



CCADET

**CENTRO DE CIENCIAS APLICADAS Y DESARROLLO
TECNOLÓGICO
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

Sistema electromecánico para el seguimiento de la trayectoria solar

Benjamín Valera Orozco
Juan Salvador Pérez Lomeli
Gerardo Antonio Ruiz Botello
Rigoberto Nava Sandoval
Roberto U. Reséndiz Núñez

Octubre 2015

Proyecto Interno

Financiamiento interno

Sistema electromecánico para el seguimiento de la trayectoria solar

Benjamín Valera Orozco, Juan Salvador Pérez Lomeli, Gerardo Antonio Ruiz Botello, Rigoberto Nava Sandoval

Centro de Ciencias Aplicadas y Desarrollo Tecnológico
Universidad Nacional Autónoma de México, Circuito Exterior S/N, Ciudad Universitaria AP 70-186, C.P.
04510, México D.F. México

Resumen

En el presente informe se describe el diseño y construcción de un sistema electromecánico para el seguimiento de la trayectoria solar con fines demostrativos. En futuras aplicaciones, el trabajo descrito podría ser útil en diversos ámbitos como la generación de energía fotovoltaica o calentadores solares. El sistema consiste de tres componentes: Una componente mecánica constituida por un prototipo de dos grados de libertad que orienta el dispositivo de captación, una componente electrónica que implementa el algoritmo de seguimiento solar y una componente software para Smartphone que implementa la interface con el usuario. Mediante los dos grados de libertad en la componente mecánica se generan los ángulos de altitud y azimut que permiten el rastreo de cualquier objeto celeste en la bóveda solar desde una posición fija en la tierra. En la componente electrónica, un microcontrolador obtiene el tiempo actual mediante un circuito de reloj en tiempo real para generar los ángulos de altitud y azimut con base en las ecuaciones solares. La componente software permite la comunicación con la componente electrónica a través de una interface WiFi para configurar o monitorear el sistema completo de seguimiento solar. Se presenta el diseño del sistema describiendo las tres componentes y resultados que verifican el desempeño del aparato en una aplicación simple de orientación de un panel solar para la captación de energía fotovoltaica.

Indice

Introducción	4
Definición del problema	4
Entorno actual	4
Descripción del problema a resolver y motivación	4
Relevancia y limitaciones	4
Relación con otras áreas	5
Método	5
Objetivos y resultados esperados	5
Organización del informe	6
1 Trayectoria del sol	7
1.1 Trayectoria tierra-sol	7
1.2 Trayectoria aparente del sol	8
1.2.1 Registro de la fecha	9
1.2.2 Traslación de la trayectoria solar	10
1.2.3 Rotación de la trayectoria solar	12
1.3 Ajuste del tiempo	13
1.4 Ángulos solares	13
1.5 Ejemplo de aplicación	14
2 Descripción del sistema electromecánico	18
2.1 Componente electrónica	19
2.1.1 Etapa de radiofrecuencia	19
2.1.2 Etapa de registro del tiempo	21
2.1.3 Etapa de acondicionamiento de señales	21
2.1.4 Etapa de control	22
2.1.5 Etapa impulsora de motores a paso	24
2.1.6 Etapa para el sensor de posición	24
2.1.7 Etapa de alimentación de energía	25
2.1.8 Firmware del microcontrolador	26
2.2 Descripción del software en el ámbito del operador	29
2.3 Descripción del software en el ámbito del programador	33
2.3.1 Descripción general del proyecto de programación	34
2.3.2 Descripción del código fuente	35
2.4 Componente mecánica	46
2.5 Procedimiento típico para operar el sistema	48
3 Resultados y discusión	51
3.1 Resultados	51
3.1.1 Prototipo desarrollado	51
3.1.2 Comparación teórica contra real para los ángulos solares	53
3.1.3 Desempeño en posición del prototipo	54
3.2 Discusión y trabajo a futuro	55
Conclusiones	57
Anexo A Diagramas electrónicos	58
A.1 Diagrama esquemático	58
A.2 Circuito impreso	59
A.3 Lista de partes	61

Anexo B Planos de construcción	63
B.1 Planos	63
B.2 Lista de partes	82
Referencias	84

Introducción

Definición del problema

El seguimiento celestial ha sido utilizado ampliamente en las áreas de astrofotografía y energía solar. En estas áreas, un dispositivo de observación localizado en una posición fija sobre la tierra, es orientado para seguir a un objeto celestial. Un ejemplo es el seguidor solar, en donde un panel fotovoltaico o lentes concentradores son dirigidos hacia el sol con el objetivo de maximizar la energía colectada. Estos dispositivos de observación se orientan utilizando seguidores solares activos utilizando aparatos electromecánicos tales como moto-reductores y mecanismos de uno a varios grados de libertad. Está comprobado que los seguidores solares activos incrementan la eficiencia en aproximadamente 30 a 60% en comparación con los sistemas estacionarios.

Entorno actual

El principal antecedente para el desarrollo del presente proyecto es el artículo [Kassem y Hamad, 2011] titulado "A microcontroller-based multi-function solar tracking system" en donde se implementa un seguidor solar activo sobre la base de un microcontrolador. En éste trabajo, se utilizan tres foto-resistores que proporcionan la posición del sol cada 45 minutos y con ello se re-direcciona el panel solar hacia la posición más cercana a la máxima intensidad solar.

Descripción del problema a resolver y motivación

En el presente informe se describe el desarrollo de un sistema electromecánico para un seguidor solar activo con dos grados de libertad que resuelva las ambigüedades que pudieran existir al generar la trayectoria solar utilizando foto-resistores. En nuestro concepto, la generación de la trayectoria solar mediante las ecuaciones solares y una base de tiempo resulta ser más precisa y elimina las ambigüedades que introducen los días nublados o de clima adverso que afectan el desempeño de los sensores foto-resistivos.

El sistema electromecánico deberá operar continuamente a lo largo del día y durante largos periodos de tiempo, típicamente meses e inclusive años. En éste sentido, el objetivo del seguidor solar es seguir el sol sin importar la fecha ni las condiciones del clima.

El sistema deberá incluir una interface de operación inalámbrica del tipo Wi-Fi. Esta interface permitirá la configuración del sistema electrónico y el monitoreo de algunas variables que durante el proceso puedan ser relevantes, como la posición y variables ambientales.

Relevancia y limitaciones

La relevancia del proyecto radica en el método de seguimiento solar, ya que comúnmente los seguidores solares utilizan sensores fotovoltaicos para estimar la posición del sol y en este caso se utilizan una base de tiempo y las ecuaciones solares. En forma adicional el sistema incluye una conexión inalámbrica Wi-Fi como interface de configuración y monitoreo. Finalmente, se fomenta la creación de infraestructura de desarrollo propio que puede ser ofrecida como una transferencia tecnológica.

El alcance del presente proyecto es la creación de un prototipo demostrativo que pueda ser utilizado para probar conceptos y algoritmos de control propios.

La principal limitación del prototipo radica en que no constituye una fuente para la generación de electricidad a la red pública. En este sentido, se espera que el panel fotovoltaico proporcione energía exclusivamente para abastecer a su propia componente electrónica.

Relación con otras áreas

Existe una gran relación con las áreas de sistemas de comunicaciones, instrumentación, programación de computadoras, microcontroladores y microprocesadores.

Método

En este informe se describe el desarrollo hardware y software para implementar un seguidor solar con dos ejes activos. El esquema general se presenta en la figura 1.

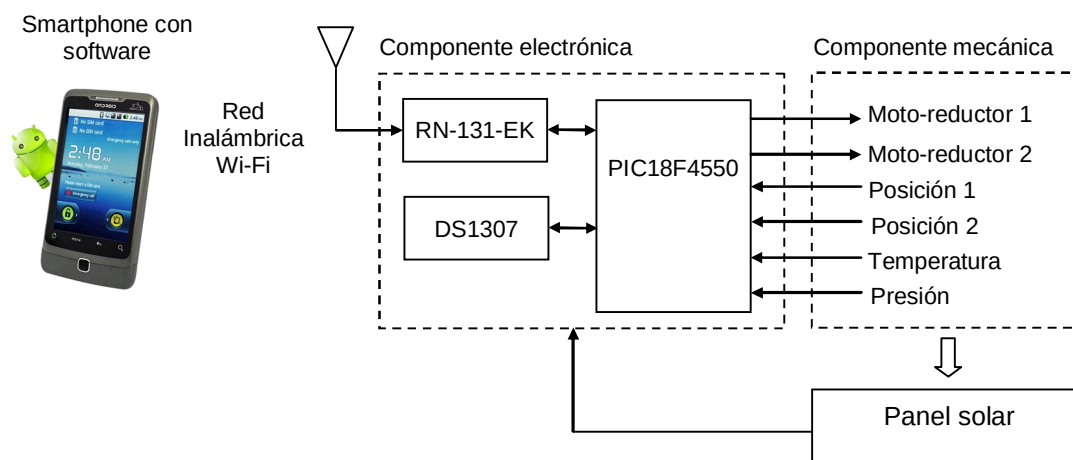


Figura 1. Método a emplear.

La componente electrónica tiene como base un microcontrolador PIC18F4550 de 8 bits con las siguientes características relevantes para la presente aplicación: Bajo consumo de potencia, interface I²C con dispositivos periféricos, 32kB de memoria programable e interface serie USART.

El microcontrolador es enlazado a un reloj de tiempo real DS1307, mediante la interface I²C. El DS1307 cuenta segundos, minutos, horas, día del mes, día de la semana y año en tiempo real y consume menos de 500nA sin perdidas en las cuentas. En este sentido, una batería estándar de litio de respaldo proporciona fuente de energía por aproximadamente 10 años, inclusive durante la ausencia de energía principal. El tiempo proporcionado por el reloj de tiempo real en conjunto con la latitud y longitud de la posición del sistema de seguimiento, son utilizados para calcular la posición aparente del sol. Con estos datos el microcontrolador ejecuta el algoritmo de seguimiento solar al mover dos moto-reductores que impulsan los dos ejes de libertad en el dispositivo electromecánico.

El microcontrolador se conecta al módulo Wi-Fi RN-131-EK mediante la interface USART. La interface Wi-Fi es utilizada para configuración y monitoreo de algunas variables que pueden ser críticas durante el funcionamiento del sistema como pueden ser temperatura y posición del panel fotovoltaico. La interface de operación es un programa específicamente desarrollado para la presente aplicación que se ejecuta desde un smartphone con Android.

La componente mecánica orienta un panel fotovoltaico con dos grados de libertad: altitud y azimut. Con estos dos grados de libertad es posible obtener la máxima irradiación solar. El panel fotovoltaico se utiliza para suministrar energía eléctrica a la componente electrónica.

Objetivos y resultados esperados

Desarrollar un sistema electromecánico para el seguimiento de la trayectoria solar que consiste de:

- Una componente electrónica con base en microcontrolador, reloj de tiempo real y tarjeta de interface Wi-Fi.
- Firmware en el microcontrolador para el control de la componente mecánica, orientación automática a la trayectoria solar y comunicación Wi-Fi.
- Una componente mecánica con dos ejes de libertad que le permiten orientar un panel solar.
- Software en un Smartphone para el monitoreo y configuración del sistema.

Organización del informe

La sección de introducción plantea el problema, el método empleado para resolverlo y expone los resultados esperados. El capítulo uno presenta los conceptos básicos que se utilizaron en el desarrollo del prototipo. Se enfatiza especialmente en las ecuaciones solares, ya que a partir de ellas es como se consigue el seguimiento solar. En el capítulo dos se describe el desarrollo de las tres componentes del sistema electromecánico: componente electrónica con base en MCU, RTC e interface WiFi, componente mecánica para el seguimiento en azimut y altitud de la trayectoria solar y software de operación desarrollado para dispositivos Android. El capítulo tres contiene los resultados, conclusiones y trabajo a futuro. Finalmente los anexos contienen información detallada acerca de los recursos desarrollados como parte del presente informe.

Se incluye un CD conteniendo la siguiente estructura de directorios.

```
Informe
CodigoFuente
    PIC
    Android
    MatLab
    PovRay
Programa
CADElectronica
```

El contenido de los directorios es como sigue:

Informe: El presente documento.

CodigoFuente: El código fuente de los programas desarrollados para el microcontrolador, la interface en el smartphone, los cálculos teóricos de los ángulos solares y la generación de figuras en un ambiente gráfico.

Programa: El archivo ejecutable de la aplicación para smartphone.

CADElectronica: Los archivos CAD para la construcción de la componente electrónica.

1 Trayectoria del sol

La trayectoria solar es un concepto dependiente de la variabilidad del ángulo que forma el eje de rotación de la tierra [Kalogirou, 2009].

Como se sabe, la existencia de las estaciones está motivada porque el eje de rotación de la tierra no es siempre perpendicular al plano de su trayectoria de traslación con respecto al sol, sino que forma un ángulo variable dependiendo del momento del año en que se encuentre.

1.1 Trayectoria tierra-sol

La órbita de la tierra es una elipse y no un círculo, por lo tanto la distancia entre la tierra y el sol varía durante el año, conduciendo a valores de irradiación solar aparente durante todo el año que están dados aproximadamente por la siguiente ecuación

$$I_0 = I_{SC} \left[1 + 0.033 \cos \left(\frac{N}{365} 360^\circ \right) \right]$$

En donde:

$$I_{SC} = 429.5 \frac{\text{Btu}}{\text{hr ft}^2} = 1353 \frac{\text{W}}{\text{m}^2}$$

Es la constante solar
es el número de días transcurridos

El punto más cercano de la tierra (alrededor de 146 millones km) al sol es llamado la perihelio y ocurre alrededor del 3 de enero; El punto más lejano de la tierra (alrededor de 156 millones km) al sol es llamado la aphelio y ocurre alrededor del 4 de julio. La figura 2 ilustra la trayectoria que genera la tierra al rotar alrededor del sol.

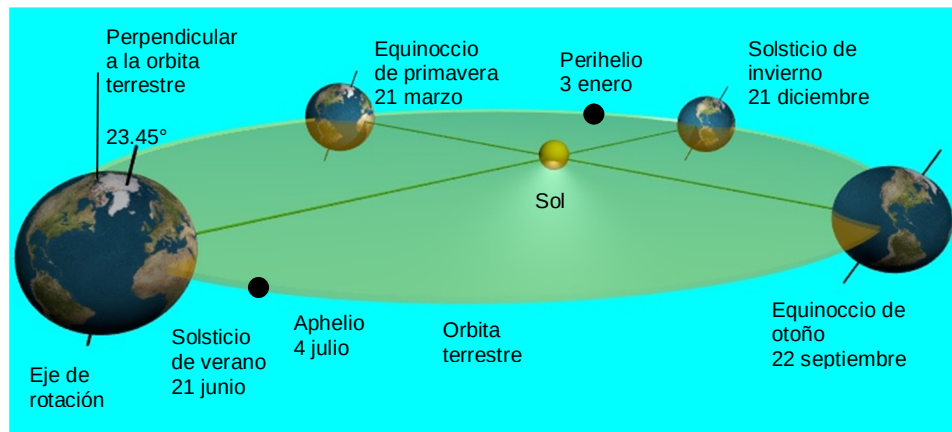


Figura 2. Trayectoria tierra-sol.

La tierra está inclinada sobre su eje en un ángulo de 23.45° grados. Como la tierra anualmente viaja alrededor del sol, la inclinación se manifiesta en nuestras estaciones del año. El sol cruza el ecuador alrededor del 21 de marzo (equinoccio de primavera) y 22 de septiembre (equinoccio de otoño). El sol alcanza su latitud más al norte alrededor del 21 de junio (solsticio de verano) y su latitud más al sur cerca del 21 de diciembre (solsticio de invierno).

La declinación es la distancia angular del norte del sol o sur del ecuador de la tierra. El ángulo de declinación, δ , para el hemisferio norte (invertir el signo del ángulo de declinación para el hemisferio sur) es:

$$\delta = 23.45^\circ \sin \left[\frac{N + 284}{365} 360^\circ \right] \quad (1)$$

donde N es el número de días del año, con el 1 de enero igual a 1.

La tierra está dividida en latitudes (división horizontal) y longitudes (divisiones norte-sur). El ecuador está a una latitud de 0° ; los polos norte y sur están a $+90^\circ$ y -90° , respectivamente; el trópico de cáncer y el trópico de capricornio están localizados a $+23.45^\circ$ y -23.45° , respectivamente. Para longitudes, la comunidad global ha definido 0° como el primer meridiano el cual está localizado en Greenwich, Inglaterra. Las longitudes se describen en términos de grados en que se encuentran al este o al oeste del meridiano de Greenwich. Un día tiene 24 hrs ó 1440 minutos, que cuando se divide por 360° significa que se tarda 4 minutos para mover cada grado de longitud. Ver figura 3.

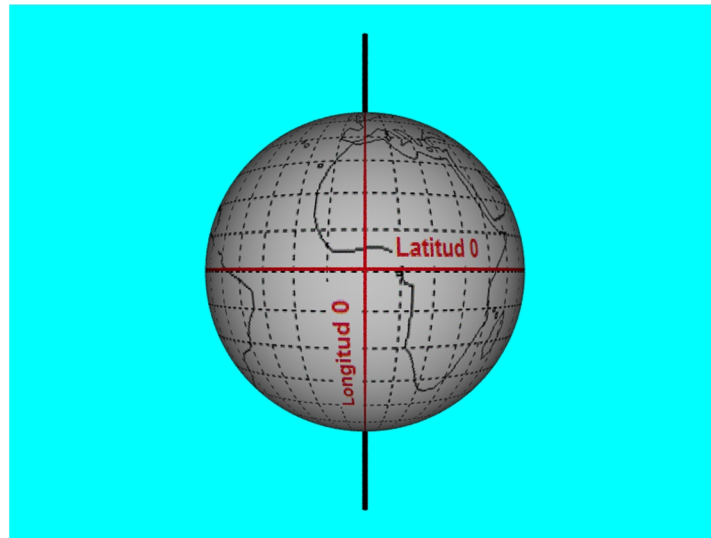


Figura 3. Longitud y latitud de la tierra.

1.2. Trayectoria aparente del sol

Si se observa el cielo desde la tierra, da la impresión de que el sol se mueve alrededor del plano terrestre, como se muestra en la figura 4.

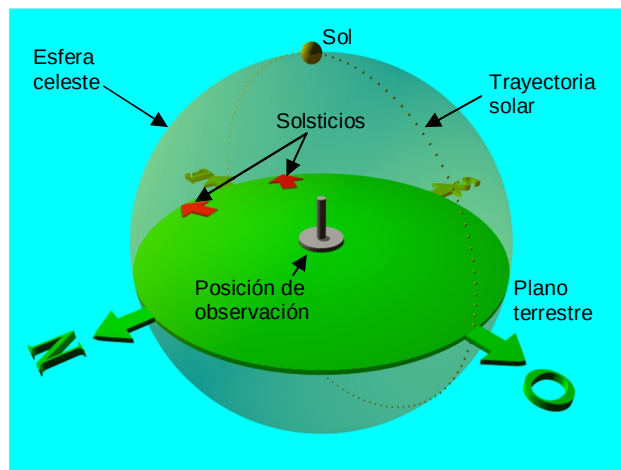


Figura 4. Modelo de la trayectoria aparente del sol.

Al modelo de la figura 4 se le conoce como “trayectoria aparente del sol” ya que da la impresión de que el observador permanece fijo y el sol se desplaza alrededor de él navegando por una esfera que circunscribe la trayectoria solar. El punto de partida para el modelo se establece en la posición terrestre con latitud cero (ecuador) y la fecha el 21 de marzo o 22 de septiembre (equinoccio). Bajo estas condiciones la trayectoria solar es una circunferencia que pasa exactamente por el este y el oeste geográficos y abarca el máximo radio de la esfera celeste, como se muestra en la figura 4.

En éste modelo, la variación de los días del año o fecha se puede representar como una traslación de la trayectoria solar sobre el eje que pasa entre el norte y el sur, desplazándose entre los solsticios. Por otra parte, la posición de observación solo es afectada por la latitud y se representa en el modelo como una rotación alrededor del eje que pasa por el este y el oeste. Cabe mencionar que todas las posiciones geográficas con igual longitud observan la misma trayectoria solar. De esta manera el modelo de la trayectoria solar contempla las siguientes transformaciones:

- Registro de la fecha.
- Traslación
- Rotación

1.2.1 Registro de la fecha

La variación de los días del año o *Fecha* (mes y día) propicia que la trayectoria solar se mueva de norte a sur y viceversa. En el modelo, la *Fecha* primero debe ser el número entero de días al año N para que se puedan realizar operaciones matemáticas. Posteriormente, el número entero N debe relacionarse con un movimiento de traslación positivo para los días de verano y negativo para los días de invierno $MesP$. Además, para simplificar la transformación de *Fecha* a número entero N , se considera que febrero tiene 28 días. La relación entre el número entero N y el movimiento entre los solsticios se normaliza a $\pm A$ en la variable $MesP$ para posteriormente proporcionarle la escala conveniente al factor de traslación geométrica. La figura 5 muestra la relación descrita.

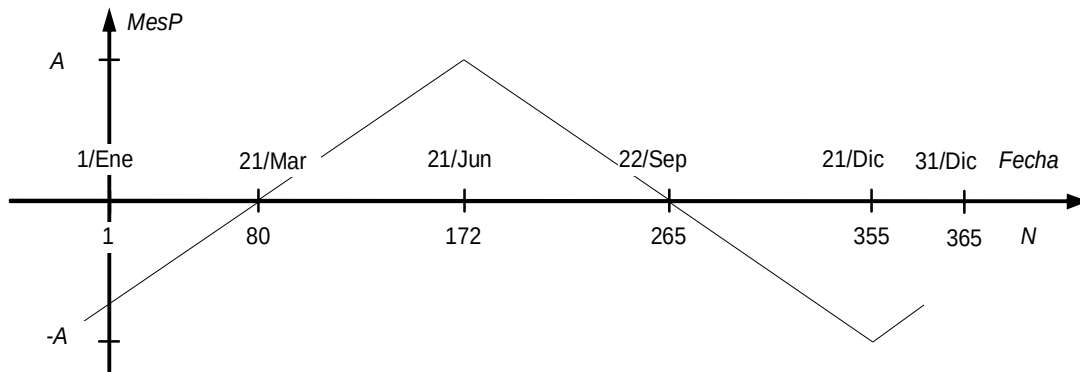


Figura 5. Relaciones Fecha-N-MesP.

Es importante mencionar que los cruces por cero de la gráfica ocurren precisamente en los equinoccios en donde la trayectoria solar cambia de dirección. De la figura 5 se obtienen las siguientes relaciones:

$$\begin{aligned}
 MesP &= \frac{A(N - 172)}{92} + A \text{ para } 1 \leq N \leq 172 \\
 MesP &= \frac{2A(N - 172)}{172 - 355} + A \text{ para } 173 \leq N \leq 355 \\
 MesP &= \frac{(13.869 - A)(N - 355)}{(355 - 365)} - A \text{ para } 356 \leq N \leq 365
 \end{aligned} \tag{2}$$

En resumen, la ecuación (2) se utiliza para trasladar geométicamente la trayectoria del sol dependiendo del día del año N .

Durante los solsticios ocurren los máximos desplazamientos de la variable $MesP$. Esto debido a que la tierra esta inclinada sobre su eje en un ángulo de 23.45° . Al proyectar el plano terrestre sobre los ejes coordenados se obtiene la representación de la figura 6.

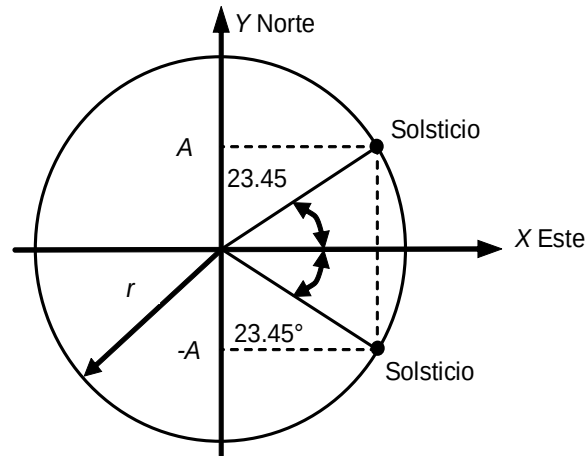


Figura 6. Solsticios.

De tal manera que la excursión máxima de la variable $MesP$, A , está relacionada con el ángulo de inclinación de la tierra y el radio r en donde se desee representar de la siguiente forma

$$A = r \text{ sen}(23.45^\circ) \quad (3)$$

1.2.2 Traslación de la trayectoria solar

Como ya se mencionó, la variación de la fecha introduce en el modelo una traslación de la trayectoria solar sobre el eje que une el norte con el sur. Adicionalmente ocurre que al trasladar la trayectoria solar cambia su radio r_c ya que la circunferencia de la trayectoria está circunscrita a la esfera celeste, como se aprecia en la figura 7.

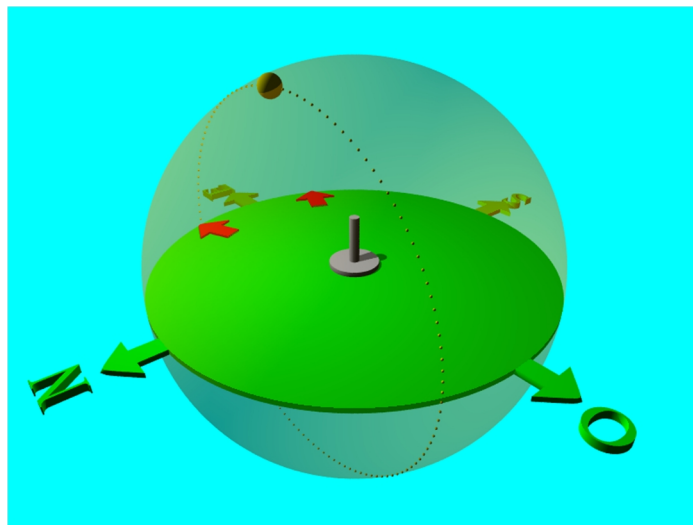


Figura 7. Traslación de la trayectoria aparente del sol.

Proyectando la trayectoria solar sobre los ejes coordenados se obtiene el arreglo que se muestra en la figura 8.

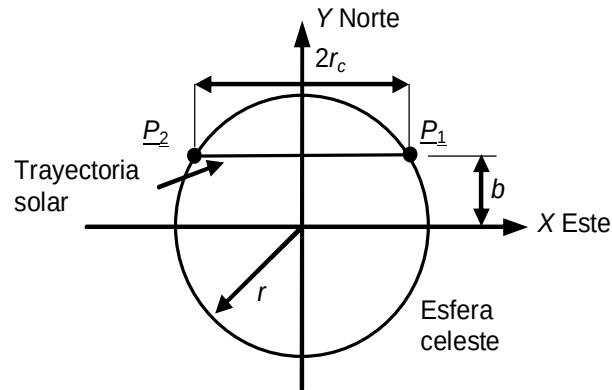


Figura 8. Radio de la trayectoria solar.

Para obtener el radio de la trayectoria solar r_c , primero se obtiene la ecuación de la circunferencia correspondiente a la esfera celeste

$$x^2 + y^2 = r^2$$

en donde r es el radio de la esfera celeste. Por otra parte, la ecuación de la recta que representa la trayectoria solar

$$y = b$$

en donde $b = MesP$, la magnitud que se trasladó la trayectoria debida a la fecha del año. Resolviendo simultáneamente las dos últimas ecuaciones para obtener los puntos de intersección P_1 y P_2

$$\begin{aligned} x &= \sqrt{r^2 - b^2} \\ P_1 &= [\sqrt{r^2 - b^2}, b] \\ P_2 &= [-\sqrt{r^2 - b^2}, b] \end{aligned}$$

El radio de la trayectoria r_c entonces es la mitad de la distancia entre P_1 y P_2

$$\begin{aligned} r_c &= \frac{1}{2} \sqrt{(-\sqrt{r^2 - b^2} - \sqrt{r^2 - b^2})^2 + (b - b)^2} \\ r_c &= \sqrt{r^2 - b^2} \\ r_c &= \sqrt{r^2 - MesP^2} \end{aligned}$$

Por lo tanto la trayectoria solar $\underline{T}$ se genera en el plano X-Z mediante la siguiente ecuación paramétrica de la circunferencia

$$\begin{aligned} T_x &= r_c \cos(index) \\ T_y &= 0 \\ T_z &= r_c \sin(index) \end{aligned}$$

Sustituyendo r_c

$$\begin{aligned}
 T_x &= \sqrt{r^2 - MesP^2} \cos(index) \\
 T_y &= 0 \\
 T_z &= \sqrt{r^2 - MesP^2} \sin(index)
 \end{aligned}
 \tag{4}$$

Con $0 \leq index \leq 2\pi$, como se muestra en la figura 9.

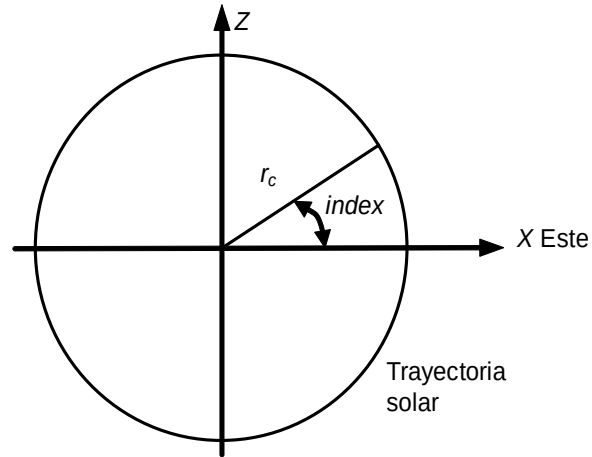


Figura 9. Trayectoria solar en el plano.

La ecuación (4) entonces se utiliza para generar la trayectoria solar que incluye la traslación y radio variable para los diferentes días del año, representados en la variable $MesP$.

1.2.3 Rotación de la trayectoria solar

La trayectoria aparente del sol, además de trasladarse, rota alrededor del eje que une al este con el oeste debido a la posición geográfica o latitud de observación, como se muestra en la figura 10.

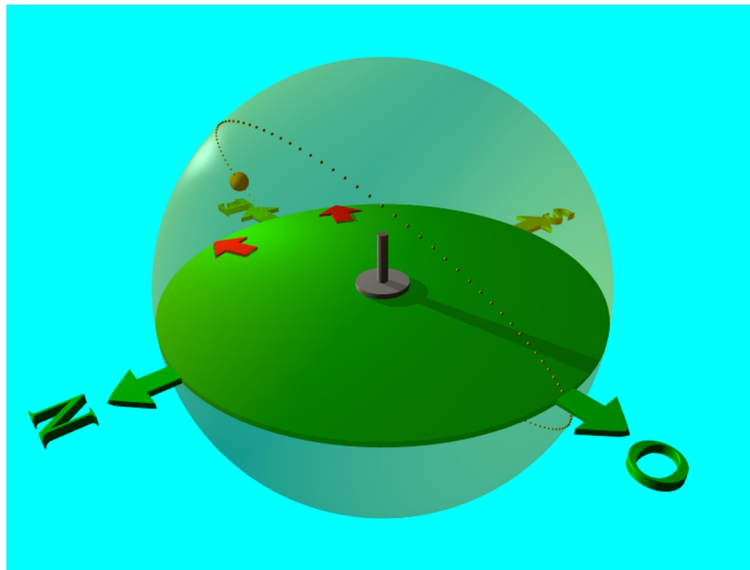


Figura 10. Rotación de la trayectoria aparente del sol.

Proyectando la trayectoria solar sobre los ejes coordenados se obtiene el arreglo que se muestra en la figura 11.

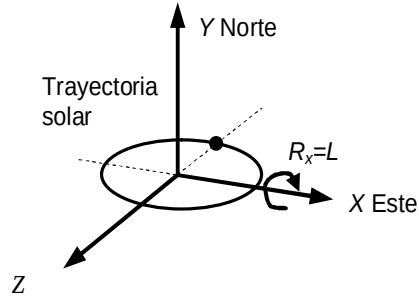


Figura 11. Trayectoria aparente del sol en los ejes coordenados.

La figura 11 muestra que la trayectoria solar rota alrededor del eje X R_x grados o radianes. En el modelo físico planteado la magnitud de la rotación es igual a la latitud de observación L .

1.3 Ajuste del tiempo

El tiempo solar aparente, AST (o tiempo solar local) en las longitudes occidentales está dado por

$$AST = LST + \left(4 \frac{\text{min}}{\text{deg}}\right) (LSTM - Long) + ET \quad (5)$$

Dónde:

LST = Tiempo local estándar u hora de reloj para aquella zona horaria (puede que tenga que ajustar el horario de verano, DST , es $LST = DST - 1$ hora).

$Long$ = Longitud local de la posición de interés.

$LSTM$ = Longitud local del tiempo estándar del meridiano dada por:

$$LSTM = 15^\circ \left(\frac{Long}{15^\circ} \right) \text{ redondeado a entero}$$

ET = Ecuación de tiempo.

La diferencia entre el tiempo solar verdadero y el tiempo solar significativo es que cambia continuamente día a día con un ciclo anual. Esta ecuación es conocida como la ecuación de tiempo. La ecuación de tiempo, ET en minutos, está dada aproximadamente por:

$$ET = 9.87 \sin(2D) - 7.53 \cos(D) - 1.5 \sin(D) \quad (6)$$

Donde:

$$D = 360^\circ \frac{(N - 81)}{365}$$

1.4 Ángulos solares

La hora angular, H , es el ángulo azimut de los rayos del sol causados por la rotación de la tierra y H puede calcularse a partir de:

$$H = \frac{(\text{Número de minutos después de la medianoche, } AST) - 720 \text{ minutos}}{4 \text{ min/deg}} \quad (7)$$

El ángulo horario tal como se define es negativo en la mañana y positivo en la tarde ($H=0^\circ$ al medio día). El ángulo de altitud solar β_1 es la altura angular aparente del sol en el cielo si se mira de frente. El ángulo cénit θ_z y su complemento, el ángulo de altitud β_1 , están dados por

$$\cos(\theta_z) = \sin(\beta_1) = \cos(L) \cos(\delta) \cos(H) + \sin(L) \sin(\delta) \quad (8)$$

Donde:

L = latitud (positivo en cualquier hemisferio) (0° a 90°).

δ = ángulo de declinación (negativo para el hemisferio sur) (-23.45° a $+23.45^\circ$)

H = ángulo horario

La altitud al mediodía es $\beta_1 = 90^\circ - L + \delta$. El sol sale y se pone cuando su altitud es 0° , no necesariamente cuando su ángulo horario es $\pm 90^\circ$. El ángulo horario al atardecer o amanecer, H_s , se puede encontrar usando la ecuación (8), donde $\beta_1 = 0$

$$\cos(H_s) = -\tan(L) \tan(\delta) \quad (9)$$

Donde H_s es negativa para el amanecer y positiva para el atardecer. Los valores absolutos de $\cos(H_s)$ mayor que la unidad se producen en las zonas árticas cuando el sol ni sale ni se esconde. El azimut solar, α_1 , es el ángulo lejos del sur.

$$\cos(\alpha_1) = \frac{\sin(\beta_1) \sin(L) - \sin(\delta)}{\cos(\beta_1) \cos(L)} \quad (10)$$

Donde α_1 es positiva hacia el oeste (tarde), y negativa hacia el este (mañana), y por lo tanto, el signo de α_1 debe coincidir con la del ángulo horario. La figura 12 ilustra los ángulos de altitud β_1 , azimut α_1 y horario H .

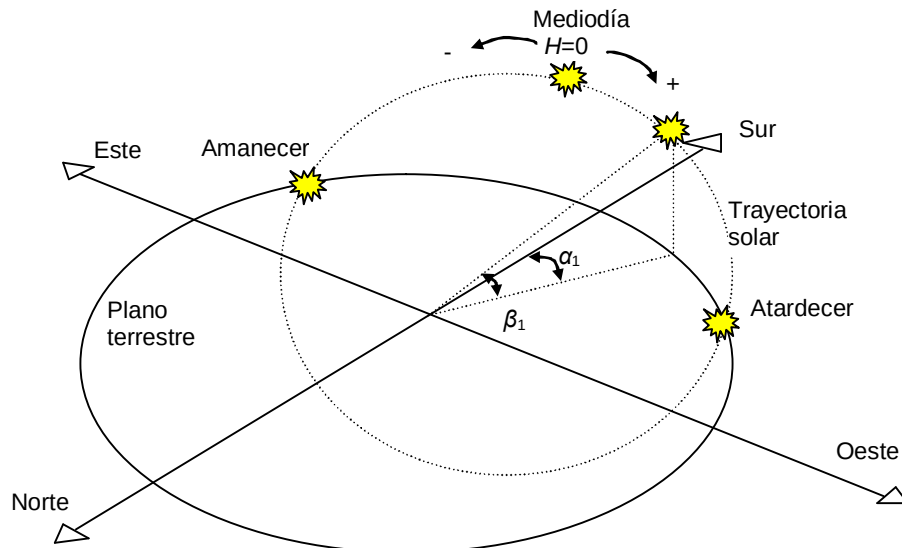


Figura 12. Ángulos de interés.

1.5 Ejemplo de aplicación

A manera de ejemplo, se muestra el cálculo de los ángulos solares de la figura 12 para la localidad de Phoenix, Arizona, USA que está localizada a 112° longitud oeste y 33.43° de latitud a las 8:00 a.m. para el 21 de julio.

De entrada, los datos del problema son:

$Long=112^\circ$
 $L=33.43^\circ$
 $LST=8:00$
 $Fecha=21$ julio

Dado que Phoenix no observa horario de verano, resulta innecesario hacer el ajuste al tiempo local. De la figura 5 se obtiene que para $Fecha=21$ de julio $N=202$. Sustituyendo en la ecuación (6)

$$D = 360^\circ \frac{(202 - 81)}{365} = 119.3$$

$$ET = 9.87 \sin(2 \times 119.3) - 7.53 \cos(119.3) - 1.5 \sin(119.3) = -6.05 \text{ min}$$

Sustituyendo $Long=112^\circ$, $LST=8:00$ y $ET=-6.05$ min en la ecuación (5)

$$LSTM = 15^\circ \left(\frac{112}{15^\circ} \right)_{\text{redondeado a entero}} = 105^\circ$$

$$AST = 8:00 + \left(4 \frac{\text{min}}{\text{deg}} \right) (105^\circ - 112^\circ) - 6.05 \text{ min}$$

$$AST = 8:00 - 28 \text{ min} - 6.05 \text{ min}$$

$$AST = 8:00 - 34.05 \text{ min} = 7:25.95$$

Sustituyendo $AST=7:25.95=7\text{hrs } 25.95\text{min}$ en la ecuación (7)

$$H = \frac{(7\text{hrs} \times 60 \frac{\text{min}}{\text{hrs}} + 25.95 \text{ min}) - 720 \text{ min}}{4 \text{ min/deg}} = -68.5^\circ$$

Sustituyendo $N=202$ en la ecuación (1)

$$\delta = 23.45^\circ \sin \left[\frac{202 + 284}{365} 360^\circ \right] = 20.44^\circ$$

Sustituyendo $L=33.43^\circ$, $H=-68.5^\circ$ y $\delta=20.44^\circ$ en la ecuación (8)

$$\beta_1 = \sin^{-1}(\cos(33.43^\circ) \cos(20.44^\circ) \cos(-68.5^\circ) + \sin(33.43^\circ) \sin(20.44^\circ))$$

$$\beta_1 = 28.61^\circ$$

Sustituyendo $\beta_1=28.61^\circ$, $L=33.43^\circ$ y $\delta=20.44^\circ$ en la ecuación (10)

$$\alpha_1 = \cos^{-1} \left(\frac{\sin(28.61^\circ) \sin(33.43^\circ) - \sin(20.44^\circ)}{\cos(28.61^\circ) \cos(33.43^\circ)} \right) = -96.7^\circ$$

Los ángulos obtenidos, α_1 , β_1 y H son suficientes para reproducir de manera ilustrativa la posición del sol como se muestra en la figura 12. Sin embargo, calculando la trayectoria solar (traslación y rotación) y las horas del amanecer y atardecer se obtendría una mejor representación.

Se inicia por definir el radio del plano terrestre que tendrá la representación gráfica, es decir $r=40$. Se sustituye este dato en la ecuación (3) para obtener la excursión máxima, A , de la variable $MesP$.

$$A = 40 \sin(23.45^\circ) = 15.92$$

Se sustituye $A=15.92$ y $N=202$ en la ecuación (2)

$$MesP = \frac{2 \times 15.92(202 - 172)}{172 - 355} + 15.92 = 10.7$$

Sustituyendo $MesP=10.7$ y $r=40$ en la ecuación (4) se obtiene la magnitud de las traslaciones en x y y de la representación gráfica

$$\begin{aligned} T_x &= \sqrt{40^2 - 10.7^2} \cos(index) \\ T_y &= 0 \\ T_z &= \sqrt{40^2 - 10.7^2} \sin(index) \end{aligned}$$

Con $0 \leq index \leq 2\pi$, para generar la trayectoria completa.

De la figura 11 se obtiene directamente que la rotación de la trayectoria solar es directamente la latitud $R_x=L=33.43^\circ$.

Para obtener el ángulo horario del amanecer para el 21 de julio en Phoenix, se sustituye $L=33.43^\circ$ y $\delta=20.44^\circ$ en la ecuación (9).

$$H_s = -\cos^{-1}(-\tan(33.43^\circ) \tan(20.44^\circ)) = -104.24^\circ$$

Para encontrar su correspondiente horario AST se sustituye $H_s=-104.24^\circ$ en la ecuación 7

$$\begin{aligned} -104.24^\circ &= \frac{(Amanecer_{AST}) - 720 \text{ minutos}}{4 \text{ min/deg}} \\ Amanecer_{AST} &= \left(-104.24^\circ \times 4 \frac{\text{min}}{\text{deg}}\right) + 720 \text{ min} \\ Amanecer_{AST} &= 303.04 \text{ min} = 5.051 \text{ hrs} = 5:03 \end{aligned}$$

El horario local del amanecer se obtiene al sustituir $Amanecer_{AST}=5:03$, $LSTM=105^\circ$, $Long=112^\circ$ y $ET=-6.05 \text{ min}$ en la ecuación (5)

$$\begin{aligned} 5:03 &= Amanecer_{LST} + \left(4 \frac{\text{min}}{\text{deg}}\right)(105^\circ - 112^\circ) - 6.05 \text{ min} \\ Amanecer_{LST} &= 5:03 - \left(4 \frac{\text{min}}{\text{deg}}\right)(105^\circ - 112^\circ) + 6.05 \text{ min} \\ Amanecer_{LST} &= 5:03 - (-28 \text{ min}) + 6.05 \text{ min} = 5:37 \end{aligned}$$

Para obtener el ángulo horario del atardecer para el 21 de julio en Phoenix, se sustituye $L=33.43^\circ$ y $\delta=20.44^\circ$ en la ecuación (9).

$$H_s = \cos^{-1}(-\tan(33.43^\circ) \tan(20.44^\circ)) = 104.24^\circ$$

Para encontrar su correspondiente horario AST se sustituye $H_s=104.24^\circ$ en la ecuación 7

$$\begin{aligned} 104.24^\circ &= \frac{(Atardecer_{AST}) - 720 \text{ minutos}}{4 \text{ min/deg}} \\ Atardecer_{AST} &= \left(104.24^\circ \times 4 \frac{\text{min}}{\text{deg}}\right) + 720 \text{ min} \\ Atardecer_{AST} &= 1136.96 \text{ min} = 18.95 \text{ hrs} = 18:57 \end{aligned}$$

El horario local del atardecer se obtiene al sustituir $Atardecer_{AST}=18:57$, $LSTM=105^\circ$, $Long=112^\circ$ y $ET=-6.05 \text{ min}$ en la ecuación (5)

$$18:57 = \text{Atardecer}_{LST} + \left(4 \frac{\text{min}}{\text{deg}}\right) (105^\circ - 112^\circ) - 6.05\text{min}$$

$$\text{Atardecer}_{LST} = 18:57 - \left(4 \frac{\text{min}}{\text{deg}}\right) (105^\circ - 112^\circ) + 6.05\text{min}$$

$$\text{Atardecer}_{LST} = 18:57 - (-28\text{min}) + 6.05\text{min} = 19:31$$

La tabla 1 muestra el ejemplo desarrollado y contiene los resultados relevantes para el presente trabajo. El directorio CodigoFuente/MatLab del CD adjunto contiene el programa en Matlab que obtiene los resultados de la tabla 1.

Información de entrada		Datos de salida	
Longitud $Long$	112°	Azimut α_1	-96.7°
		Altitud β_1	28.61°
Latitud L	33.43°	Ángulo horario H	-68.5°
		Amanecer	5:37
Hora LST	8:00	Atardecer	19:31
		Rotación R_x	33.43°
Fecha	21 de julio	MesP	10.7
		Traslación T_x	$\sqrt{40^2 - 10.7^2} \cos(index)$
Radio r	40	Traslación T_y	0
		Traslación T_z	$\sqrt{40^2 - 10.7^2} \sin(index)$

Tabla 1 Ejemplo de aplicación (resultados numéricos).

Con los resultados de la tabla 1 se genera la figura 13 que muestra de manera gráfica el ejemplo de aplicación desarrollado. El directorio CodigoFuente/PovRay del CD adjunto contiene el código fuente para reproducir la figura 13 a partir de geometrías básicas [POV-Ray, 2013]. Adicionalmente, se muestra el código fuente para generar las figuras 2 y 3. En la figura 13 se muestra el modelo de la trayectoria aparente del sol y un panel que es orientado a la posición del sol con el propósito de captar la máxima irradiación en la localidad, día y hora especificados para el ejemplo.

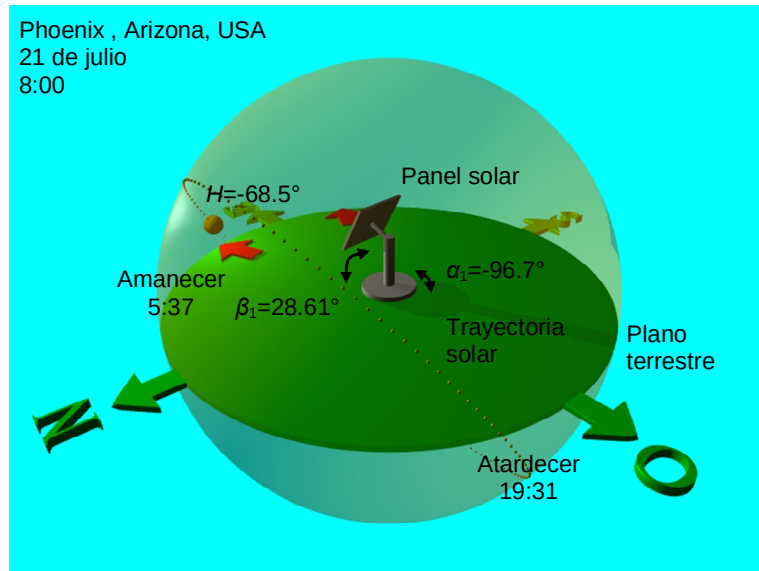


Figura 13. Ejemplo de aplicación (ambiente gráfico).

2 Descripción del sistema electromecánico

La figura 14 muestra el esquema general del sistema electromecánico desarrollado para el seguimiento de la trayectoria solar.

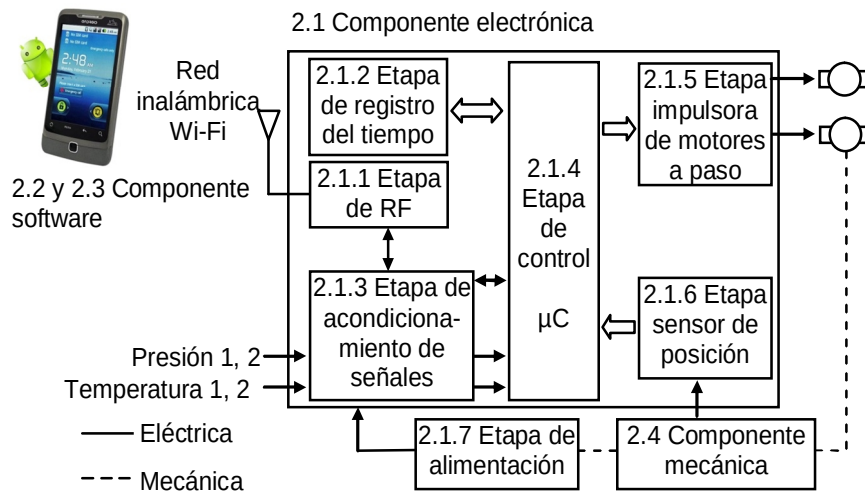


Figura 14. Esquema general del seguidor solar.

El método de posicionamiento que desempeña el esquema de la figura 14 (seguidor activo) es el seguimiento de la trayectoria del sol con base en el cálculo de los ángulos solares en función del registro del tiempo, que se ejecutan en la componente electrónica por medio de un microcontrolador, μC . Se puede realizar el seguimiento solar sobre la base del cálculo de los ángulos de altitud y azimut del sol y girar la componente mecánica donde se encuentra montada la etapa de alimentación (celda fotovoltaica) y orientarla de frente al sol, como se describe en 1.4 y 1.5. De esta manera se puede obtener la máxima radiación solar y aumentar la eficiencia en la generación de energía. El seguidor solar es un sistema autoalimentado, es decir, el sistema no necesita ningún tipo de suministro de energía del mundo exterior, sino sólo la luz del sol.

El algoritmo de control incluye dos subrutinas importantes. Los cálculos solares son para generar los ángulos de altitud y azimut del sol, mientras que los cálculos de control de pulsos para los motores paso a paso son para determinar el envío de las señales de pulsos de acuerdo con la diferencia angular. El registro de fecha y hora requerido para el cálculo de los ángulos es proporcionado por el circuito integrado DS1307 en la etapa de registro del tiempo, que proporciona al algoritmo de control el año, mes, día, hora, minuto y segundo en tiempo real.

Se utiliza una aplicación para teléfono celular, desarrollada como parte del presente trabajo, con fines de configuración y monitoreo de diversas variables con las cuales trabaja el sistema electromecánico. El medio de enlace entre la aplicación y el sistema electromecánico es a través de una red inalámbrica Wi-Fi.

Como se observa en el esquema de la figura 14, el proyecto involucra varias etapas para su funcionamiento, que van desde una aplicación para teléfono celular hasta un sistema electromecánico. En la siguiente descripción las componentes se organizan de la siguiente forma:

- Componente electrónica.
- Componente software.
- Componente mecánica.

A continuación se describen cada una de las componentes que integran el presente proyecto y que fueron desarrolladas a partir de herramientas básicas.

2.1 Componente electrónica

El circuito electrónico para el seguimiento solar se compone de las siguientes etapas, figura 14.

- Etapa de control. En ésta etapa, un microcontrolador implementa los algoritmos de control, cálculo de los ángulos solares y adquisición de señales para orientar al mecanismo hacia la trayectoria solar. Además se controla la transmisión y recepción de datos para las interfaces con las etapas de radiofrecuencia y registro del tiempo.
- Etapa de radiofrecuencia. Esta etapa se utiliza para establecer la vía de comunicación para la interface con el usuario, principalmente para monitorear y configurar parámetros de operación de la componente electrónica de manera inalámbrica.
- Etapa de registro del tiempo. En esta etapa se lleva a cabo el registro del tiempo (mes, día y hora) independientemente del microprocesador que impulsa la etapa de control. De esta manera se asegura un registro del tiempo confiable y no interrumpible ya que no depende ni del algoritmo del microcontrolador ni de la energía principal que alimenta a la componente electrónica.
- Etapa de acondicionamiento de señales. Se implementa con circuitos electrónicos sencillos que permiten el acondicionamiento de dos sensores de temperatura, dos de presión y dos niveles lógicos. Estas señales pueden eventualmente representar variables de interés en un calentador solar, no implementado en el presente trabajo.
- Etapa impulsora de motores a paso. Esta etapa se implementa con circuitos impulsores de corriente directa que permiten acoplar las señales del microcontrolador a los embobinados de los motores a paso que posicionan la componente mecánica.
- Etapa para los sensores de posición. Está conformada por circuitos opto-electrónicos de encendido/apagado para establecer la posición inicial de los motores a paso.
- Etapa de alimentación. Esta etapa principalmente consta de un panel solar fotovoltaico montado sobre la componente mecánica. Se encarga de transformar la energía solar en eléctrica para alimentar la componente electrónica en conjunto.

El anexo A y el directorio CADElectronica del CD adjunto contienen los detalles de la implementación de la componente electrónica. A continuación, una descripción detallada de las diversas etapas.

2.1.1 Etapa de radiofrecuencia

La etapa de radio-frecuencia, RF, tiene como base un módulo inalámbrico RN-131-EK [Roving Networks, 2012] que levanta una red “Punto de Acceso” (software Acces Point o soft AP) a la cual cualquier dispositivo móvil con conexión Wi-Fi puede asociarse. En el modo soft AP, el módulo es un servidor DHCP que distribuye direcciones IP a los clientes que se asocian a su red. Una vez asociado el dispositivo inalámbrico a la red, el módulo RN-131-EK es un “túnel de datos” sobre una conexión TCP, lo cual quiere decir que los datos que se reciben o transmiten en el medio inalámbrico, son transmitidos o recibidos en el puerto serie asíncrono UART “Universal Asynchronous Receiver Transmitter” que comunica el módulo con el microcontrolador. En la figura 15 se muestra la configuración electrónica del módulo inalámbrico RN-131-EK utilizado en el proyecto.

El voltaje de polarización del RN-131 de es de 3.3 volts pero tiene la ventaja que dentro de su tarjeta de desarrollo, RN-131-EK, cuenta con un regulador de voltaje que permite proporcionarle hasta un voltaje de 5V. La comunicación que este dispositivo establece con el microcontrolador lo hace mediante el puerto serie asíncrono UART. El modulo RN-131-EK además cuenta con tres leds integrados que indican el estado en el cual se encuentra dicho dispositivo. En la tabla 2 se muestra sus posibles estados. Con esta tabla es posible saber si la conexión con el dispositivo ha sido exitosa o no, además brinda información de manera visual sin tener que medir ningún parámetro en el circuito.

El módulo RN-131 tiene dos modos de operación posibles: modo de datos (por defecto) y modo de comandos. En el modo de datos, el módulo de Wi-Fi es esencialmente un tubo de datos. Cuando el

modulo recibe datos a través de Wi-Fi, quita los *headers* y *trailers* del TCP/IP y pasa los datos del usuario a la UART. Cuando los datos se escriben en la UART, el módulo inalámbrico construye un paquete TCP/IP y lo envía hacia afuera a través de Wi-Fi. Por lo tanto, todo el proceso de envío/recepción de datos al host es transparente para el microcontrolador.

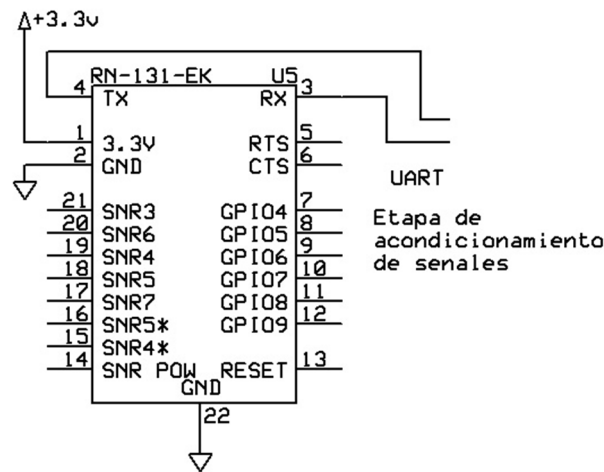


Figura 15. Configuración del módulo inalámbrico de Wi-Fi.

Condición	Led rojo	Led amarillo	Led Verde
Encendido	-	-	Conexión sobre TCP
Parpadeo rápido	Sin asociación	Transferencia de datos Tx o Rx en modo túnel	Sin dirección IP
Parpadeo lento	-	-	Con dirección IP
Apagado	Con asociación	-	-

Tabla 2. Posibles condiciones del módulo RN-131-EK.

Cuando se establece el modo de comandos, ingresando la cadena “\$\$\$”, se pueden cambiar los parámetros de operación de la red soft AP al cambiar el nombre del módulo, o sea la SSID, y también para poder establecer una contraseña para restringir que dispositivo puede o no conectarse.

En la figura 16 se aprecia la comunicación que establece el modulo inalámbrico. En esencia este módulo inalámbrico establece dos tipos de comunicaciones, UART y Wi-Fi, lo cual le permite ser empleado en toda una gama de aplicaciones para dispositivos móviles y fijos que requieran una comunicación eficaz.

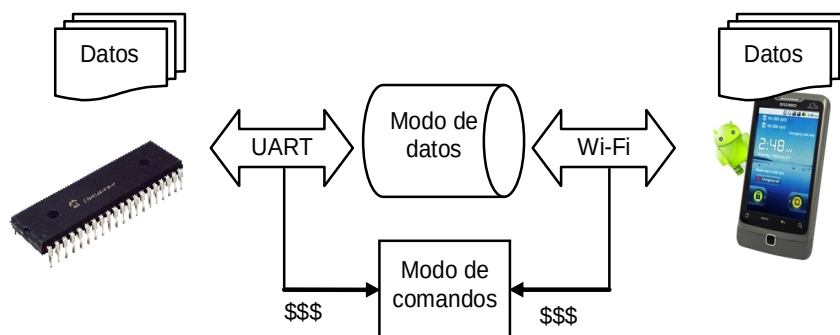


Figura 16 Modos de operación del módulo inalámbrico RN-131.

2.1.2 Etapa de registro del tiempo

Para el registro del tiempo, es decir la obtención de la hora y fecha local, se empleó el circuito integrado DS1307 [Maxim Integrated, 2015]. Este integrado proporciona segundos, minutos, horas, días, meses y años en tiempo real (todos estos datos son necesarios para los cálculos astronómicos de alta precisión que se han programado dentro del microcontrolador).

En la figura 17 se muestra la configuración del reloj de tiempo real empleado.

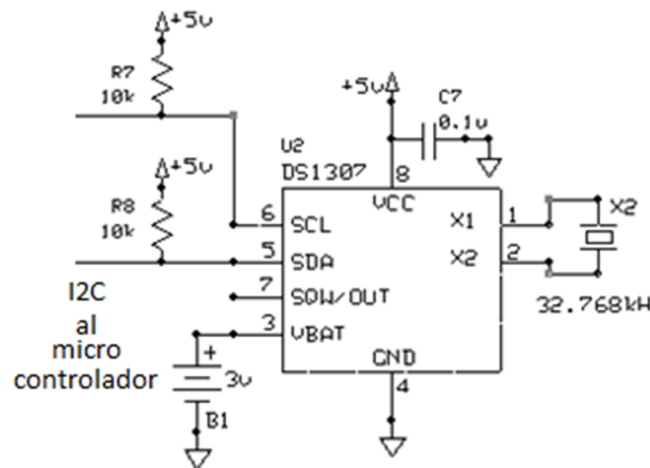


Figura 17. Configuración del reloj de tiempo real, DS1307.

El DS1307 o RTC (Real Time Clock) se polariza con un voltaje de 5 volts pero tiene un circuito de detección de potencia incorporado que detecta fallas de energía y cambia automáticamente a la fuente de la batería B1 (pila de litio, 3V), si esta se ve interrumpida, que le puede proporcionar energía al dispositivo hasta por 10 años sin interrupción. Su frecuencia de trabajo es de 32.768 KHz, X2, y envía los datos solicitados por puerto serie I²C. La configuración requiere un par de resistencias pull-up de 10kΩ para cada línea con el fin de poder transmitir y recibir correctamente.

2.1.3 Etapa de acondicionamiento de señales

En la componente electrónica se requieren adecuar las señales que generan los sensores de temperatura a los niveles de operación del convertidor analógico a digital del microcontrolador. Por otra parte se requiere adecuar los niveles lógicos TTL de 3V del puerto UART del módulo RN-131-EK a los niveles lógicos TTL de 5V del puerto UART del microcontrolador.

Acondicionamiento de la señal de temperatura

El sensor LM35 es el encargado de proporcionar y medir las señales de temperatura. El LM35 no requiere de circuitos adicionales para calibrarlo externamente. La baja impedancia de salida, su salida lineal y su precisa calibración hace posible que esté integrado sea instalado fácilmente en la componente electrónica. Debido a su baja corriente de alimentación se produce un efecto de auto calentamiento muy reducido. Se encuentra en diferentes tipos de encapsulado, el más común es el TO-92, utilizado ampliamente por transistores de baja potencia.

El sensor LM35 proporciona una salida lineal de 10mV por grado centígrado medido. De esta forma no se requiere hacer una conversión de temperatura, puesto que el programa desarrollado maneja grados Celsius. Además este dispositivo cuenta con un rango de funcionamiento de -55 a 150 grados centígrados, lo que permite tener un mayor grado de lectura. En la figura 18 se muestra la configuración del sensor de temperatura LM35.

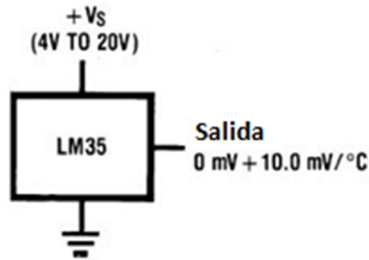


Figura 18. Configuración del sensor de temperatura LM35.

El sensor tiene una polarización de 4 hasta 20 volts para su correcto funcionamiento. Para interpretar el voltaje suministrado por el sensor para cada grado centígrado leído solo basta utilizar la siguiente fórmula.

$$Temp(^{\circ}C) = \frac{V_{sensor} \times 5 \times 100}{1023}$$

Acondicionamiento de niveles lógicos

El voltaje de polarización de la etapa de control es de 5 volts mientras que la etapa de RF es de 3.3 volts. Por esta diferencia de voltaje en el funcionamiento, la comunicación se establece por medio del convertidor de nivel BOB-08745 [SparkFun, 2015]. La configuración utilizada se muestra en la figura 19.

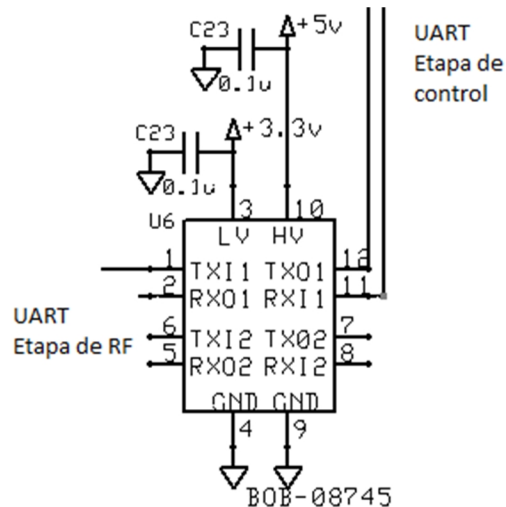


Figura 19. Convertidor de nivel BOB-08745.

El convertidor de nivel BOB-08745 hace posible la comunicación entre dispositivos que trabajan con 5 y 3.3 volts. El convertidor tiene que ser alimentado por las dos fuentes de tensión (de alta tensión y baja tensión) que se están utilizando. Alto voltaje (5V) para el pin 'HV', baja tensión (3.3 V) para 'LV', y la toma de tierra del sistema para el pin 'GND'. Cada convertidor de nivel tiene la capacidad de convertir 4 pines en la parte alta y 4 pines en la parte baja con dos entradas y dos salidas previstas para cada lado.

2.1.4 Etapa de control

En la etapa de control se encuentra el microcontrolador PIC18F4550 de 8 bits [Microchip, 2007]. En la figura 20 se muestra la configuración y puertos usados.

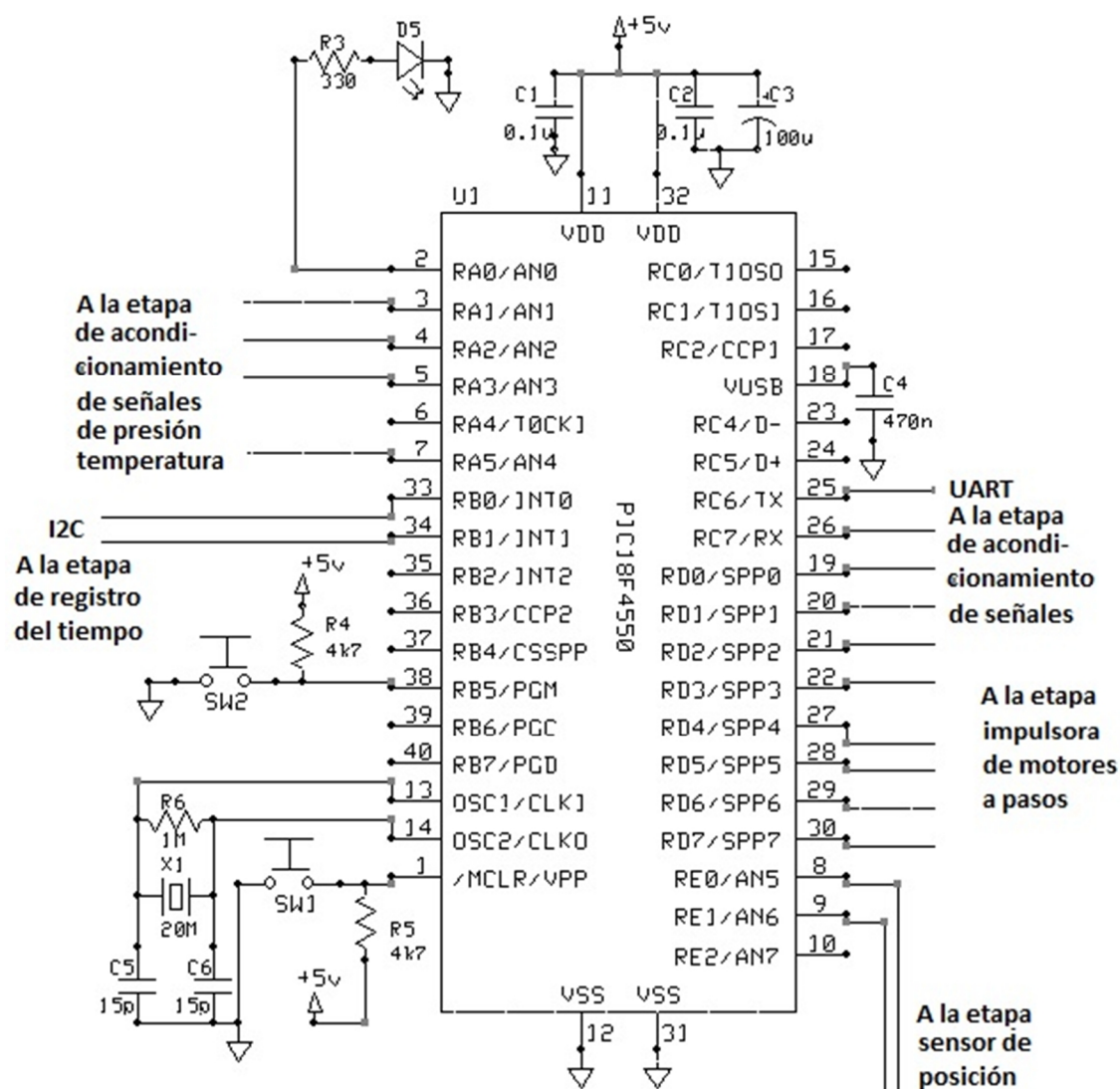


Figura 20. Configuración del PIC18F4550.

Dentro de las comunicaciones que establece el microcontrolador se encuentra la comunicación UART. Esta comunicación solo necesita de dos líneas, la de recepción de datos o Rx en RC7 y la de transmisión de datos o Tx en RC6. El UART de la etapa de control se utiliza para establecer comunicación con la etapa de RF a través del circuito de acondicionamiento de los niveles lógicos.

El microcontrolador cuenta con la comunicación I²C “Inter-Integrated Circuit” que solo necesita dos líneas: la de datos o SDA en RB0 y la del reloj o SCL en RB1. La vía de comunicación I²C se utiliza para transferir la información de la fecha y hora que se genera en la etapa de registro del tiempo hacia el microcontrolador en donde se calculan los ángulos solares.

Para la interpretación de las señales analógicas generadas por los sensores de presión y temperatura se hace uso de los convertidores analógicos digitales del microcontrolador en RA1, RA2, RA3 y RA5. De esta manera el microcontrolador puede medir, procesar y mandar la información de temperatura o presión si le es solicitado por la interface de usuario.

El puerto RD del microcontrolador, que tiene la opción de ser utilizado como entrada o salida analógica, es el puerto encargado de mandar la señal de pulsos hacia la etapa impulsora de los motores a pasos,

para cada uno de los dos motores que generan la trayectoria solar a través del cálculo de los ángulos solares.

Para detectar la posición inicial en la componente mecánica (cero altitud y azimuth), se habilitaron los puertos RE0 y RE1 como entradas para detectar las señales que envía los dos sensores de posición.

2.1.5 Etapa impulsora de motores a paso

En esta etapa se encuentran los drivers de potencia M54532P. El driver M54532P tiene en su interior un arreglo de cuatro transistores Darlington con los diodos de fijación internos [Mitsubishi, 1999]. Los circuitos están hechos de transistores NPN. Por estas características, puede manejar una corriente máxima de 1.5 Amperes, lo que lo hace idóneo para el manejo de pequeños motores a pasos. En la figura 21 se muestra la configuración de este dispositivo.

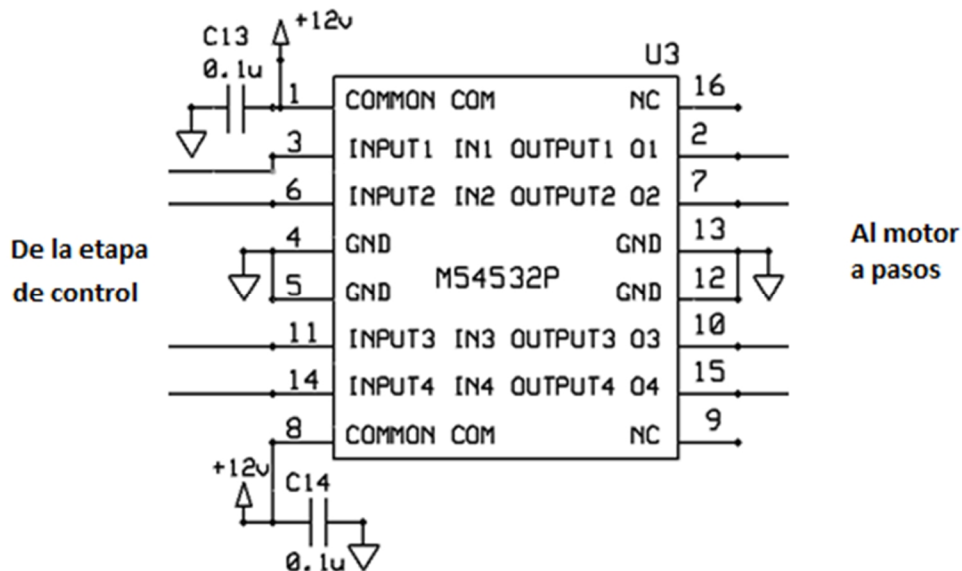


Figura 21. Configuración del dispositivo M54532P.

Como se puede observar en la figura 21 el dispositivo cuenta con 4 entradas y 4 salidas con lógica negada, el cual permite controlar motores unipolares de 5, 6 u 8 hilos sin problema alguno. Por sus diodos internos de fijación se ahorra espacio para el desarrollo de la tarjeta. El circuito de la figura 21 se implementa dos veces en la componente electrónica, uno de ellos se utiliza para motor del ángulo de altitud y el otro para el motor de azimuth.

2.1.6 Etapa para el sensor de posición

Para el posicionamiento de la componente mecánica se requiere de sensores adecuados que proporcionen señales de información cuando se encuentre en la posición inicial requerida (cero).

Para la posición a cero del sistema se emplearon dos opto-interruptores H21A1, uno por cada ángulo calculado, altitud y azimuth, que mandan una señal de 0 o 5 volts.

El funcionamiento de los opto-interruptores H21A1 consiste en un diodo emisor de infrarrojos junto con un fototransistor de silicio en una carcasa de plástico. El sistema de envasado está diseñado para optimizar la resolución mecánica, la eficiencia de acoplamiento, el rechazo de la luz ambiental, el costo y la fiabilidad. La brecha en la carcasa proporciona un medio de interrumpir la señal con un material opaco, el

cambio de la salida es de un "ON" a un estado "OFF". En la figura 22 se puede observar la configuración de los opto-interruptores.

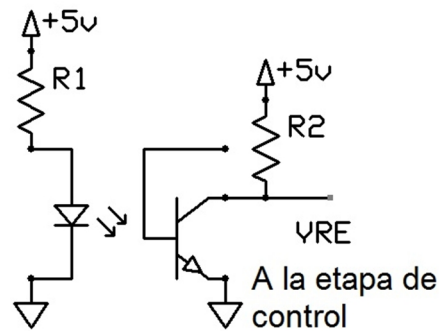


Figura 22. Configuración del opto-interruptor para el posicionamiento del sistema.

De esta manera cuando se detecta el cambio de voltaje, de 0 a 5 volts, "ON" a "OFF", el microcontrolador empleado interpreta estas señales y brinda una posición inicial al sistema, dependiendo de cuál sea el caso (azimut o altitud).

2.1.7 Etapa de alimentación de energía

La etapa de alimentación está conformada por una celda solar, un controlador de carga solar y una batería recargable. Este subsistema genera y proporciona la energía necesaria para el correcto funcionamiento del seguidor solar. La figura 23 muestra el diagrama de conexión.

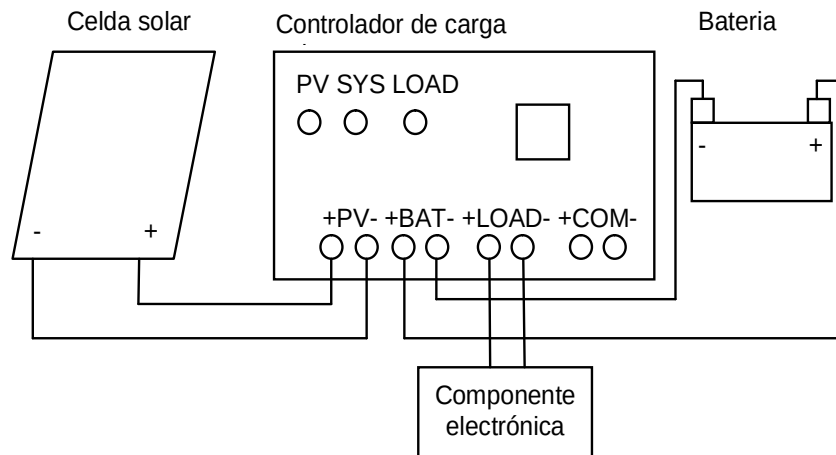


Figura 23. Conexión de la etapa de alimentación.

Pruebas hechas en el laboratorio demostraron que la celda solar proporciona un voltaje de alrededor de 17 V y una corriente de 0.5 A, esto cuando se encuentra frente al sol. También se realizaron pruebas cuando se encontraba bajo sombra, dando un voltaje de alrededor de 10V y una corriente de 0.2 A. Dichas pruebas se realizaron en un día soleado y despejado.

En condiciones adecuadas la batería tarda 6 horas en cargarse por completo, conectada al circuito electrónico, lo que proporciona alrededor de 10 horas más de energía, quedando un remanente que se puede utilizar en otras tareas.

Para saber el estado en que se encuentra la batería, la etapa de alimentación cuenta con un controlador de carga solar que indica, por medio del led marcado como SYS, el estado en que se encuentra. A continuación se muestra la tabla 3 del sistema led.

LED Color	Estado del sistema	LED Color	Estado del sistema
Verde	Normal	Naranja	Batería baja
Verde parpadeo	Batería llena	Rojo	Batería descargada
Verde parpadeo rápido	Batería sobrecargada	Rojo parpadeo rápido	Temperatura alta
Rojo parpadeo	Sobrecarga	Naranja parpadeo rápido	Fusible quemado

Tabla 3. Descripción del sistema led.

Con este sistema led en el controlador de carga solar, resulta sencillo saber el estado en que se encuentra la batería. De esta manera se sabe si la batería se encuentra cargada totalmente o viceversa. Además con la tabla 3 se sabe el estado en que se encuentra el sistema, ya que indica cuando hay una sobrecarga.

2.1.8 Firmware del microcontrolador

El firmware del microcontrolador contiene los algoritmos de control y medición que permiten mover los motores a paso y capturar las señales de interés. El directorio CódigoFuente/PIC del CD adjunto contiene los detalles de la implementación. El firmware está compuesto por una rutina de inicio, una rutina principal y una interrupción.

Rutina de inicio

La rutina de inicio tiene como objetivo posicionar la componente mecánica. Con la obtención de las señales de la etapa del sensor de posición se establece un punto de inicio en la parte mecánica del sistema, el cual será el punto de referencia para la ejecución de los ángulos solares. Adicionalmente, si el tiempo actual de la etapa de registro del tiempo, AST, corresponde con el intervalo de radiación solar, se calculan los ángulos de azimut y altitud para orientar el panel solar, de lo contrario se inverte. La figura 24 muestra las actividades que se ejecutan como parte de esta rutina inicial.

Rutina principal

La figura 25 muestra la rutina principal del programa. Establecido el punto de inicio en el sistema mecánico, se ejecuta un ciclo indefinido en donde se encuesta el paquete de datos recibido por Wi-Fi en la etapa de radiofrecuencia y a su vez transmitido por el puerto UART al microcontrolador. El paquete está compuesto por una serie de instrucciones a ejecutar. Algunas posibles instrucciones de envío de información desde la componente electrónica a la interface de usuario mediante Wi-Fi son:

- Envía la medición de la temperatura T1 almacenada en la variable ValorCAD0.
- Envía la medición de la temperatura T2 almacenada en la variable ValorCAD1.
- Envía la posición del ángulo azimut, alfa, por puerto UART.
- Envía la posición del ángulo altitud, beta, por puerto UART.
- Envía la fecha y hora del circuito electrónico. Usada con fines de depuración.

Los datos a actualizar se envían desde la interface de usuario hacia la componente electrónica mediante Wi-Fi y son:

- Posición deseada del motor 1 en azimut (alfa). Usada con fines de depuración.
- Posición deseada del motor 2 en altitud (beta). Usada con fines de depuración.
- Configuración de la fecha y hora local.

- Configuración de longitud y latitud.

Interrupción del TIMER0

La rutina mostrada en la figura 26 inicia cuando el TIMER0 del PIC18F4550 se desborda al cabo de 60s desde que se establece el "inicio" en la componente mecánica del sistema. Esto genera una interrupción e inicia una rutina que realiza la obtención de la hora local, a través de la comunicación I²C con la etapa de registro del tiempo, con el fin de calcular los ángulos astronómicos y de esta forma actualizar la posición del panel fotovoltaico.

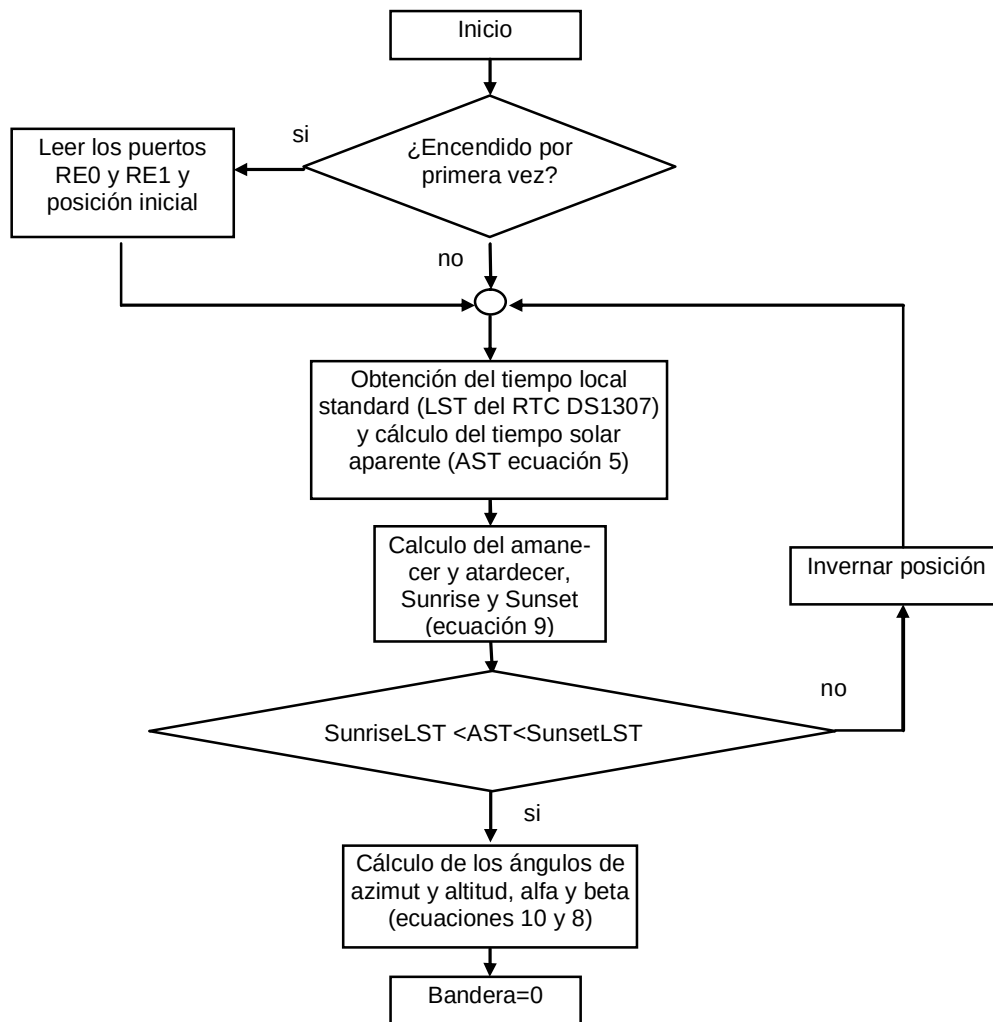


Figura 24. Rutina de inicio.

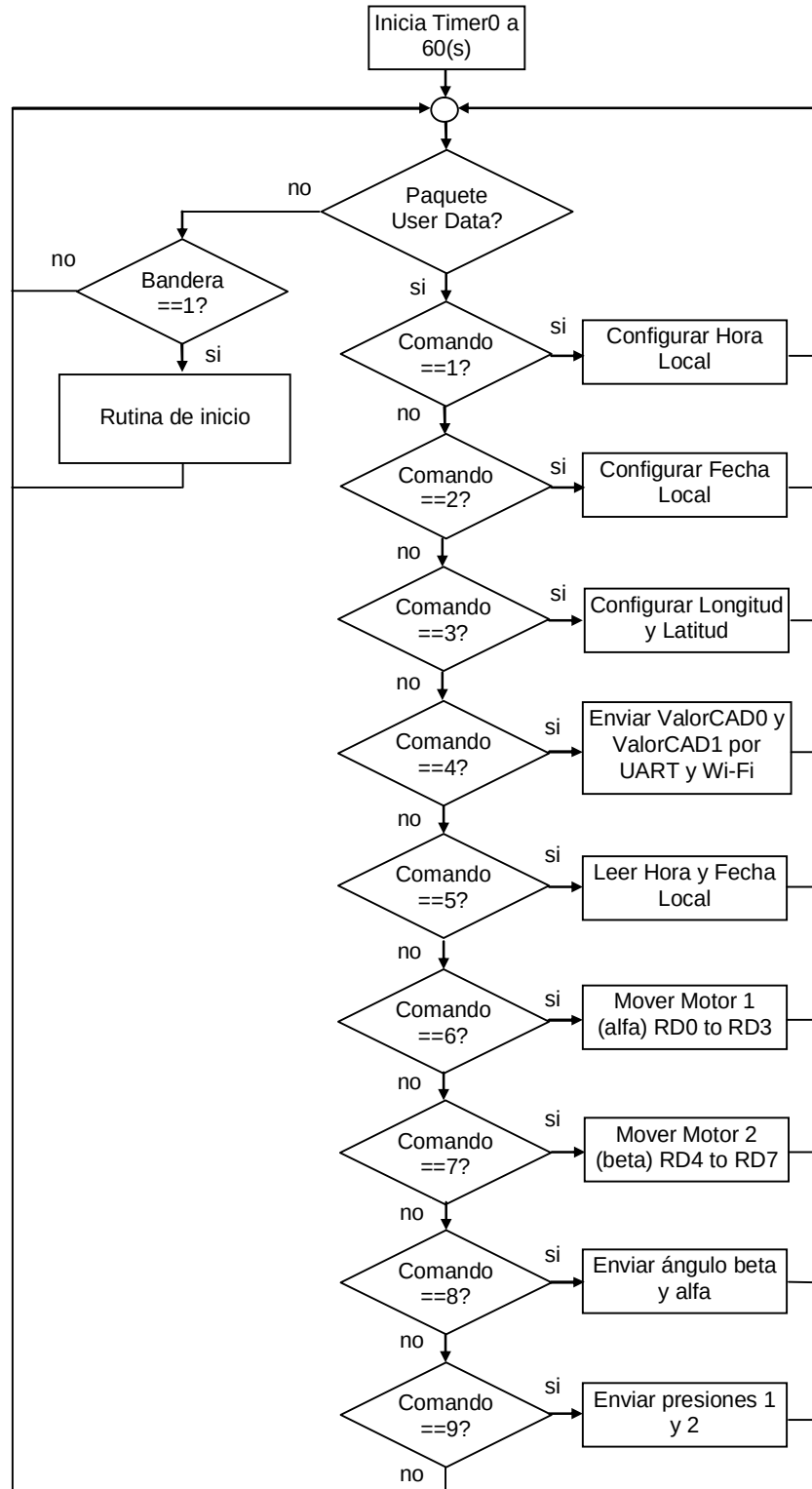


Figura 25. Rutina principal.

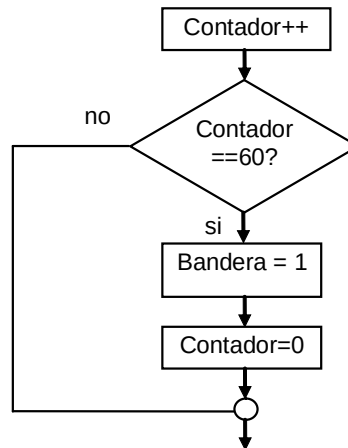


Figura 26. Interrupción del TIMER0.

2.2 Descripción del software en el ámbito del operador

La aplicación para teléfono celular inteligente (Smartphone) que implementa la interface de usuario (UI) para operar el sistema electromecánico de seguimiento solar se denomina “Seguidor solar” y está contenida en el directorio Programa del CD anexo. La pantalla principal de la aplicación se muestra en la figura 27.



Figura 27. Pantalla principal de la aplicación “Seguidor solar”.

La aplicación fue desarrollada respetando el concepto de “dinamismo” para las UI’s de los dispositivos Android, es decir, adaptándose de forma natural al contenido de la información que genera la aplicación con el menor número posible de controles o recursos. En el inicio de la aplicación se muestra el contenido de la pestaña central “Ángulos Solares”. En ésta se despliegan dos indicadores analógicos que contienen la información actual de posicionamiento en azimuth (Alfa) y altitud (Beta). La información se obtiene cada cinco segundos del sistema electromecánico mediante la interface inalámbrica Wi-Fi, no obstante que el sol se desplace 1° cada 4 min.

Al presionar la pestaña del extremo izquierdo se muestra la información de temperatura en los sensores del sistema electromecánico, como se muestra en la figura 28. La temperatura en los canales de conversión AN1 y AN2 de la componente electrónica se despliegan de forma analógica y también su valor es refrescado cada cinco segundos.



Figura 28. Pantalla para el despliegue de temperatura.

Al presionar la pestaña del extremo derecho se muestra la información de los convertidores analógico a digital AN3 y AN4 que pueden representar variables de presión en el sistema electromecánico, como se muestra en la figura 29. De igual forma que en las pantallas anteriores, la información se refresca automáticamente cada cinco segundos.

Las pantallas de las figuras 27 a 29 no requieren que el operador envíe instrucción alguna, ya que su operación es gobernada por un temporizador que refresca de manera automática los valores de los indicadores analógicos.

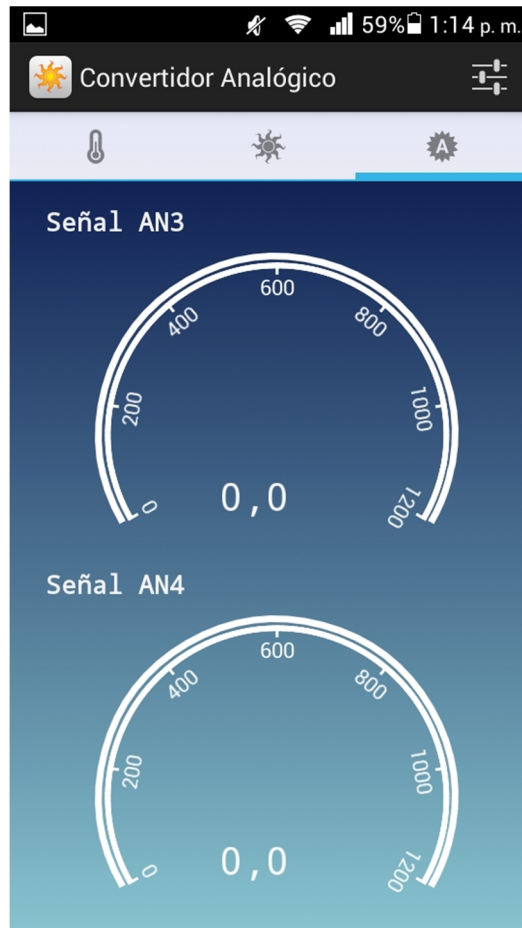


Figura 29. Pantalla para el despliegue de los convertidores analógico a digital.

Al presionar el botón “Configuraciones” en la parte superior derecha, aparece un nuevo grupo de pestañas en donde el usuario puede enviar datos de configuración al sistema electromecánico.

En primer lugar, la pestaña izquierda se utiliza para obtener la fecha y hora del dispositivo Android y enviarlas al sistema electromecánico para el ajuste del tiempo, ver figura 30. Como se muestra en la tabla 2, la fecha y la hora LST son datos de entrada al algoritmo para el cálculo de los ángulos solares que se ejecuta en la componente electrónica del sistema de seguimiento. Con estos datos de fecha y hora se configura la etapa de registro del tiempo en la componente electrónica. Una vez configurada la fecha y hora, esta se mantiene actualizada de forma automática por el reloj de tiempo real en la componente electrónica por aproximadamente 10 años, que es el periodo típico de duración de la batería de respaldo. Al presionar el botón “Ok” de la pestaña, se envían los datos mediante la interface Wi-Fi y se despliega el diálogo de confirmación que se muestra en la figura 31.

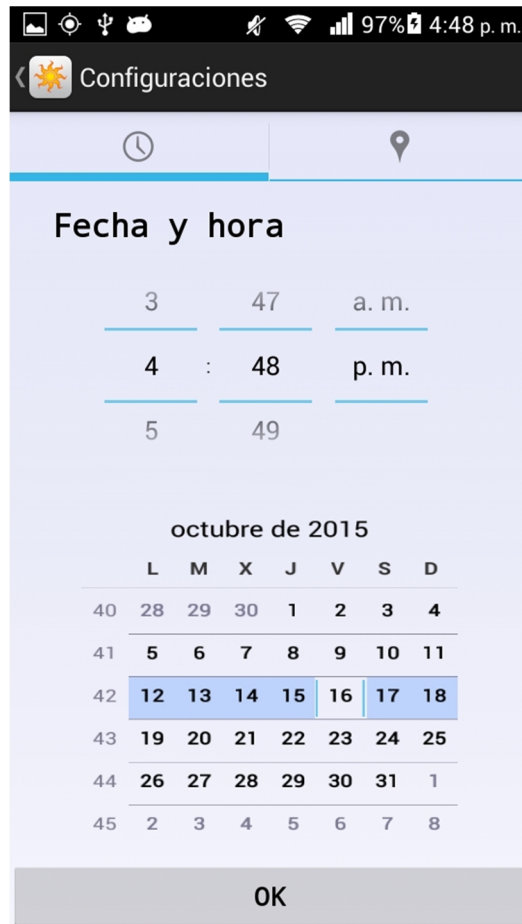


Figura 30. Pantalla para el ajuste del tiempo.

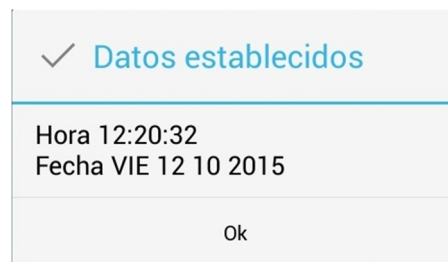


Figura 31. Diálogo de confirmación para el ajuste del tiempo.

En segundo lugar, la pestaña derecha se utiliza para obtener la latitud y longitud del dispositivo Android y enviarlas al sistema electromecánico para el ajuste de la posición terrestre, ver figura 32. Como se muestra en la tabla 2, la latitud y longitud también son datos de entrada al algoritmo para el cálculo de los ángulos solares que se ejecuta en la componente electrónica del sistema de seguimiento. Con estos datos de latitud y longitud se configuran las variables que operan en el algoritmo mencionado. Una vez configurada la latitud y longitud, esta se mantiene fija mientras se conserve la energía que alimenta al sistema electromecánico. Al presionar el botón "Ok" de la pestaña se obtienen los datos ya sea manualmente o mediante el sistema de localización GPS, se envían mediante la interface Wi-Fi y se despliega el diálogo de confirmación que se muestra en la figura 33.

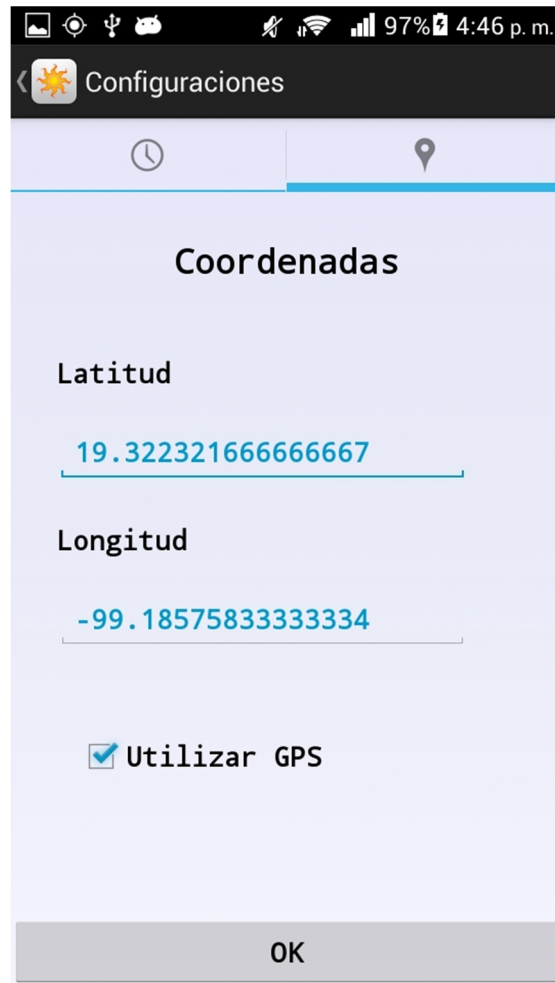


Figura 32. Pantalla para el ajuste de la posición.



Figura 33. Diálogo de confirmación para el ajuste de la posición.

2.3 Descripción del software en el ámbito del programador

Se desarrolló un programa en lenguaje de programación java para la versión del sistema operativo Android 4.2.2 [Montero, 2012] denominada Jelly Bean en el Ambiente de Desarrollo Integrado, IDE, Eclipse [Android Developer, 2015], con el objetivo de obtener información directamente de un dispositivo móvil para configurar parámetros que intervienen en el algoritmo que calcula la posición del sol, además de funcionar como interface para el monitoreo de algunas variables como los ángulos solares, así como de algunos canales analógicos habilitados en el PIC18F4550.

2.3.1 Descripción general del proyecto de programación

Las herramientas que ofrece Eclipse son capaces de crear la estructura de un proyecto con los elementos necesarios para el funcionamiento de una aplicación en Android. La estructura de la aplicación se conforma por distintos directorios y archivos que se muestran en la figura 34 y que están contenidos en el directorio CodigoFuente/Android del CD anexo. Entre los elementos más importantes destacan los siguientes:

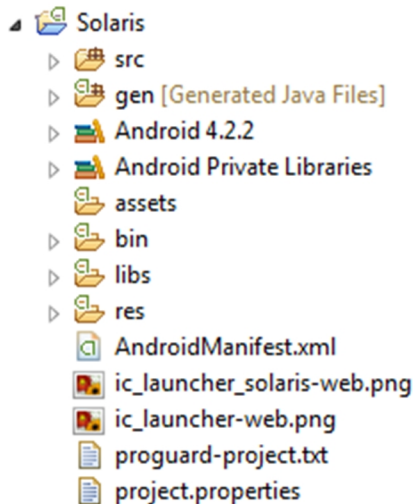


Figura 34. Estructura de la aplicación en Android.

- **src**: Este directorio contiene el código fuente de la aplicación, organizado en archivos con extensión java y contenidos en el paquete com.ccadet.unam.solaris.
- **gen**: Contiene elementos de código autogenerado por Eclipse. El archivo más importante dentro de este directorio es R.java cuyo objetivo es controlar los recursos de la aplicación. En éste se definen una serie de constantes con el identificador, ID, asociado a cada recurso incluido en la carpeta res. Por ejemplo, la constante R.drawable.ic_tab_time contendrá el ID de la imagen ic_tab_time.png ubicada en el directorio res/drawable/.
- **bin**: El compilador coloca aquí los archivos binarios que surgen tras la compilación del proyecto como clases.dex que se generan después de ejecutar la instrucción Dex que es el formato de archivo que acepta la máquina virtual Dalvik.
- **res**: Este directorio contiene los archivos que la aplicación utiliza para visualizarse en pantalla. Estos archivos pueden ser imágenes (PNG, JPEG, etc.), layouts, menús o cadenas de texto entre otros. Los distintos tipos de recursos están agrupados en los siguientes subdirectorios de res y se describen en la tabla 4.

Directorio	Contenido
res/drawable-mdpi res/drawable-hdpi res/drawable-xhdpi res/drawable-xxhdpi	Contienen elementos visuales como iconos y archivos XML que definen niveles de opacidad y degradado de colores. Las distintas carpetas contienen imágenes idénticas pero en distintas resoluciones para cada tipo de pantalla.
res/layout res/layout-land	Aquí se encuentran los ficheros XML que estructuran de manera jerárquica los componentes que se muestran en pantalla o layouts, como botones o cuadros de texto, en ambos directorios existe el mismo fichero pero diseñado para distinta orientación de pantalla ya sea horizontal o vertical. En este directorio residen los dos layouts importantes para la aplicación: activity_main para la

	actividad principal y <code>activity_config</code> para la configuración.
res/menu	Contiene un archivo XML que define un menú para la aplicación.
res/values	Contiene los siguientes archivos: • <code>attrs.xml</code>: define los atributos que puede utilizar la clase <code>Meter</code> y cualquier clase que extienda la funcionalidad de esta. • <code>string.xml</code>: define cadenas de texto utilizadas durante la aplicación. • <code>styles.xml</code>: establece el tema base para la aplicación.

Tabla 4. Organización del directorio `res`.

- **AndroidManifest.xml**: Este archivo describe las especificaciones generales del proyecto como el nombre de la aplicación, versión de Android a la que va dirigida, actividades que lo integran y permisos necesarios para interactuar con los recursos de hardware necesarios para su ejecución. Esta aplicación ejecuta dos actividades: `MainActivity` que es el punto de entrada para el programa y `ConfigActivity`, también requiere saber la ubicación precisa mediante el sistema de posicionamiento GPS, utilizar el servicio de Wi-Fi y acceder a la memoria del teléfono.

2.3.2. Descripción del código fuente

El siguiente apartado describe los métodos y atributos que conforman las clases desarrolladas para la aplicación. Las clases están contenidas en un paquete que identifica a la aplicación de forma única, como se muestra en la figura 35.

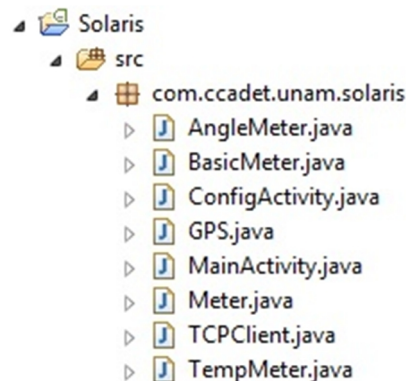


Figura 35. Clases de la aplicación.

Clase `AngleMeter`

Es una subclase de `Meter`, despliega en la interfaz gráfica del usuario un control que muestra el valor de un determinado ángulo. Los métodos y atributos que conforman la clase se muestran en la figura 36.

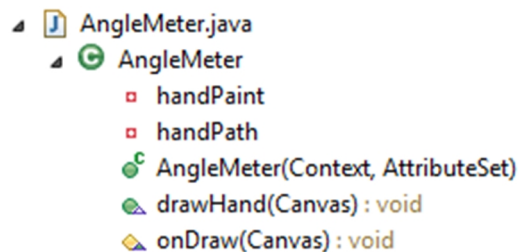


Figura 36. Elementos de la Clase `AngleMeter`.

Al ser una clase más especializada que Meter, debe implementar el método abstracto drawHand en él se diseña un indicador, que se posiciona en el valor actual del ángulo. Este indicador tiene la figura de un sol. La clase debe implementar un constructor que es el único método que no es heredado y sobrescribir el método onDraw(Canvas), colocando en él, el método drawHand para poder dibujar el indicador. Los métodos se muestran en la tabla 5.

Método	Descripción
AngleMeter(Context, AttributeSet)	Constructor de la clase.
drawHand(Canvas)	Prepara los elementos necesarios para dibujar un indicador en pantalla que señale el valor del ángulo correspondiente.
onDraw(Canvas)	Dibuja la forma de un sol en pantalla.

Tabla 5. Métodos de la clase AngleMeter.

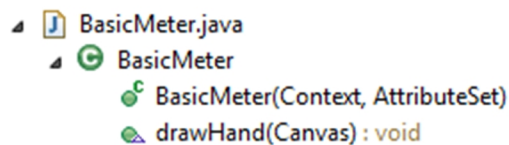
Los atributos necesarios para dibujar este elemento se describen en la tabla 6.

Tipo	Atributo	Significado
Paint	handPaint	Objeto que contiene información respecto al color, estilo y efectos visuales.
Path	handPatht	Objeto que define el contorno de la figura con base en formas primitivas.

Tabla 6. Atributos de la clase AngleMeter.

Clase BasicMeter

Extiende la funcionalidad de la clase Meter, su función es desplegar en pantalla el valor de alguna señal registrada en los canales analógicos AN3 y AN4 del microcontrolador. La figura 37 muestra los métodos de esta clase, es la versión más genérica del medidor ya que además del constructor no implementa algún otro método, aunque el método drawHand aparezca en realidad no está implementado.



```

BasicMeter.java
├── BasicMeter
│   ├── BasicMeter(Context, AttributeSet)
│   └── drawHand(Canvas) : void

```

Figura 37. Metodos de la clase BasicMeter.

Clase ConfigActivity

El propósito de esta actividad es obtener la ubicación geográfica mediante GPS así como hora y fecha que proporcione el dispositivo móvil, esta información es encapsulada en un objeto tipo Bundle y es colocada en una estructura de datos denominada Intent cuyo propósito es pasar la información obtenida a MainActivity. En la figura 38. se muestran los atributos y métodos de esta clase.

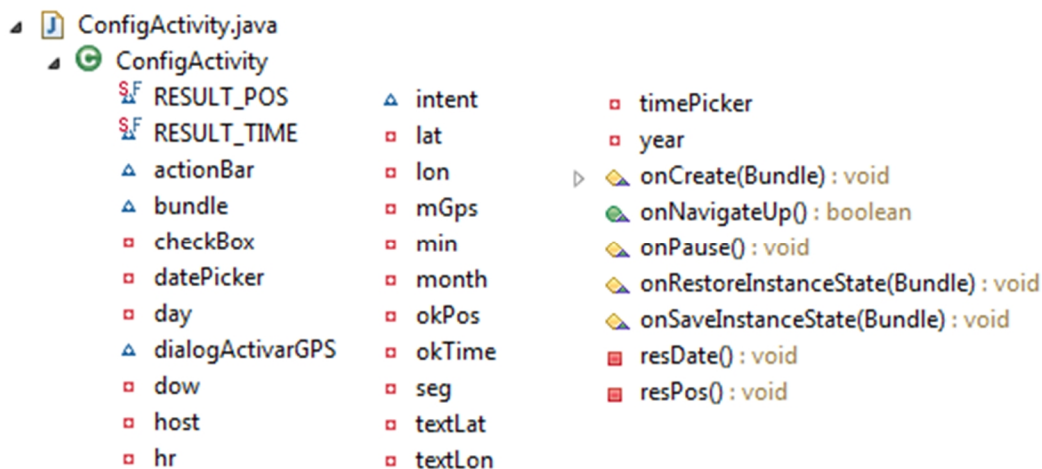


Figura 38. Métodos y atributos de la clase ConfigActivity.

Los datos de hora y fecha sirven para configurar inicialmente los registros del reloj de tiempo real DS1307 mientras que las coordenadas GPS (latitud y longitud) son utilizadas en el algoritmo programado en el microcontrolador para determinar la magnitud de los ángulos solares.

La estructura de la interface de usuario está organizada por dos vistas: la vista inicial muestra elementos prácticos para seleccionar hora y fecha respectivamente, la segunda contiene dos cajas de edición, EditText, para introducir manualmente o automáticamente la latitud y longitud, ambas vistas cuentan con un botón para seleccionar los datos y finalizar la actividad. La tabla 7 muestra los atributos y métodos de la clase

Tipo	Atributo	Significado
int	RESULT_TIME=7 RESULT_POS=5	La actividad tiene dos posibles resultados: establecer coordenadas o tiempo. Cada constante representa un código asignado a ese resultado.
GPS	mGPS	Objeto que proporciona los valores de latitud y longitud obtenidos por el dispositivo móvil mediante la antena de GPS.
String	lat, lon hr, min, seg dow, day, month, year	Almacenan la información de latitud, longitud, hora del día, minuto, segundo, día de la semana, día del mes, mes y año respectivamente.
TabHost	host	Objeto que sirve como contenedor de las distintas vistas de ConfigActivity.
TimePicker	timePicker	Control para seleccionar la hora.
DatePicker	datePicker	Control para seleccionar la fecha.
EditText	textLat textLon	Objetos en los que se muestra la latitud y longitud.
CheckBox	checkBox	Objeto que muestra una casilla en pantalla, al seleccionarla muestra las coordenadas en textLat y textLon respectivamente.
Button	okPos okTime	Controles en la pantalla para seleccionar datos de posición y tiempo.
ActionBar	actionBar	Característica en la pantalla que muestra el nombre de la actividad y habilita un

		botón para regresar a la actividad principal.
Bundle	bundle	Objeto en donde se colocan las variables tipo String con la información obtenida durante la actividad.
Intent	intent	Representa la descripción de una operación para ser ejecutada, en este caso, pasar datos entre dos actividades.
DialogActivarGPS	dialogActivarGPS	Dialogo utilizado para activar el GPS del dispositivo en caso de estar desactivado.

Tabla 7. Atributos de ConfigActivity.

Los controles que se muestran en pantalla tienen como base a la clase android.view.View, estos controles son capaces de registrar y gestionar acciones que el usuario puede ejercer. Los dos objetos Button implementan el método `setOnClickListener(OnClickListener listener)`, el cual recibe como parámetro un componente que registra el evento al oprimir un botón. En el método `onCreate(Bundle)` se implementa la funcionalidad de estos controles. La tabla 8 describe los métodos utilizados en esta clase.

Método	Descripción
<code>onCreate(Bundle)</code>	Establece el contenido de la interface gráfica definida por el archivo <code>activity_config.xml</code> , también aquí se implementa la funcionalidad de los controles y se inicializan algunas variables.
<code>onSaveInstanceState(Bundle)</code>	Almacena en un estado el valor actual de la vista mostrada en pantalla.
<code>onRestoreInstanceState(Bundle)</code>	Restaura el estado de la vista antes de visualizar la actividad.
<code>onNavigateUp()</code>	Finaliza la actividad y regresa a MainActivity, es invocado al oprimir el botón en la parte izquierda de la barra de acciones.
<code>onPause()</code>	Limpia la casilla de verificación que ofrece la opción de utilizar las coordenadas mediante GPS y cancela las notificaciones de cambio de posición.
<code>resPos()</code>	Obtiene la información que se encuentra en los objetos <code>EditText</code> en formato String para ser colocados en un objeto bundle.
<code>resDate()</code>	Obtiene la información de hora y fecha, mediante los distintos métodos que ofrecen los controles <code>timePicker</code> y <code>datePicker</code> y la clase <code>Calendar</code> , estos datos son colocado en el objeto bundle.

Tabla 8. Métodos de ConfigActivity.

Clase GPS

Actualmente los dispositivos Android incorporan una variedad de sistemas de posicionamiento, uno de los más populares es el servicio de GPS, el cual proporciona entre otras cosas latitud y longitud correspondientes a la ubicación real del dispositivo. Esta clase se encarga de obtener el servicio de localización mediante GPS y proporcionar, de ser posible, una ubicación actual. La figura 39 muestra los atributos y métodos que conforman esta clase.



Figura 39. Elementos de la clase GPS.

Debido a las condiciones del servicio, existe un tiempo de espera entre cada notificación de cambio de posición, por lo que en primer instancia las coordenadas se obtienen por medio del método `getLastKnownLocation(String provider)` el cual devuelve la última posición conocida por el proveedor indicado, si es que esta posición existe. Las coordenadas son válidas hasta que el método `onLocationChanged(Location location)` registre una actualización en la posición. La tabla 9 describe los atributos de la clase GPS.

Tipo	Atributo	Significado
Context	mcontext	Información global sobre el entorno de la aplicación.
Location Manager	mLocationManager	Componente que gestiona el acceso al servicio de localización.
Location	mLocation	Tipo de dato que encapsula los datos de una posición geográfica.
double	lat lon acc	Variables que almacenan latitud, longitud en grados y precisión en metros determinadas por un objeto tipo Location.
boolean	isGPSEnable	Variable para comprobar el estado del GPS, su valor es true si el GPS está habilitado.
boolean	canGetLocation	Variable que determina si es posible obtener una posición.

Tabla 9. Atributos de la clase GPS.

GPS extiende de la clase abstracta `android.app.Service` por lo que las herramientas de Eclipse incluyen de manera automática todos los métodos aunque no se implementen. La tabla 10 muestra los métodos utilizados por esta clase.

Método	Descripción
GPS(Context)	Constructor de la clase.
getLocation()	Se encarga de obtener el servicio de GPS y suscribirse a las notificaciones de cambio de posición indicando que las actualizaciones se reciban tan pronto estén disponibles.
onLocationChanged(Location)	Es invocado cada que se recibe una nueva posición.
onStatusChanged(String, int, Bundle)	No utilizado.

onProviderEnabled(String)	No utilizado.
onProviderDisabled(String)	No utilizado.
onBind(Intent)	No utilizado.
stopGPS()	Cancela las actualizaciones del proveedor de localización.
getLat()	Devuelve la latitud en una variable tipo double
getLon()	Devuelve la longitud en una variable tipo double

Tabla 10. Métodos de la clase GPS.

Clase MainActivity

Normalmente al crear un nuevo proyecto en Eclipse, la clase que se lanzará al ejecutarse la aplicación se denomina MainActivity. Esta clase, derivada de Activity, es la encargada de cargar el layout y ejecutar los métodos que le dan funcionalidad a toda la aplicación. La figura 40 muestra los atributos y métodos de la clase MainActivity.

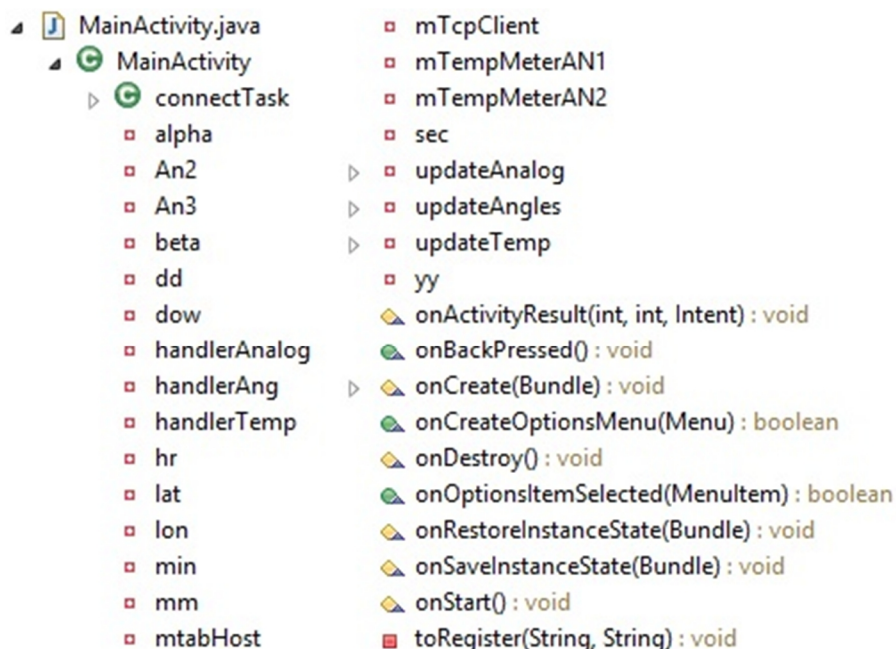


Figura 40. Elementos de la clase MainActivity.

La tabla 11 muestra los atributos de la clase MainActivity.

Tipo	Atributo	Significado
TabHost	mtabHost	Variable utilizada para configurar la interface de pestañas y asignar el método listener que atiende el evento por cambio de pestaña.
TCPClient	mTcpClient	Variable utilizada para crear un objeto del tipo TCPClient con el objeto de acceder a los métodos de esta clase y habilitar la funcionalidad de cliente TCP de la presente aplicación.
AngleMeter	beta alpha	Variables utilizadas para crear objetos del tipo AngleMeter que permiten modificar los valores que se muestran en los despliegues analógicos asignados a los ángulos de azimut (alfa) y altitud (beta).
TempMeter	mTempMeterAN1 mTempMeterAN2	Variables utilizadas para crear objetos del tipo TempMeter que permiten modificar los valores que se muestran en los despliegues

		analógicos asignados a las temperaturas.
BasicMeter	An2 An3	Variables utilizadas para crear objetos del tipo BasicMeter que permiten modificar los valores que se muestran en los despliegues analógicos asignados a los canales de conversión analógico a digital.
Handler	handlerAng handlerTemp handlerAnalog	Variables que permiten el acceso a los eventos periódicos.
String	lon lat hr min sec dd mm yy dow	Variables que almacenan temporalmente los datos de posición, fecha y hora. Esta información es encapsulada en un objeto tipo Bundle y es recibida en una estructura de datos denominada Intent cuyo propósito es recibir la información obtenida de ConfigActivity.
Runnable	updateAnalog updateAngles updateTemp	Componentes que ejecutan los eventos periódicos del tipo Runnable cada 5 segundos que consisten en el envío, por conexión TCP, de las cadenas de comando al servidor en la componente electrónica del seguidor solar.
AsyncTask	connectTask	Componente de ejecución asíncrona de los métodos doInBackground, messageRecived y onProgressUpdate para recibir y procesar las cadenas recibidas por el cliente TCP.

Tabla 11. Atributos de la clase MainActivity.

Los métodos de esta clase, descritos en la tabla 12, son los que soportan gran parte de la funcionalidad de la aplicación.

Método	Descripción
onCreate(Bundle savedInstanceState)	Es el método que se ejecuta al iniciar la aplicación. Lanza el layout activity_main. Habilita el servicio WiFi. Inicializa las variables asignadas a los componentes del layout. Crea el componente connectTask para el procesamiento asíncrono de mensajes TCP. Implementa el método listener onTabChanged para procesar los cambios de pestaña.
onDestroy()	Procesa la finalización de la aplicación al desconectar el cliente TCP.
onCreateOptionsMenu(Menu menu)	Crea el menú de opciones para la configuración de la aplicación.
onStart()	Inicializa los valores de los despliegues analógicos.
onSaveInstanceState(Bundle outState)	Permite almacenar el estado previo a la destrucción de la aplicación.
onRestoreInstanceState(Bundle savedInstanceState)	Permite recuperar el estado de la aplicación.
onOptionsItemSelected(MenuItem item)	Procesa la selección del menú de opciones para desplegar la actividad ConfigActivity en donde se configura la aplicación.

onBackPressed()	Procesa el evento por presionar el botón de retroceso y despliega un dialogo para confirmar la finalización de la aplicación.
onActivityResult(int requestCode, int resultCode, Intent data)	Procesa el evento por presionar el botón ok en la actividad de configuración. Forma las cadenas que se enviaran mediante el cliente TCP para configurar la componente electrónica del seguidor solar. Las cadenas contienen los datos de fecha y posición.
toRegister(String v_title,String v_cont) throws IOException	Utilizada con fines de depuración para escribir en archivo los resultados parciales del desempeño de la aplicación.

Tabla 12. Métodos de la clase MainActivity.

Clase Meter

La clase Meter es un tipo abstracto de vista que se utiliza para dibujar en el lienzo (canvas) la forma general de un indicador analógico a partir de primitivas geométricas. Los atributos y métodos de esta clase se muestran en la figura 41.

- Meter.java - Meter - centerDegrees - gaugePosition - scalePosition - colorA - colorB - colorC - currentValue - currentValuePaint - currentValuePath - degreeMaxValue - degreeMinValue - degreesPerNotch - dialAcceleration - dialVelocity - facePaint	- faceRect - gaugeRect - incrementPerLargeNotch - incrementPerSmallNotch - lastDialMoveTime - scaleCenterValue - scaleMaxValue - scaleMinValue - scalePaintLargeNotch - scalePaintNumber - scalePaintSmallNotch - scaleRect - targetValue - totalNotches - valueAllPaint - valueOkPaint - Meter(Context)	- Meter(Context, AttributeSet) - Meter(Context, AttributeSet, int) - calculateCurrentValue() : void - drawCurrentValue(Canvas) : void - drawFace(Canvas) : void - drawGauge(Canvas) : void - drawHand(Canvas) : void - drawScale(Canvas) : void - init() : void - notchToValue(int) : int - onAttachedToWindow() : void - onDetachedFromWindow() : void - onDraw(Canvas) : void - onRestoreInstanceState(Parcelable) : void - onSaveInstanceState() : Parcelable - setValueAbs(float) : void - valueToAngle(float) : float
---	--	--

Figura 41. Elementos de la clase Meter.

El dibujo sobre el lienzo utiliza diversos atributos que definen la geometría a dibujar y su apariencia. Los atributos utilizados se muestran en la tabla 13.

Tipo	Atributo	Significado
float	degreeMinValue	Constantes geométricas que se utilizan para dimensionar el dibujo sobre el lienzo.
float	degreeMaxValue	
int	totalNotches	
int	incrementPerLargeNotch	
int	incrementPerSmallNotch	

int int int float float int float float float float float	scaleMinValue scaleCenterValue scaleMaxValue <i>scalePosition</i> <i>gaugePosition</i> <i>centerDegrees</i> degreesPerNotch currentValue targetValue dialVelocity dialAcceleration	
int int int	colorA colorB colorC	Constantes predefinidas para representar los colores de las primitivas geométricas.
Paint Paint Paint Paint Paint Paint Paint	valueOkPaint facePaint valueAllPaint scalePaintNumber scalePaintLargeNotch scalePaintSmallNotch currentValuePaint	Clases públicas de Android que contienen la información de estilo y color con que se dibujan las geometrías.
Path	currentValuePath	Objeto que se utiliza para encapsular contornos compuestos de rutas a partir de geometrías básicas.
long	lastDialMoveTime	Variable que almacena el estado previo de posicionamiento en la aguja del indicador analógico.
RectF RectF RectF	faceRect gaugeRect scaleRect	Objetos que encapsulan las coordenadas rectangulares de áreas predestinadas a geometrías sobre el lienzo.

Tabla 13. Atributos de la clase Meter.

La aplicación utiliza los métodos mostrados en la tabla 14 para dibujar un indicador analógico a partir de geometrías elementales.

Método	Descripción
Meter(Context context) Meter(Context context, AttributeSet attributeSet) Meter(Context context, AttributeSet atrAttributeSet,int defStyleAttr)	Constructores de la clase que reciben diversos parámetros con fines de inicialización.
onDetachedFromWindow() onAttachedToWindow()	Métodos sobrecargados para habilitar la desvinculación y vinculación del lienzo de su vista.
onRestoreInstanceState(Parcelable state) Parcelable onSaveInstanceState()	Métodos utilizados para restablecer y resguardar variables importantes para el dibujo sobre el lienzo.
init()	Inicia los objetos que son utilizados en el dibujo sobre el lienzo.
drawFace(Canvas canvas)	Dibuja el marco que rodea la escala elipsoidal del indicador analógico.
drawGauge(Canvas canvas)	Dibuja la barra de relleno que recorre la escala elipsoidal desde el inicio hasta el valor medido.
drawCurrentValue(Canvas canvas)	Dibuja el valor numérico medido.
drawScale(Canvas canvas)	Dibuja la escala graduada del indicador analógico.
calculateCurrentValue()	Calcula el desplazamiento y aceleración que

	realizará el indicador analógico a partir del valor medido.
valueToAngle(float value)	Transforma el valor medido a desplazamiento angular.
onDraw(Canvas canvas)	Ejecuta los métodos drawFace, drawGauge, drawCurrentValue, drawScale y calculateCurrentValue que dibujan el indicador analógico en conjunto.
notchToValue(int notch)	Calcula el valor de la división mínima dibujada sobre la escala del indicador analógico.
setValueAbs(float value)	Recorta el valor medido para que no sobrepase el rango del indicador analógico.
drawHand(Canvas canvas)	Dibuja la aguja que indica el valor medido.

Tabla 14. Métodos de la clase Meter.

Clase TCPClient

La clase TCPClient implementa las funciones básicas para el cliente TCP Android a partir de los elementos que se muestran en la figura 42.

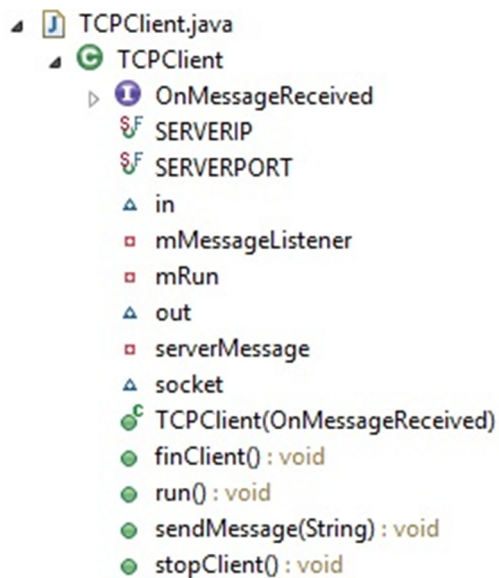


Figura 42. Elementos de la clase TCPClient.

El cliente implementado se utiliza básicamente para transmitir datos de configuración al sistema electromecánico y recibir de él datos de mediciones de ángulo y variables analógicas. Los atributos utilizados en esta clase se muestran en la tabla 15.

Tipo	Atributo	Significado
String	SERVERIP	Constante que contiene la dirección IP del servidor que se levanta en la componente electrónica del seguidor.
int	SERVERPORT	Constante que contiene el puerto del servidor que se levanta en la componente electrónica del seguidor.
interface	OnMessageReceived	Permite intercambiar el mensaje TCP recibido con la actividad principal.
String	serverMessage	Contiene el mensaje recibido mediante TCP.
boolean	mRun	Variable que habilita o deshabilita la recepción de

		mensajes mediante TCP.
Socket	socket	Es el componente abstracto que se utiliza para el intercambio de datos mediante la interface TCP.
BufferedReader	in	Implementa el buffer de datos en donde se almacenarán los datos que arriban de la interface TCP.
PrintWriter	out	Formatea el flujo de información de salida (mensaje) para que pueda ser enviado por la interface TCP hacia el servidor en la componente electrónica.

Tabla 15. Atributos de la clase TCPClient.

Los métodos de esta clase se describen en la tabla 16.

Método	Descripción
TCPClient(OnMessageReceived listener)	Es el método constructor de la clase.
sendMessage(String message)	Vacía el flujo de información (mensaje) por la interface TCP para que sea enviado.
stopClient()	Deshabilita la recepción de mensajes TCP al modificar la variable booleana mRun.
finClient()	Cierra el socket de comunicación TCP.
run()	Es el método de ejecución periódica que se encarga de encuestar el estado del buffer de lectura in para recibir información TCP.

Tabla 16. Métodos de la clase TCPClient.

Clase TempMeter

TempMeter es una clase derivada de Meter. Proporciona una interface gráfica para desplegar los indicadores analógicos de temperatura a partir de su clase derivada funcional Meter. Los atributos y métodos de esta clase se muestran en la figura 43.

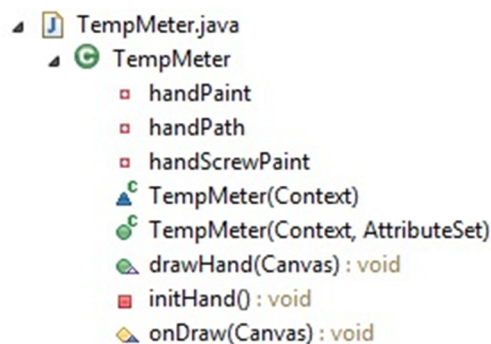


Figura 43. Elementos de la clase TempMeter.

Los atributos de esta clase se describen en la tabla 17.

Tipo	Atributo	Significado
Path	handPath	Objeto que se utiliza para encapsular contornos compuestos de rutas a partir de geometrías básicas.
Paint	handPaint handScrewPaint	Clases públicas de Android que contienen la información de estilo y color con que se dibujan las geometrías.

Tabla 17. Atributos de la clase TempMeter.

Los métodos de esta clase se describen en la tabla 18.

Método	Descripción
TempMeter (Context context) TempMeter(Context context, AttributeSet attributeSet)	Constructores de la clase.
initHand()	Prepara los elementos necesarios para dibujar sobre el lienzo.
drawHand(Canvas canvas)	Dibuja la aguja que se utiliza para señalar, en el despliegue analógico, la medición de temperatura.
onDraw(Canvas canvas)	Es el método sobrecargado que se utiliza para dibujar en el lienzo.

Tabla 18. Métodos de la clase TempMeter.

2.4 Componente mecánica

Se diseñó y construyó la componente mecánica de la figura 44 para el seguimiento de la trayectoria solar con dos grados de libertad (azimut y altitud).

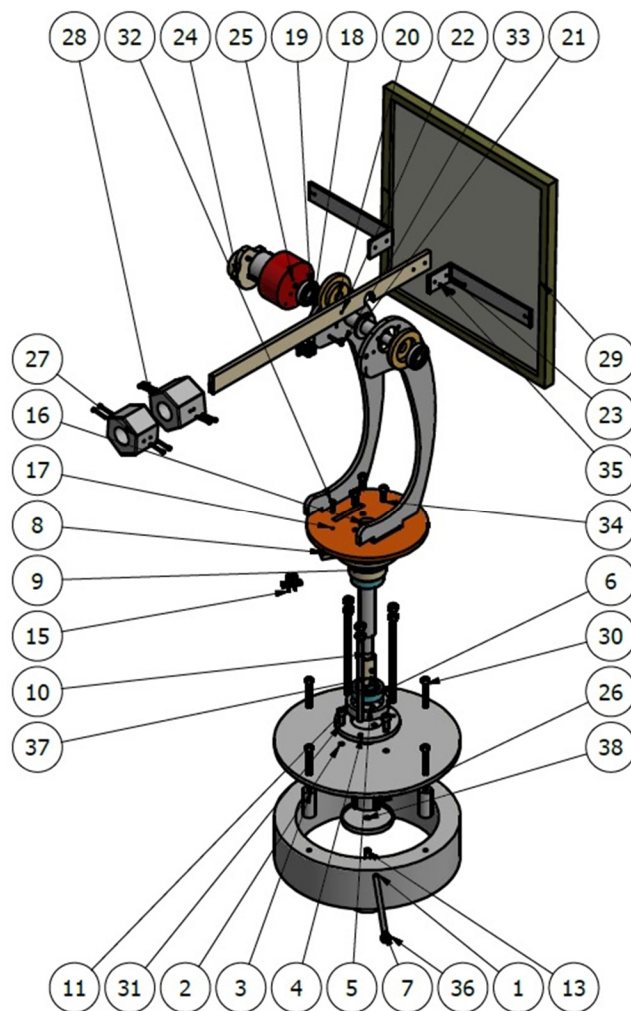


Figura 44. Componente mecánica.

El diseño está enfocado a transmitir el movimiento de dos moto-reductores a paso a los mecanismos de azimut y altitud que orientan el panel solar disminuyendo, en la medida de lo posible, fricciones y esfuerzos mecánicos. Los detalles de las piezas construidas se muestran en el anexo B. A continuación una descripción de los elementos de la figura 44.

1. Base principal. Es un cilindro hueco de cold rolled, robusto de manera que le proporciona estabilidad mecánica al resto de los componentes que se montan sobre éste.
2. Base superior de la base principal. Es una placa de cold rolled, de igual manera robusta que conforma el medio plano de sujeción para la unión inferior con rodamiento del eje azimutal y los tornillos de sujeción con la unión superior con rodamiento del eje azimutal.
3. Postes inferiores. Conforman el medio espaciador para proporcionar alojamiento al moto-reductor azimutal.
4. Base pvc inferior. Es la base de PVC para guiar verticalmente el eje azimutal.
5. Base para rodamiento cónico. Es la base inferior para colocar el rodamiento cónico azimutal.
6. Rodamiento cónico comercial. Es el rodamiento cónico azimutal inferior sobre el cual descansa el peso completo del mecanismo de elevación y libera de esfuerzos al moto-reductor azimutal.
7. Varilla de soporte. Es la varilla de soporte para el moto-reductor azimutal sobre la base principal. Utilizado con base en tornillos comerciales.
8. Base superior de aluminio. Es la base superior fija a la base principal mediante tornillos largos. En esta base se establece la referencia para la rotación del mecanismo azimutal y se coloca el foto-detector.
9. Base superior de PVC para el rodamiento radial. Es la base superior para el rodamiento radial que orienta el eje azimutal verticalmente y elimina los esfuerzos en el ensamble superior.
10. Flecha principal de pvc. Es el eje azimutal que descansa libre de esfuerzos en los rodamientos cónico inferior y radial superior.
11. Buje para el motor de azimut. Es un buje de ensamble entre el eje azimutal y el moto-reductor azimutal.
13. Moto-reductor de azimut. Moto-reductor azimutal con operación libre de esfuerzos, sobre-cargas y fricciones mecánicas. Motor comercial.
15. Sensor de posición en azimut. Opto-interruptor para identificar la posición inicial (cero) en azimut. Sensor comercial.
16. Brida para la base de la plataforma. Brida de sujeción entre la plataforma de soporte para el mecanismo de elevación y el eje azimutal.
17. Plataforma de movimiento. Es el plano con movimiento azimutal sobre el cual descansan los dispositivos de movimiento de elevación. Su rotación es solidaria con el eje azimutal.
18. Base orgánica. Son un par de palancas de soporte del mecanismo de elevación que proporcionan libertad de movimiento al eje de elevación, contrapesos y panel solar.
19. Rodamiento radial comercial. Par de rodamientos radiales que le proporcionan movimiento libre de esfuerzos al eje de elevación.
20. Brida de soporte. Brida que da soporte al rodamiento de elevación con la base orgánica.
21. Eje de movimiento de elevación. Eje de giro sobre rodamientos radiales que transmite el movimiento de elevación.
22. Barra real de soporte. Es la barra que une el conjunto de contrapesos y panel solar al eje de movimiento de elevación.
23. Barras en L. Par de barras que dan soporte al panel solar.
24. Moto-reductor de elevación. Es el dispositivo que opera libre de esfuerzos y fricciones para proporcionar el movimiento de elevación al panel solar. Motor comercial.
25. Base para los reductores. Base para sujetar los motores de azimut y elevación a sus respectivas cajas de engranes reductores.
26. Cilindro interior. Cilindro de acoplamiento entre el moto-reductor de azimut y su respectivo eje.
27. Contrapeso chico. Utilizado para equilibrar la barra real de soporte sobre el eje de movimiento de elevación.
28. Contrapeso grande. Utilizado para equilibrar la barra real de soporte sobre el eje de movimiento de elevación.
29. Panel solar. Elemento comercial para la generación de energía eléctrica.

- 30. a 37. Tornillos comerciales de sujeción de diversos componentes.
- 38. Tapadera inferior. Tapa para el moto-reductor azimuthal.

2.5 Procedimiento típico para operar el sistema

1. Alambrar el sistema como se muestra en las figuras 45 y 46.

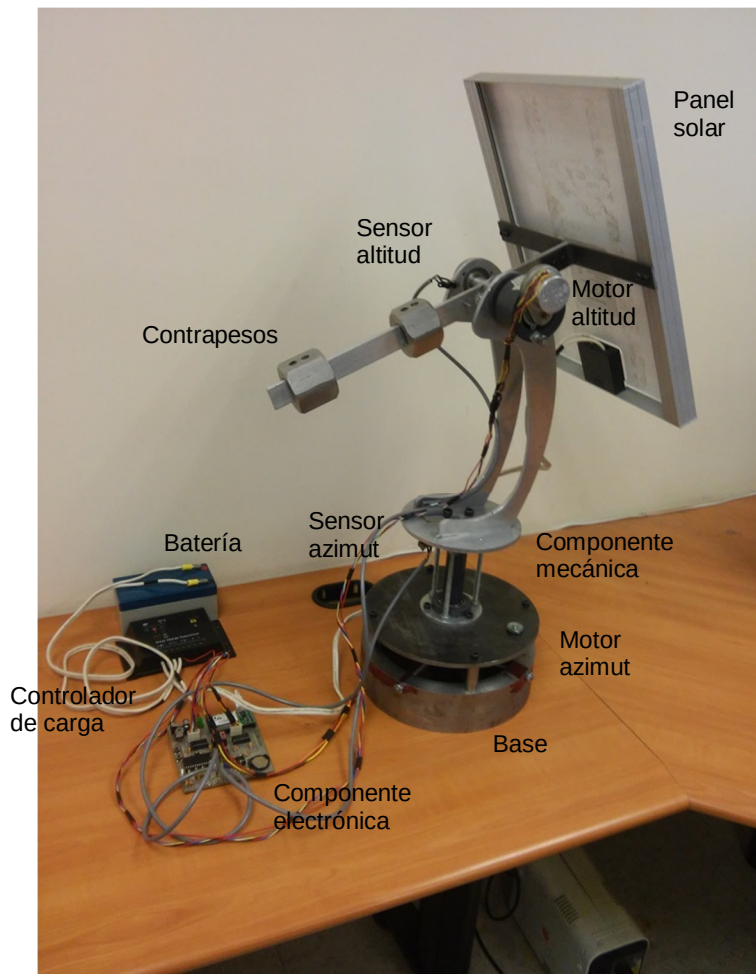


Figura 45. Alambrado del sistema de seguimiento solar.

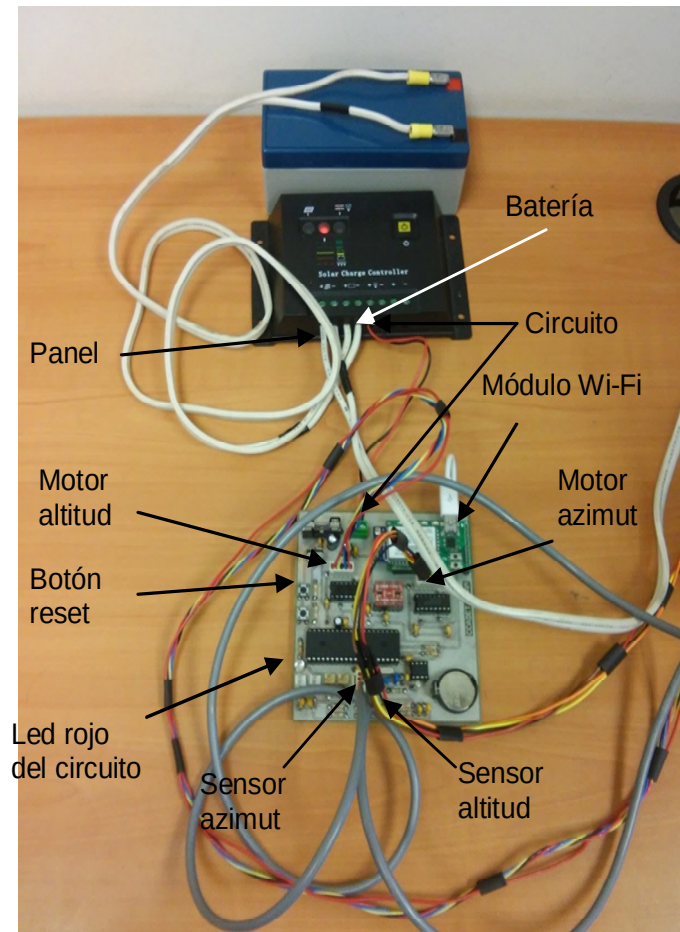


Figura 46. Alambrado del sistema de seguimiento solar (detalle).

2. Localizar el sistema completo en un área soleada y orientar la componente mecánica con el norte geográfico con la ayuda de una brújula, como se muestra en la figura 47.

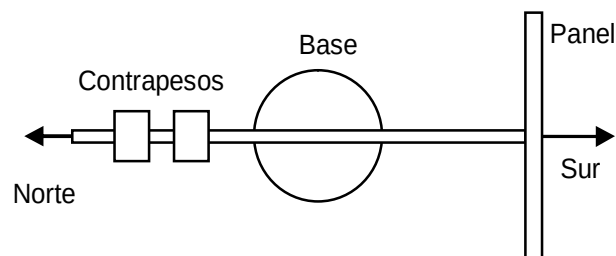


Figura 47. Orientación de la componente mecánica (vista superior).

3. Asegurarse que al inicio la batería este con carga completa.
4. Bajo estas condiciones el controlador de carga indicará que el panel no carga la batería ya que ésta se encuentra con carga completa, como se muestra en la figura 48. El led azul PV destellará cuando la batería demande carga y se disponga de irradiación solar.

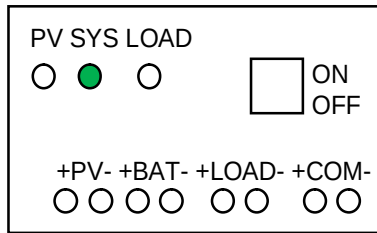


Figura 48. Estado inicial del controlador de carga.

5. Encender el controlador de carga para aplicar energía a la componente electrónica (led LOAD encendido). El led rojo del circuito deberá permanecer encendido, mientras que el led verde del módulo Wi-Fi deberá destellar de forma lenta y el led rojo del mismo módulo deberá destellar rápidamente. En este instante la componente mecánica se desplazará hasta alcanzar los ángulos de altitud y azimut cero. Posteriormente, la componente mecánica se desplazará a los ángulos de altitud y azimut calculados para la fecha, hora, longitud y latitud actuales.
6. En el dispositivo móvil con la aplicación Solaris instalada, establecer una conexión con la red inalámbrica denominada BenjasRN. Lo anterior es un procedimiento estándar en los dispositivos Android que generalmente se configura en la opción Aplicaciones/Ajustes/Wi-Fi/BenjasRN. La red no requiere de contraseña. En este momento el led rojo del módulo Wi-Fi deberá apagarse.
7. Presionar el botón de reset en la componente electrónica. En este instante el led rojo del circuito destellará 5 veces y el mecanismo buscará la posición cero. Posteriormente el mecanismo se desplazará a la posición que actualmente genera el algoritmo de generación de la trayectoria solar y, una vez alcanzada, el led rojo del circuito volverá a destellar 5 veces indicando que la aplicación Solaris puede iniciarse.
8. Ejecutar la aplicación Solaris y operar el sistema como se mencionó en 2.2. En éste momento el led rojo del circuito permanecerá encendido y el led verde del módulo Wi-Fi también.
9. Al finalizar la aplicación Solaris el led verde del módulo Wi-Fi deberá destellar de forma lenta.

3 Resultados y discusión

Se construyó un sistema electromecánico para el seguimiento de la trayectoria solar a partir de herramientas básicas prescindiendo de soluciones comerciales.

3.1 Resultados

Los resultados obtenidos para el sistema electromecánico se reportan de manera cualitativa y cuantitativa. En la evaluación cualitativa se reportan fotografías de las componentes electrónica, software y mecánica. Para la evaluación cuantitativa se desarrollaron dos experimentos que verifican el desempeño.

3.1.1 Prototipo desarrollado

La figura 49 muestra el prototipo desarrollado en conjunto.



Figura 49. Prototipo desarrollado.

La figura 50 muestra la componente electrónica fabricada en circuito impreso.



Figura 50. Componente electrónica.

La figura 51 muestra el software de operación con la pestaña correspondiente a los ángulos solares calculados en el sistema electromecánico.



Figura 51. Componente software.

La figura 52 muestra dos detalles de la componente mecánica en donde se aprecian los mecanismos que permiten el posicionamiento del panel en los dos grados de libertad (elevación y azimut).

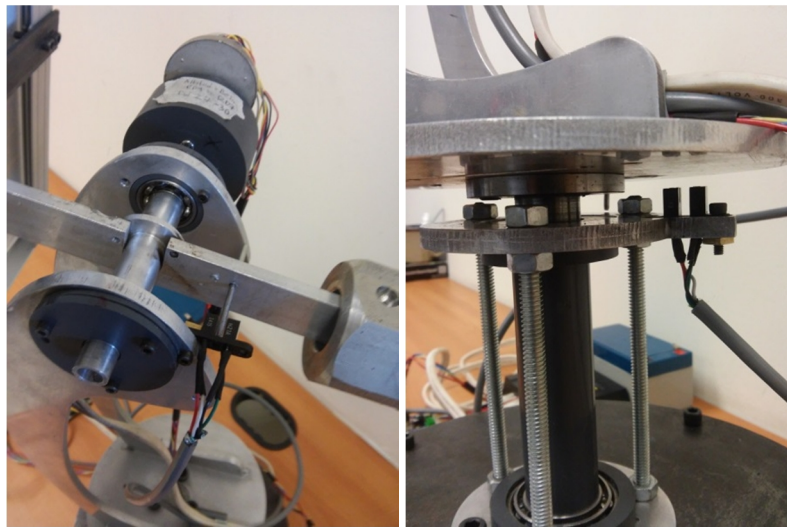


Figura 52. Componente mecánica.

3.1.2 Comparación teórica contra real para los ángulos solares

Para verificar el desempeño del algoritmo que calcula los ángulos solares, se eligieron tres horarios representativos del amanecer (8:00), mediodía (13:00) y atardecer (18:00) en la Ciudad de México. Con esta hora, latitud y longitud se calcularon los datos de hora del amanecer (Sunrise), hora del atardecer (Sunset), altitud (Beta) y azimut (Alfa) para tres fechas representativas de los equinoccios y solsticios. Los cálculos se realizaron tomando como referencia el algoritmo descrito en 1.4 y 1.5 programado en MatLab y se compararon contra el mismo algoritmo programado en lenguaje C para el microcontrolador PIC18F4550. Las tablas 19 a 21 muestran los resultados obtenidos en donde se puede observar el gran parecido entre ambos algoritmos, lo que comprueba el buen desempeño del algoritmo de cálculo.

Ciudad de México 8:00 tiempo local	Microcontrolador				Matlab			
	Sunrise	Sunset	Beta	Alfa	Sunrise	Sunset	Beta	Alfa
Equinoccio de primavera 21 de marzo	06:44	18:43	17.6	83.1	06:44	18:43	17.65	83.1
Equinoccio de otoño 22 de septiembre	06:30	18:28	21.0	81.0	06:29	18:27	21.3	81.4
Solsticio de verano 21 de junio	06:02	19:13	25.8	107.3	06:02	19:13	25.8	107.3
Solsticio de invierno 21 de diciembre	07:10	17:59	10.4	60.4	07:10	17:59	10.41	60.4

Tabla 19. Cálculos para la Ciudad de México 8:00.

Ciudad de México 13:00 tiempo local	Microcontrolador				Matlab			
	Sunrise	Sunset	Beta	Alfa	Sunrise	Sunset	Beta	Alfa
Equinoccio de primavera 21 de marzo	06:44	18:43	69.7	-11.38	06:44	18:43	69.7	-11.3
Equinoccio de otoño 22 de septiembre	06:30	18:28	68.5	-22.0	06:29	18:27	68.5	-22.0

Solsticio de verano 21 de junio	06:02	19:13	83.4	-129.07	06:02	19:13	83.4	-129
Solsticio de invierno 21 de diciembre	07:10	17:59	46.6	-8.65	07:10	17:59	46.7	-8.3

Tabla 20. Cálculos para la Ciudad de México 13:00.

Ciudad de México 18:00 tiempo local	Microcontrolador				Matlab			
	Sunrise	Sunset	Beta	Alfa	Sunrise	Sunset	Beta	Alfa
Equinoccio de primavera 21 de Marzo	06:44	18:43	10.3	-85.8	06:44	18:43	10.3	-85.8
Eq. De otoño 22 de Septiembre	06:30	18:28	6.5	-87	06:29	18:27	6.5	-87
Solsticio de verano 21 de Junio	06:02	19:13	15.7	-109.8	06:02	19:13	15.9	-109.7
Solsticio de invierno 21 de Dic.	07:10	17:59	-0.1	-65.046	07:10	17:59	0	0

Tabla 21. Cálculos para la Ciudad de México 18:00.

3.1.3 Desempeño en posición del prototipo

Para el segundo experimento, se registró el ángulo observado en distintas posiciones con el fin de comprobar si el ángulo que se fija como objetivo corresponde al que se mueve físicamente en los motores. En el arreglo experimental se tomó el conjunto de motor y reductor de engranes y una escala angular graduada como se muestra en la figura 53.

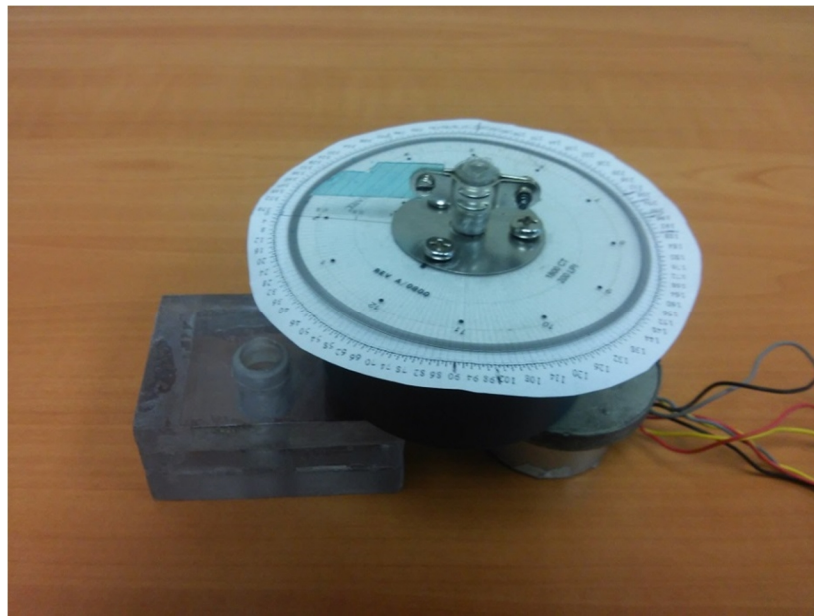


Figura 53. Arreglo experimental para evaluar el desempeño en posición.

Se hizo un recorrido de 0 a 180° y de regreso hasta -180° en intervalos de 10°, desde una posición inicial que equivale a cero. Existe una diferencia de hasta 4° entre el objetivo y el ángulo observado, aunque si se moviera el motor de 0° a 180° el ángulo que se fijó como objetivo corresponde al observado. La tabla 22 muestra los datos obtenidos.

Objetivo	Angulo observado	Objetivo	Angulo observado	Objetivo	Angulo observado	Objetivo	Angulo observado
0	0	170	174	-10	-10	-170	-172
10	9	160	164	-20	-20	-160	-162
20	20	150	154	-30	-30	-150	-152
30	30	140	144	-40	-40	-140	-142
40	40	130	134	-50	-50	-130	-132
50	50	120	124	-60	-60	-120	-122
60	61	110	114	-70	-70	-110	-112
70	71	100	103	-80	-80	-100	-102
80	81	90	93	-90	-91	-90	-92
90	91	80	83	-100	-101	-80	-82
100	101	70	73	-110	-111	-70	-72
110	112	60	62	-120	-121	-60	-62
120	122	50	52	-130	-131	-50	-52
130	132	40	42	-140	-142	-40	-42
140	142	30	31	-150	-152	-30	-31
150	152	20	21	-160	-162	-20	-21
160	162	10	10	-170	-172	-10	-10
170	172	0	0	-180	-182	0	0
180	182						

Tabla 22. Datos registrados para el azimut.

Se realizó el mismo procedimiento para el motor designado para la altitud, los datos obtenidos se muestran en la tabla 23.

Objetivo	Angulo observado	Objetivo	Angulo observado	Objetivo	Angulo observado	Objetivo	Angulo observado
0	0	80	81	-10	-10	-80	-81
10	9	70	71	-20	-21	-70	-71
20	19	60	61	-30	-31	-60	-61
30	29	50	51	-40	-41	-50	-52
40	39	40	40	-50	-51	-40	-42
50	49	30	30	-60	-61	-30	-32
60	60	20	20	-70	-71	-20	-22
70	70	10	10	-80	-81	-10	12
80	80	0	0	-90	-91	0	-2
90	90						

Tabla 23. Datos registrados para el motor para la altitud.

3.2 Discusión y trabajo a futuro

Se desarrolló una herramienta de trabajo que permite depurar algoritmos propios para el seguimiento de la trayectoria solar. No constituye una solución a la generación de energía limpia. No obstante, con el desarrollo de esta herramienta, se adquiere experiencia y se identifican posibles problemas que puedan ser relevantes en el momento de ofrecer una solución a un problema en particular, ya sea en la generación de energía eléctrica o la concentración de energía solar.

El prototipo descrito en este trabajo tiene el mérito de haber sido desarrollado a partir de herramientas básicas, prescindiendo de soluciones comerciales. Como prueba de ello se pueden mencionar los siguientes aspectos. En primer lugar, el sistema electrónico fue desarrollado con base en un microcontrolador del cual el fabricante ofrece gratuitamente sus herramientas de desarrollo y uso libre de licencias. En segundo lugar, se construyó un dispositivo mecánico de diseño propio utilizando los recursos propios del taller del CCADET UNAM. Finalmente, para el desarrollo de la componente software

se utilizó una plataforma de programación de uso libre en la cual participan programadores que comparten sus experiencias en diversos foros abiertos sin fines de lucro.

Como trabajo a futuro se plantea lo siguiente:

- Mejorar el sistema de cableado que facilite un mejor movimiento en los mecanismos rotatorios de altitud y azimut.
- Mejorar el acoplamiento entre los motores y el dispositivo mecánico para evitar efectos de "juego mecánico o backlash".
- Mejorar la robustez del firmware en el microcontrolador con un sistema supervisor (watchdog) que permita aumentar la autonomía del sistema y su eventual restablecimiento después de fallas técnicas.
- Mejorar el desempeño de la aplicación para Smartphone que restrinja el acceso a usuarios no autorizados o que permita un acceso confiable y seguro a la red inalámbrica.
- Establecer un mecanismo de reset en la componente electrónica que asegure siempre un correcto inicio del sistema.

Conclusiones

Se diseñó y construyó un sistema electromecánico con dos grados de libertad para el seguimiento de la trayectoria solar. Además se elaboró una aplicación para teléfono celular que es utilizada como interface de usuario inalámbrica. El sistema desarrollado constituye una herramienta de evaluación para algoritmos de control. Las componentes electrónica, mecánica y software de operación se construyeron con base en herramientas básicas de desarrollo (microcontroladores, material crudo, software libre) lo cual facilita su eventual transferencia tecnológica.

Los experimentos realizados con el sistema comprueban su buen desempeño y viabilidad para usar esta tecnología en la construcción de calentadores solares o sistemas de generación eléctrica.

Del análisis de la información recolectada en los experimentos realizados, se identificaron oportunidades de mejora para trabajos futuros.

Especificaciones del equipo

Fuente de energía: 12VDC generados de manera autónoma mediante panel solar.

Consumo de energía: 0.5A.

Resolución en altitud: 129 pasos por grado.

Resolución en azimut: 7 pasos por grado

Interface inalámbrica de operación mediante aplicación para Smartphone.

Anexo A Diagramas electrónicos

Los archivos de diseño asistido por computadora se encuentran en la carpeta CADElectronica del CD anexo.

A.1 Diagrama esquemático

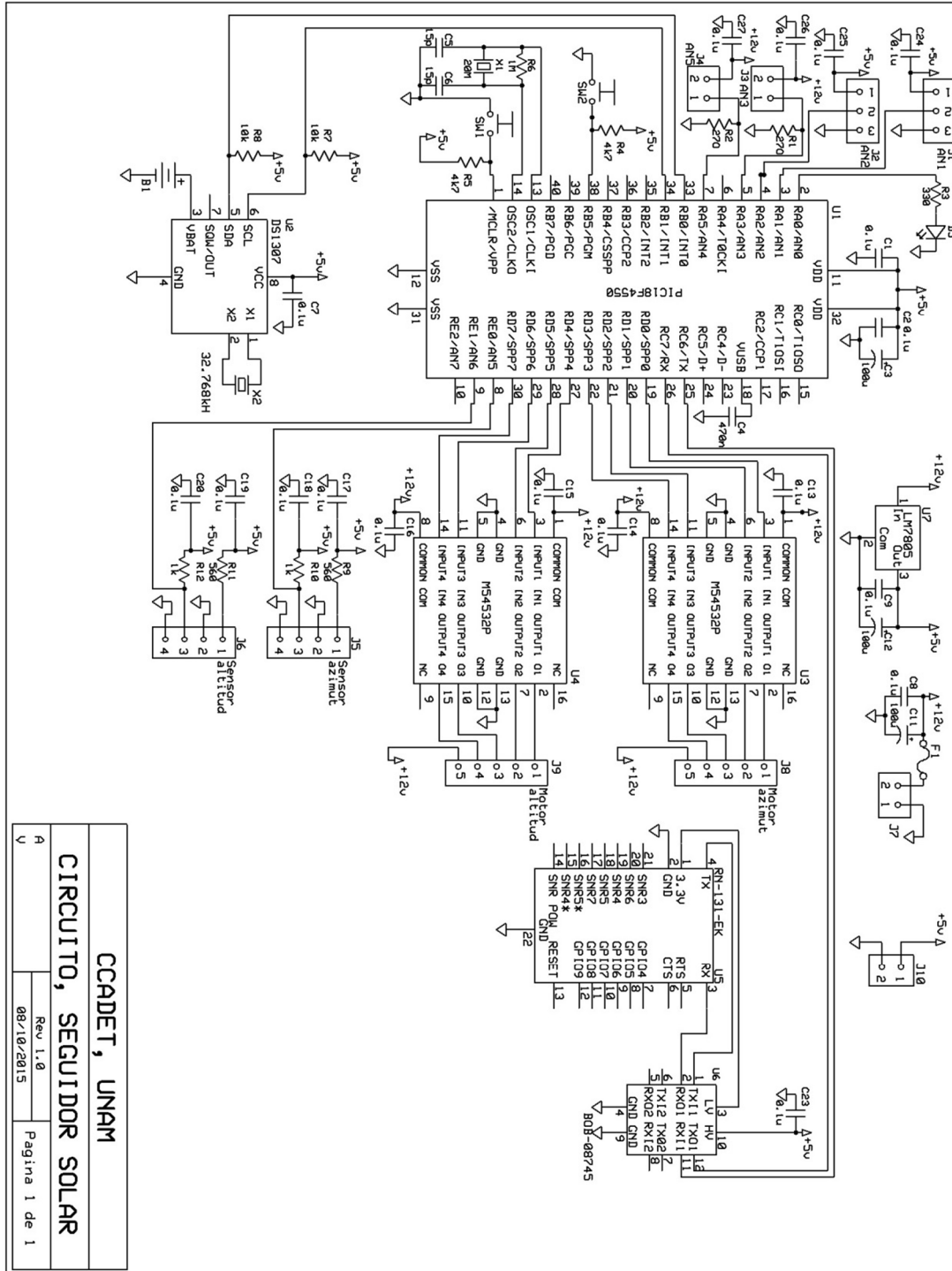


Figura 54. Dibujo esquemático.

A.2 Circuito impreso

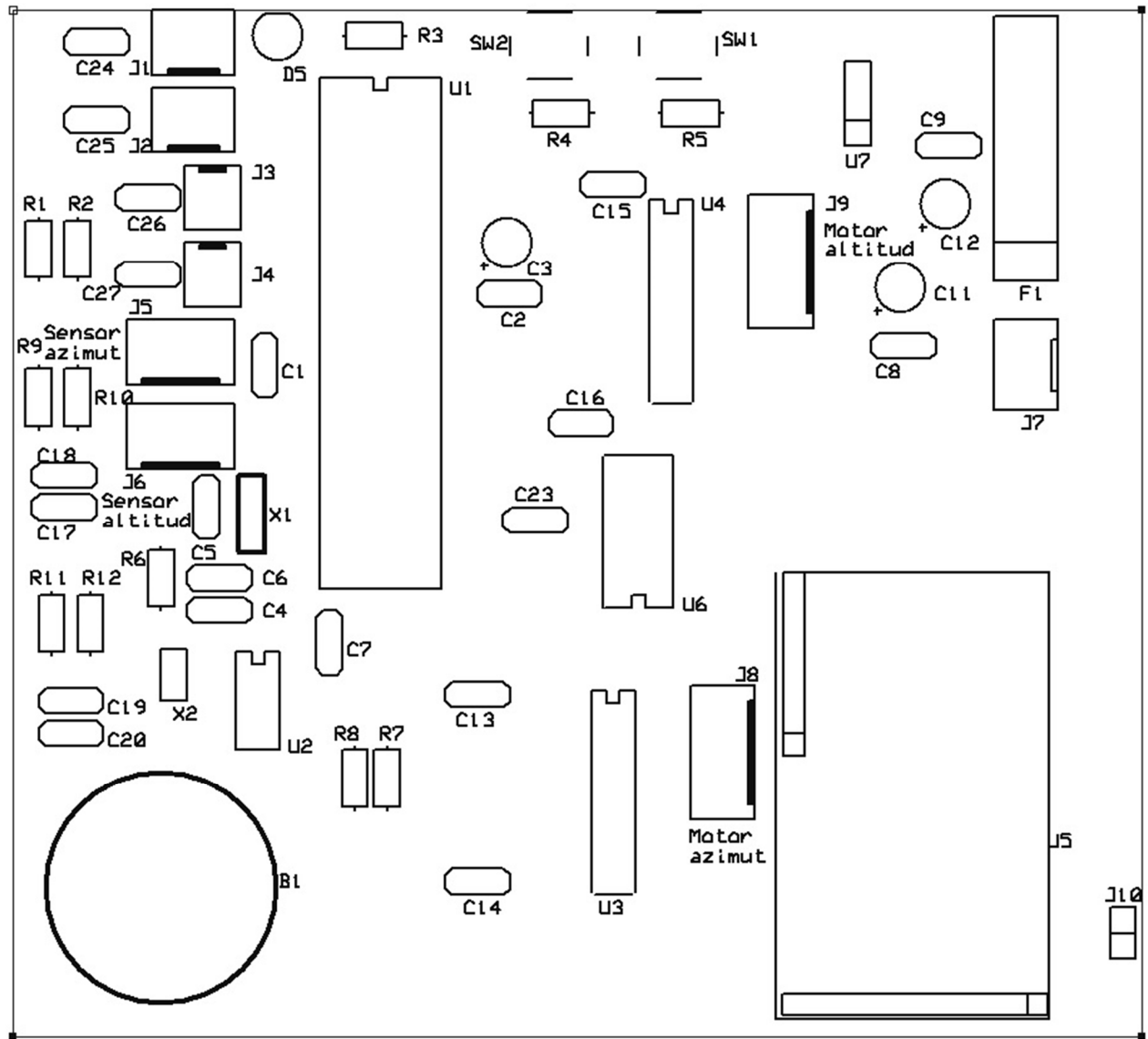


Figura 55. Circuito impreso, mascarilla.

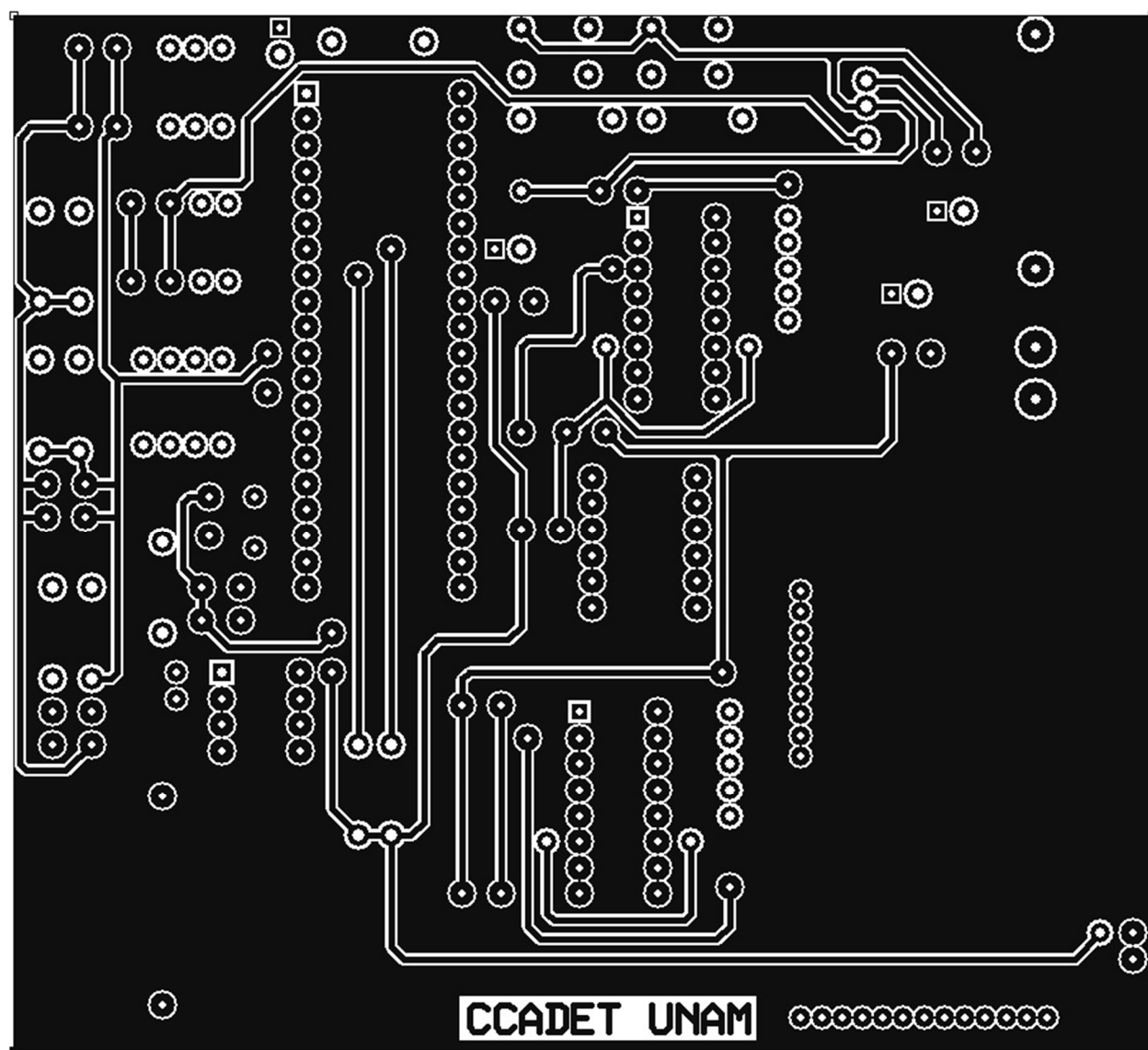


Figura 56. Circuito impreso, componentes.

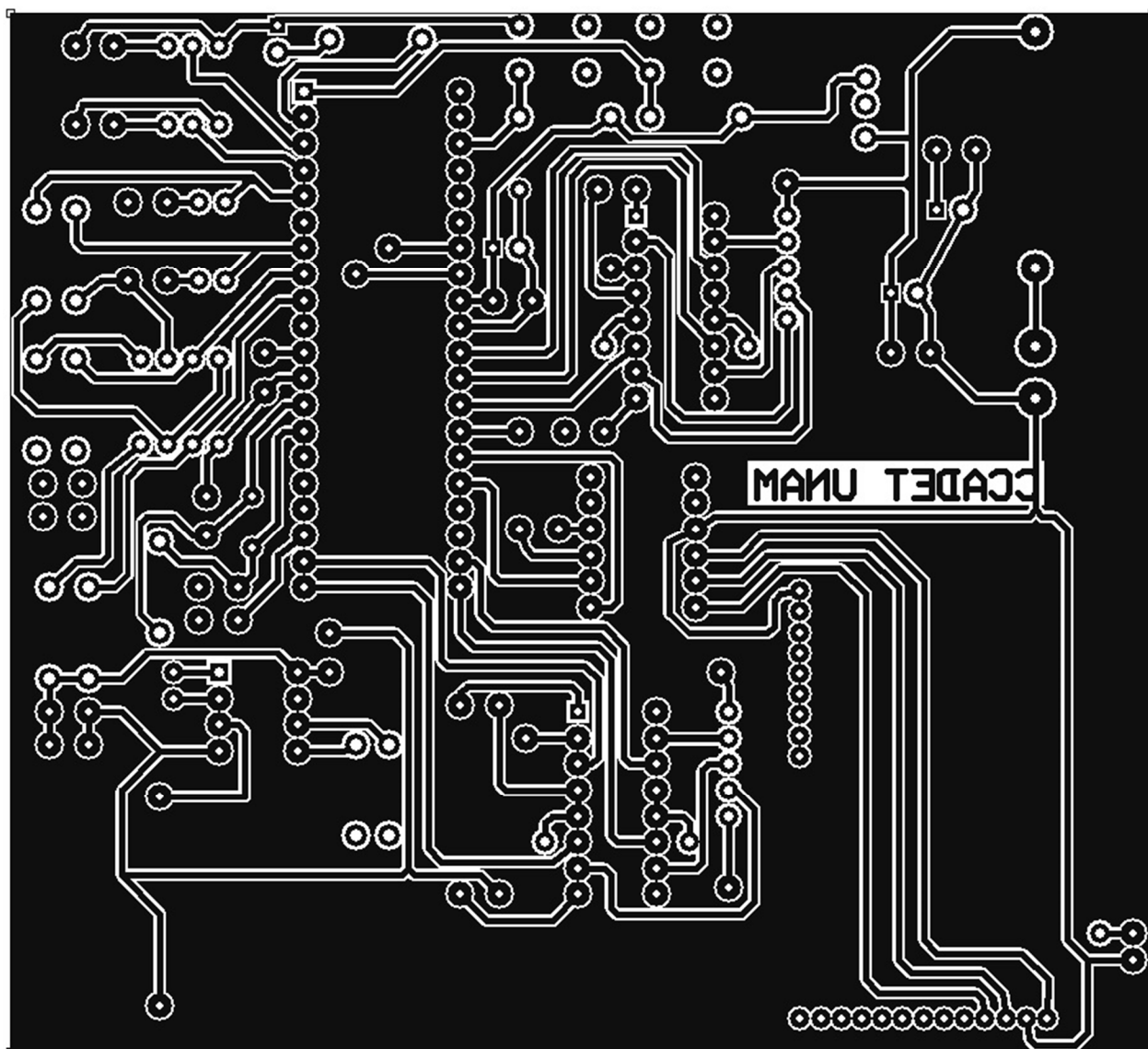


Figura 57. Circuito impreso, soldadura.

A.3 Lista de partes

ID	Descripción	Proveedor Num. Parte
B1	Base de pila	AG Electronica CR-BASE-20
C1	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C2	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C3	Capacitor electrolítico 100u	AG Electronica CE-100/10V-TEAPO
C4	Capacitor monolítico 470n	AG Electronica CM-.47/50V
C5	Capacitor cerámico 15p	AG Electronica CC-15/1000V-WELSON
C6	Capacitor cerámico 15p	AG Electronica CC-15/1000V-WELSON
C7	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C8	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C9	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C11	Capacitor electrolítico 100u	AG Electronica CE-100/50V-TEAPO
C12	Capacitor electrolítico 100u	AG Electronica CE-100/10V-TEAPO

C13	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C14	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C15	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C16	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C17	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C18	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C19	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C20	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C23	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C24	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C25	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C26	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
C27	Capacitor monolítico 0.1u	AG Electronica CM-.1/50V
D5	Led	Steren 5/AZUL MEGA
F1	Fusible europeo 0.5	Ultimo recurso
J1	Molex 3 pines	Ultimo recurso
J2	Molex 3 pines	Ultimo recurso
J3	Molex 2 pines	Ultimo recurso
J4	Molex 2 pines	Ultimo recurso
J5	Molex 4 pines	Ultimo recurso
J6	Molex 4 pines	Ultimo recurso
J7	Conector 2 tornillos	Steren TRT-02
J8	Molex 5 pines	Ultimo recurso
J9	Molex 5 pines	Ultimo recurso
J10	Molex 2 pines	Ultimo recurso
R1	Resistencia 27	AG Electronica RC-270/1/4
R2	Resistencia 27	AG Electronica RC-270/1/4
R3	Resistencia 330	AG Electronica RC-330/1/4
R4	Resistencia 4k7	AG Electronica RC-4k7/1/4
R5	Resistencia 4k7	AG Electronica RC-4k7/1/4
R6	Resistencia 1M	AG Electronica RC-1M/1/4
R7	Resistencia 10k	AG Electronica RC-10k/1/4
R8	Resistencia 10k	AG Electronica RC-10k/1/4
R9	Resistencia 560	AG Electronica RC-560/1/4
R10	Resistencia 1k	AG Electronica RC-1k/1/4
R11	Resistencia 560	AG Electronica RC-560/1/4
R12	Resistencia 1k	AG Electronica RC-1k/1/4
SW1	Push botton	Steren AU-101
SW2	Push botton	Steren AU-101
U1	PIC18F4550	AG Electronica PIC18F4550-I/P
U2	DS1307	AG Electronica DS1307
U3	M54532P	AG Electronica M54532P
U4	M54532P	AG Electronica M54532P
U5	RN-131-EK	Newark 47W7781
U6	BOB-08745	AG Electrónica BOB-08745
U7	LM7805	AG Electronica L7805CV
X1	Cristal 20M	AG Electronica 20MHZ-MINI
X2	Cristal32.768kHz	AG Electronica 32.768KHz
	Panel solar	AG Electronica SM-12V/10W-SU
	Batería recargable	AG Electronica PS1230
	Controlador de carga	AG Electronica RYPV-A2-5A-12/24V

Tabla 24. Lista de partes.

Anexo B Planos de construcción

Se muestran los planos de construcción y lista de materiales.

B.1 Planos

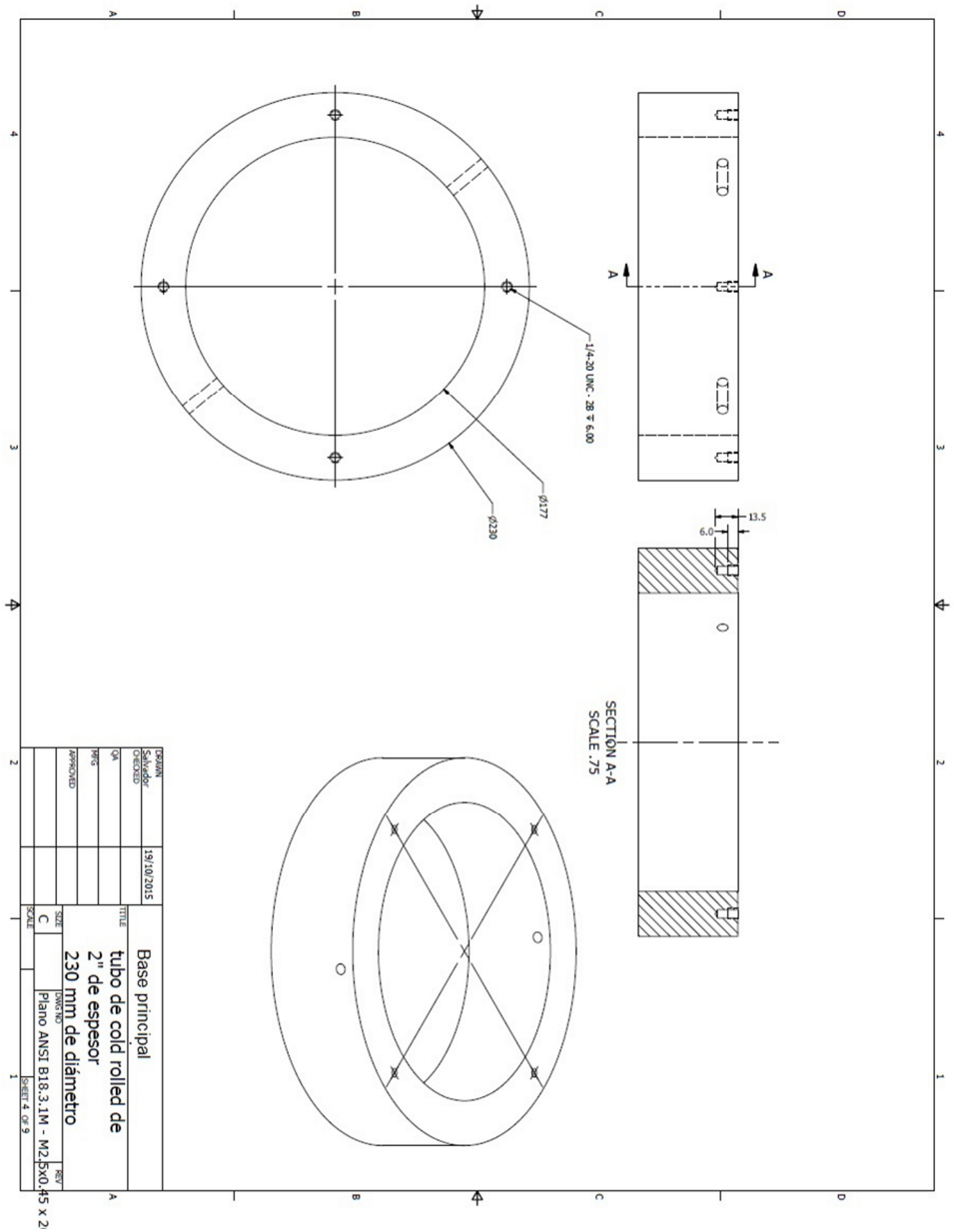


Figura 58. Base principal.

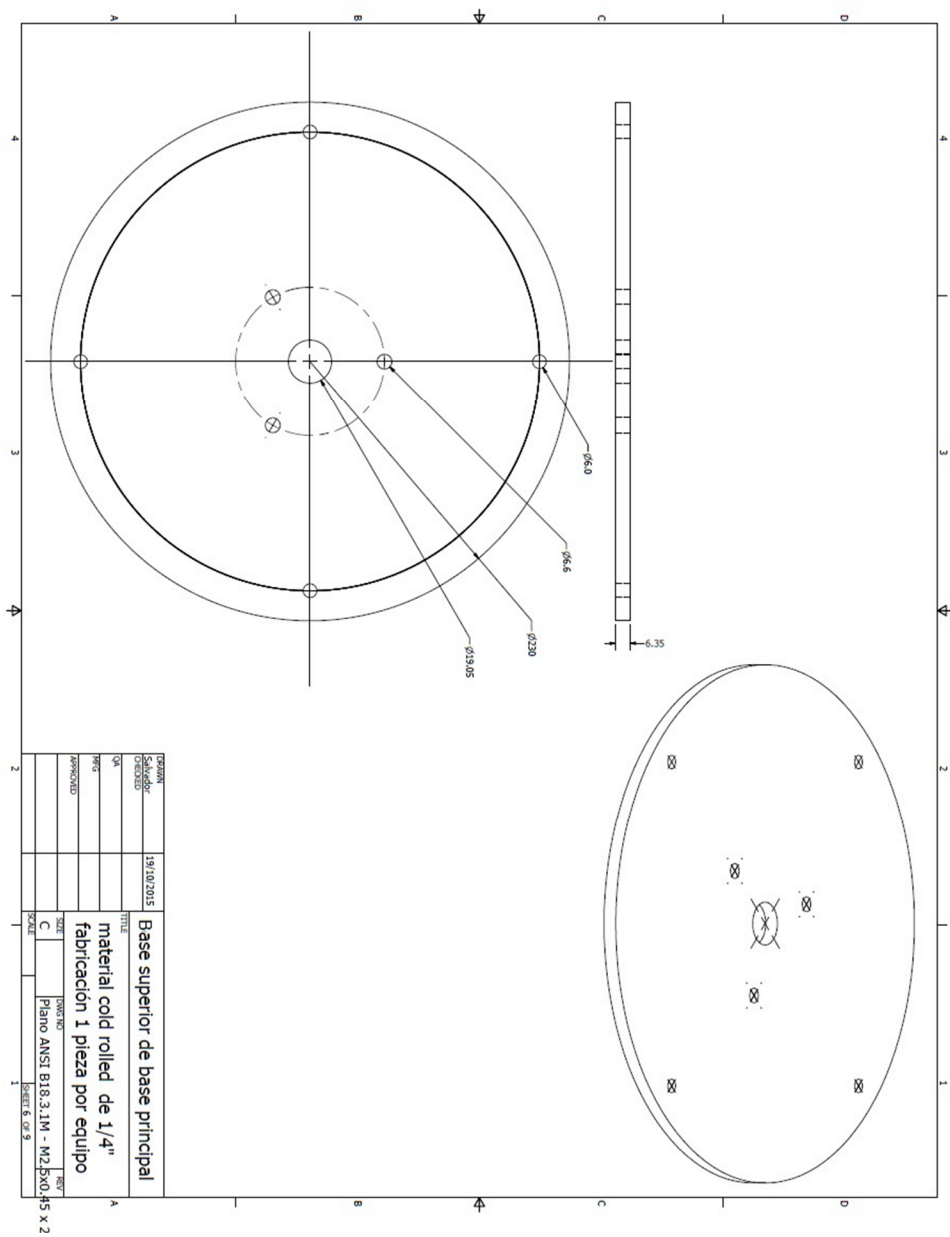


Figura 59. Base superior de la base principal.

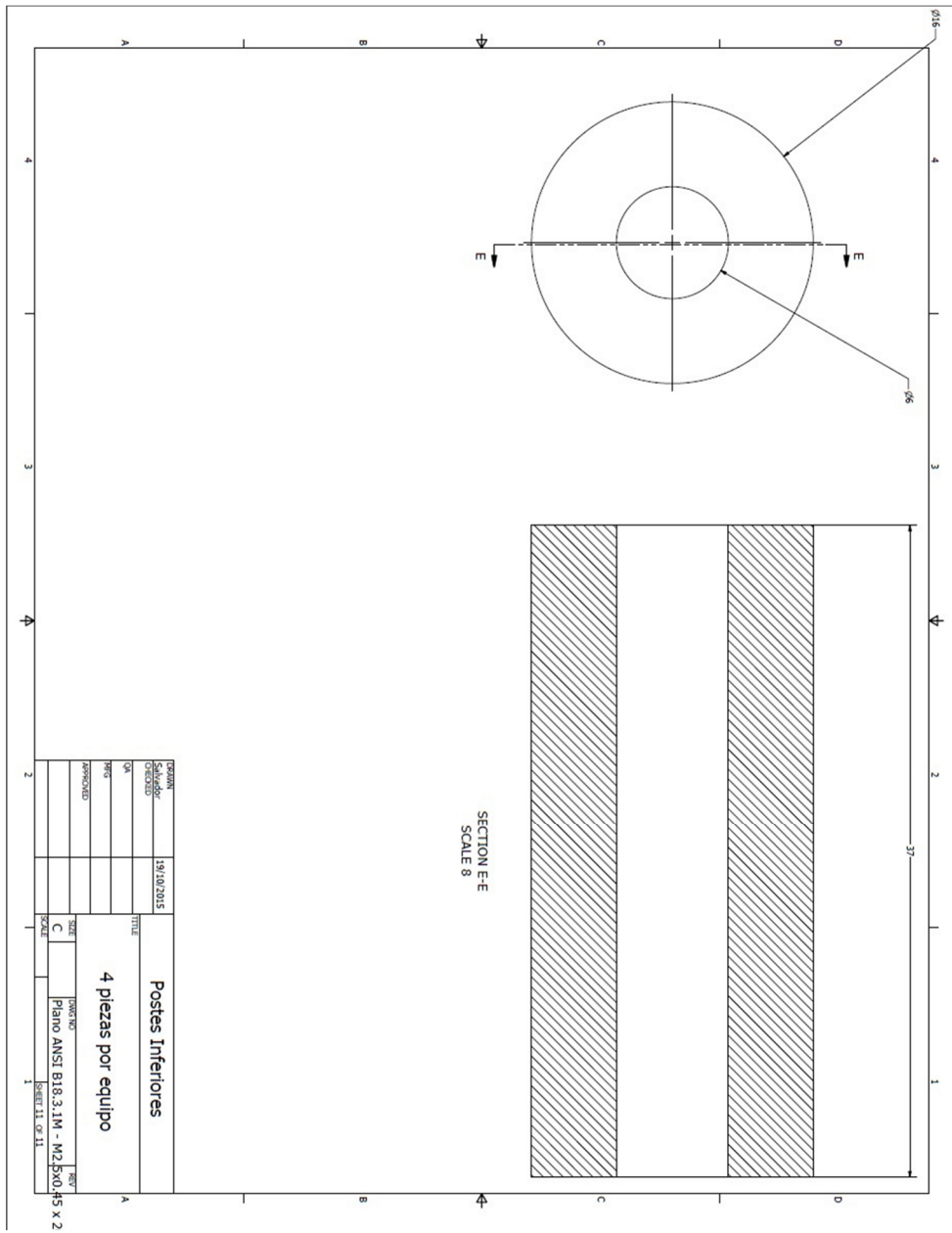


Figura 60. Postes inferiores.

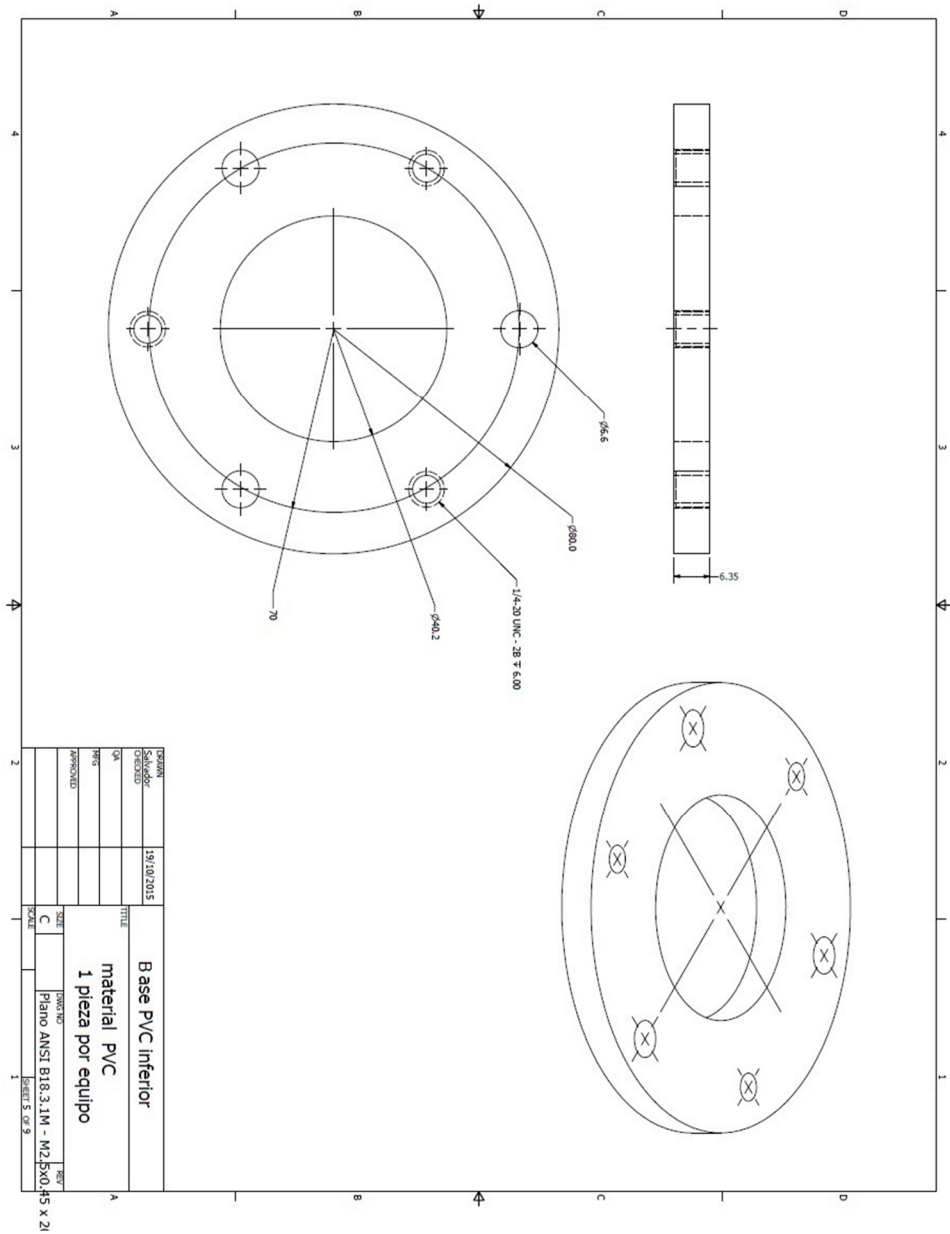


Figura 61. Base pvc inferior.

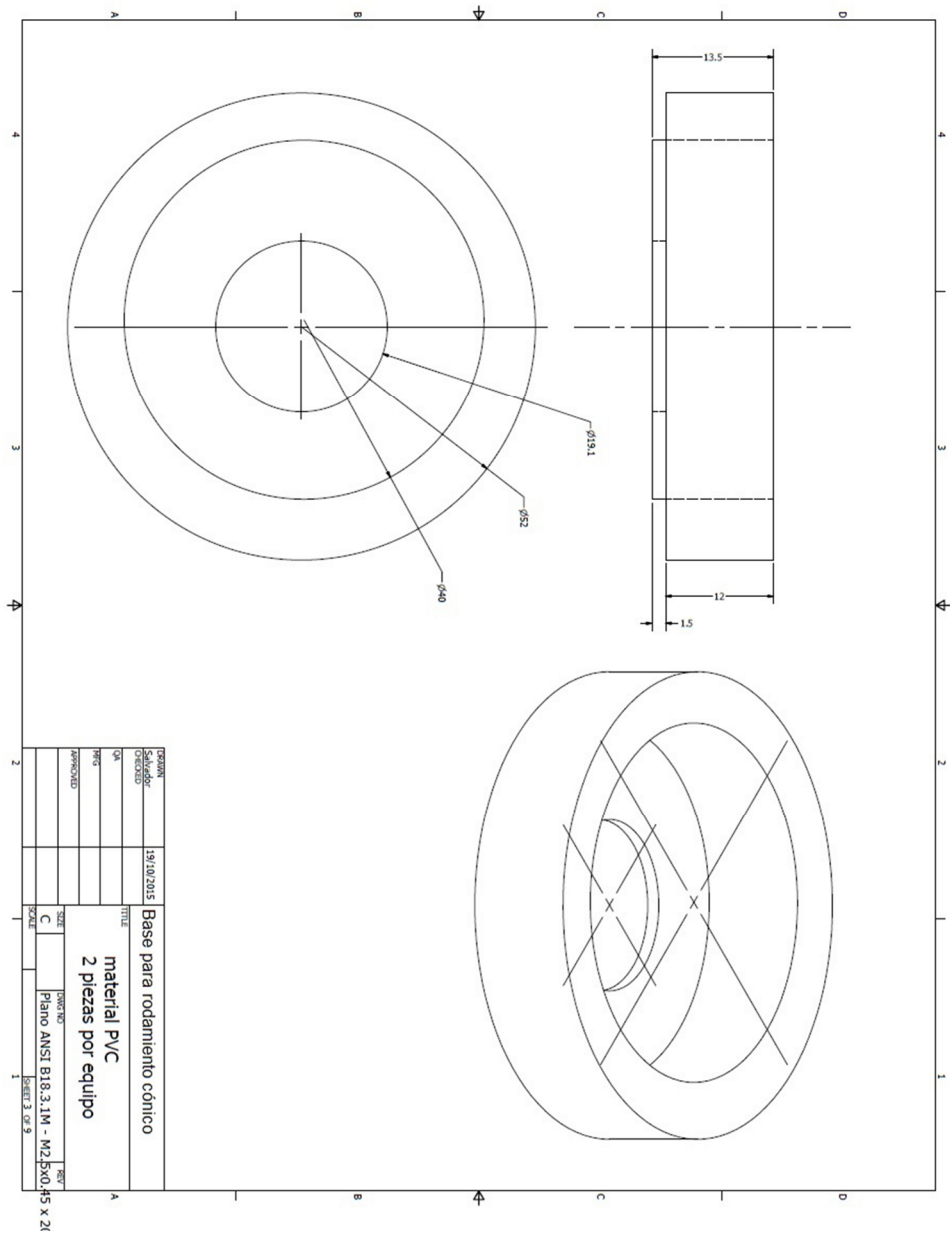


Figura 62. Base para rodamiento cónico.

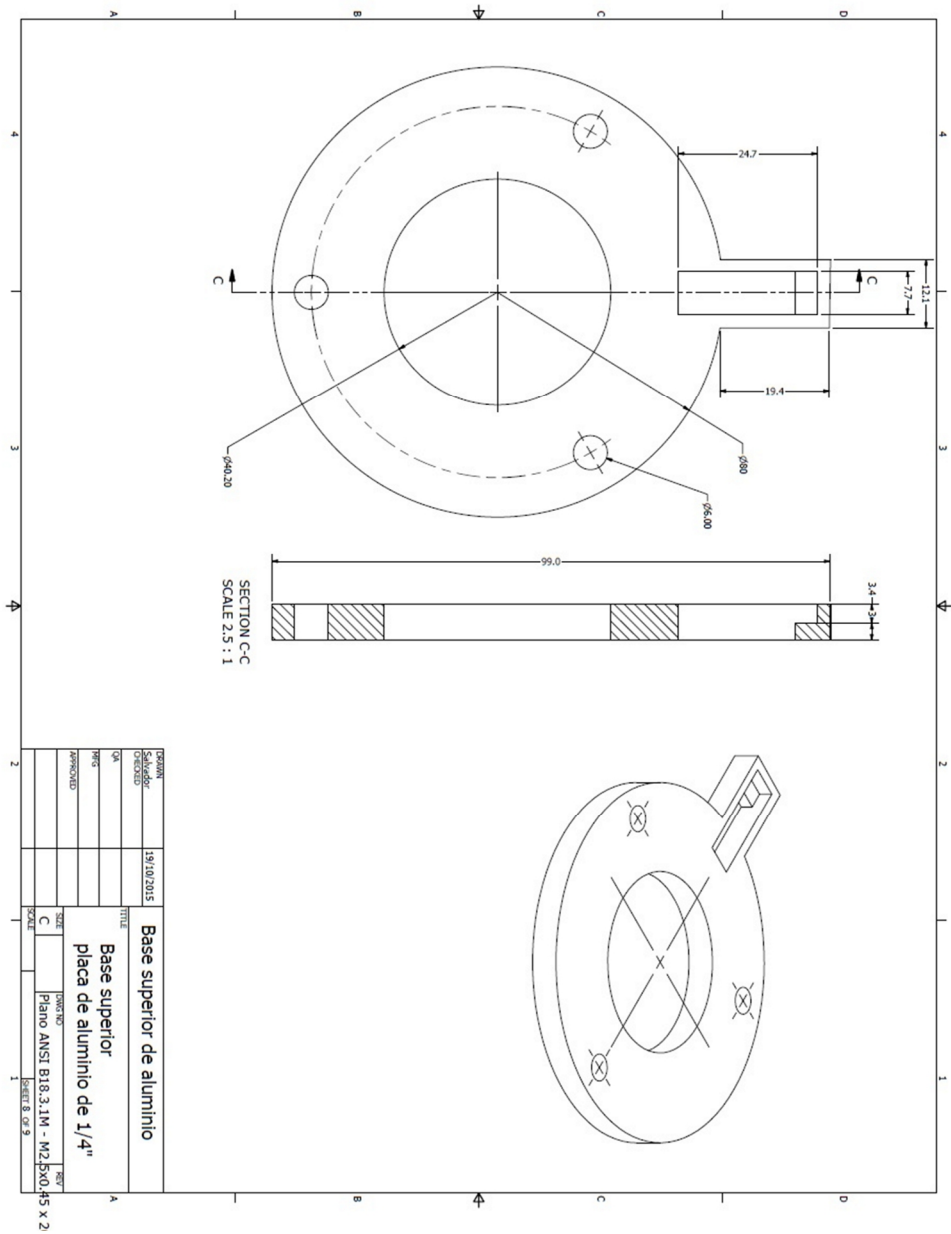


Figura 63. Base superior de aluminio.

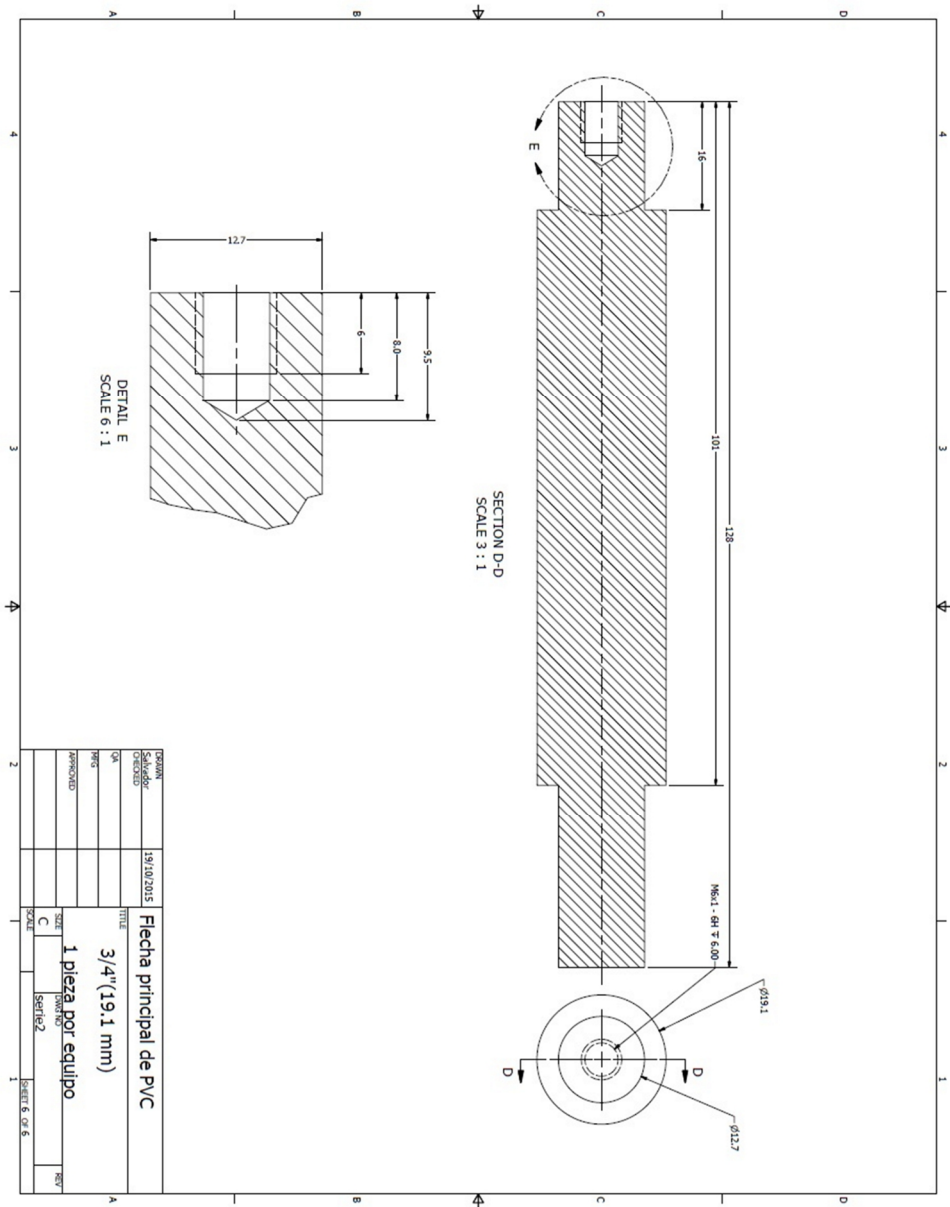


Figura 65. Flecha principal de pvc.

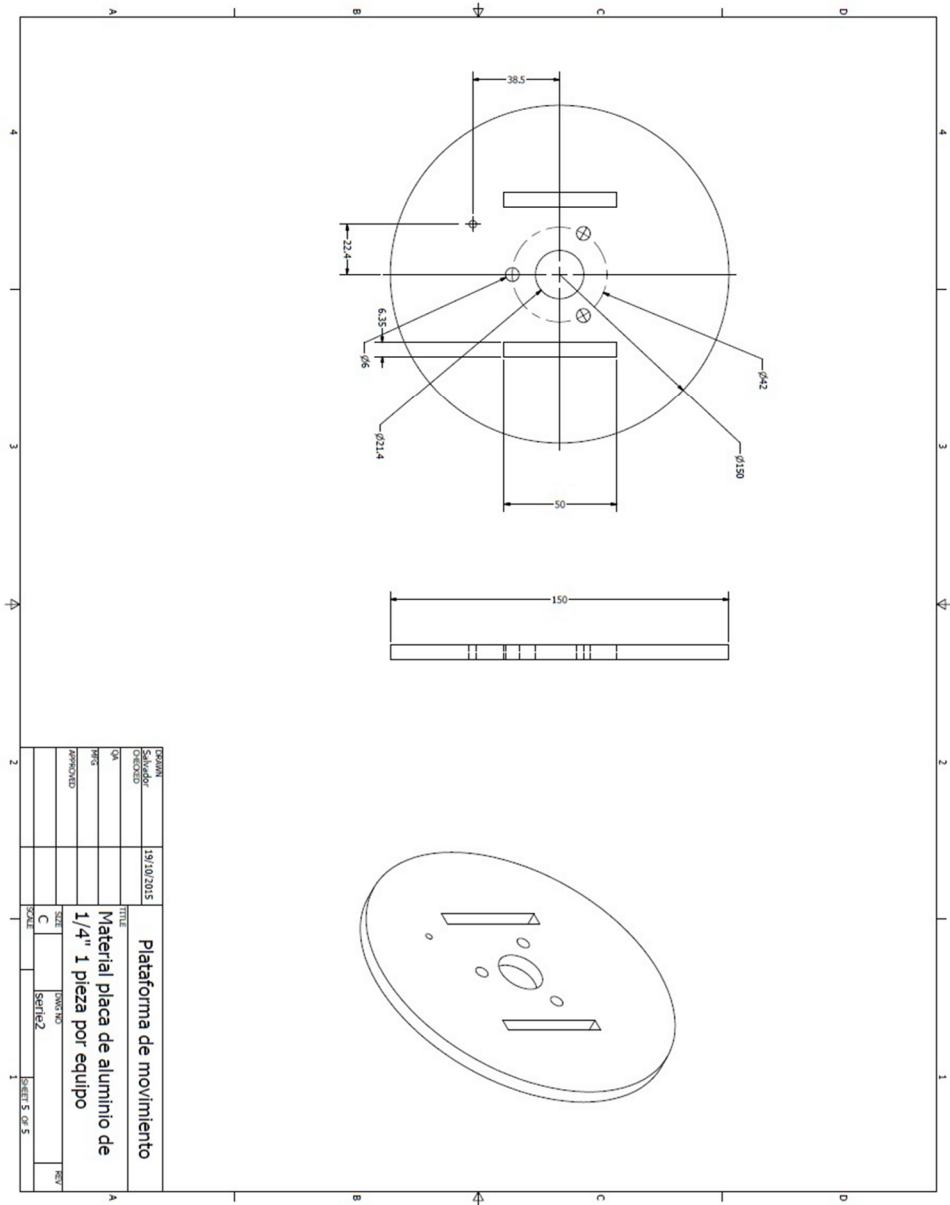


Figura 68. Plataforma de movimiento.

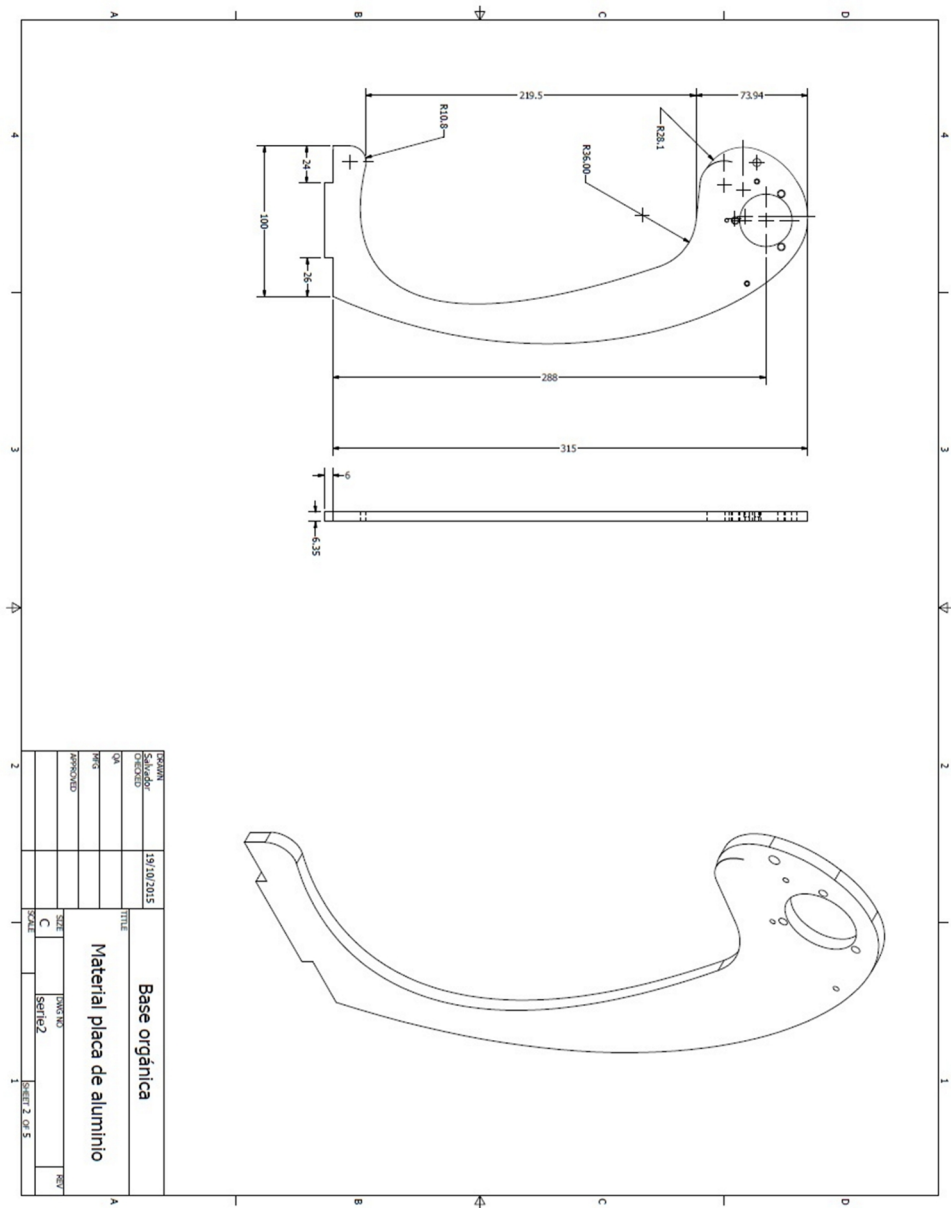


Figura 69. Base orgânica.

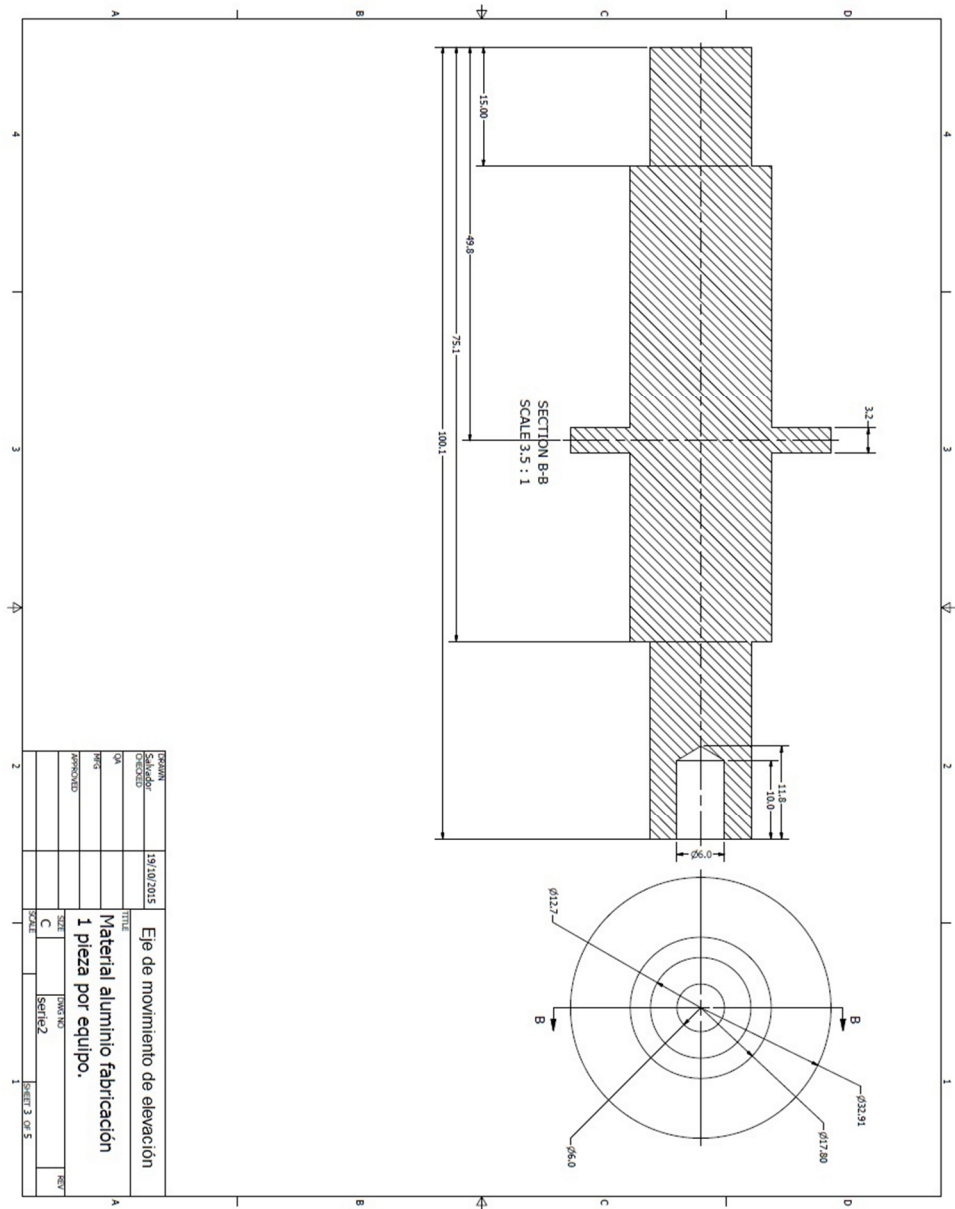


Figura 71. Eje de movimiento de elevación.

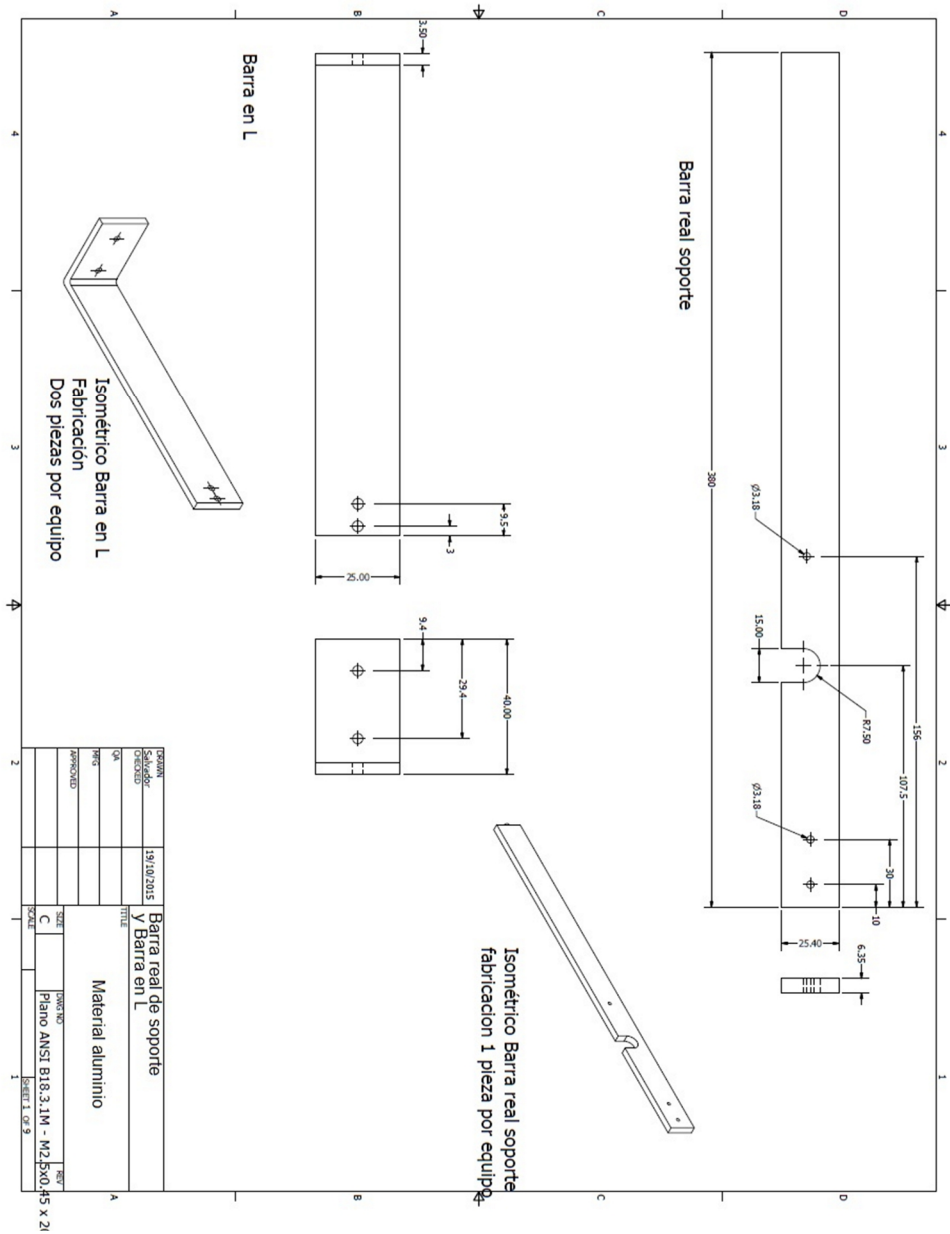


Figura 72. Barra real de soporte y Barras en L.

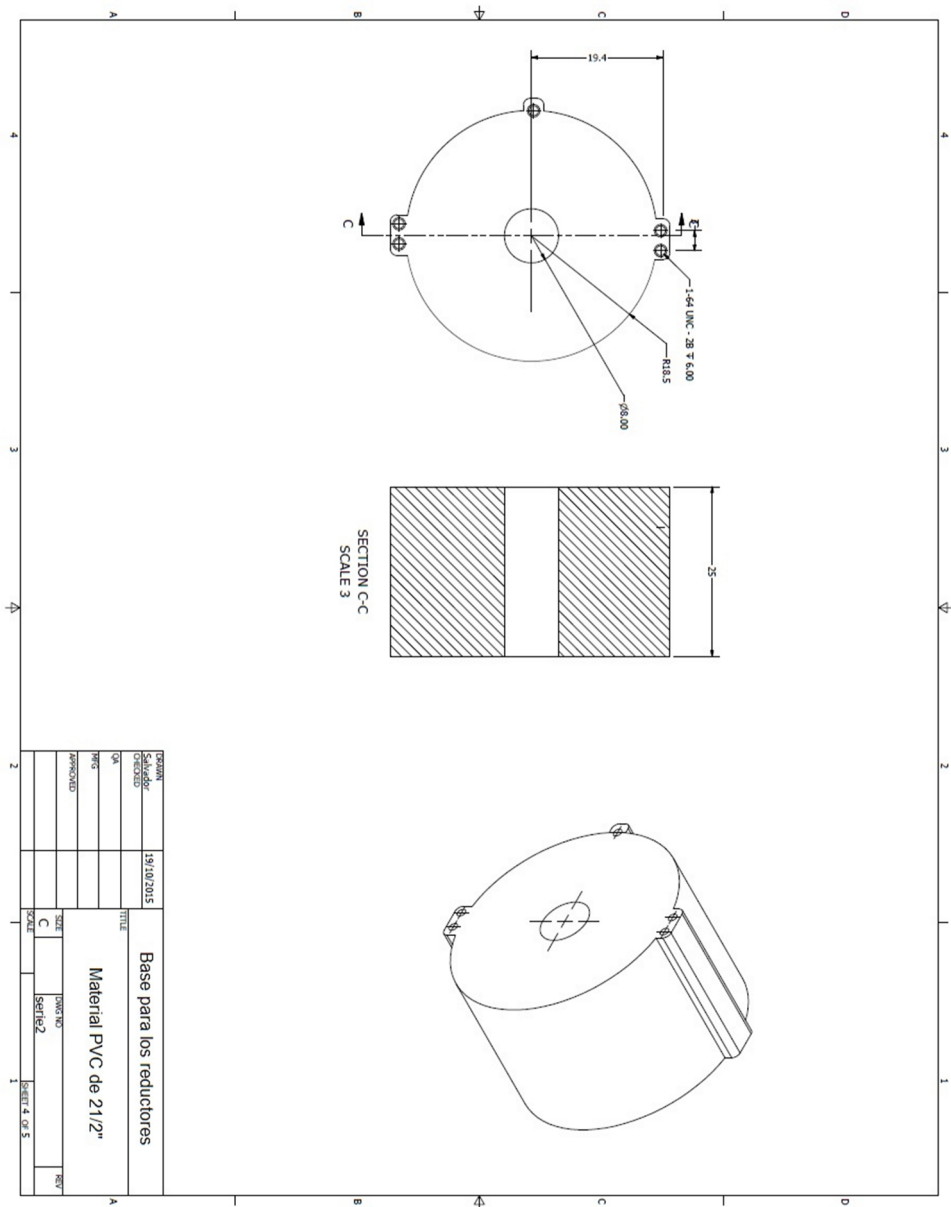


Figura 73. Base para los reductores.

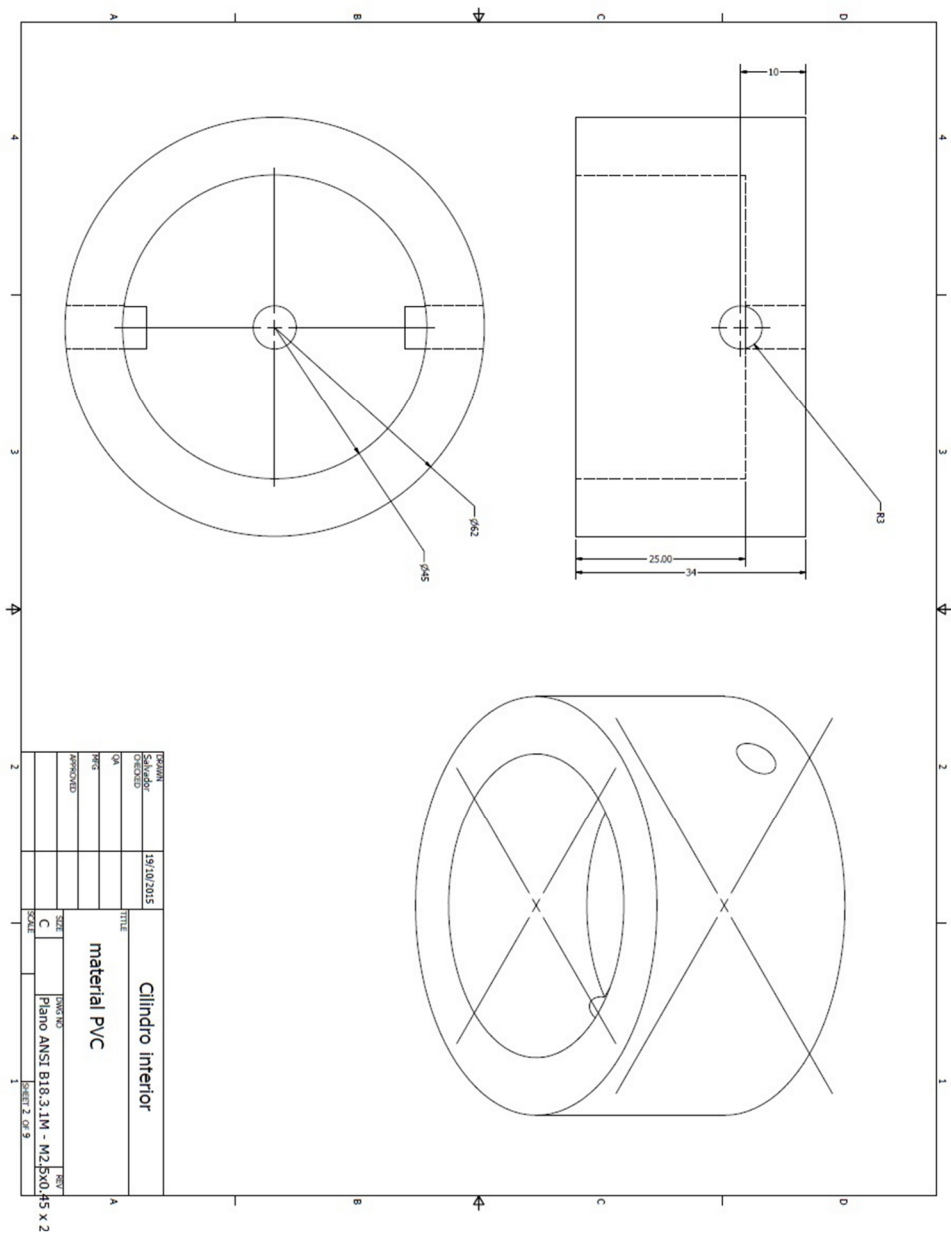


Figura 74. Cilindro interior.

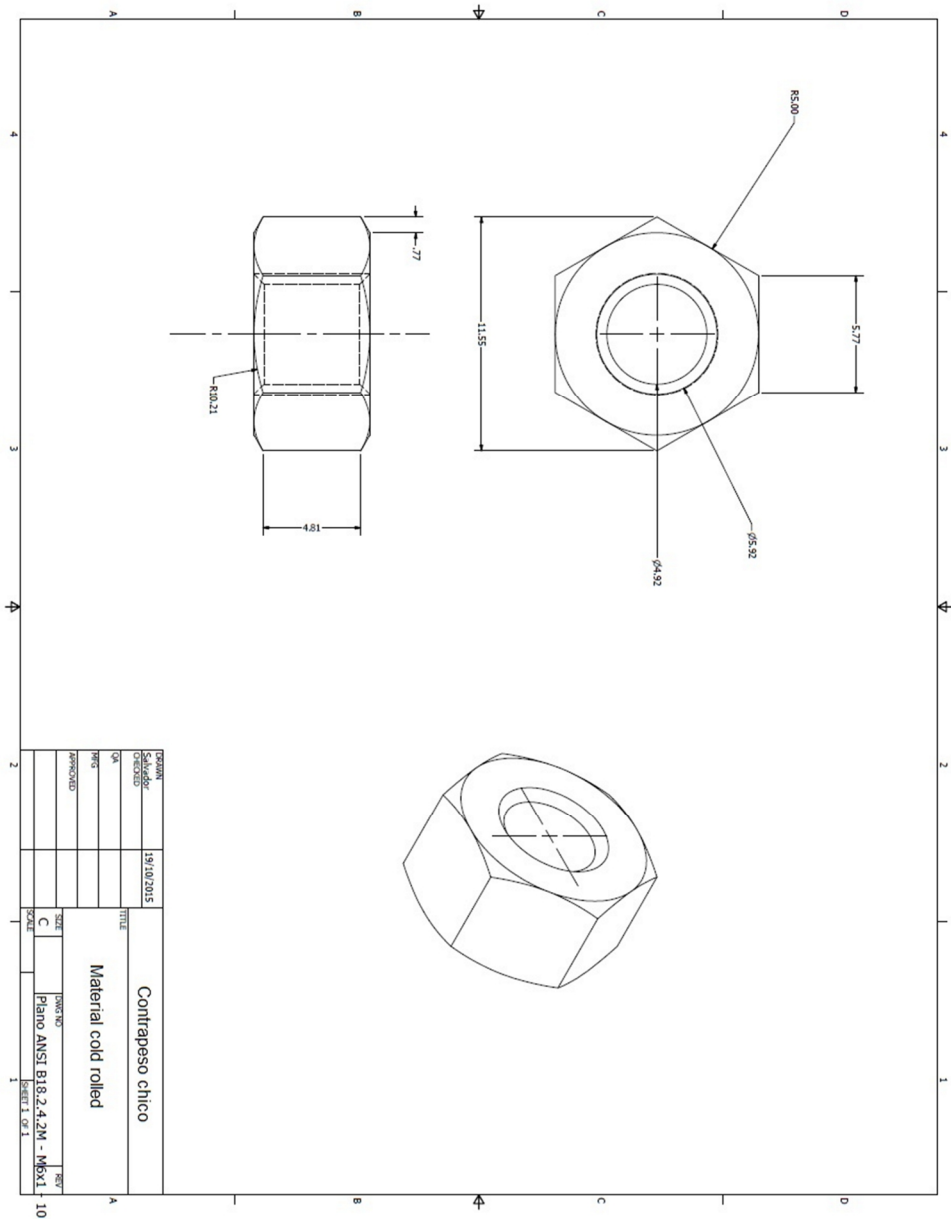


Figura 75. Contrapeso chico.

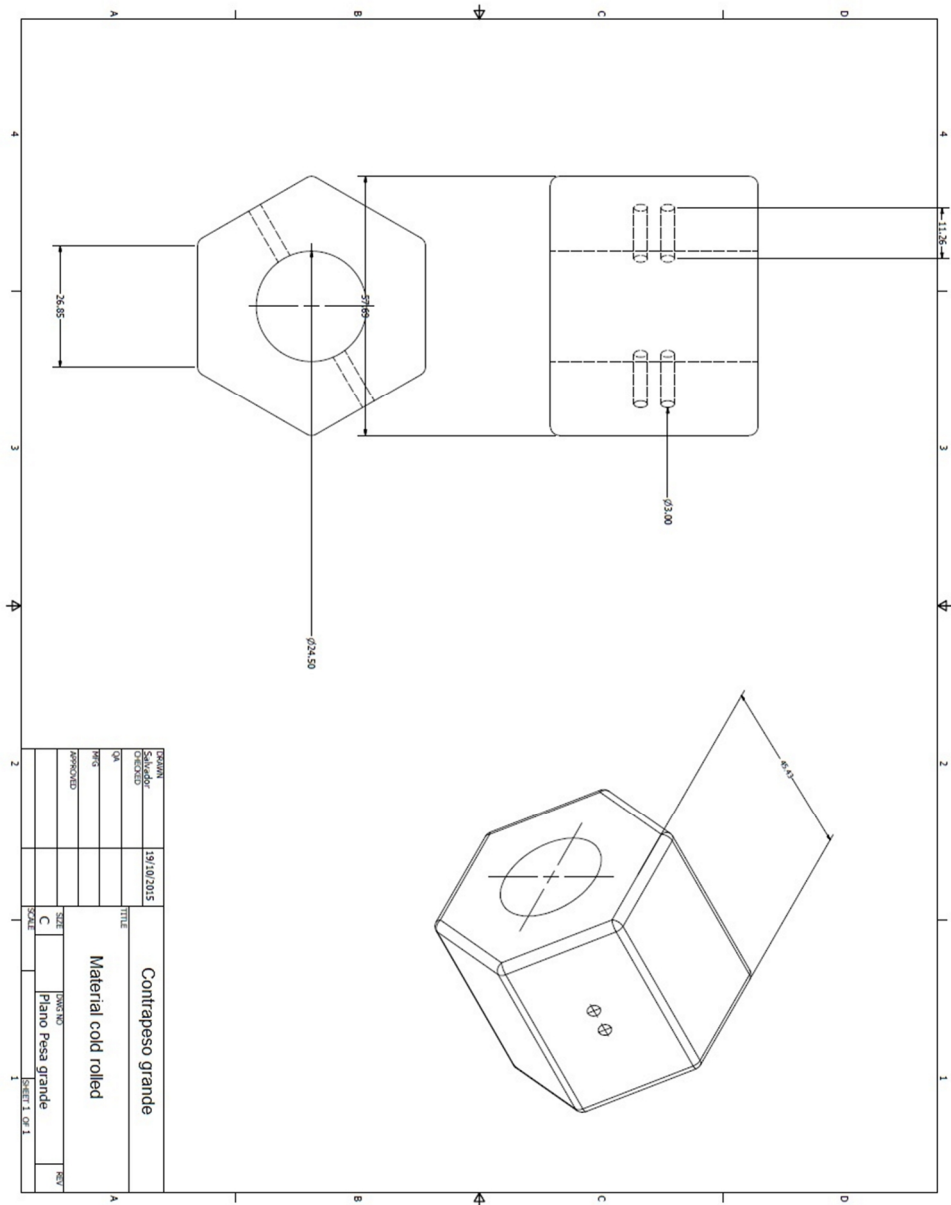


Figura 76. Contrapeso grande.

7	5	Varilla de soporte	Varilla roscada comercial de 1/4"
8	1	Base superior de aluminio	Placa de aluminio de 1/4"
9	1	Base superior de PVC para el rodamiento radial	Barra de PVC de 2" de diametro
10	1	Flecha principal de pvc	Barra de PVC de 1/2"
11	1	Buje para el motor de azimut	Buje de latón de 3/4"
13	1	Moto-reductor de azimut	Moto-reductor
15	2	Sensor de posición en azimut	Sensor pieza comercial
16	1	Brida para la base de la plataforma	Barra de PVC de 2"
17	1	Plataforma de movimiento	Placa de aluminio de 1/4"
18	2	Base orgánica	Placa de aluminio de 1/4"
19	2	Rodamiento comercial	DIN 615 - E 11 - 11 x 32 x 7
20	2	Brida de soporte	Barra de PVC de 2"
21	1	Eje de movimiento de elevación	Barra de aluminio de 1 1/2"
22	1	Barra real de soporte	Barra de cold rolled de 1/4" espesor
23	2	Barras en L	Barra de cold rolled de 1/4" espesor
24	1	Moto-reductor de elevación	Moto-reductor
25	1	Base para el moto-reductor de elevación	Barra de PVC de 2"
26	2	Cilindro interior	Barra de PVC de 1"
27	1	Contrapeso chico	Hexágono de aluminio de 2"
28	1	Contrapeso grande	Hexágono de aluminio de 2"
29	1	Panel solar	celda comercial
30	4	Tornillo	Screw GB/T 70.1 M5 x 50
31	3	Tornillo	Screw GB/T 70.1 M6 x 14
32	1	Tornillo	ISO 2341 - A - 3 x 18
33	1	Tornillo	ISO 2341 - A - 3 x 22
34	3	Tornillo	ANSI B18.3.1M - M5x0.8 x 16
35	10	Tornillo	ANSI B18.3.1M - M2.5x0.45 x 20
36	8	Tornillo	ANSI B18.2.4.2M - M6x1
37	2	Tornillo	ANSI B18.3.1M - M2.5x0.45 x 10
38	1	Tapadera inferior	Barra de PVC

Tabla 25. Lista de partes.

Referencias

- Android Developer (2015), "Android Developer Tools" [On line]
Available at: <https://developer.android.com/tools/help/adt.html> [Accessed on Octubre 8, 2015].
- Kalogirou S. A. (2009) *Solar Energy Engineering*. Academic Press, USA, p. 755.
- Kassem A. & Hamad M. (2011), *A microcontroller-based multi-function solar tracking system*, Systems Conference (SysCon), IEEE International Digital Object Identifier, p. 13–16.
- Maxim Integrated Products (2015) *DS1307 64 x 8, Serial, I2C Real-Time Clock*, Maxim Integrated Products, USA, p. 14.
- Microchip. (2007), *PIC18F2455/2550/4455/4550 Data Sheet*, Microchip, USA, p. 430.
- Mitsubishi Semiconductor (1999) *M54532p/fP 4-unit 1.5a darlington transistor array with clamp diode*, Mitsubishi Semiconductor, USA, p. 4.
- Montero MR. (2012) *Desarrollo de aplicaciones para Android*, Editorial Ra-Ma, España, p. 350.
- POV-Ray. (2013) "POV-Ray2 en Persistence of Vision". [On line]
Available at: <http://www.povray.org/>. [Accessed on Agosto 22, 2014].
- Roving Networks (2012) *WiFly Evaluation Kit USER MANUAL*. Roving Networks, USA, p. 22.
- SparkFun Electronics (2015), "Using the Logic Level Converter" [On line]
Available at: <https://learn.sparkfun.com/tutorials/using-the-logic-level-converter> [Accessed on Octubre 8, 2015].