

# Sumario Proyecto CATIE-FunCR FASE III

## ***Pruebas en sitio de drones, entrenamiento práctico en campo***

Tomás de Camino Beck, Ph.D., Abraham Pineda Zelaya

### Índice

Sobre este sumario	2
Descripción de Actividades	3
Objetivo General	3
Objetivo Específicos	3
Zona de trabajo	4
Sobre Hardware	4
Sobre Software	5
Recomendaciones Finales	5
Conclusiones	7
Acciones de Ejecución de fondos	7
Memoria Visual	11
Anexos	12

## Sobre este Sumario

Este documento es un sumario que describe las actividades realizadas por la Fundación Costa Rica para la Innovación para el desarrollo de prototipos de drones de ala fija, de quadcopter con lidar, y de un escáner con LIDAR. Todo el material de respaldo de los prototipos (diseños, códigos, material de respaldo). Que responde al último entregable del contrato sobre el proyecto “DESARROLLO DE MODELOS SOSTENIBLES DE SILVICULTURA Y ENLACES A LA FINANCIACIÓN PRIVADA DE LOS BOSQUES SECUNDARIOS”-Fase II.

El proyecto representa el resultado de cuatro tipos de actividades:

1. **Construcción de hardware.** La Fundación Costa Rica para la Innovación ha creado dispositivos con microcontroladores, componentes electrónicos y otros materiales de construcción, tanto para la construcción física de drones, y adaptaciones de drones, como para el uso de sensores LIDAR y otros.
2. **Creación de software.** se entiende por software al el código necesario para el funcionamiento de microcontroladores, para la navegación y funcionamiento de drones, para la navegación autónoma, y para el uso de sensores tipo LIDAR. Todo el software (código), se encuentra en el repositorio [GitHub<sup>1</sup>](https://github.com/FunCR/ProyectoCATIE) que la fundación creó para el proyecto, para poder mantener los códigos de forma dinámica. Adicionalmente se desarrolló un software en la plataforma de acceso abierto Processing.org para poder visualizar los datos provenientes del escáner LIDAR.
3. **Creación de diseños digitales.** Para la construcción de prototipos se desarrollaron y modificaron diseño múltiples para la fabricación digital de los

---

<sup>1</sup> <https://github.com/FunCR/ProyectoCATIE>

drones de ala fija, del andamio para el dron DJI, y para el escáner LIDAR. Todos los archivos digitales 2D y 3D se encuentran en el repositorio GitHub<sup>2</sup>.

4. **Transferencia de Conocimiento.** Durante esta fase del proyecto, se capacitó en tiempo real sobre el uso de las tecnologías que serán importantes para el personal de CATIE. Sin embargo, mucho del material generado es de carácter experimental y requiere posterior validación en algún proceso de programa de investigación.

## Descripción de Actividades

A partir de los resultados de la fase anterior (ver [sumario II](#)), se trabajó en los siguientes elementos:

1. Desarrollo de versión con vuelo autónomo, más robusta del dron de ala fija
2. Pruebas de vuelo autónomos
3. Capacitación de personal de CATIE, para vuelo de drones
4. Desarrollo de prototipo para pruebas de campo del escáner LIDAR
5. Desarrollo de código (software) tanto de los dispositivos electrónicos, como de interfaz para la integración de datos del escáner LIDAR

## Objetivo General

Validar el funcionamiento de los prototipos en su etapa final de producción para la medición de datos en el bosque.

## Objetivo específico

- Entrenar al equipo de Catie en la fabricación y manejo de drones de ala fija, su reparación y uso a partir de procesos de transferencia de conocimiento en sitio.

---

<sup>2</sup> <https://github.com/FunCR/ProyectoCATIE>

- Generar pruebas de prototipos de versión avanzada en sitio, con esquemas de riesgo y de explorar su validación
- Transferencia de conocimientos en uso y construcción de drones.
- Llevar a cabo el proceso de fabricación de la versión 2 de andamios de Quadcopter.
- Generar la Construcción de la versión de prototipo final para pruebas de escáner Lidar.
- Desarrollar código (software) para el uso de los dispositivos y para visualización de datos .

## Zona de Trabajo

Para el desarrollo y prueba de drones se trabajó en las instalaciones de CATIE. Para el desarrollo del escáner LIDAR, se trabajó en el laboratorio Inventoria, pues su desarrollo se concentró en la construcción de la versión funcional para pruebas. Del mismo modo la construcción de drones con corte láser se realizó en el laboratorio Inventoría de la Universidad Lead

## Sobre el Hardware

En el proyecto se probaron y construyeron elementos de hardware cuyos detalles de construcción se especifican en anexos. Estos incluyen:

1. Construcción de drones de ala fija ver ANEXO.
2. Construcción de armazón para quadcopter DJI Phantom 4 para soportar sensores múltiples
3. Construcción de un escáner con LIDAR (ver ANEXO ), para detectar árboles en un área, y explorara la posibilidad de calcular diámetros de árboles.

## Sobre el Software

Se programaron varios microcontroladores cuyos códigos fuente (software) se puede encontrar en el repositorio GitHub<sup>3</sup> y en el ANEXO .

En este proyecto se desarrollaron varias capas de software, desde el código para programar los microcontroladores, así como un código desarrollado para visualizar los datos de forma directa del Escáner LIDAR. El código de visualización corre sobre la plataforma de Processing.org, la cual se puede descargar gratis. Instrucciones de cómo ejecutar el código en processing se encuentra en el ANEXO.

## Recomendaciones Finales

- Los prototipos construidos muestran la posibilidad de construir tecnología propia. Sin embargo es necesario definir objetivos de investigación definidos en un proyecto de análisis de datos con el fin de poder validar el uso de la tecnología en un proyecto piloto.
- Es necesario realizar ensayos experimentales para validar todas las tecnologías desarrolladas.
- Muchas de las tecnologías en el tiempo del proyectos ya han cambiado. Así por ejemplo en Enero de este año, no existía un LIDAR giratorio en el mercado de bajo costo, pero ya para final de año, hay varias ofertas en el mercado, y aun costo similar del LIDAR ite, pero con la estructura para girar 360 grados, y algunos de ellos con los sensores para navegación<sup>4</sup>. Por lo que se recomienda la actualización continua a partir del análisis del mercado y su oferta de nuevos componentes que se puede continuar probando.

---

<sup>3</sup> <https://github.com/FunCR/ProyectoCATIE>

<sup>4</sup> Ver por ejemplo <https://www.seeedstudio.com/s/lidar.html>

- Probar drones de ala fija es difícil, se necesitan espacios amplios para poder hacer pruebas. Esto dificulta el desarrollo del dron, pues solamente se podían probar en las instalaciones de CATIE.
- Debido a atrasos con la compra de algunos componentes, no se pudieron probar cámaras digitales estereoscópicas, controles sofisticados de vuelo, y otros componentes que eran necesarias para probar aún más las capacidades de vuelo de estos drones. Por lo que se recomienda que con el material entregado en esta fase y la transferencia de conocimiento trasladada al equipo de CATIE , se continúe con las pruebas.
- El escáner LIDAR requiere de un análisis profundo de datos de campo. En este fase se diseñó un algoritmo para poder agrupar en tiempo real los puntos e identificar árboles, se recomienda continuar con más pruebas para poder validar el algoritmo en diferentes aplicaciones.
- Se recomienda explorar el uso de cámaras NDVI, las cuales fueron encargadas pero no se recibieron para la conclusión de este proyecto.
- Algunos de los controladores requeridos para el manejo de drones en este proyecto no fueron recibidos a tiempo, sin embargo FunCR, utilizó equipo propio para darle continuidad al proyecto. FunCR recomienda a los encargados de CATIE, continuar las pruebas con los componentes que no llegaron a tiempo,estudiar estos nuevos controladores de manera independiente (las indicaciones se dejaron documentadas)
- Este tipo de investigación en el desarrollo de prototipos físicos requiere un flujo de trabajo constante y la disponibilidad de un laboratorio para pruebas y cambios constantes, por esta razón recomendamos a CATIE, establecer un laboratorio de fabricación para acelerar los procesos de prototipado y desarrollo de tecnología.
- Este tipo de exploración de tecnologías puede resultar en prototipos que no funcionan, por tanto en el flujo de proyectos de esta naturaleza, se debe tener la capacidad de adaptar los objetivos y estrategias de forma constante. Esto

quiere decir que muchas veces lo que se propone cambia de dirección por varias razones. Por ejemplo, en el proceso de desarrollo del escáner LIDAR, el precio de LIDARs para desarrollo bajó considerablemente, lo cual hubiera permitido tener un mejor Lidar con mayores opciones a un costo menor.

- Para procesar datos existen paquetes de software abierto disponibles que se recomiendan utilizar para analizar los datos de vuelo, las imágenes capturadas, y la nube de puntos. Entre estos están
  - Pix4D <https://www.pix4d.com/>
  - Librería Pint Cloud <http://pointclouds.org>
  - R <https://www.r-project.org/>
  - Wolfram Mathematica <http://www.wolfram.com>
  - Processing <https://processing.org/>
  - Arduino IDE <https://www.arduino.cc/>
  - TinkerCad <https://www.tinkercad.com/>
  - Meshlab <http://www.meshlab.net/>

## Conclusiones

El desarrollo de tecnologías propias, para extender las posibilidades de investigación es vital para generar innovación en las organizaciones de investigación. En el proyecto se desarrollaron tecnologías importantes a muy bajo costo, y con alta capacidad de modificación y transformación. Las tecnologías creadas son una base importante para poder seguir construyendo y mejorando de forma continua.

Es importante ahora incluir estas tecnologías en ensayos experimentales, primeramente con un piloto de investigación, y luego como proyectos formales de investigación. Es importante hacer estudios comparativos para ver el valor de dichas tecnologías comparadas con las tecnologías tradicionales.

Aunque mucho de lo desarrollado potencialmente se puede desarrollar aún mejor con tecnologías más caras, el poder construir estas tecnologías dentro de CATIE, permite entender las posibilidades tecnológicas, y la necesidad de adecuar tecnologías para que cumplan con los objetivos de investigación, y no que la investigación esté limitada las posibilidades tecnológicas.

En cuanto a las tecnologías desarrolladas, tanto los drones de ala fija, como el escáner LIDAR, son un punto de arranque importante para continuar investigación y desarrollo en esta línea. Las modificaciones al Drone DJI, en particular el andamiaje para colocar múltiples sensores, permiten ahora extender el set de sensores del drone, para poder medir otro tipo de variables.

Adicionalmente se adquirieron otros dispositivos de alto valor para el vuelo de drones, como lo es los controles de alta precisión y las cámaras estereoscópicas, pensamos que el equipo de CATIE debe continuar en el desarrollo de este proyecto.

Es de gran valor que el CATIE haya entrado al desarrollo de tecnologías, puesto que las tecnologías creadas permiten crecer de manera rápida, mientras personas en la organización entiendan las posibilidades y se capaciten en el desarrollo de estas tecnologías.

### **Acciones de ejecución de fondos**

#### **Participación de Expertos**

La siguiente sección presenta las acciones llevadas a cabo por cada uno de los expertos involucrados en el proyecto, para completar esta fase del proyecto.

### **Equipo especialista.**

La ejecución del proceso de investigación y desarrollo del presente contrato se encuentra bajo el liderazgo del Dr. Tomás de Camino Beck, Director de Investigación y Desarrollo de la Fundación Costa Rica para la Innovación. Entre sus funciones principales se encuentran:

- Conceptualización de la estrategia tecnológica a implementar en el proceso.
- Análisis de datos y presentación de los mismo ante el CATIE.
- Desarrollo de dispositivos basados en la tecnología de LIDAR.
- Generación de Informes técnicos.
- Coordinación con la contraparte de CATIE.
- Liderazgo de las pruebas en campo.

El diseño y desarrollo de la tecnología de drones así como de software y hardware para las demás tecnologías están a cargo del Director en Tecnología de la Fundación Costa Rica para la innovación Abraham Pineda Zelaya, entre sus principales funciones se encuentran:

- Investigación aplicada de tecnología de drones.
- Diseño y fabricación de dispositivos de mensura forestal para adherir al dron.
- Diseño y fabricación de Drones de ala fija.
- Recopilación de los datos, realizar recomendaciones para la optimización de la tecnología.
- Participación en las pruebas de campo de las tecnologías.
- Transferencia de Conocimiento al equipo de CATIE.
- Generación de datos para los informes técnicos.

El diseñador Gabriel Wheelock está a cargo de optimizar el diseño de los dispositivos y los drones fabricados, así como la ejecución de las partes que se llevan a cabo en fabricación digital.

La gestión de calidad, vinculación con el CATIE en términos contractuales , revisión de informes y rendición de cuentas está a cargo de la Directora General de la Fundación Costa Rica para la Innovación la Licenciada Carolina Taborda Kruse.

### Ejecución de Tareas

Cantidad de días	Tarea	Responsable
40	Optimización del prototipo del LIdar: diseño y fabricación	Tomas de Camino
300	Diseño y fabricación en 3D de las estructuras del dron de ala fija , el	Gabriel Wheelock

	dron Phantom y del Lidar, preparación de planos de especificaciones técnicas y preparación de archivos digitales.	
8	Reuniones de planeamiento y diseño de capacitación en campo	Tomas de Camino, Abraham Pineda y Gabriel Wheelock
40	Fabricación de prototipo final del LIDAR y desarrollo de software en proccessing	Tomas de Camino
240	Fabricación de Drones de ala fija y diseño de software para capacitación en campo	Abraham Pineda
48	Capacitación en Campo	Abraham Pineda
80	Análisis de datos capacitación en campo	Tomas de Camino & Abraham Pineda
8	Generación de Informe segunda fase	Carolina Taborda, Tomas de camino, Abraham Pineda
<b>764</b>		

## Memoria Visual de las visitas a CATIE.







## ANEXOS

## Anexo 1

### Bitácora Pruebas de Dron de Ala Fija en CATIE

1. Lunes
  - a. Pruebas de vuelo con drone que llevamos listo.
  - b. Return to home: Volvo correctamente al punto de inicio desde una distancia de 500 metros.
  - c. Loiter: Uso combinado de dos modos de vuelo, Position hold y Altitude hold, el drone hace un círculo de 50 metros radio alrededor del punto GPS.
  - d. Vuelo manual por parte de funcionarios Catie.
  - e. Despegue y vuelo exitoso.
  - f. El ultimo aterrizaje estrello el drone contra una palmera y se destruyo. Las partes electrónicas quedaron intactas y se utilizo en el siguiente modelo.
  - g. Preparacion del espacio de trabajo
2. Martes
  - a. Impermeabilizacion de 4 kits de drones
  - b. Inicio de entrenamiento para fabricación de drones
  - c. Traspaso de partes del drone dañado a un nuevo frame
3. Miércoles
  - a. Fabricacion de drones
  - b. Explicación de cada una de las partes electrónicas
  - c. Introduction a Nav Configurator
  - d. Manejo, almacenamiento, uso recomendado y proceso de carga de baterías.
  - e. Vuelo preliminar del drone transplantado.
  - f. Se quemó el ESC 2 veces.

- g. El nuevo CartoFoam utilizado para la fabricación es mas pesado, el gramaje de del papel es mas pesado que el anterior material, un cambio de pero de alrededor de 300 gramos. La misma situación esta presente con el frame interno de impresión 3D.
- h. El puerto USB de las tarjeta resulto ser muy débil no soporta la conexión y desconexión constante del cable en ángulos incomodos, se desprende.
- i. Se agrego Bluetooth al diseño como setup standard.

#### 4. Jueves

- a. Montaje de electrónicos
- b. Montaje correcto del GPS
- c. Montaje del controlador de vuelo
- d. Montaje del ESC
- e. Vuelo preliminar
- f. Se quemó otro ESC: \*El primer ESC se quemó por que le entro agua, los otros dos tuvieron un problema del voltage entre el Administrador de energía y el regulador interno de 5v. Se resolvió al quitar el cable de 5 voltios del breakout connector del ESC\*
- g. Los ESC de 30 Amperios mostraron problemas para mover el peso del nuevo frame.

#### 5. Viernes

- a. Uso de Inav Configurator:
- b. Configuración específica para Ala fija.
- c. Comandos de Inav Configurator CLI:
  - i. set max\_angle\_inclination\_rll = 600
  - ii. set max\_angle\_inclination\_pit = 600
  - iii. small\_angle = 180 // Para poder armar en cualquier ángulo
  - iv. set nav\_rth\_altitude = 8000
  - v. set nav\_rth\_alt\_mode = CURRENT

- vi. set nav\_rth\_allow\_landing = NEVER
  - d. La batería tuvo que moverse mas atrás para estabilizar el nuevo centro de gravedad del air frame.
  - e. Los ESC y motores de los prototipos no manejan bien baterias de 4 celdas, \*cambiar estos componentes\*
  - f. Las aspas de 10 pulgadas de vuelo lento consumen demasiado amperaje y causa calentamiento excesivo en los ESC actuales.
  - g. Este día entregaron parte de los componentes faltantes.
6. Sábado
- a. Vuelos
  - b. Reparacion
  - c. Introduction a impression 3D y uso de Prusa MK2

---

## Guia Drones Ala Fija

### Materiales:

- 1 Kit completo cortado en carto-foam, 22 piezas en total.
- 1 Controlador de vuelo, F3 o superior.
- 1 GPS para el controlador de vuelo F3
- 1 Receptor de control remoto de 2.4 Ghz o similar.
- 1 Control remoto de 2.4Ghz o similar
- 1 Administrador de voltaje
- 1 Motor Brushless 2215 de 1050KV o superior
- 1 Controlador de velocidad de 30 amperios o superior para materias de hasta 6 celdas
- 2 Micro servos de 9.5g preferiblemente de engranaje metálico
- 4 Extensiones para servos
- Camara FPV de 600TVL o superior
- Transmisor de video de 5.8ghz de 200 miliwatts a 600 miliwatts
- Module de reception de video en 5.8 Ghz
- Monitor de video para modulo: Pueden ser los lentes Fatshark o los monitores integrados grandes
- Batería de 4 celdas de amenos 2000mah
- Placa para baterias
- Velcro
- Cinta de empaque
- Silicon
- Pistola de silicon caliente
- Cutter
- 2 Alambres de acero inoxidable calibre 20 de 30 cm de largo

### Proceso de ensamble

### Fuselaje

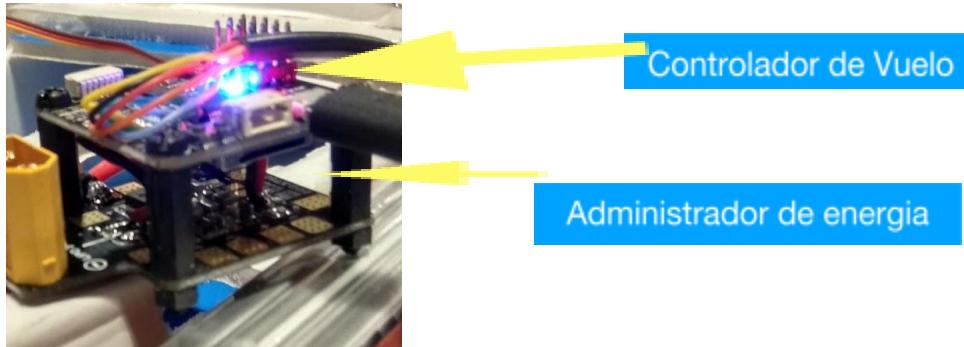
Una vez se cuente con el kit completo del fuselaje en cartón foam, ya sea fabricado a mano o en cortadora Láser, debe seguir el siguiente video instructivo para armar el fuselaje:

<https://youtu.be/B5dqMtpdbcU>

Existen dos modos de ensamble definidos por si tiene o no administración de energía.

Es necesario poner atención a los diferentes tipos de dobleces en las partes del fuselaje, en el dobles “A” las paredes caen encima de la placa central, en el dobles “B” las paredes quedan al lado de la placa central y el dobles “C” las paredes se doblan una encima de la otra.

# Controlador de Vuelo y Administrador de Energía

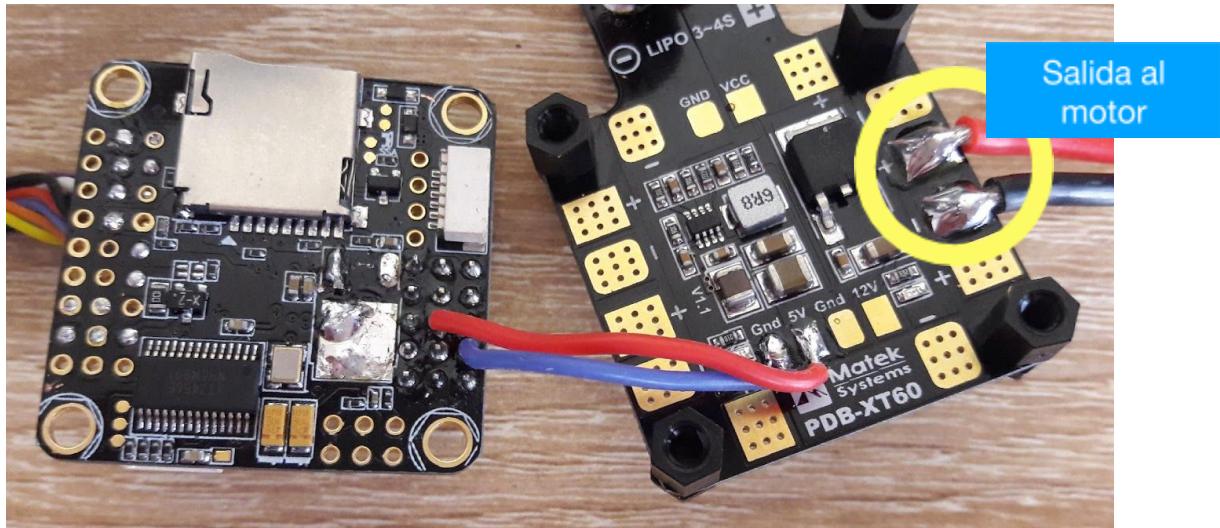


El video completa la versión del fuselaje preparada para la versión sin el administrador de energía. El administrador de energía y el controlador de vuelo se ensamblan en torre.

## Administrador de energía

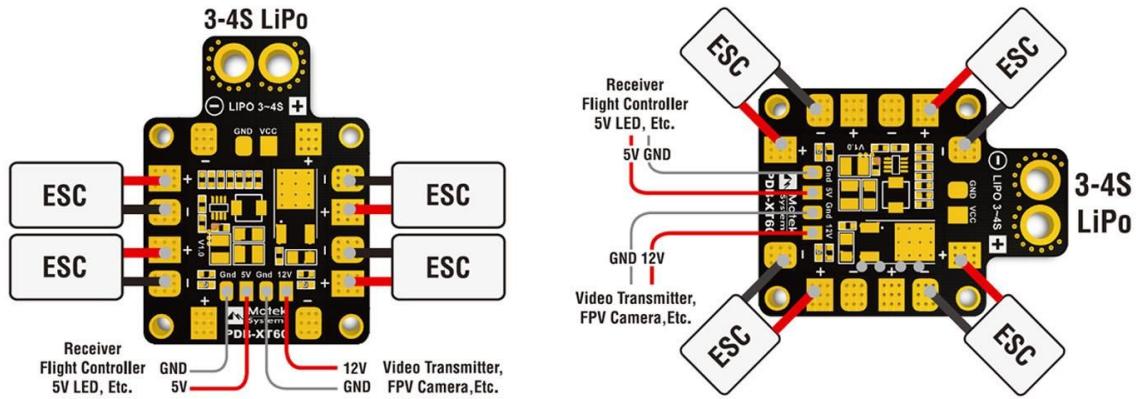
El administrador de energía proporciona diferentes voltajes:

- Voltaje directo de la batería para los motores.
- 12 voltios para transmisor de video y tracker GPS.



- 5 voltios para el controlador de vuelo, servos y otros sensores.

\*Ensamble del administrador de vuelo con administrador de energía. Conexión de 5v (5 voltios) para controlador de vuelo, sensores y servos y puerto de energía al motor.



El administrador cuenta con puertos de energía para los ESC (controladores de velocidad del motor), transmisor de video, controlador de vuelo y otros.

*\*Ensamble del administrador de vuelo con administrador de energía. Conexión de 5v (5 voltios) para controlador de vuelo, sensores y servos y puerto de energía al motor.*

El diagrama muestra todas las posibles conexiones del administrador, importante tomar en cuenta que todas la conexiones son soldadas. Conectar a todas las cosas según su disposición de vuelo.

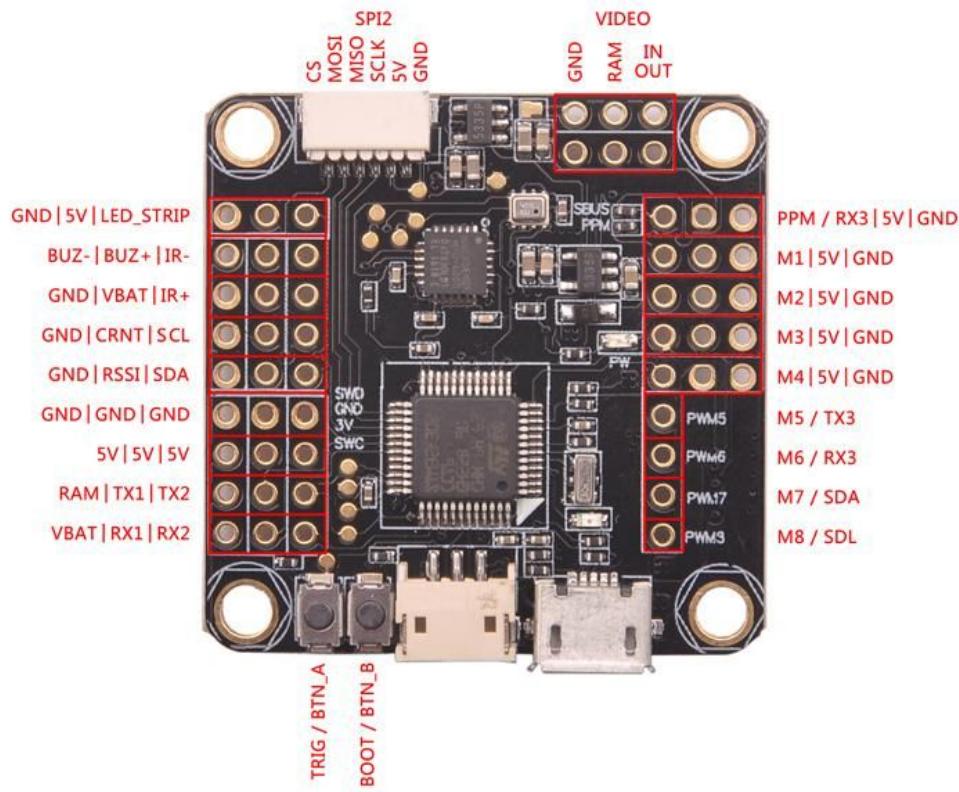
El modelo que no contiene administración de energía está dispuesto a ser una fabricación más rápida y un modelo que se puede usar para el entrenamiento de nuevos pilotos o vuelos que no requieren sensores o dispositivos que demanden mucha energía.

El administrador de energía incluye un circuito que filtra la energía para que no existe interferencia entre los diferentes sistemas presentes en el aeronave.

## Controlador de vuelo

El controlador de vuelo que estamos utilizando es de aplicación en carreras de drones, estos drones alcanzan velocidades de hasta 150 Km/h. El diseño de alta precisión y gran velocidad de procesamiento nos permite utilizar este estándar en aplicaciones que explota al máximo las posibilidades de la plataforma mientras entrega un excelente desempeño.

El controlador de vuelo es un Omnibus F3, empaquetado en una versión especial llamada Inav F3. Este controlador está dispuesto para este tipo de aplicaciones autónomas.



El diagrama de conexiones es el siguiente:

\* *Diagrama completo de conexiones.*

El controlador de vuelo tiene múltiples tipos de conexión y comunicación:

- \* Motores: M1, M2, M3, M4 son los que vamos a utilizar en la configuración de ala fija.
- \* Puerto serial: TX1-RX1 Lo utilizamos para conectar el bluetooth. TX2-RX2 funciona para el GPS
- \* Puerto I2C( SDA, SCL ): Aca se conecta el magnetómetro externo presente en el GPS.
- \* Video In -Video Out: Entrada del video de cámara para ser procesado y poner los datos del drone en pantalla y la salida del video procesado.

### Actualización de firmware y calibración

Es necesario antes de montar el controlador de vuelo al fuselaje actualizar el firmware del controlador (software de control de vuelo que se corre dentro del controlador) y calibrar los sensores. Todo este proceso se hace con el software Inav. Este se encarga de todo el proceso de calibración, actualización y puesta en marcha del controlador de vuelo.

Descargue e instale software desde este link:

<https://github.com/iNavFlight/inav-configurator/releases>

En caso de que su computador utilice el sistema operativo Windows por favor Instale los drivers para la tarjeta utilizando la herramienta que encontrará en el siguiente link:

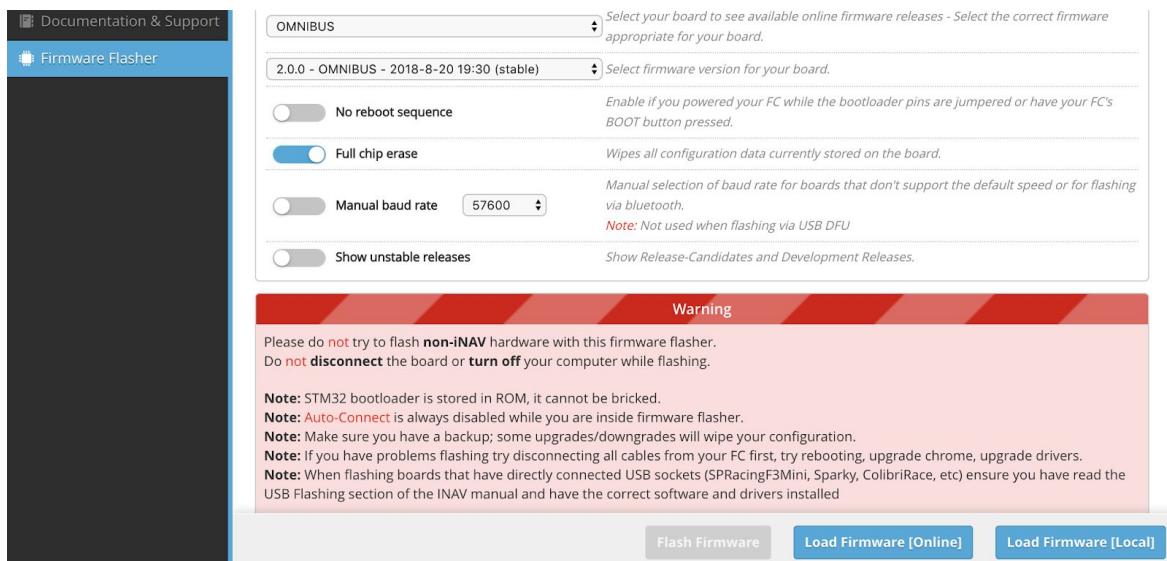
<http://zadig.akeo.ie/>

Instale la herramienta y siga los siguientes paso:

- Mantenga presionado el botón de boot loader y conecte la tarjeta a su computador usando el cable usb.
- Abra Zadig, escoja Options > List All Devices
- Seleccione STM32 Bootloader, WinUSB
- Haga clic en “replace driver” o “install driver”

Espere a que el proceso termine y continúe.

Para actualizar el software del controlador diríjase a el menú del extremo izquierdo del software y haga clic en “Firmware Flasher”.



Seleccione las opciones que muestra la imagen:

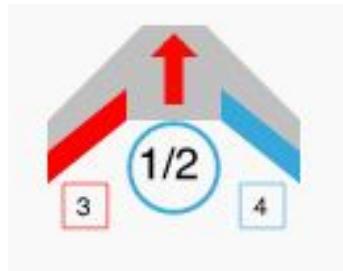
Luego click en el botón “Load Firmware [Online]”, una vez termine de cargar el firmware se activará el botón “Flash Firmware”, haga clic en este botón y espere que el sistema le muestre que está listo el proceso.

Ahora con el firmware actualizado haga clic en “Connect”, ahora en el nuevo menú de la izquierda clic en “Calibration” y siga las instrucciones de “Accelerometer Calibration” y “Compass Calibration”.

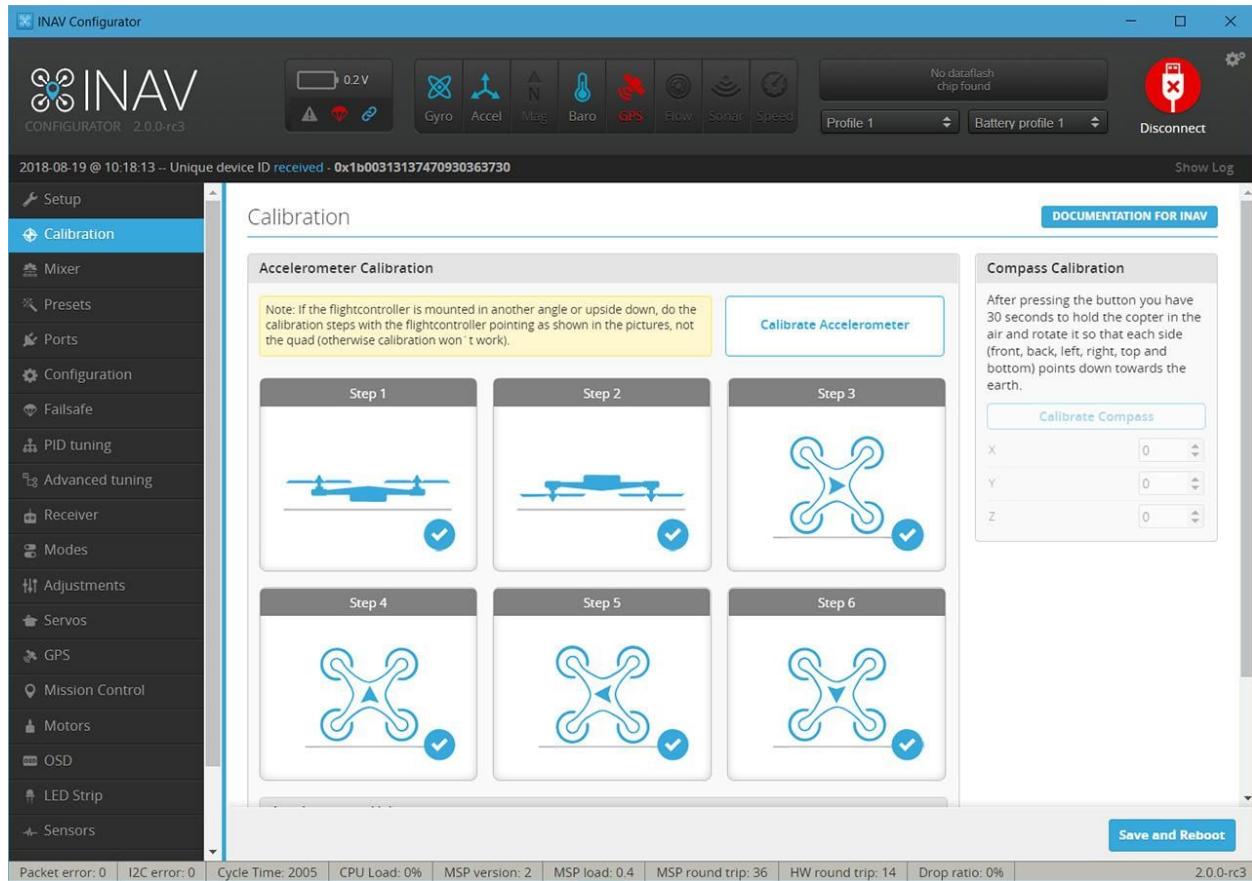
\*Ventana de calibración en “Inav Configurator”.

## Conexión de motor y servos

La configuración de ala fija requiere de un motor para impulsar la aeronave y el uso de servos que a través del alambre de acero mueven los alerones, partes móviles en el final del ala, para darle dirección al ala.



\*Diagrama de conexión motores y servos. Los números corresponden a los puertos M1, M2, M3 y M4 del controlador de vuelo.



El puerto uno y dos se pueden utilizar para conectar el controlador de velocidad del motor, a través de este puerto el controlador de vuelo hace uso del controlador de velocidad para impulsar el aeronave y cumplir con las misiones.



Es necesario conectar correctamente el motor al controlador de velocidad para asegurar que el motor gira en la dirección correcta, correctamente en nuestro caso significa que el motor

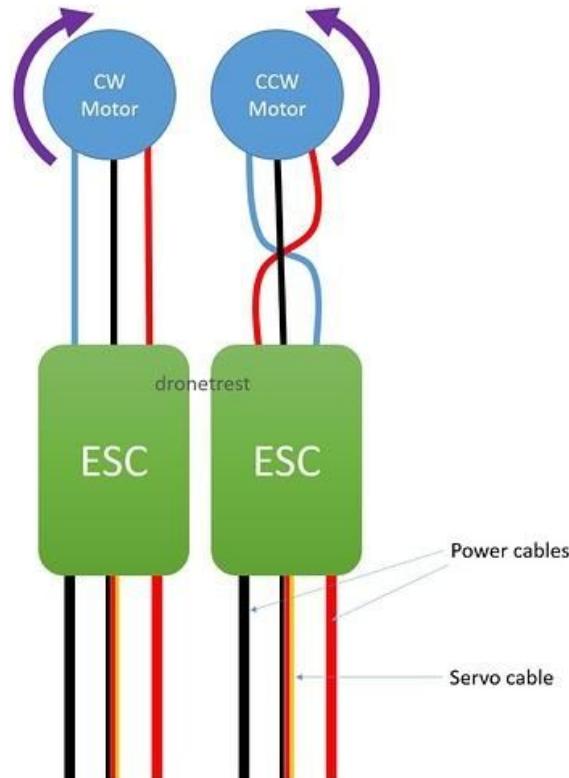
gira en la dirección que soca la tuerca del eje en el que se monta la propela. En nuestro caso queremos que el motor gire contra las manecillas del reloj.

Este es un proceso de prueba y error. Lamentablemente no todos los controladores de velocidad están montados igual y los motores no tienen el mismo embobinado.

## Este proceso se debe de hacer sin tener la propela instalada.

El primer paso es conectar los cables uno a uno de forma paralela, acto seguido se conecta el cable de control del ESC (electronic speed controller - controlador de velocidad) a un recibidor del control remoto en el canal del Throttle (potencia) y se mueve la palanca de ese canal para mover el motor.

En caso de que el motor gire en sentido erroneo se invierte la conexión de los cables de los extremos, el centro se mantiene.



\* Diagrama muestra cómo invertir los cables para cambiar la dirección del motor, dirección puede variar según el ESC y el motor. Imagen tomada de [dronetrest.com](http://dronetrest.com).

Los servos se conectan a los puertos M3 y M4, el servo de la izquierda en el M3 y el de la derecha en M4.

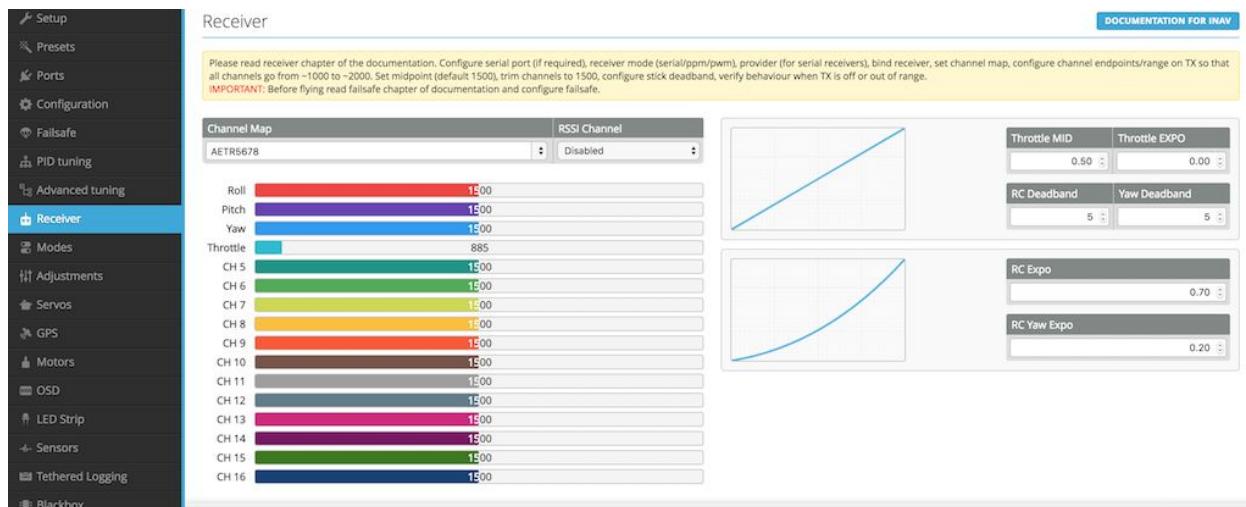
Tal como lo muestra el diagrama de conexión de motores.



# Recibidor del control remoto

En nuestro controlador de vuelo el recibidor del control remoto utiliza el protocolo de comunicación PPM, mas información en este link [https://en.wikipedia.org/wiki/Pulse-position\\_modulation](https://en.wikipedia.org/wiki/Pulse-position_modulation). Y se conecta en el puerto PPM del controlador de vuelo.

Una vez esto está hecho debemos dirigirnos al menú de la izquierda en el botón “Receiver”, si el recibidor está correctamente conectado a el puerto PPM usted verá las lecturas del recibidor moverse al mover las palancas y switches del control.



Si esto funciona correctamente tenemos que configurar los extremos y centros de la señal.

\*Lecturas del recibidor del control remoto en “Inav Configurator”.

Cuando las palancas del control se encuentran en los centros la lectura en el software debe ser de 1500. Es normal que la lectura “brinque” un poco entre números cercanos. Si esto no es así debe de corregirlo desde el control remoto. Todo control remoto es diferente refiérase al manual del fabricante para saber como corregir el “Sub-trim” de cada canal. Al hacer cambios en el control estos se reflejaran en la interfaz de Inav.

Cuando las palancas de encuentran en el extremo mínimo la lectura en el software debe ser de 1000 y cuando está en el extremo máximo debe ser de 2000. Si esto no es así debe de corregirlo desde el control remoto. Todo control remoto es diferente refiérase al manual del fabricante para saber como corregir el “Channel Endpoints” de cada canal. Al hacer cambios en el control estos se reflejaran en la interfaz de Inav.

Puede referirse a estos links para ver algunos detalles:

<https://github.com/iNavFlight/inav/blob/master/docs/Rx.md>  
<https://www.youtube.com/watch?v=nDsNWZgxmw4>

# Failsafe - Procedimiento de emergencia

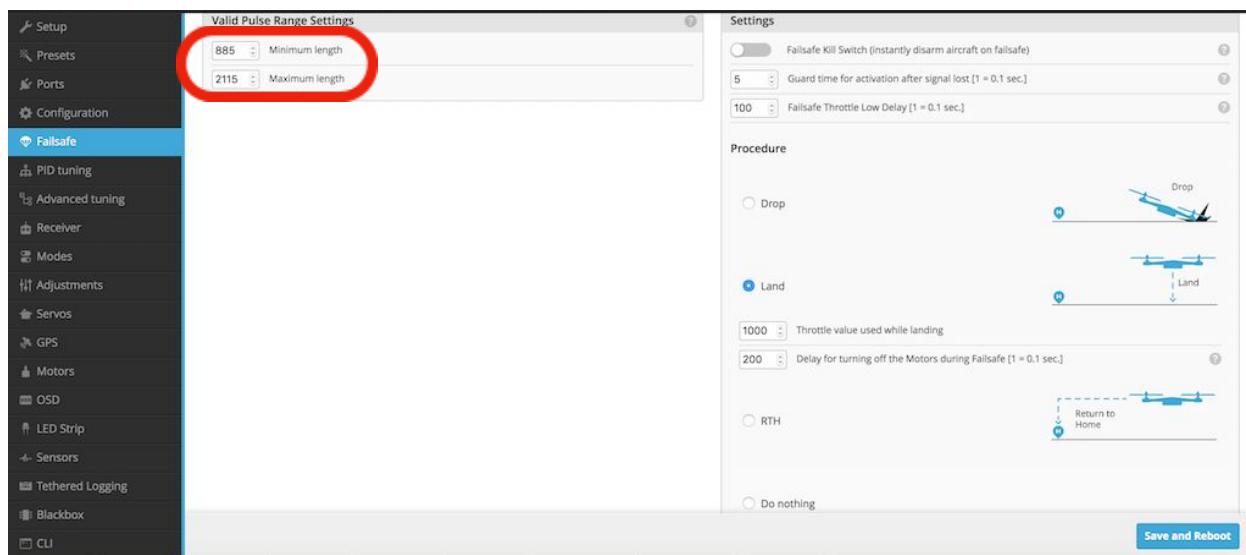
El Failsafe es la herramienta que nos permite decidir cómo se va a comportar el aeronave en caso de que pierda conexión con el control o la estación base.

Es este caso deseamos que el failsafe se active en caso de que se desconecte del control y no este ejecutando una navegación autónoma.

Para configurar esta característica es necesario ver la ventana del recibidor en Inav Configurator y una vez abierta asegurarse de que tenemos conexión con el control, este se sabe al mover las palancas del recibidor y los valores de los canales cambian en el software.

Una vez asegurado esto, el siguiente paso es apagar el control remoto y ver el cambio en las lecturas de la pantalla del recibidor. Todo recibidor de control remoto tiene un failsafe interno, que cambia la lecturas que envía para hacerle saber al controlador de vuelo que perdió conexión con el control, en algunas ocasiones es posible modificar el comportamiento de este failsafe, consulte el manual del fabricante.

Al apagar el control tome nota de cuál es la lectura más baja y más alta de los canales que antes estaban activos.



Ahora diríjase al menú de la izquierda y haga click en “Failsafe”, y en el espacio de “Minimun length” poner la lectura más baja y en el de “Maximum length” la lectura más alta.

En “Procedure” haga clic en RTH, que significa Return to home. Esto significa que el aeronave volverá al punto del que despegó.

## GPS

El Módulo de GPS viene soldado a la place Inav F3, conectado al puerto serial número 2 tal y como lo vimos al inicio de esta guía.

Ahora diríjase a la ventana de “Ports” y en línea de “UART 2” y la columna de “Sensors” escoja la opción “GPS”.

Ahora en la ventana de Configuración active el GPS deslizando el Switch hacia la derecha.

## Instalación de controlador de vuelo en el fuselaje

Una vez el controlador de vuelo se encuentre instalado con administrador de energía (en la configuración que lo incluye), controlador de velocidad, GPS y receptor de control remoto. El paso siguiente es instalarlo en el centro del fuselaje.



El compartimiento del centro del fuselaje es el lugar idóneo para ubicar el controlador.

Esta es la zona más cercana al centro de gravedad del fuselaje y por ende el mejor lugar para que los sensores del controlador de vuelo tengan las lecturas más eficientes.

## Ejemplo de instalación



## Configuración específica del software

Es necesario correr algunas configuraciones específicas para el ala fija. En el menú de la izquierda en Inav Configurator el último ítem dice “CLI” haga click en el botón. CLI significa “Command Line Interface”

Al ingresar a la ventana en la parte de abajo verá un renglón blanco en el cual puede escribir o pegar comandos, por favor copie los siguientes comandos, peguelos en el renglón blanco y presione enter:

```
set max_angle_inclination_rll = 600
set max_angle_inclination_pit = 600
small_angle = 180
set nav_rth_altitude = 8000
set nav_rth_alt_mode = CURRENT
set nav_rth_allow_landing = NEVER
set align_mag = CW270_DEG_FLIP
set align_board_pitch = 10
```

**Ahora se encuentra listo para su primer vuelo de prueba.**

```

#include <SPI.h>
#include <SD.h>
#include <Adafruit_BMP085.h>
#include <Wire.h>
#include <LIDARLite.h>

float myVal = 3.1415;

//GPS
#include "Ublox.h"
#define SERIAL_BAUD 115200
#define GPS_BAUD 9600
#define N_FLOATS 4

Ublox M8_Gps;
// Altitude - Latitude - Longitude - N Satellites
float gpsArray[N_FLOATS] = {0, 0, 0, 0};
//GPS

//Lidar
LIDARLite myLidarLite;

//BAROMETRO
Adafruit_BMP085 bmp;

// SD CARD Set the pins used
#define cardSelect 4

File logfile;

char filename[15];

// blink out an error code
void error(uint8_t errno) {
    while(1) {
        uint8_t i;
        for (i=0; i<errno; i++) {
            digitalWrite(13, HIGH);
            delay(100);
            digitalWrite(13, LOW);
            delay(100);
        }
        for (i=errno; i<10; i++) {
            delay(200);
        }
    }
}

void setup() {
    // connect at 115200 so we can read the GPS fast enough and echo without dropping chars
    // also spit it out
    Serial.begin(SERIAL_BAUD);
    Serial1.begin(GPS_BAUD);
}

```

```

// Serial.println("\r\nAnalog logger test");
pinMode(13, OUTPUT);

//LIDAR
myLidarLite.begin(0, true); // Set configuration to default and I2C to 400 kHz
myLidarLite.configure(0); // Change this number to try out alternate configurations

if (!bmp.begin()) {
  // Serial.println("Could not find a valid BMP085 sensor, check wiring!");
  error(2);
}

// see if the card is present and can be initialized:
if (!SD.begin(cardSelect)) {
  Serial.println("Card init. failed!");
  error(2);
}

strcpy(filename, "CATIE_00.TXT");
for (uint8_t i = 0; i < 100; i++) {
  filename[6] = '0' + i/10;
  filename[7] = '0' + i%10;
  // create if does not exist, do not open existing, write, sync after write
  if (! SD.exists(filename)) {
    break;
  }
}

logfile = SD.open(filename, FILE_WRITE);
if( ! logfile ) {
  // Serial.print("Couldnt create ");
  // Serial.println(filename);
  error(3);
}
// Serial.print("Writing to ");
// Serial.println(filename);

pinMode(13, OUTPUT);
pinMode(8, OUTPUT);
// Serial.println("Ready!");

}

uint8_t i=0;
void loop() {
  String dataString = "";

  while(Serial1.available()){
    char c = Serial1.read();
    //Serial.println(c);
    if (M8_Gps.encode(c)) {
      gpsArray[0] = M8_Gps.altitude;
      gpsArray[1] = M8_Gps.latitude;
      gpsArray[2] = M8_Gps.longitude;
    }
  }
}

```

```

gpsArray[3] = M8_Gps.sats_in_use;

}

}

//LEER BAROMETRO
//dataString = bmp.readTemperature();
//dataString += ",";
// dataString += bmp.readPressure();
// dataString += ",";
dataString += String(gpsArray[1],10);
dataString += ",";
dataString += String(gpsArray[2],10);
dataString += ",";
dataString += String(gpsArray[0],10);
dataString += ",";
dataString += bmp.readAltitude();
dataString += ",";
// dataString += bmp.readSealevelPressure();
// dataString += ",";
// dataString += bmp.readAltitude(101500);
// dataString += ",";
dataString += myLidarLite.distance(false);
dataString += ",";
dataString += gpsArray[3];

digitalWrite(8, HIGH);

//Guardamos datos
logfile = SD.open(filename, FILE_WRITE);

logfile.println(dataString);
logfile.close();
Serial.println(dataString);

}

```

# Construcción Escáner LIDAR

Tomás de Camino Beck, Ph.D.

## Introducción

El escáner LIDAr es un escáner de lecturas del sensor de distancia láser de alta resolución. El escáner rota 360 grados haciendo lecturas de distancia cada ciertos intervalos, y registrando el ángulo con respecto al norte magnético. La distancia máxima de lectura del LIDAR es de 50 metros.

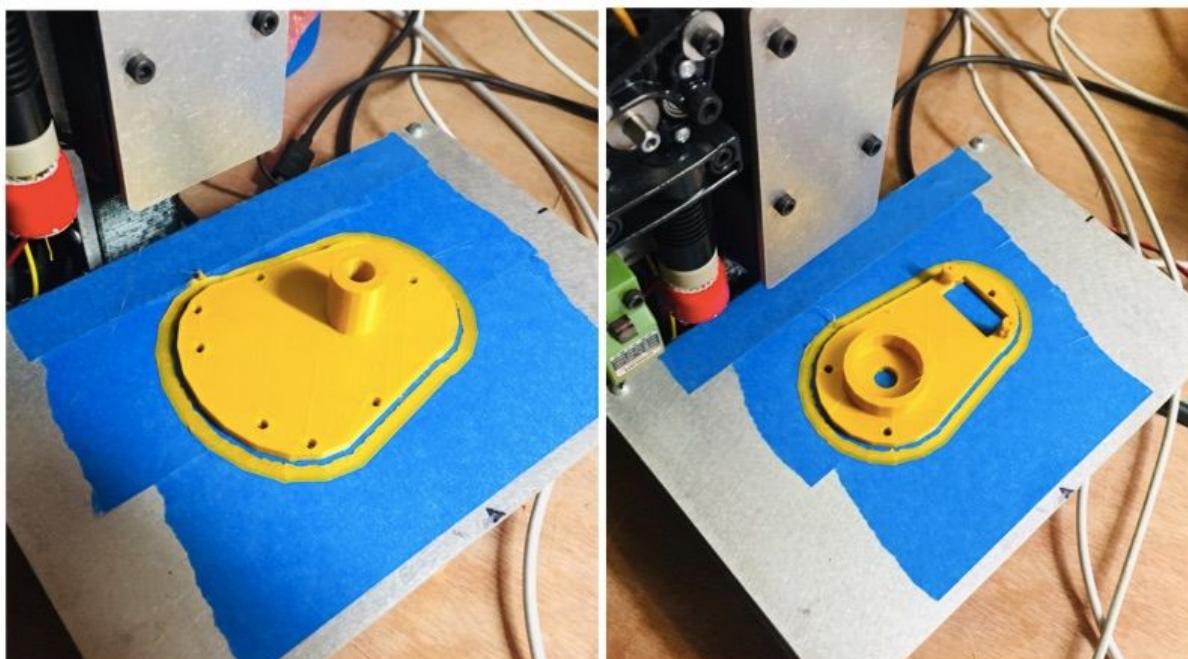
La función del escáner es detectar una nube de puntos para luego determinar la presencia de árboles en un rodal, y posteriormente calcular diámetros y áreas basales. este dispositivo se encuentra en fase de prototipo, por tanto es necesario realizar ensayos de campo para validar su funcionamiento.

## Materiales

- Rol genérico de patineta
- Microcontrolador (en este documento se utiliza un S4A)
- Servomotor de rotación continua
- Extensiones (stand off)
- Slip ring de 6 cables
- LIDAR lite
- ADAfruit 9DOF
- Impresora 3D
- Taco anclaje metal 1/2"
- Elásticos varios

## Impresión 3D

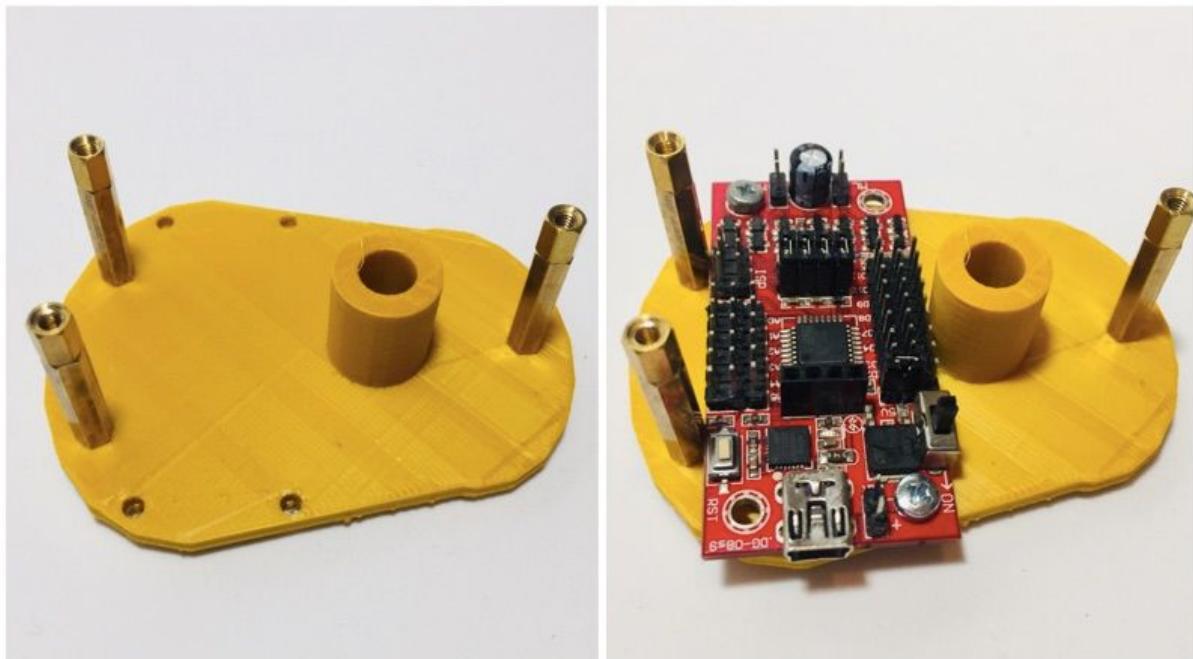
Imprimir en impresora 3D en resolución de 300 micras (0.3mm) las piezas del siguiente repositorio [GitHub](#), como se ve en la fotos.



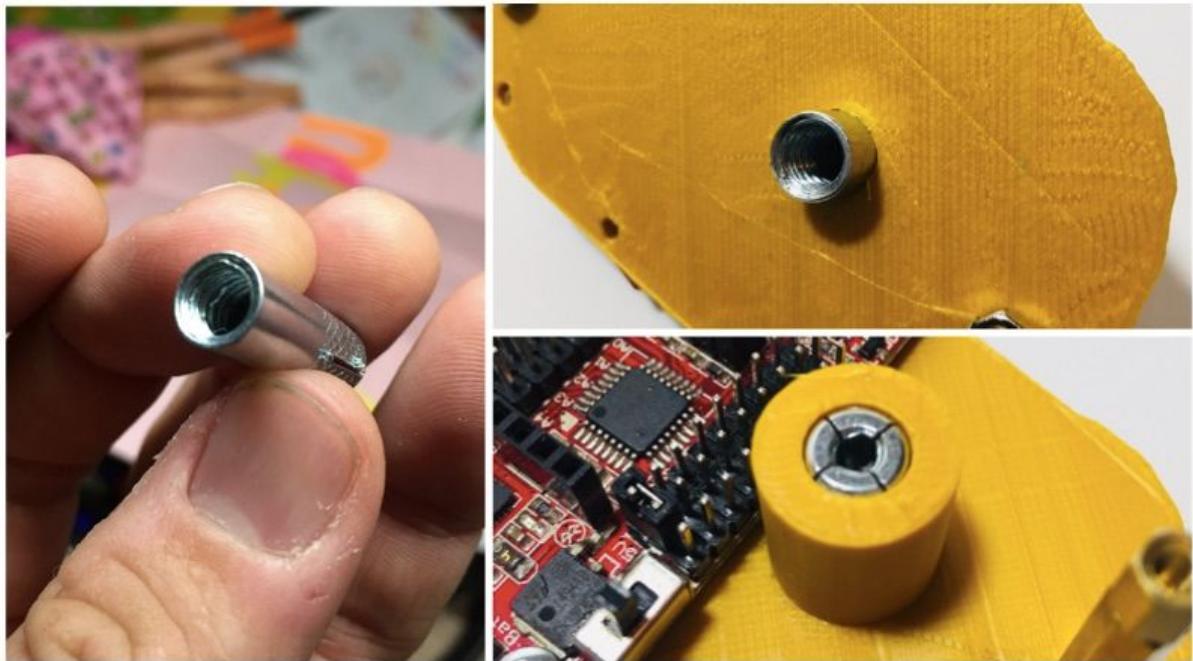
Para la impresión del prototipo se utilizó plástico PLA, con una impresora Printrbot Smalls.

## Ensamblado de la base

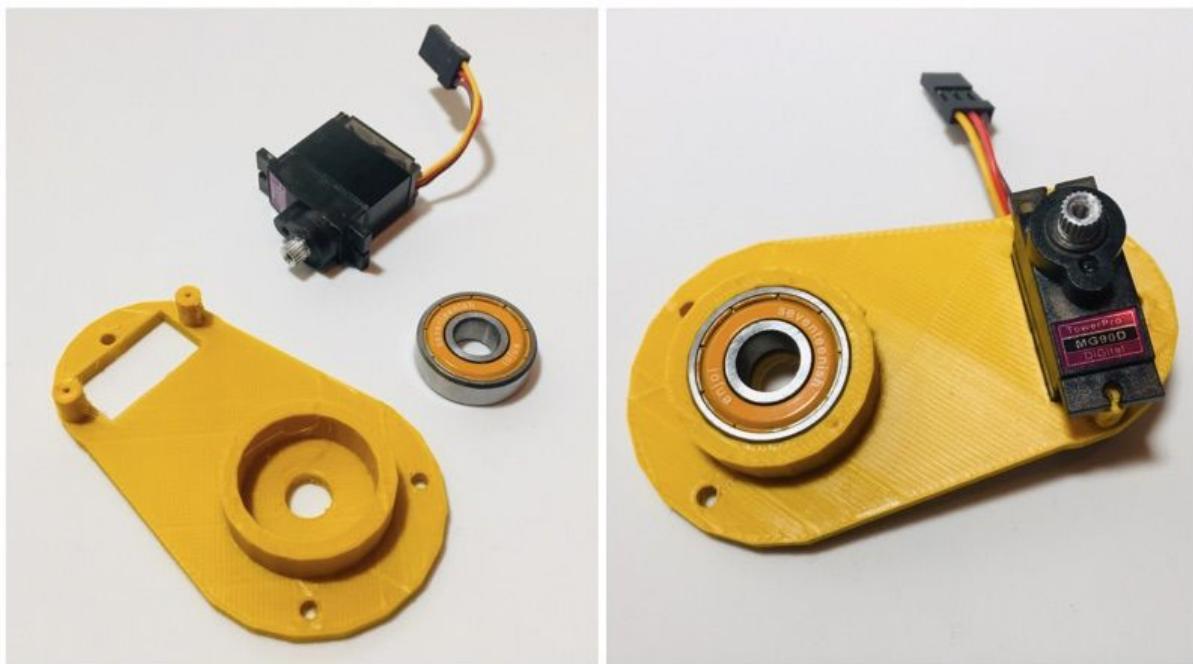
Colocar el microcontrolador y fijarlo con tornillos a la base impresa en 3D. Colocar las extensiones. En este caso se utilizan extensiones de 2cm y una adicional de 6mm. (ver fotos)



Importante colocar el anclaje de metal. El anclaje permite luego utilizar un trípode fotográfico para fijar el escáner. En la siguiente foto se muestra cómo colocarlo



En la segunda base, montar el servomotor de rotación continua y fijarlo con tornillos como se observa en la foto.

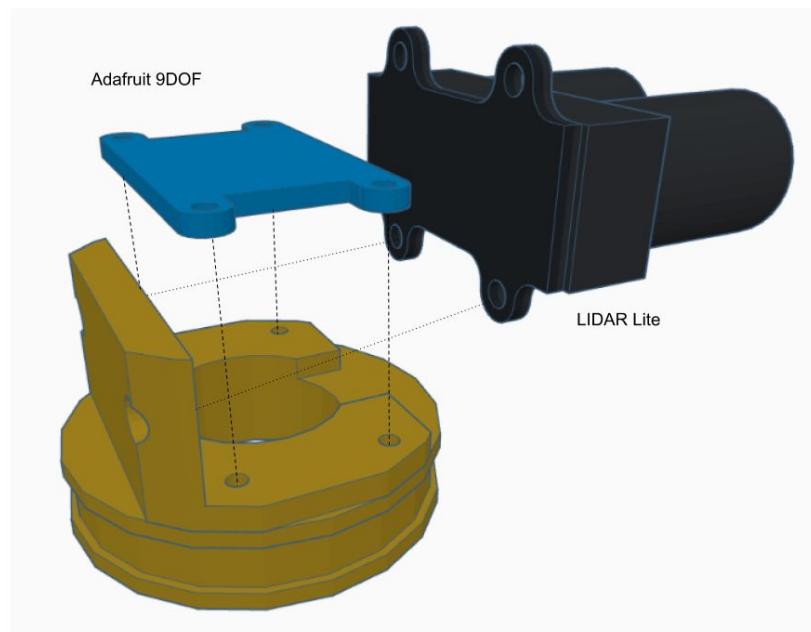


Una vez que las dos bases estén ensambladas, atornillar la base superior a la inferior como se muestra en la siguiente foto

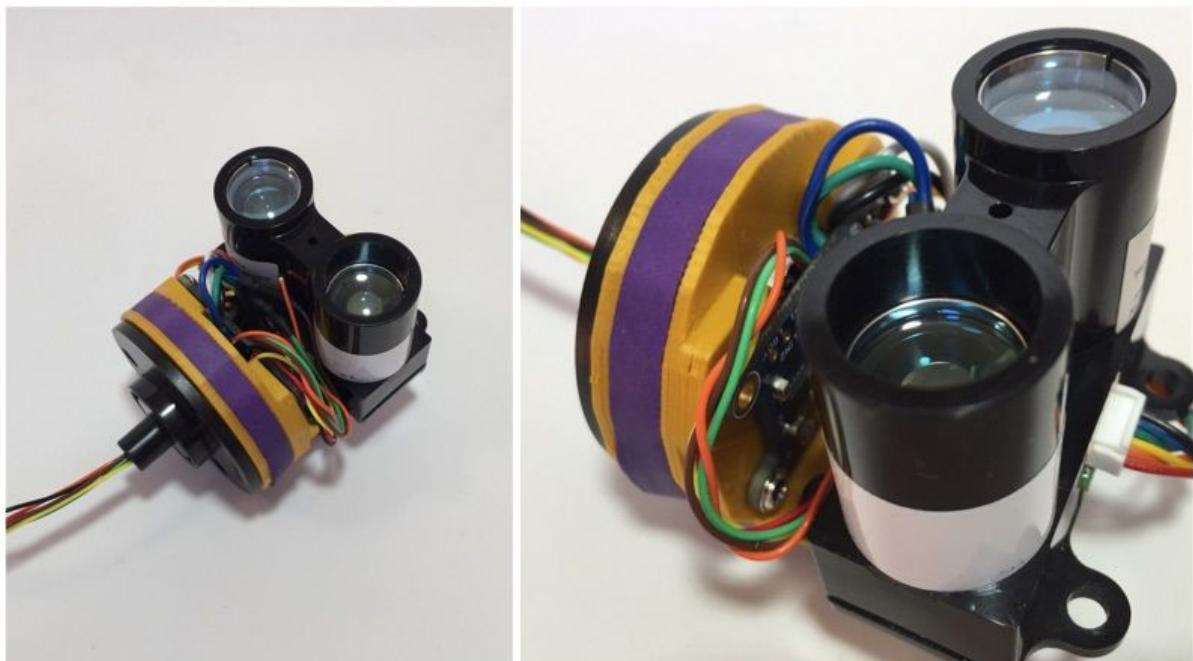


## Ensamblado del Lidar y magnetómetro

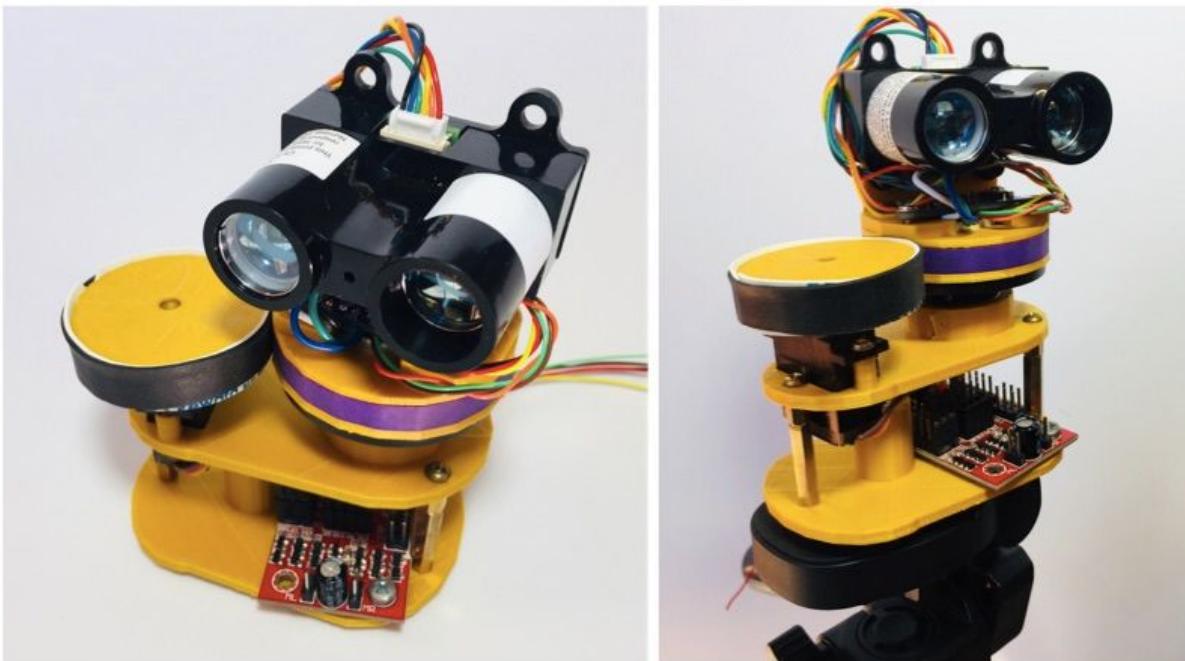
El LIDAR y el magnetómetro van colocados sobre el “slip ring” que permite la rotación de cables y que ambos sensores se muevan libremente. Ambos sensores se colocan sobre la pieza impresa de PLA como se ve en el siguiente esquema



Ensamblar el LIDAR, con el magnetómetro como se observa en la fotografía

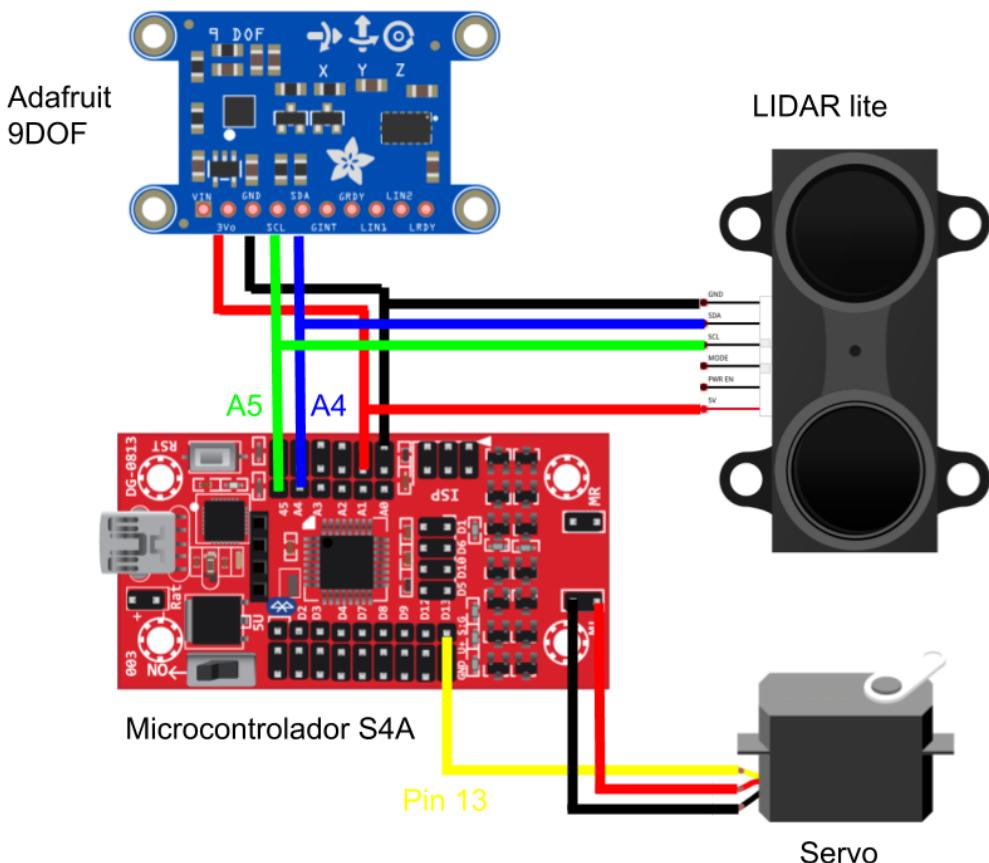


Una vez colocados ambos sensores, se colocan junto con la rueda del servo sobre la base,



## Hardware y código

El hardware se conecta de manera bastante simple. Las conexiones SDA, SCL, del LIDAR van conectadas directamente con las conexiones SDA y SCL del magnetómetro 9DOF. De igual manera las conexiones de GND y 3V. Estas se conectan a los cables de “Slip Ring”, y de allí a las conexiones de SDA (A4) y SCL (A5) del controlador S4A. El servomotor se conecta directamente al S4A en el pin número 9, pero la alimentación de poder se conecta a ML en el S4A. El esquema de conexión se puede ver en el siguiente esquema

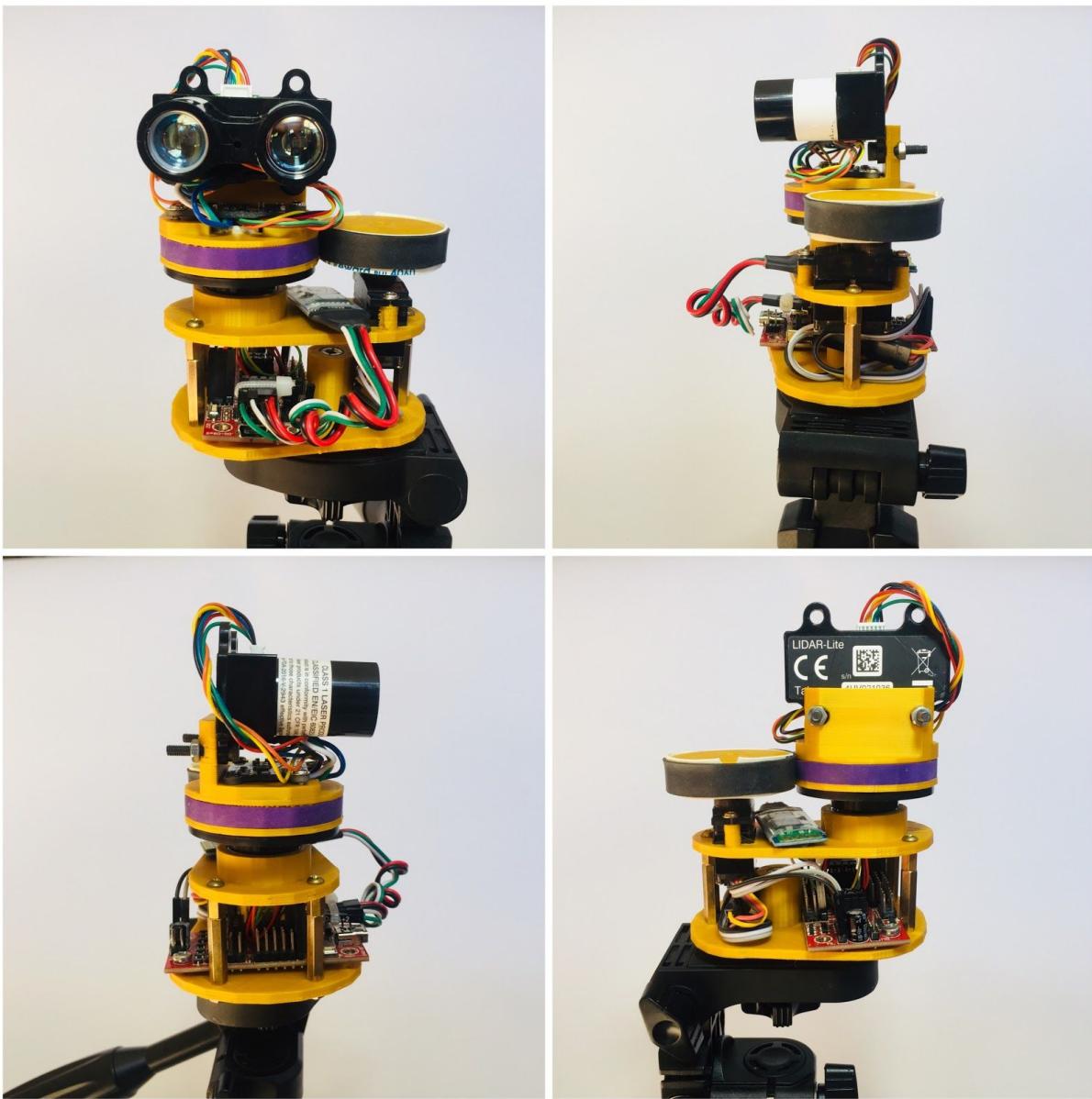


El código se encuentra en el ANEXO, o en el siguiente repositorio de GitHub<sup>1</sup>.

## Operación

Colocar el escáner sobre el trípode

<sup>1</sup> <https://github.com/FunCR/ProyectoCATIE/tree/master/TreeScanner/Code>



- Conectar el puerto USB del escáner a la computadora. El escáner comenzará a rotar y a tomar datos de forma inmediata. Para ver los datos, abrir el Arduino IDE, y abrir el monitor serial.
- Los datos son arrojados por el escáner en conexión serial a 9600 Baudios. Los datos se leen en formato “ángulo (grados), distancia (en cm)”. El ángulo es la lectura del magnetómetro que corresponde a los grados con respecto al norte magnético.
- El escáner se debe dejar leyenda por varios minutos para que recoja una “nube” de puntos suficientemente densa, para luego detectar la presencia de



árboles en un rodal. El LIDAR hace una lectura de 1, cuando no puede determinar la distancia.

- Los datos recogidos en la terminal serial del Arduino IDE, se puede copiar y pegar en un archivo de texto (sin formato), para luego se procesados (recomendado)
- Del mismo modo, se puede leer los datos de forma visual a través del Lidar Viewer, que se desarrolló como parte de este proyecto y que corre en la plataforma Processing.org.

# LIDAR Viewer

Tomás de Camino Beck, Ph.D.

## Introducción

Lidar viewer es el prototipo de un visualizador de datos del escáner LIDAR. El software se desarrolló en Processing.org una plataforma de código abierto que permite el desarrollo de prototipos de software rápidamente, con códigos escritos en JAVA. Esta plataforma además permite ir modificando el código y distribuyéndolo en forma rápida, pues corre de manera transparente en cualquier versión instalada de Processing.

El LIDAR Viewer, permite la visualización de datos crudos del escáner LIDAR, pero además implementa un algoritmo de agregación que permite identificar estructuras constantes en el escanéo. En el caso de esta aplicación, permite la identificación de árboles. A partir de identificar las estructuras o árboles, es necesario una investigación para establecer la relación entre puntos de borde de estructuras y estimación de diámetro.

El software se conecta al Escáner LIDAR a través del puerto serial con un módulo Bluetooth 2.0 que se encuentra en el escáner. Sin embargo también es posible visualizar los datos a través de cable USB

El código se encuentra en prototipo, y está sujeta a cambios. El código para correr en Processing, se puede descargar en el repositorio GitHub<sup>2</sup> (Se adjunta como ANEXO).

## Recepción de datos

Los datos son recibidos por el LiDAR Viewer, como tuplas de datos,

---

<sup>2</sup>

[https://github.com/FunCR/ProyectoCATIE/blob/master/TreeScanner/Code/LIDAR\\_Viewer/LIDAR\\_Viewer.pde](https://github.com/FunCR/ProyectoCATIE/blob/master/TreeScanner/Code/LIDAR_Viewer/LIDAR_Viewer.pde)

$$\text{datum} = \langle \text{angle}, \text{distance} \rangle$$

Una nube de puntos lo constituye un conjunto grande de puntos ángulo distancia. Mediante esta nube de puntos es posible reconstruir información espacial en 2 dimensiones.

Cada 50 milisegundos el software recibe los datos. estos datos se pueden visualizar crudos (toda la nube de puntos) o como agregados de datos

## Algoritmo

La meta del algoritmo es crear “clusters” de puntos para identificar estructuras permanentes (árboles). La estrategia para lograrlo es agrupar los puntos por proximidad espacial, es decir, se asume que cuando un fuste es detectado, existirá un grupo de puntos muy cercanos, con los bordes marcados por puntos distantes.

Los valores ángulo distancia se convierten a coordenadas cartesianas mediante las siguientes fórmulas,

$$\langle x, y \rangle = \langle \cos(\text{angle}) \times \text{distance}, \sin(\text{angle}) \times \text{distance} \rangle$$

Cada par ordenado  $x,y$  representa un punto  $p_i$ . La distancia entre dos puntos se calcula mediante la distancia euclidiana definida como,

$$|p_1, p_2| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Donde  $p_1$  y  $p_2$  representan dos puntos cualesquiera en la nube de puntos.

Consideremos el siguiente conjunto de datos agrupados,

$$\mathbb{D} = \{\bar{x}, \bar{y}, c\}$$

Donde  $x$  y  $y$  son un promedio de puntos dentro de un rango de distancia predeterminado (centroide), y  $c$  es el conteo de puntos con ese centroide. Las estructuras predominantes o más “sólidas” tendrán un conteo de puntos más alto. Al escanear un bosque, los troncos de árboles serán estructuras predominantes.

El pseudocódigo del algoritmo es el siguiente,

```

Require:  $datum = \langle angle, distance \rangle$ 
Ensure:  $distance > 1$ 
 $x \Leftarrow \cos(angle) \times distance$ 
 $y \Leftarrow \sin(angle) \times distance$ 
 $D \neq \{\}$ 
 $i = 0$ 
while  $p_i \in D$  do
     $p \Leftarrow \langle x, y \rangle$ 
    if  $|p, p_i| < threshold$  then
         $\langle p_i.x, p_i.y \rangle \Leftarrow \langle \bar{x}, \bar{p_i.x}, \bar{y}, \bar{p_i.y} \rangle$ 
         $p_i.c \Leftarrow p_i.c + 1$ 
         $i \Leftarrow i + 1$ 
    else
         $D \Leftarrow D + p$ 
    end if
end while
```

El código completo en JAVA se puede ver en el ANEXO.

## Utilización

1. Se debe instalar Processing. Para hacerlo, descargar e instalar el programa en el sitio de Processing.org<sup>3</sup>
2. Descargar el código del repositorio de GitHub<sup>4</sup>.
3. EN processing, abrir el código de Lidar\_Viever.

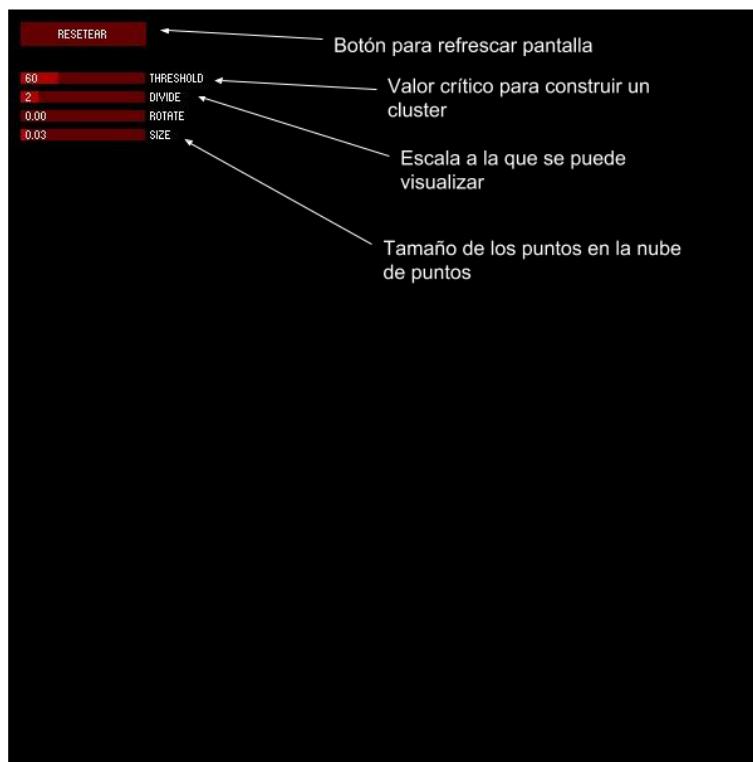
---

<sup>3</sup><https://processing.org/>

<sup>4</sup> [https://github.com/FunCR/ProyectoCATIE/tree/master/TreeScanner/Code/LIDAR\\_Viever](https://github.com/FunCR/ProyectoCATIE/tree/master/TreeScanner/Code/LIDAR_Viever)

4. En la línea `String portname = "/dev/tty.HC-06-DevB";` el valor `"/dev/tty.HC-06-DevB"` indica el nombre del puerto a donde se encuentra el bluetooth. Este se indica una vez que hayan hecho el “pairing” con el bluetooth “HC-06” del escáner Lidar.
5. Una vez que se aseguren que el Bluetooth está registrado en la computadora, presionar el botón de play en processing.

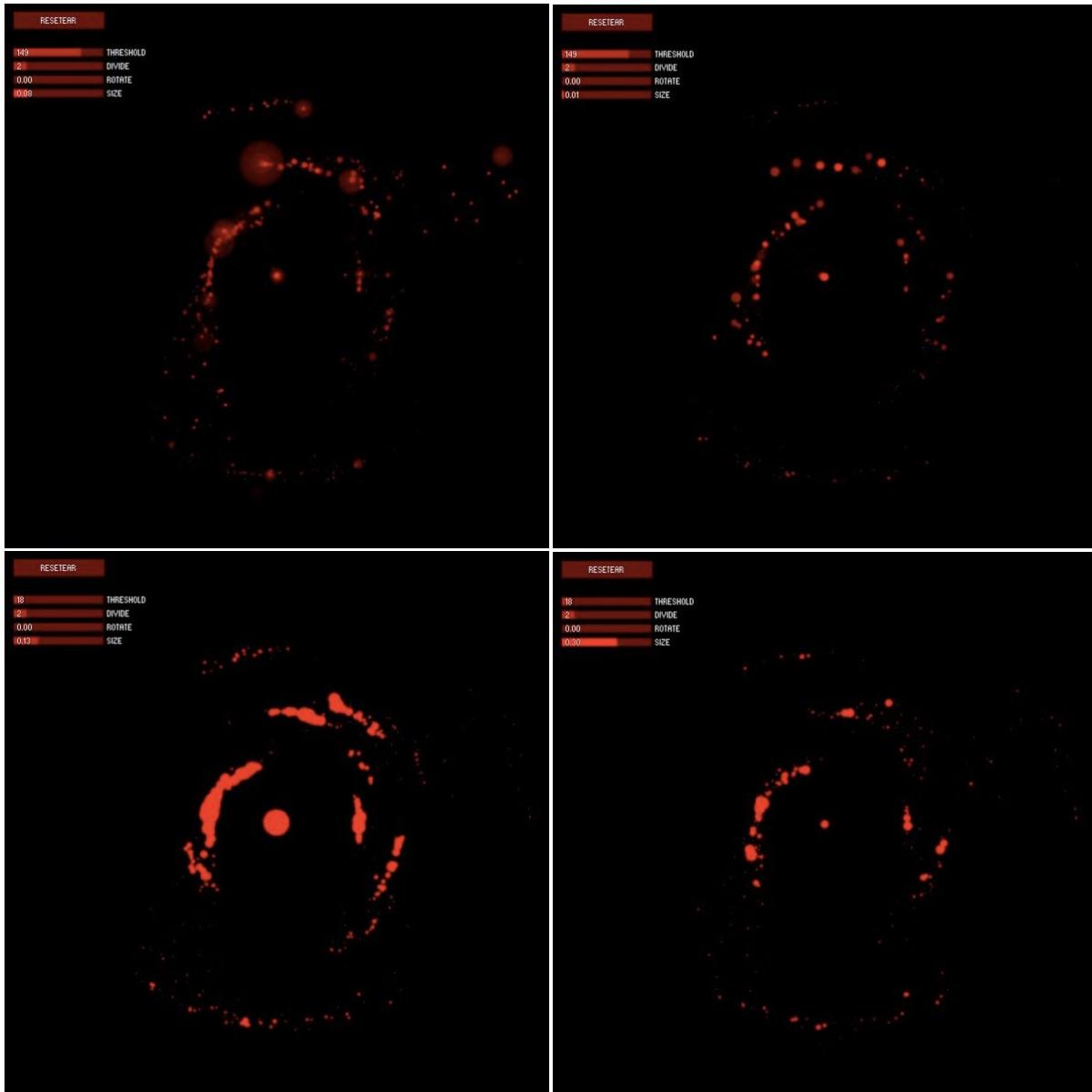
## Interfaz



Una vez hecha la conexión con el USB, el programa comenzará a despegar la nube de puntos. Los puntos de un cluster, se verán como círculos cada vez más grandes. El tamaño de esos círculos de puede cambiar con el slider etiquetado “SIZE”. Si se mueve el slider “THRESHOLD” se puede cambiar la tolerancia de la distancia euclíadiana para formar los clusters.

El Slider “DIVIDE” simplemente es la escala a la cual despliega los puntos, y el botón “RESETEAR” permite refrescar la pantalla.

Luego de algunos minutos la pantalla se verá de la siguiente manera,



Cada “banda” que se forma, representa un árbol. En la imagen se muestran varias configuraciones de formación de cluster.

## Extensiones Necesarias

- Con los puntos agrupados, es necesario extender el análisis para determinar los diámetros. Este trabajo requiere un ensayo experimental tanto en campo, como en medidas diamétricas exactas para calibrara el algoritmo. Con medidas diamétricas exactas, nos referimos a estructuras cilíndricas de diámetro exacto que son escaneadas por el escáner LIDAR, para determinar los diámetros y validar el algoritmo de reconocimiento de troncos, así como extenderlo para la medición de diámetros y cálculo de área basal y de diámetros.
- La forma de exportar datos todavía no es directa, para ellos hay que grabar los datos en la consola terminal y exportar a archivo separado por comas

```

//Código del Escaner LIDAR
//Creado por Tomas de Camino Beck

#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_LSM303_U.h>
#include <Adafruit_L3GD20_U.h>
#include <Adafruit_9DOF.h>
#include <LIDARLite.h>
#include <Servo.h>

#define ACCEL 150
#define SERVOPIN 13

/* Assign a unique ID to the sensors */
Adafruit_9DOF          dof    = Adafruit_9DOF();
Adafruit_LSM303_Accel_Unified accel = Adafruit_LSM303_Accel_Unified(30301);
Adafruit_LSM303_Mag_Unified   mag   = Adafruit_LSM303_Mag_Unified(30302);

/* Update this with the correct SLP for accurate altitude measurements */
float seaLevelPressure = SENSORS_PRESSURE_SEALEVELHPA;

Servo servo;
LIDARLite myLidarLite;

//****************************************************************************
/*! @brief Initialises all the sensors used by this example
*/
//****************************************************************************
void initSensors()
{
    if (!accel.begin())
    {
        /* There was a problem detecting the LSM303 ... check your connections */
        Serial.println(F("Ooops, no LSM303 detected ... Check your wiring!"));
        while (1);
    }
    if (!mag.begin())
    {
        /* There was a problem detecting the LSM303 ... check your connections */
        Serial.println("Ooops, no LSM303 detected ... Check your wiring!");
        while (1);
    }
}

```

```

}

//****************************************************************************
/*!

*/
//****************************************************************************
void setup(void)
{
    Serial.begin(9600);
    Serial.println(F("Adafruit 9 DOF Pitch/Roll/Heading Example")); Serial.
println("");

/* Initialise the sensors */
initSensors();
//setMotors();
myLidarLite.begin(0, true);
myLidarLite.configure(5);
servo.attach(SERVOPIN);
pinMode(5, OUTPUT);
pinMode(10, OUTPUT);
digitalWrite(5, HIGH);
digitalWrite(10, LOW);

}

//****************************************************************************
/*!
@brief Constantly check the roll/pitch/heading/altitude/temperature
*/
//****************************************************************************
void loop(void)
{
    servo.write(0);
    sensors_event_t mag_event;
    sensors_vec_t orientation;

    /* Calculate the heading using the magnetometer */
    mag.getEvent(&mag_event);

    if (dof.magGetOrientation(SENSOR_AXIS_Z, &mag_event, &orientation))
    {
        /* 'orientation' should have valid .heading data now */
        Serial.print(orientation.heading);
    }
}

```

```
Serial.print(F(", "));  
}  
  
Serial.println(myLidarLite.distance());  
delay(100);  
}
```

## LIDAR\_Viewer

//creado por Tomas de Camino Beck

```
import controlP5.*;
import processing.serial.*;
```

```
Serial port;
String data;
float[] val = new float[2];
float x, y=0;
```

```
int threshold = 60;
int divide = 2;
float rotate=0;
float size=0.03;
```

```
ControlP5 cp5;
```

```
ArrayList<PVector> feature = new ArrayList<PVector>();
```

```
void setup() {
```

```
cp5 = new ControlP5(this);
```

```
//controles de menu
```

```
cp5.addButton("Resetear")
.setValue(0)
.setPosition(10, 10)
.setColorBackground(color(100, 0, 0))
.setSize(100, 19)
;
```

```
cp5.addSlider("threshold")
```

```
.setPosition(10, 50)
.setColorForeground(color(180, 0, 0))
.setColorBackground(color(100, 0, 0))
.setColorActive(color(255, 0, 0))
.setRange(0, 200)
;
```

```
cp5.addSlider("divide")
```

```
.setPosition(10, 65)
.setColorForeground(color(180, 0, 0))
.setColorBackground(color(100, 0, 0))
.setColorActive(color(255, 0, 0))
.setRange(1, 8)
;
```

```
cp5.addSlider("rotate")
.setPosition(10, 80)
.setColorForeground(color(180, 0, 0))
.setColorBackground(color(100, 0, 0))
.setColorActive(color(255, 0, 0))
.setRange(0, 2*PI)
;
```

```
cp5.addSlider("size")
.setPosition(10, 95)
.setColorForeground(color(180, 0, 0))
.setColorBackground(color(100, 0, 0))
.setColorActive(color(255, 0, 0))
.setRange(0.001, 0.5)
;
```

```
//fin controles de menu
```

```
println(Serial.list());
//Seleccionar puerto de la lista
port = new Serial(this, "/dev/tty.HC-06-DevB", 9600);
port.bufferUntil('\'\n\'');
size(600, 600);
background(0);
}
```

```
void draw() {
//background(0);
fill(0);
//text(val[0],10,10);
pushMatrix();
translate(width/2, height/2);
rotate(val[0]);
float px = x;
float py = y;
x = cos(radians(val[0]))*val[1];
y = sin(radians(val[0]))*val[1];

stroke(255, 0, 0, 5);
//strokeWeight(3);
if (x!=0) {
  searchFeature(x, y, threshold);
}
drawPoints();
popMatrix();
//if (mousePressed) {
```

```

// saveFrame("frames/####.jpg");
//}

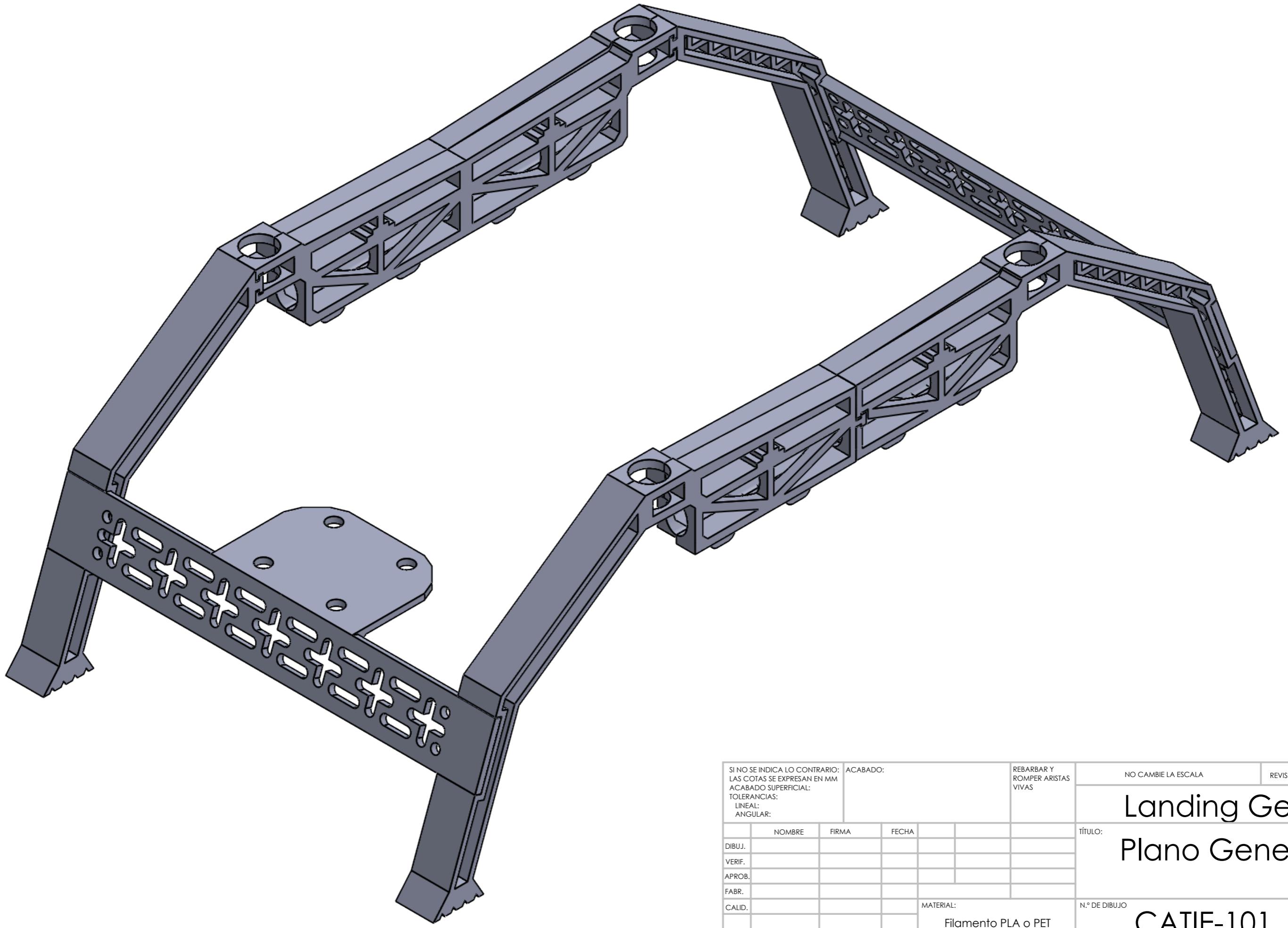
}

void serialEvent(Serial port) {
  data = port.readString();
  String[] list = split(data, ',');
  val[0]=float(list[0]);
  val[1]=float(list[1]);
}

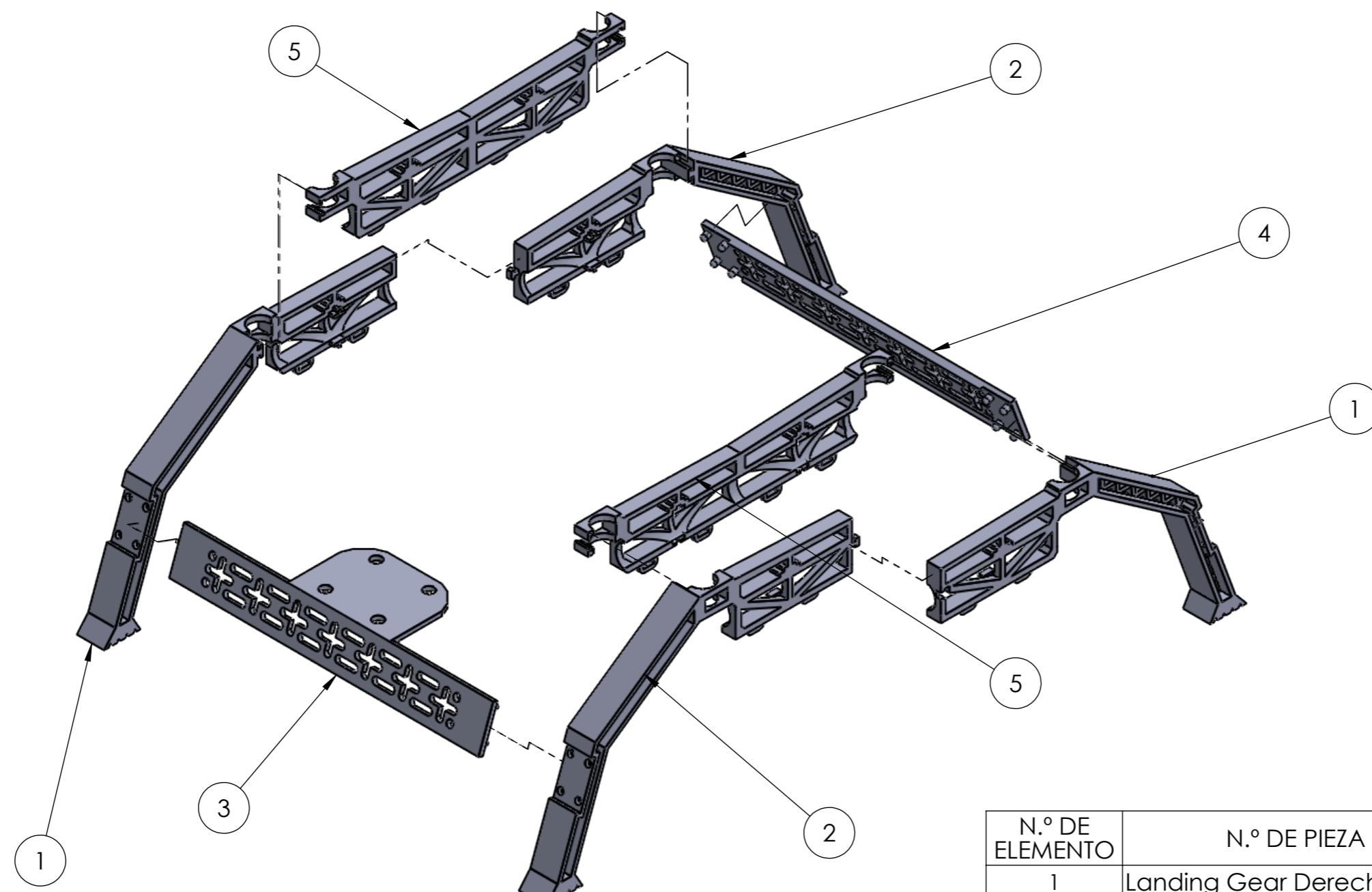
public void Resetear(int theValue) {
  println("reset");
  background(0);
}

public int searchFeature(float x, float y, float th) {
  //busqueda por distancia
  int i=0;
  PVector f;
  boolean found = false;
  while (i<feature.size() && !found) {
    f = feature.get(i);
    float d =dist(x, y, f.x, f.y);
    if (d  |
```





SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:			ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
DIBUJ.	NOMBRE	FIRMA	FECHA		TÍTULO:	
VERIF.					Landing Gear	
APROB.					Plano General	
FABR.						
CALID.				MATERIAL:	N.º DE DIBUJO	
				Filamento PLA o PET	CATIE-101	A3
				PESO:	ESCALA:1:1	HOJA 1 DE 1



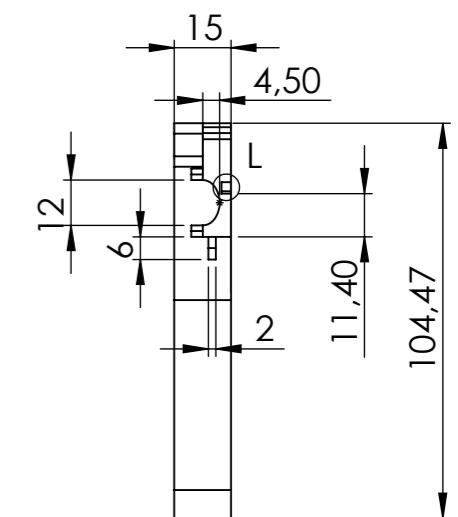
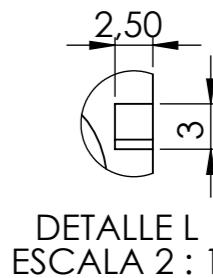
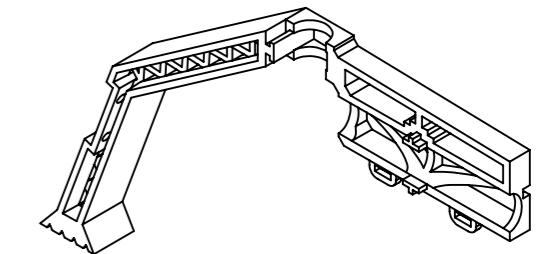
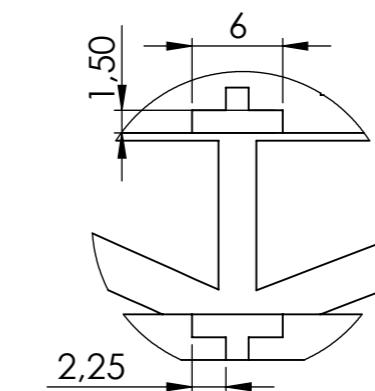
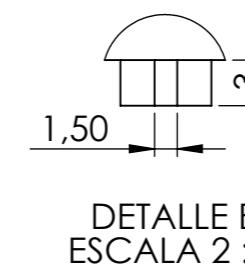
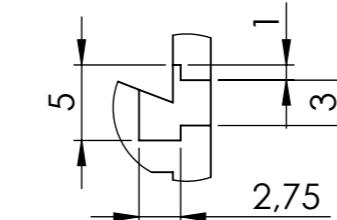
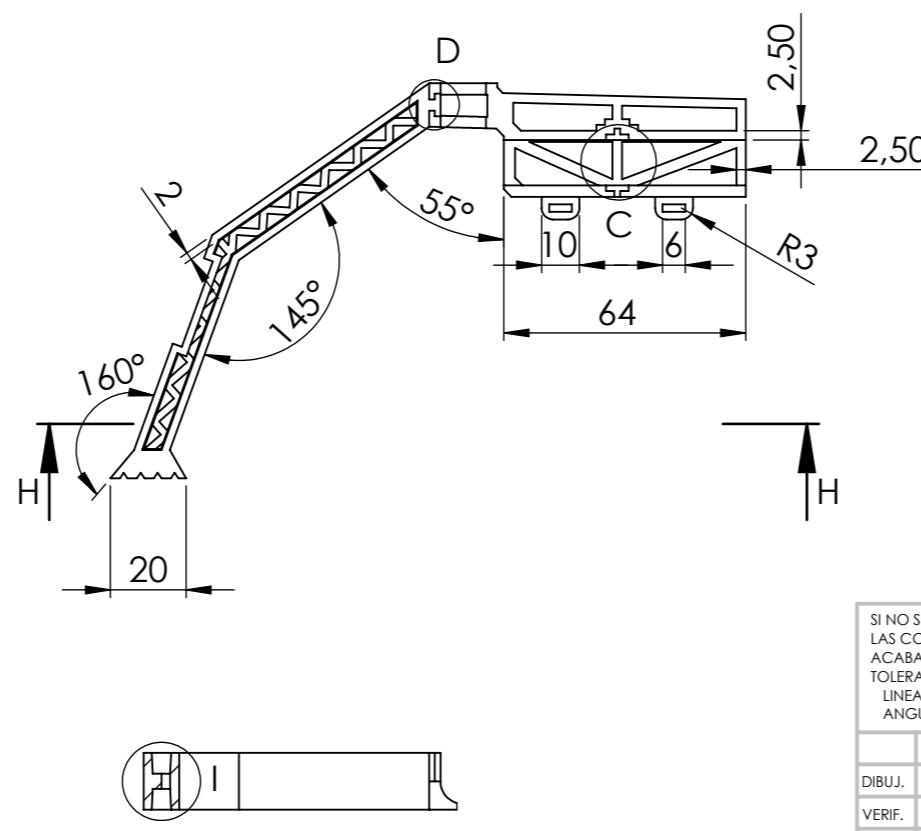
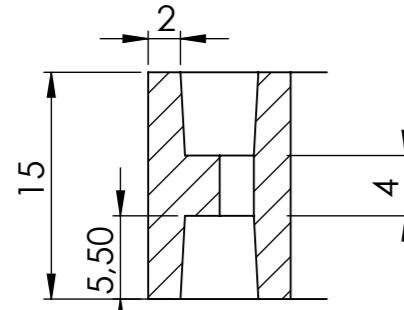
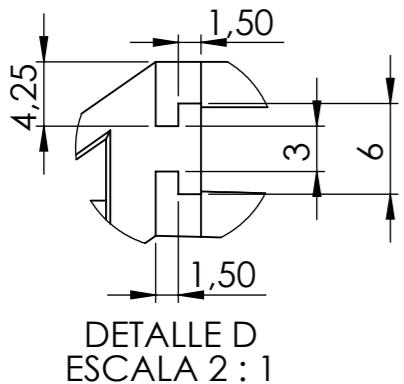
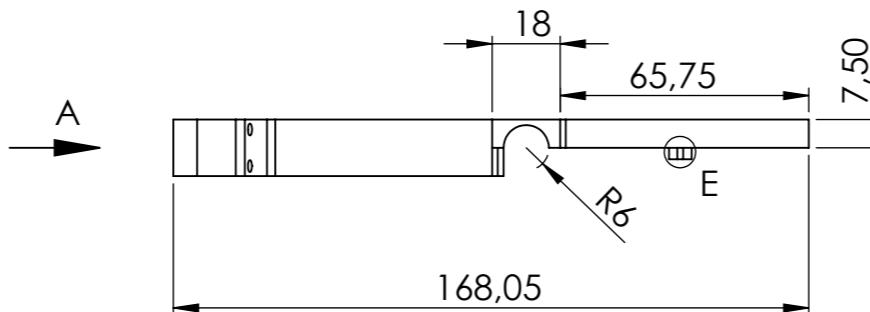
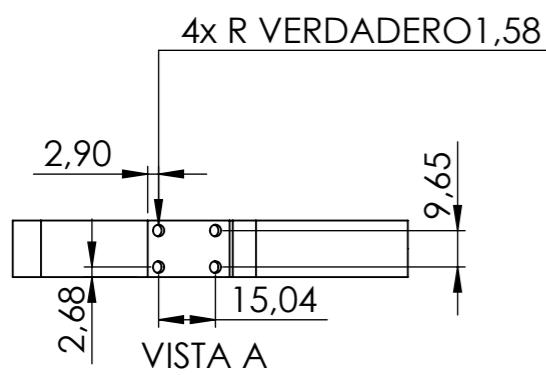
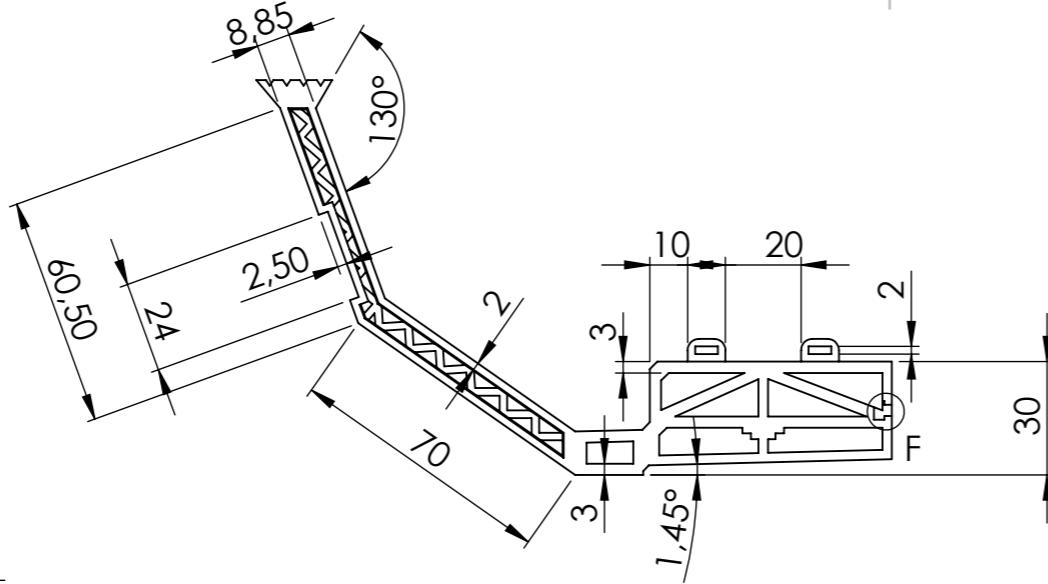
N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANTIDAD
1	Landing Gear Derecho		2
2	Landing Gear Izquierdo		2
3	Snap para Lidar		1
4	Snap		1
5	Landing Gear Amarre		2

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:			ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA		REVISIÓN
DIBUJ.	NOMBRE	FIRMA	FECHA				
VERIF.							
APROB.							
FABR.							
CALID.							
	MATERIAL:						
				PESO:			
					ESCALA:1:2		
						HOJA 1 DE 1	
							CATIE-102
							A3

**Landing Gear**  
**Plano explosivo**

**Nota:**

Esta pieza es en espejo  
de la pieza Landing  
Gear Izquierdo



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: 0,25 mm ANGULAR: 1°			ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	
DIBUJ.	NOMBRE	FIRMA	FECHA		TÍTULO:	REVISIÓN
VERIF.					Landing Gear	
APROB.					Landing Gear	
FABR.					Derecho	
CALID.					CATIE-103	A3
			MATERIAL: Filamento de PLA o PET	N.º DE DIBUJO	ESCALA: 1:2	HOJA 1 DE 1
			PESO:			

8

7

6

5

4

3

2

1

F

F

E

E

D

D

C

C

B

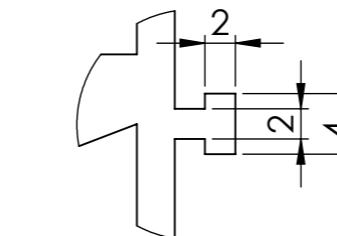
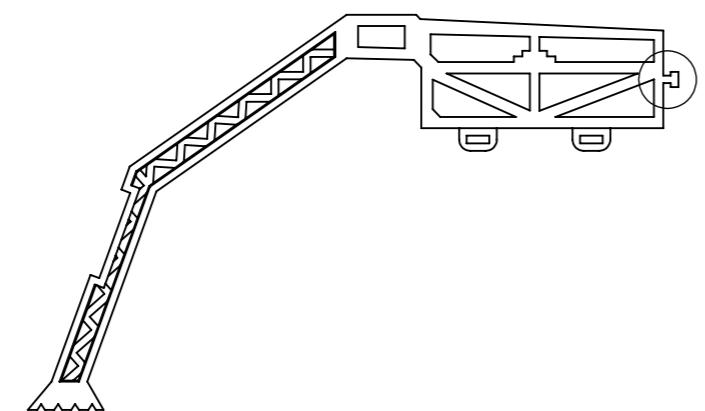
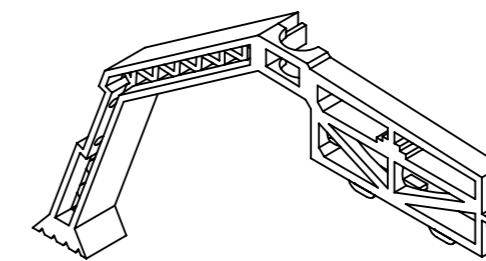
B

A

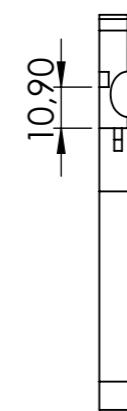
A



DETALLE K  
ESCALA 2 : 1



DETALLE J  
ESCALA 2 : 1



### Nota:

La construcción de esta pieza es en espejo de la pieza Landing Gear Derecho.

Todas las cotas son idénticas a excepción de las presentadas en esta lámina.

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: 0.25 mm ANGULAR: 1°		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA		REVISIÓN
DIBUJ.	NOMBRE	FIRMA	FECHA				
VERIF.							
APROB.							
FABR.							
CALID.				MATERIAL: Filamento de PLA o PET			
				PESO:		ESCALA: 1:2	
							HOJA 1 DE 1
							A3

Landing Gear

Landing Gear  
Izquierdo

CATIE-104

8

7

6

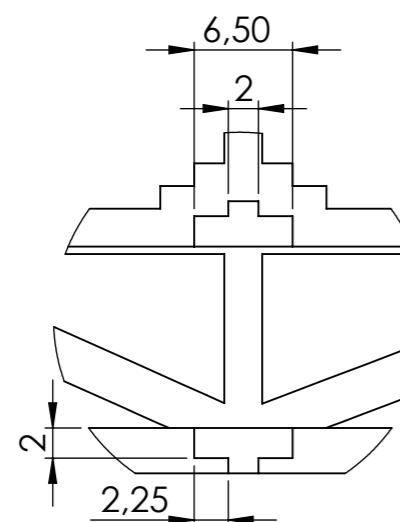
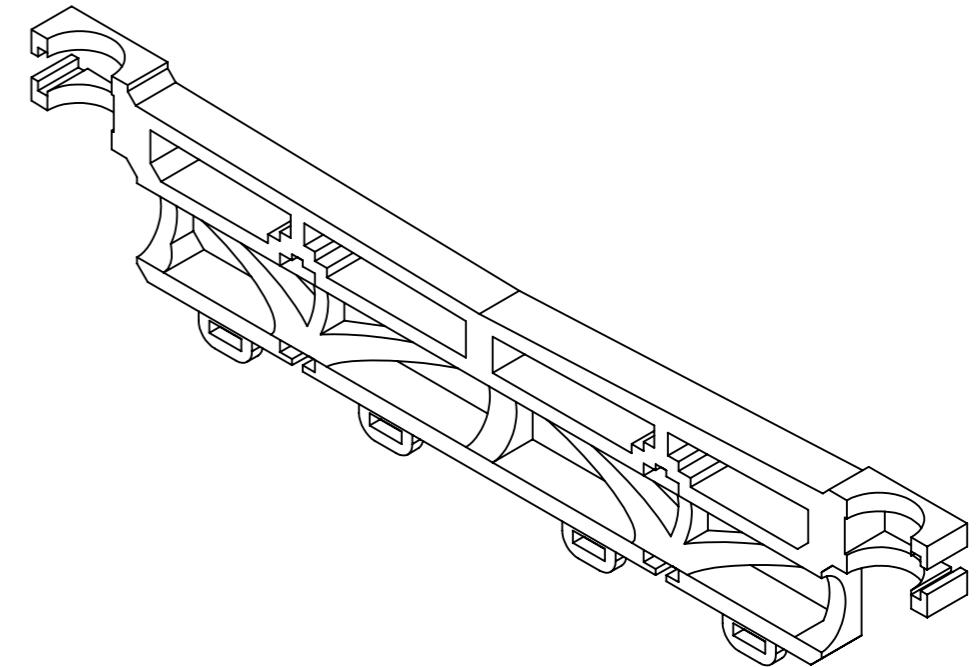
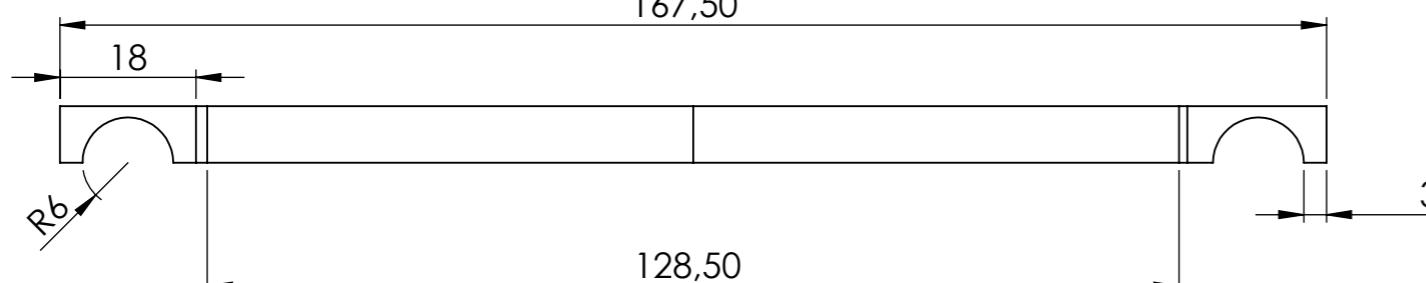
5

4

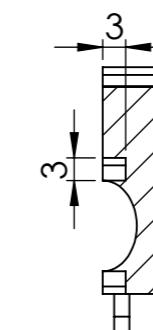
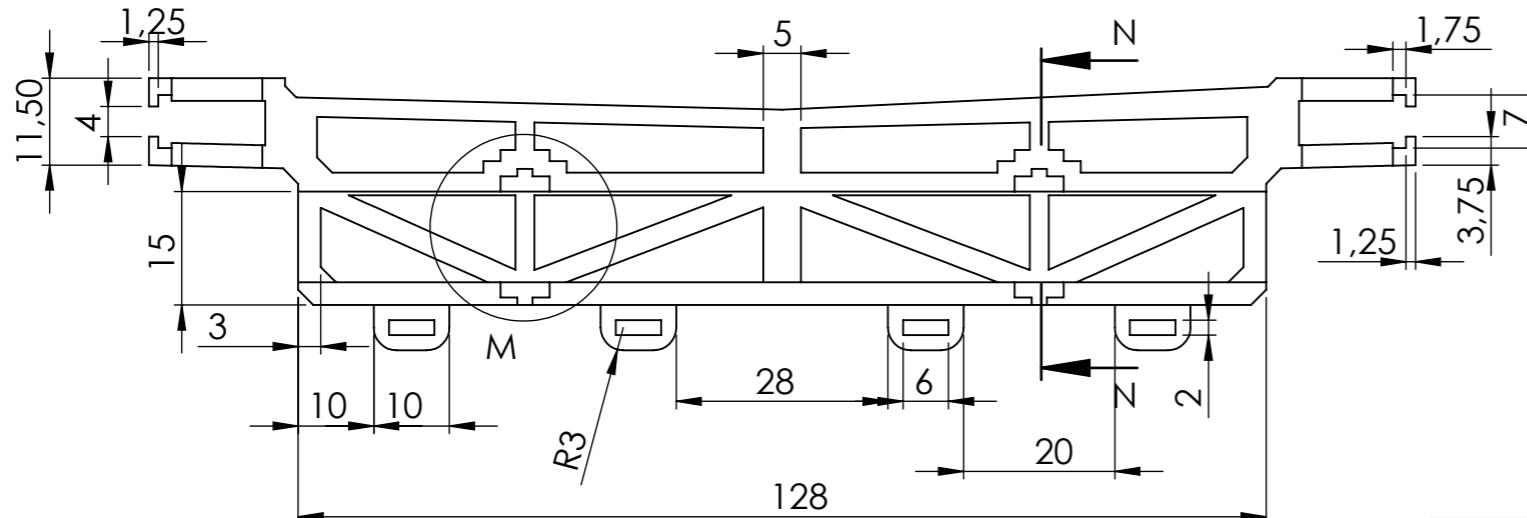
3

2

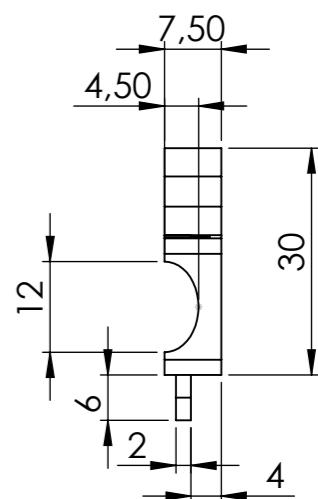
1



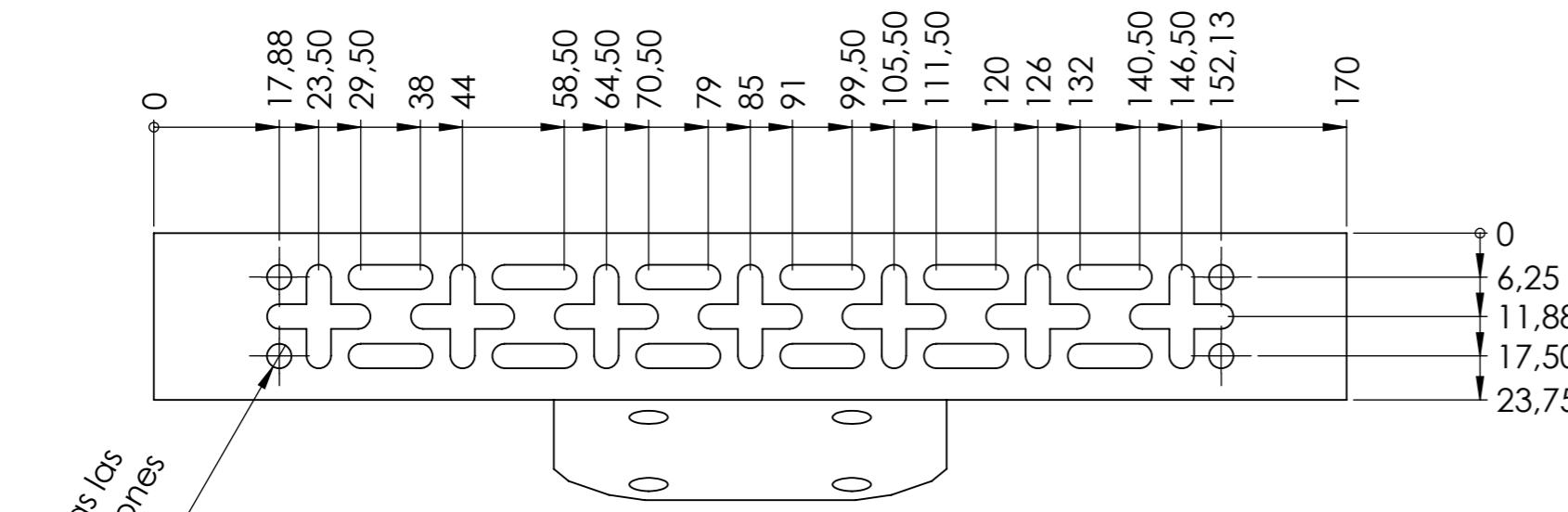
DETALLE M  
ESCALA 2 : 1



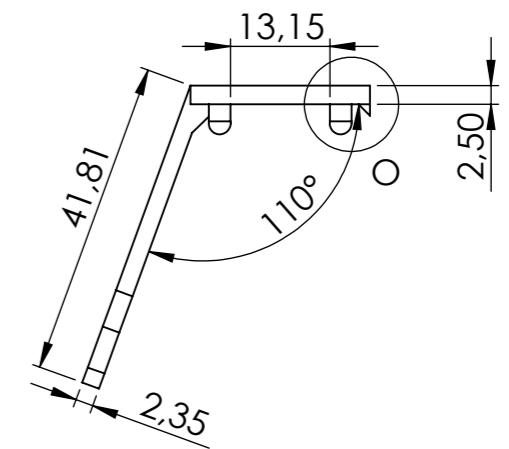
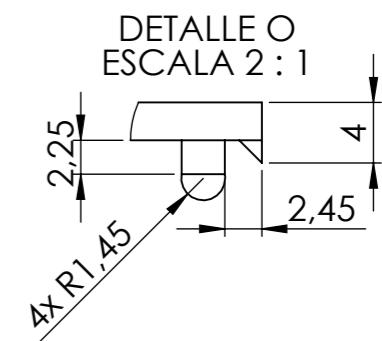
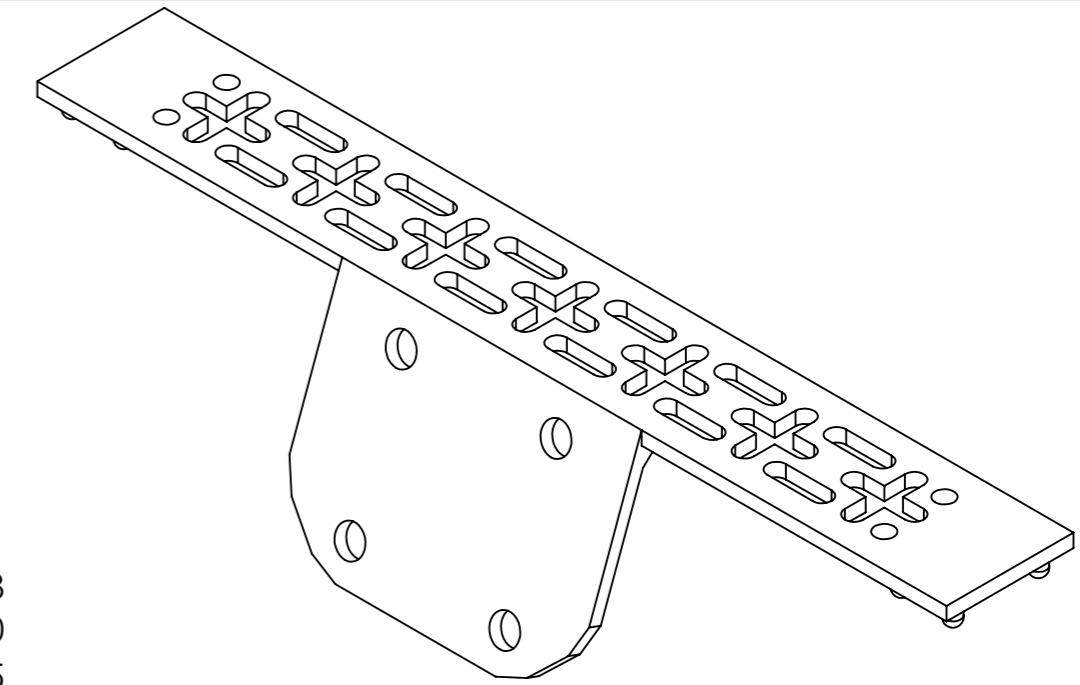
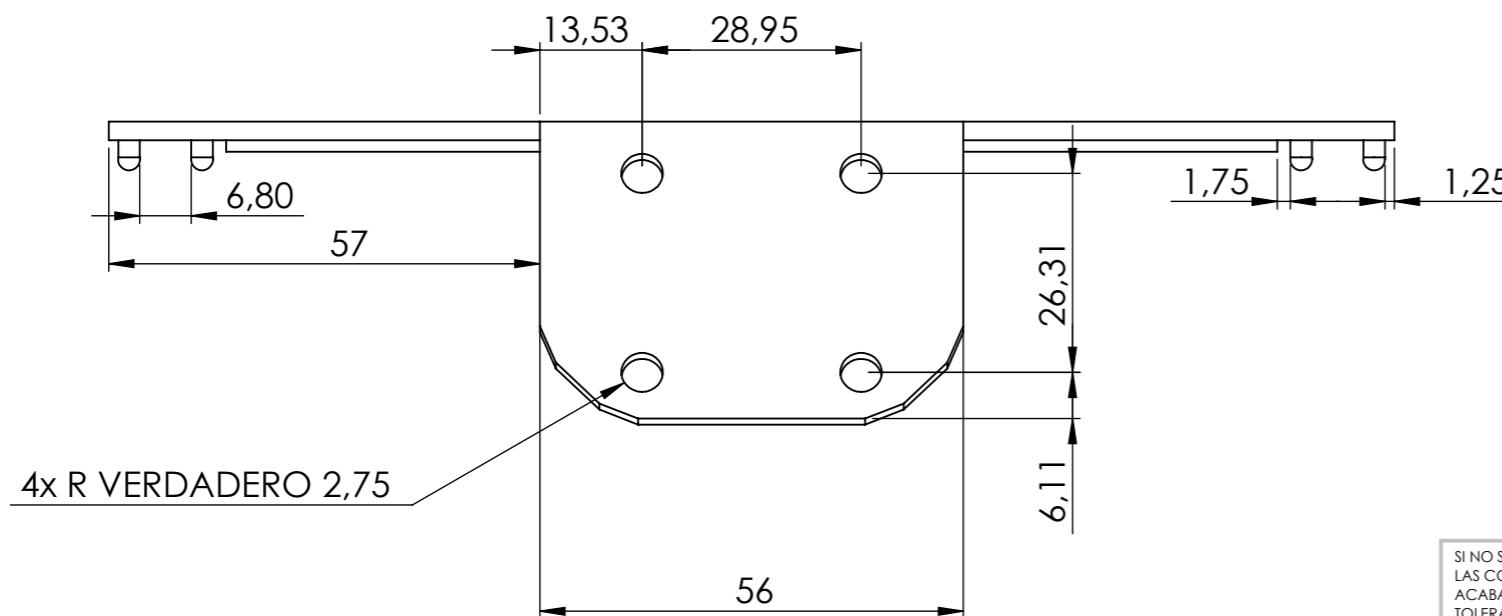
## SECCIÓN N-N



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: 0.25 mm ANGULAR: 1°		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
<h1>Landing Gear</h1> <p>TÍTULO:</p> <h2>Landing Gear Amarre</h2>						
DIBUJ.	NOMBRE	FIRMA	FECHA			
VERIF.						
APROB.						
FABR.						
CALID.	MATERIAL: Filamento de PLA o PET			N.º DE DIBUJO	CATIE-105	
	PESO:			ESCALA:1:1	HOJA 1 DE 1	



En todas las  
perforaciones  
Ø3,50



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: 0,25 mm ANGULAR: 1°		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
DIBUJ.	NOMBRE	FIRMA	FECHA			
VERIF.						
APROB.						
FABR.						
CALID.						
		MATERIAL: Filamento de PLA o PET		N.º DE DIBUJO	CATIE-106	
		PESO:		ESCALA: 1:1	A3	
		HOJA 1 DE 1				

CATIE-106

A3

1

Snap para Lidar

Landing Gear

8 7 6 5 4 3 2 1

F

F

E

E

D

D

C

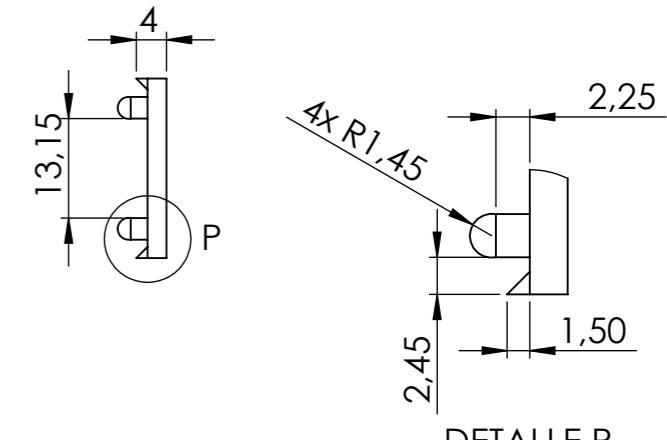
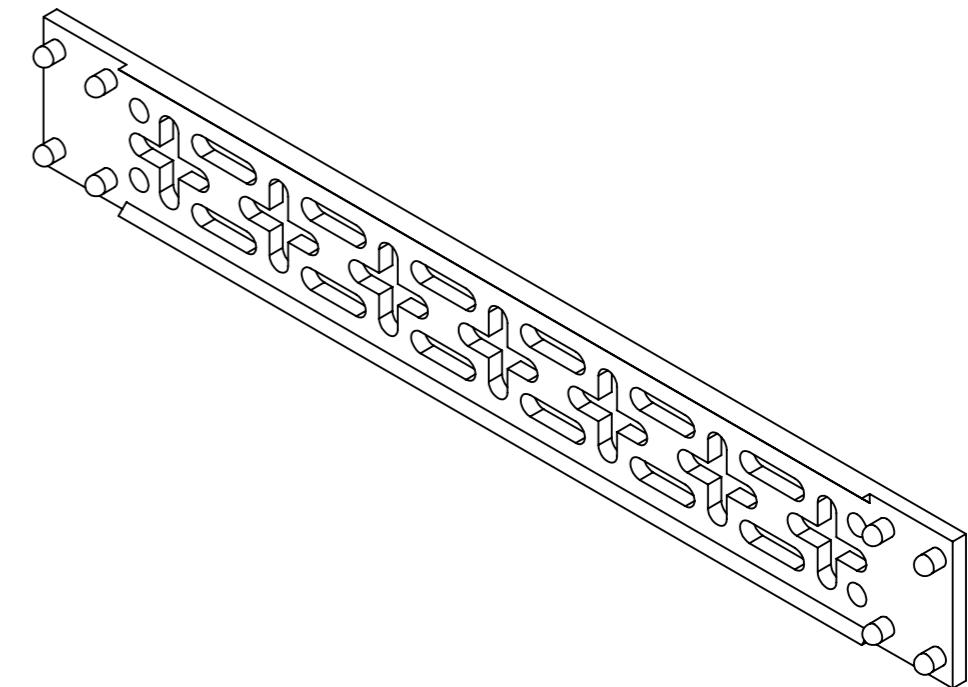
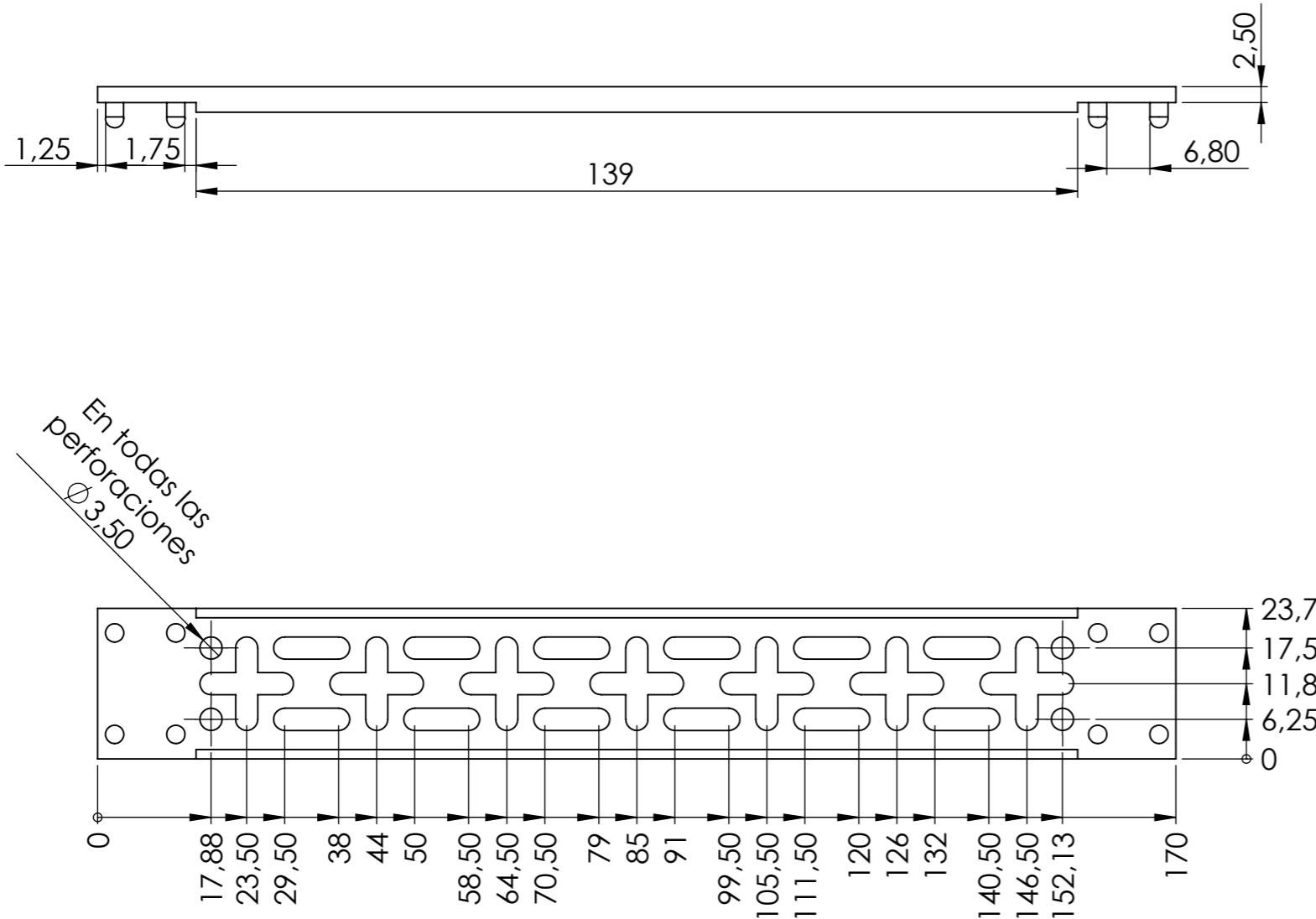
C

B

B

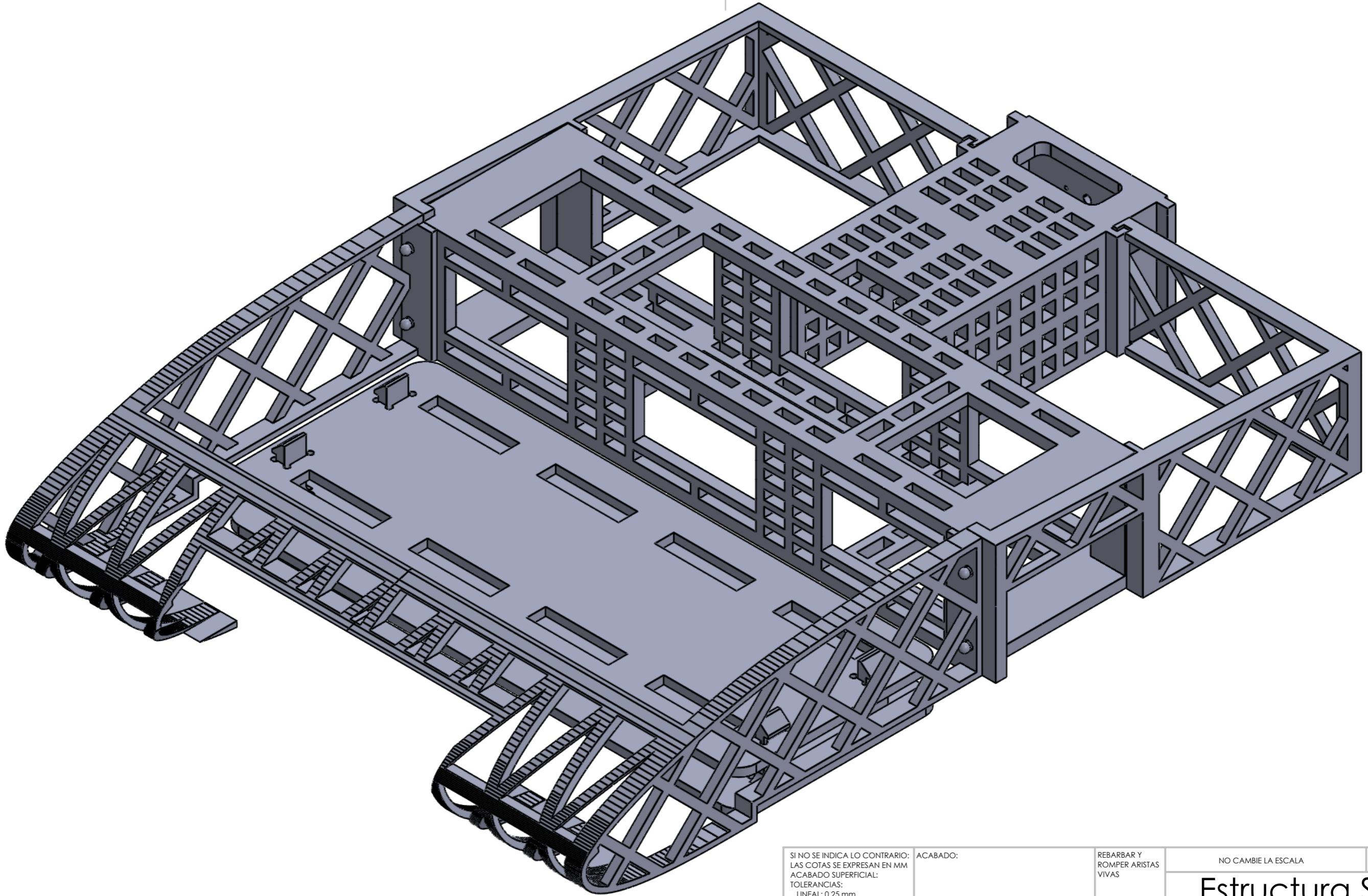
A

A



DETALLE P  
ESCALA 2 : 1

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: 0,25mm ANGULAR: 1°			ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
DIBUJ.	NOMBRE	FIRMA	FECHA		TÍTULO:	Landing Gear
VERIF.						
APROB.						
FABR.						
CALID.						
				MATERIAL: Filamento de PLA o PET	N.º DE DIBUJO	CATIE-107
				PESO:	ESCALA:1:1	A3
					HOJA 1 DE 1	



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: 0.25 mm ANGULAR: 1°		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
DIBUJ.	NOMBRE	FIRMA	FECHA		TÍTULO: <b>Estructura Spear</b>	
VERIF.						
APROB.						
FABR.						
CALID.			MATERIAL: Filamento de PLA o PET	N.º DE DIBUJO	CATIE-201	A3
			PESO:	ESCALA:1:1	HOJA 1 DE 1	

8 7 6 5 4 3 2 1

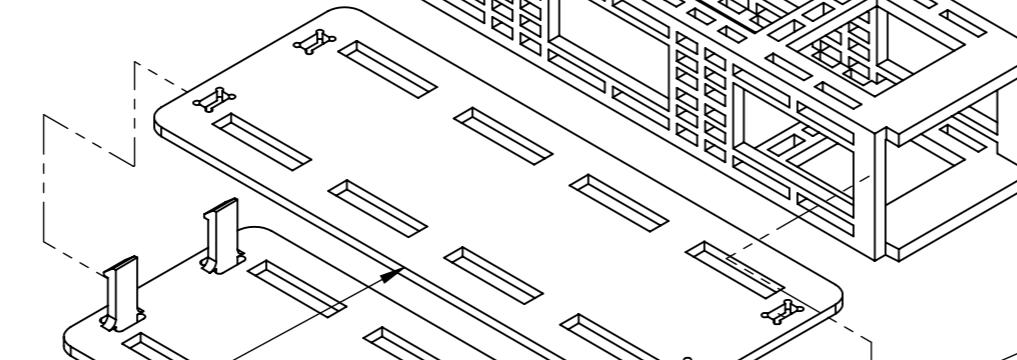
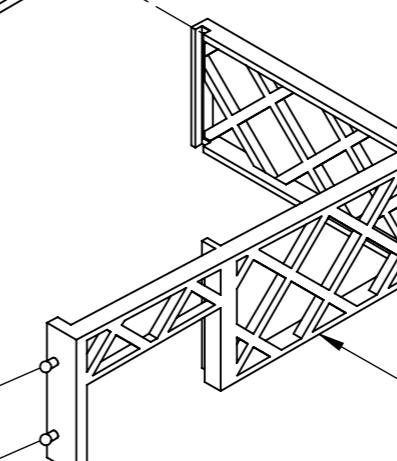
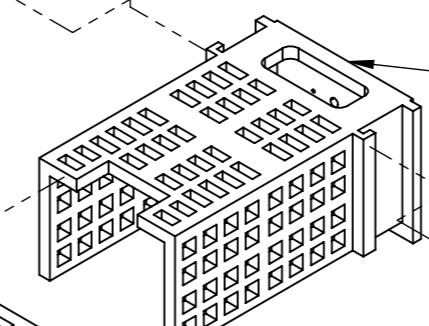
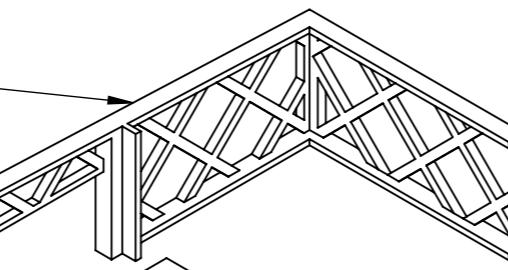
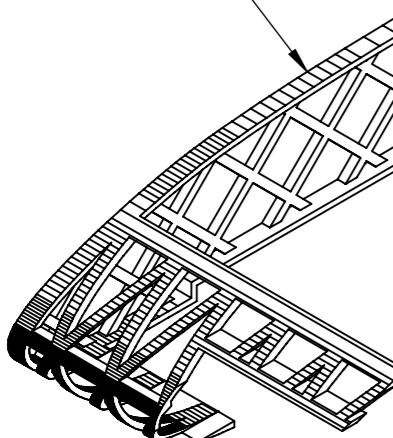
F

5

6

9

10



12

11

8

D

C

B

A

F

E

D

C

B

A

N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANTIDAD
5	Chasis frente derecho		1
6	Chasis atrás derecho		1
7	Chasis atrás izquierdo		1
8	Chasis frente izquierdo		1
9	Base central		1
10	Case Motor		1
11	Base snap batería		1
12	Soporte Baterías		1

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:			ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA		REVISIÓN
DIBUJ.	NOMBRE	FIRMA	FECHA				
VERIF.							
APROB.							
FABR.							
CALID.							
MATERIAL:				PESO:		N.º DE DIBUJO	
						ESCALA: 1:2	
						HOJA 1 DE 1	
						CATIE-202 A3	

Estructura Spear

Estructura Spear

Plano Explosivo

CATIE-202

A3

8

7

6

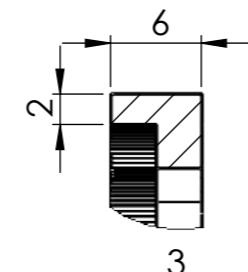
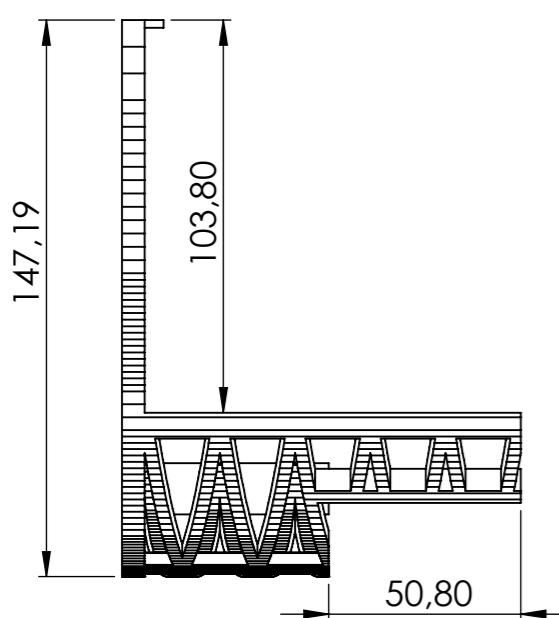
5

4

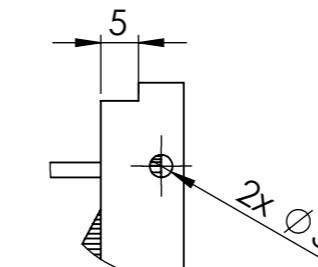
3

2

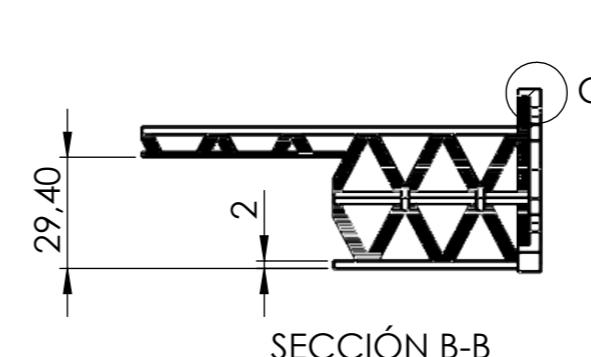
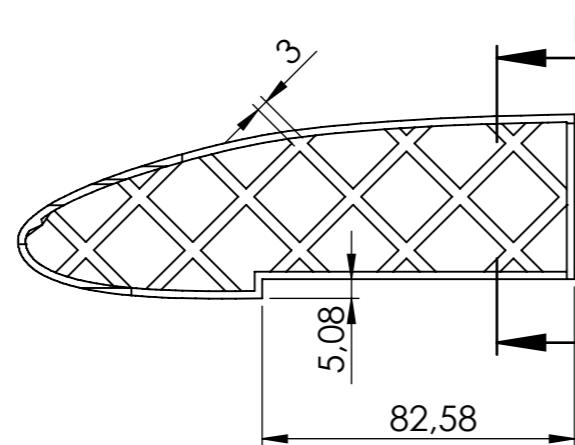
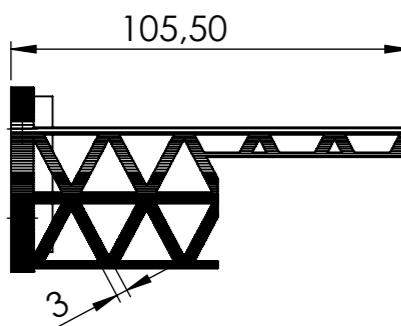
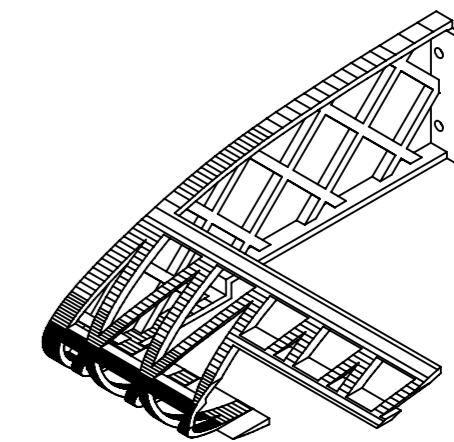
1



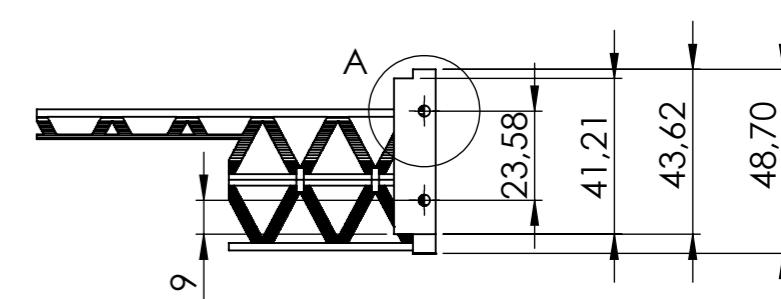
DETALLE C  
ESCALA 2 : 1



DETALLE A  
ESCALA 1 : 1



SECCIÓN B-B



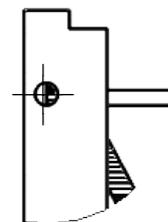
### Notas:

1. Esta pieza está hecha en espejo con la pieza Chasis Frente Izquierdo, todas las cotas son idénticas.
2. Esta pieza está diseñada a partir de un modelo CAD ya existente, producto de otro diseño. Para mayor referencia sobre diámetros de curvas y cotas no indicadas en este plano, consultar los archivos CAD que se adjuntan con este plano y llevan el mismo nombre.

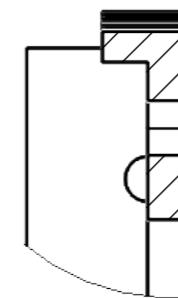
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: 0,25 mm ANGULAR: 1°		ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
DIBUJ.	NOMBRE	FIRMA	FECHA	TÍTULO:	Estructura Spear
VERIF.				Chasis Frente	
APROB.				Derecho	
FABR.				CATIE-203	A3
CALID.				MATERIAL: Filamento de PLA o PET	
				N.º DE DIBUJO	
				ESCALA: 1:2	
				HOJA 1 DE 1	
PESO:					

8 7 6 5 4 3 2 1

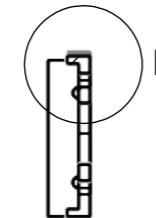
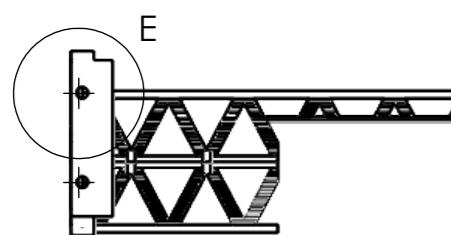
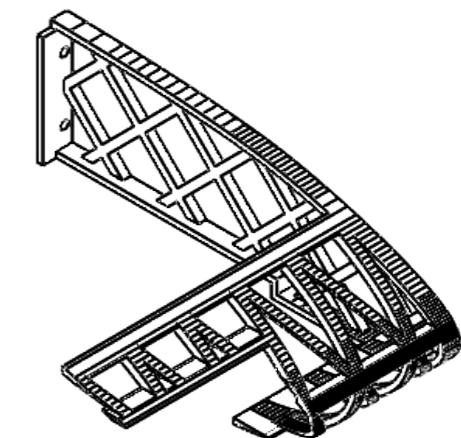
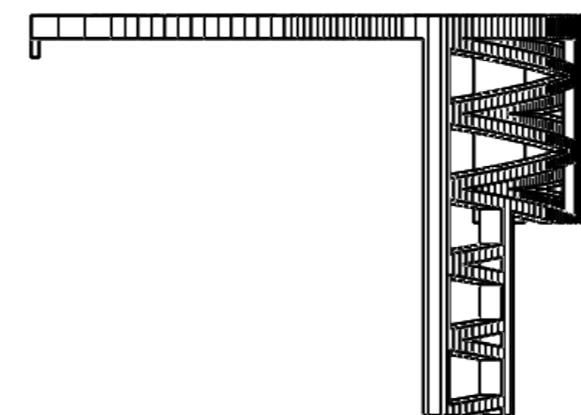
F



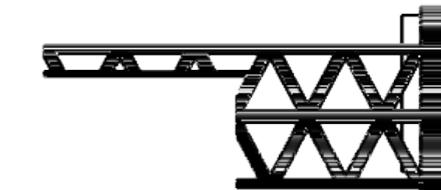
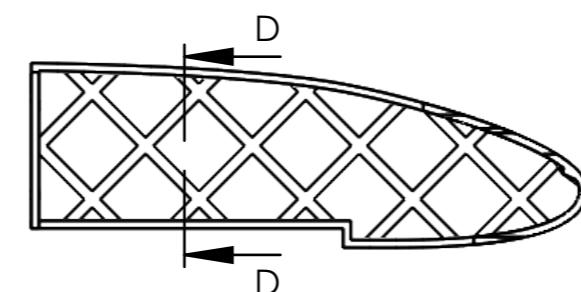
DETALLE E  
ESCALA 1 : 1



DETALLE F  
ESCALA 2 : 1



SECCIÓN D-D



### Notas:

1. Esta pieza está hecha en espejo con la pieza Chasis Frente Derecho.
2. En esta lámina se presentan las vistas solo para referencia.
3. Consultar cotas de la Chasis Frente Derecho para construcción de la pieza de esta lámina.
4. Esta pieza está diseñada a partir de un modelo CAD ya existente, producto de otro diseño. Para mayor referencia sobre diámetros de curvas y cotas no indicadas en este plano, consultar los archivos CAD que se adjuntan con este plano y llevan el mismo nombre.

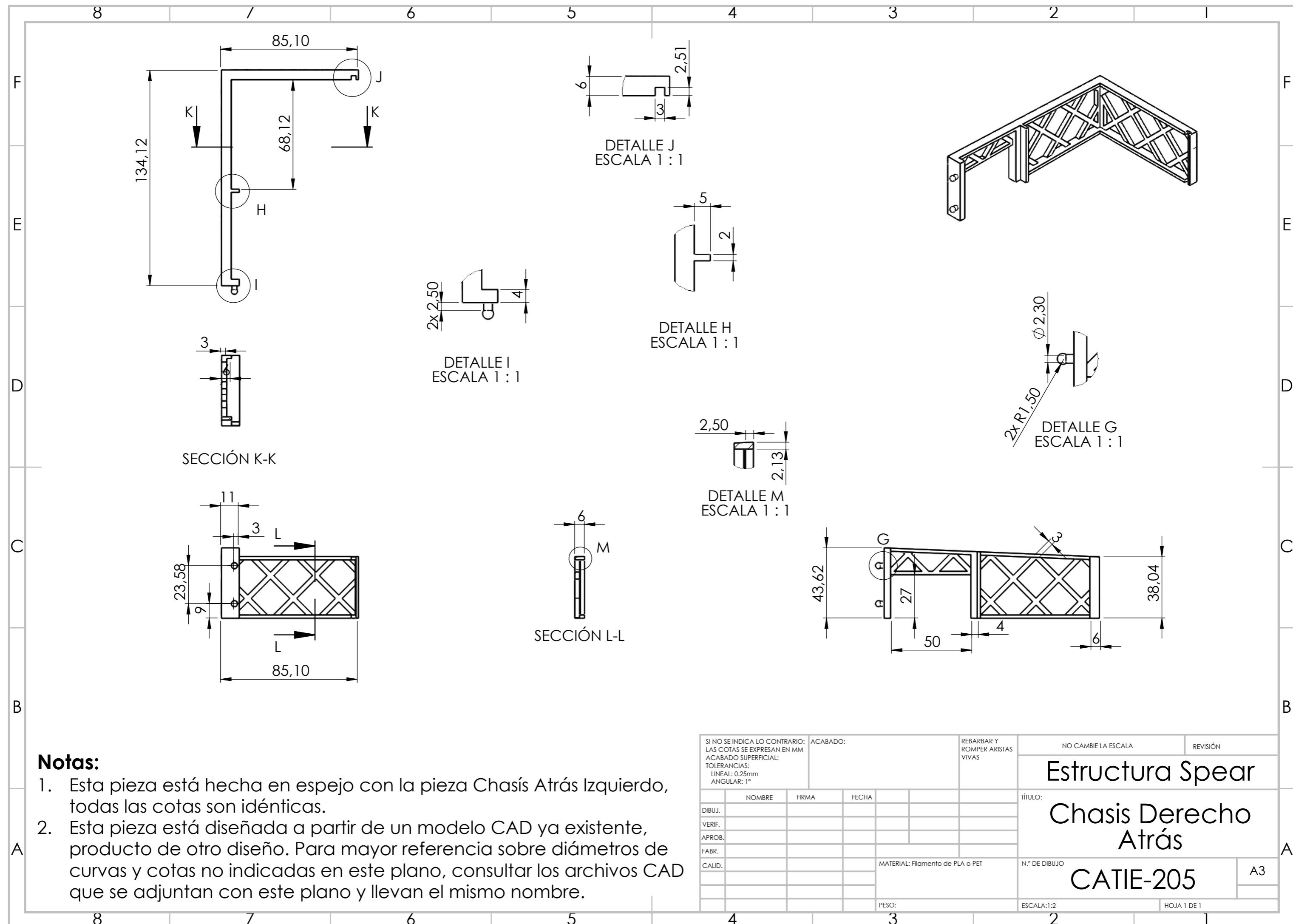
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: 0.25 mm ANGULAR: 1°	ACABADO:    	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA
DIBUJ.	NOMBRE	FIRMA	FECHA
VERIF.			
APROB.			
FABR.			
CALID.			
MATERIAL: Filamento PLA o PET			N.º DE DIBUJO
			A3
PESO:			ESCALA:1:2
			HOJA 1 DE 1

Estructura Spear

Chasis Frente  
Izquierdo

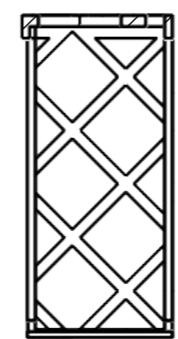
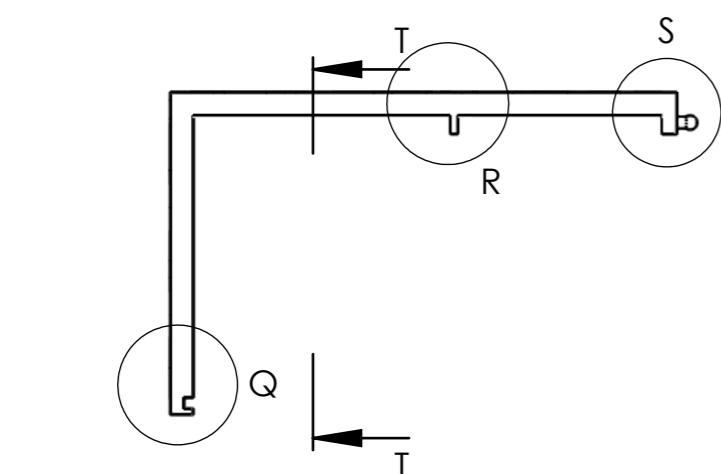
CATIE-204

8 7 6 5 4 3 2 1

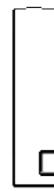
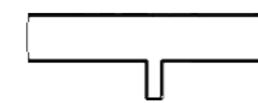
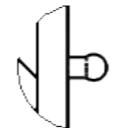
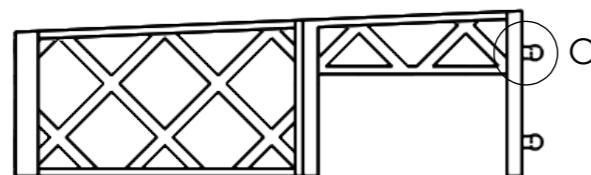
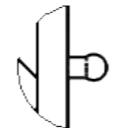
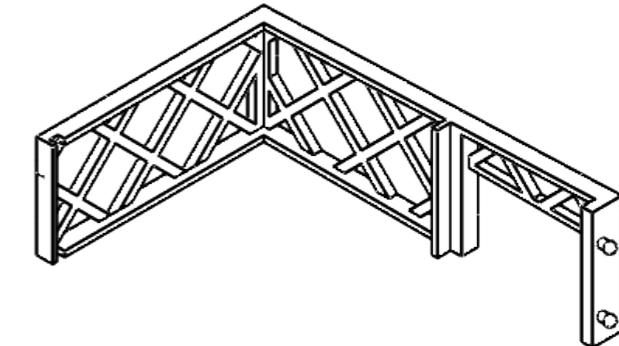
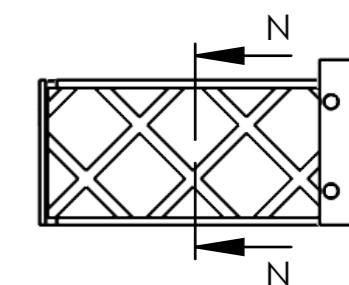


8 7 6 5 4 3 2 1

F



SECCIÓN T-T

DETALLE Q  
ESCALA 1 : 1DETALLE R  
ESCALA 1 : 1DETALLE O  
ESCALA 1 : 1DETALLE S  
ESCALA 1 : 1DETALLE P  
ESCALA 1 : 1

SECCIÓN N-N

B

**Notas:**

1. Esta pieza está hecha en espejo con la pieza Chasis Atrás Derecho.
2. En esta lámina se presentan las vistas solo para referencia.
3. Consultar cotas de la Chasis Frente Derecho para construcción de la pieza de esta lámina.
4. Esta pieza está diseñada a partir de un modelo CAD ya existente, producto de otro diseño. Para mayor referencia sobre diámetros de curvas y cotas no indicadas en este plano, consultar los archivos CAD que se adjuntan con este plano y llevan el mismo nombre.

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: 0.25 ANGULAR: 1°		ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
DIBUJ.	NOMBRE	FIRMA	FECHA	TÍTULO:	Estructura Spear
VERIF.				Chasis Atrás	Izquierdo
APROB.				CATIE-206	A3
FABR.					
CALID.					
			MATERIAL: Filamento de PLA o PET	N.º DE DIBUJO	
				ESCALA: 1:2	HOJA 1 DE 1
			PESO:		

8

7

6

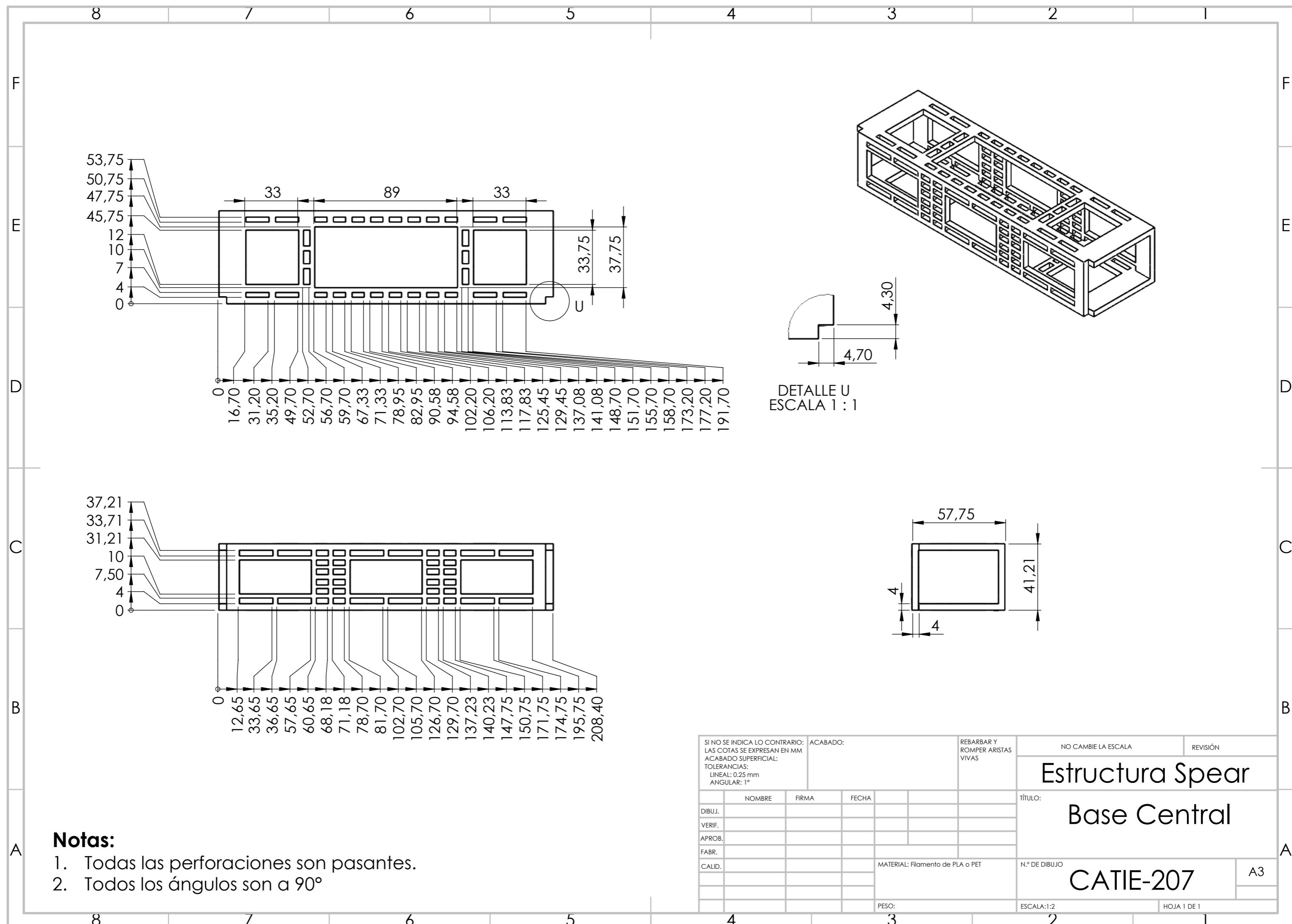
5

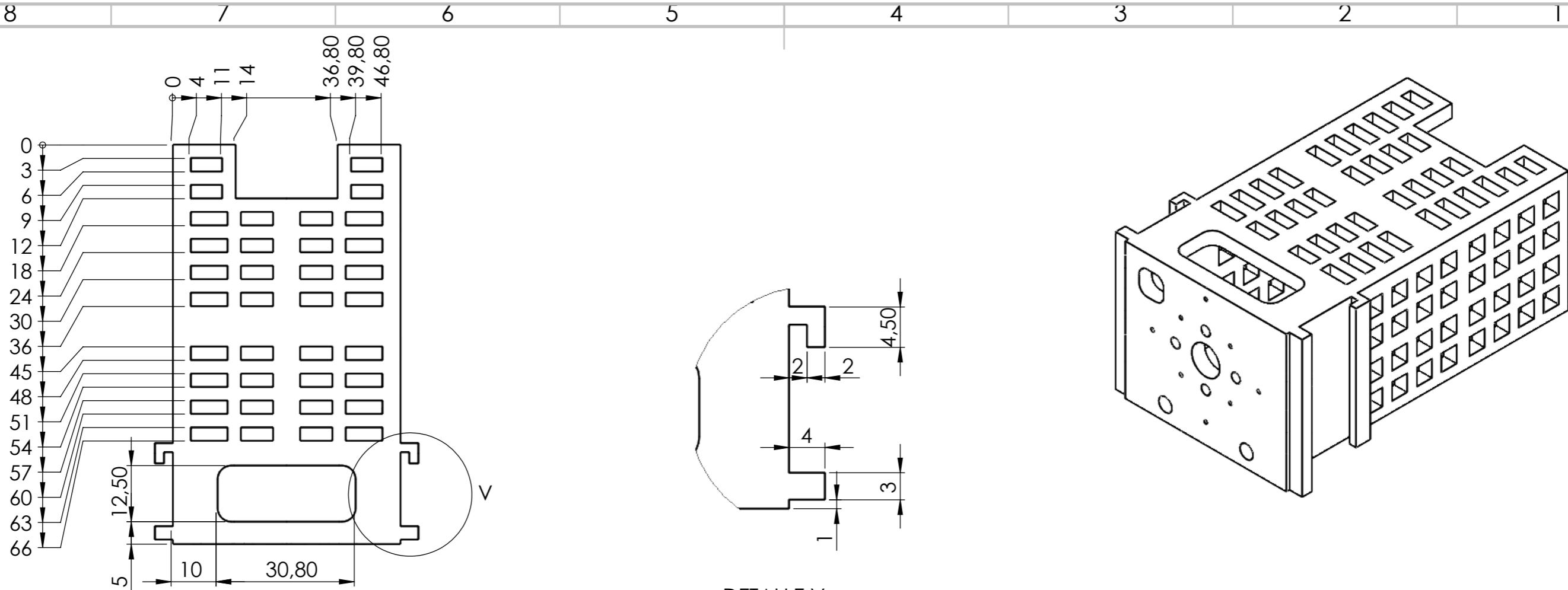
4

3

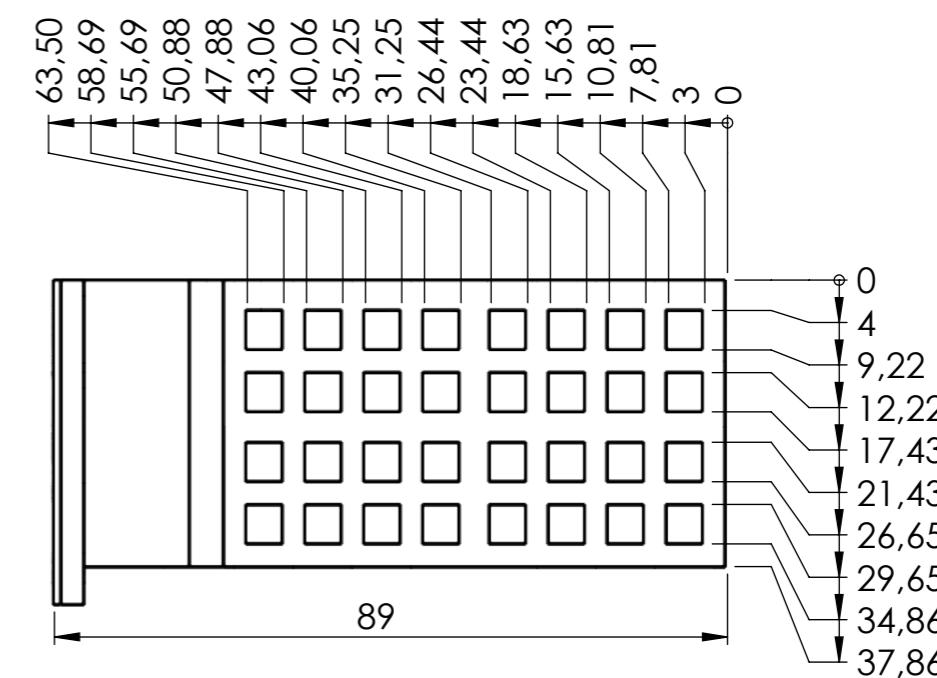
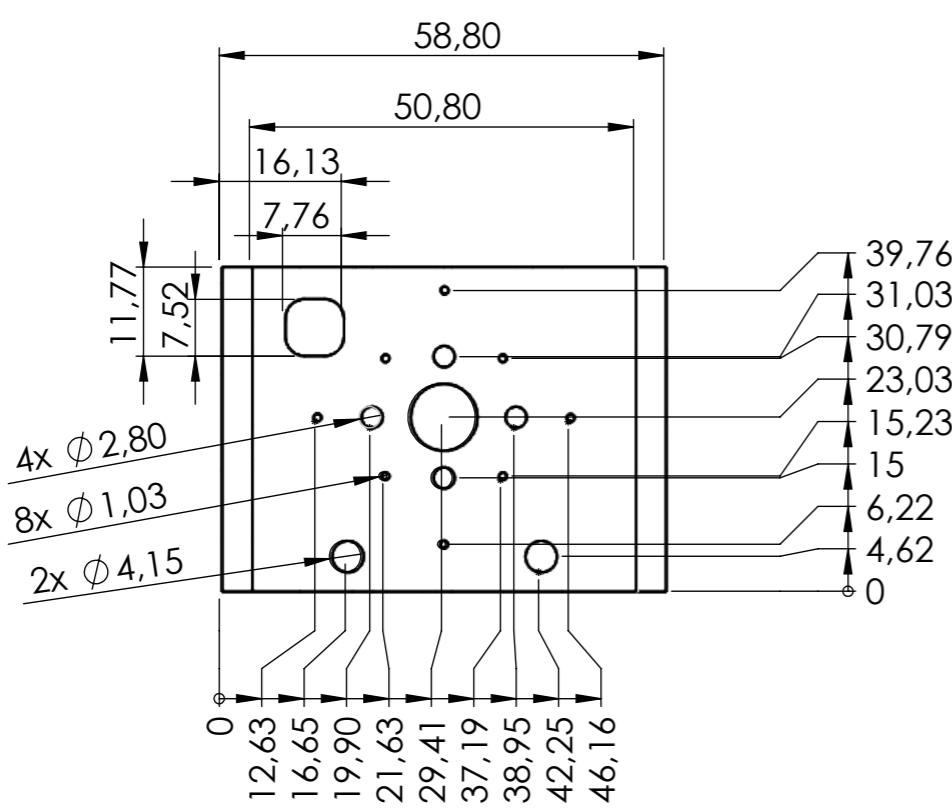
2

1





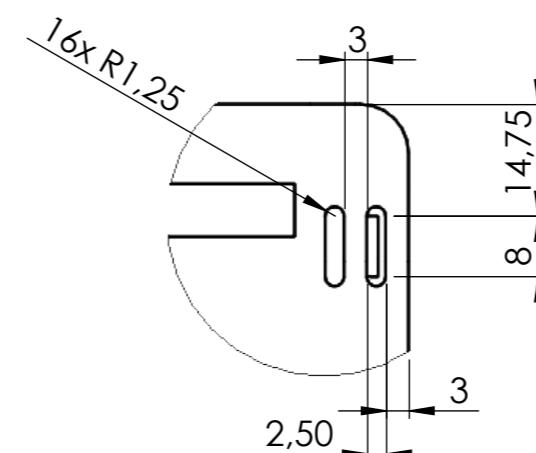
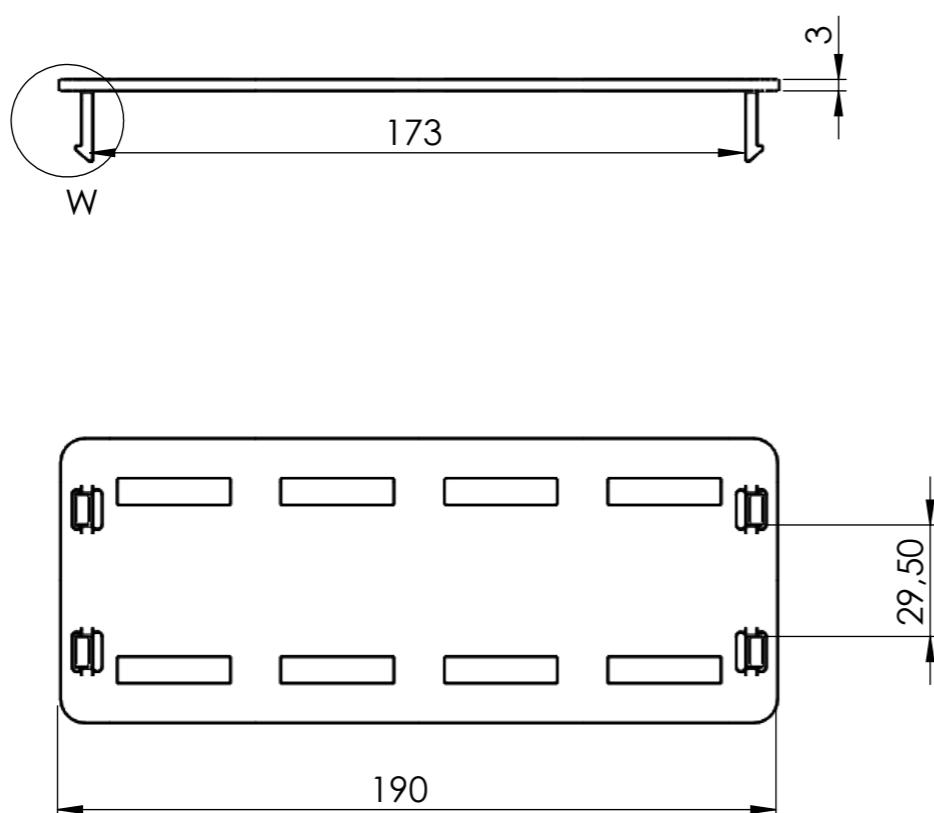
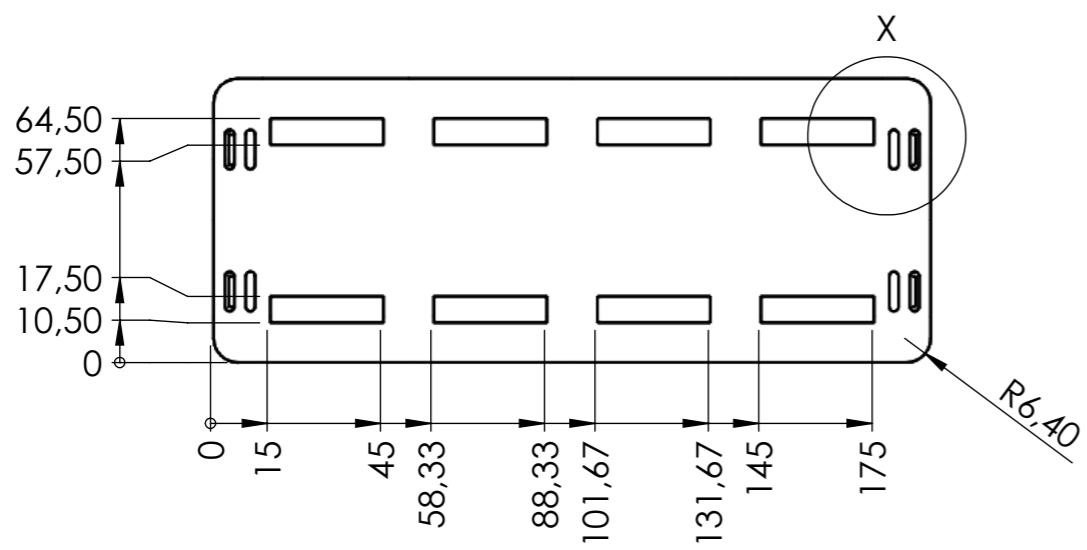
DETALLE V  
ESCALA 2 : 1



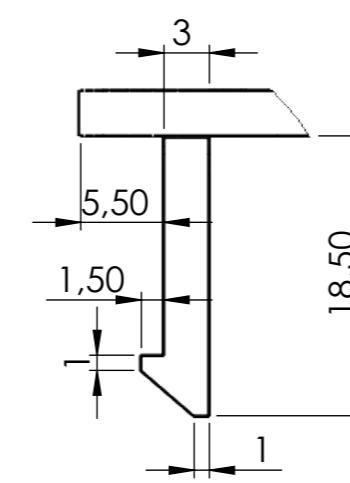
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: 0.25 m ANGULAR: 1°			ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
DIBUJ.	NOMBRE	FIRMA	FECHA		TÍTULO:	
VERIF.					CASE MOTOR	
APROB.						
FABR.						
CALID.						
			MATERIAL: Filamento de PLA o PET	N.º DE DIBUJO	CATIE-208	A3
			PESO:	ESCALA: 1:1	HOJA 1 DE 1	

### Notas:

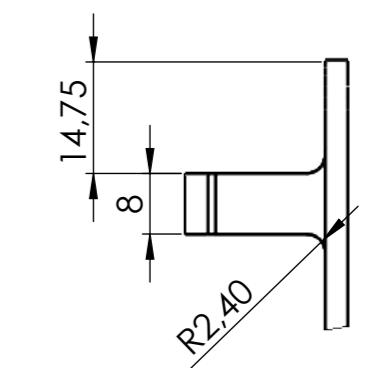
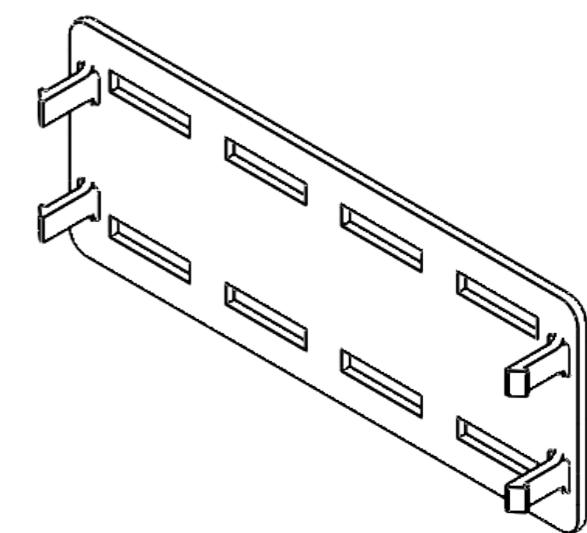
1. Todas las perforaciones son pasantes.



DETALLE X  
ESCALA 1 : 1



DETALLE W  
ESCALA 2 : 1



DETALLE Y  
ESCALA 1 : 1

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: 0.25 mm ANGULAR: 1°		ACABADO:			REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA		REVISIÓN
DIBUJ.	VERIF.	NOMBRE	FIRMA	FECHA		TÍTULO:		
VERIF.						Estructura Spear		
APROB.						Base con snap para baterías		
FABR.						MATERIAL: Filamento de PLA o PET		
CALID.						N.º DE DIBUJO		
						CATIE-209	A3	
						ESCALA: 1:2		
						HOJA 1 DE 1		

