



**UNIVERSIDAD NACIONAL AUTONOMA  
DE  
MEXICO**

**FACULTAD DE INGENIERÍA**

**Sistema electrónico para el seguimiento  
de la trayectoria solar**

**T E S I S**

QUE PARA OBTENER EL TITULO DE:

**INGENIERO EN ELECTRICA ELECTRONICA**

P R E S E N T A

**López Hernández Victor  
Ríos Pérez Arturo**

DIRIGIDA POR

**Benjamín Valera Orozco**

CIUDAD UNIVERSITARIA  
Diciembre de 2015





Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



# Agradecimientos

A Juan Salvador Pérez Lomeli por el desarrollo y construcción del mecanismo de seguimiento solar.

Al Taller de Construcción de Prototipos del CCADET UNAM por las facilidades otorgadas en la construcción del mecanismo.

Al M. en I. Rigoberto Nava Sandoval por el diseño del mecanismo.



# Índice

<b>Resumen de la tesis</b>	<b>1</b>
<b>Introducción</b>	<b>3</b>
Definición del problema	3
Antecedentes	3
Descripción del problema a resolver	3
Relevancia y justificación	4
Alcance y limitaciones	4
Relación con otras áreas	4
Método	4
Resultados esperados	5
<b>1. Conceptos básicos</b>	<b>7</b>
1.1. Energías renovables	7
1.1.1. Energía solar	7
1.1.2. Energía eólica	9
1.2. Trayectoria del sol	11
1.2.1. Trayectoria tierra-sol	11
1.2.2. Trayectoria aparente del sol	13
1.2.3. Ajuste del tiempo	19
1.2.4. Ángulos solares	20
1.2.5. Ejemplo de aplicación	21
1.3. Microcontrolador PIC18F4550	25
1.3.1. Especificaciones generales	25
1.3.2. Herramientas de desarrollo	27
1.4. Android	29
1.5. Dispositivos electrónicos empleados	32

---

1.5.1. Módulo LAN inalámbrico RN-131C	32
1.5.2. Driver M54532P/FP	33
1.5.3. Convertidor de nivel lógico bidireccional BOB-08745	33
1.5.4. Reloj de tiempo real DS1307	34
1.5.5. Sensor de temperatura LM35	35
1.5.6. Optoacoplador H22A1	35
1.5.7. Batería recargable	36
1.5.8. Celda solar	37
1.5.9. Controlador de carga solar	37
<b>2. Descripción del proyecto</b>	<b>39</b>
2.1. Esquema general	39
2.2. Subsistema electrónico	41
2.2.1. Etapa de radiofrecuencia	42
2.2.2. Etapa de registro del tiempo	44
2.2.3. Etapa de acondicionamiento de señales	45
2.2.4. Etapa de control	46
2.2.5. Etapa impulsora de motores a pasos	48
2.2.6. Etapa para el sensor de posición	49
2.2.7. Etapa de alimentación de energía	49
2.2.8. Firmware del microcontrolador	51
2.3. Subsistema mecánico	54
2.4. Descripción del software de operación en el ámbito del operador	57
2.5. Descripción del software de operación en el ámbito del programador	62
2.5.1. Descripción general del proyecto	62
2.5.2. Descripción del código fuente	64
<b>3. Resultados</b>	<b>79</b>
3.1. Resultados	79
3.1.1. Prototipo desarrollado	79
3.1.2. Comparación teórica contra real para los ángulos solares	81
3.1.3. Desempeño en posición del prototipo	82
3.2. Discusión	84
3.3. Trabajo a futuro	85
<b>Conclusiones</b>	<b>87</b>

<b>Bibliografía</b>	<b>89</b>
<b>Anexos</b>	<b>91</b>
A. Código fuente MatLab	91
B. Código fuente PovRay	94
C. Diagramas electrónicos	96
C.1. Diagrama esquemático	97
C.2. Circuito impreso	98
C.3. Lista de partes	100
D. Planos de construcción	102
D.1. Planos	102
D.2. Lista de material	122





# Índice de figuras

Figura 1. Método a emplear	5
Figura 1.1. Energía solar fotovoltaica	8
Figura 1.2. Zonas potenciales para el desarrollo de proyectos de energía eólica	10
Figura 1.3. Trayectoria tierra-sol	12
Figura 1.4. Longitud y latitud de la tierra	13
Figura 1.5. Modelo de la trayectoria aparente del sol	13
Figura 1.6. Relaciones <i>Fecha-N-MesP</i>	14
Figura 1.7. Solsticios	15
Figura 1.8. Traslación de la trayectoria aparente del sol	16
Figura 1.9. Radio de la trayectoria solar	16
Figura 1.10. Trayectoria solar en el plano	18
Figura 1.11. Rotación de la trayectoria aparente del sol	18
Figura 1.12. Trayectoria aparente del sol en los ejes coordenados	19
Figura 1.13. Ángulos de interés	21
Figura 1.14. Ejemplo de aplicación (ambiente gráfico)	25
Figura 1.15. Esquema de pines del PIC18F4550	26
Figura 1.16. Compilador CCS	29
Figura 1.17. Programación para Android	31
Figura 1.18. Módulo WiFi	32
Figura 1.19. Driver de corriente directa	33
Figura 1.20. Convertidor de nivel lógico	34
Figura 1.21. Reloj de tiempo real	34
Figura 1.22. Sensor de temperatura	35
Figura 1.23. Sensor de posición	36
Figura 1.24. Batería recargable	36
Figura 1.25. Celda solar	37
Figura 1.26. Controlador de carga solar	38
Figura 2.1. Esquema general del seguidor solar	39
Figura 2.2. Configuración del módulo inalámbrico de Wi-Fi	42
Figura 2.3. Modos de operación del módulo inalámbrico RN-131	44
Figura 2.4. Configuración del reloj de tiempo real, DS1307	44
Figura 2.5. Configuración del sensor de temperatura LM35	45
Figura 2.6. Convertidor de nivel BOB-08745	46
Figura 2.7. Configuración del PIC18F4550	47

Figura 2.8. Configuración del dispositivo M54532P	48
Figura 2.9. Configuración del optoacoplador para el posicionamiento del sistema	49
Figura 2.10. Conexión de la etapa de alimentación	50
Figura 2.11. Rutina de inicio	51
Figura 2.12. Rutina principal	52
Figura 2.13. Interrupción del TIMER0	53
Figura 2.14. Subsistema mecánico	54
Figura 2.15. Pantalla principal de la aplicación “Seguidor solar”	57
Figura 2.16. Pantalla para el despliegue de temperatura	58
Figura 2.17. Pantalla para el despliegue de los convertidores analógico a digital	59
Figura 2.18. Pantalla para el ajuste del tiempo	60
Figura 2.19. Diálogo de confirmación para el ajuste del tiempo	60
Figura 2.20. Pantalla para el ajuste de la posición	61
Figura 2.21. Diálogo de confirmación para el ajuste de la posición	62
Figura 2.22. Estructura de la aplicación en Android	62
Figura 2.23. Clases de la aplicación	64
Figura 2.24. Elementos de la Clase AngleMeter	65
Figura 2.25. Métodos de la clase BasicMeter	66
Figura 2.26. Métodos y atributos de la clase ConfigActivity	66
Figura 2.27. Elementos de la clase GPS	69
Figura 2.28. Elementos de la clase MainActivity	71
Figura 2.29. Elementos de la clase Meter	73
Figura 2.30. Elementos de la clase TCPClient	76
Figura 2.31. Elementos de la clase TempMeter	77
Figura 3.1. Prototipo desarrollado	79
Figura 3.2. Subsistema electrónico	80
Figura 3.3. Software de operación	80
Figura 3.4. Componente mecánica	81
Figura 3.5. Arreglo experimental para evaluar el desempeño en posición	83
Figura C.1. Dibujo esquemático	97
Figura C.2. Circuito impreso (mascarilla)	98
Figura C.3. Circuito impreso (componentes)	99
Figura C.4. Circuito impreso (soldadura)	100
Figura D.1. Base principal	103
Figura D.2. Base superior de la base principal	104
Figura D.3. Postes inferiores	105
Figura D.4. Base pvc inferior	106
Figura D.5. Base para rodamiento cónico	107
Figura D.6. Base superior de aluminio	108
Figura D.7. Base superior de PVC para el rodamiento radial	109
Figura D.8. Flecha principal de pvc	110
Figura D.9. Buje para el motor de azimut	111
Figura D.10. Brida para la base de la plataforma	112
Figura D.11. Plataforma de movimiento	113

---

Figura D.12. Base orgánica	114
Figura D.13. Brida de soporte	115
Figura D.14. Eje de movimiento de elevación	116
Figura D.15. Barra real de soporte y Barras en L	117
Figura D.17. Base para los reductores	118
Figura D.18. Cilindro interior	119
Figura D.19. Contrapeso chico	120
Figura D.20. Contrapeso grande	121
Figura D.21. Tapadera inferior	122



# Resumen de la tesis

El seguimiento astronómico es utilizado ampliamente en las áreas de astrofotografía y energía solar. El propósito del seguimiento es el de orientar dispositivos colectores hacia el objeto celeste en cuestión con el fin de concentrar la energía que emite. Los seguidores activos utilizan motores eléctricos y sistemas de transmisión de movimiento para orientar el dispositivo colector desde una posición fija en la tierra. Generalmente el seguimiento es gobernado por un sistema electrónico y algoritmos de control de movimiento.

En esta tesis se presenta el desarrollo y construcción de un sistema electromecánico para el seguimiento solar activo con dos ejes de libertad. El circuito electrónico está sobre la base de un microcontrolador, MCU, PIC18F4550, circuito electrónico denominado Reloj de Tiempo Real, RTC, e interface inalámbrica WiFi. El algoritmo de control en el MCU es calculado a partir de la fecha y hora en el calendario del RTC y las ecuaciones solares para mover los dos motores a pasos del mecanismo de seguimiento. La interface WiFi se utiliza con fines de monitoreo y configuración.

La sección de introducción plantea el problema enfrentado en este trabajo de tesis, el método empleado para resolverlo y expone los resultados esperados. El capítulo uno presenta los conceptos básicos que se utilizaron en el desarrollo del proyecto de tesis. Se expone la importancia de las energías renovables, las ecuaciones que rigen la trayectoria solar y las herramientas y dispositivos que se utilizaron para desarrollar el dispositivo de seguimiento solar. 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: sistema electrónico con base en MCU, RTC e interface WiFi, dispositivo mecánico

para el seguimiento en azimut y elevación 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 proyecto de tesis.

# 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 por medio de 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.

## Antecedentes

El principal antecedente para el desarrollo del presente proyecto es el artículo “A microcontroller-based multi-function solar tracking system” en donde se implementa un seguidor solar activo sobre la base de un microcontrolador [Kassem y Hamad, 2011]. 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

En el presente proyecto de tesis se propone desarrollar un sistema electrónico para un seguidor solar activo de dos grados de libertad que resuelva las ambigüedades que pudieran existir al generar la trayectoria solar utilizando foto-resistores. El concepto de este trabajo es la generación de la trayectoria solar mediante las ecuaciones solares y una base de tiempo que 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.

En el presente proyecto, el sistema electrónico 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 a desarrollar es seguir el sol, sin importar la fecha ni las condiciones del clima.

El sistema electrónico 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 temperatura.

### **Relevancia y justificación**

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 utiliza una base de tiempo y las ecuaciones solares. En forma adicional el sistema electrónico incluirá 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.

### **Alcance y limitaciones**

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

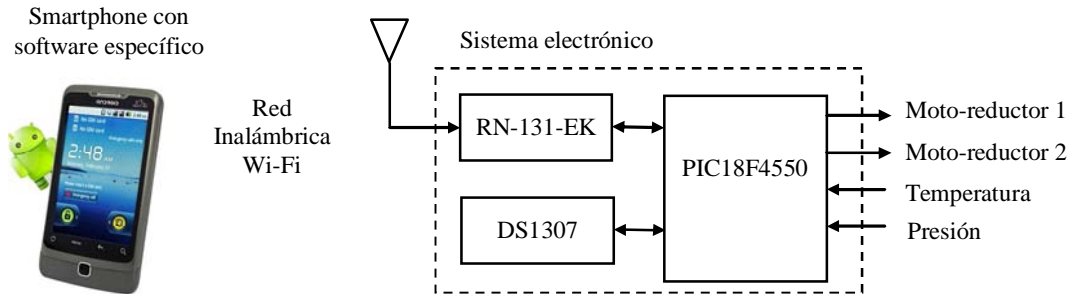
La principal limitación del prototipo por desarrollar radica en que no constituye una fuente para la generación de electricidad a la red pública. En este sentido, se espera que el panel fotovoltaico proporcione energía exclusivamente para abastecer a su propio sistema electrónico.

### **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 trabajo se describe el desarrollo de 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.

El sistema electrónico tiene como base un microcontrolador PIC18F4550 de 8 bits, con las siguientes características relevantes para la presente aplicación: Bajo consumo de potencia, interface I<sup>2</sup>C 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 I<sup>2</sup>C. El DS1307 cuenta segundos, minutos, horas, día del mes, día de la semana y año en tiempo real y consume menos de 500nA sin pérdidas 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.

## Resultados esperados

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

- Un circuito electrónico con base en microcontrolador, reloj de tiempo real y tarjeta de interfaz Wi-Fi.

- Firmware en el microcontrolador para el control del sistema mecánico, orientación automática a la trayectoria solar y comunicación Wi-Fi.
- Un prototipo mecánico con dos ejes de libertad que le permiten orientar un panel solar.
- Software en un Smartphone para el monitoreo y configuración del circuito electrónico.

# Capítulo 1

## Conceptos básicos

En el presente capítulo se revisan los conceptos básicos y herramientas utilizadas para la realización del presente proyecto de tesis.

### 1.1. Energías renovables

Las energías renovables son fuentes de energía natural y limpia que contribuyen a la conservación del medio ambiente. Existen una gran cantidad de energías renovables pero las principales son las del tipo solar y eólica que se describen a continuación.

#### 1.1.1. Energía solar

La energía solar es una fuente de energía emitida por el sol. El sol cada año arroja a nuestro planeta 4 mil veces más energía de la que el ser humano consume.

Los efectos que se perciben del sol cada día son el calor y la luz visible. La tierra y otras superficies expuestas al sol se calientan. Este calor puede aprovecharse para uso doméstico, comercial, o para procesos industriales. Los dispositivos que facilitan este aprovechamiento se llaman calentadores solares. En el proceso de calentamiento participan las regiones ultravioleta, visible e infrarrojo de la radiación solar.

Para la generación de energía eléctrica mediante celdas solares sólo participa una parte de la radiación solar; el resto de la energía solar se disipa como calor o sólo queda inútil para los materiales que constituyen la celda. A diferencia de la conversión solar foto-térmica, en la conversión solar fotovoltaica se considera la radiación como una fuente de fotones (cuanto de energía de la radiación) distribuida respecto a su energía.

En los últimos años la energía solar ha tomado una mayor importancia en las líneas de investigación de empresas públicas y privadas a nivel internacional. El simposio SolarPACES ha sido el evento científico más importante de concentración solar a nivel internacional durante casi 25 años. Una cincuentena de científicos e ingenieros, se han reunido en este evento cada dos años para intercambiar su experiencia técnica y buscar impulsar el desarrollo de dichas tecnologías. Sin embargo, en el año de 2009 el evento fue diferente; las inscripciones se cerraron antes de lo previsto, debido a un número inesperadamente grande de participantes inscritos. El grupo de científicos e ingenieros de siempre es ahora ampliamente superado por más de 300 representantes de empresas privadas o públicas, vendedores y consultores, todos en busca de “oportunidades de negocio”. Ver la figura 1.1.



Figura 1.1. Energía solar fotovoltaica.

México se encuentra entre los cinco países más atractivos del mundo para invertir en proyectos de energía solar fotovoltaica, tan solo detrás de China y Singapur. Lo anterior debido a que el país forma parte del “cinturón solar” con una radiación mayor a 5KWh por metro cuadrado al día. México cuenta con la base manufacturera de módulos fotovoltaicos más grande de América Latina.

Hasta 2012, México contaba con una capacidad efectiva instalada para la generación de energía eléctrica de 63,195 MW, de los cuáles 14,501 MW provinieron de fuentes renovables de energía (eólica, solar, hidráulica, geotérmica y de biomasa), lo que representa el 23% del total de la capacidad instalada, de acuerdo con estimaciones de ProMéxico con datos de la Comisión Reguladora de Energía (CRE) y la Comisión Federal de Electricidad

(CFE). Donde la energía solar aportaba unos 37 MW a la capacidad efectiva instalada. Ver tabla 1.1.

TIPO DE ENERGÍA	CAPACIDAD INSTALADA EN OPERACIÓN (MW)	CAPACIDAD AUTORIZADA EN CONSTRUCCIÓN (MW)
Eólica	1,289	2,460
Geotérmica	823	0
Hidráulica	11,707	289
Solar	37*	156
Biomasa	645	100
<b>Total</b>	<b>14,501</b>	<b>3,006</b>

Fuente: Comisión Federal de Electricidad (CFE) / Comisión Reguladora de Energía (CRE).

\*Incluye proyectos fotovoltaicos de pequeña y mediana escala, principalmente en aplicaciones de electrificación rural y residencial.

Tabla 1.1. Capacidad instalada en operación y en construcción para la generación de ER.

Los principales problemas que se tienen con este tipo de energía son diversos, ya que se requiere de una inversión considerablemente grande en su instalación además de que se requieren extensos territorios para poder captar la mayor energía solar posible.

### 1.1.2. Energía eólica

La energía eólica es la energía obtenida a partir del viento, es decir, la energía cinética generada por efecto de las corrientes de aire, y que es convertida por medio de generadores eólicos en otras formas útiles de energía para el desarrollo de actividades humanas.

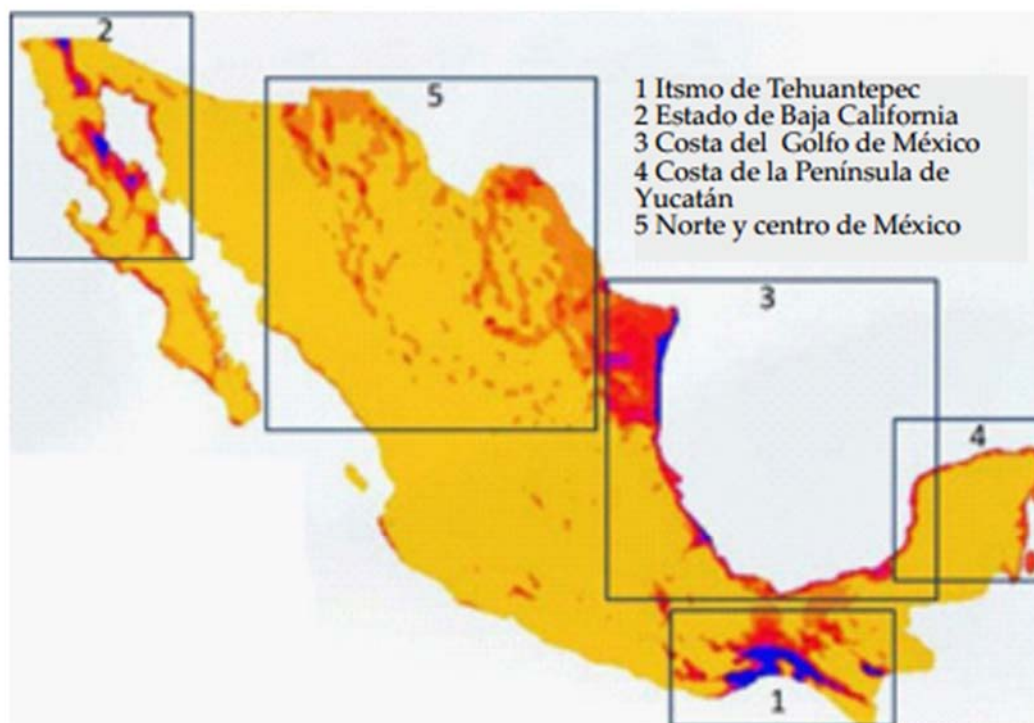
La energía eólica constituye una de las fuentes de energía renovable con mayor potencial de desarrollo, tanto por su técnica avanzada como por su viabilidad económica. Asimismo, es una de las fuentes de energía más interesantes para reducir las emisiones de CO<sub>2</sub>.

Actualmente se pueden encontrar en el mercado máquinas eólicas en una amplia gama de potencia, con aplicaciones tan diversas como el accionamiento de bombas de agua, para riego, la producción autónoma de electricidad o la generación de energía eléctrica a gran escala (parques eólicos). Los avances de la tecnología eólica han permitido disponer de

máquinas capaces de trabajar a buen rendimiento en lugares con distribución de velocidades bastante comunes.

En México, dentro de las energías renovables, la energía eólica ha jugado un papel muy importante en cuanto a la capacidad instalada y desarrollo se refiere. México cuenta con 26 parques eólicos a lo largo y ancho de su territorio. Esto lo coloca en el primer puesto en generación de energía en cuanto a energías renovables se refiere.

México cuenta con 40,268 MW de potencial de energía eólica, lo que representa más de la capacidad instalada de todo sistema eléctrico nacional. Sin embargo, actualmente solo se aprovecha el 3.2% de dicha capacidad. En la figura 1.2 se muestra las zonas potenciales con las que cuenta México para la producción de energía eólica.



Fuente: AMDEE

Figura 1.2. Zonas potenciales para el desarrollo de proyectos de energía eólica, Asociación Mexicana de Energía Eólica (AMDEE).

Generando más de las dos terceras partes en cuanto a toda la energía se produce. Las energías renovables y en especial la eólica han mostrado su gran eficiencia y producción.

El principal problema de la generación de energía eólica es la incertidumbre respecto a la disponibilidad de viento cuando se necesita. Lo que implica que la energía eólica no puede ser utilizada como fuente de energía única y debe de estar respaldada siempre por otras fuentes de energía con mayor capacidad de regulación (térmicas, hidroeléctricas, nucleares, etc.)

## 1.2. 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.

### 1.2.1. 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:

$$I_{sc} = 429.5 \frac{\text{Btu}}{\text{hr ft}^2} = 1353 \frac{\text{W}}{\text{m}^2} \quad \text{Es la constante solar}$$

$N$  es el número de días transcurridos

El punto más cercano de la tierra (alrededor de 146 millones km) al sol es llamado la *perihelion* y ocurre alrededor del 3 de enero; El punto más lejano de la tierra (alrededor de 156 millones km) al sol es llamado la *aphelion* y ocurre alrededor del 4 de julio. La figura 1.3 ilustra la trayectoria que genera la tierra al rotar alrededor del sol.



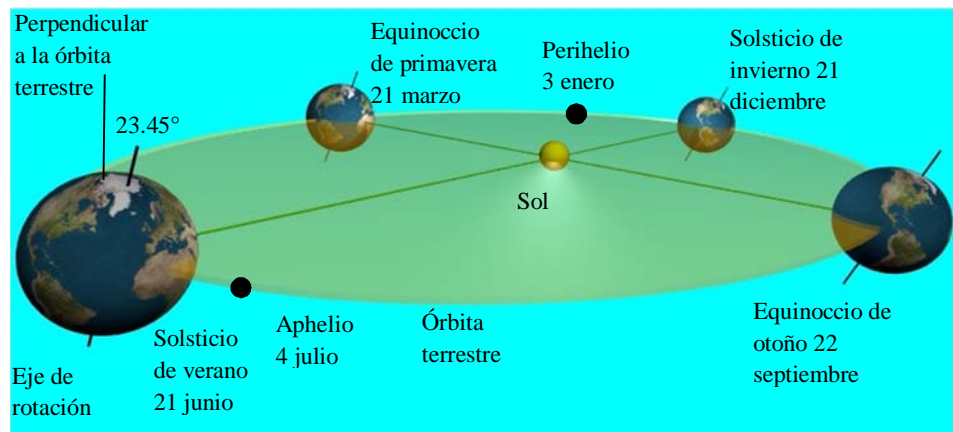


Figura 1.3. Trayectoria tierra-sol.

La tierra esta inclinada sobre su eje en un ángulo de  $23.45^\circ$  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,  $\delta$ , para el hemisferio norte (invertir el signo del ángulo de declinación para el hemisferio sur) es:

$$\delta = 23.45^\circ \text{sen} \left[ \frac{N + 284}{365} 360^\circ \right] \quad (1.1)$$

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  $0^\circ$ ; los polos norte y sur están a  $+90^\circ$  y  $-90^\circ$ , respectivamente; el trópico de cáncer y el trópico de capricornio están localizados a  $+23.45^\circ$  y  $-23.45^\circ$ , respectivamente.

Para longitudes, la comunidad global ha definido  $0^\circ$  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  $360^\circ$  significa que se tarda 4 minutos para mover cada grado de longitud. Ver figura 1.4.

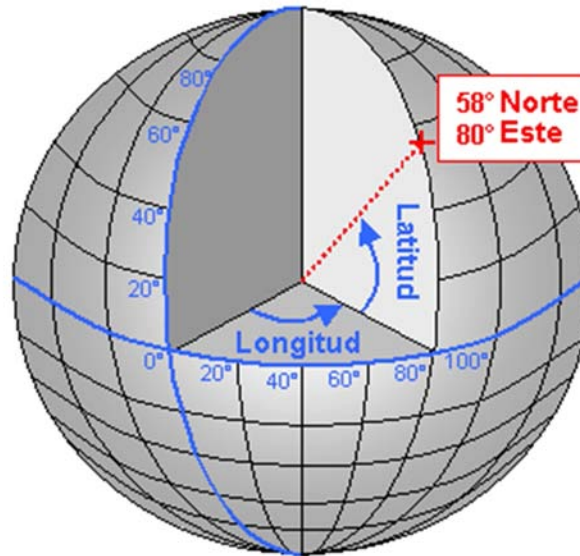


Figura 1.4. Longitud y latitud de la tierra.

### 1.2.2. 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 1.5.

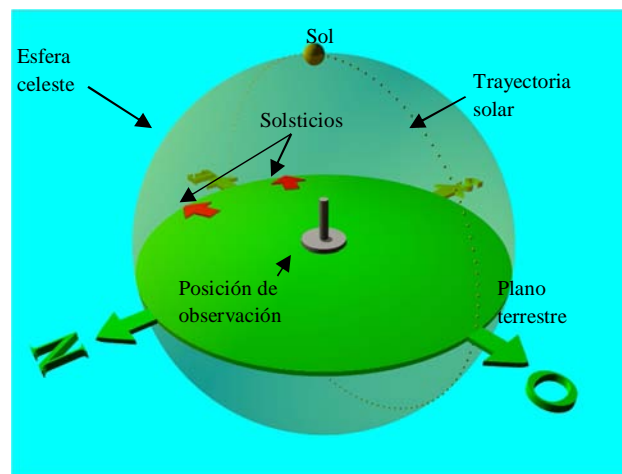


Figura 1.5. Modelo de la trayectoria aparente del sol.

Al modelo de la figura 1.5 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 1.5.

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.

### 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 1.6 muestra la relación descrita.

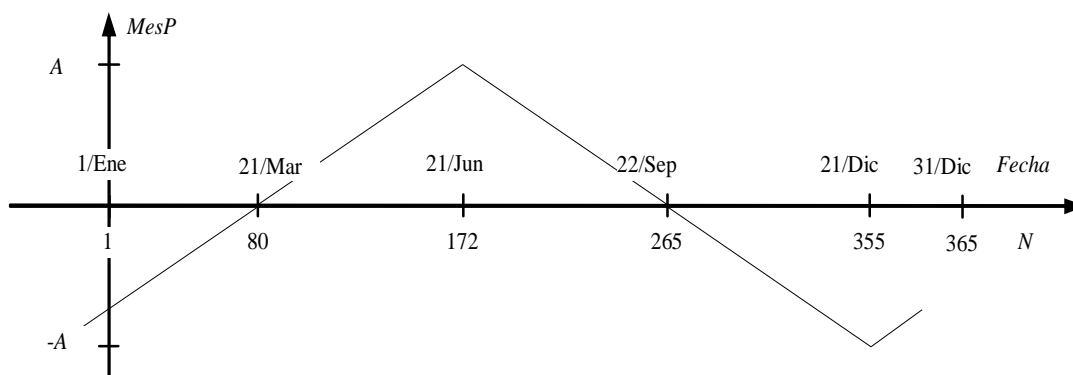


Figura 1.6. Relaciones *Fecha-N-MesP*.

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 1.6 se obtienen las siguientes relaciones:

$$\begin{aligned} MesP &= \frac{A(N - 172)}{92} + A \text{ para } 1 \leq N \leq 172 \\ MesP &= \frac{2A(N - 172)}{172 - 355} + A \text{ para } 173 \leq N \leq 355 \\ MesP &= \frac{(13.869 - A)(N - 355)}{(355 - 365)} - A \text{ para } 356 \leq N \leq 365 \end{aligned} \quad (1.2)$$

En resumen, la ecuación (1.2) se utiliza para trasladar geoméricamente 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  $23.45^\circ$ . Al proyectar el plano terrestre sobre los ejes coordenados se obtiene la representación de la figura 1.7.

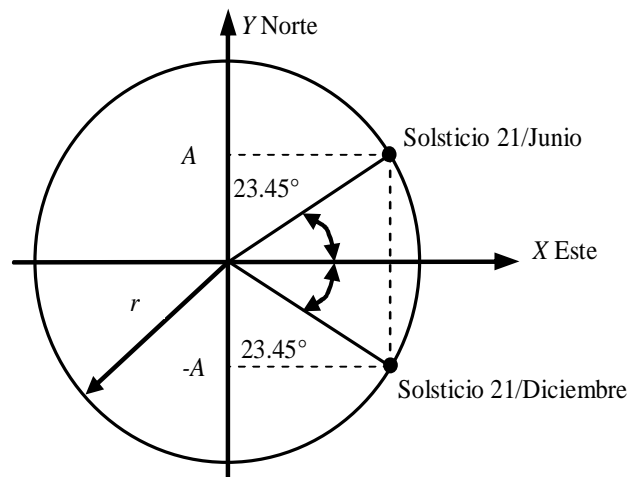


Figura 1.7. Solsticios.

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

$$A = r \text{ sen}(23.45^\circ) \quad (1.3)$$

### 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 1.8.

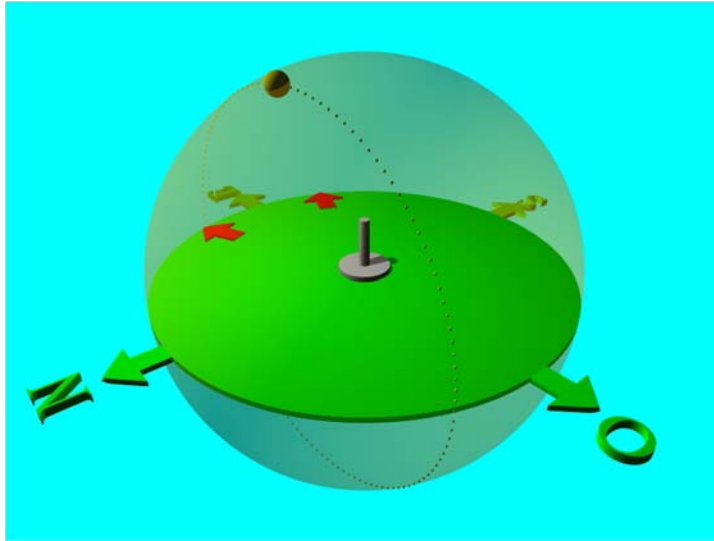


Figura 1.8. Traslación de la trayectoria aparente del sol.

Proyectando la trayectoria solar sobre los ejes coordenados se obtiene el arreglo que se muestra en la figura 1.9.

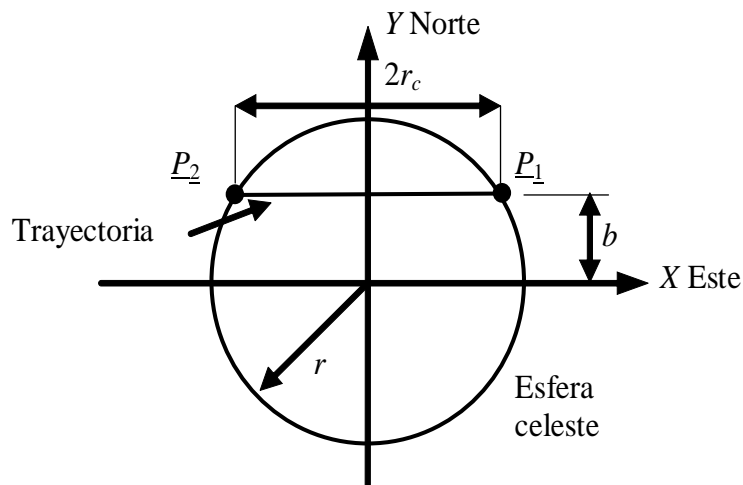


Figura 1.9. Radio de la trayectoria solar.

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  $\underline{P_1}$  y  $\underline{P_2}$

$$x = \sqrt{r^2 - b^2}$$

$$\underline{P_1} = [\sqrt{r^2 - b^2}, b]$$

$$\underline{P_2} = [-\sqrt{r^2 - b^2}, b]$$

El radio de la trayectoria  $r_c$  entonces es la mitad de la distancia entre  $\underline{P_1}$  y  $\underline{P_2}$

$$r_c = \frac{1}{2} \sqrt{\left(-\sqrt{r^2 - b^2} - \sqrt{r^2 - b^2}\right)^2 + (b - b)^2}$$

$$r_c = \sqrt{r^2 - b^2}$$

$$r_c = \sqrt{r^2 - MesP^2}$$

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

$$T_x = r_c \cos(index)$$

$$T_y = 0$$

$$T_z = r_c \sin(index)$$

Sustituyendo  $r_c$

$$T_x = \sqrt{r^2 - MesP^2} \cos(index)$$

$$T_y = 0 \tag{1.4}$$

$$T_z = \sqrt{r^2 - MesP^2} \sin(index)$$

Con  $0 \leq index \leq 2\pi$ , como se muestra en la figura 1.10.

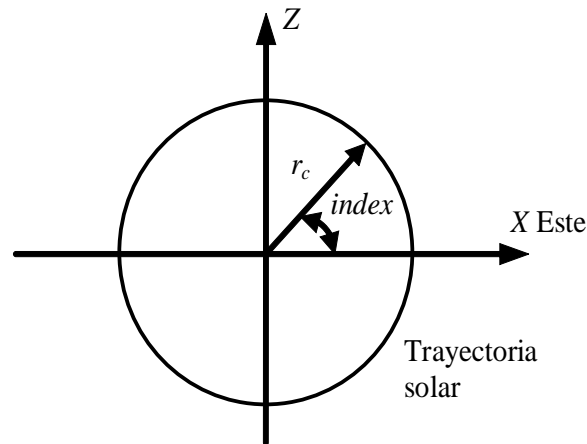


Figura 1.10. Trayectoria solar en el plano.

La ecuación (1.4) 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  $MesP$ .

### 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 1.11.

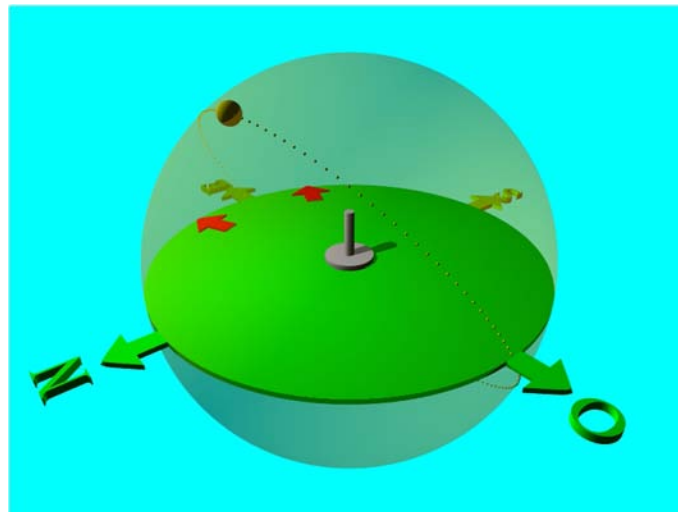


Figura 1.11. Rotación de la trayectoria aparente del sol.

Proyectando la trayectoria solar sobre los ejes coordenados se obtiene el arreglo que se muestra en la figura 1.12.

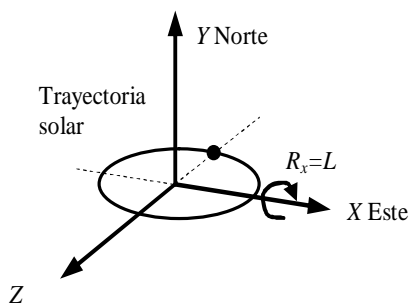


Figura 1.12. Trayectoria aparente del sol en los ejes coordenados.

La figura 1.12 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  $L$ .

### 1.2.3. Ajuste del tiempo

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

$$AST = LST + \left(4 \frac{\text{min}}{\text{deg}}\right) (LSTM - Long) + ET \quad (1.5)$$

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:

$$ET = 9.87 \text{sen}(2D) - 7.53 \text{cos}(D) - 1.5 \text{sen}(D) \quad (1.6)$$

Donde:



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

### 1.2.4. 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:

$$H = \frac{(\text{Número de minutos después de la medianoche, } AST) - 720 \text{ minutos}}{4 \text{ min/deg}} \quad (1.7)$$

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  $\beta_1$  es la altura angular aparente del sol en el cielo si se mira de frente. El ángulo cénit  $\theta_z$  y su complemento, el ángulo de altitud  $\beta_1$ , están dados por

$$\cos(\theta_z) = \sin(\beta_1) = \cos(L) \cos(\delta) \cos(H) + \sin(L) \sin(\delta) \quad (1.8)$$

Donde:

$L$ = latitud (positivo en cualquier hemisferio) ( $0^\circ$  a  $90^\circ$ ).

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

$H$ =ángulo horario.

La altitud al mediodía es  $\beta=90^\circ-L+\delta$ . El sol sale y se pone cuando su altitud es  $0^\circ$ , 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 (1.8), donde  $\beta_1=0$

$$\cos(H_s) = -\tan(L) \tan(\delta) \quad (1.9)$$

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,  $\alpha_1$ , es el ángulo lejos del sur.

$$\cos(\alpha_1) = \frac{\sin(\beta_1) \sin(L) - \sin(\delta)}{\cos(\beta_1) \cos(L)} \quad (1.10)$$

Donde  $\alpha_1$  es positiva hacia el oeste (tarde), y negativa hacia el este (mañana), y por lo tanto, el signo de  $\alpha_1$  debe coincidir con la del ángulo

horario. La figura 1.13 ilustra los ángulos de altitud  $\beta_1$ , azimuth  $\alpha_1$  y horario  $H$ .

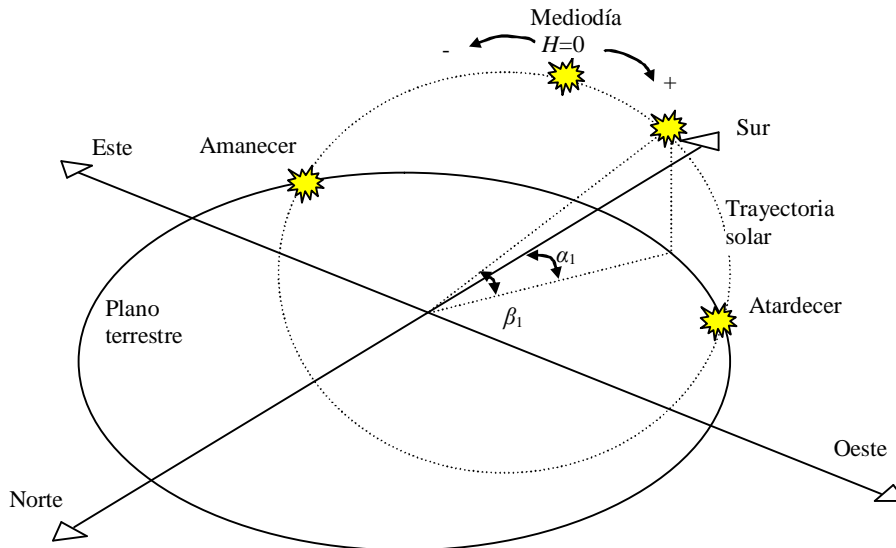


Figura 1.13. Ángulos de interés.

### 1.2.5. Ejemplo de aplicación

A manera de ejemplo, se muestra el cálculo de los ángulos solares de la figura 1.13 para la localidad de Phoenix, Arizona, USA que está localizada a  $112^\circ$  longitud oeste y  $33.43^\circ$  de latitud a las 8:00 a.m. para el 21 de julio.

Dado que Phoenix no observa horario de verano, resulta innecesario hacer el ajuste al tiempo local. De la figura 1.6 se obtiene que para el 21 de julio  $N=202$ . Sustituyendo en la ecuación (1.6)

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

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

Sustituyendo  $Long=112^\circ$  y  $LST=8:00$  en la ecuación (1.5)

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

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

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

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

Sustituyendo  $AST=7:25.95=7\text{hrs } 25.95\text{min}$  en la ecuación (1.7)

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

Sustituyendo  $N=202$  en la ecuación (1.1)

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

Sustituyendo  $L=33.43^\circ$ ,  $H=-68.5^\circ$  y  $\delta=20.44^\circ$  en la ecuación (1.8)

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

$$\beta_1 = 28.61^\circ$$

Sustituyendo  $\beta_1=28.61^\circ$ ,  $L=33.43^\circ$  y  $\delta=20.44^\circ$  en la ecuación (1.10)

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

Los ángulos obtenidos,  $\alpha_1$ ,  $\beta_1$  y  $H$  son suficientes para reproducir de manera ilustrativa la posición del sol como se muestra en la figura 1.13. 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 (1.3) 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 (1.2)

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

Sustituyendo  $MesP=10.7$  y  $r=40$  en la ecuación (1.4) se obtiene la magnitud de las traslaciones en  $x$  y  $y$  de la representación gráfica

$$T_x = \sqrt{40^2 - 10.7^2} \cos(index)$$

$$T_y = 0$$

$$T_z = \sqrt{40^2 - 10.7^2} \sin(index)$$

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

De la figura 1.12 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 (1.9).

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

Para encontrar su correspondiente horario AST se sustituye  $H_s=-104.24^\circ$  en la ecuación 1.7

$$-104.24^\circ = \frac{(Amanecer_{AST}) - 720 \text{ minutos}}{4 \text{ min/deg}}$$

$$Amanecer_{AST} = \left(-104.24^\circ \times 4 \frac{\text{min}}{\text{deg}}\right) + 720 \text{min}$$

$$Amanecer_{AST} = 303.04 \text{min} = 5.051 \text{hrs} = 5:03$$

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 (1.5)

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

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

$$Amanecer_{LST} = 5:03 - (-28 \text{min}) + 6.05 \text{min} = 5:37$$

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 (1.9).

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

Para encontrar su correspondiente horario AST se sustituye  $H_s=104.24^\circ$  en la ecuación 1.7

$$104.24^\circ = \frac{(Atardecer_{AST}) - 720 \text{ minutos}}{4 \text{ min/deg}}$$

$$Atardecer_{AST} = \left(104.24^\circ \times 4 \frac{\text{min}}{\text{deg}}\right) + 720 \text{min}$$

$$Atardecer_{AST} = 1136.96 \text{min} = 18.95 \text{hrs} = 18:57$$

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 (1.5)

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

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

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

La tabla 1.2 muestra el ejemplo desarrollado y contiene los resultados relevantes para el presente proyecto de tesis. El anexo A contiene el programa en Matlab que obtiene los resultados de la tabla 1.2.

Información de entrada		Datos de salida	
Longitud $Long$	112°	Azimut $\alpha_1$	-96.7°
		Altitud $\beta_1$	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 1.2. Ejemplo de aplicación (resultados numéricos).

Con los resultados de la tabla 1.2 se genera la figura 1.14 que muestra de manera gráfica el ejemplo de aplicación desarrollado. El anexo B contiene el código fuente para reproducir la figura 1.14 a partir de geometrías básicas [Pov-Ray, 2013]. En la figura 1.14 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.

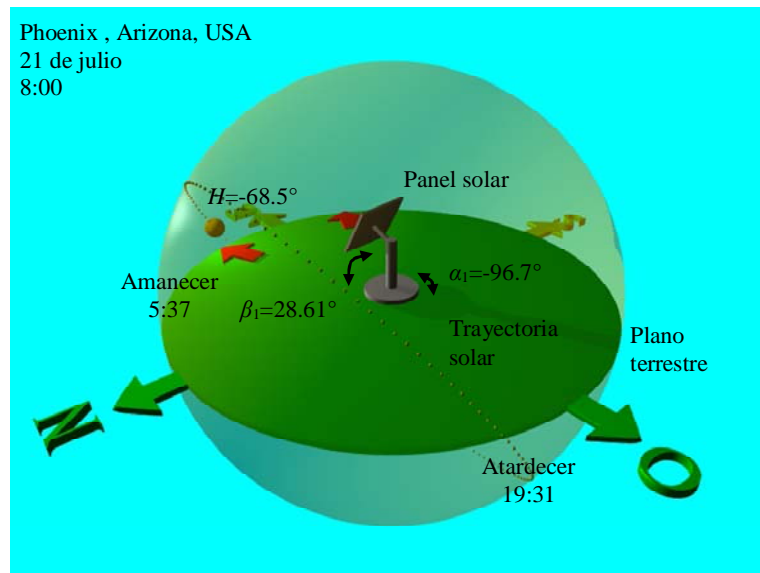


Figura 1.14. Ejemplo de aplicación (ambiente gráfico).

### 1.3. Microcontrolador PIC18F4550

Los microcontroladores PIC existen en gamas de 8 bits, 16 bits y 32 bits. Dentro de la gama más simple de 8 bits se encuentra el microcontrolador PIC18F4550.

#### 1.3.1. Especificaciones generales

Sus características de memoria RAM, memoria EEPROM, tipos de puertos de comunicación, número de entradas/salidas, número de canales analógicos y tipos de puertos de comunicación, han hecho de este PIC uno de los más utilizados para diversas aplicaciones.

A continuación en la tabla 1.3 se puede apreciar algunas características técnicas del PIC18F4550.

CARACTERISTICAS	PIC18F4450
Frecuencia de Operación	DC - 48MHZ
Memoria de programa(bytes)	32768
Memoria RAM (bytes)	2048
Memoria EEPROM (bytes)	256
Puertos E/S	Puertos A,B,C , D, E
Interrupciones	20
Líneas de E/S	35

Temporizadores	4
Módulos de comparación/PWM	1
Canales de comunicación serie	MSSP, EUSART
Canal USB	1
Puerto Paralelo de transmisión de datos (SSP)	1
Canales de conversión A/D de 10 bits	13 canales
Comparadores analógicos	2
Encapsulados	PDIP 40 pines QFN 40 pines TQFP 40 pines

Tabla 1.3. Características generales del PIC18F4550

En la tabla 1.3 se presentan las características principales del PIC18F4550. Este microcontrolador cuenta con 5 puertos E/S, 4 temporizadores, 20 fuentes de interrupción, comunicación serial, módulo USB, 13 canales de entradas analógicas y 2 comparadores analógicos. Con estas características el microcontrolador puede ser usado en toda una gama de posibles aplicaciones, manejando diferentes tipos de variables analógicas y digitales.

En la figura 1.15 se presenta la descripción de pines del microcontrolador PIC18F4550.

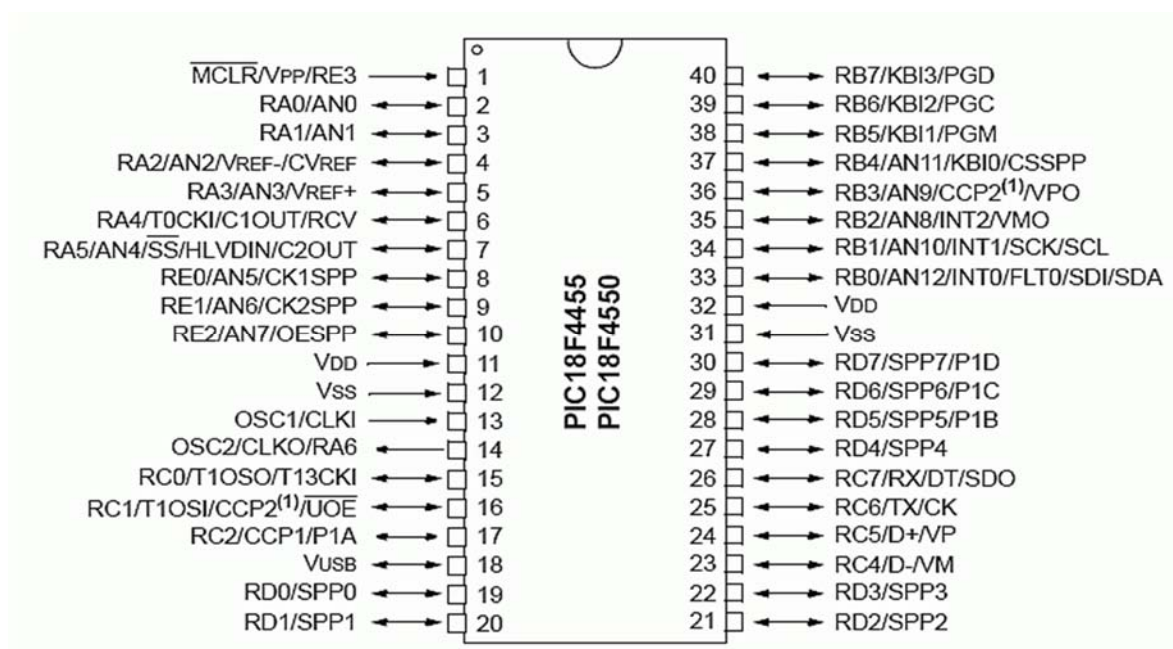


Figura 1.15. Esquema de pines del PIC18F4550.

El microcontrolador PIC18F4550 se alimenta con 5V DC, los cuales se aplican en los pines de alimentación VDD (+) y VSS (-). El consumo de corriente del dispositivo depende de las cargas conectadas al microcontrolador y de su frecuencia de trabajo. Se aconseja que se coloque un capacitor de desacoplo de 100nF lo más cerca posible de los pines de alimentación. Una fuente DC de 5V y por lo menos 700mA será suficiente para el trabajo eficiente de este microcontrolador.

El microcontrolador requiere de un circuito que indique la velocidad de trabajo al cual este operara. Para esto se utiliza un circuito externo que puede ser un cristal o un reloj. En la hoja de especificaciones de este microcontrolador se recomienda uno de 20MHz.

### **1.3.2. Herramientas de desarrollo**

Para desarrollar aplicaciones utilizando los microcontroladores PIC se requieren diversas herramientas, algunas de las cuales se describen a continuación.

#### **MPLAB**

MPLAB es un editor IDE (siglas en inglés de Integrated Development Environment) gratuito, destinado a productos de la marca Microchip. Este editor es modular, permite seleccionar los distintos microcontroladores soportados, además de permitir la grabación de estos circuitos integrados directamente al programador.

Es un programa que corre bajo Windows, Mac OS y Linux. Presenta las clásicas barras de programa, de menú, de herramientas de estado, etc. El ambiente MPLAB posee editor de texto, compilador y simulador (no en tiempo real). Para comenzar un programa desde cero para luego grabarlo al microcontrolador en MPLAB se siguen los siguientes pasos:

1. Crear un nuevo archivo con extensión .ASM y nombre cualquiera.
2. Crear un Proyecto nuevo eligiendo un nombre y ubicación.
3. Agregar el archivo .ASM como un SOURCE FILE.
4. Elegir el microcontrolador a utilizar desde SELECT DEVICE del menú CONFIGURE.



Una vez realizado esto, se está en condiciones de empezar a escribir el programa respetando las directivas necesarias y la sintaxis para luego compilarlo, grabarlo y ejecutarlo directamente en un microcontrolador PIC.

## **Lenguaje de programación C**

Uno de los lenguajes de programación más antiguos, llamado lenguaje C, fue creado en 1972 por Dennis Ritchie.

Entre las características principales del lenguaje de programación que debemos conocer a la hora de crear un programa son las siguientes:

1. En un principio fue un lenguaje ligado a UNIX, pero después se implementó en otros ambientes como Microsoft y MS-DOS, además, adquirió el estándar ANSI y luego el ISO.
2. No es un lenguaje complejo como el Basic, el Cobol o el Pascal, pero es superior a un lenguaje ensamblador.
3. El lenguaje de programación sirve para crear aplicaciones y software de sistemas.
4. Es un lenguaje muy flexible que permite programar con múltiples estilos.
5. Acceso a memoria de bajo nivel mediante el uso de punteros.
6. Interrupciones al procesador con uniones.
7. Punteros a funciones y variables estáticas, que permiten una forma rudimentaria de encapsulado y polimorfismo.

## **CCS Compiler**

La empresa CCS (Custom Computer Services) desarrolló el primer compilador C para microcontroladores Microchip hace más de 20 años y continúa ofreciendo soluciones de software para desarrolladores de aplicaciones que utilizan microcontroladores PIC y dispositivos DSC PIC24/dsPIC. Los compiladores CCS son fáciles de usar y de aprender.

Incluye la optimización a nivel profesional, la mayor biblioteca de funciones integradas, y programas de ejemplo listos para funcionar de forma rápida y poner en marcha cualquier proyecto. Ver figura 1.16.

Características claves del compilador:

1. Fácil migración entre todos los dispositivos de Microchip PIC® MCUs.

2. Reduce al mínimo el tiempo de desarrollo.
3. C++ de entrada/salida con datos completos en formato a cualquier dispositivo o para datos del tipo string.
4. Utiliza bibliotecas de CCS y código libre.
5. Diversas funciones como #bit y #byte que le permiten colocar variables de C en direcciones absolutas.
6. Definir, configurar y gestionar las interrupciones.

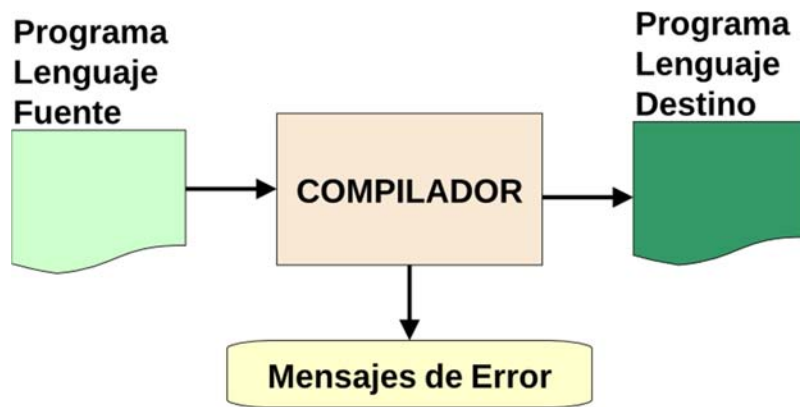


Figura 1.16. Compilador CCS.

## 1.4 Android

El sistema Operativo Android es una plataforma de software con base en el núcleo de Linux. Diseñada en un principio para dispositivos móviles, que permiten controlar dispositivos por medio de bibliotecas desarrolladas o adaptadas por Google mediante el lenguaje de programación Java.

Android es una plataforma de código libre. Esto quiere decir, que cualquier programador puede crear y desarrollar aplicaciones para esta plataforma, así como cualquier empresa puede implementar el sistema operativo en los dispositivos móviles que produce.

Las principales características que presenta el sistema operativo Android son:

1. Diseño de dispositivo: La plataforma es adaptable a pantallas de diversas resoluciones, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D con base en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales.

2. Almacenamiento: Android permite el almacenamiento de los datos en archivos de diversos formatos, soporta el manejo de memoria interna y externa, y como herramienta de bases de datos está SQLite, una base de datos liviana.
3. Conectividad: El sistema operativo soporta las siguientes tecnologías de conectividad: GSM/ADGE, IDEN, CDMA, EVDO, UMTS, Bluetooth, WiFi, LTE, HSDPA, HSPA+ y WiMAX.
4. Mensajería: SMS y MMS son formas de mensajería, incluyendo mensajería de texto y ahora la Android Cloud to Device Messaging Framework (C2DM) es parte del servicio de Push Messaging de Android.
5. Navegador Web: El navegador web incluido en Android está sobre la base del motor de renderizado de código abierto WebKit, emparejado con el motor de JavaScript V8 de Google Chrome.
6. Soporte Multimedia: Android soporta los siguientes formatos multimedia; WebM, 3GP, MP4, MPEG-4 SP, AMR, AMR-WB, AAC, HE- AAC, MP3, MIDI, Ogg, Vorbis, WAV, JPEG, PNG, GIF y BMP.
7. Soporte para streaming: Streamning RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML. Adobe Flash Streaming (RTMP) es soportado mediante Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player.
8. Soporte para hardware adicional: Android soporta cámaras de fotos, video, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.
9. Multi-táctil: Android tiene soporte nativo para pantallas capacitivas con soporte multi-táctil que inicialmente hicieron su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de kernel. Más tarde, Google publicó una actualización para el Nexus One y el Motorola Droid que activa el soporte multi-táctil de forma nativa.
10. Multitarea: Multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj, a diferencia de otros sistemas de la competencia en la que la multitarea es congelada.

11. Tethering: Android soporta tethering, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico.
12. Características con base en voz: La búsqueda en Google a través de voz está disponible como “Entrada de Búsqueda” desde la versión inicial del sistema.
13. Video llamada: Android soporta video llamada a través de Google Talk desde su versión HoneyComb.

El entorno de desarrollo incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento de software. El entorno de desarrollo integrado es Eclipse usando un plugin de Herramientas de Desarrollo de Android, ver figura 1.17.

Google Play es un catálogo de aplicaciones gratuitas o de pago que pueden ser descargadas e instaladas en dispositivos Android sin necesidad de una PC.



Figura 1.17. Programación para Android.

Soporta Java aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma. El bytecode Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y corre en la Máquina Virtual Dalvik. Dalvik es una máquina virtual especializada diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados.

## 1.5. Dispositivos electrónicos empleados

Para el desarrollo del presente proyecto de tesis se construyó un sistema electrónico con base en el microcontrolador de la sección 1.3 y los dispositivos electrónicos que se describen a continuación.

### 1.5.1. Módulo LAN inalámbrico RN-131C

El módulo de comunicaciones inalámbricas RN-131 de Roving Networks, figura 1.18, está habilitado para trabajar en modo local con muy bajo consumo energético dando soporte para protocolo inalámbrico WiFi con velocidades de transferencia de hasta 54 Mbps, mediante la cual es posible adquirir o medir hasta 5 variables análogas con resolución de 14 bits e incluye un sensor de temperatura y humedad relativa, medición de dos señales digitales y actuación por medio de dos salidas digitales, todas ellas acondicionadas en 0-5V e incluye puerto USB para comunicación y/o programación.

Su forma compacta y muy bajo consumo de energía lo hace ideal para aplicaciones inalámbricas móviles tales como el seguimiento de activos, seguimiento de GPS y sensores de baterías.



Figura 1.18. Módulo WiFi.

Sus principales características son las siguientes:

- Ultra-bajo consumo de energía: 4  $\mu$ A inactivo, 40 mA recibiendo datos y 210 mA transmitiendo datos.
- Alto rendimiento, 1 Mbps de velocidad de datos sostenidos con TCP/IP y WPA2.
- 8 Mbits de memoria flash y 128 KB de RAM.
- 10 pines de propósito general E/S digitales.
- Reloj en tiempo real para activar o desactivar.
- Acepta 3.3 V regulado o de 2 a 3 V de la batería.
- Wi-Fi para WPA2-PSK.

- Industrial (RN-131G) y comercial (RN-131C) son las opciones de temperatura.

La sencillez del módulo inalámbrico y su bajo consumo de energía en la recepción, transmisión y en su estado inactivo hicieron que se optara por este dispositivo para la comunicación entre el sistema electrónico desarrollado y una aplicación para teléfono móvil.

### 1.5.2. Driver M54532P/FP

El circuito M54532P/FP está hecho de 4 arreglos de transistores NPN en su interior, en una configuración Darlington con diodos de acoplamiento en cada uno, lo cual le permite manejar altas corrientes con muy bajas corrientes de entrada. Ver figura 1.19.

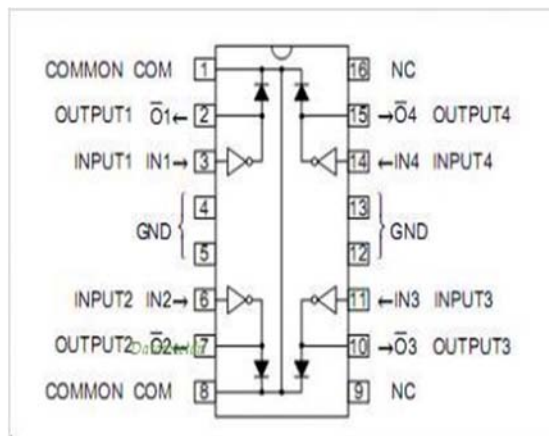


Figura 1.19. Driver de corriente directa.

Sus principales características son las siguientes:

- Alto voltaje de ruptura ( $V_{CEO} \geq 50V$ )
- Conducción de altas corrientes ( $I_C(\max)=1.5A$ )
- Con diodos de sujeción para cargas inductivas.
- Ancho rango de operatividad ( $T_a = -20$  a  $+75^\circ C$ ).

### 1.5.3. Convertidor de nivel lógico bidireccional BOB-08745

El convertidor de nivel lógico bidireccional BOB-08745 es un dispositivo que de forma segura convierte y transmite señales de 5V a 3.3V o de 3.3V a 5V al mismo tiempo. Ver figura 1.20.

El convertidor de nivel es muy fácil de usar. Tiene que ser alimentado de las dos fuentes de voltaje (alta tensión y baja tensión) que su sistema está

utilizando. Alto voltaje (5V por ejemplo) para el pin 'HV', baja tensión (3.3 V por ejemplo) para 'LV', y de tierra del sistema para el pin 'GND'.



Figura 1.20. Convertidor de nivel lógico.

#### 1.5.4. Reloj de tiempo real DS1307

El reloj de tiempo real de la figura 1.21 (por sus siglas en inglés RTC, Real Time Clock) DS1307 es un dispositivo de bajo consumo de energía. La dirección de datos se transfiere en serie a través de I<sup>2</sup>C. El reloj y calendario proporciona segundos, minutos, horas, días, mes y año. El reloj funciona tanto en el formato de 24 horas o de 12 horas con indicador AM/PM. El DS1307 tiene un circuito integrado que detecta fallas de energía y cambia automáticamente a la alimentación de reserva. Consume menos de 500nA sin pérdidas. 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.

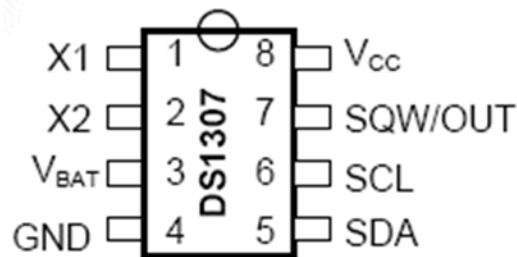


Figura 1.21. Reloj de tiempo real.

Sus principales características son las siguientes:

- Reloj de tiempo real (RTC). Cuenta segundos, minutos, horas, fecha del mes, día de la semana y año válido hasta 2100.

- Interface Serial I<sup>2</sup>C.
- Consume menos de 500 nA en modo Batería-Backup.
- Rango de operatividad industrial de temperaturas: -40 °C a +85 °C.

### 1.5.5. Sensor de temperatura LM35

El LM35 es un sensor de temperatura integrado de precisión, cuya tensión de salida es linealmente proporcional a temperatura en °C (grados centígrados), figura 1.22. El LM35 no requiere ninguna calibración externa o ajuste para proporcionar una precisión típica de  $\pm 1.4$  °C a temperatura ambiente y  $\pm 3.4$  °C a lo largo de su rango de temperatura (de -55 a 150 °C).

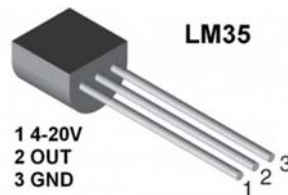


Figura 1.22. Sensor de temperatura.

Principales características:

- Calibrado directamente en grados Celsius (Centígrados).
- Factor de escala lineal de +10 mV/°C.
- 0.5°C de precisión a +25 °C.
- Rango de trabajo: -55 °C a +150 °C.
- Funciona con alimentaciones entre 4V y 30V.
- Menos de 60  $\mu$ A de consumo.
- Baja impedancia de salida, 0.1W para cargas de 1mA.

El LM35 tiene una ventaja sobre los sensores de temperatura lineal calibrada en grados Kelvin: que el usuario no está obligado a restar una gran tensión constante para obtener grados centígrados.

### 1.5.6. Optoacoplador H22A1

El interruptor óptico ranurado H22A1 de la figura 1.23 es un diodo emisor de luz de arseniuro de galio acoplado a un fotodarlington de silicio en una carcasa de plástico. El sistema de empaquetado está diseñado para optimizar la resolución mecánica del dispositivo. La brecha en el encapsulado proporciona un medio de interrupción de la señal con un material opaco, cambiando la salida de ON a OFF.





Figura 1.23. Sensor de posición.

### 1.5.7. Batería recargable

La batería recargable PS-1230, de la figura 1.24, suministra 12 V con 3.4 A/H. Sus principales características son:

- Tecnología de vidrio absorbente (AGM) para un rendimiento superior.
- Válvula regulada, permite una operación segura en cualquier posición.
- Cubierta resistente a impactos (UL94-HB).
- Resistencia interna de 45mΩ
- Preso aproximado, 1.32 Kg.



Figura 1.24. Batería recargable.

La tabla 1.4 muestra la forma en que la batería puede suministrar corriente cuando se opera continuamente.

Tiempo de operación continua	Corriente	Voltaje
20 hr	170 mA	10.50 V
10 hr	330 mA	10.50 V
5 hr	580 mA	10.20 V
1 hr	2.2 A	9.0 V
15 min	6.5 A	9.0 V

Tabla 1.4. Suministro de corriente.

### 1.5.8. Celda solar

La celda solar o celda fotovoltaica, de la figura 1.25, es el dispositivo que genera electricidad directamente de la luz visible. Sus principales características son:

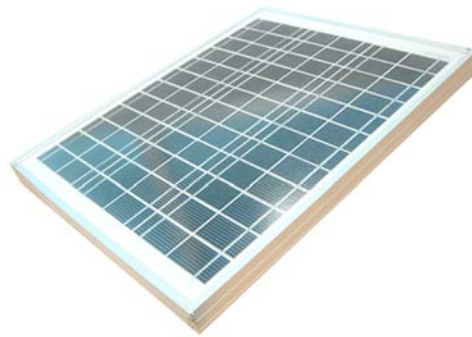


Figura 1.25. Celda solar.

- Voltaje real: 12 V.
- Voltaje nominal: 17.5 V.
- Corriente nominal: 0.58 A.
- Corriente de corto circuito: 0.64 A.
- Tensión de circuito abierto: 21.5 V.
- Tensión máxima del sistema: 1000 V.
- Potencia 10W.
- Temperatura de operación  $-40^{\circ}\text{C} - 85^{\circ}\text{C}$ .
- Eficiencia de conversión: 15%.
- Peso: 1.342 Kg.
- Dimensiones de la celda: 35 cm x 29 cm x 2.5 cm.

### 1.5.9. Controlador de carga solar

El dispositivo que administra la energía que proporciona la celda solar hacia la batería recargable o la carga es el mostrado en la figura 1.26. Cuenta con funciones para el estado de carga (SOC) de las baterías, para regular su proceso de carga y protección contra descarga profunda. Sus características principales son las siguientes.



Figura 1.26. Controlador de carga solar.

- Identificación automática de voltaje.
- Corrección de temperatura.
- Indicador LED para cada condición de trabajo.
- Control por PWM (Modulación por ancho de pulso).
- Incrementar, recuperar e inhabilitar la carga para una larga duración de la batería.
- Protección contra corto circuito, sobre carga, carga excesiva y aumento de temperatura.

# Capítulo 2

## Descripción del proyecto

Las soluciones de energía renovables se están convirtiendo en opción cada vez más popular. Los sistemas fotovoltaicos (solares) no son más que un ejemplo. Para maximizar la potencia de salida de un sistema solar es deseable aumentar su eficiencia. Con el fin de maximizar la potencia de salida de los paneles solares, éstos necesitan ser alineados con el sol. Como tal, se requiere un medio de seguimiento solar. A continuación se describe el sistema electromecánico para el seguimiento solar desarrollado en el presente proyecto de tesis

### 2.1. Esquema general

La figura 2.1 muestra el esquema general del sistema electromecánico desarrollado para el seguimiento de la trayectoria solar.

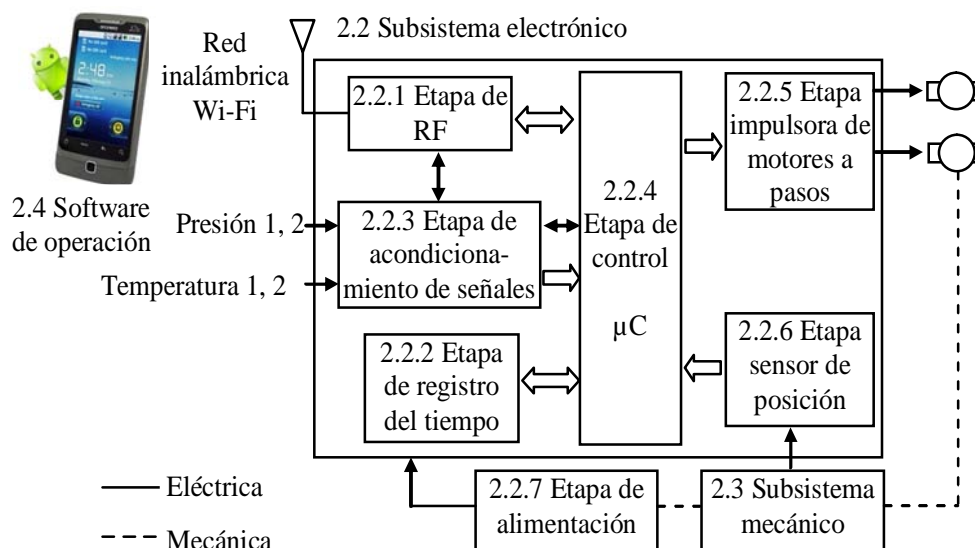


Figura 2.1. Esquema general del seguidor solar.

El método de posicionamiento que desempeña el esquema de la figura 2.1 (seguidor activo) es el seguimiento de la trayectoria del sol con base en cálculos astronómicos de la posición solar en función del registro del tiempo, que se ejecutan en el subsistema electrónico por medio de un microcontrolador,  $\mu\text{C}$ . Se puede realizar el seguimiento solar sobre la base del cálculo de los ángulos de altitud y azimut del sol y girar al subsistema mecánico donde se encuentra montada la etapa de alimentación (celda fotovoltaica) y orientarla de frente al sol. 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 astronómicos 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 del tiempo requerido para el cálculo de los ángulos es proporcionado por el circuito integrado DS1307, 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 de tesis, 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 2.1, el proyecto involucra varias etapas para su funcionamiento, que van desde una aplicación para teléfono celular hasta un sistema electromecánico, pero todo esto se puede clasificar en tres componentes esenciales, las cuales son:

- Subsistema electrónico.
- Subsistema mecánico.
- Software de operación

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

## **2.2. Subsistema electrónico**

El circuito electrónico para el seguimiento solar se compone de las siguientes etapas, figura 2.1.

- 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 la interface con el usuario.
- 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 del subsistema electrónico 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 al sistema electrónico.
- Etapa de acondicionamiento de señales. Se implementa con sencillos circuitos electrónicos 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 la presente tesis.
- 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 al subsistema mecánico.
- Etapa para los sensores de posición. Está conformada por circuitos optoelectrónicos 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 el subsistema mecánico. Se encarga de transformar la energía solar en eléctrica para alimentar al subsistema electrónico en conjunto.

El anexo C contiene los detalles de la implementación del subsistema electrónico. A continuación, una descripción detallada de las diversas etapas del subsistema electrónico.

### 2.2.1. Etapa de radio frecuencia

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, 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 que comunica el módulo con el microcontrolador.

En la figura 2.2 se muestra la configuración electrónica del módulo inalámbrico RN-131-EK utilizado en el proyecto.

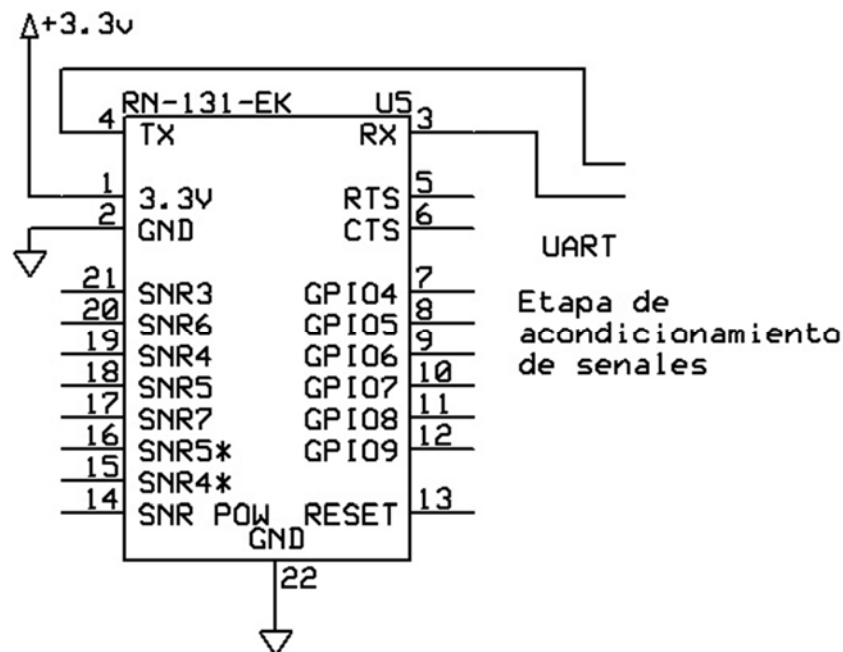


Figura 2.2. Configuración del módulo inalámbrico de Wi-Fi.

El voltaje de polarización del RN-131 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 módulo RN-131-EK además cuenta con tres leds integrados que indican el estado en el cual se encuentra dicho dispositivo. En la tabla 2.1 se muestra sus posibles estados.

<b>Condición</b>	<b>Led rojo</b>	<b>Led amarillo</b>	<b>Led Verde</b>
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 2.1. Posibles condiciones del módulo RN-131-EK.

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 módulo 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.

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 2.3 se aprecia la comunicación que establece el módulo 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.



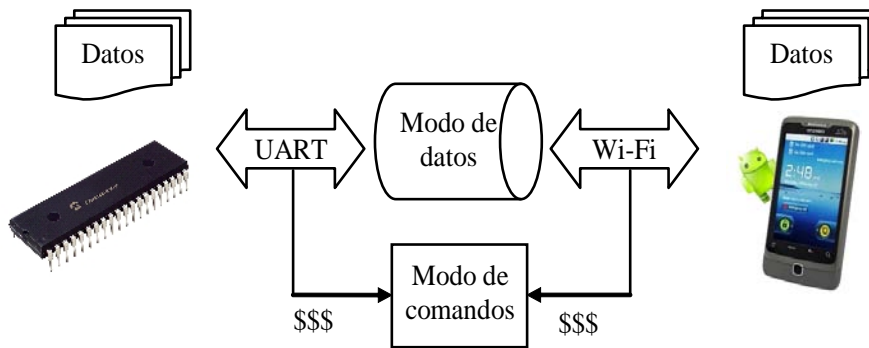


Figura 2.3. Modos de operación del módulo inalámbrico RN-131.

### 2.2.2. Etapa de registro del tiempo

Para el registro del tiempo, o sea 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 2.4 se muestra la configuración del reloj de tiempo real empleado.

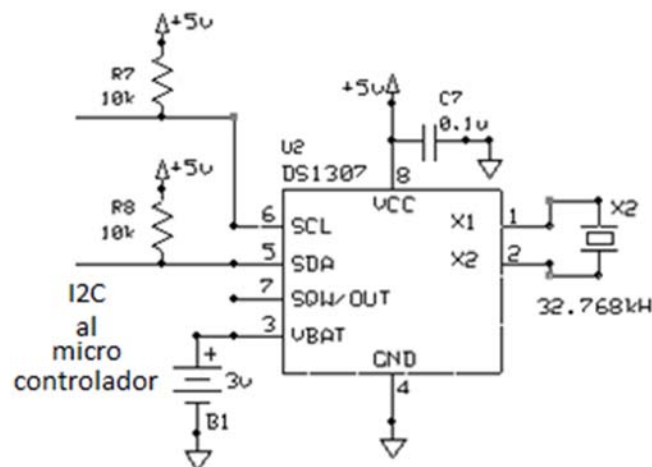


Figura 2.4. Configuración del reloj de tiempo real, DS1307.

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 I<sup>2</sup>C, pero se

necesitan un par de resistencias Pull-up para cada línea con el fin de poder transmitir y recibir correctamente.

### 2.2.3. Etapa de acondicionamiento de señales

El subsistema electrónico requiere 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 el subsistema electrónico. 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, utilizada 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 2.5 se muestra la configuración del sensor de temperatura LM35.

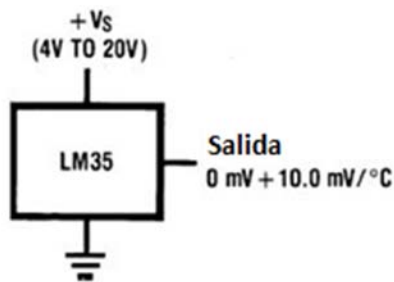


Figura 2.5. Configuración del sensor de temperatura LM35.

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 de utilizar la siguiente formula.

$$Temp(^{\circ}C) = \frac{148}{1023}V(Hex) + 11(^{\circ}C)$$

### 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 Electronics, 2015]. La configuración utilizada se muestra en la figura 2.6.

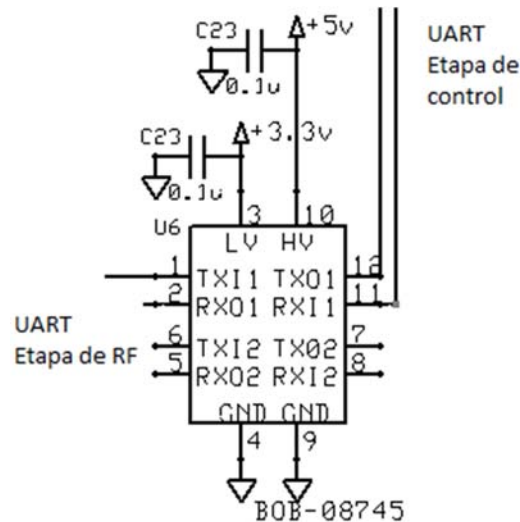


Figura 2.6. Convertidor de nivel BOB-08745.

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 está 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 de 4 pines en la parte baja con dos entradas y dos salidas previstas para cada lado.

#### 2.2.4. Etapa de control

En la etapa de control se encuentra el microcontrolador PIC18F4550 de 8 bits descrito en la sección 1.3 [Microchip, 2007]. En la figura 2.7 se muestra la configuración y puertos usados.

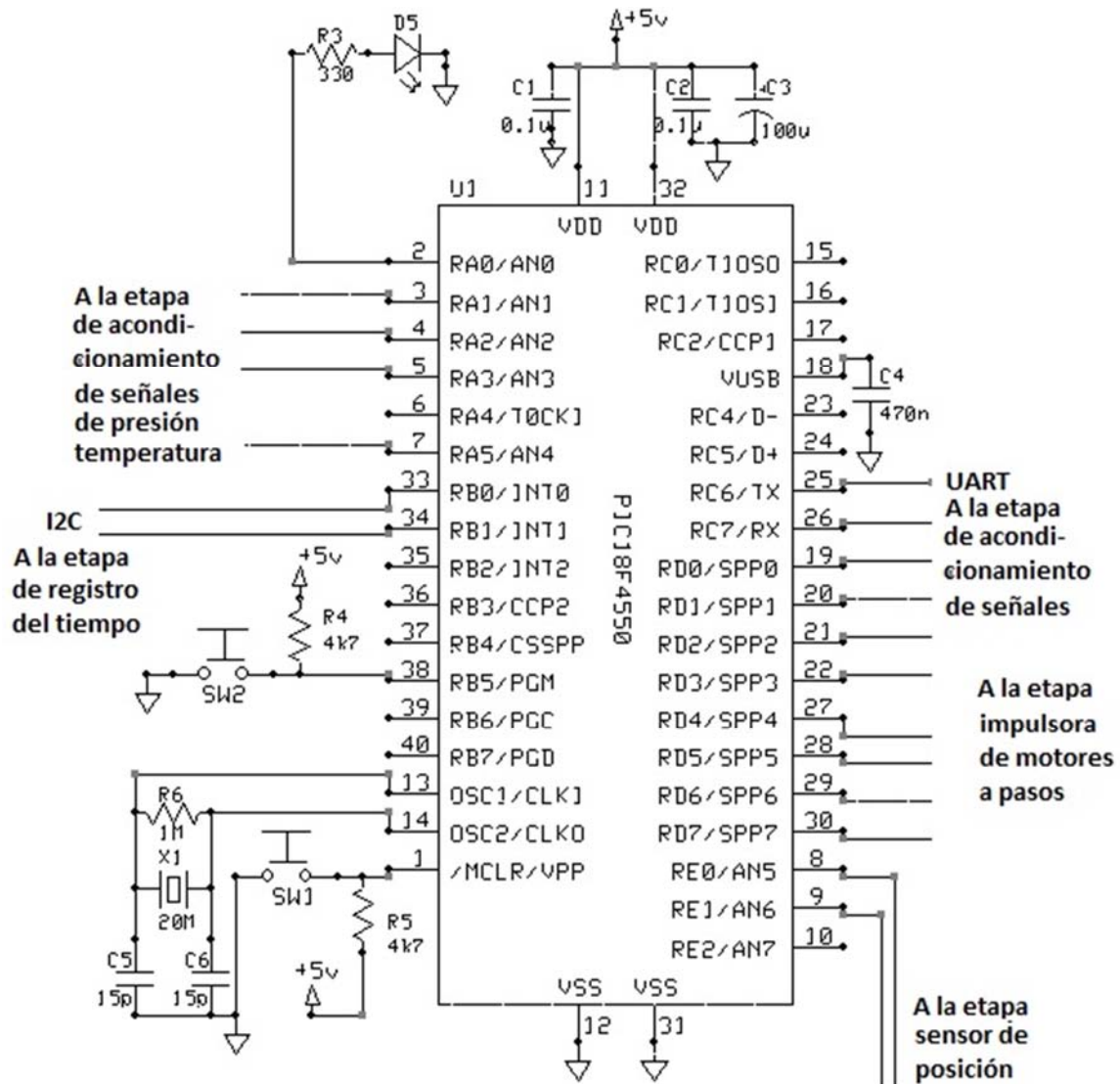


Figura 2.7. Configuración del PIC18F4550.

Dentro de las comunicaciones que establece el microcontrolador se encuentra la comunicación UART “Universal Asynchronous Receiver - Transmitter”. 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 I<sup>2</sup>C “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 I<sup>2</sup>C 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 tienen 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 el subsistema mecánico (cero altitud y azimut), se habilitaron los puertos RE0 y RE1 como entradas para detectar las señales que envía los dos sensores de posición.

### 2.2.5. Etapa impulsora de motores a pasos

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 Semiconductor, 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 2.8 se muestra la configuración de este dispositivo.

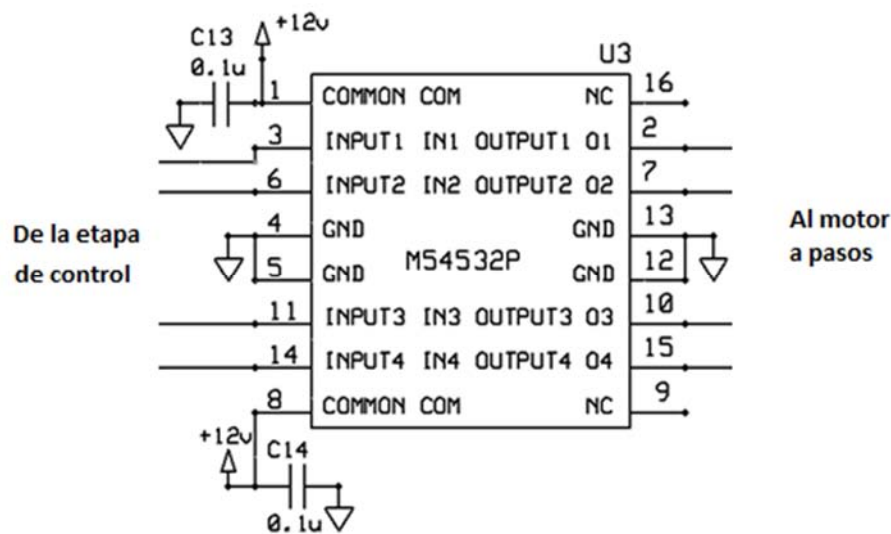


Figura 2.8. Configuración del dispositivo M54532P.

Como se puede observar en la figura 2.8 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 2.8 se implementa dos veces en el subsistema electrónico, uno de ellos se utiliza para motor del ángulo de altitud y el otro para el motor de azimut.

### 2.2.6. Etapa para el sensor de posición

Para el posicionamiento del subsistema mecánico se requiere de sensores adecuados que proporcionen señales de información cuando se encuentre en la posición deseada o requerida.

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

El funcionamiento de los optoacopladores 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 2.9 se puede observar la configuración de los optoacopladores.

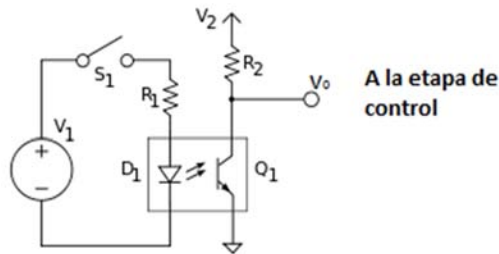


Figura 2.9. Configuración del optoacoplador para el posicionamiento del sistema.

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, altitud o azimut.

### 2.2.7. 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 2.10 muestra el diagrama de conexión.

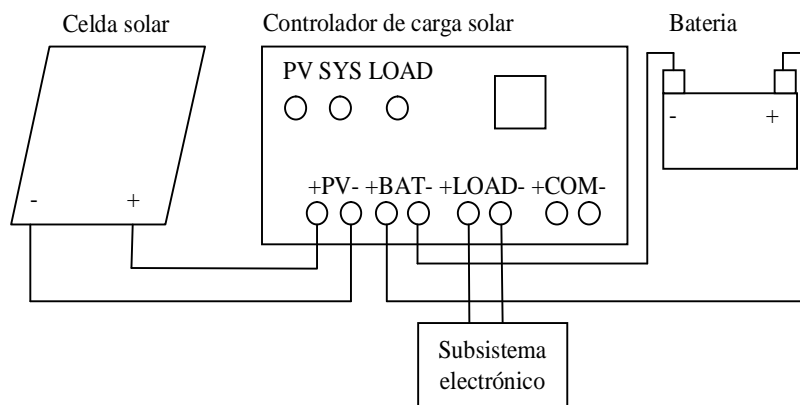


Figura 2.10. Conexión de la etapa de alimentación.

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, como se mencionó en la sección 1.5.9, el subsistema 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 2.2 del led SYS.

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 2.2. Descripción del led SYS.

Con este sistema de 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 2.2

se sabe el estado en que se encuentra el sistema, ya que indica cuando hay una sobrecarga.

### 2.2.8. 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. Está compuesto por una rutina de inicio, una rutina principal y una interrupción.

#### Rutina de inicio

La rutina de inicio tiene como objetivo posicionar el subsistema mecánico. 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 subsistema mecánico, de lo contrario se inverna. La figura 2.11 muestra las actividades que se ejecutan como parte de esta rutina inicial.

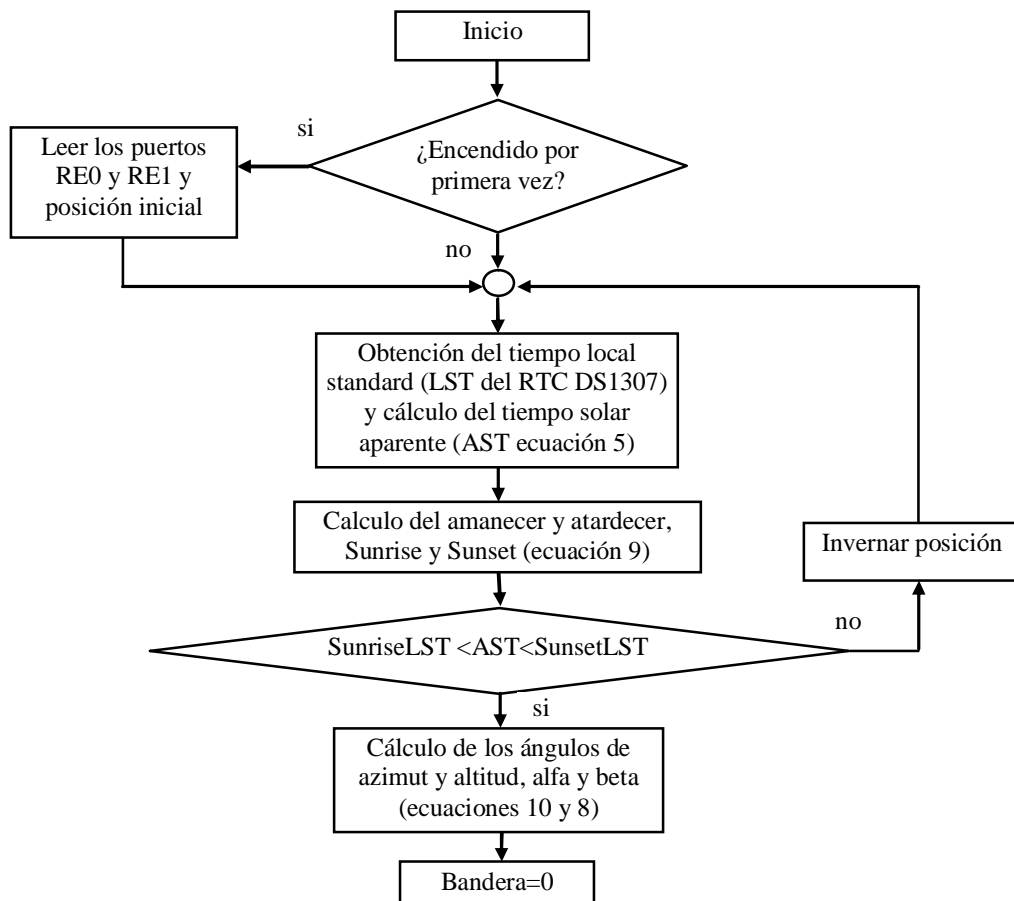


Figura 2.11. Rutina de inicio.



## Rutina principal

La figura 2.12 muestra la rutina principal del programa. Establecido el punto de inicio en el sistema mecánico, se ejecuta un ciclo en donde se encuesta el paquete de datos recibido por Wi-Fi en la etapa de RF y a su vez transmitido por el puerto UART al microcontrolador.

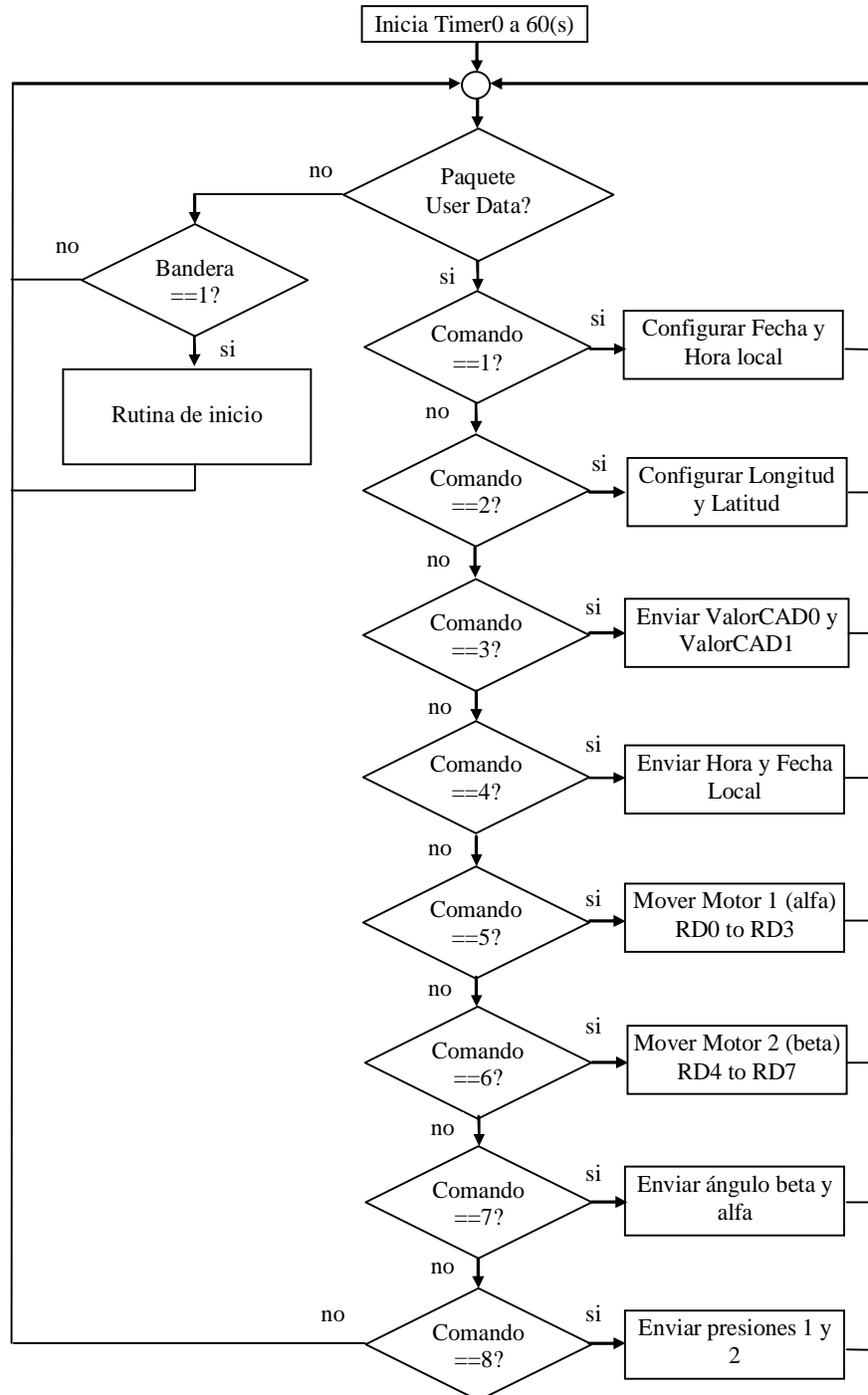


Figura 2.12. Rutina principal.

El paquete está compuesto por una serie de instrucciones a ejecutar. Algunas posibles instrucciones de envío de información desde el sistema electrónico a la interface de usuario mediante Wi-Fi son:

- Envía las mediciones de la temperatura T1 y T2 almacenadas en las variables ValorCAD0 y ValorCAD1 respectivamente.
- Envía la posición del ángulo azimut, alfa.
- Envía la posición del ángulo altitud, beta.
- Envía la fecha y hora del circuito electrónico.

Los datos a actualizar se envían desde la interface de usuario hacia el sistema electrónico 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 2.13 inicia cuando el TIMER0 del PIC18F4550 se desborda al cabo de 60(s) desde que se establece el “inicio” en la parte mecánica del sistema. Esto genera una interrupción e inicia una rutina que realiza la obtención de la hora local, a través de la comunicación I<sup>2</sup>C 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.

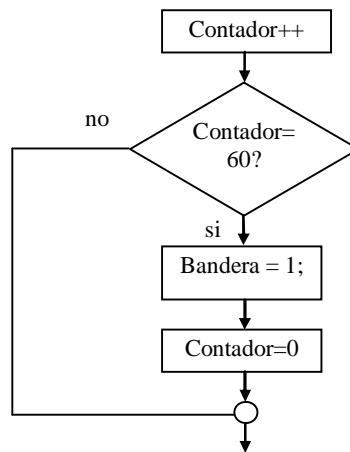


Figura 2.13. Interrupción del TIMER0.

### 2.3. Subsistema mecánico

Se diseñó y construyó el subsistema mecánico de la figura 2.14 para el seguimiento de la trayectoria solar con dos grados de libertad (azimut y altitud).

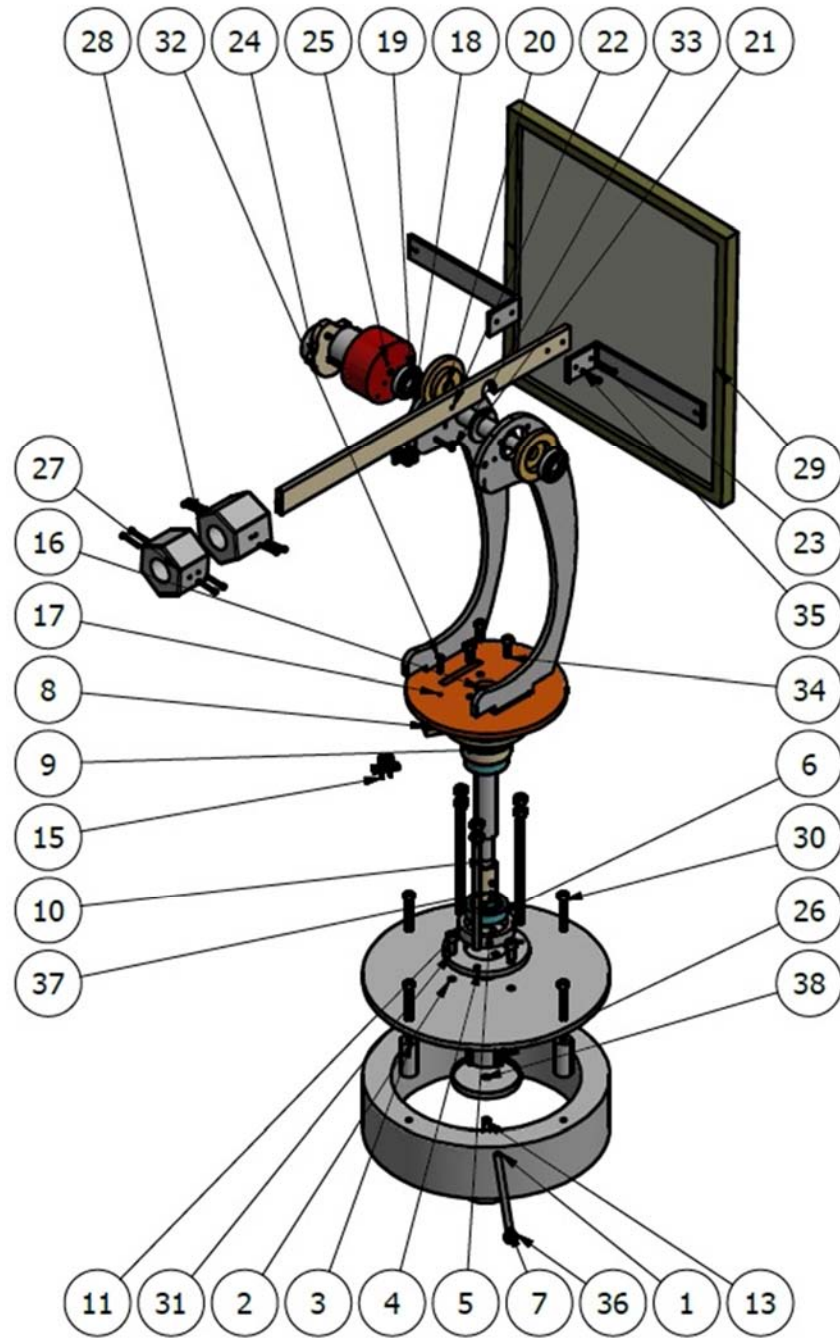


Figura 2.14. Subsistema mecánico.

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 componentes del subsistema mecánico se describen a continuación. Los detalles de las piezas construidas se muestran en el anexo D.

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 azimutal.

## 2.4. Descripción del software de operación 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”. La pantalla principal de la aplicación se muestra en la figura 2.15.

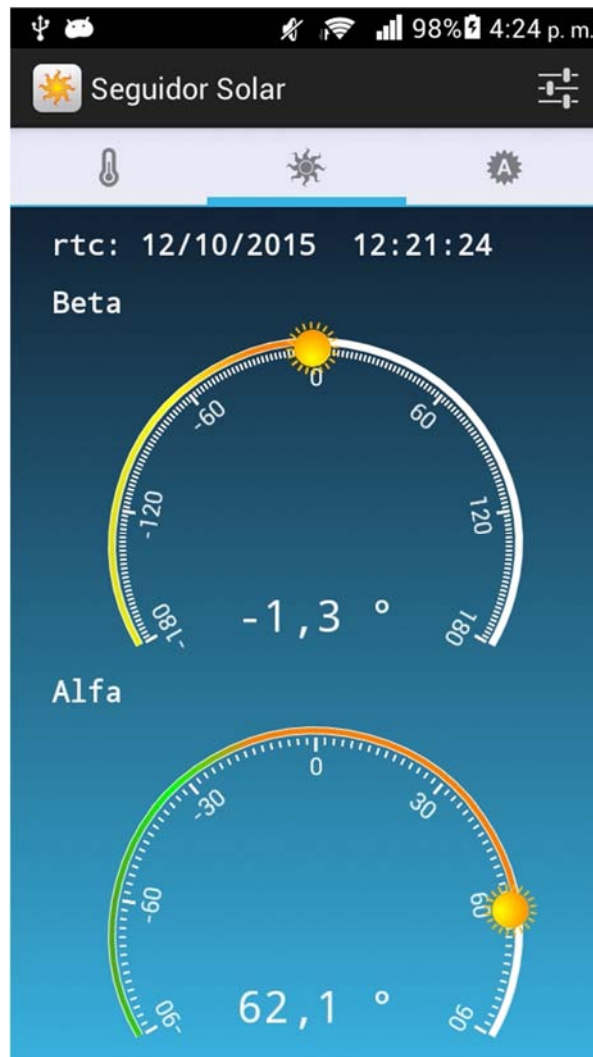


Figura 2.15. Pantalla principal de la aplicación “Seguidor solar”.

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 azimut (Alfa) y altitud (Beta). Adicionalmente, se despliega la fecha y hora actualmente configuradas en el subsistema electrónico (rtc). Es importante mencionar que la fecha y hora del subsistema electrónico pueden ser distintas a las correspondientes del dispositivo móvil. La información se obtiene cada 5 segundos del sistema electromecánico mediante la interface inalámbrica Wi-Fi, no obstante que el sol se desplace  $1^\circ$  cada 4 min.

Al presionar la pestaña del extremo izquierdo se muestra la información de temperatura en los sensores del subsistema electromecánico, como se muestra en la figura 2.16.

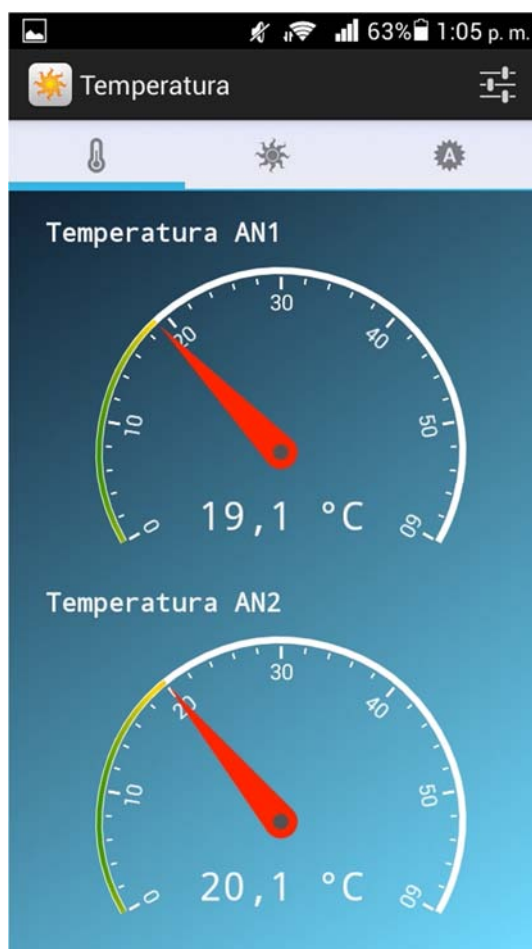


Figura 2.16. Pantalla para el despliegue de temperatura.

La temperatura en los canales de conversión AN1 y AN2 del subsistema electrónico se despliega de forma analógica y también su valor es refrescado cada 5 segundos.

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 2.17. De igual forma que en las pantallas anteriores, la información se refresca automáticamente cada 5 segundos.

Las pantallas de las figuras 2.15 a 2.17 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.

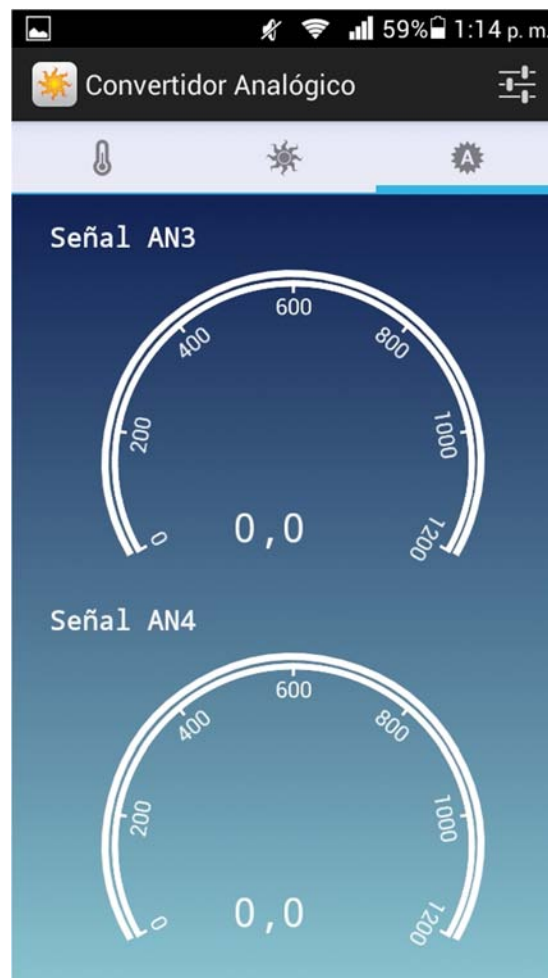


Figura 2.17. Pantalla para el despliegue de los convertidores analógico a digital.

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 2.18. Como se muestra en la tabla 1.2, la fecha y la hora LST son datos de entrada al algoritmo para el cálculo de los ángulos solares que se ejecuta en el subsistema electrónico. Con estos datos de fecha y hora se configura la etapa de registro del tiempo. Una vez configurada la fecha y hora, esta se mantiene actualizada de forma automática por el reloj de tiempo real en el subsistema electrónico 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 2.19.



Figura 2.18. Pantalla para el ajuste del tiempo.

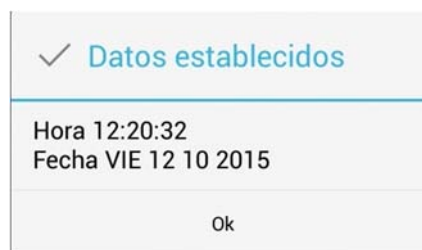


Figura 2.19. Diálogo de confirmación para el ajuste del tiempo.

Finalmente, 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 2.20. Como se muestra en la tabla 1.2, la latitud y longitud también son datos de entrada al algoritmo para el cálculo de los ángulos solares que se ejecuta en el subsistema electrónico. 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 2.21.



Figura 2.20. Pantalla para el ajuste de la posición.



Figura 2.21. Diálogo de confirmación para el ajuste de la posición.

## 2.5. Descripción del software de operación 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.

### 2.5.1. Descripción general del proyecto

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 2.22. Entre los elementos más importantes destacan los siguientes:

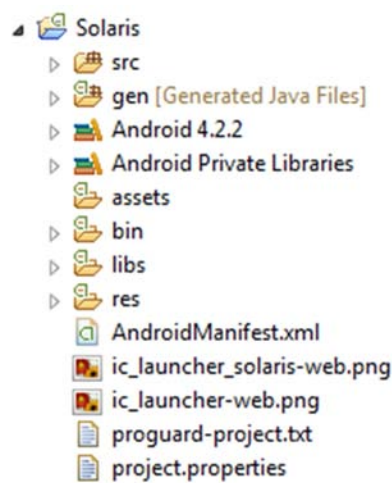


Figura 2.22. Estructura de la aplicación en Android.

- **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 2.3.

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 activity_config para la configuración.
res/menu	Contiene un archivo XML que define un menú para la aplicación.
res/values	Contiene los siguientes archivos: <ul style="list-style-type: none"> <li>• attrs.xml: define los atributos que puede utilizar la clase Meter y cualquier clase que extienda la funcionalidad de esta.</li> <li>• string.xml: define cadenas de texto</li> </ul>

	utilizadas durante la aplicación. <ul style="list-style-type: none"> <li>• <code>styles.xml</code>: establece el tema base para la aplicación.</li> </ul>
--	---

Tabla 2.3. 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.

### 2.5.2. 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 2.23.

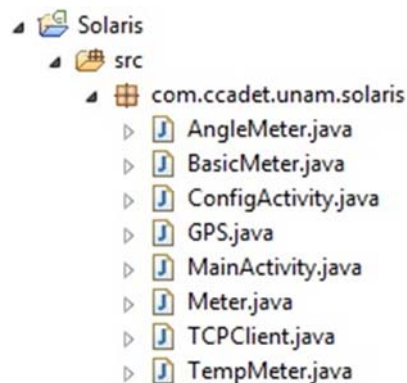


Figura 2.23. Clases de la aplicación.

#### 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 2.24.

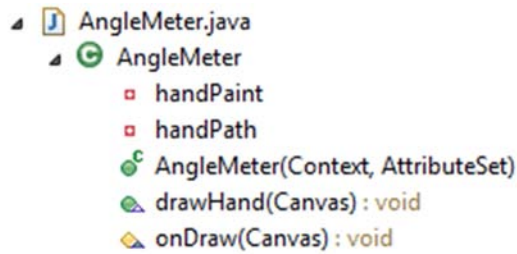


Figura 2.24. Elementos de la Clase AngleMeter.

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 2.4.

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 2.4. Métodos de la clase AngleMeter.

Los atributos necesarios para dibujar este elemento se describen en la tabla 2.5.

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

Tabla 2.5. 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 2.25 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.

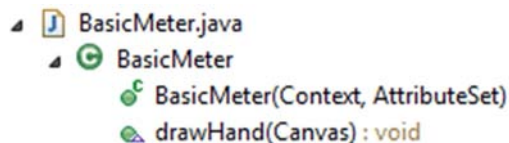


Figura 2.25. Metodos de la clase BasicMeter.

## 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 2.26. se muestran los atributos y métodos de esta clase.

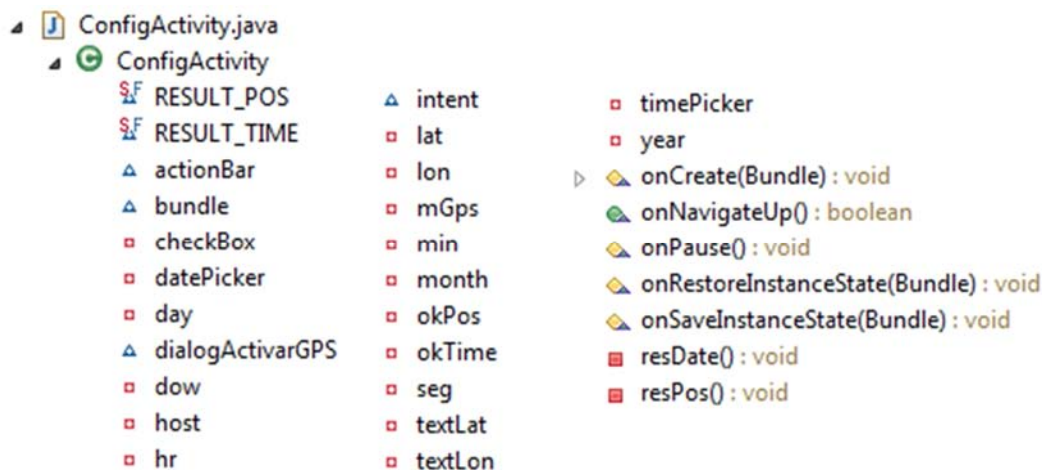


Figura 2.26. Métodos y atributos de la clase ConfigActivity.

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 de forma manual o automática la latitud y longitud, ambas vistas cuentan con un botón para seleccionar los datos y finalizar la actividad. La tabla 2.6 muestra los atributos y métodos de la clase.

<b>Tipo</b>	<b>Atributo</b>	<b>Significado</b>
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.
DialogActivar GPS	dialogActivarGPS	Dialogo utilizado para activar el GPS del dispositivo en caso de no estar desactivado.

Tabla 2.6. 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 2.7 describe los métodos utilizados en esta clase.

Método	Descripción
<code>onCreate(Bundle)</code>	Establece el contenido de la interface gráfica definida por el <code>layout 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 <code>MainActivity</code> , 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 <code>String</code> para ser colocados en un objeto <code>Bundle</code> .
<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 <code>Bundle</code> .

Tabla 2.7. 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 2.27 muestra los atributos y métodos que conforman esta clase.



Figura 2.27. 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 2.8 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 2.8. 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 2.9 muestra los métodos utilizados por esta clase.

<b>Método</b>	<b>Descripción</b>
<code>GPS(Context)</code>	Constructor de la clase.
<code>getLocation()</code>	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.
<code>onLocationChanged(Location)</code>	Es invocado cada que se recibe una nueva posición.
<code>onStatusChanged(String, int, Bundle)</code>	No utilizado.
<code>onProviderEnabled(String)</code>	No utilizado.
<code>onProviderDisabled(String)</code>	No utilizado.
<code>onBind(Intent)</code>	No utilizado.
<code>stopGPS()</code>	Cancela las actualizaciones del proveedor de localización.
<code>getLat()</code>	Devuelve la latitud en una variable tipo <code>double</code>
<code>getLon()</code>	Devuelve la longitud en una variable tipo <code>double</code>

Tabla 2.9. 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 2.28 muestra los atributos y métodos de la clase `MainActivity`.

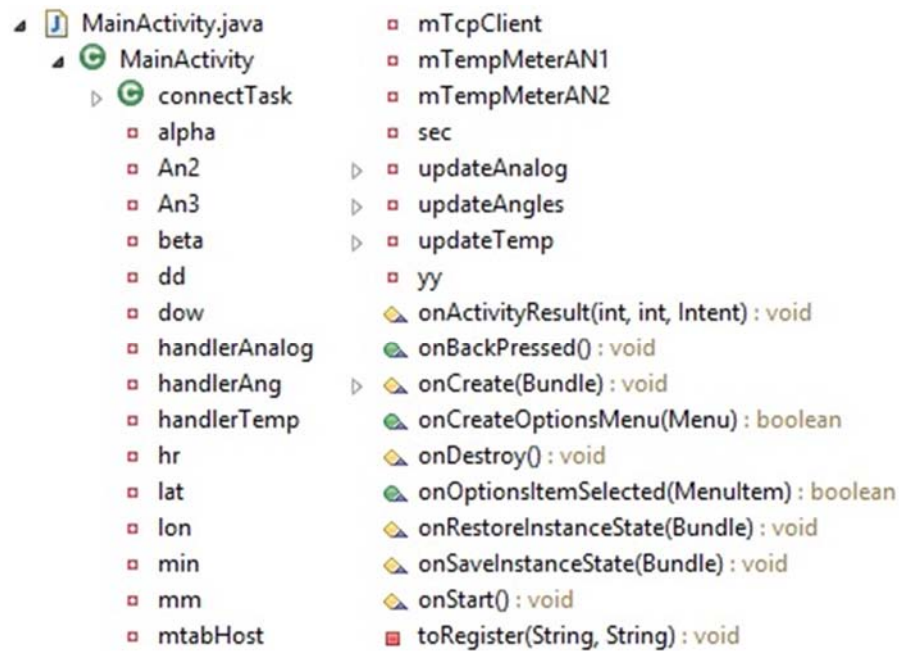


Figura 2.28. Elementos de la clase MainActivity.

La tabla 2.10 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 del 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	Variables que permiten el acceso a los eventos periódicos.

	handlerAnalog	
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 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 2.10. Atributos de la clase MainActivity.

Los métodos de esta clase, descritos en la tabla 2.11, 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 2.11. 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 2.29.

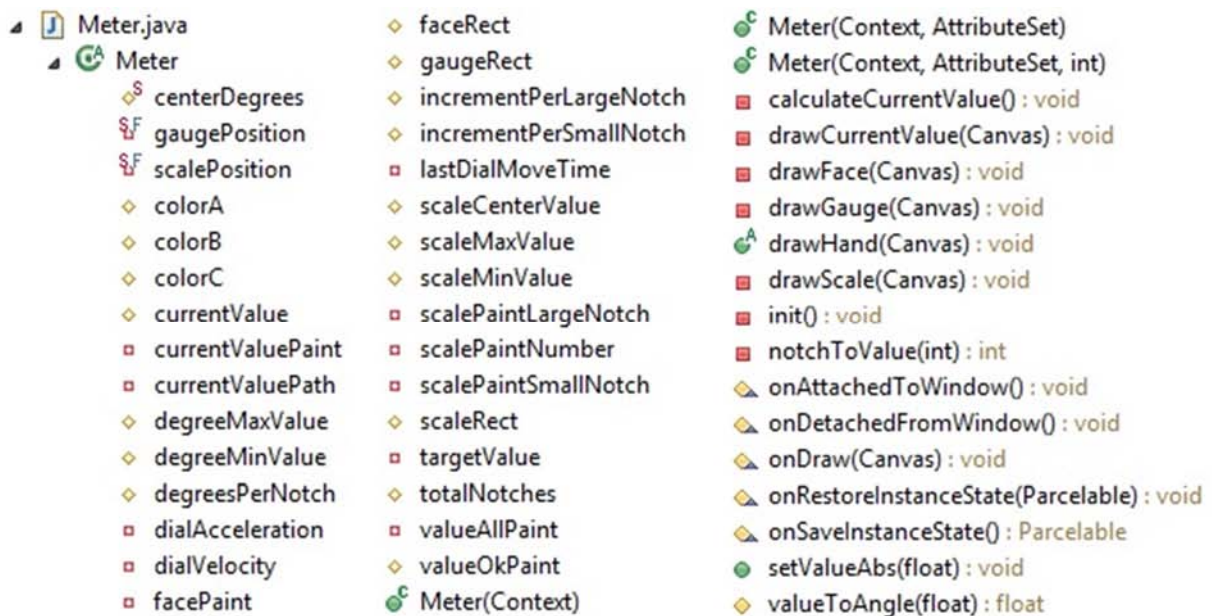


Figura 2.29. Elementos de la clase Meter.

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 2.12.

<b>Tipo</b>	<b>Atributo</b>	<b>Significado</b>
float float int int int int int int float float int float float float float float	degreeMinValue degreeMaxValue totalNotches incrementPerLargeNotch incrementPerSmallNotch scaleMinValue scaleCenterValue scaleMaxValue <i>scalePosition</i> <i>gaugePosition</i> <i>centerDegrees</i> degreesPerNotch currentValue targetValue dialVelocity dialAcceleration	Constantes geométricas que se utilizan para dimensionar el dibujo sobre el lienzo.
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 2.12. Atributos de la clase Meter.

La aplicación utiliza los métodos mostrados en la tabla 2.13 para dibujar un indicador analógico a partir de geometrías elementales.

<b>Método</b>	<b>Descripción</b>
Meter(Context context) Meter(Context context, AttributeSet attributeSet) Meter(Context context, AttributeSet	Constructores de la clase que reciben diversos parámetros con fines de inicialización.

atrAttributeSet,int defStyleAttr)	
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 2.13. 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 2.30.



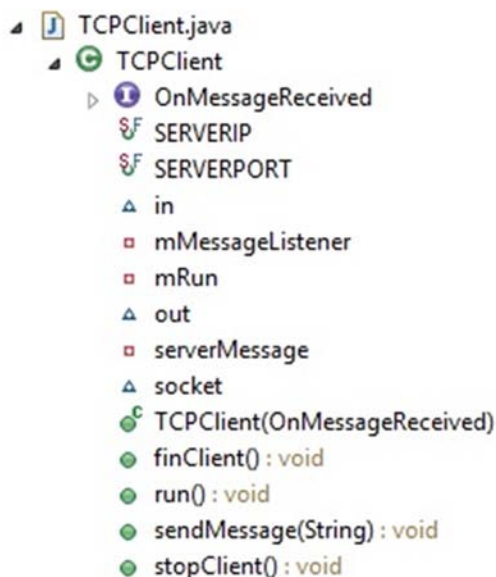


Figura 2.30. Elementos de la clase TCPClient.

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 2.14.

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 TCP recibido.
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 2.14. Atributos de la clase TCPClient.

Los métodos de esta clase se describen en la tabla 2.15.

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 2.15. 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 2.31.

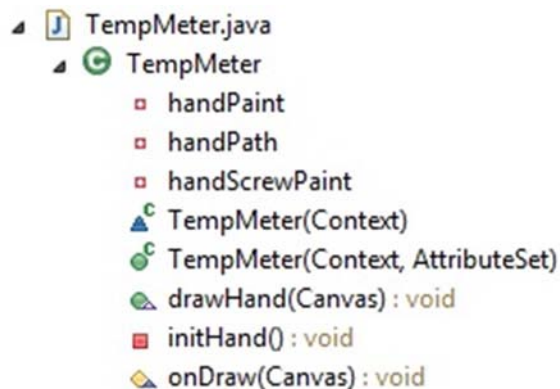


Figura 2.31. Elementos de la clase TempMeter.

Los atributos de esta clase se describen en la tabla 2.16.

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 2.16. Atributos de la clase TempMeter.

Los métodos de esta clase se describen en la tabla 2.17.

<b>Método</b>	<b>Descripción</b>
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 2.17. Métodos de la clase TempMeter.

# Capítulo 3

## Resultados

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 los subsistemas electrónico, mecánico y software. Para la evaluación cuantitativa se desarrollaron dos experimentos que verifican el desempeño.

#### 3.1.1. Prototipo desarrollado

La figura 3.1 muestra el prototipo desarrollado en conjunto.



Figura 3.1. Prototipo desarrollado.

La figura 3.2 muestra el subsistema electrónico fabricado en circuito impreso.



Figura 3.2. Subsistema electrónico.

La figura 3.3 muestra el software de operación con la pestaña correspondiente a los ángulos solares calculados en el sistema electromecánico.

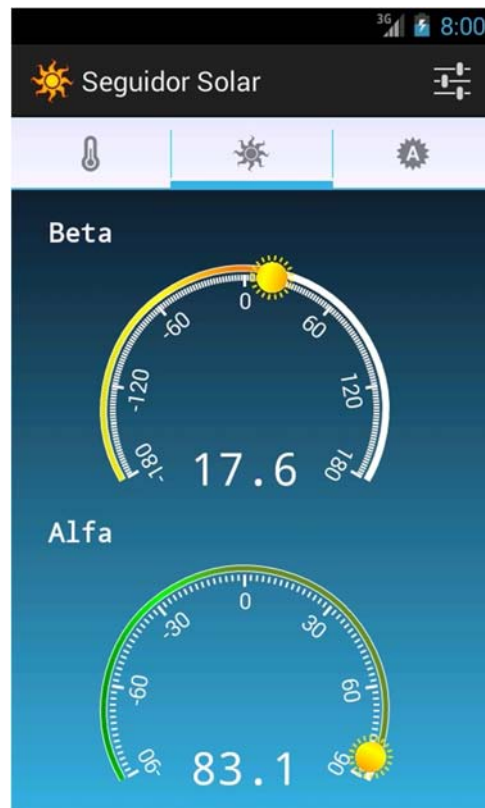


Figura 3.3. Software de operación.

La figura 3.4 muestra dos detalles del subsistema mecánico en donde se aprecian los mecanismos que permiten el posicionamiento del panel en los dos grados de libertad (elevación y azimut).

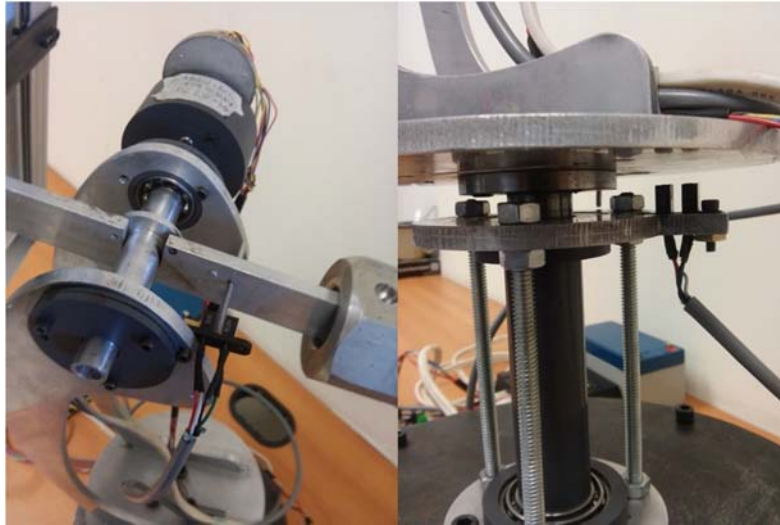


Figura 3.4. Componente mecánica.

### 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 1.2.4 y 1.2.5 programado en MatLab y se compararon contra el mismo algoritmo programado en lenguaje C para el microcontrolador PIC18F4550. Las tablas 3.1 a 3.3 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

<b>Solsticio de verano 21 de junio</b>	06:02	19:13	25.8	107.3	06:02	19:13	25.8	107.3
<b>Solsticio de invierno 21 de diciembre</b>	07:10	17:59	10.4	60.4	07:10	17:59	10.41	60.4

Tabla 3.1. Cálculos para la Ciudad de México 8:00.

<b>Ciudad de México 13:00 tiempo local</b>	<b>Microcontrolador</b>				<b>Matlab</b>			
	<b>Sunrise</b>	<b>Sunset</b>	<b>Beta</b>	<b>Alfa</b>	<b>Sunrise</b>	<b>Sunset</b>	<b>Beta</b>	<b>Alfa</b>
<b>Equinoccio de primavera 21 de marzo</b>	06:44	18:43	69.7	-11.38	06:44	18:43	69.7	-11.3
<b>Equinoccio de otoño 22 de septiembre</b>	06:30	18:28	68.5	-22.0	06:29	18:27	68.5	-22.0
<b>Solsticio de verano 21 de junio</b>	06:02	19:13	83.4	-129.07	06:02	19:13	83.4	-129
<b>Solsticio de invierno 21 de diciembre</b>	07:10	17:59	46.6	-8.65	07:10	17:59	46.7	-8.3

Tabla 3.2. Cálculos para la Ciudad de México 13:00.

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

Tabla 3.3. 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 3.5.

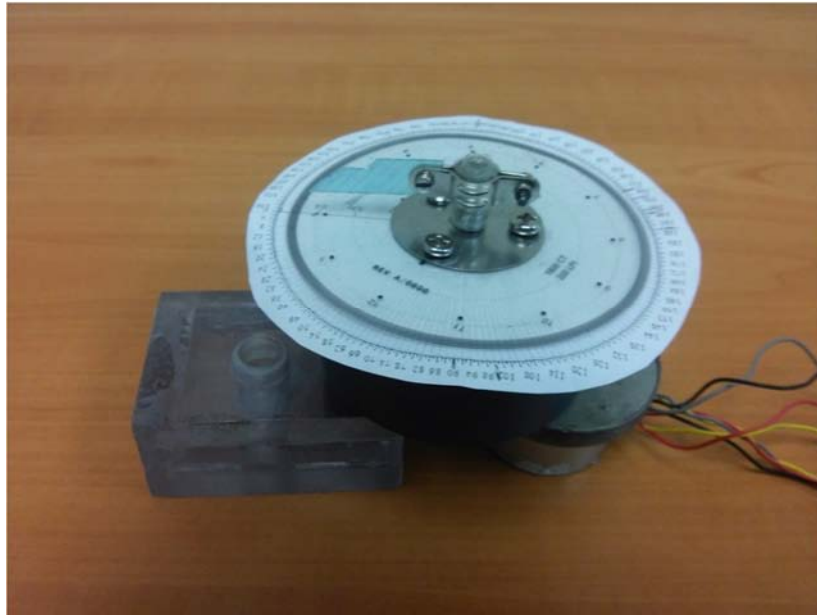


Figura 3.5. Arreglo experimental para evaluar el desempeño en posición.

Se hizo un recorrido de  $0$  a  $180^\circ$  y de regreso hasta  $-180^\circ$  en intervalos de  $10^\circ$ , desde una posición inicial que equivale a cero. Existe una diferencia de hasta  $4^\circ$  entre el objetivo y el ángulo observado, aunque si se moviera el motor de  $0^\circ$  a  $180^\circ$  el ángulo que se fijó como objetivo corresponde al observado. La tabla 3.4 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 3.4. 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 3.5.

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 3.5. Datos registrados para el motor para la altitud.

## 3.2. Discusión

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.

### **3.3. Trabajo a futuro**

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.
- Optimizar el número de componentes de la tarjeta electrónica para reducir su costo y tamaño.

# Conclusiones

Se diseñó y construyó un sistema electromecánico con dos grados de libertad para el seguimiento de la trayectoria solar que es calculada mediante ecuaciones solares y una base de tiempo. 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. El subsistema electrónico, mecánico 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.

# Bibliografía

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 Rama, 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].

# Anexos

En esta sección se presenta la información complementaria al presente trabajo de tesis.

## A. Código fuente MatLab

```
clear all;
close all;

Ano=2014;
Mes=7;
Dia=21;

Hora=8;
Minutos=0;

Latitud=33.43;
Longitud=112; %Longitudes oeste

Latitudrad=Latitud*(pi/180);

switch Mes
case 1
    N=Dia;
case 2
    N=31+Dia;
case 3
    N=31+28+Dia;
case 4
    N=31+28+31+Dia;
case 5
    N=31+28+31+30+Dia;
case 6
    N=31+28+31+30+31+Dia;
case 7
```

```

    N=31+28+31+30+31+30+Dia;
case 8
    N=31+28+31+30+31+30+31+Dia;
case 9
    N=31+28+31+30+31+30+31+31+Dia;
case 10
    N=31+28+31+30+31+30+31+31+30+Dia;
case 11
    N=31+28+31+30+31+30+31+31+30+31+Dia;
case 12
    N=31+28+31+30+31+30+31+31+30+31+30+Dia;
end

if((N>=1)&&(N<=172))
    MesP=((15.95*(N-172))/92)+15.95;
elseif((N>=173)&&(N<=355))
    MesP=((2*15.95*(N-172))/(172-355))+15.95;
elseif((N>=356)&&(N<=365))
    MesP(((13.869-15.95)*(N-355))/(355-365))-15.95;
end

D=(360/365)*(N-81);
Drad=(pi*D)/180;
ET=(9.87*sin(2*Drad))-(7.53*cos(Drad))-(1.5*sin(Drad));
LSTM=15*fix(Longitud/15);

angulorad=(N+284)*(360/365)*(pi/180);
delta=23.45*sin(angulorad);
deltarad=delta*(pi/180);

angulorad=-tan(Latitudrad)*tan(deltarad);
Hsrad=-acos(angulorad);
Hs=Hsrad*(180/pi);

sunriseAST=720+(Hs*4);
sunriseASTHora=fix(sunriseAST/60);
sunriseASTMinuto=sunriseAST-(sunriseASTHora*60);
sunriseLSTHora=sunriseASTHora;
sunriseLSTMinuto=sunriseASTMinuto-(4*(LSTM-Longitud))-ET;
% sunrise(N)=(sunriseLSTHora*60)+sunriseLSTMinuto;
sunrise=(sunriseLSTHora*60)+sunriseLSTMinuto;

sunsetAST=720+(-Hs*4);
sunsetASTHora=fix(sunsetAST/60);
sunsetASTMinuto=sunsetAST-(sunsetASTHora*60);
sunsetLSTHora=sunsetASTHora;
sunsetLSTMinuto=sunsetASTMinuto-(4*(LSTM-Longitud))-ET;
% sunset(N)=(sunsetLSTHora*60)+sunsetLSTMinuto;
sunset=(sunsetLSTHora*60)+sunsetLSTMinuto;

```

```

Hnrad=0;
Hn=0;

sunnoonAST=720+(Hn*4);
sunnoonASTHora=fix(sunnoonAST/60);
sunnoonASTMinuto=sunnoonAST-(sunnoonASTHora*60);
sunnoonLSTHora=sunnoonASTHora;
sunnoonLSTMinuto=sunnoonASTMinuto-(4*(LSTM-Longitud))-ET;
% sunrise(N)=(sunriseLSTHora*60)+sunriseLSTMinuto;
sunnoon=(sunnoonLSTHora*60)+sunnoonLSTMinuto;

ASTminutos=Minutos+(4*(LSTM-Longitud))+ET;
H=((Hora*60)+ASTminutos)-720)/4;
Hrad=H*(pi/180);

angulorad=(cos(Latitudrad)*cos(deltarad)*cos(Hrad))+(sin(Latitudrad)*sin(deltarad));
betalrad=asin(angulorad);
betal=betalrad*(180/pi);

angulorad=((sin(betalrad)*sin(Latitudrad))-sin(deltarad))/(cos(betalrad)*cos(Latitudrad));
alfalrad=acos(angulorad);
if(H>0) alfa=alfalrad*(180/pi);
else alfa=-alfalrad*(180/pi);
end

tiempo=(Hora*60)+Minutos;
if((tiempo>sunrise)&&(tiempo<sunset))
    alfa=alfal;
    beta=betal;
else
    alfa=0;
    beta=0;
end

pos=(H+90)*(pi/180);
HourAngle=H;

alfa
beta
H
sunriseLSTHora
sunriseLSTMinuto
sunsetLSTHora
sunsetLSTMinuto
Rx=Latitud;
Rx
MesP

```



## B. Código fuente PovRay

```
// Vers: MegaPOV 0.6

#version 3.1;
#include "colors.inc"

// Valores obtenidos de MatLab
declare Alfa = -96.6969;
declare Beta = 28.6108;
declare H = -68.5125;
declare Rx = 33.43;
declare r=40;
declare MesP = 10.7205;

// Correcciones para acoplar los sistemas de coordenadas
declare Alfa = -Alfa;
declare Posicion =1.570796+H*0.0174532925;

// Especificaciones generales
global_settings{assumed_gamma 1.8}
background { color Cyan }

// CAMARAS

// Camara global
camera {
    perspective
    location <0, 0, -110>
    look_at <0, 0, 0>
    rotate <-60, 0, -130>
    angle 65
}

// LUCES

// Luz ambiental
light_source {<100, 100, -100> color White shadowless}

//OBJETOS

//Flecha
#declare My_Flecha=
union {
    intersection {
        cylinder { <45,0,0>, <40,0,0>, 2 }
        box { <50, 5, -0.5>, < 40, -5, 0.5> }
    }
    intersection {
```

```
        cone { <50, 0, 0>, 0 <45, 0, 0>, 5 }
        box { <50, 5, -0.5>, < 40, -5, 0.5> }
    }
}

//Plano
#declare My_Plano=
intersection {
    sphere { <0, 0, 0>, r }
    box { <-r, -r, 0.5>, < r, r, -0.5> }
}

//Poste del seguidor
#declare My_Poste=
union {
    intersection {
        sphere { <0, 0, 0>, 5 }
        box { <-5, -5, 0.5>, < 5, 5, -0.5> }
    }
    cylinder { <0,0,0>, <0,0,-10>, 1.0 }
}

// Panel solar
#declare My_Panel=
union {
    cylinder { <0,0,0>, <0,-5,0>, 0.5 }
    box { <-5, -4.75, -5>, < 5, -5.25, 5> }
}

// DIBUJA OBJETOS

// Plano de tierra y referencias
object { My_Flecha rotate <0, 0, 0> translate <0, 0, 0> pigment{
color Green }}
object { My_Flecha rotate <0, 0, 90> translate <0, 0, 0> pigment{
color Green }}
object { My_Flecha rotate <0, 0, 180> translate <0, 0, 0> pigment{
color Green }}
object { My_Flecha rotate <0, 0, 270> translate <0, 0, 0> pigment{
color Green }}
object { My_Plano rotate <0, 0, 0> translate <0, 0, 0> pigment{ color
Green }}

// Seguidor solar
object { My_Poste rotate <0, 0, Alfa> translate <0, 0, -1.0> pigment{
color Gray } }
object { My_Panel rotate <Beta, 0, 0> rotate <0, 0, Alfa> translate
<0, 0, -10> pigment{ color Gray } }

// Texto
```

```

text { ttf "timrom.ttf" "E" 0.15, 0 texture {pigment { color Green }}
scale 10 translate <52, -2, 0> }
text { ttf "timrom.ttf" "N" 0.15, 0 texture {pigment { color Green }}
scale 10 translate <-2, 52, 0> }
text { ttf "timrom.ttf" "O" 0.15, 0 texture {pigment { color Green }}
scale 10 translate <-58, -2, 0> }
text { ttf "timrom.ttf" "S" 0.15, 0 texture {pigment { color Green }}
scale 10 translate <-2, -58, 0> }

// Boveda solar
sphere { <0, 0, 0>, r texture { pigment{ color rgbt<1,0.7,0,0.7>}
finish { diffuse 0.9 phong 0.5}} }

// Solsticios
object { My_Flecha translate <-10, 0, -0.5> rotate <0, 0, 23.45>
pigment{ color Red }}
object { My_Flecha translate <-10, 0, -0.5> rotate <0, 0, -23.45>
pigment{ color Red }}

// Sol
light_source {<sqrt((r*r)-(MesP*MesP))*cos(Posicion), 0, -sqrt((r*r)-
(MesP*MesP))*sin(Posicion)> color White translate <0,MesP, 0> rotate
<Rx, 0, 0> looks_like {
sphere { <0, 0, 0>, 2 texture {pigment { color Yellow }} } }

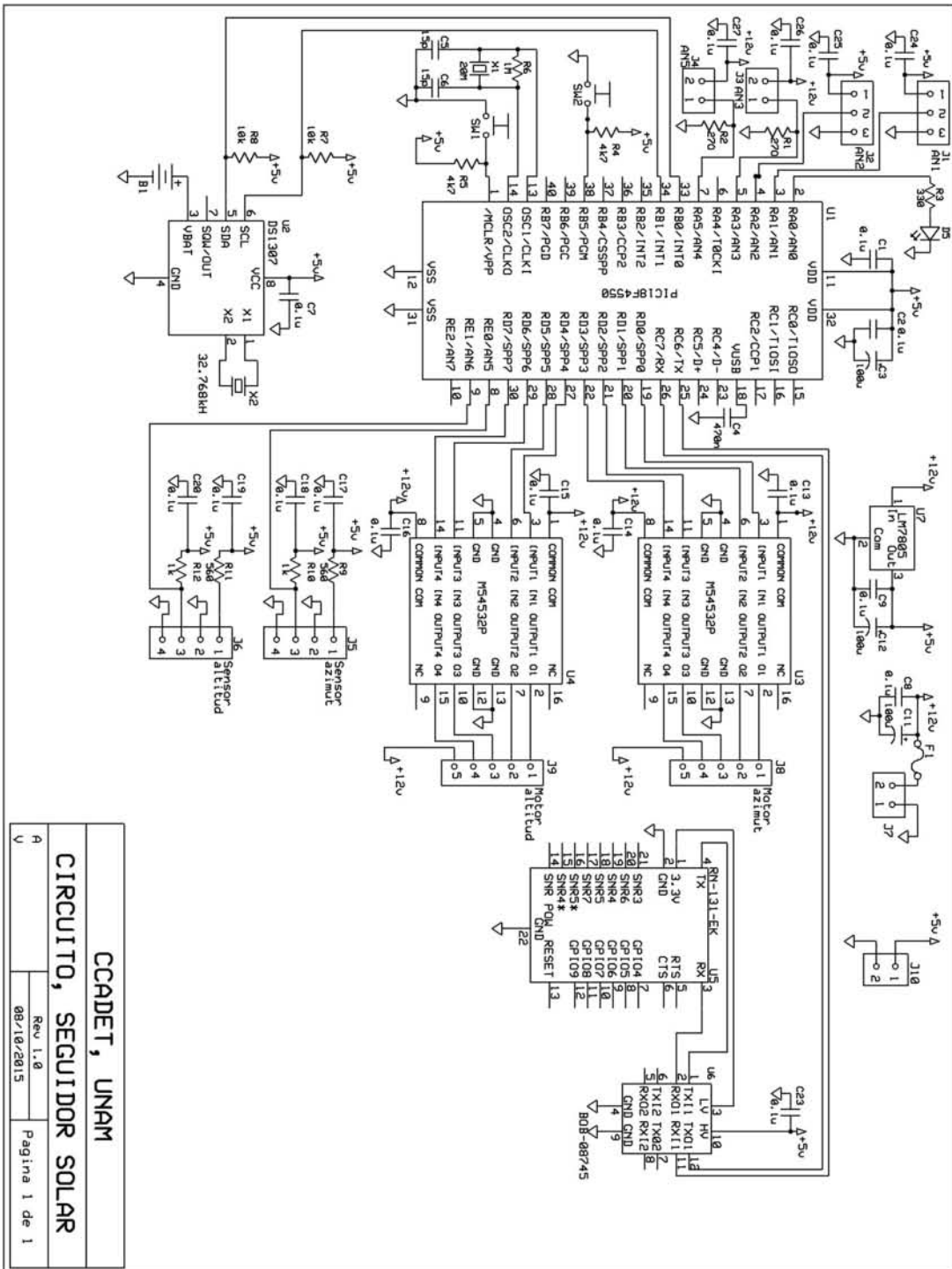
//Trayectoria solar
#declare index=0;
#while (index<6.28)
  sphere{<sqrt((r*r)-(MesP*MesP))*cos(index), 0, -sqrt((r*r)-
(MesP*MesP))*sin(index)>, 0.2 translate <0,MesP, 0> rotate <Rx, 0,
0> texture {pigment { color Yellow }} no_shadow }
  #declare index=index+0.05;
#end

```

## C. Diagramas electrónicos

Diagramas del subsistema electrónico.

### C.1. Diagrama esquemático



<b>CCADET, UNAM</b>	
<b>CIRCUITO, SEGUIDOR SOLAR</b>	
A	Rev 1.0
V	08/10/2015
Página 1 de 1	

Figura C.1. Dibujo esquemático.

## C.2. Circuito impreso

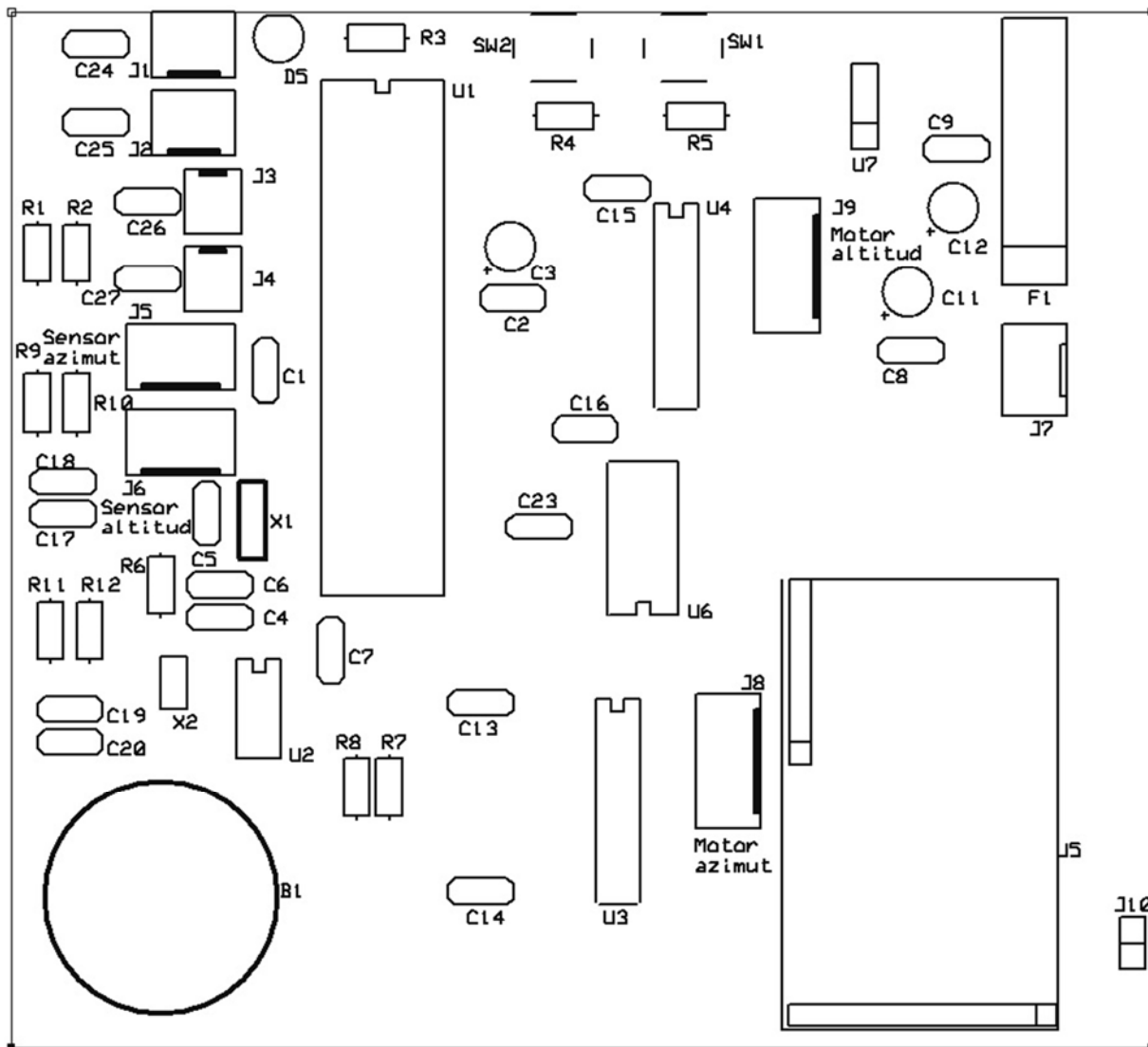


Figura C.2. Circuito impreso (mascarilla).

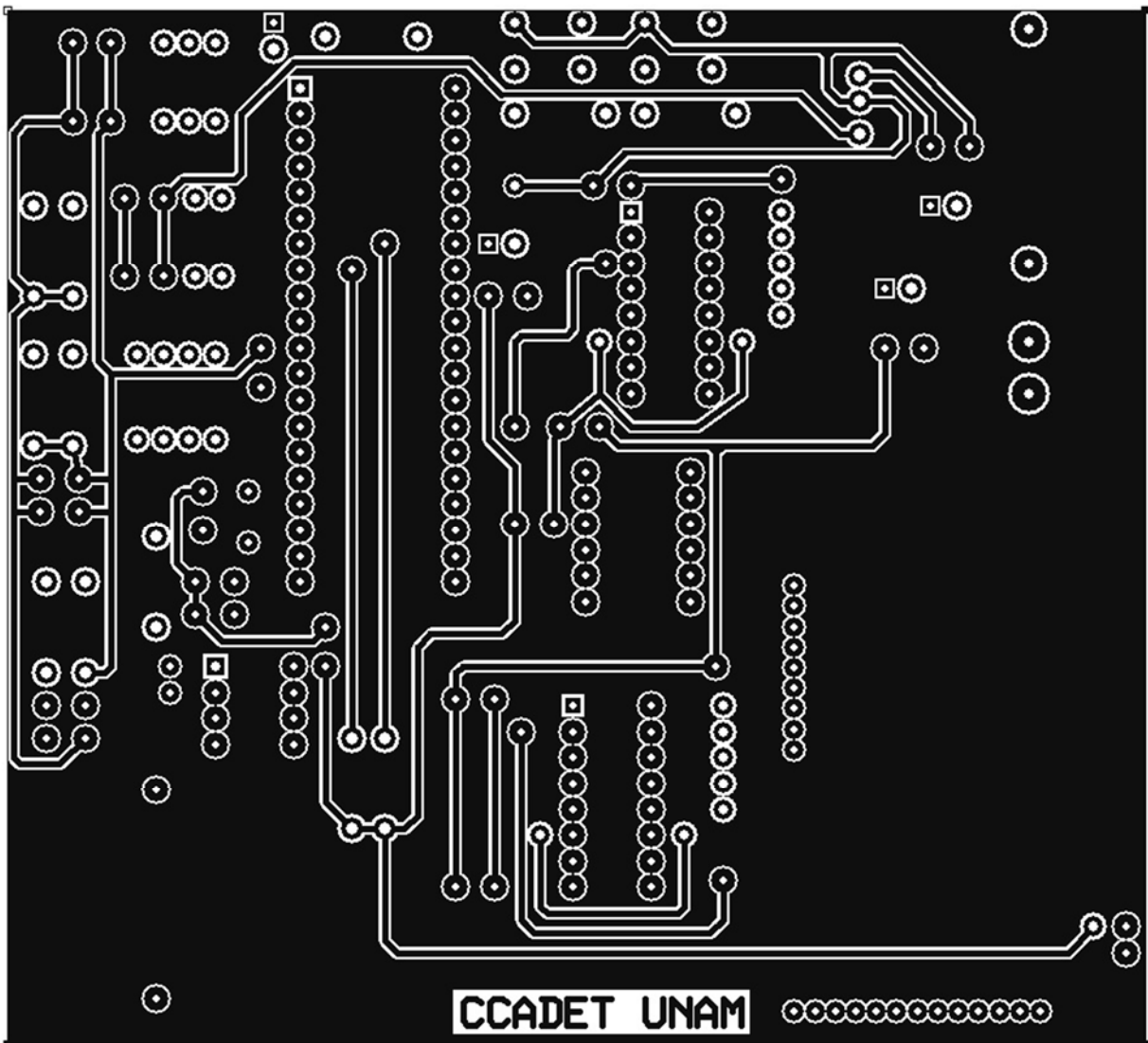


Figura C.3. Circuito impreso (componentes).

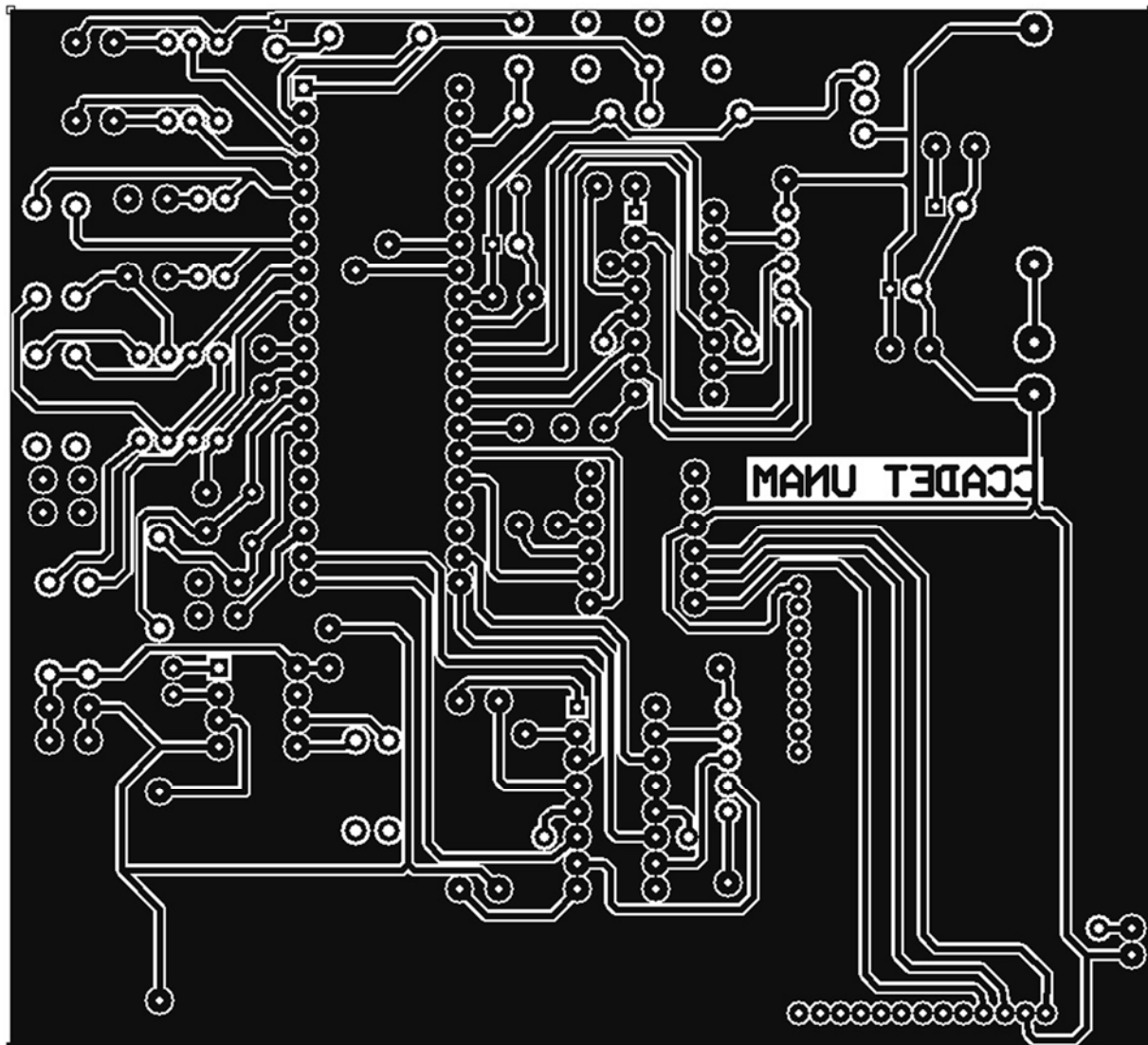


Figura C.4. Circuito impreso (soldadura).

### C.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 C.1. Lista de partes.

## **D. Planos de construcción**

Planos de construcción de las piezas de diseño específico.

### **D.1. Planos**

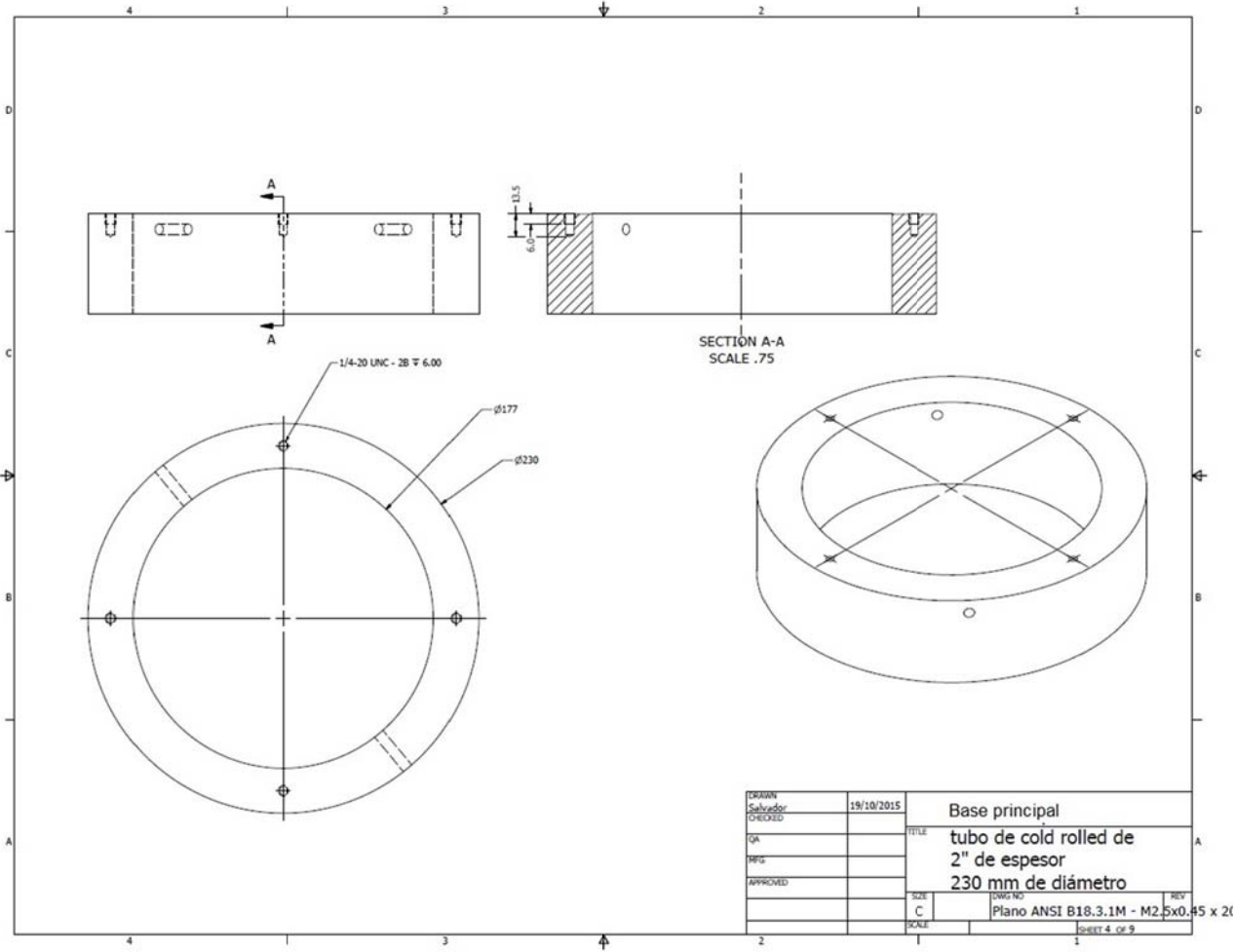


Figura D.1. Base principal.

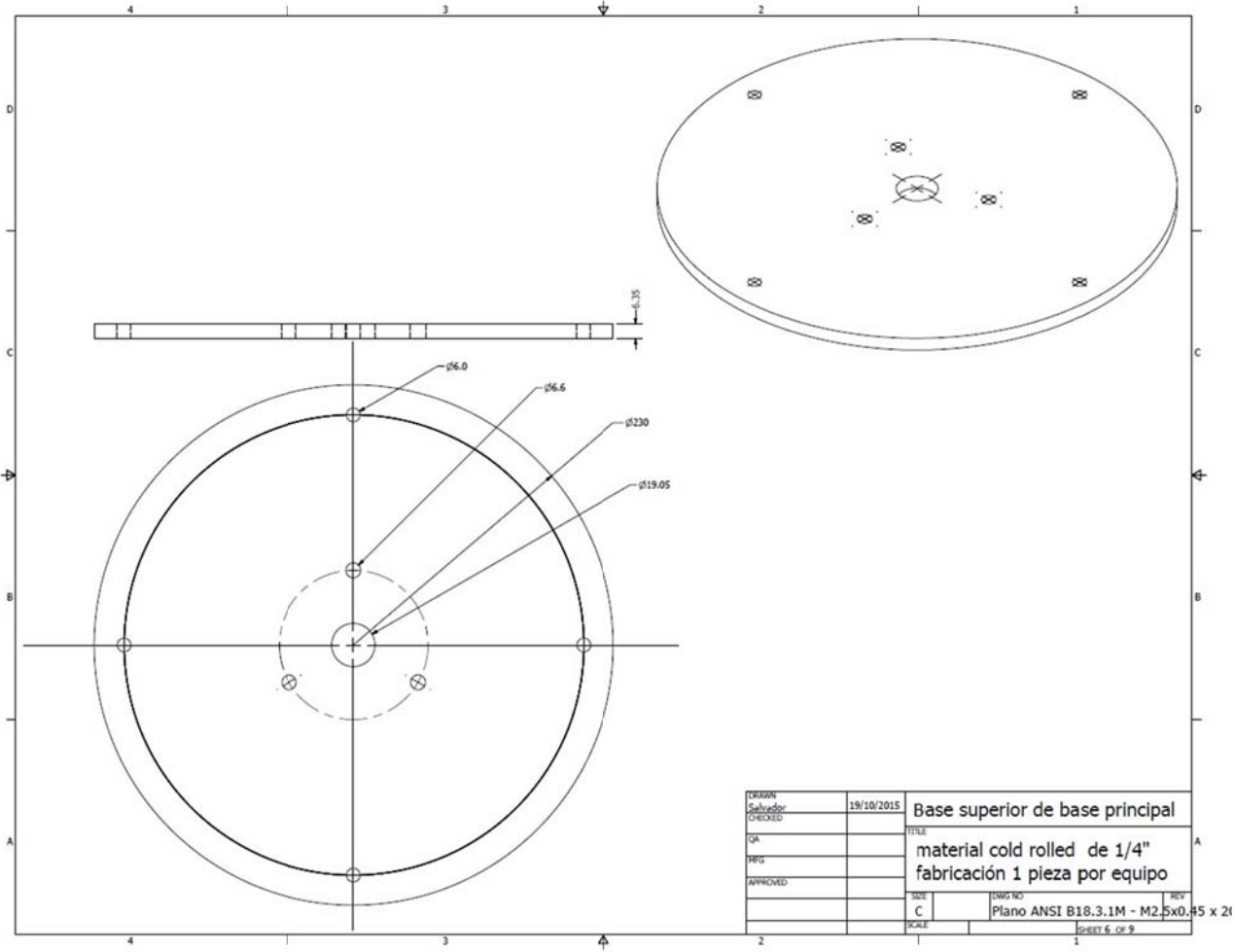


Figura D.2. Base superior de la base principal.

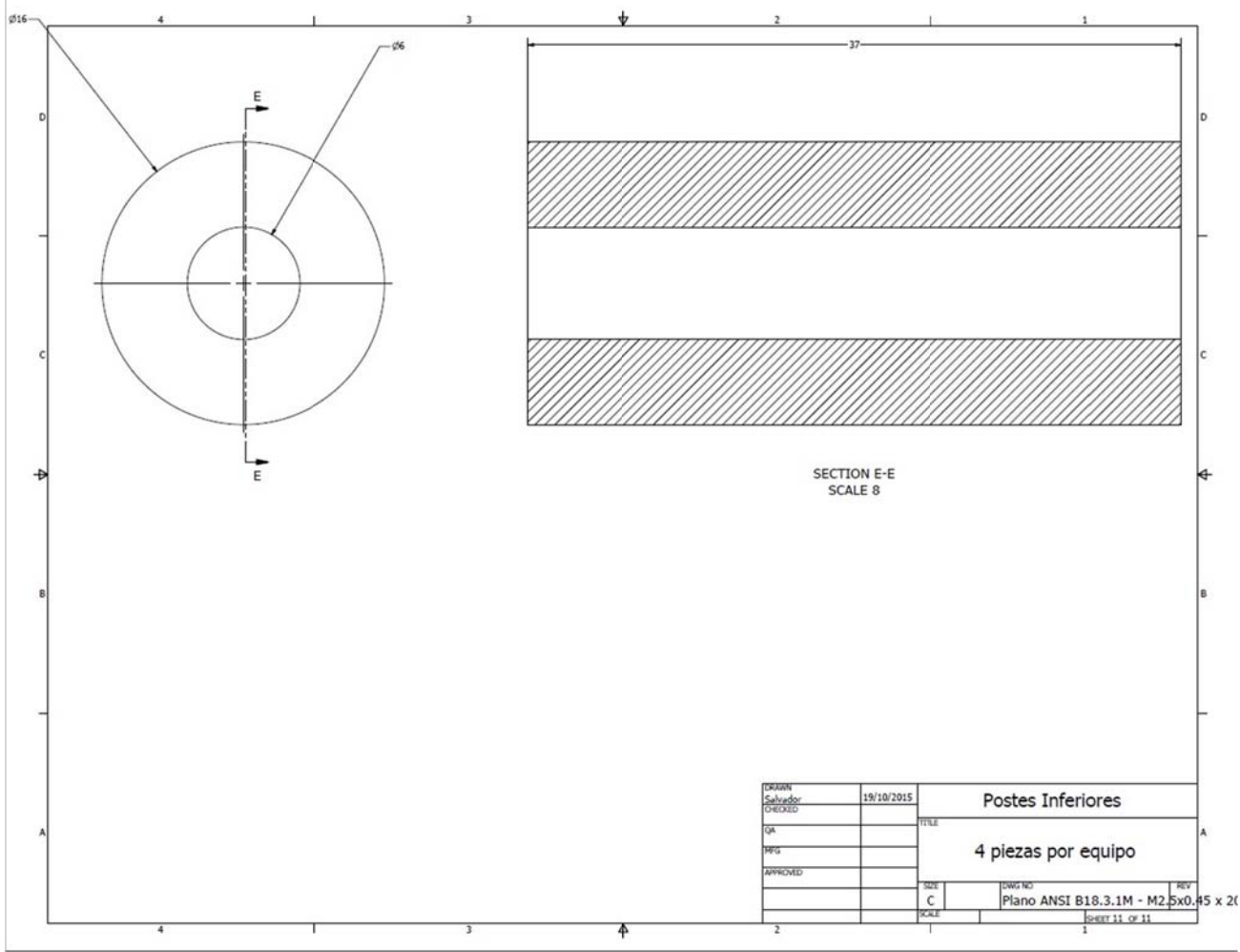


Figura D.3. Postes inferiores.

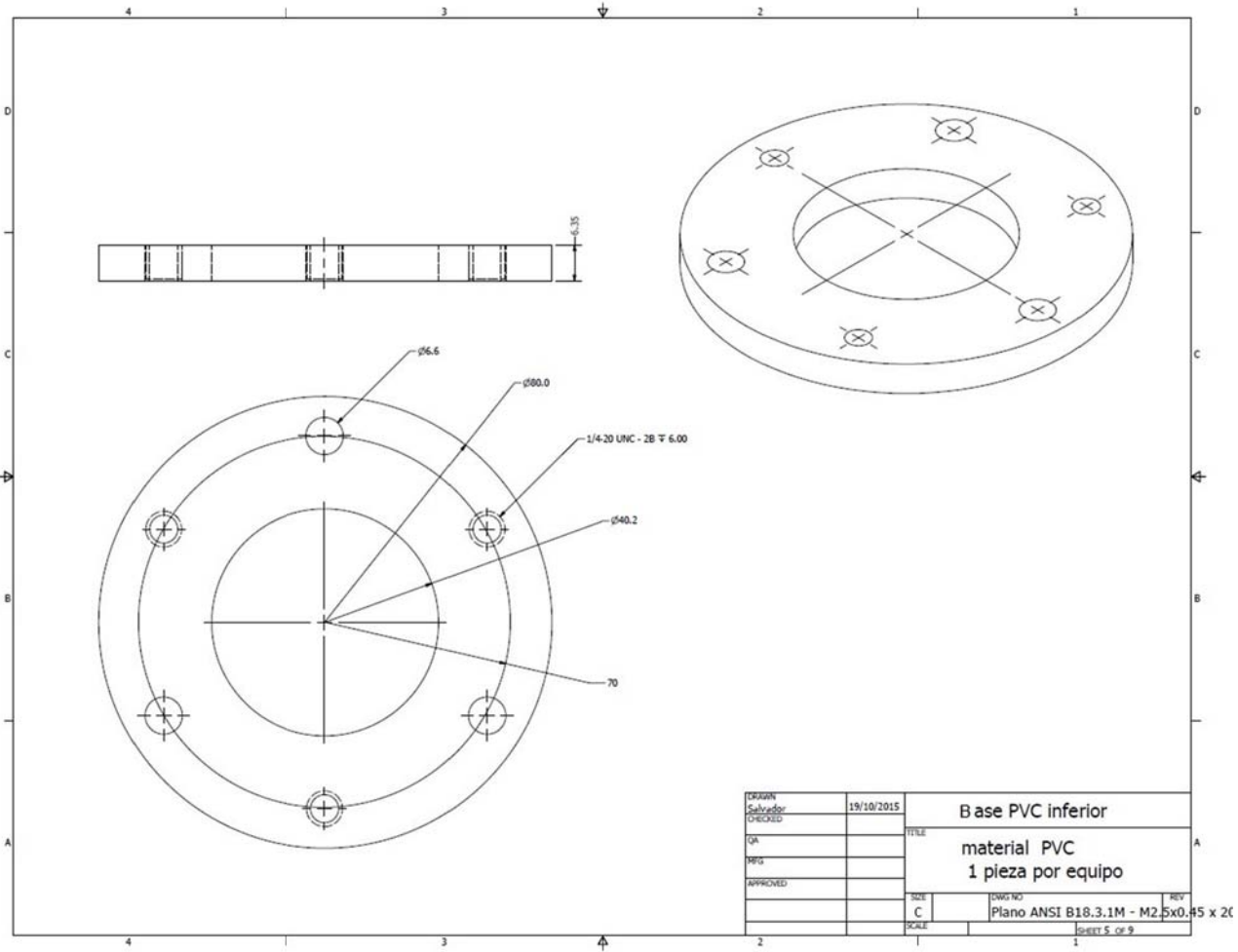


Figura D.4. Base pvc inferior.

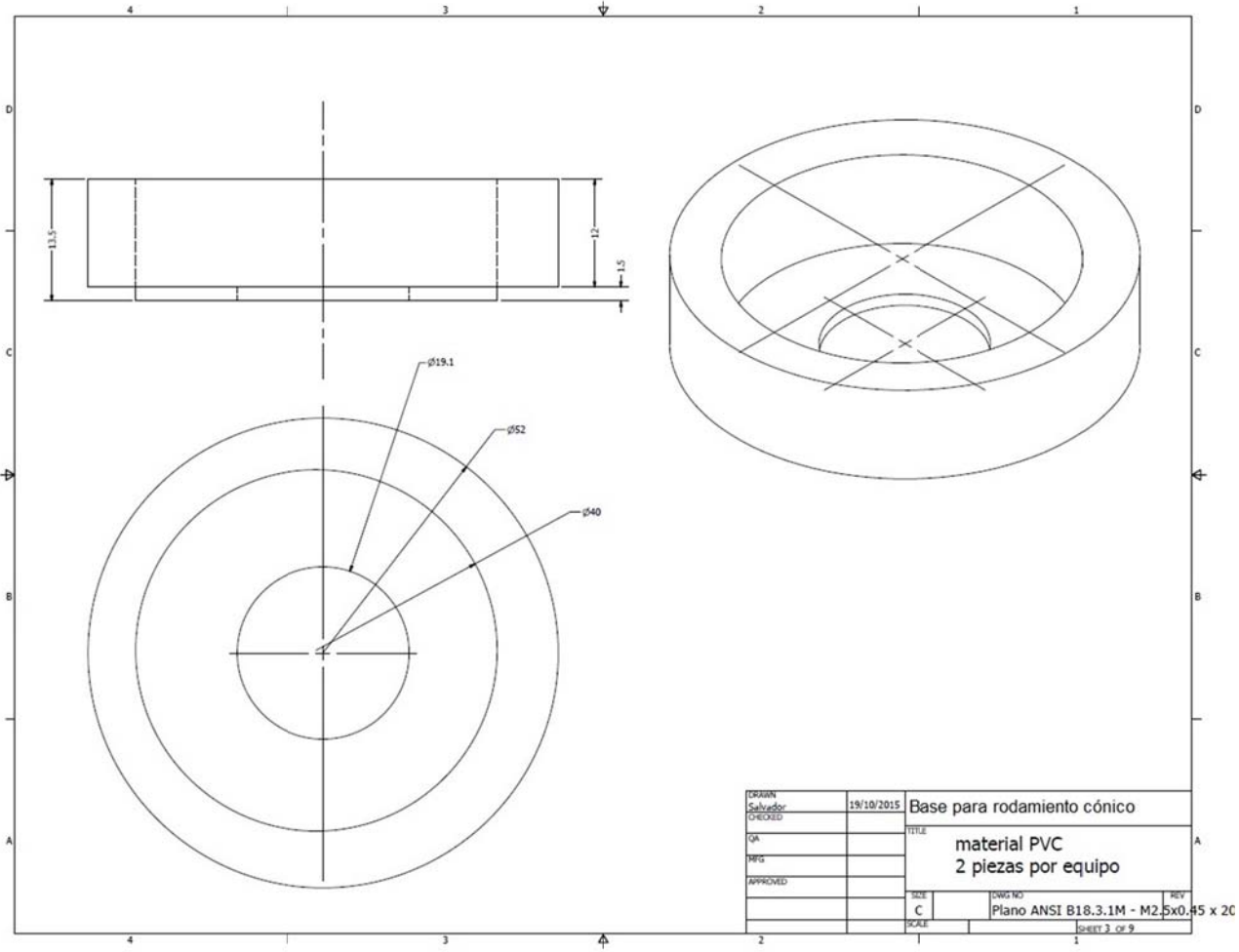


Figura D.5. Base para rodamiento cónico.

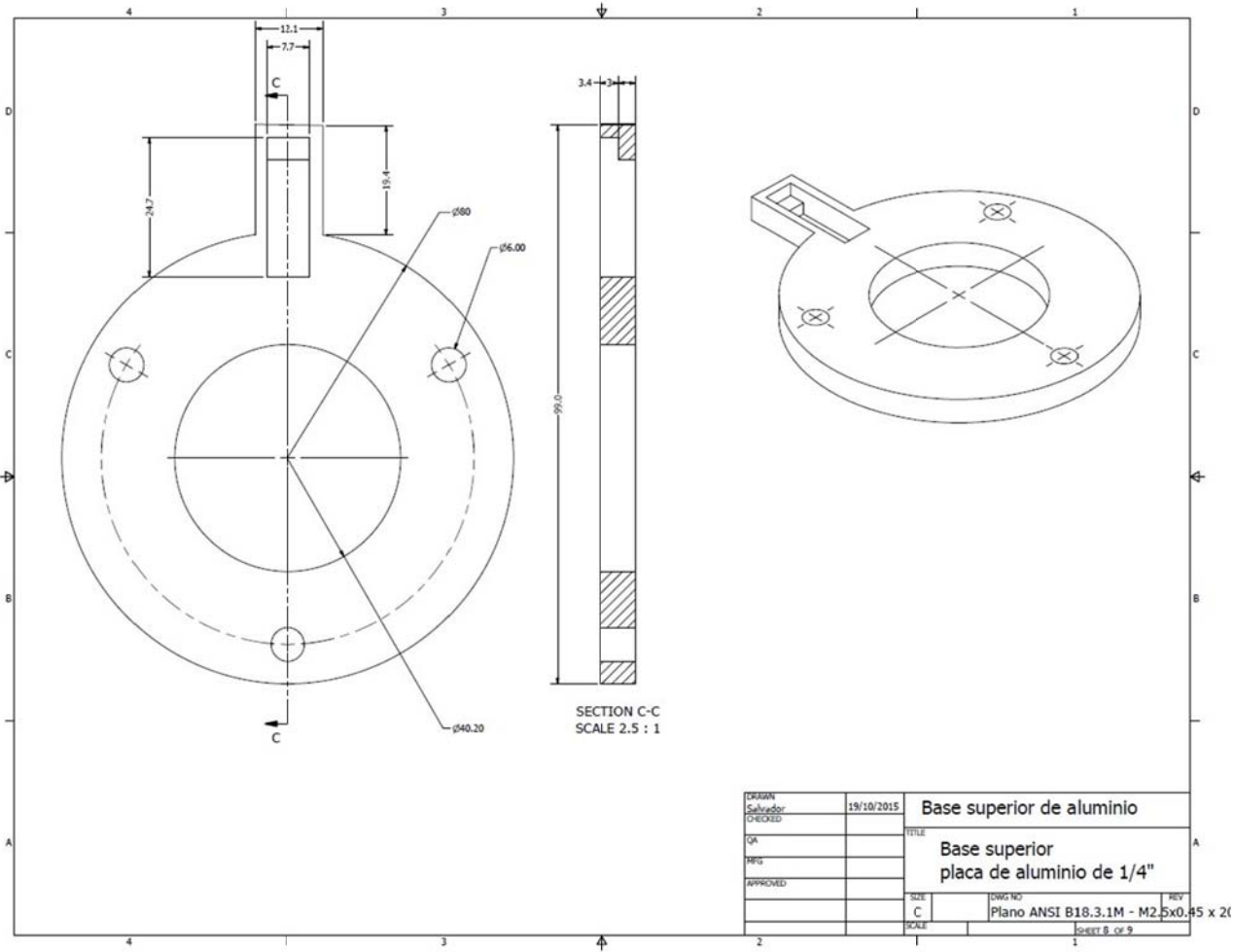


Figura D.6. Base superior de aluminio.

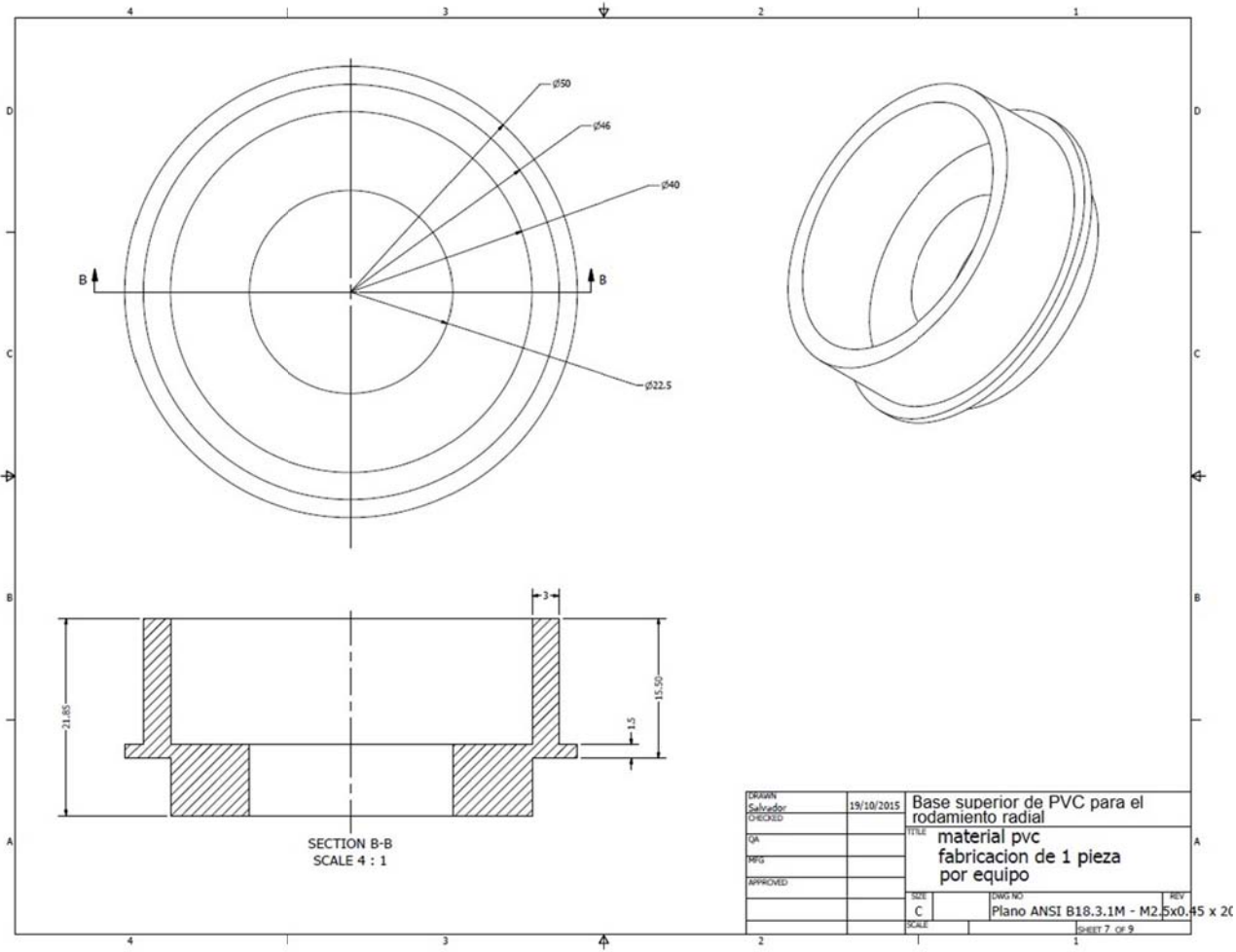


Figura D.7. Base superior de PVC para el rodamiento radial.



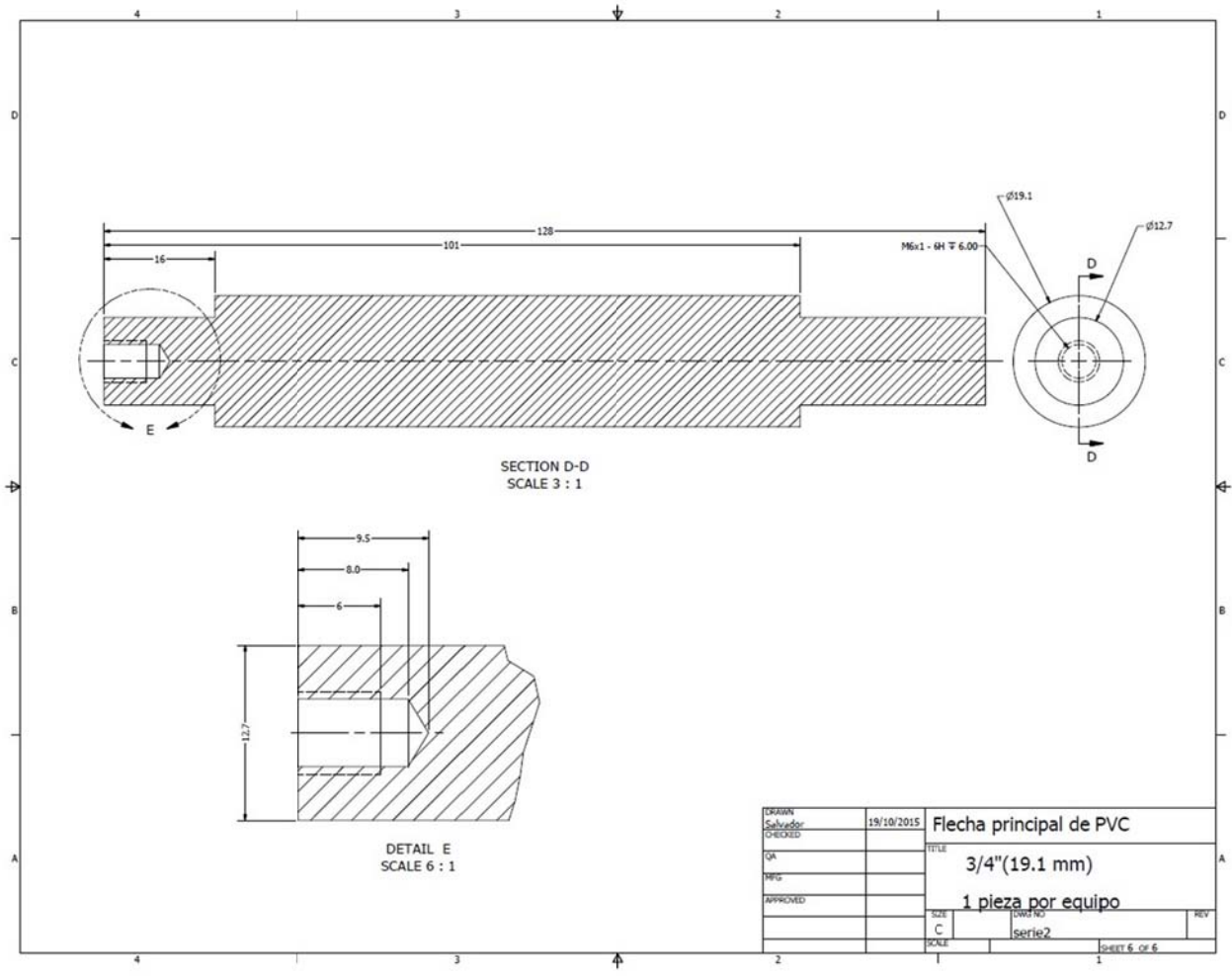


Figura D.8. Flecha principal de pvc.

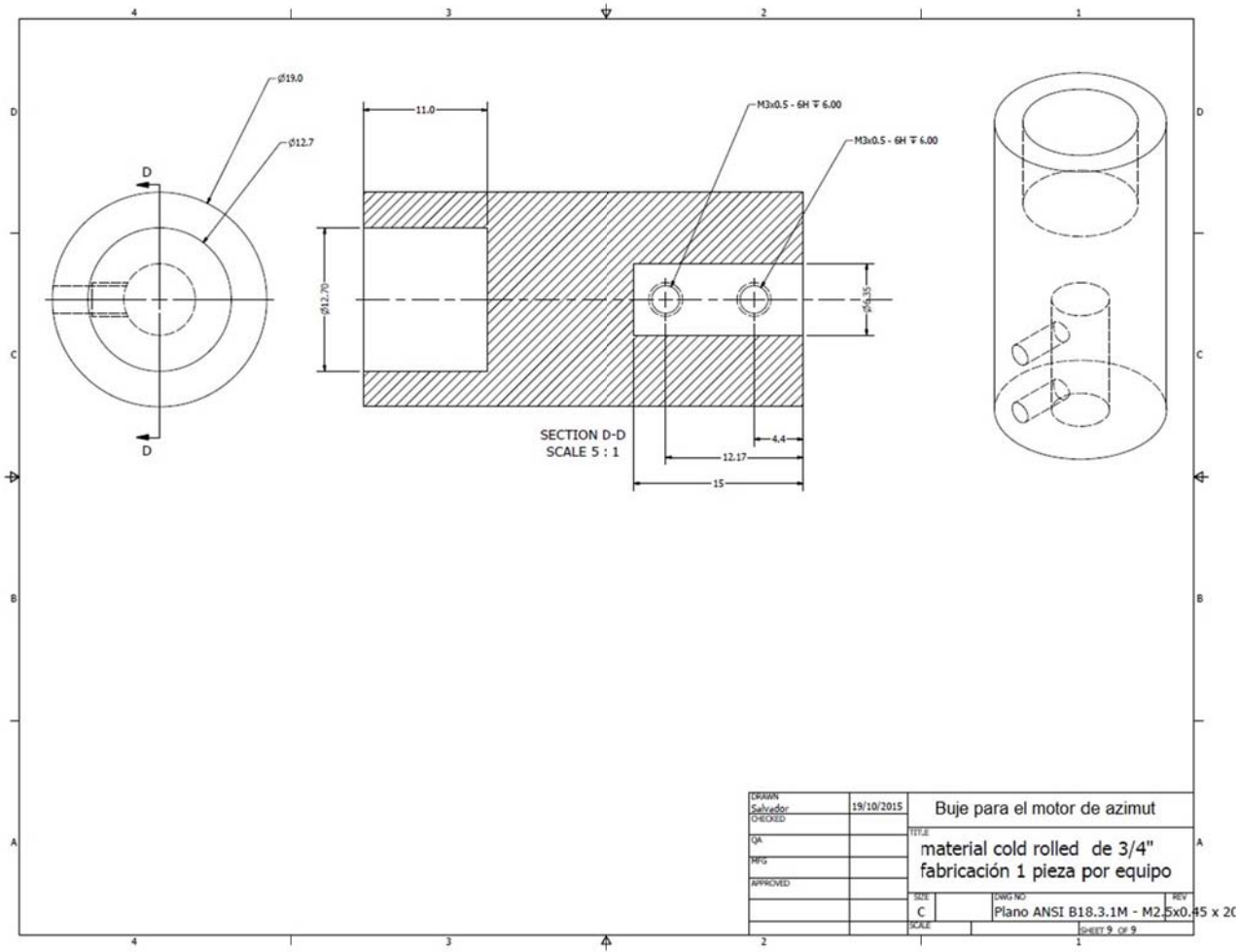


Figura D.9. Buje para el motor de azimut.

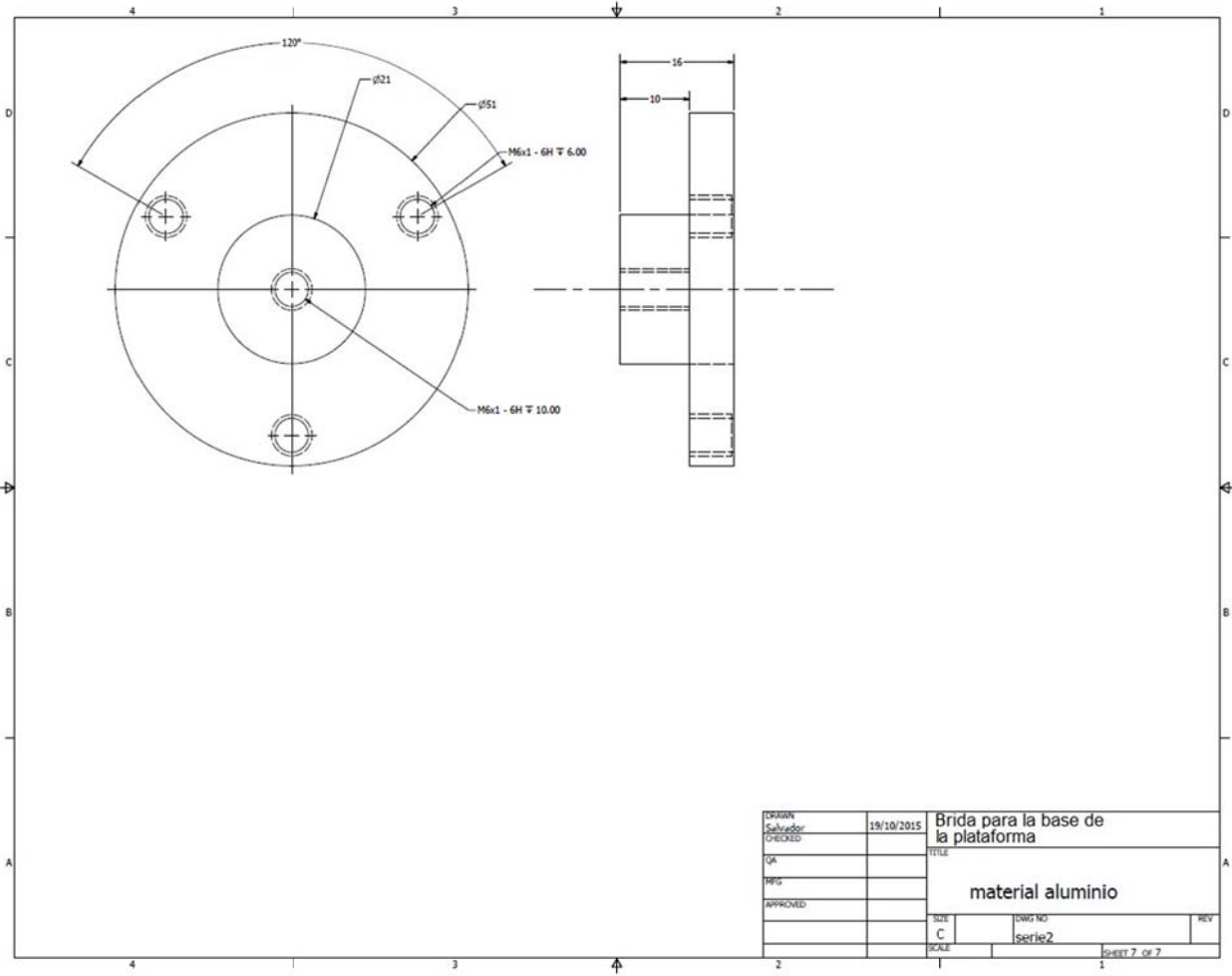


Figura D.10. Brida para la base de la plataforma.

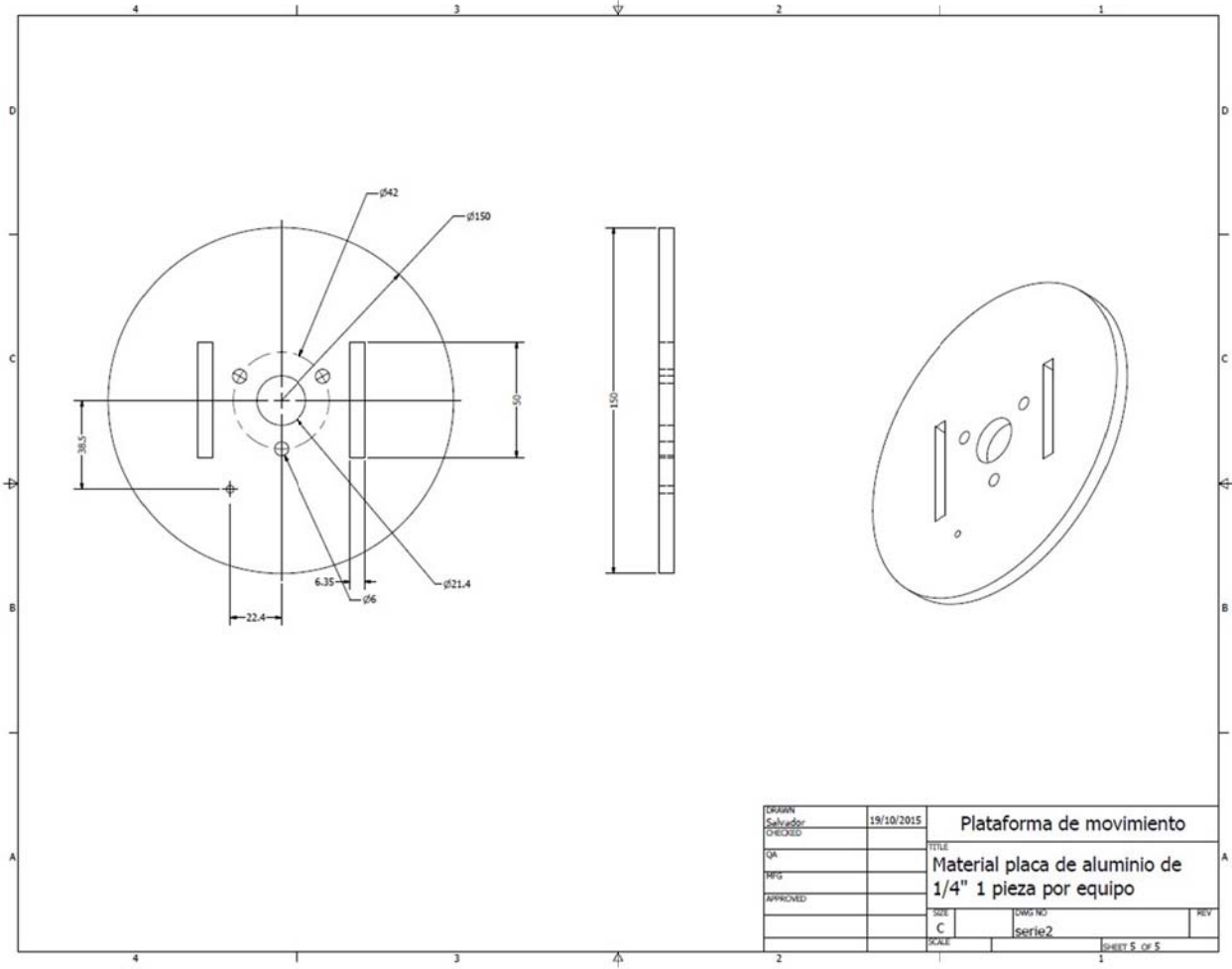


Figura D.11. Plataforma de movimiento.

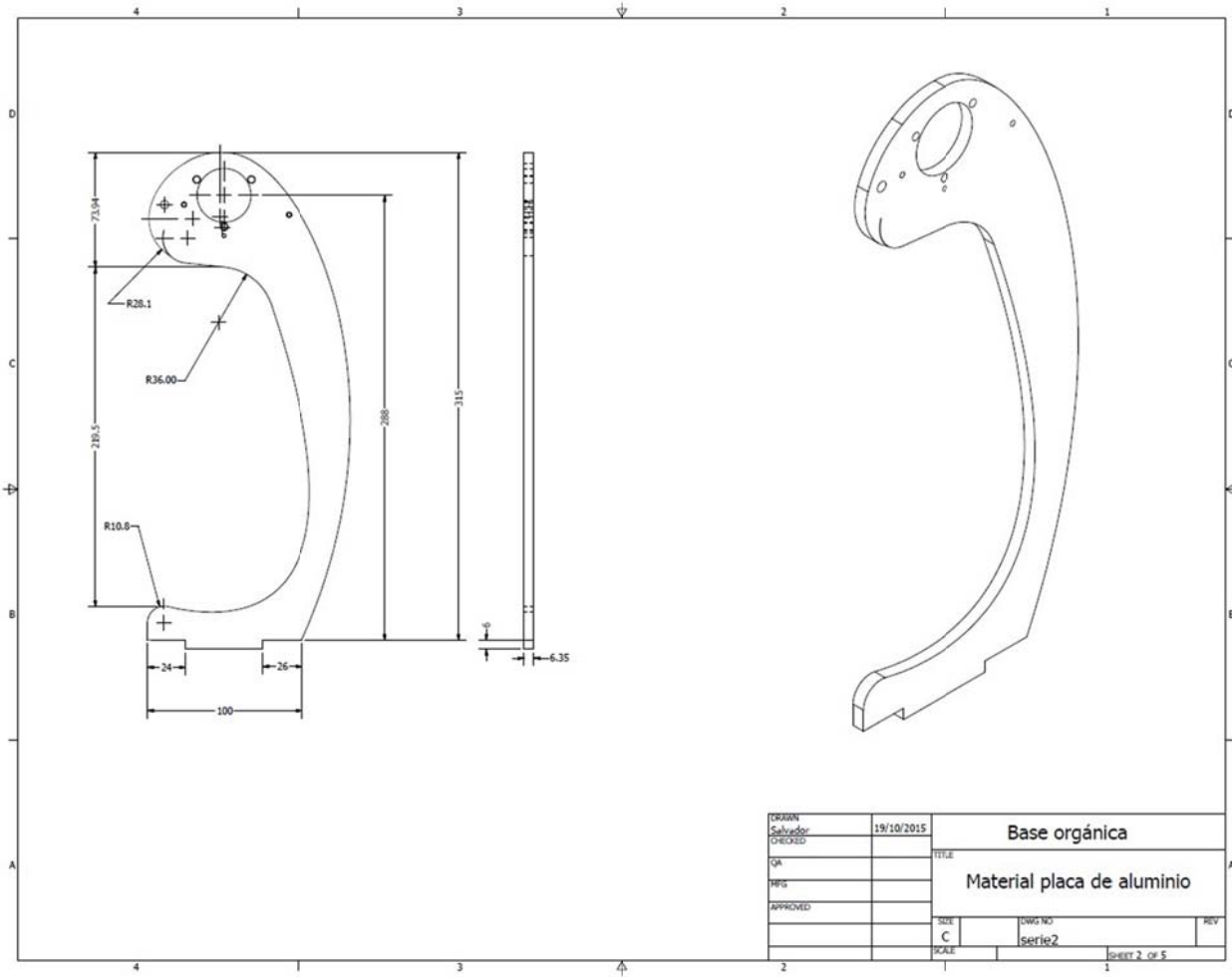


Figura D.12. Base orgánica.

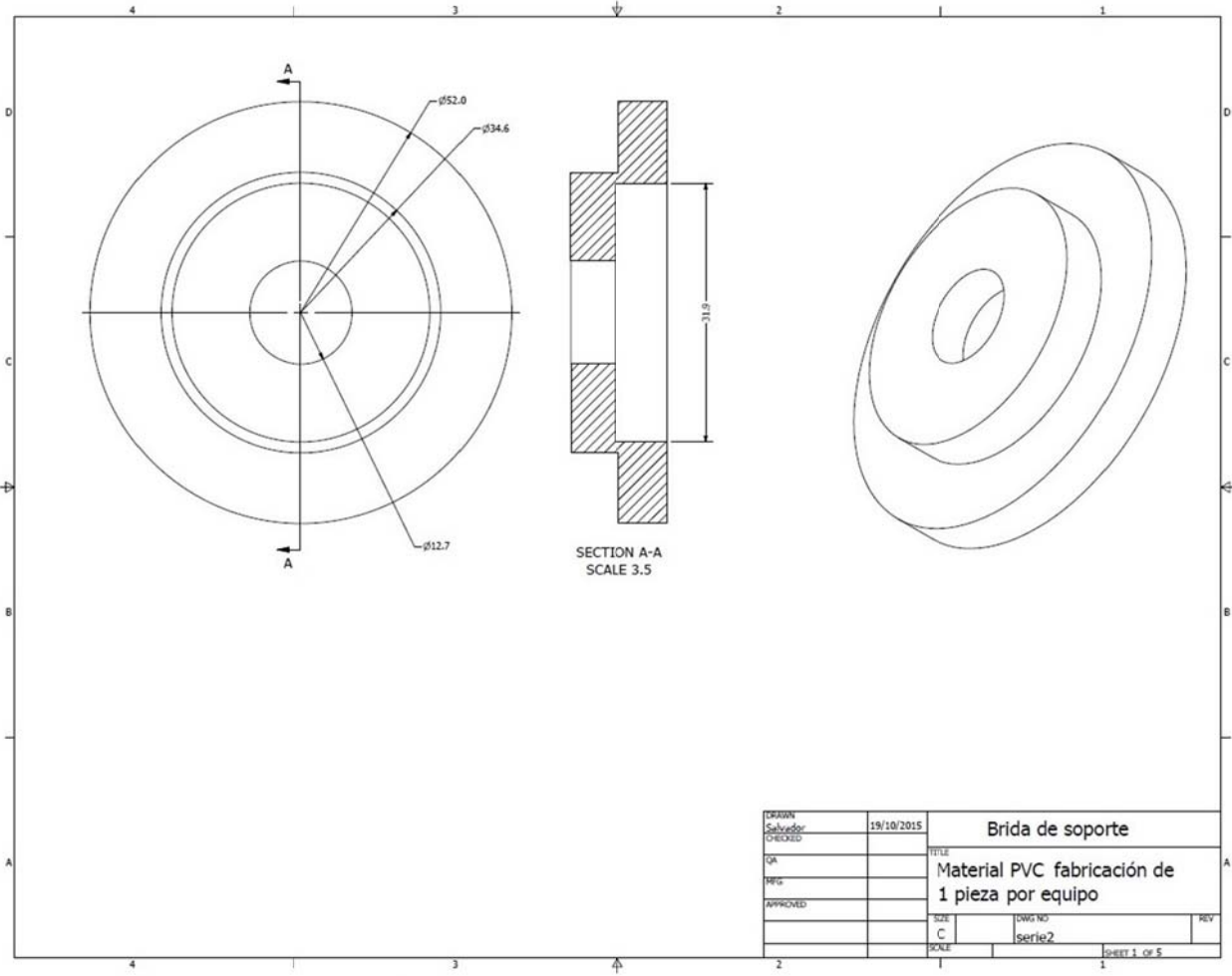


Figura D.13. Brida de soporte.

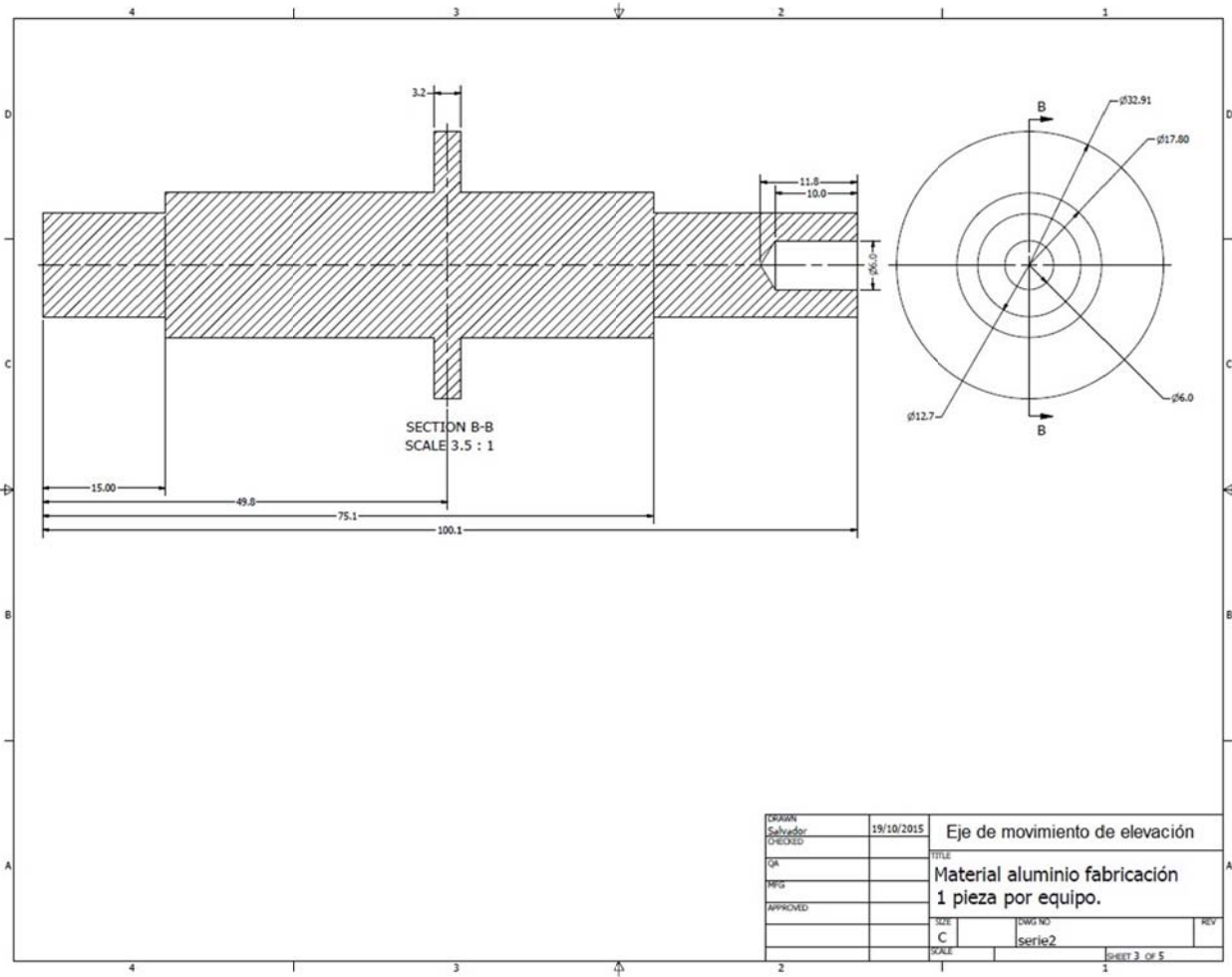


Figura D.14. Eje de movimiento de elevación.

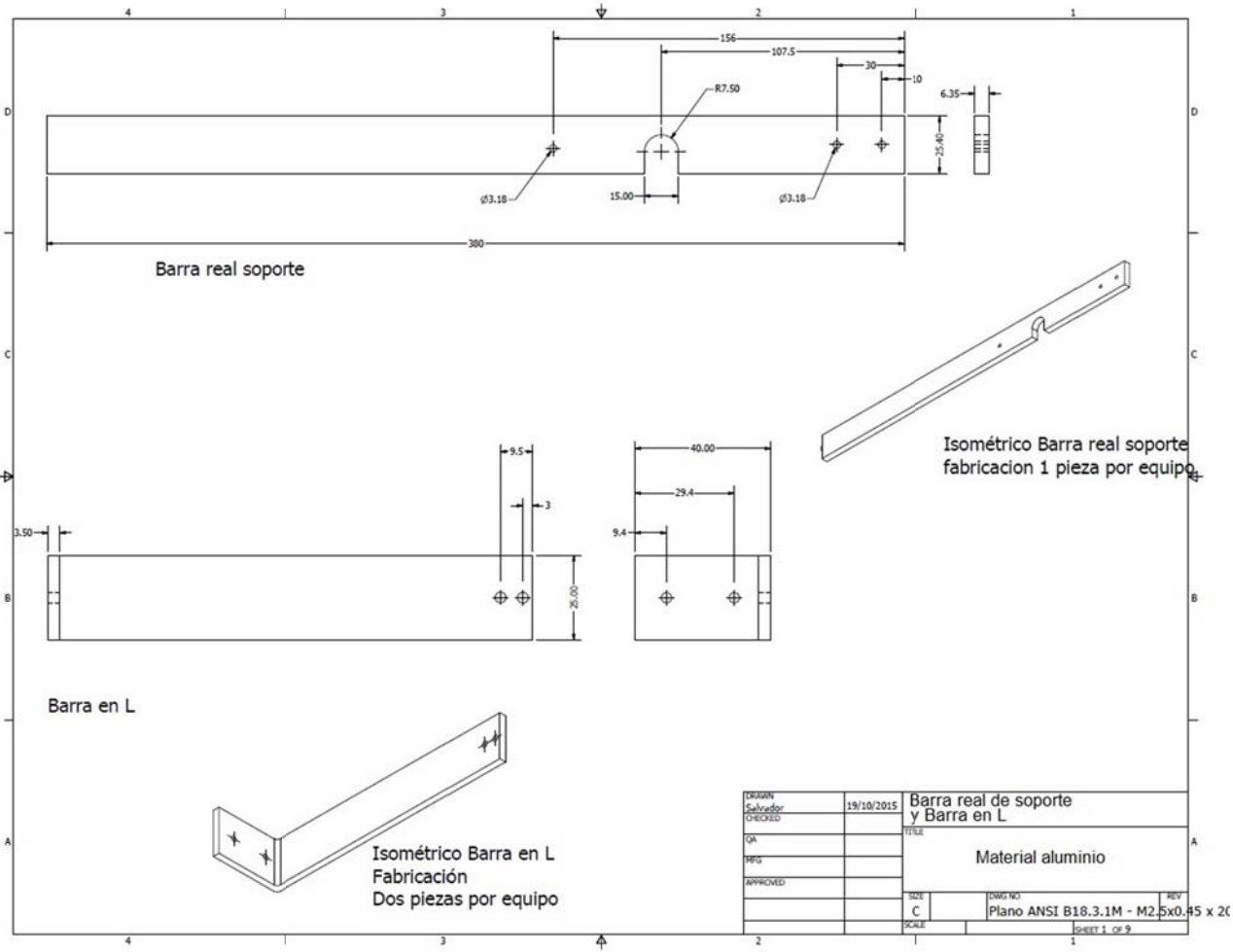


Figura D.15. Barra real de soporte y Barras en L.



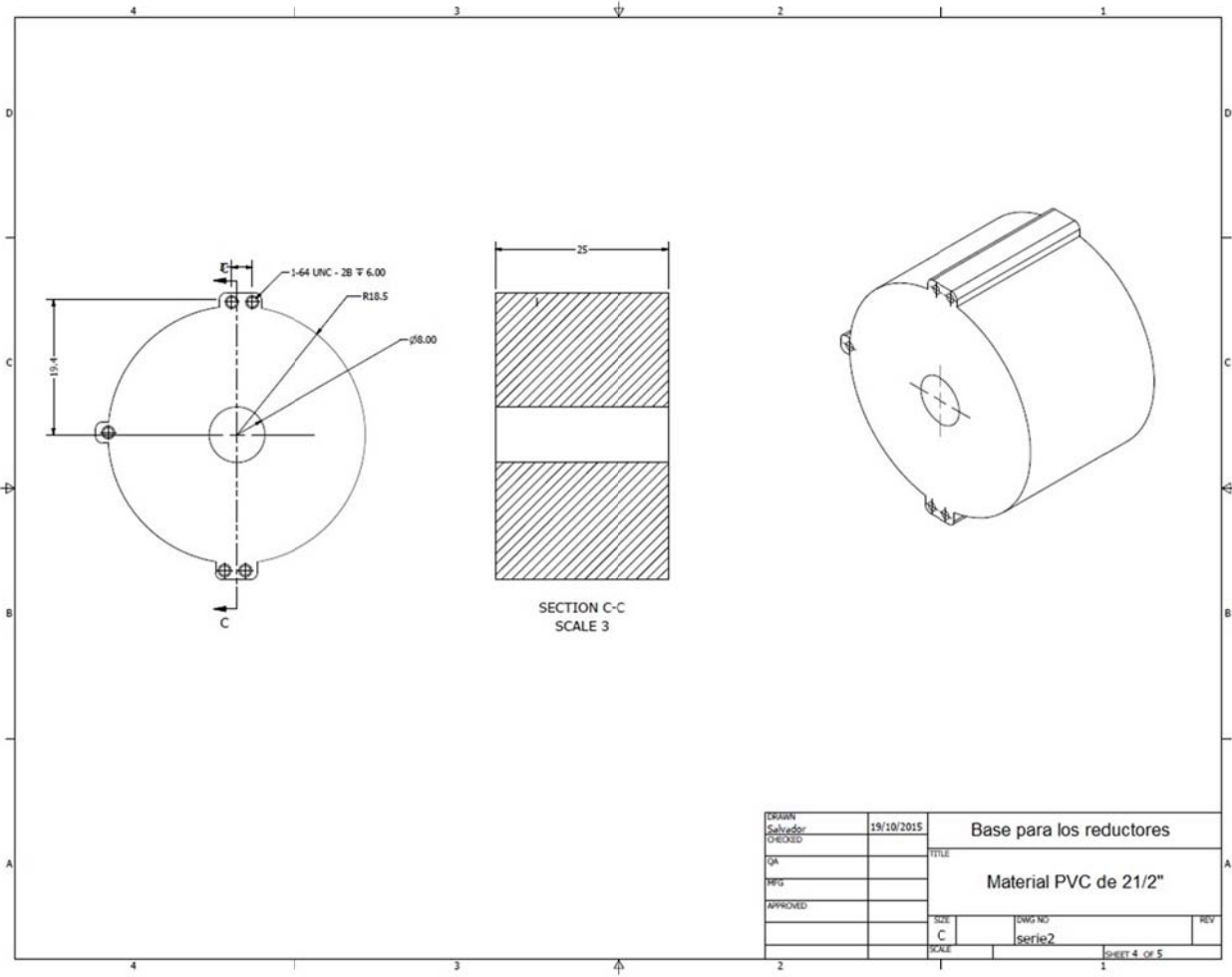


Figura D.17. Base para los reductores.

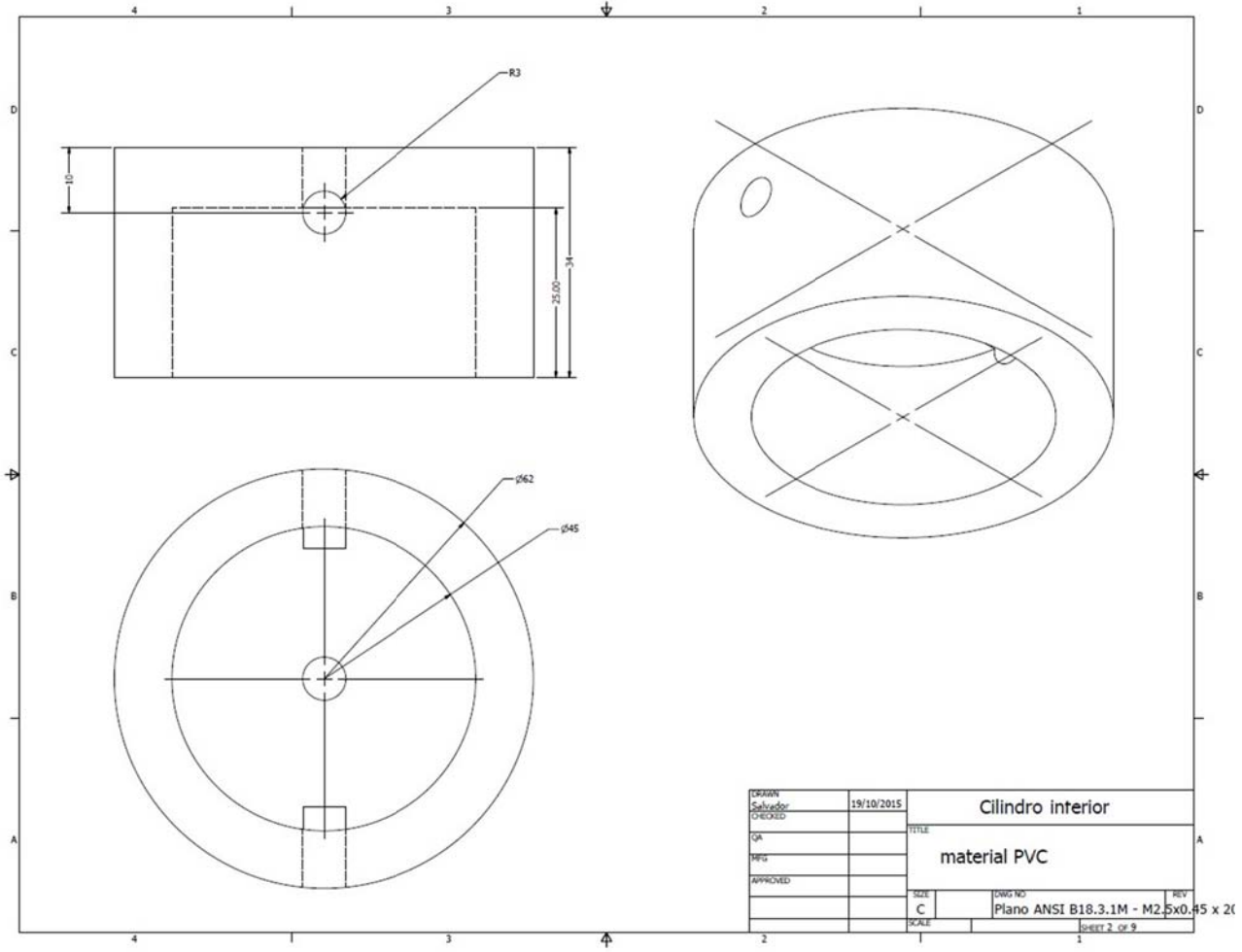


Figura D.18. Cilindro interior.

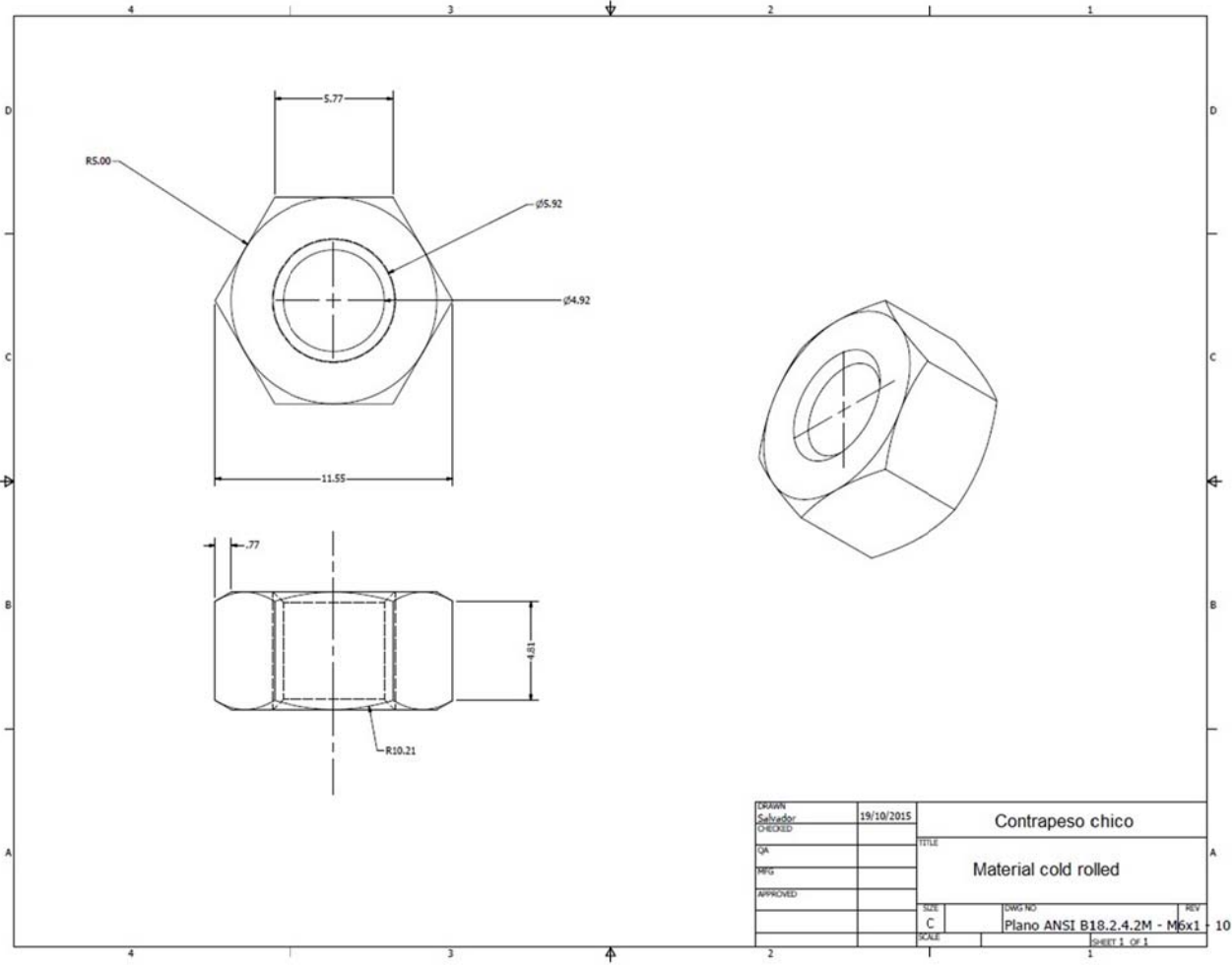


Figura D.19. Contrapeso chico.

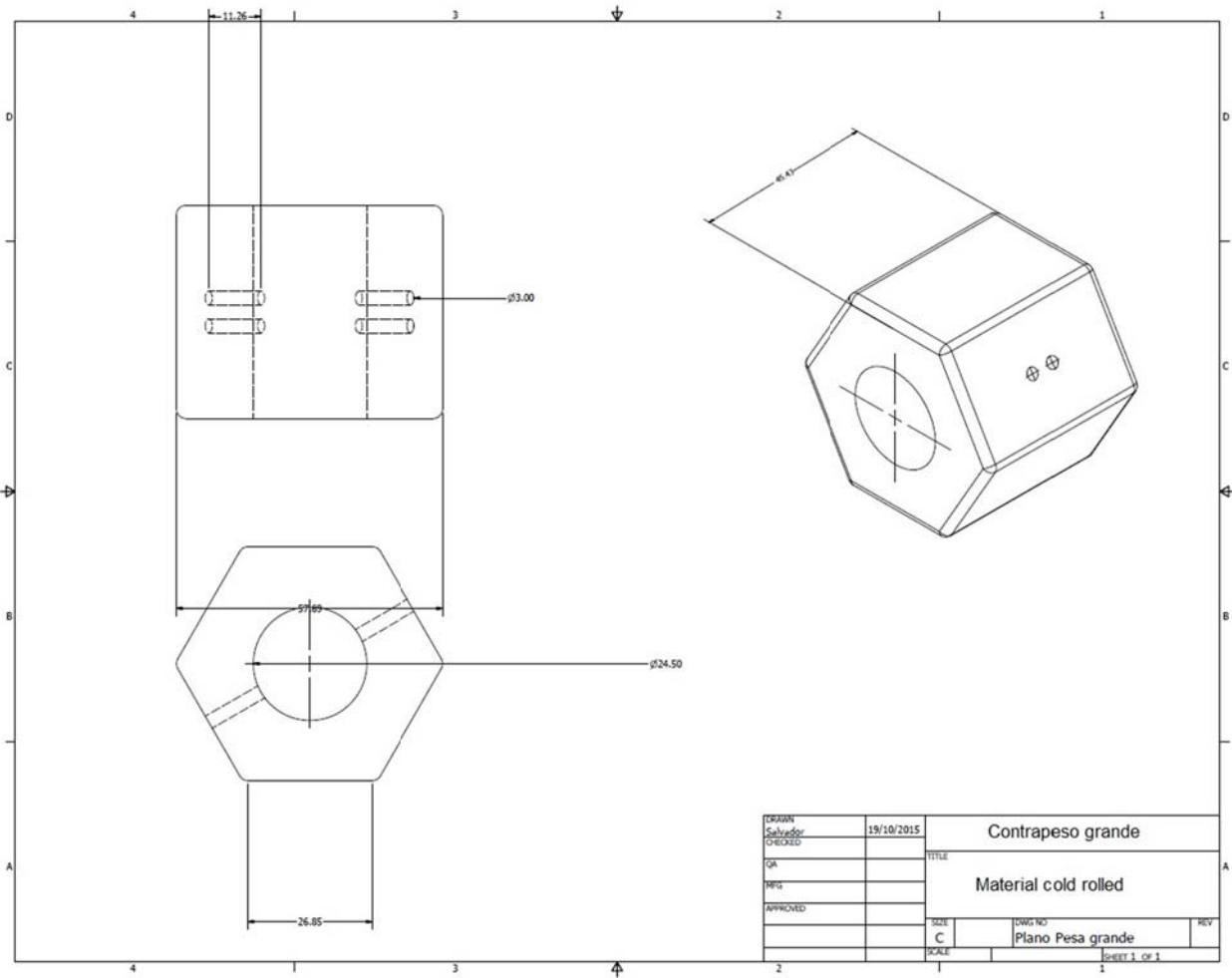


Figura D.20. Contrapeso grande.

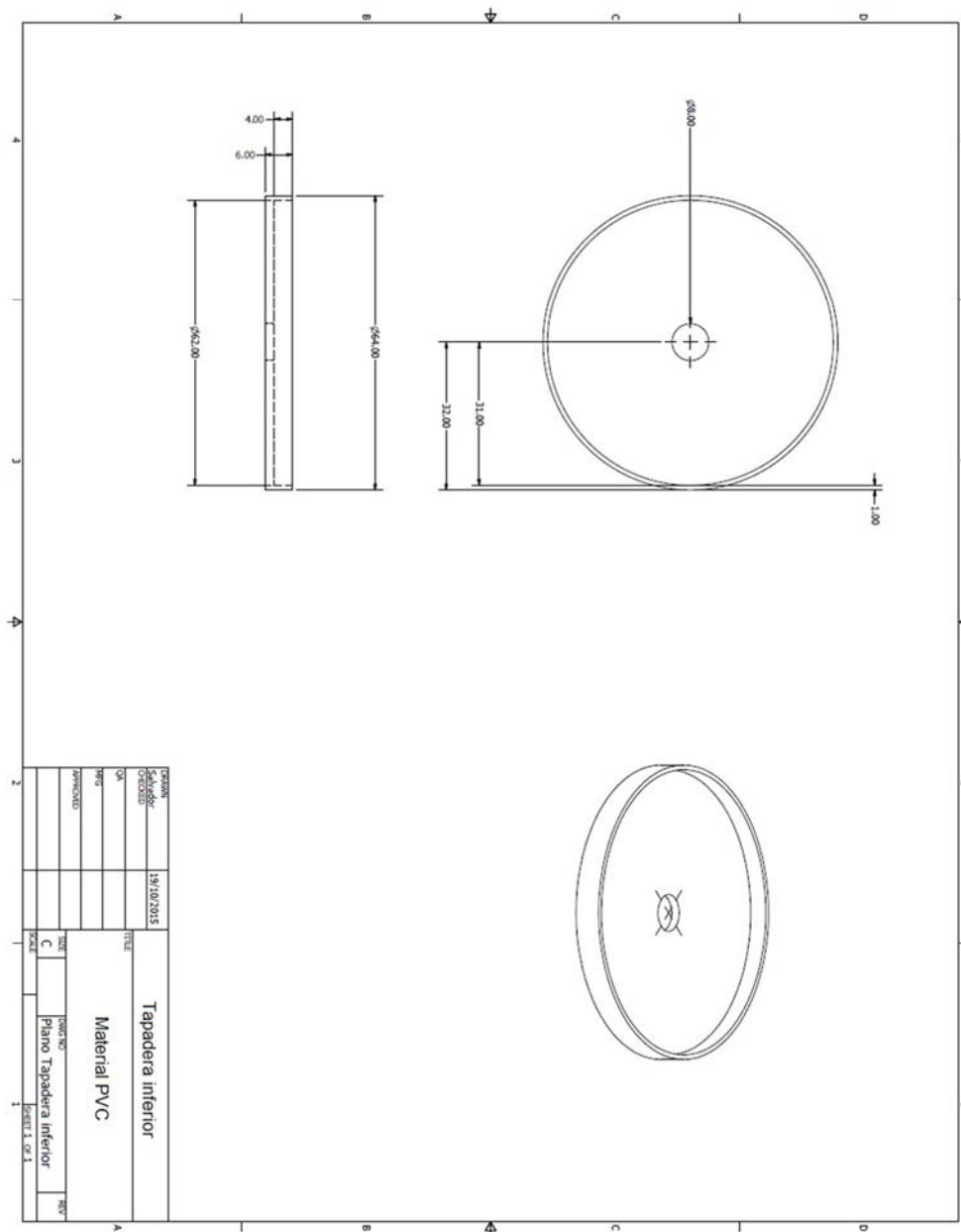


Figura D.21. Tapadera inferior.

## D.2. Lista de material

Id	Cantidad	Descripción	Material
1	1	Base principal	Material tubo de cold rolled
2	1	Base superior de la base principal	Placa de 1/4" cold rolled
3	4	Postes inferiores	Buje de cold rolled de 3/4"
4	1	Base pvc inferior	Placa de pvc de 1/4"
5	1	Base para rodamiento cónico	Barra de PVC de 2" de diametro
6	2	Rodamiento cónico comercial	DIN 720 - 30203 - 17 x 40 x 13.25

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 D.1. Lista de material subsistema mecánico.