



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Industrial

Diseño e implementación mediante impresión 3-D de un
dispositivo para el control de velocidad de objetos basado
en Arduino y ultrasonidos

Trabajo Fin de Grado

Grado en Ingeniería en Tecnologías Industriales

AUTOR/A: Rubia Duque, Carlos de la

Tutor/a: Gracia Morán, Joaquín

Cotutor/a: Baraza Calvo, Juan Carlos

CURSO ACADÉMICO: 2023/2024

Agradecimientos

A mis padres y mi hermana,
por ser mis pilares fundamentales.

A mis tutores,
por la ayuda, paciencia y dedicación que me han dado

A mis amigos,
por todo el apoyo que me han dado durante estos años.

Gracias por todo.

Resumen

Este proyecto se centra en el diseño y la implementación de un velocímetro digital utilizando Arduino, además del diseño e impresión 3D de su carcasa.

El objetivo principal es crear un dispositivo capaz de medir y mostrar la velocidad de un objeto en movimiento a tiempo real. Para ello se ha utilizado un sensor de ultrasonido conectado a un Arduino, permitiendo la lectura precisa de la velocidad y su visualización en una pantalla.

El proyecto combina conceptos de electrónica, programación e impresión 3D, para lograr un dispositivo que cumpla con los objetivos establecidos.

Abstract

This project focuses on the design and implementation of a digital speedometer using Arduino, along with the design and 3D printing of its casing.

The main objective is to create a device capable of measuring and displaying the real-time speed of a moving object. To achieve this, an ultrasonic sensor connected to an Arduino has been used, enabling precise speed reading and visualization on a screen.

The project combines concepts from electronics, programming, and 3D printing to achieve a device that meets the established objectives.

Resum

Aquest projecte es centra en el disseny i la implementació d'un velocímetre digital utilitzant Arduino, a més del disseny i la impressió 3D de la seua carcassa.

L'objectiu principal és crear un dispositiu capaç de mesurar i mostrar la velocitat d'un objecte en moviment en temps real. Per a aconseguir-ho, s'ha utilitzat un sensor d'ultrasons connectat a un Arduino, permetent una lectura precisa de la velocitat i la seua visualització en una pantalla.

El projecte combina conceptes d'electrònica, programació i impressió 3D per a aconseguir un dispositiu que compleix els objectius establits.

Índice

- **Memoria**

- **Presupuesto**

- **Anexo 1** (Programación, código completo)

- **Anexo 2** (Planos del proyecto)

Memoria

Índice de contenido de la memoria

ÍNDICE DE FIGURAS	8
ÍNDICE DE TABLAS	9
1. INTRODUCCIÓN.....	10
1.1. MOTIVACIÓN	10
1.2. OBJETIVOS.....	10
2. Fundamentos teóricos del proyecto.....	11
2.1. Sensor de ultrasonido	11
2.2. Arduino.....	12
2.3. Velocímetros en la actualidad	16
2.4. Características de dispositivos similares.....	17
3. Componentes	20
3.1. Hardware	20
3.2. Elección del hardware.....	21
3.3. Hardware final.....	23
3.4. Conexión de Hardware.....	27
4. Programación.....	30
4.1. Programa implementado.....	31
5. Diseño de la carcasa	37
5.1. Introducción a la impresión 3D	37
5.2. Diseño de la carcasa.....	38
5.3. Despiece	41
5.4. Montaje final.....	45
6. Montaje Final.....	48
6.1. Montaje	48
6.2. Pruebas	49
7. Conclusiones y trabajo futuro.....	50
7.1. Conclusión	50
7.2. Trabajos futuros	51
8. Bibliografía.....	51
9. Cálculo de presupuesto.....	54

ÍNDICE DE FIGURAS

Figura 1. Ejemplo de funcionamiento de un sensor ultrasónico	12
Figura 2. Tipos de placa de Arduino	13
Figura 3. Captura de la web Arduino	15
Figura 4. Cadena de producción	16
Figura 5. Dispositivo similar del mercado.....	17
Figura 6. Portable Digital Stroboscope	18
Figura 7. GLM 50-27 C PROFESSIONAL	19
Figura 8. VELOCITY SPEED GUN	19
Figura 9. Sensor ultrasonido de uso industrial	22
Figura 10. Sensor ultrasonido con conexión por pines.....	22
Figura 11. Arduino nano R3	23
Figura 12. Adafruit Standard LCD 16x2.....	24
Figura 13. Sensor ultrasónico HC-SR04.....	24
Figura 14. LED RGB	24
Figura 15. Potenciómetro	25
Figura 16. Botón	25
Figura 17. Interruptor	26
Figura 18. Batería recargable 18650 de lón-Litio.....	26
Figura 19. Esquema del circuito.....	28
Figura 20. Fotografía del circuito en fase de prueba	29
Figura 21. Fotografía del circuito en fase de prueba	30
Figura 22. Diagrama de flujo de biblioteca y definición de variables.....	32
Figura 23. Diagrama de flujo función "ultrasonicRead"	33
Figura 24. Diagrama de flujo función "setup"	34
Figura 25. Diagrama de flujo de estado y accionamiento del pulsador	35
Figura 26. Diagrama de flujo función "Calculo"	36
Figura 27. Diagrama de flujo del código completo.....	36
Figura 28. Logo Autodesk Inventor.....	39
Figura 29. Captura de pantalla de la aplicación Inventor	39
Figura 30. Logo PrusaSlicer.....	39
Figura 31. Captura de pantalla de la aplicación PrusaSlicer	40
Figura 32. Prototipo 1	41
Figura 33. Prototipo 4	41
Figura 34. Pieza carcasa de la pantalla	42
Figura 35. Tapa de la carcasa de la pantalla	42
Figura 36. Pieza carcasa del sensor	43
Figura 37. Tapa de la carcasa del sensor.....	43
Figura 38. Mango.....	44
Figura 39. Tapa del mango.....	44
Figura 40. Enganche del módulo de la pantalla.....	45
Figura 41. Enganche del módulo de la pantalla (detalle)	45
Figura 42. Enganche del módulo del sensor.....	46
Figura 43. Enganche del módulo del sensor (detalle)	46
Figura 44. Foto 1 ensamblaje	47
Figura 45. Foto 2 ensamblaje	47

Figura 46. Foto de implementación.....	48
Figura 47. Prueba 1 (medidas correctas).....	49
Figura 48. Prueba 2 (distancia de medida).....	49
Figura 49. Prueba de limitaciones	50

ÍNDICE DE TABLAS

Tabla 1. Presupuesto mano de obra.....	54
Tabla 2. Listado y coste de componentes.....	55
Tabla 3. Tiempo y coste de impresión de piezas	55
Tabla 4. Coste de impresión y diseño	55
Tabla 5. Coste de materiales.....	56
Tabla 6. Presupuesto de equipo	57
Tabla 7. Presupuesto general	57

1. INTRODUCCIÓN

1.1. MOTIVACIÓN

El gran número de avances tecnológicos en los últimos años ha llevado a que la medición precisa de la velocidad sea una herramienta esencial en una gran variedad de aplicaciones, desde la seguridad vial hasta la optimización de procesos industriales. Además, estos mismos avances han permitido el desarrollo de sistemas de microprocesadores con una amplia gama de aplicaciones a un precio asequible. Este proyecto se sumerge en el diseño y la implementación de un velocímetro digital, utilizando la versátil plataforma Arduino como base.

La elección de diseñar y desarrollar un velocímetro digital funcional utilizando Arduino permite aplicar los conceptos de electrónica y programación de manera práctica para el desarrollo de un dispositivo operativo que cumpla con unas funciones específicas, y además permite el uso de la impresión 3D como una herramienta de diseño y personalización.

Este proyecto busca ofrecer una solución práctica a través del uso de herramientas ya conocidas y el estudio de otras nuevas para su implementación en la creación de un dispositivo capaz de medir la velocidad de manera precisa. Además, se pretende demostrar que la integración de conceptos tanto de electrónica y programación unidos, a la impresión 3D permiten de manera efectiva satisfacer una necesidad del mundo real.

1.2. OBJETIVOS

Este proyecto se separa en una serie de objetivos tanto para el desarrollo y diseño del velocímetro, como para la documentación de todo el proceso. Dichos objetivos son los siguientes:

- **Selección de componentes y diseño del circuito:** Montaje de un circuito electrónico tras la selección de los componentes apropiados que, añadidos a un sensor de ultrasonido y junto con un microcontrolador, sea capaz de medir la velocidad de un objeto en movimiento.
- **Programación del microcontrolador:** Desarrollar el código necesario para que, a partir del circuito diseñado, el microcontrolador sea capaz de leer los datos provenientes del sensor ultrasonido, los procese y los represente en la pantalla de manera útil para el usuario, además de mandar las señales pertinentes al resto de receptores que puede haber en el circuito.
- **Calibración y precisión:** Calibrar el sensor de ultrasonido usando el código del Arduino para intentar minimizar el error en la lectura de la velocidad, para de esta forma obtener un dispositivo más fiable.
- **Diseño de la carcasa:** Con la ayuda de una aplicación de modelado digital, diseñar una carcasa para el posterior ensamblado de los componentes que forman parte del circuito del velocímetro.

- **Impresión 3D y ensamblaje:** Utilizando la tecnología de impresión 3D, se construirá la carcasa anteriormente diseñada para su posterior ensamblaje, y así poder dar al dispositivo un cuerpo funcional y totalmente operativo.
- **Pruebas y validación:** Tanto durante la fase de programación del Arduino, como después del ensamblaje del circuito en la carcasa, es necesario realizar una serie de pruebas y validaciones para comprobar el correcto funcionamiento, la precisión de la velocidad y la respuesta del sensor a las diferentes condiciones de movimiento.
- **Documentación:** Realizar un documento en el que plasmar todas las partes del proceso del proyecto, incluidos el diseño del circuito, el diseño 3D y su posterior impresión y los resultados de las pruebas.
- **Aprendizajes y aplicación de los conocimientos:** Durante el proceso de este proyecto habrá que aplicar una serie de conocimientos y aprender algunos otros nuevos, para poder llegar a un producto final que cumpla con las características esperadas.

2. Fundamentos teóricos del proyecto

El sensor ultrasónico o sensor de ultrasonido, es un tipo de sensor muy útil a la hora de medir distancias o detectar obstáculos basándose en la emisión y recepción de ultrasonidos. En este caso, el sensor se va a utilizar para medir la velocidad de un objeto en movimiento. Esto se llevará a cabo con la ayuda del Arduino y el código necesario, que calculará la velocidad que tiene un objeto mediante la repetición de mediciones de la distancia a la que este se encuentra del sensor y el tiempo que haya transcurrido entre dichas medidas.

Para entender un poco mejor el proyecto que se va a realizar, y el funcionamiento del dispositivo que se va a fabricar, a continuación, se expondrá de manera resumida el funcionamiento del sensor de ultrasonido, una breve explicación de lo que es la plataforma Arduino y se hablará del uso de los velocímetros en la actualidad.

2.1. Sensor de ultrasonido

Un sensor de ultrasonido es un dispositivo que, utilizando ondas sonoras de alta frecuencia (ultrasonidos), es capaz de medir la distancia entre un objeto y el propio sensor. Estos sensores tienen diversas utilidades en aplicaciones donde es necesaria la medición de distancias con precisión. [1]

El funcionamiento básico de estos sensores comienza con la emisión de una onda corta de alta frecuencia (desde los 20KHz). Estas ondas son generadas en el sensor por un transductor piezoeléctrico que convierte la energía eléctrica en energía acústica.

Las ondas ultrasónicas viajan por el aire a la velocidad del sonido hasta que se encuentran con el objeto (Figura 1), en el cual rebotan y vuelven hasta el sensor.

El sensor posee un transductor, que es el que se encarga de recibir las ondas reflejadas y de transformarlas nuevamente en señales eléctricas. [3]

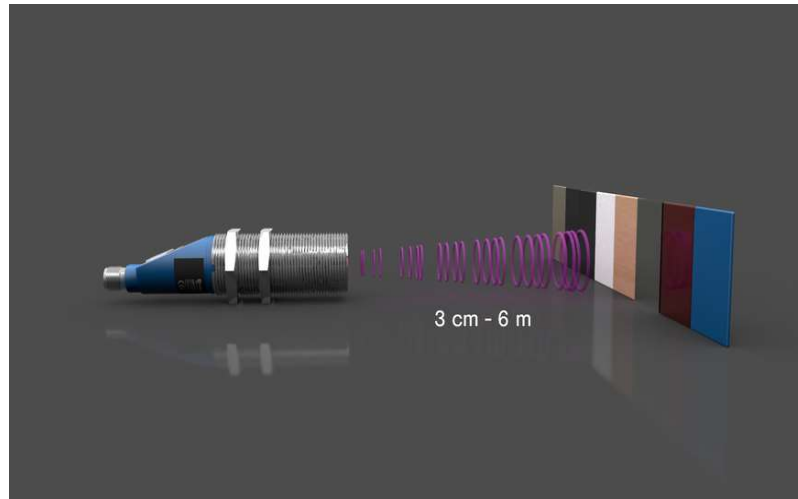


Figura 1. Ejemplo de funcionamiento de un sensor ultrasónico
Fuente: Referencia bibliográfica [2]

Dado que la velocidad del sonido es constante, el tiempo transcurrido entre la emisión y la recepción de dichas ondas es el dato que mide el sensor. Con ambos datos, es sencillo calcular la distancia que existe entre el sensor y el objeto en el que han rebotado las ondas.[4]

Después de esto, un computador (un microcontrolador Arduino en el caso de este proyecto) es el que se encarga de transformar esta información en una medida de distancia legible. Esta información posteriormente se puede utilizar para ser mostrada por pantalla (como es el caso de este proyecto) o para activar acciones específicas, como puede ser abrir una puerta o detener un vehículo.

A partir de este concepto, y dependiendo de la complejidad del sensor o los sensores, el sensor ultrasónico tiene múltiples funciones y utilidades en la industria debido a su gran capacidad de medir distancias de manera precisa y confiable. Hoy en día se utiliza en la mayoría de los vehículos en las herramientas de seguridad de conducción, como la ayuda al estacionamiento o la frenada automática por proximidad. Como ejemplo de usos más industriales, se emplea para la medida de la velocidad de fabricación de múltiples objetos y así llevar el control de funcionamiento de la maquinaria.

2.2. Arduino

Arduino es una plataforma de creación electrónica de código abierto, la cual ofrece hardware y software libre, que ha sido diseñada para simplificar la creación de proyectos para los creadores y desarrolladores. Esta plataforma ofrece diferentes microordenadores, para que después los usuarios puedan darle una infinidad de usos diferentes a cada uno de ellos. [5]

La presentación de hardware y software de código abierto proporciona, por un lado, acceso gratuito y público a todos los códigos desarrollados por diversos programadores (Figura 3) que deseen compartir

sus proyectos. Por otro lado, permite el acceso al diseño de todas sus placas, lo que facilita la creación personalizada de una amplia variedad de proyectos mediante el aprovechamiento de esta herramienta.

Esto lleva a que existan diferentes tipos de placas de Arduino, todas diferentes, pero igualmente funcionales a partir de la misma base. La placa contiene un microcontrolador, que es el cerebro del sistema, y pines de entrada y salida donde se conectan los diferentes componentes electrónicos. El hecho de que existan diferentes placas es solo para poder elegir la que se ajuste mejor al proyecto que se quiera realizar, en función de diferentes características que presentan cada una de las placas (Figura 2), como número de pines (según el número de dispositivos a conectar), memoria del microprocesador (según la complejidad de la tarea a realizar), etc.



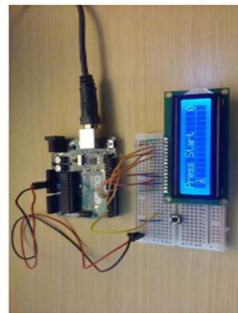
Figura 2. Tipos de placa de Arduino
Fuente: Referencia bibliográfica [6]

Usando el entorno de desarrollo de Arduino, se crea el código, el cual define cómo el microcontrolador interactúa con los componentes conectados en los diferentes pines de este. Este código se carga en la placa a través de un cable USB o una conexión de características similares. Una vez cargado, el microcontrolador ejecuta el código repetidamente según las instrucciones definidas.

Los diferentes pines de salida y entrada que se encuentran en la placa permiten la conexión de diferentes sensores y actuadores, además de otros tipos de componentes. El microcontrolador se encarga de recibir las diferentes señales de entrada que llegan desde los sensores, y controla la salida de los actuadores según lo definido en el programa. De esta forma se ajusta el funcionamiento del circuito según las necesidades del diseñador.

Por ejemplo, podría leer la señal de un sensor de temperatura y encender un led si esta pasa por encima de un valor programado o como es el caso de este proyecto, calcular la velocidad de un objeto a través de la repetición de la lectura de su posición y representar dicha velocidad en una pantalla.

Esta serie de características convierte a Arduino en una plataforma muy personalizable, permitiendo la creación de proyectos simples y complejos. Además, la comunidad posee una amplia cantidad de ejemplos, bibliotecas y tutoriales para ayudar en el aprendizaje, uso y construcción de los proyectos.



jumpman lcd game

Getting started by crepeguy

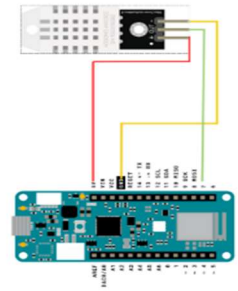
3713 · 0 · 3



flix

Showcase by okalachev

3114 · 0 · 5



Temperature Monitoring with Arduino IoT Cloud using DHT22

Tutorial by attar1

1899 · 0 · 1



Full-sized photovoltaic system

Showcase by arduinojacky

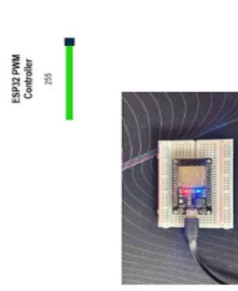
3164 · 1 · 4



Analog Hardware monitor for Windows

Tutorial by nathanb07

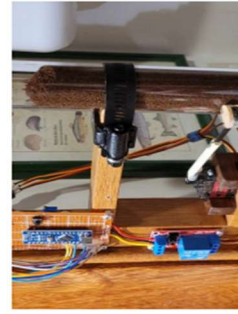
2763 · 0 · 3



ESP32 Web Server with Slider

Tutorial by cotech11

731 · 0 · 0



Cafeino 1.0

Work in progress by mateuseccato

727 · 0 · 0



Arduino Door Lock with Smartphone Flashlight Login (Li...

Tutorial by mircemk

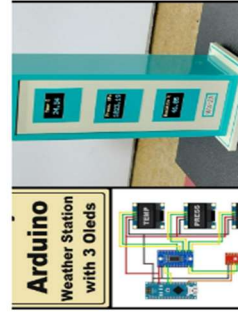
1805 · 0 · 0



Kinetic sand Drawing machine

Tutorial by lirven3579

1169 · 0 · 2



Classic Style Arduino Weather Station with three Oled Displays

Tutorial by mircemk

913 · 0 · 0

Figura 3. Captura de la web Arduino Fuente: Referencia bibliográfica [7]

2.3. Velocímetros en la actualidad

Actualmente, el término velocímetro se aplica en su mayoría al instrumento que mide y muestra el valor de la rapidez instantánea de un vehículo. No obstante, existen otros muchos usos de los sensores de velocidad tanto en la industria como en otros sectores.

En el caso de los usos industriales de los sensores de velocidad, en su mayoría se utilizan en el sector de la automatización. En estas empresas es muy importante poder llevar un control del funcionamiento y la producción. Para ello, se hace uso de múltiples sensores y herramientas para recoger todos los datos posibles acerca de cómo va la producción (Figura 4). Dentro de estos sensores, uno de los más importantes es el sensor de la velocidad, ya que este es el que se encarga de recoger los datos de la rapidez con la que se mueven los productos a lo largo de las líneas de producción. El tener el control de la velocidad permite determinar flujos de operación y producción, además de que en términos generales suele ser un recurso barato, de fácil instalación, y con grandes beneficios para la producción.[8]

Dentro del uso de los sensores de velocidad, en el sector industrial existen varios tipos, diferenciándose en cómo funcionan:

- Sensores láser.
- Sensores por cilindro.
- Sensores ópticos.
- Sensores de ultrasonido.

Además, existen diferentes funciones dentro de la industria, desde el control de flujo como se ha comentado anteriormente, hasta la medición de velocidad sin contacto como herramienta de localización de averías para encontrar las causas de los daños en el material, las marcas de fricción, los arañazos, y eliminar los problemas de roturas de las cadenas/cintas de producción.



*Figura 4. Cadena de producción.
Fuente: Referencia bibliográfica [9]*

Fuera del término industrial los sensores de velocidad, además del velocímetro del coche que se ha comentado, existen otros usos muchos usos como por ejemplo en el campo deportivo o el campo de la seguridad.

Dentro del campo deportivo, se utilizan los sensores de velocidad para recoger marcas, ya sean de vehículos dentro de los deportes de motor, como de las velocidades de lanzamiento en deportes como tenis, fútbol, béisbol, golf, etc.

Por otro lado, dentro del campo de la seguridad, los sensores de velocidad se utilizan para el control de velocidad de los vehículos (Figura 5); en su mayoría son la base del funcionamiento de los radares de tráfico.



*Figura 5. Dispositivo similar del mercado
Fuente: Referencia bibliográfica [10]*

2.4. Características de dispositivos similares

Ya que en el ámbito industrial es más habitual la medición de la velocidad de grandes cadenas de producción, ya sea de los productos o de algunas de las piezas de maquinarias, no se suelen encontrar dispositivos que tengan unas características similares a las de este proyecto. No obstante, a continuación, se van a presentar una serie de ejemplos de dispositivos similares para posteriormente entender las ventajas que presenta el dispositivo diseñado en el proyecto.

Omega Engineering no tiene en su catálogo ningún velocímetro. No obstante, sí que posee una serie de modelos de tacómetros. Un tacómetro es un instrumento utilizado para medir la velocidad de rotación de un objeto, generalmente un eje o un disco, y aunque esta no es la función del dispositivo que se diseña en este proyecto, las características de los dispositivos se asemejan. La marca presenta dos tipos de tacómetros, unos de sensor láser y otros de sensores estroboscópicos. Los modelos de sensor láser tienen un coste de entre 270 y 295 dólares [12][13], mientras que los modelos de sensor estroboscópico tienen un coste de entre 860 y 870 [14] dólares según el modelo. Ya que el tacómetro lo que mide son revoluciones, no nos es de utilidad la comparación con las velocidades que son capaces

de medir los diferentes modelos. Sin embargo, y ya que uno de los modelos de sensor estroboscópico, tiene una forma geométrica similar a la del dispositivo del proyecto (Figura 6), sí que se puede hacer una comparación de los tamaños. En la hoja técnica del producto se puede ver que el dispositivo en cuestión es de dimensiones mayores que la de este proyecto y que puede llegar hasta el peso de 1,1Kg.



*Figura 6. Portable Digital Stroboscope
Fuente: Referencia bibliográfica [15]*

La marca Grainger también presenta una serie de modelos de tacómetros. Dentro de cada modelo, presentan variaciones que se diferencian en algunas de sus características. Todos están entre los 120 y los 400 dólares. [16]

Por otro lado, también existen herramientas de uso profesional que consisten en la medida de distancia mediante el uso de un sensor láser. Aunque la finalidad del dispositivo de este proyecto no es la medida de la distancia, sí que es la tarea que lleva a cabo el sensor y sí que se representa en pantalla la distancia a la que está el objeto del cual se está midiendo la velocidad. Por ello podemos comparar las características de dichas herramientas existentes en el mercado.

La marca Bosh [17] presenta una serie de herramientas de medida de distancia con el uso de sensor láser (Figura 7). Estas herramientas presentan un amplio rango de medida, ya que, según el modelo, la distancia mínima que pueden medir esta entre los 5 y los 15 cm, y pueden llegar hasta una distancia máxima de medida de entre 40 y 150m de distancia. En peso, los modelos varían entre los 0,1 y los 0,25 kg. Y, por último, el precio de estas herramientas varía entre los 100 y los 370 euros, dependiendo del modelo.



Figura 7. GLM 50-27 C PROFESSIONAL
Fuente: Referencia bibliográfica [18]

También otros ejemplos son las herramientas de medida de distancia de la marca Fluke, que tiene un par de modelos de características similares a los de Bosch. Estas herramientas presentan las mismas características de dimensiones y capacidad de lectura, llegando a unos costes también similares a los de su competencia en el mercado. [19]

Por último, y como se mencionó en el apartado anterior, aunque no sean de carácter industrial, existen en el mercado algunos dispositivos de medida de velocidad a partir de sensores que presentan características similares a las del proyecto.

La marca Bushnell, aunque enfocada para registrar datos deportivos (Figura 8), presenta una pistola que mide hasta 320 km/h hasta unos 460m de distancia. Con estas especificaciones técnicas su precio en la tienda de la propia marca es de 150 dólares. No obstante, no presenta diferentes opciones de regulación de medidas ni otros ajustes, que podría llevar a un uso más personalizado para cada individuo.



Figura 8. VELOCITY SPEED GUN
Fuente: [20]

En el caso del velocímetro que se va a diseñar en este proyecto, su uso sería más apropiado para el campo de seguridad, ya que se va a diseñar de forma que sea necesario un operador, por lo que, para el ámbito industrial, solo tendría sentido su uso en el caso de alguna revisión o prueba de funcionamiento, y no para la automatización de una cadena de montaje. Por otro lado, el dispositivo

ha llegado a medir velocidades cercanas a los 100km/h. Sin embargo, el sensor solo alcanza a medir objetos que están a 4m de distancia del dispositivo. Esto lleva a que, si el objeto que se quiere medir está a más de 4m o una distancia cercana a esta, empiezan a existir interferencias que llevan a una mala medida de velocidad. Esto hace que el dispositivo diseñado no sea muy apropiado para el ámbito deportivo.

El proyecto quiere demostrar que con el uso del Arduino y la impresión 3D se puede conseguir una herramienta de gran utilidad a un precio más reducido que el que tienen las herramientas de usos similares en el mercado.

3. Componentes

3.1. Hardware

A la hora de fabricar un velocímetro, el hardware que lo compone puede variar según la complejidad o la función que el diseñador quisiera darle al dispositivo final. Por ejemplo, podría disponer de una pantalla en la que mostrar los datos, o simplemente algunos ledes con los que mostrar diferentes colores según la velocidad detectada.

No obstante, los componentes más habituales que suelen poseer los velocímetros de los cuales se ha hablado anteriormente, son los siguientes:

- **Sensor:** Es el que se encargará de medir la distancia entre el velocímetro y el objeto en movimiento. Posteriormente, esta distancia se utilizará para calcular la velocidad. En el caso del dispositivo de este proyecto, será un sensor de ultrasonido. Al utilizar este tipo de sensor, otro componente que es necesario en el dispositivo será un transductor piezométrico, que es el que se encarga de recibir de vuelta la señal enviada por el sensor y poder leer los datos.
- **Placa base:** Es el cerebro del sistema, que se encargará de procesar la información proveniente del sensor y controlar el funcionamiento del resto de componentes. En el caso de este proyecto, se va a utilizar una placa Arduino.
- **Pantalla:** Ya sea una pantalla de tipo LCD, OLED, o cualquier otro tipo de pantalla compatible, es donde se verán representados los datos según se haya programado en el código.
- **Carcasa:** En ella se van a alojar todos los componentes, y a su vez protegerlos, dando forma al dispositivo final.
- **Botones o Interruptores:** Permitirán al usuario activar, desactivar o interactuar con el velocímetro, una vez montado en su carcasa.
- **Resistencias y condensadores:** Componentes electrónicos utilizados para ajustar y estabilizar las señales eléctricas que atravesarán el circuito durante el funcionamiento del dispositivo.
- **Cables y conectores:** Necesarios para conectar los diferentes componentes entre sí, y a la placa Arduino.

- **Fuente de alimentación:** Será la encargada de suministrar energía a nuestro circuito; es decir, a la placa de Arduino y el resto de los componentes. Puede ser una batería, un adaptador de corriente o una fuente de energía.

Como se ha comentado anteriormente, dependiendo del diseño específico, podrían ser necesarios otros componentes como diodos, transistores o reguladores de voltaje.

3.2. Elección del hardware

Ya que no existe una necesidad específica para el dispositivo que se diseña en este proyecto, no se requieren unas características particulares que este deba cumplir. No obstante, este proyecto pretende demostrar que uniendo la tecnología de Arduino y la impresión 3D se puede obtener una herramienta útil de menor coste que las de mercado. Por ello se busca que los componentes del dispositivo sean del menor coste posible, y con una calidad razonable, para así desarrollar una herramienta de bajo coste, que sirva para realizar pruebas de su funcionamiento. Una vez se conocen las características que se quiere que cumpla el velocímetro, se puede comenzar con la elección de los componentes que van a formar el dispositivo.

Desde el comienzo del proyecto hasta la elección final de los componentes, ocurren muchos cambios en la decisión de características que se quiere que cumpla el velocímetro. Se comenzó usando como referencia un diseño ya existente en uno de los múltiples foros de diseños y fabricación a base de Arduino. En este diseño, el dispositivo presenta los componentes básicos para su funcionamiento:

- Una **fuentes de alimentación** la cual proporciona la energía necesaria al sistema para su funcionamiento.
- El **sensor de ultrasonido** el cual se encargaba de recoger los datos de distancia que había hasta el objeto del cual se mide la velocidad.
- Una **pantalla LCD** donde se mostraba la velocidad medida.
- Una **placa Arduino** la cual posee el microprocesador que se encarga del funcionamiento del dispositivo y de los cálculos para obtener los datos de la velocidad del objeto.

Posteriormente, se fue pensando en otras características que se podrían incorporar al diseño existente, que podrían darle un funcionamiento y aspecto diferente al proyecto. Se pensó en darle más control al usuario, que este pudiera decidir cuándo el dispositivo toma las medidas y cuándo no, o que el dispositivo mostrara de alguna forma, aparte de por la pantalla LCD, información sobre las medidas que se obtienen. Por otro lado, también se tuvo en cuenta la potencia y alcance que se buscaba en el dispositivo, lo que conllevó a una búsqueda y estudio de diferentes sensores.

Se descartó por completo el uso de sensores de nivel industrial (Figura 9), debido a la complejidad y precio que suponen, ya que estas características no coinciden con las deseadas en este proyecto.



Figura 9. Sensor ultrasonido de uso industrial
Fuente: Referencia bibliográfica [21]

Esto llevó a la búsqueda directamente de sensores más sencillos (Figura 10) que pudieran ser conectados al sistema de Arduino sin gran complejidad, y sin suponer un gran coste.



Figura 10. Sensor ultrasonido con conexión por pines
Fuente: Referencia bibliográfica [22]

Por otro lado, se planteó cómo el usuario iba a controlar este dispositivo, es decir, si este podría iniciar y/o parar las mediciones en cualquier momento. Para esto, serían necesarios una serie de pulsadores o interruptores que permitirían al usuario el control manual del dispositivo.

Por último, se eligió cómo se representaría la información capturada, y si el velocímetro presentaría algún otro tipo de señal, sonora o luminosa, dependiendo del valor de la medida que se ha obtenido.

También se planteó cual podría ser la forma que tendría el dispositivo, según el uso que se le iba a dar. Finalmente, se decidió darle forma de “pistola”; es decir, se le puso una empuñadura al dispositivo para que fuera más cómodo de manejar para el usuario. Se buscaba un diseño similar a los productos que se habían encontrado en el mercado con las mismas funciones, pero intentando disminuir en la medida de lo posible el tamaño para así reducir el coste en material.

3.3. Hardware final

Finalmente, se decidió que el funcionamiento completo del velocímetro en desarrollo consistiera en que después de que el Arduino calculara la velocidad, a partir de los valores de medida de la distancia obtenidos por el sensor, se mostraría en una pantalla los datos obtenidos de velocidad y distancia del objeto. Además, se iluminaría un led mostrando diferentes colores (rojo y verde) según el rango en el que se encuentre la velocidad medida, siendo verde para cualquier velocidad que no supere una velocidad establecida como “velocidad límite”. Cualquier medición de la velocidad que supere este valor hará que el led se ilumine de color rojo.

Para que el dispositivo final cumpla estas características y funciones, necesita una serie de componentes que permitan su correcto funcionamiento. Estos componentes, tal y como se comentó en el apartado anterior, pueden variar según las elecciones del diseñador. En el caso de este proyecto, los componentes seleccionados para formar el dispositivo en desarrollo son los siguientes:

- ❖ **Arduino Nano R3.** La placa Arduino que será el controlador de nuestro dispositivo. Esta placa Arduino tiene un tamaño reducido que encaja a la perfección en el diseño del dispositivo del proyecto. El dispositivo a diseñar no necesita un gran número de pines en el controlador ya que no son muchos los componentes necesarios para su funcionamiento, y por estas características se decidió por este modelo (Figura 11).



*Figura 11. Arduino nano R3
Fuente: Referencia bibliográfica [23]*

- ❖ **Adafruit Standard LCD 16x2.** La pantalla en la que se podrán leer los datos de velocidad obtenidos. Se eligió esta pantalla ya que presenta una buena compatibilidad en los proyectos de Arduino (Figura 12).



Figura 12. Adafruit Standard LCD 16x2
Fuente: Referencia bibliográfica [24]

- ❖ **Sensor ultrasónico HC-SR04.** El sensor que se encargará de obtener los datos para poder calcular la velocidad. El rango de detección de este módulo de sensor ultrasónico es de 3 cm a 4 m, lo cual es suficiente para las pruebas de funcionamiento y demostración que pretende dar este proyecto. Por esto, y por su simplicidad y bajo coste, son los motivos por los que finalmente se eligió este sensor (Figura 13).



Figura 13. Sensor ultrasónico HC-SR04
Fuente: Referencia bibliográfica [25]

- ❖ **Led.** Servirá para mostrar un código de colores según lo alta que sea la velocidad medida. Se ha elegido un led de pines RGB para poder programar con un código de colores según lo que se quiera expresar con esta señal (Figura 14).



Figura 14. LED RGB
Fuente: Referencia bibliográfica [26]

- ❖ **3 resistencias de 330 ohmios.** Componente importante para ajustar la señal eléctrica que llega hasta el led. Se necesitan resistencias de 330 ohmios para el correcto funcionamiento del led, en este caso serán 3 resistencias por que hace falta una resistencia por cada uno de los pines correspondientes a las entradas R, G y B del LED.
- ❖ **Potenciómetro (Single Turn) 10 kilo ohmios.** Dispositivo que permitirá calibrar o ajustar la sensibilidad de nuestro circuito. Este potenciómetro es necesario para poder regular la potencia que llega hasta la pantalla LCD y que el visionado de los datos sea adecuado (Figura 15).



Figura 15. Potenciómetro
Fuente: Referencia bibliográfica [27]

- ❖ **Accionador en forma de pulsador.** Componente que permitirá al usuario el interactuar con el circuito, activando y deteniendo su funcionamiento. Sin este elemento, el velocímetro no podría reiniciar los datos que presenta en pantalla (Figura 16).



Figura 16. Botón
Fuente: Referencia bibliográfica [28]

- ❖ **Interruptor.** Permitirá al usuario cortar la alimentación de energía del dispositivo, para mantenimiento de energía mientras este no esté en uso (Figura 17).



Figura 17. Interruptor
Fuente: Referencia bibliográfica [29]

- ❖ **2 baterías recargables 18650 de Ión-Litio, 3.7V, 2.6Ah**, que serán la fuente de alimentación del dispositivo final (Figura 18).



Figura 18. Batería recargable 18650 de Ión-Litio
Fuente: Elaboración propia

- ❖ **Porta pilas 18650**. Las baterías recargables se colocan en el porta pilas para suministrar la energía al dispositivo.

Con estos componentes y la carcasa diseñada en este proyecto, finalmente se obtiene un dispositivo que, aunque tiene unas características de medida menores a los que se han mencionado en apartados anteriores, sigue siendo una herramienta de utilidad y bajo coste, como se verá más adelante, en el presupuesto de este proyecto.

3.4. Conexión de Hardware

Durante el diseño y la programación del hardware del dispositivo, se ha hecho uso de unas placas de inserción o placa de pruebas (*protoboard*), las cuales se han usado temporalmente mientras se realizaban todas las pruebas necesarias.

Los pines correspondientes tanto a cada uno de los componentes, como los de entrada al Arduino, deben permanecer con el orden seleccionado a la hora de la programación del código, ya que el código se programa en base a los puertos de entrada y salida de información. Esto lleva a que hay que tener mucha cautela a la hora de pasar el circuito a las placas donde se van a soldar los cables y los diferentes componentes para obtener el circuito final, el cual se introduce en la carcasa final del dispositivo, ya que un cambio por error, en el orden de cualquiera de los pines llevaría a un error en el funcionamiento del código, y por lo tanto a un mal funcionamiento del dispositivo.

Para poder mover el circuito de las *protoboards* a las placas finales, se ha dibujado un esquema del circuito (Figura 19) utilizando la herramienta *tinkercad*.

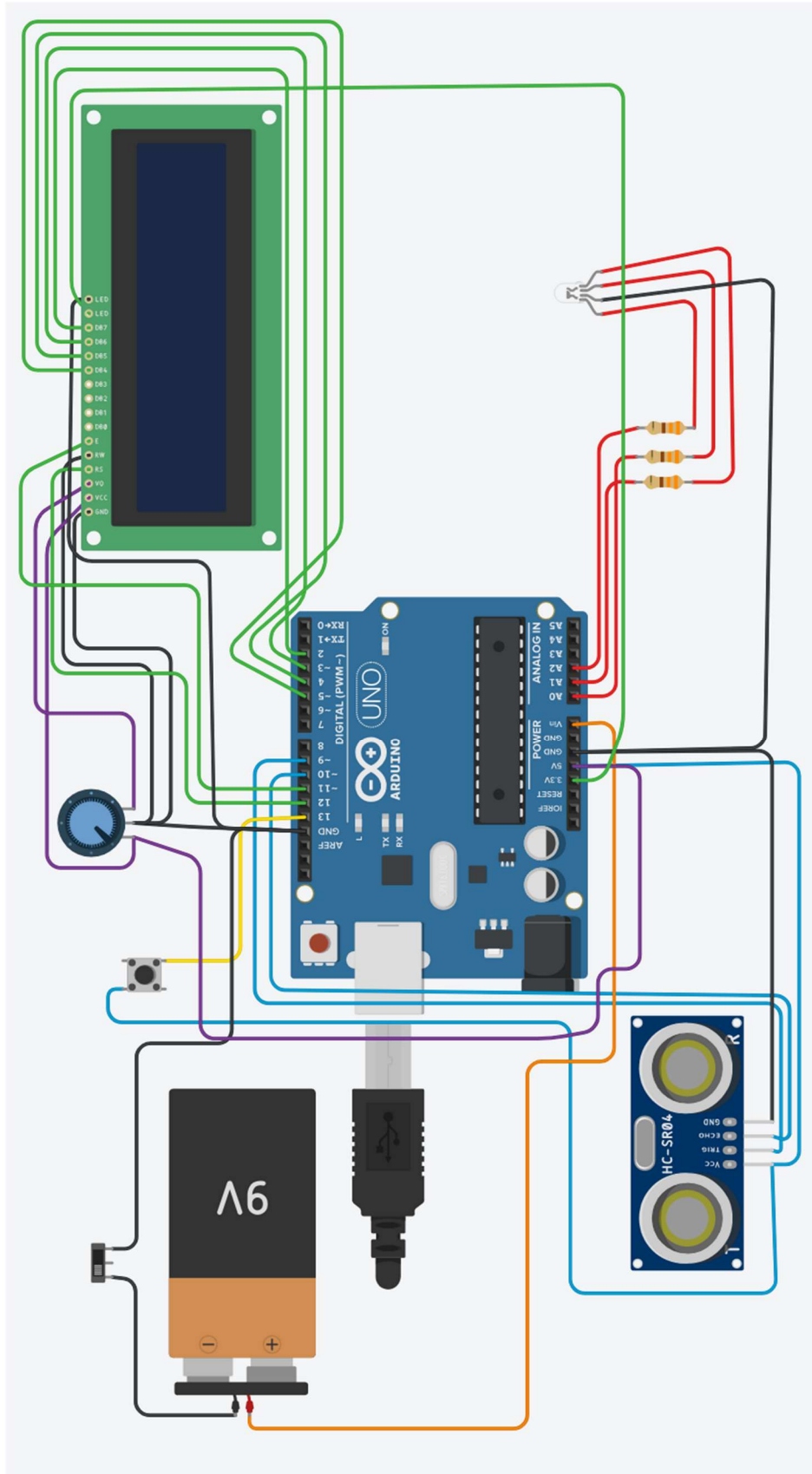
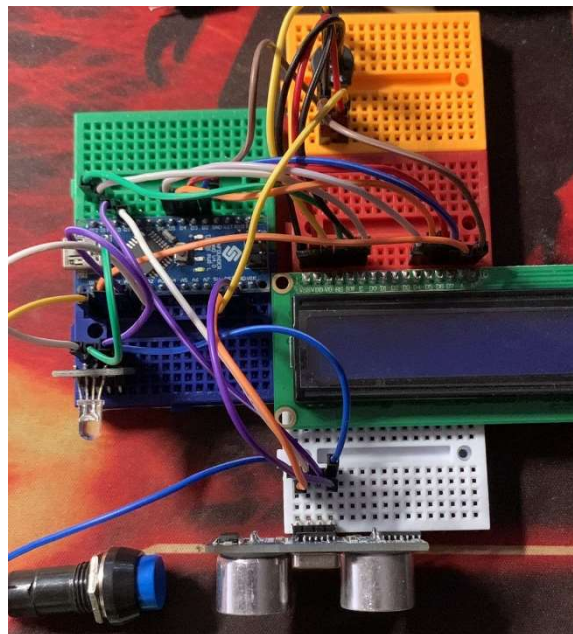


Figura 19. Esquema del circuito
Fuente: Elaboración propia

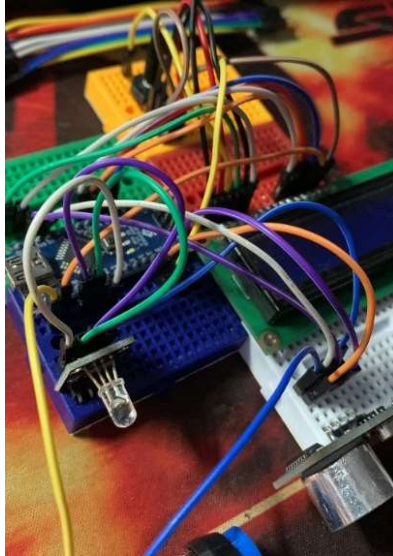
En el esquema se pueden observar los diferentes componentes del dispositivo y cómo quedan las conexiones de los diferentes pines de la placa Arduino a los pines del resto de los componentes. Hay que destacar que el Arduino que aparece en el esquema no es el modelo que se encuentra en el dispositivo. No obstante, como ambos modelos presentan los mismos pines de salida y entrada, se ha utilizado este para el esquema, ya que el software *tinkercad* no tiene representación del otro modelo.

Además, durante este proceso, el Arduino no está conectado a ninguna fuente de alimentación, sino que se conecta directamente al ordenador desde donde se introduce y modifica el código que luego se va a ejecutar.

A continuación, se adjuntan algunas fotografías de la conexión del hardware durante el proceso de programación y pruebas del código:



*Figura 20. Fotografía del circuito en fase de prueba
Fuente: Elaboración propia*



*Figura 21. Fotografía del circuito en fase de prueba
Fuente: Elaboración propia*

Las imágenes de cómo queda la conexión final del dispositivo dentro de la carcasa se encuentran en el apartado de Implementación final de este mismo documento.

4. Programación

Se podría decir que la programación es una de las tareas más importantes que componen este proyecto, ya que esta es la que va a permitir definir cómo va a funcionar el dispositivo final. Una mala programación llevaría a que el Arduino no fuera capaz de leer bien los datos obtenidos por el sensor, que no pudiera calcular la velocidad y/o que los componentes no funcionaran de la manera deseada para el correcto funcionamiento del dispositivo.

En este caso, Arduino presenta su propio lenguaje de programación, por lo que no ha habido ningún tipo de proceso de selección en ese aspecto. Esta tarea consiste en, con el uso de dicho lenguaje de programación, describir las tareas que el controlador tendrá que llevar a cabo, enviando las señales correspondientes al resto de componentes y utilizando los datos recibidos de manera correcta, y así conseguir que en conjunto se realice la tarea deseada.

En este proceso, se realizan diferentes pruebas y modificaciones de código para ajustar el funcionamiento de cada uno de los componentes del circuito hasta conseguir un dispositivo final que cumpla con las características deseadas.

En este apartado se explica de manera detallada el código final con el que se programará el controlador del dispositivo de este proyecto.

4.1. Programa implementado

En el código implementado en el Arduino del dispositivo de este proyecto, se pueden identificar diferentes partes, las cuales llevan al funcionamiento completo y adecuado de los diferentes componentes.

- **Biblioteca y definición de variables (figura 22)**

Este fragmento de código Arduino se utiliza para definir las bibliotecas y variables necesarias. La biblioteca "LiquidCrystal" se utiliza para controlar pantallas LCD, y la biblioteca "TimerOne" se utiliza para configurar un temporizador que mide el tiempo entre eventos.

Posteriormente, crea un objeto "lcd" que se utilizará para controlar una pantalla LCD. Los números que siguen representan los pines conectados a RS, E, D4, D5, D6 y D7 en la pantalla, en ese orden.

Luego, definen dos constantes para los pines del sensor de ultrasonido. El pin "trigPin" se utiliza para enviar señales ultrasónicas y el pin "echoPin" se utiliza para recibir las señales reflejadas.

También se define una constante para el pin de un pulsador pushbutton. Se utiliza para detectar el estado del pulsador. Además, se declara una variable llamada "buttonState" que se utiliza para almacenar el estado del pulsador (pulsado o no pulsado). La inicializa a "0", lo que significa que inicialmente se considera que el pulsador no está presionado.

Posteriormente, se declara una variable llamada "duration" que se utiliza para almacenar la duración de un pulso ultrasónico, es decir, el tiempo que tarda la señal ultrasónica en viajar desde el sensor al objeto y regresar.

Y, por último, se declaran varias variables que se utilizarán para el cálculo de la velocidad. La palabra clave "volatile" se usa para indicar que estas variables pueden cambiar en cualquier momento debido a interrupciones. "distanceanterior" y "distanceactual" almacenan las distancias anteriores y actuales medidas por el sensor. "Speed" se utilizará para calcular la velocidad. "SpeedMax" almacena la velocidad máxima registrada, y "distance" se utiliza para almacenar la distancia medida actualmente.



Figura 22. Diagrama de flujo de biblioteca y definición de variables
Fuente: Elaboración propia

- **Sensor de ultrasonido (Figura 23)**

Este fragmento de código describe una función llamada “ultrasonicRead()”, que se utiliza para medir la distancia entre un sensor ultrasónico y un objeto, y luego devuelve esa distancia.

Primero, se define la función “ultrasonicRead()”, que devuelve un valor flotante (float). Se establece el pin “trigPin” en un estado bajo (LOW) y luego esperan 2 microsegundos. Esto limpia el pin de activación del sensor ultrasónico.

Luego se genera un pulso ultrasónico, el pin “trigPin” se establece en un estado alto (HIGH) durante 10 microsegundos y luego se vuelve a bajar. Este pulso ultrasónico se utiliza para enviar una señal de sonido hacia el objeto que se desea medir.

Posteriormente, se mide el tiempo que tarda la señal ultrasónica en viajar desde el sensor al objeto y regresar. La función “pulseIn()” calcula la duración del pulso de alta (HIGH) en el pin “echoPin” y almacena ese valor en la variable “duration”.

Luego, se calcula la distancia en centímetros utilizando el valor de “duration”. La velocidad del sonido en el aire es aproximadamente 0.034 cm/microsegundo. Dividiendo “duration” por 2 se tiene en cuenta el viaje de ida y vuelta de la señal ultrasónica.

Seguido de esto, se muestra la distancia medida en centímetros en el Monitor Serie para que se pueda leer en el programa y se muestra la distancia medida en la pantalla LCD. Se establece la posición del cursor en la primera línea y se muestra el valor de distancia con un mensaje.

Finalmente, la función devuelve la distancia medida en centímetros como un valor flotante (float).

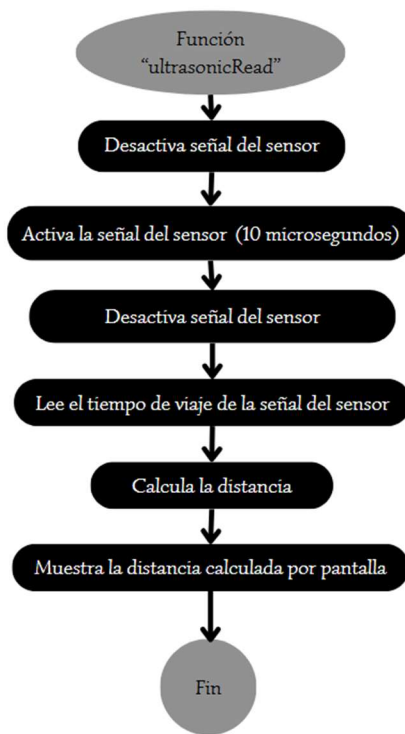


Figura 23. Diagrama de flujo función "ultrasonicRead"
Fuente: Elaboración propia

- **Inicialización del microcontrolador (Figura 24)**

Este fragmento de código corresponde a la función "setup()" en un programa de Arduino y se ejecuta una vez al inicio cuando el microcontrolador se inicia. En resumen, se utiliza para configurar y preparar los componentes y periféricos necesarios para el funcionamiento del proyecto. Esto incluye la inicialización de la pantalla LCD, la configuración de pines como entradas o salidas, la configuración del temporizador y la activación de la comunicación serie.

El temporizador se inicializa para que dispare cada 250 ms (milisegundos) y se asocia a la interrupción "Calculo". Esto puede ser parte de un sistema de temporización en el proyecto para medir la velocidad.



Figura 24. Diagrama de flujo función "setup"
Fuente: Elaboración propia

- **Estado y accionamiento del pulsador (Figura 25)**

Este fragmento de código pertenece a la función "loop()", que se ejecuta continuamente después de que la función "setup()" se haya ejecutado una vez al inicio. Aquí se verifica el estado de un botón y se toman decisiones basadas en su estado.

Primero, se lee el estado del pulsador, que está conectado al pin "buttonPin". La función "digitalRead()" devuelve el estado del pin como HIGH (1) si el botón está presionado y LOW (0) si no lo está. El estado leído se almacena en la variable "buttonState".

Si el pulsador está presionado, se establece la variable "SpeedMax" en 0, lo que significa que se está reiniciando el valor máximo de velocidad registrada y se muestra en pantalla un mensaje que indica "Speed in cm/s" seguido de "0".

En el caso contrario, es decir, si el botón no está presionado (el estado del botón es LOW), el código dentro del bloque "else" se ejecuta:

Si la velocidad actual, almacenada en la variable "Speed", es mayor que la velocidad máxima registrada, almacenada en la variable "SpeedMax" y si la velocidad actual es mayor que la velocidad máxima registrada, se muestra en pantalla un mensaje que indica "Speed in cm/s" seguido del valor de "Speed", lo que refleja la velocidad actual.

Si la velocidad actual es mayor que la velocidad máxima registrada, se actualiza el valor de SpeedMax con la velocidad actual, lo que significa que la velocidad máxima registrada se actualiza a medida que se detectan velocidades más altas.

Además, también se programa que si la velocidad actual es superior a la velocidad indicada como límite (LimitVelocity) el led del sistema se iluminará de color rojo, mientras que, si es inferior a dicha velocidad, se iluminará de color verde.

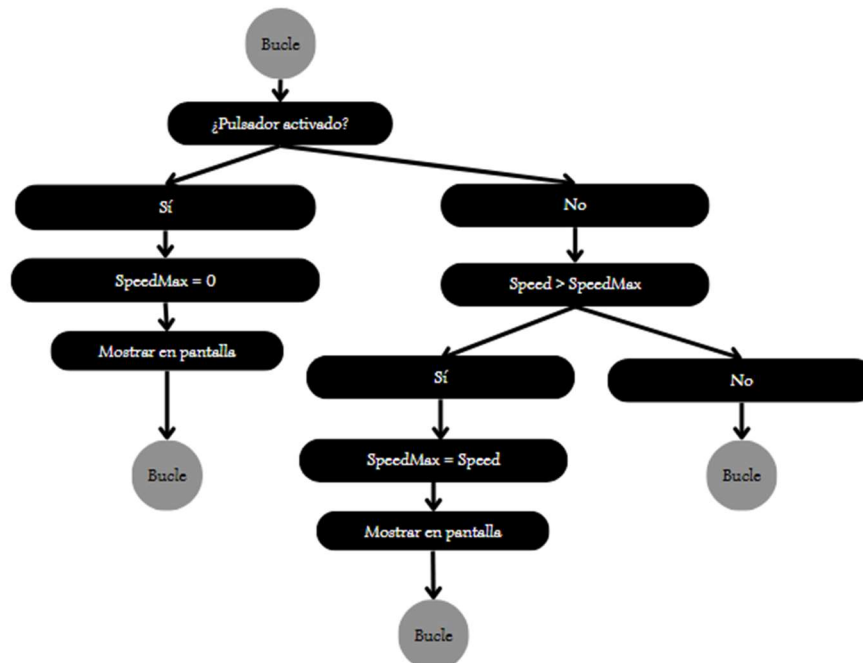


Figura 25. Diagrama de flujo de estado y accionamiento del pulsador
Fuente: Elaboración propia

- **Cálculo de la velocidad (Figura 26)**

En resumen, esta función "Calculo()" se encarga de calcular la velocidad actual en función de las lecturas del sensor ultrasónico. La diferencia en la distancia medida en intervalos de tiempo regulares (en este caso, 1 segundo) se utiliza para calcular la velocidad en centímetros por segundo (cm/s). La velocidad calculada se almacena en la variable "Speed", y la variable "distanceanterior" se actualiza para la próxima iteración del cálculo.

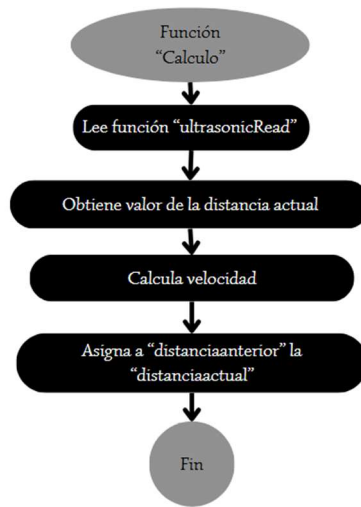


Figura 26. Diagrama de flujo función "Calculo"
Fuente: Elaboración propia

A continuación, se muestra el diagrama de flujo del código completo con el que funciona el dispositivo del proyecto.

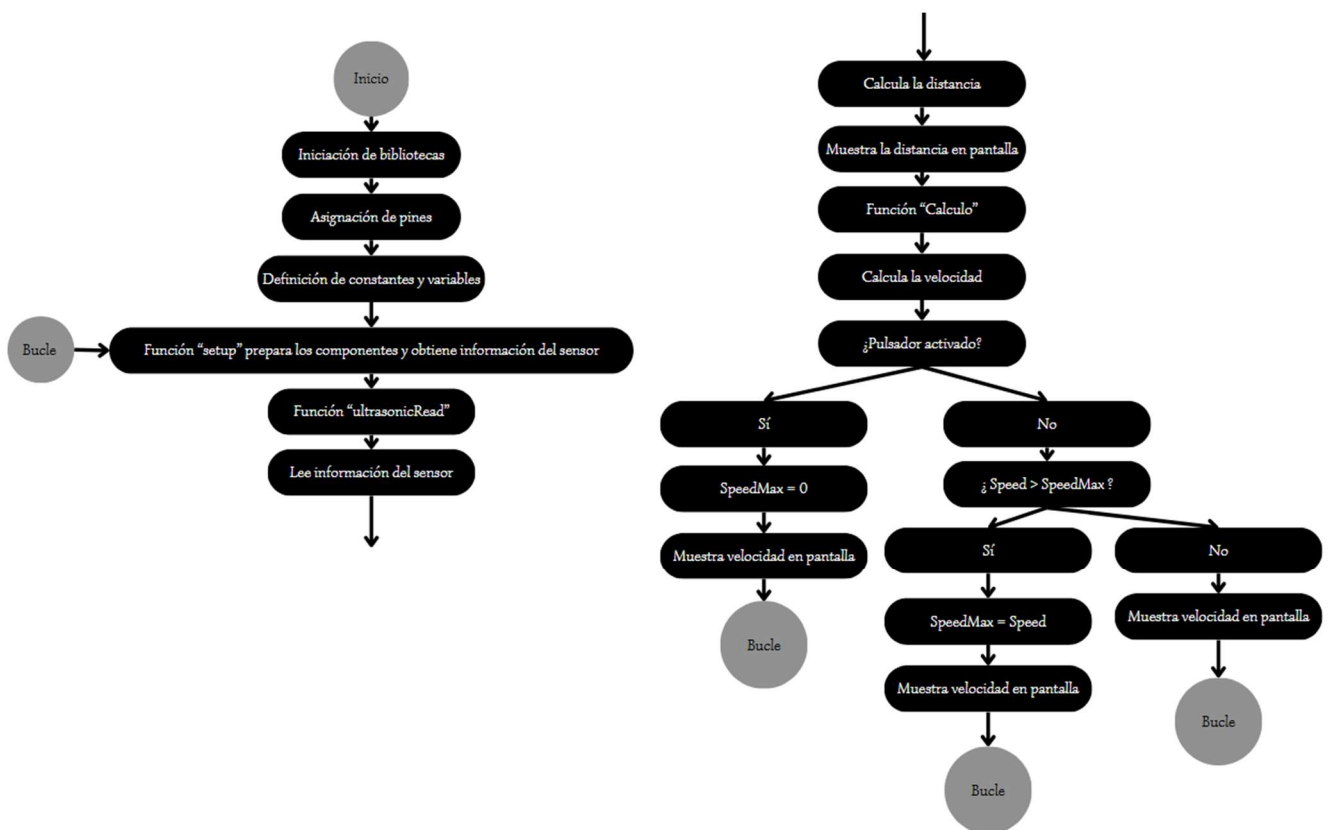


Figura 27. Diagrama de flujo del código completo
Fuente: Elaboración propia

El programa completo se muestra en el "Anexo 1".

5. Diseño de la carcasa

En este apartado se hablará resumidamente sobre la impresión 3D y cómo funciona, ya que es la herramienta que se va a utilizar para el diseño y la creación de la carcasa del dispositivo. Posteriormente se tratará cómo ha sido el proceso que se ha llevado a cabo.

5.1. Introducción a la impresión 3D

Hoy en día, gracias a los diferentes avances tecnológicos, existen diferentes campos donde la impresión 3D se ha convertido en una innovación revolucionaria que está cambiando totalmente la manera en la que se diseña, se crean prototipos y se fabrican objetos. La impresión 3D permite la creación de objetos tridimensionales capa por capa a partir de modelos digitales.

A diferencia de los métodos de fabricación tradicionales, que se basan en la eliminación de material (como cortar o perforar), la impresión 3D se basa en la adición de material de manera gradual para construir un objeto completo desde cero. Esto da la capacidad de construir de manera precisa y personalizada, lo que conlleva a un gran número de aplicaciones en una amplia variedad de campos, desde medicina e ingeniería, hasta en moda y diseño de productos. [11]

Existen 7 categorías de tecnología 3D, recogidas en la norma ISO/ASTM*52900, variando en cómo funciona su impresión. Cada una presenta sus puntos fuertes y sus puntos débiles, ajustándose así a diferentes campos de la industria. Estas categorías son: [Barnatt, 2016]

Extrusión de Material (*Material extrusion*)

- **Descripción:** Un filamento termoplástico sólido es extruido a través de una boquilla caliente y depositado en capas en un estado semifluido. Luego, se enfría para solidificarse.

Fotopolimerización (*Vat Photopolymerization*)

- **Descripción:** Una resina fotosensible en un depósito se endurece selectivamente mediante una fuente de luz.

Proyección de Material (*Material Jetting*)

- **Descripción:** Gotas de material líquido se depositan selectivamente y luego se solidifican. Este método permite imprimir diferentes materiales en un solo objeto.

Proyección de Adhesivo (*Binder Jetting*)

- **Descripción:** Similar a la proyección de material, pero en este caso se depositan gotas de adhesivo selectivamente sobre capas sucesivas de material en polvo.

Fusión de Material en Polvo (*Powder Bed Fusion*)

- **Descripción:** Una fuente de calor funde selectivamente el material en polvo en capas sucesivas. Se logra la fusión de las partículas en polvo dentro de un área de construcción para formar el objeto final.

Deposición Dirigida de Energía (*Directed Energy Deposition*)

- **Descripción:** Similar a la fusión de material en polvo, con la diferencia de que la fuente de calor funde el material a medida que se deposita. Utiliza un rayo láser o haz de electrones alimentado con polvo o hilo de metal para formar las capas del objeto final.

Laminación de Hojas (*Sheet Lamination*)

- **Descripción:** Esta tecnología, ahora en desuso debido a su elevado desperdicio de material, coloca finas láminas de material (papel, plástico o metal) recortadas con una fresadora o un láser de CO₂.

El primer paso de la impresión 3D es el diseño digital, mediante software de diseño asistido por computadora (CAD). Este diseño se descompone en capas individuales. Posteriormente, la impresora 3D construye el objeto diseñado capa a capa. Aunque los métodos de impresión 3D pueden variar, la mayoría implica la deposición de material, como plástico fundido o resina fotosensible, en la forma deseada. Esto variará según el uso que se le vaya a dar al objeto diseñado.

La impresión 3D ofrece una serie de ventajas significativas, como es la creación rápida de prototipos y la producción relativamente sencilla de objetos totalmente personalizados. Esto es especialmente valioso en campos como la medicina, donde se pueden crear prótesis a medida, o en la industria aeroespacial, donde se pueden fabricar componentes livianos y complejos. La industria de la moda también ha adoptado la impresión 3D para la creación de ropa y accesorios únicos. Además, la educación se ha beneficiado al permitir que los estudiantes fabriquen modelos físicos permitiendo una educación más dinámica y práctica de diferentes materias.

A pesar de que la impresión 3D ha avanzado considerablemente en los últimos años, todavía enfrenta desafíos relacionados con la velocidad, los costos y la disponibilidad de materiales. No obstante, la investigación y el desarrollo continuos llevarán a un gran avance de esta tecnología en los próximos años ampliando aún más sus usos y mejorando los actuales.

En resumen, la impresión 3D ha abierto una nueva frontera en la fabricación y el diseño. Su capacidad para producir objetos personalizados y funcionales ha revolucionado nuestra aproximación a la producción y la creatividad. A medida que la tecnología avanza, la impresión 3D promete seguir transformando una amplia gama de industrias y sectores en los próximos años.

5.2. Diseño de la carcasa

En el caso de este proyecto, el diseño digital de la carcasa se hizo utilizando el software "Inventor" y posteriormente para configurar su impresión se utilizó el software "PrusaSlicer".

- **Autodesk Inventor** (Figura 28) es un software de diseño asistido por computadora (CAD) utilizado principalmente para la creación de modelos en 3D, simulaciones y documentación técnica de productos. Este software es especialmente popular en las industrias de ingeniería y manufactura.



Figura 28. Logo Autodesk Inventor
Fuente: Referencia bibliográfica [30]

Autodesk Inventor es una poderosa herramienta CAD que ayuda a ingenieros y diseñadores a crear, simular y documentar productos de manera eficiente y precisa, optimizando el proceso de diseño y reduciendo costos de producción. (Figura 29)

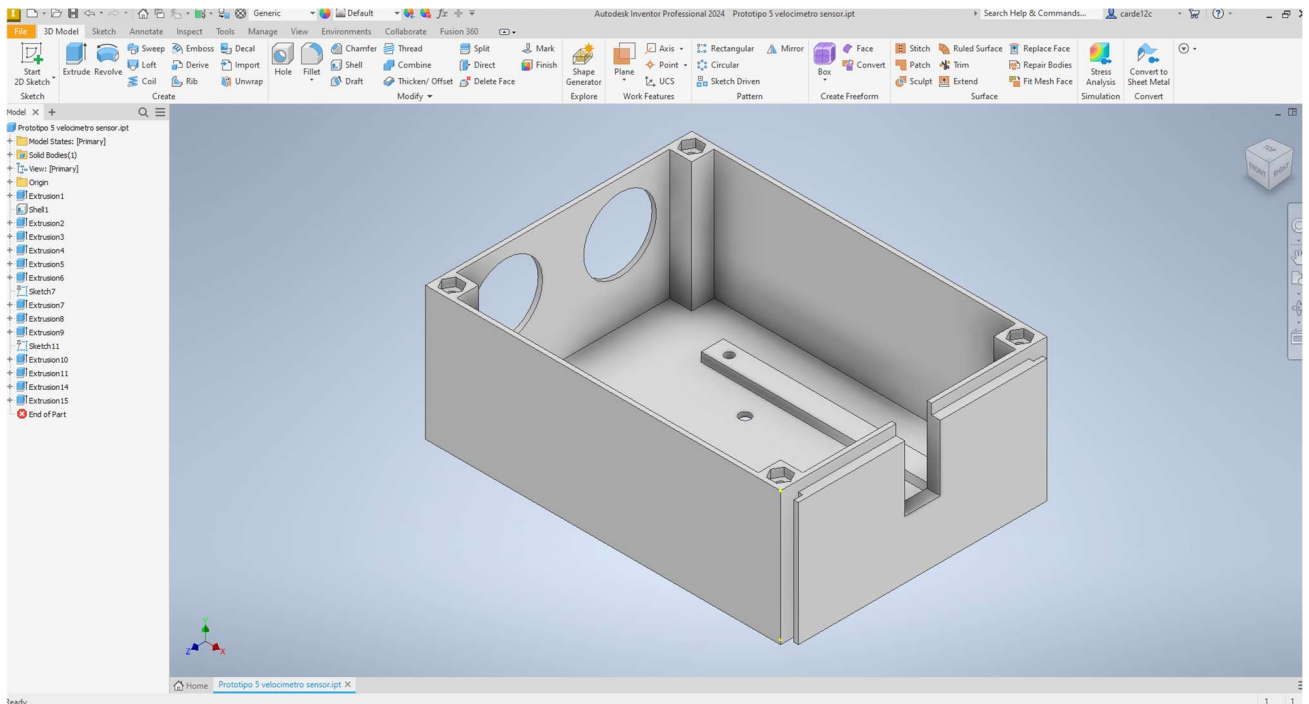


Figura 29. Captura de pantalla de la aplicación Inventor
Fuente: Elaboración propia

- **PrusaSlicer** (Figura 30) es un software de código abierto utilizado para preparar modelos 3D para impresión en impresoras 3D. Es desarrollado por Prusa Research y es especialmente conocido por su integración con las impresoras 3D Prusa, aunque también es compatible con una amplia gama de impresoras 3D.



Figura 30. Logo PrusaSlicer
Fuente: Referencia bibliográfica [31]

PrusaSlicer es una herramienta esencial para usuarios de impresoras 3D que buscan un software potente y flexible para preparar sus modelos para la impresión. Con una interfaz intuitiva y opciones avanzadas, facilita tanto el trabajo de principiantes como de usuarios avanzados en la optimización de sus impresiones 3D. (Figura 31)

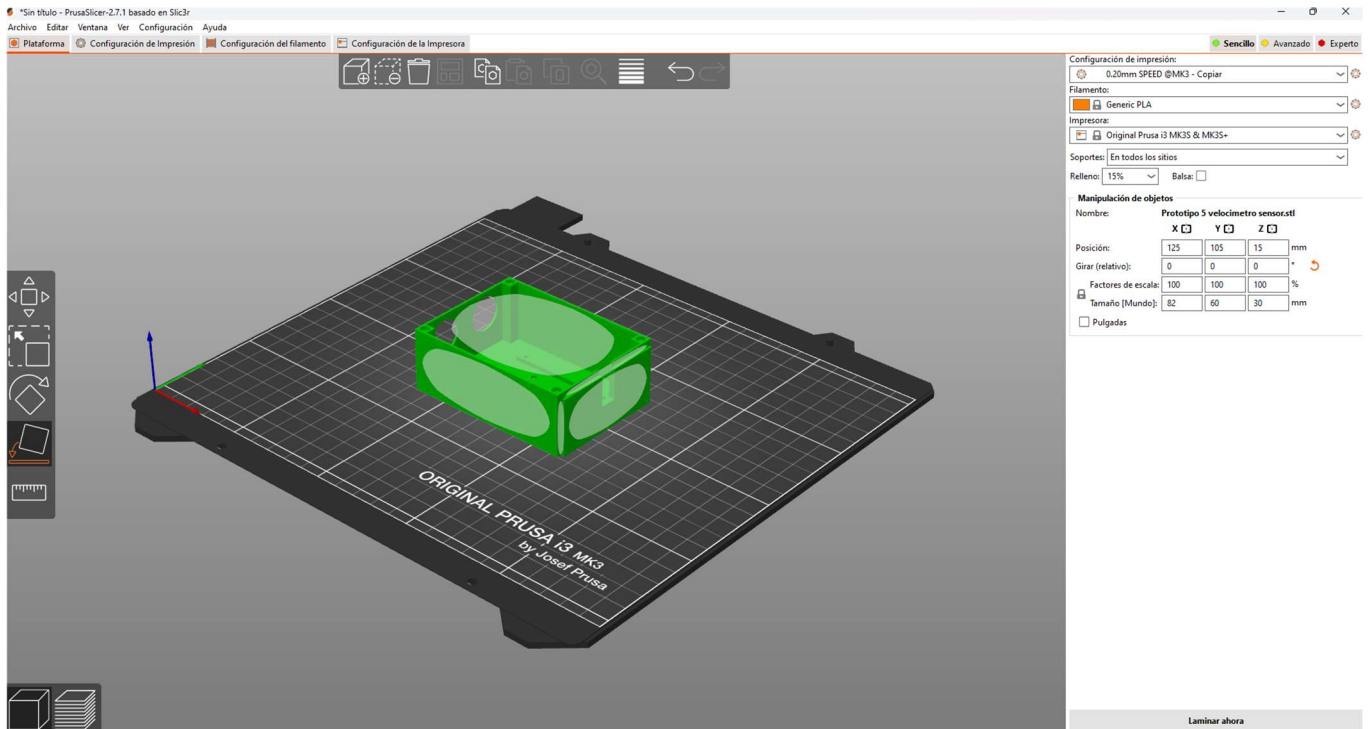


Figura 31. Captura de pantalla de la aplicación PrusaSlicer
Fuente: Elaboración propia

Para llegar hasta el diseño final de la carcasa, primero se pasó por diferentes pruebas. Una vez decidida la forma que se quería para el dispositivo, se buscó que tuviera el menor tamaño posible. Para ello, se planteó en qué partes de la carcasa irían colocados los diferentes componentes que forman el velocímetro.

Al querer darle forma de pistola, como ya se ha comentado anteriormente, la carcasa constaría de dos partes principales. Una “caja” superior donde se colocarían la mayoría de los componentes del dispositivo y un mango, donde por comodidad se decidió meter las baterías que suministran la energía al velocímetro.

Una vez decidido en qué partes de la carcasa se acomodarán los diferentes componentes del dispositivo, y teniendo las medidas de todos ellos, se plantearon diferentes alternativas de diseño, las cuales se imprimieron y se hicieron pruebas de ensamblaje.

A continuación, se muestran algunos de los diseños que se realizaron antes de obtener el que finalmente se eligió para el dispositivo. (Figuras 32 y 33)

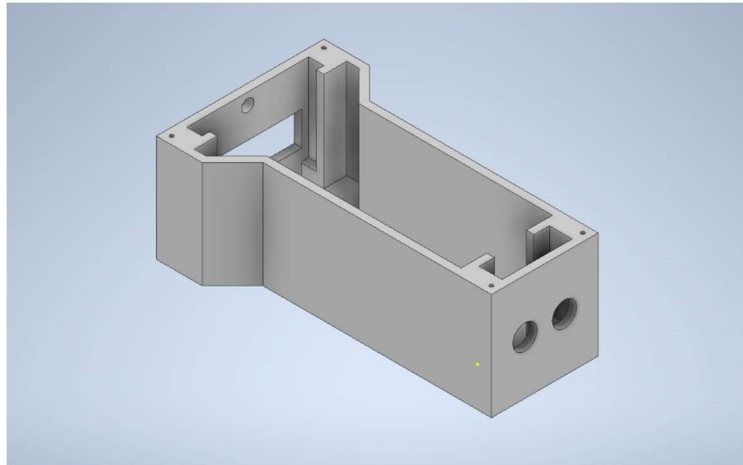


Figura 32. Prototipo 1
Fuente: Elaboración propia

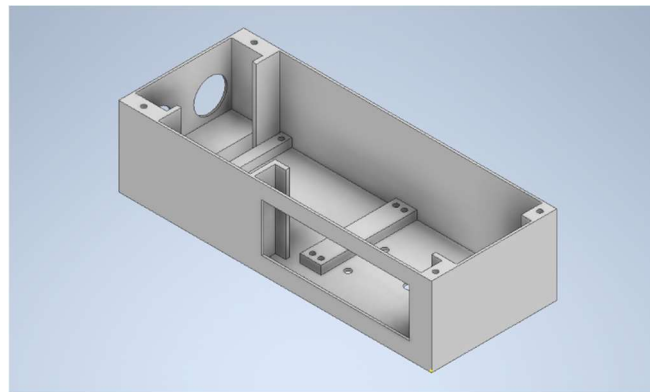


Figura 33. Prototipo 4
Fuente: Elaboración propia

5.3. Despiece

Tras las diferentes pruebas con los prototipos de carcasa, se optó por separar la parte superior en dos piezas y luego el mango. En este apartado se muestran las diferentes piezas que componen la carcasa final del dispositivo, con una breve explicación de cada una.

- Caja de la pantalla y tapa. En esta parte de la carcasa es donde irán la pantalla y el interruptor del dispositivo. Además, la tapa posee un orificio y una pestaña de sujeción donde se acomodará el led. (figuras 34 y 35)

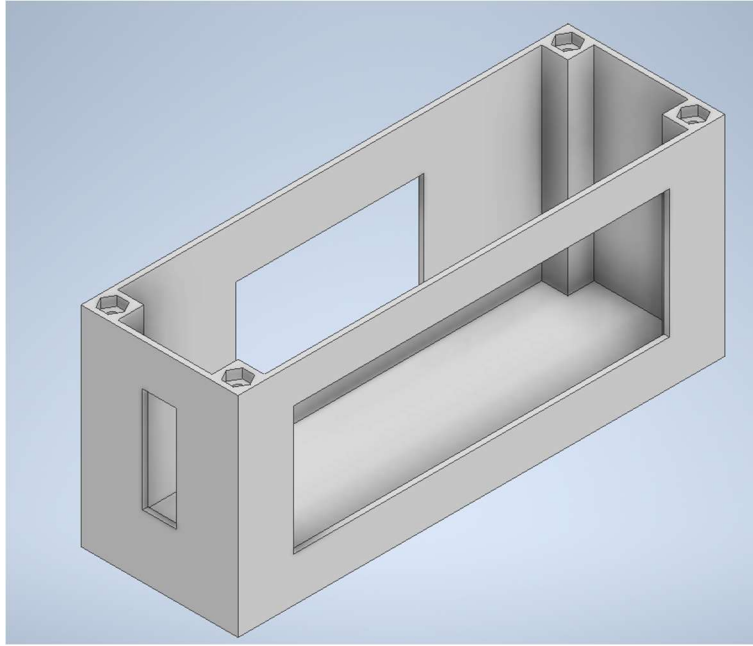


Figura 34. Pieza carcasa de la pantalla
Fuente: Elaboración propia

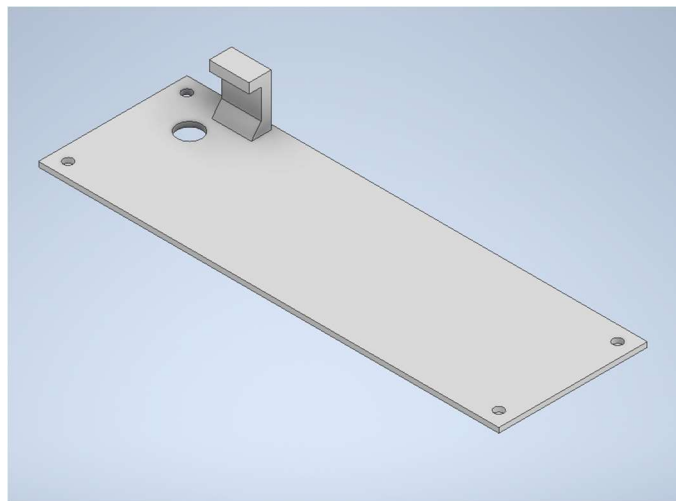
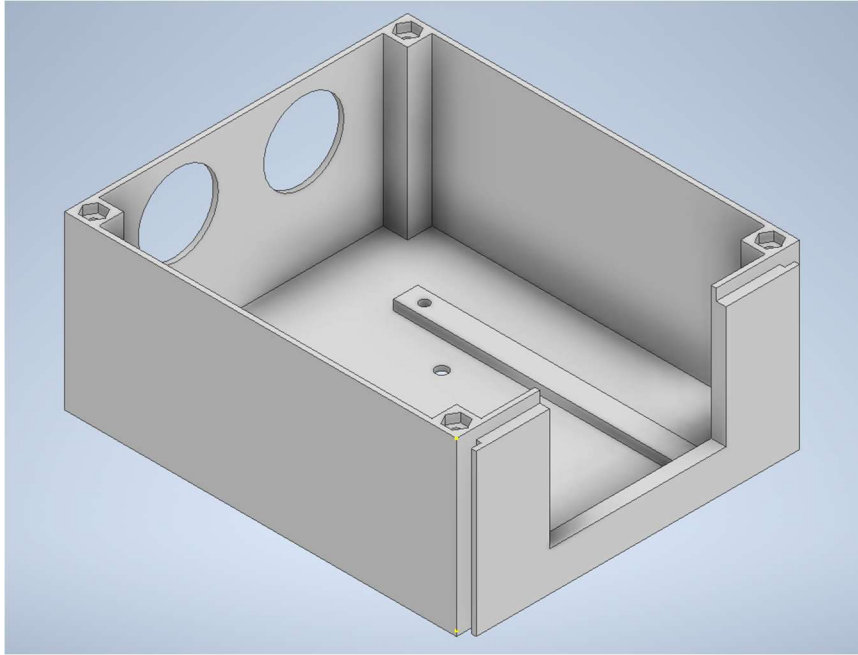
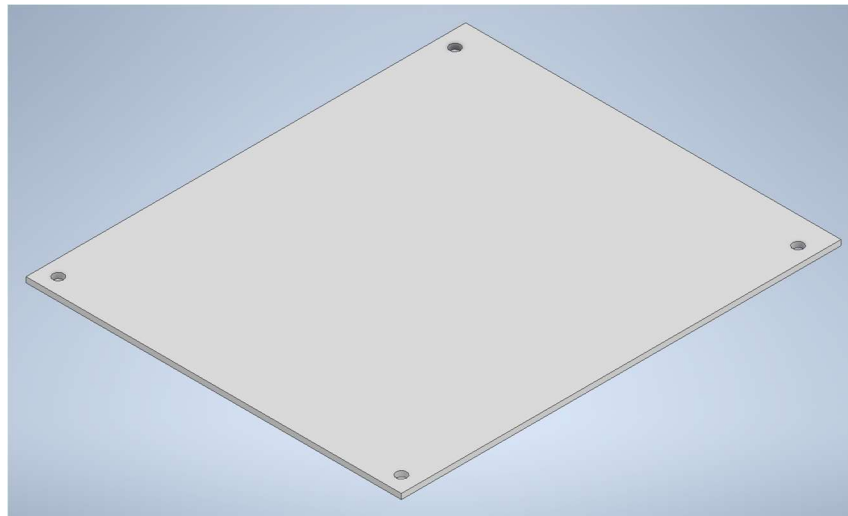


Figura 35. Tapa de la carcasa de la pantalla
Fuente: Elaboración propia

- Caja del sensor y tapa. En esta pieza de la carcasa es donde estarán la mayoría de los componentes del dispositivo, ya que es aquí donde estarán tanto el sensor como las dos placas de circuitería, lo que incluye el Arduino y las resistencias. (Figuras 36 y 37)



*Figura 36. Pieza carcasa del sensor
Fuente: Elaboración propia*



*Figura 37. Tapa de la carcasa del sensor
Fuente: Elaboración propia*

- Mango y tapa. El mango finalmente se ajustó para poder alojar en su interior las baterías del dispositivo, y que además que sobrara espacio suficiente para poder incluir el pulsador. (Figuras 38 y 39)

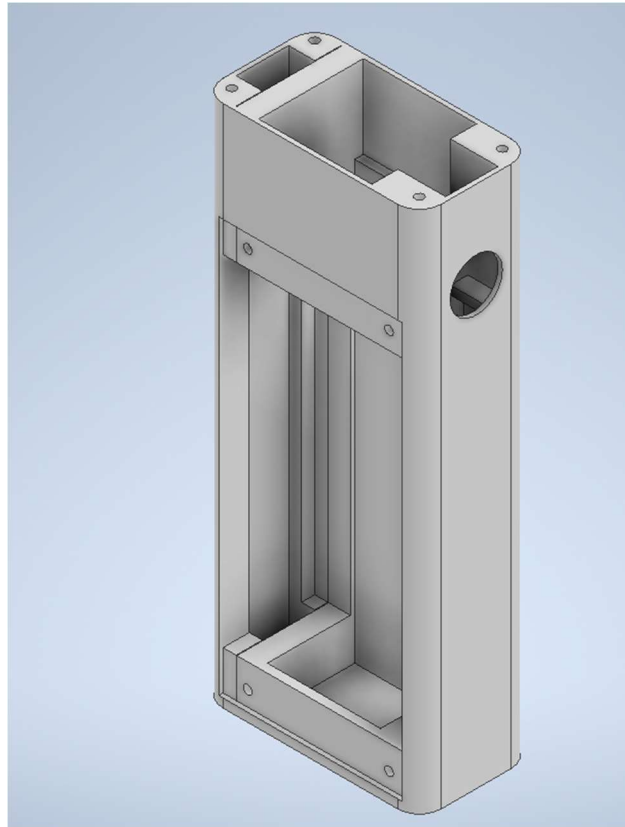


Figura 38. Mango
Fuente: Elaboración propia

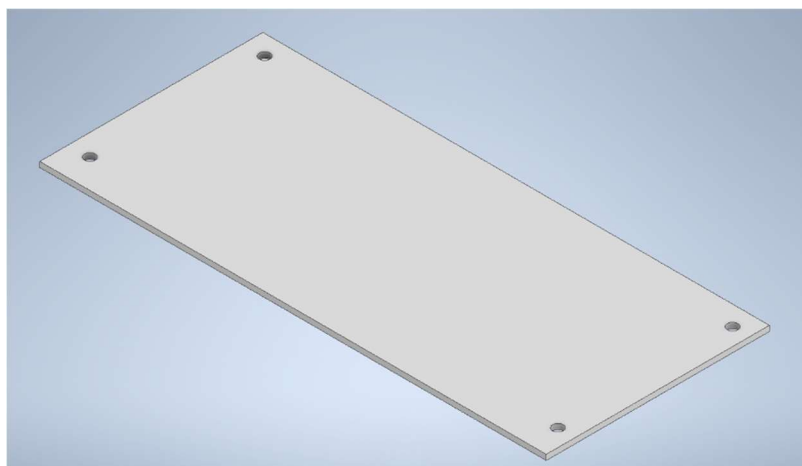


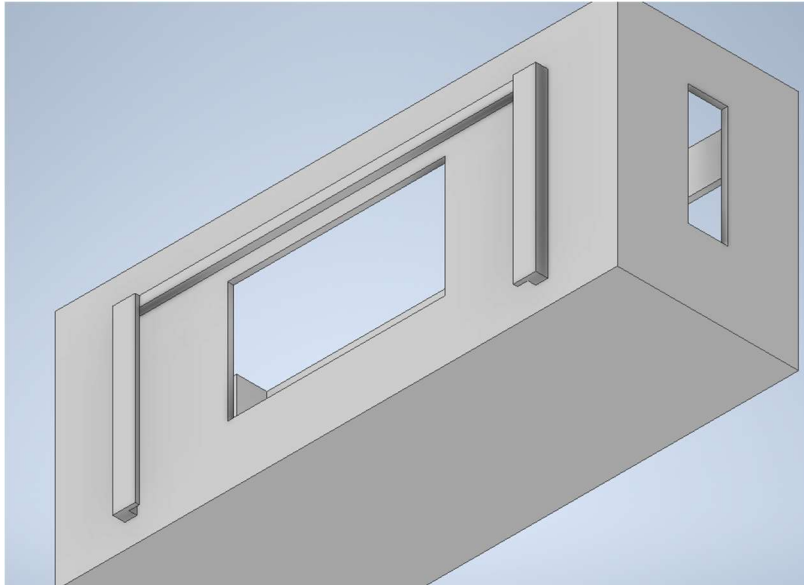
Figura 39. Tapa del mango
Fuente: Elaboración propia

Los planos de diseño de toda la carcasa aparecerán en el **"Anexo 2"** de este proyecto.

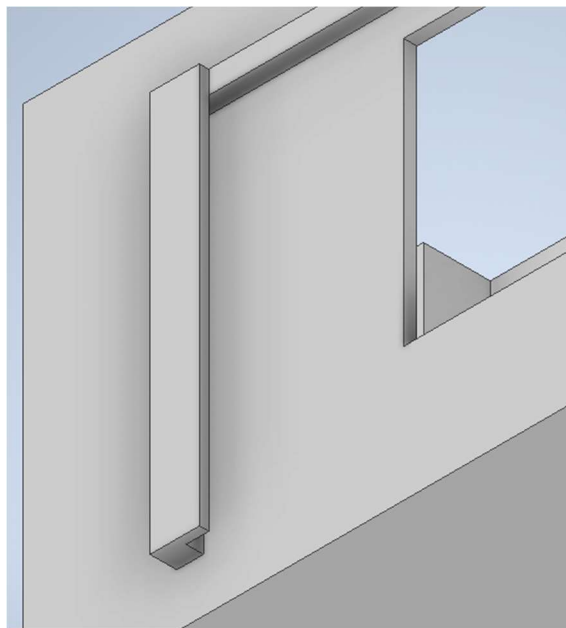
5.4. Montaje final

Tras imprimir todas las piezas, se hizo la comprobación final de montaje antes de incluir todos los componentes. Para el montaje entre las piezas se decidió usar tornillería exceptuando entre las dos piezas de la caja, que ya en el diseño se decidió incluirle unas pestañas para que las dos piezas encajaran entre sí mediante deslizamiento, sin necesidad de tornillería.

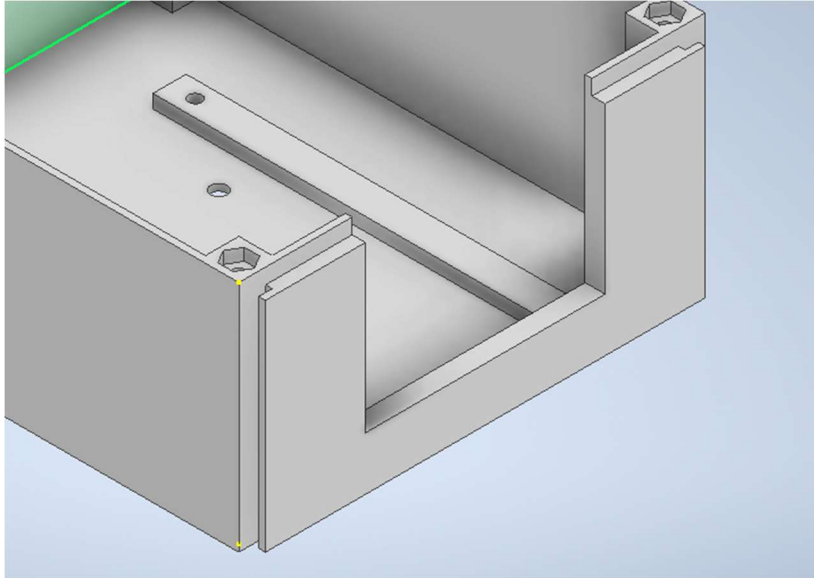
A continuación, se presentan unas imágenes del detalle de este sistema.



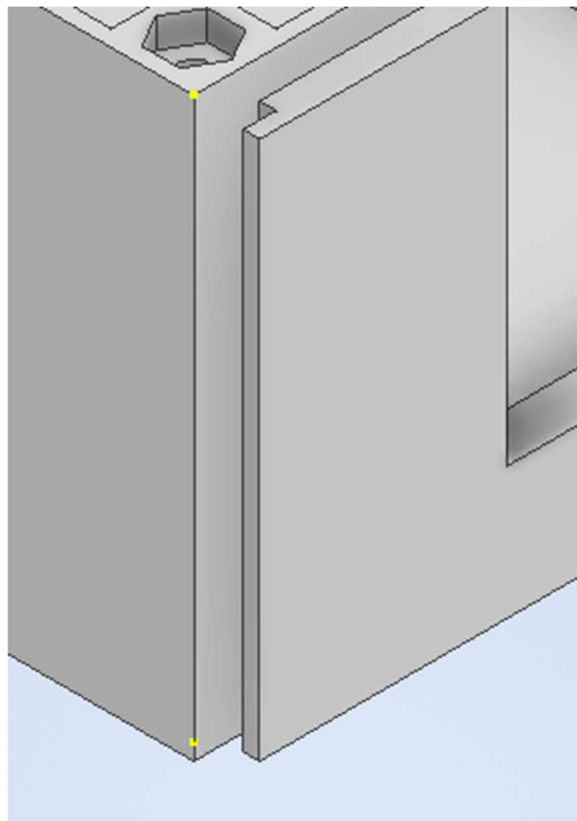
*Figura 40. Enganche del módulo de la pantalla
Fuente: Elaboración propia*



*Figura 41. Enganche del módulo de la pantalla (detalle)
Fuente: Elaboración propia*



*Figura 42. Enganche del módulo del sensor
Fuente: Elaboración propia*



*Figura 43. Enganche del módulo del sensor (detalle)
Fuente: Elaboración propia*

A continuación, se muestran algunas imágenes de cómo queda la carcasa del dispositivo una vez montada con los componentes clave en sus posiciones, pero aún sin la circuitería lista.



Figura 44. Foto 1 ensamblaje
Fuente: Elaboración propia



Figura 45. Foto 2 ensamblaje
Fuente: Elaboración propia

6. Montaje Final

En la etapa final del proyecto queda montar y ensamblar la circuitería dentro de la carcasa. Además, una vez todo montado queda comprobar que el dispositivo cumple con las funciones que se esperaban en su diseño.

6.1. Montaje

Para la circuitería se utilizó estaño y un soldador para poder crear el circuito definitivo del dispositivo, también se aprovechó que varios de los componentes poseían pines, para poder hacer más sencilla la sustitución de alguno de ellos si fuera necesarios por alguna avería o fallo.

Una vez completado el circuito, el último paso es el ensamblaje de este en la carcasa del dispositivo. Al estar diseñado desde cero, y estar hecho a medida, este paso es realmente sencillo ya que todos los componentes encajan a la perfección en sus correspondientes espacios, por lo que con ayuda de algo de tornillería todo queda bien conectado y listo para su funcionamiento. (Figura 46)

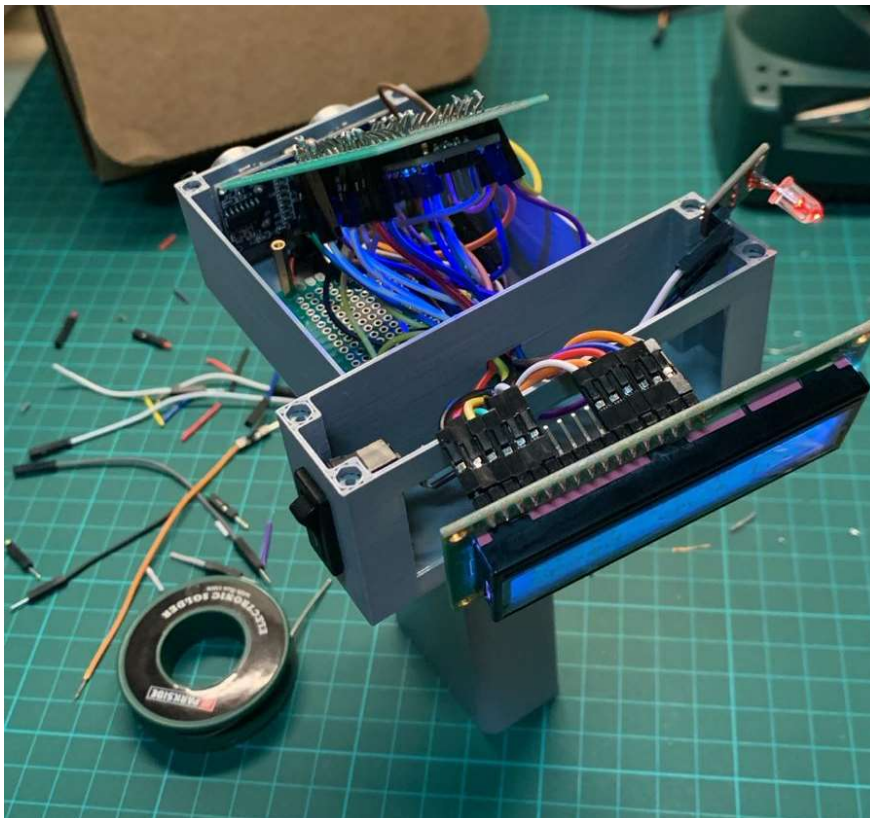


Figura 46. Foto de implementación
Fuente: Elaboración propia

6.2. Pruebas

Una vez todo está montado, queda comprobar que el dispositivo funciona tal y como está previsto, para ello se realizan una serie de pruebas que, aunque simples, muestran que el velocímetro funciona correctamente. Se han realizado una serie de pruebas para comprobar las capacidades del dispositivo:

- **Prueba de medidas correctas.** (Figura 47) se quería comprobar que la velocidad medida por el dispositivo es correcta. Para ello, se midió la velocidad de una caja la cual se estaba moviendo a velocidad constante a lo largo de una mesa. Además, se cronometró el tiempo que tardó la caja en llegar de un lado a otro de la mesa. Conociendo las medidas de la mesa, se calculó la velocidad y se compara con la de la lectura del dispositivo.



Figura 47. Prueba 1 (medidas correctas)
Fuente: Elaboración propia

- **Prueba de distancia.** (Figura 48) se midió la velocidad de otra caja situada sobre una cinta que se desplazaba a una velocidad conocida. Esta prueba permitió comprobar también su funcionamiento a mayores distancias.



Figura 48. Prueba 2 (distancia de medida)
Fuente: Elaboración propia

- **Prueba de medida de limitaciones.** (Figura 49) Una vez comprobado que el dispositivo obtiene medidas correctas de velocidades y encontrado cual es la distancias a las que el sensor es capaz de obtener lecturas de distancia, se quería buscar cuales son las limitaciones del dispositivo que se ha desarrollado. Para ello se hizo varias pruebas midiendo la velocidad de un coche, partiendo del coche parado, se midió la velocidad máxima a la que llegaba este.



Figura 49. Prueba de limitaciones
Fuente: Elaboración propia

Tras esta serie de pruebas se ha podido determinar las capacidades del dispositivo. Este puede medir un amplio rango de velocidades, ya que, aunque el dispositivo muestre las velocidades en cm/s en las pruebas del coche se obtuvieron lecturas de hasta 2500cm/s equivalente a 90km/h. No obstante, el problema que se ha detectado es que, por la capacidad de medida del sensor de hasta 4m, y el funcionamiento físico del sensor ultrasónico, el dispositivo da errores de medida de la velocidad según se va ampliando la distancia.

A continuación, se presentan los enlaces donde se pueden encontrar los videos de las pruebas descritas anteriormente:

- Prueba 1. <https://media.upv.es/player/?id=7b3241d0-1c74-11ef-a087-3524a69ceafe>
- Prueba 2. <https://media.upv.es/player/?id=0a44bb90-1c76-11ef-958f-3b1b47278153>
- Prueba 3. <https://media.upv.es/player/?id=139a5ed0-1c75-11ef-a520-a111d113fa5e>

7. Conclusiones y trabajo futuro

7.1. Conclusión

La realización de este proyecto me ha llevado a adentrarme algo más en campos que había visto a lo largo del grado y a conocer otros nuevos.

Por una parte, el haberme adentrado con este proyecto en el diseño digital para crear la carcasa me ha permitido ver ese uso práctico que solo había podido estudiar de manera teórica durante el grado, ya que al darle forma gracias a la impresión 3D, comprendes mejor la inmensidad de usos y la versatilidad que tiene la herramienta del diseño digital.

Por otra parte, el tema del montaje del circuito, empezando con las placas de pruebas y más tarde con la soldadura, permite ver la infinidad de dispositivos que se puede llegar a diseñar gracias a las tecnologías actuales. Añadiendo a esto el campo del Arduino, abre un sinfín de posibilidades de proyectos a crear gracias al avance tecnológico que este entorno supone.

La finalidad de este proyecto en su mayoría era demostrar como gracias al avance tecnológico y a implementación de unos conocimientos en los diferentes campos nombrados, hoy en día se puede llegar a diseñar una serie de dispositivos funcionales de una manera muy sencilla y más barata de lo que salen los dispositivos de las mismas características en el mercado.

Por último, es importante añadir que la versatilidad del entorno Arduino permite ajustar las características del dispositivo, lo cual lleva a que el dispositivo diseñado puede tener más funciones que otros de las mismas características en el mercado.

7.2. Trabajos futuros

Para optimizar el dispositivo de este proyecto y asegurar su máxima utilidad, es fundamental, antes de su desarrollo, tener claramente definida la tarea que se espera que realice. Conociendo esta tarea de antemano, se puede programar y/o diseñar el dispositivo de manera más personalizada y eficiente para cumplir dicha función.

Por otro lado, invertiría algo más de presupuesto en el sensor, ya que aun estando fuera de los sensores industriales, existen algunos sensores algo más potentes, para así obtener un dispositivo de mayores capacidades por si fuera necesario. También buscaría un sensor de otro tipo, como podría ser el láser si la intención del proyecto es la medida de velocidades.

Por último, se me ocurre que para mayor facilidad de manejo del dispositivo final y si por si fuera necesario un cambio de código de este, se podría ajustar la carcasa del dispositivo para que se pudiera conectar el USB a la placa de Arduino sin tener que abrir o desmontar el dispositivo una vez montado. Esto permitiría un fácil reajuste del código, para variar los rangos de medida del dispositivo en caso de tener que cubrir diferentes necesidades.

8. Bibliografía

A continuación, se presentan las diferentes fuentes que se han utilizado para respaldar información o búsqueda de imágenes, para completar la elaboración de esta memoria:

1. ¿Cómo funciona el sensor de ultrasonidos medidor de distancia? (2021, 10 diciembre). <https://solectroshop.com/es/blog/como-funciona-el-sensor-de-ultrasonidos-medidor-de-distancia--n99>

2. Funcionamiento y tecnología de un sensor de ultrasonidos. (s. f.). Wenglor Sensoric Group. <https://www.wenglor.com/es/Principio-de-funcionamiento-y-tecnologia-de-un-sensor-de-ultrasonidos/s/Funktionsprinzip+und+Technologie+eines+Ultraschall-Sensors?BranchenundIndustrienlinks=%3A1&BranchenundIndustrienrechts=%3A1>
3. Mecafenix, I. (2023, 28 noviembre). Que es un sensor ultrasónico y cómo funciona. Ingeniería Mecafenix. <https://www.ingmecafenix.com/automatizacion/sensores/ultrasonico/>
4. Comprender el funcionamiento de los sensores ultrasónicos. Jeff Smoot. (20/05/2021) <https://www.digikey.es/es/articles/understanding-ultrasonic-sensors>
5. Fundación Aquae. (2020, 30 octubre). ¿Sabes qué es un Arduino y para qué sirve? <https://www.fundacionaquae.org/wiki/sabes-arduino-sirve/>
6. Fernández, Y. (2022, 23 septiembre). Qué es Arduino, cómo funciona y qué puedes hacer con uno. Xataka. <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
7. Aguayo, P. (2023, 28 diciembre). Inicio | Arduino.cl - Plataforma de Código Abierto para Electrónica. Arduino.cl - Compra Tu Arduino En Línea. <https://arduino.cl/>
8. ELOVIS. (2024, 4 febrero). Medición sin contacto de longitud, velocidad, longitud de pieza - % Berührungslose Messung von Länge, Geschwindigkeit, Stücklänge. <https://www.elovis.com/es/>
9. Gandhi, M., Gandhi, M., & Gandhi, M. (2020, 5 febrero). Tipos de sensores de velocidad para bandas transportadoras. AUTYCOM - AUTYCOM. <https://www.autycom.com/tipos-de-sensores-de-velocidad-bandas-transportadoras/>
10. Buy Velocity Speed Gun | Amazon. (s. f.). Amazon. <https://www.amazon.es/Bushnell-Velocity-Speed-Point-101911/dp/B0002X7V1Q?th=1>
11. Systèmes, D. (2022, 5 julio). Impresión 3D. Dassault Systèmes. <https://www.3ds.com/es/make/guide/process/3d-printing>
12. Mian. (2016, June 3). *Omega Engineering*. Omega. <https://www.omega.com/en-us/test-inspection/handheld-meters/speed/p/HHT13-Series>
13. Illinois, J. H. (2024, March 25). Omega Engineering. <https://www.omega.com/en-us/test-inspection/handheld-meters/speed/p/HHT12-Series>
14. Omega Engineering. (s.f.). <https://www.omega.com/en-us/test-inspection/handheld-meters/speed/p/HHT41B-series>
15. Ind, D. K. M. (2023, August 15). Omega Engineering. <https://www.omega.com/en-us/test-inspection/handheld-meters/speed/p/HHT30R>
16. *Tacometros* - Grainger Industrial Supply. (s.f.). <https://www.grainger.com/category/test-instruments/non-electrical-properties-testing/tachometers>
17. *Medidor láser de distancias técnica de medición*. Técnica de medición. (s.f.). <https://www.bosch-professional.com/es/es/medidor-laser-de-distancias-101300-ocs-c/>
18. *GLM 50-27 C Medidor Láser de Distancias: Bosch Professional*. GLM 50-27 C | Bosch Professional. (s.f.). <https://www.bosch-professional.com/es/es/products/glm-50-27-c-0601072T00>
19. *Medidor Láser de Distancia: Medidor Láser de Distancia Fluke 417D: 40 m*. Fluke. (s.f.). <https://www.fluke.com/es-mx/producto/infraestructura-del-edificio/medidores-laser-de-distancia/417d>
20. *Buy velocity speed gun and more*. Bushnell. (s.f.). <https://www.bushnell.com/additional-products/speed-guns/velocity-speed-gun/BU-101911.html>
21. *Sensores de proximidad ultrasónicos*. Amazon.es: Sensores de proximidad ultrasónicos: Industria, empresas y ciencia. (s.f.). <https://www.amazon.es/Sensores-ProximidadUltrasonicos/b?ie=UTF8&node=10229657031>

22. Nimomo US42 I2C Pixhawk APM Módulo de Medición de Distancia Ultrasónico Con Sensor De Control de Vuelo. Amazon.es: Industria, empresas y ciencia. (s.f.). https://www.amazon.es/Pixhawk-medici%C3%B3n-distancia-ultras%C3%B3nico-control/dp/B09QHNQNJ5/ref=sr_1_21?mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=U9EPYR2U43BV&dib=eyJ2IjoiMSJ9.Gi1Y5lYk0lr72ac5dQIQDVFu5gbbNRcmTnnOdzdNQu8ilY_2lzkjLTE8RL_2RQypW52Sl2vQLi2Z5ySDyh3LpFkxTqSqHRuUz13K3ERNIOKAfrue0pOk70DvZhB4bxWKccqmGA323aZzLbbgnNIjDDVD7VoHVntDQZbrbmC3j6kslBYID6Ft8aHYI4T39E334Kbgs8_IBvG2n47YnWEbpywglS36IEFlfdil2gyH1W7dlJmmPDpKYL03GREFQdjL5oA_FkQmp0ACldicnjlY5QDS7c5FV319gFpfoY.Aq_422UbHXxQXR4DEaW2QBdl2vpFGUg2dxlyE9ODz_U&dib_tag=se&keywords=sensor%2Bultrasonido%2Barduino%2Bpixhawk&qid=1716143289&srefix=sensor%2Bultrasonido%2Barduino%2Bpixhawk%2Caps%2C104&sr=8-21
23. Ociodual Nano v3.0 Placa de control ATmega328P ... (s.f.). <https://www.pccomponentes.com/ociodual-nano-v30-placa-de-control-atmega328p-ch340g-compatible-con-arduino>
24. Azdelivery HD44780 1602 modulo pantalla LCD display 2x16 caracteres con Fondo Azul y Caracteres Blancos compatible con Arduino y Raspberry Pi. Amazon.es: Informática. (s.f.). https://www.amazon.es/AZDelivery-HD44780-Pantallacaracteresincluido/dp/B08216WMDJ/ref=sr_1_14?mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=35SW471VBGLCY&dib=eyJ2IjoiMSJ9.eQ3M3OcA2p8xfwvqFPtOkNbugAXctd3BL8mtzO8_LfDgIhckX_aD4bVDVs4Tkjsdo1YkGotvCeoXCna5OnEv0drLU3pZF182WR_ojb12eCMPUIQVXwUve0wunl0fpWVzQ0PzhHKxHuJVwGPfyE4aORMsY9We_1lel15Tndh0t1mtq3GXwtvC421z5BOnnhQsnD6s6u9XjSmJBrcFJdkYuhaBKVuGnXhdRu_4J5xYlvz7_Yyoz4ssjjP0kpWZMueQFpooSzwJLLZ9PzSTeVtJFdPjHje2ZsPbyD_1rjbX4.myS6g3p8FmdXWT2MoS46bwKohyaDtwUlmoQDhbWPwc&dib_tag=se&keywords=pantalla%2Blcd&qid=1716143063&srefix=pantalla%2Blcd%2Caps%2C121&sr=8-14&th=1
25. Azdelivery sensor compatible con HC-SR04 de Distancia modulo Ultrasonico compatible con Arduino y Raspberry Pi. Amazon.es: Industria, empresas y ciencia. (s.f.). <https://www.amazon.es/AZDelivery-HCSR04-Modulo Parent /dp/B07Z4RTTV6>
26. Azdelivery KY-016 FZ0455 3-color RGB Módulo led 3 color compatible con arduino. Amazon.es: Iluminación. (s.f.). <https://www.amazon.es/AZDelivery-KY-016-Parent/dp/B082FM9C8Q>
27. 3362p-1-103LF - trimpot, Vuelta única, Cermet, Ajuste Superior, 10 Kohm, Agujero Pasante, 1 Vueltas. Farnell. (s.f.). <https://es.farnell.com/bourns/3362p-1-103lf/pot-trimmer-10k-3-pin/dp/9354301>
28. Aerezetix - c10596 - conmutador/Interruptor de Botón SPST-no 1A/250VAC pulsador momentáneo redondo - off-(on) - 1 Posición Estable - Color Negro/amarillo - en Plástico. Amazon.es: Coche y moto. (s.f.). <https://www.amazon.es/AERZETIX-Conmutador-interruptor-moment%C3%A1neo-OFF/dp/B09ZH7PBRX>
29. Cable pelado Interruptor Basculante, interruptor 2 posiciones, Interruptor 2 pines, interruptor 2 terminales, interruptor on/off SPST, 16 a 250 VAC: 10a 125VAC, negro sin luz. Amazon.es (s.f.). <https://www.amazon.es/CABLEPELADO-Interruptor-universal-pin-off/dp/B0B1MVN8PV>
30. Descargar Inventor | Prueba gratuita de Inventor | Autodesk. (s. f.). <https://www.autodesk.es/products/inventor/free-trial>
31. PrusaSlicer | Impresoras 3D Original Prusa vendidas directamente por Josef Prusa. (s. f.). Prusa3D By Josef Prusa. https://www.prusa3d.com/es/pagina/prusaslicer_424/

También se sacó información de **Barnatt, C. (2016). 3D Printing (3rd ed.)**, para obtener información acerca de la impresión 3D.

9. Cálculo de presupuesto

En este anexo se va a llevar a presentar el cálculo del presupuesto que supone este proyecto.

- **Coste de mano de obra**

Para el desarrollo de este proyecto se ha contado con un estudiante de ingeniería y la supervisión de dos ingenieros senior. De manera informativa se plantea el coste de la mano de obra, aunque este no sea un coste real para el presupuesto del proyecto que se ha llevado a cabo.

Descripción	Unidades	Coste (€/h)	Coste €
Estudiante de Ingeniería	300h	25,00	7500,00
Ingeniero sénior	20h	60,00	1200,00

Tabla 1. Presupuesto mano de obra

- **Listado de materiales y coste**

Para el montaje del dispositivo han sido necesarios una serie de componentes, tanto para su montaje final como para la fase de elaboración y pruebas del prototipo. En la siguiente tabla se presentan las diferentes piezas y componentes junto con su coste.

Descripción	Unidades	Coste (€/ud)
Arduino Nano R3	ud	8,99
Adafruit Standard LCD 16x2	ud	6,50
Sensor ultrasónico HC-SR04	ud	8,99
LED	ud	1,67
Resistencia 330 ohm	ud	0,10
Potenciómetro (Single Turn) 10 kilo ohmios	ud	9,50
Interruptor	ud	4,50
Accionador en forma de pulsador	ud	4,80
Batería recargable 18650 de Ión-Litio, 3.7V, 2.6Ah	ud	8,36
Porta pilas	ud	1,99
Sistema de alimentación ininterrumpida UPS HAT	ud	25,95

Placa de prototipado	ud	2,41
Set de elementos de conexión: 120 Piezas de Cable DuPont, 40 Pines Macho-Hembra, 40 Pines Macho-Macho, 40 Pines Hembra-Hembra	ud	6,99
Pilar hexagonal espaciador hexagonal macho-hembra	ud	0,10
Tornillo	ud	0,05
Tuerca	ud	0,05
Filamento para impresora 3D FDM, PLA, 1.75mm, metálico, 1kg RS PRO	kg	40,57
Estaño para soldar	kg	60,61

Tabla 2. Listado y coste de componentes

- **Presupuesto de impresión y diseño**

El dispositivo se ha diseñado a partir del uso de varias aplicaciones. Sin embargo, no se valorará el coste de estas ya que se una se obtuvo por un convenio universitario con el desarrollador y otra es de uso gratuito. Por otro lado, en el coste de la impresión, hay que tener en cuenta el gasto energético y el uso del filamento que se utilizó en la impresión de la carcasa del dispositivo.

La siguiente tabla muestra, los tiempos y gasto de material de las piezas de la carcasa del dispositivo.

Pieza	Tiempo de impresión	Gasto de material
Modulo del sensor	2h 22min	22,53g
Módulo de la pantalla	2h 9min	18,49g
Mango	3h 19min	38g
Tapas	1h 14min	14,59g

Tabla 3. Tiempo y coste de impresión de piezas

En la siguiente tabla se muestra el presupuesto de la impresión y el diseño del dispositivo, contando con el coste de maquinaria y coste tanto energético como el de gasto de material en la impresión. El coste de las licencias de los softwares utilizados no se tendrá en cuenta, ya que se han utilizado versiones gratuitas para estudiante, concedidas por convenios de la universidad (como la licencia de Autodesk) o eran de carácter gratuito (como es el caso de PrusaSlicer) por lo que no han supuesto un coste para la realización del proyecto. Añadir, que el coste energético que figura en la siguiente tabla se debe a que la impresora gasta unos 80w cada hora de impresión y el coste eléctrico es de 0,20(€/kWh)

Descripción	Unidades	Coste unidad	Coste
Filamento para impresora 3D FDM, PLA, 1.75mm, metálico, 1kg RS PRO	0,0936 kg	40,57 (€/kg)	3,80 €
Coste energético de la impresora	9h	0,20 (€/kWh)	0,17€
Software Inventor	1 ud	0,00€	0,00€
Software PrusaSlicer	1 ud	0,00€	0,00€

Tabla 4. Coste de impresión y diseño

Coste total de la impresión y diseño: 3,97 €

Cabe añadir que en este coste no se está teniendo en cuenta el gasto que supusieron las pruebas de impresión anteriores al diseño final. De tener estas en cuenta el coste ascendería a unos **38,49 €**.

- **Presupuesto del montaje**

El montaje se ha llevado a cabo mediante el uso de herramientas que han sido facilitadas por el **Departamento de Informática de Sistemas y Computadores (DISCA)** de la Universidad Politécnica de Valencia, al que pertenecen los tutores de este proyecto, por lo que su coste no se tendrá en cuenta para este presupuesto. A continuación, se presenta una tabla con el coste que ha tenido el montaje del dispositivo del proyecto.

Descripción	Unidades	Coste (€)
Arduino Nano R3	1	8,99
Adafruit Standard LCD 16x2	1	6,50
Sensor ultrasónico HC-SR04	1	8,99
LED	1	1,67
Resistencia 330 ohm	3	0,30
Potenciómetro (Single Turn) 10 kilo ohmios	1	9,50
Interruptor	1	4,50
Accionador en forma de pulsador	1	4,80
Batería recargable 18650 de Ión-Litio, 3.7V, 2.6Ah	2	16,72
Porta pilas	1	1,99
Sistema de alimentación ininterrumpida UPS HAT	1	25,95
Placa de prototipado	6	14,46
Set de elementos de conexión: 120 Piezas de Cable DuPont, 40 Pines Macho-Hembra, 40 Pines Macho-Macho, 40 Pines Hembra-Hembra	1	6,99
Pilar hexagonal espaciador hexagonal macho-hembra M2 20mm + 3mm	4	0,40
Tornillo M2	20	1,00
Tuerca M2	8	0,40
Estaño para soldar	0,02	1,21

Tabla 5. Coste de materiales

Coste total de montaje del dispositivo: 114,37 €

Coste final del dispositivo: 120,94 €

- **Presupuesto de equipo**

El coste amortización del equipo utilizado tanto en el montaje, como en el diseño y en la impresión de las piezas del proyecto, además del precio de algunas herramientas utilizadas durante la etapa de montaje del proyecto. Se tendrá en cuenta la amortización de los equipos según su uso durante el tiempo del proyecto.

Descripción	Precio	Amortización	Coste
Impresora 3D Modelo Prusa i3 MK3S+	999 €	14%	139,86 €
Portátil OMEN 16	1600 €	7%	112 €
Soldador de estaño	42,30 €	-	42,30 €

Tabla 6. Presupuesto de equipo

Coste total del presupuesto de equipos: 294,16 €

- **Presupuesto total del proyecto**

El coste total del dispositivo final es de un total de **152,86 euros**, sumando el coste de diseño e impresión más el coste de los materiales que han sido necesarios para el mismo.

Por otro lado, a continuación, se presenta en detalle el coste total que ha tenido el proyecto.

Concepto	Importe
Mano de obra	8.700,00 €
Impresión 3D	38,49 €
Materiales	114,37 €
Software	0,00 €
Equipo	294,16 €
Presupuesto de Ejecución Material	9.147,02 €
13% Gastos Generales	1.189,11 €
6% Beneficio Industrial	548,82 €
Presupuesto de Ejecución por Contrata	10.884,95 €
I.V.A. 21%	2.285,84 €
Presupuesto Base de Licitación	13.170,79 €

Tabla 7. Presupuesto general

Anexo 1

Programación. Código completo

```
#include <LiquidCrystal.h>

#include <TimerOne.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // RS,E,D4,D5,D6,D7

// Definir número de los pines

const int trigPin = 9;

const int echoPin = 10;

int ledRojo = A0;

int ledVerde = A1;

int ledAzul = A2;

// Las contantes no varían, se usan para establecer número de pines:

const int buttonPin = 13; // El número del pin del botón

int buttonState = 0; // variable para leer el estado del botón

// definición de variables

long duration;

volatile int distanceanterior=0;

volatile int distanceactual=0;

volatile double Speed=0;

double SpeedMax=0;

int distance=0;

double LimitVelocity=10;

float ultrasonicRead ()

{

// Limpia el trigpin
```

```

digitalWrite(trigPin, LOW);
delayMicroseconds(2);

// Establece el trigPin en HIGH durante 10 microsegundos
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Lee el echoPin, devuelve el tiempo de la onda sónica en microsegundos
duration = pulseIn(echoPin, HIGH);

// Calculo de distancia
distance= duration*0.034/2;

// Guarda el valor de la distancia y los muestra en pantalla
Serial.print("Distance in cm : ");
Serial.println(distance);
lcd.setCursor(0,0);
lcd.print("Dist. in cm ");
lcd.print(distance);
lcd.print(" ");
return distance;

}

void setup() {
  lcd.begin(16, 2);// LCD 16X2
  pinMode(trigPin, OUTPUT); // Establece el trigPin como Output
  pinMode(echoPin, INPUT); // Establece el echoPin como Input
  pinMode( 7 , OUTPUT);

```

```

Timer1.initialize(1000000);    // Dispara cada 250 ms

    Timer1.attachInterrupt(Calculo); // Activa la interrupcion y la asocia a ISR_Blink
pinMode(buttonPin, INPUT);

Serial.begin(9600); // Inicio de la comunicación Serial

    // inicializa el pushbottom como Input
pinMode(ledRojo,OUTPUT);
pinMode(ledVerde,OUTPUT);
pinMode(ledAzul,OUTPUT);
}

void loop() {

    // Lee el valor de estado del pushbotton
    buttonState = digitalRead(buttonPin);

    // Comprueba si el pushbottom esta pulsado, si es así lo establece como HIGH:
    if (buttonState == HIGH) {
        SpeedMax = 0;
        // Muestra el valor de la velocidad
        Serial.print("Speed in cm/s :");
        Serial.println("0");
        lcd.setCursor(0,1);
        lcd.print("Speed cm/s ");
        lcd.print(SpeedMax);
        Serial.println("BOTON PULSADO");
    } else {

        if(Speed > SpeedMax){

            Serial.print("Speed in cm/s :");

```

```

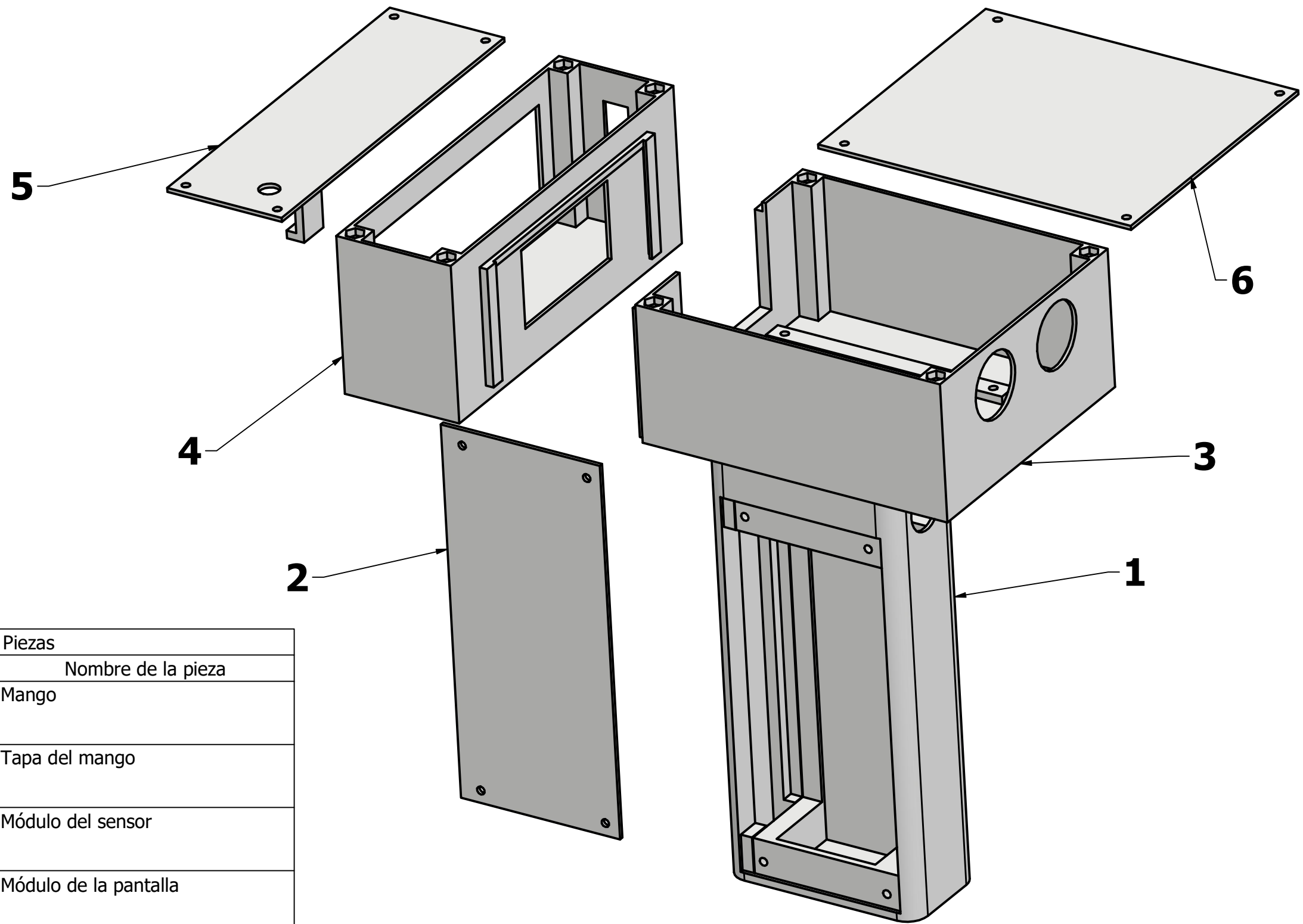
Serial.println(Speed);
lcd.setCursor(0,1);
lcd.print("Speed cm/s ");
lcd.print(Speed);
SpeedMax = Speed;

    if(SpeedMax > LimitVelocity) {
digitalWrite(ledRojo,0);
digitalWrite(ledVerde,0);
digitalWrite(ledAzul,255);
    } else {
        digitalWrite(ledRojo,0);
        digitalWrite(ledVerde,255);
        digitalWrite(ledAzul,0);
    }
}
}
}
}

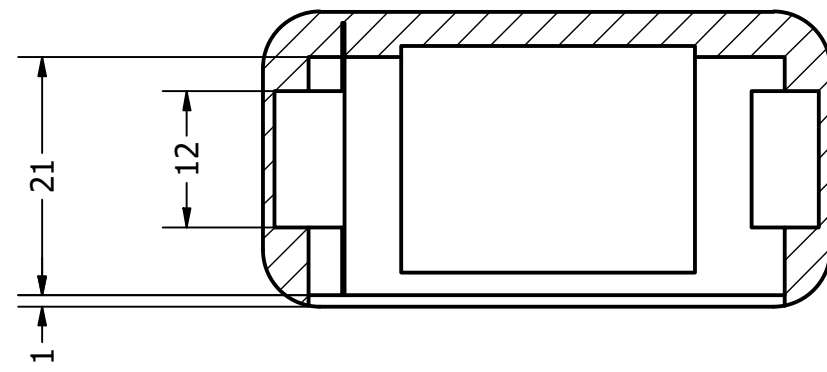
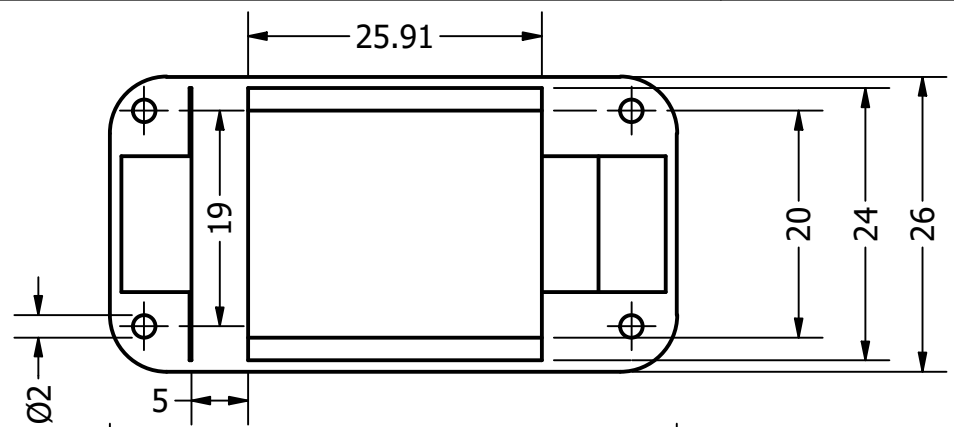
void Calculo () {
    distanceactual = ultrasonicRead(); //llama a la función ultrasonicRead()
    // Formula es la variación de distancia entre el tiempo
    Speed = abs(distanceactual - distanceanterior)/1.0; // Como el tiempo de lectura es cada segundo
    dividimos entre 1.
    distanceanterior = distanceactual;
}

```

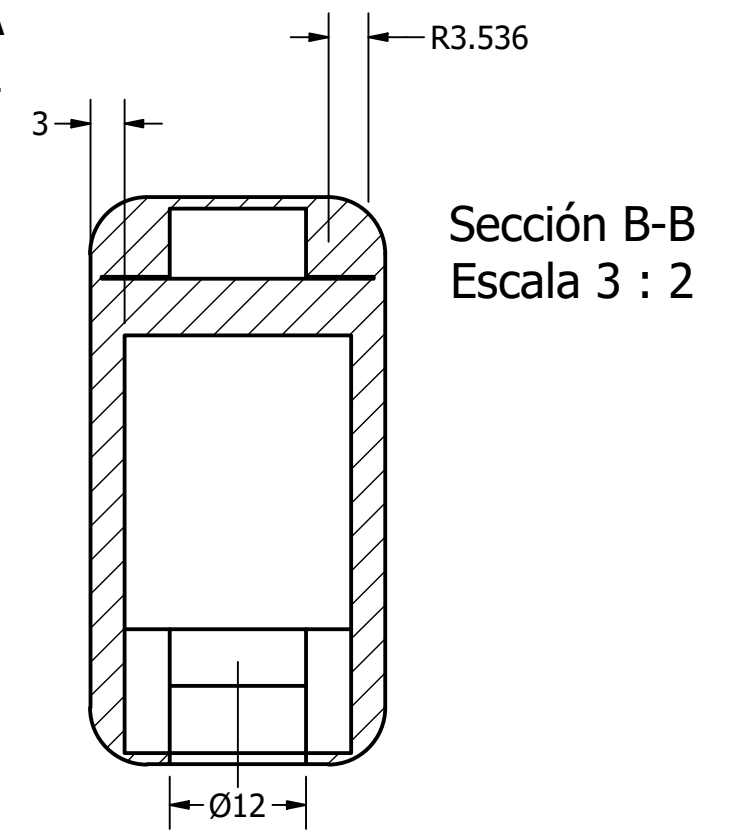
Anexo 2



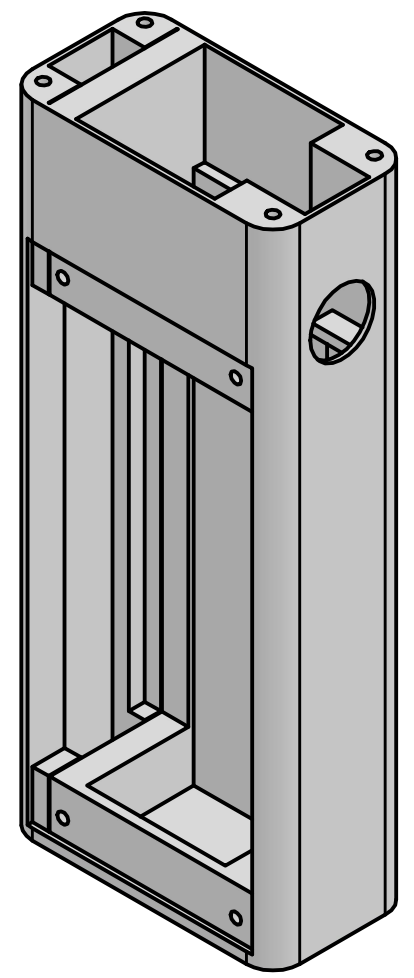
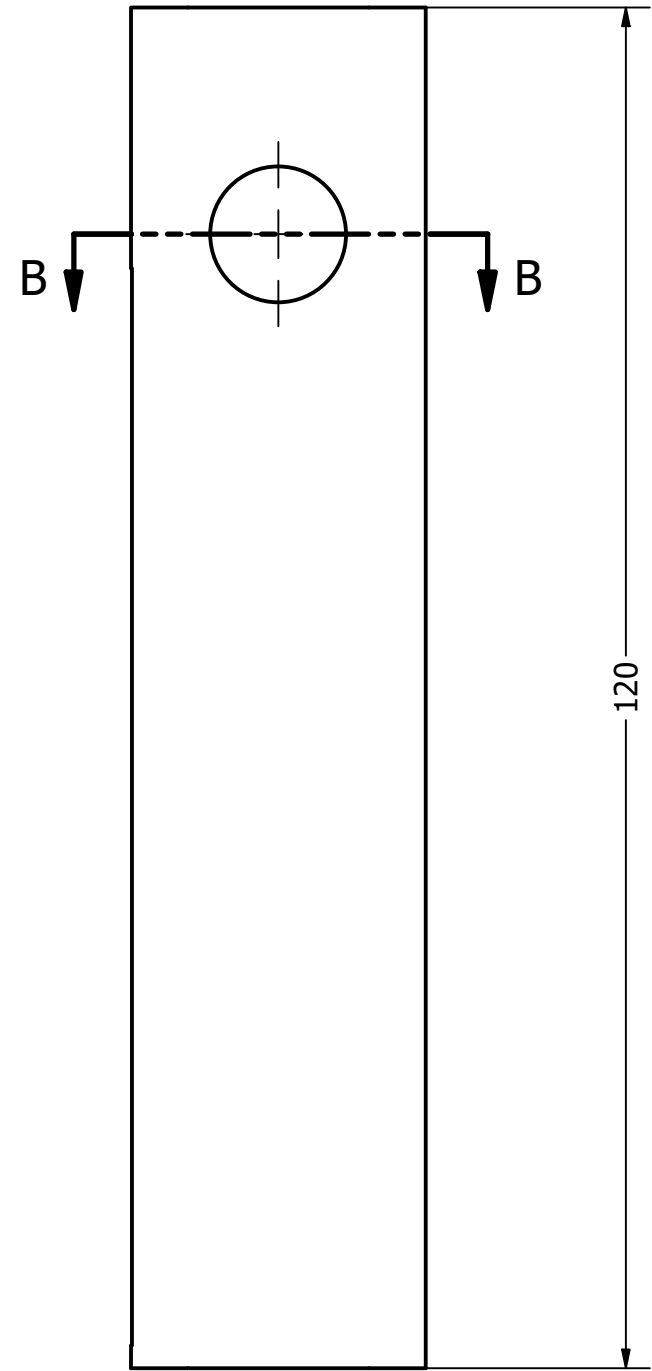
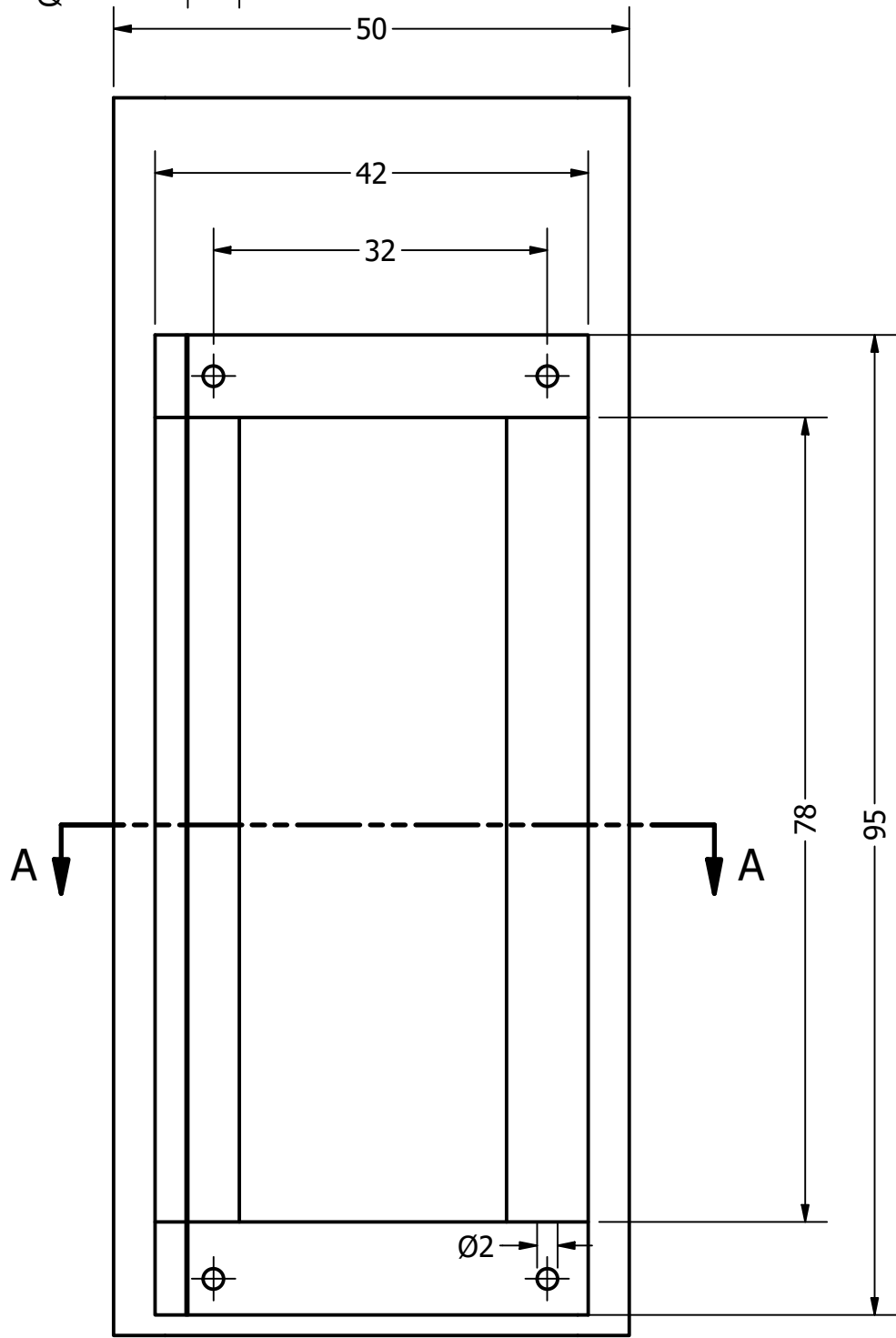
Listado de Piezas		
Pieza	Cantidad	Nombre de la pieza
1	1	Mango
2	1	Tapa del mango
3	1	Módulo del sensor
4	1	Módulo de la pantalla
5	1	Tapa del módulo de la pantalla
6	1	Tapa del módulo del sensor

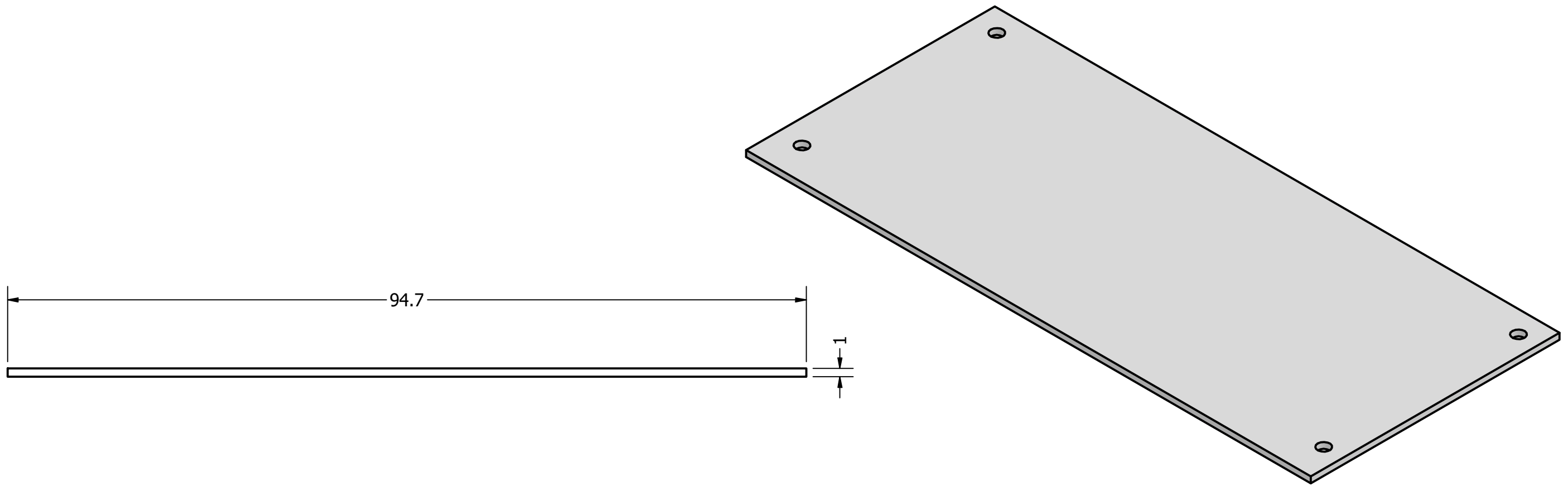
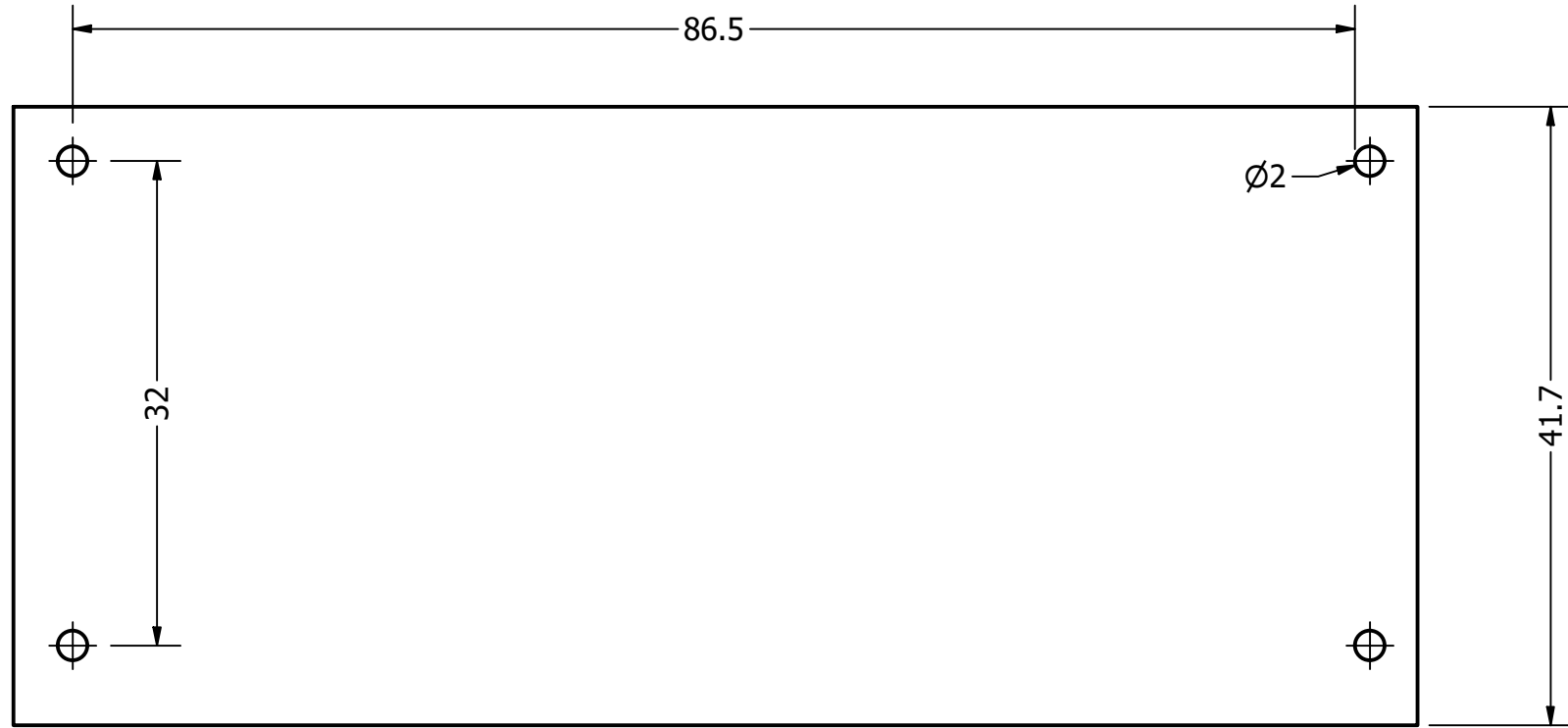




Sección A-A
Escala 3 : 2

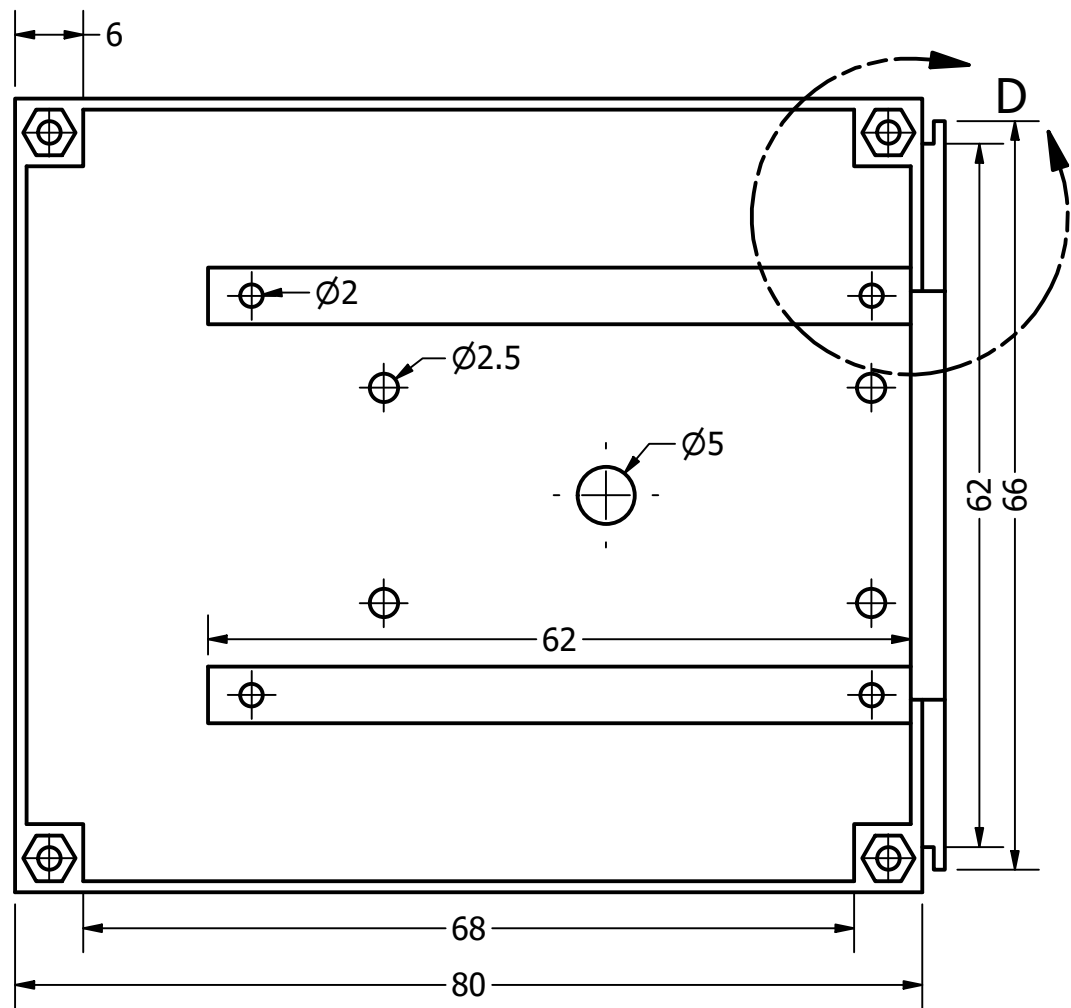


Sección B-B
Escala 3 : 2

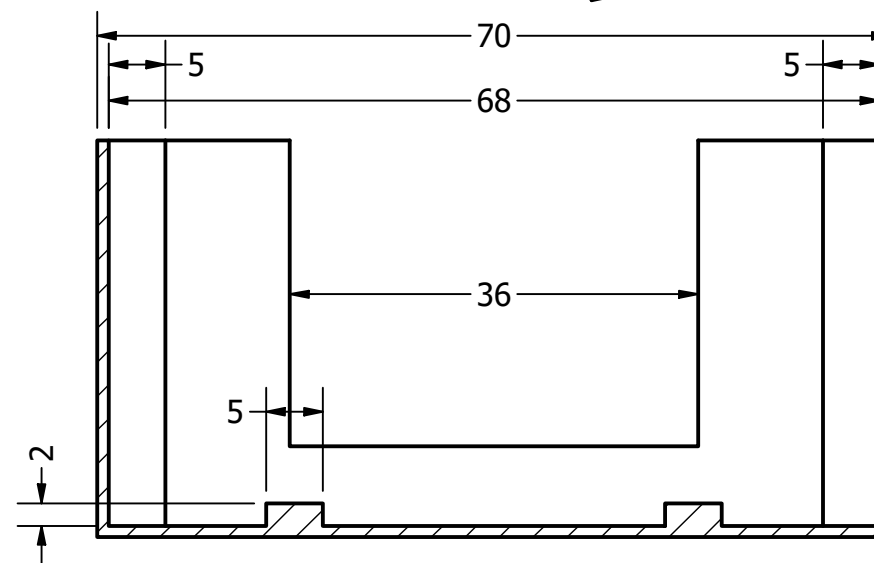
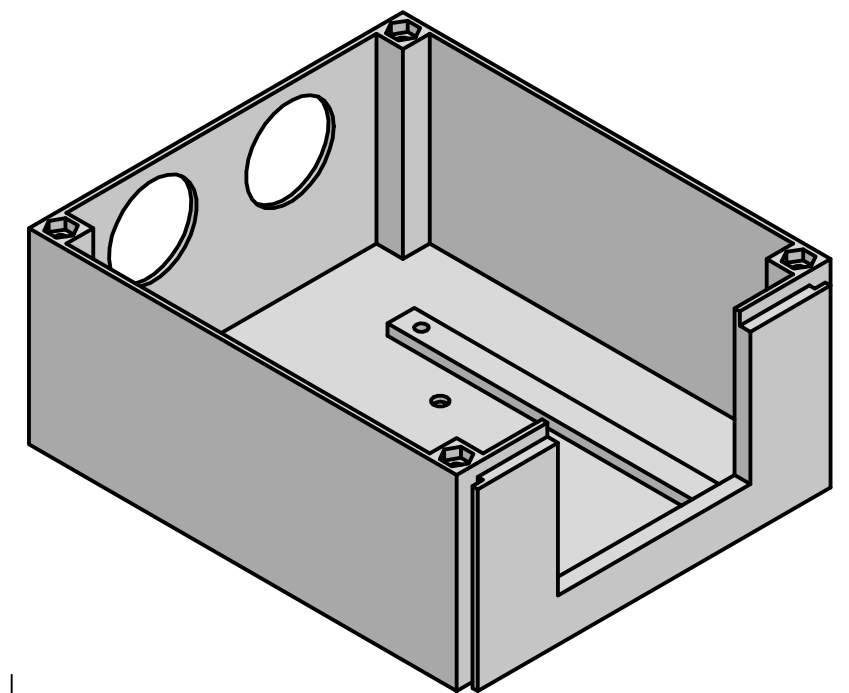
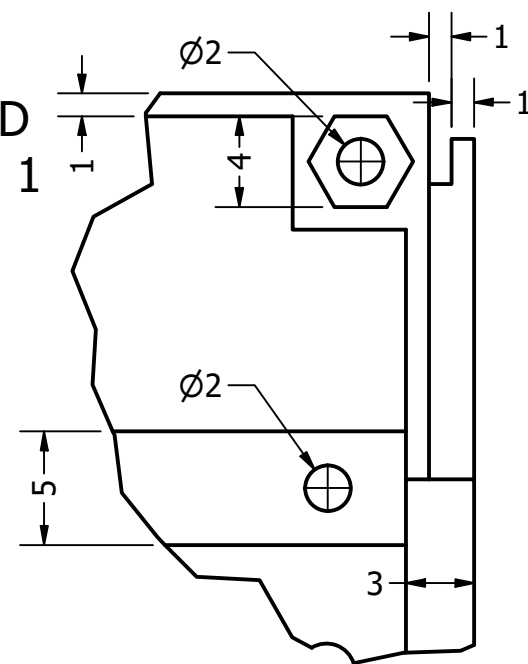




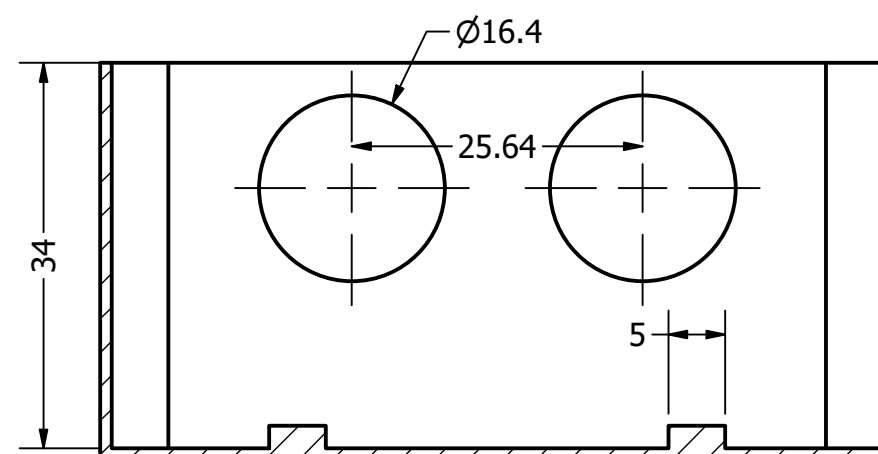
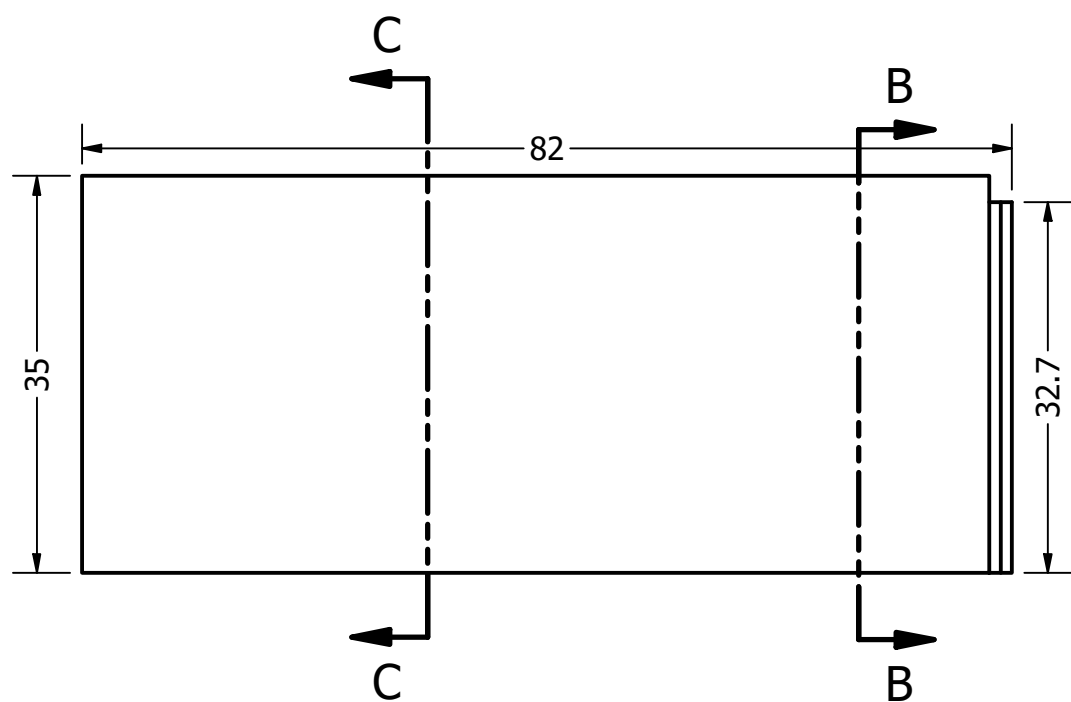
TRABAJO FINAL DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES  UNIVERSITAT POLITÈCNICA DE VALÈNCIA  ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL VALENCIA	Proyecto: DISEÑO E IMPLEMENTACIÓN DE UN VELOCÍMETRO BASADO EN ARDUINO	Plano: Tapa del mango	Fecha: Mayo 2024	Nº Plano: 3
		Autor: Carlos de la Rubia Duque	Escala: 2 : 1	



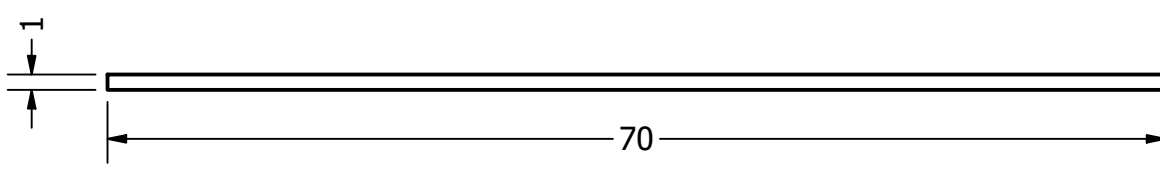
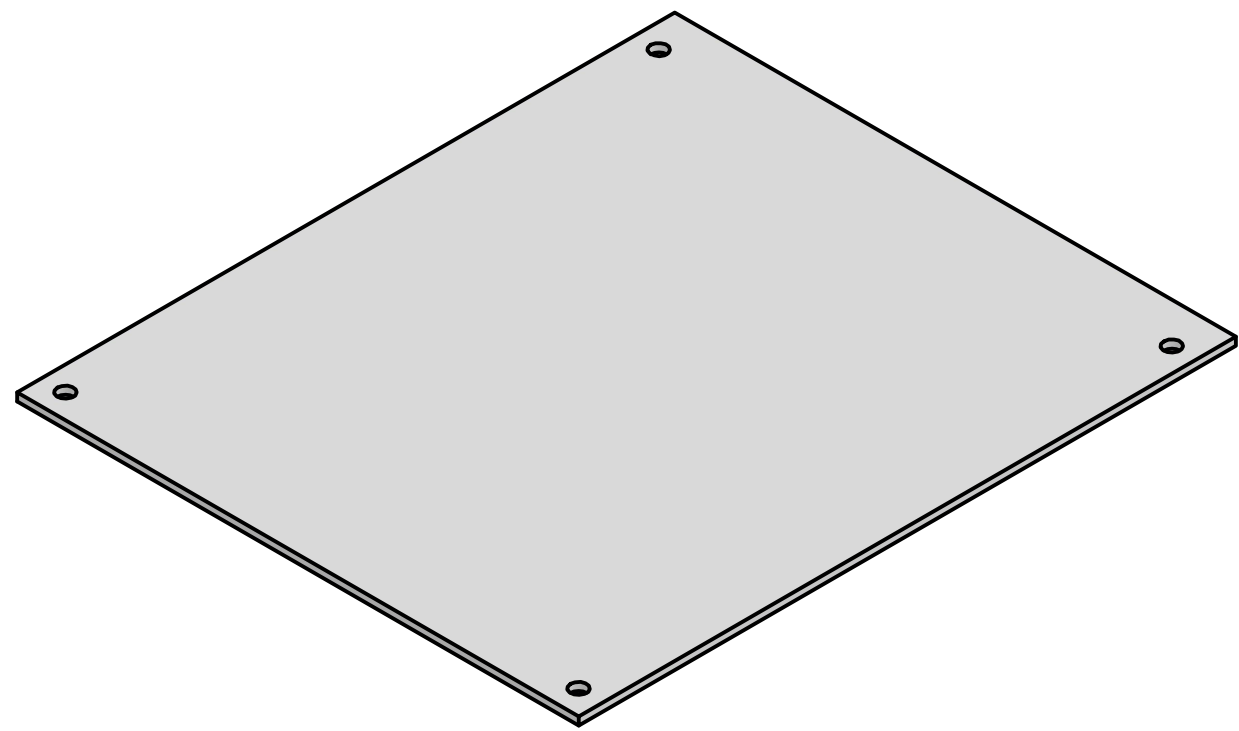
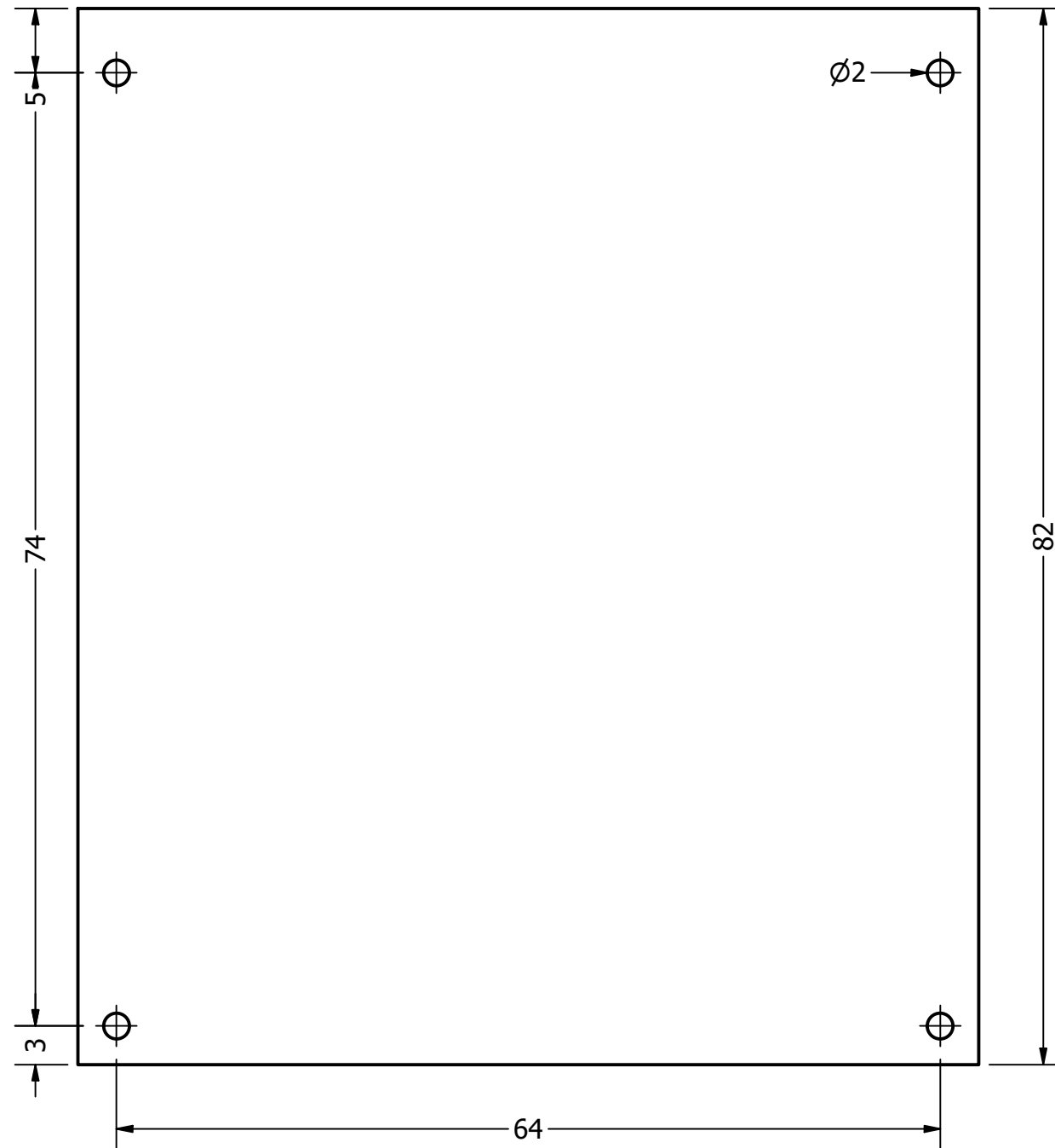
DETALLE D
ESCALA 3 : 1

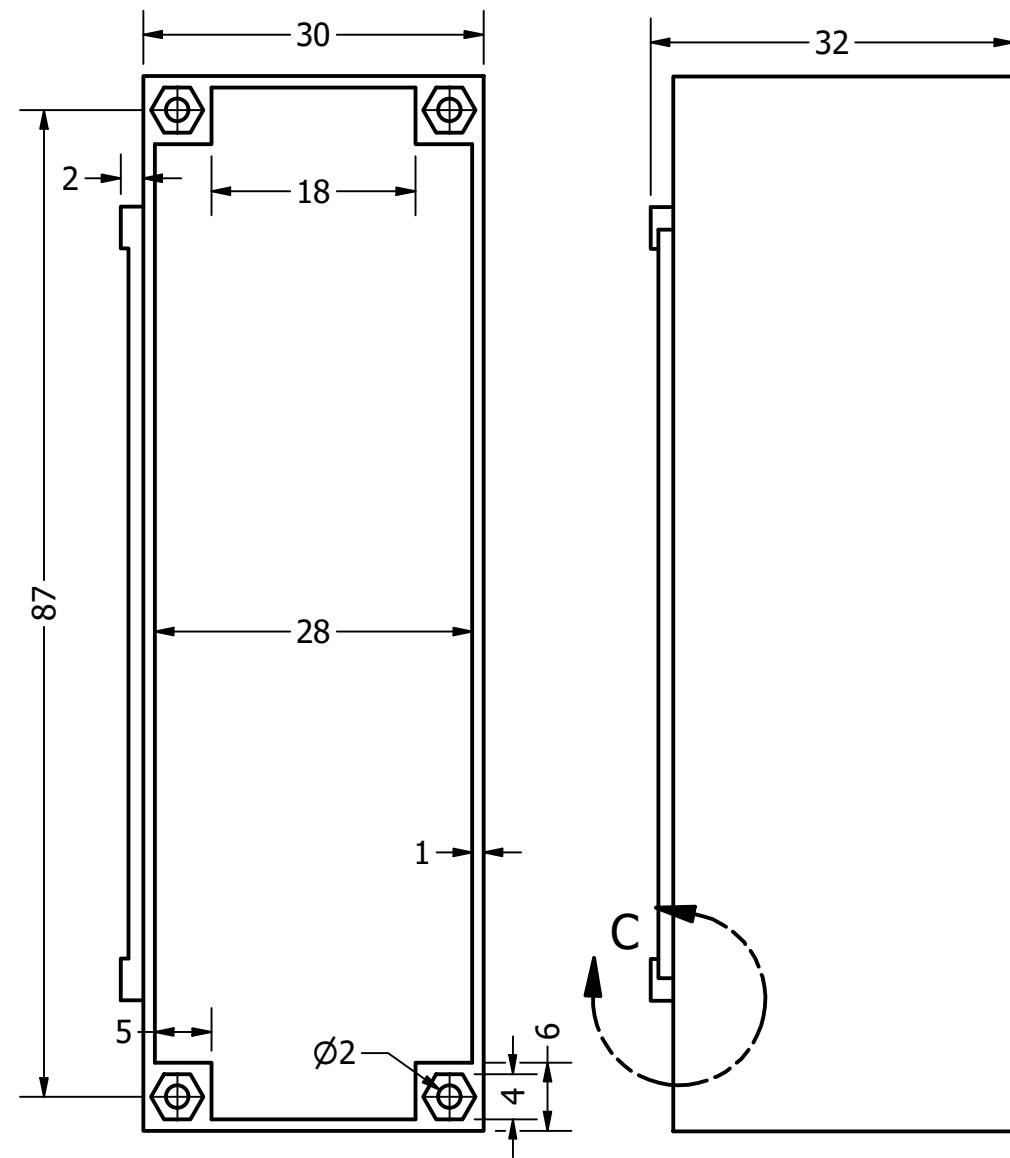


SECCIÓN B-B
ESCALA 3 : 2

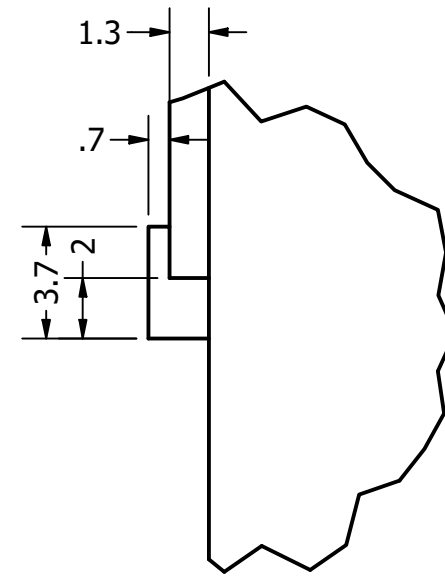


SECCIÓN C-C
ESCALA 3 : 2

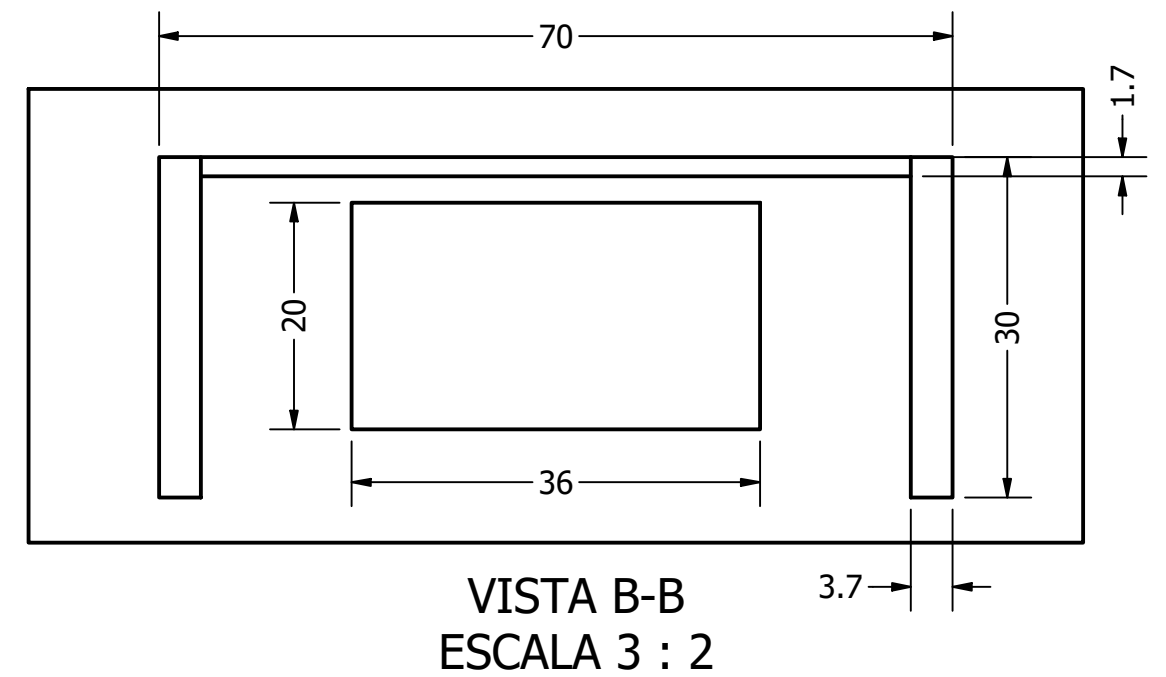
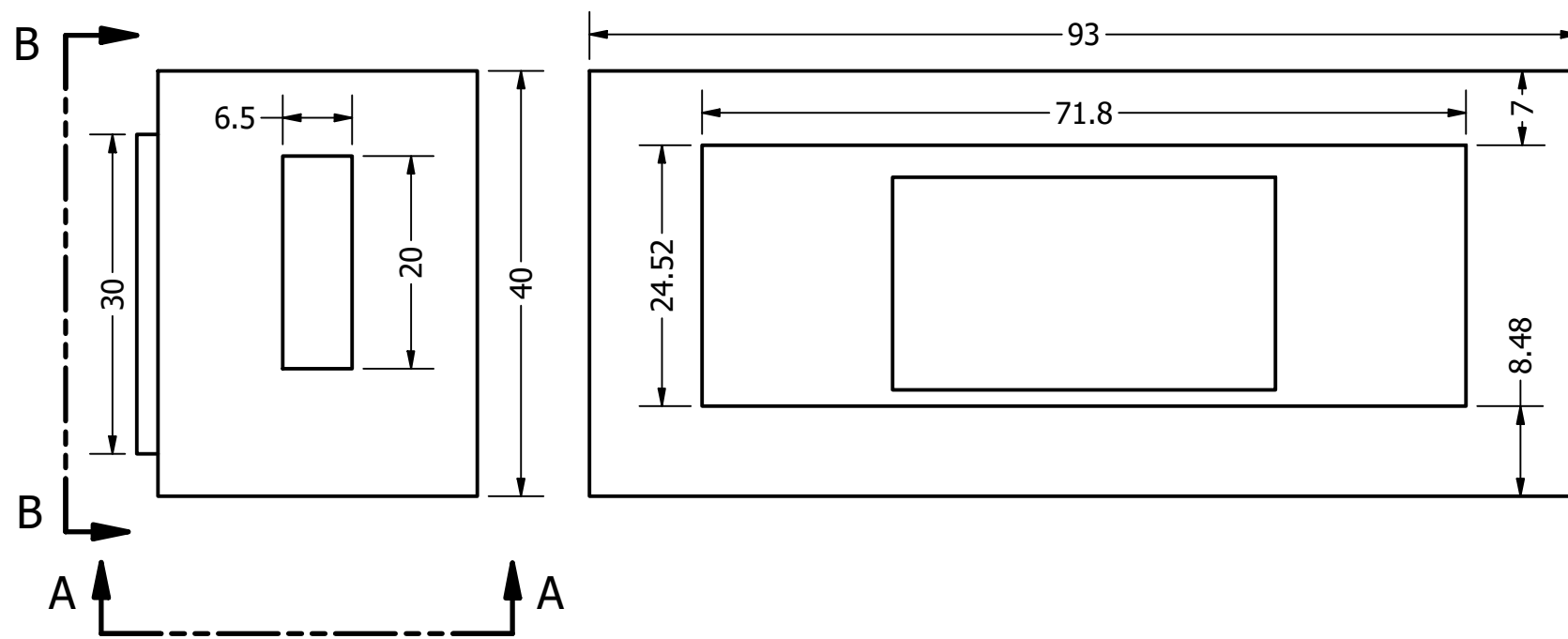
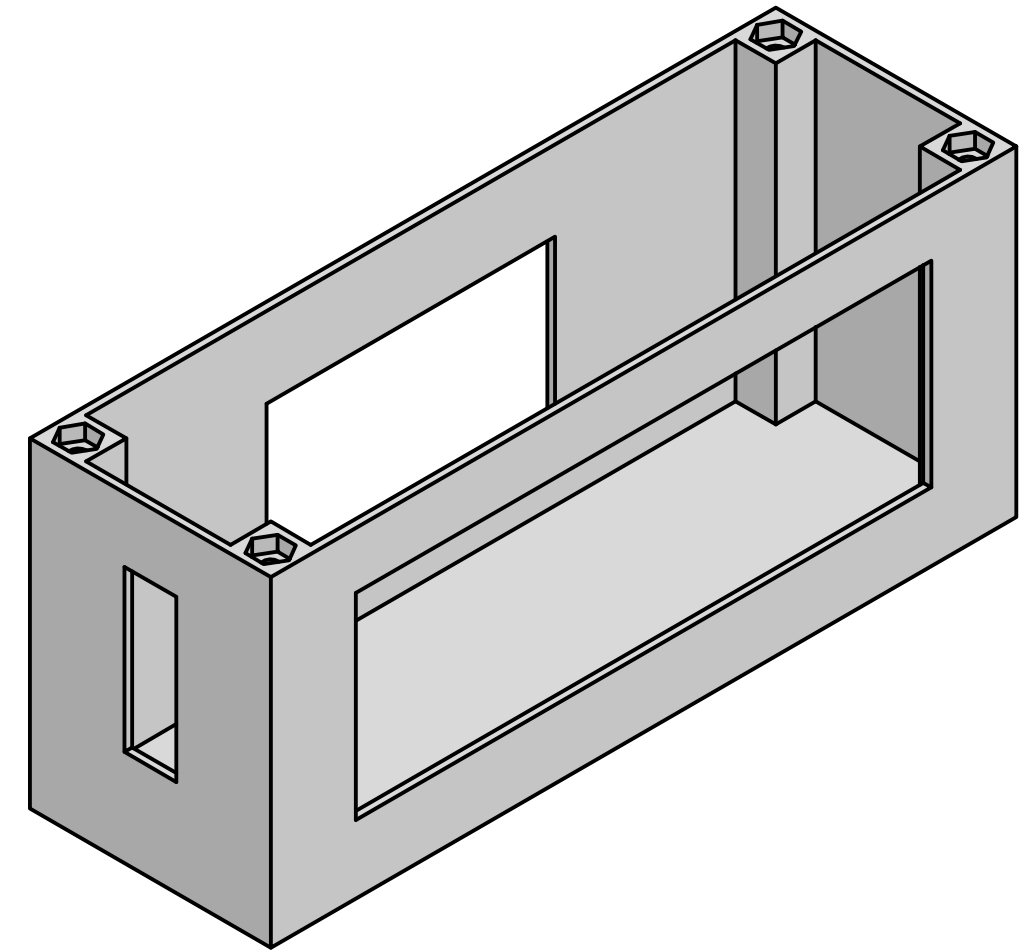




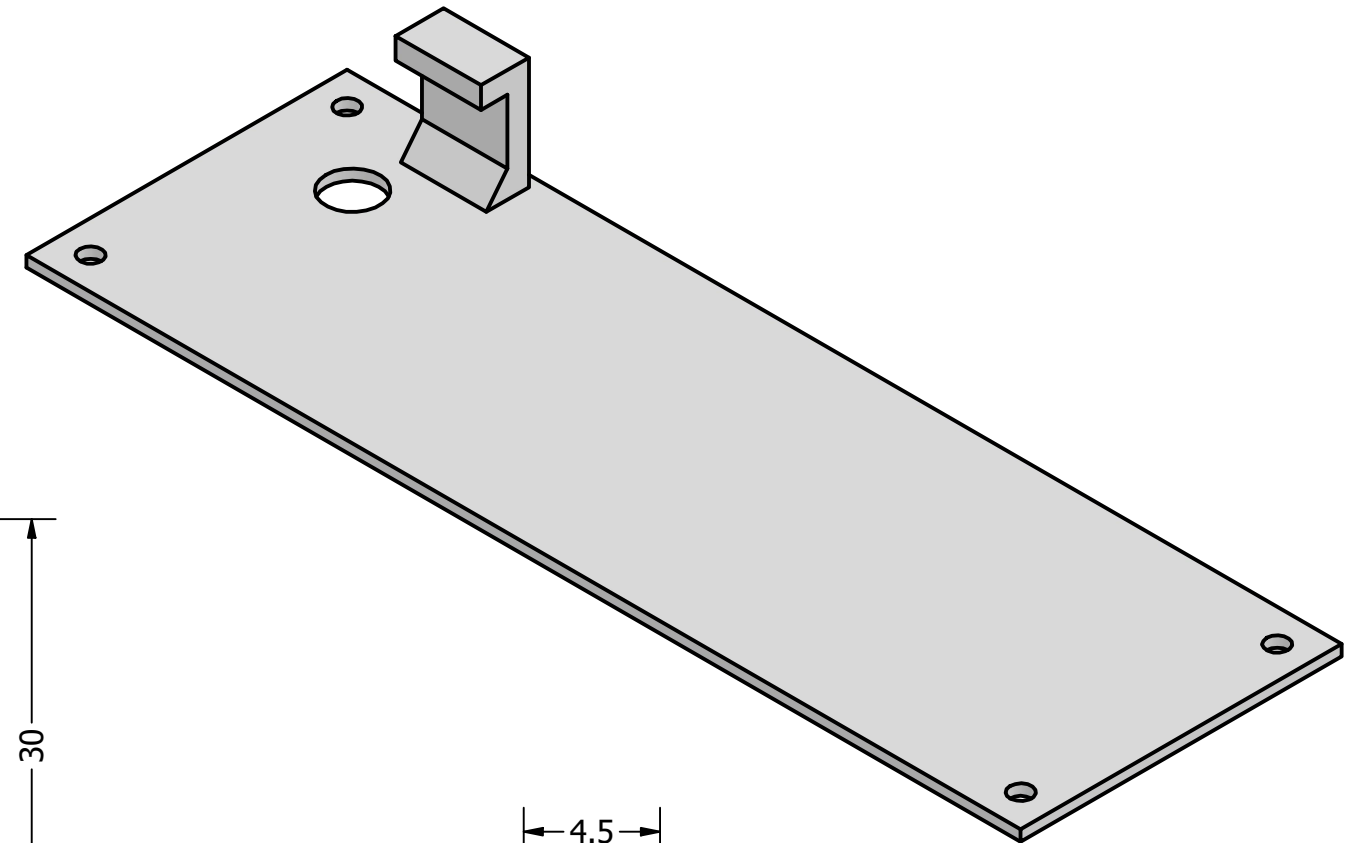
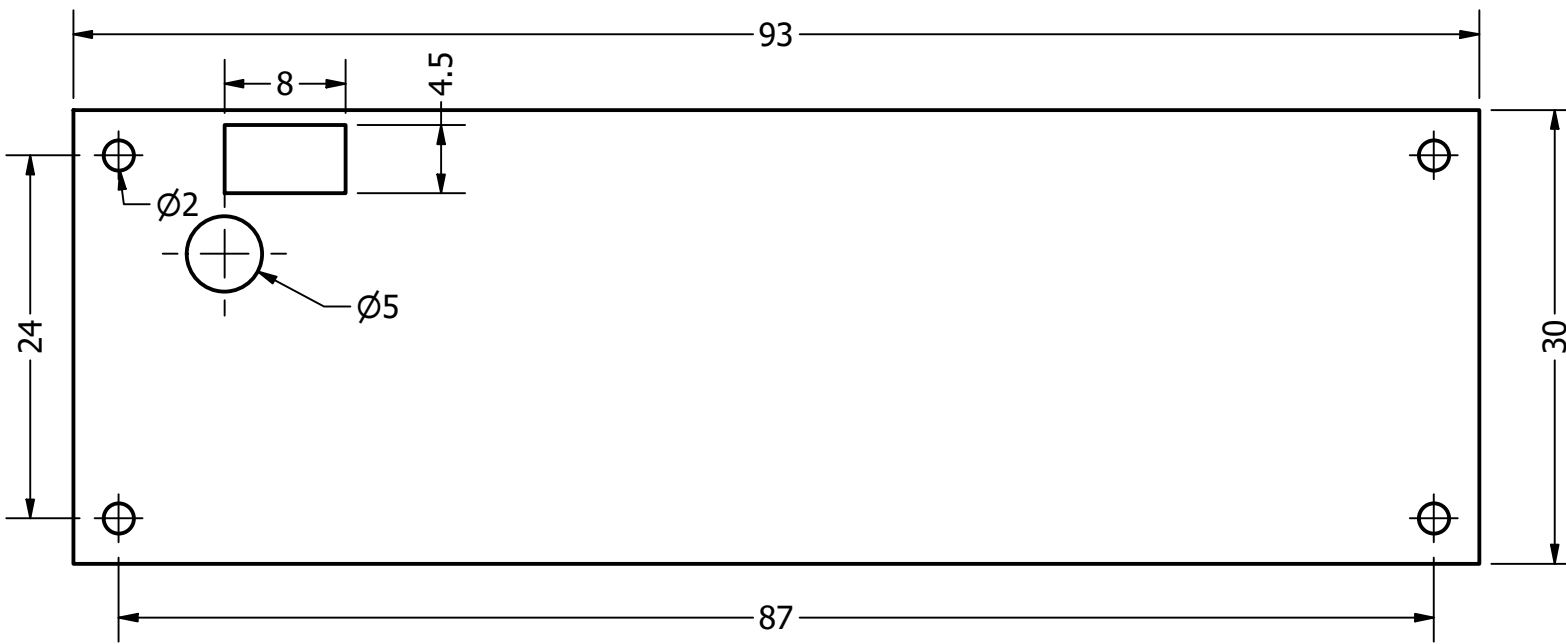
VISTA A-A
ESCALA 3 : 2



DETALLE C
ESCALA 4 : 1



VISTA B-B
ESCALA 3 : 2



DETALLE A
ESCALA 4 : 1

