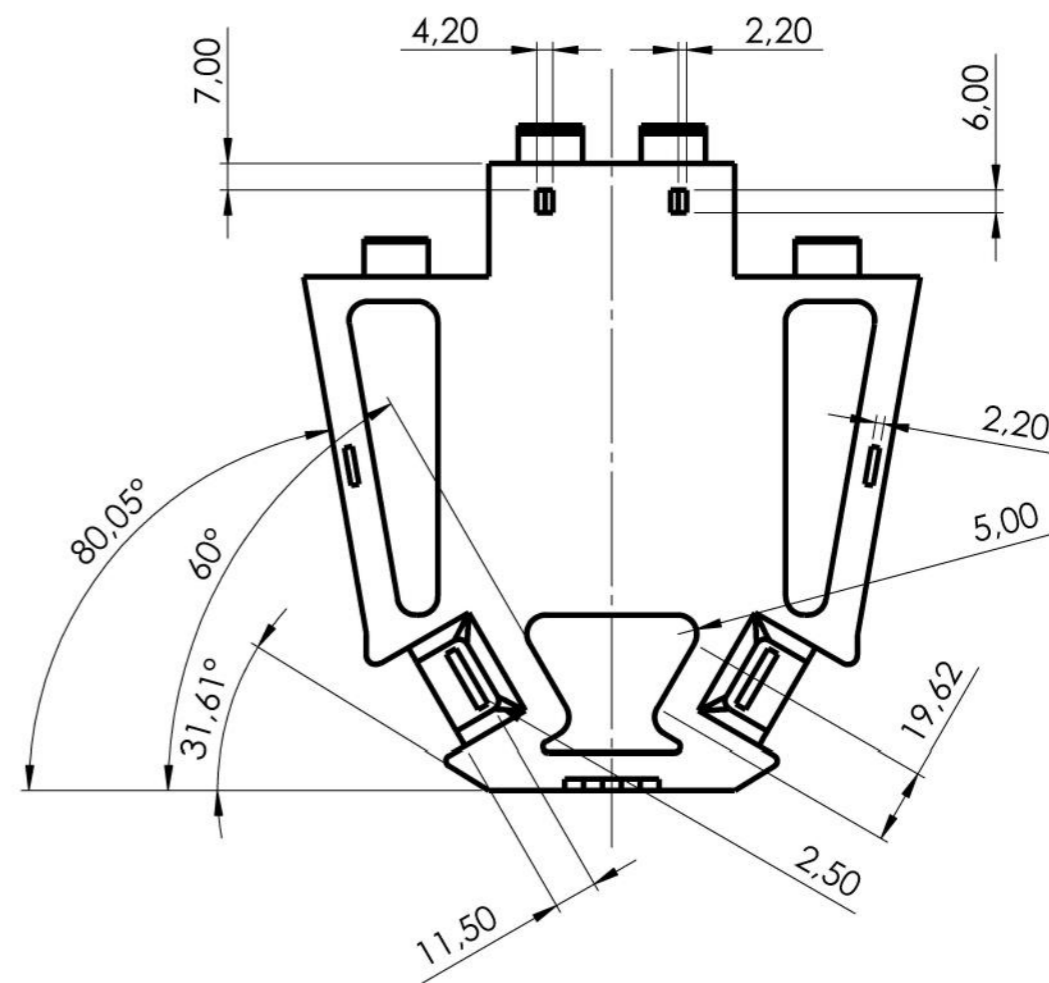
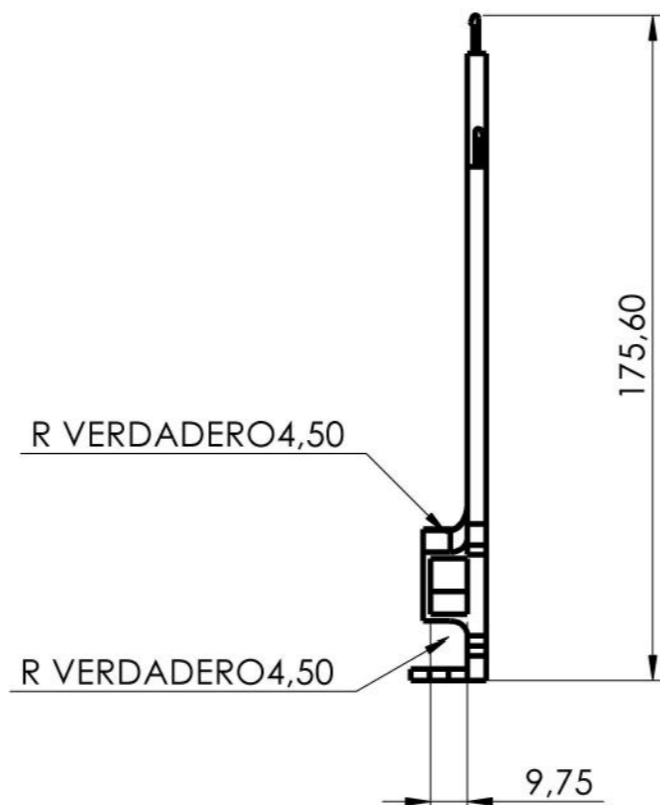
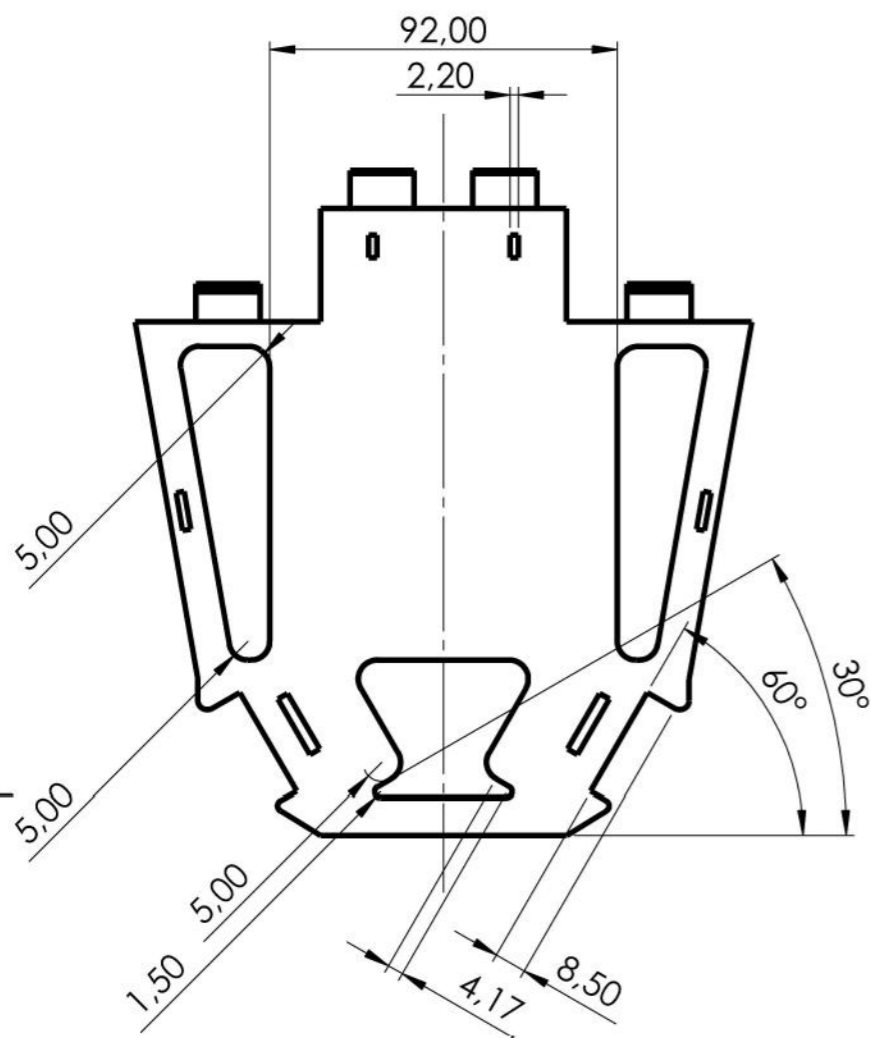
FIRMA: Álvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 2111
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 140420-2A	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: CHASIS INFERIOR DELANTERO	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 1:2	Nº PLANO: 2.1.1.1
		FECHA DE EDICIÓN: 11/04/20	HOJA: 1/2
			A3



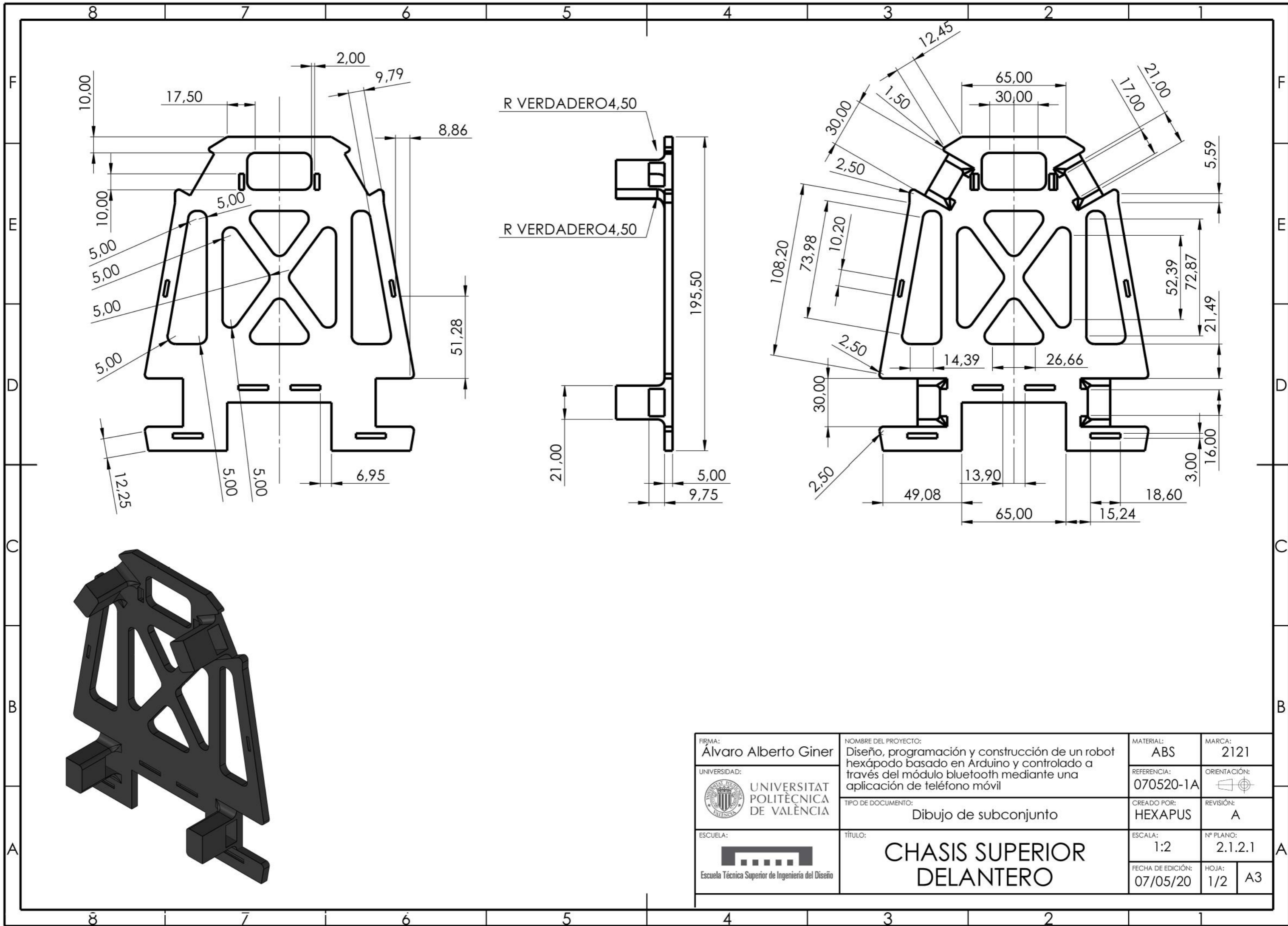
FIRMA: Álvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL:	ABS	MARCA:	2111
		UNIVERSIDAD:	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA		REFERENCIA:
ESCUELA:	 Escuela Técnica Superior de Ingeniería del Diseño		TIPO DE DOCUMENTO:	Dibujo de subconjunto	
CHASIS INFERIOR DELANTERO		CREADO POR:	HEXAPUS	REVISIÓN:	A
		ESCALA:	1:2	Nº PLANO:	2.1.1.1
		FECHA DE EDICIÓN:	11/04/20	HOJA:	2/2
					A3



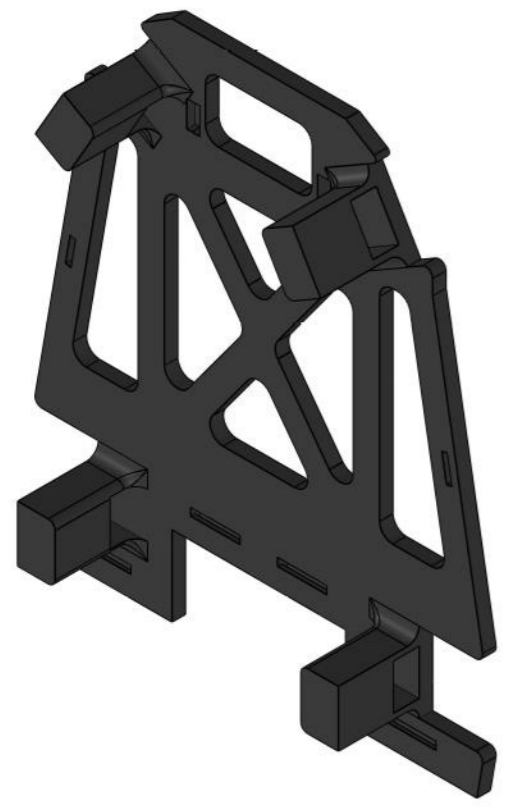
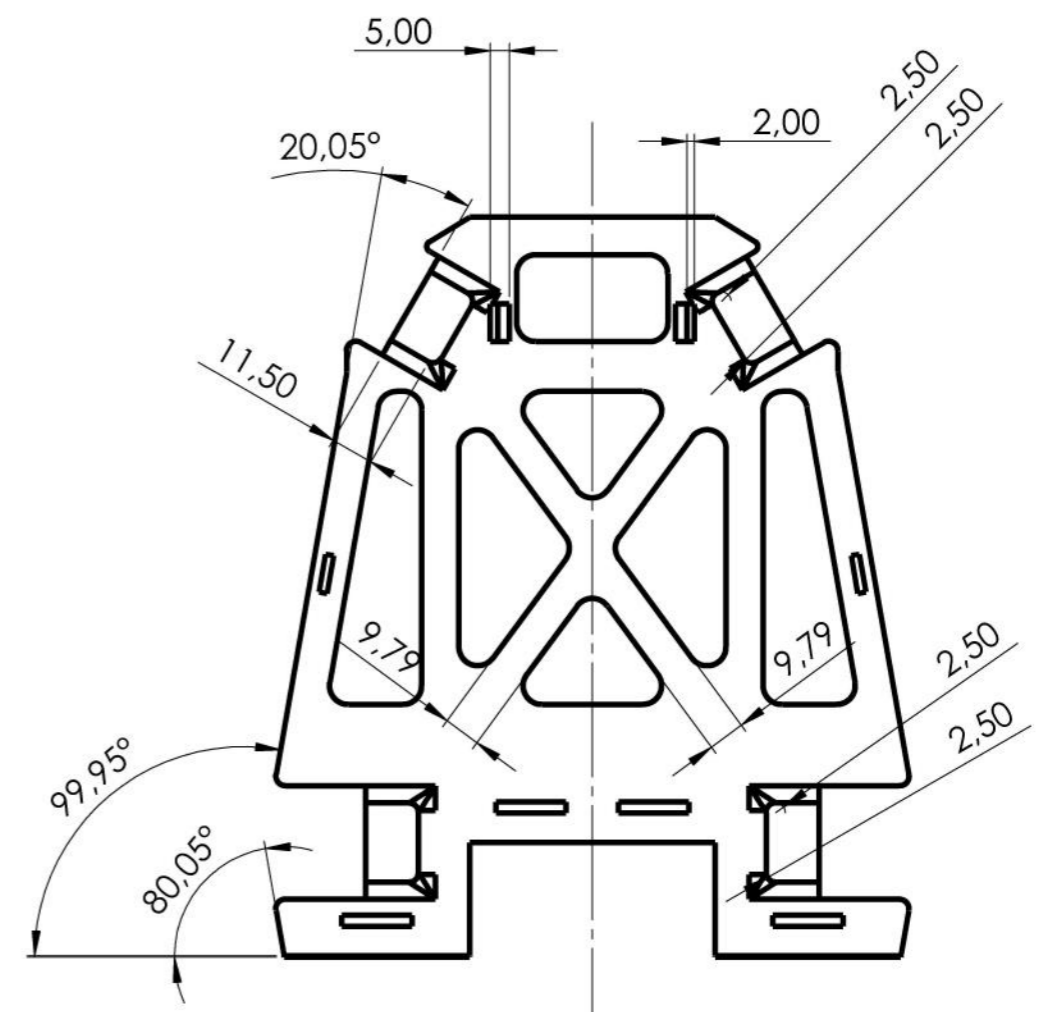
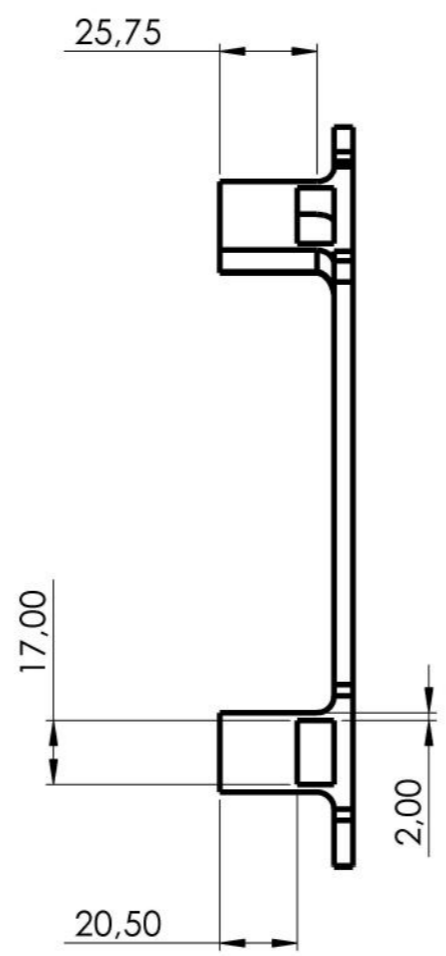
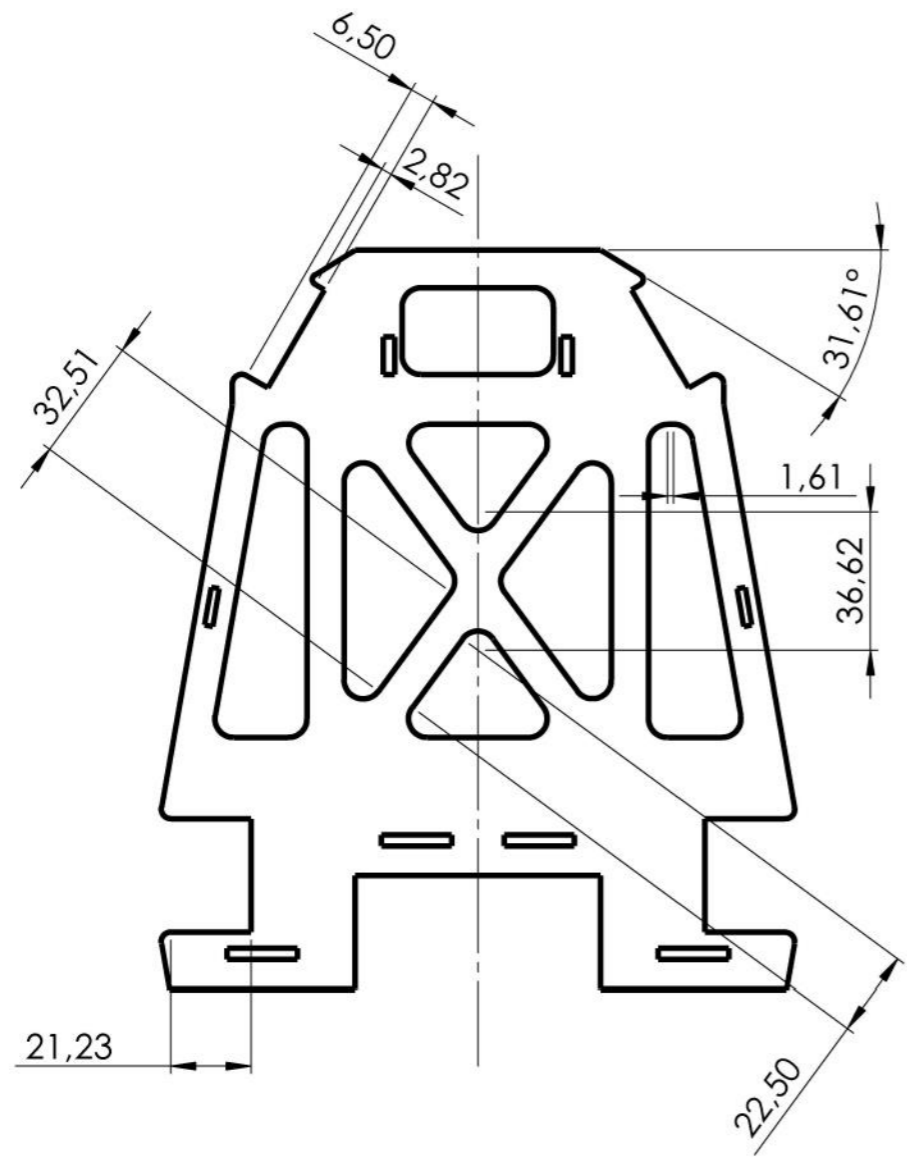
FIRMA: Alvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 2112
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 050520-1A	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: CHASIS INFERIOR TRASERO	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 1:2	Nº PLANO: 2.1.1.2
		FECHA DE EDICIÓN: 05/05/20	HOJA: 1/2
			A3



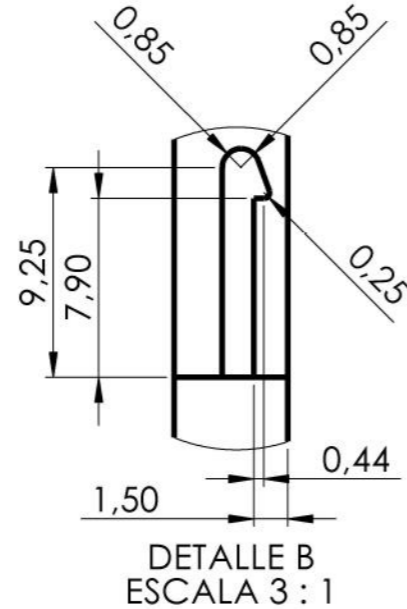
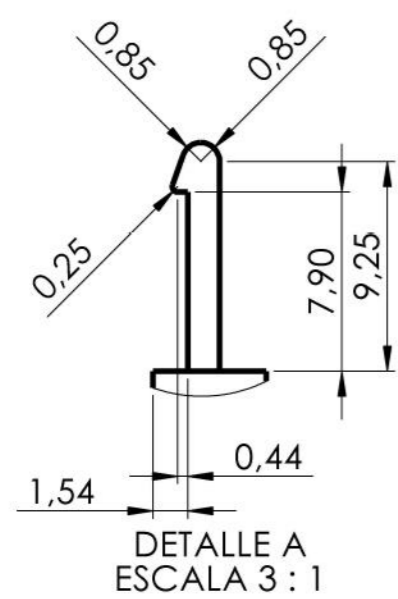
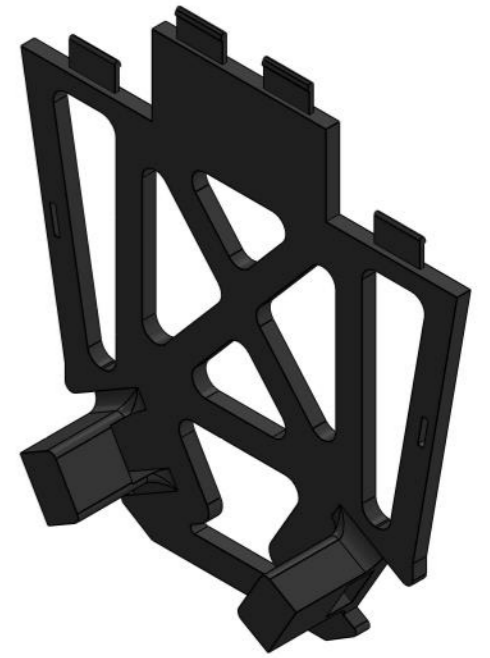
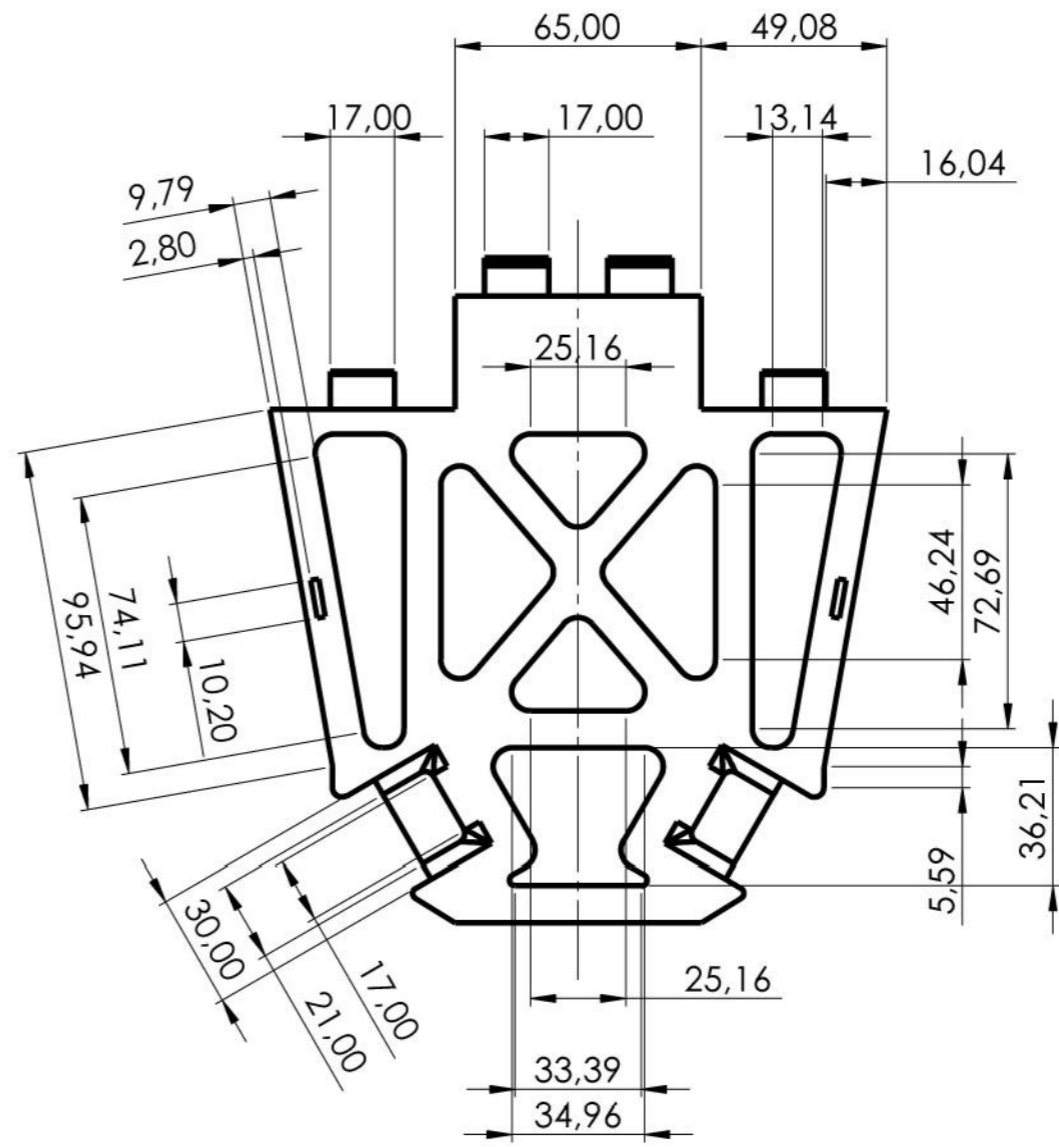
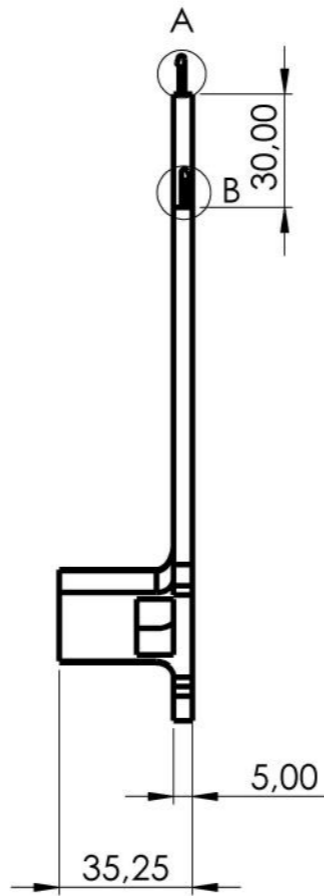
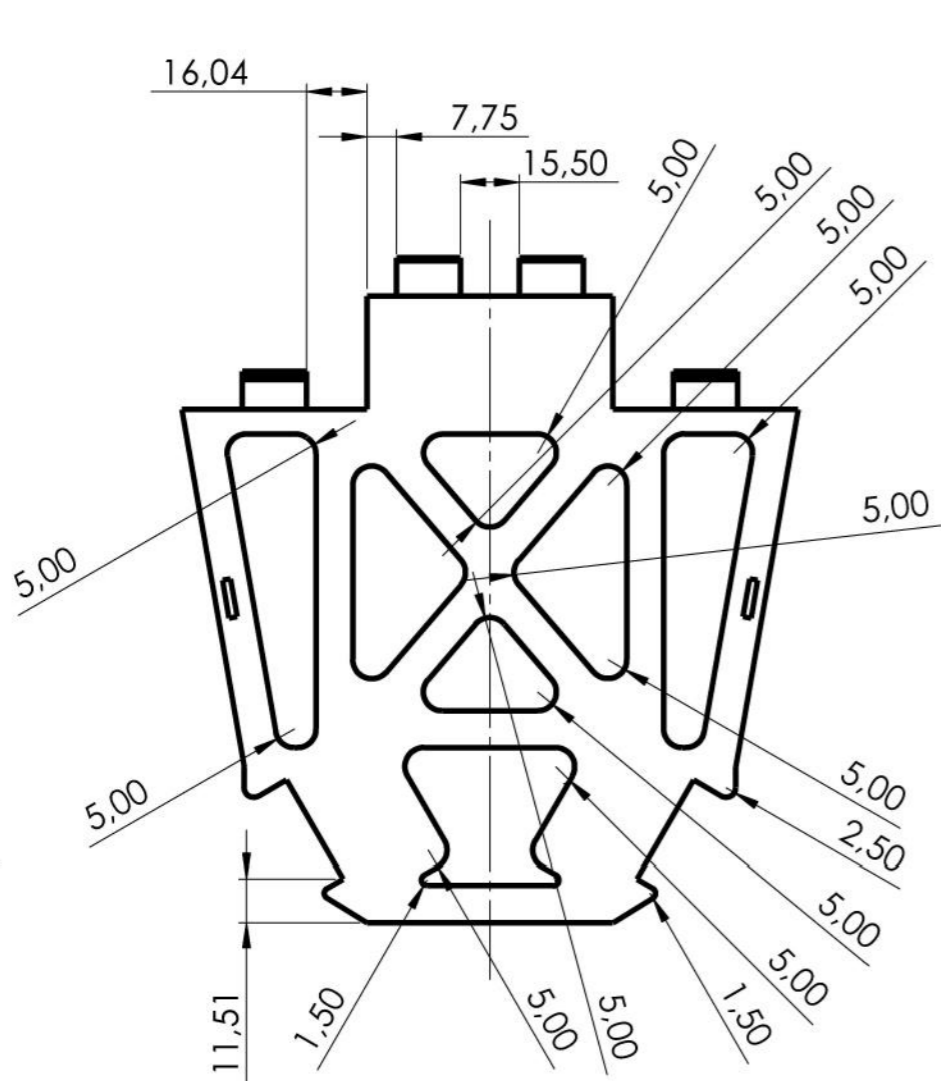
FIRMA: Álvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 2112
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 050520-1B	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: CHASIS INFERIOR TRASERO	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 1:2	Nº PLANO: 2.1.1.2
		FECHA DE EDICIÓN: 05/05/20	HOJA: 2/2
			A3



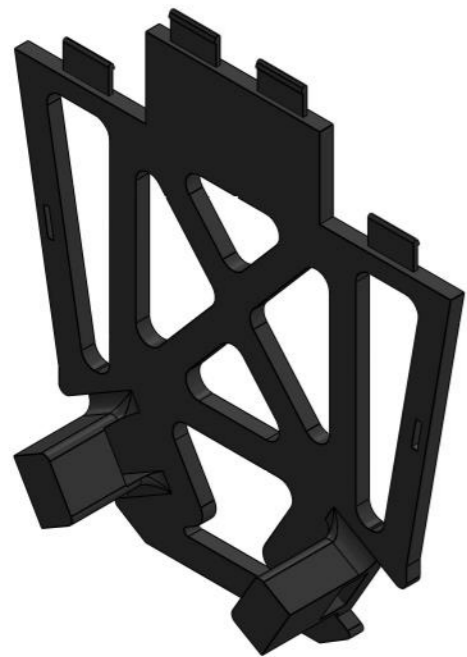
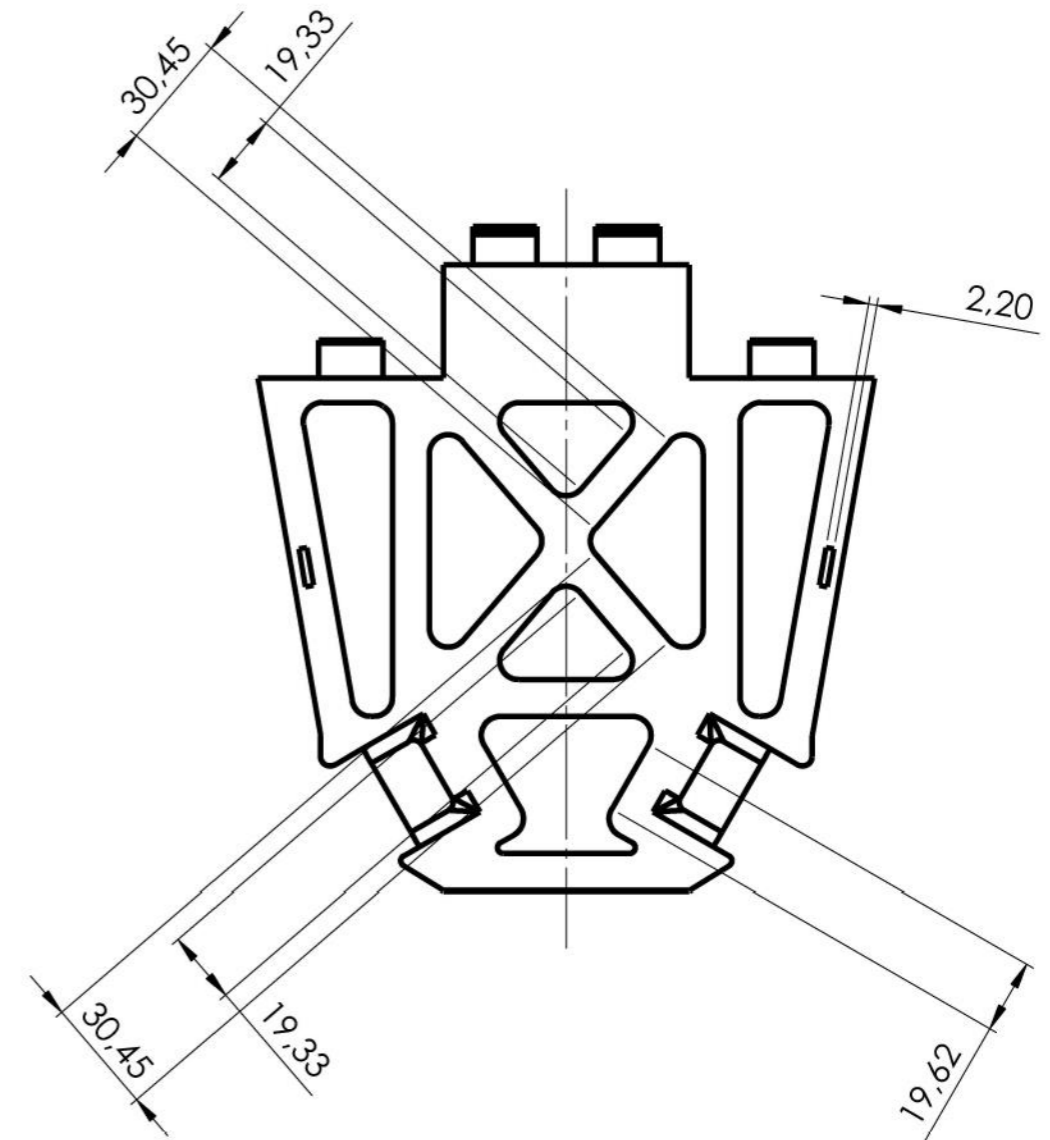
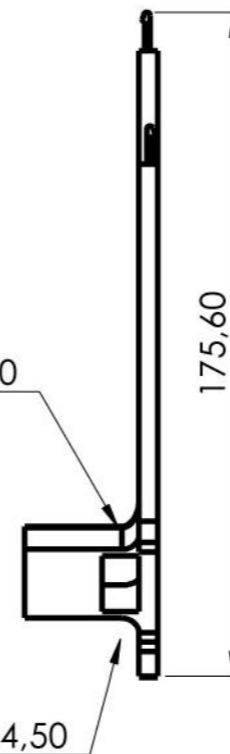
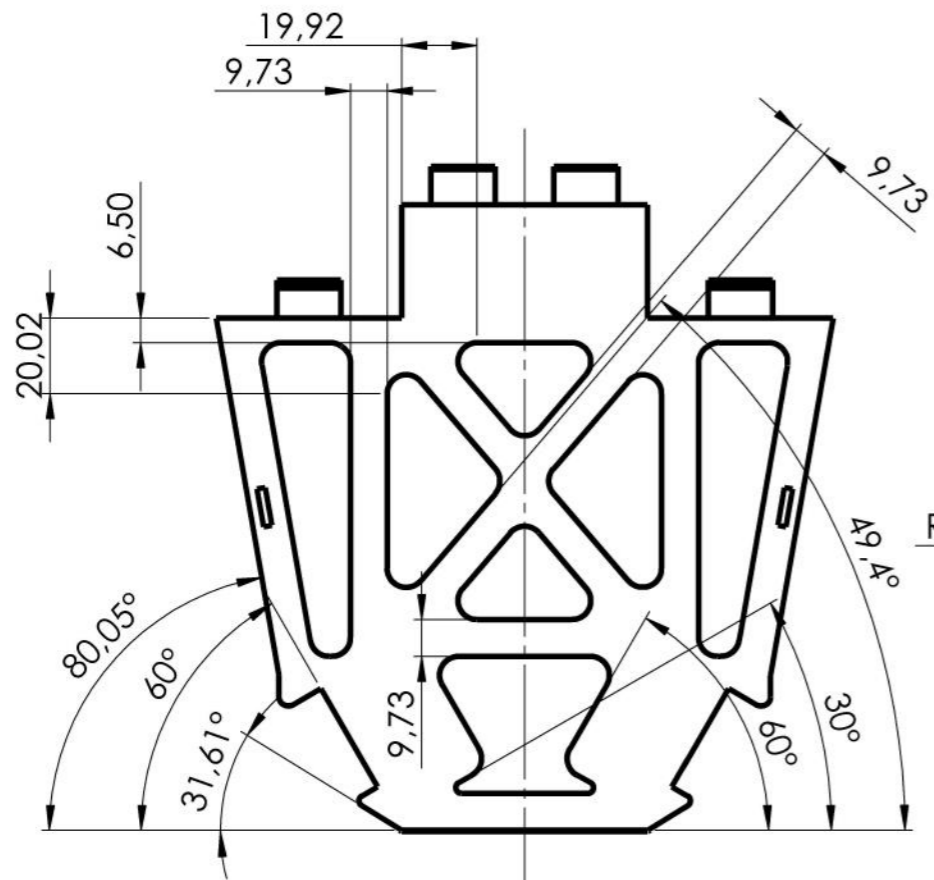
FIRMA: Alvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 2121
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 070520-1A	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: CHASIS SUPERIOR DELANTERO	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 1:2	Nº PLANO: 2.1.2.1
		FECHA DE EDICIÓN: 07/05/20	HOJA: 1/2
			A3



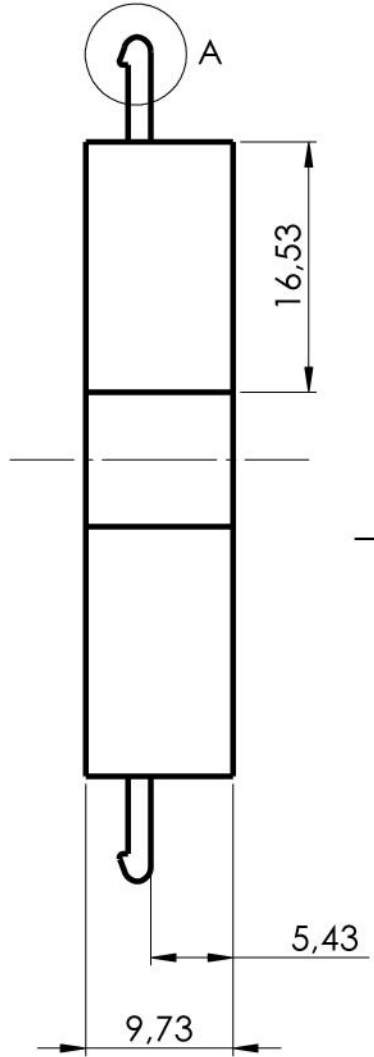
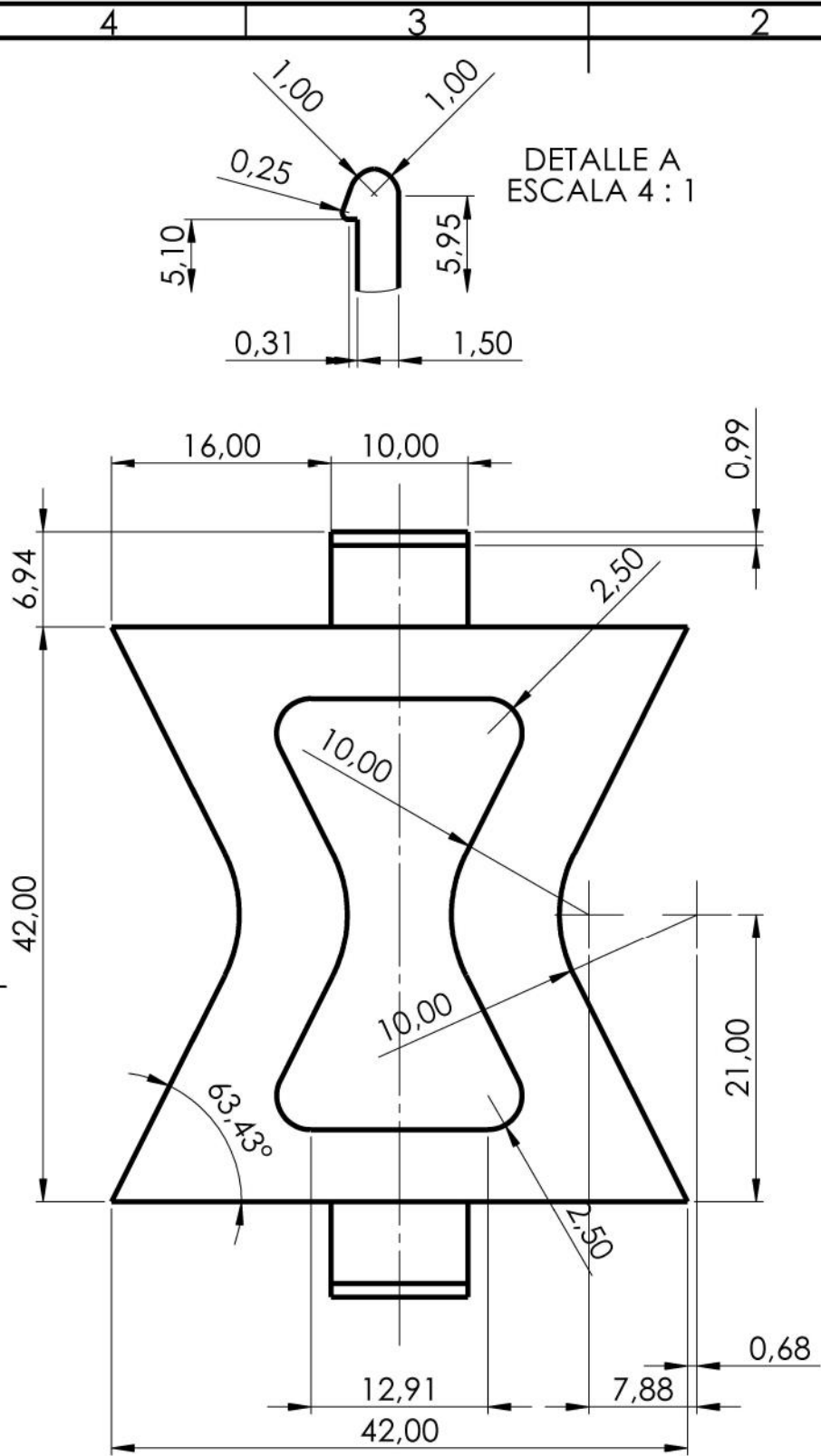
FIRMA: Álvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 2121
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 070520-1B	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: CHASIS SUPERIOR DELANTERO	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 1:2	Nº PLANO: 2.1.2.1
		FECHA DE EDICIÓN: 07/05/20	HOJA: 2/2
			A3



FIRMA: Alvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 2122
UNIVERSIDAD: UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 050520-2A	ORIENTACIÓN:
ESCUELA: Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: CHASIS SUPERIOR TRASERO	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 1:2	Nº PLANO: 2.1.2.2
		FECHA DE EDICIÓN: 05/05/20	HOJA: 1/2 A3



FIRMA: Alvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL:	ABS	MARCA:	2122				
		UNIVERSIDAD:	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA	REFERENCIA:	050520-2B	ORIENTACIÓN:			
ESCUELA:	 Escuela Técnica Superior de Ingeniería del Diseño	TIPO DE DOCUMENTO:	Dibujo de subconjunto		CREADO POR:	HEXAPUS	REVISIÓN:	A	
		TÍTULO:	CHASIS SUPERIOR TRASERO		ESCALA:	1:2	Nº PLANO:	2.1.2.2	
					FECHA DE EDICIÓN:	05/05/20	HOJA:	2/2	A3



FIRMA:
Alvaro Alberto Giner



NOMBRE DEL PROYECTO:
Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil

TIPO DE DOCUMENTO:
Dibujo de subconjunto

TÍTULO:
PILAR

MATERIAL:
ABS

REFERENCIA:
130420-8

CREADO POR:
HEXAPUS

ESCALA:
2:1

FECHA DE EDICIÓN:
13/04/20

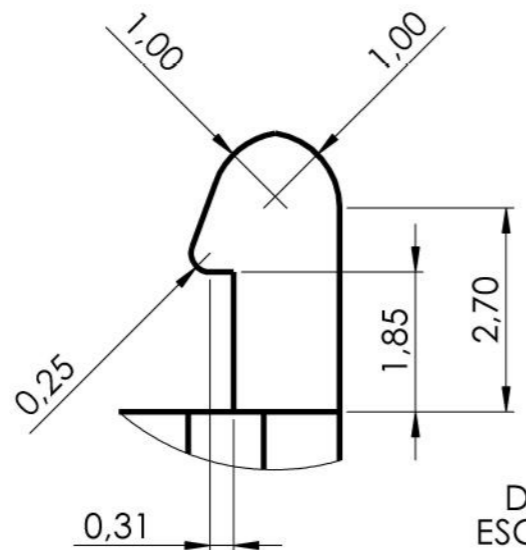
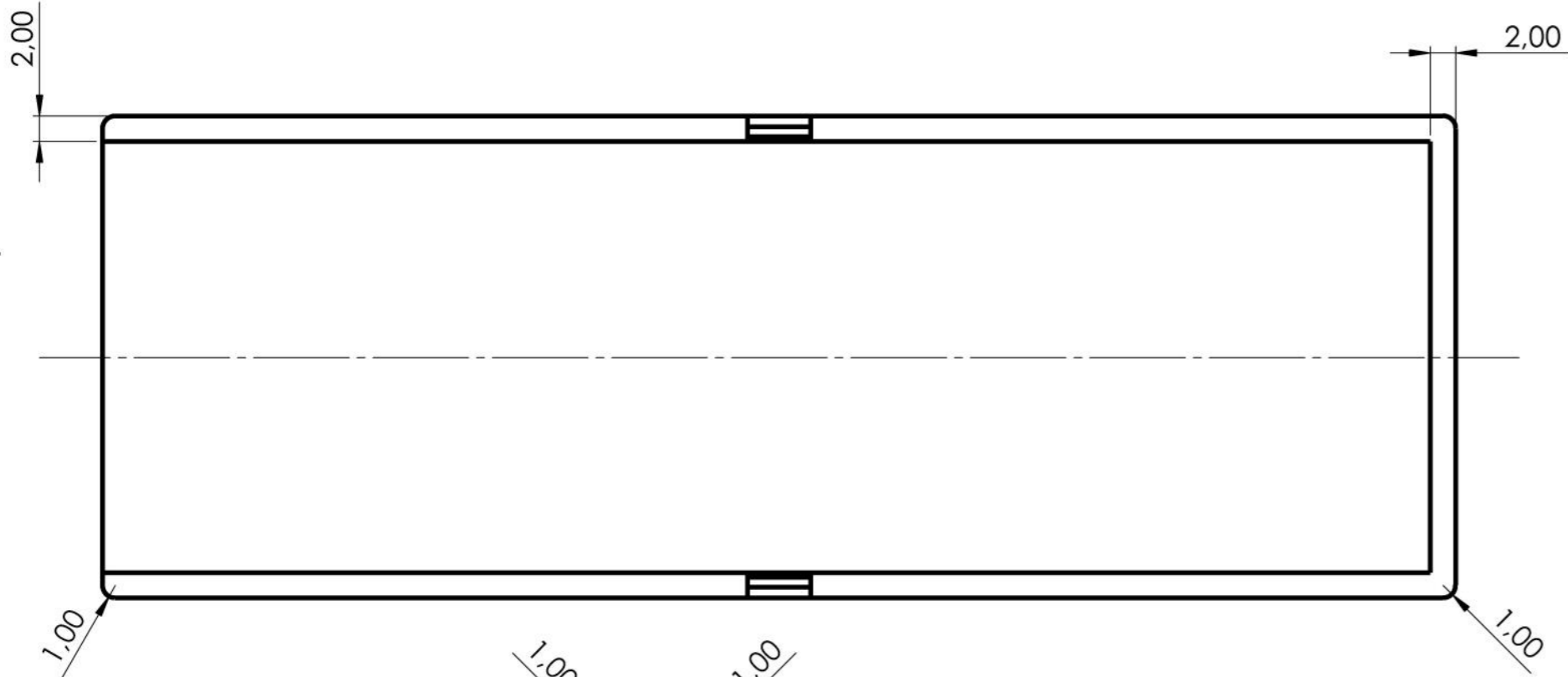
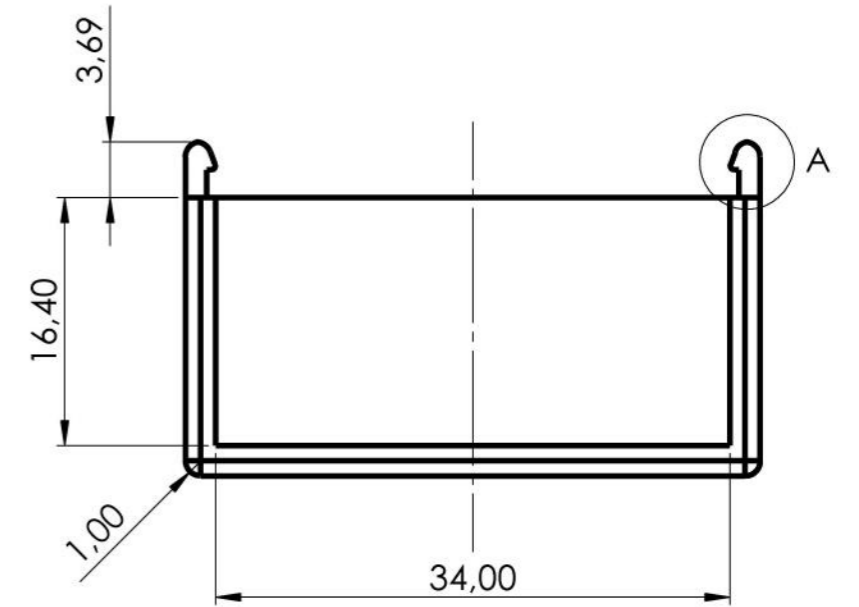
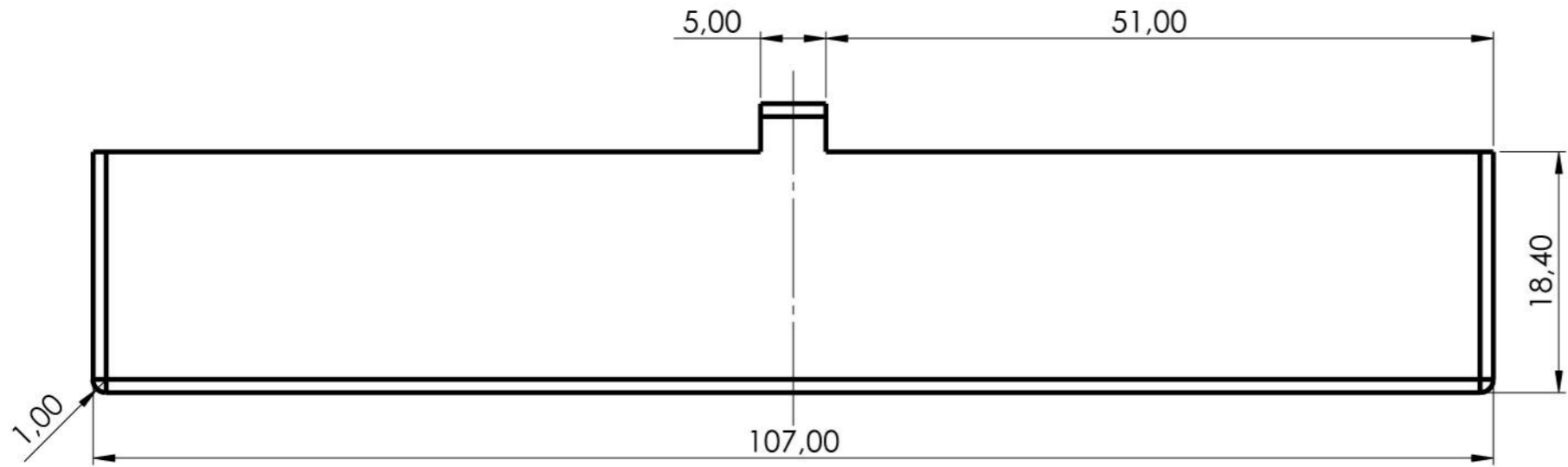
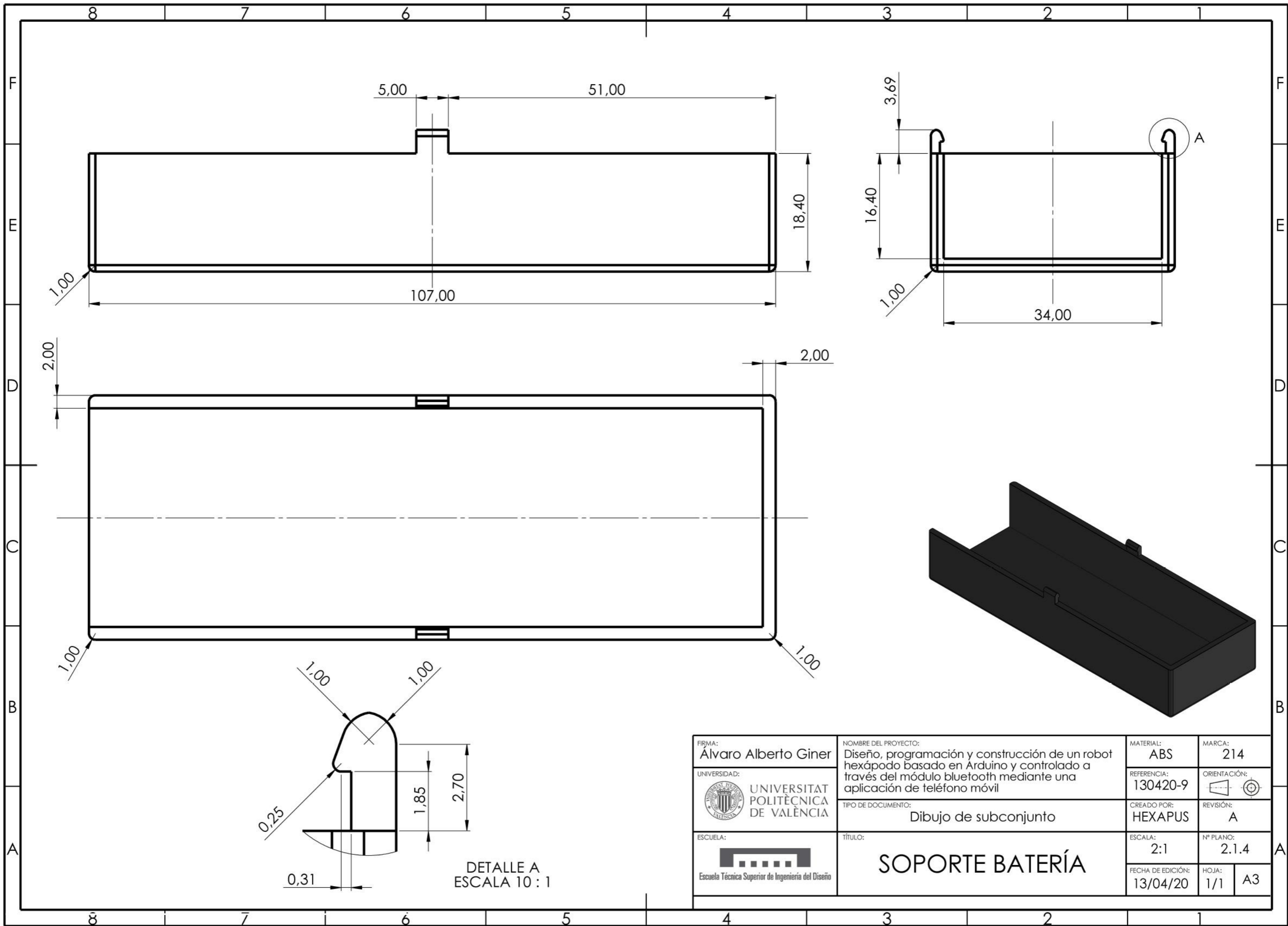
MARCA:
213



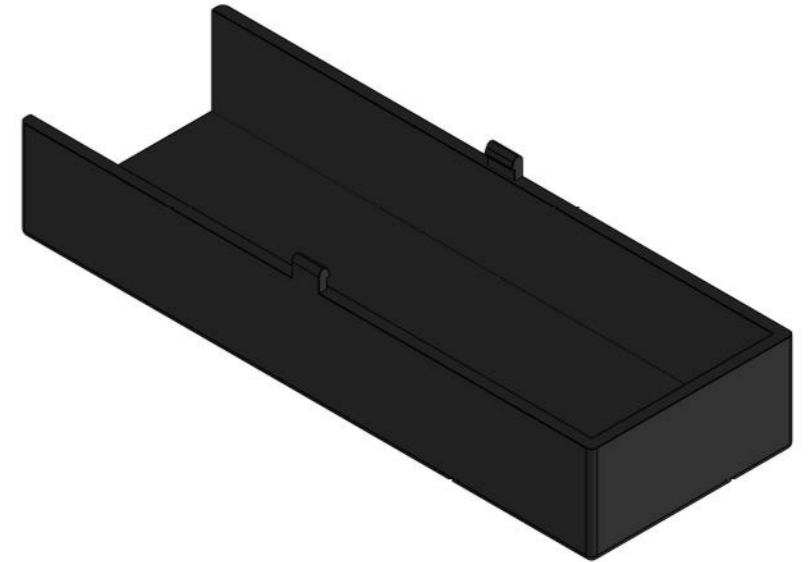
REVISIÓN:
A

Nº PLANO:
2.1.3

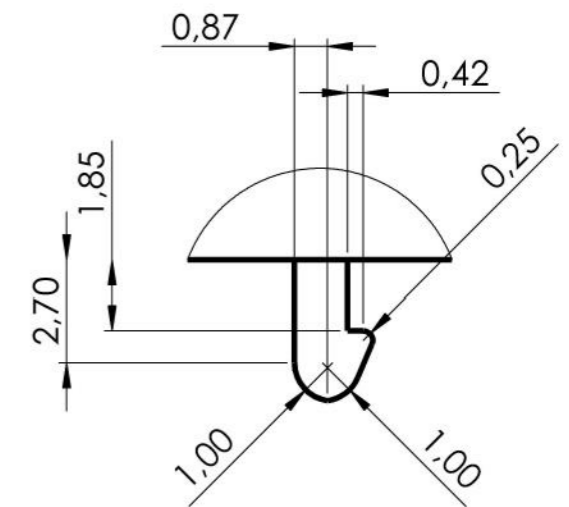
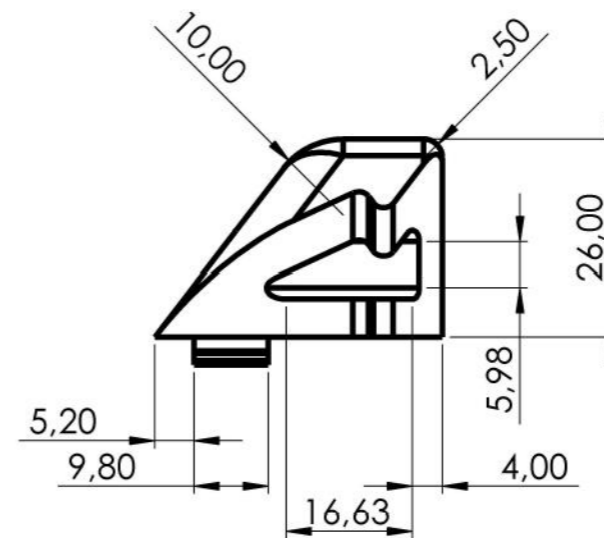
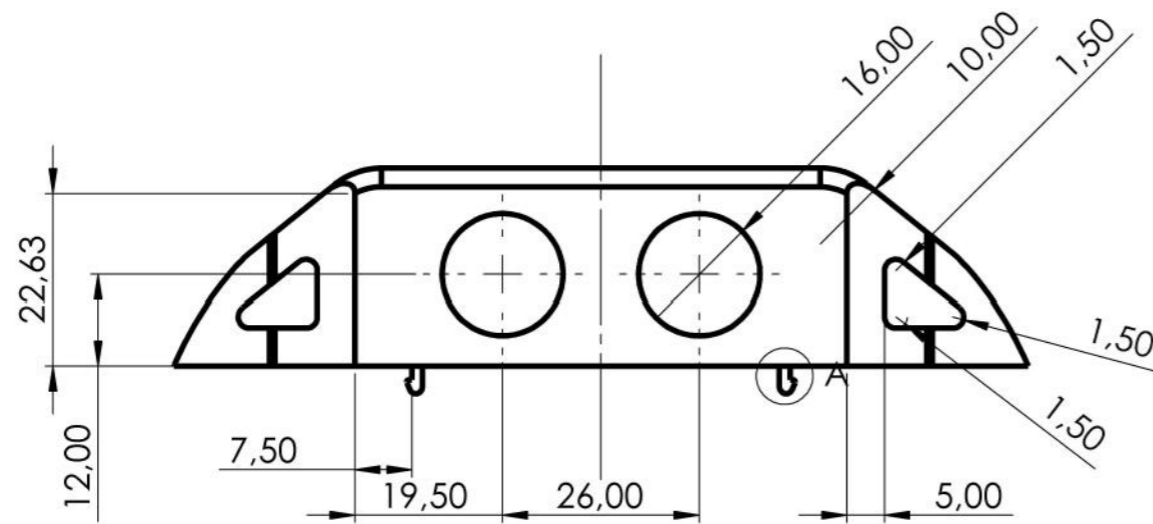
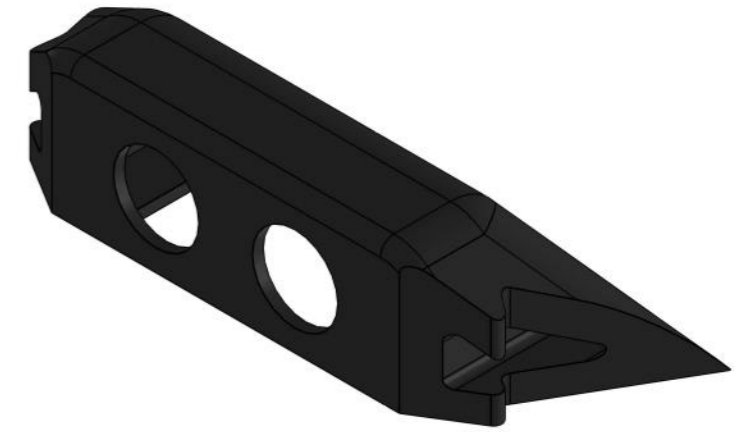
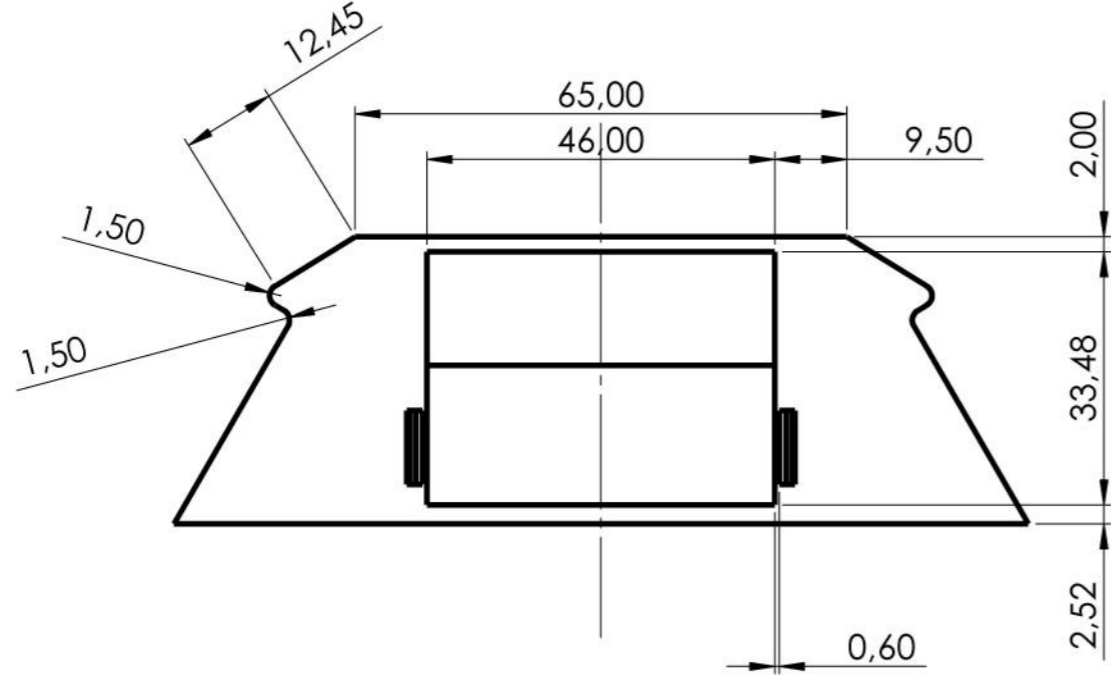
HOJA:
1/1 A4



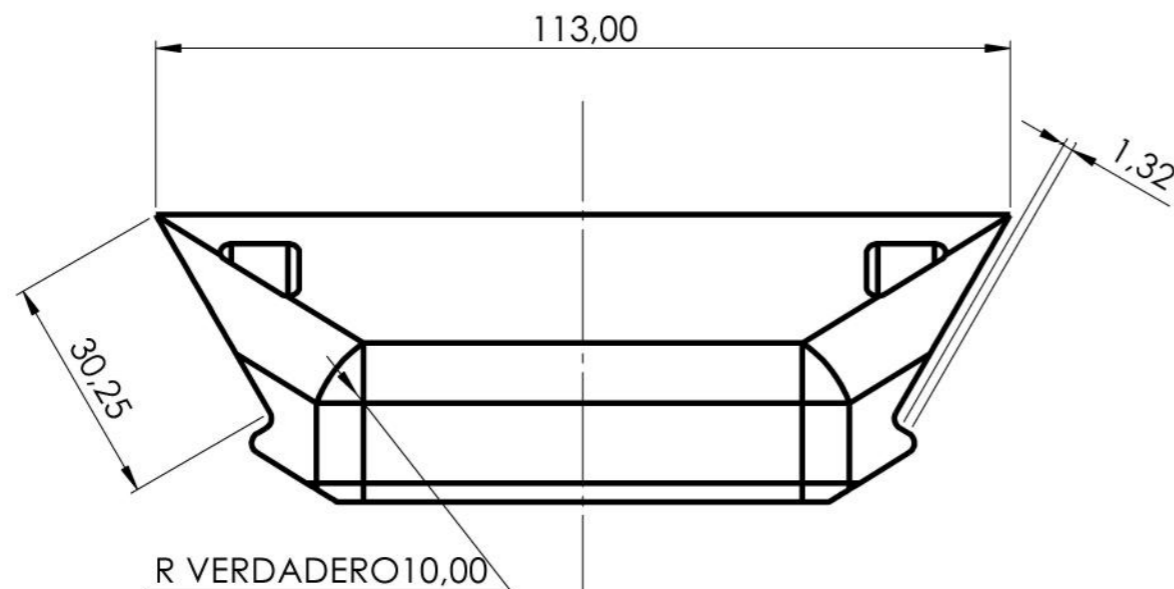
DETALLE A
ESCALA 10 : 1



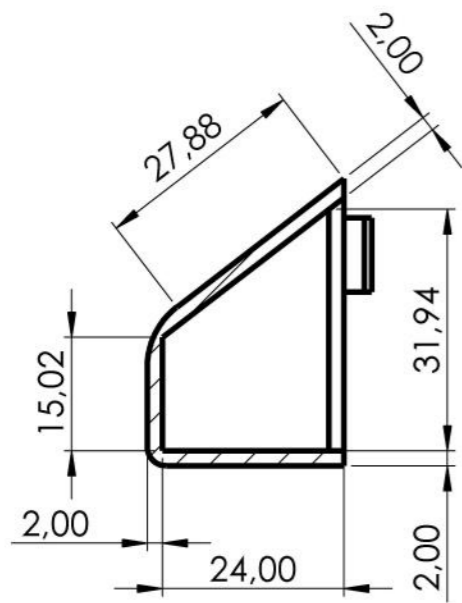
FIRMA: Álvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 214
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 130420-9	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: SOPORTE BATERÍA	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 2:1	Nº PLANO: 2.1.4
		FECHA DE EDICIÓN: 13/04/20	HOJA: 1/1 A3



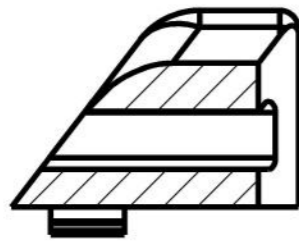
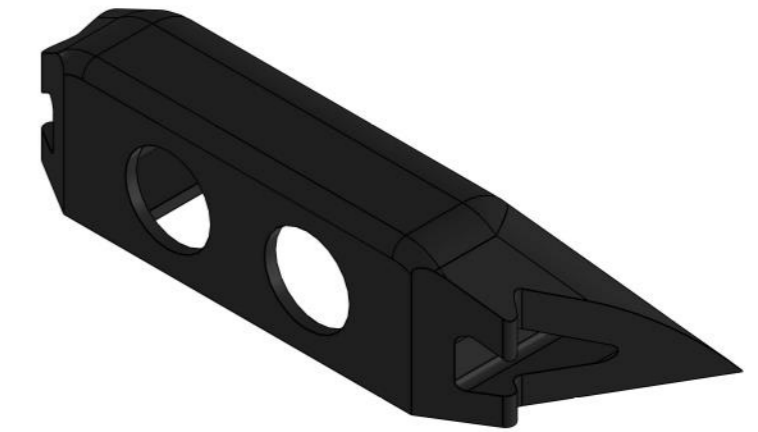
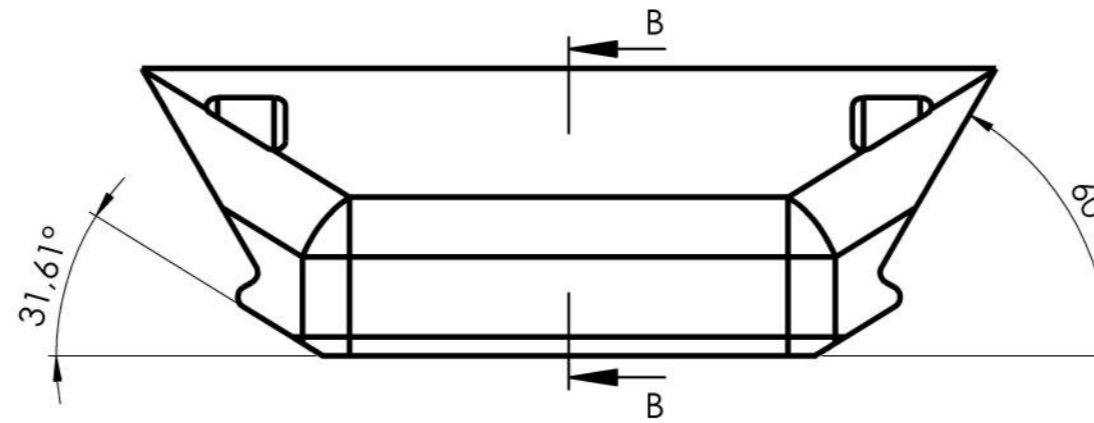
DETALLE A
ESCALA 5 : 1



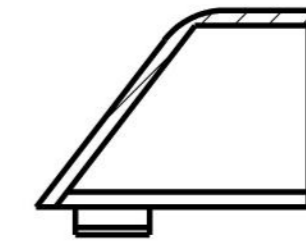
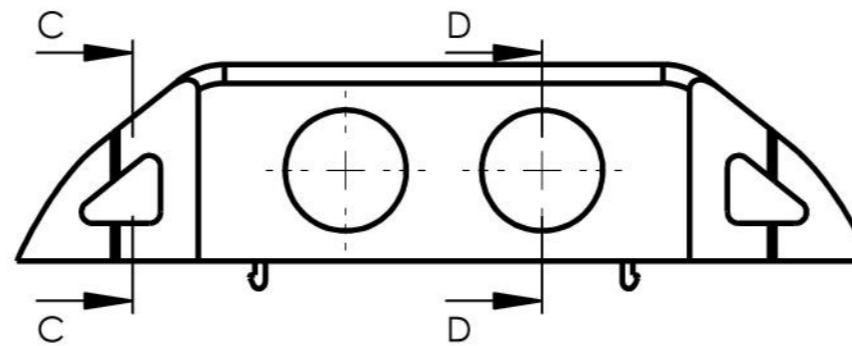
FIRMA: Álvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 215
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 080520-1A	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: SOPORTE ULTRASONIDOS	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 1:1	Nº PLANO: 2.1.5
		FECHA DE EDICIÓN: 08/05/20	HOJA: 1/2
			A3



SECCIÓN B-B

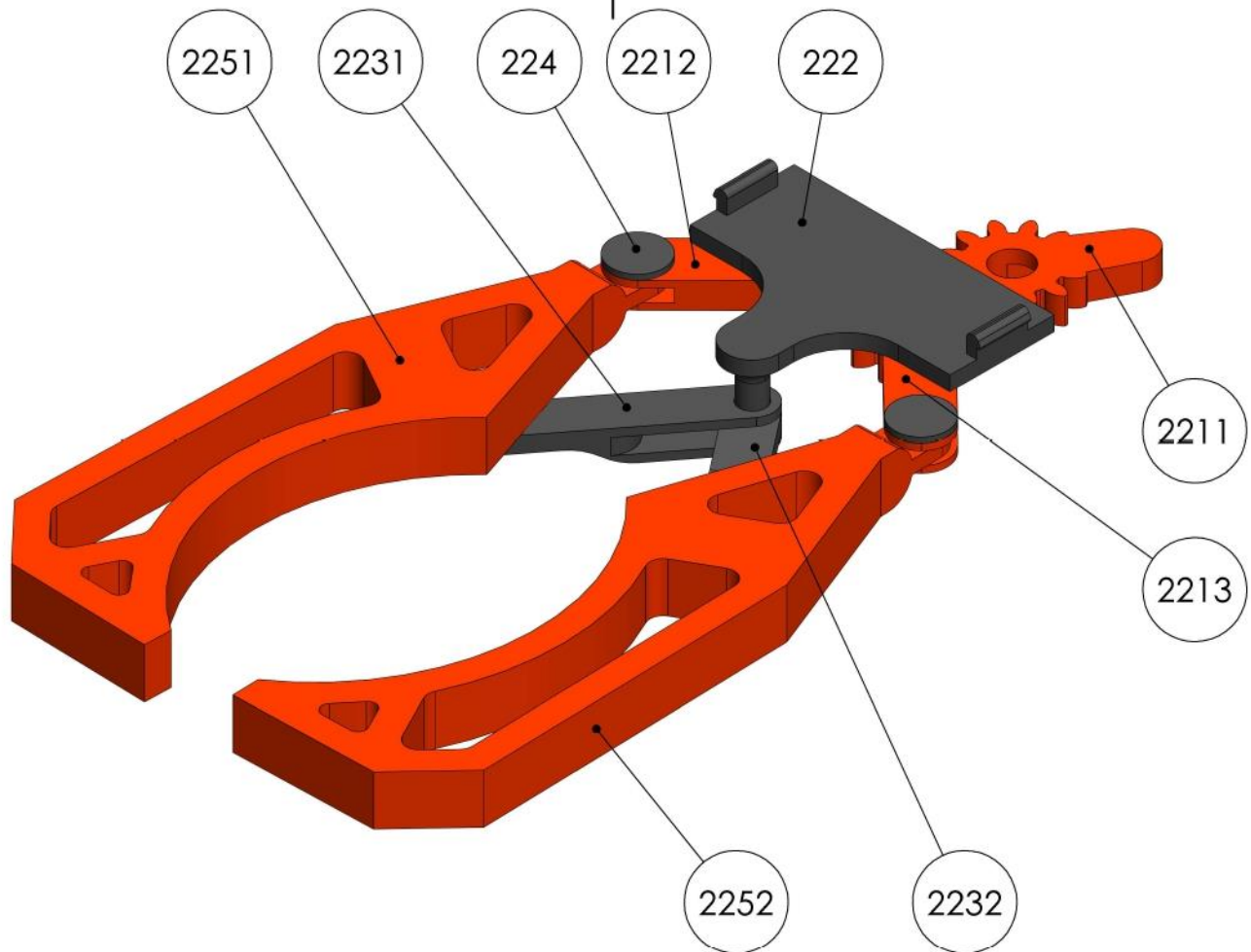


SECCIÓN C-C


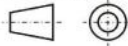



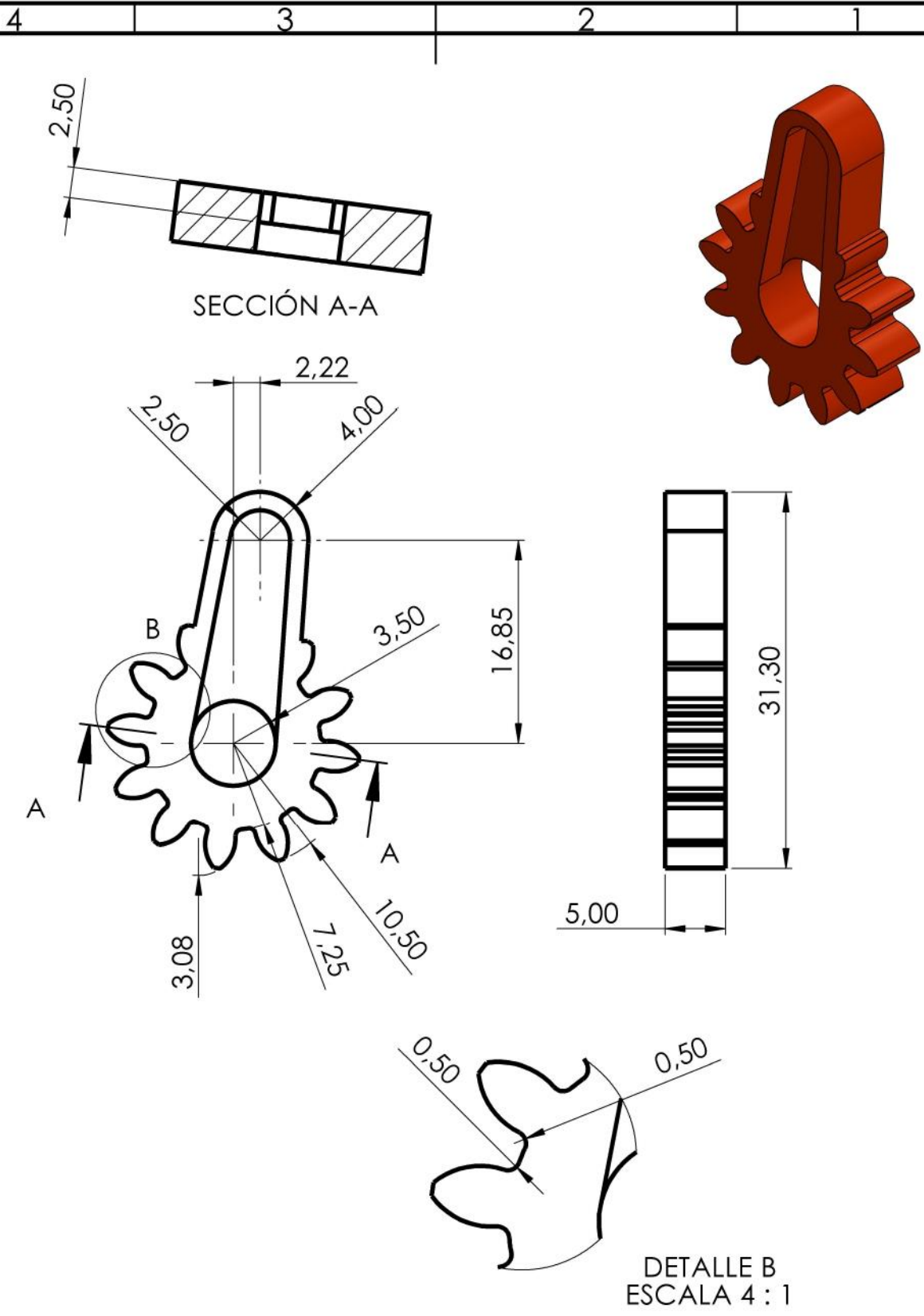
SECCIÓN D-D

FIRMA: Álvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL:	ABS	MARCA:	215
		REFERENCIA:	080520-1B	ORIENTACIÓN:	
UNIVERSIDAD: UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	CREADO POR:	HEXAPUS	REVISIÓN:	A
		ESCALA:	1:1	Nº PLANO:	2.1.5
ESCUELA: Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: SOPORTE ULTRASONIDOS	FECHA DE EDICIÓN:	08/05/20	HOJA:	2/2
					A3



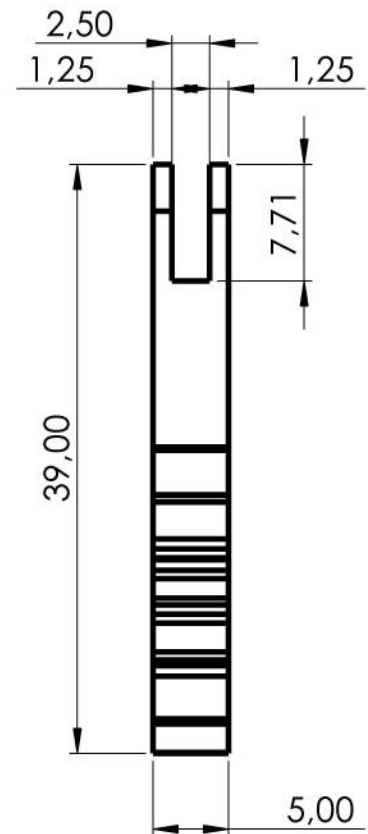
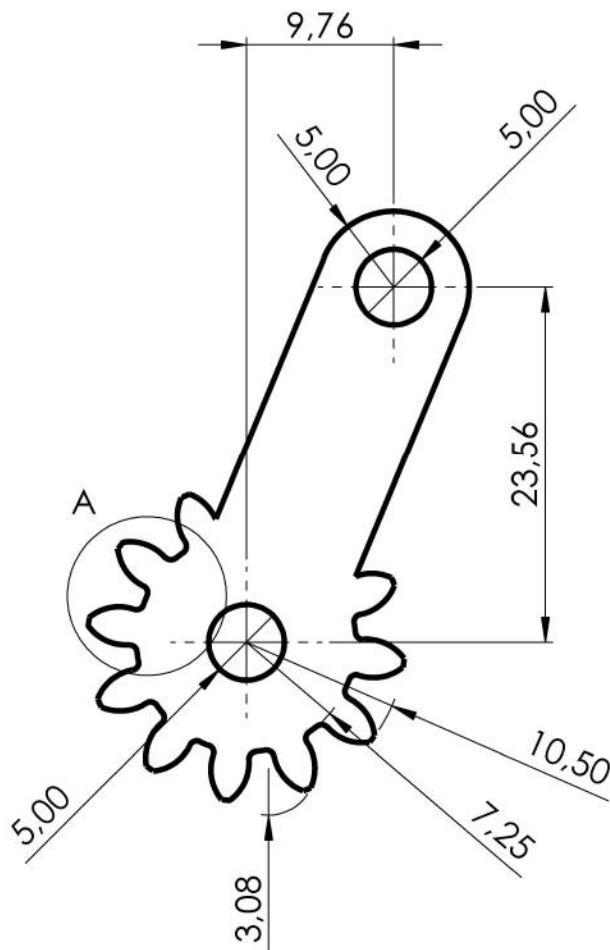
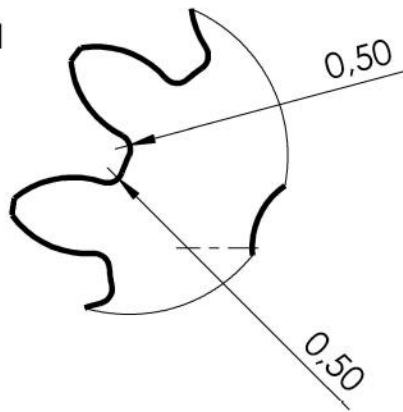
2252	1	Mandíbula Izquierda	UNE-EN ISO 15015	ABS
2251	1	Mandíbula Derecha	UNE-EN ISO 15015	ABS
224	1	Bulón	UNE-EN ISO 15015	ABS
2232	1	Estabilizador Izquierdo	UNE-EN ISO 15015	ABS
2231	1	Estabilizador Derecho	UNE-EN ISO 15015	ABS
222	1	Soporte Boca	UNE-EN ISO 15015	ABS
2213	1	Engranaje Izquierdo	UNE-EN ISO 15015	ABS
2212	1	Engranaje Derecho	UNE-EN ISO 15015	ABS
2211	1	Engranaje Motriz	UNE-EN ISO 15015	ABS
Marca	Nº Pieza	Designación	Norma	Material

FIRMA: Álvaro Alberto Giner		NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil		MATERIAL: ABS		MARCA: 2	
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA		TIPO DE DOCUMENTO: Vista General de Subconjunto		REFERENCIA: 120420-8		ORIENTACIÓN: 	
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño				TÍTULO: <h1 style="text-align: center;">BOCA</h1>		CREADO POR: HEXAPUS	
				ESCALA: 1:1		Nº PLANO: 2.2	
				FECHA DE EDICIÓN: 12/04/20		HOJA: 1/1	
						A4	



FIRMA: Álvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 2211
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 130420-5	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: ENGRANAJE MOTRIZ	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 2:1	Nº PLANO: 2.2.1.1
		FECHA DE EDICIÓN: 13/04/20	HOJA: 1/1 A4

DETALLE A
ESCALA 4 : 1



FIRMA:
Álvaro Alberto Giner

UNIVERSIDAD:



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

ESCUELA:



Escuela Técnica Superior de Ingeniería del Diseño

NOMBRE DEL PROYECTO:

Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil

TIPO DE DOCUMENTO:

Dibujo de subconjunto

TÍTULO:

ENGRANAJE DERECHO

MATERIAL:

ABS

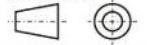
MARCA:

2212

REFERENCIA:

130420-6

ORIENTACIÓN:



CREADO POR:

HEXAPUS

REVISIÓN:

A

ESCALA:

2:1

Nº PLANO:

2.2.1.2

FECHA DE EDICIÓN:

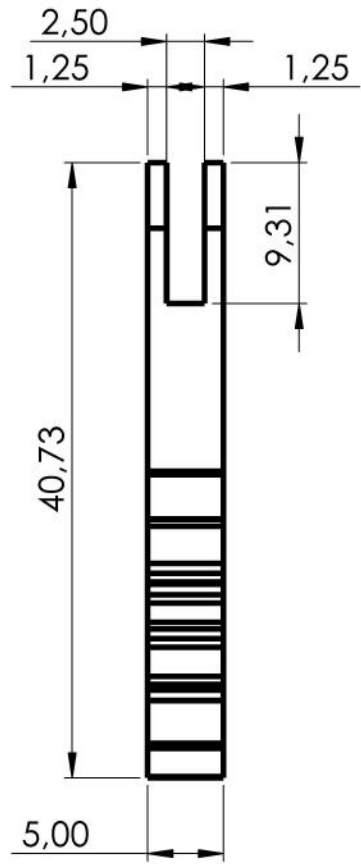
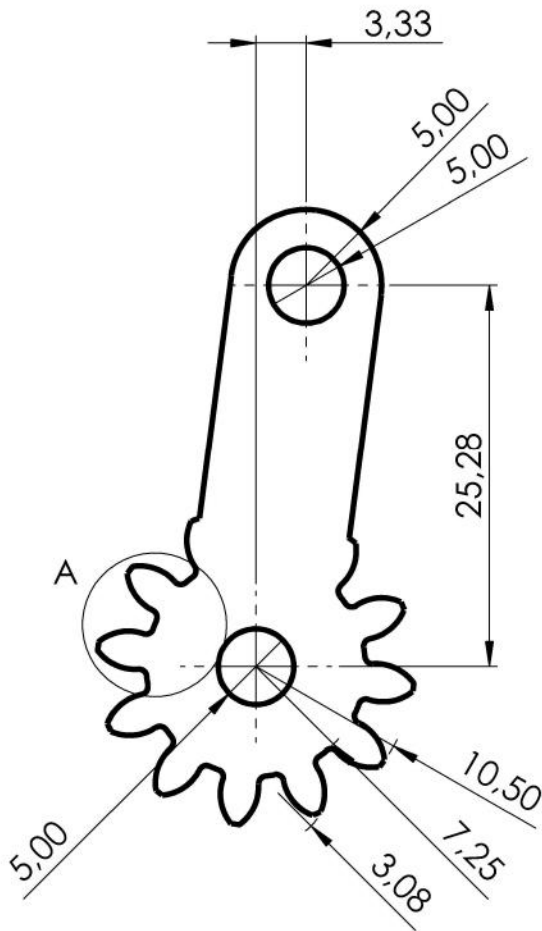
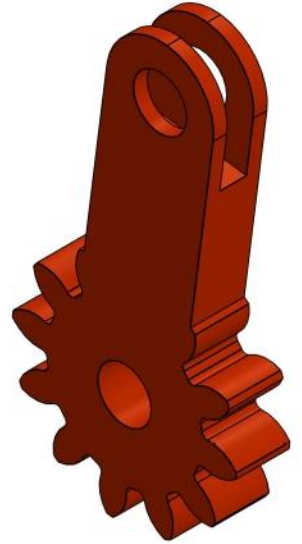
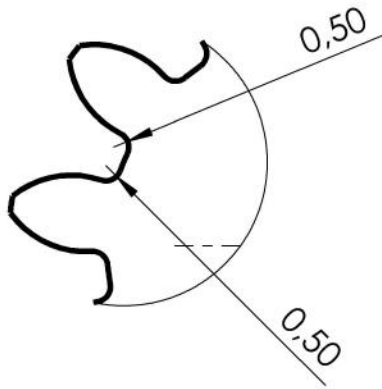
13/04/20

HOJA:

1/1

A4

DETALLE A
ESCALA 4 : 1



FIRMA:
Álvaro Alberto Giner



NOMBRE DEL PROYECTO:
Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil

TIPO DE DOCUMENTO:
Dibujo de subconjunto

TÍTULO:
ENGRANAJE IZQUIERDO

MATERIAL:
ABS

MARCA:
2213

REFERENCIA:
130420-7

ORIENTACIÓN:

CREADO POR:
HEXAPUS

REVISIÓN:
A

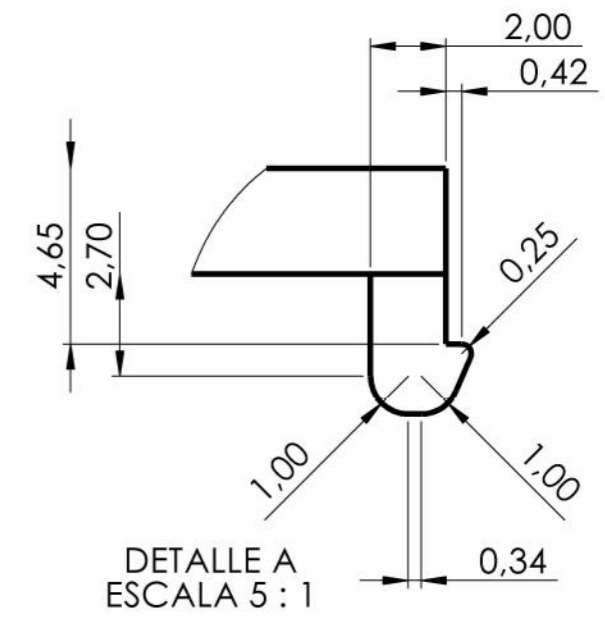
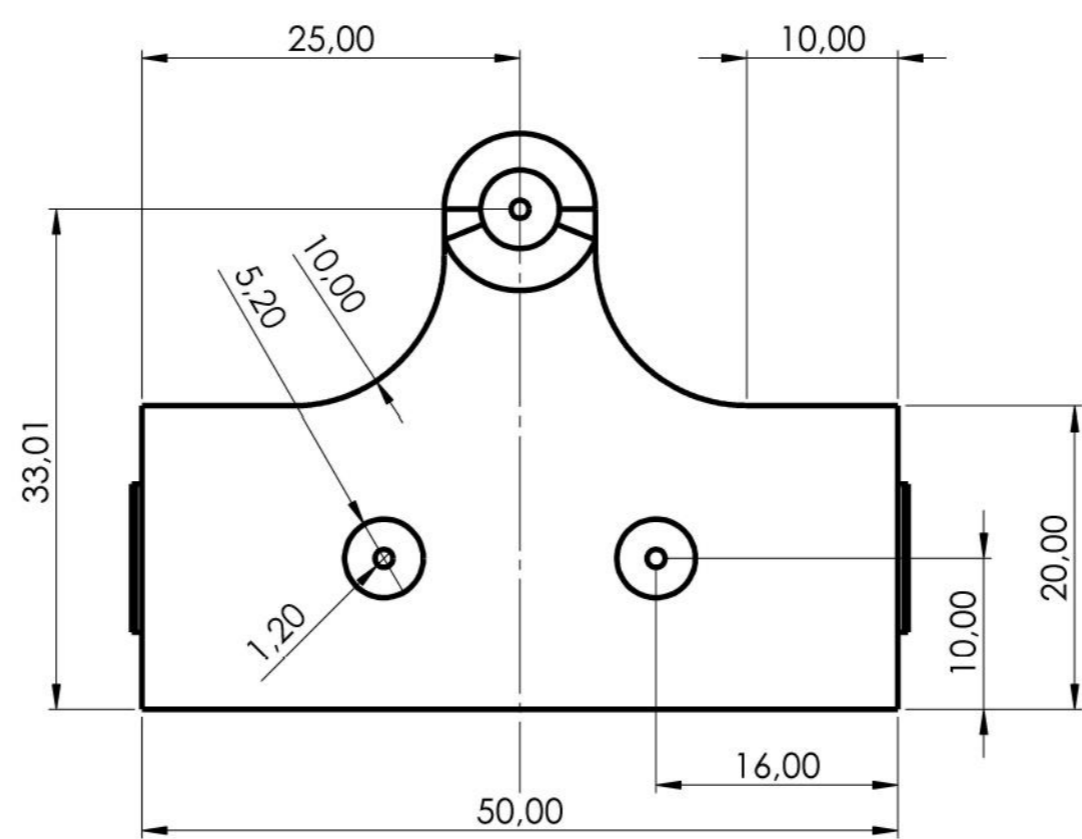
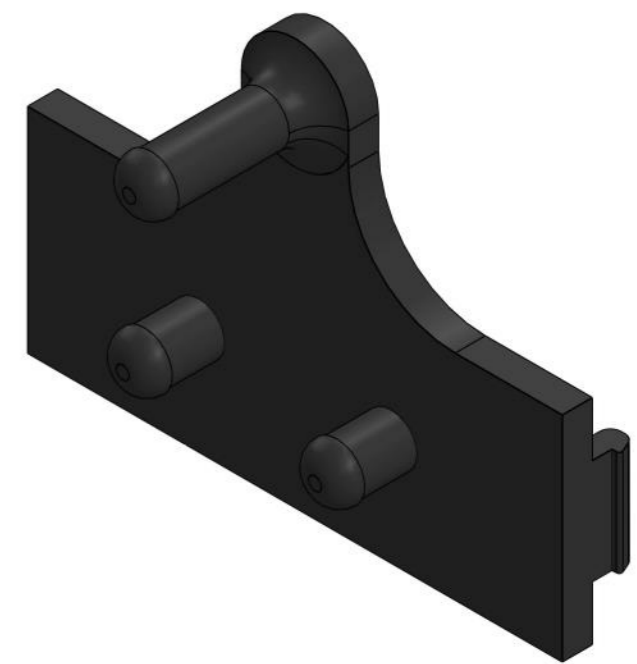
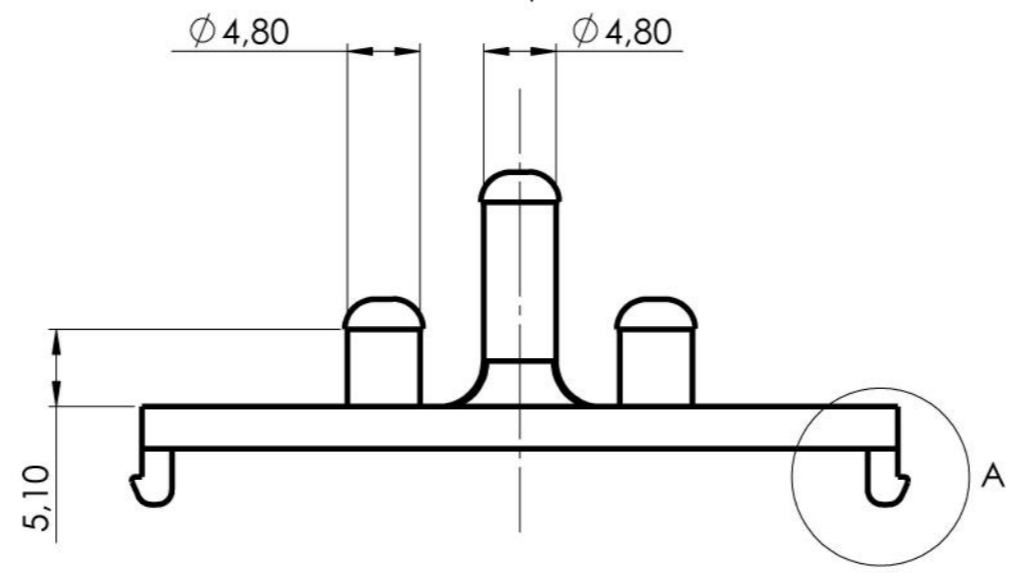
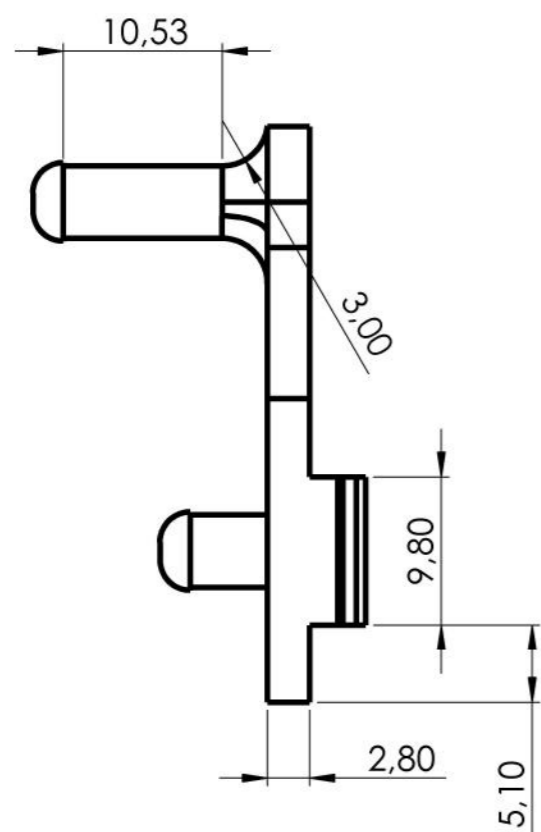
ESCALA:
2:1

Nº PLANO:
2.2.1.3

FECHA DE EDICIÓN:
13/04/20

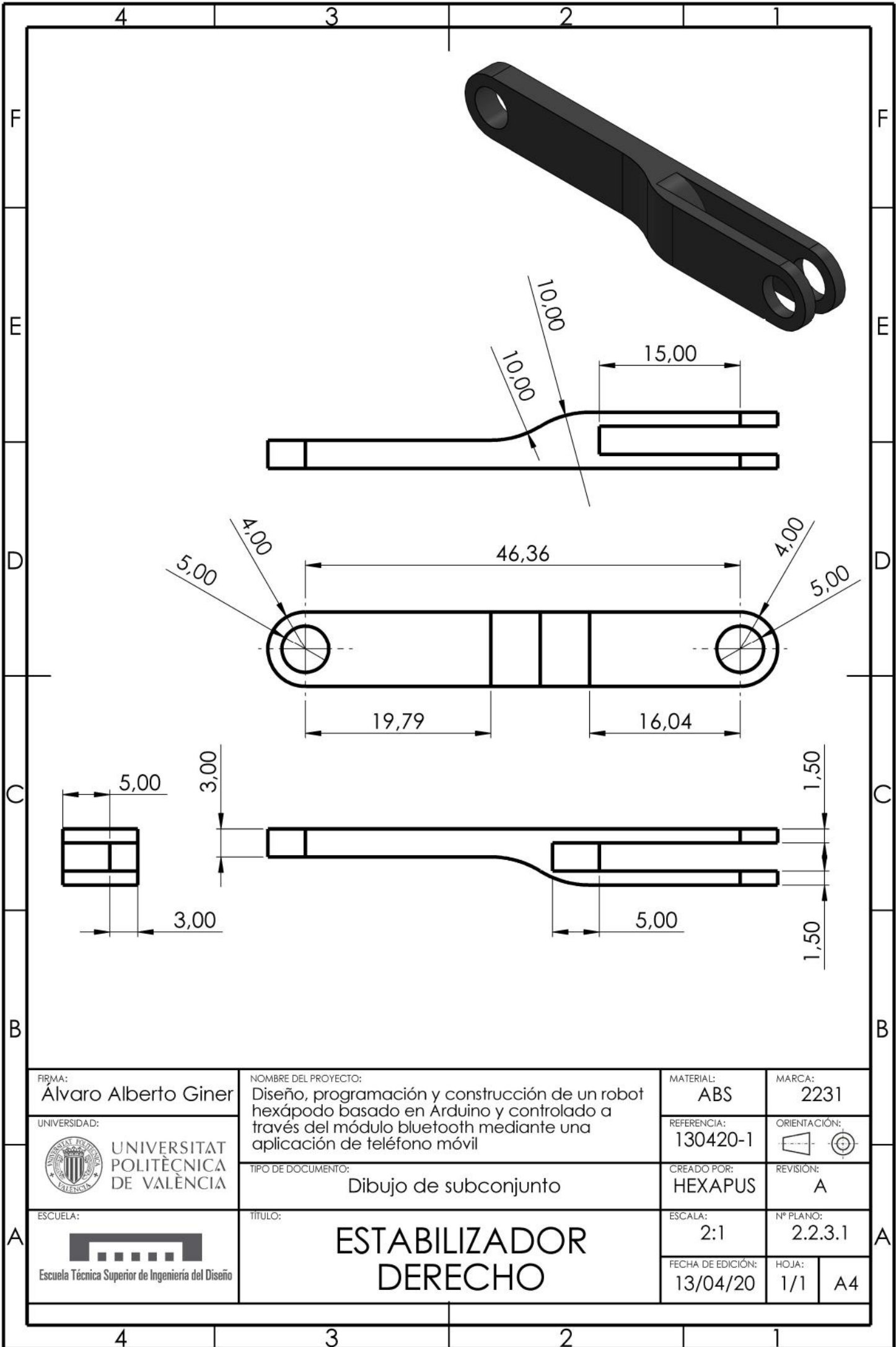
HOJA:
1/1

A4

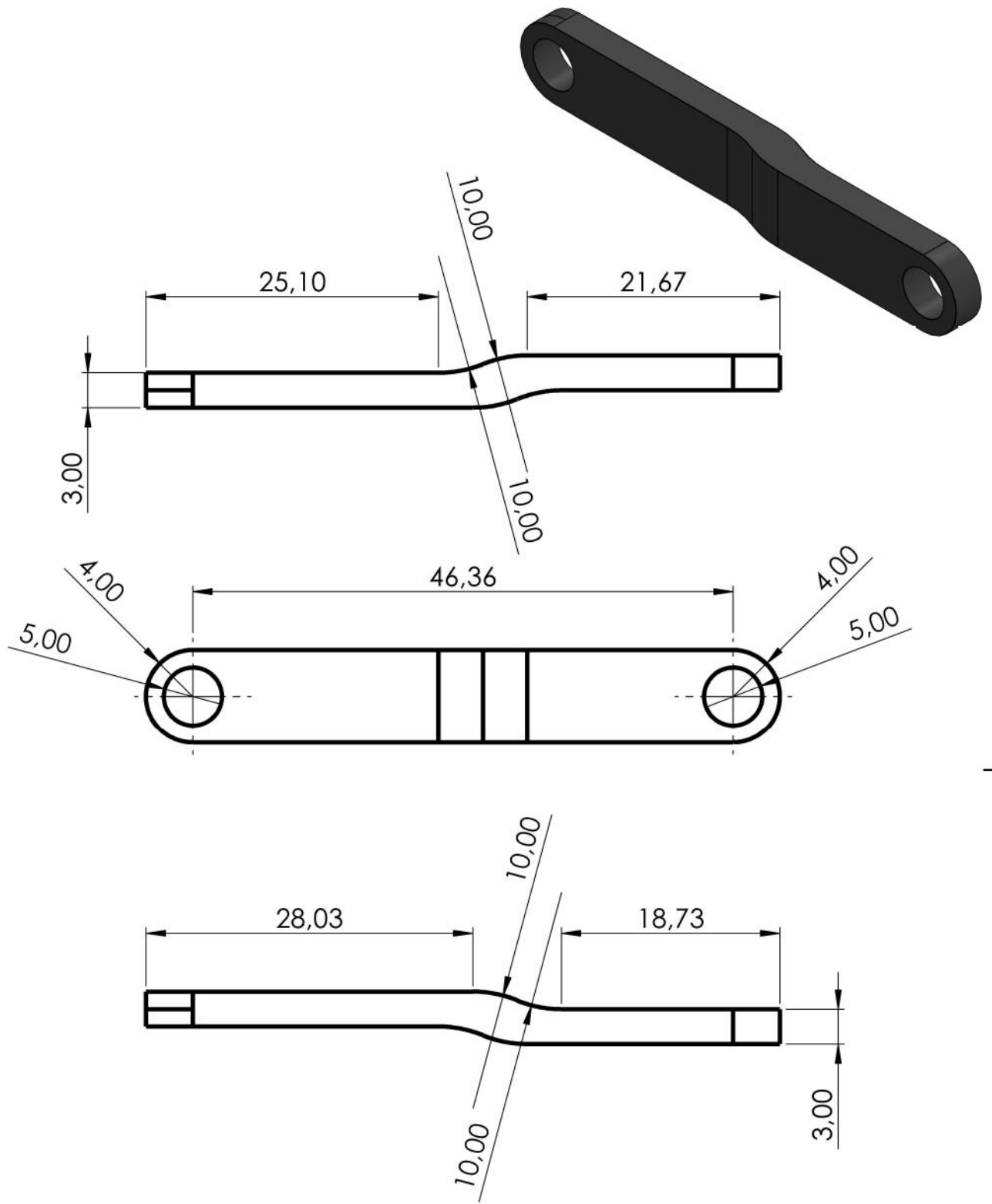


DETALLE A
ESCALA 5 : 1

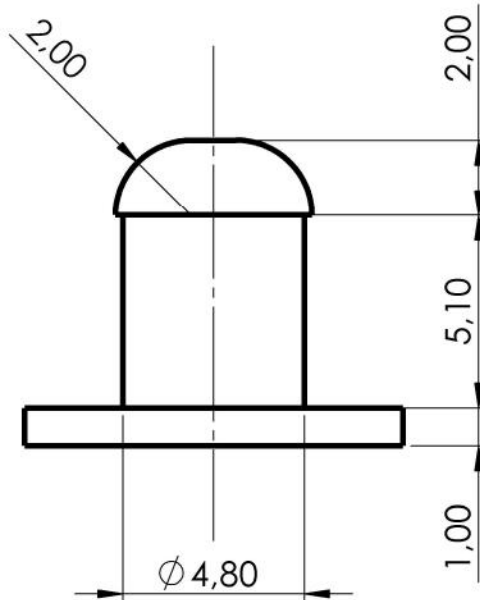
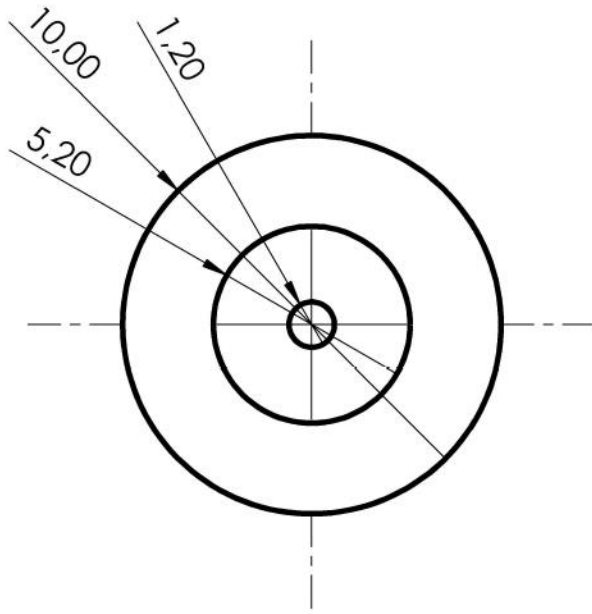
FIRMA: Alvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 222
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 130420-4	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: SOPORTE BOCA	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 2:1	Nº PLANO: 2.2.2
		FECHA DE EDICIÓN: 13/04/20	HOJA: 1/1 A3



FIRMA: Álvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 2231
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 130420-1	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: ESTABILIZADOR DERECHO	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 2:1	Nº PLANO: 2.2.3.1
		FECHA DE EDICIÓN: 13/04/20	HOJA: 1/1 A4



FIRMA: Alvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 2232
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 130420-2	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: ESTABILIZADOR IZQUIERDO	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 2:1	Nº PLANO: 2.2.3.2
		FECHA DE EDICIÓN: 13/04/20	HOJA: 1/1 A4



FIRMA:
Álvaro Alberto Giner

UNIVERSIDAD:



UNIVERSITAT
POLITÀCNICA
DE VALÈNCIA

ESCUELA:



Escuela Técnica Superior de Ingeniería del Diseño

NOMBRE DEL PROYECTO:

Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil

TIPO DE DOCUMENTO:

Dibujo de subconjunto

TÍTULO:

BULÓN BOCA

MATERIAL:

ABS

MARCA:

224

REFERENCIA:

120420-4

ORIENTACIÓN:



CREADO POR:

HEXAPUS

REVISIÓN:

A

ESCALA:

5:1

Nº PLANO:

2.2.4

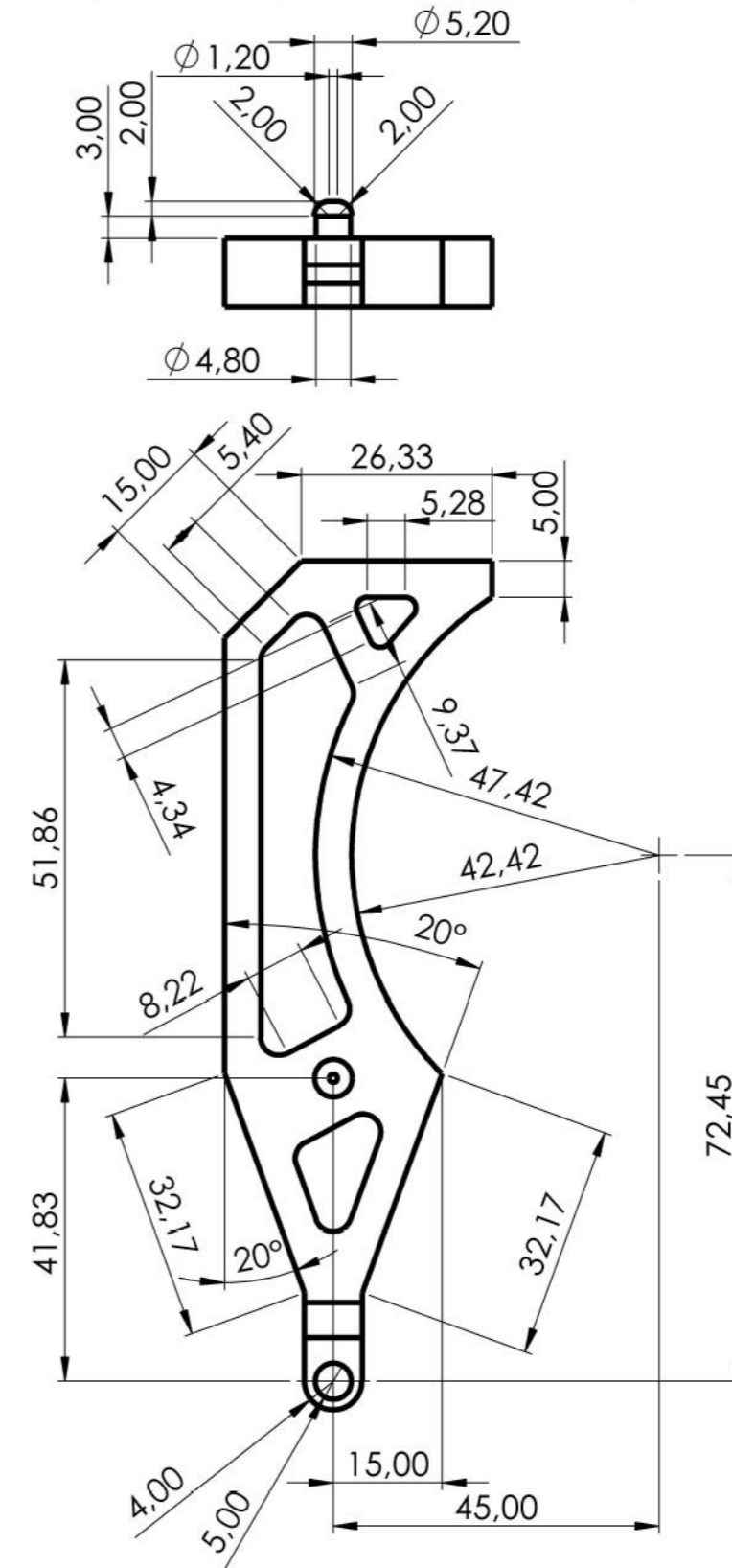
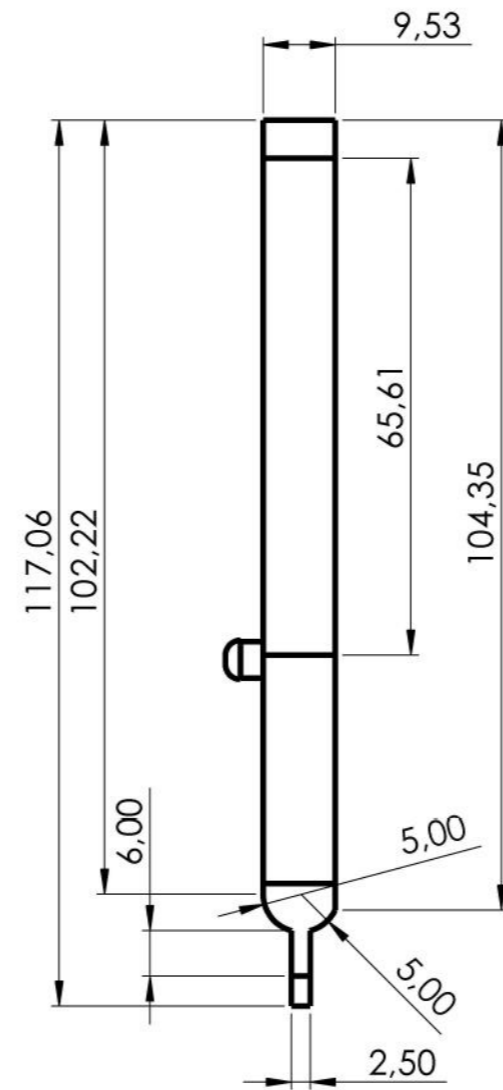
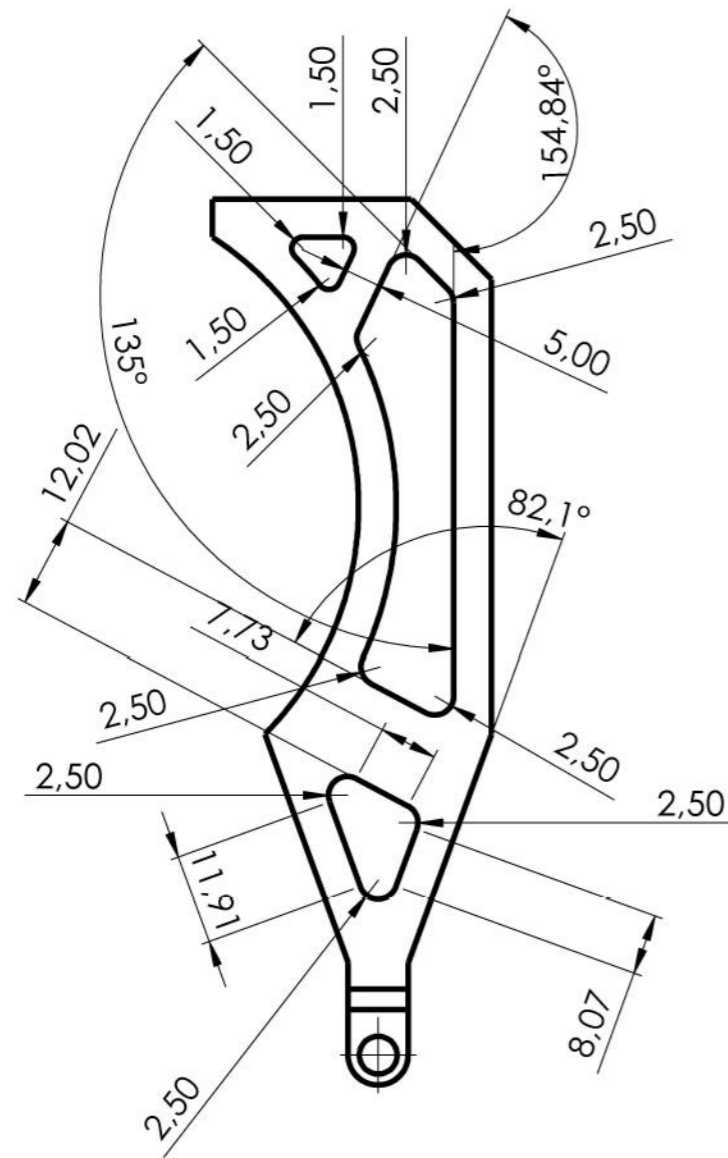
FECHA DE EDICIÓN:


12/04/20

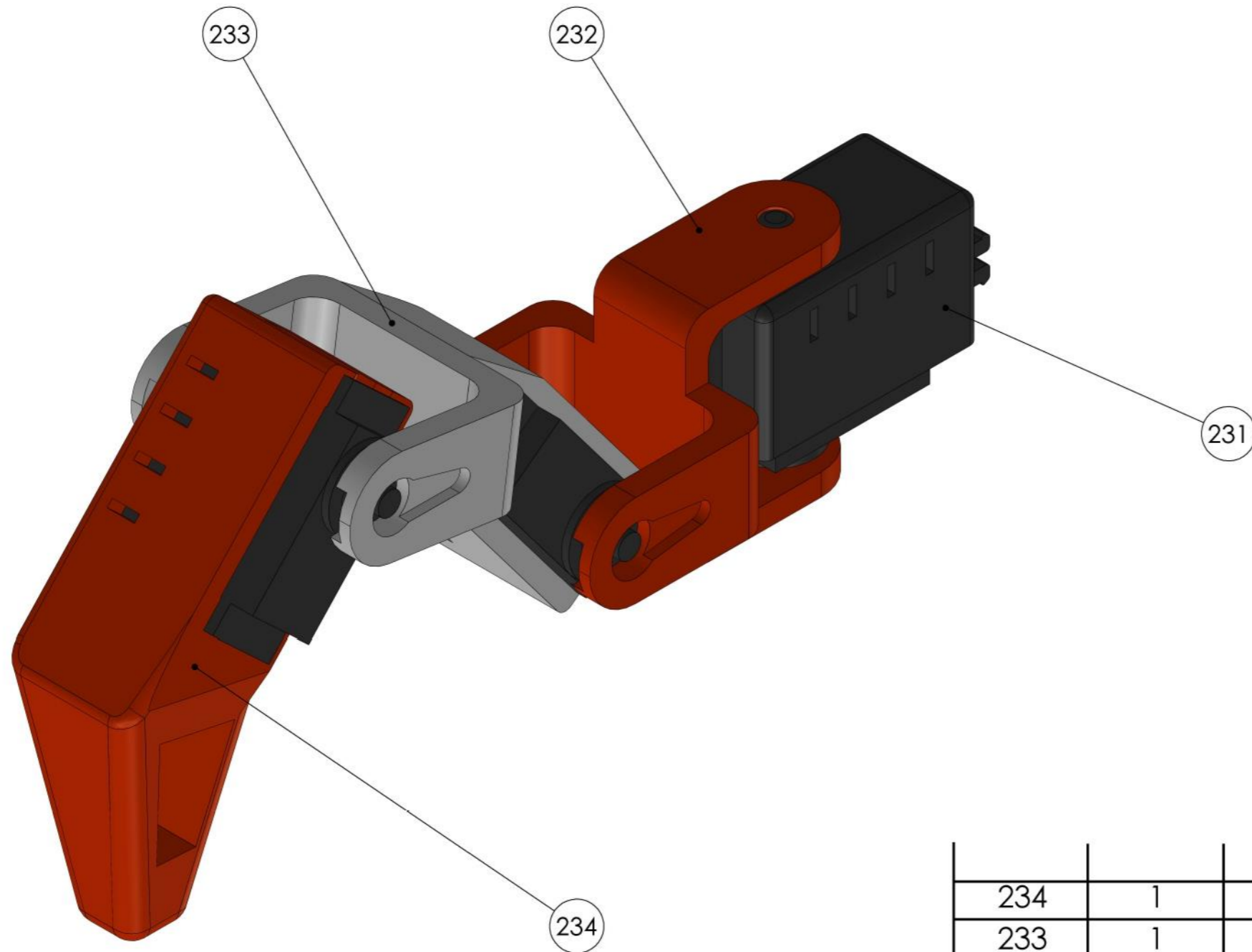
HOJA:

1/1




A4

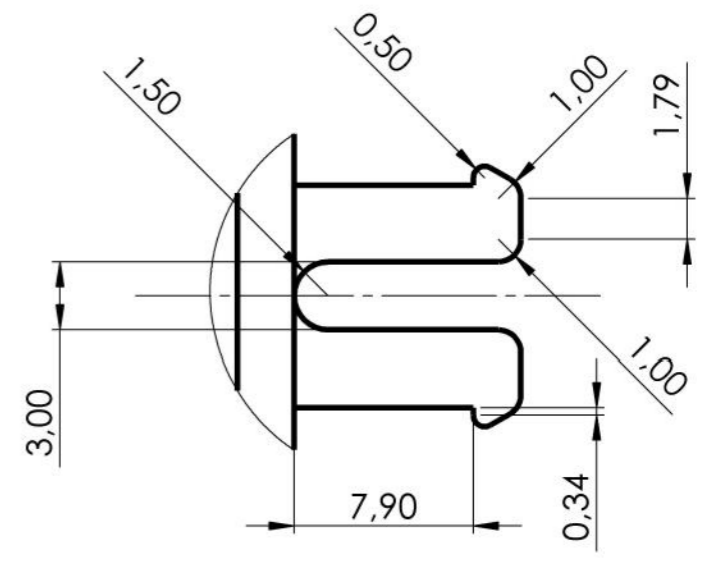
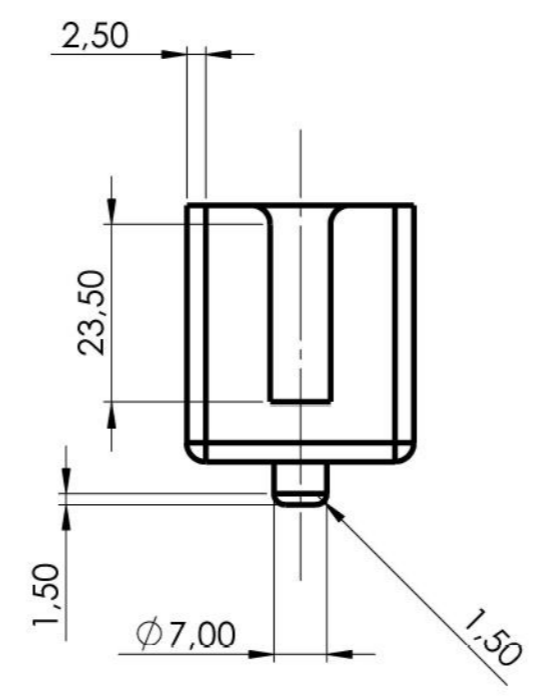
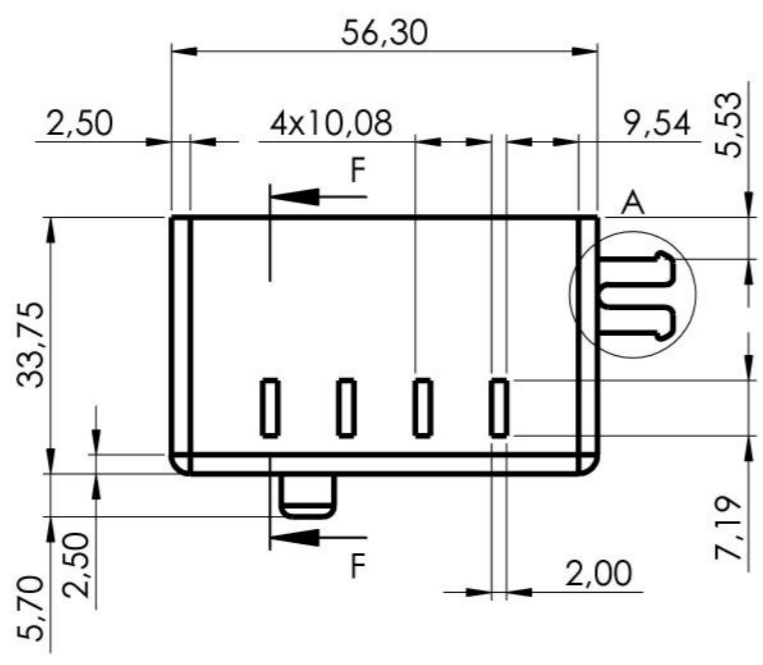
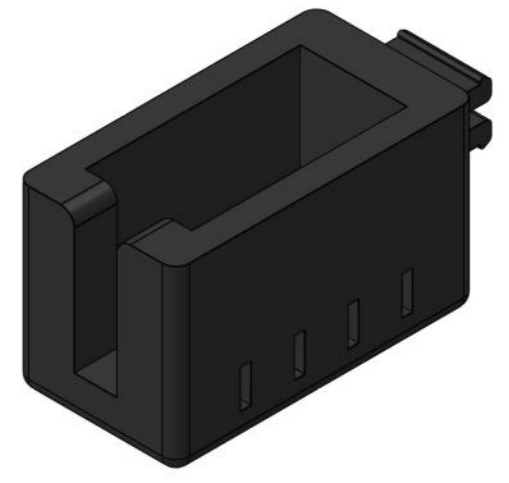
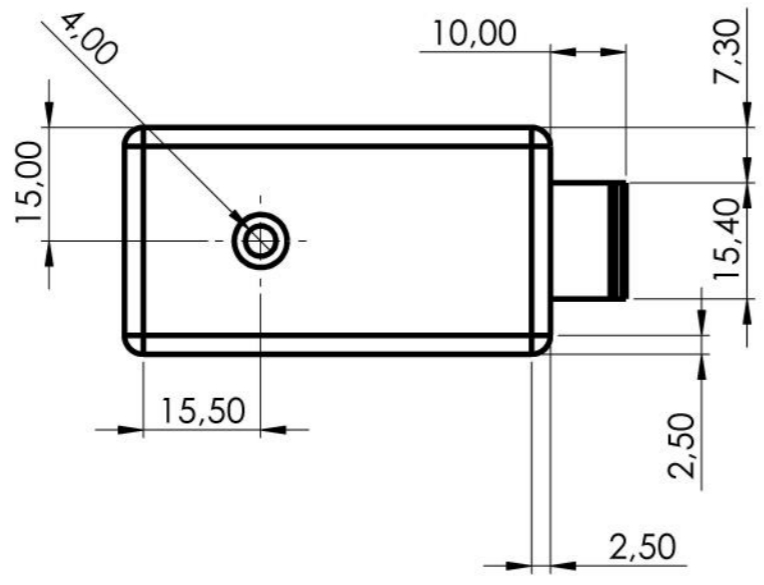


FIRMA: Alvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 2251
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 130420-3	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: MANDÍBULA DERECHA	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 1:1	Nº PLANO: 2.2.5.1
		FECHA DE EDICIÓN: 13/04/20	HOJA: 1/1
			A3

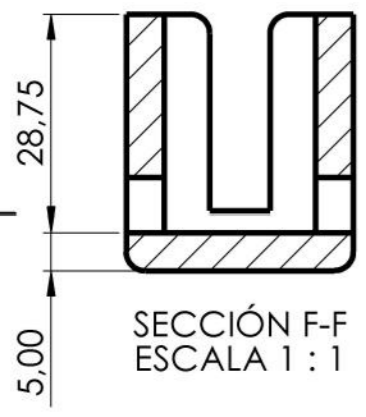


Marca	Nº Pieza	Designación	Norma	Material
	234	Muñeca	UNE-EN ISO 15015	ABS
	233	Codo	UNE-EN ISO 15015	ABS
	232	Unión Hombro-Codo	UNE-EN ISO 15015	ABS
	231	Hombro	UNE-EN ISO 15015	ABS

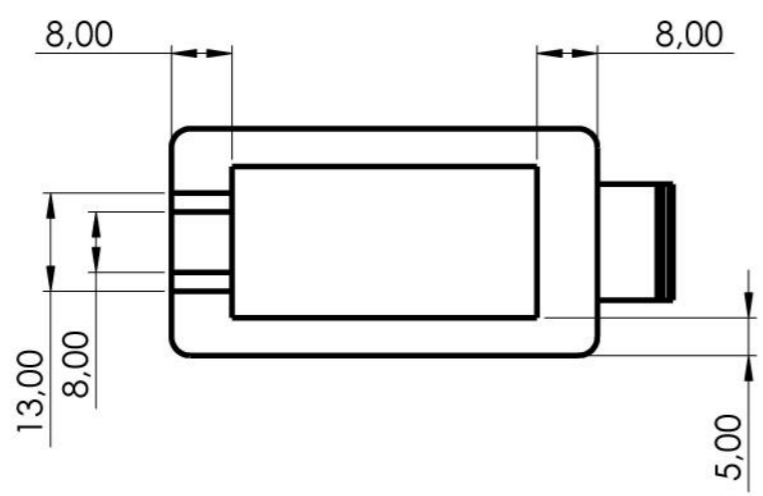
FIRMA: Alvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 3
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Vista General de subconjunto	REFERENCIA: 120420-6	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: PATA	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 1:1	Nº PLANO: 2.3
		FECHA DE EDICIÓN: 12/04/20	HOJA: 1/1 A3



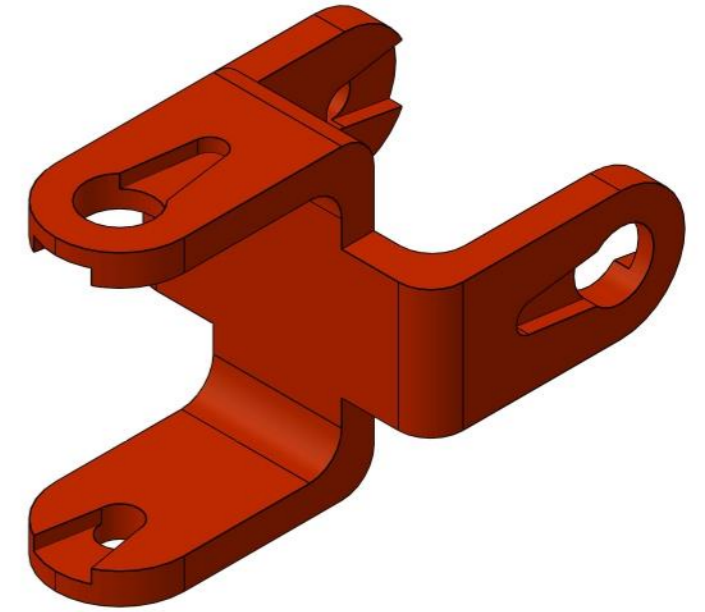
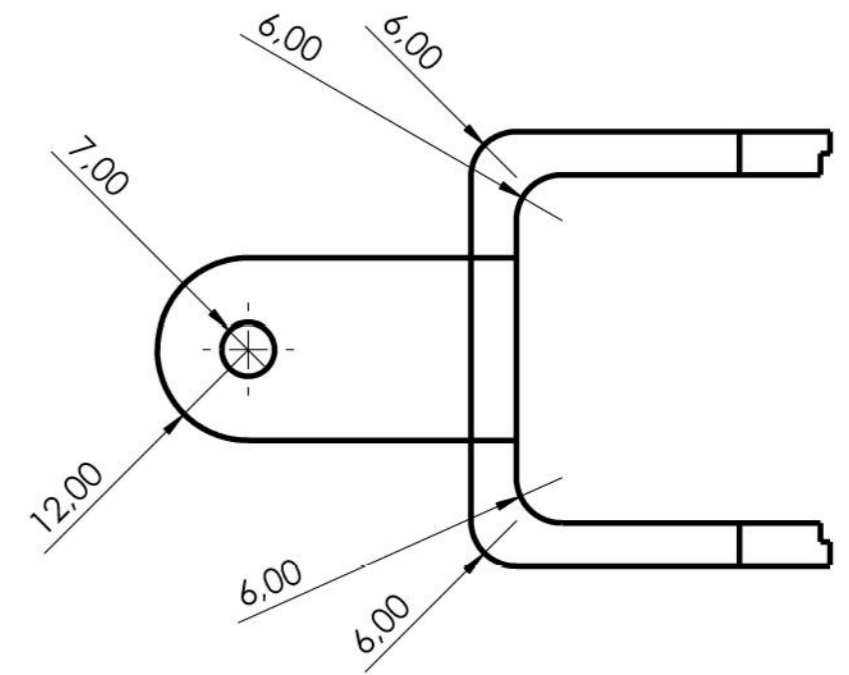
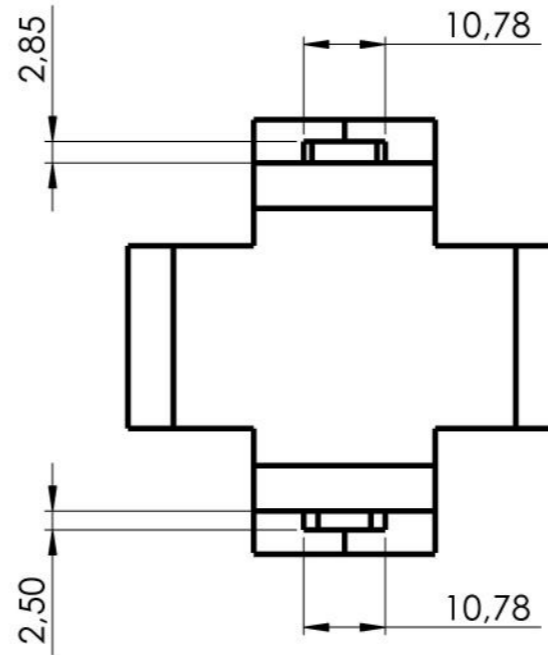
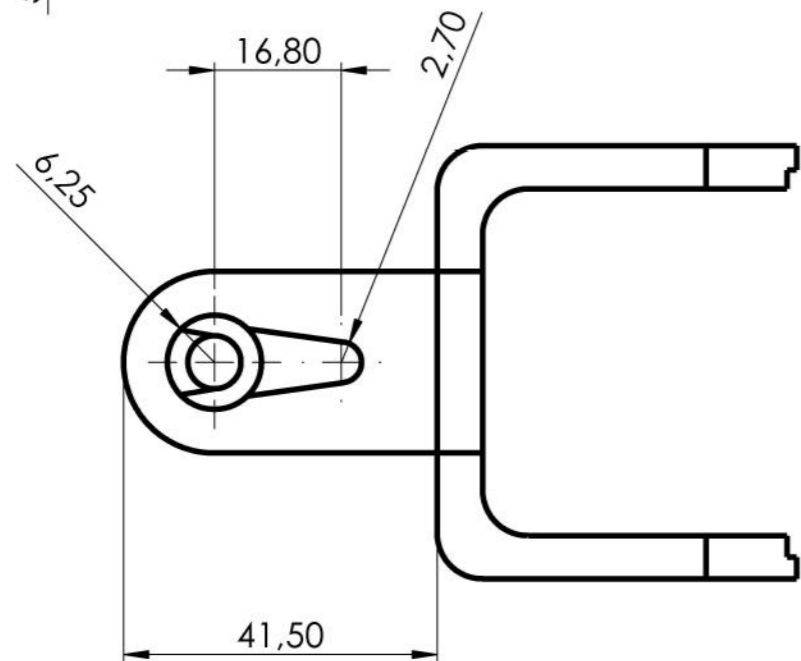
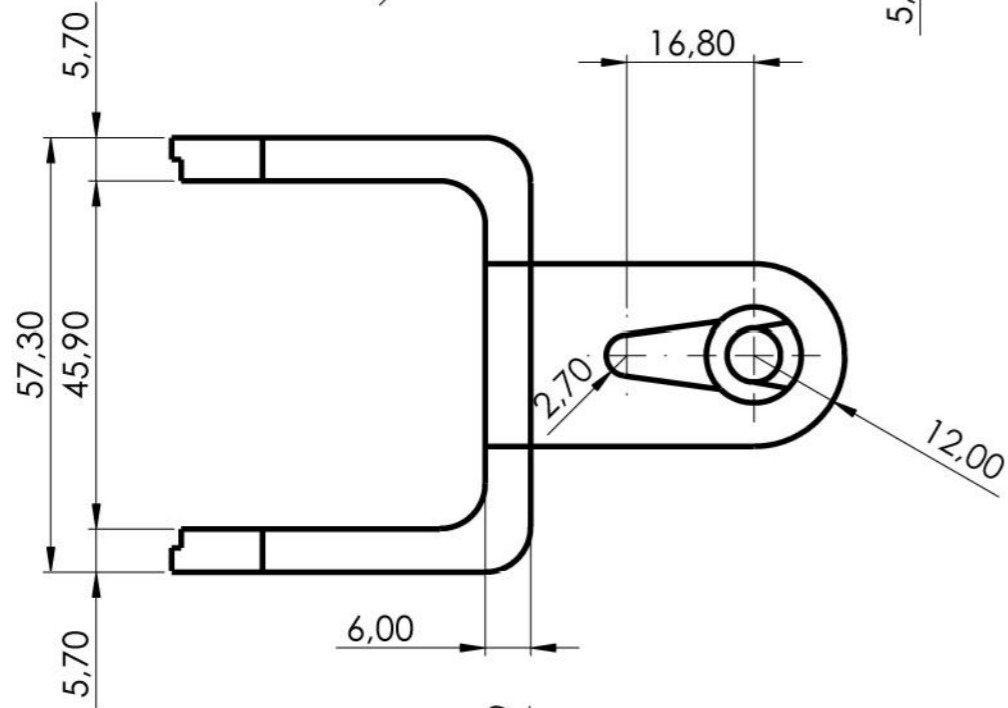
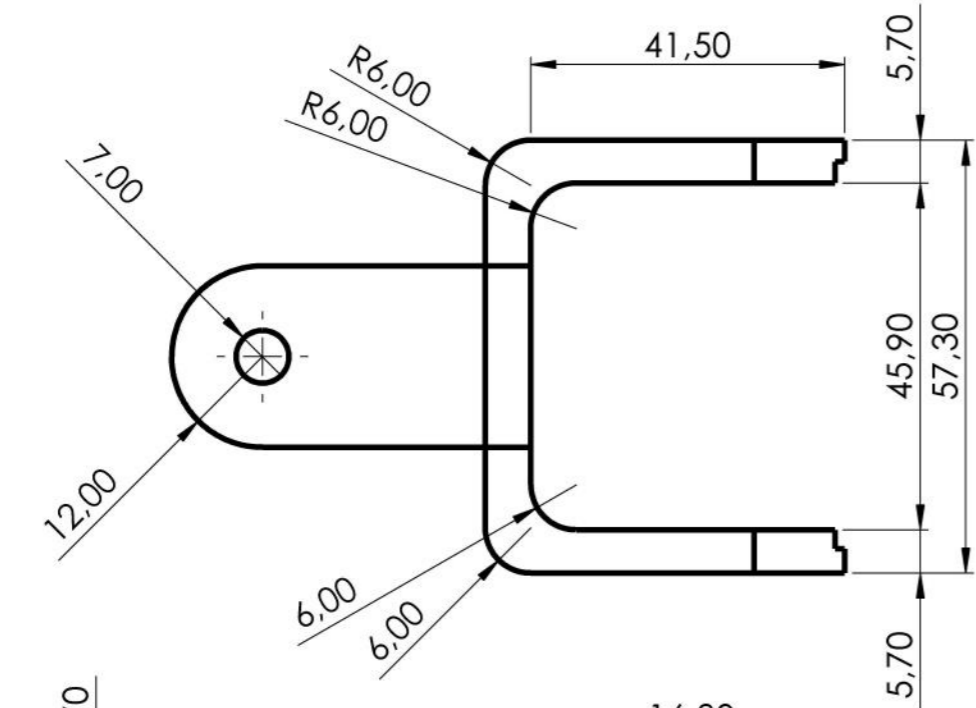
DETALLE A
ESCALA 3 : 1



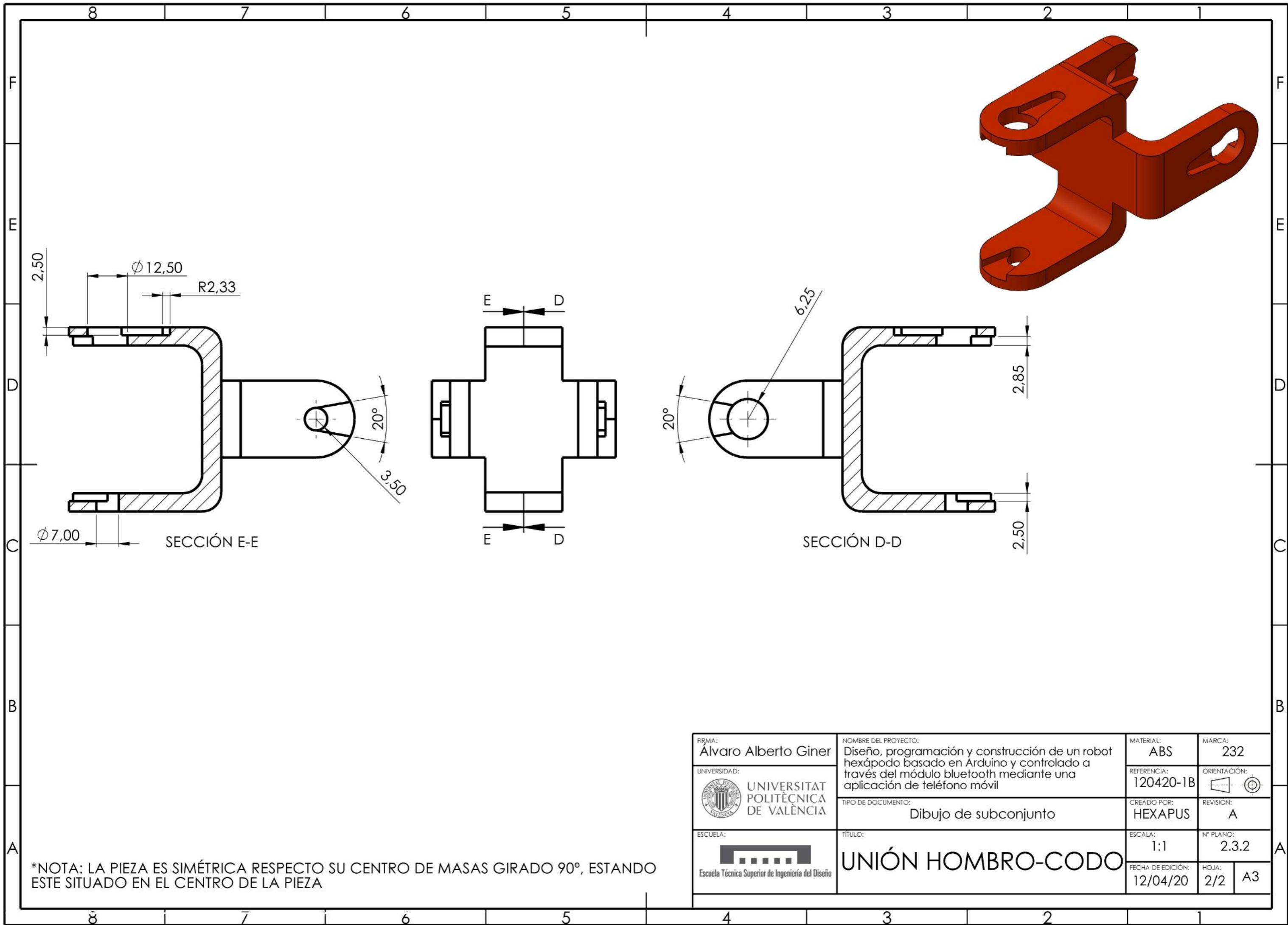
SECCIÓN F-F
ESCALA 1 : 1



FIRMA: Alvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 231
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 110420-2	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: HOMBRO	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 1:1	Nº PLANO: 2.3.1
		FECHA DE EDICIÓN: 11/04/20	HOJA: 1/1
			A3

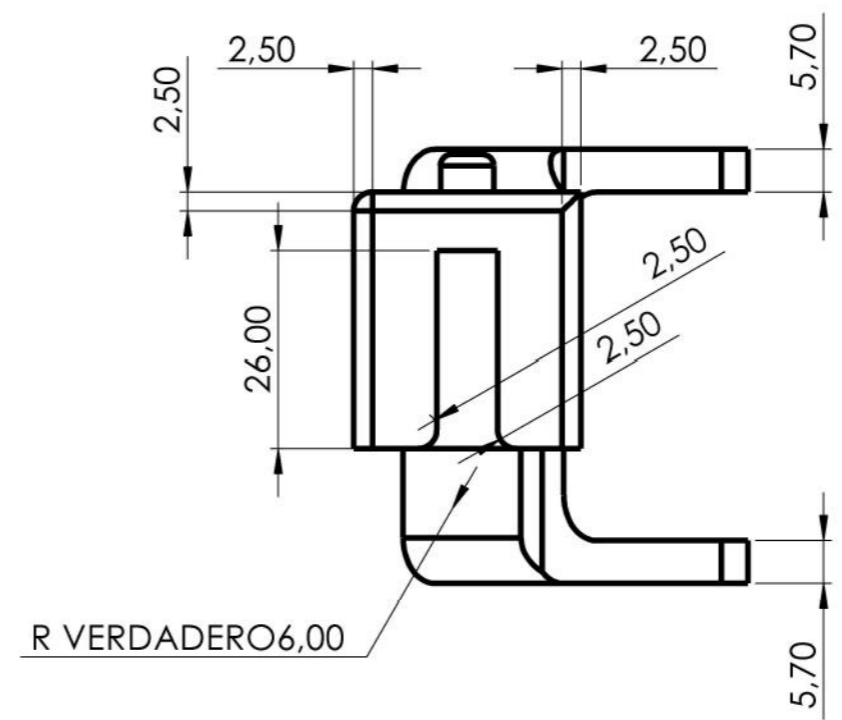
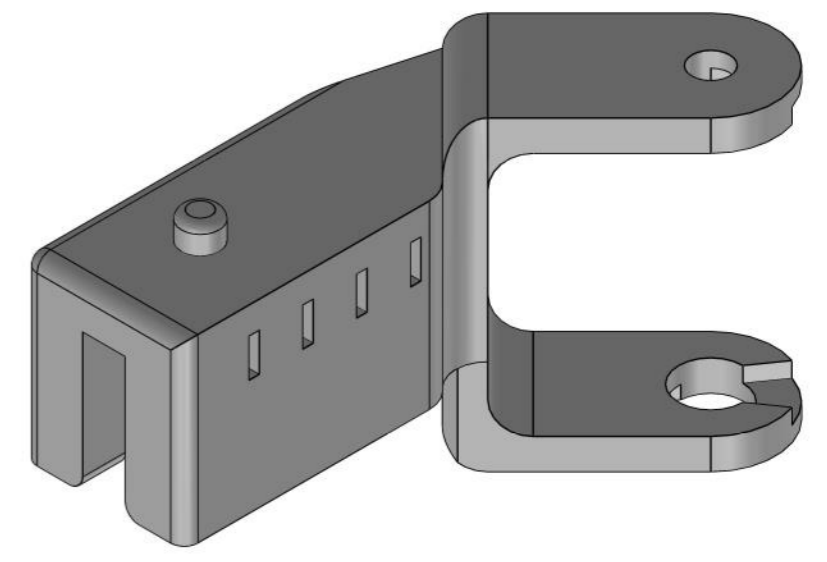
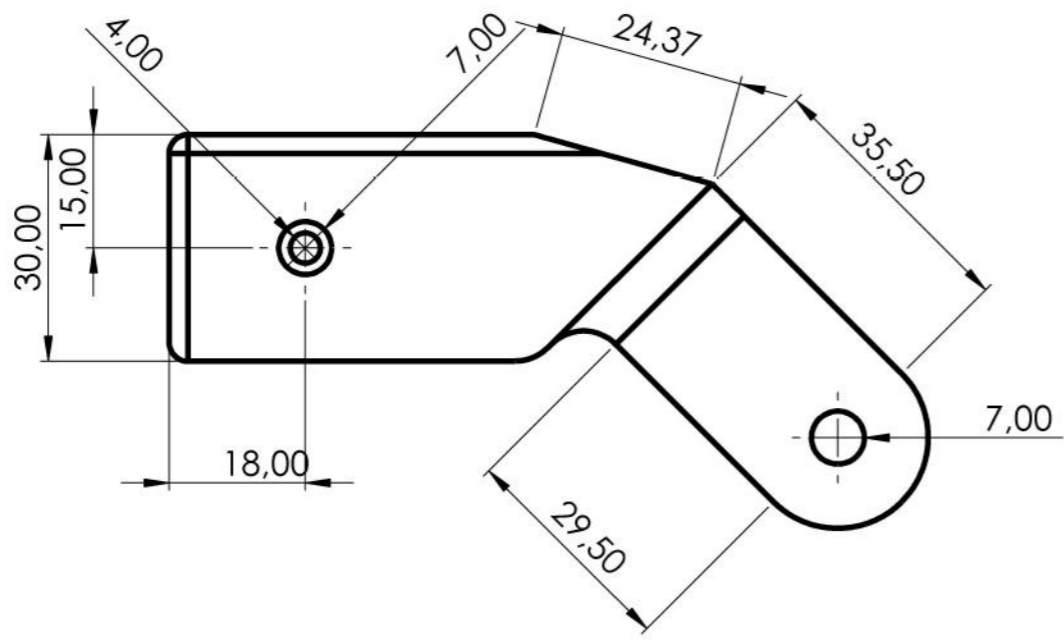
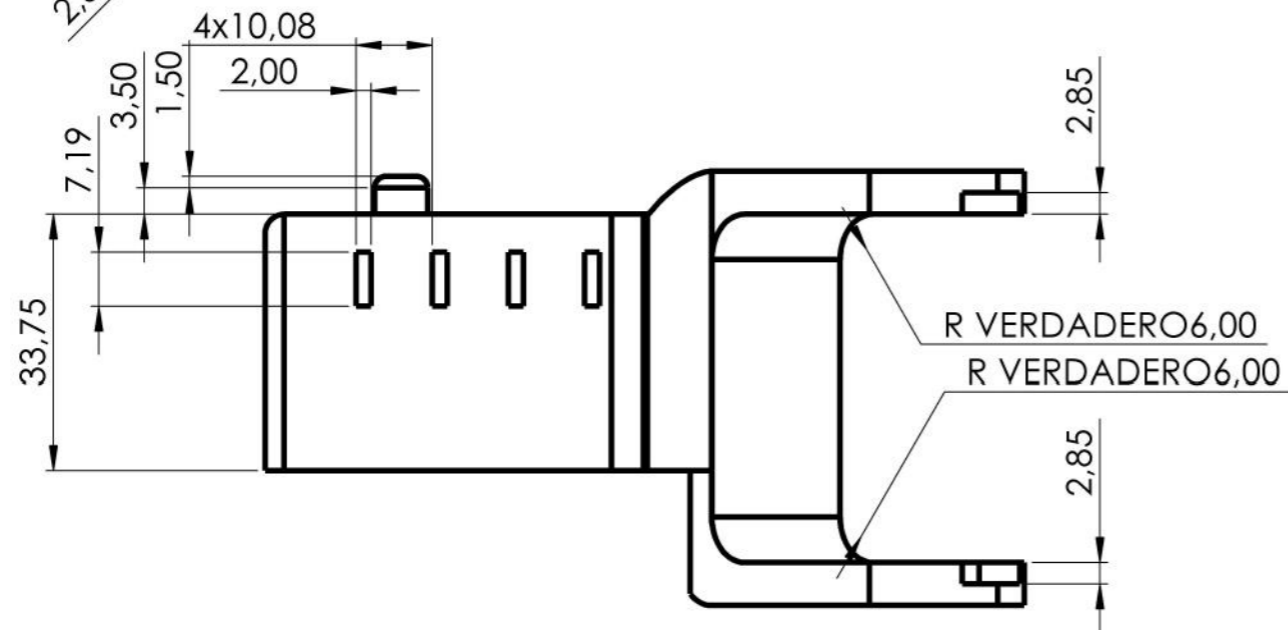
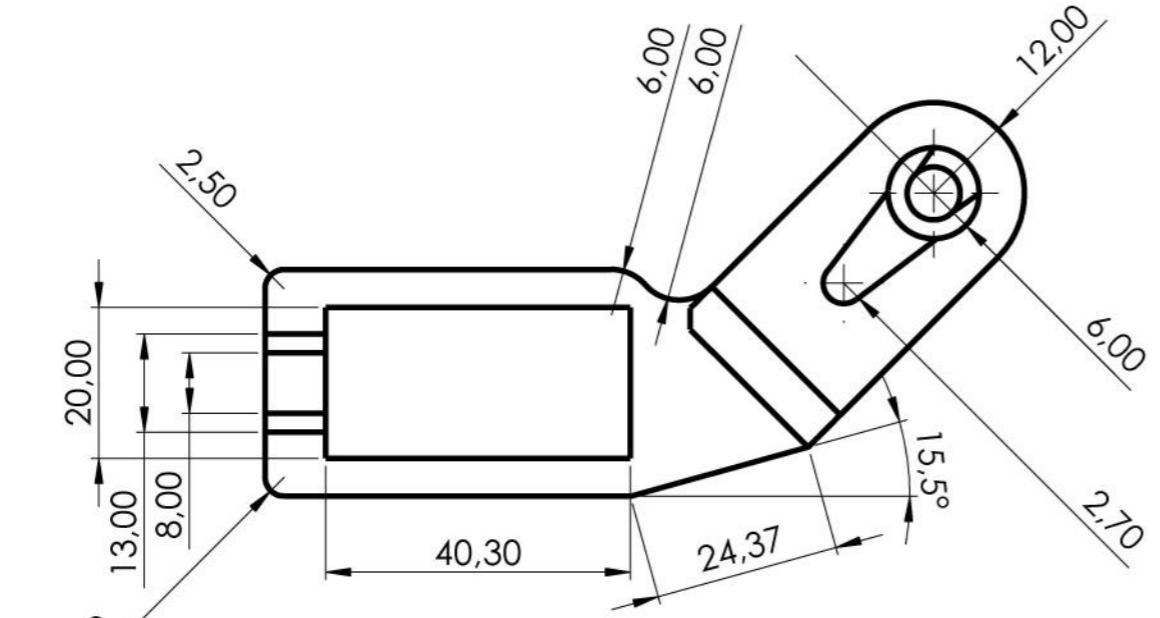


FIRMA: Álvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 232
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 120420-1A	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: UNIÓN HOMBRO-CODO	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 1:1	Nº PLANO: 2.3.2
		FECHA DE EDICIÓN: 12/04/20	HOJA: 1/2 A3

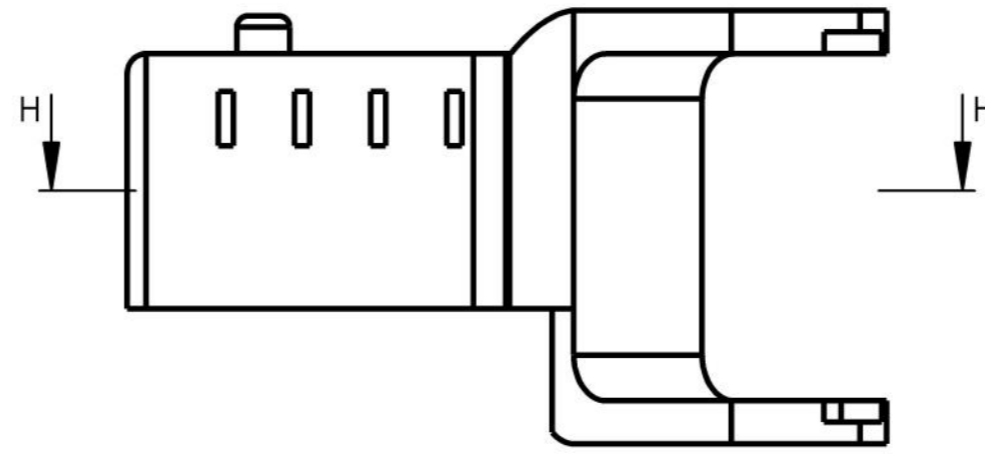
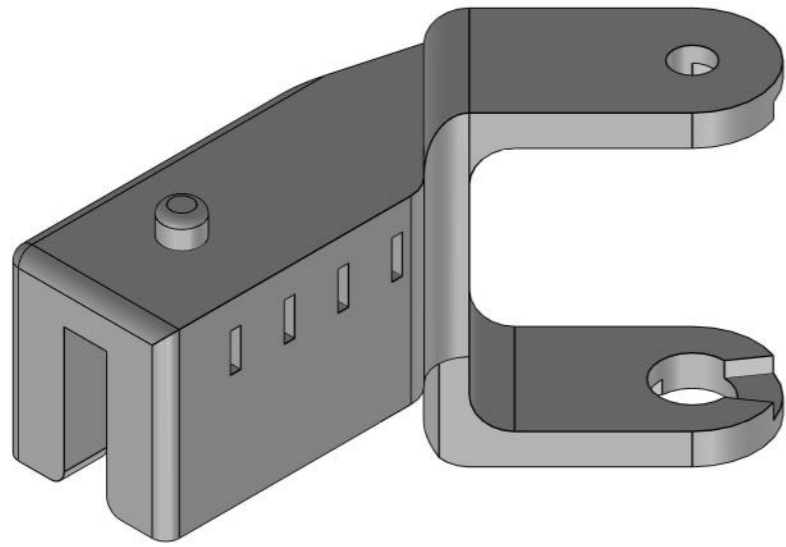


*NOTA: LA PIEZA ES SIMÉTRICA RESPECTO SU CENTRO DE MASAS GIRADO 90°, ESTANDO ESTE SITUADO EN EL CENTRO DE LA PIEZA

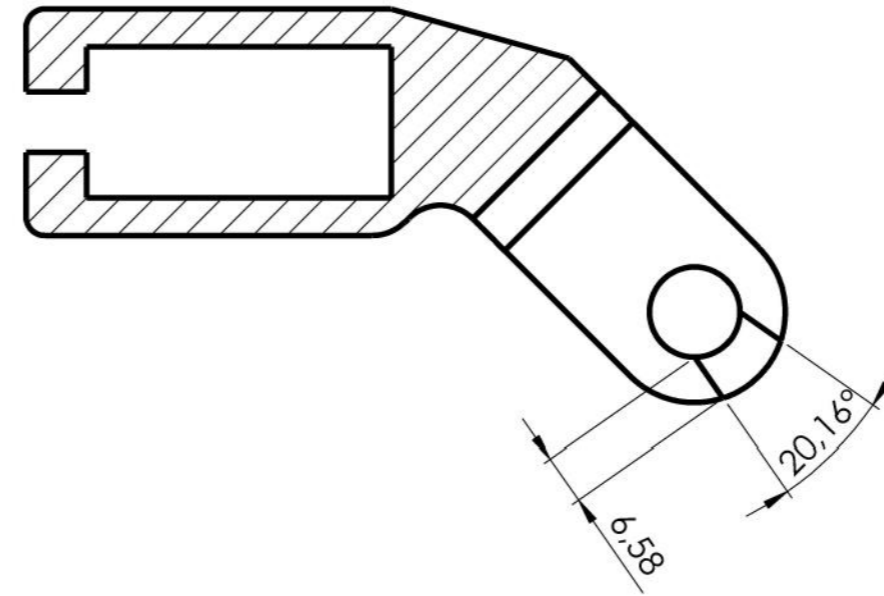
FIRMA: Álvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 232
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 120420-1B	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: UNIÓN HOMBRO-CODO	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 1:1	Nº PLANO: 2.3.2
		FECHA DE EDICIÓN: 12/04/20	HOJA: 2/2 A3



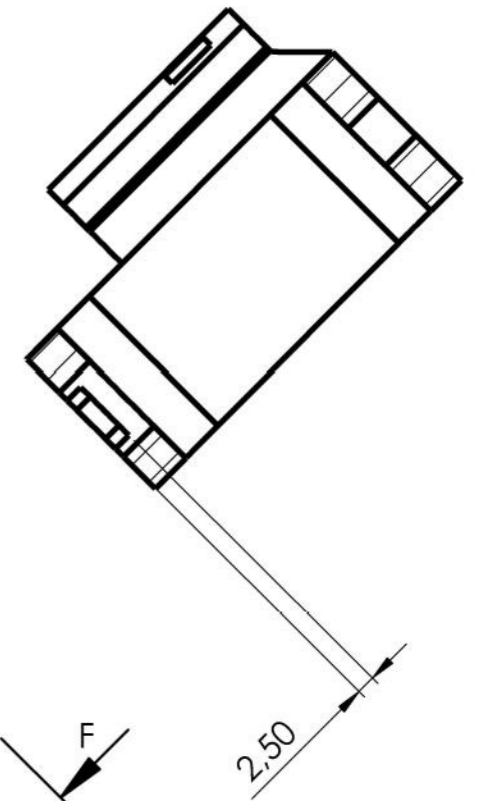
FIRMA: Alvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL:	ABS	MARCA:	233
		REFERENCIA:	110420-3A	ORIENTACIÓN:	
UNIVERSIDAD: UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	CREADO POR:	HEXAPUS	REVISIÓN:	A
		ESCUELA:	Escuela Técnica Superior de Ingeniería del Diseño		TÍTULO:
		ESCALA:	1:1	Nº PLANO:	2.3.3
		FECHA DE EDICIÓN:	11/04/20	HOJA:	1/2
					A3



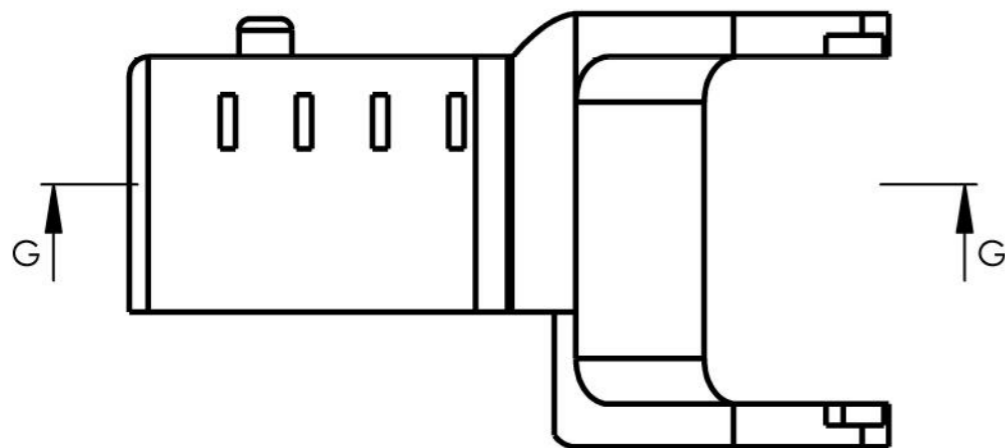
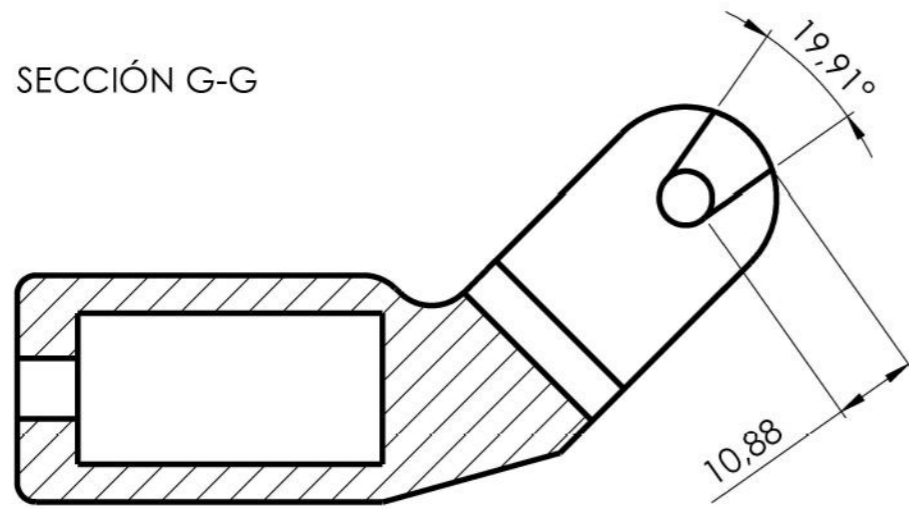
SECCIÓN H-H



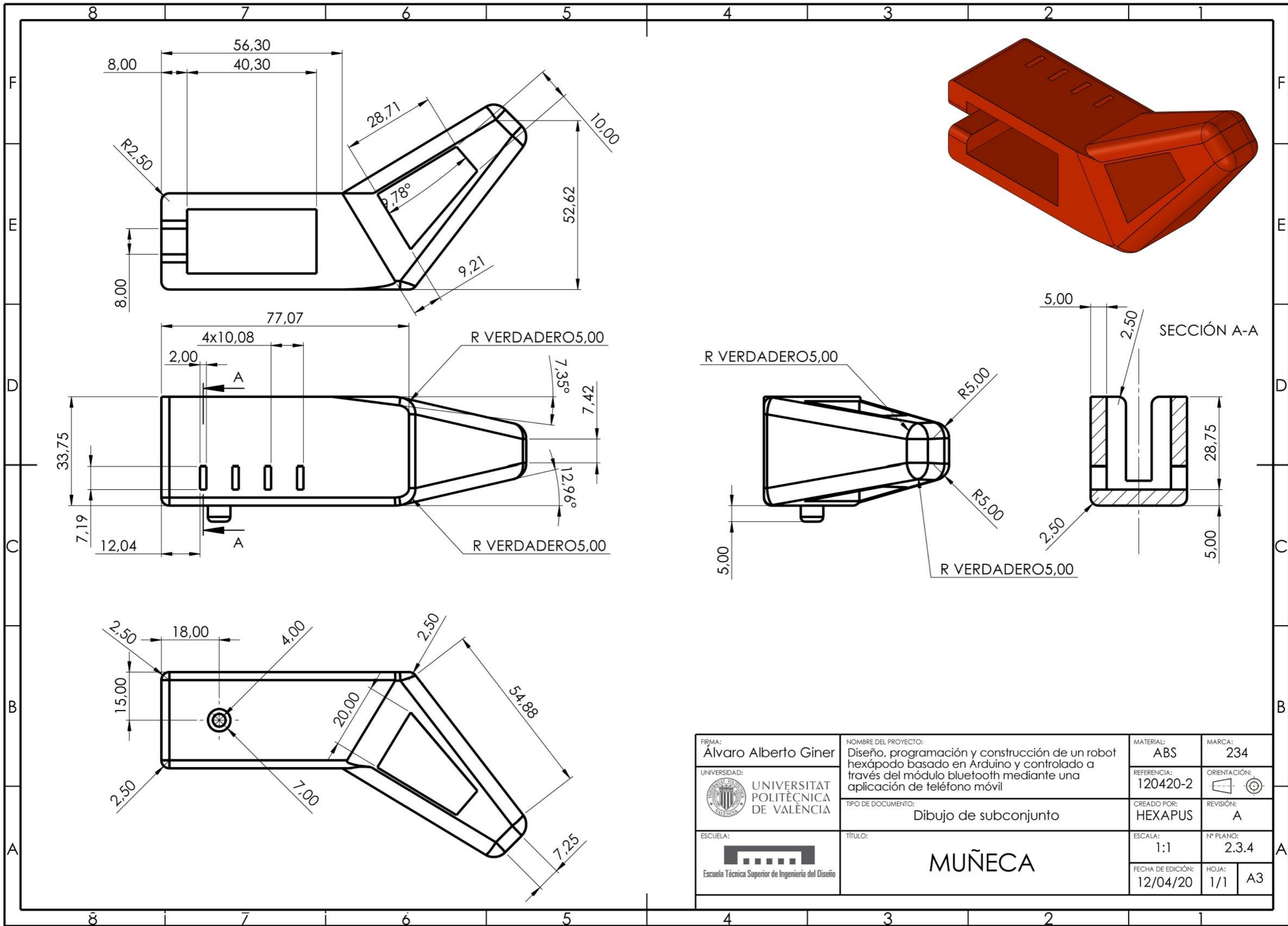
SECCIÓN F-F



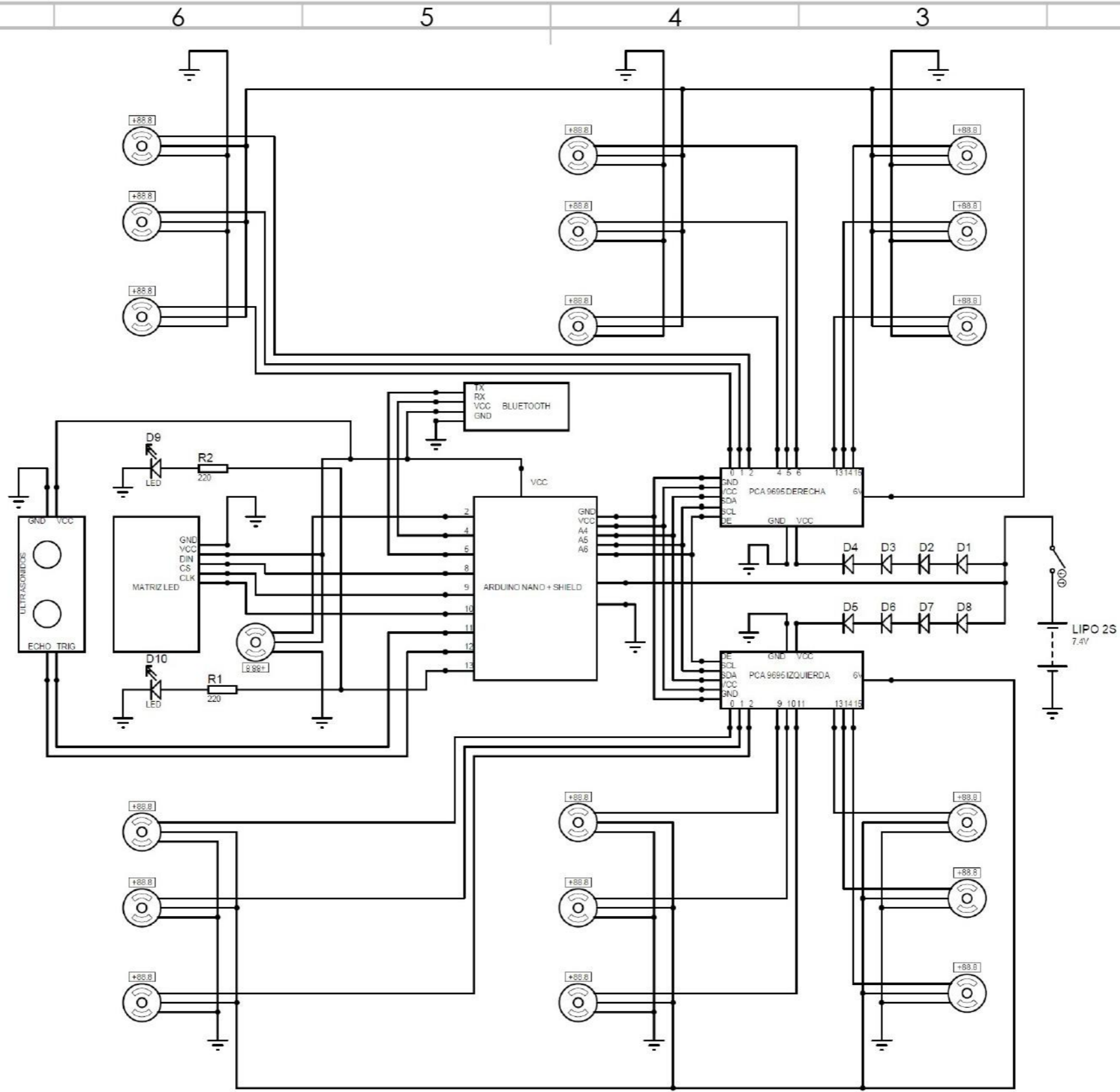
SECCIÓN G-G



FIRMA: Alvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 233
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 110420-3B	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: CODO	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 1:1	Nº PLANO: 2.3.3
		FECHA DE EDICIÓN: 11/04/20	HOJA: 2/2 A3



FIRMA: Álvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: ABS	MARCA: 234
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 120420-2	ORIENTACIÓN: 
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: MUÑECA	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: 1:1	Nº PLANO: 2.3.4
		FECHA DE EDICIÓN: 12/04/20	HOJA: 1/1
			A3



FIRMA: Alvaro Alberto Giner	NOMBRE DEL PROYECTO: Diseño, programación y construcción de un robot hexápodo basado en Arduino y controlado a través del módulo bluetooth mediante una aplicación de teléfono móvil	MATERIAL: N/A	MARCA: N/A
UNIVERSIDAD:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	TIPO DE DOCUMENTO: Dibujo de subconjunto	REFERENCIA: 140420-1	ORIENTACIÓN: N/A
ESCUELA:  Escuela Técnica Superior de Ingeniería del Diseño	TÍTULO: ESQUEMA ELÉCTRICO	CREADO POR: HEXAPUS	REVISIÓN: A
		ESCALA: S/E	Nº PLANO: 2.4
		FECHA DE EDICIÓN: 14/04/20	HOJA: 1/1
			A3



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

**DOCUMENTO NÚMERO 3:
PLIEGO DE CONDICIONES**

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

Álvaro Alberto Giner

TUTORIZADO POR

Leopoldo Armesto Ángel

CURSO ACADÉMICO: 2019/2020

Hoja en blanco dejada a propósito

1.- OBJETO

El departamento de Ingeniería de Sistemas y Automática de la Universidad Politécnica de Valencia va a promover el diseño, programación y la construcción de un prototipo de robot hexápodo destinado principalmente al uso educativo, así como para el transporte a pequeña escala.

El robot hexápodo pretende simular e imitar el comportamiento de un insecto hexápodo. Constará de seis patas con 3 grados de libertad cada una los cuales le permitirán una gran movilidad y variedad de movimientos. Cabe añadir que este robot estará dotado de unas pinzas en la parte delantera las cuales le permitirán interaccionar con objetos de su entorno, teniendo la suficiente fuerza para transportarlos al igual que poder transportar objetos en la parte superior.

Se controlará desde una aplicación móvil propia que se podrá descargar en el smartphone del usuario (únicamente para dispositivos Android cuyas versiones de éste sean superiores a la 4.0) (Código QR adjunto en el “Anexo A-1: Manual de Usuario y Mantenimiento”) con la posibilidad de hacer que el robot se mueva libremente sin colisionar con objetos de su entorno. Éste podrá ser controlado mediante la aplicación hasta un alcance máximo de 15 metros gracias a la conectividad Bluetooth. Además, contará con una batería recargable con una autonomía mínima de 7 minutos.

El proyecto no se debe demorar más allá del mes de junio de 2020, en caso de suceder, el promotor tendrá un descuento del 8% sobre coste total del proyecto.

El promotor del proyecto, el departamento de Ingeniería de Sistemas y Automática de la Universidad Politécnica de Valencia, con C.I.F. A20917125 y con dirección Camí de Vera, CP 46022, Valencia, se trata de un departamento especializado en automatización, control y optimización de procesos, robótica y visión artificial entre otros. Con este proyecto pretende crear un prototipo de robot hexápodo cuya finalidad es educativa, permitiendo que los alumnos lo puedan reprogramar y aplicar sus conocimientos.

2.- CONDICIONES DE LOS MATERIALES

Todos los materiales adquiridos para la fabricación de dicho producto estarán conforme a la normativa vigente y exigible (marcado CE).

2.1.- CHASIS

Todas las partes del chasis se realizarán con ABS u otro material equivalente cumpliendo con la norma UNE-EN ISO 15015 con una elasticidad de tracción de 1681.5 MPa y una resistencia a la flexión de 39 MPa.

Los métodos de ensayo y controles de calidad a pasar por el material ABS, se encuentran descritos en el apartado 5 de la norma UNE-EN ISO 15015.

Las piezas del robot se deberán poder imprimir con una impresora 3D cuyas dimensiones de impresión no excedan los 220x220x250 mm además de imprimir las piezas con las siguientes características:

- Temperatura de impresión: 240 °C
- Temperatura de la cama caliente: Primera capa a 70 °C y el resto a 85 °C
- Velocidad de impresión: 60 mm/s
- Número de paredes: 3

- Número de capas inferiores: 3
- Número de capas superiores: 3
- Relleno y geometría: 20 % y con forma de rejilla
- Altura de capa: 0.2 mm
- Boquilla: 0.4 mm

2.2.- UNIDAD DE CONTROL

Se deberá asegurar para el correcto funcionamiento del dispositivo las siguientes características:

El microprocesador, Arduino Nano o similar, deberá tener una velocidad de reloj igual o superior a 16 MHz al igual que poder ser alimentado constantemente a 3.3 V como mínimo y nunca superando los 12 V.

El sensor de proximidad deberá tener una sensibilidad de mínima de 3 mm y un consumo máximo de 20 mA, así mismo tener un rango mínimo de medida de 200 centímetros o más.

En lo que respecta a la matriz de leds de la parte delantera del dispositivo deberá contener un controlador MAX-7219 o equivalente contando con exactamente 64 leds de color rojo cuyo consumo máximo sea de 2 mA/led.

El servomotor de 180° dedicado a la apertura de las pinzas deberá ser un SG90 o similar, cuyo peso máximo no superará los 10 g y su control deberá poder realizarse mediante señales PWM así como tener una tensión de funcionamiento de 5V.

El dispositivo deberá contener 2 controladores PCA9685 o similar, de 16 canales cada uno, pudiendo ser sustituido por un único controlador PCA de 32 pines. La tensión de alimentación de este controlador no deberá superar en ningún caso los 6 V y no ser inferior de 4.8 V.

La duración de la batería deberá asegurar un uso de al menos 7 minutos pudiéndose cargar si fuera necesario durante la duración media de una clase de laboratorio, aproximadamente una hora.

Por último, el dispositivo Bluetooth deberá tener un área de trabajo mínima de 10 m y un consumo máximo de 40 mA.

Cabe destacar que todos los componentes de la unidad de control deberán haber pasado sus correspondientes controles de calidad en sus respectivas fábricas además de poseer su marcado CE y cumplir la normativa vigente.

2.3.- SISTEMA DE MOVIMIENTO

Por otro lado, los servomotores utilizados en las extremidades del robot deberán ser los MG996R o similares. Deberán tener un par mínimo de 10 Kg-cm y una tensión mínima de trabajo de 4.8 V y una tensión máxima de 7.2 V. El robot estará dotado exactamente de 18 servomotores iguales, 3 para cada pata. El consumo máximo no deberá superar el amperio.

Las dimensiones no deberán exceder los 40.7 x 19.7 x 42.9 mm y su peso neto no exceder los 60 g.

Las piezas que componen el sistema de movimiento serán impresas en 3D con las mismas características del chasis.

Los servos deberán haber pasado los controles de calidad en la fábrica en la que se han producido cumpliendo la normativa vigente y contando con el marcado CE.

Los métodos de ensayo y controles de calidad a pasar por el material ABS, como ya se ha comentado anteriormente en este documento, vienen descritos en el apartado 5 de la norma UNE-EN ISO 15015.

3.- EJECUCIÓN Y MONTAJE

3.1.- SUBMONTAJE DEL CHASIS DE LA PARTE SUPERIOR

Para ensamblar la parte superior del chasis es necesario introducir las patillas del chasis superior trasero en los orificios del chasis superior delantero (Figura 35). Una vez se escuche un “click” éste estará ensamblado correctamente.

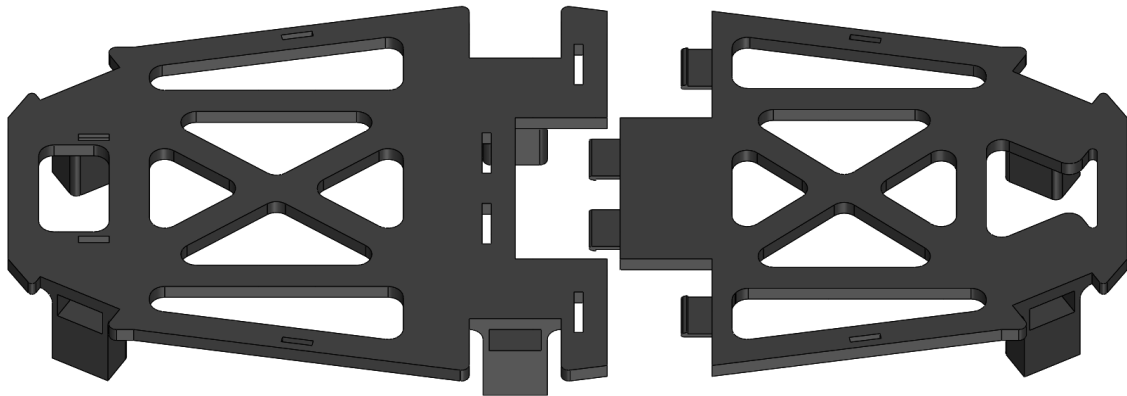


Figura 35: Vista explosionada de la parte del chasis superior. A la derecha chasis superior trasero y a la izquierda el delantero.

3.2.- SUBMONTAJE DEL CHASIS DE LA PARTE INFERIOR

Para ensamblar la parte inferior del chasis, al igual que la parte superior, es necesario introducir las patillas del chasis inferior trasero en los orificios del chasis inferior delantero (Figura 36). Una vez se escuche un “click” éste estará ensamblado correctamente.

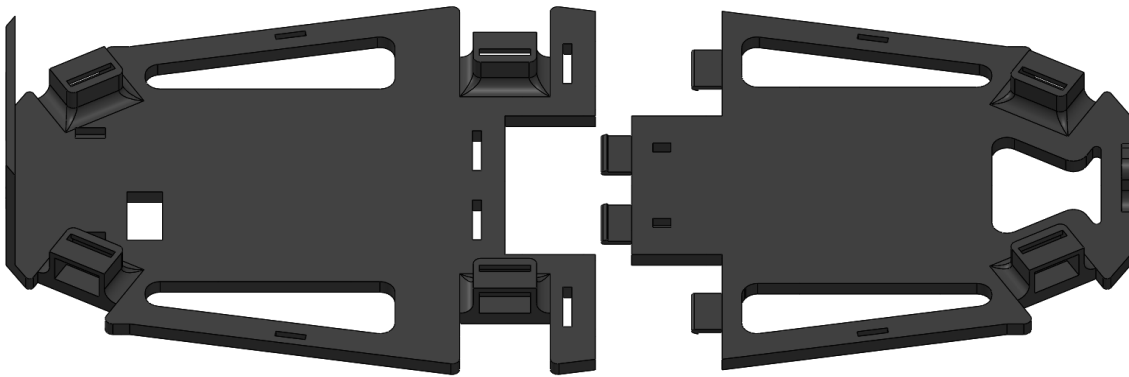


Figura 36: Vista explosionada de la parte del chasis superior. A la derecha chasis superior trasero, a la izquierda el delantero.

3.3.- ENSAMBLAJE DEL CHASIS

Una vez ensamblados el chasis superior e inferior, se procederá de la siguiente forma: En primer lugar, se deben anclar los pilares, se recomienda que estos se enganchen primero en el chasis inferior, para saber que esté bien anclado se debe haber escuchado un “click”. Posteriormente, se enganchará mediante otro “click” el soporte del módulo de ultrasonidos al chasis superior. A continuación, y tras haber realizado y situado la electrónica en su interior, se procederá a unir el chasis inferior y superior hasta la escucha de un “click” por cada pilar. Por último, se enganchará el soporte de la batería en los orificios del chasis inferior (Figura 37).

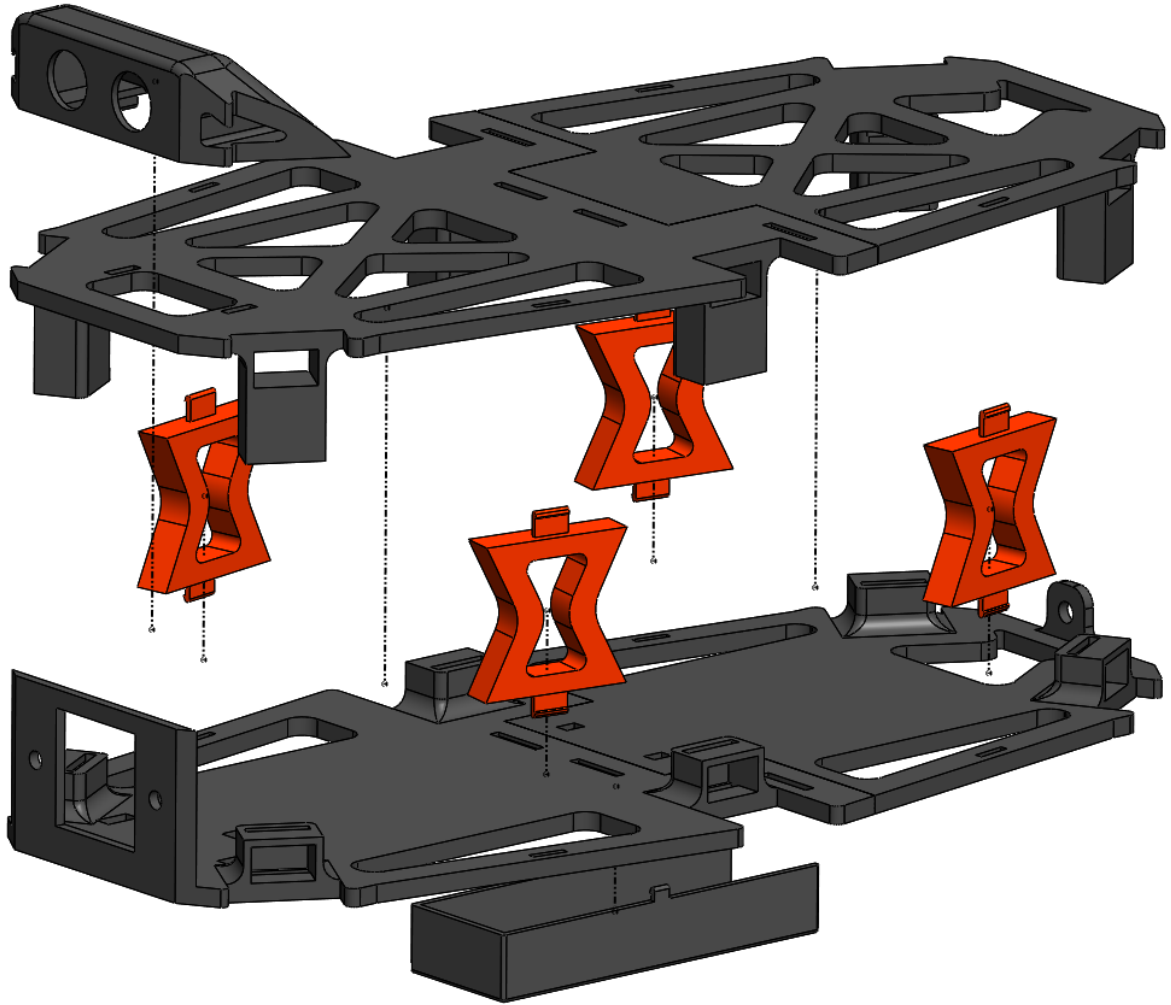


Figura 37: Vista explosionada del chasis. Describiendo la figura de arriba a abajo y en orden, encontramos el soporte de ultrasonidos, chasis superior, pilares, chasis inferior y soporte de la batería.

3.4.- SUBMONTAJE DE LA BOCA

Para ensamblar correctamente las pinzas, se deben seguir los siguientes pasos:

En primer lugar, se deberán insertar en el soporte tanto los engranajes como los estabilizadores. Tras el paso anterior, se deberán enganchar las mandíbulas en los estabilizadores y alinear los agujeros de éstas con los agujeros de los engranajes. Finalmente se deberán pasar los bulones para fijar totalmente las mandíbulas (Véase la Figura 38).

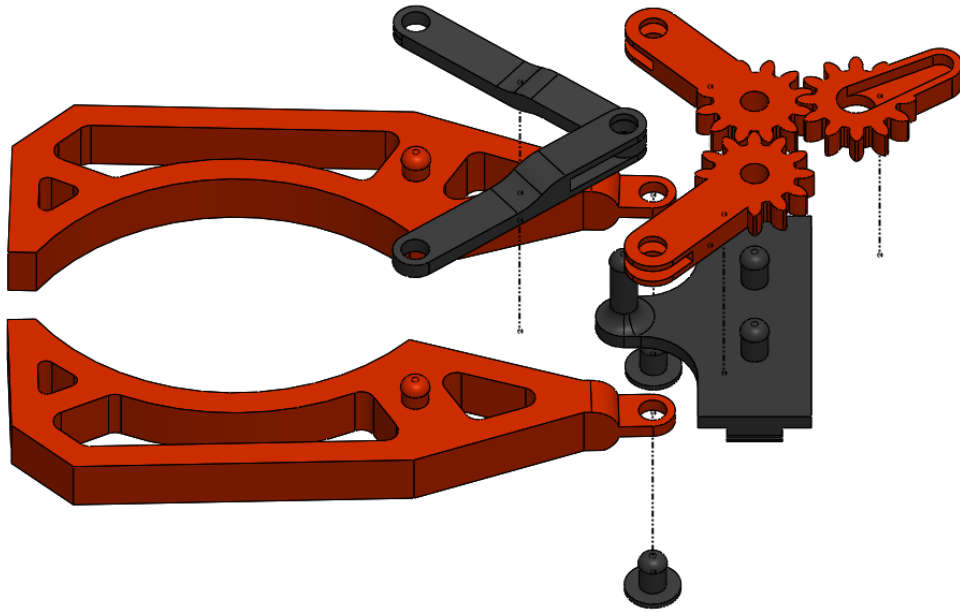


Figura 38: Vista explosionada de la boca. Describiendo la figura de arriba abajo y en orden, encontramos en rojo los engranajes y en negro los estabilizadores, en el siguiente nivel, en negro el soporte y en rojo las mandíbulas y finalmente en el último nivel los bulones.

3.5.- SUBMONTAJE DE LAS PATAS

Para el montaje de las patas es necesario introducir los servomotores dentro de las piezas pertinentes, el hombro, el codo y la muñeca. Una vez introducidos los servomotores en su lugar, se deberá ensamblar la pata (Figura 39). Cabe destacar que a la hora de montar los servomotores hace falta calibrarlos poniendo estos alineados como se muestra en la Figura 16 del documento número 1 de este proyecto. En el momento que los servos estén posicionados con un ángulo de 90° se deberán introducir en los huecos los fijadores negros que se adjuntan con los servomotores además de atornillarlos.

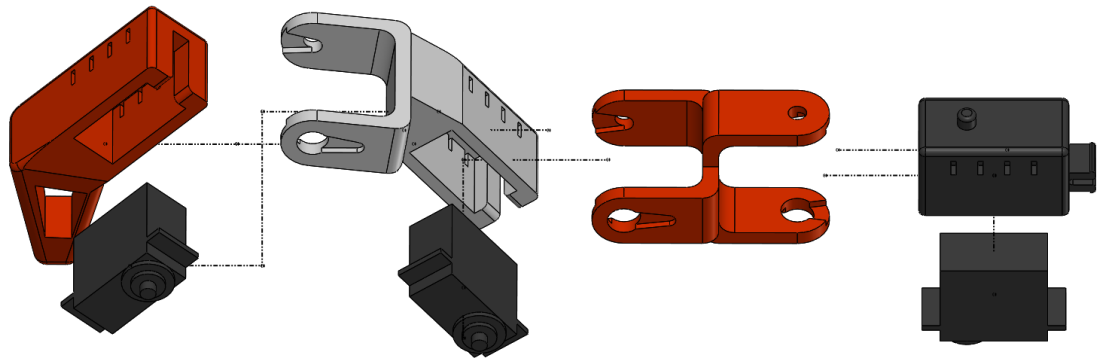


Figura 39: Vista explosionada de una pata. Describiéndola de izquierda a derecha y en orden, encontramos la muñeca y su servomotor, codo y su servomotor, unión hombro-codo y finalmente hombro y su servomotor.

3.6.- ENSAMBLAJE FINAL

Tras los ensamblajes anteriores, se fijarán cada una de las patas en los orificios habilitados para ello además de anclar en la parte inferior del chasis la boca como se muestra en la Figura 40.

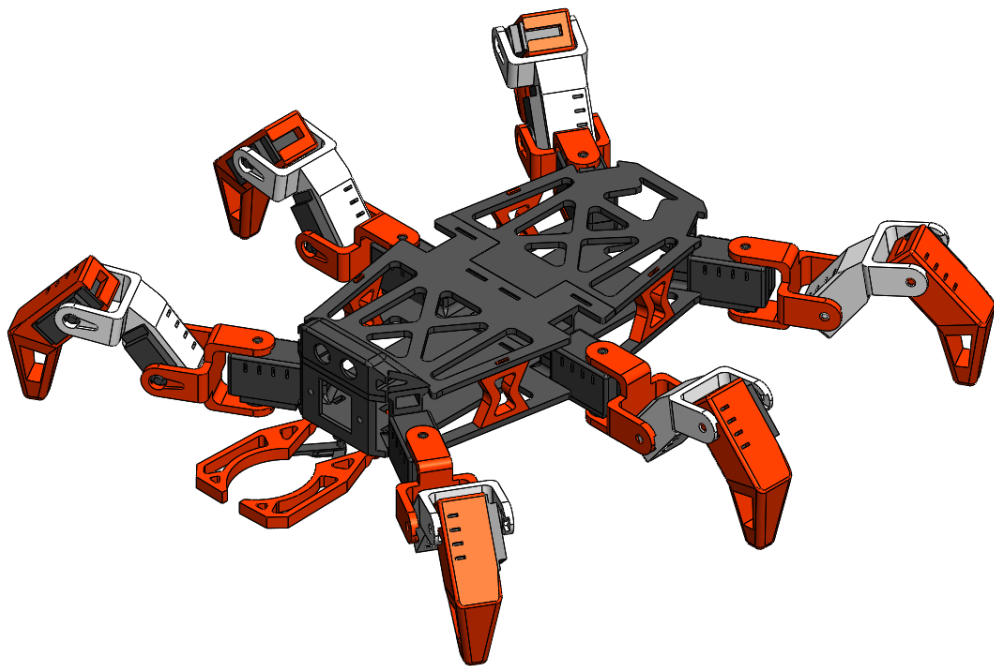


Figura 40: Vista final del ensamblaje del robot hexápodo.

3.7.- MONTAJE DE LA ELECTRÓNICA

El montaje de la electrónica correrá a cargo de la empresa que adquiera el producto, pudiendo distribuir los componentes a su gusto, siempre y cuando se cumplan las especificaciones descritas en los apartados 2.2 y 2.3 de este documento y siguiendo el plano de conexiones (número de plano: 2.4) del documento número 2 del proyecto.

4.- CONTROL DE CALIDAD

El control de calidad del ensamblaje consistirá en revisar que las piezas no contengan defectos visuales que afecten tanto a su funcionalidad como estética además de corroborar que las piezas no tengan cantos afilados ni punzantes. Si tras esta inspección visual las partes plásticas del robot no presentan roturas, deformaciones y cantos afilados o punzantes, pasará el control de calidad. Además, se deberá encender y probar todas las funciones del robot para asegurarse que todas las conexiones y módulos funcionan correctamente y ningún componente a excepción de la parte de alimentación se caliente al tacto.

Tras finalizar las pruebas de funcionamiento, se deberá cargar la batería del robot en modo almacenamiento con una tensión nominal de 7.6 V para evitar que ésta se deteriore y su vida útil disminuya, siempre balanceando sus celdas.

5.- ENSAYOS DE RECEPCIÓN

En la Tabla 19 se muestran dichos ensayos de recepción.

COMPONENTE	TIPO DE PRUEBA	CONTROLES QUE REALIZAR	NÚMERO DE CONTROLES	CONDICIONES DE NO ACEPTACIÓN
Microprocesador y módulo Bluetooth.	Recepción de comandos provenientes desde la aplicación móvil.	Mediante la aplicación móvil ofrecida con el producto se procederá a probar todos y cada uno de los comandos observando si son recibidos por el módulo Bluetooth. Ésto se podrá observar con la ayuda del Monitor serial ofrecido por Arduino.	Uno por dispositivo.	La no recepción inmediata de un comando.
Detector de proximidad.	Lectura de la distancia de los obstáculos.	Con el dispositivo encendido se procederá a medir diferentes distancias las cuales deberán coincidir con la realidad.	Uno por dispositivo.	Lectura incorrecta de las distancias o la no detección de los objetos empleados en la prueba.
Servomotor de las pinzas.	Movilidad de las pinzas.	Con el dispositivo encendido y a través de la aplicación se deberá probar si éste abre y cierra las pinzas correctamente sin calentarse.	Uno por dispositivo.	No se cierran o se abren las pinzas o no lo realizan correctamente, es decir no llegan a cerrar o abrir del todo.

Matriz de leds.	Encendido y apagado de leds.	Encender y apagar al completo la matriz led.	Uno por dispositivo.	No se enciende la matriz. No se apaga la matriz. Algún led no se enciende.
Servomotores de las patas y controladores PCA.	Posicionamiento de las patas.	Se procederá a probar todas y cada una de las funciones del movimiento mediante la aplicación móvil, además de comprobar la posición individual de cada pata y extremidad.	Uno por dispositivo. Uno por cada pata y extremidad.	No se mueve correctamente. No se posiciona alguna pata o extremidad correctamente.
Batería	Carga, descarga y duración.	Comprobación de tensión máxima y mínima, así como la duración de ésta durante un uso normal. Aguante de la carga durante una semana sin uso.	Uno por dispositivo.	No carga correctamente. Tensiones incorrectas de carga. Descarga involuntaria de la batería. Duración reducida.

Tabla 19: Ensayos de recepción.

6.- ENTREGA

El embalaje del dispositivo deberá ser adecuado para mantener el dispositivo en perfecto estado durante el transporte. Deberá seguir los principios de un Ecodiseño con el fin de conseguir un ahorro de energía y recursos materiales, contemplando al máximo el uso de materiales reciclables.

Puesto que se trata de un dispositivo electrónico, la caja deberá tener un formato que proteja el robot ante posibles impactos y golpes durante el transporte. Además, se deberá incluir en el interior un espacio reservado para el cargador de baterías que evite su desplazamiento en el interior de la caja, así como rodear el robot con una espuma protectora relleno los huecos libres del embalaje. En la parte superior, se deberá incluir el manual del usuario impreso en papel reciclado así como una tarjeta de memoria que incluya los planos 3D del robot con extensión .stl. Al lado del cargador de baterías se adjuntará el kit de piezas de repuesto.

Para evitar que el robot esté en contacto con humedad, polvo u otros agentes externos como animales, insectos o descargas electrostáticas, éste irá sellado con un plástico biodegradable al igual que el cargador de baterías y en su interior tres bolsas de gel de sílice para absorber la humedad ambiental.

Una vez esté el producto embalado correctamente se deberá sellar el embalaje tanto por arriba como por abajo mediante cinta precintadora evitando rugosidades y burbujas.

DOCUMENTO NÚMERO 3: PLIEGO DE CONDICIONES

El exterior del embalaje deberá contener:

- Fabricante, lugar de fabricación y ensamblaje.
- Nombre y/o modelo del producto.
- Imagen del producto.
- Símbolo de reciclado del embalaje.
- Código de barras del producto.
- Símbolo de producto frágil.
- Símbolo de contiene productos con baterías.
- Descripción en diferentes idiomas de qué contiene el embalaje.
- Decoración adecuada al tipo de producto.

El almacenamiento del producto deberá ser en una atmósfera con una humedad no superior al 60% y con una temperatura inferior a 30 °C y superior a 3 °C. Además, se recomienda que la altura de almacenamiento no sea superior a 2.5 metros para evitar caídas o peso excesivo sobre el producto.

No se deberá dejar el producto embalado expuesto a la intemperie.

Con el producto irá adjunta una copia de la factura.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

DOCUMENTO NÚMERO 4: PRESUPUESTO

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

Álvaro Alberto Giner

TUTORIZADO POR

Leopoldo Armesto Ángel










CURSO ACADÉMICO: 2019/2020

Hoja en blanco dejada a propósito




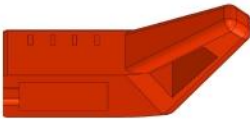
DOCUMENTO NÚMERO 4: PRESUPUESTO

Grupo	Código	Nombre del elemento	Imagen	Descripción	Cantidad	Precio Unitario (€)	Total	
PARTES ESTRUCTURALES PLÁSTICAS (A1+A2+A3)							13,99 €	
C H A S I S	A1. CHASIS							4,28 €
	2111	Chasis inferior delantero		Parte plástica. Material ABS. Relleno 20%. Altura de capa 0,2 mm.	1	1,040	1,04 €	
	2112	Chasis inferior trasero		Parte plástica. Material ABS. Relleno 20%. Altura de capa 0,2 mm.	1	0,710	0,71 €	
	2121	Chasis superior delantero		Parte plástica. Material ABS. Relleno 20%. Altura de capa 0,2 mm.	1	0,960	0,96 €	
	2122	Chasis superior trasero		Parte plástica. Material ABS. Relleno 20%. Altura de capa 0,2 mm.	1	0,680	0,68 €	
	213	Pilar		Parte plástica. Material ABS. Relleno 20%. Altura de capa 0,2 mm.	4	0,090	0,36 €	
	214	Soporte Batería		Parte plástica. Material PLA. Relleno 15%. Altura de capa 0,2 mm.	1	0,250	0,25 €	
	215	Soporte Ultrasonidos		Parte plástica. Material ABS. Relleno 20%. Altura de capa 0,2 mm.	1	0,280	0,28 €	

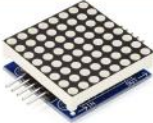









DOCUMENTO NÚMERO 4: PRESUPUESTO

Grupo	Código	Nombre del elemento	Imagen	Descripción	Cantidad	Precio Unitario (€)	Total
B O C A	A2. BOCA						0,47 €
	2211	Engranaje Motriz		Parte plástica. Material ABS. Relleno 100%. Altura de capa 0,1 mm.	1	0,020	0,02 €
	2212	Engranaje Derecho		Parte plástica. Material ABS. Relleno 100%. Altura de capa 0,1 mm.	1	0,020	0,02 €
	2213	Engranaje Izquierdo		Parte plástica. Material ABS. Relleno 100%. Altura de capa 0,1 mm.	1	0,020	0,02 €
	222	Soporte Boca		Parte plástica. Material ABS. Relleno 100%. Altura de capa 0,2 mm.	1	0,050	0,05 €
	2231	Estabilizador Derecho		Parte plástica. Material ABS. Relleno 100%. Altura de capa 0,1 mm.	1	0,020	0,02 €
	2232	Estabilizador Izquierdo		Parte plástica. Material ABS. Relleno 100%. Altura de capa 0,1 mm.	1	0,020	0,02 €
	224	Bulón		Parte plástica. Material ABS. Relleno 100%. Altura de capa 0,1 mm.	2	0,010	0,02 €
	2251	Mandíbula Derecha		Parte plástica. Material ABS. Relleno 15%. Altura de capa 0,2 mm.	1	0,150	0,15 €
	2252	Mandíbula Izquierda		Parte plástica. Material ABS. Relleno 20%. Altura de capa 0,2 mm.	1	0,150	0,15 €

DOCUMENTO NÚMERO 4: PRESUPUESTO

Grupo	Código	Nombre del elemento	Imagen	Descripción	Cantidad	Precio Unitario (€)	Total
P A T A S	A3. PATAS						9,24 €
	231	Hombro		Parte plástica. Material ABS. Relleno 20%. Altura de capa 0,2 mm.	6	0,29	1,74 €
	232	Unión Hombro Codo		Parte plástica. Material ABS. Relleno 20%. Altura de capa 0,2 mm.	6	0,31	1,86 €
	233	Codo		Parte plástica. Material ABS. Relleno 20%. Altura de capa 0,2 mm.	6	0,45	2,70 €
	234	Muñeca		Parte plástica. Material ABS. Relleno 20%. Altura de capa 0,2 mm.	6	0,49	2,94 €









DOCUMENTO NÚMERO 4: PRESUPUESTO

Grupo	Código	Nombre del elemento	Imagen	Descripción	Cantidad	Precio Unitario (€)	Total
ELECTRÓNICA							17,41 €
ELECTRÓNICA	122	Matriz Led 8x8		Matriz de 8x8 leds rojos en cátodo común basada en MAX7219. Dimensiones 32 x 32 mm.	1	1,00	1,00 €
	123	Led Blanco		LED blanco de 5mm, 1,8 -2,2V 20mA	2	0,02	0,04 €
	124	Módulo Bluetooth HC-06		Configurable mediante comandos AT. Nivel de tensión de alimentación: 3,3 V a 6 V. Frecuencia de banda de 2,4 GHz. Potencia de transmisión 4dBm. Clase 2.	1	2,00	2,00 €
	125	Microprocesador Arduino Nano		Microcontrolador ATMEGA328. Tensión de alimentación 5V. Rango de alimentación 7-12 V. Pines digitales 14 (6 PWM). Pines analógicos 8. Corriente Máxima por pin 40 mA. Memoria flash de 32 KB. Reloj 16 MHz.	1	1,50	1,50 €
	126	Placa Conexiones Arduino Nano		Placa adaptadora de pines para Arduino Nano. Tamaño 60 x 53 x 13 mm.	1	1,50	1,50 €
	127	Controlador PCA-9685		Módulo de 12 bits y 16 canales para control PWM mediante la interfaz de I2C. Posibilidad de conectar hasta 6 placas en serie.	2	2,00	4,00 €
	129	Sensor Ultrasonidos		Módulo medidor de distancias. Tensión de trabajo 5VDC. Ángulo de trabajo 15 grados. Rango 2 cm-450 cm. Precisión 0,3 cm	1	3,00	3,00 €
	130	Diodos MUR460		Diodo rectificador. Tensión máxima 600 V. Corriente Máxima 4 A. Ultrarrápido.	8	0,30	2,40 €
	135	Salva Baterías		Detector de baja tensión para baterías Li-Po además de multímetro. Posibilidad de programar tensión de alarma.	1	1,95	1,95 €
	141	Resistencia 220 Ohms		Resistencia de 220 ohm. Tolerancia de un 5%. Potencia 1/4 W.	2	0,01	0,02 €

DOCUMENTO NÚMERO 4: PRESUPUESTO

Grupo	Código	Nombre del elemento	Imagen	Descripción	Cantidad	Precio Unitario (€)	Total (€)
F U E R Z A	ACTUADORES						81,60 €
	120	Servo-Motor MG996R		Tamaño: 40 x 19 x 43 mm. Peso 67 g. Velocidad sin carga 0,17 seg/60 grados a 4,8V. Velocidad sin carga 0,13 seg/60 grados a 6V. Voltaje 4,8 V - 7,2 V. Par máximo a 6V de 13 Kgcm.	18	4,40	79,20 €
	121	Servo-Motor SG90		Tamaño: 21,5 x 11,8 x 22,7 mm. Peso 9 g. Velocidad sin carga 0,12 seg/60 grados a 4,8V. Voltaje 4,8 V - 6 V. Par máximo a 6 V. 1,5 Kgcm	1	2,40	2,40 €

DOCUMENTO NÚMERO 4: PRESUPUESTO

Grupo	Código	Nombre del elemento	Imagen	Descripción	Cantidad	Precio Unitario (€)	Total
E L É C T R I C A	ELÉCTRICA						67,71 €
	128	Placa PCB		Placa PCB perforada de 2x8cm, doble cara	1	0,50	0,50 €
	131	Pareja Conexión XT60		Conectores para Li-Po, XT60, M-H	1	1,76	1,76 €
	132	Cable 1 mm2		Cable de audio de baja impedancia. 1 mm2 de sección.	1,5	0,64	0,96 €
	133	Interruptor		Conmutador interruptor 3 pines. Voltaje/Corriente 220VAC/6A. Agujero de montaje: 6 mm. Tamaño: 13 x 7 x 9 mm	1	1,00	1,00 €
	134	Batería Li-Po 2S 2200mAh		Batería tipo LiPo. 2S (7,4 V). Carga de 2200 mAh. Descarga continua de 30C. Descarga pico 60C	1	16,99	16,99 €
	136	Cargador Baterías Li-Po Universal		Cargador LiPo Universal. Incluye alimentación y diferentes conexiones. Diferentes funcionalidades.	1	35,99	35,99 €
	137	Cable DUPONT		Cable DuPont de todas las variedades. MM HM HH	26	0,02	0,52 €
138	Set Cable extensor Servo-Motor		Pack de cable extensor para servos. Sección del cable tipo 22AWG. Tamaños de 75 mm y de 150 mm.	1	9,99	9,99 €	

DOCUMENTO NÚMERO 4: PRESUPUESTO

Grupo	Código	Nombre del elemento	Imagen	Descripción	Cantidad	Precio Unitario (€)	Total
O T R O S	OTROS						2,80 €
	139	Tubo Termorretráctil		Tubo termorretráctil. Resistente a agentes externos. Punto de fusión 120 grados Celsius.	0,2	5,00	1,00 €
	140	Bridas		Bridas de nailon. Fuerza de tensión 18,2 Kg. Resistente a la exposición a los rayos UV y agentes externos. Longitud de 150 mm x 3,6 mm.	50	0,01	0,50 €
	142	Estaño		Hilo de estaño de 1,5 mm de sección. ISO 9453:2006	20	0,05	1,00 €
	143	Barras Termofusibles		Barra termofusible. Rango de fusión de 160 a 180 grados Celsius. Diámetro de 11 mm. Longitud de 100 mm.	2	0,15	0,30 €

DOCUMENTO NÚMERO 4: PRESUPUESTO

Grupo	Nombre del elemento	Descripción	Cantidad (horas)	Precio Unitario (€)	Total
M A N O D E O B R A	MANO DE OBRA TOTAL (B1+B2)				12.560,00 €
	B1. MANO DE OBRA DEL INGENIERO				12.000,00 €
	Horas en el diseño estructural del robot.	Desarrollo de las partes plásticas estructurales del robot hexápodo	200	40	8.000,00 €
	Horas de programación del robot	Desarrollo del código del robot hexápodo.	40	40	1.600,00 €
	Horas de desarrollo de la aplicación móvil	Desarrollo del código de la aplicación móvil.	50	40	2.000,00 €
	Horas de comprobación del correcto funcionamiento del producto	Pruebas del correcto funcionamiento.	10	40	400,00 €
	B2. MANO DE OBRA DEL TÉCNICO				560,00 €
	Horas de montaje	Montaje de las partes plásticas, eléctricas y electrónicas.	10	20	200,00 €
	Horas de lijado de piezas	Lijado de piezas que presentan irregularidades en su superficie.	3	20	60,00 €
	Horas de supervisión y mantenimiento	Supervisión de la impresión, cambio de filamento, recalibración de la máquina etc.	15	20	300,00 €

DOCUMENTO NÚMERO 4: PRESUPUESTO

Grupo	Nombre del elemento	Descripción	Valor
A U M E D I O S A S E S	MEDIOS AUXILIARES		5%
	Impresora 3D	Utilizada para imprimir las partes plásticas.	
	Software	Uso de programas como Microsoft Word, Excel, PowerPoint, SolidWorks y Cura.	
	Ordenador portátil	Herramienta con la cual se han utilizado los programas para poder desarrollar el proyecto.	
	Destornilladores	Uso para atornillar los fijadores plásticos de los servomotores , así como para fijar ciertas partes electrónicas mediante tornillos.	
	Pistola de termofusible	Uso para fijar ciertas partes electrónicas como puede ser el módulo Bluetooth, los LED's blancos, la Matriz, el sensor de proximidad o ciertos cables a las placas de conexiones.	
	Plástico desechado en andamios de las piezas	Plástico desechado necesario para poder imprimir aquellas piezas con voladizos.	
	Alicates/Tenazas	Usados para cortar cables, quitar ciertos andamios difíciles de retirar con las manos y apretar correctamente las bridas.	
	Soldador de estaño	Utilizado para soldar las conexiones eléctricas y electrónicas del proyecto.	

DOCUMENTO NÚMERO 4: PRESUPUESTO

R P E R S E U S M U E P N U E D S E T L O	TOTAL DEL PRESUPUESTO	13.380,69 €
	TOTAL PARTES PLÁSTICAS	13,99 €
	TOTAL ELECTRÓNICA	17,41 €
	TOTAL FUERZA	81,60 €
	TOTAL ELÉCTRICA	67,71 €
	TOTAL OTROS	2,80 €
	TOTAL MANO DE OBRA	12.560,00 €
	TOTAL MEDIOS AUXILIARES	637,18 €

Hoja en blanco dejada a propósito

Hoja en blanco dejada a propósito