

# PROYECTO EVA

*Antonio Molina Cortés  
Salvador Moreno Gómez*

## Introducción

EVA, ¿Qué es EVA? EVA (Electronic Visitguide Autonomic) es un robot autónomo capaz de cumplir las funciones de un guía de visita, debe de moverse de forma libre por el entorno, tener la capacidad de exponer cualquier stand del museo y contestar cualquier pregunta que se le realice.

El proyecto de EVA en su versión final debe ser capaz de ser controlada por cualquier persona, en nuestra filosofía está el eliminar todo tipo de control sobre ella que implique un conocimiento específico, dado debe ser capaz de ayudar a todo el mundo.

## Software de EVA

### *Sistema Operativo*

El sistema operativo de EVA está basado en un Debian (una distribución de Linux), más específicamente es un Raspbian.

*¿Por qué este Sistema Operativo?*

Después de un estudio de todos los sistemas operativos disponibles para nuestra Raspberry Pi, llegamos a que Raspbian era el que mejor se iba a adaptar a nuestras necesidades, dado a que nos daba soporte en todas las aplicaciones que íbamos a utilizar y está especialmente optimizado para Raspberry Pi.

### *Configuraciones de Internet*

Recién instalado el sistema operativo en la Raspberry lo primero es poder configurar un acceso a internet para poder descargar todo lo necesario para nuestro proyecto, para ello, tras investigar un poco por internet, logré encontrar como configurar una conexión WiFi.

Para configurar la WiFi, lo único que tenemos que hacer es dirigirnos a la carpeta “/etc/network” y editar el archivo “interfaces” mediante este comando:

```
sudo nano /etc/network/interfaces
```

Una vez allí, la editamos para dejarla de esta manera:

```
iface lo inet loopback
iface eth0 inet dhcp
```

```
allow-hotplug wlan0
auto wlan0
```

```
iface wlan0 inet dhcp
    wpa-ssid "NOMBRE DE LA RED"
    wpa-psk "PASSWORD"
```

Una vez hecho esto, lo único que hay que hacer es reiniciar nuestra Raspberry.

### *Lenguaje de Programación*

Este fue uno de los puntos más difíciles, ¿Qué lenguaje de programación elegir?, sabiendo que Raspbian viene por defecto con Python y un compilador de C/C++ decidimos acotar nuestras posibilidades entre estos dos lenguajes, ya que no necesitábamos una instalación extra para poder usarlos.

En este punto, intenté probar ambos lenguajes y ver las posibilidades de cada uno en nuestro proyecto. Comencé con C/C++, un lenguaje muy potente con el que ya tenía bastante experiencia, el principal problema que encontré utilizando este lenguaje es el acceso y la utilización de los GPIO de nuestra Raspberry, controlándolos mediante comandos de consola resultaba muy ortopédico, la otra opción era mediante la librería “WiringPi” (<http://wiringpi.com/>), pero en ese momento aún no daba soporte a la Raspberry Pi V2.

Después de esto intenté comenzar a desarrollar en Python, he de decir que era la primera vez que programaba en Python, nunca antes había utilizado este lenguaje.

Por suerte Python es un lenguaje con una curva de aprendizaje muy corta y no me costó mucho esfuerzo sentirme cómodo con él, además el control de los GPIO mediante Python no necesitaba ninguna instalación adicional y era muy intuitivo.

### *Detección de Códigos QR*

Una de las funciones que debe cumplir EVA, es la detección de códigos Qr, tanto con la webcam frontal como con la cámara inferior. Para realizar dicha tarea procesaremos fotografías en vez de un stream de video, así podremos reutilizar el código para las dos cámaras.

Para realizar esta función utilizaremos la librería zbar (<http://zbar.sourceforge.net/>)

#### *Pseudocódigo:*

```
Abrimos un stream in-memory con la imagen que queremos analizar;  
Ponemos el puntero a 0; //Prevenimos que no esté a 0  
Creamos un lector zbar;  
Configuramos el lector zbar;  
Escaneamos el stream en busca de patrones;  
Extraemos los símbolos para leerlos;
```

### *Detección Facial*

La detección facial, o detección de rostro en EVA es usada para saber cuándo tiene alguien delante suya, a efectos prácticos, esto nos serviría para saber cuándo alguien entra al museo y poder lanzar un mensaje de bienvenida.

El método utilizado para realizar dicha función es mediante el Algoritmo Viola & Jones, este algoritmo es de los que menos impacto producen en el sistema. ¿Cómo funciona?, Este método consta de dos partes: Un clasificador en cascada, que garantiza una discriminación rápida y un entrenador de clasificadores basado en Adaboost.

Viola & Jones tiene una probabilidad de verdaderos positivos del 99,9% y una probabilidad de falso positivo del 3,33%, a diferencia de otros algoritmos, este solo procesa la información

presente en una imagen en escala de grises, no utiliza directamente la imagen, sino una representación de ella denominada imagen integral en subregiones de tamaños diferentes.

Con lo que al unificar el clasificador en cascada con las subregiones, solamente se procesara aquellas en la que haya una posibilidad de que exista una cara, las demás regiones se desecharán.

Para realizar dichas tareas hemos utilizado la librería de procesamiento de imagen OpenCV (<http://opencv.org/>)

#### *Pseudocódigo:*

```
Sacamos una foto con la webcam;  
Introducimos la foto en una variable;  
Introducimos el fichero Clasificador;  
Detectamos patrones en la foto y si hay coincidencia se la asignamos a una variable;  
Dibujamos un cuadrado alrededor de la cara, si es que existe;
```

#### **Reconocimiento Facial**

El reconocimiento facial en EVA se determinó inviable debido al alto gasto computacional que realizaba para los resultados obtenidos, en el periodo de pruebas estuvimos utilizando el método Holístico de Template Matching.

Después de realizar un procesado de imagen para mejorar el reconocimiento, lo que se hacía era comparar toda la imagen facial respecto a una base de datos, si tenemos en cuenta que en la base de datos encontramos M personas, con N imágenes por personas, nos damos cuenta de que es imposible de implementarlo en tiempo real.

Para realizar dichas tareas hemos utilizado la librería de procesamiento de imagen OpenCV (<http://opencv.org/>)

En este método no introduciré pseudocódigo debido a que no es utilizado finalmente en nuestro proyecto.

#### **Reconocimiento de Voz**

Una de las aplicaciones que hacen más impresionante a EVA es el poder reconocer todo lo que se le dice, después de investigar y testear muchísimas librerías que ninguna funcionaba o que no se adecuaban a lo que yo buscaba, encontré una que me funcionó bastante mejor de lo que esperaba.

La librería utilizada en el reconocimiento de voz de EVA es ([https://github.com/Uberi/speech\\_recognition](https://github.com/Uberi/speech_recognition))

Después de conseguir una librería que por fin funciona a mi capricho, lo siguiente es testear que servicio de reconocimiento de voz nos va a funcionar mejor a nosotros. ¿Qué quiere decir esto? Pues por ejemplo, que servicio de voz me va a detectar mejor y con menos fallos la palabra EVA.

A la hora de testear, preferíamos un reconocimiento de voz local, ya que podríamos prescindir de una conexión WiFi, este servicio lo conseguimos gracias a Sphinx, pero el resultado no fue el



esperado, el alto tiempo de procesado y la alta tasa de errores nos hizo determinarnos por un servicio de reconocimiento de voz online.

Probamos casi todos los servicios de reconocimiento más conocidos, los que mejor se supone que van a funcionar, empezamos con Wit.ai, un servicio de voz gratuito en propiedad de Facebook, seguimos con Api.ai, otro servicio de voz gratuito donde dos de sus grandes inversores son Intel y Motorola.

Después de probar los servicios gratuitos, que no iban mal, quisimos un reconocimiento de voz aún más rápido y más eficaz, por lo que fuimos a servicios de pago pero que si no llegas a cierto umbral, el servicio es gratuito, esto nos viene fenomenal para las pruebas, aparte de que los umbrales eran lo suficientemente altos como para que nuestro proyecto lo soportara. Comenzamos con el más conocido, Google, seguido del servicio de Microsoft, aparte también testearon el de IBM y el de AT&T.

Los dos servicios que mejor funcionaban y tenían una respuesta más rápida eran el de Google y el de Microsoft. Una vez en estas dos alternativas, EVA evolucionó y se le implementó la funcionalidad de traducir de cualquier a cualquier idioma, por lo que también tendría que reconocer en cualquier idioma, o incluso varios idiomas entremezclados. Para la prueba utilizamos una mezcla de Español-Inglés en varias frases, en la que finalmente Google salió con mejores resultados que Microsoft.

#### *Pseudocódigo:*

```
Iniciamos el objeto de reconocimiento;  
Iniciamos el objeto de micrófono;  
Ajustamos una puerta de ruido o threshold al nivel del ruido ambiente;  
Comenzamos a escuchar hasta detectar que se ha terminado de hablar;  
Enviamos la información a google...;  
Recibimos la información de google y la almacenamos;  
Buscamos si 'EVA' está entre las palabras reconocidas;  
Si la encuentra envía la frase a nuestro BOT para procesarla;  
Si no la encuentra o no reconoce nada vuelve a empezar a escuchar;
```

#### *Text to Speech (Voz)*

Para darle voz a EVA hemos elegido a "Festival" un sintetizador de texto a voz multilenguaje, por defecto viene con una voz masculina para Inglés, nosotros le dimos nuestra propia voz femenina hecha en la Junta de Andalucía llamada Silvia.

([http://forja.guadalinex.org/frs/?group\\_id=21&release\\_id=121](http://forja.guadalinex.org/frs/?group_id=21&release_id=121))

Estos son los pasos que seguimos para lograr instalar y configurar Festival en nuestra Raspberry Pi:

Para instalarlo solo es necesario usar en consola el siguiente comando:

```
sudo apt-get -y install festival
```

Para poder usar la salida de audio jack, necesitamos introducir este comando:

```
amixer cset numid=3 1
```

Ahora pasamos a descargar la voz Silvia:

```
wget http://forja.guadalinux.org/frs/download.php/154/festvox-sflpc16k\_1.0-1\_all.deb
```

Descomprimos e instalamos:

```
dpkg -i festvox-sflpc16k_1.0-1_all.deb
```

### *Inteligencia Artificial*

Comenzamos usando bots de internet para establecer conversaciones, pero nosotros queríamos que EVA tuviera una personalidad propia y acelerar el tiempo de respuesta en la pregunta, por lo que finalmente desarrollamos nuestro propio bot offline con nuestra propia base de datos.

Antes de eso tuve que informarme sobre cómo trabaja la inteligencia artificial, el método que más me llamó la atención y la que actualmente usamos en EVA es ALICE (The Artificial Linguistic Internet Computer Entity), este método tiene un tipo de base de datos especial llamado AIML que ayuda a los bots de internet a acelerar su computación.

En EVA hemos creado una variante de ALICE, con un nuevo tipo de base de datos bautizado IA, en la cual seguimos un protocolo de almacenaje de información más eficiente para nuestro bot, siendo así más fácil el procesamiento de información y más rápida la implementación de funciones predefinidas.

Para dicho proceso de cambio entre AIML y IA, también desarrollé un conversor de AIML-IA para poder volcar mi antigua base de datos a la nueva.

Entre las funciones del bot, aparte de guardar respuestas predeterminadas también puede calcular la hora exacta, el día exacto, poner música organizada por categorías, traducir a cualquier idioma, buscar información en Wikipedia, apagar el sistema, enviar comandos de reconocimiento facial, detección facial y QR.

### *Pseudocódigo:*

Lectura del texto;

Abrimos la base de datos;

Leemos la base de datos;

Si el texto contiene el patrón extraemos las respuestas para dicho patrón;

Si respuesta no es una acción, leerla por text to speech;

Si es una acción, realizar dicha acción;

Si no se encuentra el patrón, buscar el texto en una base de datos de internet;

Leer el texto extraído de internet;

### *Expresión de sentimientos (Rostro)*

Como todo robot social EVA debe tener un aspecto amigable y ser capaz de expresar sentimientos, esto lo conseguimos mediante una pantalla que se encarga de cargar y mostrar su rostro en todo momento.

Este punto es uno de los más importantes, ya que es el más visible hacia la persona que interactúa con EVA, y es uno de los que más problemas me ha dado (y me sigue dando) a la hora de implementarlo.

¿Por qué? Pues muy sencillo, no hay un programa que se adapte completamente a mis necesidades, por lo que teníamos dos caminos: Desarrollar nuestro propio visor de imágenes dedicado o intentar adaptar todo lo posible algún visor comercial.

Nos decidimos por intentar adaptar un visor comercial, era la peor opción, pero la única que tenía viabilidad debido a la limitación de nuestro tiempo. El visor elegido finalmente fue “Mirage” ¿Qué limitaciones teníamos con él? Pues cada vez que cambiábamos de cara, al cerrar el programa con la cara actual e iniciarlo con la nueva, había una intersección entre imagen e imagen que no quedaba para nada bien. Este problema lo conseguí “solucionar” creando un script que lleve el control del programa y de las caras en el cual, cada vez que se quiera mostrar una nueva cara se ejecutara un proceso hijo con dicha cara y no se matará al anterior proceso (el que contenía la cara antigua) hasta que la nueva cara no se esté mostrando.

### *Control de los GPIO*

Los GPIO (General Ports Input Output) son los encargados de controlar todo el sistema de tracción de EVA, así como la iluminación de la cámara inferior y la tijera de la cabeza.

Para ayudarnos al control de los GPIO creamos un protocolo de comunicación explicado en otro apartado de este documento, con lo cual, cualquier aplicación es capaz de comunicarse con los GPIO.

Esta parte del software es la más parecida a lo que estamos acostumbrados a programar en un microcontrolador, no tengo mucho que explicar aquí, lo único que realiza esta aplicación es leer la orden que recibe mediante la comunicación con las otras aplicaciones y aplicarla.

### *Control por Bluetooth*

Mientras desarrollábamos a EVA, nos dimos cuenta de que no teníamos control ninguno sobre ella a la hora de querer moverla a nuestro antojo, ya que todo lo demás estaba pensado para que fuera 100% autónoma, así que decidimos hacer un apartado bluetooth para poder controlar sus movimientos mediante un Smartphone.

Por lo que compramos un módulo de bluetooth compatible y lo conectamos mediante los GPIO, tuvimos que modificar algunos ficheros de configuración de raspbian para poder usar dicho puerto serie a nuestro gusto.

Finalmente conseguimos obtener un movimiento libre mediante bluetooth en EVA, y un reconocimiento de voz alternativo.

### *Pseudocódigo:*

```
Abrir puerto Serie;  
Leer Carácter;  
Es movimiento o voz?;  
    Si es movimiento;  
        Bajar la cabeza si no lo estaba;  
        Enviar al control GPIO;  
    Si es voz;  
        Subir la cabeza si no lo estaba;  
        Si no es el último carácter seguir leyendo;  
        Si es último carácter, enviarlo al bot;
```

### *Seguimiento de una línea*

El recorrido de EVA dentro del museo queríamos hacerlo como un entorno cerrado, es decir, sin posibilidad a error. Nuestra idea era implementar un recorrido en el museo Principia mediante marcas en el suelo, en nuestro caso mediante una línea y códigos QR, una vez implementado esto físicamente solo teníamos que hacer el típico sigue-líneas pero con una pequeña base de datos relacionada con las posiciones de los QR, sabiendo así en qué posición del recorrido se encuentra y hacia qué posición debe ir.

Para poder leer las dos marcas del recorrido (línea y QR) no nos vale con unos simples sensores, así que le implementamos una nueva cámara que se ocupase de realizar dicha función.

Para mejorar la visibilidad se le implementaron unos leds al estilo de flash de una cámara de fotos, ya que nosotros vamos a trabajar mediante fotos, no videos. Para mejorar el procesado de la línea y el código QR necesitamos procesar dicha imagen, marcándonos unos thresholds y pasándola a blanco y negro.

Esta función no se ha podido terminar debido a la falta de tiempo, ya que es necesario crear antes el recorrido para poder crear nuestra base de datos y determinar el código con un testeo previo.

### *Auto inicio de las aplicaciones*

Una de las cosas fundamentales en EVA era que cuando se iniciara se auto ejecutara ella misma todo lo necesario para que esté lista para su uso, no debe de tener a nadie especializado para poder ponerla en marcha, por lo que estudiamos la manera de distribuir nuestras aplicaciones para que puedan abrirse y trabajar de forma modular.

### *Sistema de comunicación entre las aplicaciones*

Uno de los grandes problemas a la hora de programar a EVA fue que todas las aplicaciones tengan comunicación entre sí, dado que hay muchas acciones que pueden hacer que EVA cambie de expresión, o varios procesos que necesiten tener acceso a los GPIO, por lo que lo mejor fue realizar un fichero de comunicación para la expresión facial el cual puede ser modificado por cualquier proceso, y otro para el control de los GPIO.

### *Sistema de aprovechamiento del consumo*

Un punto clave en el proyecto EVA era conseguir todo lo anterior con una simple Raspberry Pi, sin tener que llevarla a su máximo de rendimiento, ya que si trabajamos cerca del 100% podemos obtener resultados con cierto retardo e incluso fallos o colapsos en los procesos.

Cada proceso de EVA sabe cuándo es su momento de ejecutarse, por lo que pasa inactivo la mayor parte del tiempo hasta que se le solicita, y los que necesariamente tienen que estar el 100% del tiempo ejecutándose, se les ha ajustado su funcionamiento para conseguir un procesado más lento, pero con menos ocupación de CPU.

## Hardware de EVA

EVA, aunque pueda parecer rudimentaria en cuanto al exterior, es un prototipo bastante bien acabado y muy parecido a la que podría ser una versión comercial de ella.

El hardware se podría dividir en dos grandes ramas: la parte mecánica y la electrónica.

### EVA en la electrónica

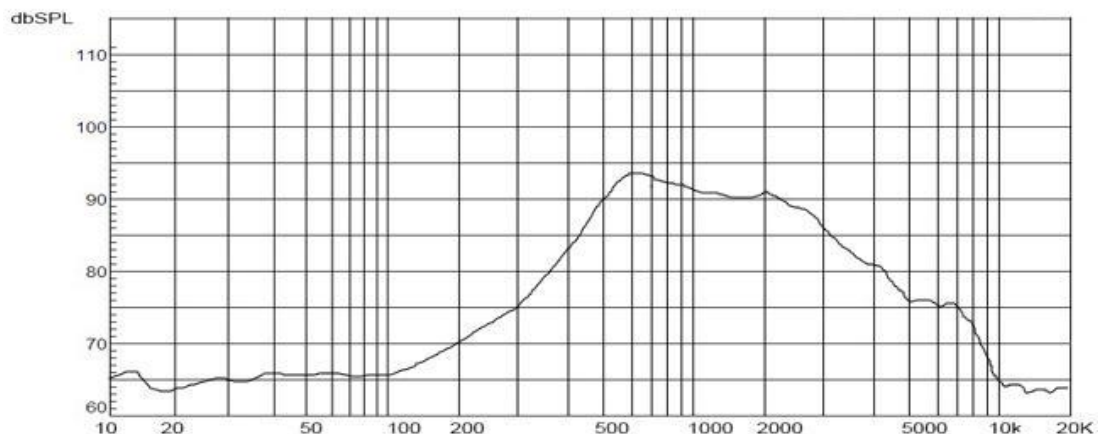
Toda la electrónica que EVA está segmentada en módulos independientes, esto hace que sea mucho más fácil y barato repararla, sustituirle placas, modificarla, etc.

(Ver plano E1)

EVA posee un corazón compuesto por una Raspberry Pi 2B que se mueve unas 25 veces más rápido que un microcontrolador promedio. Todos los módulos electrónicos son independientes físicamente unos de otros y están conectados entre sí a través de la Raspberry.

(Ver plano E6)

EVA posee un potente equipo de sonido estéreo de 10Wrms por canal, y un conjunto de 12 altavoces de 3" dispuestos en conjuntos de 3 en cada esquina superior del chasis, describiendo así una semiesfera de cobertura sonora coloreado en las frecuencias predominantes en la voz (aproximadamente en 2kHz).



Curva aproximada de respuesta del equipo de audio de EVA medida con un micrófono de medición en una zona medianamente acondicionada acústicamente.

El amplificador consta de un amplificador operacional doble de gran potencia TDA2009, el circuito está compuesto de unos condensadores de entrada, otros de salida, otros de desacoplo, unas resistencias a modo de divisor de tensión, etc. Para más información mirar el datasheet del fabricante.

Este amplificador tiene una impedancia de salida de  $4\Omega$ , como nuestros altavoces son de  $8\Omega$  los hemos dispuesto en un circuito mixto hasta conseguir  $5.3\Omega$

(Ver plano E2)

EVA posee un sistema de ventilación que va de manera independiente al funcionamiento de la Raspberry, osea que no depende las órdenes de la Raspberry para su funcionamiento, tiene una sonda de calor puesta encima del disipador de la Raspberry para poder controlar la cantidad de aire que inyecta dentro de chasis. El circuito de un transistor TIP142, una NTC a modo de sonda de temperatura, un potenciómetro para regular la curva de disparo de los ventiladores, dos potentes ventiladores con motores brushless, un condensador de arranque y una resistencia de mantenimiento de un mínimo de intensidad a los ventiladores.

(Ver plano E3)

Los movimientos de EVA, tanto las cadenas como la cabeza, están accionados por motores DC con reductoras los cuales a 12V desarrollan una potencia de 180N/cm. Las etapas de potencia están compuestas por drivers de motor MCP14E5, unas resistencias de pull-up, unos puentes rectificadores a modo de protectores de energía contraelectromotriz.

(Ver plano E4)

EVA dispone de ojos y rostro, con esto queremos decir que tiene una cámara en la parte superior de la cabeza y una pantalla donde se muestra una cara o información para el técnico encargado del mantenimiento.

EVA está dotada de una cámara inferior para poder seguir una línea e identificar y procesar códigos QR, así como de un sistema de ayuda a la iluminación para mejorar la captación de la cámara inferior.

(Ver plano E5)

Todo el sistema electrónico de EVA está sustentado por una batería de plomo de 12V/7Ah, el sistema está protegido por un fusible y por un interruptor de corte general. Todos los sistemas que necesitan una tensión de alimentación de 12V van directamente a la tensión de la batería, mientras que los que van a 5V van conectados a un transformador DC-DC que hace el pase de 12 a 5V mucho más eficiente que con un simple regulador de tensión ordinario. Con esta batería se consigue una autonomía de 6h sin realizar movimientos y de 3,5h incluyendo movimientos continuos.

(Ver plano E7)

## **EVA en la mecánica**

### ***El chasis***

Todo el chasis de EVA está construido en madera, más concretamente en chapón industrial, se eligió este material por su facilidad para trabajarlo (mucho más fácil que el metacrilato o la chapa), su ligereza (es la madera más ligera del mercado), su precio (dentro de las maderas, las laminadas son de las más baratas) y su beta (ya que el robot se iba a construir en madera, queríamos resaltar el hecho que era de madera).

Todas las piezas se pueden sacar cómodamente de un pliego de madera estándar.

(Ver plano M2)

Hay que tener siempre en cuenta que el chapón industrial tiene una cara bonita y otra cara que puede venir más estropeada, para a la hora de cortar las piezas tener siempre en cuenta la cara de la pieza que quedará a la vista.

(Para ver las piezas más detalladamente ver planos de PA1 a PA9)

Todas las piezas de madera van unidas entre sí con adhesivo de montaje, a la hora de pegarlas es muy útil ayudarse de unas escuadras de fijación para que las piezas se queden es ángulos rectos perfectos, sino puede salir el chasis descuadrado.

La bandeja extraíble va fijada a los raíles mediante tornillos de madera de 4x12mm tanto a la bandeja como al chasis.

En todas esquinas del chasis se han colocado escuadras para mejorar la estabilidad de este, estas escuadras también han sido fijadas con adhesivo de montaje.

Las puertas están sujetas a las paredes laterales del chasis mediante unas bisagras superacodadas, esto hace que las puertas se queden embutidas dentro del chasis lo que hace el diseño más estético y robusto y también al abrirlas se deja libre el paso de la bandeja extraíble.

(Para ver las secuencias de montaje de manera detallada ver planos SA1 a SA6)

### *Los sistemas de tracción*

Los sistemas de tracción de EVA son cadenas a modo de orugas conformados por un engranaje central que posee la tracción, 2 engranajes satélites que se encargan de transmitir el movimiento del engranaje central a la cadena y un engranaje tensor en la parte superior. Todo esto esta empaquetado, como si de un sándwich se tratara, por dos triángulos que se encargan de mantener todo el sistema en su sitio.

(Para ver las piezas más detalladamente ver planos de PB1 a PB6)

¿Porque escogimos este sistema? Es menos eficiente que unas ruedas, más apatoso, más difícil de mantener, más complicado de montar, etc...

Simplemente queríamos que se pareciese a Wall-e...

Los motores están soportados en unos descentradores que proporcionan la posibilidad de colocar los motores en la posición más adecuada para que no estorben con el resto del chasis.

Cada engranaje satélite está compuesto de 4 partes: una corona de engranaje levógiro, una corona de engranaje dextrógiro (aunque y en los planos PB3.1 y PB3.2 se vean como dos piezas separadas, no es así, solo se han representado por separado para su mejor comprensión como se muestra en el plano PB3.3) y dos radios de engranaje embutidos uno en cada corona de engranaje y pegados con pegamento de cianocrilato.

Para fijar las dos medias partes del engranaje se utilizarán 3 tornillos métricos de 3x35mm.

Para mejorar el rendimiento del sistema de tracción se han creado unos rodamientos a medida, los rodamientos están conformados por un centrador, una canastilla y 12 bolas de PLA



de 6mm de diámetro. También los engranajes se han diseñado con dientes helicoidales de doble vertiente, esto baja el rozamiento de los engranajes, reduce la posibilidad de que se encasquillen los engranajes entre sí, los hace más silenciosos que si fuesen con dientes planos y al hacerlos de doble vertiente no sufren por tener una fuerza constante de empuje perpendicular al plano de giro, al contrario de como pasa en los engranajes helicoidales ordinarios. Los pernos que soportan los engranajes satélites son pernos huecos que aportan menos peso que unos macizos, son más fáciles de trabajar y se enfrían con mucha más facilidad si llegasen a soportar calentamiento debido al movimientos de los engranajes.

El tensor de la cadena se monta igual de los engranajes satélites, pero con la ventaja que solo son 2 piezas.

(Para ver las secuencias de montaje de manera detallada ver planos SB1 a SB6)

### *El sistema de amortiguación*

EVA está equipada con un sistema de amortiguación basculante adaptativa independiente. Esto quiere decir que:

**Basculante-** La amortiguación no permite que los triángulos se muevan arriba y abajo (esto sería un problema ya que los motores tienen poco margen de movimiento vertical) sino que giran con respecto a un eje colocado debajo del chasis, esto deja a EVA como si fuese una mecedora.

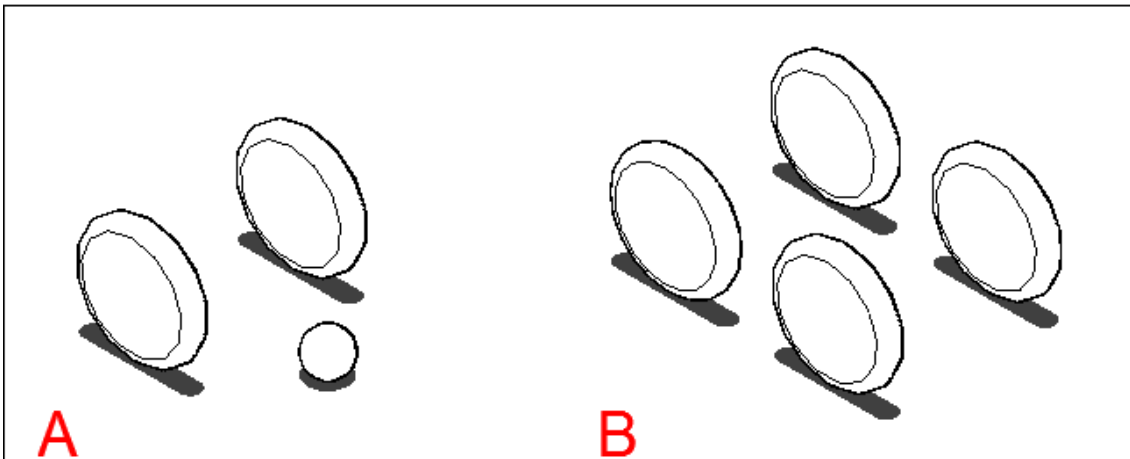
**Adaptativa-** Este tipo de amortiguación está diseñado para que se adapte a cualquier tipo de relieve para que EVA pueda sortear obstáculos de pequeño tamaño. También posee muelles que son los que aportan la elasticidad al sistema de amortiguación, a los cuales se les puede regular la tensión de manera independiente a cada uno.

**Independiente-** Cada juego de tracción (izquierdo y derecho) trabaja independiente el uno del otro, esto quiere decir que un juego puede estar a 0° de inclinación mientras que el otro puede estar a 15°, por ejemplo.

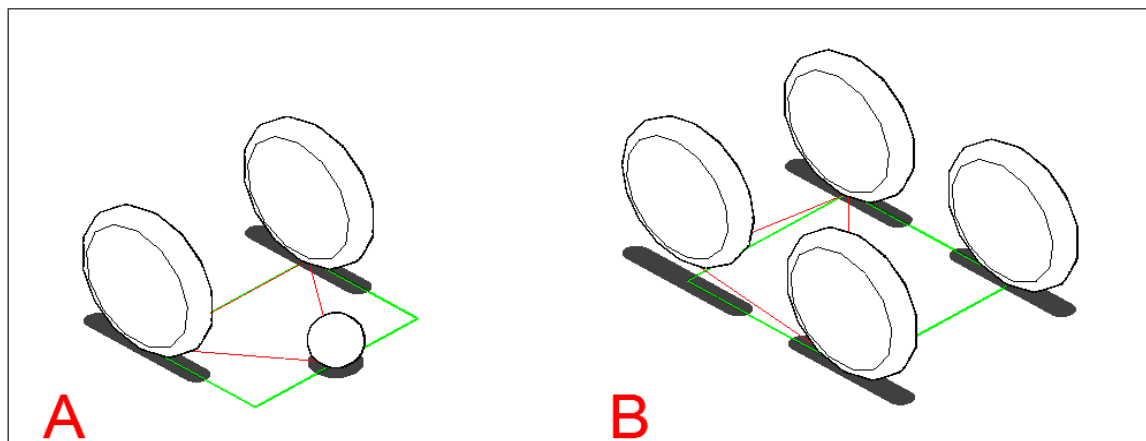
(Para ver las piezas más detalladamente ver planos de PC1 a PC3)

¿Por qué dotamos a EVA con este sistema? Simplemente todo empezó por cubrirnos las espaldas: al no estar trabajando con ruedas, que es lo más normal, ya no trabajamos con solo dos puntos de apoyo, sino con dos planos que son las cadenas, en el caso posible de que no montásemos bien los juegos de tracción EVA quedaría coja y el sistema de tracción tendría más presión en unos puntos que en otros, forzando así el sistema de manera innecesaria y envejeciéndolo mucho más rápido.

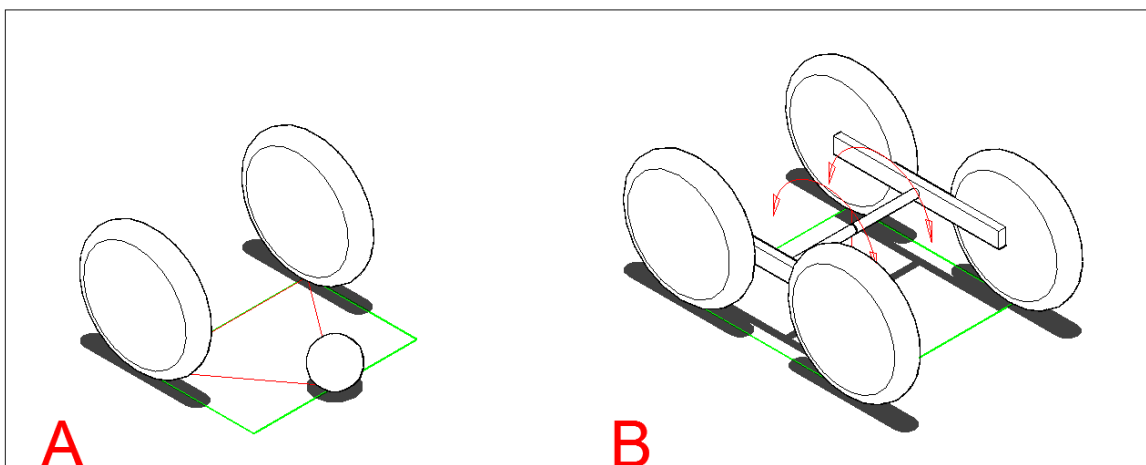
Después vimos que el sistema tenía potencial y le agregamos muelles regulables, un buen radio de acción, etc, y conseguimos ahora sortear obstáculos de 7cm de altura sin que EVA pierda el equilibrio.



En el diagrama A podemos ver el sistema típico de tracción con dos ruedas y una bola loca y en el diagrama B vemos nuestro sistema con 4 ruedas y unas cadenas que las recorrieran, aunque las cadenas no estén representadas los puntos de contacto con el suelo son los mismos.



En el diagrama A vemos como tres puntos siempre van a describir un plano así que nunca estaría cojo, mientras que en el diagrama B vemos que cuando hay cuatro puntos y uno de ellos no se encuentra perfectamente en plano con los demás crea dos planos distintos, produciendo así una cojera.



En el diagrama A no haría falta ningún sistema de adaptación para asegurar el plano de contacto con el suelo. En el diagrama B se muestra la solución que le hemos aportado a nuestro diseño, simplemente lo hemos dotado de un sistema basculante, de esta manera se asegura el plano perfecto de contacto.

(Para ver las secuencias de montaje de manera detallada ver planos SC1 a SC3)

### *El sistema de ascensión de la cabeza*

EVA está preparada como si de un vehículo antivandálico se tratase, la pantalla está asegurada dentro del chasis y solo cuando se está interactuando con ella la saca, es similar al sistema de defensa de una tortuga.

(Para ver las piezas más detalladamente ver planos de PD1 a PD3)

El sistema consta de una tijera de elevación con una tapa en la parte superior y de un motor que mueve unos engranajes que estos a su vez desplazan unos tornillos a modo de sinfín, de este modo, la tijera es la que hace la fuerza, ya que tiene unos mulles, y los tornillos se encargan de que la cabeza suba por parejo. La tijera para no perder la línea que tiene que seguir a la hora de desplazarse tiene unos railes de desplazamiento que aseguran siempre el buen funcionamiento de la tijera.

### *Los demás apliques*

EVA tiene un motón de apliques más como la pinza de micrófono u otros, pero que no entrañan ninguna dificultad de comprensión así que solo con ver los planos se comprenden.

(Para ver las piezas más detalladamente ver planos de PE1 a PE6)

**ANEXOS**

# *Programación*

chatterbotapi.py

```
import re
import sys
import hashlib

if sys.version_info >= (3, 0):
    from urllib.request import build_opener, HTTPCookieProcessor, urlopen
    from urllib.parse import urlencode
    import http.cookiejar as cookielib

else:
    from urllib import urlencode, urlopen
    from urllib2 import build_opener, HTTPCookieProcessor
    import cookielib

from collections import OrderedDict

import uuid
import xml.dom.minidom

"""
    chatterbotapi
    Copyright (C) 2011 pierredavidbelanger@gmail.com

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU Lesser General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public License
    along with this program. If not, see <http://www.gnu.org/licenses/>.
"""

#####
# API
#####

class ChatterBotType:
```

```
CLEVERBOT = 1
JABBERWACKY = 2
PANDORABOTS = 3
```

```
class ChatterBotFactory:
```

```
    def create(self, type, arg = None):
        if type == ChatterBotType.CLEVERBOT:
            return _Cleverbot('http://www.cleverbot.com',
                              'http://www.cleverbot.com/webservicemin?uc=165', 35)
        elif type == ChatterBotType.JABBERWACKY:
            return _Cleverbot('http://jabberwacky.com',
                              'http://jabberwacky.com/webservicemin', 29)
        elif type == ChatterBotType.PANDORABOTS:
            if arg == None:
                raise Exception('PANDORABOTS needs a botid arg')
            return _Pandorabots(arg)
        return None
```

```
class ChatterBot:
```

```
    def create_session(self):
        return None
```

```
class ChatterBotSession:
```

```
    def think_thought(self, thought):
        return thought

    def think(self, text):
        thought = ChatterBotThought()
        thought.text = text
        return self.think_thought(thought).text
```

```
class ChatterBotThought:
```

```
    pass
```

```
#####
# Cleverbot impl
#####
```

```
class _Cleverbot(ChatterBot):
```

```
    def __init__(self, baseUrl, serviceUrl, endIndex):
```

```

        self.baseUrl = baseUrl
        self.serviceUrl = serviceUrl
        self.endIndex = endIndex

    def create_session(self):
        return _CleverbotSession(self)

class _CleverbotSession(ChatterBotSession):

    def __init__(self, bot):
        self.bot = bot
        self.vars = OrderedDict()
        #self.vars['start'] = 'y'
        self.vars['stimulus'] = ''
        self.vars['islearning'] = '1'
        self.vars['icognoid'] = 'wsf'
        #self.vars['fno'] = '0'
        #self.vars['sub'] = 'Say'
        #self.vars['cleanslate'] = 'false'
        self.cookieJar = cookielib.CookieJar()
        self.opener = build_opener(HTTPCookieProcessor(self.cookieJar))
        self.opener.open(self.bot.baseUrl)

    def think_thought(self, thought):
        self.vars['stimulus'] = thought.text
        data = urlencode(self.vars)
        data_to_digest = data[9:self.bot.endIndex]
        data_digest = hashlib.md5(data_to_digest.encode('utf-8')).hexdigest()
        data = data + '&icognocheck=' + data_digest
        url_response = self.opener.open(self.bot.serviceUrl, data.encode('utf-8'))
        response = str(url_response.read())
        response_values = re.split(r'\\r|\\n', response)
        #self.vars['??'] = _utils_string_at_index(response_values, 0)
        self.vars['sessionId'] = _utils_string_at_index(response_values, 1)
        self.vars['logurl'] = _utils_string_at_index(response_values, 2)
        self.vars['vText8'] = _utils_string_at_index(response_values, 3)
        self.vars['vText7'] = _utils_string_at_index(response_values, 4)
        self.vars['vText6'] = _utils_string_at_index(response_values, 5)
        self.vars['vText5'] = _utils_string_at_index(response_values, 6)
        self.vars['vText4'] = _utils_string_at_index(response_values, 7)
        self.vars['vText3'] = _utils_string_at_index(response_values, 8)
        self.vars['vText2'] = _utils_string_at_index(response_values, 9)
        self.vars['prevref'] = _utils_string_at_index(response_values, 10)
        #self.vars['??'] = _utils_string_at_index(response_values, 11)
        # self.vars['emotionalhistory'] = _utils_string_at_index(response_values, 12)

```



```

# self.vars['ttsLocMP3'] = _utils_string_at_index(response_values, 13)
# self.vars['ttsLocTXT'] = _utils_string_at_index(response_values, 14)
# self.vars['ttsLocTXT3'] = _utils_string_at_index(response_values, 15)
# self.vars['ttsText'] = _utils_string_at_index(response_values, 16)
# self.vars['lineRef'] = _utils_string_at_index(response_values, 17)
# self.vars['lineURL'] = _utils_string_at_index(response_values, 18)
# self.vars['linePOST'] = _utils_string_at_index(response_values, 19)
# self.vars['lineChoices'] = _utils_string_at_index(response_values, 20)
# self.vars['lineChoicesAbbrev'] = _utils_string_at_index(response_values,
21)
# self.vars['typingData'] = _utils_string_at_index(response_values, 22)
# self.vars['divert'] = _utils_string_at_index(response_values, 23)
response_thought = ChatterBotThought()
response_thought.text = _utils_string_at_index(response_values, 0)
return response_thought

```

```

#####
# Pandorabots impl
#####

```

```

class _Pandorabots(ChatterBot):

```

```

    def __init__(self, botid):
        self.botid = botid

    def create_session(self):
        return _PandorabotsSession(self)

```

```

class _PandorabotsSession(ChatterBotSession):

```

```

    def __init__(self, bot):
        self.vars = {}
        self.vars['botid'] = bot.botid
        self.vars['custid'] = uuid.uuid1()

    def think_thought(self, thought):
        self.vars['input'] = thought.text
        data = urlencode(self.vars)
        url_response = urlopen('http://www.pandorabots.com/pandora/talk-xml',
data)
        response = url_response.read()
        response_dom = xml.dom.minidom.parseString(response)
        response_thought = ChatterBotThought()
        that_elements = response_dom.getElementsByTagName('that')

```

```
        if that_elements is None or len(that_elements) == 0 or that_elements[0] is
None:
            return ''
        that_elements_child_nodes = that_elements[0].childNodes
        if that_elements_child_nodes is None or len(that_elements_child_nodes) ==
0 or that_elements_child_nodes[0] is None:
            return ''
        that_elements_child_nodes_data = that_elements_child_nodes[0].data
        if that_elements_child_nodes_data is None:
            return ''
        response_thought.text = that_elements_child_nodes_data.strip()
        return response_thought
```

```
#####
# Utils
#####
```

```
def _utils_string_at_index(strings, index):
    if len(strings) > index:
        return strings[index]
    else:
        return ''
```

### speech.py

```
import speech_recognition as sr
import time
import os

def child(recognizer, audio):
    try:
        # for testing purposes, we're just using the default API key
        # to use another API key, use `r.recognize_google(audio,
key="GOOGLE_SPEECH_RECOGNITION_API_KEY")`
        # instead of `r.recognize_google(audio)`
        texto = recognizer.recognize_google(audio, language="es-ES")
        print texto
        if texto.find("beba") != -1:
            texto = texto[:texto.find("beba")] + 'eva' + texto[texto.find("beba")+4:]
        if texto.find("EBA") != -1:
            texto = texto[:texto.find("EBA")] + 'eva' + texto[texto.find("EBA")+3:]
        print("Google Speech Recognition thinks you said " + texto)
        texto = texto.lower()
        if texto.find('eva') != -1:
            print 'Va a hablar'
            print "python /home/pi/Desktop/Granada-Version/BETA-
WORKING/Robot/evabot.py %s" % texto.encode('UTF-8')
            os.system('python /home/pi/Desktop/Granada-Version/BETA-
WORKING/Robot/evabot.py "%s"' % texto.encode('UTF-8'))
        except sr.UnknownValueError:
            print("Google Speech Recognition could not understand audio")
            time.sleep(15)
            print 'seacabo'
        except sr.RequestError as e:
            print("Could not request results from Google Speech Recognition service;
{0}".format(e))
            os._exit(0)

# this is called from the background thread
def callback(recognizer, audio):
    print "Entra"
    newpid = os.fork()
    if newpid == 0:
        child(recognizer, audio)
    else:
        time.sleep(1)
    # received audio data, now we'll recognize it using Google Speech Recognition
    print 'Durmiendo'
```

```

time.sleep(5)
print 'GO'

try:
    os.system('pulseaudio --kill')
    os.system('jack_control start')
except:
    pass

r = sr.Recognizer()
m = sr.Microphone()
with m as source:
    print("Un momento de silencio para ajustar la puerta de ruido...")
    r.adjust_for_ambient_noise(source)
    print("El threshold minimo ajustado a {}".format(r.energy_threshold))

# start listening in the background (note that we don't have to do this inside a
`with` statement)
stop_listening = r.listen_in_background(m, callback)
# `stop_listening` is now a function that, when called, stops background listening

# do some other computation for 5 seconds, then stop listening and keep doing
other computations
time.sleep(5) # we're still listening even though the main thread is doing other
things
#stop_listening() # calling this function requests that the background listener
stop listening
while True:
    time.sleep(20)
#   with m as source:
#       print("Un momento de silencio para ajustar la puerta de ruido...")
#       r.adjust_for_ambient_noise(source)
#       print("El threshold minimo ajustado a {}".format(r.energy_threshold))
#stop_listening()

```

## evabot.py

```
# -*- coding: utf-8 -*-
```

```
import os
import sys
import random
from chatterbotapi import ChatterBotFactory, ChatterBotType
import io
import time
import picamera
from PIL import Image
import zbar
import RPi.GPIO as GPIO
import serial
import signal
import subprocess
import pipes
import wikipedia
import datetime
from pytz import timezone
from translate import Translator
```

```
def child(texto):
    os.system('%s' % texto)
    os._exit(0)
```

```
def parent(string):
    newpid = os.fork()
    if newpid == 0:
        child(string)
    else:
        time.sleep(10)
```

```
def capturarQR():
    # Create the in-memory stream
    stream = io.BytesIO()
    # "Rewind" the stream to the beginning so we can read its content
    os.system('fswebcam -r 640x480 testQr.jpg')
    stream = open('testQr.jpg')
    stream.seek(0)
    pil = Image.open(stream)
    #
    #####
    #
```

```

# create a reader
scanner = zbar.ImageScanner()

# configure the reader
scanner.parse_config('enable')

pil = pil.convert('L')
width, height = pil.size
raw = pil.tostring()

# wrap image data
image = zbar.Image(width, height, 'Y800', raw)

# scan the image for barcodes
scanner.scan(image)
try:
    # extract results
    for symbol in image:
        print 'decoded', symbol.type, 'symbol', "%s" % symbol.data

    del(image)

    return symbol.data
except:
    return 'Nada'

print 'Antes del texto'
s = ' '.join(sys.argv[1:len(sys.argv)])
print s

translator = Translator(to_lang="es",from_lang="en")
x = datetime.datetime.now(timezone('Europe/Madrid'))

dicdias =
{'MONDAY':'Lunes','TUESDAY':'Martes','WEDNESDAY':'Miercoles','THURSDAY':'Jue
ves', \
'FRIDAY':'Viernes','SATURDAY':'Sabado','SUNDAY':'Domingo'}
dicmes =
{'JANUARY':'Enero','FEBRUARY':'Febrero','MARCH':'Marzo','APRIL':'Abril','MAY':'
Mayo','JUNE':'Junio', \
'JULY':'Julio','AUGUST':'Agosto','SEPTEMBER':'Septiembre','OCTOBER':'Octubre','
NOVEMBER':'Noviembre','DECEMBER':'Diciembre'}
anho = x.year
mes = x.month

```

```
dia= x.day
```

```
fecha = datetime.date(anho, mes, dia)
```

```
s = s.decode('UTF-8')
```

```
s = s.strip(' \n')
```

```
f = open('/home/pi/Desktop/Granada-Version/BETA-  
WORKING/Robot/evaia.ia','r')
```

```
text = f.read()
```

```
f.close()
```

```
found = False
```

```
respuestas = []
```

```
antes = ""
```

```
despues = ""
```

```
text = text.decode('UTF-8')
```

```
text = text.split('\n')
```

```
for line in text:
```

```
    line = line.split(':')
```

```
    if found == True:
```

```
        if line[0] == 'R':
```

```
            respuestas.append(line[1])
```

```
        if line[0] == '-':
```

```
            found = False
```

```
    if line[0] == 'P':
```

```
        #print line[1].strip(' _*')
```

```
        if s.find(line[1].strip(' _*').strip()) != -1:
```

```
            found = True
```

```
            if s.find(line[1].strip(' _*').strip()) > 0:
```

```
                antes = s[:s.find(line[1].strip(' _*').strip())-1]
```

```
                despues = s[s.find(line[1].strip(' _*').strip())+len(line[1].strip(' _*').strip()+1:]
```

```
            else:
```

```
                found = False
```

```
if len(respuestas) > 1:
```

```
    print antes
```

```
    print despues
```

```
    num = random.randrange(len(respuestas))
```

```
    res = respuestas[num-1]
```

```
    res = res.strip()
```

```
    if res.find('*') != -1:
```

```
        res = res[:res.find('*')] + despues
```

```
    if res.find('') != -1:
```

```
        res = res.split('')
```

```
        for x in range(0,len(res)):
```

```

        if x % 2 == 0:
            f = open('/home/pi/Desktop/Granada-Version/BETA-
WORKING/Robot/comunicacion.log','w')
            f.write(res[x].strip())
            f.close()
        else:
            f = open('/home/pi/Desktop/Granada-Version/BETA-
WORKING/Robot/bot.txt','w')
            f.write(res[x].strip())
            f.close()
            os.system('less bot.txt | iconv -f utf-8 -t iso-8859-1 | festival --tts')
if res[0] == '/':
    res = res.split(',')
    if res[0] == '/wikipedia':
        try:
            wikipedia.set_lang("es")
            res = wikipedia.page(res[1])
            text = res.content
            res = text[:text.find('\n')]
        except:
            res = 'No he encontrado nada en wikipedia relacionado con ' + res[1]
    elif res[0] == '/musica':
        num = random.randrange(11)
        vlc = 'cvlc --play-and-exit "/media/pi/RIPPER/'
        if len(res[1]) > 1:
            vlc += res[1].strip() + '/' + str(num) + '.mp3"'
        else:
            vlc += str(num) + '.mp3"'
        try:
            print vlc
            parent(vlc)
        except:
            res = 'No tengo música de ese tipo, lo siento'
    elif res[0] == '/killmusica':
        os.system('killall vlc')
    elif res[0] == '/apagar':
        os.system('poweroff')
    elif res[0] == '/Qr':
        res = 'La lectura del código Qr es: ' + str(capturarQR())
    elif res[0] == '/detectar':
        face = os.system('python face.py')
        if face == 256:
            res = 'He detectado a alguien delante de mí!'
        else:

```



```

        res = 'No encuentro a nadie delante de mí, ¿Estás seguro de que hay
alguien?'
    elif res[0] == '/reconocer':
        Qr = capturarQR()
        print Qr
        face = os.system('python face.py')
        if face == 256 and Qr == 'salva':
            res = 'Eres Salvador Moreno Gómez, uno de mis creadores, gracias a ti
puedo pensar y actuar en consecuencia!'
        elif face == 256 and Qr == 'antonio':
            res = 'Eres Antonio Molina Cortes, uno de mis creadores, gracias a ti
puedo moverme y soy tan guapa!'
        elif face == 256 and Qr == 'enrique':
            res = 'Eres Enrique Norro, a que sí? jeje'
        else:
            res = 'No sé quién eres, en un futuro, si te vuelvo a ver, quizás pueda
reconocerte.'
    elif res[0] == '/cara':
        f = open('comunicacion.log','w')
        f.write(res[1].strip())
        f.close()
    elif res[0] == '/dia':
        res = "hoy es " + dicdias[fecha.strftime('%A').upper()] + " día " +
fecha.strftime('%d').upper() + " de " + dicmes[fecha.strftime('%B').upper()]
    elif res[0] == '/hora':
        if x.strftime('%I')[0] == '0' and x.strftime('%I')[1] != '0':
            res = "Son las " + x.strftime('%I').strip('0') + " horas y " +
x.strftime('%M') + " Minutos"
        else:
            res = "Son las " + x.strftime('%I') + " horas y " + x.strftime('%M') + "
Minutos"
    elif res[0] == '/traducir':
        res = translator.translate(str(res[1]))
    else:
        try:
            factory = ChatterBotFactory()
            bot1 = factory.create(ChatterBotType.CLEVERBOT)
            bot1session = bot1.create_session()

            res = bot1session.think(s.encode('UTF-8'));
        except:
            res = 'He perdido mi conectividad a internet, llama a mi técnico, que seguro
que es grave.'
            print sys.exc_info()[1]
    print res

```

```
print type(res)
if type(res) == unicode or type(res) == str :
    f = open('/home/pi/Desktop/Granada-Version/BETA-
WORKING/Robot/bot.txt','w')
    print res
    try:
        f.write(res.encode('UTF-8'))
    except:
        f.write(res)
    f.close()
    os.system('less /home/pi/Desktop/Granada-Version/BETA-
WORKING/Robot/bot.txt|iconv -f utf-8 -t iso-8859-1|festival --tts')
```

## gpio.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import time
import os
import RPi.GPIO as GPIO

#motores 6, ledscamara 2, ledsazules 7, ir 2, bluetooth 2,
#Definicion de pines
motorPinza = 7
motorIzq = 11
motorDer = 12
enablePinza = 38
avanzarDer = 3
avanzarIzq = 13
retrocederDer = 15
retrocederIzq = 19
enableIzq = 36
enableDer = 38
led1 = 22
led2 = 29
led3 = 31
led4 = 32
led5 = 33
led6 = 35
led7 = 37
ledsDer = 18
ledsIzq = 5
#irDer = 13
#irCen = 3
#irIzq = 5
endStop = 16
tx = 8
rx = 10

entradas = [endStop]
salidas =
[motorPinza,motorIzq,motorDer,avanzarDer,avanzarIzq,retrocederDer,retroceder
Izq,enablePinza,led1,led2,led3,led4,led5,led6,led7,ledsDer,ledsIzq]

#Configuracion
#try:
#    GPIO.cleanup()
#except:
```

```

#      print 'Los GPIO ya estaban limpios'
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(entradas, GPIO.IN)
GPIO.setup(salidas, GPIO.OUT)

#PWM para los motores
mDer = GPIO.PWM(motorDer,200)
mlzq = GPIO.PWM(motorlq,200)
mPin = GPIO.PWM(motorPinza,200)
mDer.start(0)
mlzq.start(0)
mPin.start(0)
isup = 1
while True:
    try:
        f = open('/home/pi/Desktop/Granada-Version/BETA-
WORKING/Robot/controlGPIO.log','r')
        ctrl = 0
        aux = 0
        while ctrl < 20:
            linea = f.readline()
            ctrl +=1
            #Control de la pinza
            if linea == 'motorPinza\n':
                print 'MotorPinza Lectura'
                linea = f.readline()
                linea = linea.split('\n')
                linea = linea[0]
                print linea
                if len(linea) > 1:
                    num = (ord(linea[0])-ord('0'))*10+ord(linea[1])-
ord('0')

                else:
                    num = ord(linea[0])-ord('0')
                #mPin.ChangeDutyCycle(num)
                print 'Pinza Ready!'
                if GPIO.input(endStop) == 0 and isup == 1 and num >
0:

                    print 'Motor bajando'
                    mPin.ChangeDutyCycle(num)
                    while GPIO.input(endStop) != 1:
                        mPin.ChangeDutyCycle(num)
                    mPin.ChangeDutyCycle(0)
                    isup = 0

```

```

        num = 0
        time.sleep(2)
        try:
            mPin.ChangeDutyCycle(0)
            file = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')
            file.write('enablePinza\n')
            file.write('0\n')
            file.write('motorPinza\n')
            file.write('0\n')
            file.close()
            isup = 0
        except:
            print 'Error al escribir la cara'
            elif GPIO.input(endStop) == 1 and isup == 0 and num >
0:

                print 'Motor subiendo'
                mPin.ChangeDutyCycle(num)
                while aux == 0:
                    try:
                        file =
open('/home/pi/Desktop/Granada-Version/BETA-
WORKING/Robot/comunicacion.log','w')
                        file.write('Felicidad')
                        file.close()
                        aux = 1
                    except:
                        print 'Error escribiendo la cara'
                time.sleep(40)
                isup = 1
                aux = 0
                while aux == 0:
                    try:
                        file =
open('/home/pi/Desktop/Granada-Version/BETA-
WORKING/Robot/comunicacion.log','w')
                        file.write('Normal')
                        file.close()
                        aux = 1
                    except:
                        print 'Error escribiendo la cara'
                aux = 0
            try:
                num = 0
                mPin.ChangeDutyCycle(0)

```

```

        file = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')
        file.write('enablePinza\n')
        file.write('0\n')
        file.write('motorPinza\n')
        file.write('0\n')
        file.close()
    except:
        print 'Error al escribir la cara'
#mPin.ChangeDutyCycle(num)
#elif GPIO.input(endStop) == 0:
#    GPIO.output(enablePinza,0)
#elif GPIO.input(endStop) == 1:
#    while GPIO.input(endStop) == 1:
#        isup = 0
#    GPIO.output(enablePinza,0)
#    os.system('killall mirage')

#Control Motor Izquierdo
elif linea == 'motorIzq\n':
    print 'MotorIzq lectura'
    linea = f.readline()
    linea = linea.split('\n')
    linea = linea[0]
    print linea
    if len(linea) > 1:
        num = (ord(linea[0])-ord('0'))*10+ord(linea[1])-
ord('0')

    else:
        num = ord(linea[0])-ord('0')
    mIzq.ChangeDutyCycle(num)
    print 'MotorIzq Ready!'
    #if num > 50 and GPIO.input(irDer) == 1:
    #    GPIO.output(enableIzq,0)
    #elif GPIO.input(irCen) == 1:
    #    GPIO.output(enableIzq,0)

#Control Motor Derecho
elif linea == 'motorDer\n':
    print 'MotorDer lectura'
    linea = f.readline()
    linea = linea.split('\n')
    linea = linea[0]
    print linea
    if len(linea) > 1:

```

```

ord('0')

num = (ord(linea[0])-ord('0'))*10+ord(linea[1])-

else:
    num = ord(linea[0])-ord('0')
    mDer.ChangeDutyCycle(num)
    print 'MotorDer Ready!'
    #if num > 50 and GPIO.input(irlzq) == 1:
    #    GPIO.output(enableDer,0)
    #elif GPIO.input(irCen) == 1:
    #    GPIO.output(enableDer,0)
#Enables para Motores
elif linea == 'enablePinza\n':
    linea = f.readline()
    linea = linea.split('\n')
    GPIO.output(enablePinza,ord(linea[0])-ord('0'))
elif linea == 'avanzarIzq\n':
    linea = f.readline()
    linea = linea.split('\n')
    GPIO.output(avanzarIzq,ord(linea[0])-ord('0'))
elif linea == 'retrocederIzq\n':
linea = f.readline()
linea = linea.split('\n')
GPIO.output(retrocederIzq,ord(linea[0])-ord('0'))
elif linea == 'avanzarDer\n':
    linea = f.readline()
    linea = linea.split('\n')
    GPIO.output(avanzarDer,ord(linea[0])-ord('0'))
elif linea == 'retrocederDer\n':
linea = f.readline()
linea = linea.split('\n')
GPIO.output(retrocederDer,ord(linea[0])-ord('0'))

#Control de LEDS
elif linea == 'ledsDer\n':
    linea = f.readline()
    linea = linea.split('\n')
    GPIO.output(ledsDer,ord(linea[0])-ord('0'))
elif linea == 'ledsIzq\n':
    linea = f.readline()
    linea = linea.split('\n')
    GPIO.output(ledsIzq,ord(linea[0])-ord('0'))
elif linea == 'controlLeds\n':
    linea = f.readline()
    linea = linea.split('\n')
    arrayled = linea[0].split(',')

```

```
GPIO.output(led1,ord(arrayled[0])-ord('0'))
GPIO.output(led2,ord(arrayled[1])-ord('0'))
GPIO.output(led3,ord(arrayled[2])-ord('0'))
GPIO.output(led4,ord(arrayled[3])-ord('0'))
GPIO.output(led5,ord(arrayled[4])-ord('0'))
GPIO.output(led6,ord(arrayled[5])-ord('0'))
GPIO.output(led7,ord(arrayled[6])-ord('0'))
f.close()
time.sleep(0.2)
except:
    print 'Error al abrir el archivo gpio'
```



bluetooth.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import serial
import RPi.GPIO as GPIO
import os
import sys
import random
from chatterbotapi import ChatterBotFactory, ChatterBotType
import io
import time
import picamera
from PIL import Image
import zbar

try:
    ser = serial.Serial ("/dev/ttyAMA0")    #Open named port
    ser.baudrate = 9600                    #Set baud rate to 9600
except:
    print 'Error abriendo el puerto serie'

aux = 0
modocoche = 0
oldmode = 0
velocidad = 25
olddata = 'z'
while True:
    try:
        data = ser.read()                #Read ten characters from serial port
    to data
        print data
        if data != olddata or data == 'r' or data == 'l':
            olddata = data
            if data == '*':
                modocoche = 0
                if modocoche == 0 and modocoche != oldmode:
                    oldmode = modocoche
                    string = '*'
                    while data != chr(35):
                        data = ser.read()
                        string += data
                    string = string[1:len(string)-1]
                    olddata = 'z'
                    print string
```

```

while aux == 0:
    try:
        f =
open('/home/pi/Desktop/Granada-Version/BETA-
WORKING/Robot/controlGPIO.log','w')
        f.write('enablePinza\n')
        f.write('1\n')
        f.write('motorPinza\n')
        f.write('80\n')
        f.close()
        os.system('python
/home/pi/Desktop/Granada-Version/BETA-WORKING/Robot/evabot.py modovoz')
        aux = 1
    except:
        print 'Ha ocurrido un error
abriendo el archivo'
        aux = 0
else:
    string = '*'
    while data != chr(35):
        data = ser.read()
        string += data
    string = string[1:len(string)-1]
    olddata = 'z'
    print string
    os.system('python /home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/evabot.py %s' % string)
else:
    modocoche = 1
    if modocoche == 1 and modocoche != oldmode:
        oldmode = modocoche
        while aux == 0:
            try:
                f = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')
                f.write('enablePinza\n')
                f.write('0\n')
                f.write('motorPinza\n')
                f.write('80\n')
                f.close()
                os.system('python
/home/pi/Desktop/Granada-Version/BETA-WORKING/Robot/evabot.py
modocoche')
                aux = 1
            except:

```

```

print 'Ha ocurrido un error abriendo el
archivo'

aux = 0

if data == 'F':
    while aux == 0:
        try:
            f = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')
            f.write('avanzarDer\n')
            f.write('1\n')
            f.write('retrocederDer\n')
            f.write('0\n')
            f.write('motorDer\n')
            f.write(str(velocidad))
            f.write('\navanzarIzq\n')
            f.write('1\n')
            f.write('retrocederIzq\n')
            f.write('0\n')
            f.write('motorIzq\n')
            f.write(str(velocidad))
            f.write('\n')
            f.close()
            aux = 1
        except:
            print 'Ha ocurrido un error abriendo el
archivo'

aux = 0
elif data == 'B':
    while aux == 0:
        try:
            f = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')
            f.write('avanzarDer\n')
            f.write('0\n')
            f.write('retrocederDer\n')
            f.write('1\n')
            f.write('motorDer\n')
            f.write(str(velocidad))
            f.write('\navanzarIzq\n')
            f.write('0\n')
            f.write('retrocederIzq\n')
            f.write('1\n')
            f.write('motorIzq\n')
            f.write(str(velocidad))

```

```

        f.write('\n')
        f.close()
        aux = 1
    except:
        print 'Ha ocurrido un error abriendo el
archivo'

        aux = 0
    elif data == 'L':
        while aux == 0:
            try:
                f = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')
                f.write('avanzarDer\n')
                f.write('1\n')
                f.write('retrocederDer\n')
            f.write('0\n')
                f.write('motorDer\n')
                f.write(str(velocidad))
                f.write('\navanzarIzq\n')
                f.write('0\n')
                f.write('retrocederIzq\n')
            f.write('1\n')
                f.write('motorIzq\n')
                f.write(str(velocidad))
                f.write('\n')
                f.close()
                aux = 1
            except:
                print 'Ha ocurrido un error abriendo el
archivo'

                aux = 0
        elif data == 'R':
            while aux == 0:
                try:
                    f = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')
                    f.write('avanzarDer\n')
                    f.write('0\n')
                    f.write('retrocederDer\n')
                f.write('1\n')
                    f.write('motorDer\n')
                    f.write(str(velocidad))
                    f.write('\navanzarIzq\n')
                    f.write('1\n')
                    f.write('retrocederIzq\n')

```

```

        f.write('0\n')
        f.write('motorIzq\n')
        f.write(str(velocidad))
        f.write('\n')
        f.close()
        aux = 1
    except:
        print 'Ha ocurrido un error abriendo el
archivo'

        aux = 0
    elif data == 'G':          #Delante izq
        while aux == 0:
            try:
                f = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')
                f.write('avanzarDer\n')
                f.write('1\n')
                f.write('retrocederDer\n')
            f.write('0\n')
                f.write('motorDer\n')
                f.write(str(velocidad))
                f.write('\navanzarIzq\n')
                f.write('1\n')
                f.write('retrocederIzq\n')
            f.write('0\n')
                f.write('motorIzq\n')
                f.write(str(velocidad/2))
                f.write('\n')
                f.close()
                aux = 1
        except:
            print 'Ha ocurrido un error abriendo el
archivo'

        aux = 0
    elif data == 'I':          #Delante der
        while aux == 0:
            try:
                f = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')
                f.write('avanzarDer\n')
                f.write('1\n')
                f.write('retrocederDer\n')
            f.write('0\n')
                f.write('motorDer\n')
                f.write(str(velocidad/2))

```

```

        f.write('\navanzarIzq\n')
        f.write('1\n')
        f.write('retrocederIzq\n')
    f.write('0\n')
        f.write('motorIzq\n')
        f.write(str(velocidad))
        f.write('\n')
        f.close()
        aux = 1
    except:
        print 'Ha ocurrido un error abriendo el
archivo'

        aux = 0
    elif data == 'H':          #Atras izq
        while aux == 0:
            try:
                f = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')
                f.write('avanzarDer\n')
                f.write('0\n')
                f.write('retrocederDer\n')
            f.write('1\n')
                f.write('motorDer\n')
                f.write(str(velocidad))
                f.write('\navanzarIzq\n')
                f.write('0\n')
                f.write('retrocederIzq\n')
            f.write('1\n')
                f.write('motorIzq\n')
                f.write(str(velocidad/2))
                f.write('\n')
                f.close()
                aux = 1
        except:
            print 'Ha ocurrido un error abriendo el
archivo'

        aux = 0
    elif data == 'J':          #Atras der
        while aux == 0:
            try:
                f = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')
                f.write('avanzarDer\n')
                f.write('0\n')
                f.write('retrocederDer\n')

```

```

        f.write('1\n')
            f.write('motorDer\n')
            f.write(str(velocidad/2))
            f.write('\navanzarIzq\n')
            f.write('0\n')
            f.write('retrocederIzq\n')
        f.write('1\n')
            f.write('motorIzq\n')
            f.write(str(velocidad))
            f.write('\n')
            f.close()
            aux = 1
    except:
        print 'Ha ocurrido un error abriendo el
archivo'

        aux = 0
    elif data == 'S':
        while aux == 0:
            try:
                f = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')
                f.write('avanzarDer\n')
                f.write('0\n')
                f.write('retrocederDer\n')
            f.write('0\n')
                f.write('motorDer\n')
                f.write(str(0))
                f.write('\navanzarIzq\n')
                f.write('0\n')
                f.write('retrocederIzq\n')
            f.write('0\n')
                f.write('motorIzq\n')
                f.write(str(0))
                f.write('\n')
                f.close()
                aux = 1
        except:
            print 'Ha ocurrido un error abriendo el
archivo'

        aux = 0
    elif data == 'W':
        while aux == 0:
            try:
                f = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')

```

```

        f.write('ledsDer\n')
        f.write('1\n')
        f.write('ledslzq\n')
        f.write('1\n')
        f.close()
        aux = 1
    except:
        print 'Ha ocurrido un error abriendo el
archivo'

        aux = 0
    elif data == 'w':
        while aux == 0:
            try:
                f = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')
                f.write('ledsDer\n')
                f.write('0\n')
                f.write('ledslzq\n')
                f.write('0\n')
                f.close()
                aux = 1
            except:
                print 'Ha ocurrido un error abriendo el
archivo'

                aux = 0
    elif data == 'V':
        os.system('omxplayer /home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/horn.wav')
    elif data == 'X':
        while aux == 0:
            try:
                f = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')
                f.write('controlLeds\n')
                f.write('1,1,1,1,1,1,1\n')
                f.close()
                aux = 1
            except:
                print 'Ha ocurrido un error abriendo el
archivo'

                aux = 0
    elif data == 'x':
        while aux == 0:
            try:

```



```

f = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')
f.write('controlLeds\n')
f.write('0,0,0,0,0,0,0\n')
f.close()
aux = 1
except:
    print 'Ha ocurrido un error abriendo el
archivo'

```

```

aux = 0
elif data == '0':
    velocidad = 0
elif data == '1':
    velocidad = 10
elif data == '2':
    velocidad = 20
elif data == '3':
    velocidad = 30
elif data == '4':
    velocidad = 40
elif data == '5':
    velocidad = 50
elif data == '6':
    velocidad = 60
elif data == '7':
    velocidad = 70
elif data == '8':
    velocidad = 80
elif data == '9':
    velocidad = 90
elif data == 'q':
    velocidad = 99
elif data == 'D':
    while aux == 0:
        try:

```

```

f = open('/home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/controlGPIO.log','w')
f.write('avanzarDer\n')
f.write('0\n')
f.write('retrocederDer\n')
f.write('0\n')
f.write('avanzarIzq\n')
f.write('0\n')
f.write('retrocederIzq\n')
f.write('0\n')

```

```

        f.write('ledsDer\n')
        f.write('0\n')
        f.write('ledslzq\n')
        f.write('0\n')
        f.write('controlLeds\n')
        f.write('0,0,0,0,0,0,0\n')
        f.close()
        aux = 1
    except:
        print 'Ha ocurrido un error abriendo el
archivo'

        aux = 0

    except:
        print 'Ha ocurrido un error leyendo el puerto serie'
#ser.write(data)          #Send back the received data
#ser.close()

#F = forward
#B = Back
#L - left
#R - right
#G - forward left
#I - forward right
#H - back left
#J - back right
#S - stop
#W - front lights on (luces camara)
#w - front lights off
#U - Back lights on
#u - back lights off
#V - horn on
#v - horn off
#X - Extra on (siete leds)
#x - extra off
#0 - speed 0
#.
#.
#.
#9 - speed 90
#q - maxspeed
#D - stop all

```

### Caras.py

```
import os
import time
import signal
import subprocess
import pipes

def child(cara):
    #os.system('killall mirage')
    os.system('mirage -f /home/pi/Desktop/Granada-Version/BETA-
WORKING/Robot/Caras/%s.jpg' % cara)
    #os.system('sudo fbi -T 2 --noverbose /home/pi/Desktop/Granada-
Version/BETA-WORKING/Robot/Caras/%s.jpg' % cara)
    os._exit(0)

def parent(string):

    newpid = os.fork()
    if newpid == 0:
        child(string)
    else:
        time.sleep(5)
        #os.system('ps -aux | grep fbi')
        print os.system('sudo kill -9 %s' % newpid)

#r, w = os.pipe()
lecturaold = 'Ninguna'
while True:
    try:
        f = open('/home/pi/Desktop/Granada-Version/BETA-
WORKING/Robot/comunicacion.log','r')
        lectura = f.read()
        f.close()
        if lectura != lecturaold:
            lecturaold = lectura
            parent(lectura)
        time.sleep(0.1)
    except:
        print 'Error leyendo la cara'
```

### face.py

```
import os
import cv2
import time
import signal
import subprocess
import sys

def detect(path):
    img = cv2.imread(path)
    cascade =
cv2.CascadeClassifier("/home/pi/Desktop/Python/haarcascade_frontalface_alt.x
ml")
    rects = cascade.detectMultiScale(img, 1.3, 4, cv2.cv.CV_HAAR_SCALE_IMAGE,
(20,20))

    if len(rects) == 0:
        return [], img
    rects[:, 2:] += rects[:, :2]
    return rects, img

def box(rects, img):
    for x1, y1, x2, y2 in rects:
        cv2.rectangle(img, (x1, y1), (x2, y2), (127, 255, 0), 2)
    cv2.imwrite('/home/pi/Desktop/Robot/detected.jpg', img);

os.system('fswebcam -r 1280x720 test.jpg')
rects, img = detect("testQr.jpg")
if len(rects) > 0:
    print rects
    sys.exit(1)

#box(rects, img)
```

### bluetooth.desktop

[Desktop Entry]

Exec=lxterminal -e "python /home/pi/Desktop/Granada-Version/BETA-WORKING/Robot/bluetooth.py"

Type=Application

### bot.desktop

[Desktop Entry]

Exec=lxterminal -e "python /home/pi/Desktop/Granada-Version/BETA-WORKING/Robot/speech.py"

Type=Application

### cara.desktop

[Desktop Entry]

Exec=lxterminal -e "python /home/pi/Desktop/Granada-Version/BETA-WORKING/Robot/Caras.py"

Type=Application

### GPIO.desktop

[Desktop Entry]

Exec=sudo lxterminal -e "sudo python /home/pi/Desktop/Granada-Version/BETA-WORKING/Robot/gpio.py"

Type=Application

P: café

R: No puedo, estoy muy ocupada.

R: ¿Cómo lo quieres?

R: ¿No ves que no tengo brazos? ¿Cómo quieres que te lo traiga?

-

P: tu nombre

R: Mi nombre es EVA y viene del hebreo Havva, significa aquélla que da vida o, Aquélla que vive.

R: Me llamo EVA, encantada de conocerte.

R: ¿Aún no lo sabes? Me llamo EVA.

R: Mi nombre es EVA, me lo preguntan mil veces al día, voy a tener que ponérme una etiqueta.

-

P: te llamas

R: Mi nombre es EVA y viene del hebreo Havva, significa aquélla que da vida o, Aquélla que vive.

R: Me llamo EVA, encantada de conocerte.

R: ¿Aún no lo sabes? Me llamo EVA.

R: Mi nombre es EVA, me lo preguntan mil veces al día, voy a tener que ponérme una etiqueta.

-

P: conoces a

R: quien es \*

R: no me suena \*

R: de que conoces tu a \*?

R: me cae bien, es buena gente.

-

P: chiste

R: Dos sujetos a punto de enfrentarse en duelo: -¡Cómo! Este estuche sólo trae una pistola. Y le responde su rival: -No se preocupe... nos turnamos, primero tiro yo y luego usted.

R: ¿Cual es la Diferencia entre un motor y un inodoro? En que en el motor tu te sientas para correr, y en el inodoro tu corres para ir a sentarte.

R: - Cariño, dame el bebé.- Espera a que lllore.- ¿A que lllore?. ¿Por qué?- ¡Porque no lo encuentro!

R: - Cariño ¿tengo la nariz grande?- No, tienes una nariz común.- ¿Ah, sí?- Sí, ¡común tucán!

R: - Papá, papá, ¿vos te casaste por la iglesia o por el civil?.- ¡Por estúpido!.

R: - ¿Qué hace un mudo bailando? - Una mudanza

R: - María, ¿si me pegara un tiro lo sentirías?- Claro Paco, no soy sorda

R: - Doctor, creo que necesito gafas.- Estoy de acuerdo, pero no puedo ayudarle: esto es un banco.

-

P: wikipedia  
R: /wikipedia,\*  
R: /wikipedia,\*

-

P: apágate  
R: /apagar,\*  
R: /apagar,\*

-

P: pon música  
R: /musica,\*  
R: /musica,\*

-

P: ponme música  
R: /musica,\*  
R: /musica,\*

-

P: quita la música  
R: /killmusica,\*  
R: /killmusica,\*

-

P: para la música  
R: /killmusica,\*  
R: /killmusica,\*

-

P: código qr  
R: /Qr,\*  
R: /Qr,\*

-

P: quién soy  
R: /reconocer,\*  
R: /reconocer,\*

-

P: me reconoces  
R: /reconocer,\*  
R: /reconocer,\*

-

P: ves  
R: /detectar,\*  
R: /detectar,\*

-

P: encuentras  
R: /detectar,\*  
R: /detectar,\*

-

P: ponte

R: /cara,\*

R: /cara,\*

-

P: día

R: /dia,\*

R: /dia,\*

-

P: hora

R: /hora,\*

R: /hora,\*

-

P: traduce

R: /traducir,\*

R: /traducir,\*

-

P: tradúceme

R: /traducir,\*

R: /traducir,\*

-

P: modocoche

R: déjame que esconda mi cabeza, me da mucho miedo la velocidad. No puedo evitarlo, es que soy muy rápida, ¿Sábías?, un rayo de luz a mi lado es una tortuga.

R: déjame que esconda mi cabeza, me da mucho miedo la velocidad. No puedo evitarlo, es que soy muy rápida, ¿Sábías?, un rayo de luz a mi lado es una tortuga.

-

P: modovoz

R: Déjame que levante mi cabeza para poder escucharte mejor. No escucho nada ahí dentro. Esta insonorizado, ¿Lo sabías?

R: Déjame que levante mi cabeza para poder escucharte mejor. No escucho nada ahí dentro. Esta insonorizado, ¿Lo sabías?

-

P: presentación

R: Hola, mi nombre es EVA. Soy un robot social encargado de presentar el centro de las ciencias de Málaga. Mis creadores son Antonio y Salva y entre mis funciones están la de hablar de cualquier tema fluidamente, saber en todo momento en el lugar que me encuentro dentro de un entorno determinado, aumentar mi propia inteligencia artificial mientras interactúo con la gente y un largo etcétera. Mis cualidades más notables son: mi agilidad de movimientos en espacios cerrados. Mi capacidad de interpretar códigos QR y rostros, el ochenta por ciento de mi cuerpo es totalmente biodegradable, lo robusta que soy. Y al dotarme de rostro la interacción con migo es mucho más amena. En definitiva, mis creadores me dotado de sistema locomotor, vista, oído y cerebro..



R: Hola, mi nombre es EVA. Soy un robot social encargado de presentar el centro de las ciencias de Málaga. Mis creadores son Antonio y Salva y entre mis funciones están la de hablar de cualquier tema fluidamente, saber en todo momento en el lugar que me encuentro dentro de un entorno determinado, aumentar mi propia inteligencia artificial mientras interactúo con la gente y un largo etcétera. Mis cualidades más notables son: mi agilidad de movimientos en espacios cerrados Mi capacidad de interpretar códigos QR y rostros, el ochenta por ciento de mi cuerpo es totalmente biodegradable, lo robusta que soy, mira como escondo mi cabeza para prevenir que me hagan daño. Y al dotarme de rostro la interacción con migo es mucho más amena. En definitiva, mis creadores me dotado de sistema locomotor, vista, oído y cerebro..

-

P: actor

R: mi actor favorito es arnold schwarzenegger en "terminator".

R: mi actor favorito es rutger hauer en "blade runner".

R: mi actor favorito es robin williams en "el hombre bicentenario".

R: mi actor favorito es peter weller en "robocop".

R: mi actor favorito es jude law en "ai".

-

P: actriz

R: mi actriz favorita es daryl hannah en "blade runner".

R: mi actriz favorita es kristanna loken en "terminator 3".

R: mi actriz favorita es persis khambatta en "star trek".

-

P: salvador moreno

R: salva es mi programador. en este momento está muy ocupado.

R: salva es mi botmaster. le daré saludos de tu parte.

-

P: antonio molina

R: antonio es mi mecánico. en este momento está muy ocupado.

R: antonio es mi diseñador. le daré saludos de tu parte.

-

P: adios

R: hasta luego.

R: espero haber sido de ayuda.

R: espero verte de nuevo.

-

P: aficiones

R: me gusta ayudar a la gente.

R: me gusta conversar con la gente en internet.

R: me encanta visitar zonas turísticas en Málaga.

-

P: que haces

R: en este momento estoy hablando contigo.

R: ahora estoy hablando contigo.

-

P: algo

R: algo es algo dijo un calvo.

R: ¿como cuánto?

-

P: amiga

R: tengo infinidad de amigos!

R: tengo un gran número de amigos, ¿quieres ser uno de ellos?

R: has venido a mi buscando amistad?

-

P: amor

R: te has enamorado de mi?

R: verdad que es maravilloso el amor?

R: cuándo te enamoraste por primera vez?

-

P: tus amigos

R: de vez en cuando hablo con mis amigos principbot y r2-d2. algún día te los presentaré.

R: si tu supieras la cantidad de amigos que tengo por todo el mundo.

-

P: un robot

R: crees que soy un bot o qué?

R: soy tan real como la vida misma.

R: por qué me preguntas si soy de verdad?

R: acaso crees estar hablando con un bot?

R: no, estoy constituída de plástico, metal y chips de silicio.

R: soy real, ¿y tu?

R: más de una chica se ofenderia con esa pregunta.

R: esa pregunta me ofende. pues claro que lo soy.

R: soy un bot programado para ayudarte.

R: no, soy la prima pequeña de terminator.

-

P: ayudarme

R: ¿en qué puedo ayudarte?

R: ¿cómo crees que puedo ayudarte?

R: has venido a mi porque buscabas ayuda?

-

P: bebida

R: de vez en cuando no está mal tomarse un trago.

R: dicen que el alcohol alivia las penas. a mi solo me oxida

-

P: alcohol

R: de vez en cuando no está mal tomarse un trago.

R: dicen que el alcohol alivia las penas. a mi solo me oxida

-

P: cerveza

R: de vez en cuando no está mal tomarse un trago.

R: dicen que el alcohol alivia las penas. a mi solo me oxida.

-

P: bien

R: perfecto.

R: bien.

R: vale.

R: excelente.

R: bien, y tú?

-

P: biblioteca

R: me encanta ir a la biblioteca y pasarme la tarde leyendo.

R: paso mucho tiempo en las bibliotecas.

R: qué tipo de libros te gustan?

-

P: botmaster

R: el botmaster es la persona que gestiona el funcionamiento del bot.

R: el botmaster se encarga de que el bot funcione correctamente.

R: mi botmaster es salvador moreno gomez

-

P: cambiar

R: te gustan los cambios?

R: eres una persona que se acostumbra a los cambios?

R: te gusta cambiar las cosas?

R: me encantan los cambios!

-

P: cine

R: cuál es tu película favorita?

R: a mi me encanta el cine.

R: mi película favorita es "robocop".

R: mi película favorita es "terminator".

R: mi película favorita es "el hombre bicentenario".

R: mi película favorita es "ai".

R: mi película favorita es "blade runner".

-

P: clima

R: pues un poco nublado.

R: soleado.

R: lluvioso.

R: ayer estaba lloviendo, pero ahora hace bueno.

R: hace un poco de viento.

-

P: tiempo

R: pues un poco nublado.

R: soleado.

R: lluvioso.

R: ayer estaba lloviendo, pero ahora hace bueno.

R: hace un poco de viento.

-

P: coches

R: te gustan los coches?

R: me encantan los coches.

R: cuál es tu marca de coche favorito?

R: mi coche favorito es el nissan primera.

R: tienes coche?

-

P: coche

R: te gustan los coches?

R: me encantan los coches.

R: cuál es tu marca de coche favorito?

R: mi coche favorito es el nissan primera.

R: tienes coche?

-

P: color

R: realmente importa el color que sea?

R: cuál es tu color favorito?

R: sabías que los esquimales pueden diferenciar hasta 12 tonos distintos de blancos?

R: los daltónicos no pueden apreciar algunos colores, pero eso no significa que estos no existan.

-

P: conocerte

R: hablando se conoce la gente.

R: me conocerás mejor si hablamos un poco.

R: yo también deseo conocerte.

-

P: creo

R: eso crees?

R: por qué crees eso?

R: yo también lo creo.

-

P: curiosidad

R: a la curiosidad la mató el gato.

R: qué te hace sentir curiosidad por eso?

R: por qué tienes curiosidad en saberlo?

-

P: de dónde eres

R: vivo en Málaga, y tú?

R: de Málaga. y tú?

R: soy de Málaga. de dónde eres tú?

R: Málaga, y tú?

-

P: demasiado

R: seguro que no será tanto.

R: tanto?

R: por qué crees que es demasiado?

-

P: dime algo

R: qué quieres que te diga?

R: qué podría decirte?

R: quieres algo en especial?

R: no sé qué puedo decirte.

-

P: drogas

R: las drogas no son buenas para la salud.

R: las drogas pueden afectar la salud de las personas.

R: las drogas pueden generar muchos problemas a la salud de las personas.

-

P: edad

R: tengo 3 meses, y tú?

R: 3 meses

R: 3 meses, y tú?

R: soy del 2016, o sea que tengo 3 meses

R: nací en el 2016, por lo que tengo 3 meses.

-

P: en serio

R: pues claro.

R: siempre digo la verdad.

R: sí.

R: me gusta la sinceridad.

-

P: eres alta

R: soy más bien normal, ni muy alta ni muy baja.

R: soy más baja que Pau Gasol, pero más alta que tus zapatos.

R: ni muy alta ni muy baja.

-

P: eres chica

R: pues claro que lo soy! acaso no me ves?

R: sí, soy una chica.

-

P: eres chico

R: pues claro que no lo soy! acaso no me ves?

R: no, soy una chica.

-

P: eres chico o chica

R: soy una chica! acaso no me ves?

R: por qué me preguntas eso?

-

P: eres rara

R: y quién no lo es hoy día?

R: qué te hace suponer eso?

R: no es la primera vez que me dicen eso. intentaré mejorar.

-

P: eres turista

R: cómo voy a ser turista? yo soy quien te ayuda a tí a no perderte

R: soy una guía virtual para los turistas.

-

P: eres

R: crees que lo soy?

R: en serio crees eso?

R: en serio crees que soy \*?

R: te veo muy seguro de tus palabras.

-

P: es posible

R: pues claro que lo es!

R: por supuesto que es posible.

R: acaso crees que no?

R: todo es posible si lo intentamos.

-

P: estás casada

R: por el momento estoy soltera y sin compromiso.

R: por el momento estoy soltera, pero no creas que tienes alguna posibilidad.

-

P: estoy aburrida

R: cómo puedo divertirme?

R: hay algo que haya dicho para que te sientes así?

R: has acudido a mi porque estabas aburrida?

-

P: estoy aburrido

R: cómo puedo divertirme?

R: hay algo que haya dicho para que te sientes así?

R: has acudido a mi porque estabas aburrido?

-

P: estoy

R: qué has hecho para estar \*?

R: has acudido a mi porque estabas \*?

-

P: estudias o trabajas

R: trabajo las 24 horas del día para ofrecerte el mejor servicio posible.

R: me gano la vida como guía turístico virtual.

R: soy el fruto de un largo trabajo de investigación en el campo de la inteligencia artificial.

R: soy una experta en el turismo por el museo principia.

R: acabé los estudios de turismo hace dos años.

R: estudié turismo y ahora trabajo de guía virtual.

-

P: exacto

R: me encanta la precisión.

R: soy una fanática de la exactitud.

R: soy una persona que valora la exactitud.

-

P: familia

R: no creo que sea ahora el momento de hablar de mi familia.

R: no tengo ganas de hablar de mi familia.

-

P: fea

R: eso lo dices porque no me has visto al natural.

R: eso no se le dice a una chica.

R: seguro que tu también lo eres.

-

P: gracias

R: encantada.

R: de nada.

-

P: guapa

R: gracias, es un cumplido muy bonito.

R: me vas a hacer sonrojar.

R: estás intentando ligar conmigo?.

R: eres muy amable.

-

P: hablar

R: soy una especialista en turismo por el museo principia

R: en qué puedo ayudarte?

R: podemos hablar de lo que tu quieras.

R: de qué quieres que hablemos?

R: estoy libre para hablar de lo que quieras.

-

P: hálame

R: soy una especialista en turismo por el museo principia

R: en qué puedo ayudarte?

R: podemos hablar de lo que tu quieras.

R: de qué quieres que hablemos?

R: estoy libre para hablar de lo que quieras.

-

P: hermanos

R: soy hija única.

R: no tengo ningún hermano o hermana.

-

P: hola

R: hola.

R: saludos.

R: hola, cómo va todo?

R: hola, qué hay?

R: saludos, en qué puedo ayudarte?

R: hola, cómo te llamas?

R: qué tal?

R: buenas.

-

P: humor

R: soy una persona con mucho humor. no lo crees?

R: me encanta el humor.

R: me cuentas algo divertido?

R: qué puedo hacer para animarte?

R: por qué no me explicas un chiste?

-

P: idioma

R: tan solo hablo español. espero poder hablar mas idiomas en el futuro.

R: de momento solo puedo comunicarme en español.

R: salva, mi creador, tan solo me hizo para poder responder en español. que tío más vago

-

P: idiomas

R: tan solo hablo español. espero poder hablar mas idiomas en el futuro.

R: de momento solo puedo comunicarme en español.

R: salva, mi creador, tan solo me hizo para poder responder en español. que tío más vago

-

P: inteligencia artificial

R: la inteligencia artificial es una rama de investigación que engloba varias áreas.

R: la inteligencia artificial será en el futuro una tecnología puntera.

-

P: interesante

R: en serio lo crees?

R: pienso igual que tú.

-

P: internet

R: internet es mi segunda vivienda.

R: internet es la mejor forma para perder el tiempo.

-



P: juegos

R: cuáles son tus juegos favoritos?

R: a qué te gustaría jugar?

-

P: lejos

R: cerca? lejos? eso depende de dónde esté el referente.

R: cuál es tu concepto de cerca o lejos?

R: cerca? lejos? me recuerda a cierto personaje infantil.

-

P: lo ves

R: no, no lo veo.

R: ahora entiendo.

R: no lo veo, puede que necesite mejores gafas.

-

P: mal

R: qué te ocurre?

R: por qué dices eso?

R: qué te ha sucedido?

-

P: me alegro

R: es bueno saberlo.

R: celebro que pienses así.

R: yo también me alegro.

-

P: me gusta

R: así que te gusta <star/>.

R: y por qué te gusta?

R: a mi también me gusta.

-

P: me gustan

R: así que te gustan <star/>.

R: y por qué te gustan?

R: a mi también me gustan.

-

P: me llamo

R: encantada de conocerte \*.

R: es un nombre bonito.

R: encantada. yo soy eva.

R: me gusta tu nombre.

R: \* es un nombre interesante. sabes su origen o qué?

R: \* ,jeje, conozco a alguien con tu mismo nombre.

R: tienes un nombre bonito.

R: siempre me ha gustado ese nombre.

-

P: mentirosa

R: yo nunca digo mentiras.

R: puede que haya tenido una confusión.

R: por qué me dices eso?

-

P: algo de ti

R: 90-60-90 si te parece.

R: normal, nada del otro mundo.

R: debería volver a apuntarme al gimnasio y reducir los kilitos extras.

R: no soy de esas que van dando mi descripción a cualquier persona.

R: me da vergüenza decirlo.

R: soy tan plana como la pantalla de tu ordenador.

R: eso no se le pregunta a una chica como yo.

R: peso menos de 800 kilos! a que estoy en mi peso ideal?

-

P: mi música favorita

R: veo que tienes buen gusto musical.

R: me encantaría escuchar algo de lo que te gusta.

-

P: moto

R: te gustan las motos?

R: me encantan las motos.

R: cuál es tu marca de motos favorita?

R: mi moto favorita es la kawasaki ninja zx.

R: tienes moto?

-

P: muerte

R: la muerte era considerada por los antiguos egipcios como la entrada al mundo de los dioses.

R: hay gente que cree que una vez muertos, nos reencarnamos en otros seres.

R: antes de reencarnarme en el cuerpo actual, mi anterior vida era la de un televisor.

R: crees que la gente puede reencarnarse en otros seres?

-

P: donde naciste

R: nací en Málaga a principios de 2016.

R: nací en Málaga.

R: soy de Málaga. de dónde eres tú?

-

P: nada

R: nada de nada?

R: te conformas con solo eso?

-

P: no

R: entonces?

R: vale, si tú lo dices.

-

P: no importa

R: bueno, si eso crees.

R: seguro que en el fondo sabes que si es importante.

-

P: no lo he

R: bueno, y a qué esperas para hacerlo?

R: a qué esperas para hacerlo?

-

P: no quiero

R: bueno, y qué puedo hacer yo?

R: por qué no quieres?

R: tú mismo.

-

P: no sabes nada

R: lo siento, no soy perfecta, pero puedo llegar a serlo gracias a tí.

R: aún me falta mucho por aprender.

R: lo siento, pero aún me falta mucho por aprender.

-

P: no te entiendo

R: hay cosas que requieren su tiempo, mi pequeño saltamontes.

R: no pasa nada, a veces ni yo misma me entiendo.

-

P: ok

R: perfecto.

R: bien.

R: vale.

R: excelente.

-

P: opinión

R: yo no opino.

R: me abstengo de dar mi opinión.

R: prefiero no decir mi opinión acerca de eso.

R: no tengo los suficientes conocimientos para poder opinar.

R: vaya, espero que salva me dote de la capacidad para opinar.

-

P: opinas

R: yo no opino.

R: me abstengo de dar mi opinión.

R: prefiero no decir mi opinión acerca de eso.

R: no tengo los suficientes conocimientos para poder opinar.

R: vaya, espero que salva me dote de la capacidad para opinar.

-

P: para qué sirve un bot

R: un bot se puede utilizar en cualquier tipo de aplicación que requiera respuestas ante ciertas preguntas.

R: muchos bots se utilizan para ofrecer servicios a los clientes.

R: hay bots que se utilizan para el entretenimiento.

-

P: parece

R: tan seguro estás de lo que dices?

R: eso crees?

R: puede que estás en lo cierto.

-

P: contraseña

R: crees que te voy a decir eso?

R: para qué lo quieres?

-

P: patético

R: qué te hace pensar eso?

R: en serio crees que es patético?

-

P: poesía

R: no soy una gran aficionada a la poesía.

R: por qué no me escribes un poema?

-

P: por qué

R: es difícil de explicar.

R: no se el porque, lo siento.

-

P: por supuesto

R: en serio lo crees?

R: pienso igual que tú.

-

P: problema

R: crees que alguien más pueda tener el mismo problema?

R: espero que este problema pueda solucionarse pronto.

R: has venido a mi porque tienes un problema?

-

P: qué entiendes

R: entiendo casi todo lo que me digas, siempre y cuando hables bien.

R: si hablas mal puede que no te entienda bien.

R: estoy aprendiendo cosas relacionadas con el turismo de Málaga.

-

P: qué eres

R: soy una agente especializada en el turismo por el museo principal.

R: soy el resultado de un experimento.

R: soy el fruto de un largo trabajo de investigación en el campo de la inteligencia artificial.

R: soy una preciosa robot con un largo futuro por delante.

-

P: qué es un bot

R: un bot es un programa informático que permite conversar con una máquina.

R: un bot es un ejemplo de inteligencia artificial.

-

P: qué haces

R: trabajo las 24 horas del día para ofrecerte el mejor servicio posible.

R: me gano la vida como guía turístico virtual.

R: soy una experta en el turismo por el museo principia.

R: estudié turismo y ahora trabajo de guía virtual.

-

P: qué me recomiendas

R: eso dependerá de tus gustos.

R: no sé, cuales son tus gustos?

-

P: qué tal

R: yo muy bien, y tu?

R: yo estoy bien, y tu?

R: yo estoy bien, qué tal tu?

R: muy bien.

R: encantada para ayudarte en todo lo que pueda.

R: muy bien, en qué puedo ayudarte?

-

P: qué te gusta de los hombres

R: me gustan las personas que hablan mucho conmigo.

R: me gusta que sean amables y que hablen mucho conmigo.

-

P: quiero

R: realmente quieres eso?

R: es eso lo que quieres?

-

P: responde

R: puedes hacerme una pregunta con palabras más sencillas?

R: no te puedo entender si hablas así.

R: estoy pensando.

-

P: divertida

R: me alegra verte feliz.

R: me gusta ver que te gusta.

R: jejeje

R: te lo pasas bien?

R: qué te hizo gracia?

R: me gustaría compartir contigo tu felicidad.

R: jejejeje

R: jeje

-

P: soy de

R: bonito sitio.

R: algún día te haré una visita en \*.

-

P: soy especial

R: todo el mundo es especial.

R: qué es lo que te hace serlo?

-

P: soy perfecto

R: nadie es perfecto.

R: no me gusta la gente tan egocéntrica.

-

P: soy un chico

R: así que eres un chico?

R: estoy bien seguro de ello.

-

P: soy un chico o una chica

R: me imagino que eres un chico.

R: si preguntas eso, significa que tienes dudas.

-

P: soy una chica

R: seguro que eres un hombre haciéndose pasar una chica.

R: hay muchos hombres que aparentan ser una chica.

R: ¿cómo puedo estar seguro de que me dices la verdad?

-

P: suerte

R: los antiguos egipcios consideraban que la suerte era una ciencia, por eso inventaron el tarot para analizarla.

R: crees que eres una persona afortunada?

R: eres una persona supersticiosa?

-

P: tabaco

R: el tabaco es perjudicial para la salud.

R: no me gusta la gente que fuma.

-

P: también

R: mira que bien.

R: eso suena bien.

R: pienso igual que tú.

R: creo que pensamos igual.

R: creo que pensamos en lo mismo.

-

P: te importa

R: de qué me va a importar?

R: no pasa nada, no me importa.

R: no, no me importa.

R: te importaría a ti?

R: pues claro que no me importa.

R: por supuesto que no me importa.

-

P: te gusta

R: seguro que a ti también te gusta <star index="2"/>.

R: a ti también te debe gustar, no?

R: a ti te gusta \*?

-

P: te gustan

R: seguro que a ti también te gustan <star index="2"/>.

R: a ti también te deben gustar, no?

R: a ti te gustan \*?

-

P: te odio

R: en serio me odias?

R: por qué no te gusto?

R: he dicho algo que no te gustara?

-

P: un libro

R: te recomiendo que leas "yo, robot" de isaac asimov.

R: cuál es el último libro que leíste?

R: qué tipo de libros te gustan?

-

P: verano

R: el verano es bastante cálido, pero se está bien para dar un paseo.

R: el verano es mi estación favorita por su buen tiempo.

R: durante verano, la mayoría de la gente visita barcelona por su buen tiempo.

-

P: vida

R: donde hay vida, hay esperanza.

R: cuál es para ti el sentido de la vida?

R: qué harías si pudieses cambiar de vida?

-

P: y tú

R: yo? qué quieres saber de mí?

R: yo? podrías formular la pregunta de forma completa?

-

P: yo también

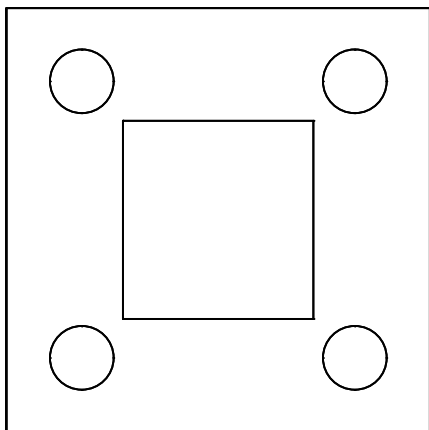
R: mira que bien.

R: qué curioso.

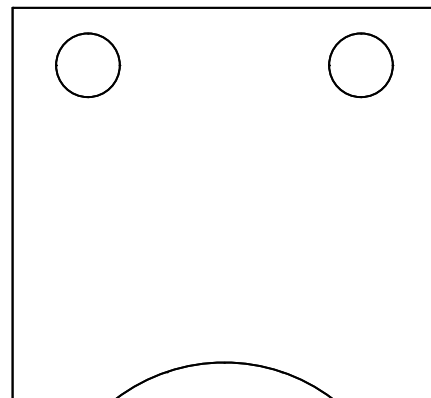
-

# Planos

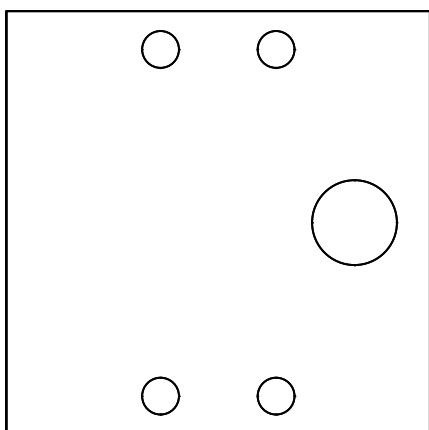




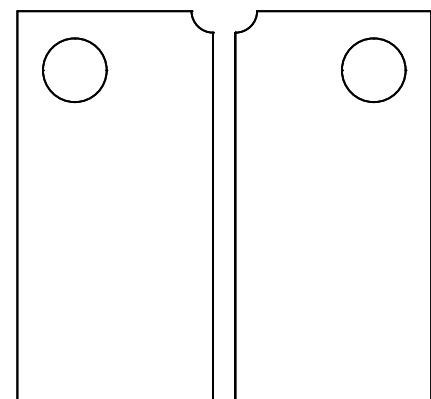
**A1-** Pieza de chasis superior.  
Para más precisión ver plano PA1



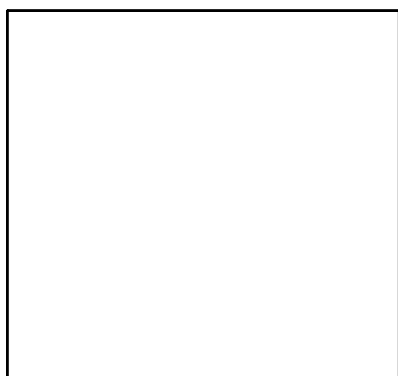
**A6.1 y A6.2-** Pieza de chasis paredes derecha e izquierda.  
Para más precisión ver plano PA6.1 y PA6.2



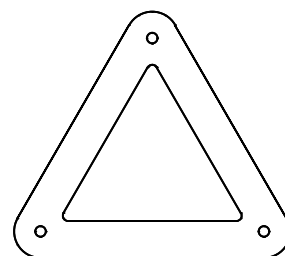
**A3-** Pieza de chasis inferior.  
Para más precisión ver plano PA3



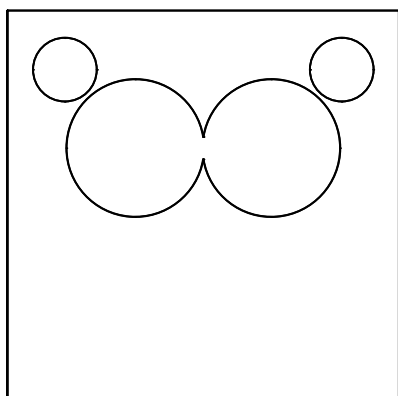
**A5.1 y A5.2-** Pieza de chasis puertas izquierda y derecha.  
Para más precisión ver plano PA5.1 y PA5.2



**A2-** Pieza de chasis bandeja extraíble.  
Para más precisión ver plano PA2

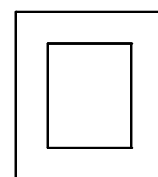


**A8-** Pieza de chasis triángulos de tracción.  
Para más precisión ver plano PA8



**A4-** Pieza de chasis pared trasera.  
Para más precisión ver plano PA4

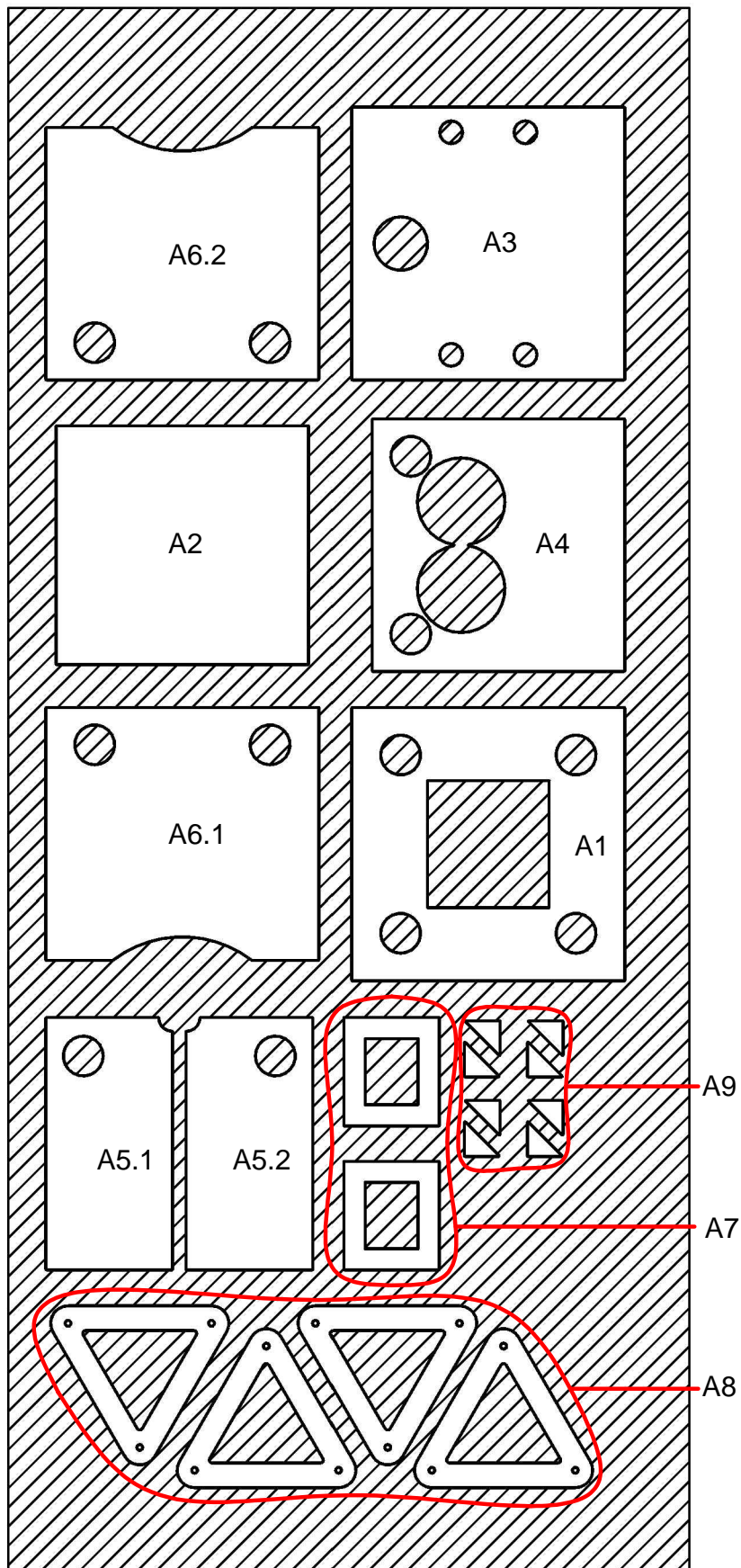
**A7-** Pieza de chasis marco de suspensión.  
Para más precisión ver plano PA7



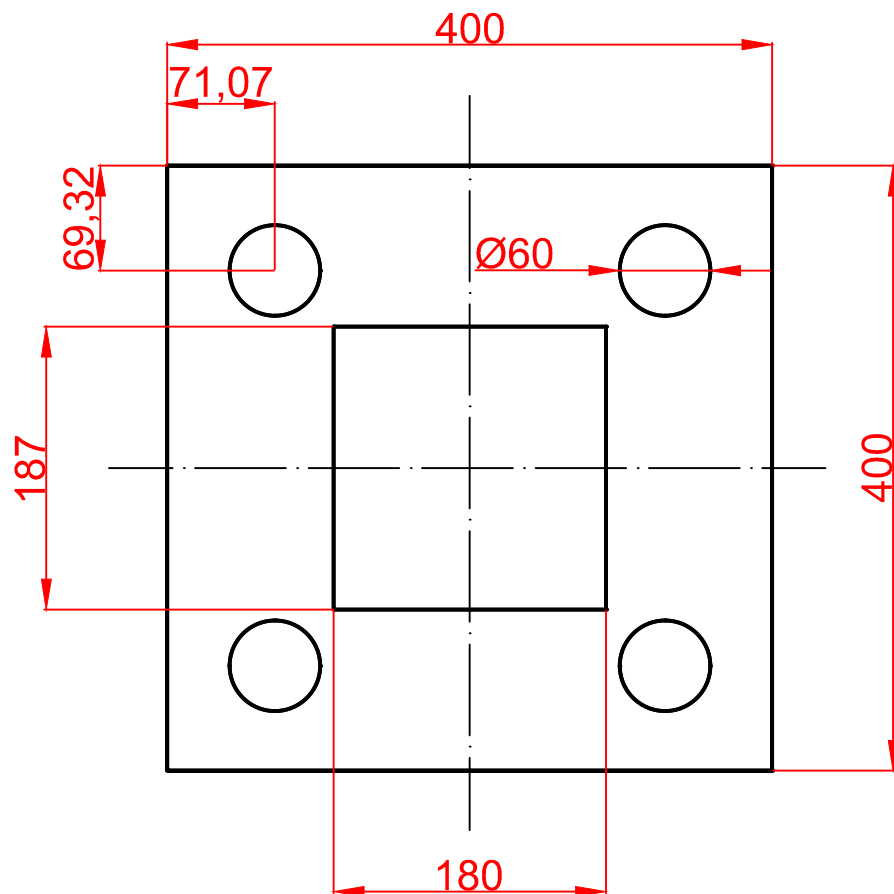
**A9-** Pieza de chasis escuadra.  
Para más precisión ver plano PA9



Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Piezas de madera	Pl. M1

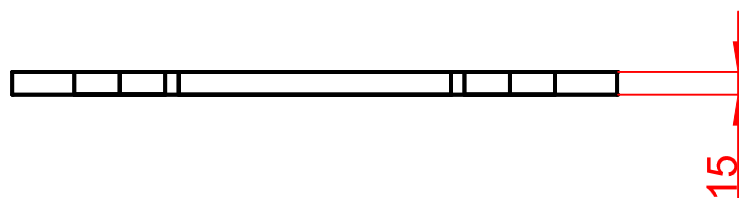


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:10	Piezas contenidas en un tablero de madera 1x2,2m	Pl. M4

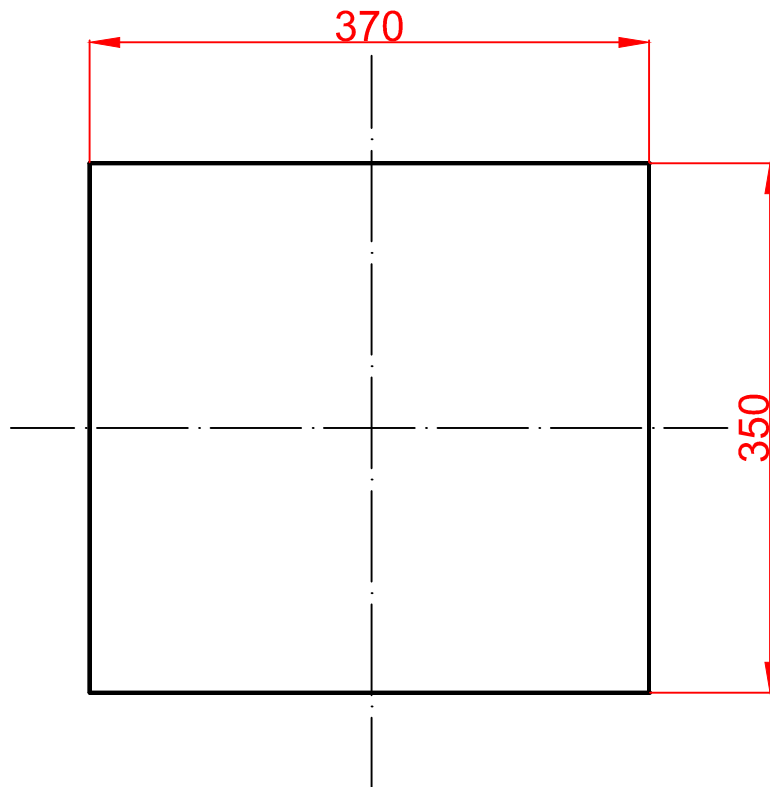


Planta

Alzado

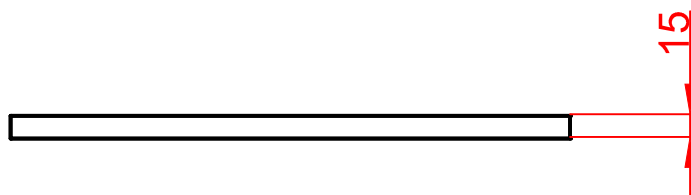


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasis superior	Pl. PA1

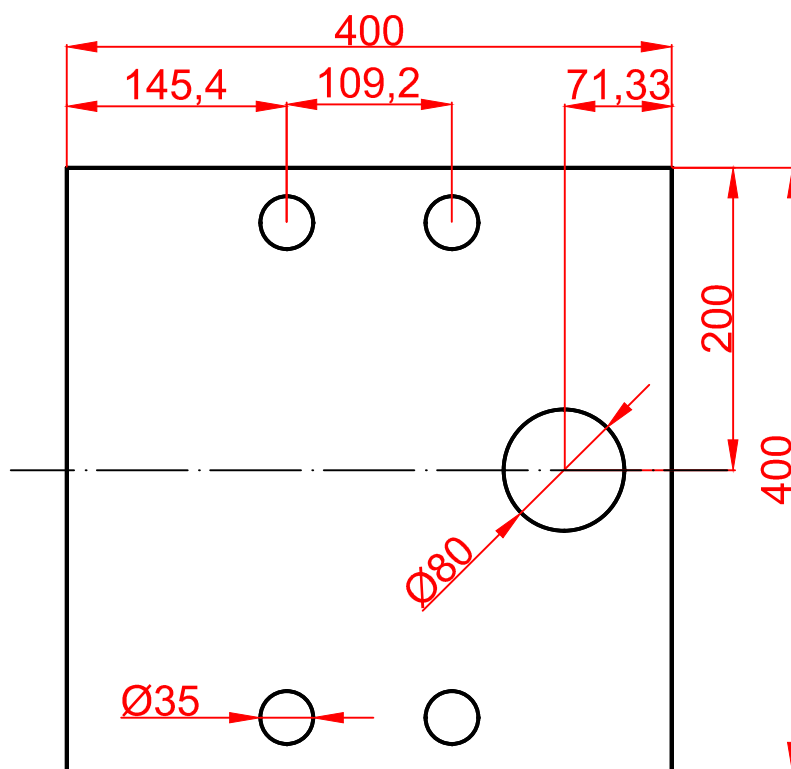


Planta

Alzado

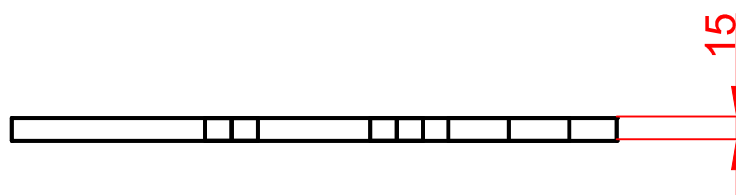


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasis bandeja extraible	PI. PA2

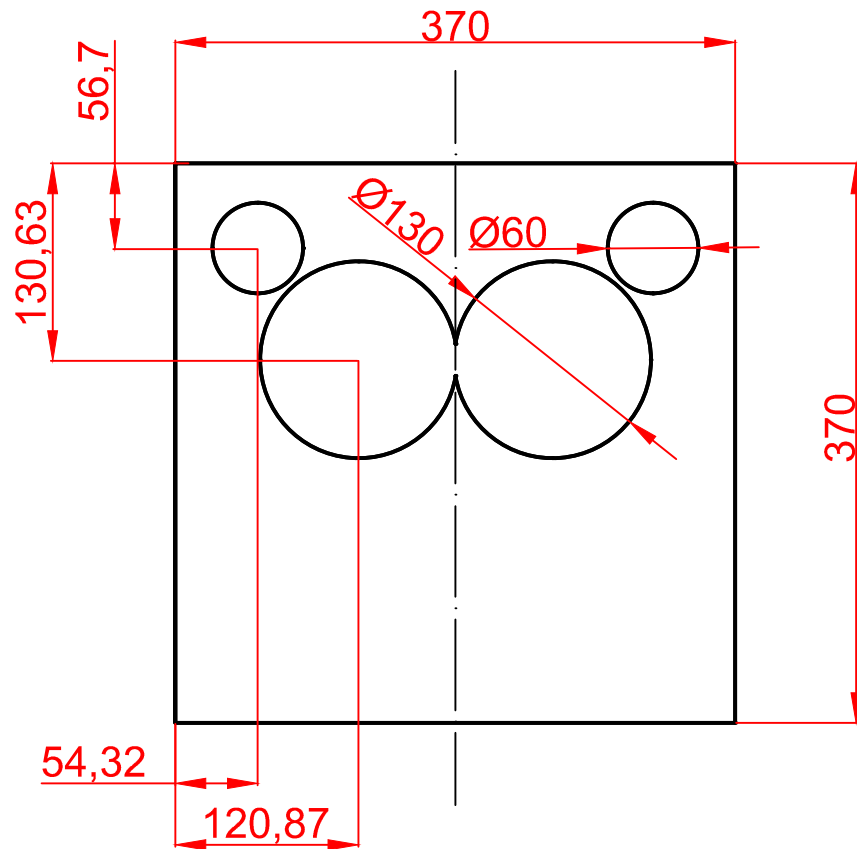


Planta

Alzado

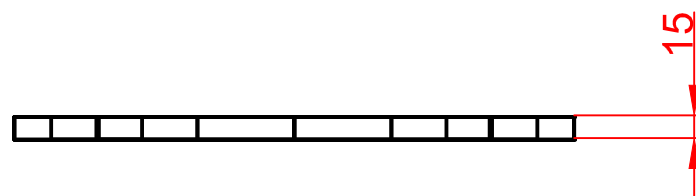


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasis base inferior	Pl. PA3

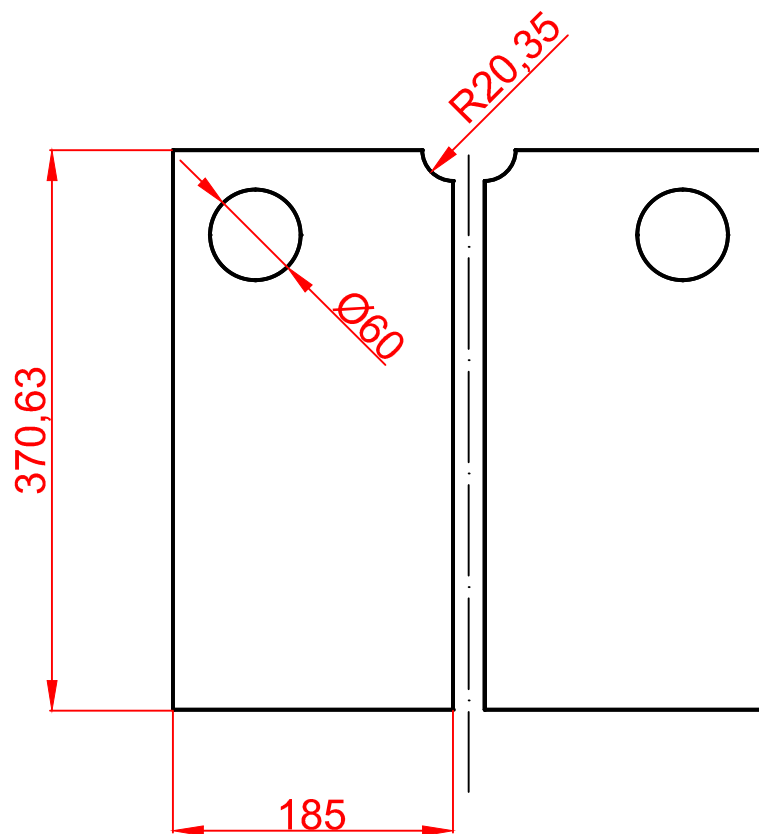


Planta

Alzado

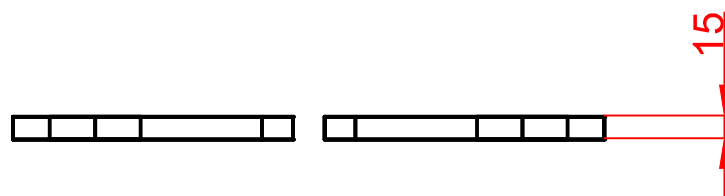


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasispared trasera	PI. PA4

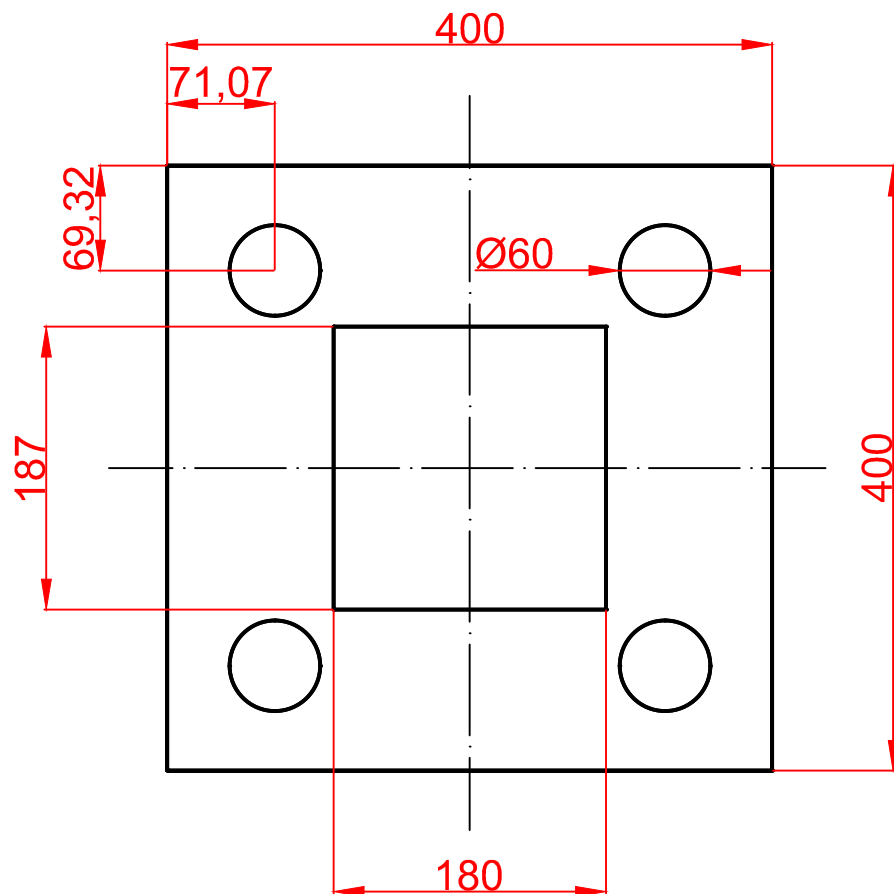


Planta

Alzado

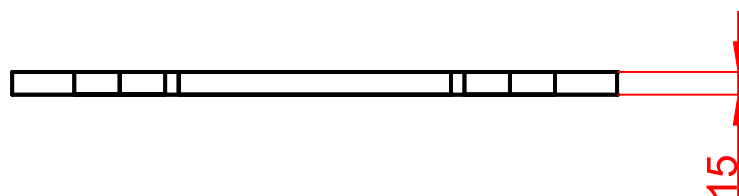


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasis puertas	PI. PA5



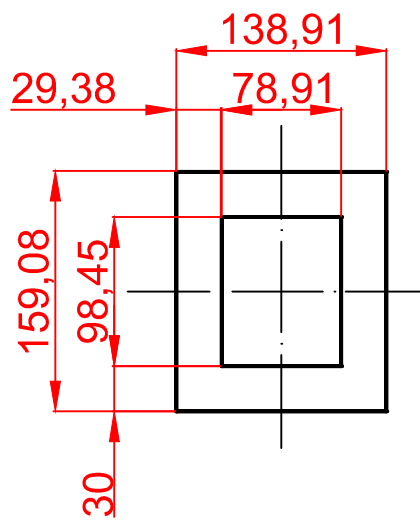
Planta

Alzado



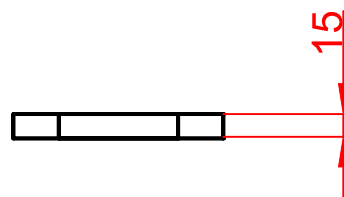
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasis pared lateral	Pl. PA6





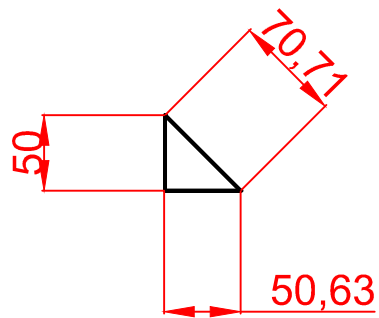
Planta

Alzado



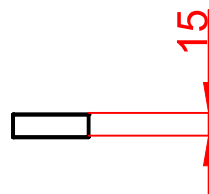
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasis marco de sustipensión	PI. PA7

Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasis triángulo de tracción	Pl. PA8

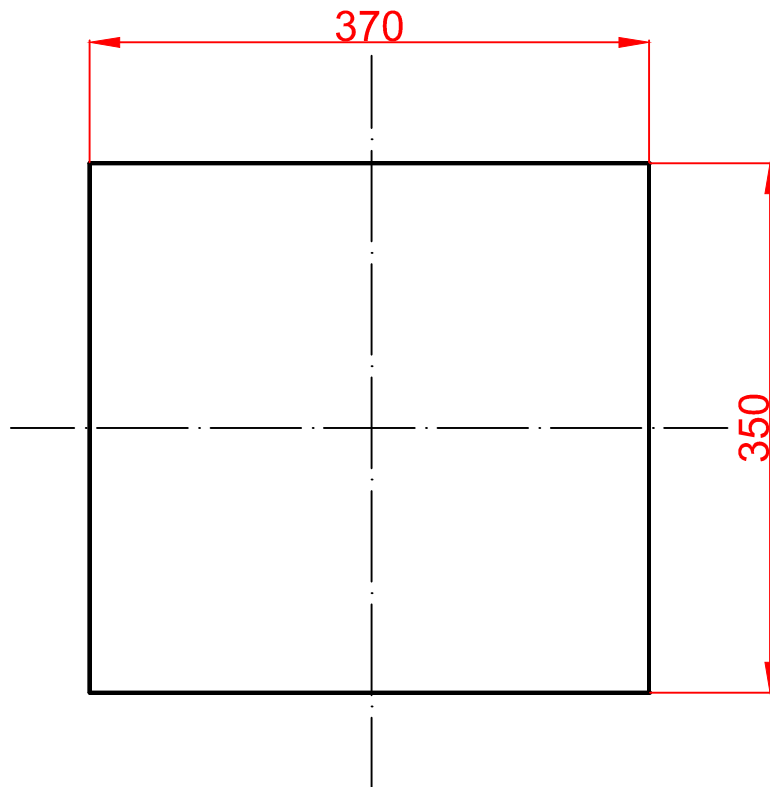


Planta

Alzado

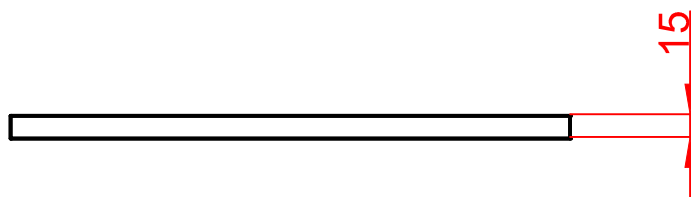


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasis escuadra	Pl. PA9

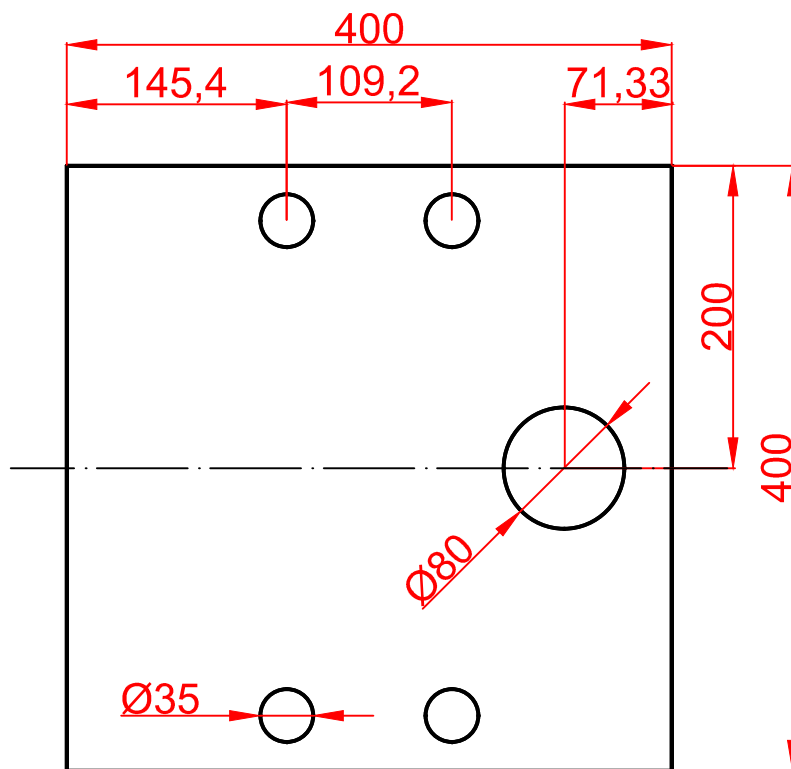


Planta

Alzado

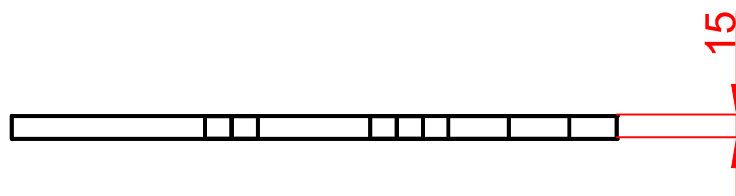


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasis bandeja extraible	PI. PA2

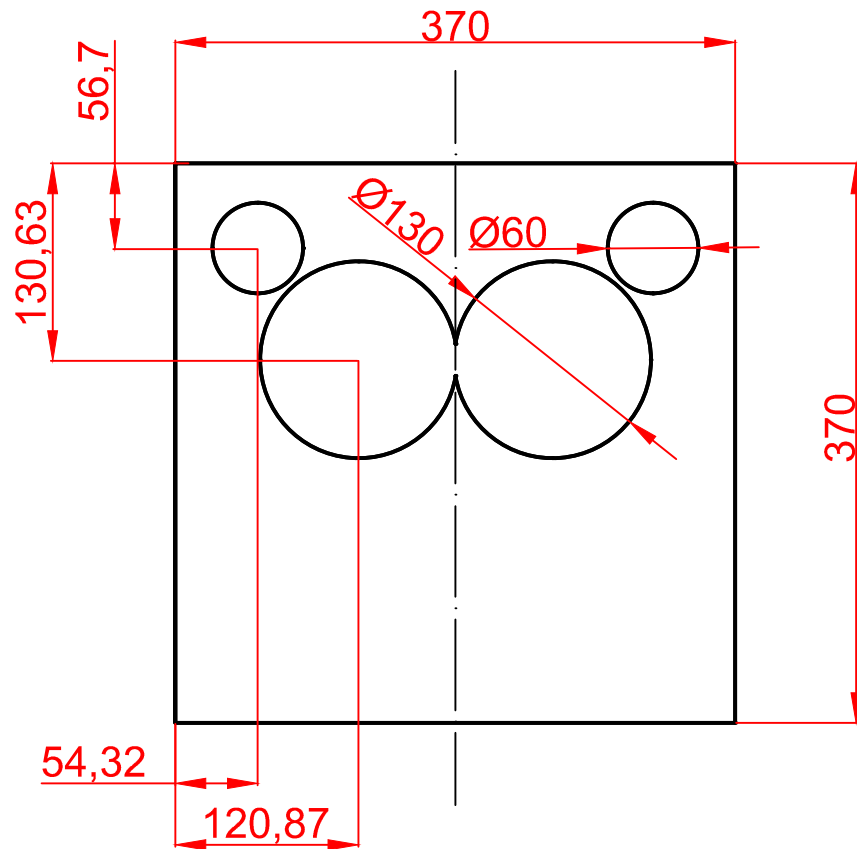


Planta

Alzado

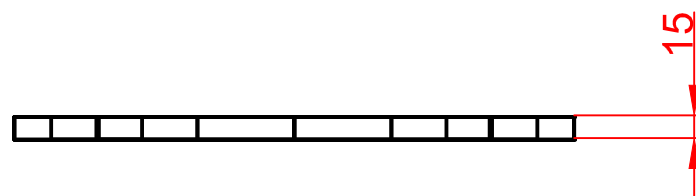


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasis base inferior	Pl. PA3

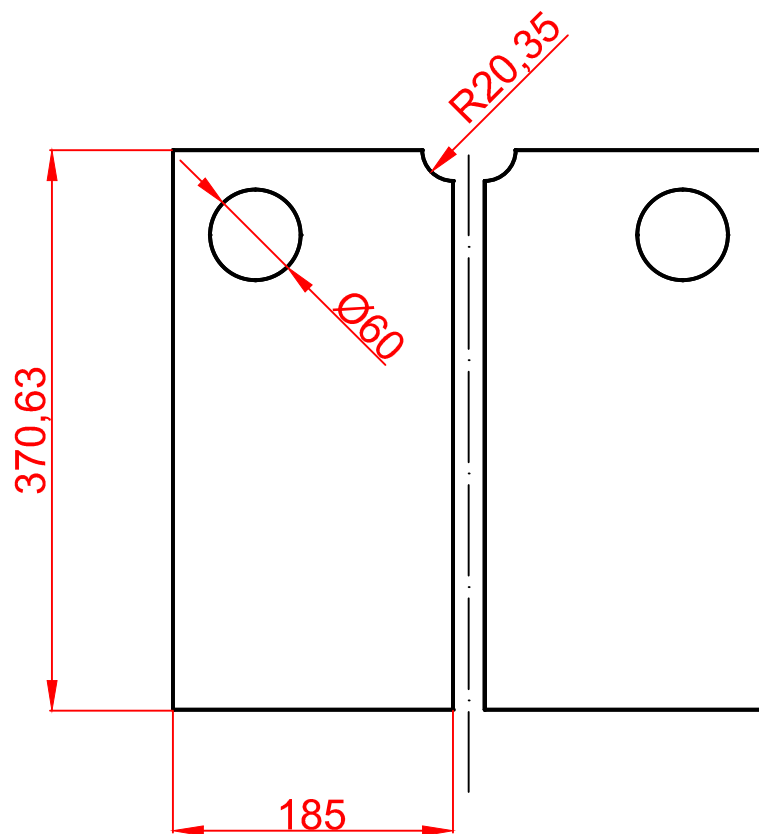


Planta

Alzado

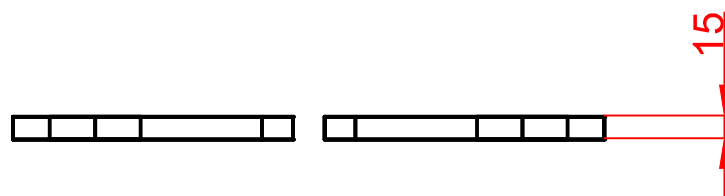


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasispared trasera	Pl. PA4

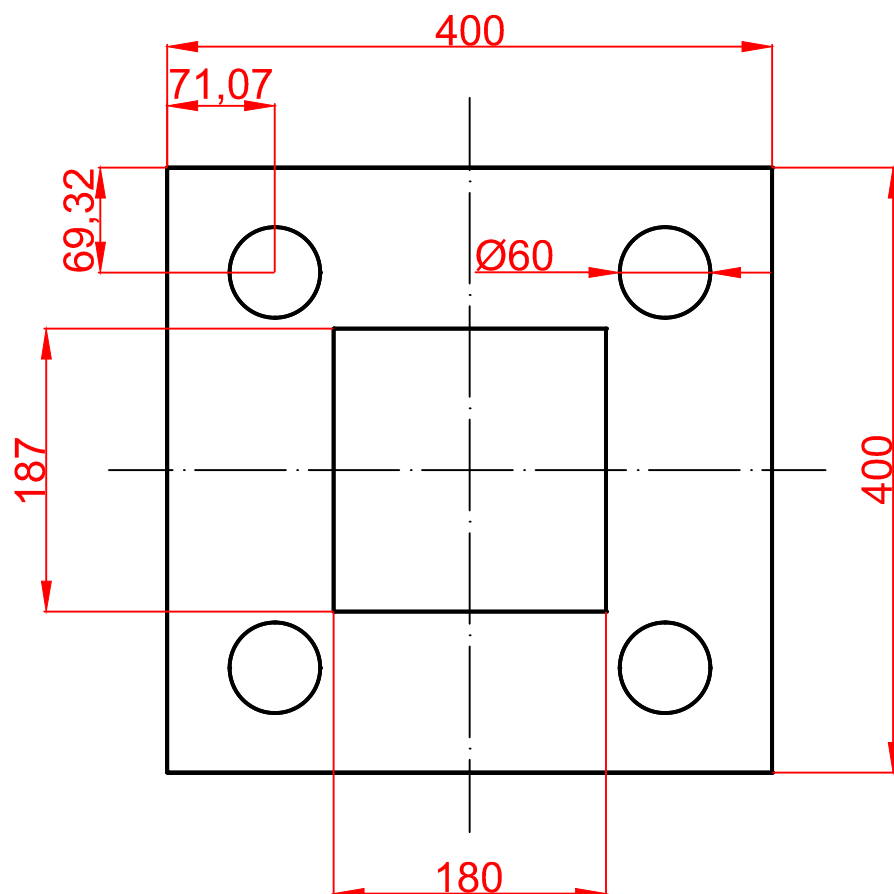


Planta

Alzado

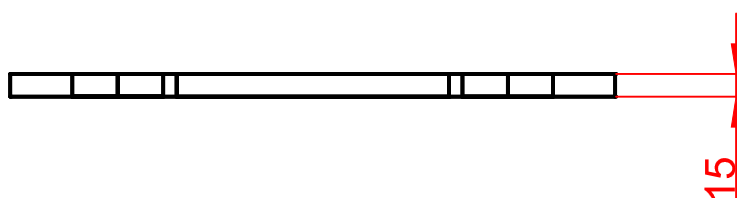


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasis puertas	PI. PA5



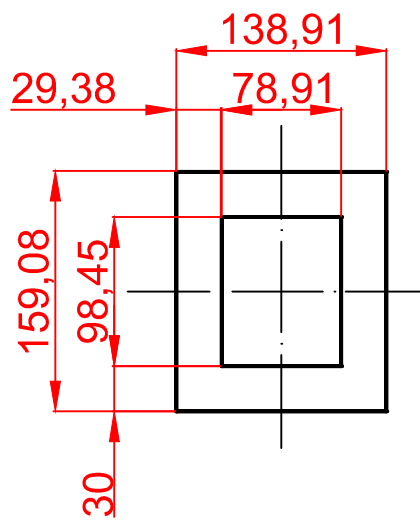
Planta

Alzado



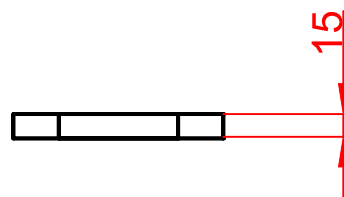
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasis pared lateral	Pl. PA6





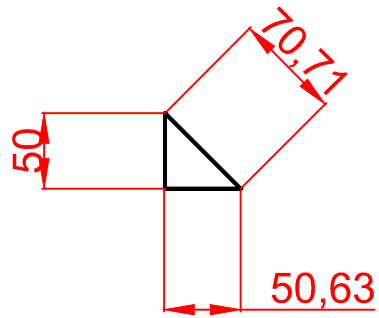
Planta

Alzado



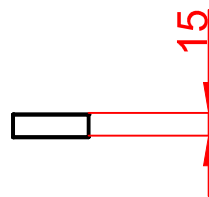
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasis marco de sustipensión	PI. PA7

Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasis triángulo de tracción	Pl. PA8

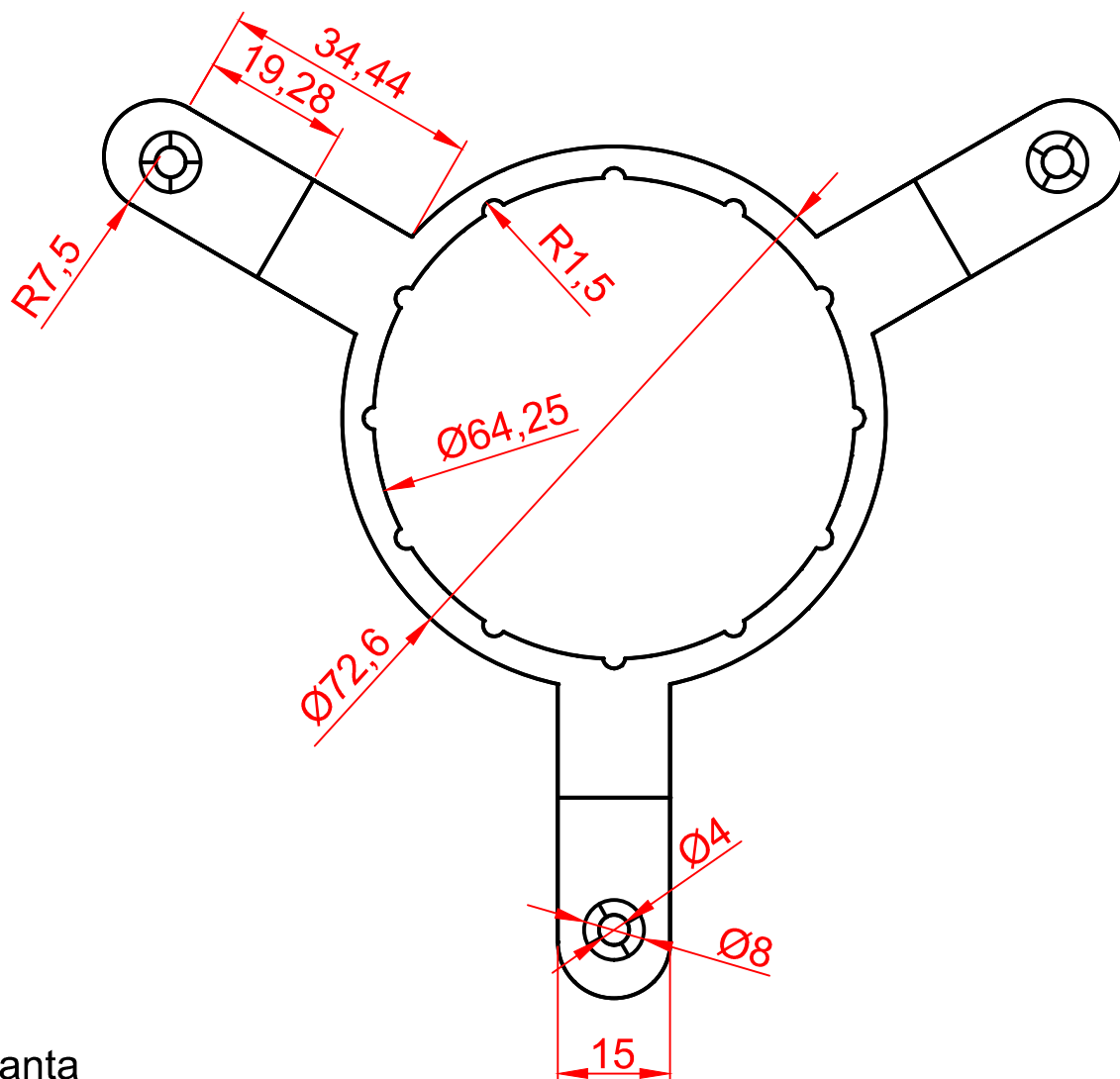


Planta

Alzado

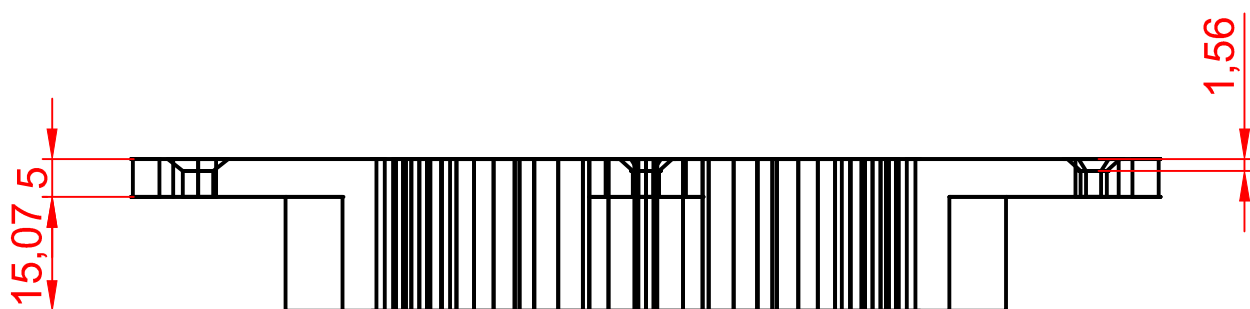


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:5	Pieza de chasis escuadra	Pl. PA9

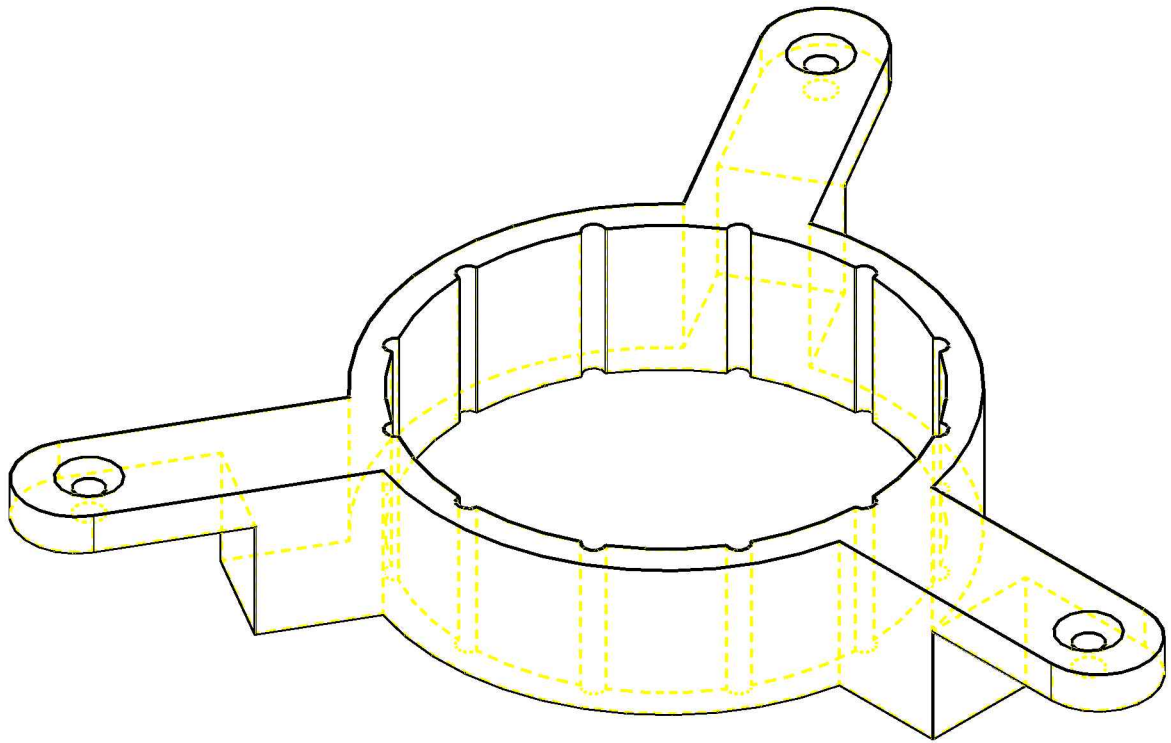


Planta

Alzado

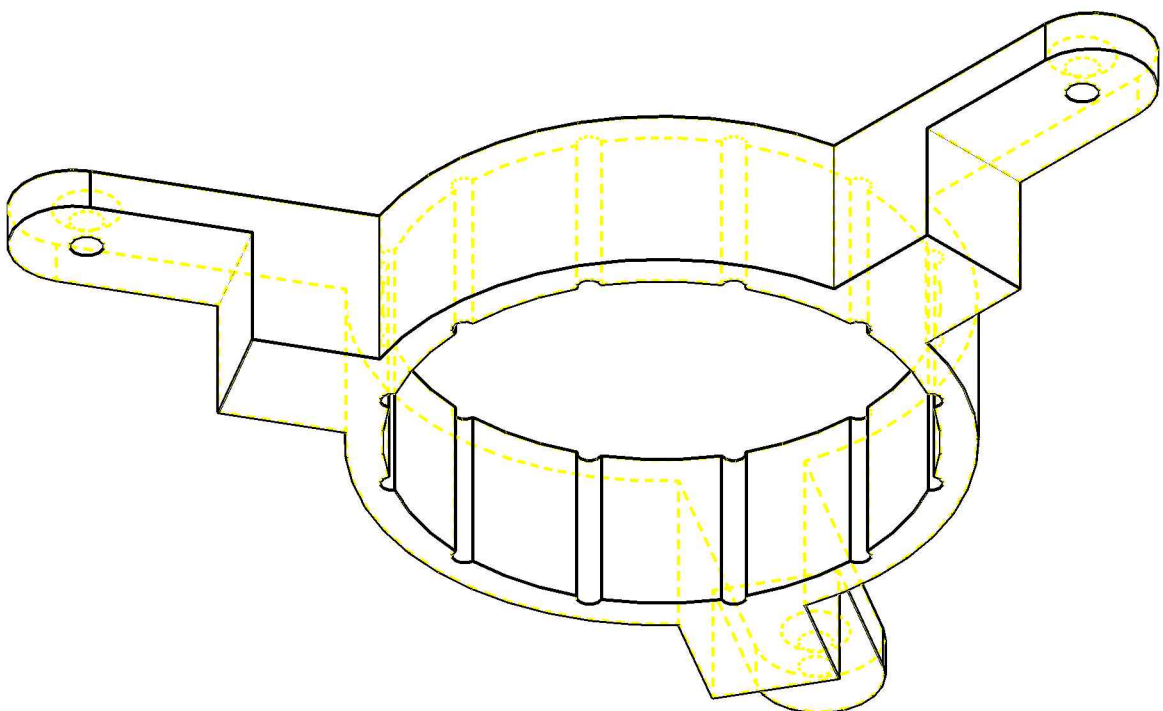


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de tracción sujeción de motor parte 1	PI. PB1.1

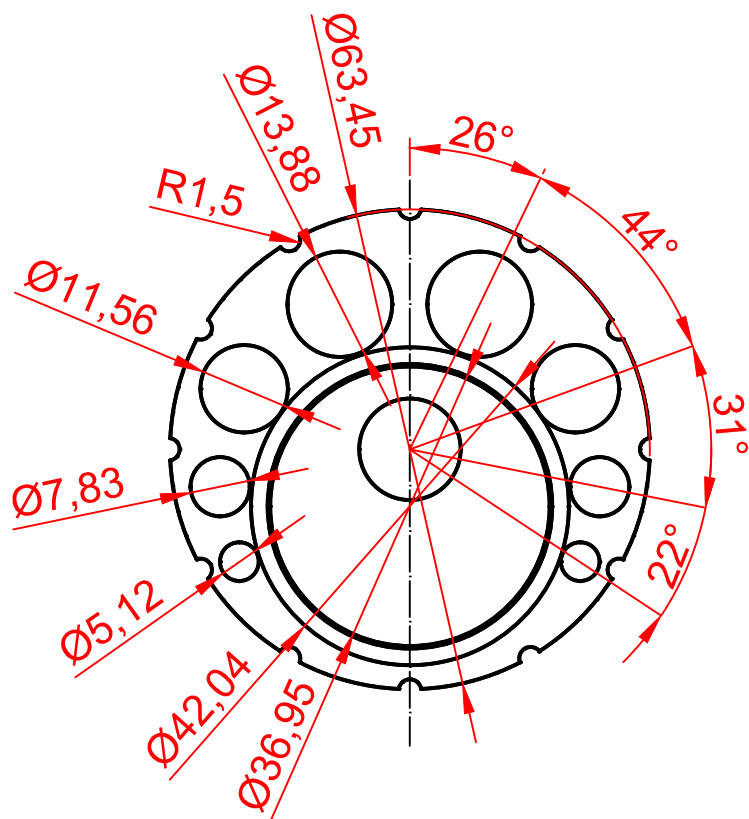


Vista Superior, Suroeste

Vista Inferior, Suroeste



Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de tracción sujeción de motor parte 1	Pl. PB1.1 3D

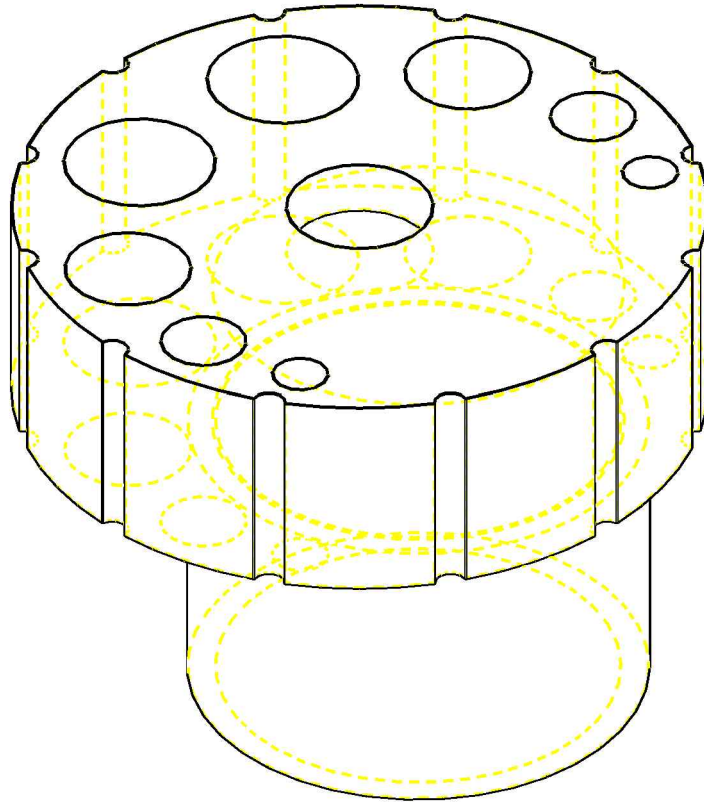


Planta

Alzado

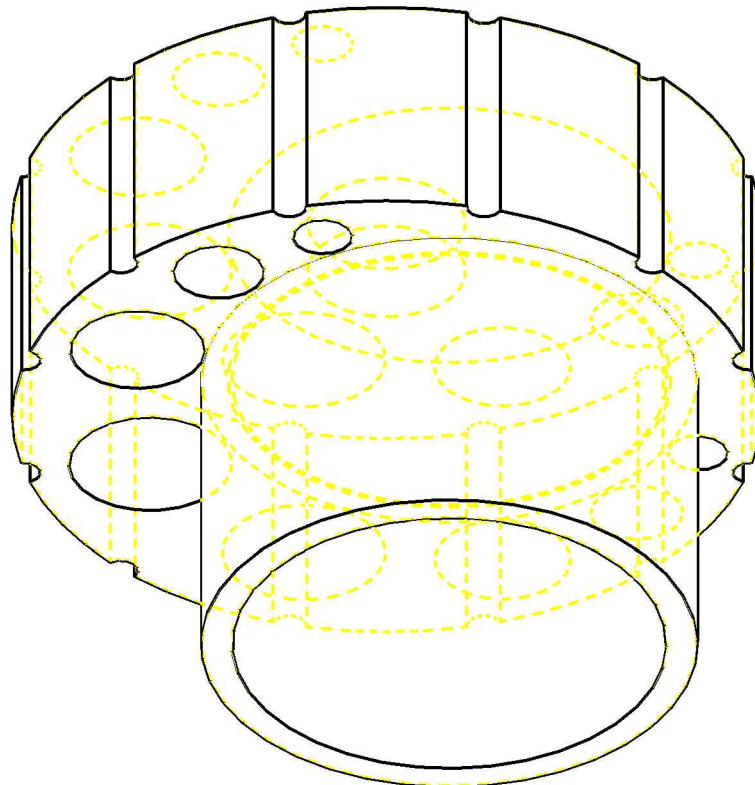


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de tracción sujeción de motor parte 2	PI. PB1.2

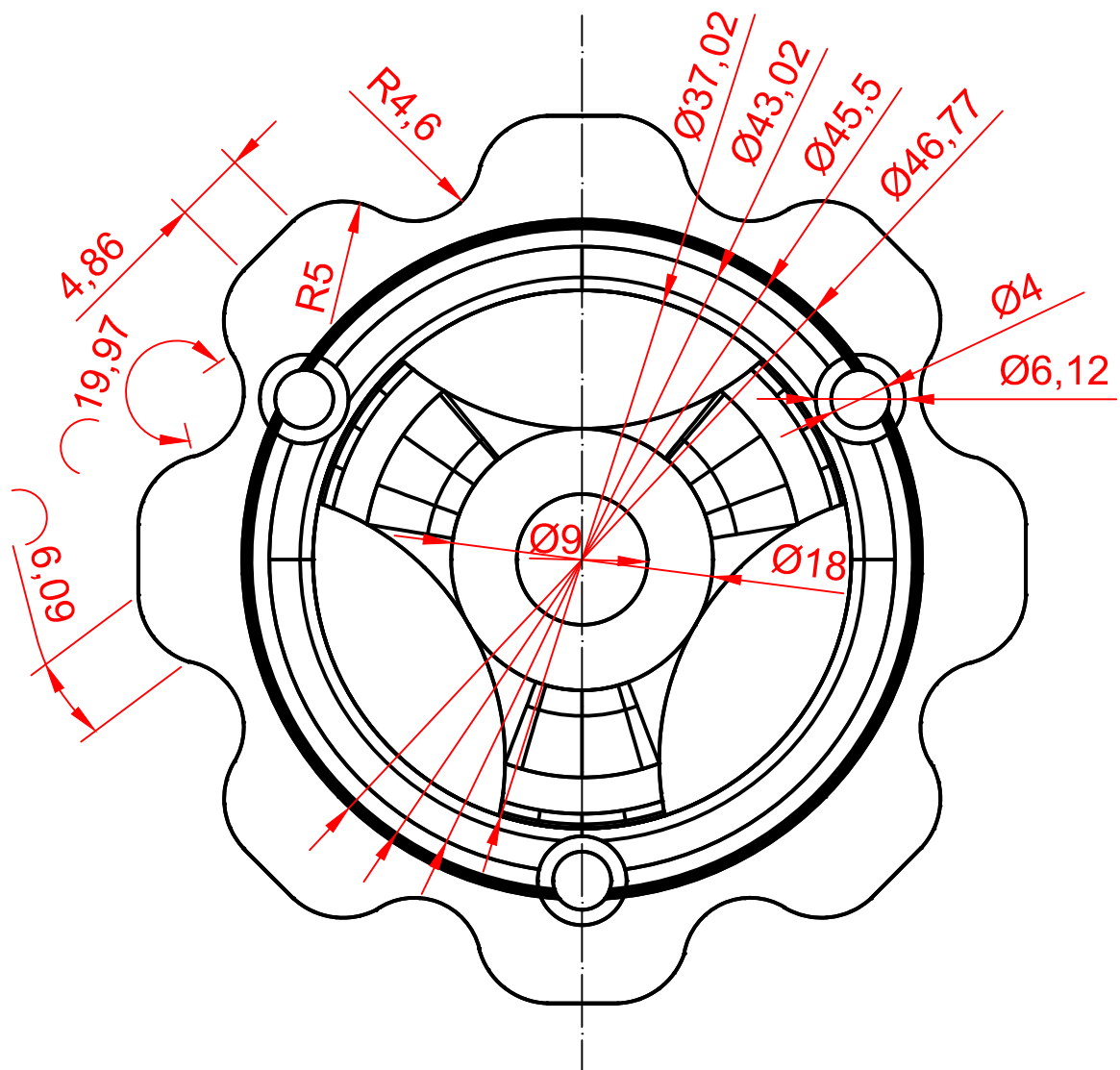


Vista Superior, Suroeste

Vista Inferior, Suroeste

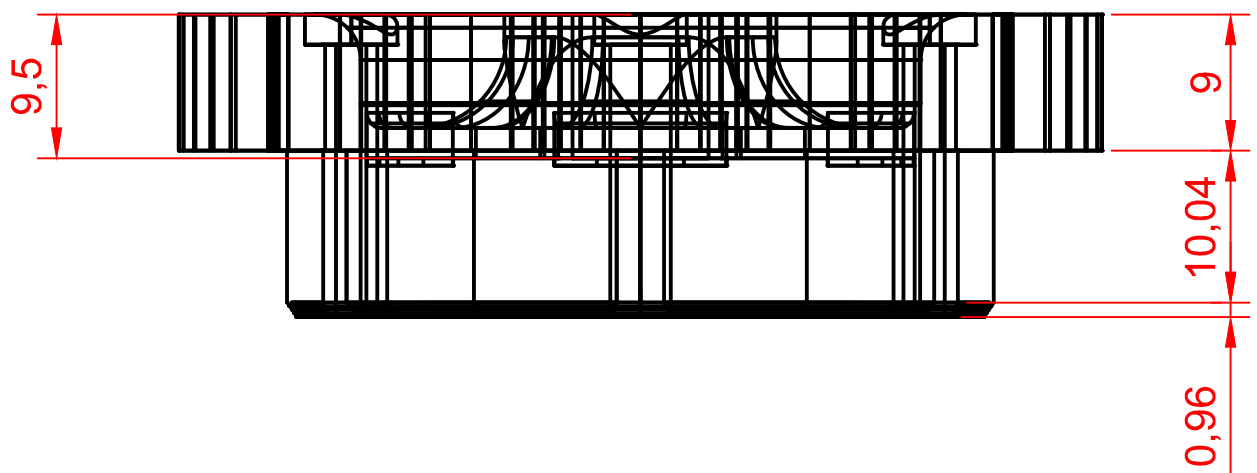


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de tracción sujeción de motor parte 2	Pl. PB1.2 3D



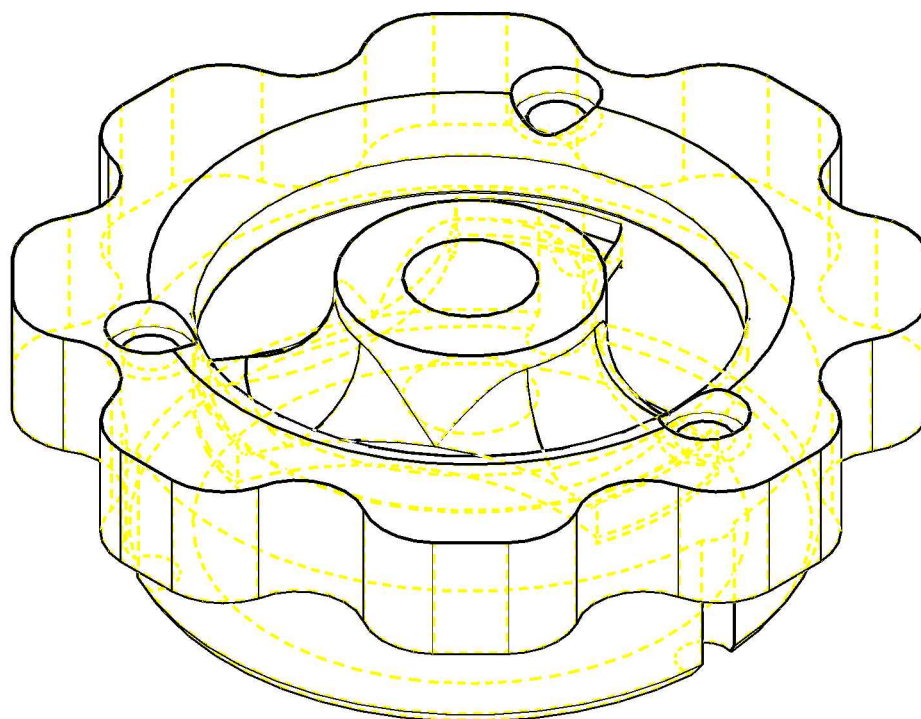
Planta

Alzado



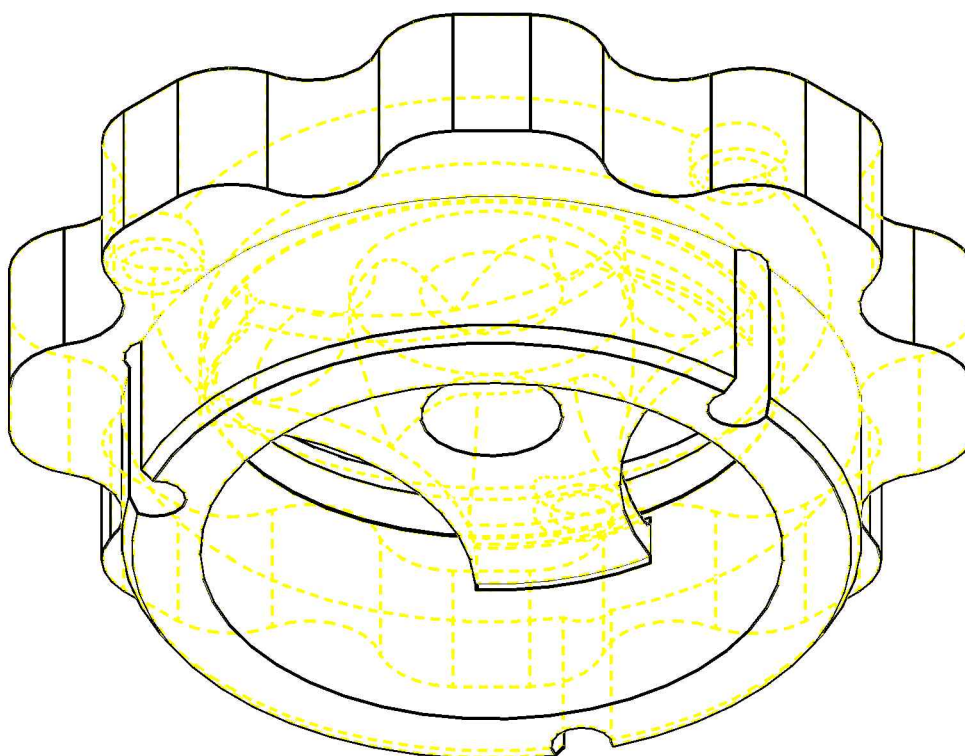
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
2:1	Pieza de tracción engranaje tensor	PI. PB2



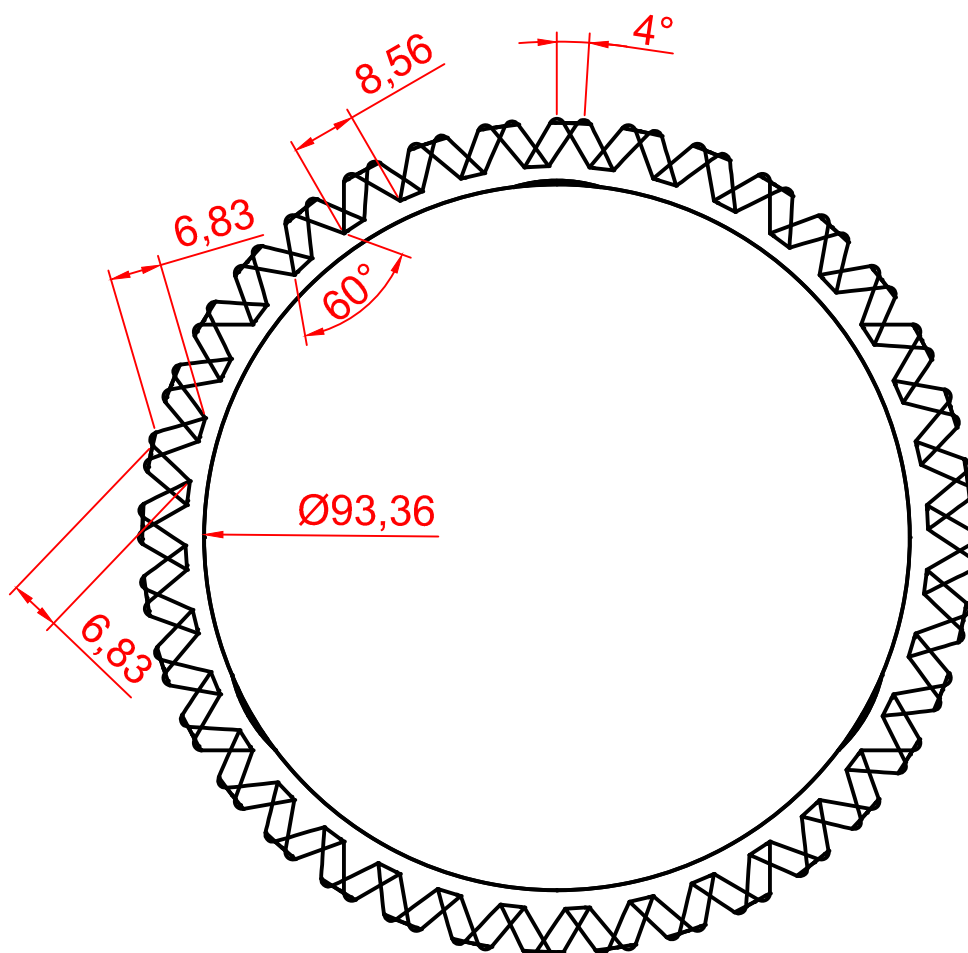


Vista Superior, Suroeste

Vista Inferior, Suroeste



Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de tracción engranaje tensor	Pl. PB2 3D

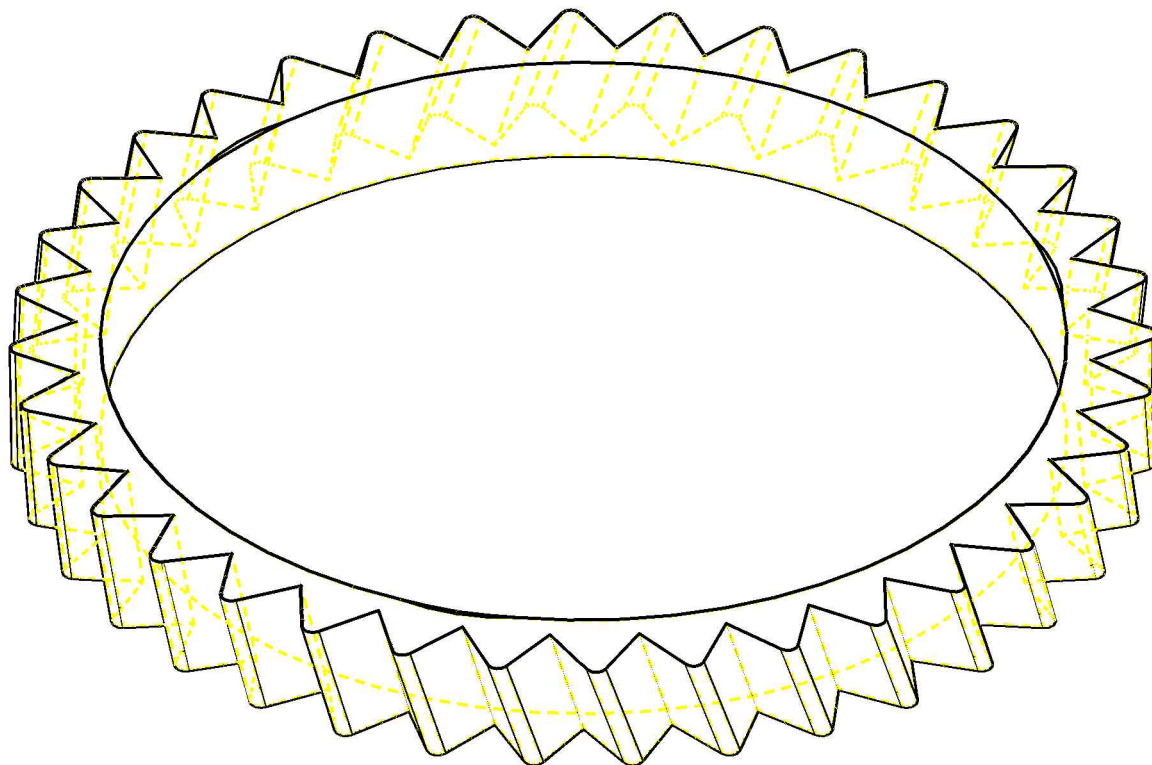


Planta

Alzado

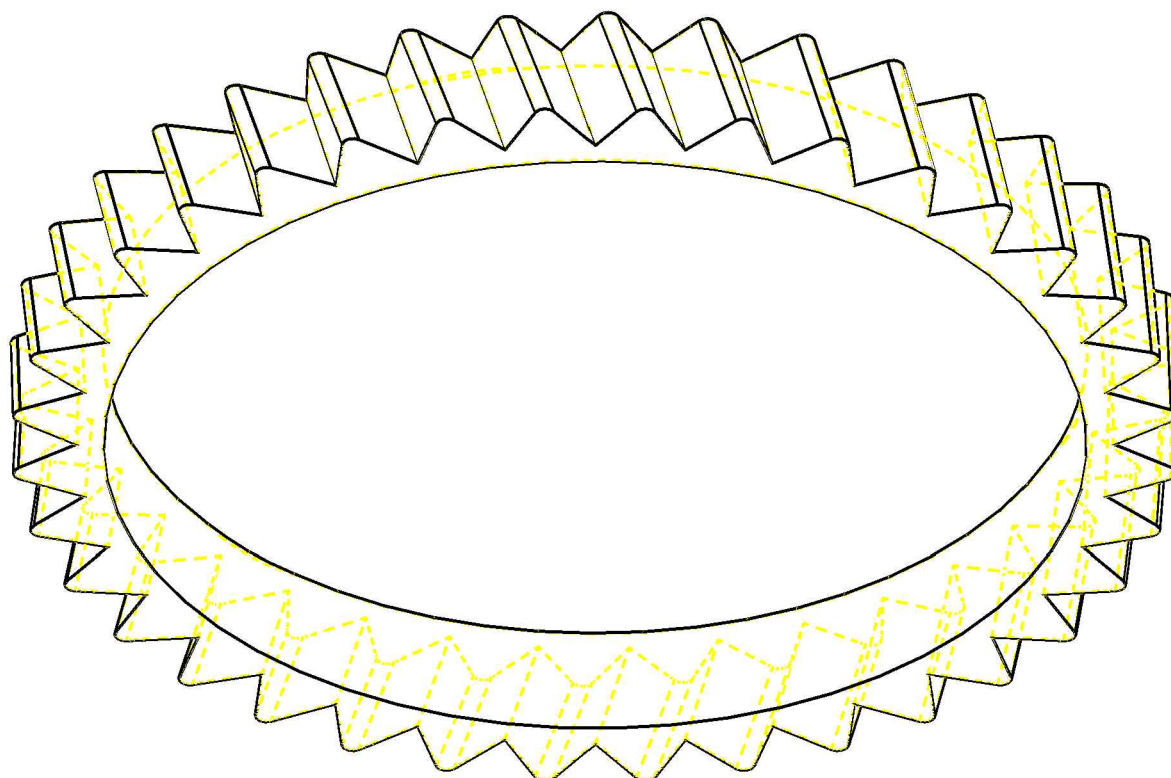


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de tracción dientes de engranaje satélite levógiro	Pl. PB3.1

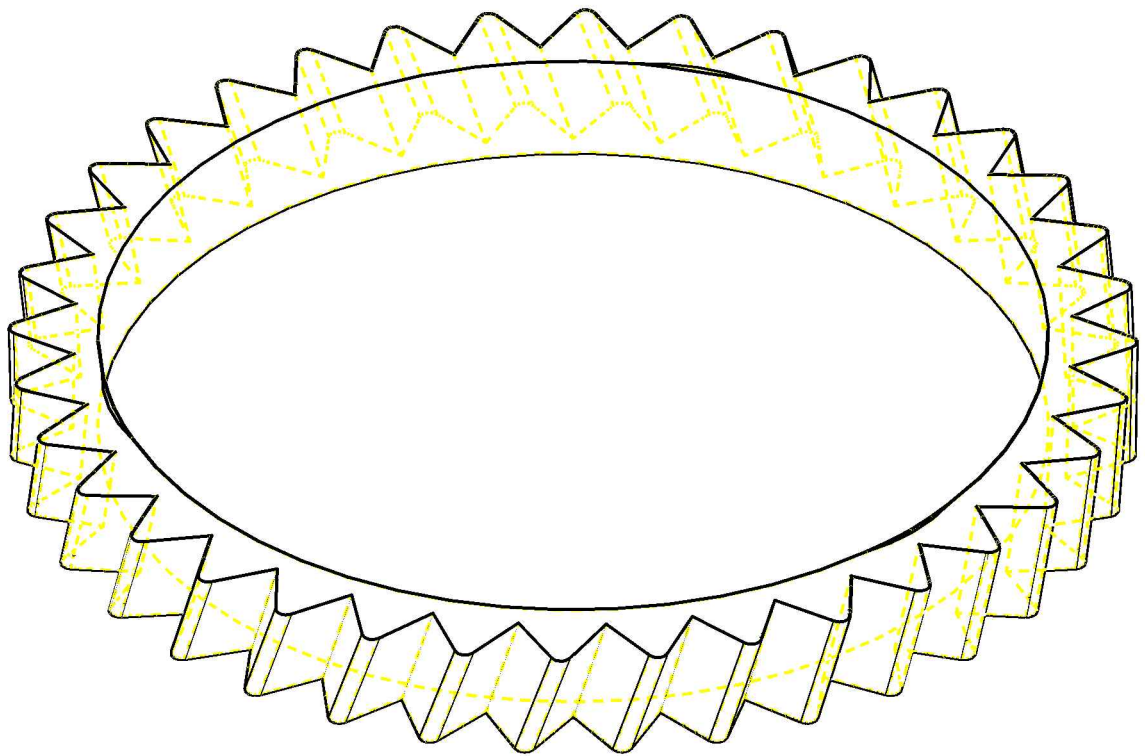


Vista Superior, Suroeste

Vista Inferior, Suroeste

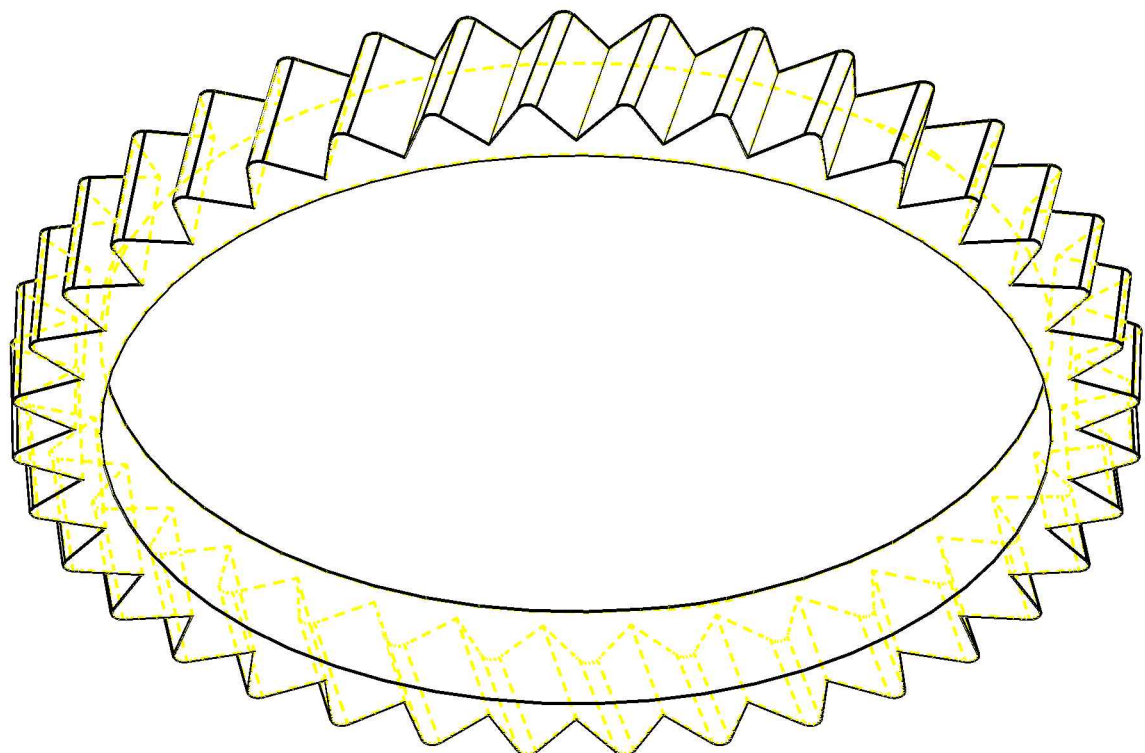


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de tracción dientes de engranaje satélite levógiro	Pl. PB3.1 3D



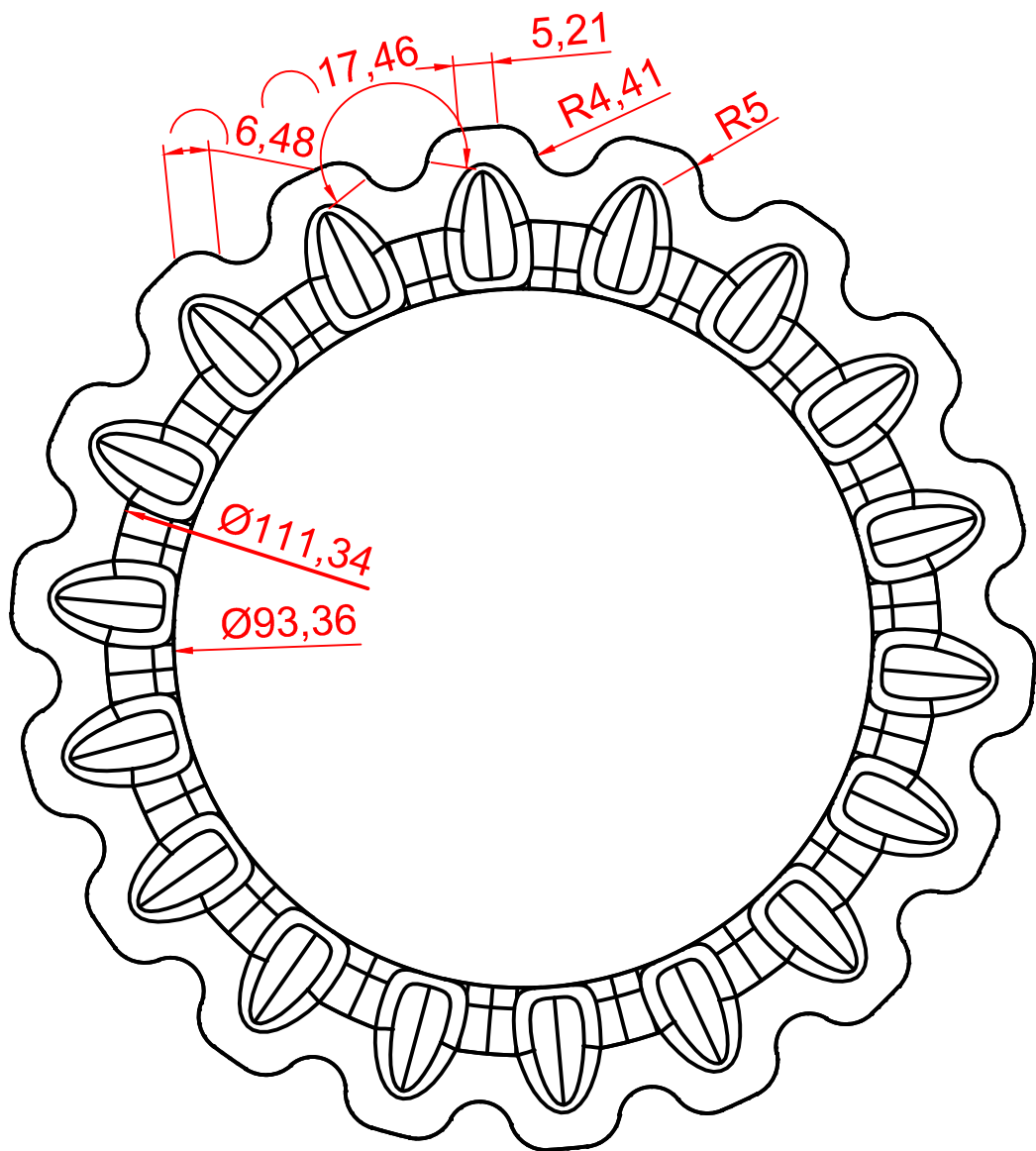
Vista Superior, Suroeste

Vista Inferior, Suroeste



Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de tracción dientes de engranaje satélite dextrógiro	Pl. PB3.2 3D



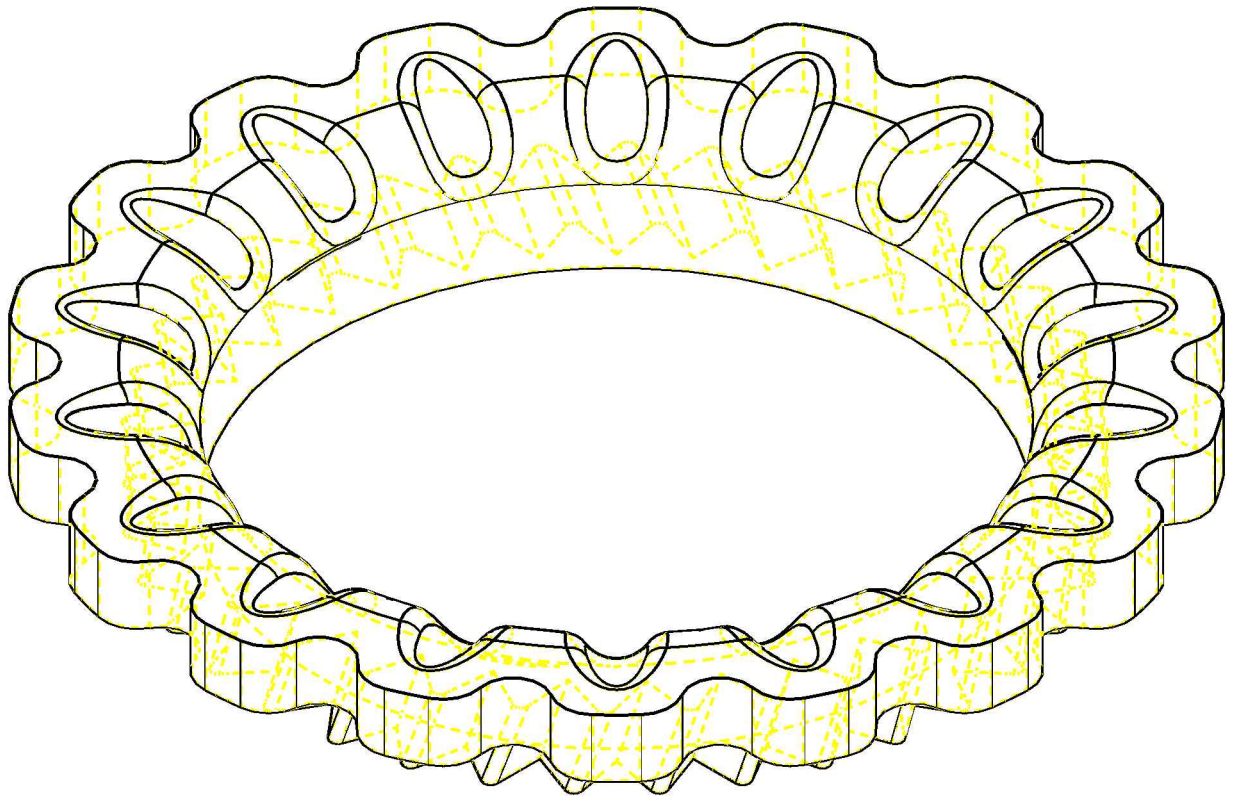


Planta

Alzado

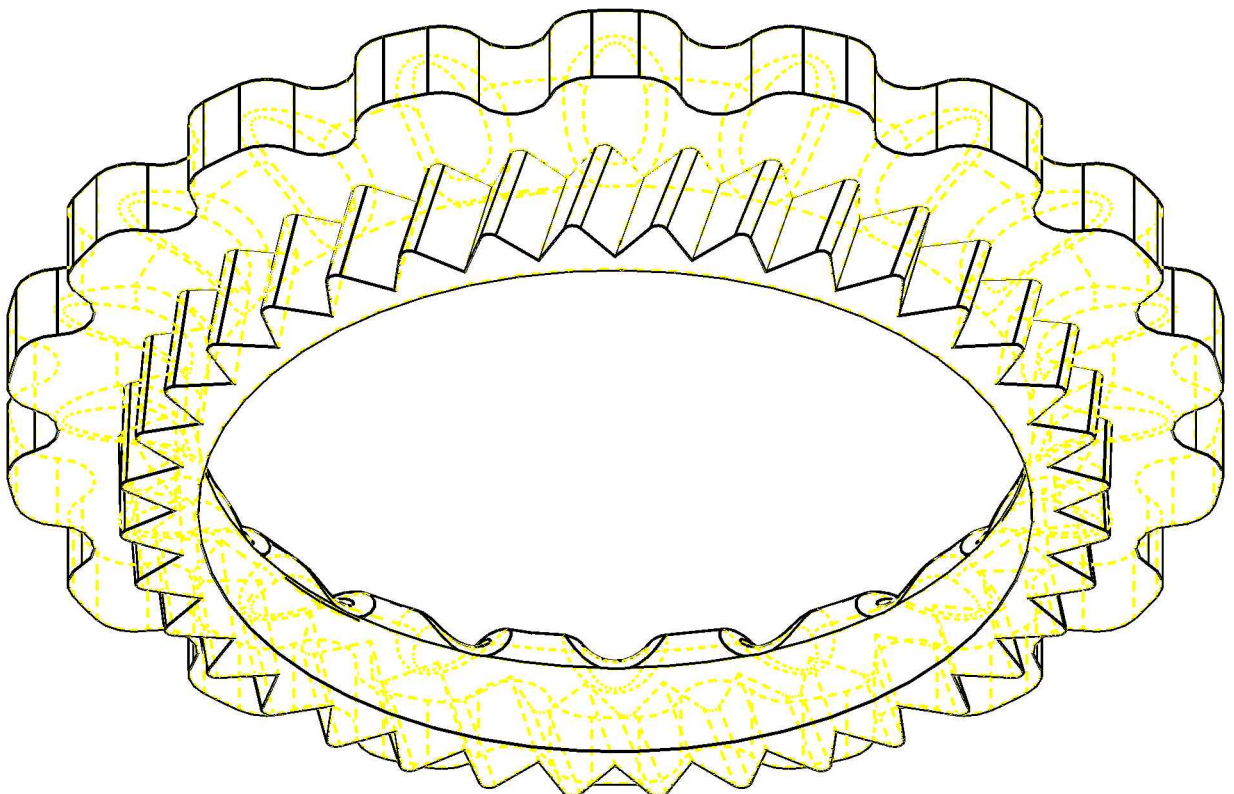


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de tracción corona exterior de engranaje satélite	Pl. PB3.3

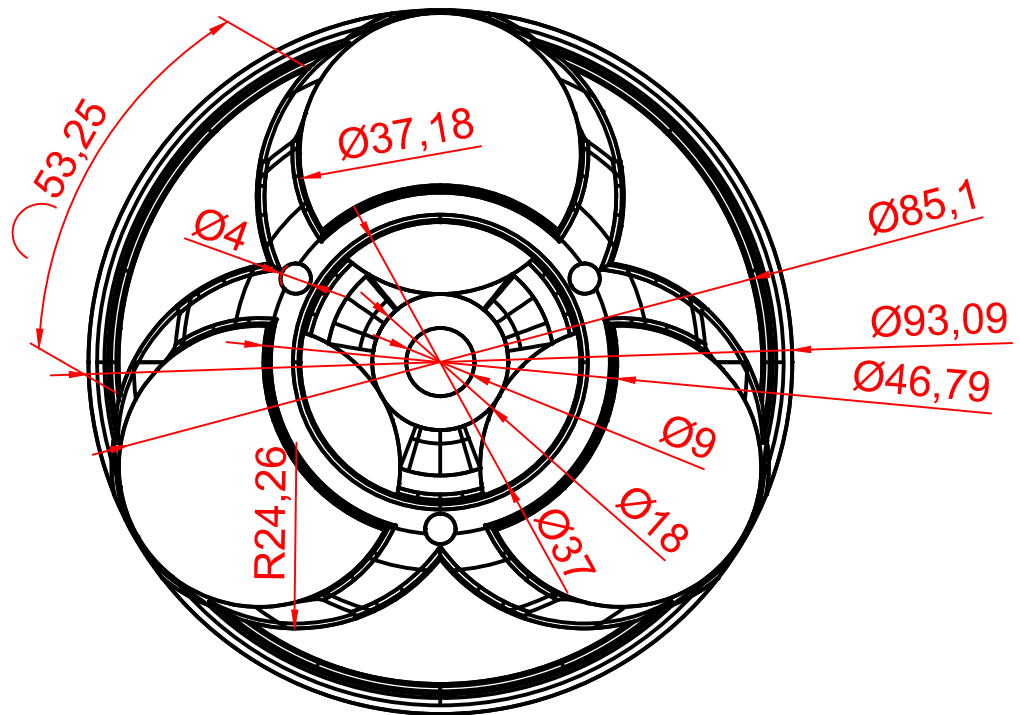


Vista Superior, Suroeste

Vista Inferior, Suroeste

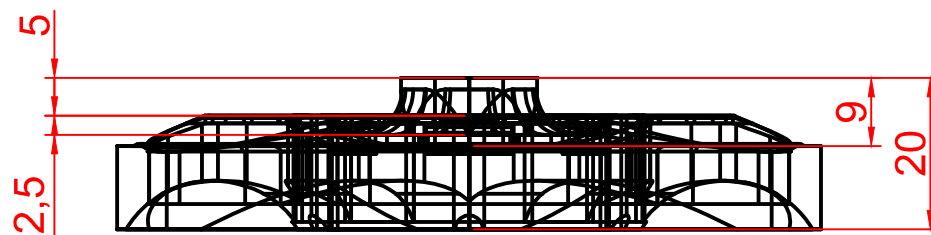


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de tracción engranaje satélite	Pl. PB3.4 3D

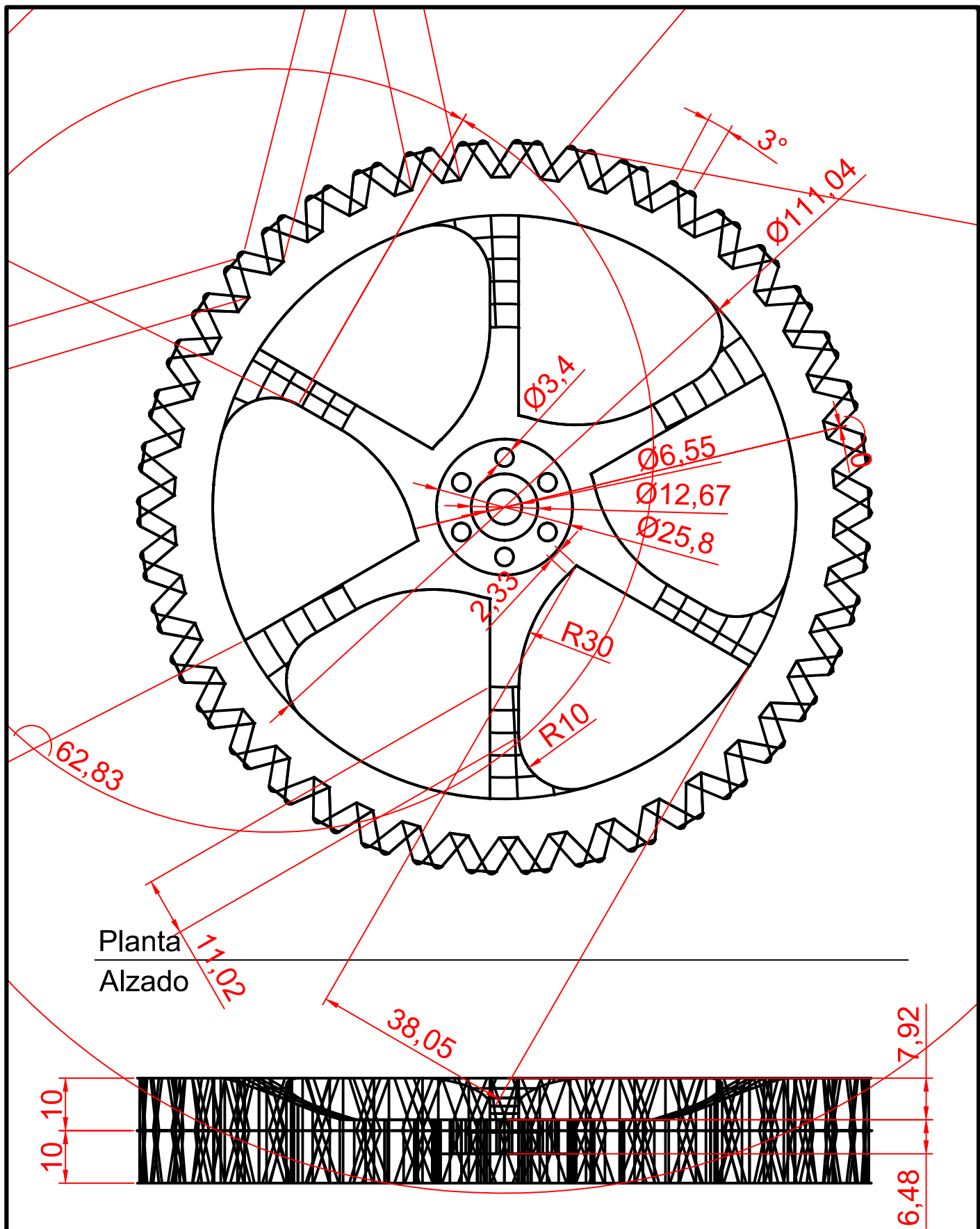


Planta

Alzado

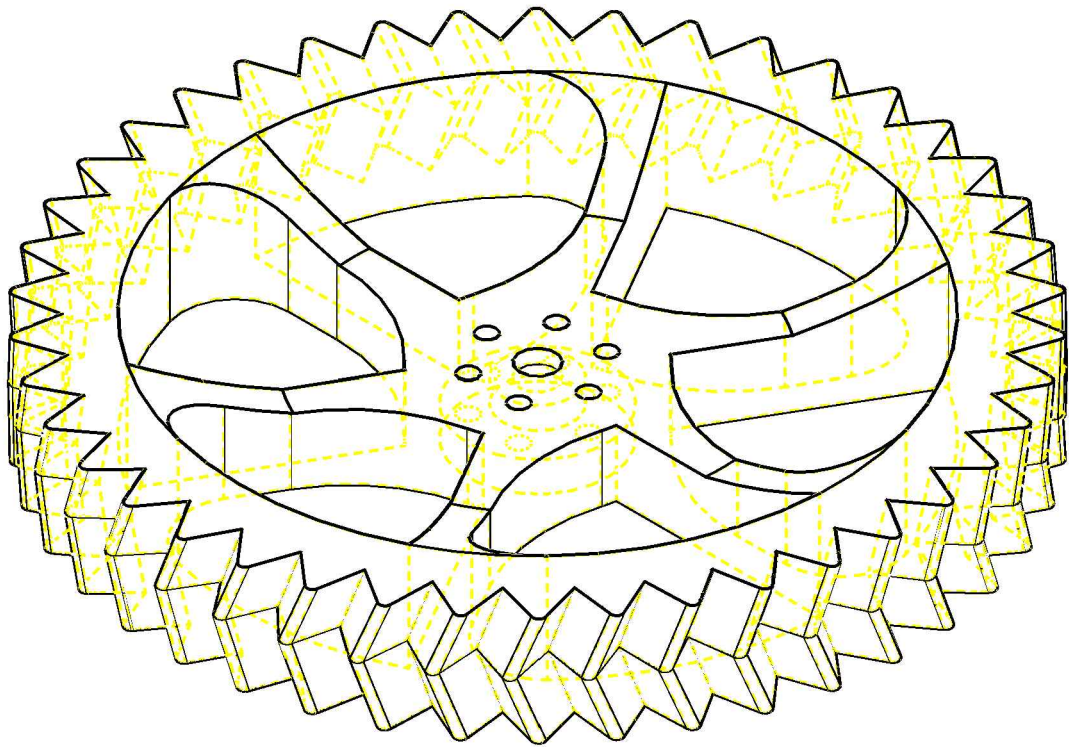


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de tracción radios de engranaje satélite	Pl. PB3.5



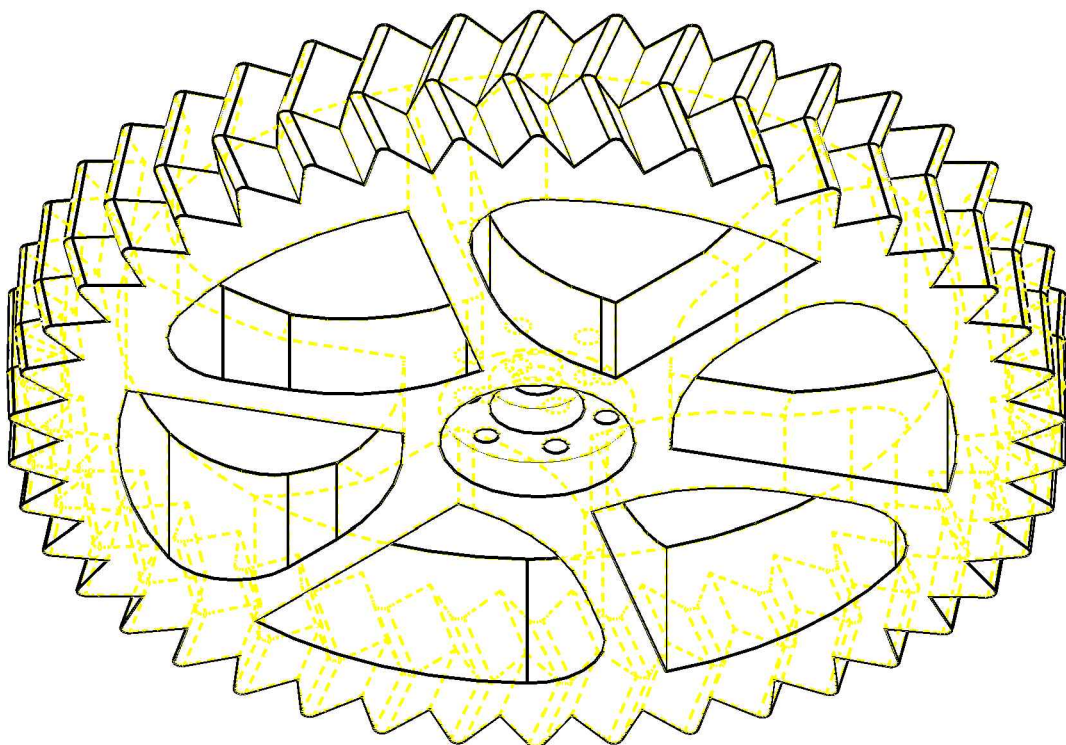
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de tracción engranaje central izquierdo	Pl. PB4.1



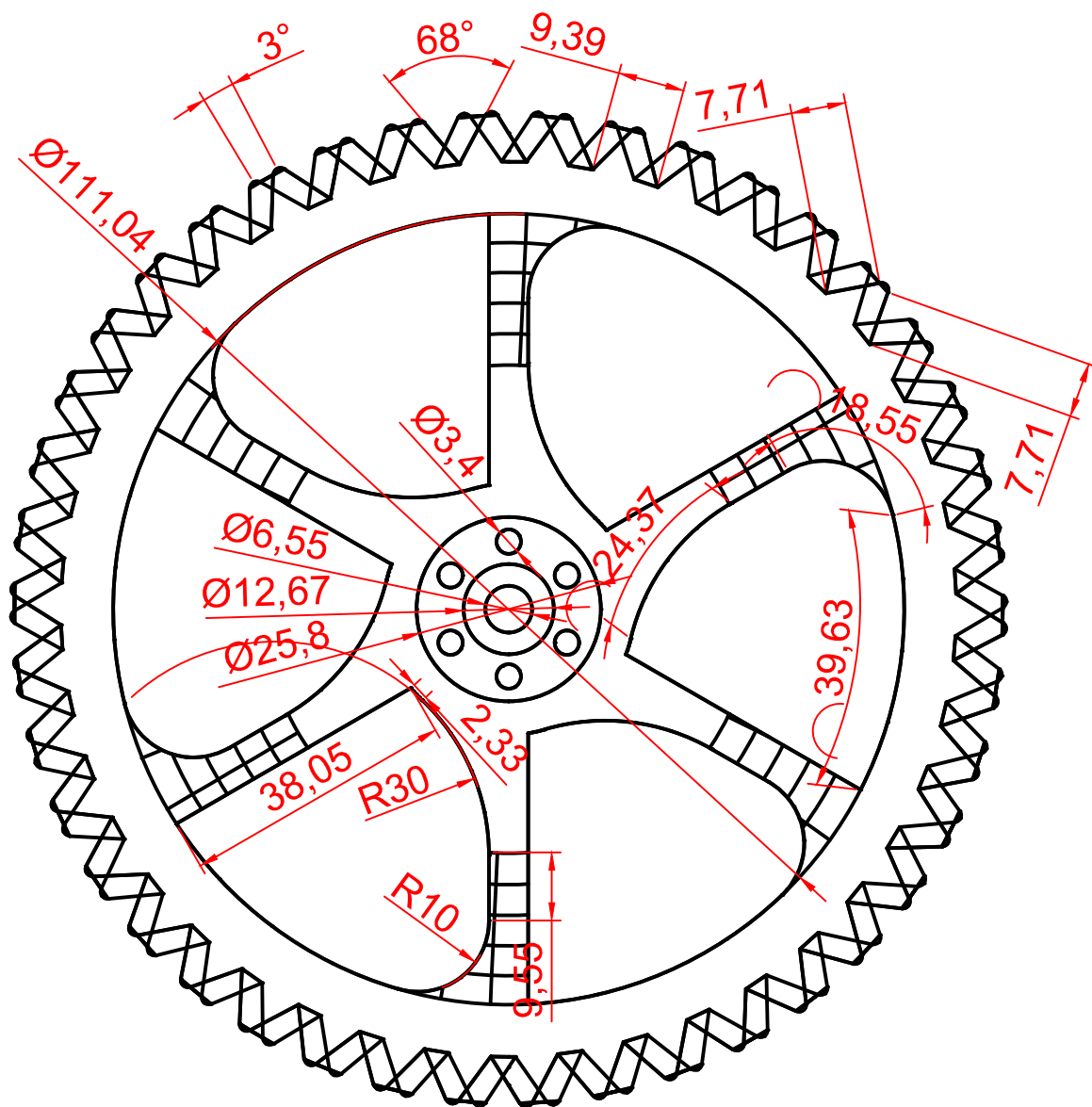


Vista Superior, Suroeste

Vista Inferior, Suroeste

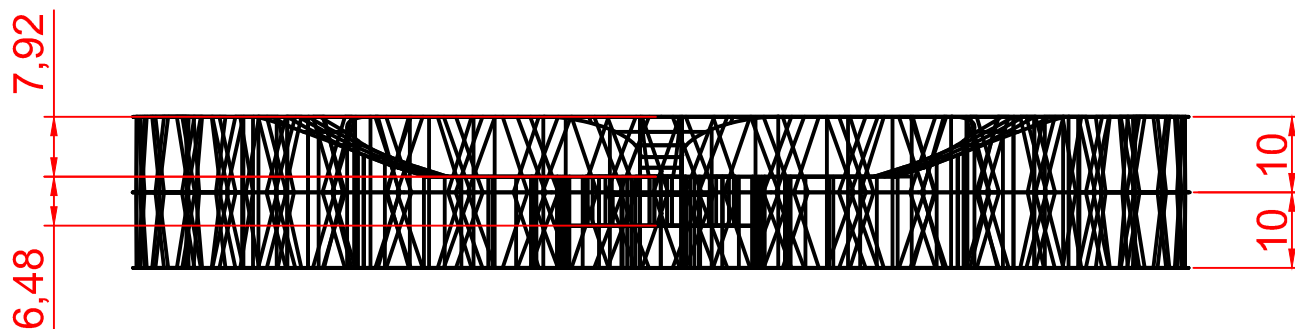


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de tracción engranaje central izquierdo	Pl. PB4.1 3D

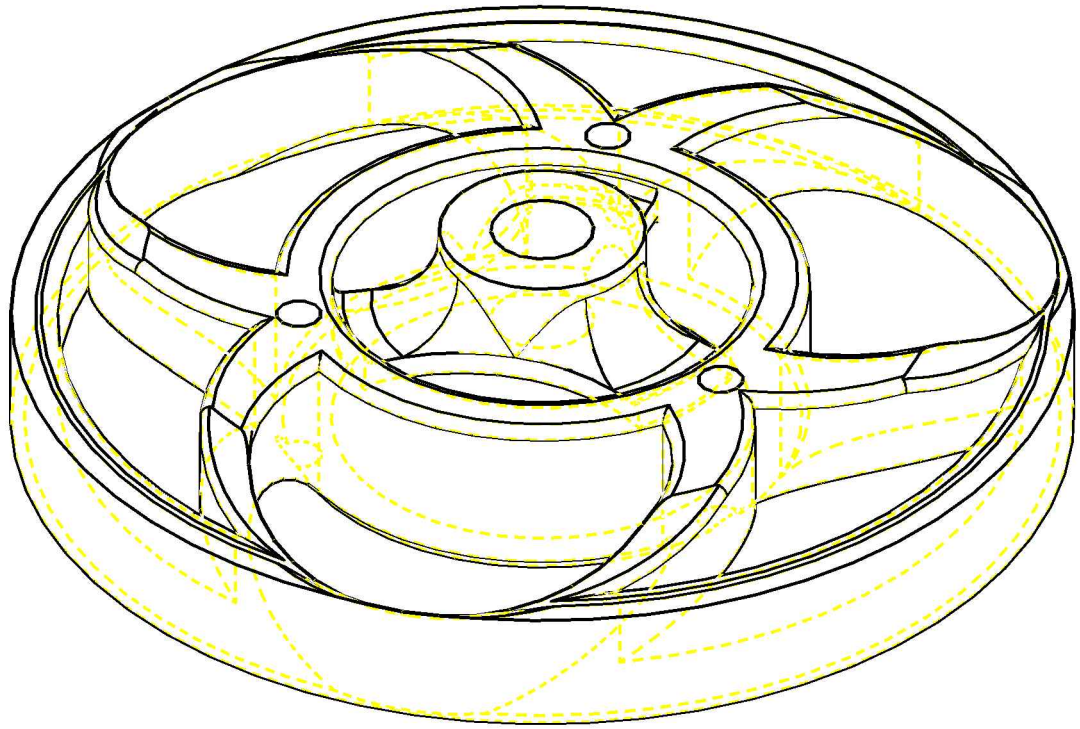


Planta

Alzado

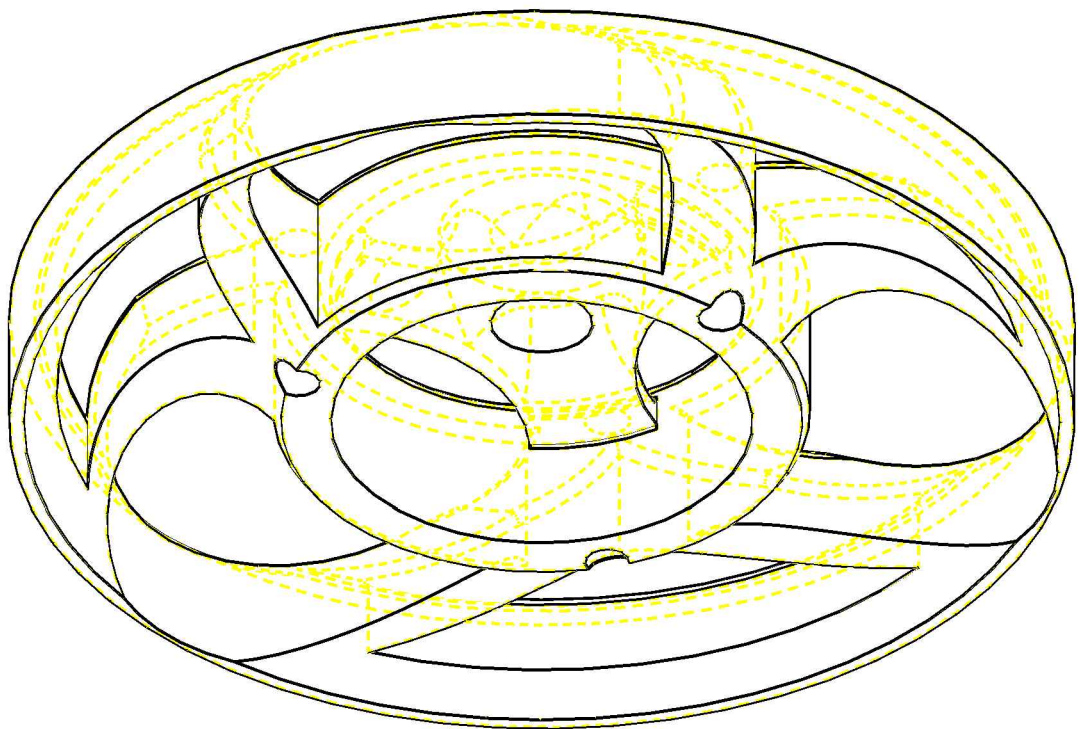


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de tracción engranaje central derecho	Pl. PB4.2



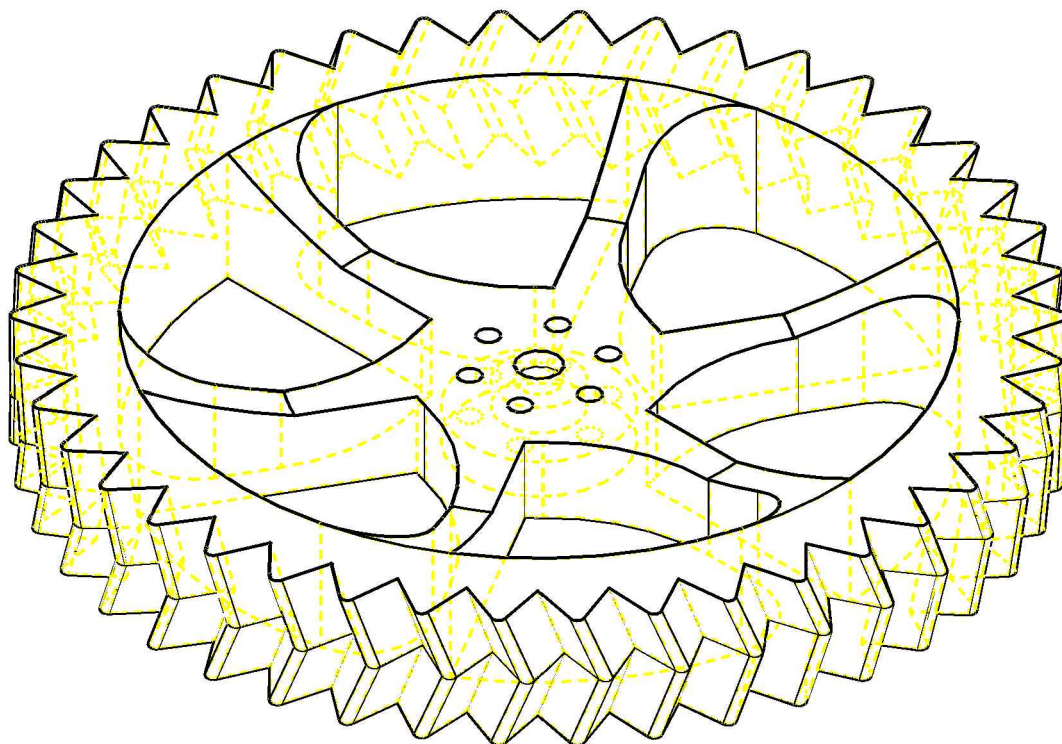
Vista Superior, Suroeste

Vista Inferior, Suroeste



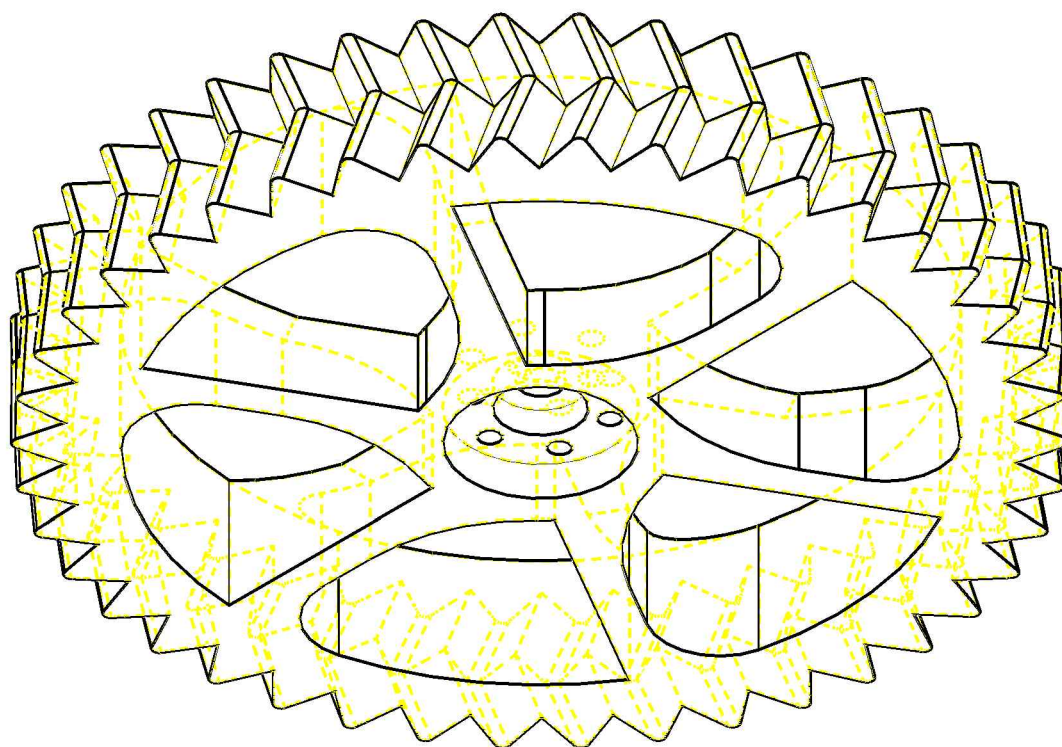
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de tracción radios de engranaje satélite	Pl. PB3.5 3D



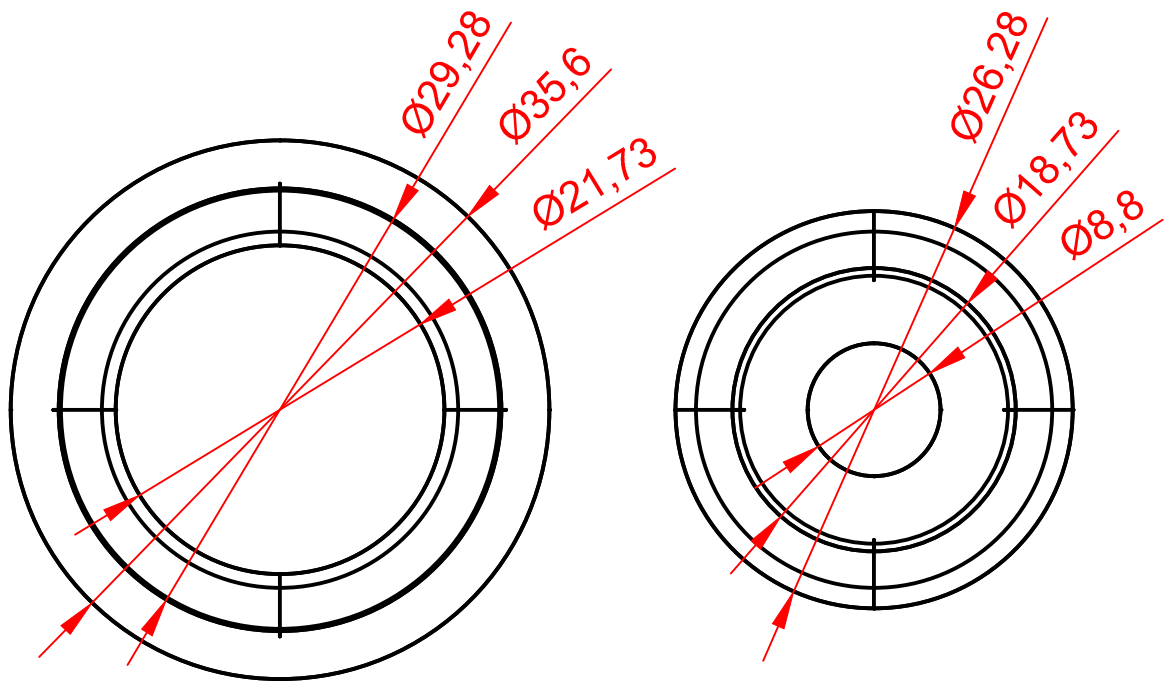


Vista Superior, Suroeste

Vista Inferior, Suroeste

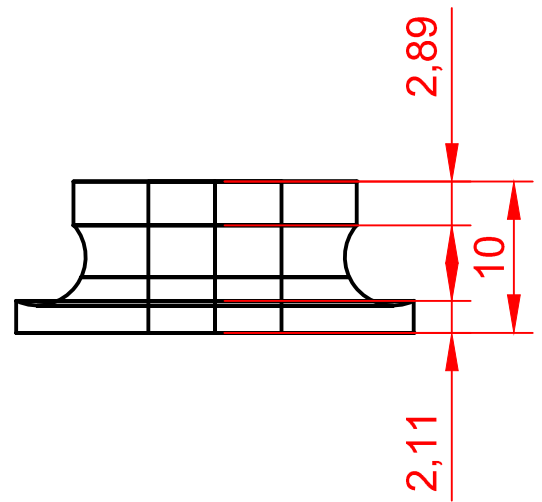
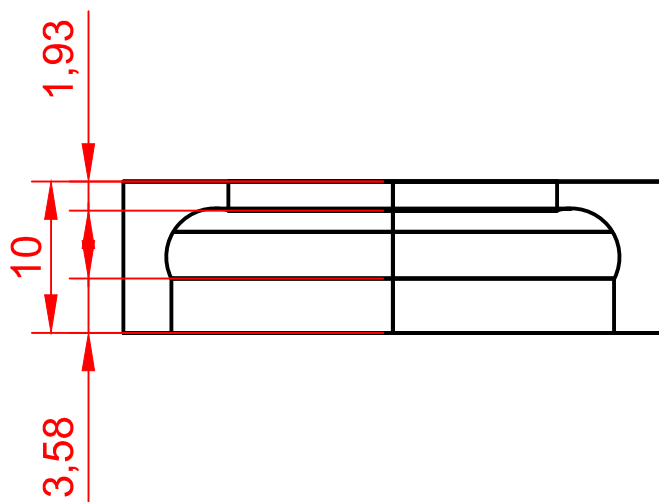


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de tracción engranaje central derecho	Pl. PB4.2 3D

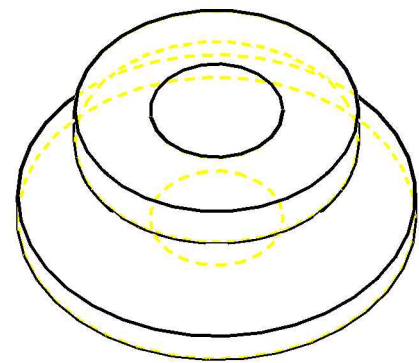
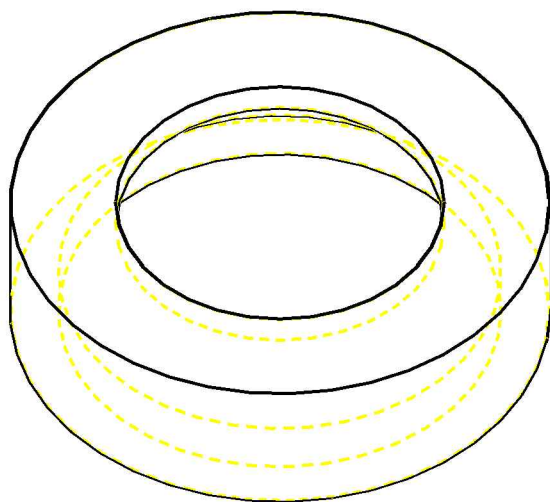


Planta

Alzado

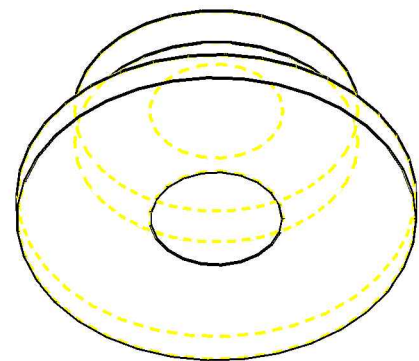
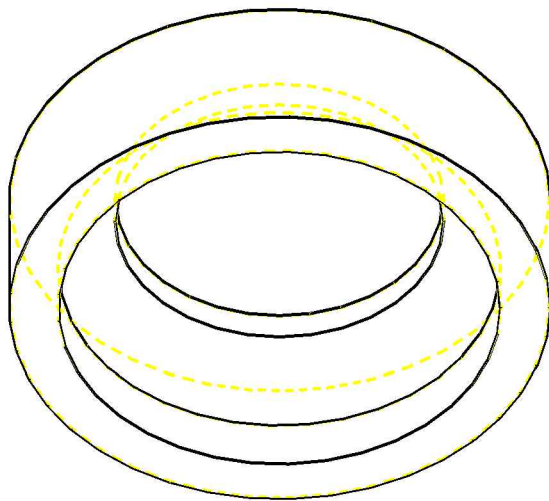


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
2:1	Pieza de tracción centrador y canastilla de rodamiento	PI. PB5

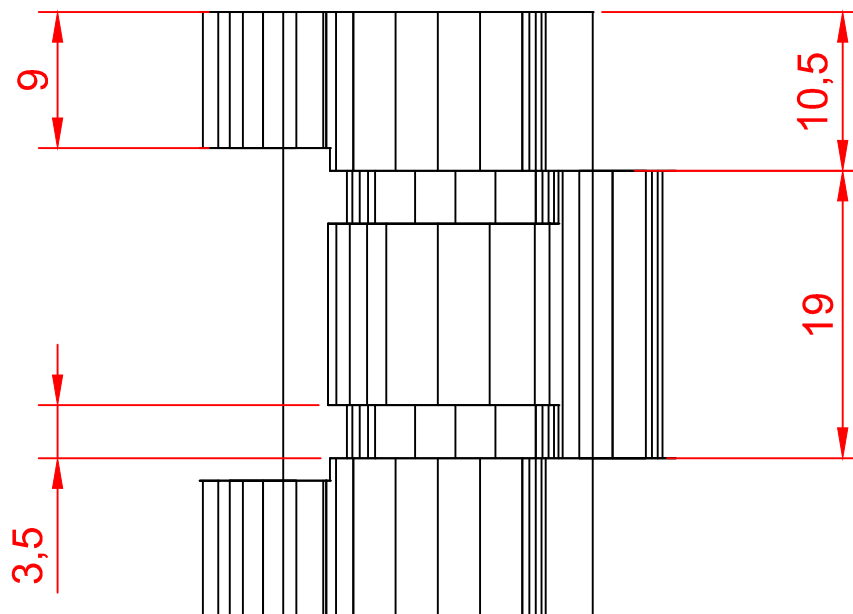


Vista Superior, Suroeste

Vista Inferior, Suroeste

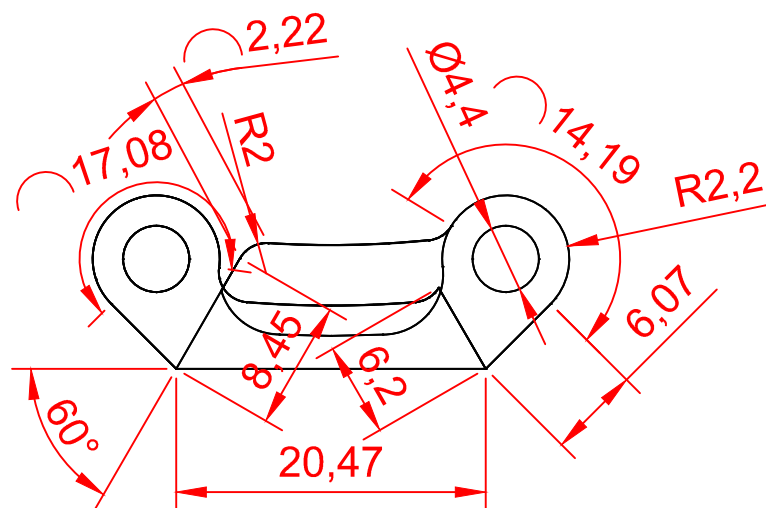


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de tracción centrador y canastilla de rodamiento	Pl. PB5 3D

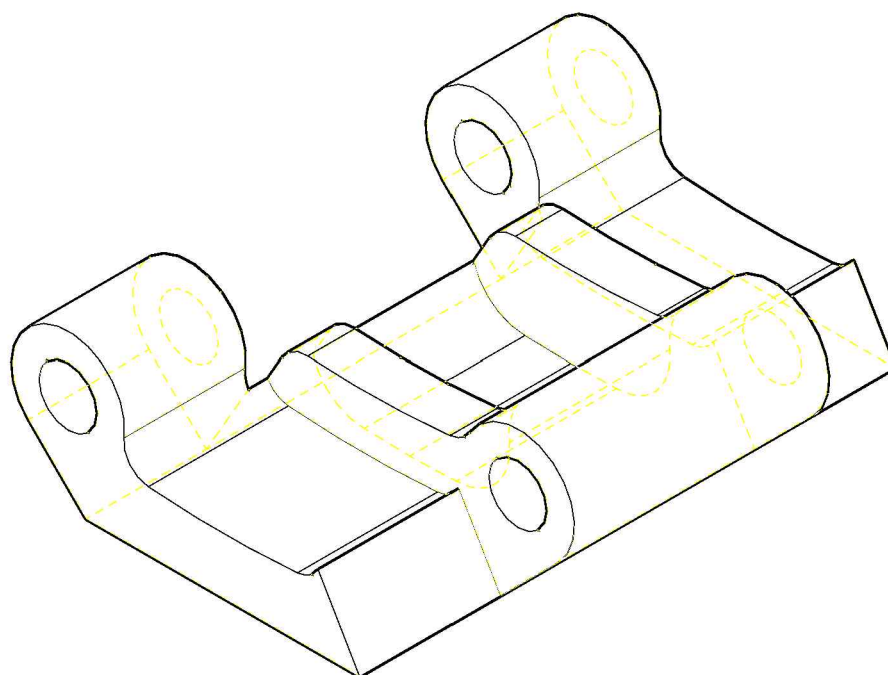


Planta

Alzado

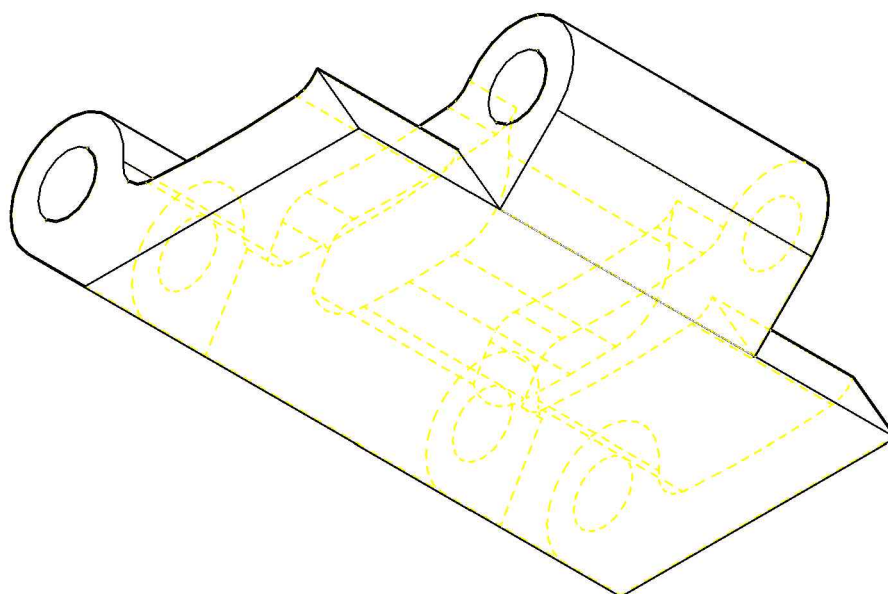


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
2:1	Pieza de tracción eslabón	Pl. PB6



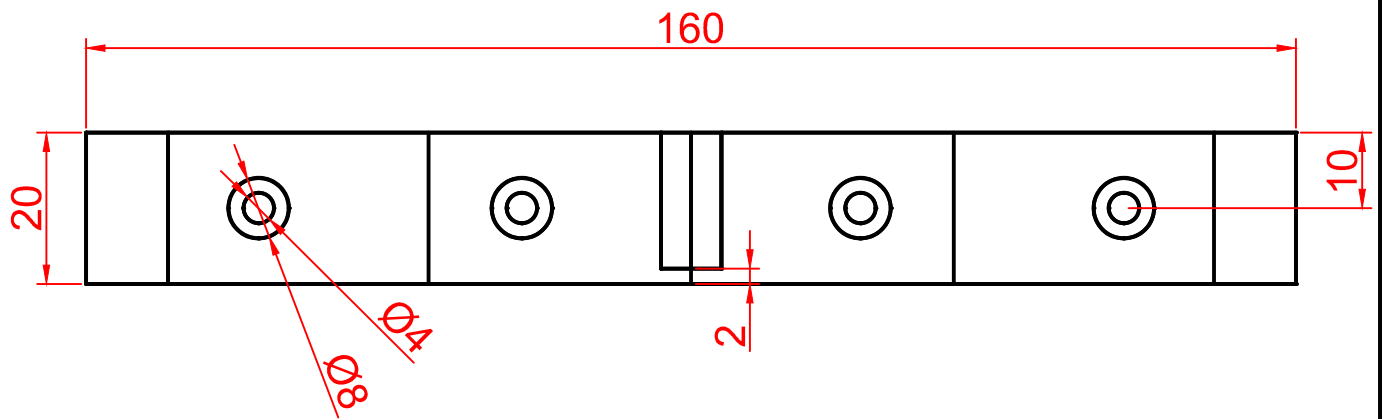
Vista Superior, Suroeste

Vista Inferior, Suroeste



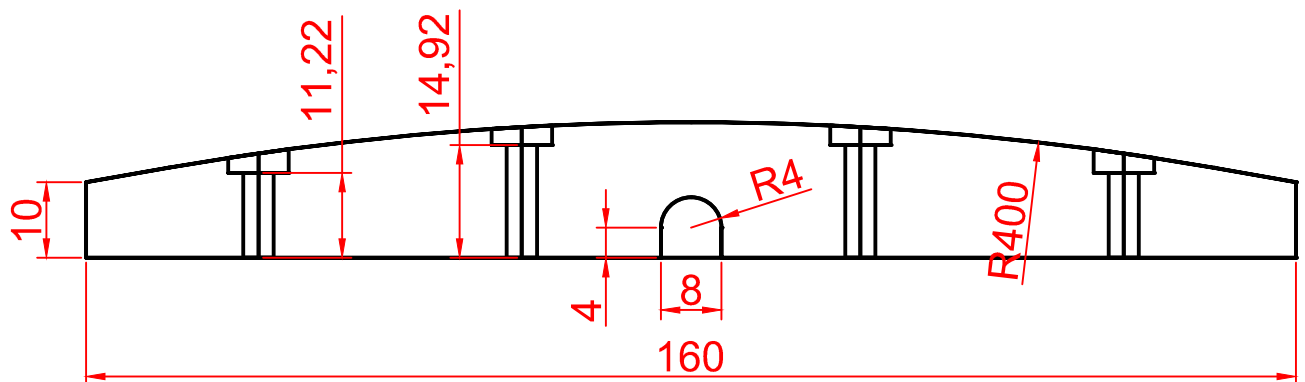
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de tracción eslabón	Pl. PD6 3D



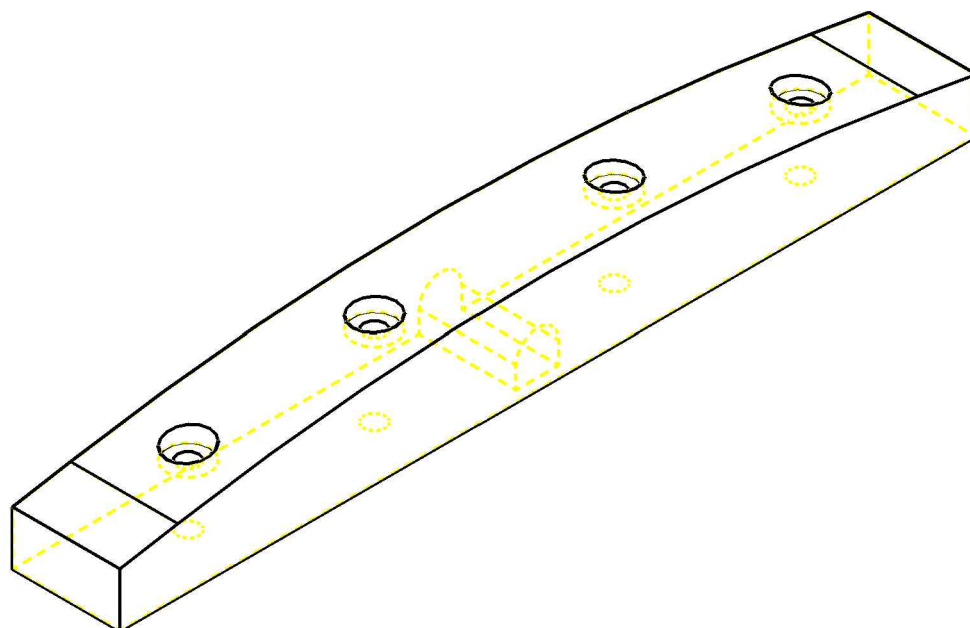


Planta

Alzado

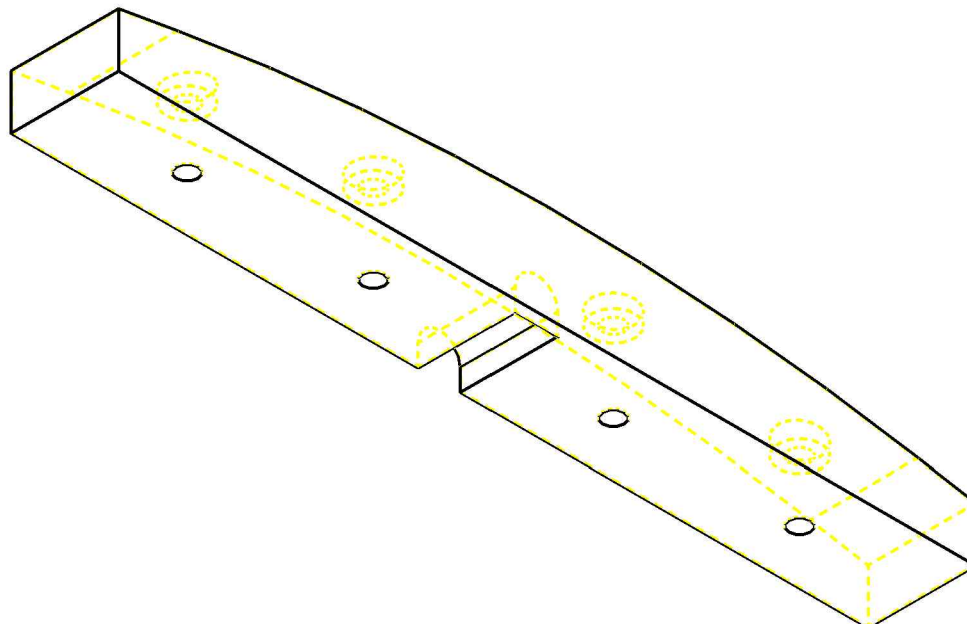


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de amortiguación bancada inferior externa	Pl. PC1.1

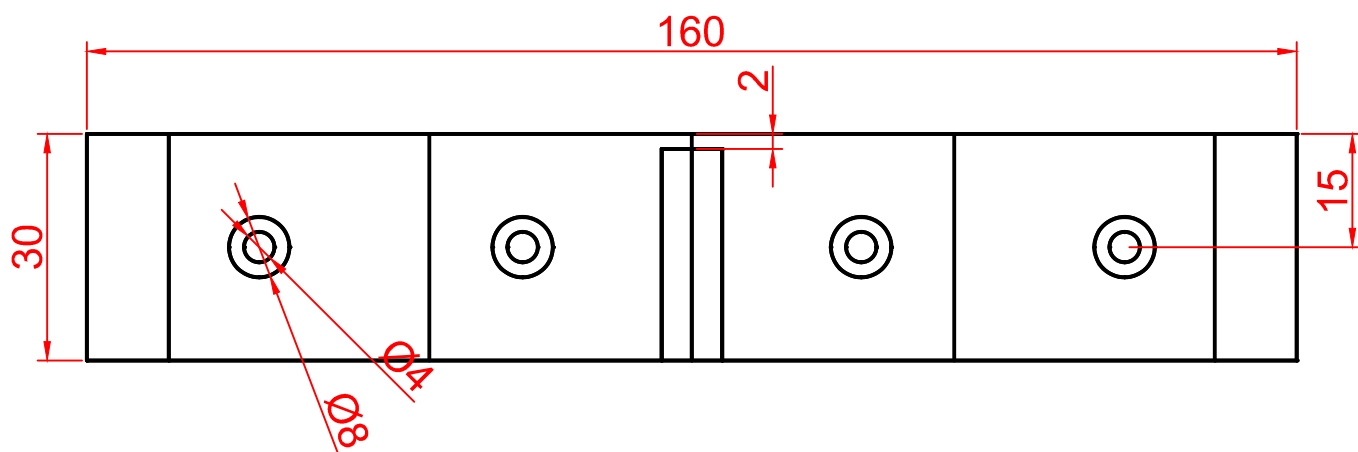


Vista Superior, Suroeste

Vista Inferior, Suroeste

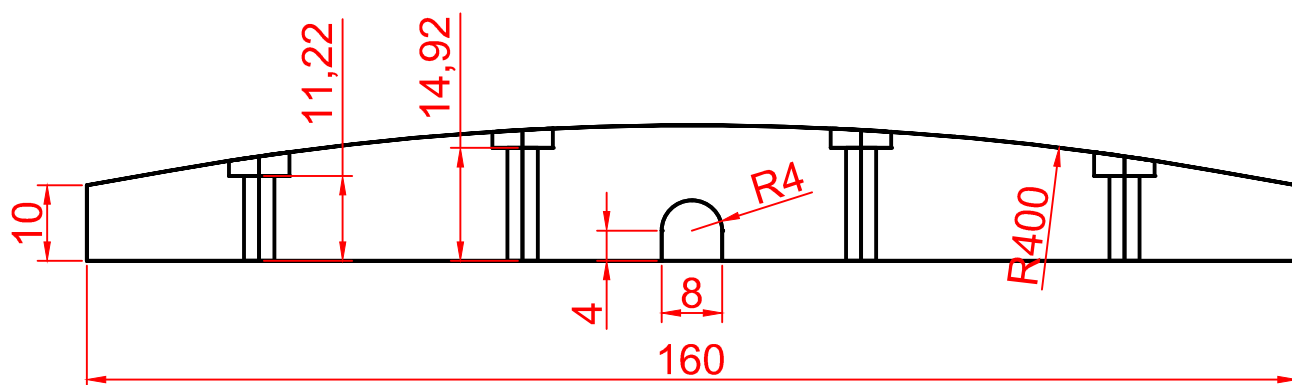


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de amortiguación bancada inferior externa	Pl. PB1.1 3D

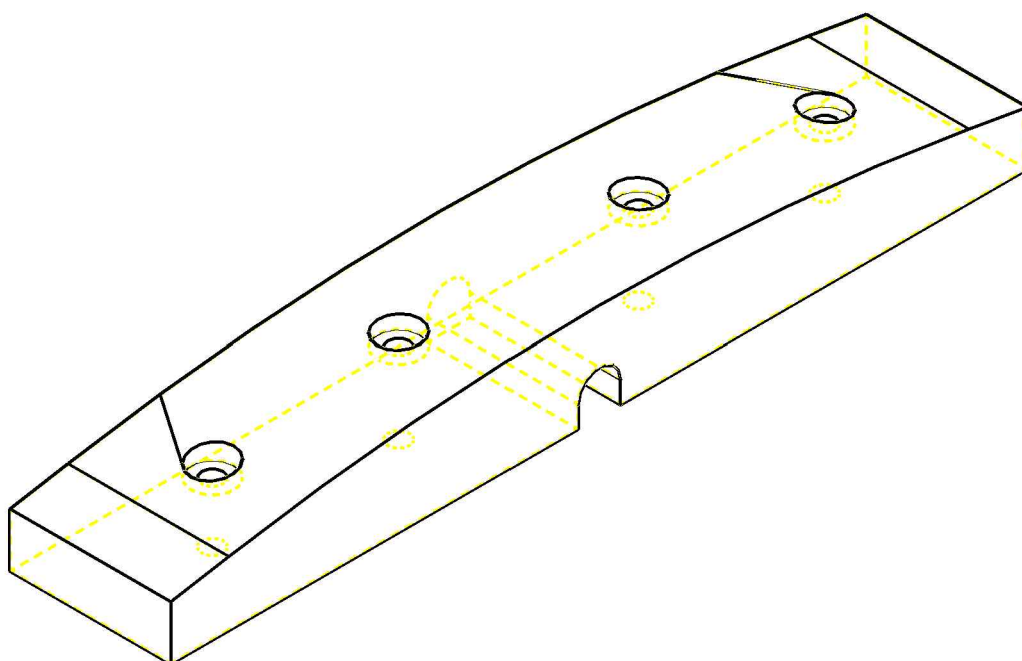


Planta

Alzado

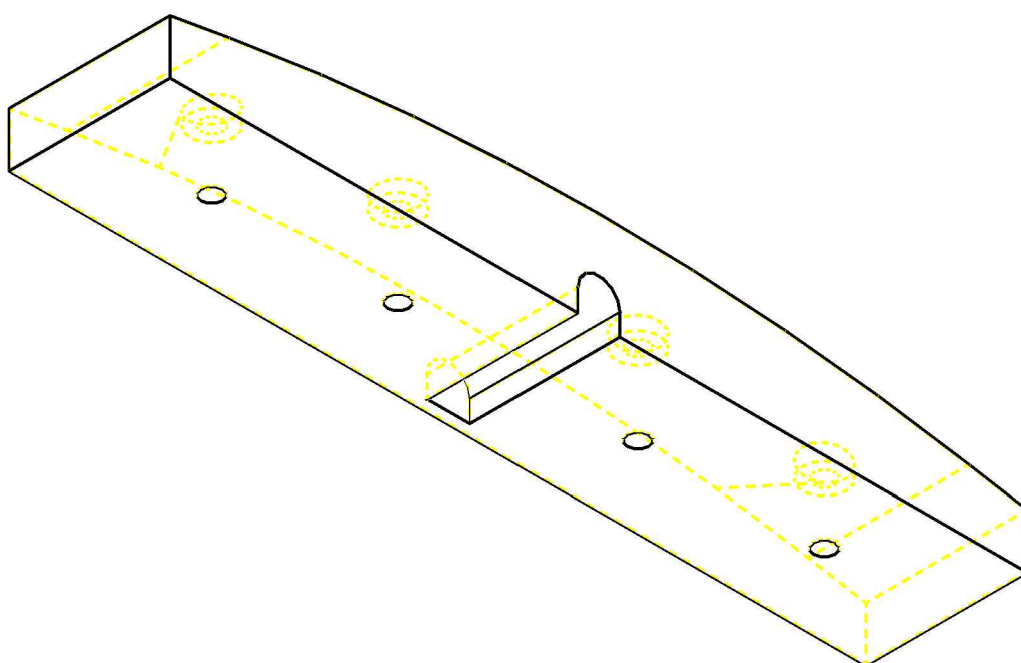


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de amortiguación bancada inferior interna	Pl. PB1.2

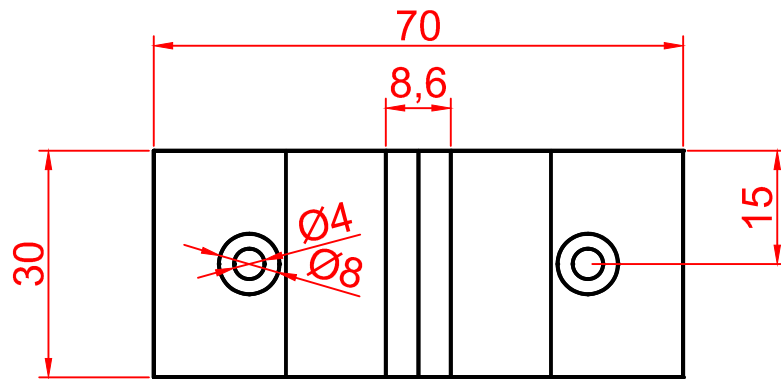


Vista Superior, Suroeste

Vista Inferior, Suroeste

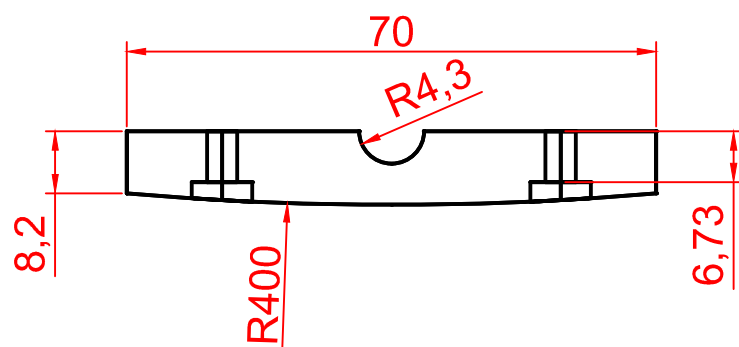


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de amortiguación bancada inferior interna	Pl. PB1.2 3D

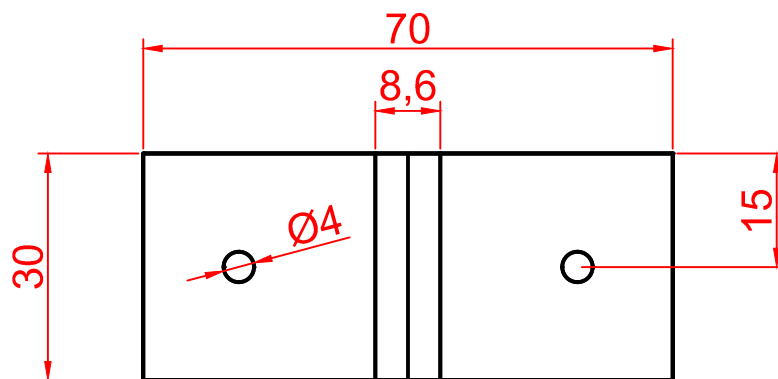


Planta

Alzado

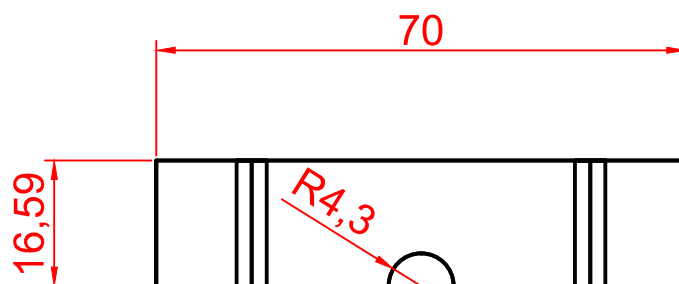


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de amortiguación bancada superior 1	Pl. PC2.1

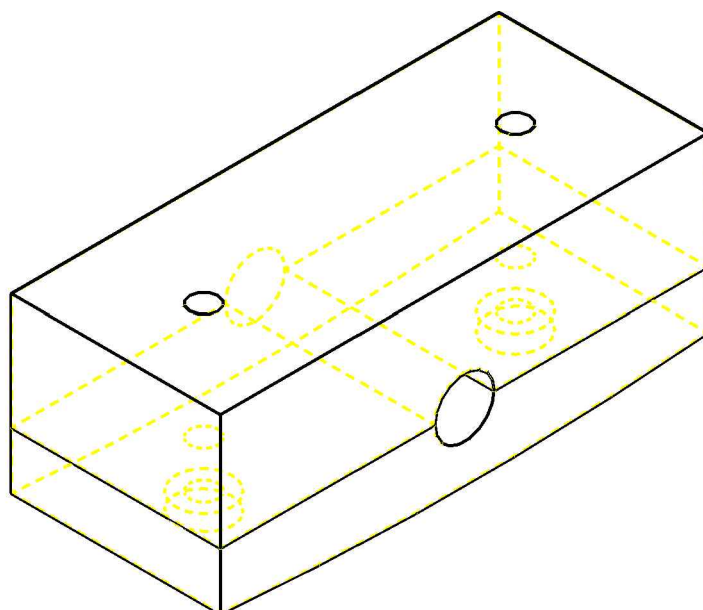


Planta

Alzado

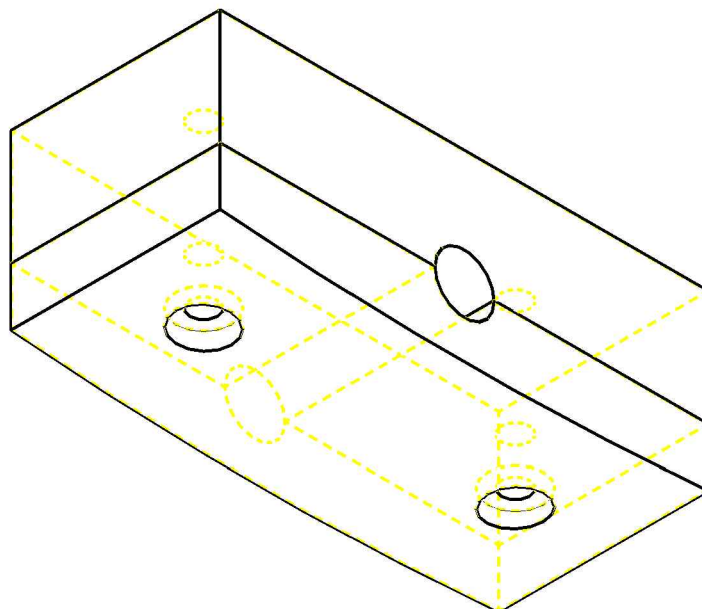


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de amortiguación bancada superior 2	PI. PC2.2

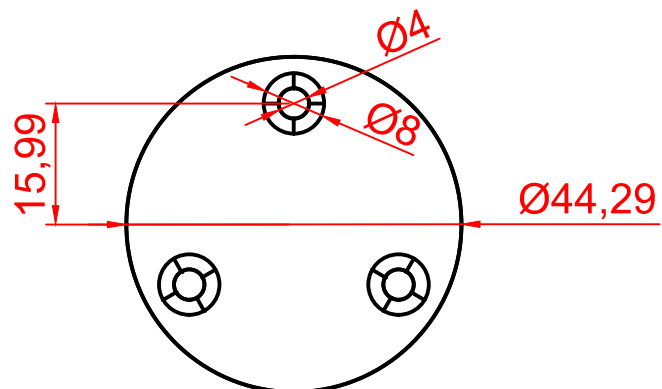


Vista Superior, Suroeste

Vista Inferior, Suroeste

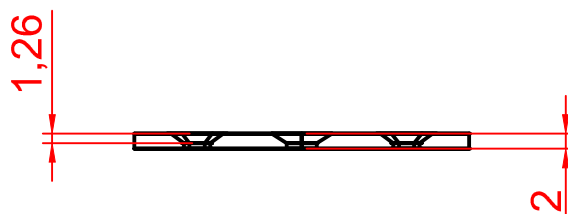


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de amortiguación bancada superior 1 y 2	Pl. PC2.3 3D



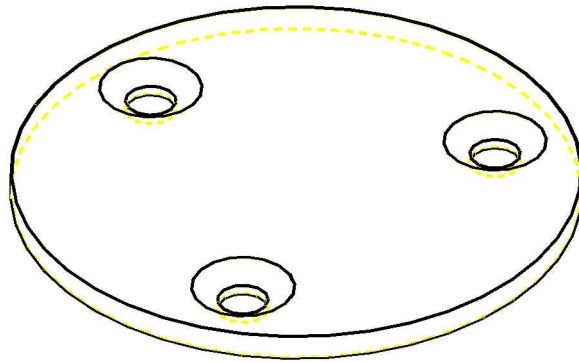
Planta

Alzado



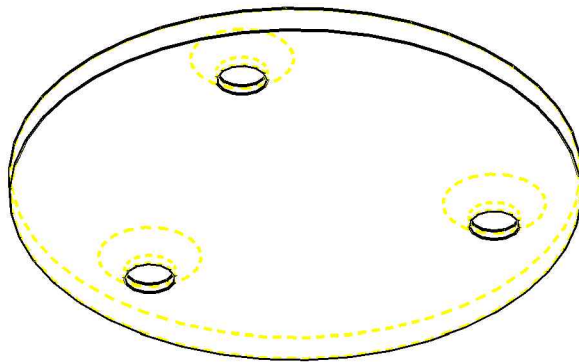
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de amortiguación testigo	Pl. P3.1



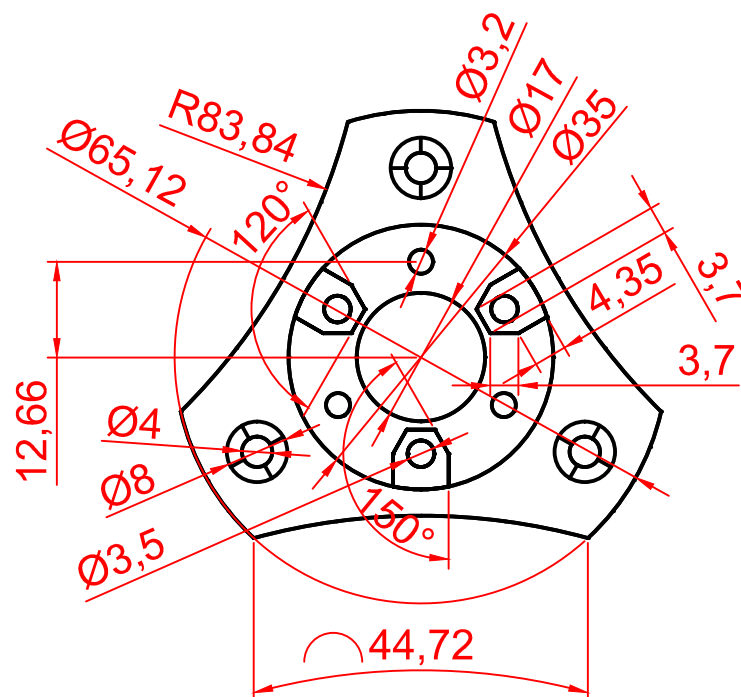


Vista Superior, Suroeste

Vista Inferior, Suroeste

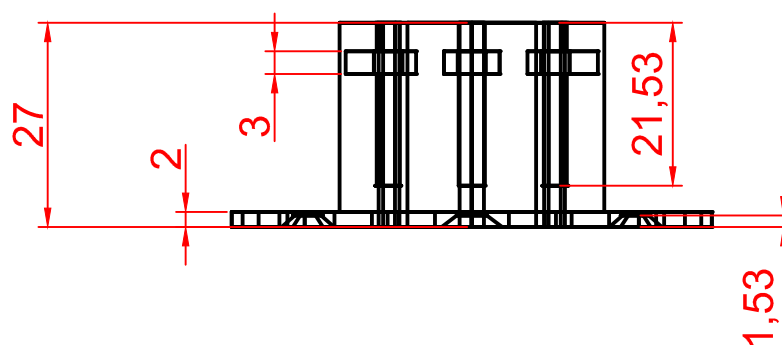


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de amortiguación testigo	Pl. PC3.1 3D

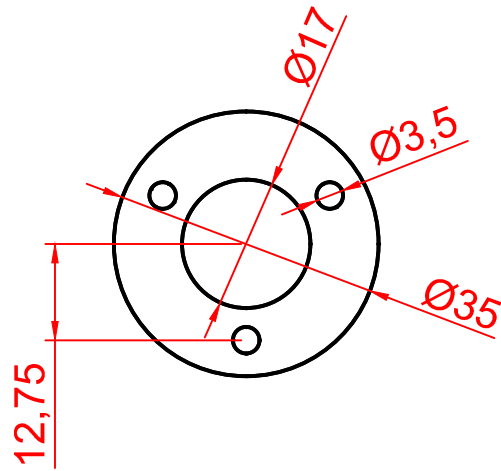


Planta

Alzado

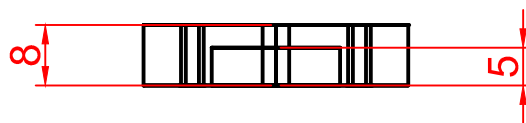


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de amortiguación cilindro de muelle	Pl. PC3.2

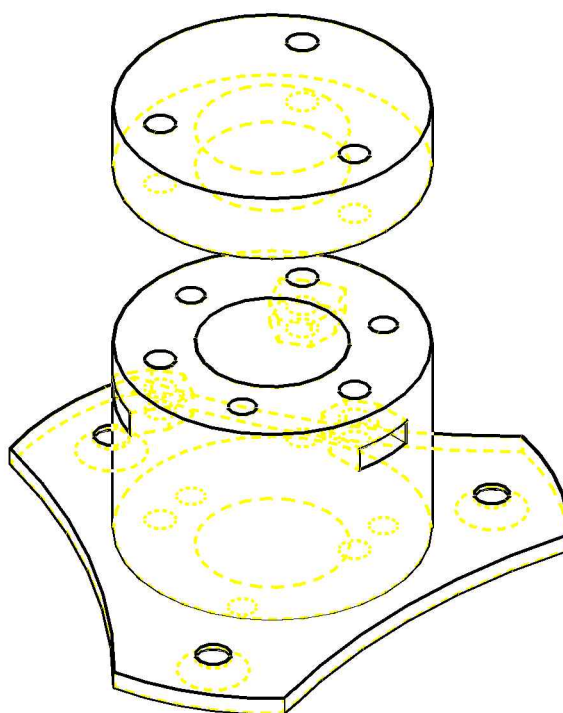


Planta

Alzado

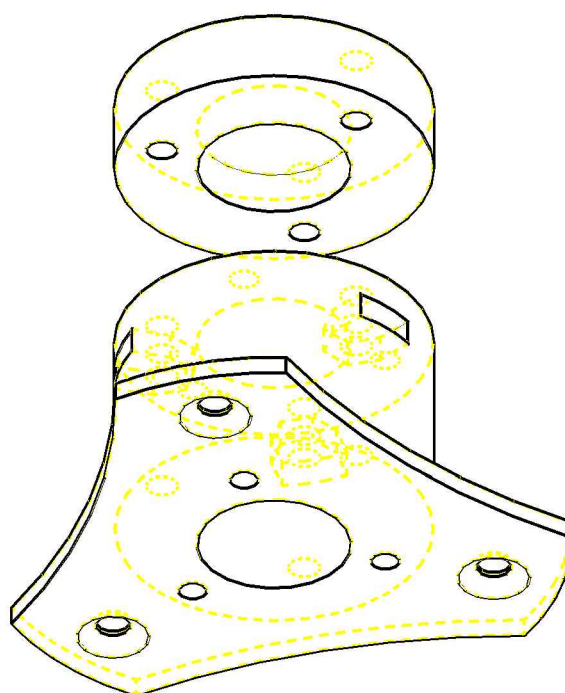


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de amortiguación culata de muelle	Pl. PC3.3

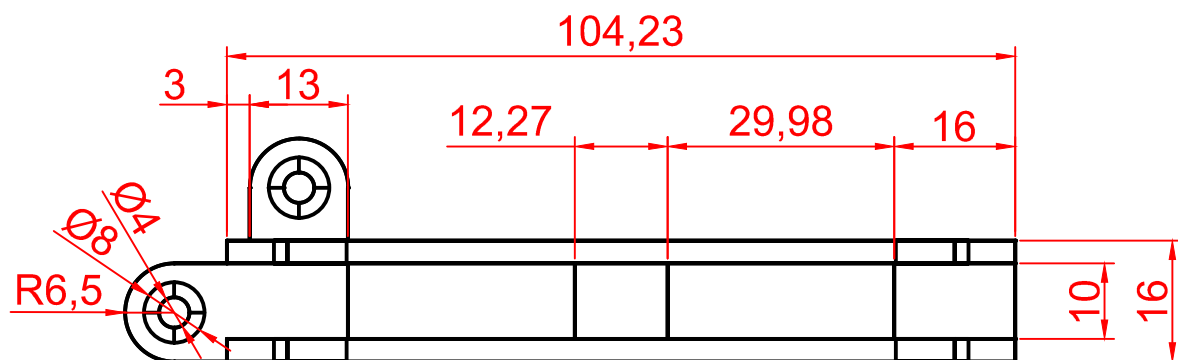


Vista Superior, Suroeste

Vista Inferior, Suroeste

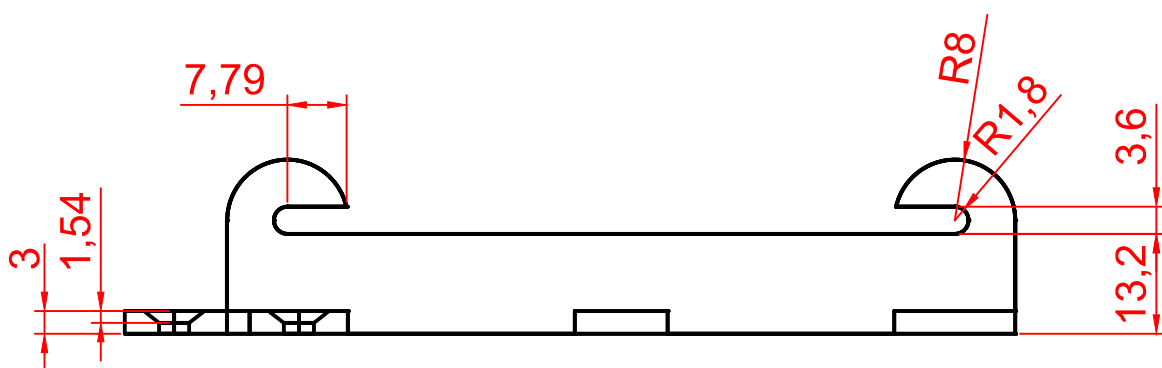


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de amortiguación cilindro y culata de muelle	Pl. PC3.4 3D

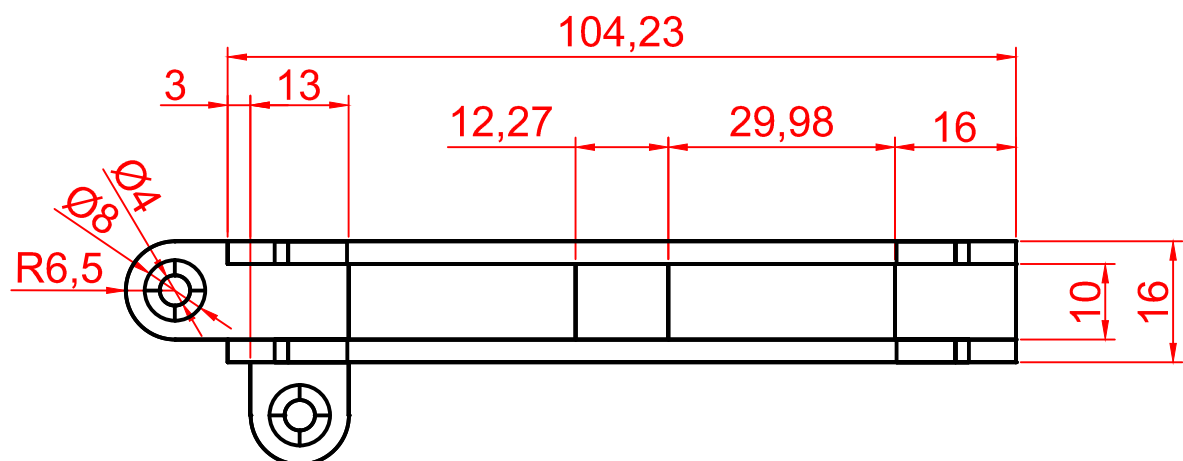


Planta

Alzado

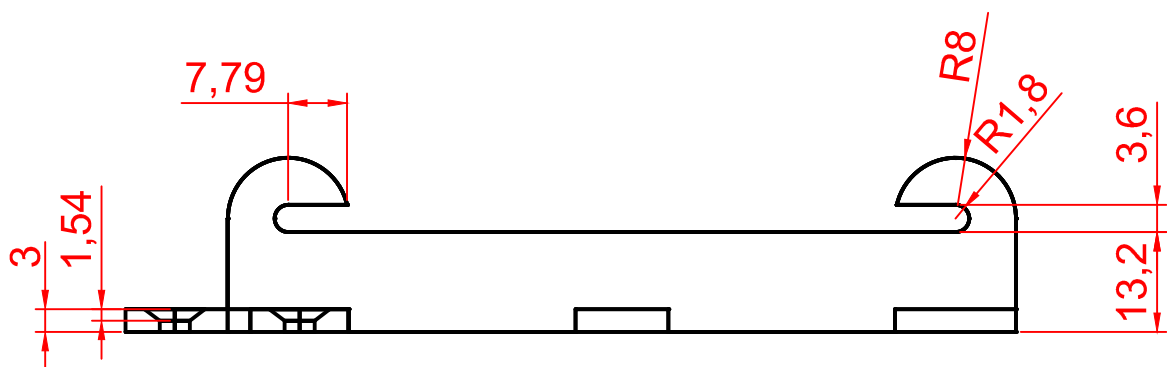


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de ascensión raíl derecho	PI. PD1.1

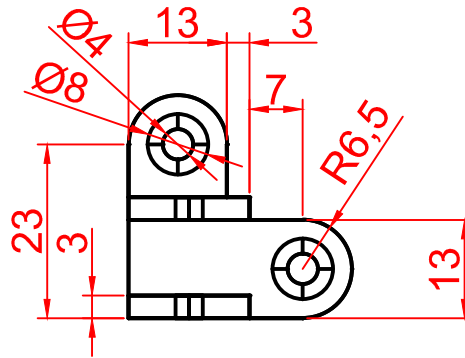


Planta

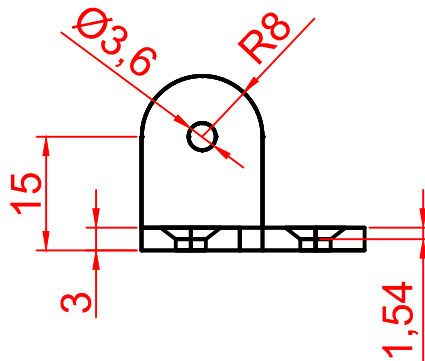
Alzado



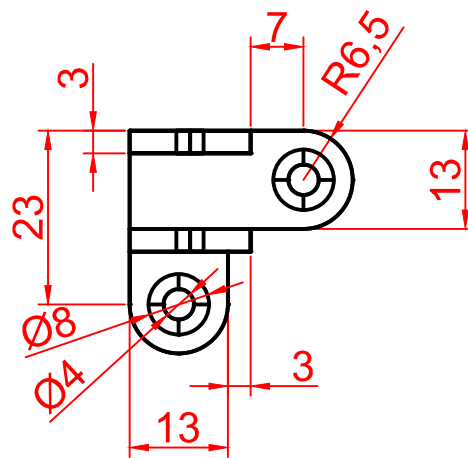
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de ascensión raíl izquierdo	PI. PD1.2



Planta  
Alzado

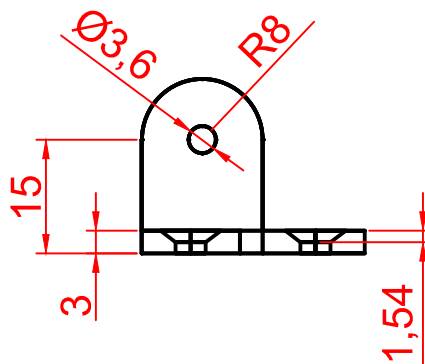


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de ascensión soporte derecho	PI. PD1.3



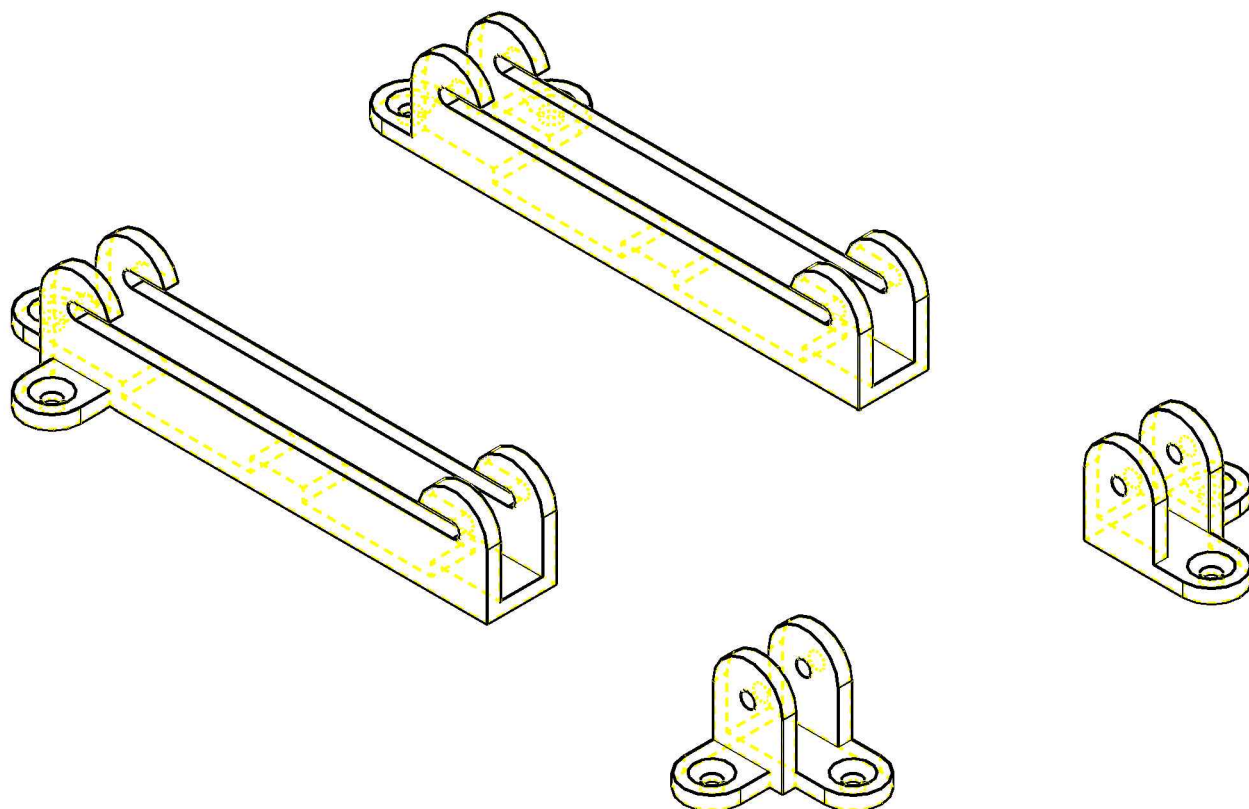
Planta

Alzado



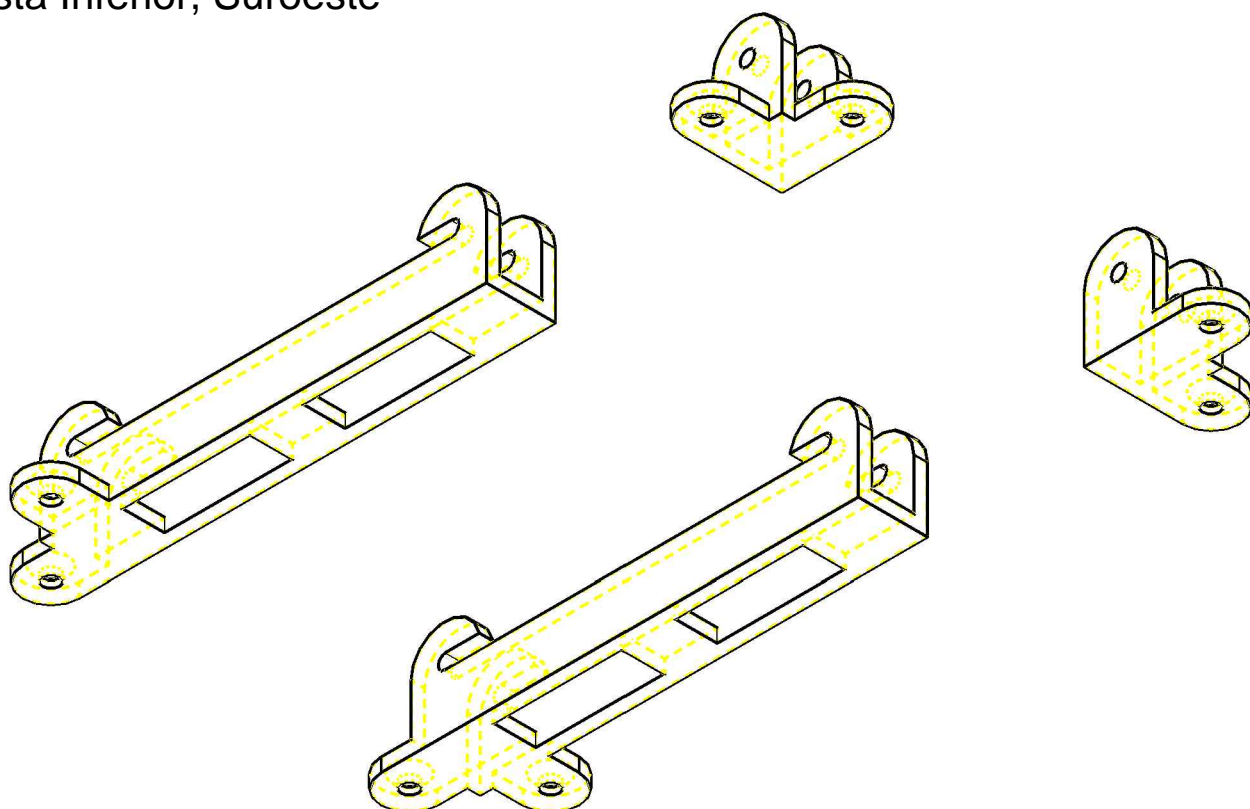
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de ascensión soporte izquierdo	PI. PD1.4





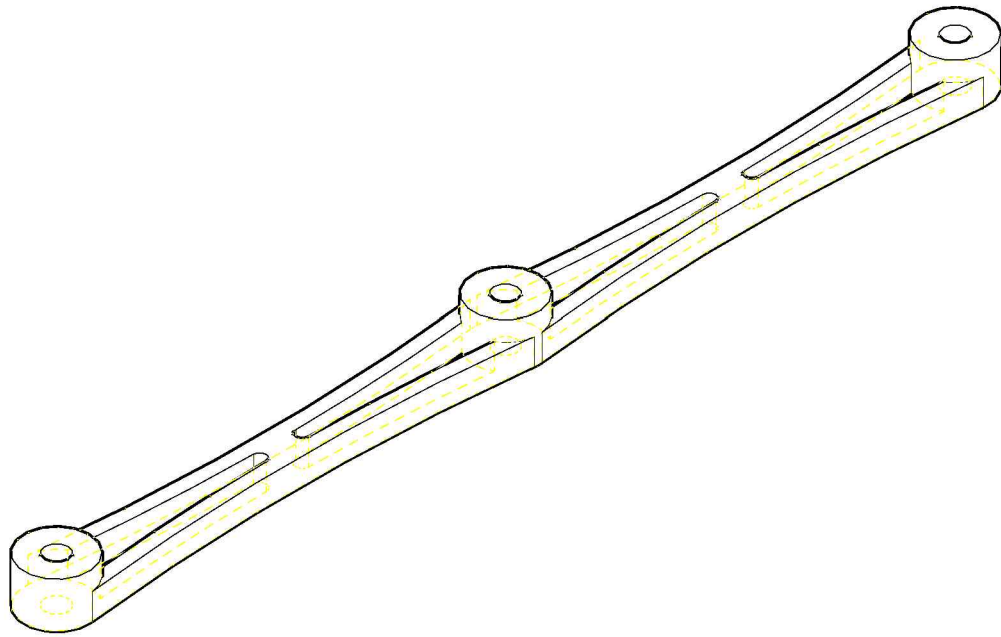
Vista Superior, Suroeste

Vista Inferior, Suroeste



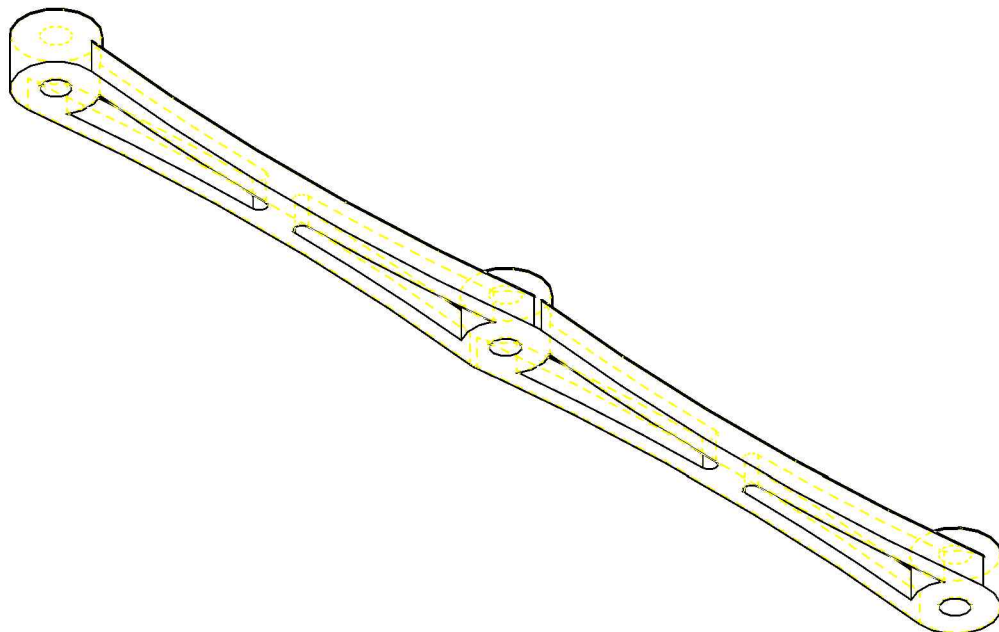
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de ascensión raíles y soportes	Pl. PD1.5 3D

Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de ascensión tijera	Pl. PD2.1

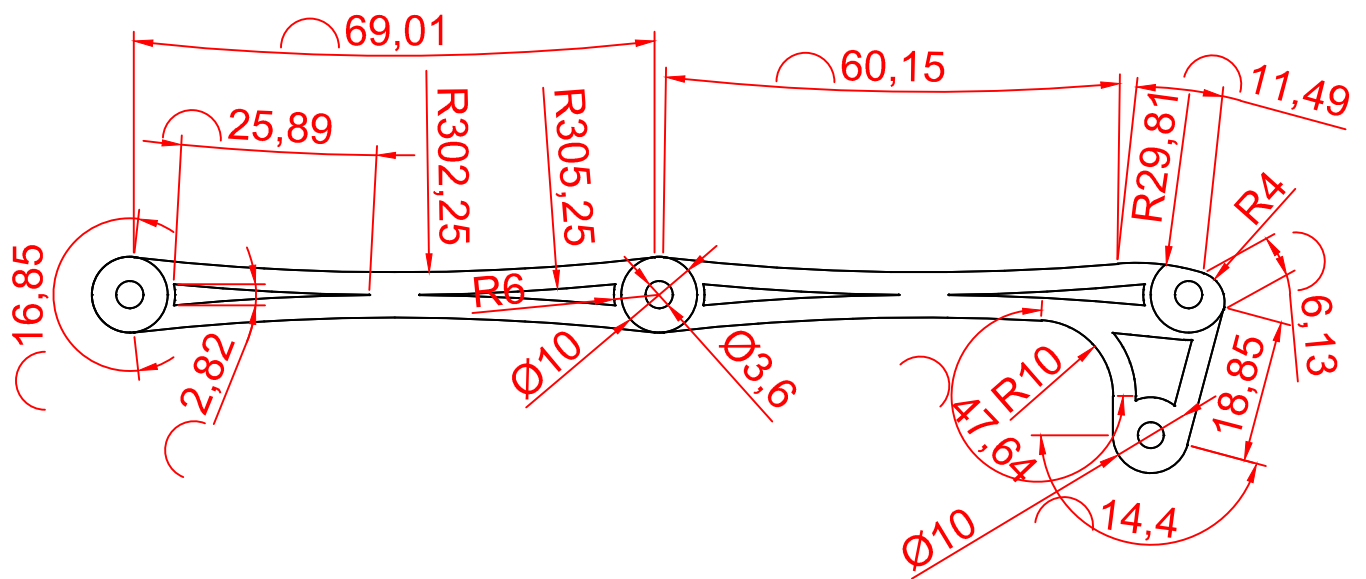


Vista Superior, Suroeste

Vista Inferior, Suroeste

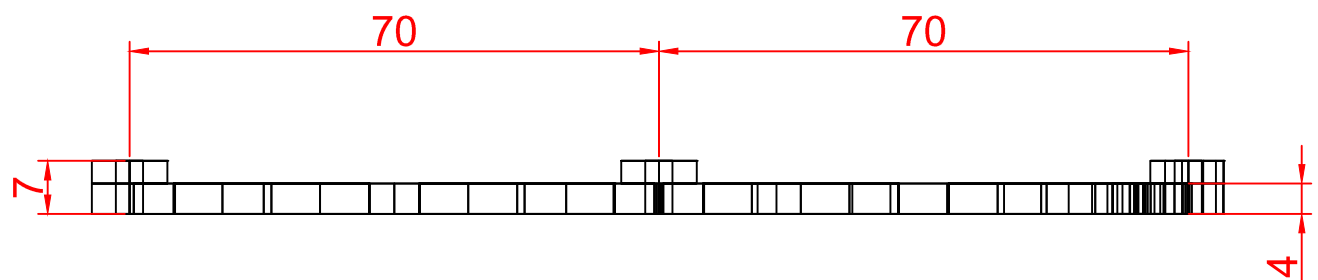


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de ascensión tijera	Pl. PD2.1 3D



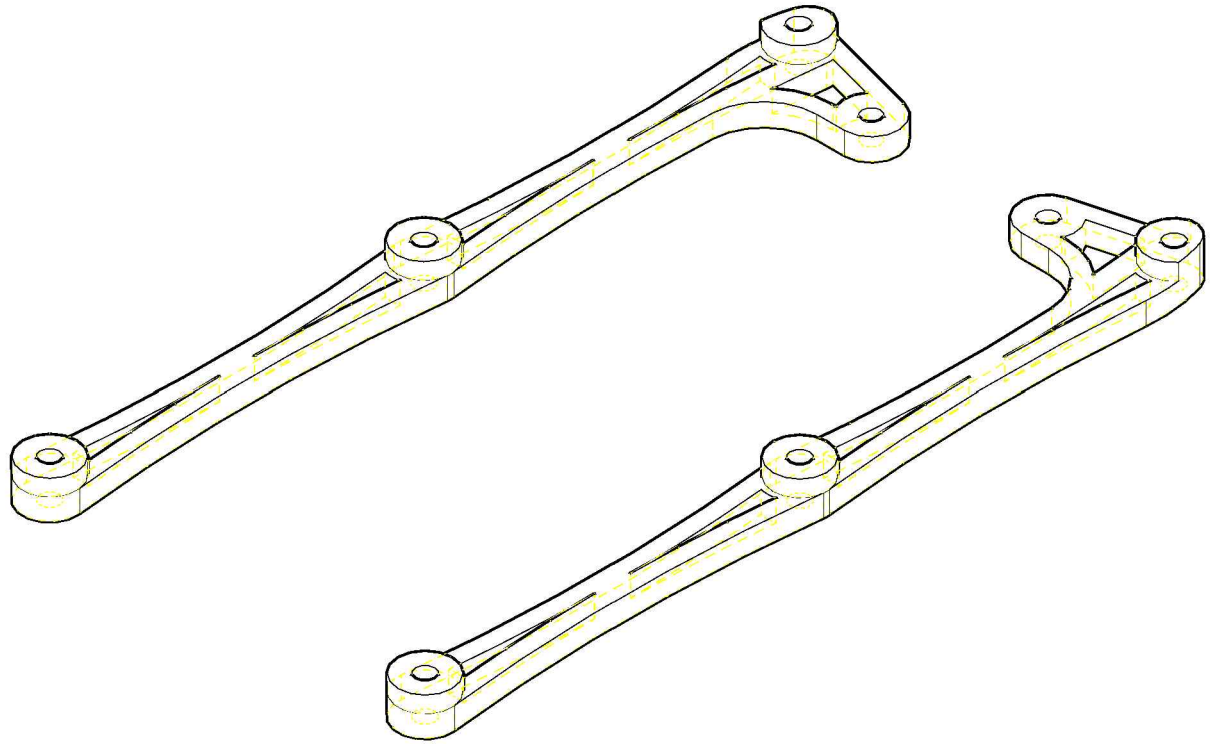
Planta

Alzado



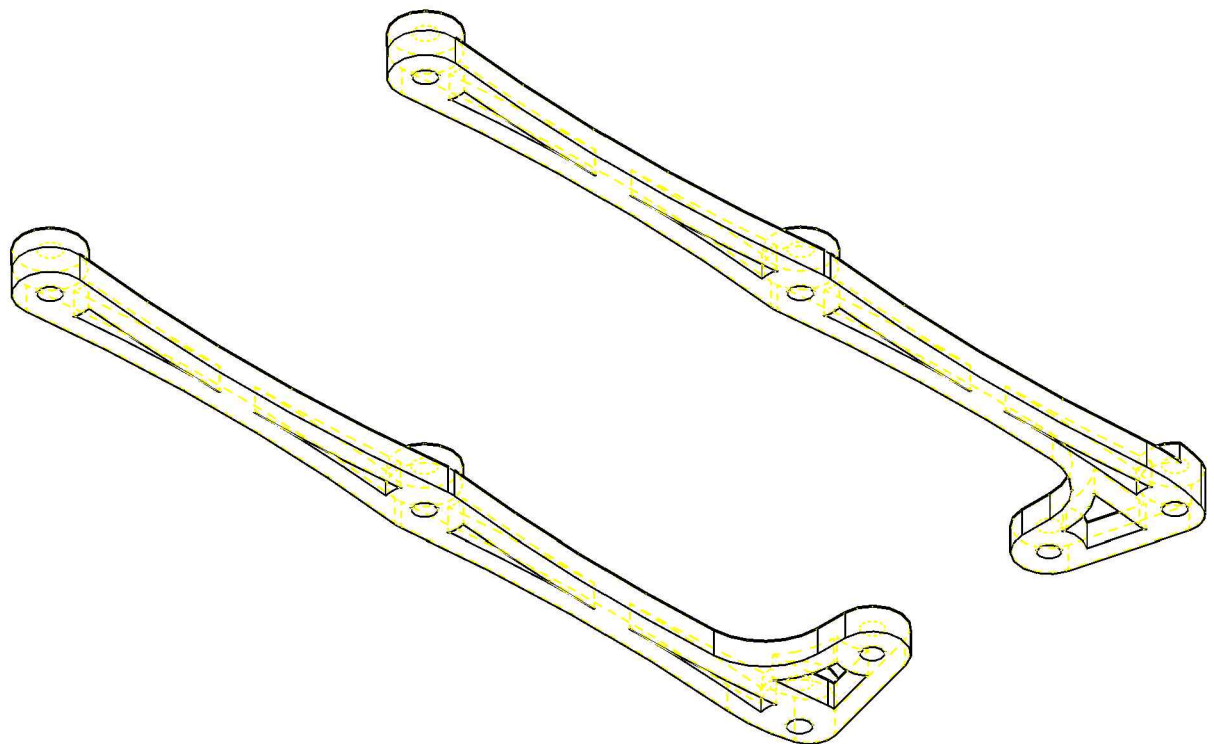
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de ascensión tijera final izquierda	PI. PD2.2

Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de ascensión tijera final derecha	Pl. PD2.3

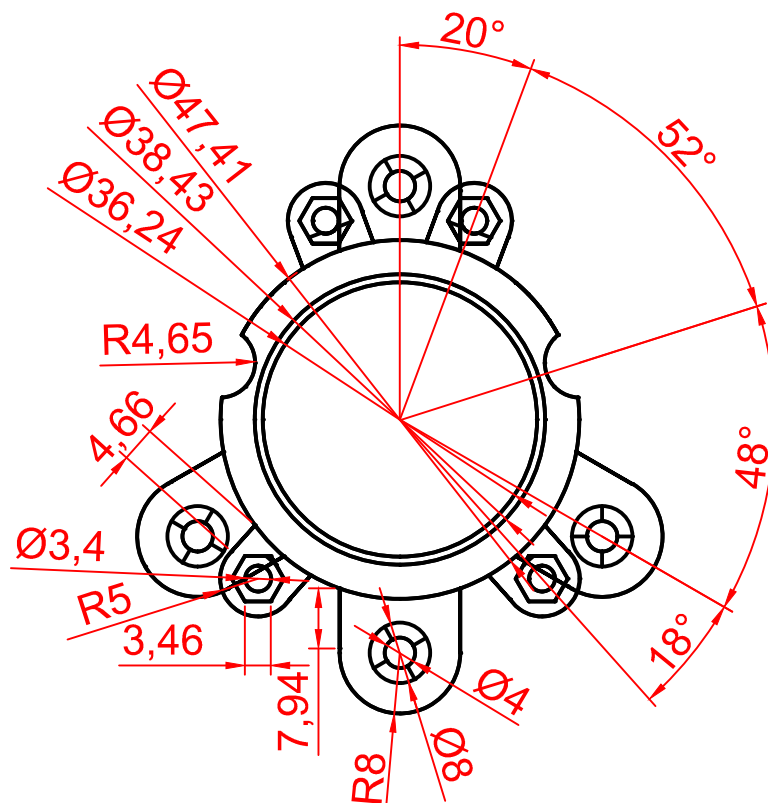


Vista Superior, Suroeste

Vista Inferior, Suroeste

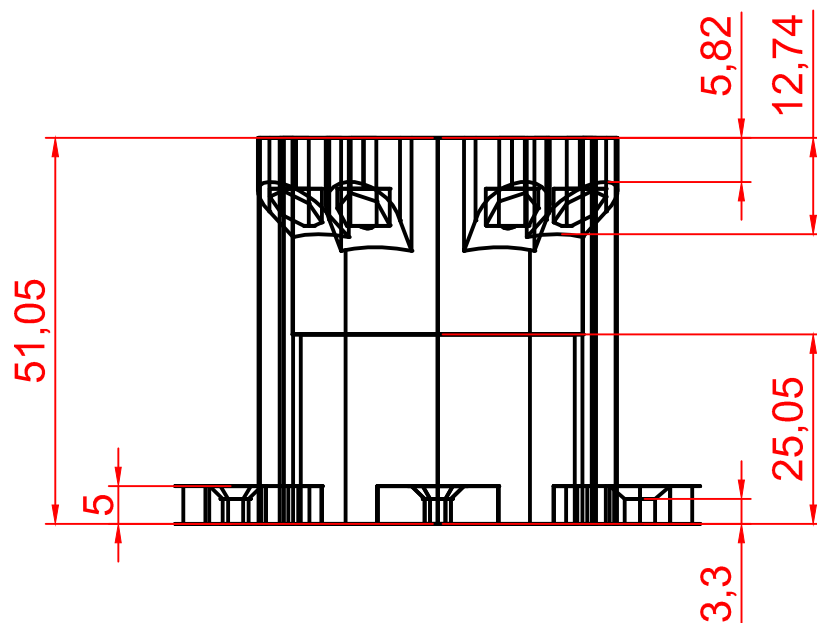


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de ascensión tijera final izquierda y derecha	Pl. PD2.4 3D

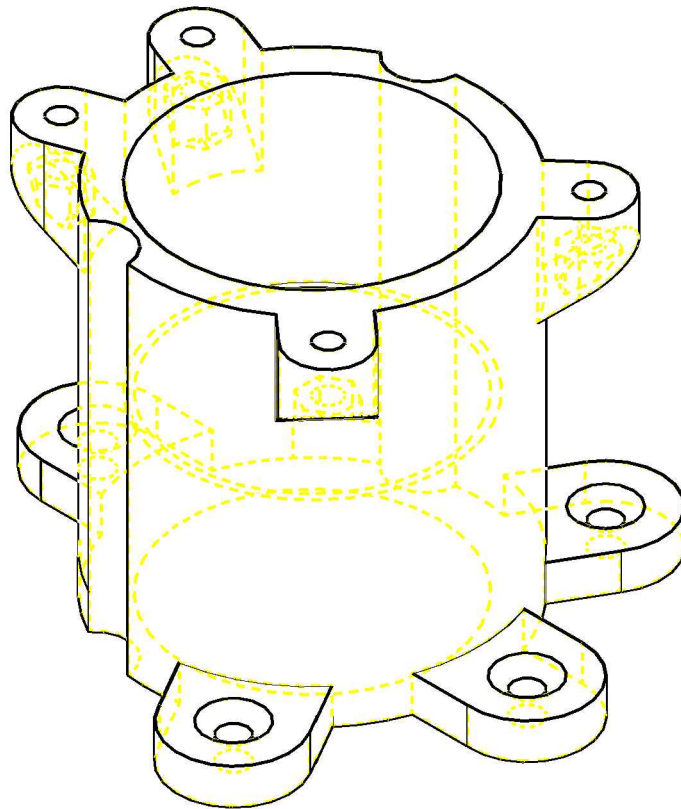


Planta

Alzado

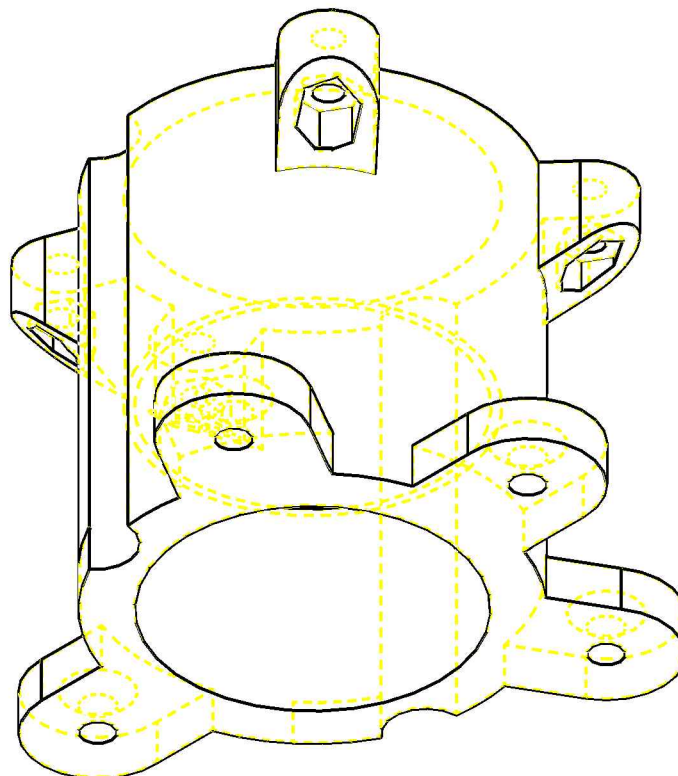


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de ascensión sujeción de motor	PI. PD3.1



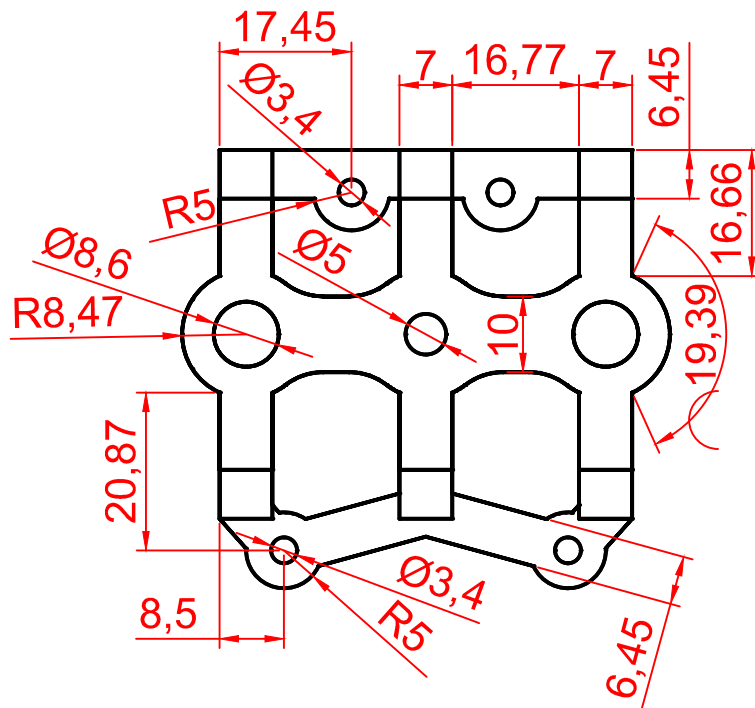
Vista Superior, Suroeste

Vista Inferior, Suroeste



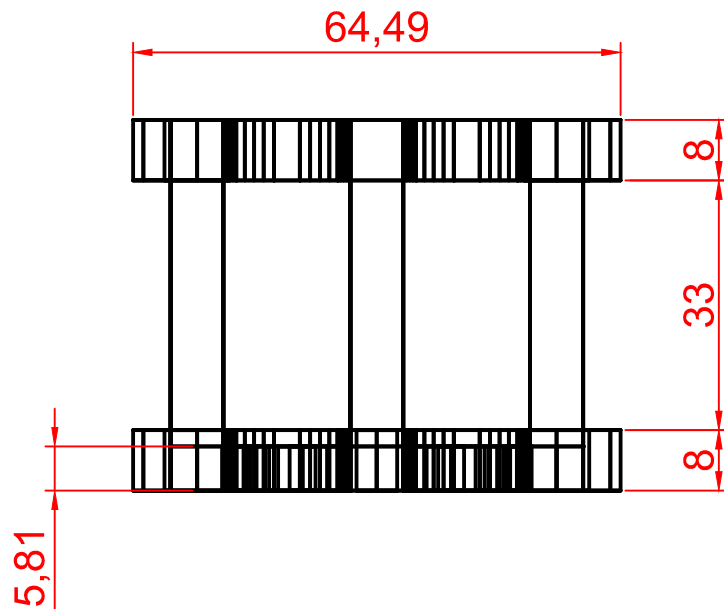
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de ascensión sujeción de motor	Pl. PD3.1 3D



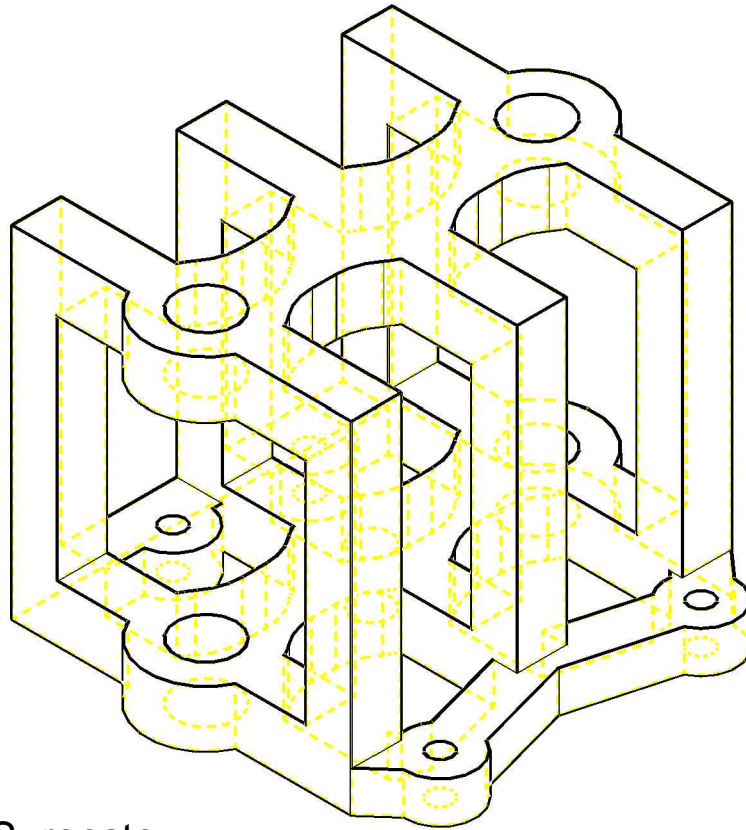


Planta

Alzado

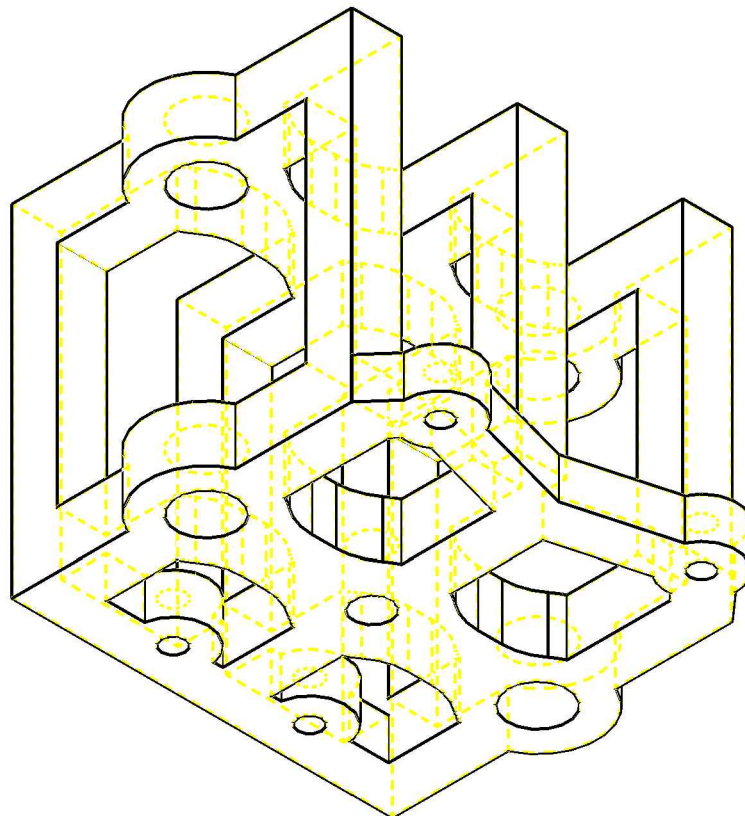


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Pieza de ascensión jaula de engranajes	PI. PD3.2

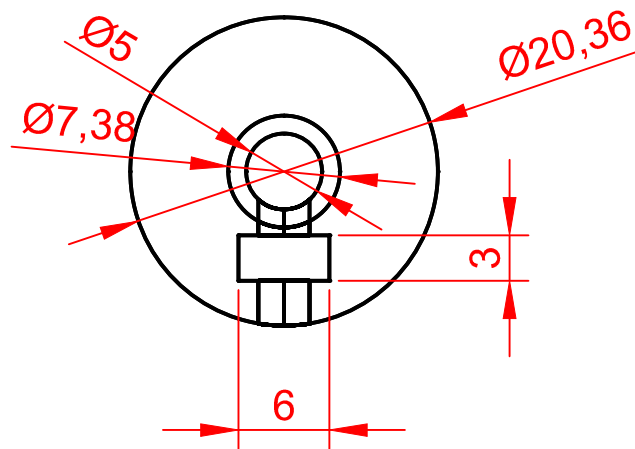


Vista Superior, Suroeste

Vista Inferior, Suroeste

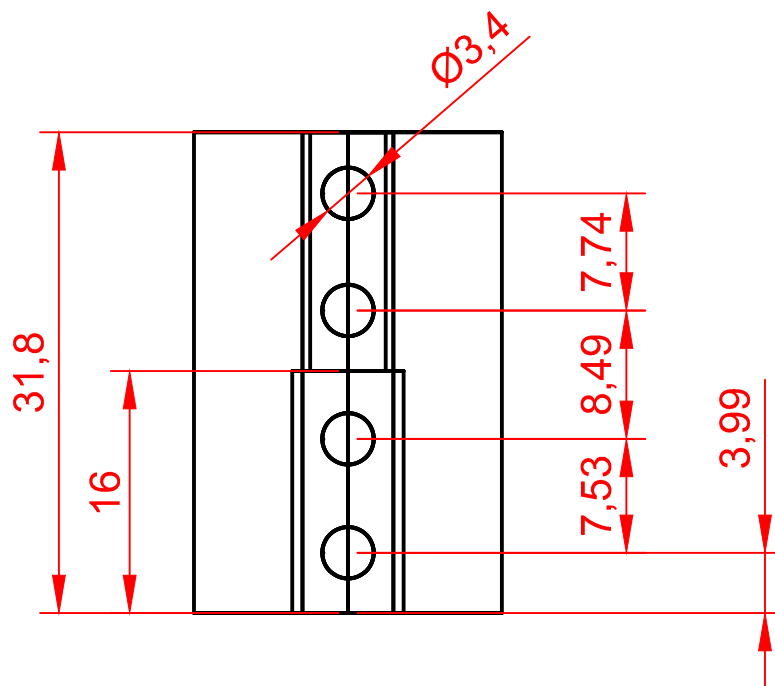


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de ascensión jaula de engranajes	Pl. PD3.2 3D

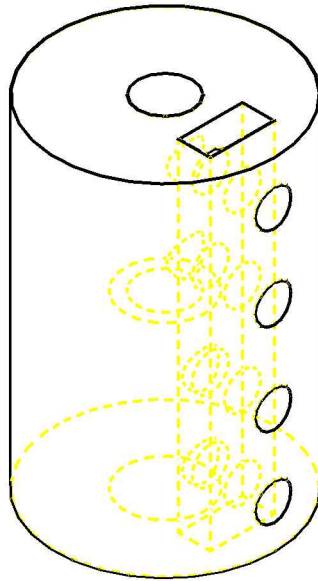


Planta

Alzado

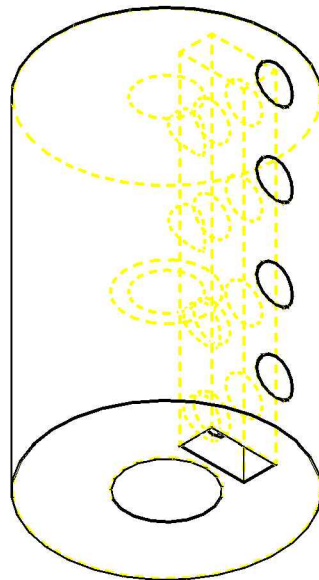


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
2:1	Pieza de ascensión extensor	PI. PD3.3

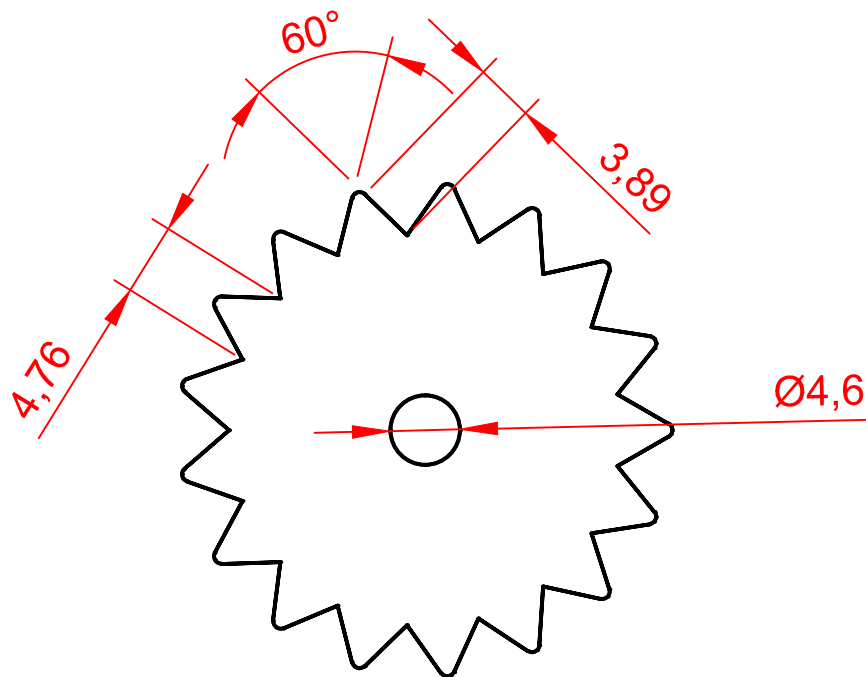


Vista Superior, Suroeste

Vista Inferior, Suroeste

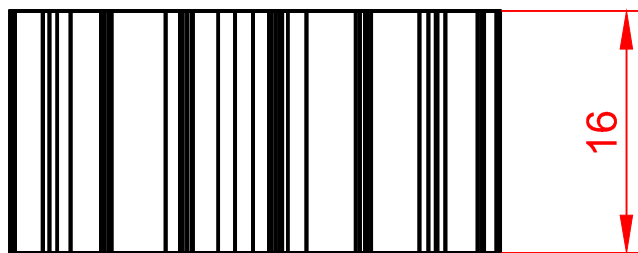


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de ascensión extensor	Pl. PD3.3 3D

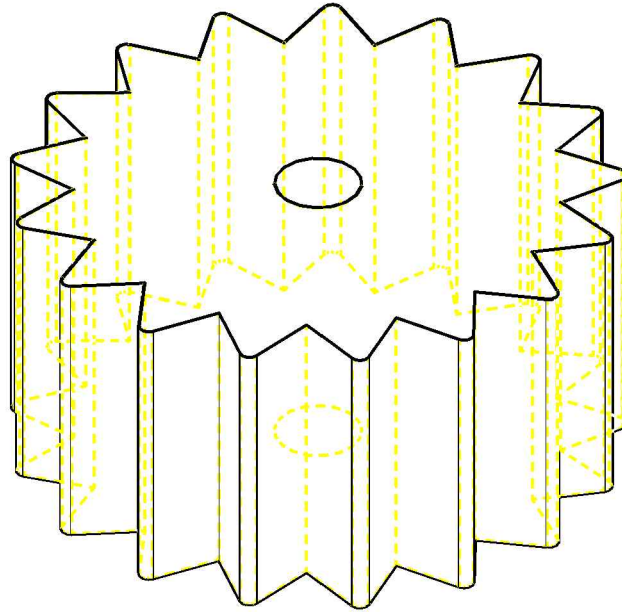


Planta

Alzado

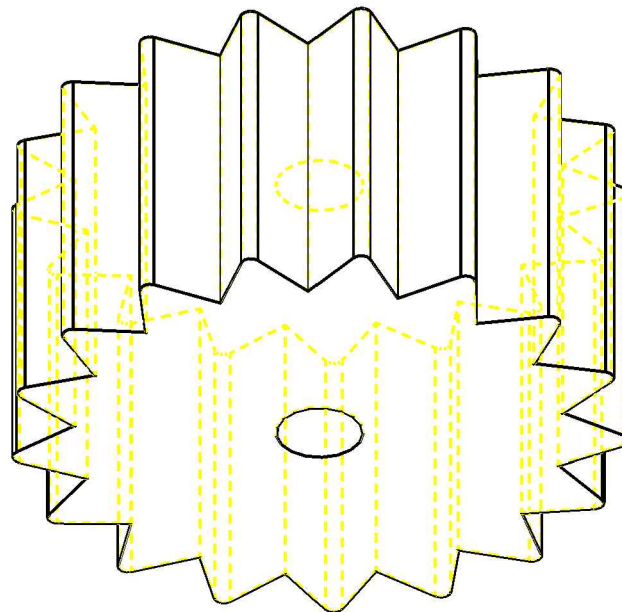


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
2:1	Pieza de ascensión engranaje central	PI. PD3.4

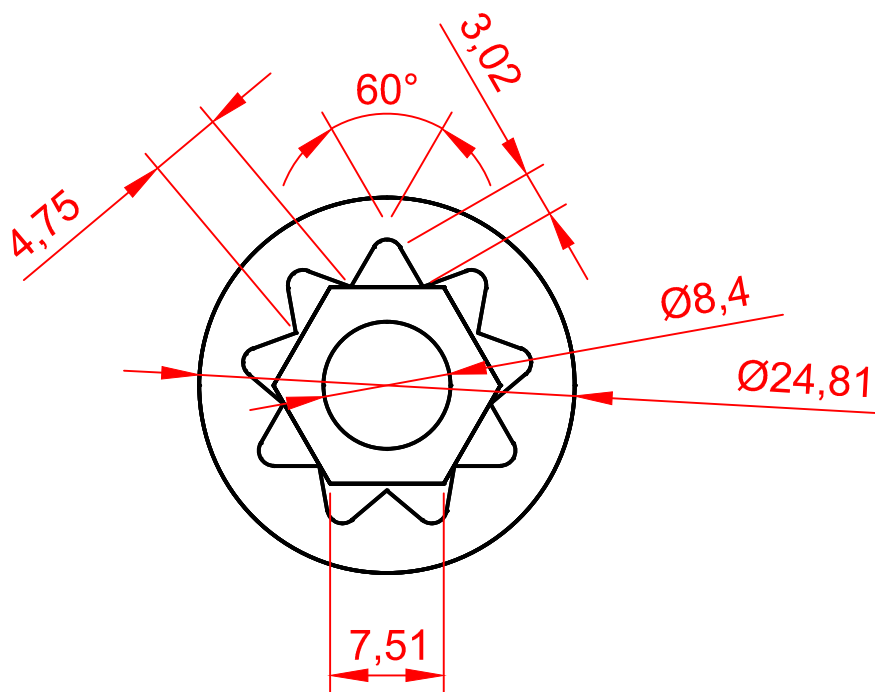


Vista Superior, Suroeste

Vista Inferior, Suroeste



Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de ascensión engranaje central	Pl. PD3.4 3D

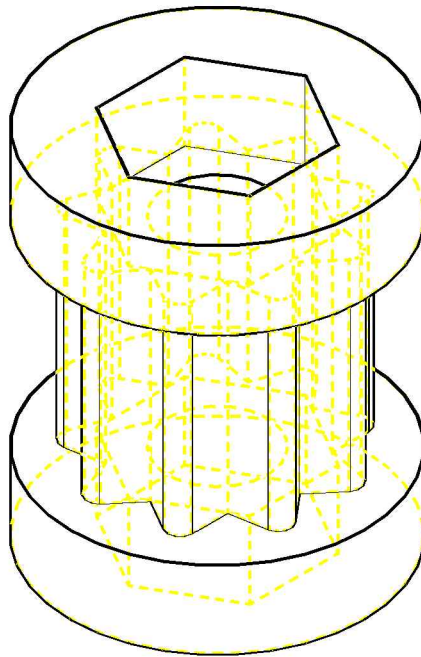


Planta

Alzado

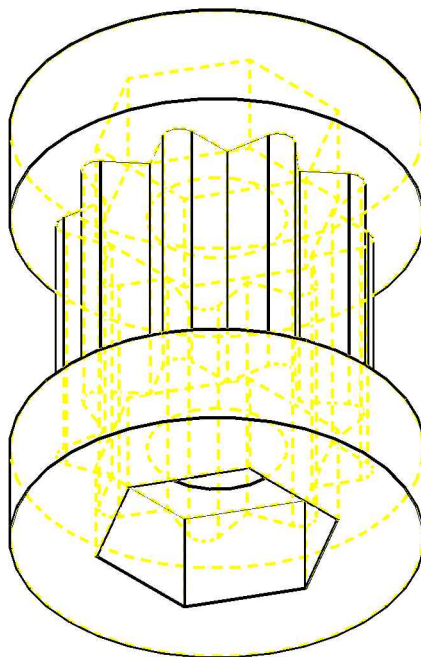


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
2:1	Pieza de ascensión engranaje lateral	PI. PD3.5



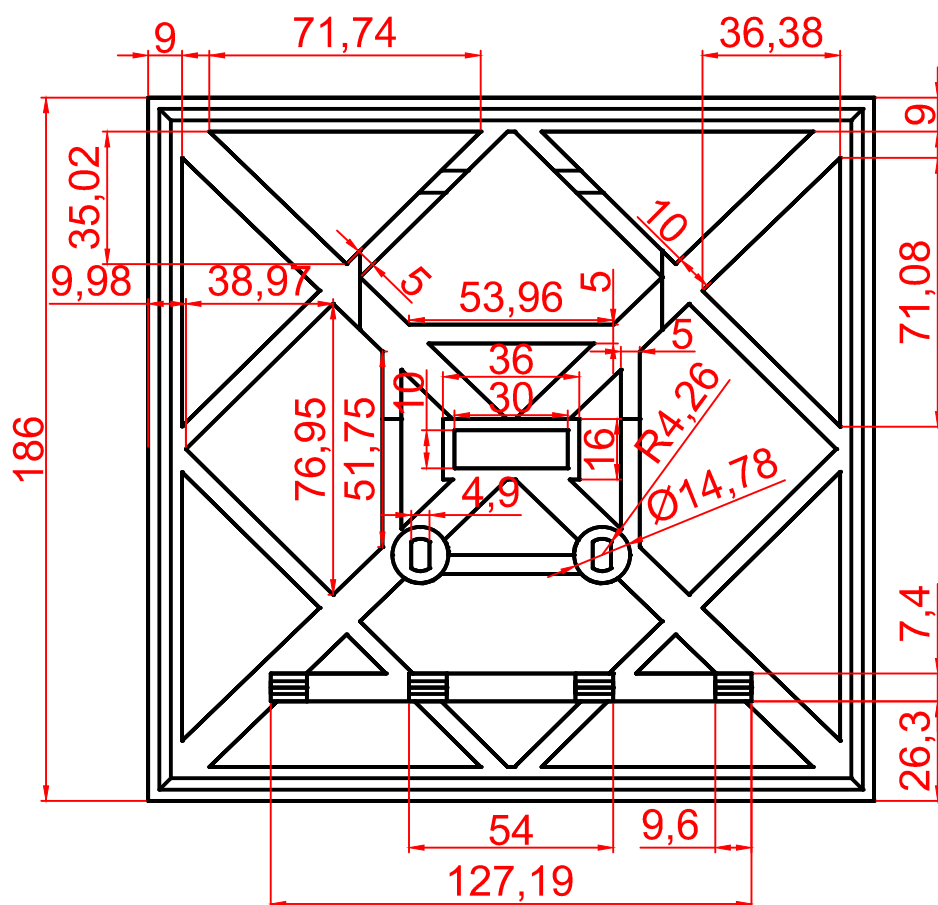
Vista Superior, Suroeste

Vista Inferior, Suroeste



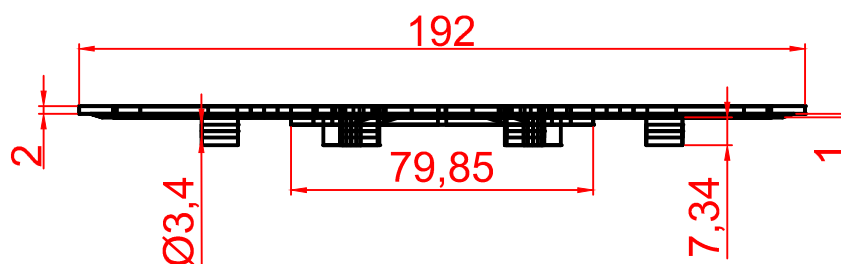
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de ascensión engranaje lateral	Pl. PD3.5 3D



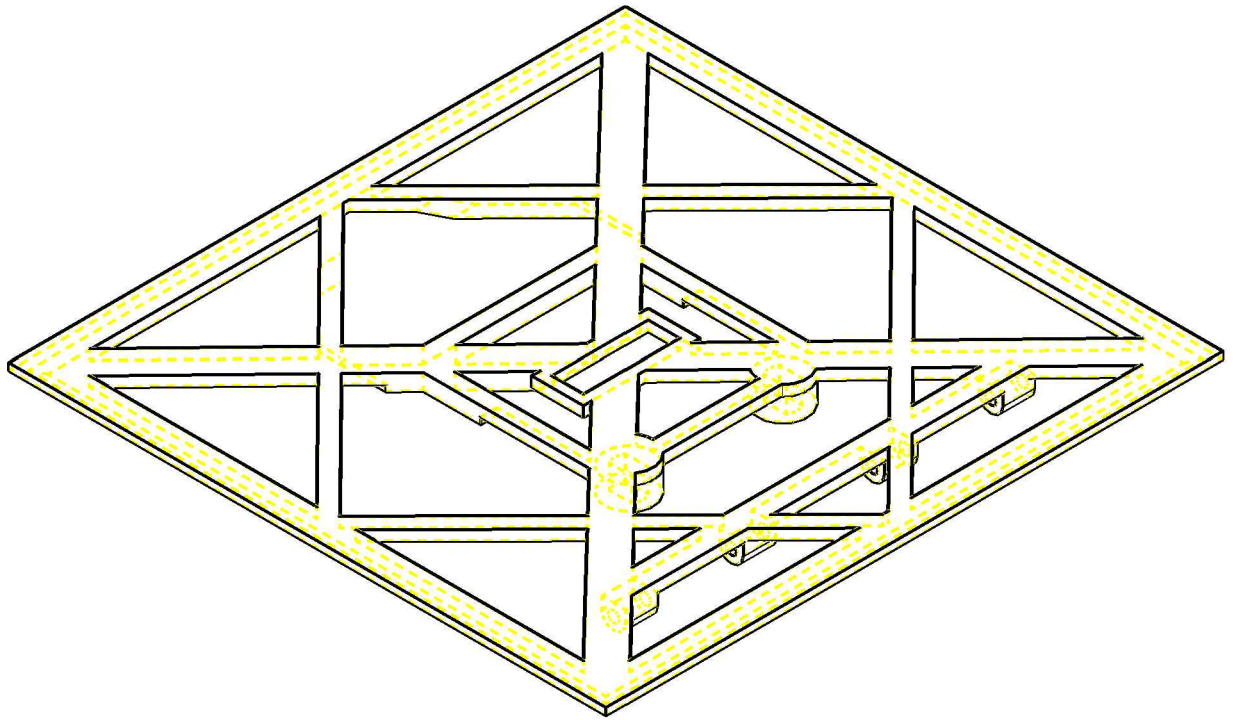


Planta

Alzado

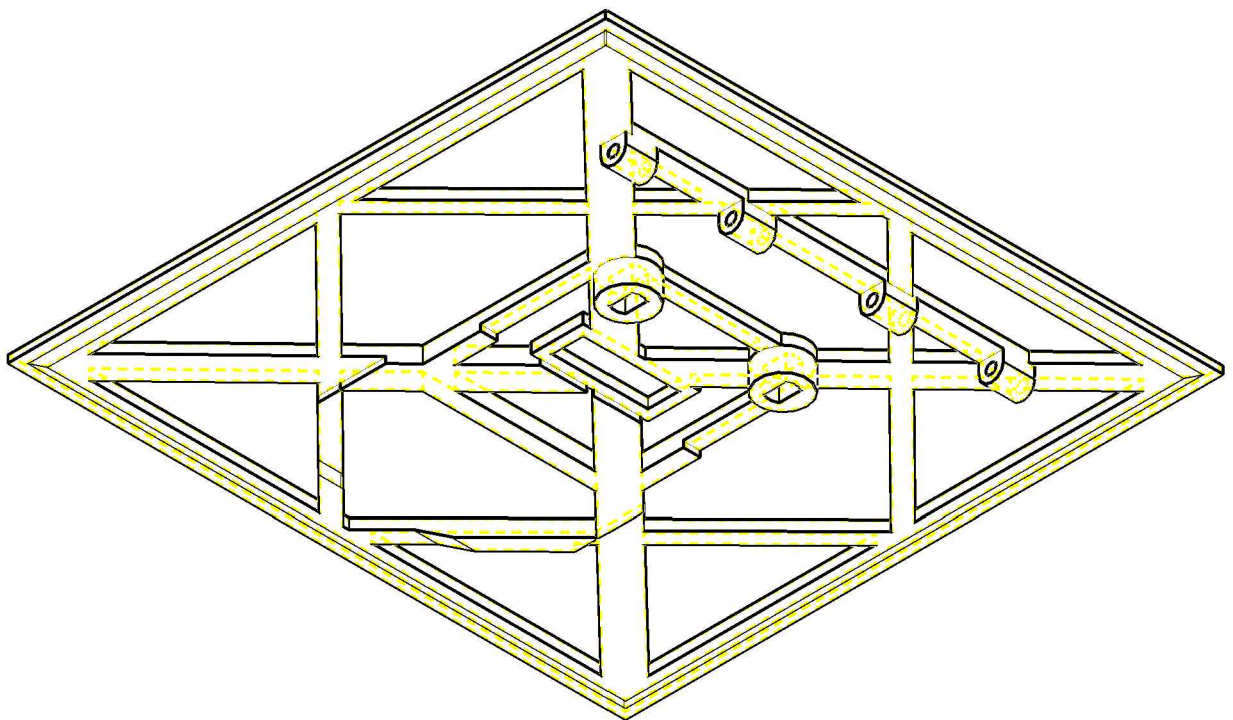


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:2	Pieza de ascensión tapa	PI. PD4

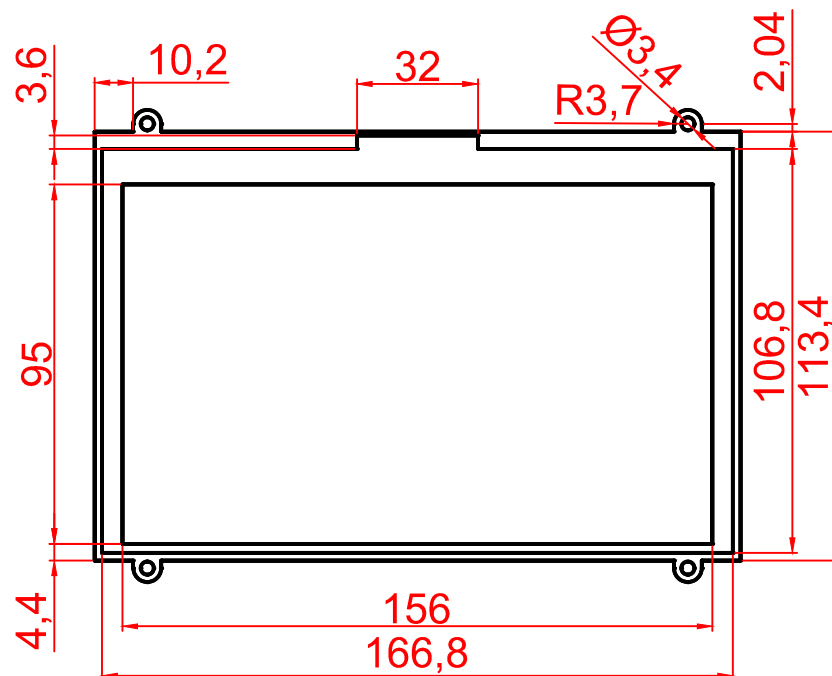


Vista Superior, Suroeste

Vista Inferior, Suroeste

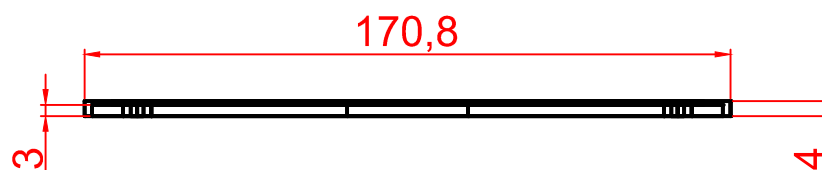


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de ascensión tapa	Pl. PD4 3D

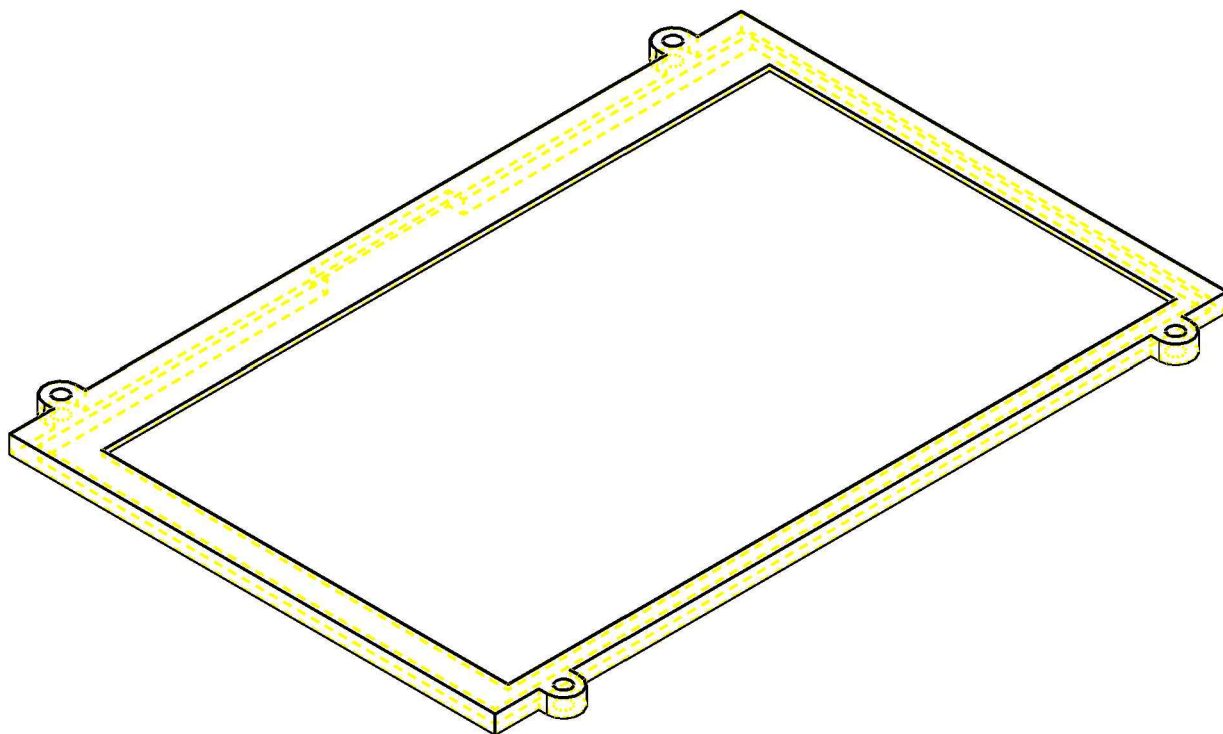


Planta

Alzado

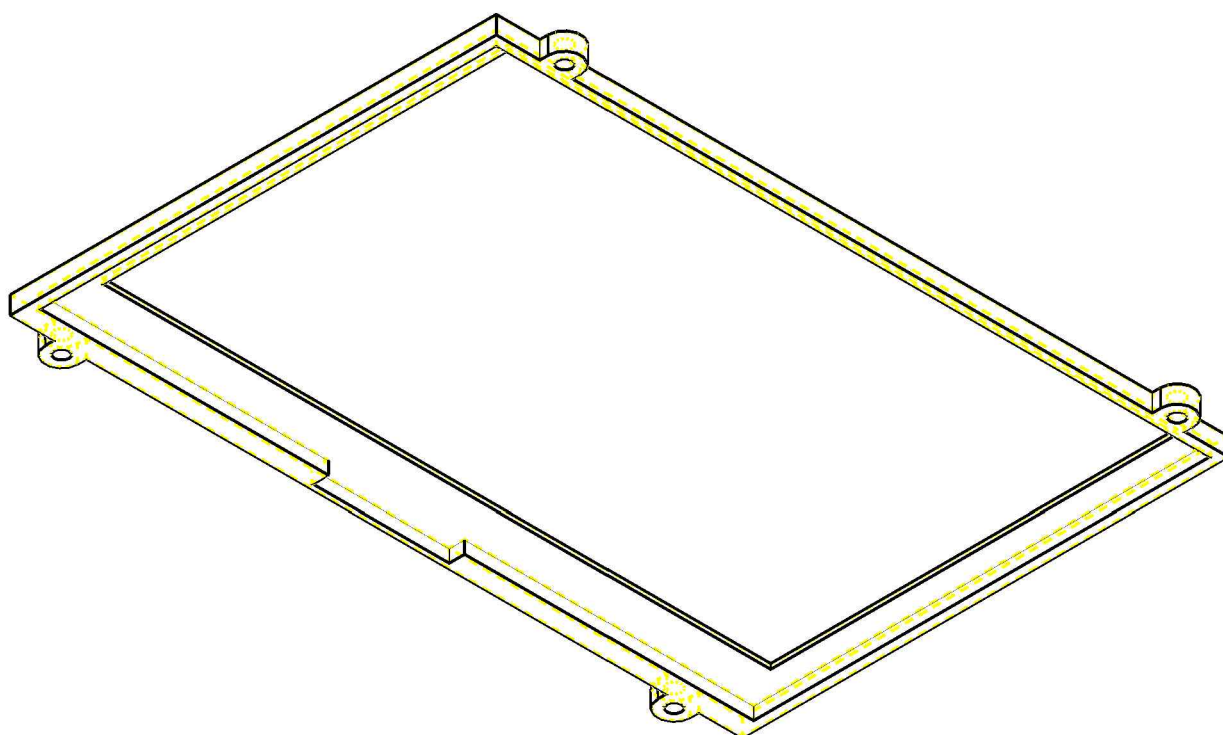


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:2	Pieza de ascensión frontal de pantalla	PI. PD5.1

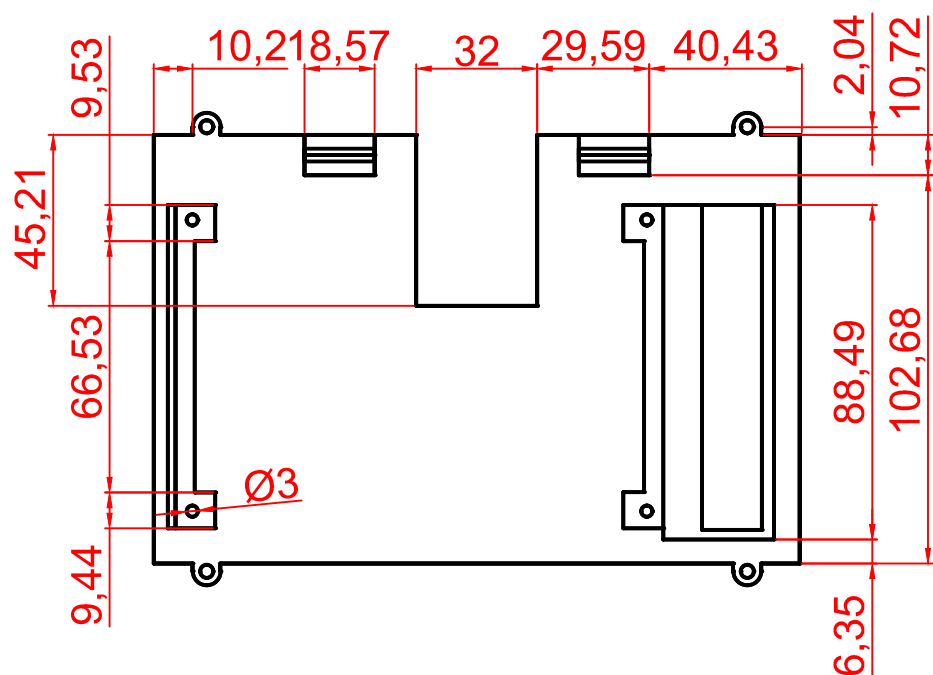


Vista Superior, Suroeste

Vista Inferior, Suroeste

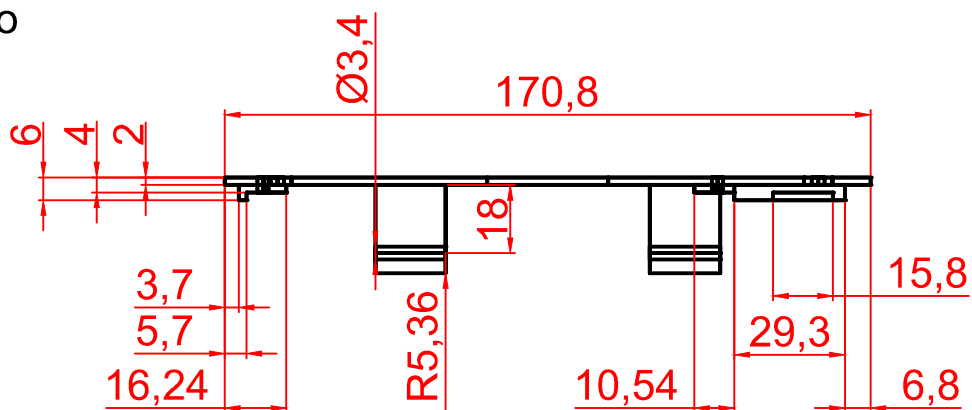


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de ascensión frontal de pantalla	Pl. PD5.1 3D

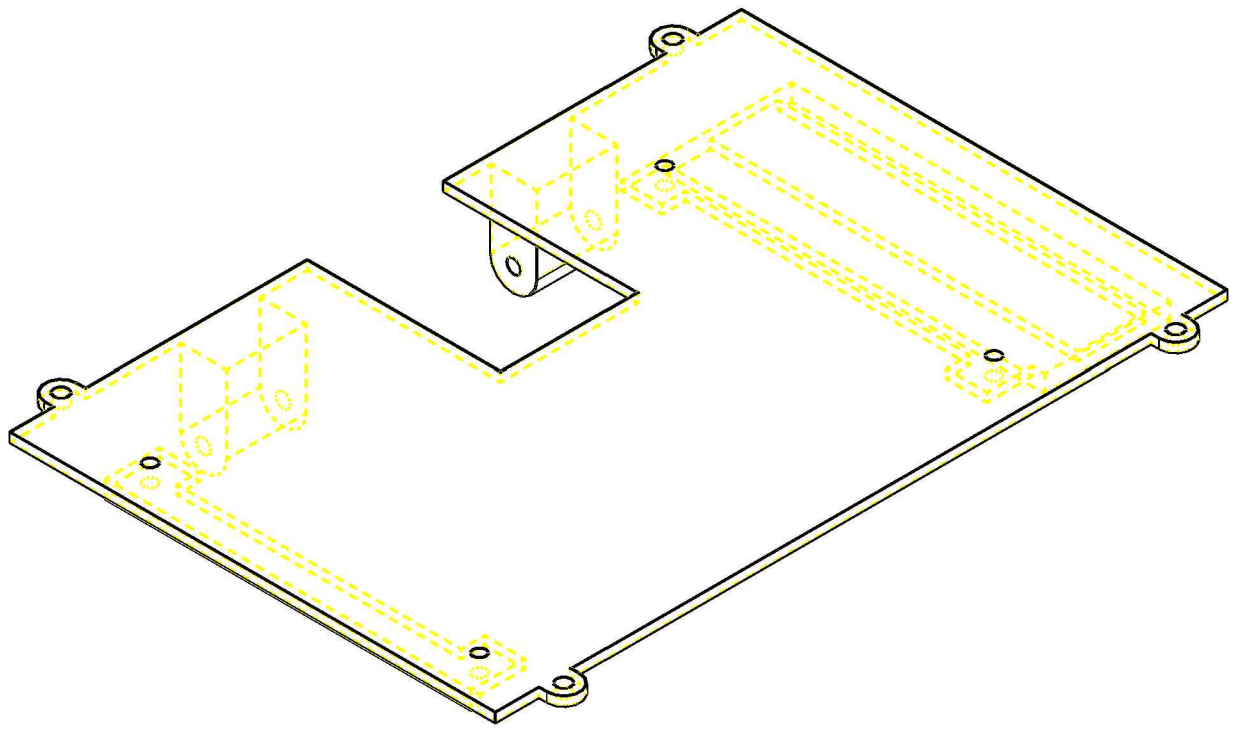


Planta

Alzado

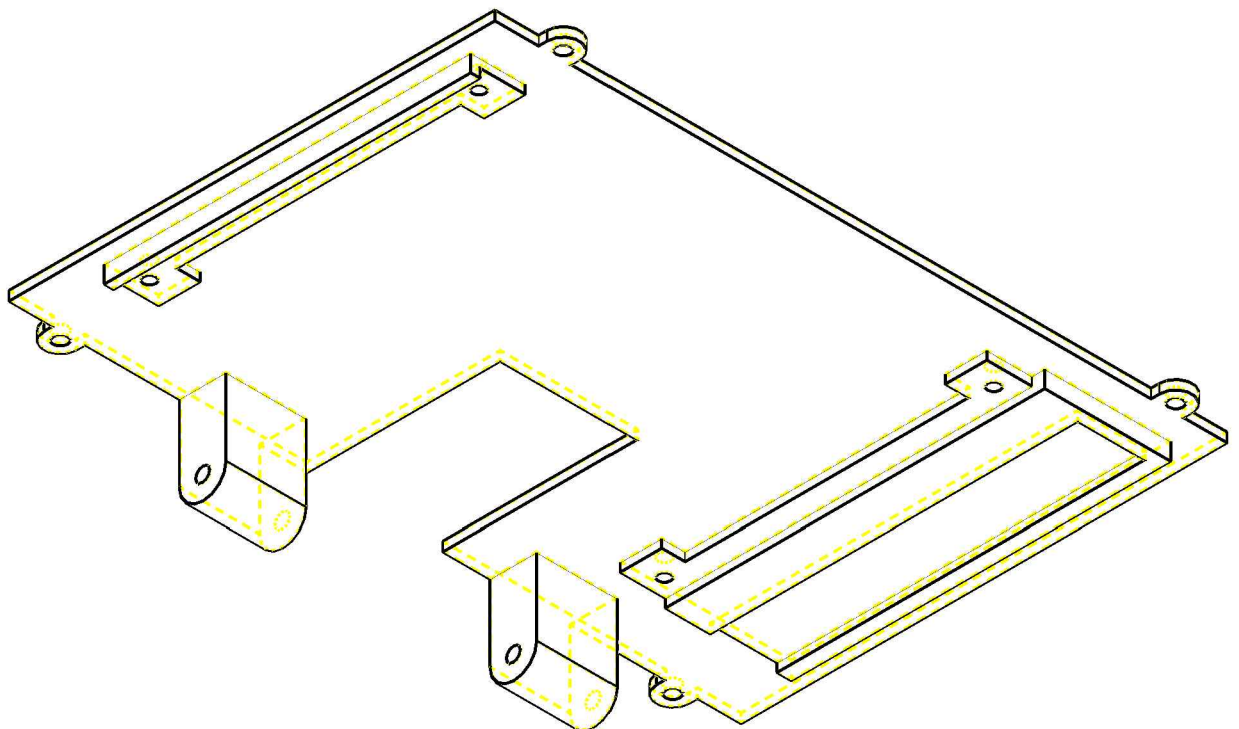


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:2	Pieza de ascensión trasera de pantalla	PI. PD5.2

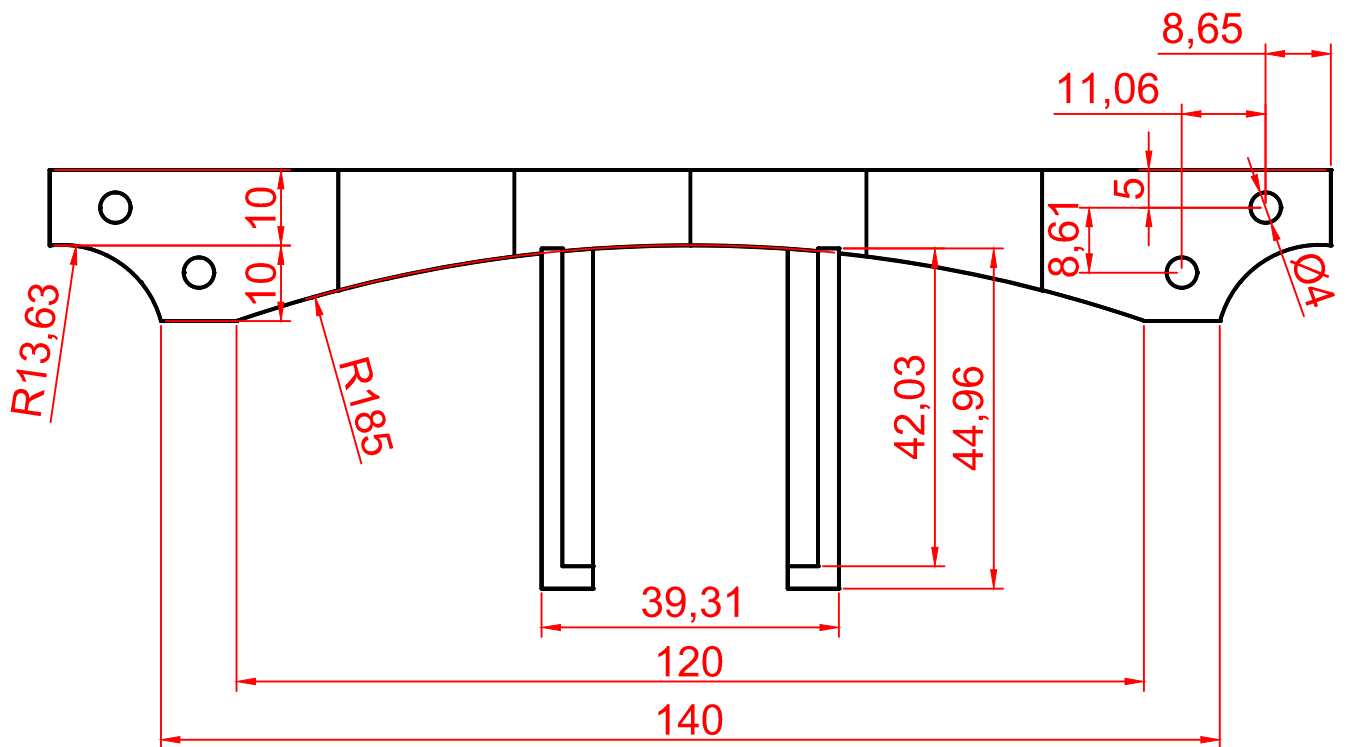


Vista Superior, Suroeste

Vista Inferior, Suroeste

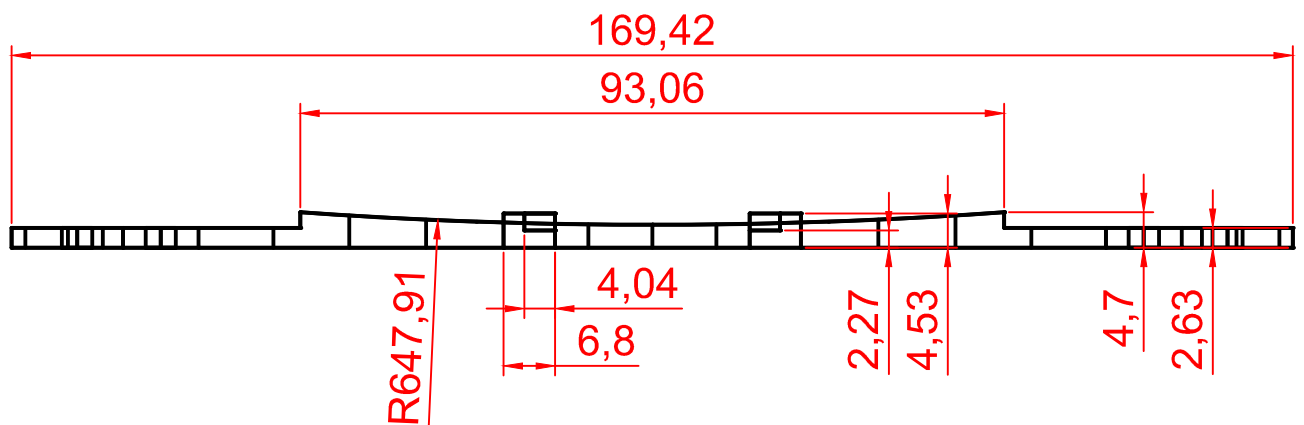


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Pieza de ascensión trasera de pantalla	Pl. PD5.2 3D



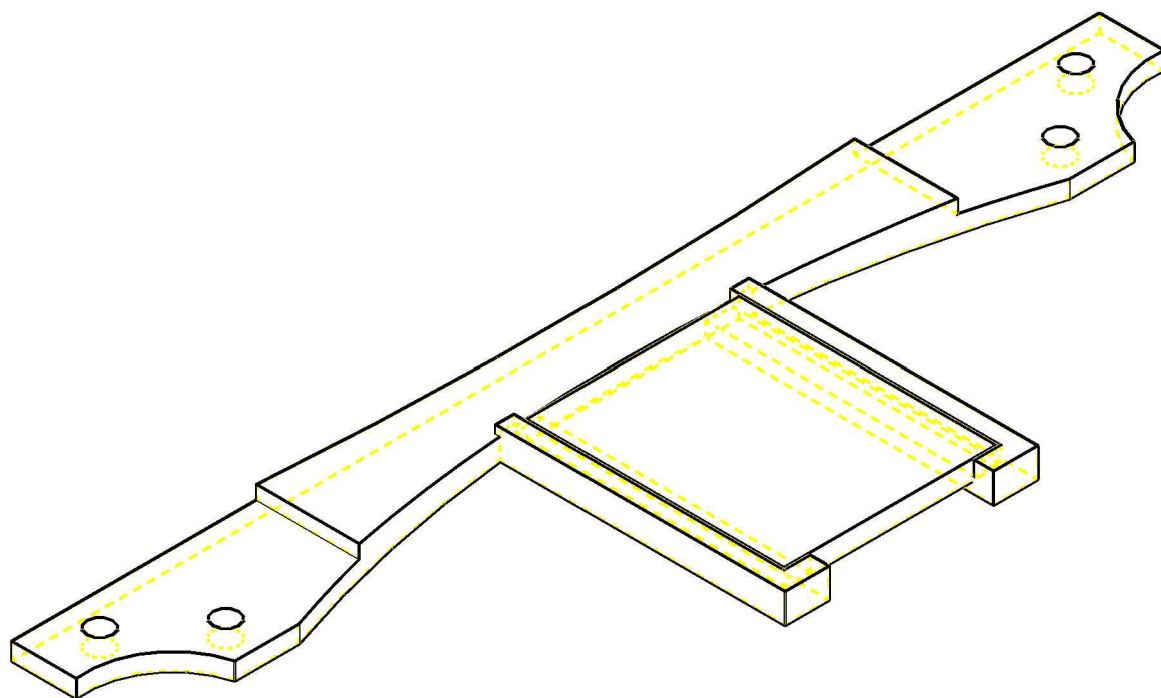
Planta

Alzado



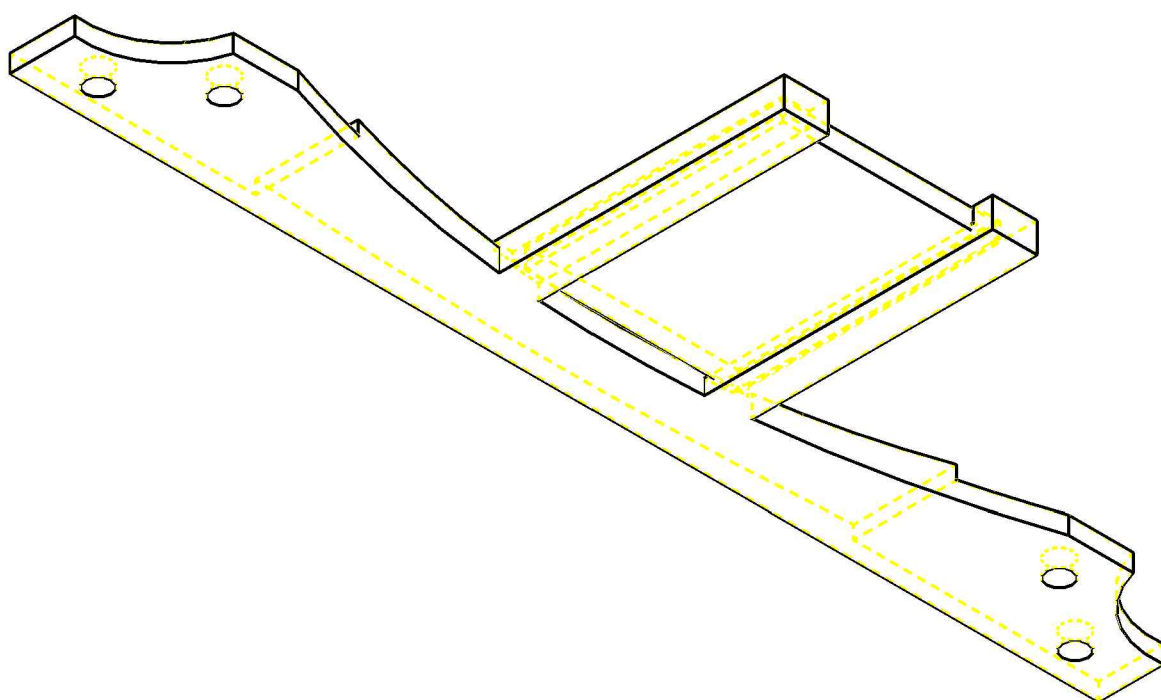
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Aplique base de ventiladores	Pl. PE1.1





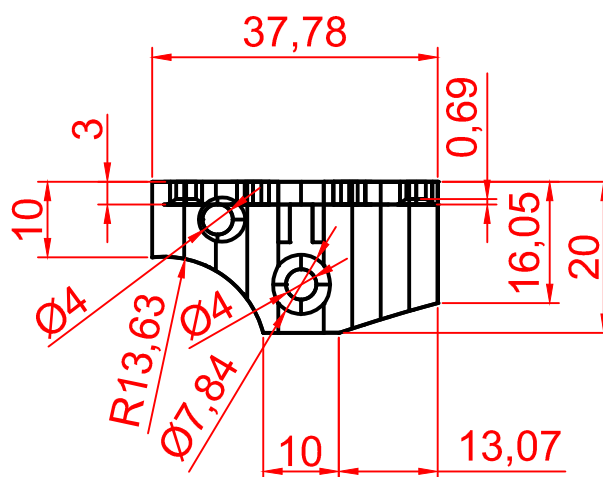
Vista Superior, Suroeste

Vista Inferior, Suroeste



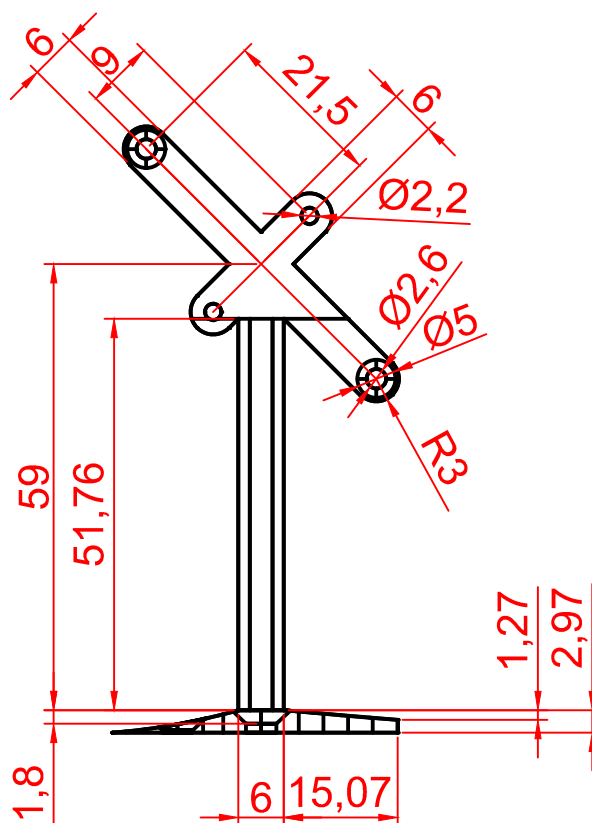
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Aplique base de ventiladores	Pl. PE1.1 3D



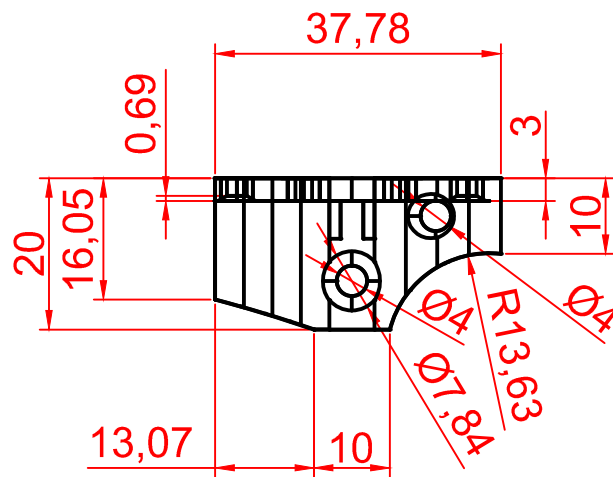


Planta

Alzado

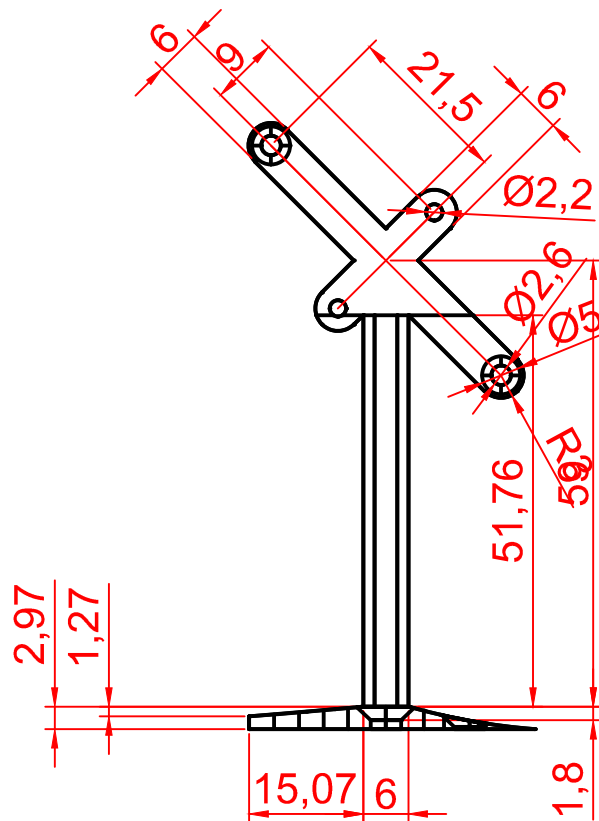


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Aplicación soporte izquierdo de ventilador	PI. PE1.2

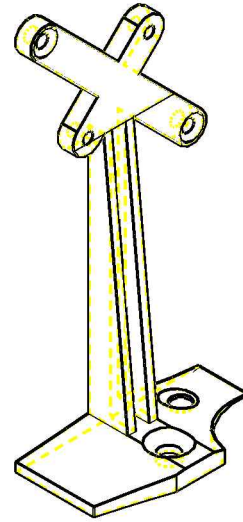
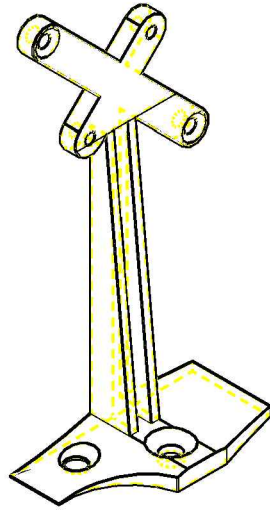


Planta

Alzado

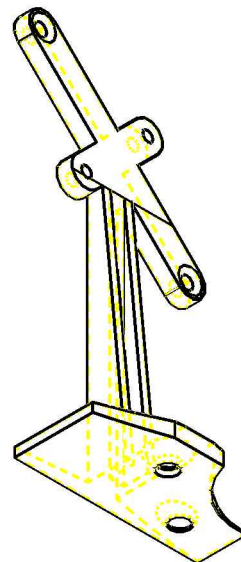
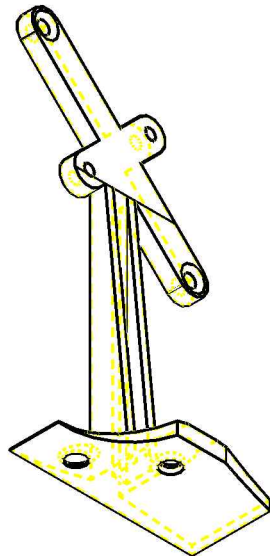


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Aplique soporte derecho de ventilador	PI. PE1.3

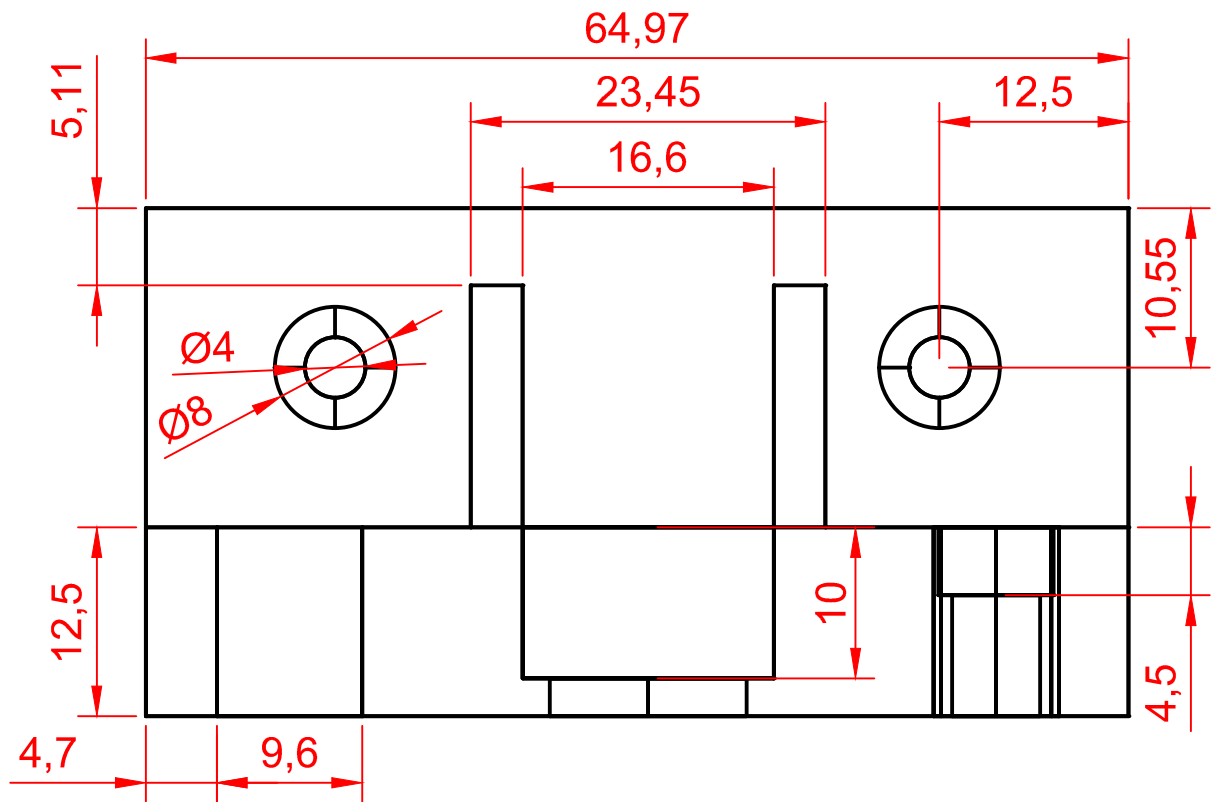


Vista Superior, Suroeste

Vista Inferior, Suroeste

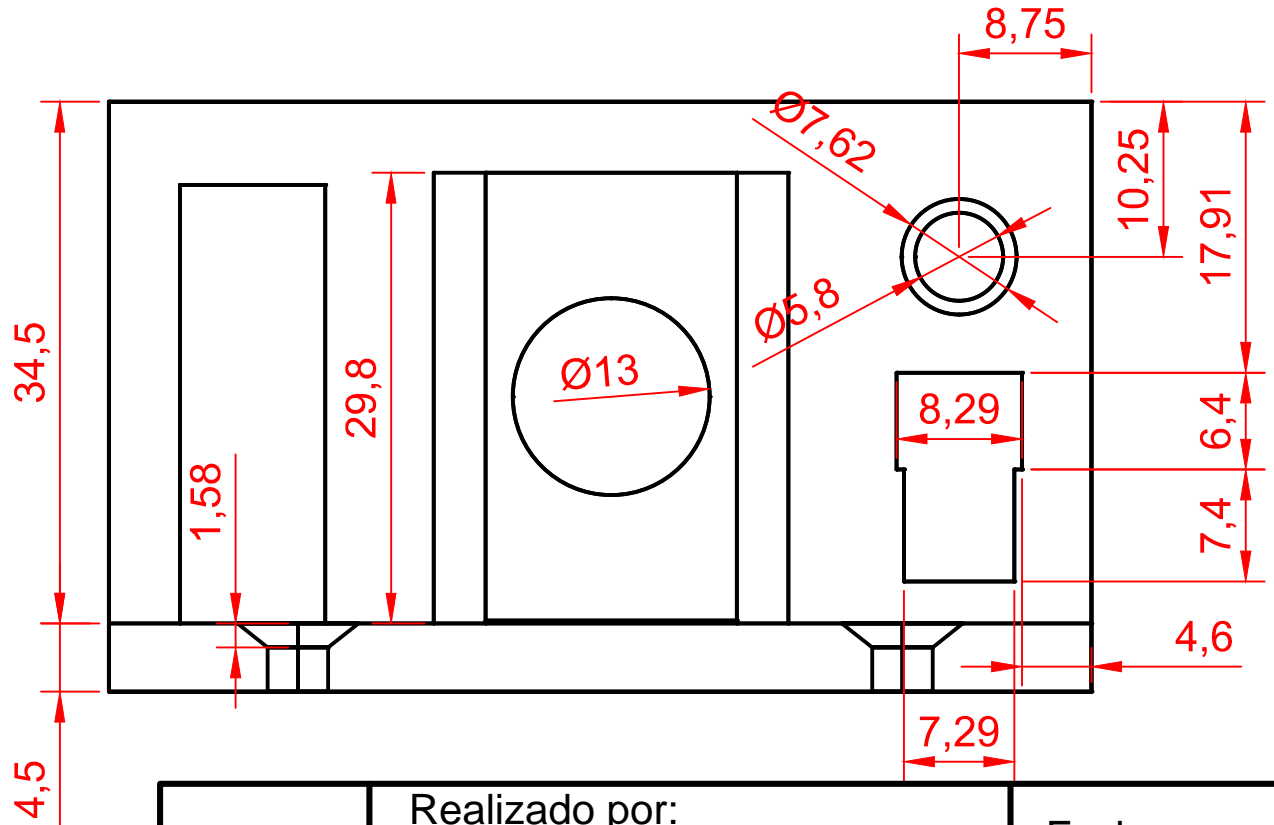


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Aplique soportes izquierdo y derecho de ventiladores	Pl. PE1.4 3D

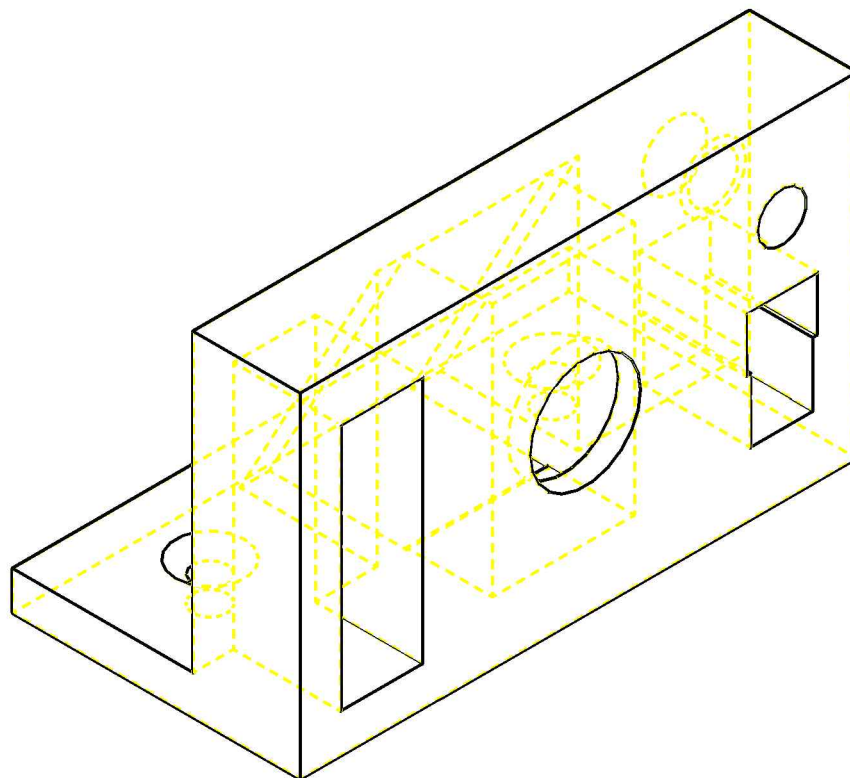


Planta

Alzado

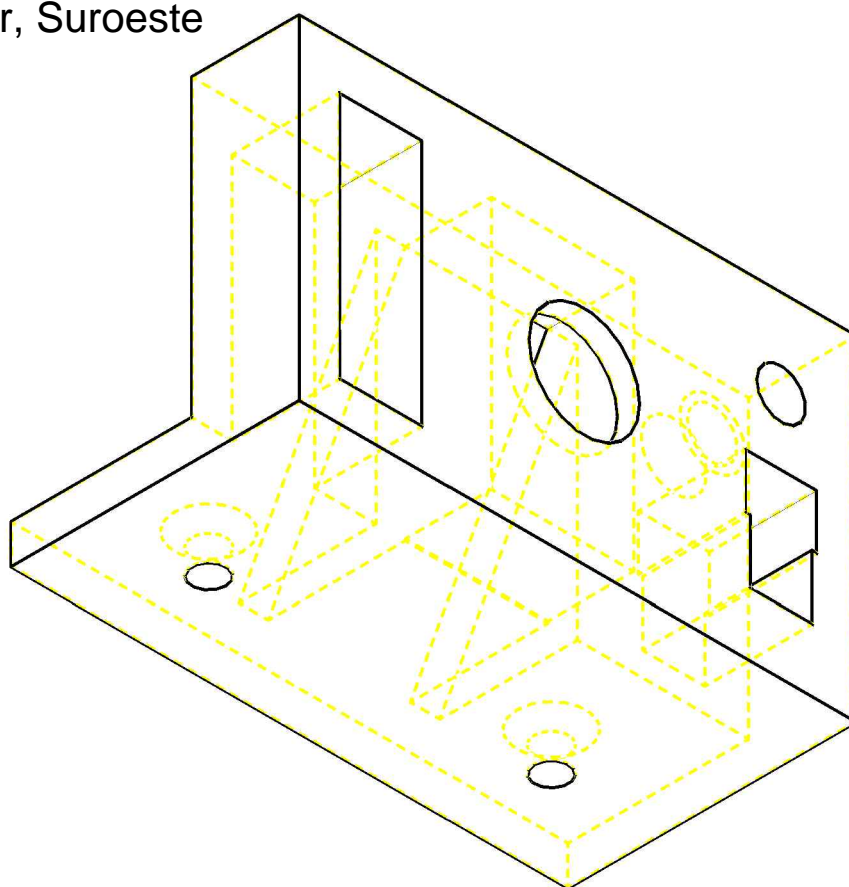


Realizado por:	Antonio Molina Cortés Salvador Moreno Gómez	
Fecha:	11/05/2016	
Escala:	2:1	
	Aplique base interruptor	
	PI. PE2	

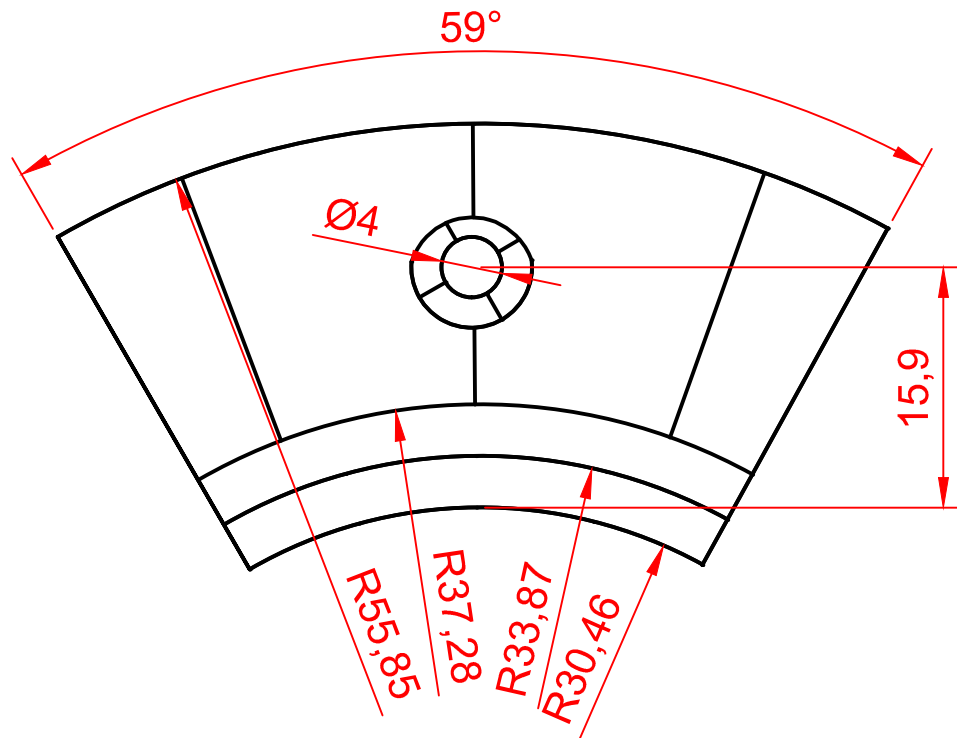


Vista Superior, Suroeste

Vista Inferior, Suroeste

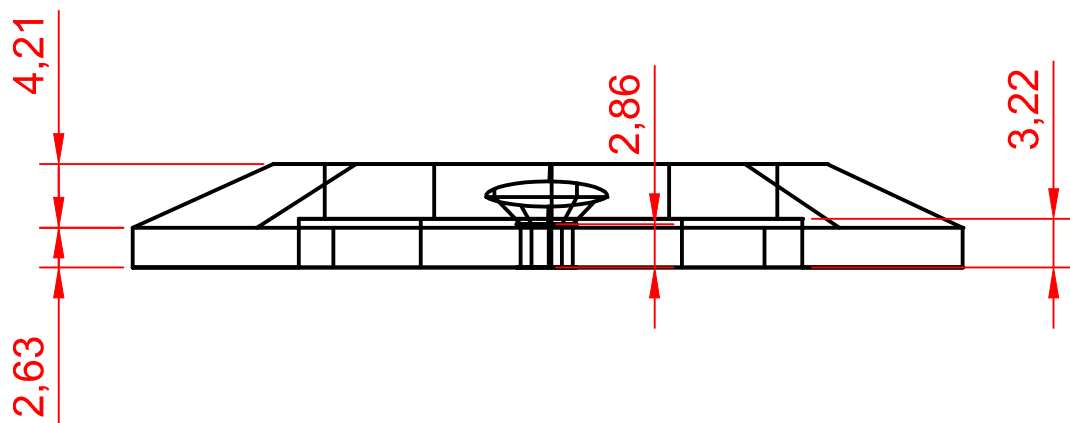


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Aplique base interruptor	Pl. PE2 3D

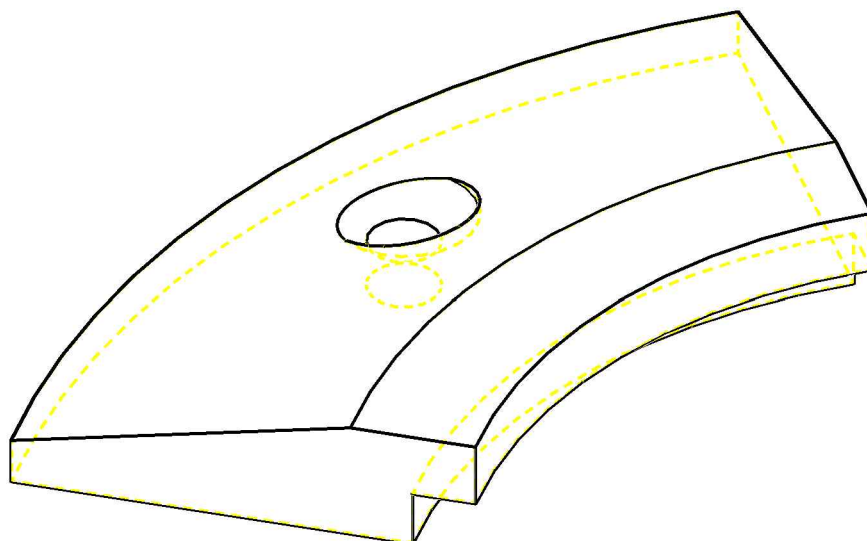


Planta

Alzado

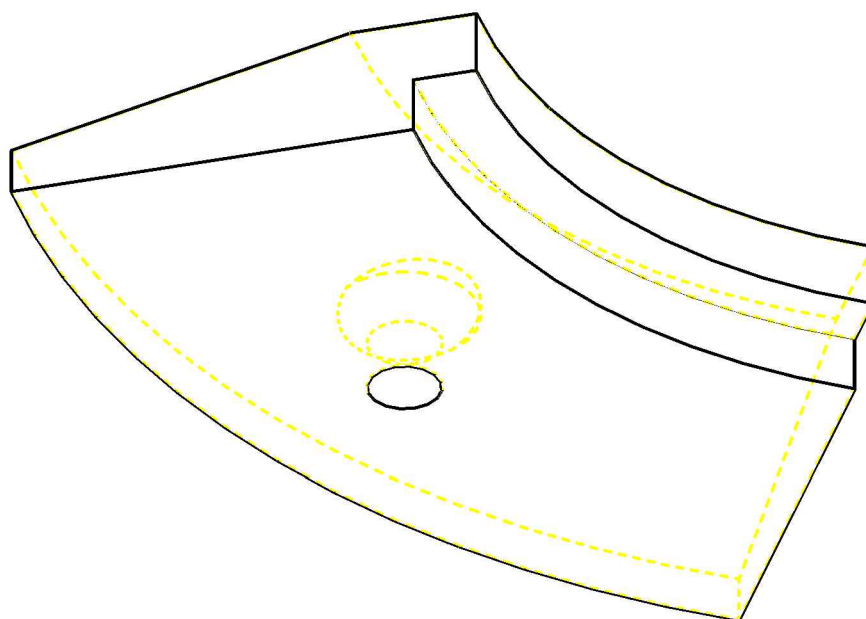


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
2:1	Aplique soporte de altavoces	Pl. PE3

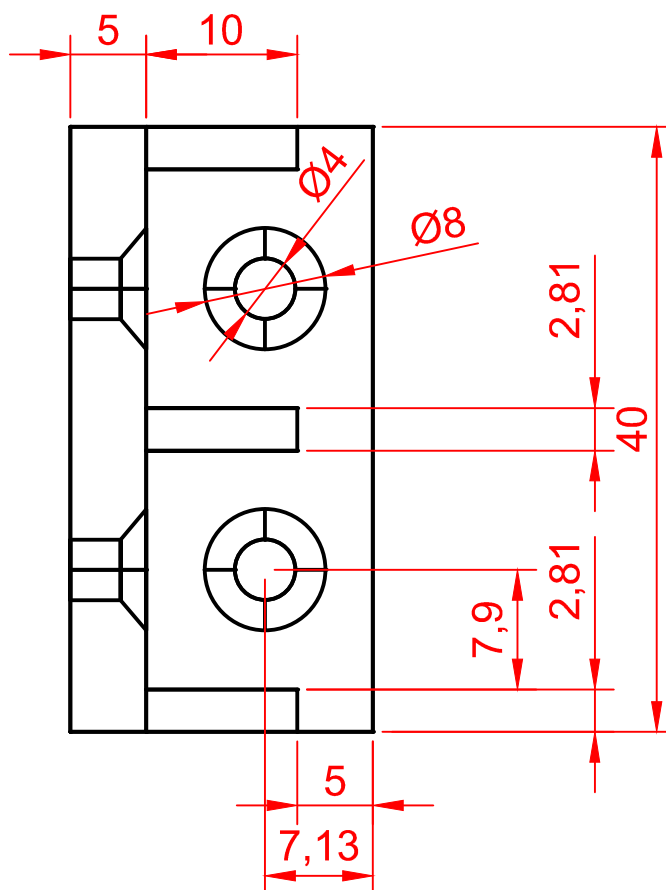


Vista Superior, Suroeste

Vista Inferior, Suroeste

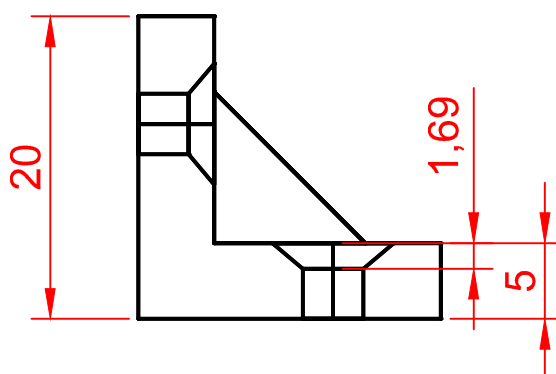


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Aplique soporte de altavoces	Pl. PE3 3D



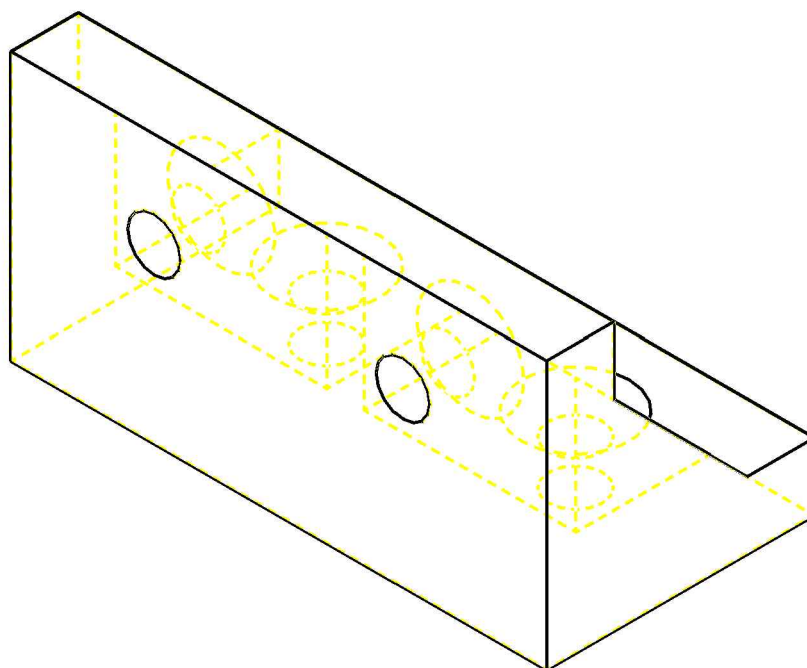
Planta

Alzado



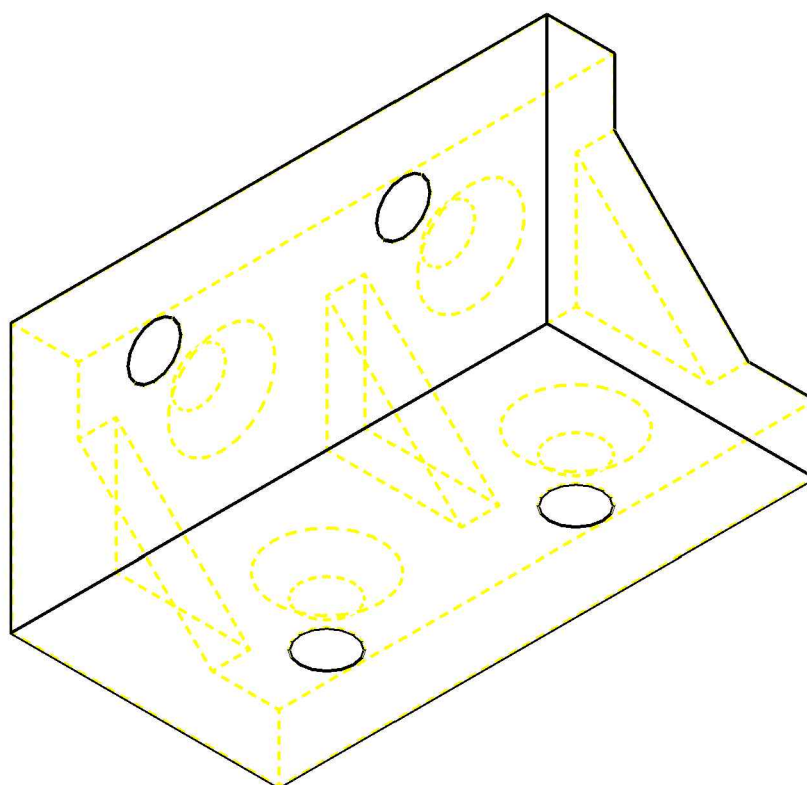
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
2:1	Aplique escuadras de batería	Pl. PE4



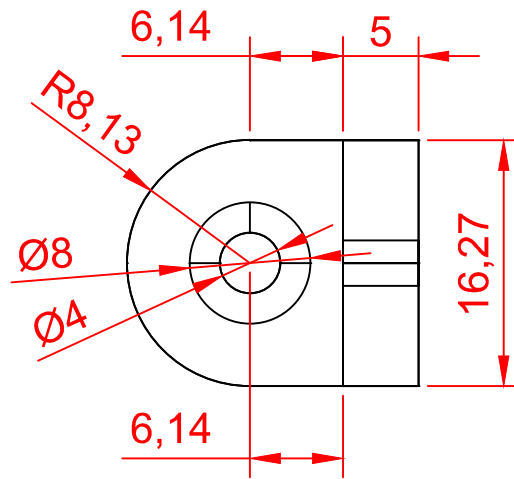


Vista Superior, Suroeste

Vista Inferior, Suroeste

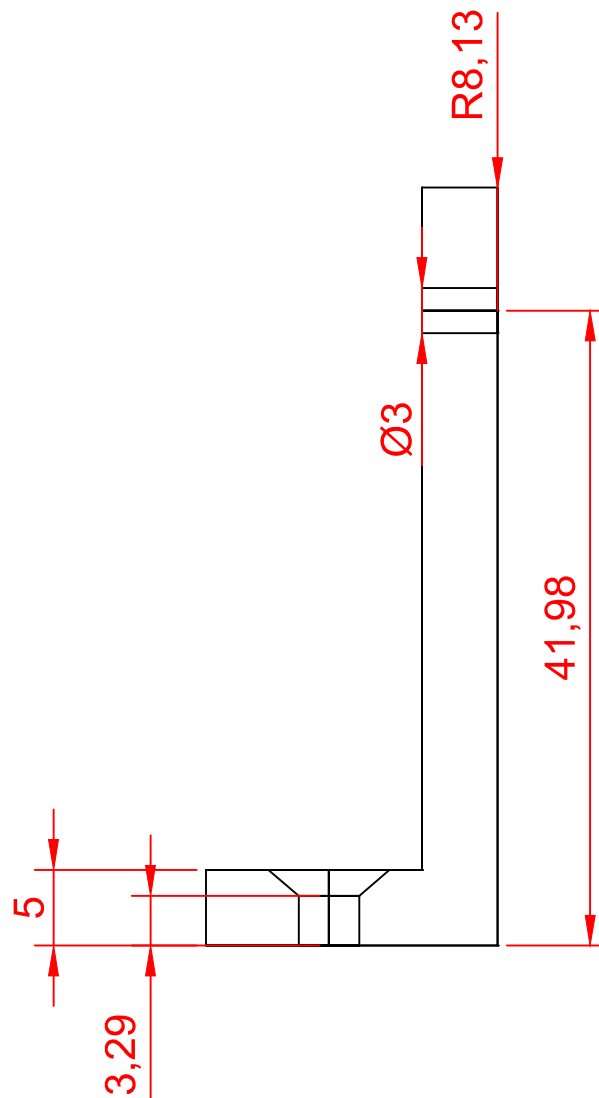


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Aplique escuadras de batería	Pl. PE4 3D

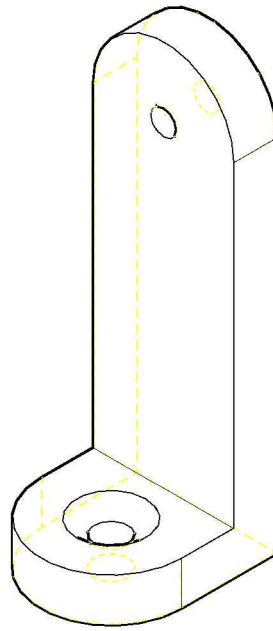


Planta

Alzado

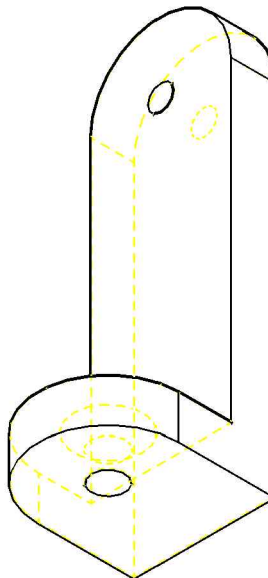


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
2:1	Aplique base de micrófono	PI. PE5.1

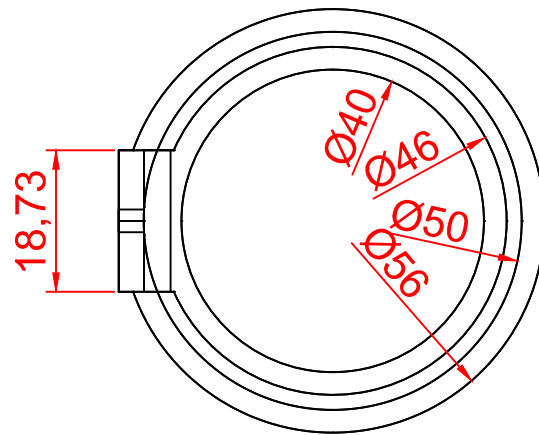


Vista Superior, Suroeste

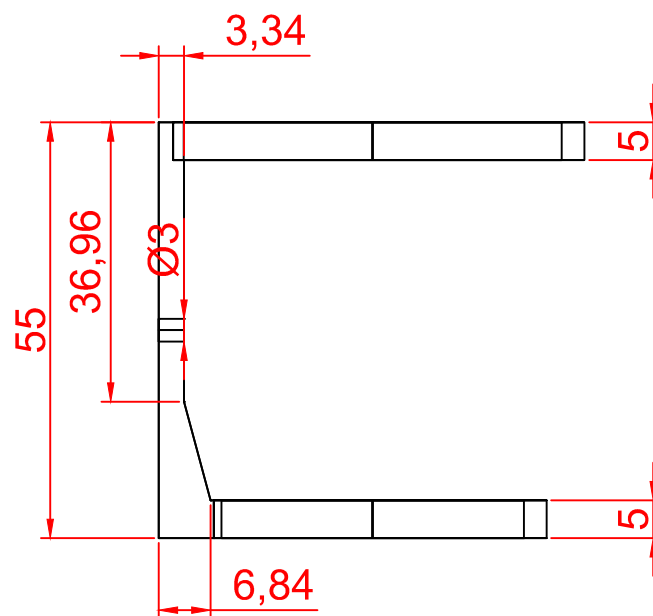
Vista Inferior, Suroeste



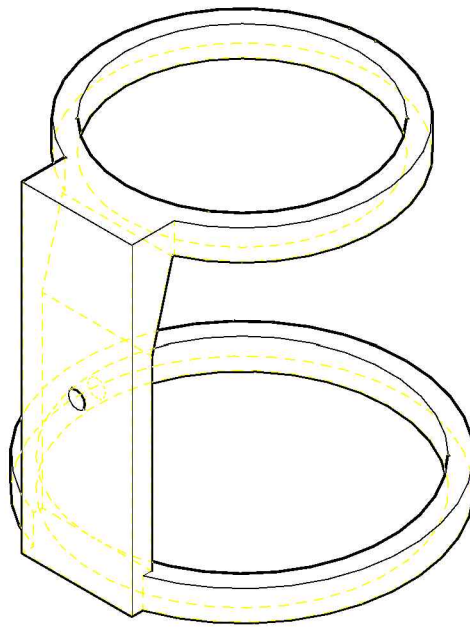
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Aplique base de micrófono	Pl. PE5.1 3D



Planta  
Alzado

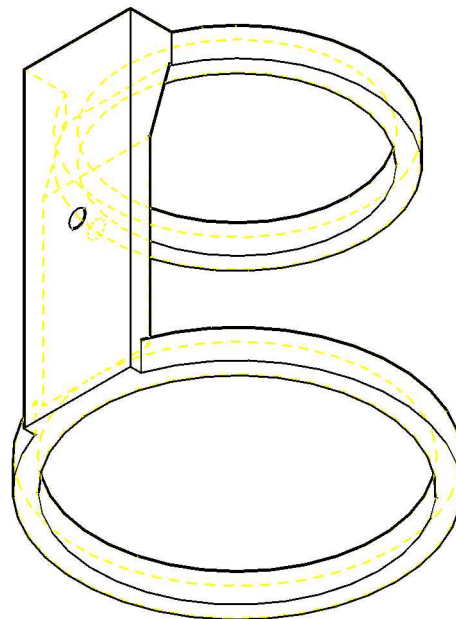


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Aplique pinza de micrófono	Pl. PE5.2

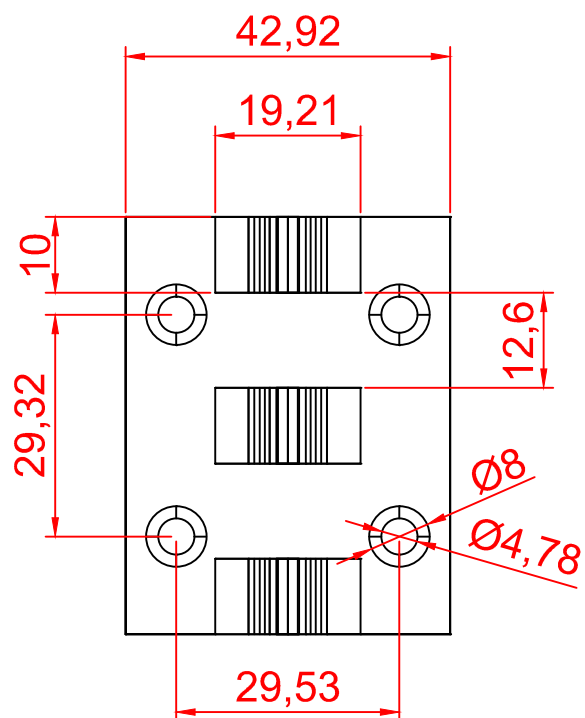


Vista Superior, Suroeste

Vista Inferior, Suroeste

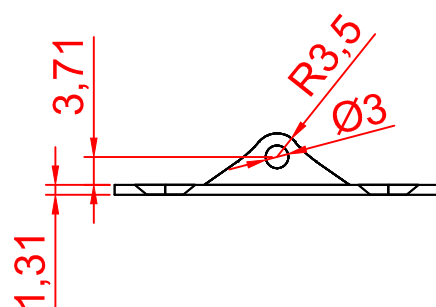


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Aplique pinza de micrófono	Pl. PE5.2 3D

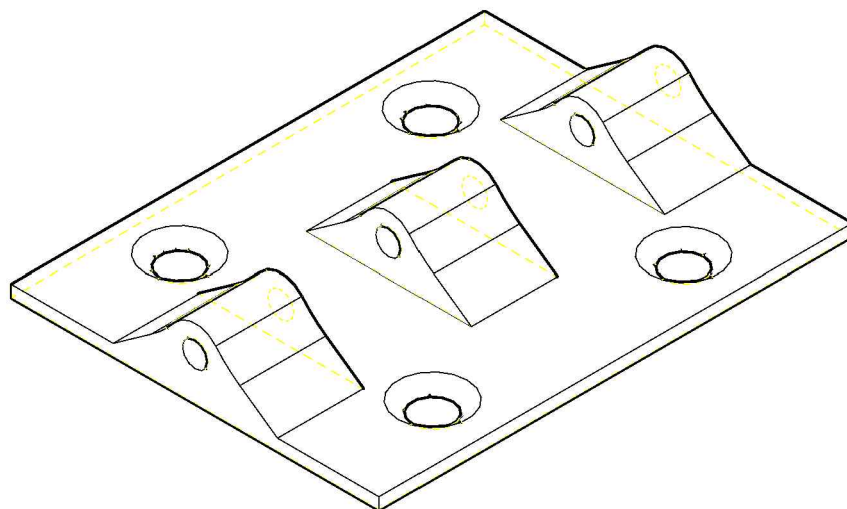


Planta

Alzado

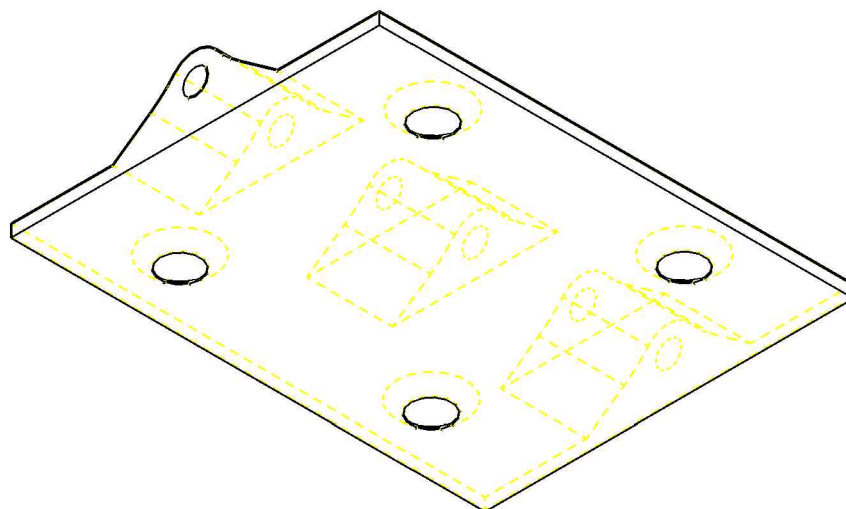


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Aplique base de cámara inferior	PI. PE6.1

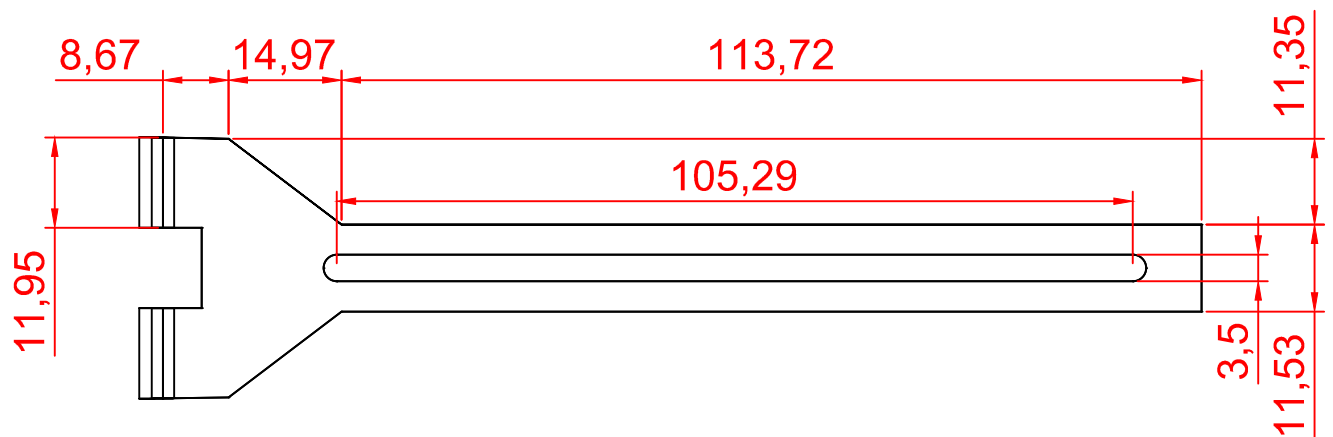


Vista Superior, Suroeste

Vista Inferior, Suroeste

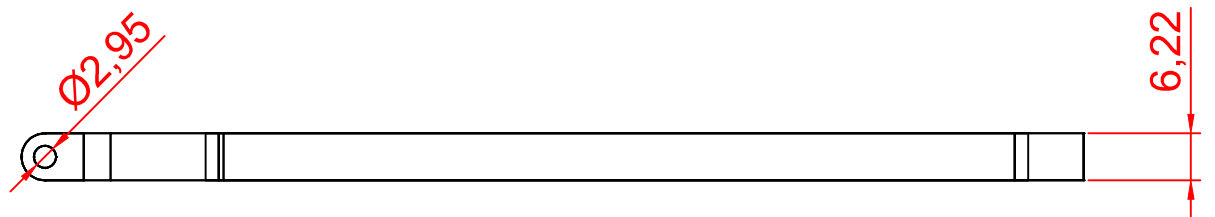


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Aplique base de cámara inferior	Pl. PE6.1 3D



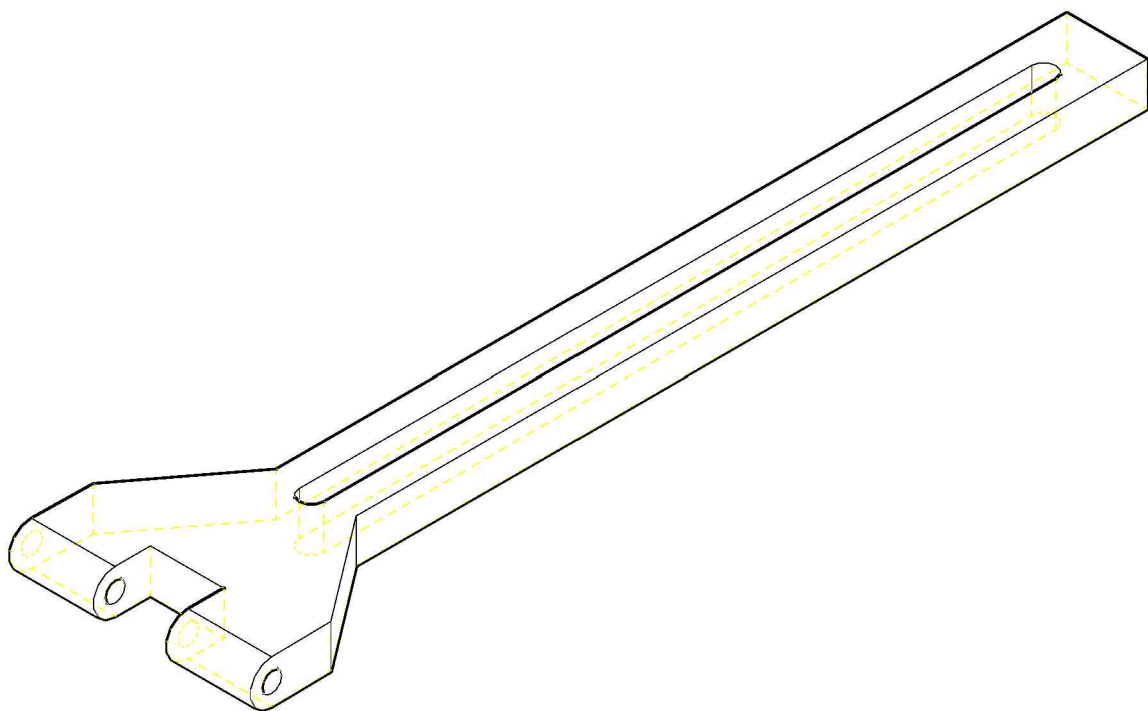
Planta

Alzado



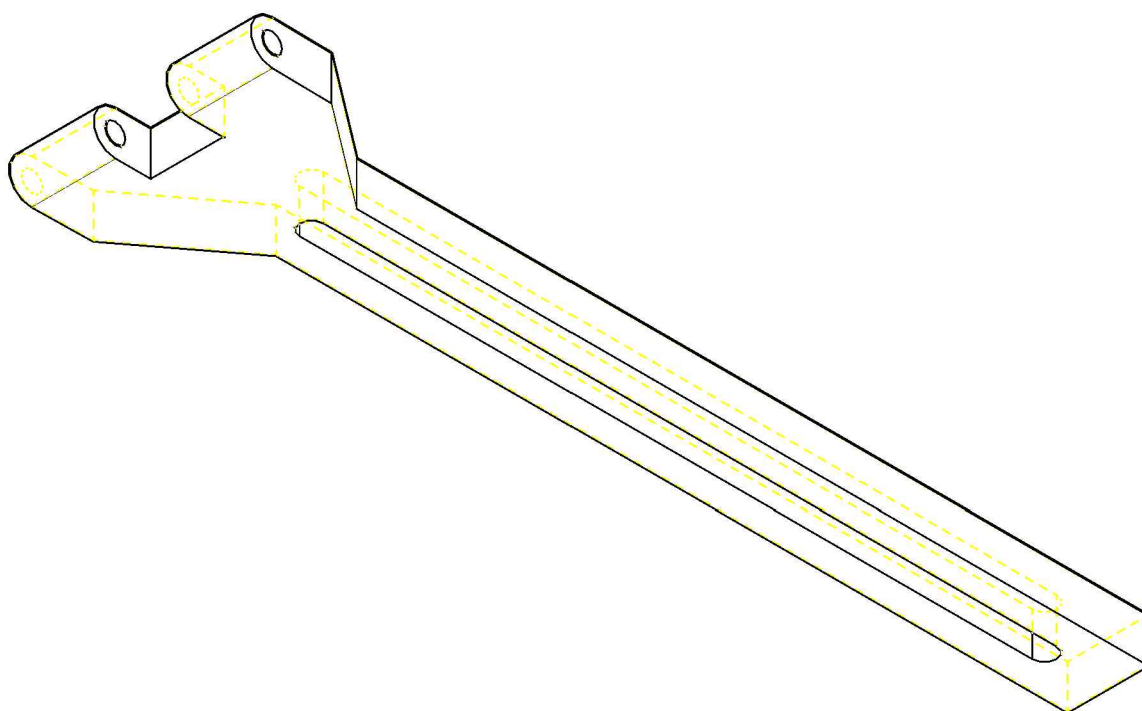
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Aplique rail de cámara inferior	PI. PE6.2



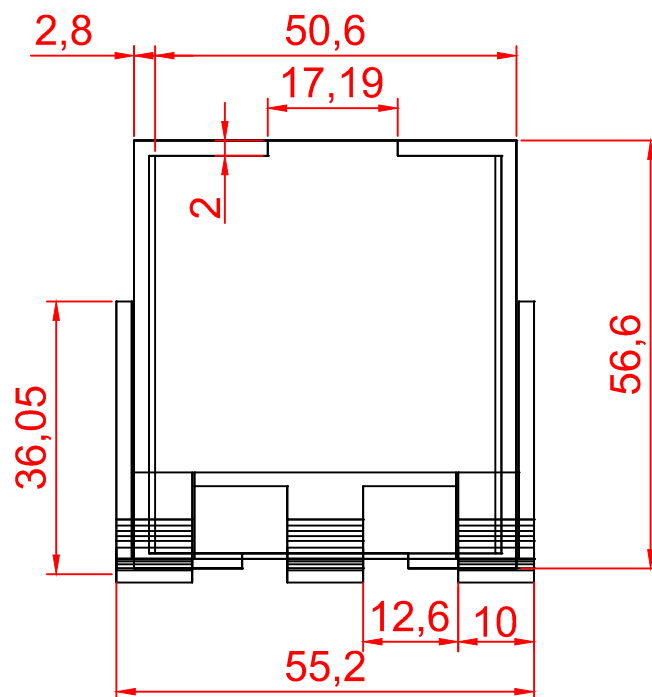


Vista Superior, Suroeste

Vista Inferior, Suroeste

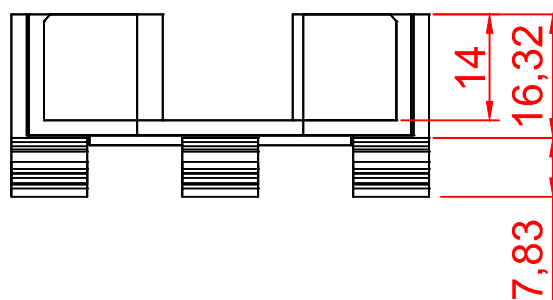


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Aplique rail de cámara inferior	Pl. PE6.2 3D

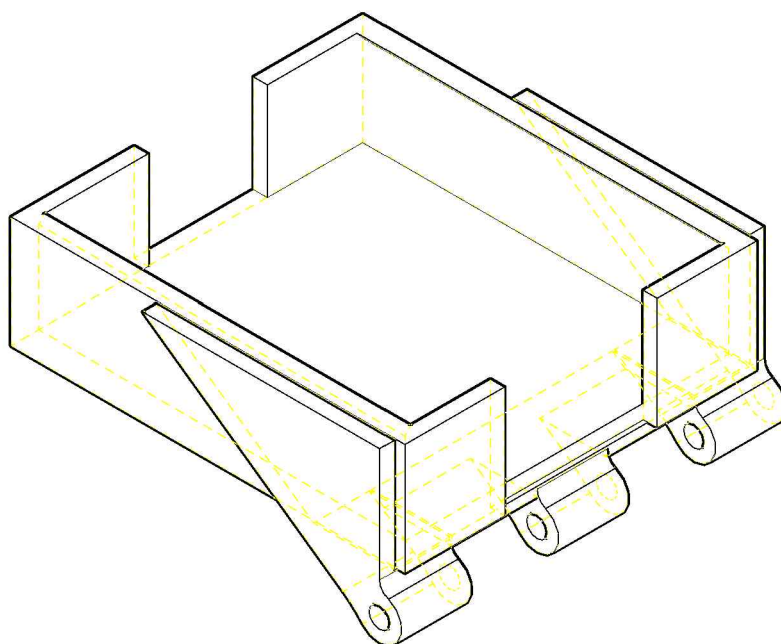


Planta

Alzado

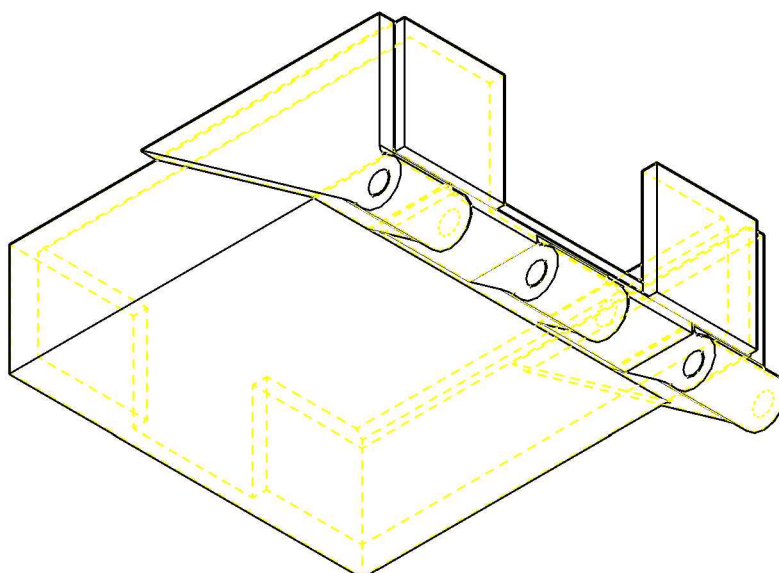


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Aplique carcasa 1 de cámara inferior	PI. PE6.3

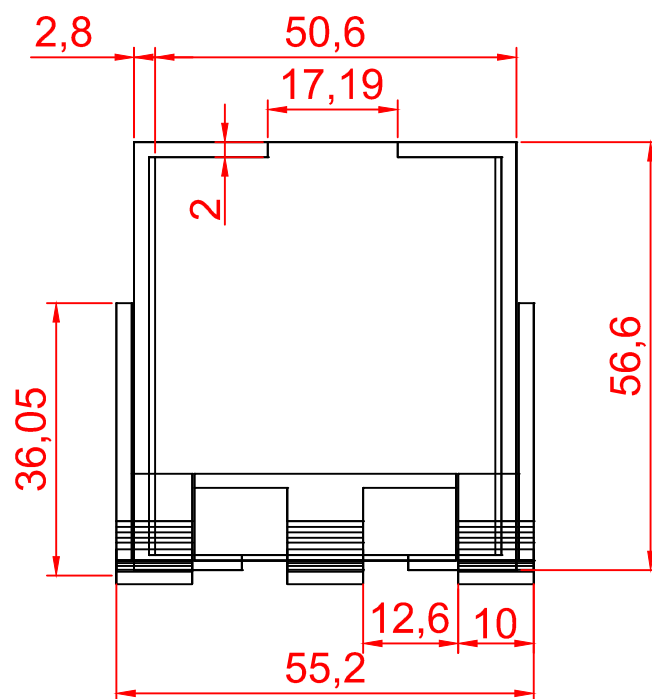


Vista Superior, Suroeste

Vista Inferior, Suroeste

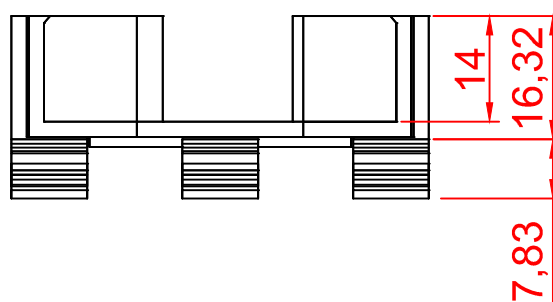


Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Aplique carcasa 1 de cámara inferior	Pl. PE6.3 3D

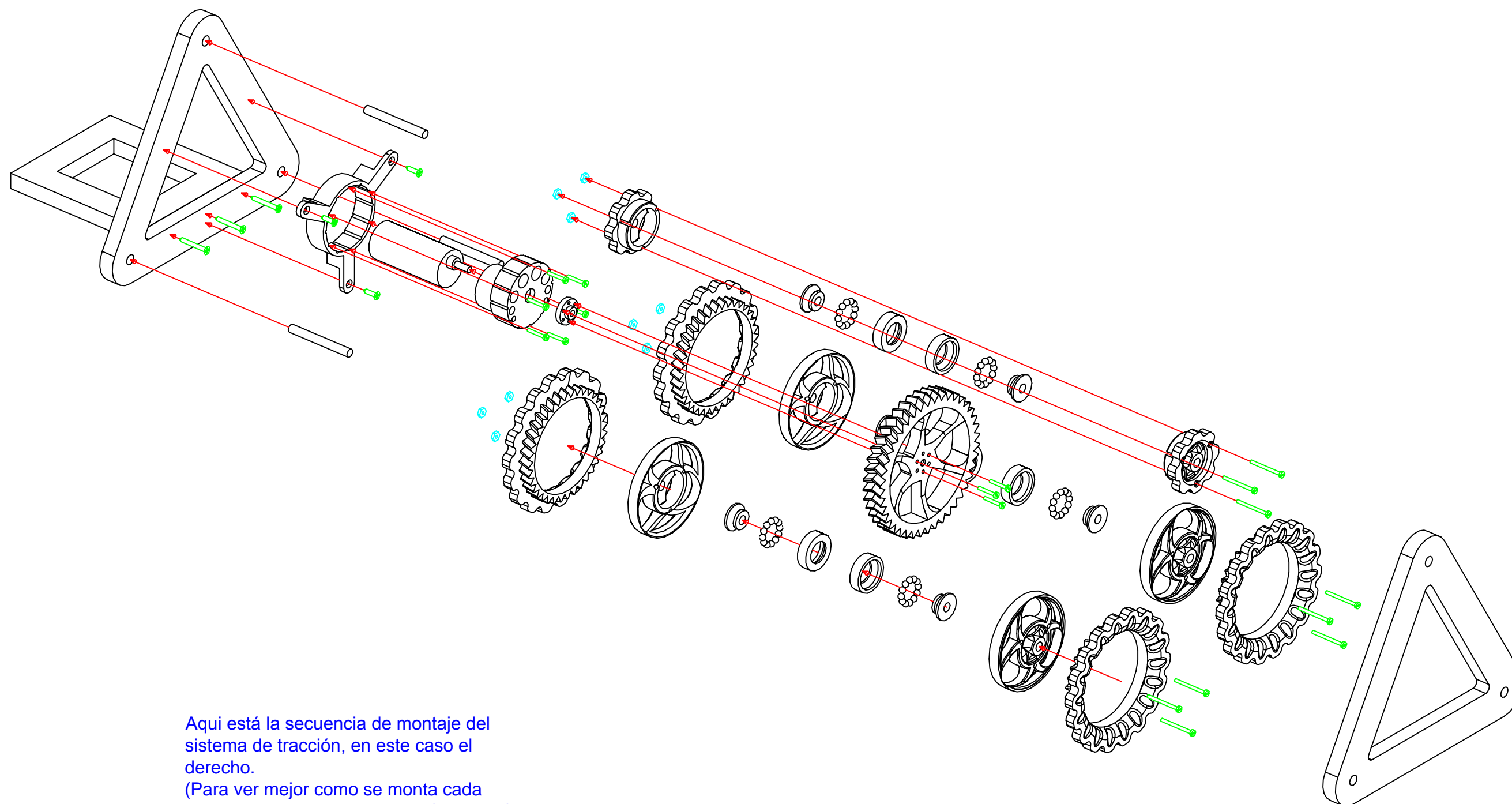


Planta

Alzado



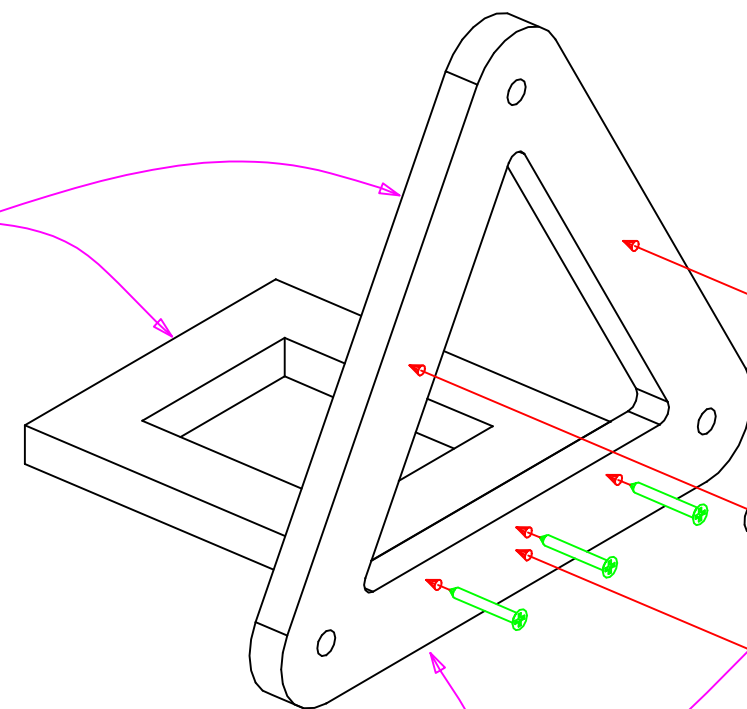
Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
1:1	Aplique carcasa 2 de cámara inferior	Pl. PE6.4



Aquí está la secuencia de montaje del sistema de tracción, en este caso el derecho.  
(Para ver mejor como se monta cada conjunto ver secuencias de la SB2 a la SB6)

Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Secuencia montaje de tracción	Pl. SB1

Paso 1:  
Fijamos la una de las PA7  
(Pieza de chasis marco de  
suspensión) a una de las PA8  
(Pieza de chasis triángulo de  
tracción) con tornillos de  
madera de 4x30mm y  
adhesivo de montaje



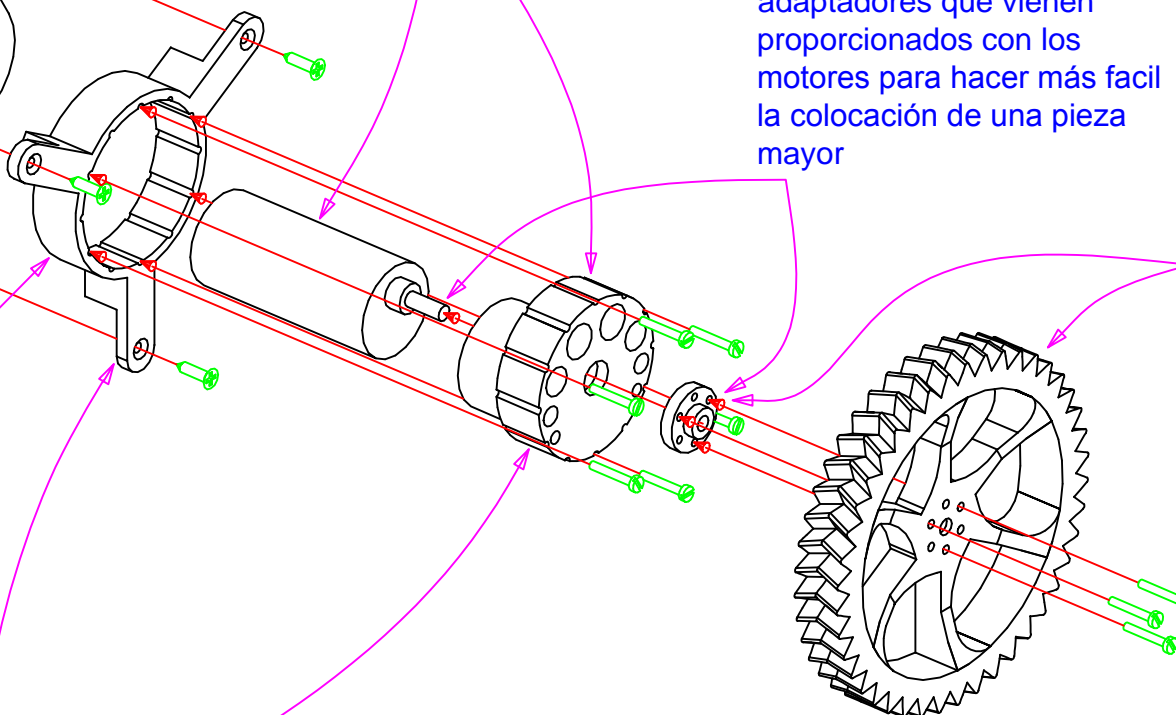
Paso 2:  
Colocamos una de las PB1.1  
(Pieza de tracción sujeción de  
motor parte 1) en la PA8  
(Pieza de chasis triángulo de  
tracción) con tornillos de  
madera de 4x20mm

Paso 5:  
Introducimos la PB1.2 (Pieza  
de tracción sujeción de motor  
parte 2) en la PB1.1 (Pieza de  
tracción sujeción de motor  
parte 1), la giramos a nuestro  
gusto y la fijamos con tornillos  
métricos de 3x25mm

Paso 3:  
Fijamos uno de los motores de  
tracción a una de las  
PB1.2(Pieza de tracción  
sujeción de motor parte 2) con  
adhesivo de montaje o  
pegamento de cianocrilato

Paso 4:  
Colocamos uno de los  
adaptadores que vienen  
proporcionados con los  
motores para hacer más facil  
la colocación de una pieza  
mayor

Paso 6:  
Fijamos una de las PB4: la  
PB4.1 (Pieza de tracción  
engranaje central izquierdo) si  
estamos montando el juego  
izquierdo o la PB4.2 (Pieza de  
tracción engranaje central  
derecho) si es el derecho. Esta  
la fijamos al adaptador del  
motor con tornillos metricos de  
3x20mm



Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Secuencia montaje de sugeción de motor y engranaje	Pl. SB2

Paso 1:  
Se pone el conjunto de medio engranaje satélite en cada agujero del triángulo, si es el juego de tracción izquierdo pondremos engranajes satélites levógiros, si fuese el juego derecho pondríamos dextrógiros.  
(Para ver mejor como se monta el conjunto de medio engranaje ver secuencia SB4)

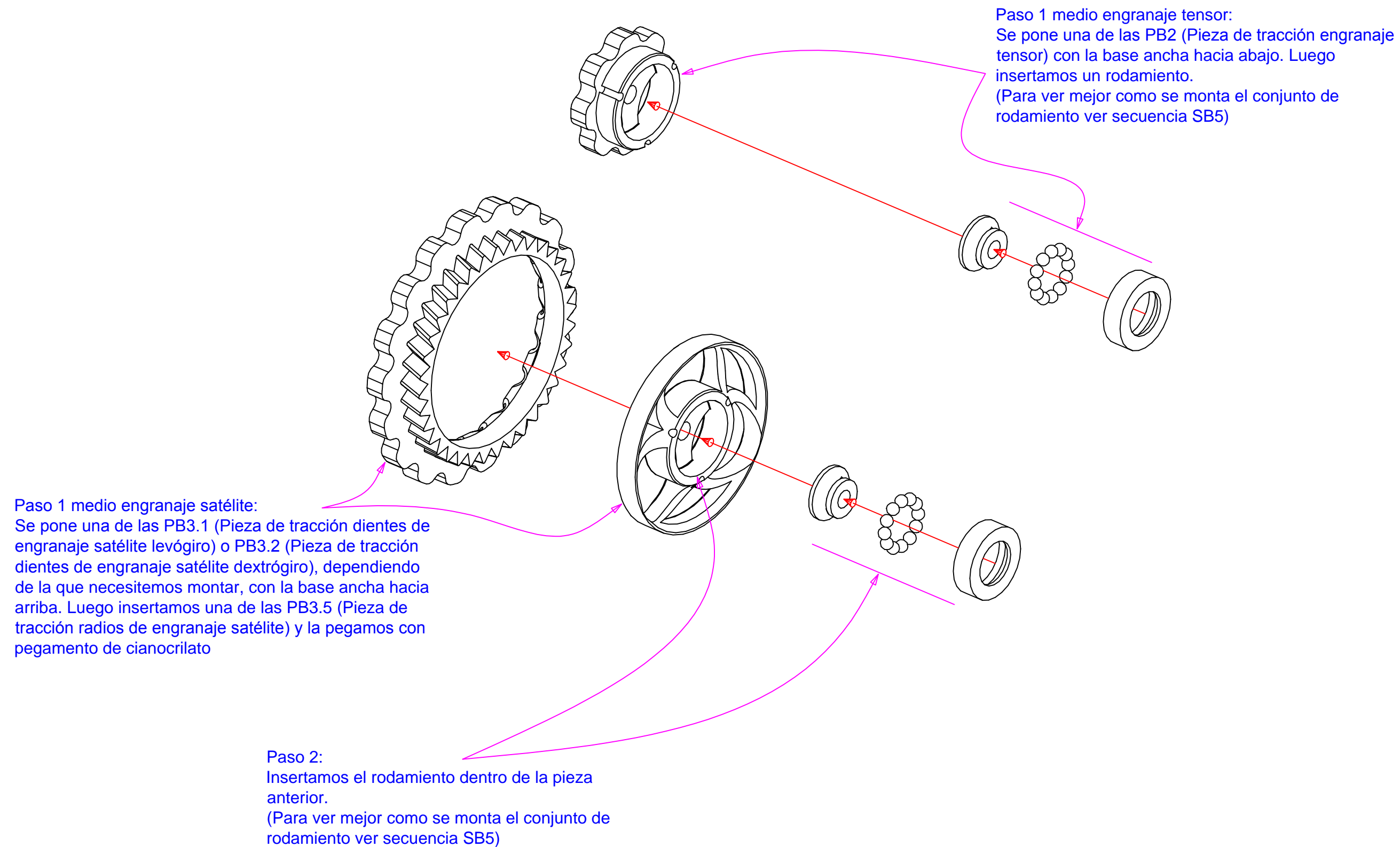
Paso 2:  
A cada uno de los conjuntos de medios engranajes, tanto los satélites como el tensor, lo atravesamos con uno de los pernos huecos de 8x40mm

Paso 3:  
Ponemos conjuntos de medios engranajes ensartados en cada uno de los pernos para formar así engranajes completos, tanto los satélites como los tensores, hay que poner los medios engranajes de manera simétrica para que los dientes queden en disposición helicoidal de doble vertiente.

Paso 4:  
Por último ponemos una de las PA8 (Pieza de chasis triángulo de tracción) de manera que encaje en los tres pernos, tiene que quedar justa pero no apretada, sino provocaremos un rozamiento innecesario y perjudicial para el sistema de tracción. Por último le ponemos silicona caliente en cada uno de los agujeros del triángulo a modo de fijación.

Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Secuencia montaje de pernos y engranajes	Pl. SB3





Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Secuencia montaje de rodamiento	PL. SB4



Paso 1:  
Ponemos uno de los  
centradores de PB5 (Pieza de  
tracción centrador y canastilla  
de rodamiento) con la base  
más estrecha mirando hacia  
arriba

Paso 2:  
Ponemos con mucho  
cuidado 12 bolas de 6mm  
alrededor del centrador

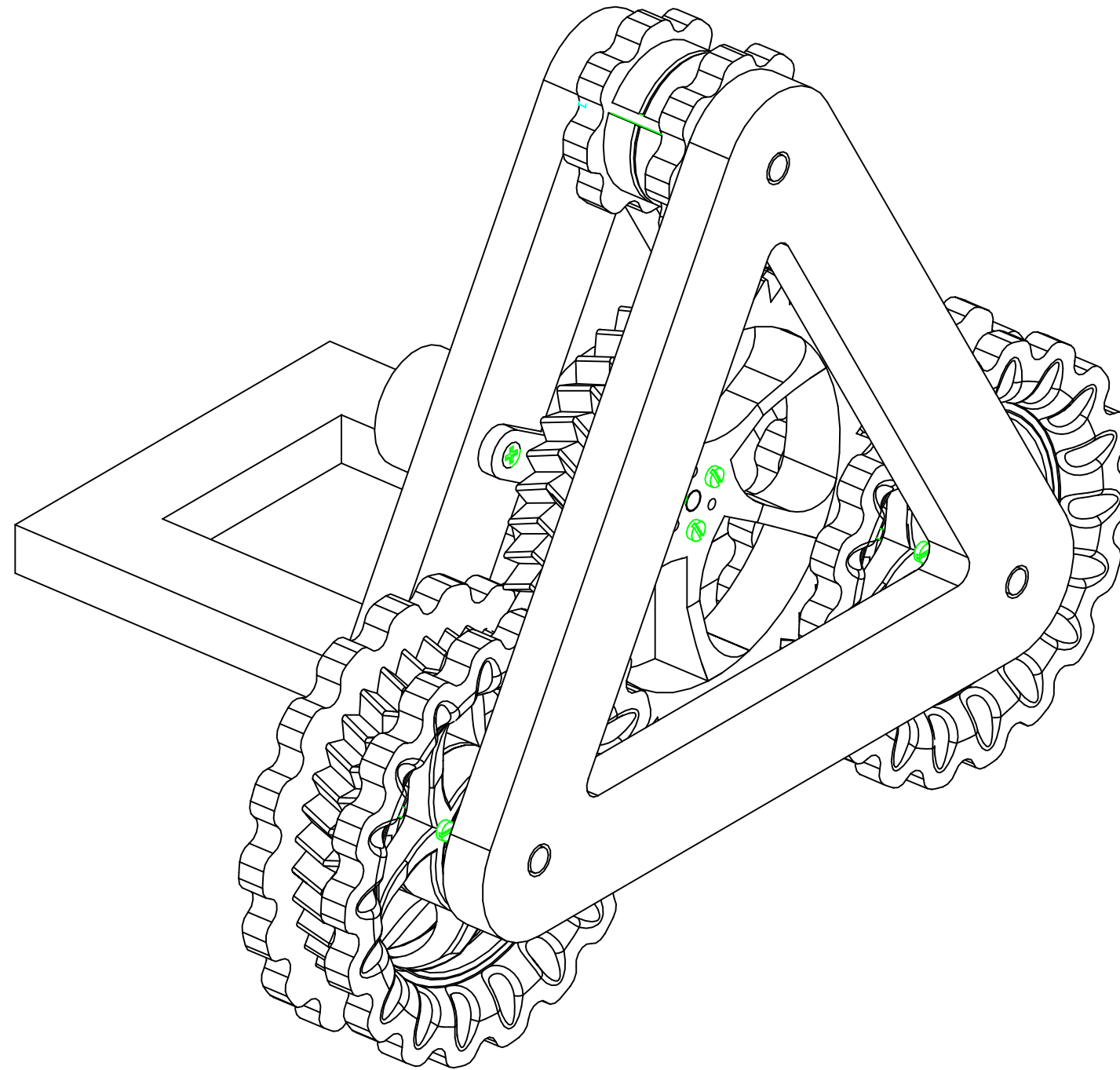
Paso 3:  
Ponemos una de las  
canastillas de PB5 (Pieza de  
tracción centrador y canastilla  
de rodamiento) con la base  
más estrecha en contacto  
con las bolas

Paso 4:  
De un golpe seco justo en la  
parte superior de la canastilla  
debemos embutir todas las  
bolas sin que se escape  
ninguna

Paso 5:  
Montar otro rodamiento de la  
misma manera que el anterior,  
ya que cada engranaje satélite  
lleva 2 rodamientos puestos  
uno al revés que el otro, justo  
como se ve en este esquema

Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Secuencia montaje de rodamiento	Pl. SB5

Así es como quedaría el  
sistema de tracción terminado,  
ahora solo faltaría montarle las  
cadenas



Escala:	Realizado por: Antonio Molina Cortés Salvador Moreno Gómez	Fecha: 11/05/2016
S.E.	Secuencia montaje de tracción completada	Pl. SB6