

# Trabajo de Fin de Grado Ingeniería Aeroespacial

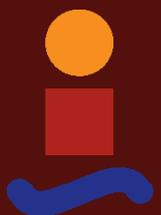
## Diseño de la automatización y la plataforma de un sensor de radiación solar directa en Arduino

Autor: Christian Naranjo Moreno-Arrones

Tutor: Ramón Andrés García Rodríguez y Eduardo Fernández Ca-  
macho

**Dpto. Ingeniería de Sistemas y Automática  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla**

Sevilla, 2022





Trabajo de Fin de Grado  
Ingeniería Aeroespacial

# **Diseño de la automatización y la plataforma de un sensor de radiación solar directa en Arduino**

Autor:

Christian Naranjo Moreno-Arrones

Tutor:

Ramón Andrés García Rodríguez y Eduardo Fernández Camacho

Dpto. Ingeniería de Sistemas y Automática  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2022



Trabajo de Fin de Grado:   Diseño de la automatización y la plataforma de un sensor de radiación solar directa en Arduino

Autor:           Christian Naranjo Moreno-Arrones

Tutor:           Ramón Andrés García Rodríguez y Eduardo Fernández Camacho

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:



# Agradecimientos

---

Quiero agradecer a todos los profesores que he tenido, tanto en la universidad como fuera de ella, por haber fomentado el desarrollo de mi curiosidad. También quiero agradecer a aquellos que me han formado como profesional y como persona, realizando un trabajo que nunca se podrá valorar lo suficiente. También a mi familia, que me ha dedicado todo su tiempo, todo su esfuerzo y todos sus recursos con tal de educarme y formarme lo mejor posible para afrontar la vida

*Christian Naranjo Moreno-Arrones  
Sevilla, 2022*



# Resumen

---

**E**n un mundo donde la forma de obtener energía está en el debate público, la que procede del Sol puede ser un valuarte importante en las próximas décadas. Por ello es crucial saber cómo aprovecharla al máximo. En ese sentido surge la idea de este TFG: diseñar la manera de automatizar de la forma más eficiente la energía solar directa recibida haciendo uso de la algoritmia y una plataforma adecuada al movimiento diario del Sol.



# Abstract

---

In a world where the way to obtain energy is in the public debate, the energy from the Sun can be an important value in the coming decades. That is why it is crucial to know how to make the most of it. That is where the idea of this TFG arises: to design the way to automate the direct solar energy received in the most efficient way, using algorithms and a suitable platform for the daily movement of the Sun.



# Índice

---

<i>Resumen</i>	III
<i>Abstract</i>	V
<b>1 Introducción</b>	<b>1</b>
1.1 Radiación solar	1
1.1.1 Origen	1
1.1.2 Dirección de incidencia	3
Irradancia solar directa	3
Irradancia solar difusa	3
Irradancia solar reflejada	3
1.1.3 Perturbaciones externas	4
1.1.4 Centrales termosolares	5
Central termosolar de canales parabólicos	5
Central termosolar de Heliostatos	6
Central termosolar con reflectores lineales Fresnel.	6
Central termosolar con discos parabólicos de Stirling.	7
<b>2 Ecuaciones solares</b>	<b>9</b>
2.1 Triángulo astronómico	9
Latitud y longitud	10
Declinación solar	10
Ángulo horario, $H_S$	10
<b>3 Hardware</b>	<b>13</b>
3.1 Arduino MICRO	13
3.2 Magnetómetro GY-271	14
3.3 Inclinómetro SW-520	15
3.4 Sensor de radiación NANO-ISSX/c	15
3.5 Servomotores de giro continuo	17
3.6 Lector de memoria tarjeta MicroSD	18
<b>4 Diseño del Pan and Tilt</b>	<b>19</b>
4.1 Base y cubierta del servo de azimuth	19
4.2 Estructura de conexión del servo de azimuth y el servo de elevación	24
4.3 Pieza de unión entre el servo de elevación y la cabeza	25
4.4 Cabeza inferior	27
4.5 Cabeza superior	31
4.6 Ensamblaje	37
<b>5 Montaje</b>	<b>39</b>
5.1 Instalación del inclinómetro SW520D	39

---

5.2	Instalación del magnetómetro GY-271	40
5.3	Instalación del sensor de radiación solar	40
5.4	Instalación de los servos	41
5.5	Instalación de la tarjeta SD	41
<b>6</b>	<b>Software</b>	<b>45</b>
6.1	Código con algoritmo igualando los 4 valores del sensor	45
6.2	Código con algoritmo octogonal	49
6.3	Algoritmo en espiral	55
<b>7</b>	<b>Resultados</b>	<b>61</b>
7.1	Toma de medidas del 5 al 7 de mayo	61
7.1.1	5 de mayo: Algoritmo octogonal	61
7.1.2	6 de mayo: Algoritmo en espiral	62
7.1.3	7 de mayo: Algoritmo con 4 valores del sensor	63
7.1.4	Comparación de los 3 días	64
7.2	Toma de medidas del 2 al 4 de junio	65
7.2.1	2 de junio: Algoritmo octogonal	65
7.2.2	3 de junio: Algoritmo con 4 valores del sensor	66
7.2.3	4 de junio: Algoritmo en espiral	67
7.2.4	Comparación de los 3 días	67
<b>8</b>	<b>Líneas de trabajo en el futuro</b>	<b>69</b>
	<i>Índice de Figuras</i>	71
	<i>Bibliografía</i>	75

# 1 Introducción

---

*“We are like tenant farmers chopping down the fence around our house for fuel when we should be using Nature's inexhaustible sources of energy – sun, wind and tide. ... I'd put my money on the sun and solar energy. What a source of power! I hope we don't have to wait until oil and coal run out before we tackle that.”*

THOMAS EDISON,

En esta introducción vamos a dilucidar qué es y cómo recibimos la energía solar en función de diversos factores como la posición geográfica del sensor, el día del año, la atmósfera terrestre, las inclemencias meteorológicas o las distintas perturbaciones que influyen en la órbita Tierra-Sol. Además, definiremos el objetivo principal de este proyecto y su utilidad en la industria.

## 1.1 Radiación solar

### 1.1.1 Origen

La energía producida por reacciones nucleares en el interior del Sol debe ser igual a la cantidad de energía que irradia a través de la superficie, de otra manera el Sol no podría ser estructuralmente estable durante largos periodos de tiempo. Una muestra de este fenómeno es que hay evidencias geográficas, en la propia composición de la superficie terrestre, de cierta estabilidad en la temperatura en un periodo cercano a 3000 millones de años. El principio físico que subyace es la liberación de energía en forma de calor y luz que resulta de transformación de hidrógeno en helio.

Prácticamente toda la radiación del Sol recibida en la Tierra se genera en la fotosfera, una delgada capa que rodea el manto convectivo. La fotosfera la componen átomos de diferente grado de ionización y electrones libres. Con estos ingredientes se producen un gran número de procesos de dispersión que generan un espectro de radiación similar a la radiación de Planck para un cuerpo negro en equilibrio a 6000K.

La estructura de la superficie solar no es regular debido al movimiento turbulento que se produce en capas más internas, columnas verticales ascendentes (más calientes) y otras descendentes (más frías). Otras irregularidades de la radiación solar incluyen fogonazos de llamas de corto periodo, así como manchas solares (regiones cercanas al fondo de la fotosfera con menor temperatura) apareciendo y desapareciendo en cuestión de días o semanas de manera irregular, pero estadísticamente con un periodo de 11 años. Las manchas solares aparecen por primera vez en latitudes de 30° o ligeramente superiores, alcanzan su actividad máxima cerca de los 15° de latitud y finaliza el ciclo cerca de los 8° de latitud.

Otro elemento a tener en cuenta es el viento solar. En la parte superior de la atmósfera terrestre, el viento solar tiene una densidad de aproximadamente  $10^{-20} \text{ kg/m}^3$ . Los iones son absorbidos por el campo magnético terrestre en los polos, dando lugar a fenómenos como auroras boreales y tormentas magnéticas. Variaciones en la actividad solar afecta al viento solar, que a su vez afecta al flujo de energía que llega a la Tierra.

La figura 1.2 resume las fuentes de radiación más importantes que llegan a la Tierra. Claramente, la radiación del Sol domina la distribución espectral. Las próximas contribuciones, unos 6 órdenes de magnitud por debajo

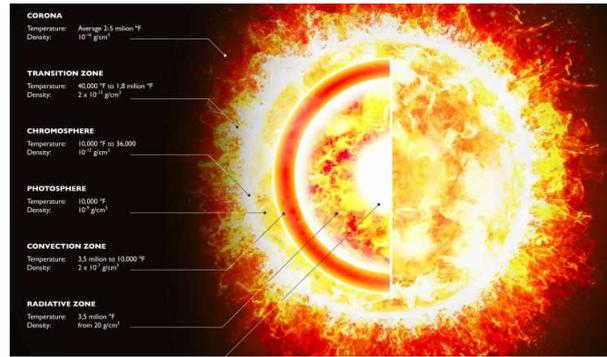


Figura 1.1 Estructura interna del Sol.

de la radiación visible, también tienen origen solar como la luz de la luna o el resplandor del aire. Más abajo, se encuentran la luz de las estrellas, la luz de nuestra propia galaxia y finalmente la luz extragaláctica, cuyo pico en la distribución espectral están en la región de microondas.

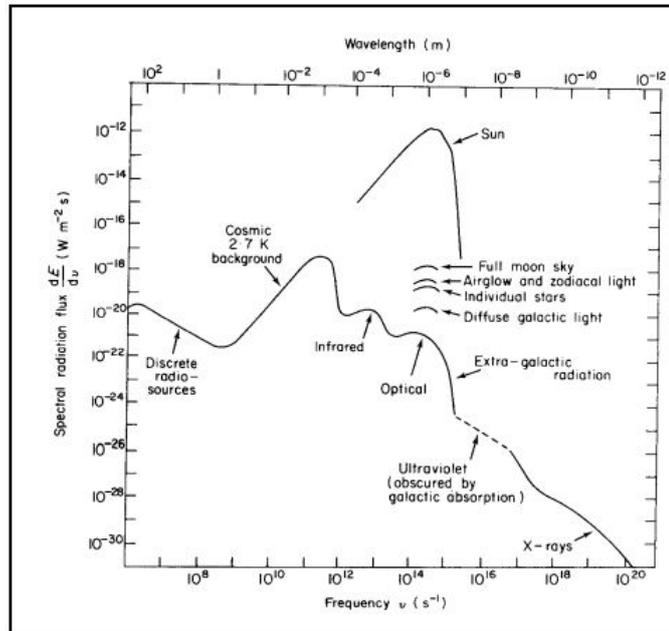


Figura 1.2 Distribución de frecuencias de algunas contribuciones a la radiación total.

El Sol emite un flujo de radiación de  $3.85 \cdot 10^{26}$  W, de lo cuales la Tierra absorbe  $1.1 \cdot 10^{17}$  W, que llega en forma de ondas electromagnéticas de difente longitud de onda, como podemos observar en la figura 1.5.

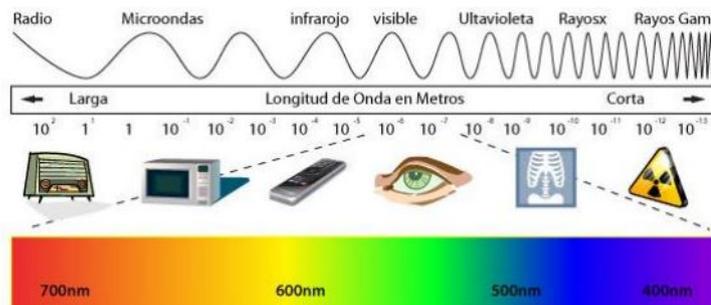


Figura 1.3 Espectro solar.

### 1.1.2 Dirección de incidencia

Una vez vista la procedencia de la radiación solar, las causas de su flujo irregular en origen, la importancia que tiene en comparación con otro tipo de radiaciones que llegan a la Tierra y la definición del espectro solar, se procede a analizar 3 categorías en función de la dirección de incidencia: directa, difusa y reflejada.

La irradiancia es la unidad utilizada para describir el valor de la potencia luminosa (energía/unidad de tiempo) incidente en un determinado instante por unidad de superficie de todo tipo de radiación electromagnética. Sus unidades de medida son:  $W/m^2$ .

El estudio de la dirección con la cual incide la radiación solar sobre los cuerpos situados en la superficie terrestre es de especial importancia cuando se desea conocer su comportamiento al ser reflejada. La dirección en que el rayo salga reflejado dependerá de la incidente.

#### Irradiancia solar directa

Es aquella que llega al cuerpo desde la dirección del Sol. Para que se vea más claro los órdenes de magnitud en términos energéticos se adjunta un mapa de la irradiancia solar normal en el planeta.

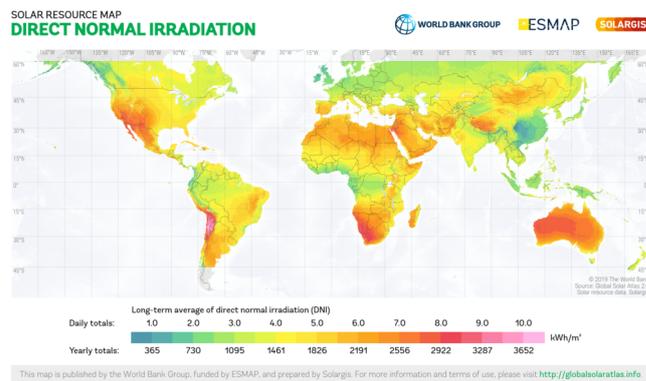


Figura 1.4 Mapa mundial de llegada de radiación solar directa.

#### Irradiancia solar difusa

Es aquella cuya dirección ha sido modificada por diversas circunstancias (densidad atmosférica, partículas u objetos con los que choca, reemisiones de cuerpos, etc.). Por sus características, esta luz se considera venida de todas direcciones. En un día nublado, por ejemplo, solo tenemos irradiancia difusa.

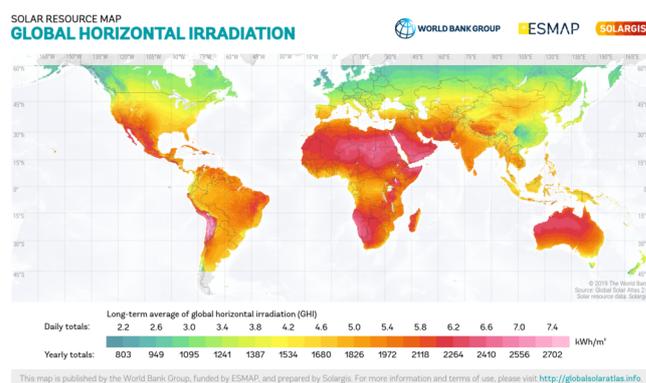


Figura 1.5 Mapa mundial de llegada de radiación solar total.

#### Irradiancia solar reflejada

Es la radiación directa y difusa que se recibe por reflexión en el suelo u otras superficies próximas.

Para comparar órdenes de magnitud de ambos tipos de radiación se adjunta un mapa mundial de la llega de radiación solar global, que junto a la figura 1.4 nos hace concebir qué peso tiene cada una. Centrémonos en

España no parece haber una gran diferencia en el mapa de colores por lo que podemos deducir que la gran mayoría de irradiación que llega es directa ( $\sim 5.5 \frac{kWh}{m^2}$ ).

### 1.1.3 Perturbaciones externas

Una vez visto cómo se genera la radiación solar y de qué manera llega a la Tierra, vamos a analizar las perturbaciones intrínsecamente ligadas a nuestro planeta que alteran la irradiación que llega a la superficie: la posición geográfica, la estacionalidad, la influencia de la atmósfera, las inclemencias meteorológicas o las características de la órbita Tierra-Sol.

Para comprender primero por qué hay variaciones en la llegada de radiación solar en función de la **posición geográfica** (latitud) y **el día del año**, hay que describir la órbita de la Tierra alrededor del Sol. La Tierra se mueve en una órbita elíptica, con el Sol en uno de sus focos y período aproximado de 365.25 días. La distancia más cercana entre la Tierra y el Sol es 147 millones de km aproximadamente, lo que se conoce como perihelio, y la más lejana 152 millones de kilómetros aproximadamente lo que se conoce como afelio. El plano de la órbita, conocido como eclíptica, forma un ángulo de  $23.5^\circ$  con el ecuador o el eje de la Tierra, lo que da lugar a la estaciones o, en un sistema inercial centrado en la Tierra, a una variación de la declinación solar de  $+23.5^\circ$ , en verano, a  $-23.5^\circ$ , en invierno.

Paradójicamente, esta disposición de los cuerpos hace que, en el Hemisferio Norte, el verano sea cuando el Sol está más alejado de la Tierra e invierno cuando está más cercano. Esto es debido a que los rayos solares serán más directos cuanto mayor sea la declinación, es decir, cuanto menor sea la diferencia entre nuestra latitud y la declinación solar con menor inclinación recibiremos los rayos. Este movimiento se ve muy bien en la siguiente figura, representando ambos cuerpos celestes en el Sistema Heliocéntrico Inercial.

El movimiento de rotación de la Tierra alrededor de su eje, con un periodo aproximado de 24 horas, hace que, desde nuestra posición, sólo "veamos" a la estrella durante un cierto tiempo de esas horas; tiempo que también dependerá de la declinación solar, es decir, del momento del año en que estemos.

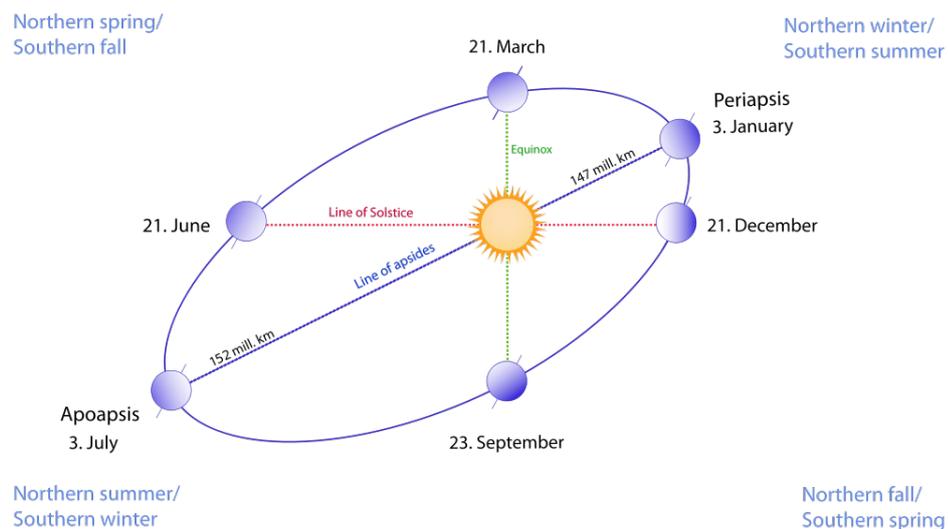


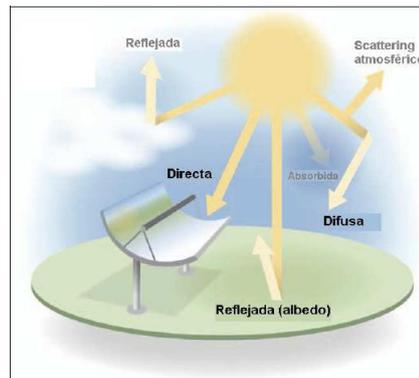
Figura 1.6 Órbita de la Tierra alrededor del Sol.

La radiación solar al pasar por la **atmósfera** sufre un proceso de debilitamiento por la dispersión (debida a los aerosoles), la reflexión (por las nubes) y la absorción (por las moléculas de gases y por partículas en suspensión), por lo tanto, la radiación solar reflejada o absorbida por la superficie terrestre (océano o continente) es menor a la del tope de la atmósfera. Esto depende de la longitud de onda de la energía transmitida y del tamaño y naturaleza de la sustancia que modifica la radiación. La superficie de la Tierra, suelos, océanos, y también la atmósfera, absorbe energía solar y la vuelven a irradiar en forma de calor en todas direcciones.

La radiación solar viaja en línea recta, pero los gases y partículas en la atmósfera pueden desviar esta energía, lo que se llama **dispersión**. La dispersión ocurre cuando un fotón se encuentra un obstáculo sin ser absorbido cambiando solamente la dirección del recorrido de ese fotón. La dispersión depende de la longitud de onda, en el sentido de que cuanto más corta sea ésta, tanto mayor será la dispersión. Moléculas de gas con tamaños

relativamente pequeño comparadas con la longitud de onda causan que la radiación incidente se disperse en todas las direcciones, hacia adelante y hacia atrás, este fenómeno es conocido como dispersión de Rayleigh. Aerosoles cuyos tamaños son comparables o exceden a las longitudes de onda de la radiación incidente, hacen que ésta no se disperse en todas las direcciones sino mayormente hacia adelante, fenómeno llamado dispersión de Mie. El proceso de la dispersión explica cómo un área con sombra o pieza sin luz solar está iluminada, le llega luz difusa o radiación difusa.

La capacidad de reflexión o fracción de la radiación reflejada por la superficie de la tierra o cualquier otra superficie se denomina **albedo**. El albedo planetario es en promedio de un 30%. El albedo es variable de un lugar a otro y de un instante a otro, depende de la cobertura nubosa, naturaleza de la superficie, inclinación de los rayos solares, partículas en el aire, etc



**Figura 1.7** Tipos de radiaciones solares que llegan a la superficie.

Con todos estos movimientos acoplados y capas heterogéneas que tiene que recorrer un rayo solar hasta llegar a la superficie parece que diseñar un modelo para predecir la radiación solar directa no es una cuestión baladí. De esa dificultad nace este proyecto para intentar crear un algoritmo que sea capaz de buscar la máxima eficiencia en la recepción de irradiancia solar mediante la entrada de datos al instante, la gestión de los mismos y, en base a estos, reposicionar el sensor.

#### 1.1.4 Centrales termosolares

Una vez vista el objeto de este proyecto es necesario extrapolar cuál es la utilidad a gran escala, es decir, analizar las distintas **centrales termosolares**.

En líneas generales, una central termosolar genera energía mediante la radiación solar, reflejando y concentrando los rayos del sol en un punto para obtener elevadas temperaturas. Sobre ese punto circula un fluido cuyo calor se utilizará para calentar agua hasta generar vapor. Dicho vapor moverá una turbina que, por la ley de inducción de Faraday, generará una corriente eléctrica que entrará en la red eléctrica. A diferencia de las centrales termosolares, las centrales fotovoltaicas generan la corriente transformando los fotones de la radiación solar en corriente (Efecto fotoeléctrico). El rendimiento de una central termosolar por área es superior a una central fotovoltaica, aunque su coste es superior. Existen diferentes tipos de centrales termosolares, estos son los principales:

##### Central termosolar de canales parabólicos

Esta tecnología usa espejos cóncavos montados en forma de canal para dirigir los rayos del sol hacia el fluido que circula por un conducto situado en el foco de la parábola. El calor adquirido se utiliza para generar vapor o se almacena en sales para su posterior uso. Los espejos son montados sobre una estructura capaz de girarlos. El objetivo es seguir la trayectoria solar a lo largo del máximo de horas posibles.

La complejidad y, por tanto, el aumento en el coste de fabricación de los espejos, radica en alcanzar la forma cóncava adecuada, necesaria para obtener un aprovechamiento óptimo de captación de la radiación solar.



**Figura 1.8** Central Termosolar de canales parabólicos de «La Africana» promovida por el grupo Magtel.

### **Central termosolar de Heliostatos**

Este tipo de centrales termosolares requiere de una gran superficie para ubicar los heliostatos, que son los espejos que utiliza este tipo de central. Los espejos se montan sobre una estructura capaz de girar en el plano horizontal y vertical, de esta manera reflejan los rayos durante el día sobre una torre concentrándose una elevada temperatura.

Esta tecnología presenta una gran complejidad como consecuencia de que cada espejo requiere de un sistema electrónico independiente de posicionamiento.



**Figura 1.9** Central Termosolar PS10 de Abengoa en Sanlúcar la Mayor, primera Central de Heliostatos en el mundo..

### **Central termosolar con reflectores lineales Fresnel.**

Las centrales termosolares con reflectores lineales Fresnel utilizan espejos planos lo que supone la gran diferencia respecto a las centrales de espejos cóncavos. Esta característica hace más barata la instalación de este tipo de centrales ya que la fabricación de estos espejos convencionales es mucho más simple.



**Figura 1.10** Central Termosolar con reflectores lineales de Magtel en Aerópolis.

**Central termosolar con discos parabólicos de Stirling.**

Una central termosolar con discos parabólicos de Stirling cuenta con concentradores parabólicos independientes. En el foco se coloca un motor que transforma la energía recibida térmica en un giro mecánico. El motor lleva acoplado a su vez un generador eléctrico que convierte el giro mecánico en corriente eléctrica. Así cada unidad produce energía eléctrica de manera independiente.



**Figura 1.11** Central Termosolar con discos parabólicos en Villarobledo, primera en España de este tipo..



## 2 Ecuaciones solares

La primera acción que llevará a cabo nuestro dispositivo será posicionarse de forma que apunte al Sol y a partir de ahí inicializar el algoritmo de búsqueda, en función de los datos de entrada de los 4 fotodiodos, y resituarse conforme el Sol se vaya desplazando.

Por ello es necesario una primera estimación teórica de la posición del Sol desde el punto de vista del sensor (Sistema de referencia topocéntrico), es decir, conocer el azimut y la elevación de la estrella. En esta sección se analizará este problema haciendo uso del triángulo astronómico, con distintas hipótesis simplificadoras y describiremos cuáles son los parámetros de entrada.

### 2.1 Triángulo astronómico

Sea un cuerpo "infinitamente" distante de la Tierra, con declinación  $\delta_S$ , cuyas coordenadas topocéntricas respecto a un observador O son su elevación,  $h$ , y su azimut,  $Az$ . El observador se encuentra en un punto de la Tierra de latitud  $\phi$  y el ángulo horario del cuerpo celeste respecto al observador es  $H_S$ . Recordemos que el ángulo horario es la diferencia entre el meridiano del observador y el meridiano en el que se encuentra S, de forma que:

$$H_S = LST - AR_S = GST_0 + \lambda + \omega_T \cdot t - AR_S \quad (2.1)$$

El triángulo astronómico permite obtener una pareja de datos (coordenadas de S topocéntricas o geocéntricas, coordenadas del observador) a partir del resto.

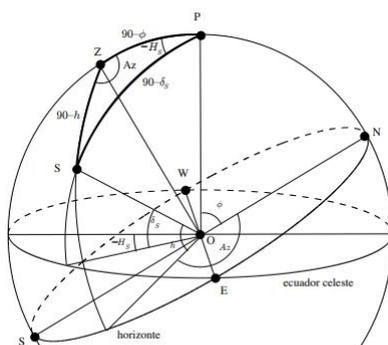


Figura 2.1 Origen del triángulo astronómico.

La clave para plantear el triángulo consiste en desplazar el centro del sistema de referencia geocéntrico al observador (esfera celeste centrada en observador); no se alteran las coordenadas angulares de S por la hipótesis de estar "infinitamente" distante de la Tierra.

En la figura, Z es el zenit, N, S, E, W los puntos cardinales, y P la dirección del eje de la Tierra (hacia la estrella polar).

Extrayendo el triángulo esférico resultante y haciendo uso de la trigonometría esférica procedemos a analizar matemáticamente el problema.

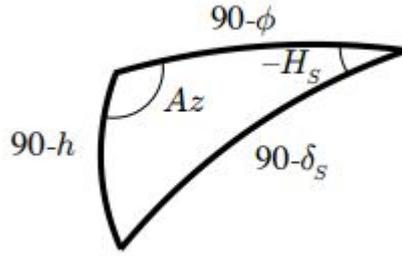


Figura 2.2 Triángulo astronómico.

Del triángulo, obtenemos aplicando dos veces la ley de cosenos y recordando que  $\text{sen}(90 - \alpha) = \text{cos}(\alpha)$  y  $\text{cos}(90 - \alpha) = \text{sen}(\alpha)$ , se tiene:

$$\text{sen}(\delta_S) = \text{sen}(\phi) \cdot \text{sen}(h) + \text{cos}(\phi) \cdot \text{cos}(h) \cdot \text{cos}(Az) \quad (2.2)$$

$$\text{sen}(h) = \text{sen}(\phi) \cdot \text{sen}(\delta_S) + \text{cos}(\phi) \cdot \text{cos}(\delta_S) \cdot \text{cos}(H_S) \quad (2.3)$$

Usando las dos ecuaciones podemos obtener una pareja cualesquiera de datos, en nuestro caso tenemos que resolver un problema de observación ya que nuestras incógnitas son  $h$  y  $Az$ .

Para resolver las ambigüedades de signo se hace uso de la ley de senos  $\frac{\text{sen}(Az)}{\text{cos}(\delta_S)} = -\frac{\text{sen}(H_S)}{\text{cos}(h)}$  y los límites físicos  $\delta_S, \phi, h \in [-90^\circ, 90^\circ]$ .

A continuación, se procederá a describir y a modelar las variables de entrada (todas aquellas que no son  $h$  y  $Az$ ).

### Latitud y longitud

La latitud ( $\phi$ ) y la longitud ( $\lambda$ ) de nuestro dispositivo son datos de entrada de nuestro algoritmo.

Haciendo uso de un mapa o, para más precisión, de alguna herramienta interactiva en Internet se pueden obtener estos parámetros.

### Declinación solar

Como se describió cuando se explicaron las características de la órbita Tierra-Sol, en la Introducción, el ángulo que forman el plano ecuatorial de la Tierra y la línea que une los centros del Sol y de la Tierra cambia en cada instante. Este ángulo es llamado declinación solar  $\delta_S$ . La declinación es  $0^\circ$  en los equinoccios de primavera y otoño y tiene un valor aproximado de  $23.5^\circ$  en el solsticio de verano y cerca de  $-23.5^\circ$  en el solsticio de invierno.

Para la declinación solar pueden consultarse efemérides. También puede determinarse mediante la aplicación de fórmulas y expresiones aproximadas, que dan su valor con diversos grados de precisión reportados por numerosos autores, la más importante de ellas y las ampliamente citada en la literatura solar corresponde a la de Spencer (1971), que es la que utilizaremos en nuestro dispositivo. La ecuación presenta la siguiente expresión, en radianes:

$$\begin{aligned} \delta_S = & 0.006918 - 0.399912 \cdot \text{cos}(\Gamma) + 0.0070257 \cdot \text{sen}(\Gamma) - 0.006758 \cdot \text{cos}(2\Gamma) + 0.000907 \cdot \text{sen}(2\Gamma) \\ & - 0.002697 \cdot \text{cos}(3\Gamma) + 0.00148 \cdot \text{sen}(3\Gamma) \end{aligned} \quad (2.4)$$

Donde  $\Gamma$ , llamado ángulo horario, viene dado por:  $\Gamma = \frac{2\pi \cdot (d_n - 1)}{365}$ , donde  $d_n$  corresponde al número del día juliano del año ( $1 \leq d_n \leq 365$ ), variando desde 1 para el 1 de enero, hasta 365 para el 31 de diciembre.

### Ángulo horario, $H_S$

Como se ha descrito anteriormente, el ángulo horario es la diferencia entre el meridiano del observador y el meridiano en el que se encuentra el Sol. La ecuación que utilizamos para calcular este ángulo en el algoritmo

es la siguiente:

$$H_S = GST + \lambda - AR_S \quad (2.5)$$

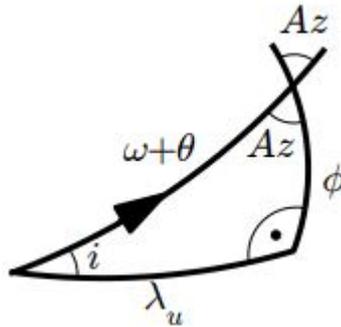
El primer término, GST (Greenwich Sidereal Time), es el ángulo que forma (en un determinado instante) el meridiano de Greenwich con el primer punto de Aries ( $\gamma$ ) el cual lo determina la línea de intersección entre la eclíptica y el ecuador terrestre, o lo que es lo mismo, la línea que une el Sol y la Tierra justo en el equinoccio de primavera.

Su interés reside en ubicar el meridiano de Greenwich, en un momento dado, en el Sistema Geocéntrico Ecuatorial, un sistema de referencia independiente de la rotación de la Tierra sobre su eje.

Como nuestras mediciones se generan en la Tierra no podemos esquivar la rotación terrestre por lo que GST debe ser calculado mediante fórmulas o efemérides. Para este proyecto hemos optado por obtenerlo mediante una herramienta web del National Astronomical Observatory of Japan (NAOJ), en el que se deben introducir el día y la hora local (URL: [https://eco.mtk.nao.ac.jp/cgi-bin/koyomi/candel/gst\\_en.cgi](https://eco.mtk.nao.ac.jp/cgi-bin/koyomi/candel/gst_en.cgi))

El segundo término es la longitud,  $\lambda$ , parámetro conocido.

El tercer término es la ascensión recta del Sol, es decir, el ángulo que forma en cada instante con el primer punto de Aries. Para resolverlo hacemos uso de un nuevo triángulo esférico determinado por el meridiano del observador, el ecuador terrestre y la eclíptica vista desde un sistema de referencia geocéntrico inercial.



**Figura 2.3** Triángulo esférico utilizado para calcular la ascensión recta del sol.

El parámetro  $i$  no es más que la diferencia angular entre el ecuador y la eclíptica, es decir,  $23.5^\circ$ .

Lo que en la figura llama  $\lambda_u$  es nuestra incógnita, la ascensión recta.

Por último el parámetro  $\omega + \theta$  caracteriza lo alejados que estamos del primer punto de Aries (Equinoccio de Primavera), es decir, el día concreto del año.

Haciendo uso de la trigonometría esférica llegamos a la siguiente expresión:

$$\tan(AR_S) = \cos(23.5^\circ) \cdot \tan(\omega + \theta) \quad (2.6)$$

Si suponemos circular la órbita de la Tierra alrededor del Sol podemos modelar  $\omega + \theta$  como:

$$\omega + \theta = \frac{\Delta T}{365.25} \cdot 360^\circ \quad (2.7)$$

El término  $\Delta T$  no es más que la diferencia entre la fecha actual y el Equinoccio de Primavera.

Por lo tanto, despejando de 2.6:

$$AR_S = \arctan(\cos(23.5^\circ) \cdot \tan(\omega + \theta)) \quad (2.8)$$

Como la órbita terrestre es simétrica habrá que resolver la ambigüedad resultante en función de la época del año.

La hipótesis introducida (órbita Tierra-Sol circular) incorpora un error máximo de  $2^\circ$ , por lo que como se está posicionando el sensor en una primera aproximación, es un error asumible.

De esta manera quedan caracterizados todos los parámetros necesarios para calcular la elevación y el azimut del Sol en una posición geográfica determinada en un momento concreto del año y del día.



## 3 Hardware

---

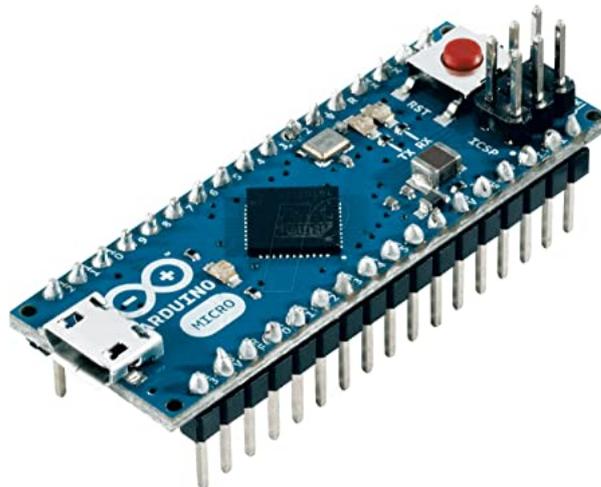
El hardware utilizado para construir el dispositivo es el siguiente:

- *Arduino Micro*
- *Magnetómetro GY-271*
- *Inclinómetro SW-520*
- *Sensor de radiación NANO-ISSX/c*
- *2 servomotores "Parallax" de giro continuo*
- *Lector de tarjeta MicroSD*

En este capítulo describiremos los medios electrónicos físicos utilizados para el desarrollo del proyecto.

### 3.1 Arduino MICRO

La placa Arduino es una placa electrónica basada en el microcontrolador ATmega32U4, desarrollado en conjunto con Adafruit. Cuenta con 20 pines digitales de entrada / salida, de los cuales 7 se pueden utilizar como salidas PWM y 12 como entradas analógicas.



**Figura 3.1** Arduino Micro.

Tiene un oscilador de cristal de 16 MHz, una conexión micro USB, una cabecera ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador, basta con conectarlo a un ordenador con un cable micro USB para empezar. Tiene un factor de forma que le permite ser fácilmente colocado en un protoboard.

La placa ArduinoMICRO se puede programar con el software de Arduino (IDE).

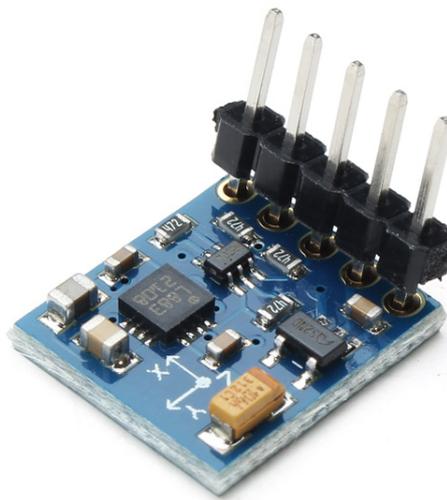
La placa puede ser alimentada a través de la conexión micro USB o con una fuente de alimentación externa.

La fuente de alimentación se selecciona automáticamente. La alimentación externa puede proceder de una fuente de alimentación o batería de CC. Los cables desde una batería o fuente de alimentación de CC deben conectarse a los pines GND y Vin. La placa puede funcionar con un suministro externo de 6 a 20 voltios, pero si se suministran menos de 7V, el pasador de 5V puede suministrar menos de cinco voltios y la placa se puede volver inestable.

El ATmega32U4 tiene 32 KB (con 4 KB utilizado por el gestor de arranque). También tiene 2,5 KB de SRAM y 1 KB de EEPROM (que puede ser leído y escrito con la biblioteca EEPROM ).

### 3.2 Magnetómetro GY-271

Una brújula digital es un sensor que mide el valor del campo magnético en tres ejes. Con esta medición, es posible estimar la orientación del dispositivo respecto al campo magnético de la Tierra. La utilizamos para orientar el dispositivo en azimut usando como referencia el Norte geográfico, que no es más que el Norte magnético corregido con la declinación, que depende de las coordenadas de medición.



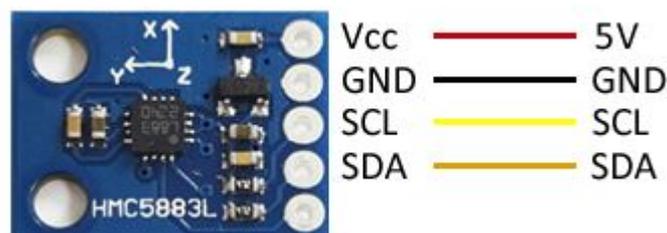
**Figura 3.2** Brújula digital GY-271.

El chip Honeywell HMC5883 es un chip de superficie que incorpora tres sensores de magnetoresistencia, cancelación de desfases, y conversores de 12 bits, lo que le proporciona una precisión de  $\pm 2^\circ$ .

La comunicación con el chip Honeywell HMC5883 se realiza a través del bus I2C, por lo que es sencillo obtener los datos medidos. La tensión de alimentación es de bajo voltaje de 1.8 a 3.3V.

Incluye un regulador de voltaje que permite alimentar directamente a 5V.

La conexión con Arduino Micro es muy sencilla:



**Figura 3.3** Conexiones del magnetómetro GY-273 (equivalente a las de GY-271) con Arduino Micro.

### 3.3 Inclinómetro SW-520

Un sensor de inclinación es un dispositivo que proporciona una señal digital en caso de que su inclinación supere un umbral. Este tipo de sensor no permite saber el grado de inclinación del dispositivo, simplemente actúa como un sensor que se cierra a partir de una cierta inclinación. En nuestro proyecto lo utilizaremos para fijar el grado de libertad de elevación en  $0^\circ$  cada vez que se inicialice el sensor, y a partir de esa posición lograr que se alcance el valor de elevación teórico que proviene de las Ecuaciones Solares.



Figura 3.4 Inclinómetro SW-520D.

Antiguamente, estos sensores se constituían ubicando una gota de mercurio en el interior de una ampolla de vidrio, en cuyo interior se alojaban dos conductores. A partir de cierta inclinación la gota de mercurio se desplazaba, cerrando el contacto entre ambos conductores.

En la actualidad, por motivos medioambientales, casi todos los sensores de mercurio han sido desplazados por sensores Tilt de doble esfera. Se dispone de un cilindro cuya pared constituye un contacto eléctrico, mientras que el otro contacto está localizado en el centro de la base. Al inclinar lo suficiente el dispositivo ambas esferas constituyen un puente entre ambos contactos, cerrando el circuito. El funcionamiento interno se muestra en la siguiente figura.

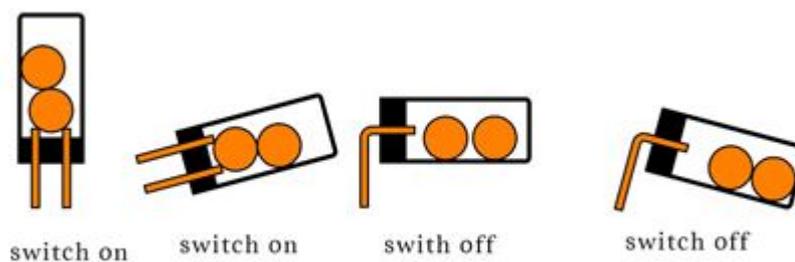
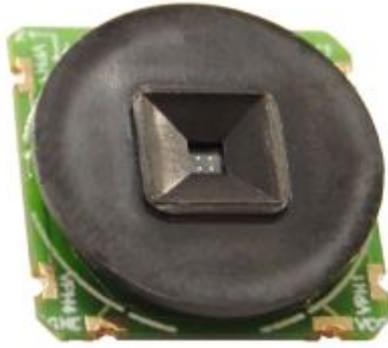


Figura 3.5 Caption.

### 3.4 Sensor de radiación NANO-ISSX/c

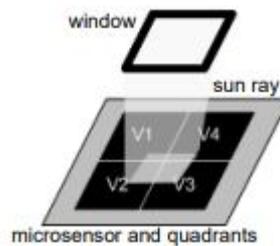
Este sensor mide la incidencia de los rayos del Sol a partir de un dispositivo de 4 fotodiodos que reciben los rayos filtrados por una ventana hace que la muestra sea más certera.



**Figura 3.6** NANO-ISSX/c.

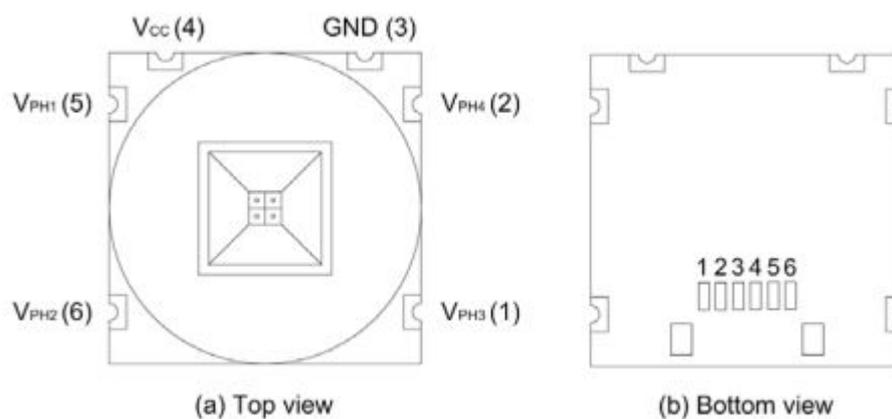
Dependiendo del ángulo de incidencia se recibirán 4 valores analógicos diferentes, pertenecientes a cada uno de los 4 fotoreceptores que contiene el sensor.

Mediante el uso de esos 4 voltajes seremos capaces de crear diferentes algoritmos que sean capaces de seguir



**Figura 3.7** Ventana de filtración del sensor de radiación.

al Sol de manera eficiente. Es un hecho geométrico que, dada la lejanía del Sol, se pueden considerar los rayos que vemos en la Tierra como paralelos, por lo que la incidencia máxima corresponderá cuando los 4 valores de incidencia coincidan. Esta será la base física del desarrollo de los diferentes algoritmos de seguimiento. Las salidas que permiten la conexión con Arduino se describen en la siguiente figura.



**Figura 3.8** Pines de conexión del sensor de radiación NANO ISSX/c.

### 3.5 Servomotores de giro continuo

Un servo de rotación continua es una variante de los servos normales, en los que la señal que enviamos al servo controla la velocidad de giro, en lugar de la posición angular como ocurre en los servos convencionales. Los servos de rotación continua, al igual que los servos convencionales, admiten tensiones de alimentación



**Figura 3.9** Servomotor Parallax de giro continuo.

de entre 4,8V a 7,2V. También al igual que sus hermanos incorporan un reductor interno por lo que, en general, proporcionan un alto par y baja velocidad máxima, en torno a 1-2 rpm

El control de un servo de rotación continua es idéntico al de un servo convencional, sólo varía el significado

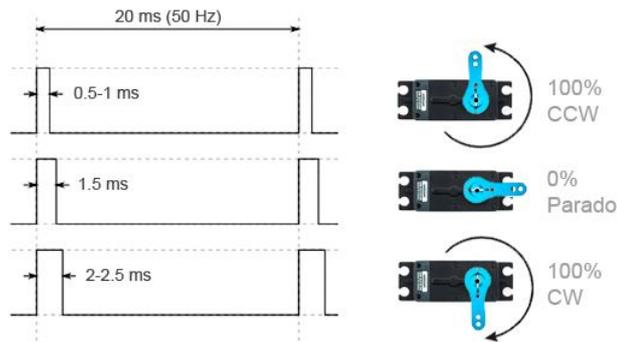


**Figura 3.10** Relación entre la velocidad angular del servo y el ancho de banda de la señal.

de la señal de control, que en lugar de transformarse en ángulo de posición se interpreta como velocidad angular, en ambos sentidos de giro.

Los servos de rotación continua son una forma sencilla de conseguir un motor con control de velocidad, sin tener que añadir dispositivos adicionales como controladores o encoders como ocurre en el caso de motores DC o paso a paso, ya que el control está integrado en el propio servo.

En nuestro caso, nos interesa un giro con velocidad pequeña y el control en posición lo lograremos con la velocidad angular, que haremos que sea una constante, y la función de Arduino `millis()`. Este método se explicará con más detalle en la descripción del Software.

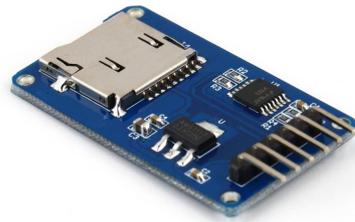


**Figura 3.11** Ancho de banda de la señal para el servo parado, a favor y en contra de las agujas del reloj.

### 3.6 Lector de memoria tarjeta MicroSD

Este elemento permite leer y escribir datos en una tarjeta MicroSD, dando la flexibilidad de poder extraerlos cada vez que se requieran y trabajar con ellos en el ordenador. Con este material en bruto, proveniente de distintos sensores, se podrá realizar un análisis y sacar conclusiones una vez acabado el experimento. Esto nos permite utilizar el sensor durante periodos de tiempo más largos (horas), salvando el obstáculo que supone la poca memoria que tiene la placa de Arduino.

A nivel interno, el dispositivo cuenta con un nivelador de señal de 5V a 3.3V, los pines para la interfaz SPI, los pines de alimentación de 5V y un regulador lineal de tensión de 5V a 3.3V.



**Figura 3.12** Lector de tarjetas MicroSD mediante bus SPI.

## 4 Diseño del Pan and Tilt

---

En este apartado se describe el proceso del diseño de un sistema mecánico con dos grados de libertad, elevación y azimuth, compatible con los servos utilizados y con la protoboard de 8.5x5.5 cm y que posteriormente se imprimirá en una impresora 3D.

Para la implementación de los diseños se ha utilizado la herramienta gráfica CATIA V5 R19.

### 4.1 Base y cubierta del servo de azimuth

Esta pieza está diseñada para fijarse a una base de madera mediante 4 tornillos y proporcionar estabilidad al conjunto.

Para ello, primero se creó en un "sketch" un rectángulo cuyas proporciones son 58 mm por 22 mm.

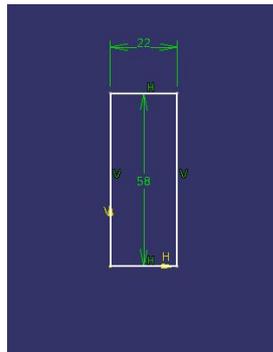


Figura 4.1 Sketch del rectángulo de la base del mecanismo.

Posteriormente, se le da la altura necesaria atendiendo a las especificaciones dimensionales del servo, en este caso 42 mm. Para hacer encajar el servo que hace el movimiento de azimuth se le da un grosor a la pieza de 1

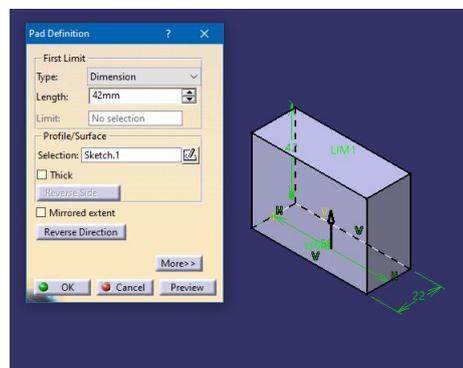


Figura 4.2 Altura de la base.

mm. Hay que tener en cuenta que por cada lado se deja un juego de 0.5 mm, suficiente para que el servo entre sin problemas pero restringiendo el movimiento para evitar inexactitudes en las medidas o que el par no ejerza suficiente fuerza sobre el resto de las piezas.

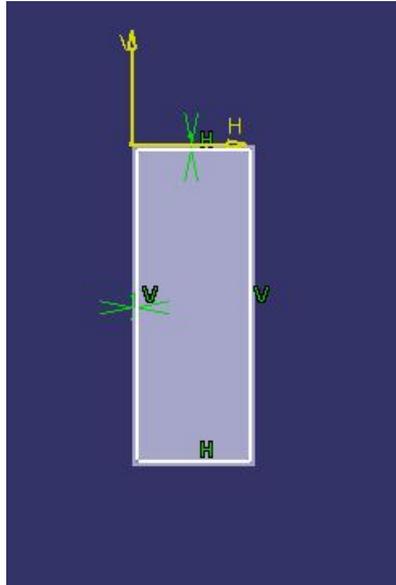


Figura 4.3 Sketch del grosor de la base.

El último paso para crear el continente del servo de azimuth es hacer hueca la pieza con las dimensiones de altura de nuestro servo y un juego de 0.5 mm; para ello la altura necesaria es 42 mm.

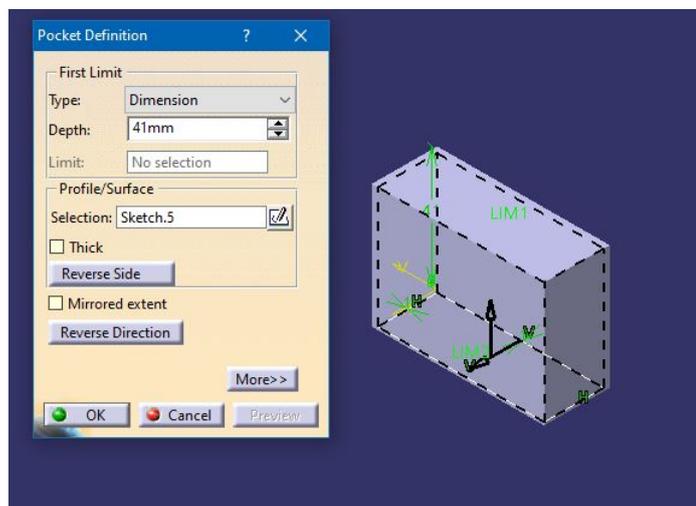
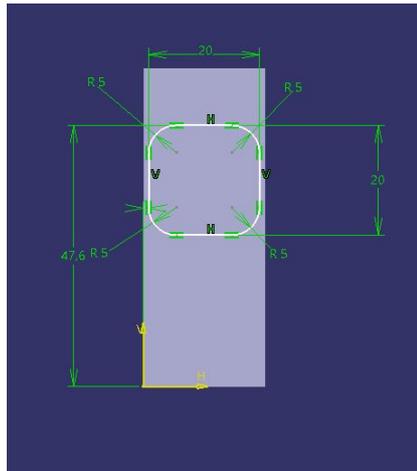


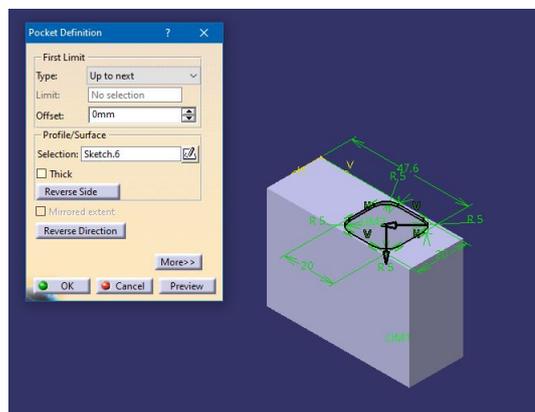
Figura 4.4 Altura para hacer la pieza hueca.

Ahora se va a proceder a crear el agujero por donde saldrán las aspas que transmitirán el par. Para ello se realiza un sketch de 20 mm por 20 mm con los bordes redondeados con radio 5 mm y en una posición tal que el eje de giro del servo coincida con el centro geométrico de la figura.



**Figura 4.5** Sketch del agujero para el eje de giro.

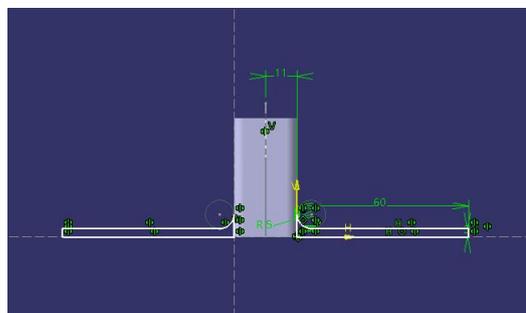
La pieza una vez comandado al programa que haga el agujero luce tal que así.



**Figura 4.6** Pieza 3D con el agujero para dar salida al eje de giro.

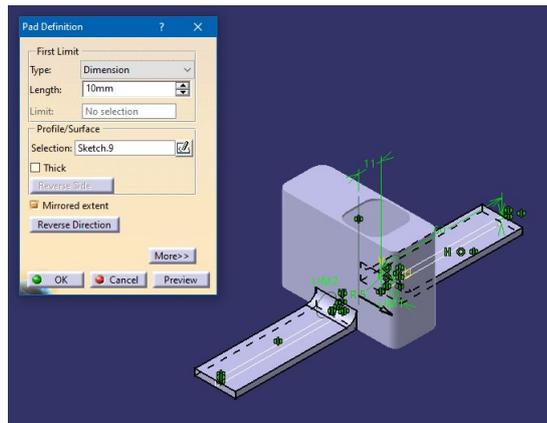
Para hacer una estructura estable es necesario aumentar la superficie de contacto de la pieza con la base de madera a la que irá atornillada. Para esto se diseñan unos salientes los cuales llevarán un agujero para anclarlos de manera correcta. El proceso de creación es el siguiente:

1. Se hace el siguiente sketch en uno de los dos planos de simetría de la pieza, con una longitud de 60 mm y un radio de redondeo de 5 mm.



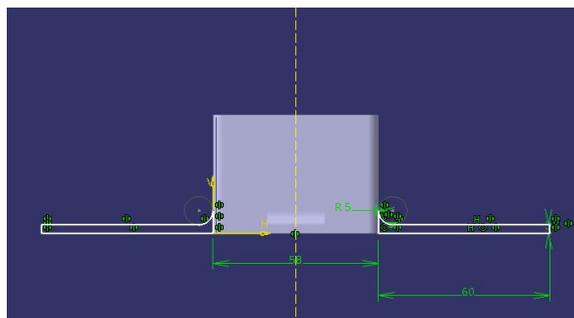
**Figura 4.7** Sketch de la primera estructura que da estabilidad al mecanismo .

2. Se le da una anchura de 10 mm.



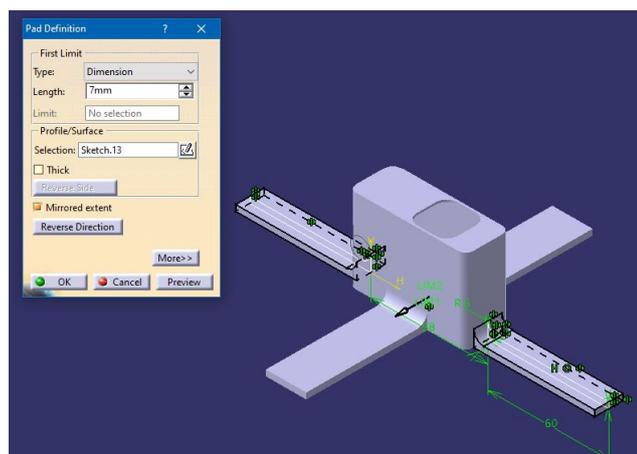
**Figura 4.8** Ancho de la primera estructura que da estabilidad al mecanismo .

3. De manera análoga se realiza el sketch de la segunda estructura en el segundo plano de simetría de la pieza.



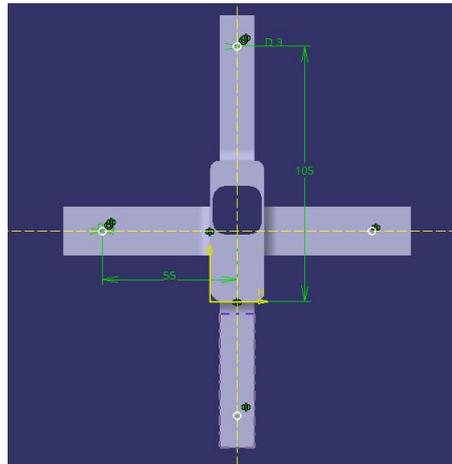
**Figura 4.9** Sketch de la segunda estructura que da estabilidad al mecanismo .

4. Se le da un ancho de 7 mm.



**Figura 4.10** Ancho de la segunda estructura que da estabilidad al mecanismo.

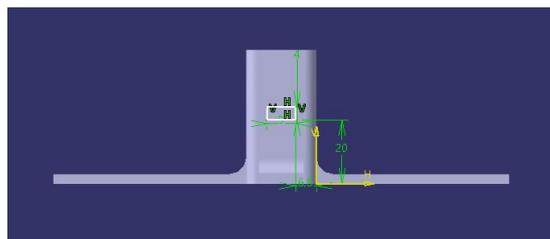
5. Para poder fijar adecuadamente la pieza base se crea un agujero de 3 mm de diámetro en cada uno de los salientes.



**Figura 4.11** Agujeros para atornillar la base.

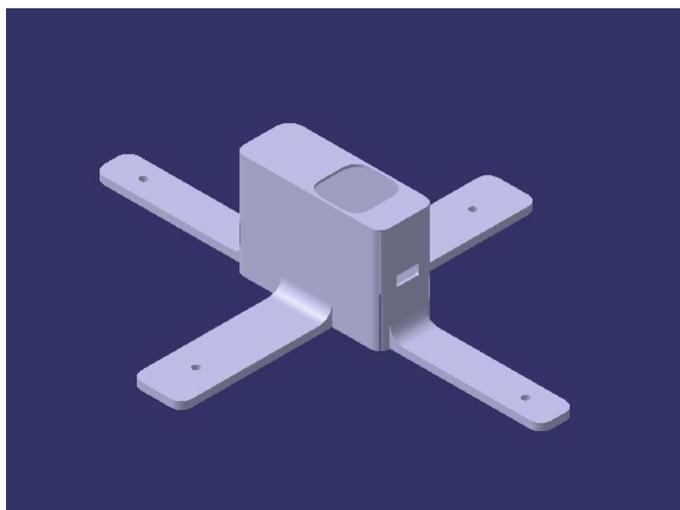
Finalmente, para poder sacar el cable al exterior y hacer posible que se conecte al Arduino se crea un agujero en uno de los lados.

5. Para poder fijar adecuadamente la pieza base se crea un agujero de 3 mm de diámetro en cada uno de los salientes.



**Figura 4.12** Agujero para el cable que conecta el servo de azimuth al Arduino.

Finalmente, la pieza acabada en perspectiva isométrica luce de la siguiente manera.



**Figura 4.13** Base y cubierta del servo de azimuth en perspectiva isométrica.

## 4.2 Estructura de conexión del servo de azimuth y el servo de elevación

Posteriormente, se diseñó una estructura que hiciera de nexo entre el servo de azimuth (fijo a la base de madera) y el servo que modifica la elevación. Para ello se optó por una plataforma en forma de L y con la "data sheet" de los servos se ajustaron las distancias y el tipo de tornillería a utilizar.

El sketch de la L tiene unas dimensiones de 65 mm por 78 mm con un radio de redondeo en la unión de 3 mm.

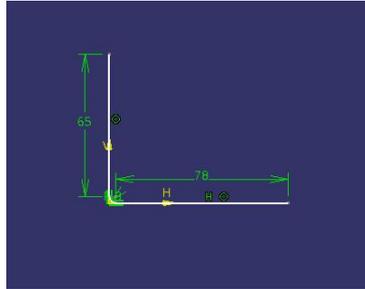


Figura 4.14 Sketch de la estructura en forma de L.

A continuación, se establece un grosor de 4 mm y un ancho de 28 mm, suficiente para aportar la rigidez para soportar el par al que será sometida por el peso de las piezas que se mostrarán después.

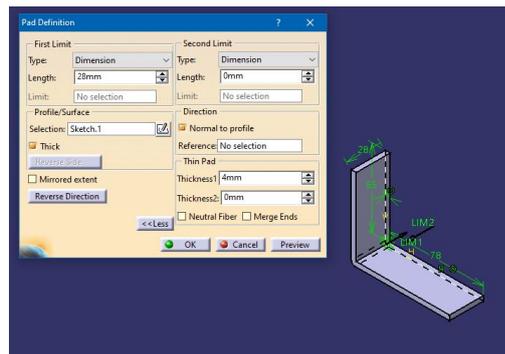


Figura 4.15 Estructura primaria en forma de L.

El servo de elevación tiene 4 agujeros de 2 mm de diámetro para permitir un anclaje más fácil. Con las restricciones dimensionales del servo se diseña un nuevo sketch en la cara vertical de la estructura.

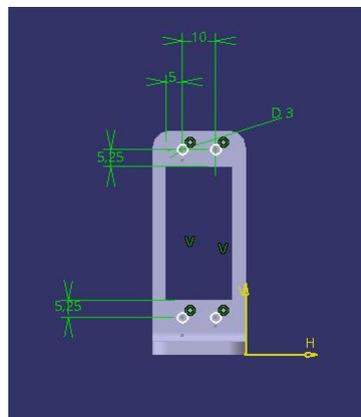
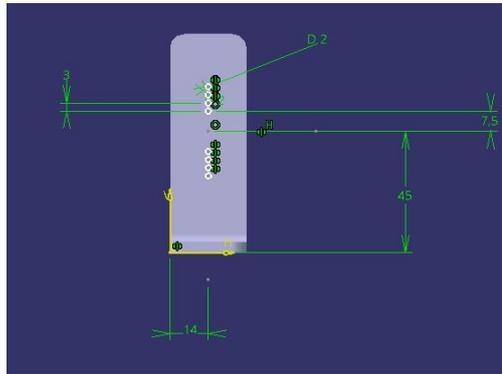


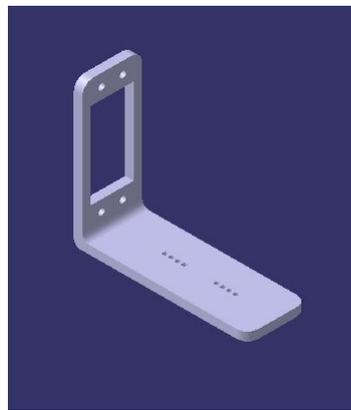
Figura 4.16 Sketch para el anclaje del servo de elevación en la estructura.

Para unir la L al servo de azimut se hacen 8 agujeros de 2 mm de diámetro para fijar a la base de la estructura dos de las aspas de la pieza que se unen al par del motor. Al unirlos mediante 8 tornillos la fuerza para mover el mecanismo en azimut se distribuye, haciéndola soportar menos esfuerzos.



**Figura 4.17** Sketch de los agujeros para unir el servo de azimut a la L.

La pieza en perspectiva isométrica luce de la siguiente manera.

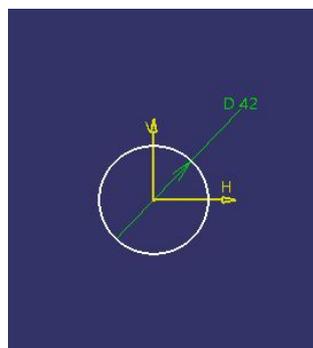


**Figura 4.18** L en perspectiva isométrica.

### 4.3 Pieza de unión entre el servo de elevación y la cabeza

Se pasa a diseñar una pieza de unión entre el servo de elevación y otra que permita contener el Arduino y todos los cables y sensores.

Primero de todo se crea una estructura cilíndrica que permita conectar con las aspas del rotor del servo de elevación. Para ello se hace un sketch con una circunferencia de 42 mm de diámetro.



**Figura 4.19** Dimensiones de la circunferencia.

El grosor del cilindro es de 4 mm, suficiente para aguantar los esfuerzos de flexión a los que será sometida la plataforma adyacente.

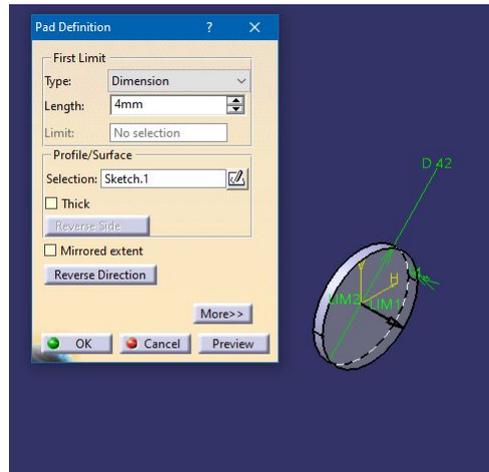


Figura 4.20 Grosor del cilindro.

Para fijar a las aspas del servo de elevación la pieza se crean 4 agujeros uniformemente distribuidos, como los creados para la fijación del servo de azimut con la L.

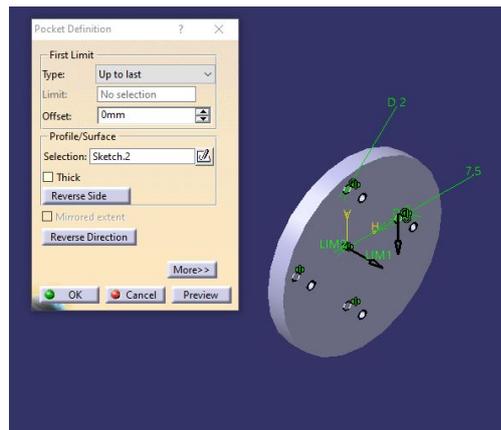


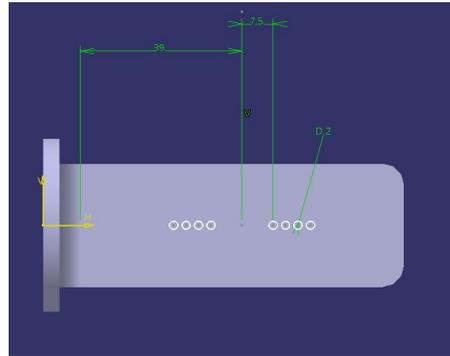
Figura 4.21 Agujeros para la fijación con el servo de elevación.

Posteriormente, se creará una base suficientemente robusta (78 mm por 4 mm de grosor) para soportar el peso de la estructura que contendrá al Arduino, los sensores y toda la circuitería.



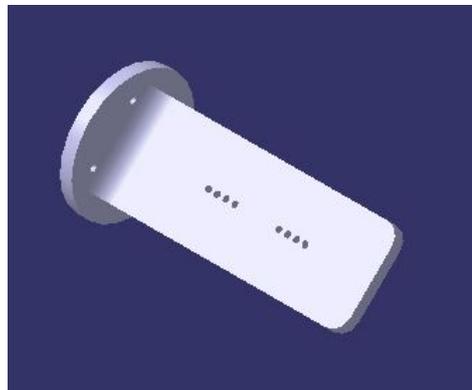
Figura 4.22 Sketch de la base de la cabeza.

Para anclar la cabeza a la base de esta pieza se optan por tornillos 4 tornillos de 2 mm de diámetro uniformemente distribuidos por la base.



**Figura 4.23** Sketch de los agujeros de fijación a la cabeza.

Finalmente, la pieza en perspectiva isométrica luce de la siguiente manera.

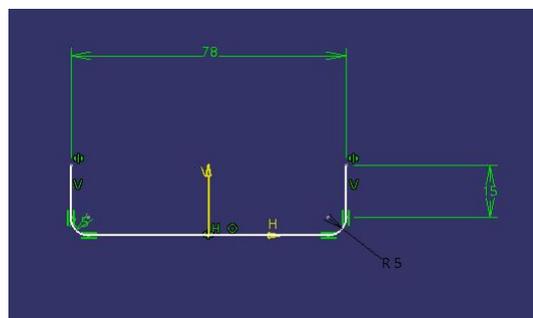


**Figura 4.24** Perspectiva isométrica del enganche del servo de elevación con la cabeza.

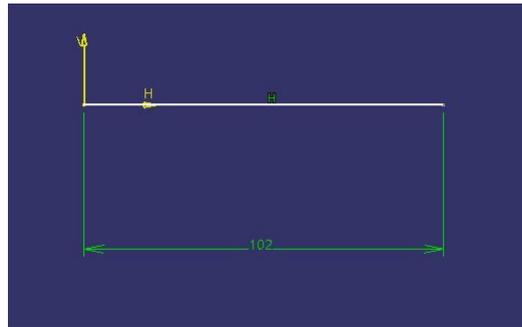
## 4.4 Cabeza inferior

El siguiente paso es el diseño de la plataforma que albergue en su interior el Arduino, los sensores y los cables. Además, se creará de forma que limite la adquisición de energía a la solar directa.

Para hacer la base se utiliza la herramienta "Rib" la cual necesita de dos sketches: uno que determina la forma y su guía.

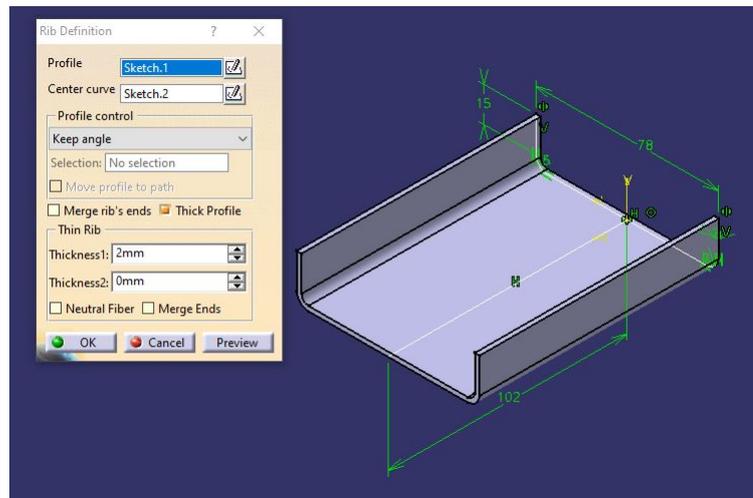


**Figura 4.25** Sketch que determina la forma de la base .



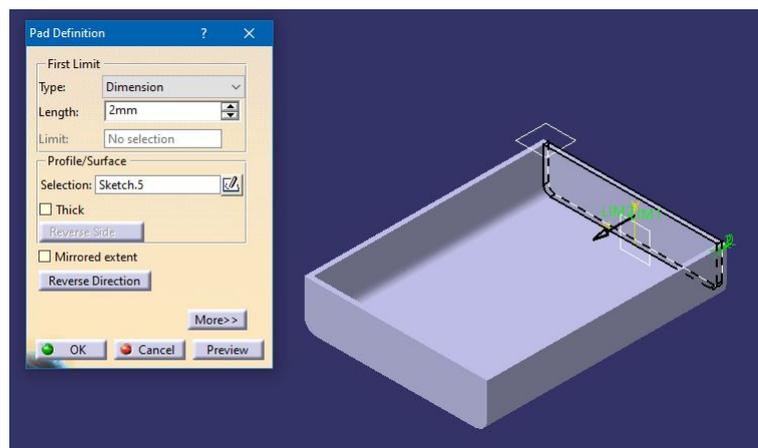
**Figura 4.26** Sketch que sigue la guía que sigue el otro sketch .

Ejecutando la herramienta de CATIA "Rib" y un espesor de 2 mm se obtiene lo siguiente.



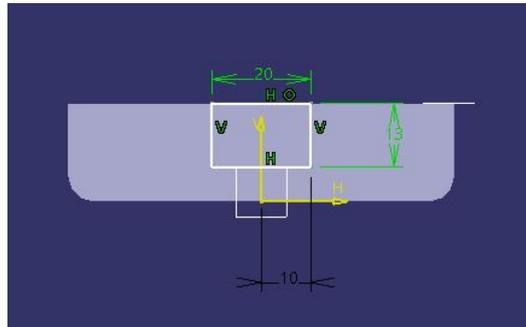
**Figura 4.27** Base primaria de la cabeza inferior.

A continuación, se cierra la estructura por ambos lados con espacio suficiente para poder contener todos los componentes electrónicos del proyecto.



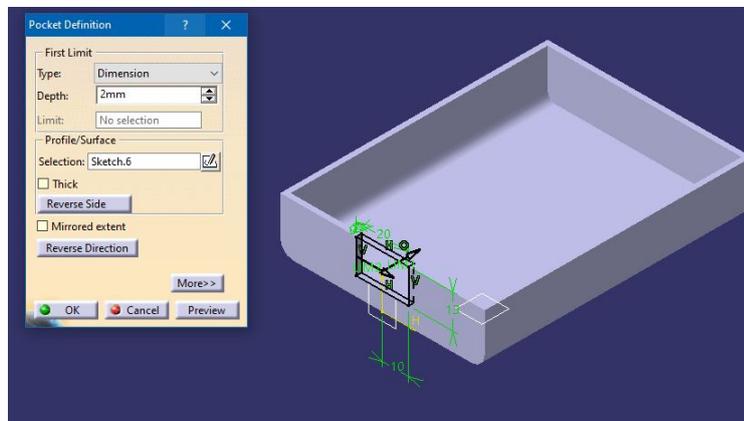
**Figura 4.28** Base primaria cerrada de la cabeza inferior.

Para poder aislar el sensor de radiación solar directa se diseña un agujero a medida para medir la mayor cantidad de radiación solar directa, evitando que otro tipo de radiación altere los resultados. Para ello se crea un sketch de 20 mm por 13 mm en forma rectangular.



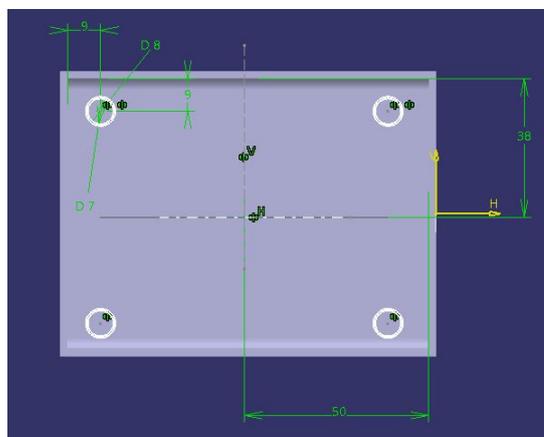
**Figura 4.29** Sketch para el agujero, en la cabeza inferior, del sensor de radiación solar directa.

Ejecutando la herramienta "Pocket" con una profundidad de 2 mm se obtiene la siguiente pieza.



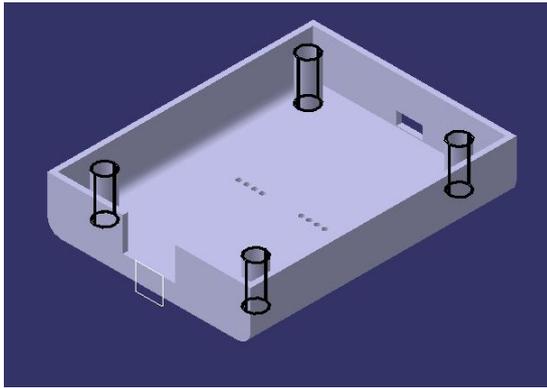
**Figura 4.30** Base primaria de la cabeza inferior con el agujero para el sensor de radiación solar.

Finalmente para conectar la base inferior con la superior se optó por un acoplamiento macho-hembra. Para ello se diseñaron 4 agujeros en la base inferior de la siguiente manera. Se realizó un sketch con 4 coronas circulares, con diámetro exterior 9 mm y diámetro interior 8 mm, uniformemente distribuidas.



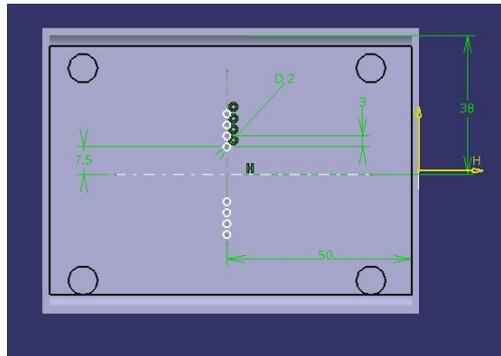
**Figura 4.31** Sketch de coronas circulares para el acoplamiento entre la base inferior y la superior.

Para finalizar se le proporciona un altura de 15 mm.



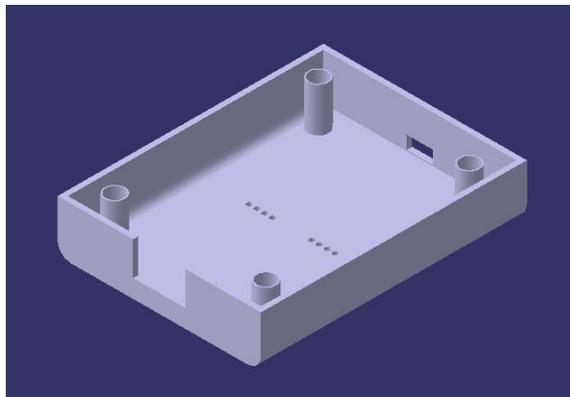
**Figura 4.32** Pieza con las coronas circulares para el acoplamiento entre la base inferior y la base superior.

El último paso para concluir esta pieza es realizar las aberturas que servirán de enlace con el resto de la estructura.



**Figura 4.33** Sketch de los agujeros de conexión..

La cabeza inferior en vista isométrica luce como se puede ver en la imagen presente debajo.



**Figura 4.34** Cabeza inferior en vista isométrica.

## 4.5 Cabeza superior

La última pieza es la complementaria de la cabeza inferior, que consta de una abertura para encajar el sensor y un saliente a modo de filtro de la radiación solar directa.

Mediante la herramienta "Rib" realizamos dos sketches: uno donde se crea la forma de la pieza a propagar y la guía.

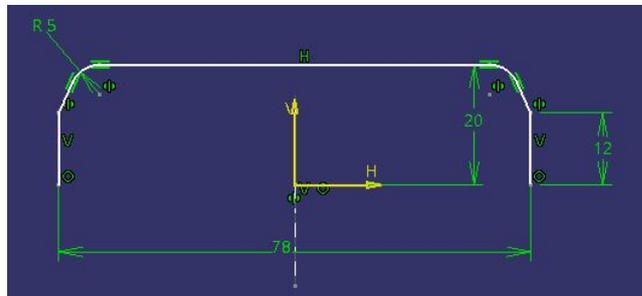


Figura 4.35 Sketch para el "Rib" que propaga la forma.

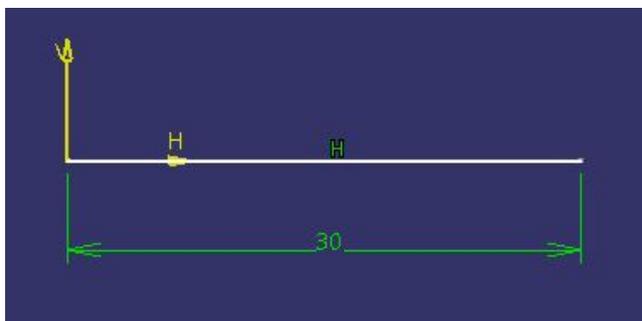


Figura 4.36 Sketch para el "Rib" que conforma la guía.

Ejecutamos el comando "Rib" con un grosor de 2 mm obteniendo lo siguiente.

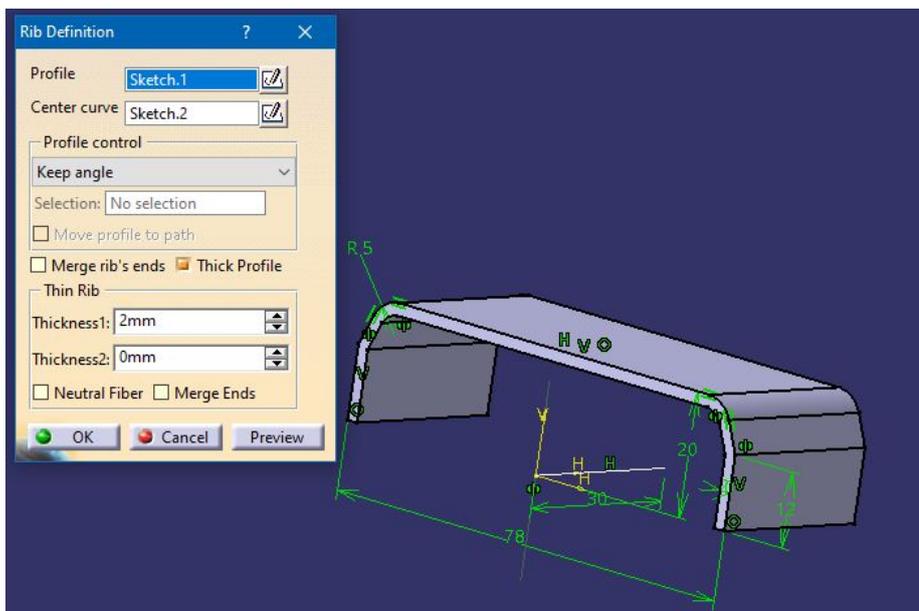


Figura 4.37 Pieza primaria que conforma el parasol.

Para proporcionar una mejor entrada a los rayos solares se optó por perfeccionar el diseño eliminando

material. Se hizo en 2 pasos para los cuales se necesitaron dos "sketchs".

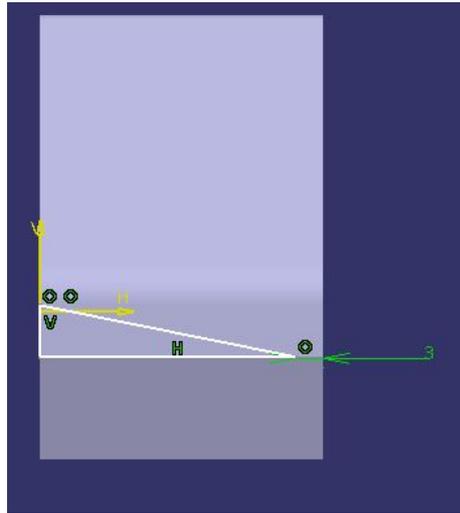


Figura 4.38 Sketch 1 para la eliminación de material.

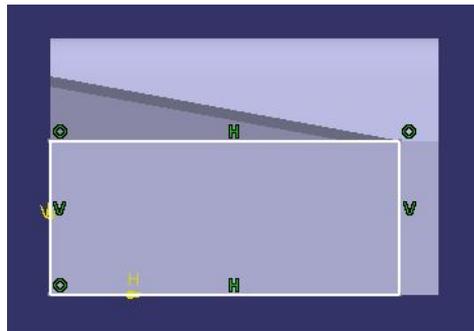


Figura 4.39 Sketch 2 para la eliminación de material.

Ejecutando dos veces el comando "Pocket" el parasol queda de la siguiente forma.

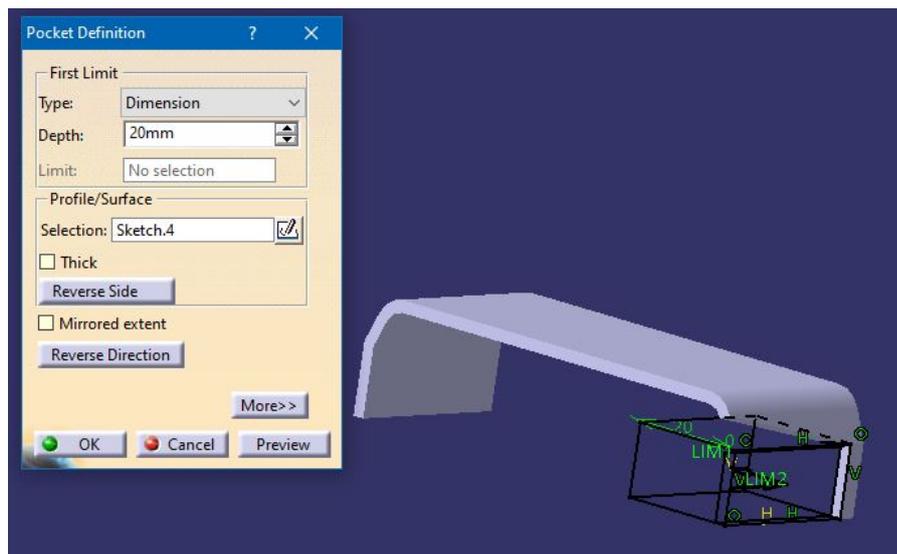


Figura 4.40 Parasol.

A continuación se procede a realizar el techo de la pieza. Para ello, mediante la herramienta "Rib" se realizan los dos sketches necesarios vistos en elementos anteriores.

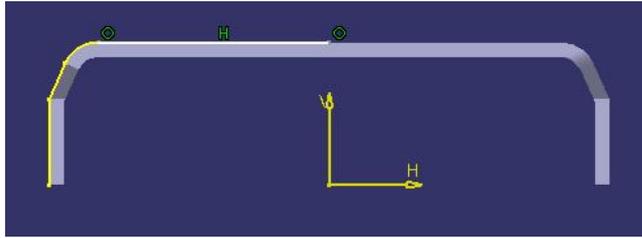


Figura 4.41 Sketch 1 para crear el techo.

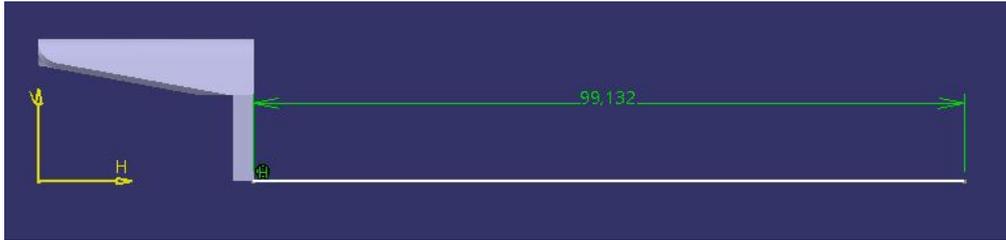


Figura 4.42 Sketch 2 para crear el techo.

Ahora ejecutamos el comando "Rib" con un grosor de 2 mm.

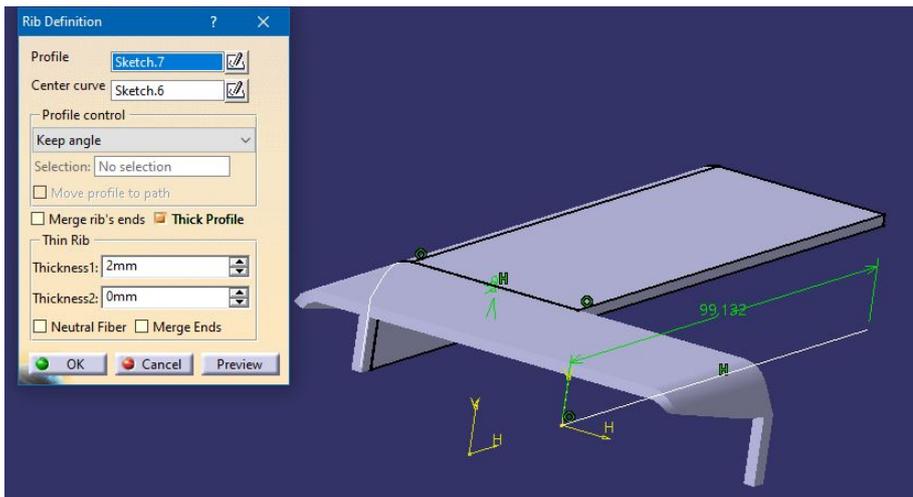


Figura 4.43 Pieza con la adhesión del techo.

El próximo paso es cerrar la estructura mediante dos paredes, que se realizarán mediante el comando "Pad" y la creación de dos sketches que rodeen la silueta de la actual pieza en las posiciones correctas. Finalmente se

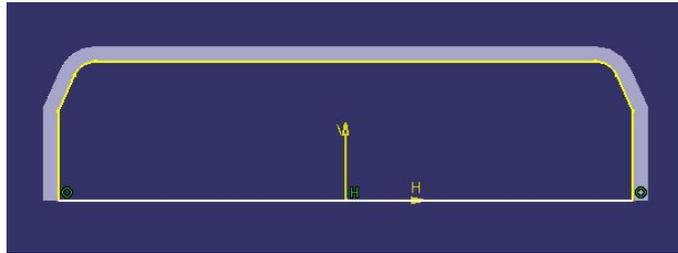


Figura 4.44 Sketchs para la pared trasera y delantera.

ejecuta el comando "Pad" con un grosor de 2 mm obteniendo la siguiente pieza.

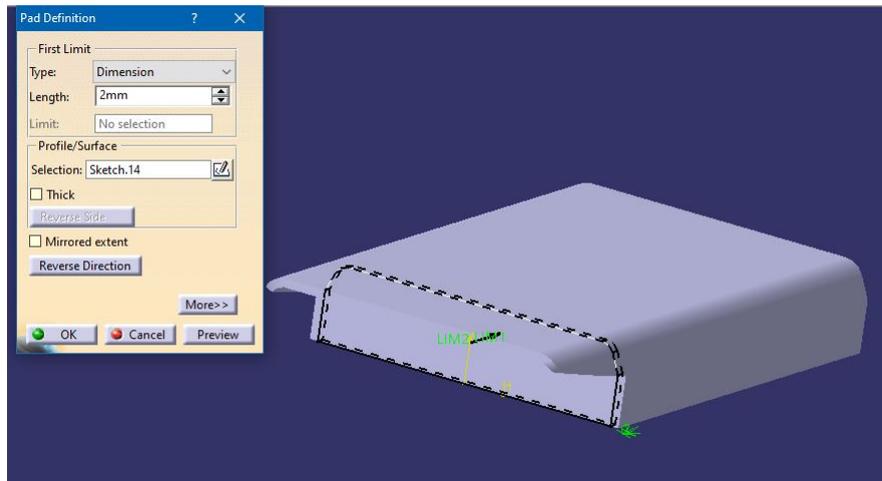


Figura 4.45 Pieza con la creación de las paredes .

Para aislar el sensor, al igual que hicimos en la cabeza inferior, realizaremos un agujero de las dimensiones del mismo. El sketch para tal cometido es el siguiente.

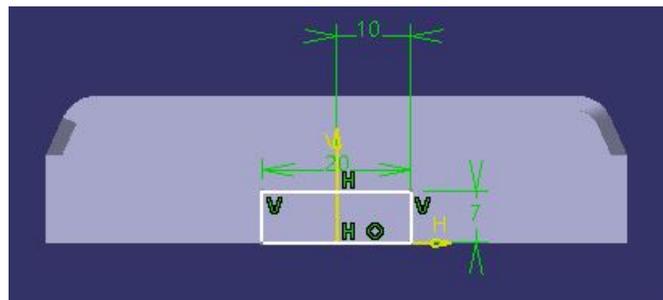
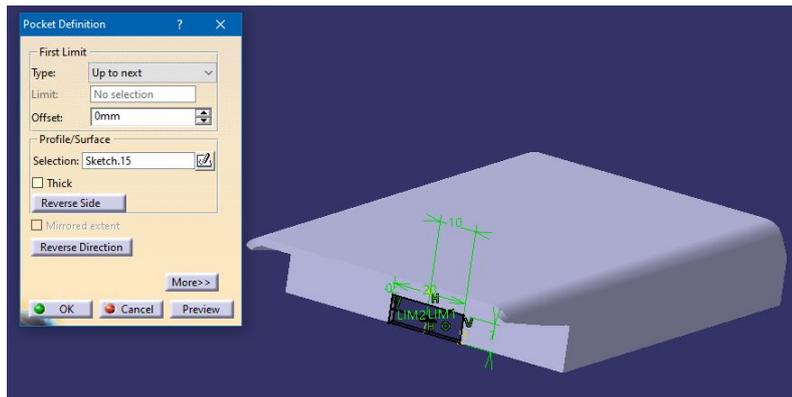


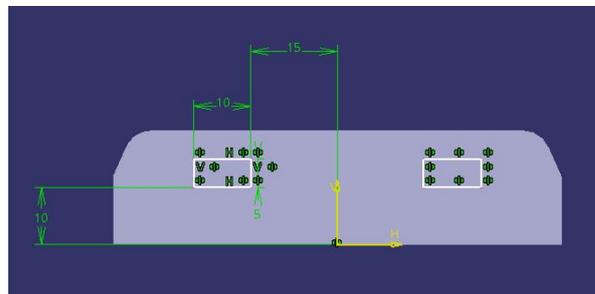
Figura 4.46 Sketch para el agujero del sensor de radiación solar directa.

Ejecutando el comando "Pocket" se obtiene la pieza con el agujero para el sensor de radiación solar directa.



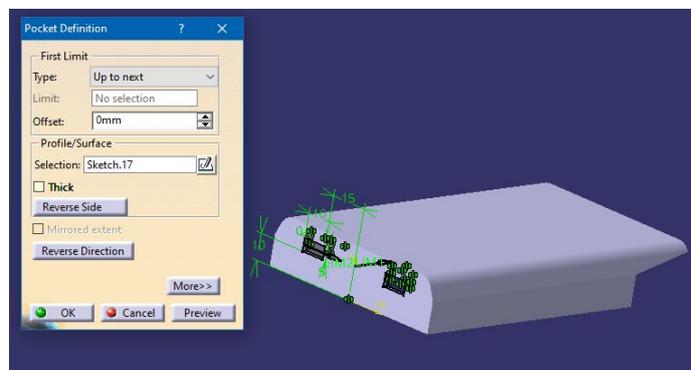
**Figura 4.47** Pieza con el agujero para el sensor.

Además, son necesarios dos agujeros por los que se conectarán los dos servos al Arduino. Para ello el sketch diseñado es el siguiente:



**Figura 4.48** Sketch para los agujeros de los servos.

Ejecutando el comando "Pocket" obtenemos la siguiente pieza.



**Figura 4.49** Cabeza superior con los agujeros para los servos.

Por último, para realizar el acoplamiento macho-hembra descrito en el diseño de la pieza anterior se necesita crear unos salientes en la pieza que encajen. Para ello se realiza un sketch con circunferencias de manera que sean coincidentes con los agujeros de la cabeza inferior pero con un diámetro 1 mm inferior.

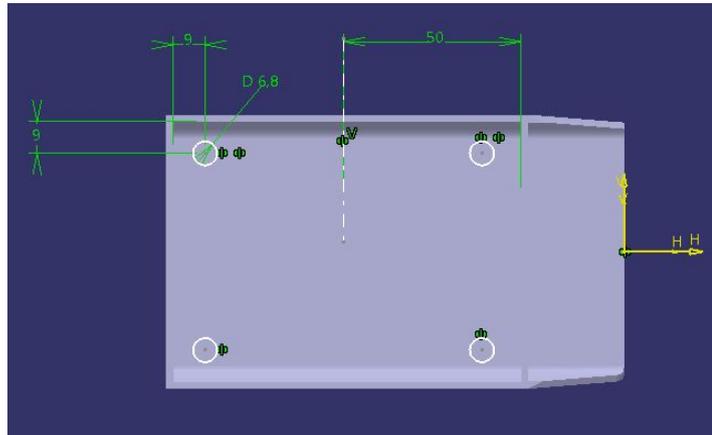


Figura 4.50 Sketch para los salientes del acoplamiento con la cabeza inferior.

Ejecutando la herramienta "Pad" con una longitud de 29 mm obtenemos lo siguiente. Por último, para

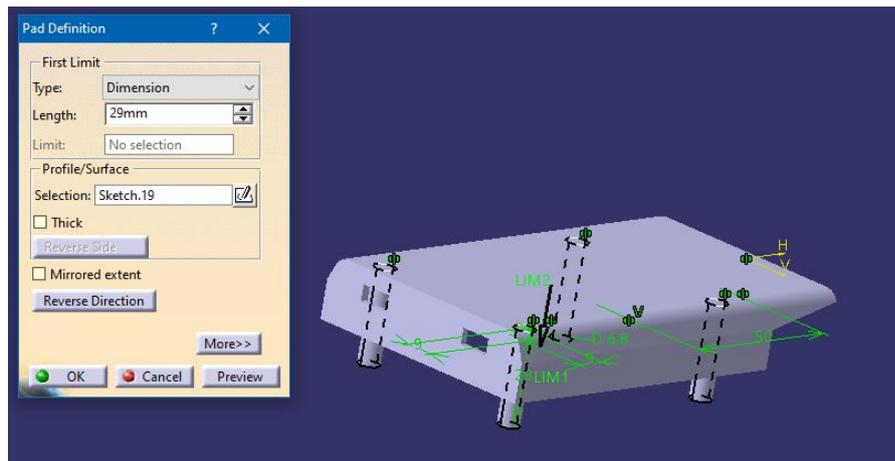


Figura 4.51 Cabeza inferior con los salientes para el acoplamiento.

facilitar el correcto encaje entre piezas se añade un suplemento a los salientes en forma de cono. Para ello se realizan 2 sketches en 3D que guiarán al cono.

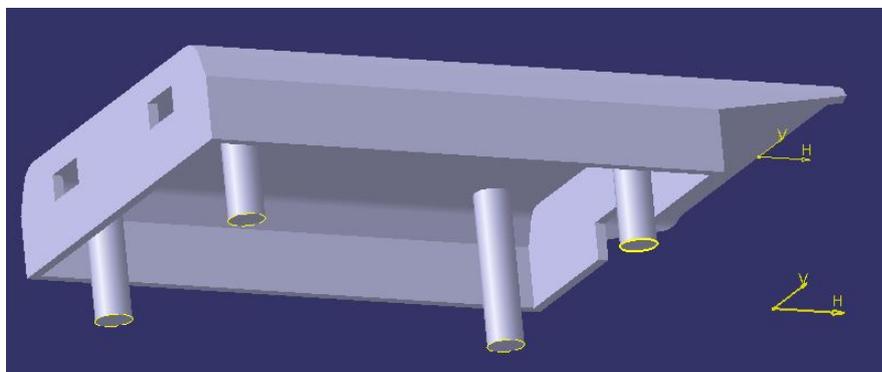
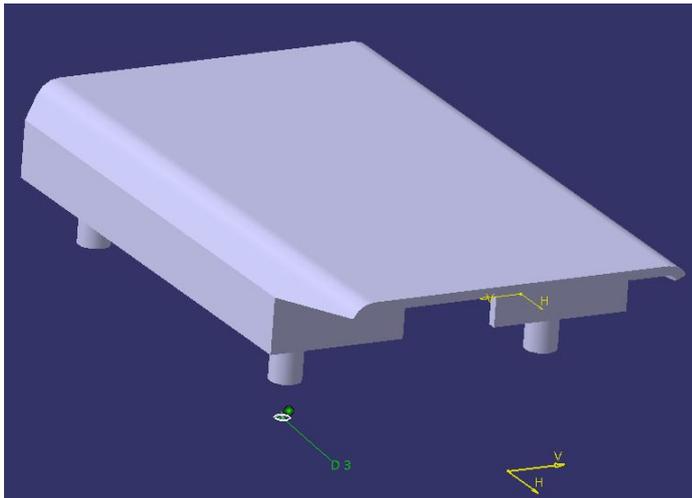
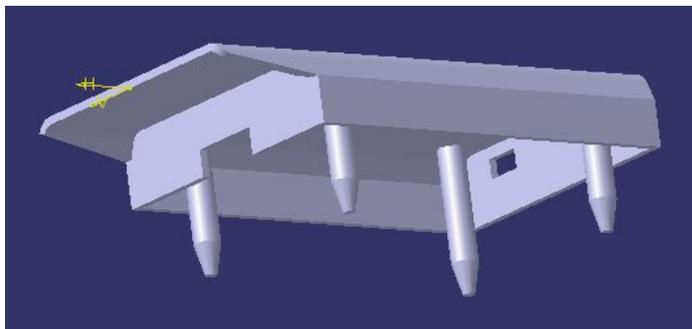


Figura 4.52 Sketch 1 guía del cono.



**Figura 4.53** Sketch 2 guía del cono.

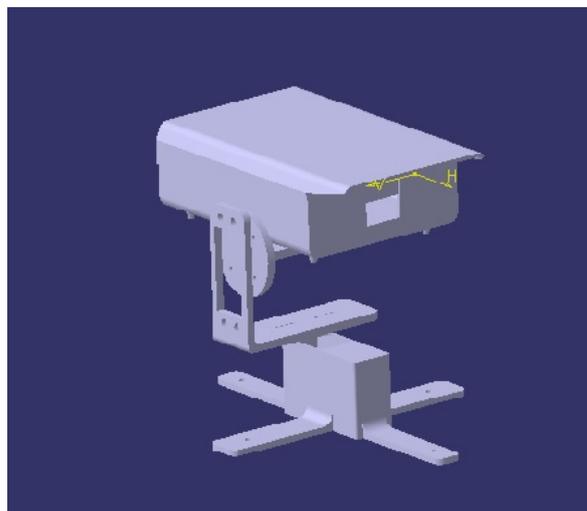
Finalmente la pieza acabada en vista isométrica luce tal que así.



**Figura 4.54** Cabeza superior en vista isométrica.

## 4.6 Ensamblaje

El ensamblaje de todas las piezas se muestra a continuación.



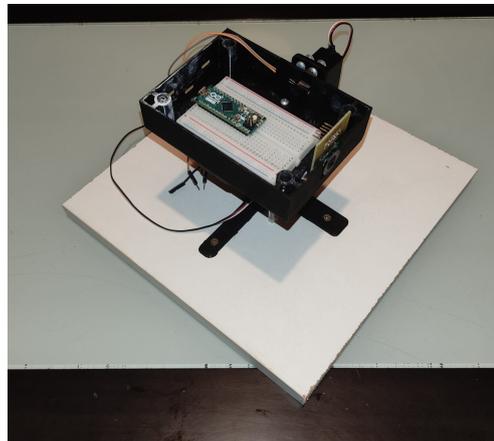
**Figura 4.55** Ensamblaje en vista isométrica.



## 5 Montaje

---

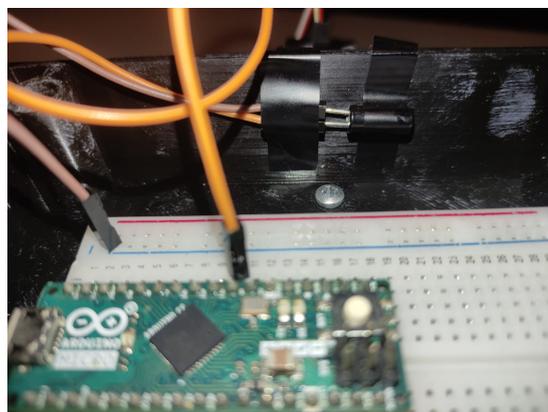
Una vez se tenían las piezas del mecanismo ensambladas se realizó el montaje de la circuitería. Lo primero de todo se pegó una protoboard pequeña a la base de la pieza que llamamos cabeza inferior. Se puede apreciar abajo la imagen de la estructura antes de iniciar el proceso de conexión del material electrónico.



**Figura 5.1** Estructura antes de iniciar el montaje de los sensores y circuitos.

### 5.1 Instalación del inclinómetro SW520D

A continuación se pegó en la base lateral de la estructura que alberga los componentes electrónicos el inclinómetro en la posición más horizontal posible para que se llevara a cabo una primera orientación del sensor lo más precisa posible. Y se conectó al Arduino, con uno de los cables a tierra y el otro al pin digital 4.



**Figura 5.2** Acoplamiento del inclinómetro a la estructura.

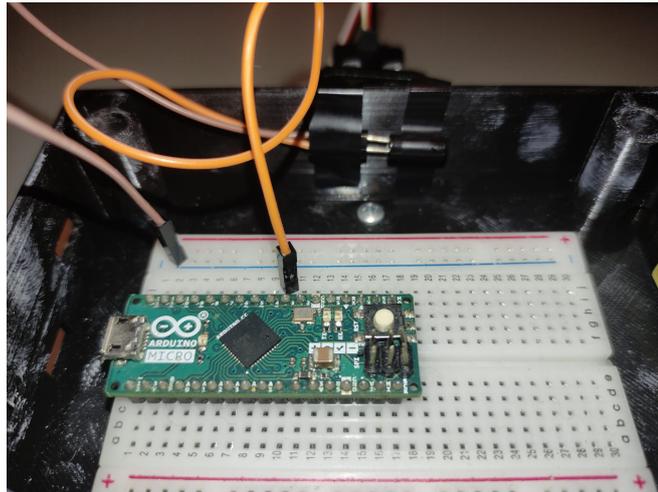


Figura 5.3 Conexión del inclinómetro.

## 5.2 Instalación del magnetómetro GY-271

Después se instala el magnetómetro GY-271 que servirá para orientar en primera aproximación la estructura en azimut. Las conexiones serán GND a tierra, VCC a la corriente de 5V, SCL corresponde en Arduino Micro al pin digital 2 y SDA corresponde al pin digital 3.

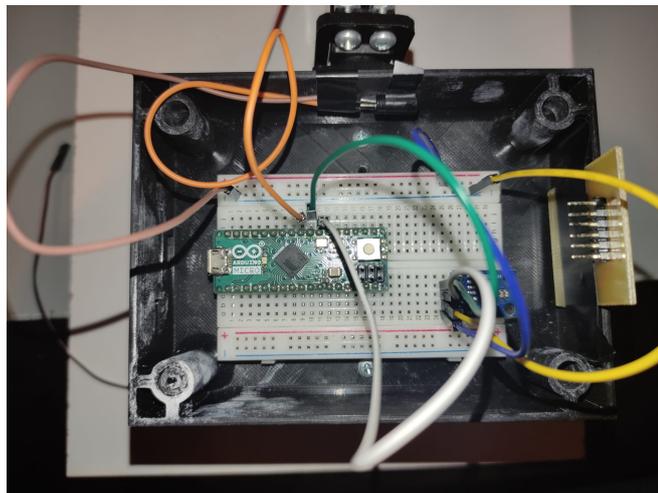


Figura 5.4 Conexión del magnetómetro.

## 5.3 Instalación del sensor de radiación solar

Posteriormente se instala el sensor de radiación solar que consta de 4 salidas analógicas. Por lo tanto, habrá una conexión a tierra, otra a la corriente de 5V y una por cada sensor a una entrada analógica del Arduino (A1,A2,A3,A4).

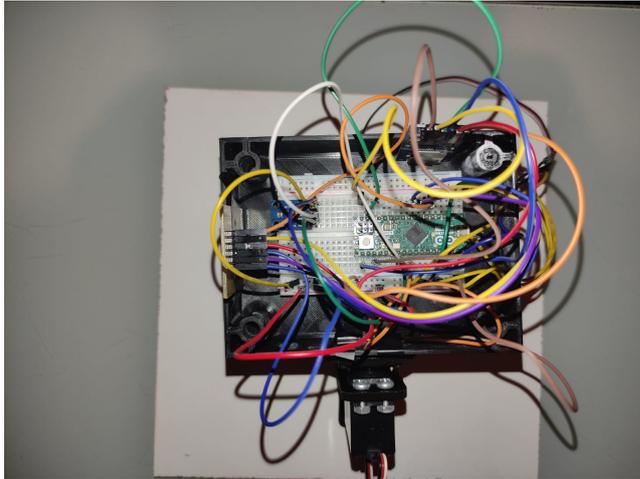


Figura 5.5 Conexión del sensor de radiación.

## 5.4 Instalación de los servos

La estructura tiene dos grados de libertad, por lo que el sistema cuenta con dos servos. Cada servo tendrá tres conexiones: una a tierra, otra a la corriente de 5V y otra a un del Arduino, en este caso D9 y D10.

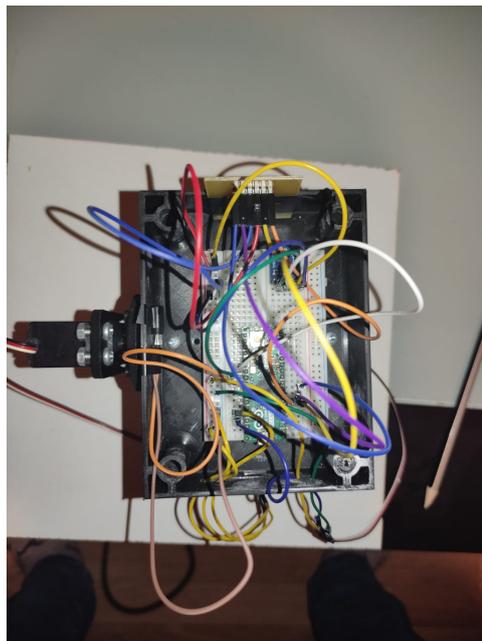
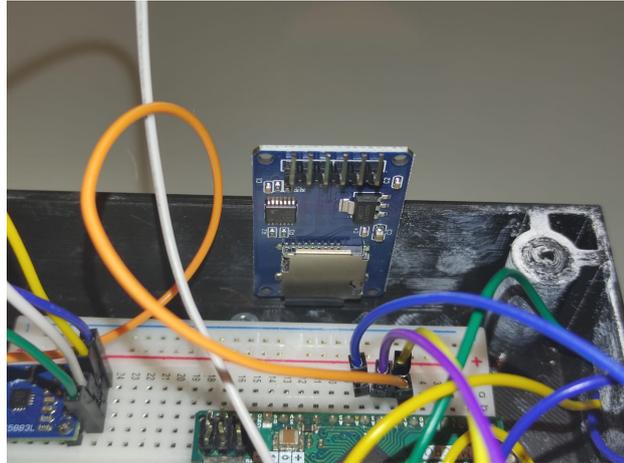


Figura 5.6 Conexión de los servos.

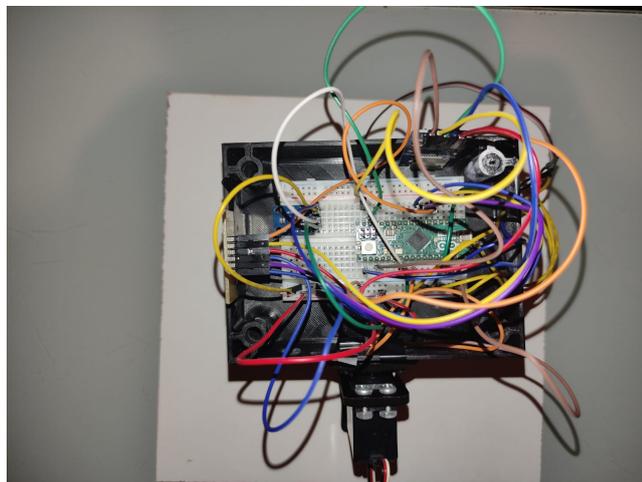
## 5.5 Instalación de la tarjeta SD

La tarjeta de memoria para recoger los datos de medición del sensor tiene 6 conexiones: a tierra, a la corriente de 5 V y a los pines especiales de Arduino CS, MOSI, SCK y MISO. La tarjeta se instala en un lateral del cubículo donde está la electrónica, justo al lado del inclinómetro.



**Figura 5.7** Disposición de la tarjeta SD.

La conexión final de todos los elementos luce tal que así.

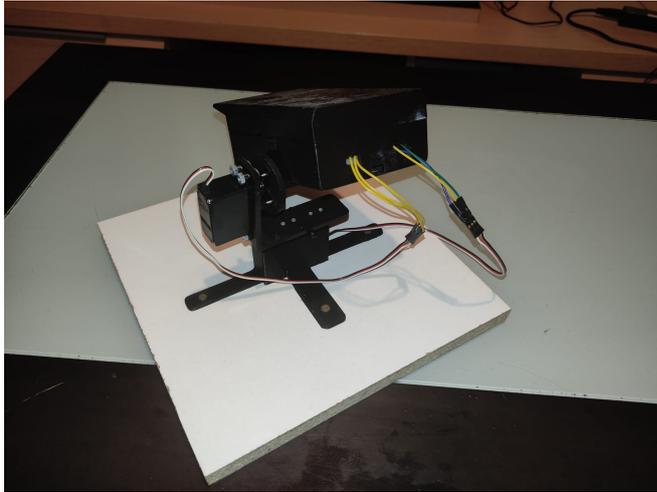


**Figura 5.8** Disposición final de la parte electrónica.

Cerrando la parte superior de la estructura el montaje queda cerrado y el mecanismo listo para la toma de datos.



**Figura 5.9** Mecanismo listo para la parte experimental (Vista frontal).



**Figura 5.10** Mecanismo listo para la parte experimental (Vista trasera).



## 6 Software

---

En este apartado se describe el proceso del diseño de los programas de Arduino que registrarán el movimiento del mecanismo y recogerá los datos necesarios para su posterior análisis.

### 6.1 Código con algoritmo igualando los 4 valores del sensor

Haciendo uso de las Ecuaciones Solares descritas en la Sección 2 se posiciona en primera aproximación el sensor de radiación solar directa que consta de 4 valores de entrada. La idea de este primer código es intentar igualar estos 4 valores, lo que supondría que el sensor está apuntando directamente al Sol, es decir, que los rayos incidentes son totalmente paralelos. Ante la dificultad de lograr esto con precisión y comprobando experimentalmente que es muy difícil conseguir que se igualen, se les aplica una tolerancia en el loop. El código contiene comentarios para el correcto entendimiento del mismo.

```
1  #include <math.h>
2  #include <HMC5883L.h>
3  #include <I2Cdev.h>
4  #include <Servo.h>
5
6
7  HMC5883L compass; //Doy orden de inicializar el procesador de la brújula
8  Servo servo1; //Servo de azimut
9  Servo servo2; //Servo de elevación
10
11 //DATOS A INTRODUCIR PARA CALCULAR ELEVACIÓN Y AZIMUT TEÓRICOS
12 #define pi 3.14159
13 #define GST 4.58 //Greenwich Sidereal Time: Se introduce en función
14   de la hora y día cuándo se empieza a medir
15 #define longitud -0.1047//Longitud dónde está el sensor
16 #define latitud 0.6524
17 #define inclinacion 0.410152
18 #define dn 52//Día juliano necesario para el cálculo de la declinación
19   en la ecuación de Spencer
20 #define t 336.25 //Diferencia de tiempo entre la fecha actual y el
21   Equinoccio de Primavera
22
23 //VARIABLES QUE SE CALCULARÁN DESPUÉS MEDIANTE EL TRIÁNGULO ASTRONÓMICO
24 float AR_sol; //Ascensión recta del Sol que se calculará después
25 float HS; //Ángulo horario (Triángulo astronómico)
26 float h; //Elevación
```

```

24 float declinacion_sol; //Declinación solar
25 float gamma; //ángulo diario necesario para el cálculo de la
    declinación en la ecuación de Spencer
26 float azimut;
27 float u_sol;
28 float seno_az;
29 float coseno_az;
30
31 //VARIABLES MAGNETÓMETRO
32 int16_t x,y,z; //Variables dónde guardaré los valores del magnetó
    metro en los 3 ejes
33 const float declinacion = -37*pi/180; //Corrección del error que da
    el magnetómetro https://www.ign.es/web/ign/portal/gmt-declinacion-
    magnetica
34
35 //VARIABLES SERVO DE AZIMUT
36 int PINSERVO=5;
37 int MIN = 1316;
38 int MAX = 1716;
39 float w=0.5035; //Velocidad angular para servo.write(91) contrario
    agujas reloj
40 float t0,t1;
41
42 //VARIABLES SERVO DE ELEVACIÓN
43 int PINSERVO2=6;
44 float t2,t3;
45
46 //VARIABLES DEL INCLINÓMETRO SW520D
47 int pin_inc=2;
48
49 //VARIABLES SENSOR NANO-ISS60
50 int sensor1=A0;
51 int sensor2=A1;
52 int sensor3=A2;
53 int sensor4=A3;
54
55 void setup(){
56   Serial.begin(9600);
57   delay(5000);
58
59   //OBTENER COMPONENTES CAMPO MAGNÉTICO
60   Wire.begin();
61   compass.initialize();
62   compass.getHeading(&x, &y, &z);
63   //Calcular ángulo el ángulo del eje X respecto al norte
64   float azimut_real = atan2(y,x);
65   azimut_real = azimut_real - declinacion; //En radianes
66   if(azimut_real < 0) {
67     azimut_real = azimut_real + 2*pi;
68   }
69
70   //CÁLCULO ASCENSIÓN RECTA DEL SOL
71   u_sol=(t/365.25)*(2*pi);
72   AR_sol=atan(cos(inclinacion)*tan(u_sol));
73   if(u_sol > pi/2 && u_sol<(3*pi/2)){
74     AR_sol=AR_sol + pi;
75   }

```

```

76 HS=GST+longitud-AR_sol;
77 //CÁLCULO DE LA DECLINACIÓN SOLAR (ECUACIÓN DE SPENCER)
78 gamma=2*pi*(dn-1)/365;
79 declinacion_sol=0.0006918-0.399912*cos(gamma)+0.070257*sin(gamma)
      -0.006758*cos(2*gamma)+0.000907*sin(2*gamma)-0.002697*cos(3*gamma)
      +0.00148*sin(3*gamma);
80
81 //CÁLCULO ELEVACIÓN
82 h=asin(sin(latitud)*sin(declinacion_sol)+cos(latitud)*cos(
      declinacion_sol)*cos(HS));
83 Serial.print(h);
84 if(h>(pi/2) || h<(-pi/2)){
85     h=pi-h;
86 }
87 //CÁLCULO AZIMUT
88 coseno_az=(sin(declinacion_sol)-sin(latitud)*sin(h))/(cos(latitud)*
      cos(h));
89 seno_az=-sin(HS)*cos(declinacion_sol)/cos(h);
90 if(coseno_az >= 0 && seno_az>=0){
91     azimut=acos(sin(declinacion_sol)-sin(latitud)*sin(h))/(cos(latitud)
      *cos(h));
92 }
93 if(coseno_az<0 && seno_az>=0){
94     azimut=acos(sin(declinacion_sol)-sin(latitud)*sin(h))/(cos(latitud)
      *cos(h));
95 }
96 }
97 if(coseno_az<0 && seno_az<0){
98     azimut=2*pi-acos(sin(declinacion_sol)-sin(latitud)*sin(h))/(cos(
      latitud)*cos(h));
99 }
100 }
101 if(coseno_az>0 && seno_az<0){
102     azimut=asin(-sin(HS)*cos(declinacion_sol)/cos(h));
103 }
104
105 //POSICIONO SISTEMA EN AZIMUT
106 servo1.attach(PINSERVO,MIN,MAX); //Esta función otorga el pin donde
      se conecta el servo y los valores mínimo y máximo del pulso
      asociados a 0° y 180°.
107 servo1.write(90);
108 t0=millis();
109 if(azimut>azimut_real){ //Si el azimut que quiero es mayor que el
      real tendré que girar en sentido de las agujas del reloj
110     servo1.write(89); //El número indica los grados requeridos
111     t1=0;
112     while((t1-t0)<((azimut-azimut_real)/w)*1000){
113         t1=millis();
114     }
115     servo1.write(90);
116 }
117 else{ //Si el azimut real es mayor que el quiero tendré que girar en
      sentido contrario a las agujas del reloj
118     servo1.write(91); //El número indica los grados requeridos
119     t1=0;
120     while((t1-t0)<((azimut_real-azimut)/w)*1000){
121         t1=millis();

```

```
122     }
123     servo1.write(90);
124
125 }
126
127 //POSICIONO SISTEMA EN ELEVACIÓN
128 pinMode(pin_inc, INPUT);
129 servo2.attach(PINSERVO2, MIN, MAX);
130 servo2.write(90);
131 digitalWrite(pin_inc, HIGH); //activamos la resistencia interna PULL
    UP
132 t3=0;
133 while(digitalRead(pin_inc)==HIGH){
134     servo2.write(91); //Colocamos el dispositivo en elevación 0
135 }
136 servo2.write(90);
137 t2=millis();
138 while((t2-t3)<(h/w)*1000){
139     servo2.write(89); //Colocamos el dispositivo con la elevación teórica
140     t3=millis();
141 }
142 servo2.write(90);
143 }
144
145
146
147 void loop(){
148
149     //CONTROL ALGORITMO 4 ENTRADAS EN AZIMUT
150     if(analogRead(sensor1)>=analogRead(sensor4)){
151         while(analogRead(sensor1)>analogRead(sensor4)){
152             servo1.write(91);
153         }
154     }
155     else{
156         while(analogRead(sensor1)<analogRead(sensor4)){
157             servo1.write(89);
158         }
159     }
160     servo1.write(90);
161
162     //CONTROL ALGORITMO 4 ENTRADAS EN ELEVACIÓN
163
164     if(analogRead(sensor1)>=analogRead(sensor2)){
165         while(analogRead(sensor1)>analogRead(sensor2)){
166             servo2.write(89);
167         }
168     }
169     else{
170         while(analogRead(sensor1)<analogRead(sensor2)){
171             servo2.write(91);
172         }
173     }
174     servo2.write(90);
175 }
```

## 6.2 Código con algoritmo octogonal

Para realizar este experimento el sistema se mueve 8 veces. Si cogemos como sistema de referencia el plano donde está el sensor mueve dos grados en vertical y horizontal, hacia arriba y hacia abajo y cubriendo las diagonales, lo que hace un total de 8 movimientos en los cuáles se recogen 32 datos.

Estos datos se ordenan mediante un algoritmo de ordenación y se consigue la posición del máximo, adonde llevamos nuestro sensor. Esta operación se realizará cada minuto.

```

1  #include <math.h>
2  #include <HMC5883L.h>
3  #include <I2Cdev.h>
4  #include <Servo.h>
5
6
7  HMC5883L compass; //Doy orden de inicializar el procesador de la brújula
8  Servo servo1; //Servo de azimut
9  Servo servo2; //Servo de elevación
10
11 //DATOS A INTRODUCIR PARA CALCULAR ELEVACIÓN Y AZIMUT TEÓRICOS
12 #define pi 3.14159
13 #define GST 4.58 //Greenwich Sidereal Time: Se introduce en función
14   de la hora y día cuándo se empieza a medir
15 #define longitud -0.1047//Longitud dónde está el sensor
16 #define latitud 0.6524
17 #define inclinacion 0.410152
18 #define dn 52//Día juliano necesario para el cálculo de la declinación
19   en la ecuación de Spencer
20 #define t 336.25 //Diferencia de tiempo entre la fecha actual y el
21   Equinoccio de Primavera
22
23 //VARIABLES QUE SE CALCULARÁN DESPUÉS MEDIANTE EL TRIÁNGULO ASTRONÓMICO
24 float AR_sol; //Ascensión recta del Sol que se calculará después
25 float HS; //Ángulo horario (Triángulo astronómico)
26 float h; //Elevación
27 float declinacion_sol; //Declinación solar
28 float gamma; //ángulo diario necesario para el cálculo de la
29   declinación en la ecuación de Spencer
30 float azimut;
31 float u_sol;
32 float seno_az;
33 float coseno_az;
34
35 //VARIABLES MAGNETÓMETRO
36 int16_t x,y,z; //Variables dónde guardaré los valores del magnetómetro
37   en los 3 ejes
38 const float declinacion = -37*pi/180; //Corrección del error que da
39   el magnetómetro https://www.ign.es/web/ign/portal/gmt-declinacion-magnetica
40
41 //VARIABLES SERVO DE AZIMUT
42 int PINSERVO=5;
43 int MIN = 1316;
44 int MAX = 1716;

```

```

39 float w=0.5035; //Velocidad angular para servo.write(91) contrario
    agujas reloj
40 float t0,t1;
41
42 //VARIABLES SERVO DE ELEVACIÓN
43 int PINSERVO2=6;
44 float t2,t3;
45
46 //VARIABLES DEL INCLINÓMETRO SW520D
47 int pin_inc=2;
48
49 //VARIABLES SENSOR NANO-ISS60
50 int sensor1=A0;
51 int sensor2=A1;
52 int sensor3=A2;
53 int sensor4=A3;
54
55 //VARIABLES PARA EL ALGORITMO OCTOGONAL
56 int v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v16,v17,v18,v19,
    v20,v21,v22,v23,v24,v25,v26,v27,v28,v29,v30,v31,v32
57 int data[32]
58
59 //VARIABLES PARA ALGORITMO DE ORDENACIÓN
60 int k, j, flag = 1;
61 int temp;
62 int posicion;
63
64 void setup() {
65   Serial.begin(9600);
66   delay(5000);
67
68   //OBTENER COMPONENTES CAMPO MAGNÉTICO
69   Wire.begin();
70   compass.initialize();
71   compass.getHeading(&x, &y, &z);
72   //Calcular ángulo el ángulo del eje X respecto al norte
73   float azimut_real = atan2(y,x);
74   azimut_real = azimut_real - declinacion; //En radianes
75   if(azimut_real < 0) {
76     azimut_real = azimut_real + 2*pi;
77   }
78
79   //CÁLCULO ASCENSIÓN RECTA DEL SOL
80   u_sol=(t/365.25)*(2*pi);
81   AR_sol=atan(cos(inclinacion)*tan(u_sol));
82   if(u_sol > pi/2 && u_sol<(3*pi/2)){
83     AR_sol=AR_sol + pi;
84   }
85   HS=GST+longitud-AR_sol;
86   //CÁLCULO DE LA DECLINACIÓN SOLAR (ECUACIÓN DE SPENCER)
87   gamma=2*pi*(dn-1)/365;
88   declinacion_sol=0.0006918-0.399912*cos(gamma)+0.070257*sin(gamma)
    -0.006758*cos(2*gamma)+0.000907*sin(2*gamma)-0.002697*cos(3*gamma)
    +0.00148*sin(3*gamma);
89
90 //CÁLCULO ELEVACIÓN

```

```

91  h=asin(sin(latitud)*sin(declinacion_sol)+cos(latitud)*cos(
      declinacion_sol)*cos(HS));
92  Serial.print(h);
93  if(h>(pi/2) || h<(-pi/2)){
94      h=pi-h;
95  }
96  //CÁLCULO AZIMUT
97  coseno_az=(sin(declinacion_sol)-sin(latitud)*sin(h))/(cos(latitud)*
      cos(h));
98  seno_az=-sin(HS)*cos(declinacion_sol)/cos(h);
99  if(coseno_az >= 0 && seno_az>=0){
100     azimut=acos(sin(declinacion_sol)-sin(latitud)*sin(h))/(cos(latitud)
      *cos(h));
101 }
102 if(coseno_az<0 && seno_az>=0){
103     azimut=acos(sin(declinacion_sol)-sin(latitud)*sin(h))/(cos(latitud)
      *cos(h));
104 }
105 }
106 if(coseno_az<0 && seno_az<0){
107     azimut=2*pi-acos(sin(declinacion_sol)-sin(latitud)*sin(h))/(cos(
      latitud)*cos(h));
108 }
109 }
110 if(coseno_az>0 && seno_az<0){
111     azimut=asin(-sin(HS)*cos(declinacion_sol)/cos(h));
112 }
113
114 //POSICIONO SISTEMA EN AZIMUT
115 servo1.attach(PINSERVO,MIN,MAX); //Esta función otorga el pin donde
      se conecta el servo y los valores mínimo y máximo del pulso
      asociados a 0° y 180°.
116 servo1.write(90);
117 t0=millis();
118 if(azimut>azimut_real){ //Si el azimut que quiero es mayor que el
      real tendré que girar en sentido de las agujas del reloj
119     servo1.write(89); //El número indica los grados requeridos
120     t1=0;
121     while((t1-t0)<((azimut-azimut_real)/w)*1000){
122         t1=millis();
123     }
124     servo1.write(90);
125 }
126 else{ //Si el azimut real es mayor que el quiero tendré que girar en
      sentido contrario a las agujas del reloj
127     servo1.write(91); //El número indica los grados requeridos
128     t1=0;
129     while((t1-t0)<((azimut_real-azimut)/w)*1000){
130         t1=millis();
131     }
132     servo1.write(90);
133 }
134 }
135
136 //POSICIONO SISTEMA EN ELEVACIÓN
137 pinMode(pin_inc, INPUT);
138 servo2.attach(PINSERVO2,MIN,MAX);

```

```
139 servo2.write(90);
140 digitalWrite(pin_inc , HIGH); //activamos la resistencia interna PULL
    UP
141 t3=0;
142 while(digitalRead(pin_inc)==HIGH){
143   servo2.write(91); //Colocamos el dispositivo en elevación 0
144 }
145 servo2.write(90);
146 t2=millis();
147 while((t2-t3)<(h/w)*1000){
148   servo2.write(89); //Colocamos el dispositivo con la elevación teórica
149   t3=millis();
150 }
151 servo2.write(90);
152 }
153
154 void loop(){
155
156   //CONTROL ALGORITMO OCTOGONAL
157   //Recogemos los valores en cada una de las 8 posiciones y los
    guardamos en el vector
158   for(int i=1;i<4;i++){
159     servo1.write(89);
160     servo2.write(89);
161     data(i)=analogRead(sensor1)
162     data(i+1)=analogRead(sensor2)
163     data(i+2)=analogRead(sensor3)
164     data(i+3)=analogRead(sensor4)
165     servo1.write(90);
166     servo2.write(90);
167   }
168
169   for(int i;i<8;i++){
170     servo1.write(92);
171     servo2.write(88);
172     data(i)=analogRead(sensor1)
173     data(i+1)=analogRead(sensor2)
174     data(i+2)=analogRead(sensor3)
175     data(i+3)=analogRead(sensor4)
176     servo1.write(90);
177     servo2.write(90);
178   }
179   for(int i;i<12;i++){
180     servo1.write(88);
181     servo2.write(92);
182     data(i)=analogRead(sensor1)
183     data(i+1)=analogRead(sensor2)
184     data(i+2)=analogRead(sensor3)
185     data(i+3)=analogRead(sensor4)
186     servo1.write(90);
187     servo2.write(90);
188   }
189   for(int i;i<12;i++){
190     servo1.write(92);
191     servo2.write(92);
192     data(i)=analogRead(sensor1)
193     data(i+1)=analogRead(sensor2)
```

```
194 data(i+2)=analogRead(sensor3)
195 data(i+3)=analogRead(sensor4)
196 servo1.write(90);
197 servo2.write(90);
198 }
199 for(int i;i<16;i++){
200 servo1.write(91);
201 servo2.write(92);
202 data(i)=analogRead(sensor1)
203 data(i+1)=analogRead(sensor2)
204 data(i+2)=analogRead(sensor3)
205 data(i+3)=analogRead(sensor4)
206 servo1.write(90);
207 servo2.write(90);
208 }
209 for(int i;i<20;i++){
210 servo1.write(92);
211 servo2.write(91);
212 data(i)=analogRead(sensor1)
213 data(i+1)=analogRead(sensor2)
214 data(i+2)=analogRead(sensor3)
215 data(i+3)=analogRead(sensor4)
216 servo1.write(90);
217 servo2.write(90);
218 }
219 for(int i;i<24;i++){
220 servo1.write(89);
221 servo2.write(92);
222 data(i)=analogRead(sensor1)
223 data(i+1)=analogRead(sensor2)
224 data(i+2)=analogRead(sensor3)
225 data(i+3)=analogRead(sensor4)
226 servo1.write(90);
227 servo2.write(90);
228 }
229 for(int i;i<28;i++){
230 servo1.write(92);
231 servo2.write(89);
232 data(i)=analogRead(sensor1)
233 data(i+1)=analogRead(sensor2)
234 data(i+2)=analogRead(sensor3)
235 data(i+3)=analogRead(sensor4)
236 servo1.write(90);
237 servo2.write(90);
238 }
239 for(int i;i<12;i++){
240 servo1.write(89);
241 servo2.write(91);
242 data(i)=analogRead(sensor1)
243 data(i+1)=analogRead(sensor2)
244 data(i+2)=analogRead(sensor3)
245 data(i+3)=analogRead(sensor4)
246 servo1.write(90);
247 servo2.write(90);
248 }
249
```

```
250 //Algoritmo Bubble Sort de ordenación de un vector de mayor a menor.
    GUARDA LA POSICIÓN DEL VALOR NO EL VALOR EN SÍ
251 for (k = 1; (k <= length) && flag; i++)
252 {
253     flag = 0;
254     for (j = 0; j < (length - 1); j++)
255     {
256         if (data[j + 1] > data[j])
257         {
258             temp = data[j];
259             data[j] = data[j + 1];
260             data[j + 1] = temp;
261             posicion=j;
262             flag = 1;
263         }
264     }
265 }
266 //En función de la posición llevamos el mecanismo al lugar donde el
    valor de entrada es mayor
267 if(posicion<=4){
268     serv1.write(89);
269     servo2.write(89);
270 }
271 else if(posicion>4 && posicion <8){
272     serv1.write(92);
273     servo2.write(88);
274 }
275 else if(posicion>8 && posicion <12){
276     serv1.write(88);
277     servo2.write(92);
278 }
279 else if(posicion>12 && posicion <16){
280     serv1.write(92);
281     servo2.write(92);
282 }
283 else if(posicion>16 && posicion <20){
284     serv1.write(91);
285     servo2.write(92);
286 }
287 else if(posicion>20 && posicion <24){
288     serv1.write(92);
289     servo2.write(91);
290 }
291 else if(posicion>24 && posicion <28){
292     serv1.write(92);
293     servo2.write(89);
294 }
295 else if(posicion>28 && posicion <32){
296     serv1.write(89);
297     servo2.write(91);
298 }
299 //Esperar un minutos
300 delay(60000)
301 }
```

## 6.3 Algoritmo en espiral

A continuación se desarrolla el código basado en un algoritmo en espiral, en el cual se recogerán 24 datos en una disposición en espiral y el mecanismo se moverá en consecuencia para lograr la posición del máximo.

```

1  #include <math.h>
2  #include <HMC5883L.h>
3  #include <I2Cdev.h>
4  #include <Servo.h>
5
6
7  HMC5883L compass; //Doy orden de inicializar el procesador de la brújula
8  Servo servo1; //Servo de azimut
9  Servo servo2; //Servo de elevación
10
11 //DATOS A INTRODUCIR PARA CALCULAR ELEVACIÓN Y AZIMUT TEÓRICOS
12 #define pi 3.14159
13 #define GST 4.58 //Greenwich Sidereal Time: Se introduce en función
    de la hora y día cuándo se empieza a medir
14 #define longitud -0.1047//Longitud dónde está el sensor
15 #define latitud 0.6524
16 #define inclinacion 0.410152
17 #define dn 52//Día juliano necesario para el cálculo de la declinación
    en la ecuación de Spencer
18 #define t 336.25 //Diferencia de tiempo entre la fecha actual y el
    Equinoccio de Primavera
19
20 //VARIABLES QUE SE CALCULARÁN DESPUÉS MEDIANTE EL TRIÁNGULO ASTRONÓMICO
21 float AR_sol; //Ascensión recta del Sol que se calculará después
22 float HS; //Ángulo horario (Triángulo astronómico)
23 float h; //Elevación
24 float declinacion_sol; //Declinación solar
25 float gamma; //ángulo diario necesario para el cálculo de la
    declinación en la ecuación de Spencer
26 float azimut;
27 float u_sol;
28 float seno_az;
29 float coseno_az;
30
31 //VARIABLES MAGNETÓMETRO
32 int16_t x,y,z; //Variables dónde guardaré los valores del magnetómetro
    en los 3 ejes
33 const float declinacion = -37*pi/180; //Corrección del error que da
    el magnetómetro https://www.ign.es/web/ign/portal/gmt-declinacion-magnetica
34
35 //VARIABLES SERVO DE AZIMUT
36 int PINSERVO=5;
37 int MIN = 1316;
38 int MAX = 1716;
39 float w=0.5035; //Velocidad angular para servo.write(91) contrario
    agujas reloj
40 float t0,t1;
41

```

```

42 //VARIABLES SERVO DE ELEVACIÓN
43 int PINSERVO2=6;
44 float t2,t3;
45
46 //VARIABLES DEL INCLINÓMETRO SW520D
47 int pin_inc=2;
48
49 //VARIABLES SENSOR NANO-ISS60
50 int sensor1=A0;
51 int sensor2=A1;
52 int sensor3=A2;
53 int sensor4=A3;
54
55 //VARIABLES PARA EL ALGORITMO OCTOGONAL
56 int v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v16,v17,v18,v19,
    v20
57 int data[20]
58
59 //VARIABLES PARA ALGORITMO DE ORDENACIÓN
60 int k, j, flag = 1;
61 int temp;
62 int posicion;
63
64 void setup() {
65   Serial.begin(9600);
66   delay(5000);
67
68   //OBTENER COMPONENTES CAMPO MAGNÉTICO
69   Wire.begin();
70   compass.initialize();
71   compass.getHeading(&x, &y, &z);
72   //Calcular ángulo el ángulo del eje X respecto al norte
73   float azimut_real = atan2(y,x);
74   azimut_real = azimut_real - declinacion; //En radianes
75   if(azimut_real < 0) {
76     azimut_real = azimut_real + 2*pi;
77   }
78
79   //CÁLCULO ASCENSIÓN RECTA DEL SOL
80   u_sol=(t/365.25)*(2*pi);
81   AR_sol=atan(cos(inclinacion)*tan(u_sol));
82   if(u_sol > pi/2 && u_sol<(3*pi/2)){
83     AR_sol=AR_sol + pi;
84   }
85   HS=GST+longitud-AR_sol;
86   //CÁLCULO DE LA DECLINACIÓN SOLAR (ECUACIÓN DE SPENCER)
87   gamma=2*pi*(dn-1)/365;
88   declinacion_sol=0.0006918-0.399912*cos(gamma)+0.070257*sin(gamma)
    -0.006758*cos(2*gamma)+0.000907*sin(2*gamma)-0.002697*cos(3*gamma
    )+0.00148*sin(3*gamma);
89
90   //CÁLCULO ELEVACIÓN
91   h=asin(sin(latitud)*sin(declinacion_sol)+cos(latitud)*cos(
    declinacion_sol)*cos(HS));
92   Serial.print(h);
93   if(h>(pi/2) || h<(-pi/2)){
94     h=pi-h;

```

```

95 }
96 //CÁLCULO AZIMUT
97 coseno_az=(sin(declinacion_sol)-sin(latitud)*sin(h))/(cos(latitud)*
    cos(h));
98 seno_az=-sin(HS)*cos(declinacion_sol)/cos(h);
99 if(coseno_az >= 0 && seno_az>=0){
100     azimut=acos(sin(declinacion_sol)-sin(latitud)*sin(h))/(cos(latitud)
        *cos(h));
101 }
102 if(coseno_az<0 && seno_az>=0){
103     azimut=acos(sin(declinacion_sol)-sin(latitud)*sin(h))/(cos(latitud)
        )*cos(h));
104 }
105 }
106 if(coseno_az<0 && seno_az<0){
107     azimut=2*pi-acos(sin(declinacion_sol)-sin(latitud)*sin(h))/(cos(
        latitud)*cos(h));
108 }
109 }
110 if(coseno_az>0 && seno_az<0){
111     azimut=asin(-sin(HS)*cos(declinacion_sol)/cos(h));
112 }
113
114 //POSICIONO SISTEMA EN AZIMUT
115 servo1.attach(PINSERVO,MIN,MAX); //Esta función otorga el pin donde
    se conecta el servo y los valores mínimo y máximo del pulso
    asociados a 0° y 180°.
116 servo1.write(90);
117 t0=millis();
118 if(azimut>azimut_real){ //Si el azimut que quiero es mayor que el
    real tendré que girar en sentido de las agujas del reloj
119     servo1.write(89); //El número indica los grados requeridos
120     t1=0;
121     while((t1-t0)<((azimut-azimut_real)/w)*1000){
122         t1=millis();
123     }
124     servo1.write(90);
125 }
126 else{ //Si el azimut real es mayor que el quiero tendré que girar en
    sentido contrario a las agujas del reloj
127     servo1.write(91); //El número indica los grados requeridos
128     t1=0;
129     while((t1-t0)<((azimut_real-azimut)/w)*1000){
130         t1=millis();
131     }
132     servo1.write(90);
133 }
134 }
135
136 //POSICIONO SISTEMA EN ELEVACIÓN
137 pinMode(pin_inc, INPUT);
138 servo2.attach(PINSERVO2,MIN,MAX);
139 servo2.write(90);
140 digitalWrite(pin_inc , HIGH); //activamos la resistencia interna PULL
    UP
141 t3=0;
142 while(digitalRead(pin_inc)==HIGH){

```

```
143   servo2.write(91); //Colocamos el dispositivo en elevación 0
144   }
145   servo2.write(90);
146   t2=millis();
147   while((t2-t3)<(h/w)*1000){
148     servo2.write(89); //Colocamos el dispositivo con la elevación teórica
149     t3=millis();
150   }
151   servo2.write(90);
152   }
153
154   void loop(){
155
156     //CONTROL ALGORITMO ESPIRAL
157     //Recogemos los valores en espiral
158     for(int i=1;i<4;i++){
159       servo1.write(90);
160       servo2.write(90);
161       data(i)=analogRead(sensor1)
162       data(i+1)=analogRead(sensor2)
163       data(i+2)=analogRead(sensor3)
164       data(i+3)=analogRead(sensor4)
165       servo1.write(90);
166       servo2.write(90);
167     }
168
169     for(int i;i<8;i++){
170       servo1.write(92);
171       servo2.write(90);
172       data(i)=analogRead(sensor1)
173       data(i+1)=analogRead(sensor2)
174       data(i+2)=analogRead(sensor3)
175       data(i+3)=analogRead(sensor4)
176       servo1.write(90);
177       servo2.write(90);
178     }
179     for(int i;i<12;i++){
180       servo1.write(92);
181       servo2.write(92);
182       data(i)=analogRead(sensor1)
183       data(i+1)=analogRead(sensor2)
184       data(i+2)=analogRead(sensor3)
185       data(i+3)=analogRead(sensor4)
186       servo1.write(90);
187       servo2.write(90);
188     }
189     for(int i;i<12;i++){
190       servo1.write(90);
191       servo2.write(92);
192       data(i)=analogRead(sensor1)
193       data(i+1)=analogRead(sensor2)
194       data(i+2)=analogRead(sensor3)
195       data(i+3)=analogRead(sensor4)
196       servo1.write(90);
197       servo2.write(90);
198     }
199     for(int i;i<16;i++){
```

```
200  servo1.write(90);
201  servo2.write(91);
202  data(i)=analogRead(sensor1)
203  data(i+1)=analogRead(sensor2)
204  data(i+2)=analogRead(sensor3)
205  data(i+3)=analogRead(sensor4)
206  servo1.write(90);
207  servo2.write(90);
208  }
209  for(int i;i<20;i++){
210  servo1.write(91);
211  servo2.write(91);
212  data(i)=analogRead(sensor1)
213  data(i+1)=analogRead(sensor2)
214  data(i+2)=analogRead(sensor3)
215  data(i+3)=analogRead(sensor4)
216  servo1.write(90);
217  servo2.write(90);
218  }
219  //Algoritmo Bubble Sort de ordenación de un vector de mayor a menor.
    GUARDA LA POSICIÓN DEL VALOR NO EL VALOR EN SÍ
220  for (k = 1; (k <= length) && flag; i++)
221  {
222  flag = 0;
223  for (j = 0; j < (length - 1); j++)
224  {
225  if (data[j + 1] > data[j])
226  {
227  temp = data[j];
228  data[j] = data[j + 1];
229  data[j + 1] = temp;
230  posicion=j;
231  flag = 1;
232  }
233  }
234  }
235  //En función de la posición llevamos el mecanismo al lugar donde el
    valor de entrada es mayor
236  if(posicion<=4){
237  servo1.write(92);
238  servo2.write(90);
239  }
240  else if(posicion>4 && posicion <8){
241  servo1.write(92);
242  servo2.write(92);
243  }
244  else if(posicion>8 && posicion <12){
245  servo1.write(90);
246  servo2.write(92);
247  }
248  else if(posicion>12 && posicion <16){
249  servo1.write(90);
250  servo2.write(91);
251  }
252  else if(posicion>16 && posicion <20){
253  servo1.write(91);
254  servo2.write(91);
```

```
255 }  
256 //Esperar un minuto  
257 delay(60000)  
258 }
```

# 7 Resultados

---

A continuación, se muestran y se comentan 6 gráficas correspondientes a 3 días consecutivos usando un código cada día de 12:00 a 14:00, una vez en mayo y otra en junio. Teniendo en cuenta que las medidas se toman cada 1 minuto, almacenamos en nuestra memoria microSD un total de 120 valores; suficientes para una representación certera.

## 7.1 Toma de medidas del 5 al 7 de mayo

Durante la primera semana de mayo se realizaron numerosas medidas en el horario comprendido entre las 12:00 y las 14:00 aproximadamente en el entorno de Ciudad Real. Se han seleccionado como medidas representativas las de los días 5, 6 y 7 de mayo, en cada uno de los cuales se usó un algoritmo diferente y la meteorología de estos días fue bastante similar.

### 7.1.1 5 de mayo: Algoritmo octogonal

El 5 de mayo de 2022 se llevaron a cabo unas mediciones entre las 12:04 y las 14:06. Haciendo uso de la información de la "data sheet" del sensor se obtuvieron unas medidas aproximadas en unidades más útiles para la comparativa y sacar mejores conclusiones. La gráfica obtenida es la siguiente:

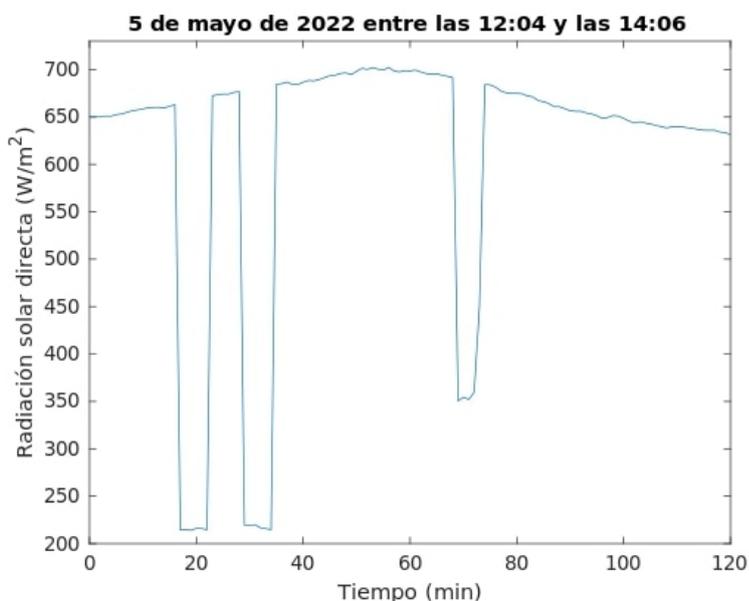


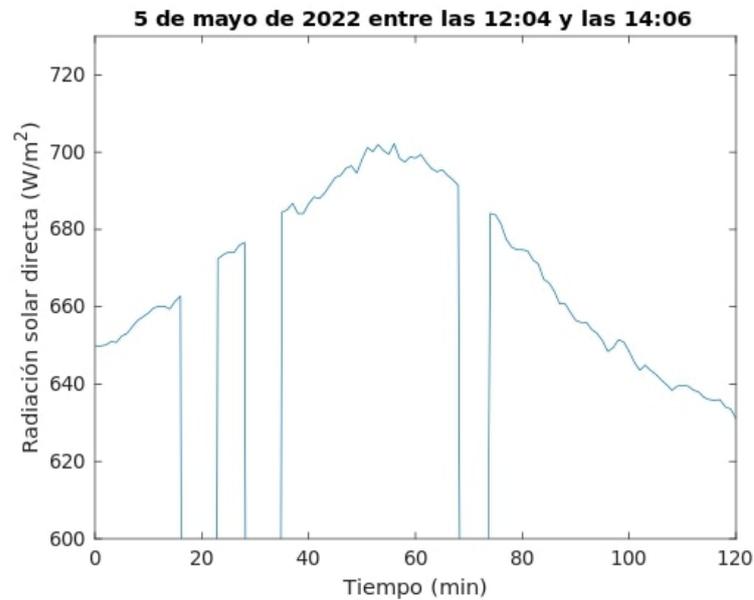
Figura 7.1 Medidas del 5 de mayo haciendo uso del algoritmo octogonal.

La elección de estas horas para los experimentos es poder apreciar la curva característica que hace la recepción de radiación solar durante un día, alcanzando el punto máximo y de inflexión cuando el Sol tiene

elevación máxima. En este caso este punto se alcanza sobre las 13:02.

Se pueden observar tres caídas repentinas en la recepción de radiación solar directa, presumiblemente relacionadas con la meteorología (nubes). Las dos primeras duraron apenas 5 minutos con una bajada del 71%. La tercera es algo menor en duración, unos 3 minutos, y en intensidad, con pérdidas aproximadas del 42%.

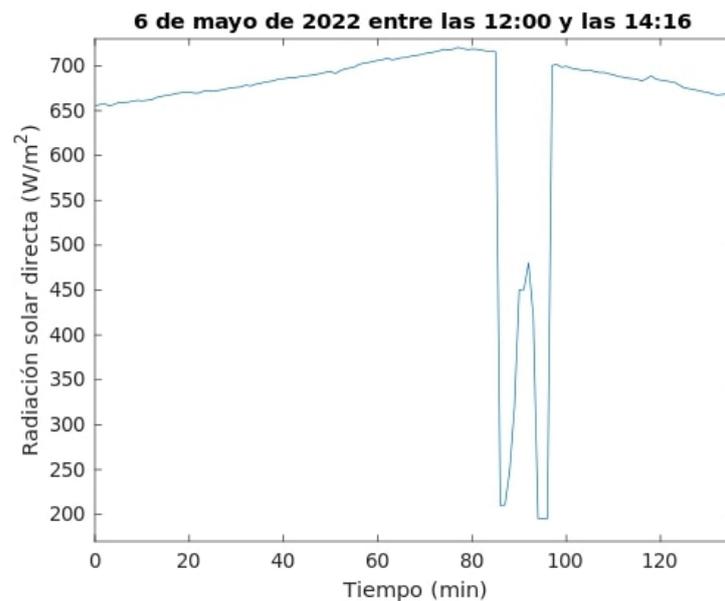
Para observar mejor la curva se obvian las inclemencias meteorológicas y se representan los datos otra vez obteniendo la siguiente gráfica.



**Figura 7.2** Zoom de las medidas del 5 de mayo haciendo uso del algoritmo octogonal .

### 7.1.2 6 de mayo: Algoritmo en espiral

El 6 de mayo de 2022 se recogieron datos entre las 12:00 y las 14:16 obteniendo lo siguiente:



**Figura 7.3** Medidas del 6 de mayo haciendo uso del algoritmo en espiral .

Alcanza el máximo de captación de radiación solar directa recibida sobre las 13:06 y la gráfica tiene la forma esperada excepto por una bajada repentina, diferente a la correspondiente al 5 de mayo. Sobre las 13:23 se produjo una bajada en la recepción de radiación solar al sensor del 70% ,aproximadamente, para 4 minutos después subir hasta los  $450 \frac{W}{m^2}$  y volver a bajar hasta los  $200 \frac{W}{m^2}$ , aproximadamente. Existen varias explicaciones a este fenómeno. Una de ellas es que el sensor o el algoritmo perdieron las referencia al dejar de tapar el sol alguna nube y tardó cerca de 8 minutos en volver a tomar datos certeros. Otra posible explicación es el paso de dos nubes pequeñas con una diferencia de 5 minutos; aunque en condiciones normales después de la primera nube tendría que haber vuelto a valores en torno  $650 \frac{W}{m^2}$  y no lo hizo; por lo que esta hipótesis es menos consistente.

Para una mejor visión de los datos característicos se amplía la zona de la curva.

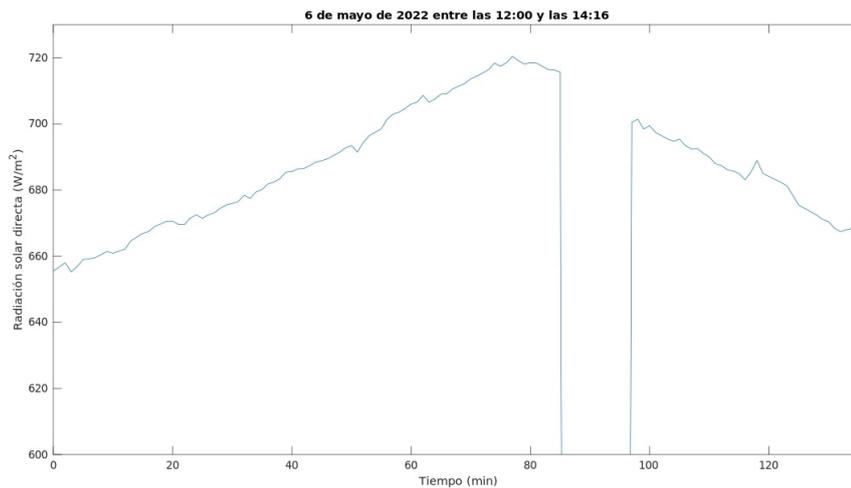


Figura 7.4 Zoom de las medidas del 6 de mayo haciendo uso del algoritmo en espiral.

### 7.1.3 7 de mayo: Algoritmo con 4 valores del sensor

El 7 de mayo de 2022 se recogieron datos entre las 12:05 y las 13:59 obteniendo lo siguiente:

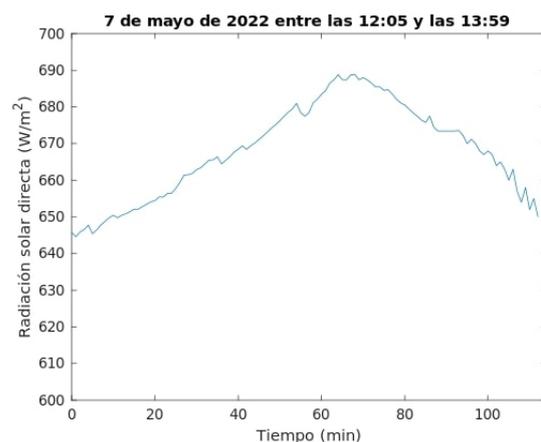


Figura 7.5 Medidas del 7 de mayo haciendo uso del algoritmo con 4 valores del sensor.

Podemos apreciar el máximo sobre las 13:10 con  $690 \frac{W}{m^2}$ . De los 7 días en los que se tomaron medidas fue el único en el que no hubo interrupciones por inclemencias meteorológicas por lo que es una toma de datos idónea para ver el funcionamiento del mecanismo. Se pueden apreciar algunas correcciones bruscas sobre las 13:53, en las cuales el sensor perdió el máximo en varias ocasiones, tardando entre 1 y 2 minutos (1 ó 2 iteraciones) en encontrarlo de nuevo.

## 7.1.4 Comparación de los 3 días

Uno de los problemas al procesar los datos de estos 7 días es que no se fue preciso en las horas de inicio y final de los experimentos (solucionado en la segunda ronda de toma de datos) por lo que la superposición de gráficas estaría desplazada en abscisas, es decir, solo valdría para comparar máximos, mínimos y forma de las curvas.

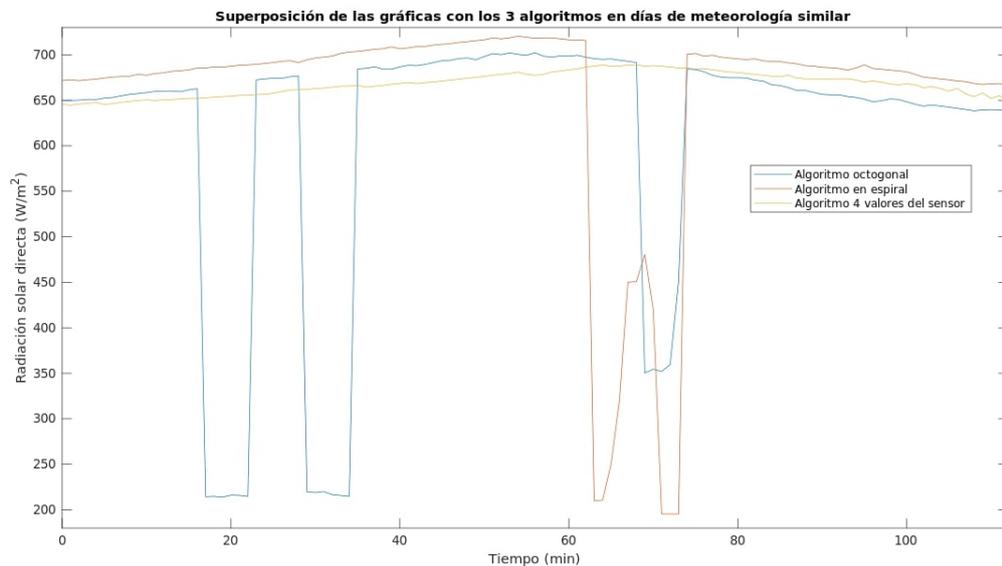


Figura 7.6 Comparación de los 3 algoritmos.

Haciendo zoom para poder apreciar las diferencias en las curvas obtenemos la siguiente representación.

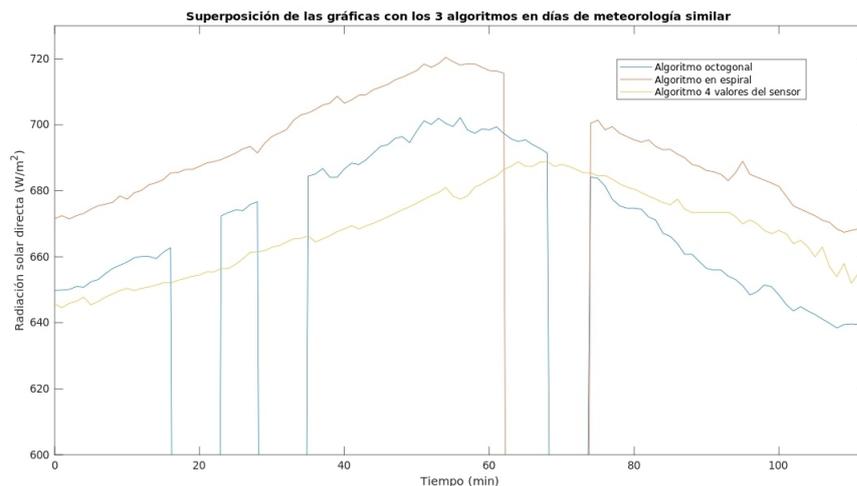


Figura 7.7 Zoom en la gráfica de comparación de los 3 algoritmos.

El algoritmo que parece más eficiente es el que recorre la búsqueda del máximo en espiral con una diferencia, en media, con respecto al algoritmo octogonal de  $27 \frac{W}{m^2}$ .

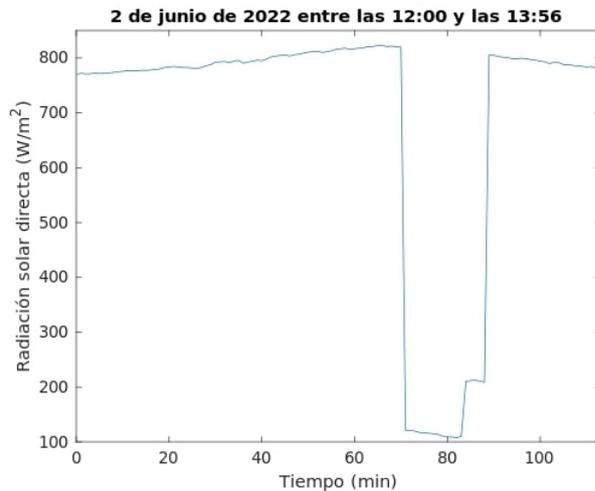
La forma del algoritmo que toma los 4 valores del sensor tiene una forma diferente a la de los otros dos, pues presenta una disposición más plana lo que indica que rara vez alcanza valores cercanos al máximo en el experimento.

## 7.2 Toma de medidas del 2 al 4 de junio

Se realizó otra toma de medidas la primera semana de junio, escogiendo los días del 2 al 4 de junio como datos representativos de los 3 algoritmos. A diferencia de la toma de mayo esta vez las horas de medida son las mismas (de 12:00 a 13:56). Como en el caso anterior los tres días presentan características climatológicas similares.

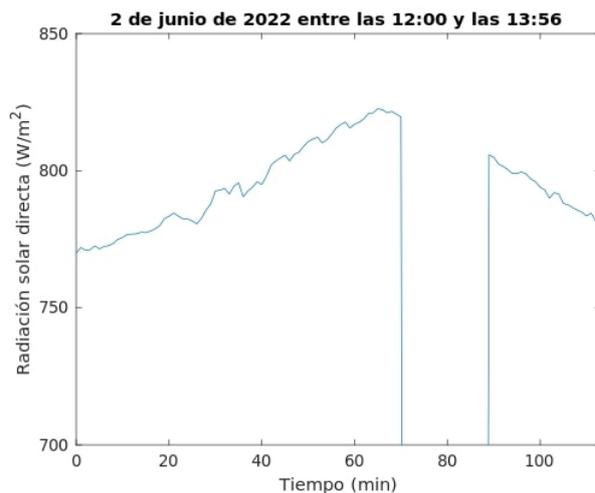
### 7.2.1 2 de junio: Algoritmo octogonal

El 2 de junio de 2022 se recogieron datos entre las 12:00 y las 13:56 obteniendo los siguientes datos:



**Figura 7.8** Medidas del 2 de junio haciendo uso del algoritmo octogonal.

De 13:10 a 13:26 se observa una caída repentina, de cerca del 85% en la recepción de energía solar directa. Como en casos anteriores se puede deber al paso de una nube (o varias) por delante del Sol. Para observar mejor la distribución de la curva se hace zoom a la zona más característica del estudio. Alcanza

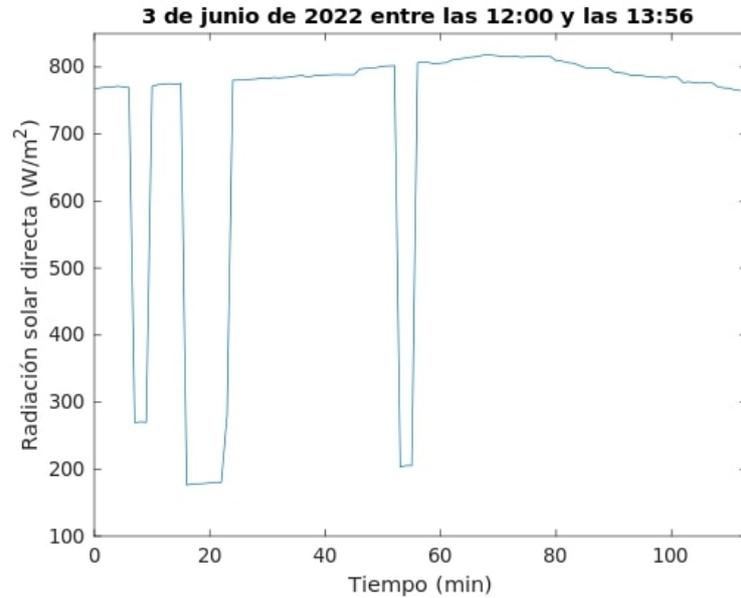


**Figura 7.9** Zoom de las medidas del 2 de junio haciendo uso del algoritmo octogonal.

el máximo sobre las 13:03 con una radiación solar directa de  $815 \frac{W}{m^2}$  aproximadamente. Se puede observar que entre las 12:20 y las 12:44 perdió el máximo en numerosas ocasiones, encontrándolo de nuevo en 1 o 2 interacciones del código.

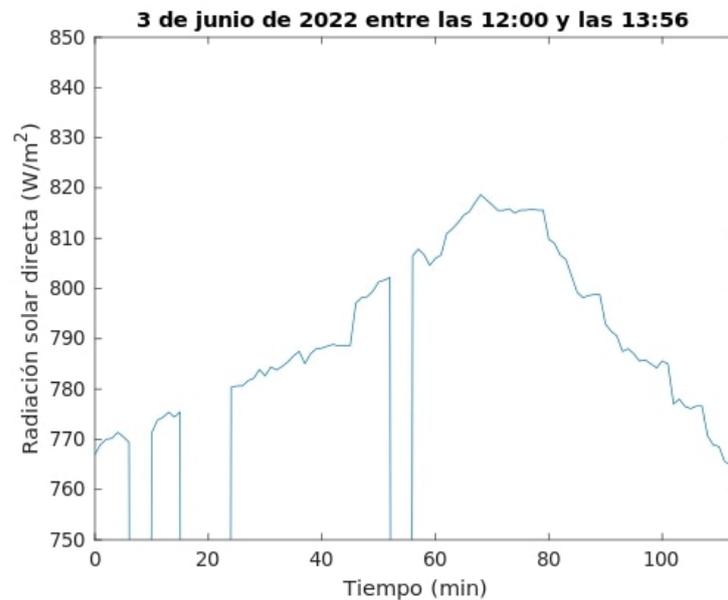
### 7.2.2 3 de junio: Algoritmo con 4 valores del sensor

El 3 de junio de 2022 se recogieron datos entre las 12:00 y las 13:56 obteniendo lo siguiente:



**Figura 7.10** Medidas del 3 de junio haciendo uso del algoritmo con 4 valores del sensor.

Como se puede observar se observan tres interrupciones por inclemencias meteorológicas: una de 12:05 a 12:07 con una bajada en la recepción del 61%, otra de 12:17 a 12:22 del 73% y la última de 12:53 a 12:58 del 68%. A continuación se representa la gráfica ampliada para ver el máximo y analizar en detalle lo ocurrido en el experimento.



**Figura 7.11** Zoom en las medidas del 3 de junio haciendo uso del algoritmo con 4 sensores del sensor.

Podemos observar que el máximo se alcanza a las 13:04 con un valor de  $820 \frac{W}{m^2}$ . Además ocurren dos fenómenos interesantes: a las 12:47 se produce una subida repentina, lo que hace indicar que antes se estaba lejos del máximo y en un iteración sube el valor de voltaje recibido sustancialmente. Por el contrario, a las 13:23 ocurre el suceso contrario, una bajada repentina de la pendiente de la curva, lo que indica la pérdida

del máximo y la consecuente pérdida de energía registrada.

### 7.2.3 4 de junio: Algoritmo en espiral

El 4 de junio de 2022 se recogieron datos entre las 12:00 y las 13:56 obteniendo lo siguiente:

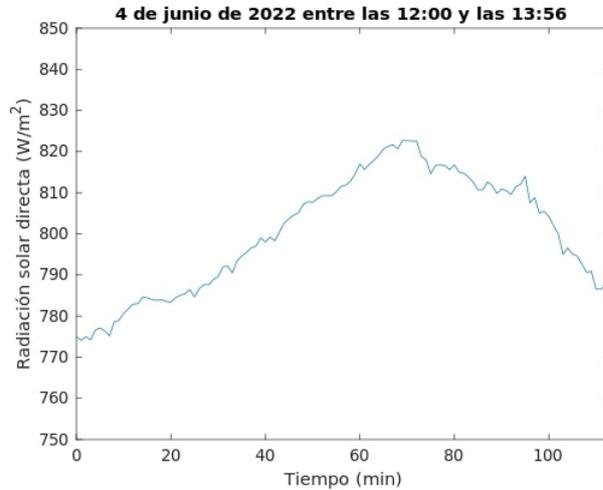


Figura 7.12 Medidas del 4 de junio haciendo uso del algoritmo en espiral.

Podemos observar que el máximo se alcanza sobre las 13:08 en un valor de  $821 \frac{W}{m^2}$ . Además entre las 12:19 y las 12:24 se repiten aproximadamente los mismos valores, lo que supone que el mecanismo no encontraba máximos a los que moverse. Otro evento a destacar es que a las 13:39 se produce un pico de energía recibida que no se mantiene en las siguientes medidas, lo que supone que el sensor encontró un valor de radiación recibida más cercano al óptimo pero no fue capaz de ubicarlo en movimientos posteriores.

### 7.2.4 Comparación de los 3 días

En este nuevo experimento se soluciona el problema de mayo de no tener los datos en las mismas ventanas de tiempo, por lo que ahora sí es posible comparar las gráficas en su conjunto.

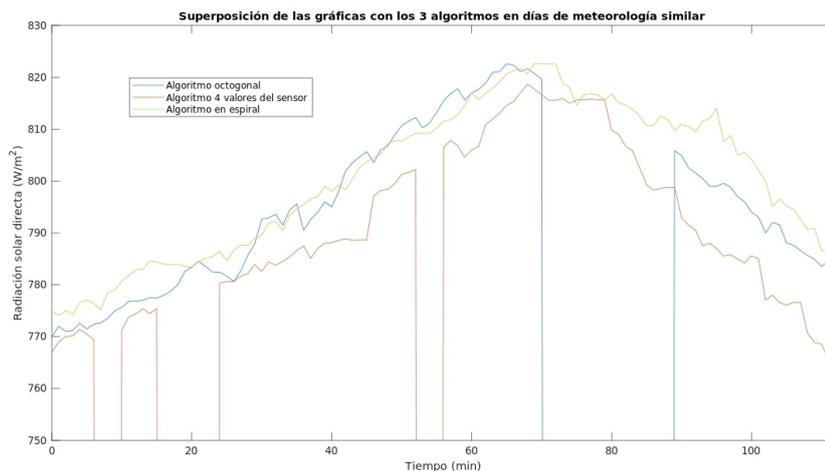


Figura 7.13 Comparación de los datos recogidos usando los 3 algoritmos.

A tenor de la superposición de gráficas podemos sacar algunas conclusiones.

En la subida los algoritmos ortogonal y espiral tienen comportamientos muy parecidos pero tras un evento meteorológico el ortogonal pierde la referencia del máximo y baja su eficiencia considerablemente.

Se parece constatar en ambos experimentos que el uso de sólo 4 medidas para encontrar el máximo es ineficiente con respecto a los otros dos códigos.

## 8 Líneas de trabajo en el futuro

---

En marzo de 2022 la irrupción de la guerra de Ucrania ha generado en Europa, dada su dependencia del gas ruso, un incremento importante del precio de la energía. Esta crisis energética ha puesto de manifiesto la importancia de la energía solar para un futuro sostenible. En España el encarecimiento de la luz ha supuesto un incremento en las instalaciones de placas solares domésticas; en Catalunya una cuarta parte de las totales se han instalado en 2022, lo que hace visible la importancia de la eficiencia en este tipo de estructuras.

Las líneas futuras que deberían seguir a este trabajo se focalizan en el tema del aumento de la eficiencia tanto del propio algoritmo desarrollado como de la estructura mecánica.

Analizando los datos se infiere que cuando, debido a inclemencias meteorológicas, se pierde la referencia del valor anterior la eficiencia cae hasta un 12%. Además, las gráficas muestran picos de subida abruptos que no mantienen su valor en el tiempo, lo que hace indicar que el recorrido que sigue la gráfica dista mucho de ser el recorrido óptimo. Estos dos hechos son elementos a mejorar del código ya que, en términos estimativos, pueden suponer mejoras de hasta el 55%.

En términos estructurales, el momento que se genera en el servo de elevación no aguanta de manera férrea los componentes electrónicos y la cabeza del mecanismo acaba 2º bajo la horizontal, lo que supone otro factor en cuanto a pérdida de eficiencia.

Hay que tener en cuenta que un avance en la eficiencia en instalaciones de grandes magnitudes tienen un gran impacto en términos económicos y energéticos. Estas tres mejoras incrementarían la eficiencia del modelo planteado en este TFG.



# Índice de Figuras

---

1.1	Estructura interna del Sol	2
1.2	Distribución de frecuencias de algunas contribuciones a la radiación total	2
1.3	Espectro solar	2
1.4	Mapa mundial de llegada de radiación solar directa	3
1.5	Mapa mundial de llegada de radiación solar total	3
1.6	Órbita de la Tierra alrededor del Sol	4
1.7	Tipos de radiaciones solares que llegan a la superficie	5
1.8	Central Termosolar de canales parabólicos de «La Africana» promovida por el grupo Magtel	6
1.9	Central Termosolar PS10 de Abengoa en Sanlúcar la Mayor, primera Central de Heliostatos en el mundo.	6
1.10	Central Termosolar con reflectores lineales de Magtel en Aerópolis	6
1.11	Central Termosolar con discos parabólicos en Villarobledo, primera en España de este tipo.	7
2.1	Origen del triángulo astronómico	9
2.2	Triángulo astronómico	10
2.3	Triángulo esférico utilizado para calcular la ascensión recta del sol	11
3.1	Arduino Micro	13
3.2	Brújula digital GY-271	14
3.3	Conexiones del magnetómetro GY-273 (equivalente a las de GY-271) con Arduino Micro	14
3.4	Inclinómetro SW-520D	15
3.5	Caption	15
3.6	NANO-ISSX/c	16
3.7	Ventana de filtración del sensor de radiación	16
3.8	Pines de conexión del sensor de radiación NANO ISSX/c	16
3.9	Servomotor Parallax de giro continuo	17
3.10	Relación entre la velocidad angular del servo y el ancho de banda de la señal	17
3.11	Ancho de banda de la señal para el servo parado, a favor y en contra de las agujas del reloj	18
3.12	Lector de tarjetas MicroSD mediante bus SPI	18
4.1	Sketch del rectángulo de la base del mecanismo	19
4.2	Altura de la base	19
4.3	Sketch del grosor de la base	20
4.4	Altura para hacer la pieza hueca	20
4.5	Sketch del agujero para el eje de giro	21
4.6	Pieza 3D con el agujero para dar salida al eje de giro	21
4.7	Sketch de la primera estructura que da estabilidad al mecanismo	21
4.8	Ancho de la primera estructura que da estabilidad al mecanismo	22
4.9	Sketch de la segunda estructura que da estabilidad al mecanismo	22
4.10	Ancho de la segunda estructura que da estabilidad al mecanismo	22
4.11	Agujeros para atornillar la base	23
4.12	Agujero para el cable que conecta el servo de azimut al Arduino	23

4.13	Base y cubierta del servo de azimut en perspectiva isométrica	23
4.14	Sketch de la estructura en forma de L	24
4.15	Estructura primaria en forma de L	24
4.16	Sketch para el anclaje del servo de elevación en la estructura	24
4.17	Sketch de los agujeros para unir el servo de azimut a la L	25
4.18	L en perspectiva isométrica	25
4.19	Dimensiones de la circunferencia	25
4.20	Grosor del cilindro	26
4.21	Agujeros para la fijación con el servo de elevación	26
4.22	Sketch de la base de la cabeza	26
4.23	Sketch de los agujeros de fijación a la cabeza	27
4.24	Perspectiva isométrica del enganche del servo de elevación con la cabeza	27
4.25	Sketch que determina la forma de la base	27
4.26	Sketch que sigue la guía que sigue el otro sketch	28
4.27	Base primaria de la cabeza inferior	28
4.28	Base primaria cerrada de la cabeza inferior	28
4.29	Sketch para el agujero, en la cabeza inferior, del sensor de radiación solar directa	29
4.30	Base primaria de la cabeza inferior con el agujero para el sensor de radiación solar	29
4.31	Sketch de coronas circulares para el acoplamiento entre la base inferior y la superior	29
4.32	Pieza con las coronas circulares para el acoplamiento entre la base inferior y la base superior	30
4.33	Sketch de los agujeros de conexión.	30
4.34	Cabeza inferior en vista isométrica	30
4.35	Sketch para el "Rib" que propaga la forma	31
4.36	Sketch para el "Rib" que conforma la guía	31
4.37	Pieza primaria que conforma el parasol	31
4.38	Sketch 1 para la eliminación de material	32
4.39	Sketch 2 para la eliminación de material	32
4.40	Parasol	32
4.41	Sketch 1 para crear el techo	33
4.42	Sketch 2 para crear el techo	33
4.43	Pieza con la adhesión del techo	33
4.44	Sketchs para la pared trasera y delantera	34
4.45	Pieza con la creación de las paredes	34
4.46	Sketch para el agujero del sensor de radiación solar directa	34
4.47	Pieza con el agujero para el sensor	35
4.48	Sketch para los agujeros de los servos	35
4.49	Cabeza superior con los agujeros para los servos	35
4.50	Sketch para los salientes del acoplamiento con la cabeza inferior	36
4.51	Cabeza inferior con los salientes para el acoplamiento	36
4.52	Sketch 1 guía del cono	36
4.53	Sketch 2 guía del cono	37
4.54	Cabeza superior en vista isométrica	37
4.55	Ensamblaje en vista isométrica	37
5.1	Estructura antes de iniciar el montaje de los sensores y circuitos	39
5.2	Acoplamiento del inclinómetro a la estructura	39
5.3	Conexión del inclinómetro	40
5.4	Conexión del magnetómetro	40
5.5	Conexión del sensor de radiación	41
5.6	Conexión de los servos	41
5.7	Disposición de la tarjeta SD	42
5.8	Disposición final de la parte electrónica	42
5.9	Mecanismo listo para la parte experimental (Vista frontal)	42
5.10	Mecanismo listo para la parte experimental (Vista trasera)	43
7.1	Medidas del 5 de mayo haciendo uso del algoritmo octogonal	61
7.2	Zoom de las medidas del 5 de mayo haciendo uso del algoritmo octogonal	62

---

7.3	Medidas del 6 de mayo haciendo uso del algoritmo en espiral	62
7.4	Zoom de las medidas del 6 de mayo haciendo uso del algoritmo en espiral	63
7.5	Medidas del 7 de mayo haciendo uso del algoritmo con 4 valores del sensor	63
7.6	Comparación de los 3 algoritmos	64
7.7	Zoom en la gráfica de comparación de los 3 algoritmos	64
7.8	Medidas del 2 de junio haciendo uso del algoritmo octogonal	65
7.9	Zoom de las medidas del 2 de junio haciendo uso del algoritmo octogonal	65
7.10	Medidas del 3 de junio haciendo uso del algoritmo con 4 valores del sensor	66
7.11	Zoom en las medidas del 3 de junio haciendo uso del algoritmo con 4 sensores del sensor	66
7.12	Medidas del 4 de junio haciendo uso del algoritmo en espiral	67
7.13	Comparación de los datos recogidos usando los 3 algoritmos	67



# Bibliografía

---

- [1] MÉNDEZ MUÑIZ, JAVIER MARÍA y CUERVO GARCÍA, RAFAEL , *Energía Solar Térmica*. Fundación Confe-metal p.37
- [2] GREENPEACE (14 DE ENERO DE 2018) «Individual action: what you can do about climate change», <http://www.greenpeace.org/international/en/>.
- [3] MEHMET KANOGLU, YUNUS A. CENGEL DR y JOHN M. CIMBALA , *Fundamentals and Applications of Renewable Energy*. Mc Graw Hill.
- [4] LUIS MIREZ TARRILLO, JORGE «Matlab/Simulink and Renewable Energy», <http://www.jmirez.wordpress.com/2013/06/05/j585-componentes-de-la-radiacion-solar/>.
- [5] ARSUSENERGIA (13 DE NOVIEMBRE DE 2020) «CENTRALES TERMOSOLARES, SU FUNCIONAMIENTO Y TIPOS», <https://arsusenergia.es/centrales-termosolares-su-funcionamiento-y-tipos/>.
- [6] IDEAM «LA RADIACIÓN SOLAR Y SU PASO POR LA ATMÓSFERA», <http://www.ideam.gov.co/web/tiempo-y-clima/la-radiacion-solar-y-su-paso-por-la-atmosfera>
- [7] VAZQUEZ VALENZUELA, RAFAEL, *Material de la asignatura "Mecánica Orbital" de Ingeniería Aeroespacial*.2022.
- [8] VERA MELLA, NELSON, *Atlas climático de irradiación solar a partir de imágenes del satélite NOAA. Aplicación a la península Ibérica*.2005.
- [9] CARACTERÍSTICAS TÉCNICAS DE ARDUINO MICRO, <https://www.web-robotica.com/arduino/placas-arduino/arduino-micro-genuino-micro-caracteristicas-tecnicas>.
- [10] IMAGEN DE ARDUINO MICRO, <https://m.media-amazon.com/images/I/71LFVqNjUxL.ACSX450.jpg>.
- [11] INFORMACIÓN DE MAGNETÓMETRO HMC5883, <https://www.luisllamas.es/brujula-magnetica-con-arduino-compass-digital-hmc5883/>.
- [12] IMÁGENES DE CONEXIONES DE MAGNETÓMETRO HMC5883, <https://www.luisllamas.es/wp-content/uploads/2016/09/arduino-brujula-HMC5883-GY273-esquema.png>.
- [13] IMAGEN SW520D, <https://www.sunrom.com/media/product/1782.jpg>.
- [14] INFORMACIÓN DEL SENSOR SW520D, <https://www.luisllamas.es/medir-inclinacion-con-arduino-y-sensor-tilt-sw-520d/>.
- [15] IMAGEN DEL FUNCIONAMIENTO DEL SENSOR SW520D, <https://www.luisllamas.es/wp-content/uploads/2015/08/sw520d.png>.
- [16] INFORMACIÓN SOBRE CONTROL DEL SERVOMOTOR, <https://www.luisllamas.es/controlar-un-servo-de-rotacion-continua-con-arduino/>.
- [17] IMAGEN TARJETA MICRO SD, <https://leantec.es/wp-content/uploads/2018/02/p3499499-MODULO-MICRO-SD-LECTOR-TARJETA-ARDUINO.jpg>.