



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Desarrollo de una plataforma mecánica
de posicionamiento de antenas para el
seguimiento automático de satélites*

Grado en Ingeniería Mecánica

ALUMNO: Álvaro Liaño Cuquerella

DIRECTORES: José María Núñez Ortuño

CURSO ACADÉMICO: 2020-2021

UniversidadeVigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Desarrollo de una plataforma mecánica
de posicionamiento de antenas para el
seguimiento automático de satélites*

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Cuerpo General

UniversidadeVigo

RESUMEN

Desde que el hombre se lanzó a la conquista del espacio, el número de satélites no ha parado de crecer debido a su gran utilidad en aplicaciones tales como el geo-posicionamiento, vigilancia y predicción meteorológica, comunicaciones de todo tipo, aplicaciones militares, científicas y una infinidad más de usos diversos. Es más, su crecimiento es exponencial, y expertos en la materia aseguran que su número puede quintuplicarse, tan solo en la próxima década.

La proliferación de estos ingenios obliga cada vez más a contar con sistemas capaces de predecir sus orbitas y que proporcionen en cada momento su situación concreta, de modo que podamos realizar un seguimiento continuo de ellos. El conocimiento preciso de la situación instantánea de los satélites obedece a diferentes motivos entre los que cabe mencionar el establecimiento de ventanas temporales para la descarga la información que recopilan desde puntos concretos de la superficie terrestre, establecer una vigilancia precisa de su correcto funcionamiento, reducción de pérdidas de potencia y eficiencia derivadas del uso de grandes haces de transmisión o incluso evitar que colisionen entre ellos o contra basura espacial.

Además, debido a la complejidad que conlleva el seguimiento de tal cantidad de satélites, cerca de 5.000 según la Oficina de Naciones Unidas para Asuntos del Espacio Exterior (en inglés, UNOOSA) y sus diferentes órbitas se hace imprescindible que dichos sistemas alcancen un elevado grado de automatización y autonomía.

Así, el objetivo de este trabajo es desarrollar de una plataforma electro-mecánica de seguimiento de satélites en tiempo real que, mediante una órbita conocida, calcule los ángulos de azimut y elevación necesarios para el apuntamiento de un sistema de antenas.

PALABRAS CLAVE

Satélites, antenas, órbitas, posicionamiento, telecomunicaciones.

ABSTRACT

Since humankind began the conquest of space, the number of satellites has not stopped growing due to their great utility in applications such as geo-positioning, weather monitoring and prediction, communications of all kinds, military applications, and an infinite number of other uses.

Moreover, their growth is exponential, as many experts in the field assure that their number can increase fivefold, just in the next decade. The proliferation of these devices increasingly requires systems capable of predicting their orbits and providing their specific situation at all times, so that we can continuously monitor them.

The precise knowledge of the instantaneous satellite situation awareness is due to different reasons including the establishment of temporary windows for downloading the information they collect from specific points on the Earth's surface, the establishment of precise monitoring of its proper functioning, reducing power and efficiency losses from the use of large transmission beams or even preventing them from colliding with each other or against space debris.

What is more, the complexity of tracking such a large number of satellites, about 5,000 according to the United Nations Office for Outer Space Affairs (UNOOSA) and their different orbits makes imperative that these systems achieve a high degree of automation and autonomy.

Thus, the objective of this work is to develop an electro-mechanical real-time satellite tracking platform that, using a known orbit, calculates the azimuth and elevation angles necessary for pointing an antenna system.

KEY WORDS

Satellites, antenna, orbit, positioning, telecommunications.

AGRADECIMIENTOS

En primer lugar, me gustaría dedicarle unas palabras de agradecimiento a mi tutor, José María Núñez Ortuño, que con su continuo interés y dedicación me ayudó a resolver los problemas que encontré a lo largo de este proyecto y me motivó a hacerlo de la mejor manera posible.

En segundo lugar, dedicarle mi profundo agradecimiento a la Escuela Naval y el Centro Universitario de la Defensa, que de forma conjunta me han dado las herramientas necesarias para llevar a cabo este proyecto y han forjado mi carácter para ser la persona que soy hoy en día.

En tercer y último lugar, dedicarle mi más sincero agradecimiento a mis padres, que me han dado los valores más esenciales que me guían en la vida y han fomentado mi esfuerzo constante, y a mis hermanos, que han supuesto un constante ejemplo que me ha ayudado a mejorar, y amigos, que han sido un continuo apoyo en todo momento y un sustento en los momentos más difíciles.

CONTENIDO

Contenido	1
Índice de Figuras	4
Índice de Tablas.....	7
1 Introducción y objetivos	1
1.1 Descripción del capítulo.....	1
1.2 Antecedentes	1
1.3 Justificación	2
1.4 Objetivos	3
1.5 Estructura del trabajo	4
2 Estado del arte	5
2.1 Descripción del apartado.....	5
2.2 Satélites	5
2.2.1 Evolución histórica	5
2.2.2 Satélites.....	9
2.2.3 Comunicaciones por satélite	17
2.2.4 Satélites de comunicaciones	19
2.2.5 Descripción del sistema español de comunicaciones por satélite (SECOMSAT).....	23
2.2.6 Satélites de experimentación y de radioaficionados	25
2.3 Plataformas de seguimiento de satélites.....	27
2.3.1 Componentes de una plataforma de seguimiento	27
2.3.2 Tipos de plataformas.....	28
2.4 Programas de seguimiento de satélites.....	33
2.4.1 Aplicaciones Web	34
2.4.2 Aplicaciones para móviles y tablets	39
2.4.3 Aplicaciones de escritorio.....	41
3 Fundamentos teóricos.....	45
3.1 Descripción del apartado.....	45
3.2 Leyes de Kepler	45
3.3 Mecánica orbital.....	46
3.4 Secciones cónicas.....	47
3.5 Elementos orbitales	48
3.5.1 Perturbaciones orbitales.....	51
3.6 Coordenadas orbitales	52

3.6.1	Sistemas de coordenadas	52
3.6.2	Ángulos de visión	53
3.6.3	Punto sub-satélite.....	54
3.6.4	Cálculo del ángulo de elevación	55
3.6.5	Cálculo del ángulo de azimut	56
3.6.6	Visibilidad.....	57
3.6.7	Ángulos de cobertura.....	57
3.7	Fuentes de datos orbitales	58
3.7.1	Formato de datos orbitales de la Norad: TLE.....	58
4	Desarrollo	63
4.1	Requisitos funcionales	63
4.2	Diseño mecánico.....	64
4.2.1	Diseño conceptual mecánico	64
4.2.2	Integración de componentes mecánicos	70
4.2.3	Límites mecánicos	70
4.2.4	Dimensionamiento de componentes	72
4.2.5	Conclusión al estudio del comportamiento mecánico del sistema	73
4.3	Diseño del hardware.....	73
4.3.1	Elección del controlador. Arduino desde el punto de vista del hardware.	73
4.3.2	Control de servomotores.....	74
4.3.3	Calibración de los servomotores.....	75
4.4	Diseño del software.....	75
4.4.1	Entorno de desarrollo.....	76
4.4.2	Arduino desde el punto de vista de software	82
4.4.3	Hamlib	82
4.5	Esquema global de funcionamiento	86
4.5.1	Funcionamiento general del sistema.....	86
4.5.2	Comportamiento general del software.....	87
4.5.3	Diseño conceptual.....	87
4.6	Integración y pruebas	93
4.6.1	Pruebas unitarias	93
4.6.2	Pruebas de integración.....	96
5	Resultados	98
5.1	Descripción del apartado.....	98
5.2	Ejemplo de un pase de un satélite NOAA.....	98
6	Conclusiones	101

6.1 Descripción del apartado.....	101
6.2 Plataforma	101
6.3 Conclusión	101
7 Líneas futuras	102
7.1 Hardware.....	102
7.2 Software	102
7.3 Conclusiones	103
8 Bibliografía.....	104
Anexo I: Planos del servomotor SG12-70.....	108
Anexo II: Especificaciones del servomotor SG12-70	109
Anexo III: Planos y especificaciones de los elementos estructurales	110
Anexo IV: Especificaciones de la placa arduino uno.....	114
Anexo V: Menú.py	115
Anexo VI: Connection.py.....	117
Anexo VII: Choose_satellite.py	118
Anexo VIII: Tracker.py	120
Anexo IX: Client.py	123
Anexo X: Cargas de pruebas para tornillos. Normativa ISO	127

ÍNDICE DE FIGURAS

Figura 2-1 Sputnik 1 (Fuente: TourHistoria)	6
Figura 2-2 Telstar I (Fuente: NASA History)	7
Figura 2-3 Lanzamiento satélites por año y países (Fuente: BBC)	8
Figura 2-4 N.º satélites en órbita (Fuente: Statista [17])	9
Figura 2-5 Arquitectura general de un satélite	10
Figura 2-6 Órbita geoestacionaria (Fuente: ESA) [29]	11
Figura 2-7 Satélites LEO (Fuente: ESA)	12
Figura 2-8 Órbita Polar (Fuente: ESA)	13
Figura 2-9 Órbitas de transferencia. En amarillo, trayectoria de lanzamiento, en azul, órbita de transferencia y en rojo, órbita final (Fuente: ESA)	14
Figura 2-10 Representación de los puntos de Lagrange elaborada por la NASA (Fuente: Wikimedia Commons)	15
Figura 2-11 Enlace satelital (Fuente: Byjus)	17
Figura 2-12 Componentes de los enlaces satelitales	18
Figura 2-13 Cobertura de INMARSAT (Fuente: INMARSAT) [30]	20
Figura 2-14 Cobertura de HISPASAT (Fuente: HISPASAT) [31]	21
Figura 2-15 Cobertura del XTAR-EUR	22
Figura 2-16 Cobertura de los satélites Iridium (Fuente: Iridium)	23
Figura 2-17 Repartición de la banda de frecuencias por los satélites de SECOMSAT	24
Figura 2-18 Estación de Maspalomas (Fuente: El Mundo)	25
Figura 2-19 Satélite de radioaficionado AO-73 (Fuente: AMSAT) [5]	27
Figura 2-20 Plataformas de seguimiento de satélites [34]	28
Figura 2-21 Estación de Robledo de Chavela	28
Figura 2-22 Vehículos de SATCOM de Infantería de Marina [2]	29
Figura 2-23 Satcom On The Move [32]	30
Figura 2-24 Vehículo SATQH [33]	31
Figura 2-25 Comparativa SOTM-SATQH [2]	31
Figura 2-26 SATCOM en buques [29]	32
Figura 2-27 Fragata de la clase Álvaro de Bazán (Fuente: Armada Española) [35]	33
Figura 2-28 Presentación satelital (Fuente: CelesTrak [3])	34
Figura 2-29 Patrones de movimiento (Fuente: CelesTrak [3])	35
Figura 2-30 Presentación general (Fuente: N2YO) [4]	35
Figura 2-31 Herramienta de seguimiento de la ISS y visión de datos en streaming (Fuente: N2YO) [4]	36

Figura 2-32 Aplicación de tracking (Fuente: AMSAT) [5]	37
Figura 2-33 Red de SatNOGS (Fuente: SatNOGS) [7].....	38
Figura 2-34 Interfaz de Satellite Tracker (Fuente: Satellite Tracker) [8].....	39
Figura 2-35 Interfaz de Orbitrack (Fuente: Orbitrack) [36]	40
Figura 2-36 Presentación sobre mapamundi (Fuente: Orbitron).....	41
Figura 2-37 Presentación inicial (Fuente: GPredict) [18]	42
Figura 2-38 Lista de satélites visibles (Fuente: GPredict) [18].....	43
Figura 2-39 Cálculos orbitales (Fuente: GPredict) [18]	43
Figura 2-40 Presentación general (Fuente: SATPC32).....	44
Figura 3-1 2ª Ley de Kepler	45
Figura 3-2 Secciones cónicas	47
Figura 3-3 Órbita situada en el plano orbital.....	49
Figura 3-4 Elementos orbitales [37]	49
Figura 3-5 Coordenadas cartesianas geocéntricas.....	52
Figura 3-6 Coordenadas polares.....	53
Figura 3-7 Ángulos de visión	54
Figura 3-8 <i>Ground track</i> de la ISS (Fuente: Wikimedia Commons)	54
Figura 3-9 Esquema para el cálculo del ángulo de elevación [38].....	55
Figura 3-10 Triángulo esférico.....	56
Figura 3-11 Ejemplo de TLE (Fuente: CelesTrak) [3].....	60
Figura 3-12 Movimiento medio [24].....	61
Figura 3-13 Representación de la anomalía media (Fuente: WikiMedia Commons)	62
Figura 4-1 Trípode empleado en el proyecto	64
Figura 4-2 Servomotor SG12-70 [16]	65
Figura 4-3 Antena Maldol hs-fox727	66
Figura 4-4 Diseño CAD de la antena	66
Figura 4-5 Placa Arduino Uno (Fuente: Arduino) [12].....	67
Figura 4-6 Conjunto de agarres y sujeciones del proyecto	67
Figura 4-7 Fotografía del soporte a 90° en el montaje	68
Figura 4-8 Fotografía del soporte en U, el soporte en C y la fijación circular en el montaje	68
Figura 4-9 Fuente de alimentación de 6V	69
Figura 4-10 Montaje del sistema	70
Figura 4-11 Alzado del anclaje de la antena al servo.....	71
Figura 4-12 Perfil del anclado de la antena al servo	71
Figura 4-13 Ejemplo de señales PWM (Fuente: Wikimedia Commons).....	74

Figura 4-14 Descripción de elementos de la placa Arduino Uno (Fuente: Arduino) [12]	75
Figura 4-15 Logo Python (Fuente: Python.org) [26]	76
Figura 4-16 Logo de Anaconda (Fuente: Anaconda) [27]	77
Figura 4-17 Logo de Spyder (Fuente: Spyder) [11]	77
Figura 4-18 Captura de pantalla de Spyder	78
Figura 4-19 Crecimiento de los principales lenguajes de programación (Fuente: StackOverflow) [28]	79
Figura 4-20 Comparativa Python-Java.....	80
Figura 4-21 Comparativa Python- C++.....	81
Figura 4-22 Esquema de funcionamiento de Hamlib [14]	83
Figura 4-23 Programas de la librería Hamlib	84
Figura 4-24 Funcionamiento global del sistema	86
Figura 4-25 Esquema básico de funcionamiento del software	87
Figura 4-26 Diagrama de flujo del módulo Connection.py.....	89
Figura 4-27 Diagrama de flujo de Connection.py	90
Figura 4-28 Diagrama de flujo del programa principal.....	92
Figura 4-29 Captura de pantalla de Celestrak (16:43 del 6 de marzo).....	94
Figura 4-30 Captura de pantalla del programa (16:43 del 6 de marzo)	94
Figura 4-31 Captura de pantalla del fichero TLE.txt	94
Figura 4-32 Comparativa de los cálculos de Tracker.py y GPredict.....	95
Figura 4-33 Prueba de comunicación con Hamlib	95
Figura 4-34 Antena realizando un pase	97
Figura 5-1 NOAA-19 (Fuente: Lockheed Martin).....	98
Figura 5-2 Prueba de pase con los softwares GPredict y SDR-Sharp.....	99
Figura 5-3 Pase del NOAA-19	100

ÍNDICE DE TABLAS

Tabla 2-1 Resumen de los satélites según tipo de órbitas	16
Tabla 2-2 Flota de satélites HISPASAT (Fuente: HISPASAT).....	22
Tabla 2-3 Bandas de frecuencia	24
Tabla 2-4 Predicción de pases de la ISS sobre la ENM (Fuente: AMSAT) [5].....	37
Tabla 3-1 Sección cónica- Excentricidad.....	48
Tabla 3-2 Resumen para el cálculo del Azimut	57
Tabla 3-3 Línea 1 del TLE	59
Tabla 3-4 Línea 2 del TLE	60
Tabla 4-1 Valores nominales de límite elástico según la calidad del tornillo (Fuente: ingmecanica) [40]	72
Tabla 4-2 Códigos de error de rotctld	85

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Descripción del capítulo

En este primer capítulo introductorio, se pretende ofrecer una visión inicial y generalista del trabajo sobre el que se desarrolla esta memoria. Para ello, empezaremos sentando las bases con un apartado de antecedentes, seguiremos con un apartado de justificación del proyecto, para continuar con los objetivos de este trabajo y finalizar con una breve explicación de la estructura de la memoria.

1.2 Antecedentes

La vertiginosa evolución sufrida por las comunicaciones por satélite en los últimos 50 años y el aumento exponencial del número de satélites en órbita han sido algunos de los actores principales en la globalización, al dotar a nuestras sociedades de: telefonía móvil, televisión en directo o Internet. Este velocísimo proceso de crecimiento, y la previsión de su rápida aceleración, ha traído consigo la, cada vez más imperiosa, necesidad de administrar y controlar el espacio en la órbita terrestre, obligándonos a desarrollar sistemas que sean capaces de realizar el seguimiento y determinación de órbitas de nuestros satélites, y el consecuente apuntamiento de antenas, de forma precisa, fiable y con elevados grados de automatización y autonomía.

En el contexto actual, se hace hincapié en la reducción de costes y una de las principales medidas adoptadas para afrontar esta problemática es la de ampliar la vida útil de los satélites. Para ello, la opción principal que tienen los controladores de las naves espaciales es la de disminuir los requisitos impuestos al satélite en cuanto a la precisión en el seguimiento de la órbita diseñada. Esta política permite que la órbita se desarrolle más allá de los límites preestablecidos, y que habían sido un requisito imprescindible hasta hace poco. Para el operador de satélites, esto ofrece una ventaja económica en términos de una vida útil prolongada del satélite, ya que se reduce el consumo de combustible a bordo. Sin embargo, para el operador de la estación terrestre, el aflojamiento de la especificación orbital da lugar a que la nave espacial describa unas órbitas menos lógicas y estables. Como consecuencia, las estaciones terrenas ahora tienen que implementar y emplear mejores y más precisos sistemas de seguimiento y apuntamiento donde antes no eran requeridos.

De manera general, se pueden agrupar los sistemas de seguimiento en dos grandes familias, los que lo realizan a través de softwares de cálculo avanzado o por controles de lazo cerrado. Estos primeros, los de software, se diseñan para que calculen el comportamiento del cuerpo orbital a fin de orientar la antena de manera precisa, por medio de desarrollos físicos-matemáticos que puedan simular con exactitud y precisión el comportamiento de los satélites, y, al comunicarse con los sistemas de posicionamiento de las antenas, ubicarlas para un correcto

establecimiento del enlace. Esto supone controles de lazo abierto, ya que los softwares no tienen capacidad de recibir señales sobre la correcta comunicación o enlace con el satélite, por lo que se basará únicamente en los cálculos de su programa. La otra gran familia de seguimiento son los sistemas que a través de diferentes mecanismos y lógicas (control de ángulo de, controladores PID¹ o lógica difusa, por citar algunos ejemplos), son capaces de percibir y manipular las señales procedentes del satélite, obteniendo la capacidad de hacer un escaneo de la señal, lo que permite desarrollar sistemas que realicen un posicionamiento de la antena hacia la zona con mayor intensidad de la señal, con la finalidad de establecer los mejores enlaces posibles.

En términos generales, se trabaja en tres ámbitos, dependiendo del tipo de órbita y los requerimientos del enlace:

- El primero consiste en el seguimiento de una de satélites en órbitas sub-síncronas. Aquí el satélite está a la vista desde cualquier punto de la superficie terrestre, durante un período limitado de tiempo y recorre un camino a través del cielo del observador en un período de tiempo determinado por los parámetros orbitales de la nave espacial. Sin alguna forma de seguimiento, incluso para una antena con un ancho de haz muy amplio, la estación terrestre pronto perdería contacto y el enlace de comunicaciones se vería gravemente afectado.
- La segunda situación se refiere al rastreo de una estación terrestre de un satélite geoestacionario. En este caso, el movimiento del satélite se limita a una ventana estrecha que no suele exceder los $\pm 30^\circ$. Para una antena con un ancho de haz del mismo orden que este movimiento es posible que no se requiera ningún tipo de seguimiento y que la antena se puede arreglar, tal vez requiriendo una corrección manual ocasional para tener en cuenta la deriva del satélite. Sin embargo, en el caso de utilizar una antena con un ancho de haz más estrecho, el seguimiento de nuevo se hace imprescindible.
- La tercera esfera, aunque no la estudiaremos ya que no es objeto de este trabajo, se refiere a los enlaces satélite a satélite. Aquí las características predominantes de la técnica de seguimiento no son sólo la precisión, sino también el peso y la fiabilidad. Esto impone nuevas limitaciones al método empleado.

Así pues, resulta evidente la imperiosa necesidad de desarrollar sistemas que no solo sean capaces de realizar un seguimiento preciso y fiable de forma autónoma, si no que además deben ser capaces de adaptarse a distintos tipos de órbitas con el fin de: reducir costes en las operaciones satelitales y disponer de sistemas de antenas que, con haces muy estrechos, puedan establecer enlaces estables con los satélites.

1.3 Justificación

En los tiempos que corren, en un mundo crecientemente globalizado, dependiente de las tecnologías satelitales para establecer sus redes de comunicaciones en todos los ámbitos de la vida cotidiana y con demanda “*in crescendo*” de más y mejores sistemas de conexión, redes y enlaces, se hace imprescindible el desarrollo de sistemas capaces de establecer conexiones de alta velocidad de transmisión de datos con satélites.

Además, para no tener grandes pérdidas de potencia y rendimiento, se exige que las antenas y satélites establezcan las conexiones con haces más y más estrechos, mejorando la eficiencia, alcance y capacidad de los enlaces. Esto provoca que el apuntamiento se vuelva crítico. Esta precisión milimétrica de seguimiento se hace especialmente necesaria en situaciones en las que

¹ Proporcional, integral y derivativo. Se refiere a los sistemas de control en los que los valores de salida son proporcionales a la cantidad y velocidad de variación de la señal actuante.

la especificación del enlace de comunicaciones exige que los niveles de señal recibida/transmitida del enlace se mantengan en valores muy bajos (esto suele ser necesario cuando el ancho de haz de la antena es del mismo orden de magnitud que el movimiento angular del satélite). La precisión de puntería requerida se determina principalmente por el ancho de haz de la antena (que es en función del diámetro de la antena y la frecuencia de funcionamiento) y las pérdidas admisibles. Otros factores que también juegan un papel protagonista, como son las condiciones de propagación, el rendimiento mecánico de los sistemas y la climatología presente, también han de ser minuciosamente estudiados ya que todos afectarán a la precisión de nuestros sistemas.

Es por esto que deberemos estudiar tanto el método por el que realizamos el seguimiento, como los elementos conformantes de nuestra plataforma, especialmente el tipo de antena, el tipo de enlace que queramos mantener y los atributos de los que queramos dotar a nuestro sistema.

En los años recientes, se han desarrollado principalmente cuatro clases de sistemas de seguimiento y apuntamiento para satisfacer las diversas necesidades de las comunicaciones por satélite, derivadas de las dos grandes familias desarrolladas en el apartado anterior. Estos pueden describirse como:

- Seguimiento mecánico a través de software de cálculo
- Detección de monopulso o multipulso
- Detección secuencial de amplitud
- Observación electrónica de haz

Estas tres últimas técnicas, se encuentran entre los sistemas de lazo cerrado o auto seguimiento. Una vez realizada la adquisición de la señal, el seguimiento continúa sin la participación del operador. Estos sistemas, dependen de la recepción de señales de error procedentes del satélite de manera continua para derivar la información y posicionar la antena de manera acorde. Aunque “*a priori*” puedan parecer como sistemas a prueba de fallos, tienen una gran cantidad de debilidades y aspectos a mejorar como la necesidad de disponer de grandes anchos de haz de antena para realizar adquisición de manera fiable, los fallos en el seguimiento derivados de las interferencias, que pueden afectar confundiendo a la antena sobre la señal deseada o polarizando y modificando los parámetros de la señal enviada por el satélite y su elevado precio. Son unos sistemas con una gran proyección pero que a día de hoy no suponen una solución precisa y fiable. Es por ello que, optaremos por desarrollar un sistema de rastreo relativo a la primera familia, la del cálculo de seguimiento y apuntamiento en base a cálculos orbitales, ya que son más baratos y emplean métodos contrastados.

Además, hay que tener en cuenta que las comunicaciones siempre han sido una parte integral en el ámbito de la guerra y un elemento vital para alcanzar el éxito en el desarrollo de las operaciones. Especialmente en el escenario geopolítico actual, con nuevas potencias emergentes amenazando zonas de soberanía e interés nacional y un mundo centrado alrededor del Océano Pacífico, debemos asegurar y garantizar a nuestras fuerzas los medios para establecer conexiones robustas y seguras. Por lo tanto, para poder ser de verdadera utilidad y ser empleable por nuestras Fuerzas Armadas, nuestro sistema debe ser, además ligero y duradero, y fácilmente transportable por las fuerzas desplegadas a imagen de las tecnologías SOTM (*Satcom On The Move*) y SATQH (*Satcom At The Quick Halt*), tecnologías de comunicaciones satelitales enfocadas a su integración en todo tipo de vehículos, tanto terrestres como barcos y aviones, de las que hablaremos más tarde en esta memoria.

1.4 Objetivos

Teniendo en cuenta todas las razones anteriormente citadas, para que nuestro sistema suponga una solución fiable a los problemas anteriormente mencionados se debe desarrollar un sistema de rastreo y seguimiento de satélites, basado en métodos de guiado por software de

cálculos orbitales, de un alto grado de precisión y fiabilidad para asegurar el establecimiento de comunicaciones y redes satelitales.

Este sistema, debe estar dotado con una antena altamente directiva para ser lo más eficiente posible con la potencia disponible y aumentar su alcance, optimizando así la capacidad del enlace y su velocidad de transmisión. Lo que, es más, para poder servir de utilidad en las FAS el sistema debe ser robusto, ligero y fácilmente transportable para poder ser desplegable en todo tipo de escenarios, por lo que debe estar compuesta de elementos ligeros pero resistentes y duraderos.

Resumiendo, se pueden establecer los siguientes objetivos para este proyecto:

- Crear un sistema de seguimiento de satélites y cálculo de órbitas mediante un programa informático de desarrollo físico-matemático, es decir, de lazo abierto.
- Desarrollar, a partir del sistema anterior, un sistema posicionador que apunte de manera automática nuestra antena en dirección al satélite del que se realiza el seguimiento.
- Equipar a nuestro sistema con una antena con un alto grado de directividad para hacerlo más eficiente y proporcionar mayores alcances.
- Diseñar nuestro sistema con elementos ligeros, duraderos y resistentes para poder ser desplegado en todo tipo de misiones y escenarios

1.5 Estructura del trabajo

En este subapartado se realiza una breve descripción del proceso de desarrollo de esta memoria por capítulos.

Comienza con un **primer capítulo** en el que se establecen los antecedentes a partir de los que surge nuestro sistema, las necesidades que tenemos y los objetivos que queremos cubrir en base a ellas.

En el **segundo capítulo**, se procederá a hacer un desarrollo del estado del arte de las tecnologías satelitales hablando en primer lugar de los propios satélites, de las plataformas de seguimiento después, para finalizar describiendo diferentes programas y aplicaciones de seguimiento de satélites.

En el **tercer capítulo** se hará una explicación breve pero detallada de la base teórica de este trabajo. Para ello, desarrollaremos las leyes de Kepler y los conceptos principales de la mecánica orbital, las secciones cónicas y los elementos orbitales clásicos, imprescindibles para realizar los cálculos, para acabar explicando los diferentes sistemas de coordenadas empleados y analizando nuestras fuentes de datos orbitales.

En el **cuarto capítulo**, en un proceso de lo más general a lo más específico, desarrollaremos a partir de los requisitos funcionales de nuestro proyecto, el diseño y dimensionamiento de los componentes mecánicos, siguiendo con el diseño del hardware y el software y su posterior integración y pruebas.

En el **quinto capítulo** expondremos los resultados obtenidos en nuestros experimentos y valoraremos el grado de cumplimiento de los objetivos que nos impusimos.

El **sexto capítulo** servirá para analizar los capítulos anteriores y realizar una síntesis sobre nuestro proyecto en general, valorando todos los aspectos de nuestro trabajo, el grado de exigencia de nuestros objetivos, el grado de cumplimiento y las características del sistema.

El **séptimo capítulo** se usará para definir los posibles futuros avances y características con las que se podrá complementar y mejorar nuestro sistema.

El **octavo capítulo** incluirá la bibliografía y referencias usadas.

Posteriormente, en los **anexos**, se muestran las especificaciones técnicas de nuestros componentes y los códigos de nuestro software.

2 ESTADO DEL ARTE

2.1 Descripción del apartado

En el presente apartado se pretende desarrollar a grandes rasgos el panorama actual de los diversos bloques que componen el proyecto; comenzando por una reseña histórica, procediendo después a hacer un breve análisis sobre los distintos programas de predicción de pases y seguimiento de satélites, tanto de licencia abierta, como de pago, así como de las distintas funcionalidades que ofrecen.

2.2 Satélites

2.2.1 Evolución histórica

2.2.1.1 La conquista del espacio: la carrera espacial

El 4 de octubre de 1957, en el marco de la Guerra Fría que opuso a Estados Unidos y a la Unión Soviética, se produjo el primer lanzamiento y puesta en órbita de un satélite artificial exitosamente, el **Sputnik**.

Este primer satélite, de unos 80 kilogramos de peso, se puso en órbita a unos 6.500 kilómetros de altura sobre el Ecuador terrestre, con un período orbital de aproximadamente una hora y media y una velocidad angular de unos 30.000 kilómetros/hora.

Su órbita elíptica tenía un apogeo² y un perigeo³ de 940 y 215 kilómetros, respectivamente. En esencia el cuerpo del satélite era una esfera de aluminio de 60 cm de diámetro, equipada con dos radios y dos pares de antenas de 2,4 y 2,9 metros respectivamente, y cuyo propósito era la obtención de datos en la ionosfera. A pesar de que solo estuvo 92 días en órbita, su lanzamiento y puesta en órbita constituyó un rotundo éxito, ya que la última frontera había sido atravesada y la humanidad daba sus primeros y vacilantes pasos en el espacio.

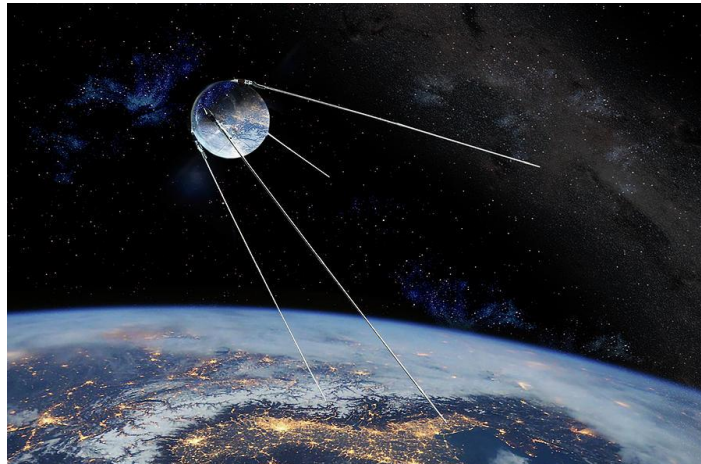


Figura 2-1 Sputnik 1 (Fuente: TourHistoria)

Con este punto de partida, empieza la que se denominará Era espacial, caracterizada por un vertiginoso avance de la tecnología en vectores de lanzamiento, sistemas de comunicaciones y control, radares de seguimiento, nuevos materiales y en infinidad de otras áreas. Tan sólo tres meses más tarde, Estados Unidos consigue poner en órbita el **Explorer**⁴ mientras que en la URSS ponían al primer ser vivo en órbita alrededor de la Tierra, una perra llamada Laika, a bordo del **Sputnik 2**. Una década más tarde, el 20 de julio de 1969, Estados Unidos consigue alunizar con el **Apolo XI**, y Neil Armstrong se convierte en el primer ser humano que pisa la Luna.

² Punto más lejano la Tierra de la órbita.

³ Punto más cercano la Tierra en la órbita.

⁴ El Explorer-1, oficialmente Alpha 1 se lanzó el 31 de enero de 1958 desde Cabo Cañaveral a bordo del cohete Juno I, en respuesta al lanzamiento del Sputnik I, dando comienzo a la carrera espacial, estrechamente relacionada con la Guerra Fría. Envío datos durante algo menos de cuatro meses, hasta que sus baterías se agotaron. El pequeño satélite permaneció en órbita hasta el 31 de marzo de 1970, cuando se produjo su reentrada en la atmósfera y se precipitó al océano Pacífico. Fue el primer lanzamiento del programa Explorer, una larga serie de más de noventa satélites estadounidenses.

2.2.1.2 Evolución de las comunicaciones por satélite

Dentro de este desarrollo tecnológico sin precedentes, la primera comunicación por satélite se produjo el 10 de julio de 1962 a través del primer satélite de comunicaciones, el **Telstar I** (véase Figura 2-2 Telstar I (Fuente: NASA History)), que también fue el primero en transmitir



Figura 2-2 Telstar I (Fuente: NASA History)

señales para la televisión.

Este artefacto, que operaba en órbita baja, era capaz de atender hasta 600 comunicaciones con un receptor en la banda de los 6 GHz y un transmisor de 4 GHz. Fue seguido por estaciones en tierra en diferentes localizaciones: Maine (Estados Unidos), Goonhilly Downs (Reino Unido) y Pleumeur-Bodou (Francia). Dado la escasa potencia del Telstar, escasamente 600W (menos potencia con la que se alimenta hoy en día un microondas), las antenas de estas estaciones ocupaban más de 300 metros cuadrados.

En 1963, el satélite **Syncom** consigue establecerse en una órbita geoestacionaria, le suceden el **Syncom II**, y, en 1964, el **Syncom III**, el satélite más famoso de la serie, utilizado para retransmitir los Juegos Olímpicos en vivo desde Tokio. En 1965 es lanzado el primer satélite comercial geoestacionario, el **Early Bird** (Intelsat). A raíz de estos éxitos, y de la ampliación de sus posibilidades y funciones, el crecimiento de la tecnología satelital es exponencial. Así, sus hermanos, **Intelsat II** y **III**, proporcionaron más enlaces para completar la primera red global, que, en 1969, permitió a más de medio billón de personas ver aterrizar al hombre en la Luna el 20 de julio de 1969.

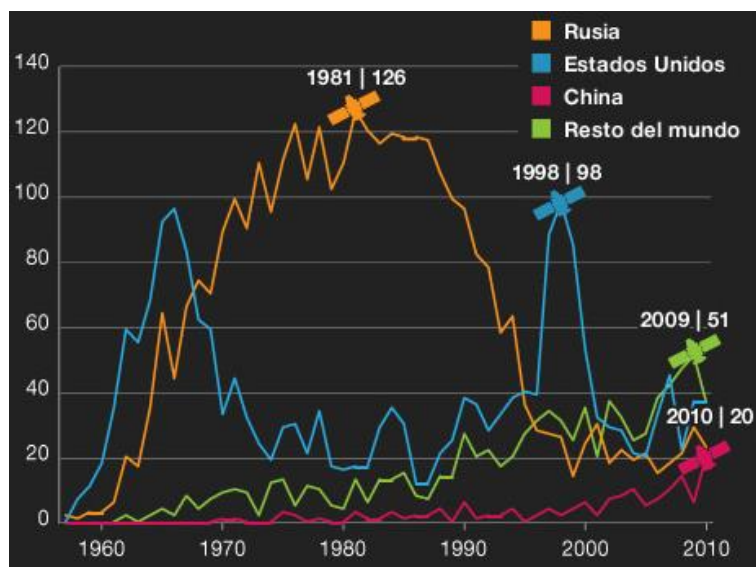


Figura 2-3 Lanzamiento satélites por año y países (Fuente: BBC)

Estas transmisiones globales en vivo junto con las transmisiones telefónicas internacionales se convirtieron en la base de las comunicaciones globales por satélite. Grandes compañías de televisión, como HBO, o de telefonía, como AT&T, lanzan sus propios ingenios al espacio. De esta forma, se empiezan a gestar tecnologías imprescindibles hoy en día como los GPS, los teléfonos móviles, o la televisión en directo e incluso Internet.

A principios de la década de 1970, tanto Canadá como Estados Unidos comienzan a desarrollar redes nacionales de satélites. En Canadá, el primer satélite de comunicaciones nacionales de América del Norte, el **Anik**, fue lanzado en noviembre de 1972. El primer satélite de comunicaciones nacionales estadounidense fue el **Westar**, lanzado en abril de 1974.

Al principio, los satélites norteamericanos se utilizaban principalmente para comunicaciones telefónicas y de datos de larga distancia. Luego, en 1975, un servicio de televisión de pago estadounidense anunció que comenzaría a usar un satélite para proporcionar sus emisiones a estaciones de televisión por cable en todo el país. El 30 de septiembre de 1975, Home Box Office, Inc., salió al aire, ofreciendo cobertura en vivo de un combate de boxeo de campeonato mundial. Muy rápidamente, otros servicios se unieron a HBO en puntos de arrendamiento a bordo de satélites, alimentando una explosión en el desarrollo de miles de sistemas de televisión por cable estadounidenses.

Estas primeras aplicaciones comerciales por satélite (telefonía internacional, televisión, televisión por cable y redes de datos) se vieron debilitadas por el crecimiento de la industria a finales del siglo XX. A partir de 1976 gracias al lanzamiento del **Marisat**, primer sistema de comunicaciones en proporcionar servicios móviles (a la Marina de los Estados Unidos y otros clientes marítimos), las comunicaciones globales crecieron exponencialmente, a medida que se introdujeron nuevas tecnologías. Tres de los avances más significativos derivados de estas tecnologías: los teléfonos celulares, televisión por satélite de transmisión en directo e Internet; incitaron a un rápido progreso en el desarrollo de satélites de órbita terrestre baja (LEO).

Se desplegaron cientos de sofisticados LEO, proporcionando comunicaciones digitales de alta velocidad incluso a las regiones más aisladas del mundo. Desde los teléfonos satelitales globales portátiles de **Iridium**, activados por primera vez en noviembre de 1998, hasta la tecnología de satélites de transmisión directa de DirecTV, lanzada en 1994, pasando por el servicio Web Inalámbrico, introducido a los clientes de telefonía digital en 1999, los satélites de órbita terrestre baja convirtieron el mundo en un lugar más pequeño.

A día de hoy, se estima que hay más de 2.000 satélites activos orbitando alrededor de la Tierra y forman ya una parte intrínseca del bienestar social de hoy en día al ser fundamentales en el funcionamiento de los servicios de telefonía, ubicación o Internet. Además, su número va en aumento, según expertos del MIT (*Massachusetts Institute of Technology*), su número podría quintuplicarse en tan solo una década, debido a la creciente red de Internet 5G, así como a otros ambiciosos megaproyectos como **Starlink**, de SpaceX, que planea poner en órbita 12.000 satélites para el año 2027.



Figura 2-4 N.º satélites en órbita (Fuente: Statista [17])

A modo de conclusión, se puede decir que, desde que la humanidad empezó sus incursiones espaciales, el número de cuerpos que orbitan, transmiten y reciben datos continuamente, no ha hecho si no aumentar y se prevé que, en los próximos años no solo no pare de crecer, si no que el número de lanzamientos sea más y más frecuente hasta alcanzar los más de 15000 satélites en órbita en únicamente los próximos diez años. (Figura 2-4) ⁵

2.2.2 Satélites

En tecnología aeroespacial se define el término satélite como un cuerpo que ha sido lanzado y puesto en órbita de algún astro persiguiendo una finalidad. Es importante destacar que en este trabajo siempre nos referiremos a satélites artificiales, ya que los naturales, como la Luna, no son útiles para este estudio.

Un satélite es, básicamente, un sistema de comunicaciones autónomo con la capacidad de recibir señales de la Tierra y retransmitir esas señales con el uso de un transpondedor, un receptor integrado y un transmisor de señales de radio. Los satélites pueden operar por sí mismos o como parte de un sistema más grande, en constelaciones de satélites.

2.2.2.1 Arquitectura general

Para comprender la arquitectura básica de estos artefactos podemos definir una serie de componentes comunes. Además, en función del objetivo que persigan, los satélites añadirán una serie de objetivos específicos y propios para su misión.

A grandes rasgos, estos componentes comunes son:

- **Sistema de suministro de energía:** proporciona al satélite la energía necesaria para su funcionamiento. Prácticamente la totalidad de satélites usa placas de paneles solares para alimentarse.

⁵ En este gráfico solo se incluyen proyectos llevados a cabo por agencias de forma individual, los proyectos en conjunto no son tenidos en cuenta

- **Sistema de control:** ordenador principal del satélite. Ejecuta las instrucciones que tenga cargadas y las órdenes enviadas desde el segmento terrestre
- **Sistema de comunicaciones:** es el conjunto de antenas, receptoras y transmisoras, que permiten la comunicación con la plataforma de seguimiento para enviar los datos y recibir instrucciones.
- **Sistema de posicionamiento:** sirven para mantener y controlar al satélite en su órbita y dirigirlo hacia sus objetivos.
- **Blindaje:** estructura básica del satélite que contiene y protege al resto de componentes. Además, hará la función de aislante térmico para proteger los instrumentos sensibles del satélite.
- **Sistemas específicos:** carga útil del satélite. Es el conjunto de elementos y sistemas diseñados para cumplir con la misión del satélite. Pueden ser de muy diversos tipos destacando, por ejemplo, sistemas militares o sistemas meteorológicos.

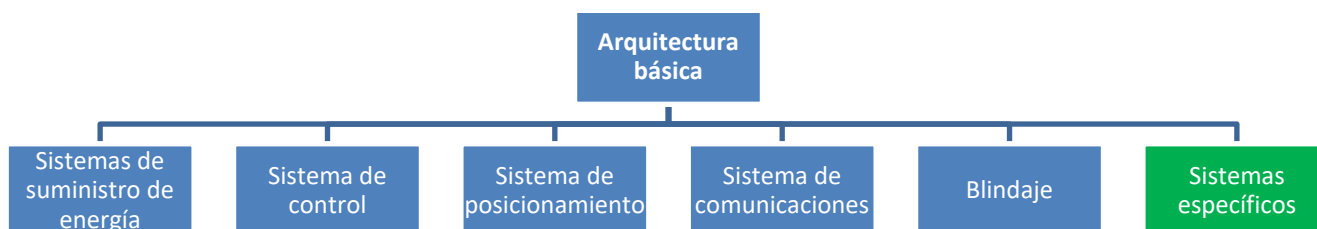


Figura 2-5 Arquitectura general de un satélite

2.2.2.2 Clasificación y tipos de satélites

En cuanto a los tipos de satélites, se pueden ordenar y distinguir de diferentes formas, pero las clasificaciones principales se hacen en función de la finalidad que persiguen y en función del tipo de órbita que describen.

A continuación, desarrollaremos brevemente los distintos tipos de clasificaciones.

2.2.2.3 Satélites según su finalidad

Los satélites pueden tener usos muy diversos y, en general, tener varias finalidades debido al coste de elaboración y puesta en órbita. Sin embargo, podemos establecer unas diferencias claras que facilitará la comprensión de sus usos. Se puede decir, por tanto, que los satélites se clasifican según el objetivo que persiguen en:

1. **Satélites de telecomunicaciones:** su finalidad principal es la de permitir el enlace y comunicación dentro de puntos en su área de cobertura.
2. **Satélites de usos militares:** los hay de diferentes tipos. Seguimiento y destrucción de cabezas nucleares, comunicaciones seguras, guiado de misiles balísticos...
3. **Satélites meteorológicos:** para hacer estudios del clima y las condiciones meteorológicas terrestres.
4. **Satélites de geoposicionamiento:** sirven para, por medio de triangulación, los sistemas de ubicación en tiempo real (*Global Positioning System*, GPS)
5. **Estaciones espaciales:** satélites diseñados para alojar a seres humanos en el espacio. Pueden estar diseñadas para estancias de semanas, meses o incluso años.

6. **Satélites astronómicos:** para estudio y observación de otros planetas y galaxias lejanas.
7. **Satélites biológicos:** para la recogida y estudio de muestras biológicas y la composición de la atmósfera.

Posteriormente en este trabajo desarrollaremos en mayor detalle algunos de estos satélites, ya que serán de los que obtengamos los datos y realicemos el seguimiento.

2.2.2.4 Satélites según su órbita

2.2.2.4.1 Satélites GEO (Geostationary Equatorial Orbit)

Satélites de órbita caracterizada por conseguir estar permanente ubicados en la misma posición respecto a un punto terrestre. Esto se consigue ya que orbitan por encima del Ecuador terrestre a una velocidad, aproximadamente unos 3 km/h, y una altura, de 35786 km referida al Ecuador, que les permite tener el mismo periodo orbital que la Tierra.

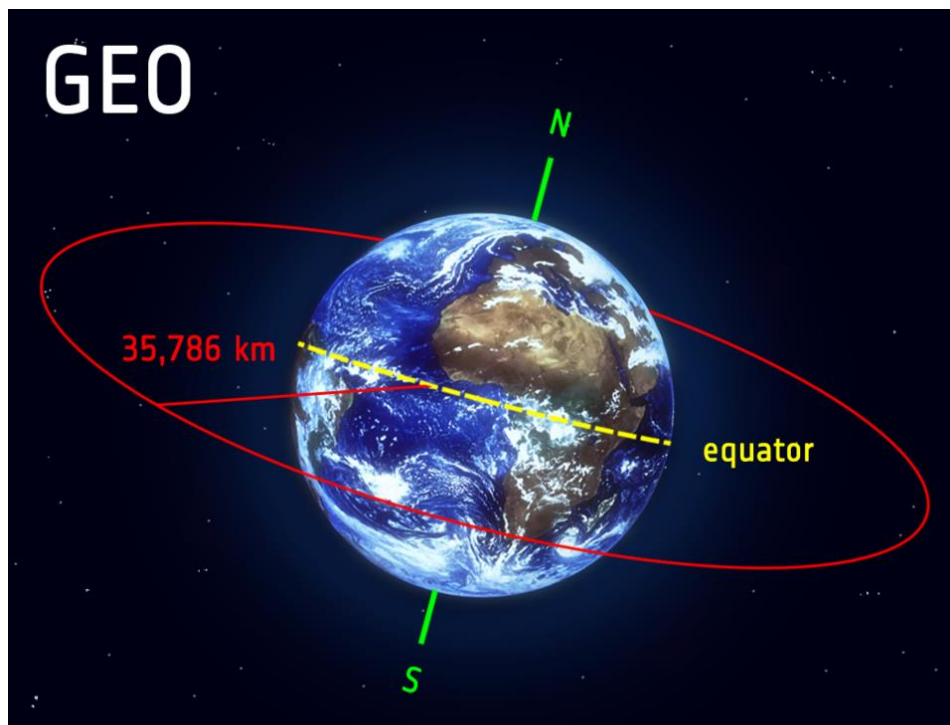


Figura 2-6 Órbita geoestacionaria (Fuente: ESA) [29]

Este tipo de órbitas son muy utilizados por los satélites de telecomunicaciones, diseñados para dar cobertura a una zona geográfica concreta, y que de esta forma permiten que las estaciones terrestres apunten siempre a los satélites sin necesidad de hacer cálculos de seguimiento y apuntamiento. También es frecuente el uso de este tipo de órbitas en satélites meteorológicos para el estudio de las condiciones climáticas en áreas concretas.

Gracias a la gran distancia a la que orbitan estos satélites ofrecen zonas de cobertura muy amplias, cabe destacar que, con solo tres satélites GEO con un desfase de 120°, seremos capaces de ofrecer cobertura global.

2.2.2.4.2 Satélites LEO (Low Earth Orbit)

Satélites de órbitas cercanas a la superficie Terrestre. Estas órbitas suelen estar situadas a una altura de unos 1000 km sobre la superficie, pero se puede dar el caso de satélites que orbiten a una altura de hasta 160 km.

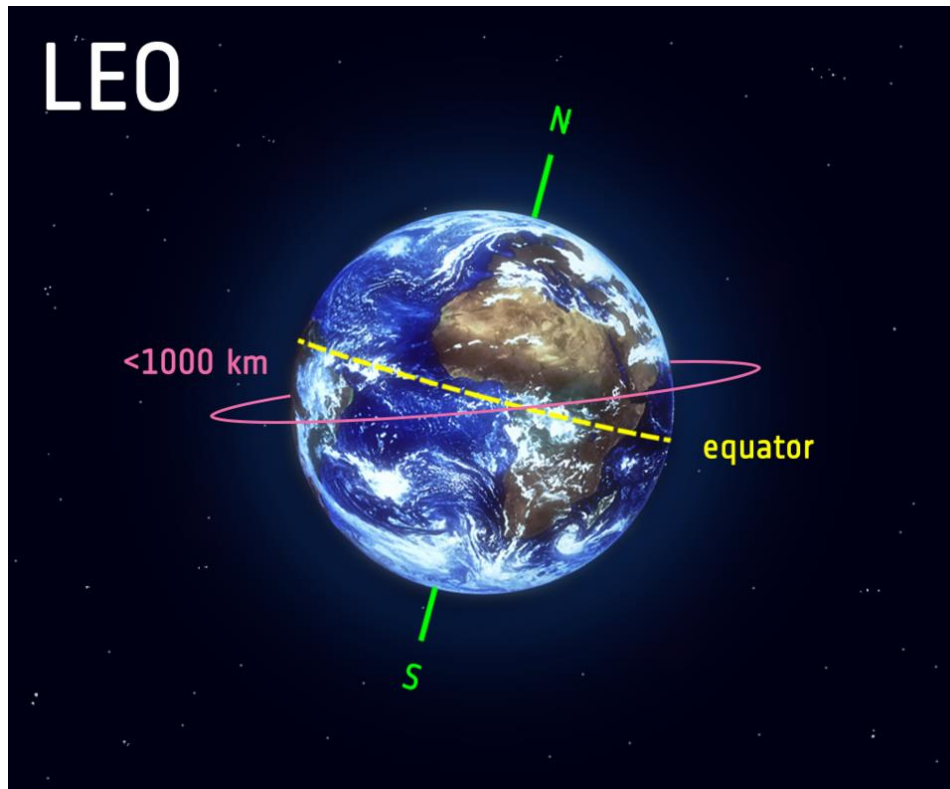


Figura 2-7 Satélites LEO (Fuente: ESA)

A diferencia de los satélites GEO, sus órbitas no son tan fijas lo que permite modificarlas para conseguir distintas áreas de cobertura, lo que les ofrece una gran ventaja y es una de las principales razones por las que se usa este tipo de órbitas.

Otra de las ventajas que ofrece es que, dado a su cercanía, son los que ofrecen una mejor resolución a la hora de obtener imágenes satelitales. También es la órbita usada por la ISS (*International Spatial Station*) dado que facilita el embarco y desembarco de astronautas.

Su órbita se caracteriza por tener una velocidad orbital de unos 8 km/s lo que provoca que tengan un periodo orbital del orden de los 100 minutos, llegando a dar unas 15 vueltas a la Tierra al día.

Sin embargo, estos satélites, individualmente, son poco útiles para las telecomunicaciones ya que suponen un mayor esfuerzo de seguimiento y apuntamiento para las plataformas terrestres. Es por ello que, se suelen emplear constelaciones de satélites LEO para obtener cobertura constante por medio de redes satelitales.

2.2.2.4.3 Satélites MEO (Medium Earth Orbit)

Estos satélites están diseñados como solución de compromiso entre las ventajas que ofrecen los satélites GEO, amplias áreas de cobertura y facilidad de apuntamiento y seguimiento, y los LEO, menor distancia a la Tierra y mayor facilidad a la hora de cambiar las órbitas.

Es muy usado por satélites de posicionamiento como el satélite europeo Galileo, que usa una constelación de satélites de media altura para aportar cobertura a nivel mundial.

2.2.2.4.4 Satélites de órbita polar

Estos satélites orbitan la Tierra de Norte a Sur en lugar de Este a Oeste, pasando por puntos cercanos a los polos, es decir, tienen una inclinación orbital cercana a los 90° . Esta clase de órbitas se usa principalmente para cometidos de teledetección y captura de imágenes. Es por ello que es usado por muchos satélites meteorológicos como los satélites Tiros de la NOAA (*National Oceanic and Atmospheric Administration*).

La constelación de satélites Iridium también utiliza una órbita polar para proporcionar servicios de telecomunicaciones. Otra aplicación importante son los satélites de salvamento marítimo Cospas-Sarsat, que dan cobertura en los polos también.

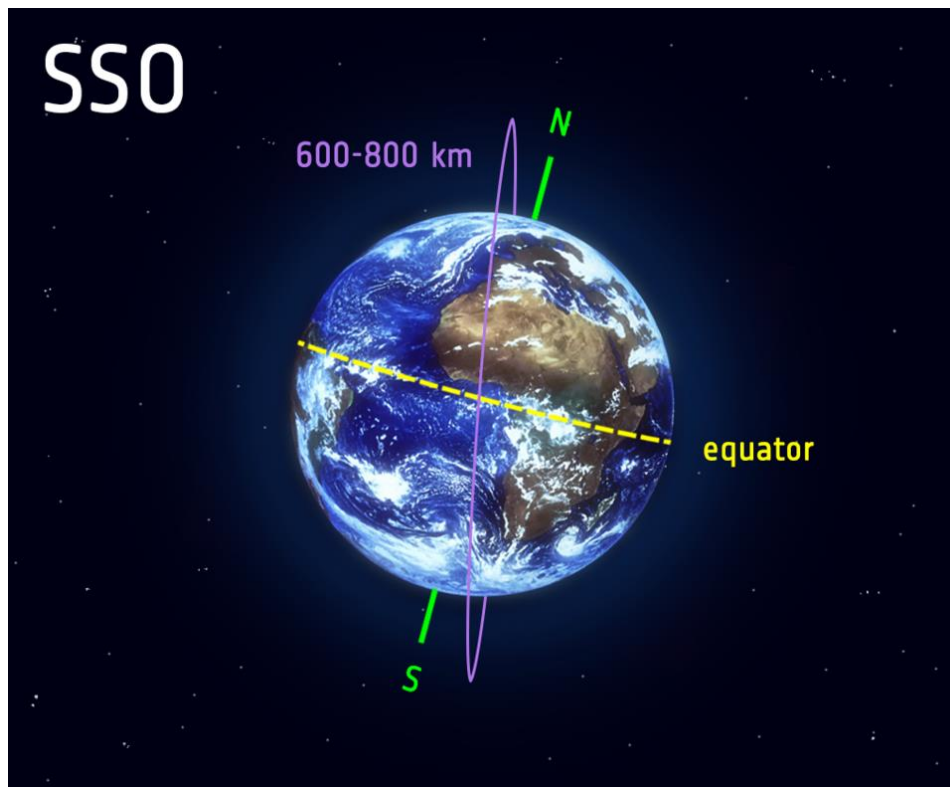


Figura 2-8 Órbita Polar (Fuente: ESA)

2.2.2.4.5 Satélites helio-síncronos (Sun-Synchronous Orbit, SSO)

Son un tipo particular de órbitas polares. Orbitan de Norte a Sur de forma sincronizada al Sol, lo que provoca que siempre pasan a la misma hora por cada determinado punto geográfico. De esta forma, estos satélites permiten la observación de un punto geográfico a una misma hora todos los días lo que permite su uso para medir la evolución de ciertos fenómenos como la deforestación del Amazonas, por ejemplo.

2.2.2.4.6 Órbitas de transferencia (Geostationary Transfer Orbit, GTO)

Un tipo de órbita especial diseñado para el paso de un tipo de órbitas a otro, sobre todo en el lanzamiento y puesta en órbita inicial. Para ello, utilizan una órbita inicial, pero con una alta excentricidad⁶ que les permite, con poco gasto de energía, el de llegada a la órbita, llegar a otras órbitas más altas como las GEO, en lugar de diseñar un lanzamiento con el combustible necesario para alcanzar esa altura.

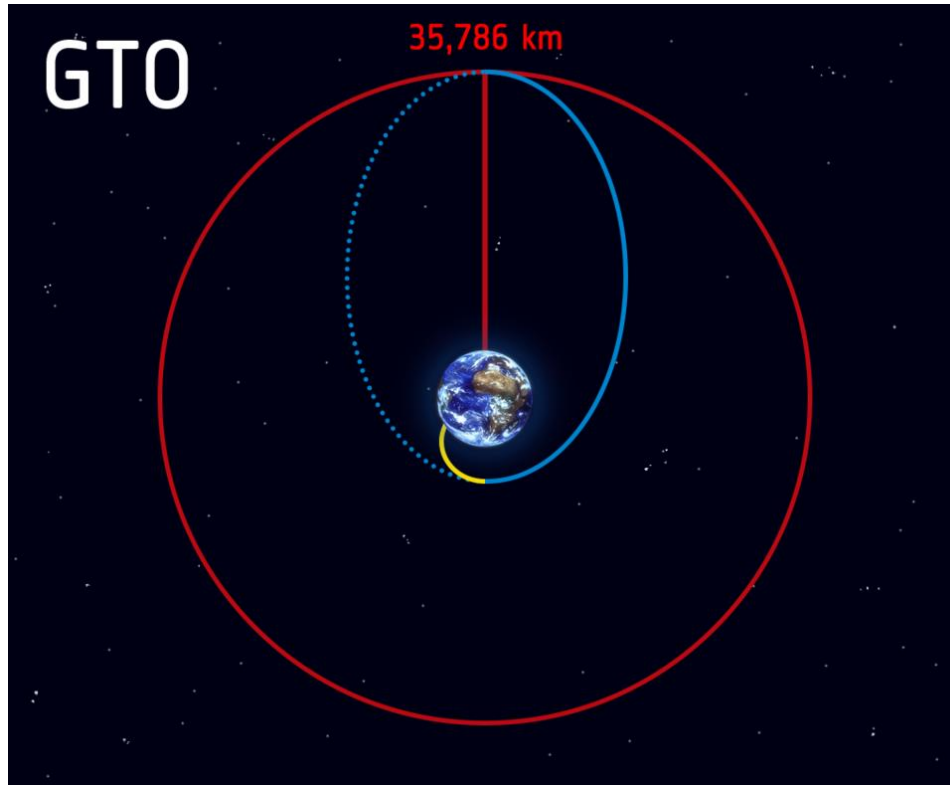


Figura 2-9 Órbitas de transferencia. En amarillo, trayectoria de lanzamiento, en azul, órbita de transferencia y en rojo, órbita final (Fuente: ESA)

Cuando la órbita excéntrica de transferencia llega a la altura deseada, un grupo de propulsión auxiliar lo redirige a la órbita final deseada.

⁶ Grado de desviación respecto a una circunferencia perfecta.

2.2.2.4.7 Puntos de Lagrange (L-points)

Se define punto de Lagrange como la posición en la que un cuerpo pequeño, afectado únicamente por fuerzas gravitacionales, puede estar teóricamente estacionario respecto a dos cuerpos mayores. Estos puntos marcan, las posiciones donde la combinación de las fuerzas gravitacionales de las dos masas grandes proporciona la fuerza centrípeta necesaria para rotar sincrónicamente con la menor de ellas.

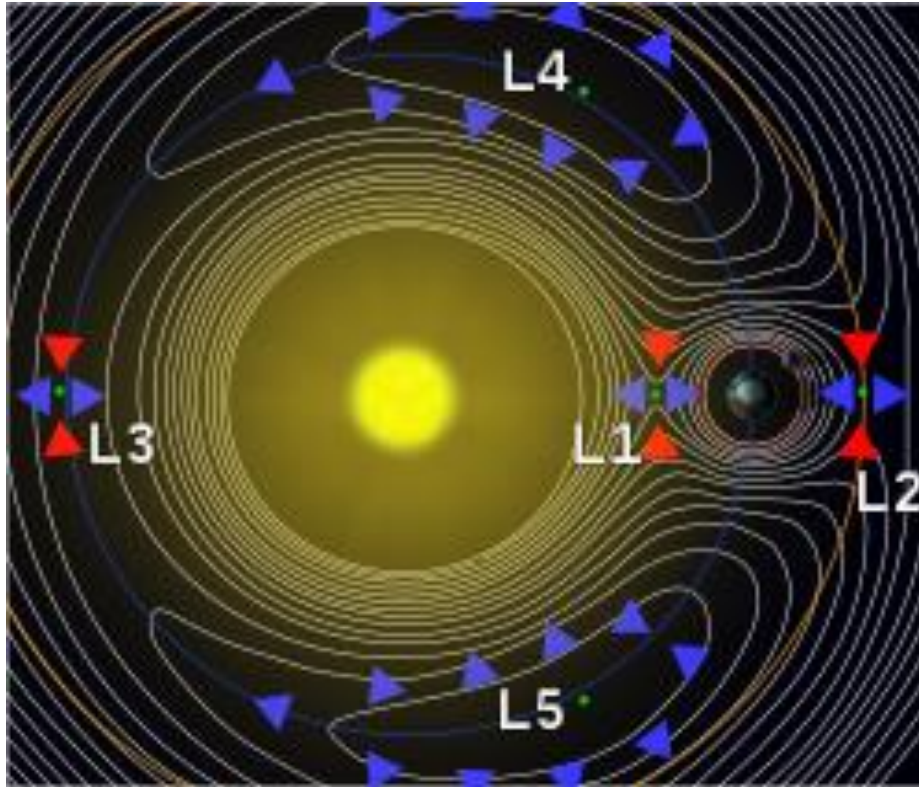


Figura 2-10 Representación de los puntos de Lagrange elaborada por la NASA (Fuente: Wikimedia Commons)

Estos 5 puntos son usados en aplicaciones para las que la cercanía a la Tierra supondría un obstáculo y para exploraciones en el espacio profundo. Algunos ejemplos son el telescopio solar SOHO o el LISA Pathfinder en la zona L1.

Para finalizar, resumimos las características principales y atributos que nos ofrece cada tipo de órbita:

Satélites según su órbita

<i>Característica</i>	GEO	LEO	MEO
<i>Distancia a la Tierra</i>	36786 km	200- 1000 km	1000 – 35876 km
<i>N.º satélites para cobertura global</i>	3	Muy alto (>20)	Dependiendo de la altura
<i>Apuntamiento y seguimiento</i>	Muy sencillo	Difícil	Dificultad media
<i>Pérdidas de propagación</i>	Altas	Bajas	Medias
<i>Uso de terminales móviles</i>	Muy complejo	Posible	Posible
<i>Complejidad de la red</i>	Sencilla	Compleja	Media

Tabla 2-1 Resumen de los satélites según tipo de órbitas

2.2.3 Comunicaciones por satélite

Las comunicaciones por satélite son la tecnología satelital más madura de todas. Empezaron hace 50 años con el Telstar I en 1962 y el Syncom en 1963 y no han parado de evolucionar desde entonces, hasta el punto en que hoy en día forman parte de nuestra vida cotidiana.

Al principio, el rendimiento y la potencia de los satélites era muy limitado. Para compensar esto, se requirieron estaciones terrestres muy grandes con antenas parabólicas de más de 20 metros de diámetro para establecer vínculos con ellas. El uso de satélites se limitaba a la telefonía de larga distancia y al transporte de señales de televisión entre diferentes cadenas.

Las nuevas tecnologías y los diferentes tipos de demanda han cambiado la forma en que se utilizan los satélites de comunicaciones. El uso de satélites más potentes y de frecuencias más altas han hecho posible que muchas personas puedan recibir señales satelitales desde el espacio.

Este avance tecnológico se basa principalmente en: el aumento de la potencia disponible, mejora del rendimiento, uso de frecuencias más altas y, sobre todo, mejoras en la directividad del lóbulo y el seguimiento de satélites. Hoy en día, forman una parte imprescindible de nuestra vida, y nuestro actual estilo de vida no sería comprensible sin estas tecnologías.

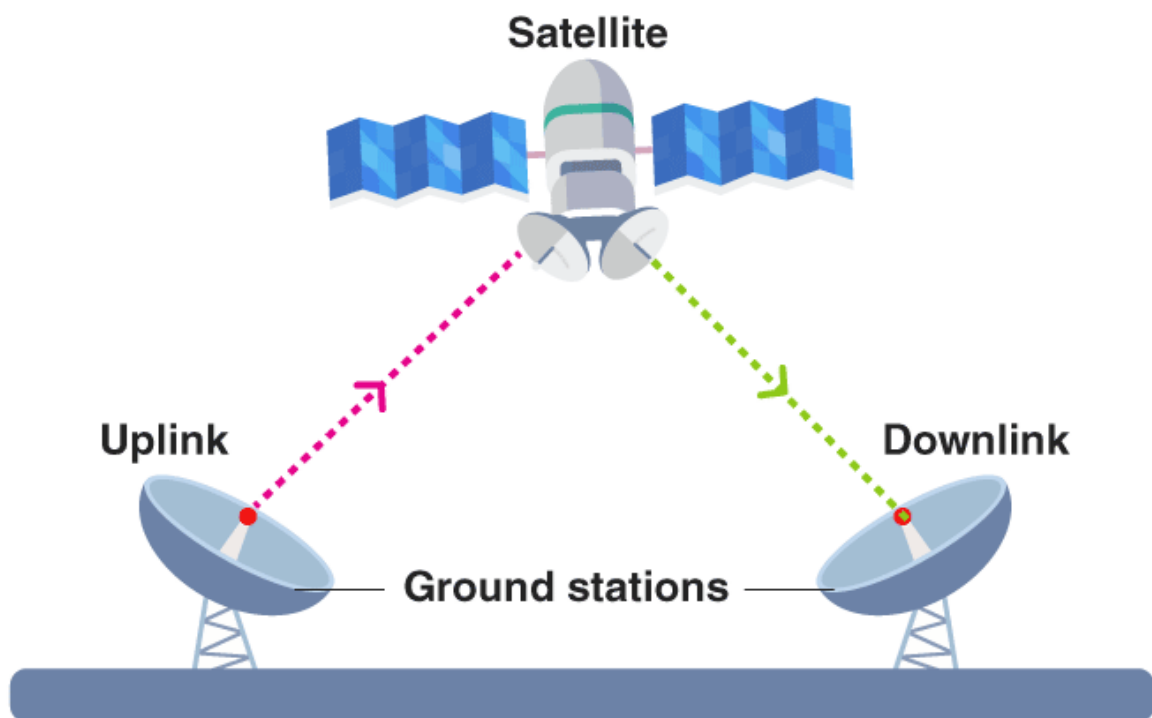


Figura 2-11 Enlace satelital (Fuente: Byjus)

2.2.3.1 Componentes de las comunicaciones satelitales

Entrando en detalles más técnicos, podemos decir que, todas las comunicaciones por satélite tienen dos componentes principales: el segmento terrestre, que consiste en transmisión, fija o móvil, y recepción de datos e instrucciones, y el segmento espacial, que es, principalmente, el propio satélite.

Un enlace satelital típico implica la transmisión o el envío de una señal de una estación terrestre a un satélite. A continuación, el satélite recibe y amplifica la señal y la retransmite a la Tierra, donde es recibida y re-amplificada por las estaciones y terminales terrestres, para su posterior distribución por cable u otros medios.

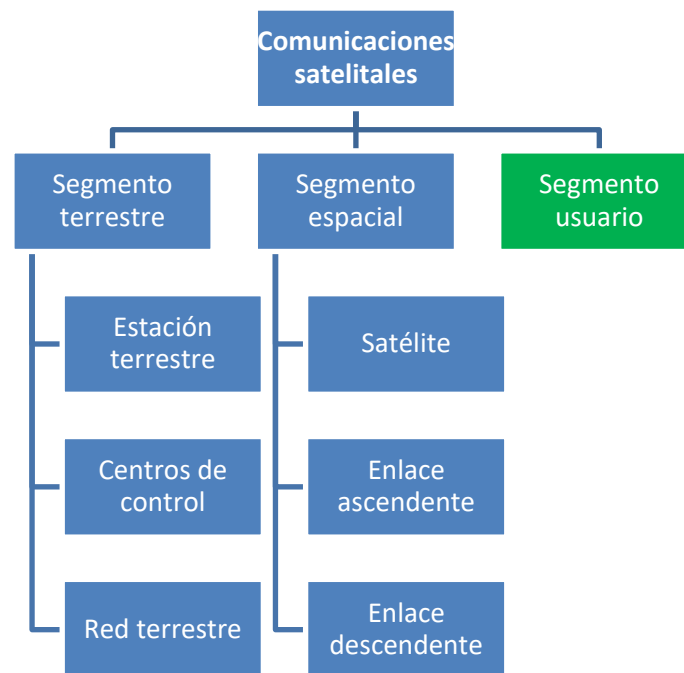


Figura 2-12 Componentes de los enlaces satelitales

2.2.3.2 Segmento terrestre

Es la parte de las comunicaciones por satélite que consta de todos los elementos necesarios para la recepción y transmisión, comunicación, amplificación de la señal y mando y control del satélite. Este segmento permite la descarga de datos y la telemetría entre las partes interesadas.

Los elementos primarios del segmento terrestre son:

- **Estaciones terrestres:** encargadas de realizar la comunicación con el satélite para la recepción/transmisión de los datos. Para que puedan desarrollar su función, es crítico un correcto y muy preciso apuntamiento y seguimiento al satélite.
- **Centros de control de misión:** encargados de la comprobación, seguimiento y corrección de la órbita del satélite, así como de darle las órdenes e instrucciones necesarias.
- **Redes terrestres:** para la redistribución de datos a los destinatarios interesados.

2.2.3.3 Segmento espacial

El segmento espacial está compuesto por el satélite propiamente dicho y los datos desde (*Downlink*) y hacia (*Uplink*) el satélite.

- **Satélite:** es básicamente un sistema de comunicaciones autónomo con la capacidad de recibir señales de la Tierra y retransmitir esas señales con el uso de transpondedores, con receptores integrados y transmisores de señales de radio.

Pueden distinguirse cuatro componentes básicos y comunes para todo tipo de satélites: sección de recepción, de conmutación, de conversión y de transmisión.

- **Uplink:** se denomina así a la transmisión que envía la estación terrena al satélite
- **Downlink:** transmisión del satélite hacia la estación terrestre.

Es muy importante destacar la diferencia que existe entre el diseño del enlace ascendente y el enlace descendente, ya que supone uno de los principales factores de diseño en las comunicaciones satelitales.

Ambos enlaces se propagan en condiciones de espacio libre, lo que conlleva una atenuación de la señal proporcional al cuadrado de la distancia y de la distancia.

$$\alpha_{EL} = \frac{P_{ISO}}{P_{RAD}} = \frac{1}{r^2} (\lambda/4\pi)^2$$

En unidades logarítmicas:

$$L_{EL} = -10 \log \alpha = 20 \log(r) + 20 \log\left(\frac{4\pi}{\lambda}\right) \text{ dB}$$

El enlace descendente, el más desfavorable, debido a las limitaciones en potencia del satélite tendrá una frecuencia menor que el enlace ascendente.

Además, pueden apreciarse más interferencias, siendo las principales:

- **Interferencia por satélites adyacentes:** las antenas de las estaciones de tierra no transmiten toda la potencia en una única dirección (lóbulo principal) sino que pequeños lóbulos laterales radian potencia hacia otras direcciones. La potencia radiada por estos lóbulos puede incidir en satélites adyacentes y dar lugar a interferencias.
- **Interferencia terrestre:** la banda de frecuencia empleada por los enlaces terrestres de microondas. La interferencia mutua entre una estación de tierra y un enlace de microondas terrestre viene dada por la potencia de las señales, así como por el margen entre la banda de frecuencias de cada señal.
- **Interferencia por polarización cruzada:** se pueden producir interferencias por acoplamiento de energía de señal. A estos efectos también es necesario tener en cuenta que a altas frecuencias los efectos de la lluvia conllevan la despolarización de las señales.
- **Interferencia por canales adyacentes:** la energía de alguna señal limitada en banda cae dentro de algún canal adyacente debido, fundamentalmente, a las características del filtro del canal.
- **Interferencia de intermodulación:** interferencia está causada por los productos de intermodulación generados dentro del transpondedor como resultado de la amplificación de múltiples portadoras.

2.2.4 Satélites de comunicaciones

Los satélites son esencialmente artefactos para recibir y transmitir datos, por lo que se podría considerar a cualquier satélite de comunicaciones, pero en este subapartado nos referimos a los

satélites cuya finalidad es posibilitar la comunicación entre estaciones terrenas distintas en forma de datos, voz, imágenes etc.

Como hemos visto en los apartados anteriores, los satélites son el nexo entre distintas estaciones terrestres, que se comunican entre sí gracias a la intermediación del satélite. Las señales electromagnéticas enviadas pueden transportar todo tipo de información, desde llamadas telefónicas o datos de Internet hasta transmisiones de radio y televisión.

La gran ventaja que ofrecen las comunicaciones es la de vencer los problemas de la radiodifusión por la cobertura terrestre. Además, hacen posible la comunicación desde y hacia áreas remotas donde los sistemas comunes de comunicación, cableado o radio, por ejemplo, no pueden llegar.

A continuación, desarrollaremos brevemente los principales sistemas y compañías de telecomunicaciones vía satélite, haciendo hincapié en aquellos que ofrecen servicios a la Armada Española.

2.2.4.1 INMARSAT

INMARSAT es una compañía británica que ofrece servicios de telefonía satelital con cobertura casi global, no ofrece cobertura en los polos. Cuenta con una flota de 12 satélites geoestacionarios de los que cuatro están operativos y uno permanece en reserva.

En la siguiente imagen podemos observar cómo están ubicados los satélites para ofrecer cobertura a nivel mundial.

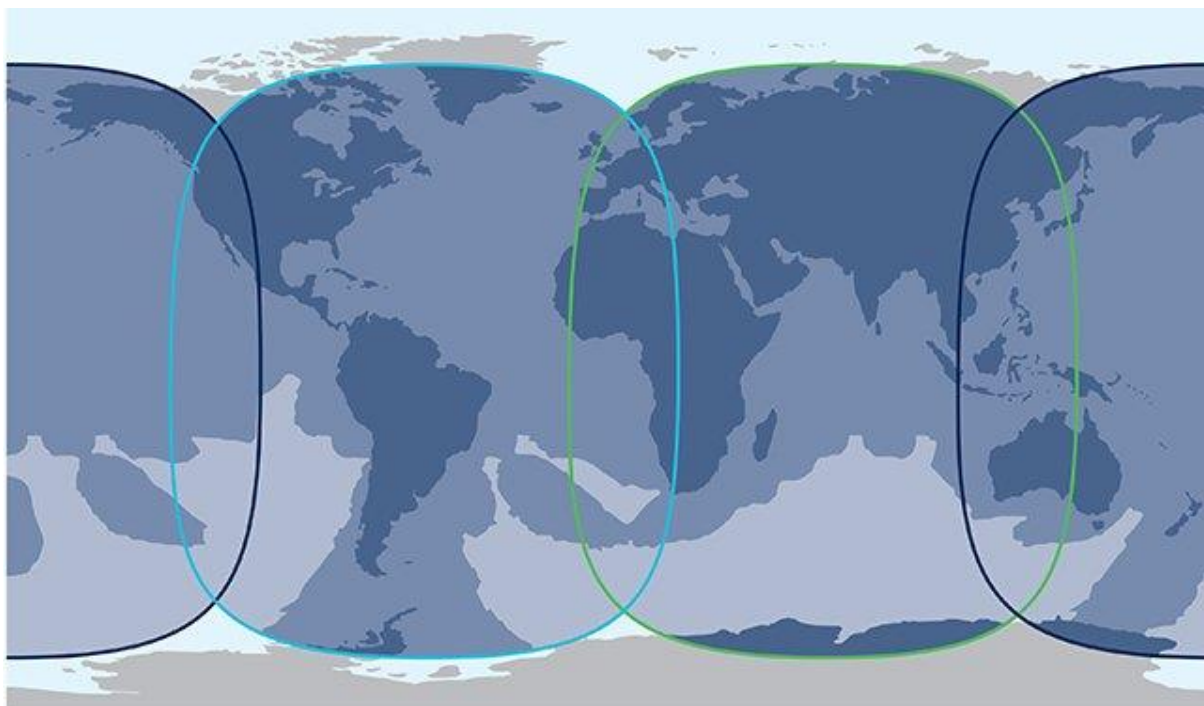


Figura 2-13 Cobertura de INMARSAT (Fuente: INMARSAT) [30]

La red INMARSAT está formada por una serie de subredes que responden a una serie de estándares de comunicaciones. Además, se subdividen en cuatro regiones oceánicas que constan de los siguientes elementos:

- **NCS (Network Control Station, estaciones de control de la red):** gestionan la subred, asignan canales satelitales para cada llamada, liberan los canales desocupados y mantienen las bases de datos de los sistemas.
- **LES (Land Earth Station, estaciones terrestres):** facilitan el acceso de las redes fijas a la constelación de satélite. Por ejemplo, Télex o RPTC hacen uso de ellas. Hay más de una por región lo que permite al terminal decidir con cuál de ellas enlazar.

- **MES (Mobile Earth Station, estaciones móviles):** son los usuarios de la red propiamente dichos, los terminales. Pueden ser desde barcos, aviones o incluso personas.

Entre los servicios que ofrece INMARSAT destacan los de telefonía y datos de conexión a Internet de forma automática. Su uso está creciendo entre los navegantes al ser un medio probado y seguro para comunicarse con tierra en caso de necesidad.

Otro de los servicios más importantes que nos da son los “Safety Net”, mensajes de seguridad enviados a zonas geográficas para alertar de posibles riesgos a la navegación enviados desde agencias costeras, meteorológicas o de salvamento y rescate.

Como conclusión, INMARSAT es una empresa con sede en Reino Unido que, gracias a sus 12 satélites GEO, ofrece cobertura prácticamente a nivel global, con la salvedad de los polos terrestres.

2.2.4.2 HISPASAT

HISPASAT es una empresa con sede en España que ofrece cobertura en Norte América, Sudamérica, África Occidental y Europa. Trabaja en el marco de las comunicaciones comerciales y gubernamentales, ofreciendo funcionalidades como la transmisión de datos, imágenes, llamadas o videoconferencias.

Dentro de sus capacidades, se encuentra la de poder distribuir más de 1250 canales de radio y televisión a millones de hogares, así como de proveer banda ancha a zonas fijas, es un motor clave de la industria aeroespacial española.



Figura 2-14 Cobertura de HISPASAT (Fuente: HISPASAT) [31]

HISPASAT cuenta con una flota de 9 satélites geoestacionarios, situados entre las longitudes 30° y 70° Oeste todos ellos, cuyo primer lanzamiento fue el HISPASAT 1, en el año 1992 desde la Guayana Francesa, quedando situado en una posición sobre el Ecuador a una longitud de 30° Oeste, centrado para ofrecer cobertura en los continentes americano y europeo. Hoy en día, sobre todo tras los lanzamientos de la serie Amazonas (Amazonas 1, 2, 3, 4 y 5) y el HISPASAT 1F es uno de los mayores proveedores de cobertura a nivel mundial.

POSICIÓN ORBITAL	SATÉLITE	FINES	AÑO DE LANZAMIENTO
30° OESTE	Hispasat 30W-5	Comerciales	2010
30° OESTE	Hispasat 30W-6	Comerciales	2018
36° OESTE	Hispasat 36W-1	Comerciales	2017
55.5° OESTE	Hispasat 55W-2	Comerciales	205
61° OESTE	Amazonas 2	Comerciales	2009
61° OESTE	Amazonas 3	Comerciales	2013
74° OESTE	Hispasat 74W-1	Comerciales	2014
61° OESTE	Amazonas 5	Comerciales	2017
70° OESTE	Hispasat 70W-1	Comerciales	2015
29° ESTE	Xtar-Eur	Gubernamentales	2005
30° OESTE	Spainsat	Gubernamentales	2006

Tabla 2-2 Flota de satélites HISPASAT (Fuente: HISPASAT)

Además de los anteriormente mencionados, con fines comerciales, HISPASAT puso en órbita dos satélites con fines gubernamentales. La sociedad participada por HISPASAT, HISDESAT enfocada en estudio de servicios y comunicaciones estratégicas, realizó el lanzamiento y puesta en órbita del satélite XTAR-EUR en el año 2005, en una posición ecuatorial a 29° Este, y del SPAINSAT en el 2006, situado a 30° Oeste, que proporcionan enlaces al Ministerio de Defensa, principalmente, pero también a otras agencias del gobierno.

A modo de ejemplo, con visualizaciones ofrecidas por Satbeams [1], se ha elaborado la siguiente composición de la cobertura ofrecida por el satélite gubernamental XTAR-EUR. En ella podemos ver, en la fila superior de izquierda a derecha: el rayo de cobertura europea, el de

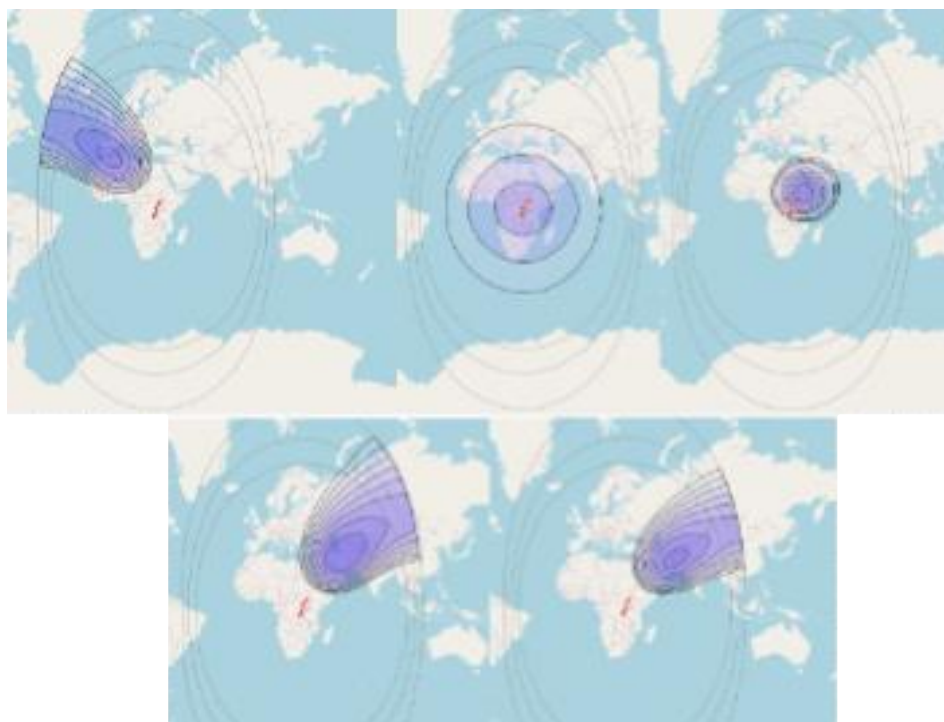


Figura 2-15 Cobertura del XTAR-EUR

cobertura global y el de África Oriental; en la fila inferior: a la izquierda el rayo de cobertura de Oriente Medio y a la derecha el Suroeste asiático (Figura 2-15).

De esta manera, España adquiere un grado de independencia, disponibilidad y fiabilidad de comunicaciones satelitales clave en este mundo globalizado y esenciales para el desempeño de las misiones y operaciones de nuestras Fuerzas Armadas.

2.2.4.3 Iridium

Es una red de 66 satélites LEO distribuidos en seis órbitas bajas, cada una contiene 11 satélites, a una altura aproximada de 780 km que son equidistantes entre sí y tienen un período orbital de unos 100 minutos.

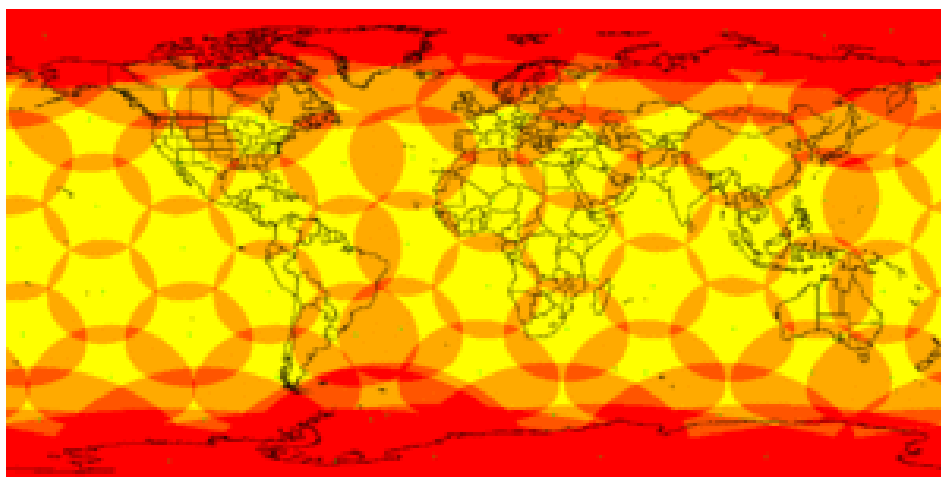


Figura 2-16 Cobertura de los satélites Iridium (Fuente: Iridium)

La red Iridium fue diseñada para dar cobertura global a teléfonos satelitales móviles, y, a diferencia de la red de INMARSAT, también ofrece cobertura polar, actualmente es capaz de proporcionar comunicaciones de datos, acceso a Internet e intercambio de voz y vídeo.

Además, cabe destacar que, desde el atentado del 11-S, los organismos de seguridad estadounidenses lo utilizan como medio preferente de comunicaciones satelitales debido a su fiabilidad, seguridad y rapidez.

También es usado por los buques de la Armada como método alternativo a los satélites de Hispasat, aunque es solo un método secundario debido al alto coste de uso y para uso de emergencia en regiones polares, que carecen de cobertura tanto de INMARSAT como de HISPASAT.

2.2.5 Descripción del sistema español de comunicaciones por satélite (SECOMSAT)

En este apartado, desarrollaremos sin entrar en detalle, la infraestructura de las comunicaciones satelitales en las Fuerzas Armadas, aunque mencionada parcialmente anteriormente, a través de la explicación de la estructura de las comunicaciones satelitales y distintas compañías satelitales, ahora podremos ponerlo en contexto más fácilmente y desarrollar los conceptos de forma más clara.

Para empezar, debemos mencionar a HISDESAT S.A., que se autodefine como: “proveedor comercial internacional de servicios de comunicaciones por satélite en Banda X y Banda Ka, para aplicaciones exclusivamente gubernamentales, tanto en el ámbito militar como en el civil.”

Es decir, es la agencia que gestiona y proporciona la estructura para las telecomunicaciones a los organismos del gobierno.

Banda	Frecuencia (GHz)
L	1 – 2
S	2 – 4
C	4 – 8
X	8 – 12
Ku	12 – 18
K	18 – 27
Ka	27 – 40
V	40 – 75

Tabla 2-3 Bandas de frecuencia

La flota espacial cuenta con tres satélites: SPAINSAT, XTAR-EUR y el HISPASAT 1D (ya inactivo); que transmiten en las bandas: Ka, Ku y X (Tabla 2-3). De esta forma el margen de frecuencias queda definido de la siguiente manera (Figura 2-17).



Figura 2-17 Repartición de la banda de frecuencias por los satélites de SECOMSAT

Al tratarse SECOMSAT de un sistema redundante (Figura 2-17), el ancho de banda gubernamental de SPAINSAT (satélite principal) se encuentra disponible en su totalidad para

este uso, mientras que el ancho banda de XTAR-EUR (satélite secundario) se contrata anualmente según la previsión de misiones a desarrollar, aunque MINISDEF dispone según sus necesidades del resto de ancho de banda contratable.

De igual manera en ambos satélites, se puede incrementar la capacidad de ancho de banda en el margen “X” de forma temporal y según necesidades puntuales en canales comerciales.

2.2.5.1 Centros de Control de los satélites SECOMSAT

Los sistemas de comunicaciones de HISDESAT cuentan con un sistema de control terrestre para sus satélites, centros donde se analiza la telemetría y donde se preparan los comandos que se enviarán a los satélites. Estos centros están equipados con antenas TTC (*Telemetry, Tracking and Control*, en inglés Rastreo, Telemetría y Comando) y operan diariamente y que cuentan con los sistemas informáticos para controlar y monitorear satélites desde tierra. Debido a la naturaleza gubernamental de los servicios ofrecidos, el sistema es altamente seguro y se vuelve totalmente redundante mediante el uso de componentes de respaldo.

Cada satélite tiene dos centros de control idénticos ubicados en áreas geográficamente separadas además de las estaciones y antenas TTC en dos bandas diferentes (X y S) ubicadas en zonas, Madrid, en Arganda del Rey, y Gran Canaria, en Maspalomas. Ambos satélites pueden ser controlados desde tierra incluso si una de las dos instalaciones está completamente inoperante, garantizando así que el servicio estará disponible continuamente en todo momento bajo cualquier circunstancia.



**Figura 2-18 Estación de Maspalomas
(Fuente: El Mundo)**

2.2.6 Satélites de experimentación y de radioaficionados

Otros tipos fundamentales de satélites son los satélites científicos o de experimentación y los satélites de radioaficionados, de los que hablaremos de forma más extensa al ser motivo de estudio para este trabajo.

2.2.6.1 Satélites científicos

Un satélite o sonda científica lleva instrumentos para obtener datos sobre campos magnéticos, radiación espacial, la Tierra y su atmósfera, el Sol u otras estrellas, planetas y sus lunas, y otros objetos y fenómenos astronómicos. Se podrían establecer las siguientes subclasificaciones dentro de ellos:

- **Satélites atmosféricos:** para el estudio de los fenómenos ocurridos en la atmósfera y estudio de su composición
- **Satélites geodésicos:** investigación de característica de la Tierra como el tamaño y la forma, su campo gravitacional o los campos electromagnéticos
- **Satélites biológicos:** aquellos que transportan seres vivos para estudiar la influencia que tiene en ellos el medio espacial
- **Satélites astronómicos y astrofísicos:** estudian, registran y toman fotos de otros astros, exploración de agujeros negros y de supernovas

Además, pueden ser considerados satélites de experimentación otros como los de exploración del espacio profundo o las sondas enviadas a explorar otros astros y planetas.

2.2.6.2 Satélites de radioaficionados

Son pequeños satélites contruidos por radioaficionados y para usos de radioaficionados. En inglés se les suele denominar como OSCAR (*Orbiting Satellite Carrying Amateur Radio*, en inglés, satélite orbital que transporta radio de aficionado) por la AMSAT (Asociación de radioaficionados), organismo que promueve el uso y conocimiento de estas plataformas.

Pueden estar diseñados para fines muy diversos, pero los principales usos son los de actuar como repetidores, transpondedores lineales, y como relés digitales de almacenamiento y avance.

La comunidad de radioaficionados es muy activa en la construcción de satélites y en la búsqueda de oportunidades de lanzamiento. Las listas de satélites en funcionamiento deben actualizarse periódicamente, a medida que se lanzan nuevos satélites y acaban su vida útil otros más antiguos. La AMSAT es la agencia encargada de publicar información actualizada.

Entre los satélites OSCAR destaca el lanzamiento de CubeSat, un estándar de nanosatélites cuya estructura tiene forma cúbica. Esto se debe a que son sistemas baratos que fueron desarrollados precisamente para su uso por universidades y pequeñas agencias.

Actualmente una gran cantidad de satélites OSCAR están operativos, permitiendo tanto su utilización en fonía, como obtener su telemetría. Los más populares son:

- **AO-7:** porta un transpondedor lineal que permite contactos intercontinentales entre radioaficionados.
- **FO-29:** de la misma manera que el anterior, permite enlaces intercontinentales
- **SO-50:** el satélite de radioaficionado en activo más popular. Usa un único canal de FM V/U y nos permite realizar enlaces continentales.
- **FOX-1A:** desarrollado por la AMSAT NA, fue el primer CubeSat. Tiene similares características a los anteriores.

- **AO-73:** es un CubeSat desarrollado conjuntamente por las agencias AMSAT UK, AMSAT NL y la empresa ISIS-BV. Posee un transpondedor lineal y permite realizar telemetría.

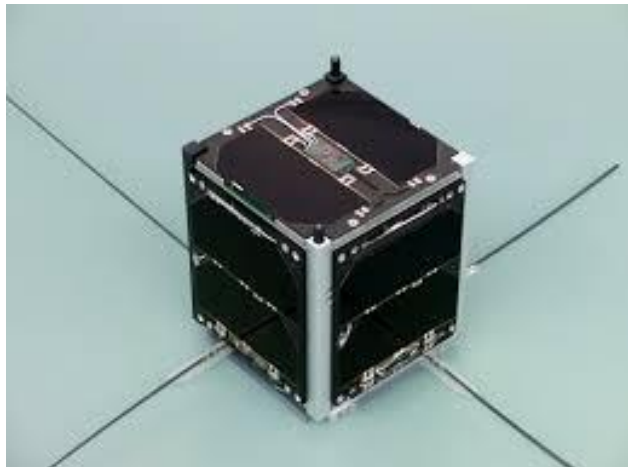


Figura 2-19 Satélite de radioaficionado AO-73 (Fuente: AMSAT) [5]

- **XW-2A/B/C/D/F:** Se trata de una constelación de satélites Xi Wang-2 Hope que fueron lanzados simultáneamente con características similares.

2.3 Plataformas de seguimiento de satélites

En este apartado procederemos a hacer una breve introducción a los sistemas de apuntamiento y seguimiento de satélites, explicando brevemente sus componentes principales, como medio para dar entrada a los sistemas de seguimiento, sus diferentes procedimientos y sus limitaciones.

2.3.1 Componentes de una plataforma de seguimiento

El segmento terrestre de una comunicación satelital tiene la finalidad de recibir y transmitir datos por medio de los satélites, para ello tiene que hacer uso de herramientas propias de la ingeniería y de la de telecomunicaciones, entre otras, lo que las convierte en plataformas complejas y muy técnicas con componentes de gran precisión y fiabilidad. Su tamaño, estructura y complejidad varía en función de su finalidad, sería absurdo decir que una plataforma de seguimiento y control de Hispasat es igual que la de un radioaficionado, pero, sin embargo, hay una serie de elementos comunes que todas deben poseer.

Los elementos principales que deben estar presentes en todo este tipo de sistemas son:

- Un **segmento espacial** al que realizar el apuntamiento y seguimiento y sobre el que desarrollar los cálculos y envíos/transmisiones que deseamos.
- Un **calculador**, software que realice los cálculos preceptivos para el correcto direccionamiento y que puede ser de muchos tipos en función de los datos que le proporcionemos, la órbita que siga el satélite o el grado de precisión y fiabilidad perseguidos.
- Una **estructura** sobre la que montar nuestros componentes y que puede ser de mayor o menor tamaño, fija o móvil.
- Los **motores** para mover la antena en los planos de azimuth y elevación al realizar el seguimiento, suelen ser motores eléctricos.

- **Terminales** del tamaño apropiado y la complejidad necesarios para satisfacer los objetivos buscados por el usuario: distintos tipos y tamaños de antenas, que trabajen en determinadas bandas...
- Una **estación de control de la red** y del terminal para organizar, gestionar, racionalizar y almacenar los datos, mandar las instrucciones del computador y muchas otras.

2.3.2 Tipos de plataformas

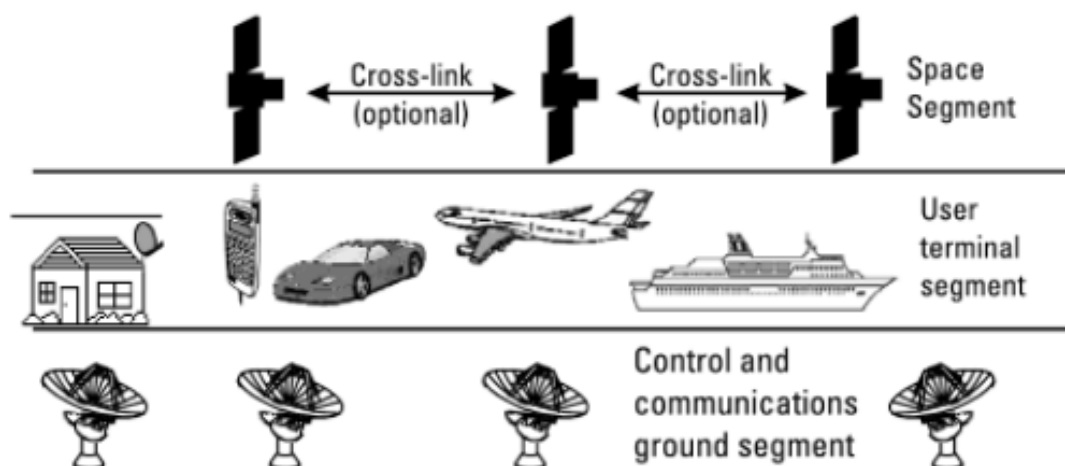


Figura 2-20 Plataformas de seguimiento de satélites [34]

Los tipos de plataformas vienen condicionados por muchas razones: la misión y características del satélite, el tipo de órbita que describe, la precisión que queramos darle a nuestro sistema respecto a peso, movilidad, estabilidad o materiales y otras muchas razones, la principal característica que les diferencia es la estructura sobre la que irán instalados, es decir, si serán sobre plataformas fijas o en plataformas móviles (barcos, coches, aviones...)

A continuación, desarrollaremos brevemente algunos tipos de cada uno de ellos.

2.3.2.1 Plataformas fijas

Este tipo de plataformas suelen ser de grandes dimensiones y son las que generalmente se establecen como NCS (*Network Control Station*). Se caracterizan por sus grandes antenas parabólicas, de hasta 50 m², al no tener las mismas limitaciones de potencia que las que veremos posteriormente.



Figura 2-21 Estación de Robledo de Chavela

Uno de los parámetros clave que tienen este tipo de configuraciones, aparte de su tamaño y alta complejidad tecnológica, es su localización y ubicación, estando emplazadas general en

zonas con una altura importante sobre el nivel del mar para facilitar el establecimiento de enlaces satelitales.

2.3.2.2 Plataformas móviles

2.3.2.2.1 SOTM

El SOTM (*Satcom On the Move*, comunicaciones por satélite en movimiento) es uno de los sistemas militares de comunicaciones satelitales por los que unidades del Ejército de Tierra y de Infantería de Marina se aseguran la disponibilidad, fiabilidad y seguridad de las comunicaciones cuando están desplegados.



Figura 2-22 Vehículos de SATCOM de Infantería de Marina [2]

Consiste, básicamente, en dotar a un vehículo de los instrumentos necesarios para convertirlo en un terminal en movimiento, con enlaces satélite de alta velocidad, siendo compatible con los equipos existentes full-dúplex de la red SECOMSAT, enlazando con los satélites XTAR-EUR o Spainsat.

Los principales requerimientos que viene a satisfacer este sistema son:

- **Fiabilidad y rapidez:** establecimiento veloz de enlaces y recuperación rápida y directa de las comunicaciones en caso de fallo, debido a las circunstancias tácticas derivadas de la misión.
- **Versatilidad:** este sistema debe ser modular para poder adaptarse a muchos tipos de vehículos todoterreno que puedan recorrer distintos terrenos (arena, carretera, bosques...)
- **Adaptabilidad:** el sistema debe ser capaz de funcionar correctamente a pesar de las posibles limitaciones de ancho de banda.
- **Discreción:** estas antenas no pueden ser ni muy visibles visualmente ni a través de equipos electromagnéticos como los RADAR.
- **Ligereza:** no pueden suponer una carga importante al tener que ser cargado por vehículos con cantidades de combustible limitadas.

Para hacer frente a estos requerimientos tiene que afrontar y superar, además, las limitaciones inherentes a ser transportado por un vehículo. Las principales son:

- La dificultad que supone la instalación, sujeción y fijación de antenas, de forma segura y estable, desplegables sobre diferentes configuraciones de shelters y vehículos de distintas clases.
- La integración de los equipos electrónicos para alimentación y su adaptación a los espacios reducidos del shelter, especialmente las baterías más voluminosas y pesadas.
- La inestabilidad provocada por el movimiento del vehículo en toda clase de terrenos.

De esta forma, se podría resumir esta tecnología, como el medio que posibilita la capacidad de apuntar y mantener una conexión estable a un satélite desde un vehículo en cualquier clase de terreno y condiciones muy variadas.

Al haber sido desarrollados para aplicaciones militares, los terminales SOTM son, por necesidad, fiables y robustos, ya que deben ser capaces de operar en las condiciones más exigentes. Estos sistemas suelen utilizar antenas de baja RCS (Superficie Equivalente Radar) para ser más discretos y alta eficiencia, así como una unidad de posicionador y seguimiento de alta precisión, para hacer viable el uso del terminal en cualquier región.



Figura 2-23 Satcom On The Move [32]

En el Ejército de Tierra, para hacerlo más útil y poder proporcionar servicio a más unidades, utilizan radios ligeras UHF y HF *SpearNet* con el fin de completar la red de comunicaciones satelitales.

2.3.2.2.2 SATQH

La tecnología SATQH (*Satcom At The Quick Halt*, Comunicaciones por satélite en parado) es muy similar a la anterior en el sentido de que busca dotar un vehículo de la capacidad de establecer comunicaciones por satélite fiables, rápidas y seguras, cuando esté desplegado, aunque cuenta con una diferencia fundamental, está pensado para su uso en parado y no en movimiento como los SOTM.

Esta diferencia de enfoque provoca también grandes diferencias en su diseño y estructura ya que al tener más estabilidad y estar pensado para trabajar de forma fija habrá menos restricciones respecto a la carga de baterías y aporte de energía eléctrica y respecto a la sujeción y estabilidad de la antena, entre otras. Por tanto, estamos hablando de sistemas con antenas más grandes y de mayor precisión que las SOTM y que permiten disponer de un mayor ancho de banda y más cobertura.



Figura 2-24 Vehículo SATQH [33]

A continuación, para mayor claridad, estableceremos una breve comparativa de las dos tecnologías para facilitar la comprensión del tema.

	SOTM	SATQH
Peso	2000 kg	750 kg
Capacidad de despliegue	10 minutos	En movimiento
Tamaño de antenas	> 2 m	< 1 m
Capacidad transmisión	Multiportadora	Portadora simple

Figura 2-25 Comparativa SOTM-SATQH [2]

2.3.2.2.3 Plataformas a bordo

Aunque a priori puede resultar una tecnología muy compleja, técnicamente resulta más sencilla que la instalación de antenas de comunicaciones satelitales en vehículo. Esta tecnología es imprescindible tanto para la marina mercante como para la militar al ser un medio de geolocalización, comunicaciones fuera de alcance radar y un método fiable de transmisión de datos.

Como norma general, los satélites que se usan son los satélites GEO ya que ofrecen enormes áreas de cobertura y el seguimiento y apuntamiento resulta muy sencillo, como es el caso de

INMARSAT o HISDESAT, pero también hay casos de uso de satélites en otras órbitas como los LEO de Iridium.

Generalmente, se tratan de sistemas pequeños, compuestos fundamentalmente de tres elementos:

- **Antena:** comúnmente trabajan en banda X, pero se puede dar el caso de que lo hagan en banda Ka, o, las más modernas, en ambas al mismo tiempo.
- **Posicionador:** es el sistema que tiene en cuenta el movimiento en todos los ejes para recalcular el posicionamiento de la antena en dirección al satélite, en banda X ya está suficientemente extendido. No de la misma manera en bandas Ku y Ka, ya que requerirá de unas precisiones de apuntamiento de mucha mayor magnitud, al tener estas bandas el lóbulo principal más estrecho, para poder apuntar a un satélite GEO.
- **Radomo:** es el compartimento que aloja al resto de componentes. Es clave disponer de un radomo que proteja la antena ante inclemencias de la mar, humedad, salinidad, adversidades climatológicas y que, a su vez, no impida el paso de las señales electromagnéticas en ninguna de las bandas, o lo haga con unas pérdidas pequeñas.

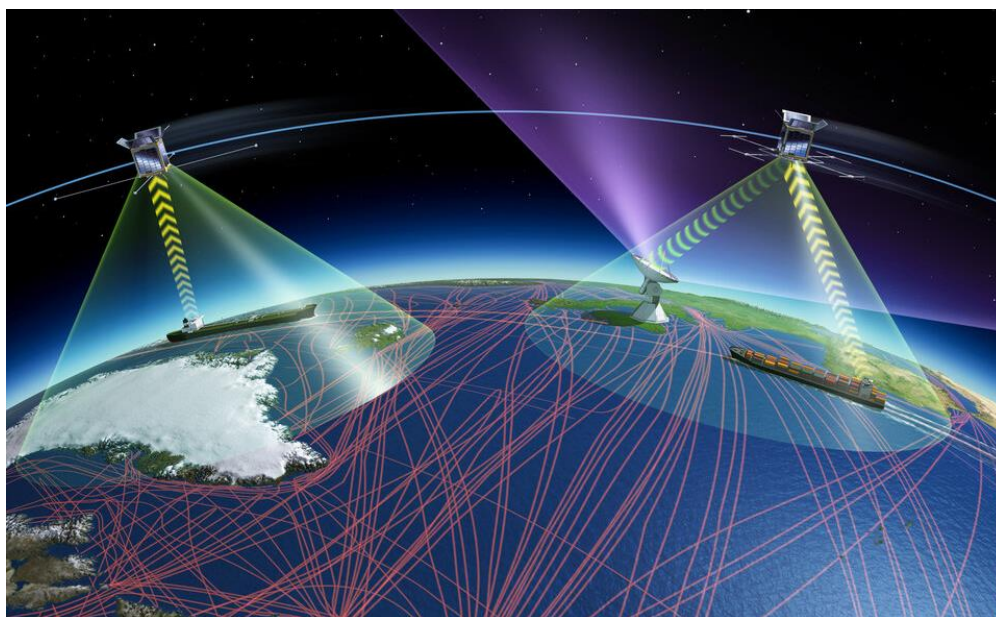


Figura 2-26 SATCOM en buques [29]

Nos centraremos en explicar este tipo de sistemas por medio de ejemplos extraídos de aquellos ingenios usados en las Fuerzas Armadas. La lista de elementos de comunicaciones satelitales usados en la Armada es la siguiente:

- **Terminales para buques:** TNX-50, TNX-100 y TNM-180
- **Terminales para vehículos:** SATQH, SOTP y SOTM
- **Terminales para submarinos:** TSUB40X
- **Terminales para aeronaves:** TAERO 45X, 60X/KA y H30KA

A continuación, procedemos a detallar las características técnicas del sistema TNX-100, instalado a bordo de las fragatas de la clase Álvaro de Bazán y otros buques de la Armada como los LPD Castilla y Galicia.

La terminal TNX-100 representa la parte superior de la gama de terminales de comunicaciones navales por satélite de Indra. Operando en banda X militar, puede trabajar a

través de una gran variedad de satélites como: SPAINSAT, SKYNET, SYRACUSE, SYCRAL, XTAR, OTAN...

Sus principales funcionalidades son:

- Subsistema de antena compuesto por dos antenas ubicadas en las dos posiciones más adecuadas para obtener la mejor cobertura satelital combinada sin trabas, eligiendo el haz satelital requerido para cada zona geográfica.
- Cobertura completa de las bandas militares de comunicaciones por satélite SHF
- Localización y seguimiento automáticos de satélites
- Comunicaciones asimétricas y bidireccionales
- Se pueden integrar diferentes módulos en el terminal TNX-100 dependiendo de las necesidades del usuario final
- Se admiten diferentes métodos de acceso (FDMA, TDMA o CDMA)
- Seguridad disponible a través de sistemas de cifrado y saltos de frecuencia
- Sistemas de seguridad: apagado de emergencia, apagado y corte rápido de comunicaciones



Figura 2-27 Fragata de la clase Álvaro de Bazán (Fuente: Armada Española) [35]

2.4 Programas de seguimiento de satélites

Después de haber explicado y desarrollado las tecnologías para el seguimiento y apuntamiento de satélites vamos a desarrollar brevemente algunos sistemas a nivel de software que ejecutan los cálculos y proporcionan al usuario, centrándonos en el enfoque de un radioaficionado, las herramientas necesarias para realizar el apuntamiento y seguimiento.

Existe una gran variedad de programas, tanto de pago como de software abierto, aplicaciones y páginas web para la obtención de los TLE, formato oficial de datos satelitales, que desarrollaremos en detalle más tarde en esta memoria, y que es usado para los cálculos de la gran mayoría de las plataformas para obtener datos satelitales y realizar la predicción de pases y seguimiento automático, similares a lo que se pretende desarrollar en esta memoria.

Para facilitar una visión general nos centraremos en aquellos de mayor relevancia, introduciendo diferentes tipos de aplicaciones y sistemas, en distintas plataformas: Internet, smartphones, ordenadores...

2.4.1 Aplicaciones Web

En este apartado trataremos los principales programas de predicción y seguimiento disponibles en la red.

2.4.1.1 CelesTrak

Es una página web [3] clave para el funcionamiento de la gran mayoría de programas de predicción y seguimiento de satélites ya que su función principal, entre muchas otras que trataremos brevemente, es la de proporcionar los de los que se nutren la mayoría de softwares.

Cabe destacar que CelesTrak obtiene los datos de la propia NORAD (North American Aerospace Defense Command) y que estos datos se actualizan con gran frecuencia (hasta varias veces al día para los satélites LEO diarios para los MEO y semanales para los GEO) para ofrecer la posibilidad de hacer seguimientos precisos y fiables.

A continuación, veremos una somera explicación de los principales servicios y funcionalidades que nos ofrece:

- **SOCRATES** (*Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space*): operando desde 2004, fue el primer sistema de predicción de conjunciones. Actualmente ofrece predicciones de hasta una semana con el fin de ayudar a evitar posibles colisiones entre satélites con órbitas cercanas.

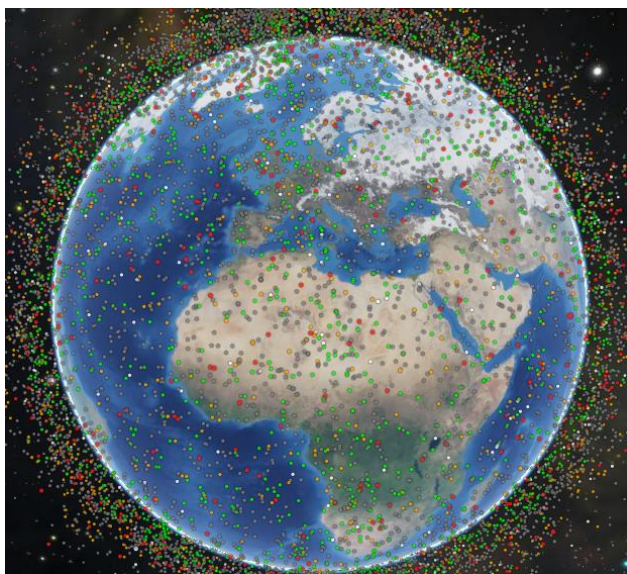


Figura 2-28 Presentación satelital (Fuente: CelesTrak [3])

- **Visualización orbital:** el programa nos muestra en pantalla la situación de una gran cantidad de satélites y elementos celestes con los datos más actualizados de una forma sencilla e interactiva (ver Figura 2-28)
- **Patrones de movimiento:** esta herramienta **CAT** (*Conjunction Analysis Tool*): nos permite predecir y calcular el movimiento de los satélites analizando sus órbitas y velocidades. Tiene una importancia vital a la hora de planificar tanto puestas en órbita como relocalizaciones de los cuerpos satelitales. Facilita además el dominio y comprensión SSA (*Spatial Situational Awareness*)
- **Proporcionar datos satelitales:** nutriéndose de datos de la NORAD, transforma los datos obtenidos a su propio formato para establecer referencias ordinales y facilitar

la SSA. Según el tipo de órbita lo tendrá que hacer con más frecuencia, varias veces al día, LEO, o con menos, semanalmente, GEO.

- **Calendario de visibilidad:** nos proporciona la información sobre los periodos de tiempo en los que podremos avistar y seguir los satélites según nuestra ubicación y época del año.

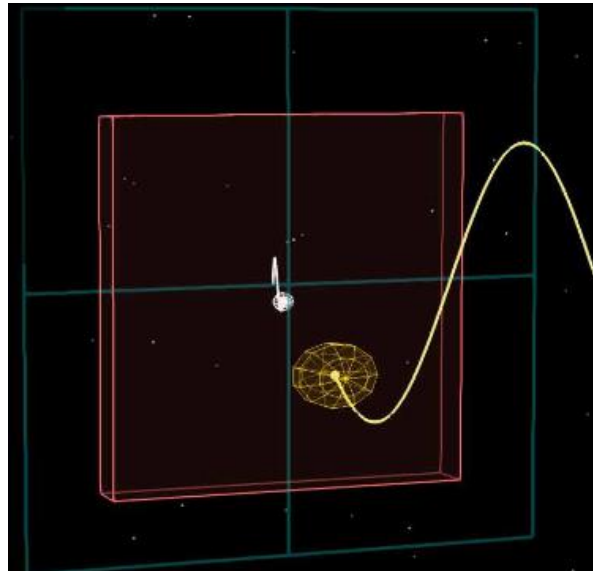


Figura 2-29 Patrones de movimiento (Fuente: Celestrak [3])

2.4.1.2 N2YO

N2YO [4] es una página web para el seguimiento en tiempo real de satélites, siendo prácticamente la única que te permite hacerlo sin necesidad de descargar ningún tipo de software. Solo será necesario concederle los permisos de ubicación para que pueda realizar los cálculos necesarios para predecir los pases y realizar el seguimiento.

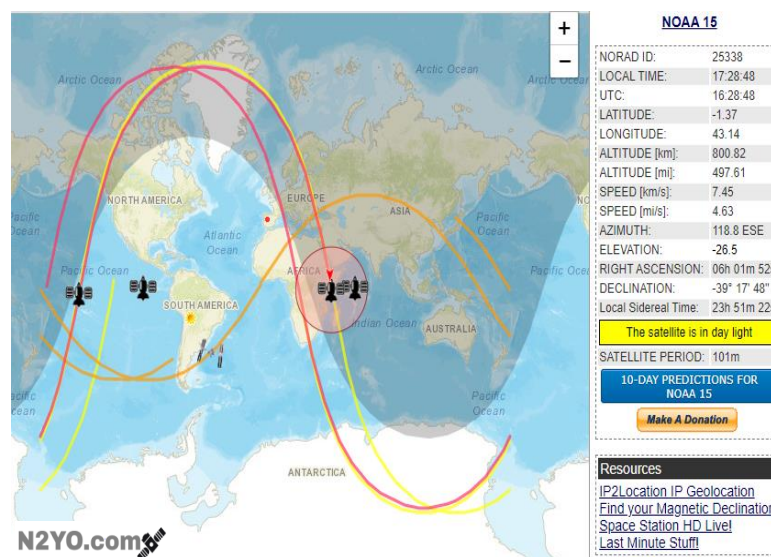


Figura 2-30 Presentación general (Fuente: N2YO) [4]

Esta página ofrece una gran variedad de satélites, los de la NOAA o la ISS (*International Spatial Station*, en inglés Estación Internacional Espacial) entre otros, permitiendo hacer el tracking de hasta 5 satélites simultáneamente sin necesidad de registrarse.

Otros servicios muy útiles que ofrece son:

- **Información detallada:** la página tiene una base de datos muy detallada y ofrece información amplia sobre todos los satélites.
- **Seguimiento permanente de la ISS:** no solo realiza el “tracking” si no que permite la descarga en tiempo real de las imágenes ofrecidas por la ISS.
- **Noticias de interés:** la página ofrece una serie de artículos y blogs enfocados principalmente a los radioaficionados.

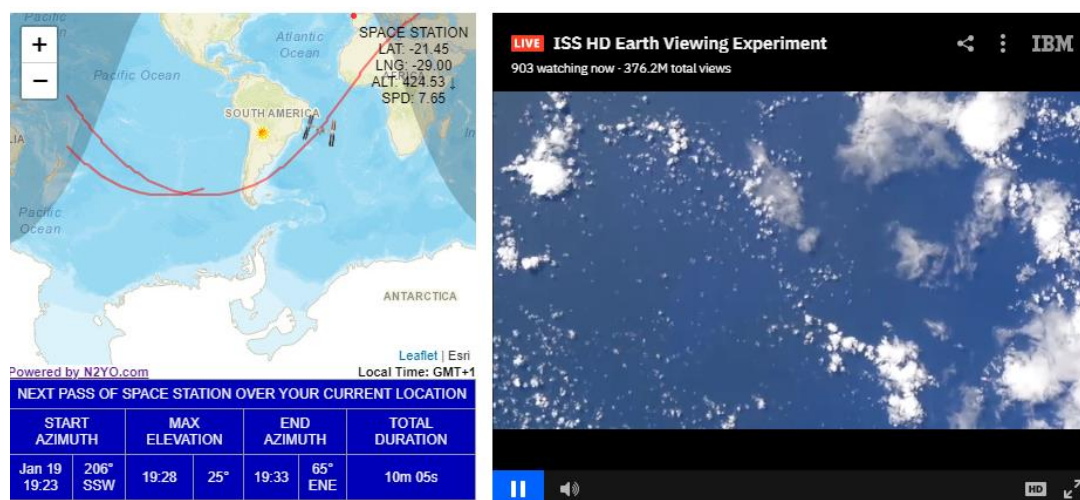


Figura 2-31 Herramienta de seguimiento de la ISS y visión de datos en streaming (Fuente: N2YO) [4]

2.4.1.3 AMSAT (Radio Amateur Satellites)

La página web [5] de la asociación de radioaficionados por satélite (por sus siglas en inglés, AMSAT) también nos ofrece dos aplicaciones web muy interesantes, una para la predicción del pase y otro para el seguimiento.

Son aplicaciones más simples y limitadas que la anteriormente mencionada, pero cumplen con el enfoque deseado, satisfacer a un radioaficionado.

Estas aplicaciones realizan la predicción de pases con solo con la latitud, longitud y elevación sobre el nivel del mar como entradas requeridas. Debajo, se muestra una imagen de la tabla de predicción de pases de la ISS sobre la Escuela Naval Militar el 20 de enero a las 09:00.

AMSAT Online Satellite Pass Predictions - ISS							
View the current location of ISS							
Date (UTC)	AOS (UTC)	Duration	AOS Azimuth	Maximum Elevation	Max El Azimuth	LOS Azimuth	LOS (UTC)
20 Jan 21	17:36:22	00:09:21	192	15	132	71	17:45:43
20 Jan 21	19:12:05	00:10:55	241	69	331	55	19:23:00
20 Jan 21	20:49:42	00:10:09	278	20	337	55	20:59:51
20 Jan 21	22:27:29	00:09:57	302	18	1	73	22:37:26
21 Jan 21	00:04:29	00:10:47	306	42	40	108	00:15:16
21 Jan 21	01:41:23	00:10:18	296	26	240	153	01:51:41
21 Jan 21	16:49:30	00:08:14	177	8	137	78	16:57:44
21 Jan 21	18:24:22	00:10:53	230	72	141	58	18:35:15
21 Jan 21	20:01:40	00:10:22	270	24	327	54	20:12:02
21 Jan 21	21:39:33	00:09:51	298	17	357	67	21:49:24

Tabla 2-4 Predicción de pases de la ISS sobre la ENM (Fuente: AMSAT) [5]

También, a través de otra aplicación web [6], la AMSAT nos ofrece la posibilidad de realizar el seguimiento de satélites dándole órdenes a los rotores de elevación y azimut mediante los softwares de Windows WispDDE y AVROT.

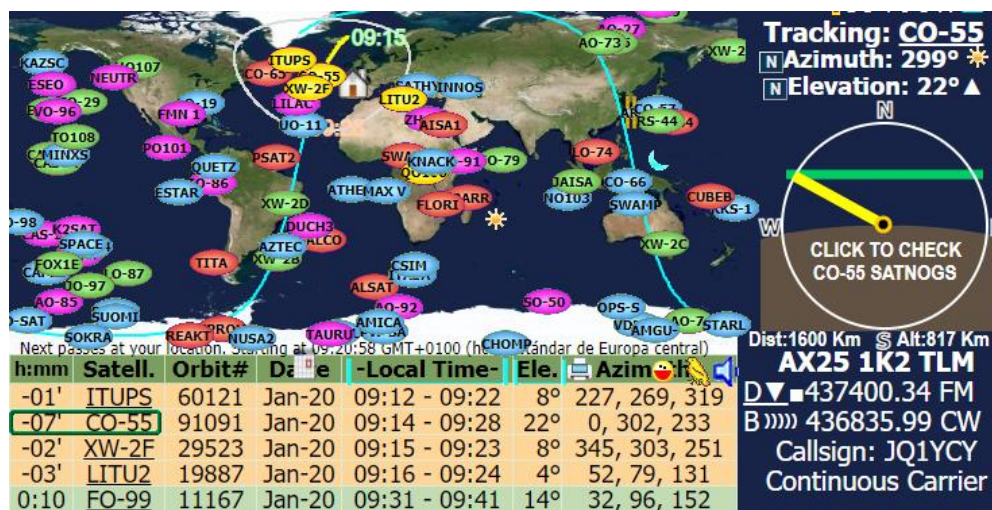


Figura 2-32 Aplicación de tracking (Fuente: AMSAT) [5]

2.4.1.4 SatNOGS

El proyecto SatNOGS [7] pretende ofrecer una plataforma completa de una red terrestre, abierta de forma libre y gratuita para todos los interesados, de estaciones satélite. El objetivo del proyecto es crear una pila completa de tecnologías abiertas basadas en estándares abiertos, y la construcción de una estación de tierra completa como escaparate del conjunto de antenas.

Para ello, se basan en la libre compartición de recursos de sistemas y antenas de los usuarios quienes, tras registrarse, pueden poner a disposición de cualquier usuario su antena para que estos programen los pases que deseen o compartir las paramétricas obtenidas.

Resumiendo, SatNOGS permite al usuario las herramientas para:

- Construcción y puesta en marcha de estaciones de seguimiento satélite baratas y ligeras.
- Diseño modular para la integración con tecnologías existentes y futuras.

- Una plataforma para disponer de una gran variedad de herramientas en el marco de las operaciones de los segmentos terrenos.
- Una red colaborativa de segmento terreno (uno a uno, uno a muchos, muchos a muchos), con enfoque basado en una comunidad abierta y colaborativa para el desarrollo de sistemas y antenas.
- Soluciones para la automatización masiva de estaciones terrestres sin operadores altamente cualificados y de forma accesible al gran público.

SatNOGS es una pila modular y escalable para la implementación de estaciones terrenas de predicción de pases y seguimiento de satélites. Totalmente basado en tecnologías de código abierto y estándares abiertos, proporciona interoperabilidad con subsistemas existentes o futuros.

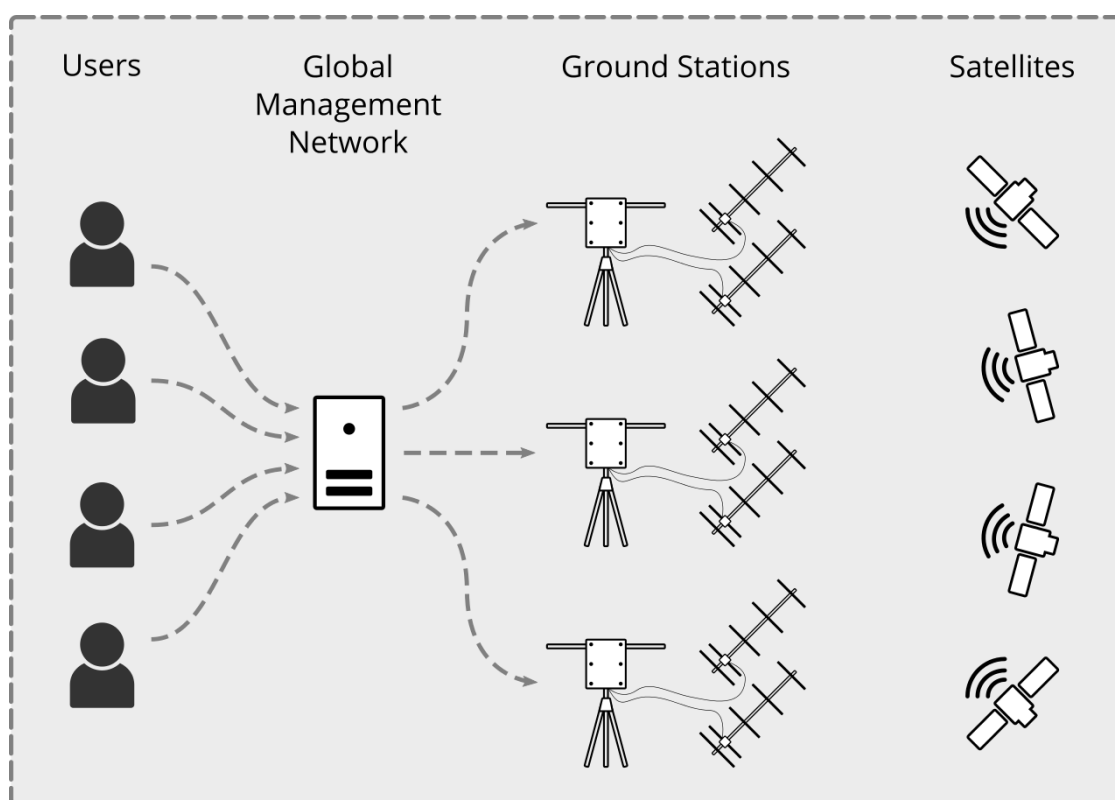


Figura 2-33 Red de SatNOGS (Fuente: SatNOGS) [7]

La estructura de red de acceso global es la parte clave de nuestra pila, conectando a múltiples observadores con múltiples estaciones terrestres que permiten el seguimiento y monitoreo de satélites desde múltiples ubicaciones en todo el mundo. Los datos recopilados serán accesibles al público a través del sitio web de la red.

Además, SatNOGS cuenta con 4 subproyectos para uso y disfrute de todos los usuarios:

- **SatNOGS Network:** el servidor de observaciones, programación de pases y seguimiento.
- **SatNOGS DB:** sitio web de información, noticias y sugerencias para el público.
- **SatNOGS Client:** sistema integrado que recibe la operación programada de la red, registra el pase y la información y la devuelve
- **SatNOGS Ground Station:** la parte que atañe a la instrumentación real de la estación terrestre con calculador, motores, antenas... Está conectado al cliente.

2.4.2 Aplicaciones para móviles y tablets

Las aplicaciones móviles centradas en los satélites son probablemente las mejores herramientas para rastrear la miríada de satélites visibles a simple vista. Nos ayudarán a, entre muchas otras funcionalidades, distinguir un satélite de otro, avisarnos cuando un cuerpo orbital esté a punto de aparecer en el cielo nocturno y mostrarnos exactamente dónde buscarlo.

Suelen ser aplicaciones con una interfaz sencilla y amigable para el usuario, destacan por su portabilidad y su versatilidad. Como únicas entradas suelen requerir conexión a Internet y ubicación GPS. A continuación, hablaremos de las más relevantes por encima.

2.4.2.1 Satellite Tracker

Satellite Tracker [8] es una potente aplicación gratuita, disponible para iOS y Android, que nos ayudará a observar todo tipo de satélites artificiales, cruzando el cielo nocturno. Entre sus capacidades está la de mostrar la posición exacta de cualquier satélite sobre la Tierra y enviar notificaciones cuando el satélite pasará sobre nuestra ubicación. La función integrada "Sky View" permite apuntar su dispositivo al cielo para ver el movimiento y la posición del satélite en tiempo real.

Se puede usar esta aplicación para:

- Seleccionar un satélite de la lista y ver su ubicación actual sobre la Tierra.
- Predicción de pases, consultando el tiempo restante para saber cuánto tiempo queda hasta que el satélite que está siguiendo vuela sobre nuestra ubicación. Además de obtener unas predicciones precisas sobre cuándo pasará el satélite por encima la próxima vez.
- Establecer alertas para recibir notificaciones sobre los pases.
- Elegir el modo "*Vista del cielo*" y apuntar nuestro dispositivo al cielo nocturno para ver la ubicación exacta del satélite en el cielo sobre nosotros.
- Ver imágenes 3D de satélites volando sobre la Tierra con la velocidad en tiempo real.

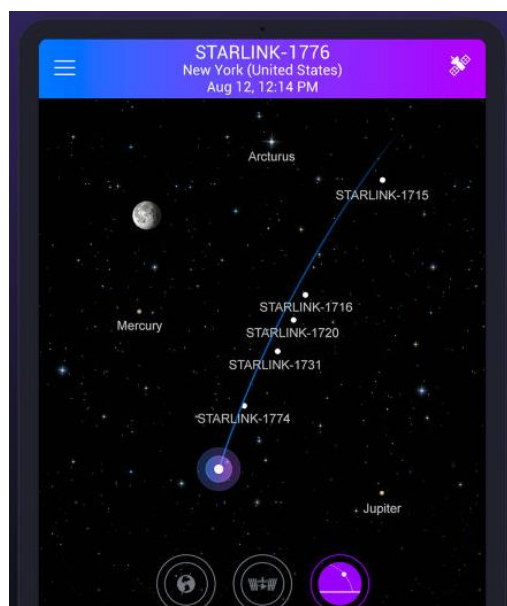


Figura 2-34 Interfaz de Satellite Tracker
(Fuente: Satellite Tracker) [8]

2.4.2.2 Orbitrack

La aplicación Orbitrack, para iOS y Android, ofrece muchas funcionalidades interesantes para la predicción de pases y seguimiento de satélites. Es un simulador de vuelo espacial y rastreador satelital de realidad aumentada. Entre las herramientas y servicios que nos ofrece podemos resaltar:

- Información sobre más de 4000 naves espaciales, incluidos todos los satélites activos, satélites militares clasificados, la Estación Espacial Internacional y los satélites de comunicación Starlink de SpaceX.
- Un modo de "realidad aumentada" que nos ayuda a encontrar satélites en el cielo utilizando el GPS y los sensores de movimiento de nuestro dispositivo. Funciona, además, con varios tipos de presentación.



Figura 2-35 Interfaz de Orbitrack (Fuente: Orbitrack) [36]

- Datos de radiofrecuencia para radioaficionados.
- Rastreo de miles de satélites. Orbitrack nos informa del momento en el que la nave espacial pase por encima, nos muestra dónde encontrarlos en el cielo, y nos permite rastrearlos a través del planeta.

2.4.2.3 NASA App

La aplicación de la NASA presenta una enorme colección de los últimos contenidos de la NASA, incluyendo más de 19.000 imágenes, videos bajo demanda, exploración del Sistema Solar, podcasts, noticias y reportajes, y herramientas de observación de la ISS y seguimiento, información de la misión y mucho más. La aplicación de la NASA está disponible de forma gratuita para iOS, Android, Apple TV, Kindle Fire, Fire TV y Roku.

Entre las funcionalidades que nos interesan, las relacionadas con el trabajo de esta memoria, podemos destacar:

- Obtener la lista actual de las próximas oportunidades de avistamiento, los pases, de la Estación Espacial Internacional (ISS) para nuestra ubicación.
- Seguimiento en tiempo real de la ubicación de la Estación Espacial Internacional y otros satélites en órbita terrestre de la NASA.

2.4.3 Aplicaciones de escritorio

2.4.3.1 Orbitron

Es una aplicación gratuita [9] (mediante el uso de cardware) para sistema operativo Windows y de código abierto que nos ofrece un sistema de rastreo satelital para fines de radioaficionado y observación principalmente, aunque también es utilizado por meteorólogos, operadores de comunicaciones por satélite o astrónomos.

Nos muestra las posiciones de los satélites, utilizando los modelos orbitales SGP4 y SGD4, en un momento dado (en tiempo real o simulado) a través de los datos TLE de CelesTrak con una capacidad para mostrar hasta 20.000 satélites simultáneamente. Es uno de los rastreadores de satélite más fáciles y potentes, del mercado y ofrece una experiencia sencilla al usuario.

Las principales funcionalidades que ofrece son:

- **Localización de satélites:** descargando los TLE, automática o manualmente, de la página web CelesTrak, nos permite representar las órbitas de todo tipo de satélites en una proyección sobre el mapamundi y permitiendo la opción de mostrar su área de cobertura
- **Predicción de pases y seguimiento de satélites:** introduciendo nuestra posición GPS el programa hace de forma automática los cálculos, con los keplerianos obtenidos de los TLE, necesarios para elaborar la predicción de pases y su duración y permitir el “tracking” durante su órbita en el área de cobertura. Esta función es válida para todo tipo de órbitas.
- **Conexión con Arduino:** a través del Arduino, les da a los rotores la información necesaria para el apuntamiento de la antena de forma continua. Esta funcionalidad es similar a la que se pretende desarrollar en este proyecto.
- **Diferentes tipos de visualizaciones:** para facilitar la comprensión al usuario y un mejor manejo de los datos. Principalmente dos tipos: radar y mapamundi.
- **Datos sobre los objetos:** tiene una gran base de datos para ofrecer al usuario información sobre una gran mayoría de los cuerpos satelitales mostrados.

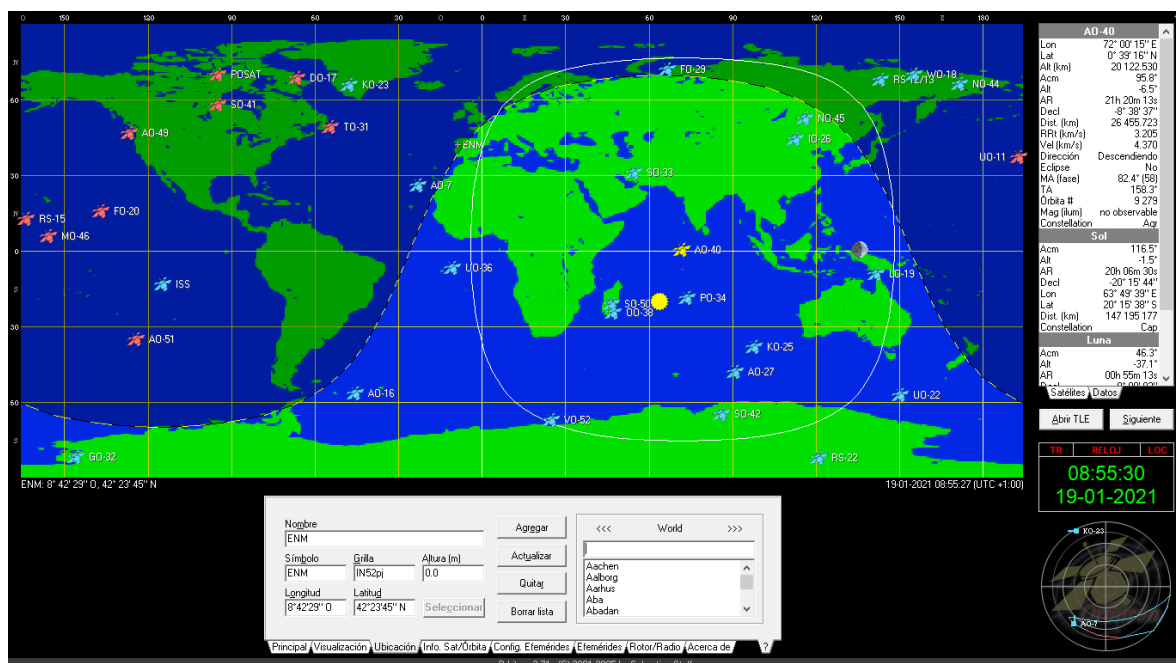


Figura 2-36 Presentación sobre mapamundi (Fuente: Orbitron)

2.4.3.2 GPredict

GPredict es un programa gratuito con versión para Windows, Mac OS x y GNU/Linux de código abierto. Es el programa más potente en el ámbito de predicción de pases y seguimiento. Está pensado tanto para radioaficionados como para profesionales. Entre otras cosas puede rastrear una gran cantidad de satélites y mostrar su posición además de otros datos en listas, tablas, mapas y diagramas polares.

Gracias al programa vamos a poder obtener información muy detallada sobre cada uno de los pases, además, una funcionalidad que le distingue de software similares es la de agrupar satélites en módulos de visualización.

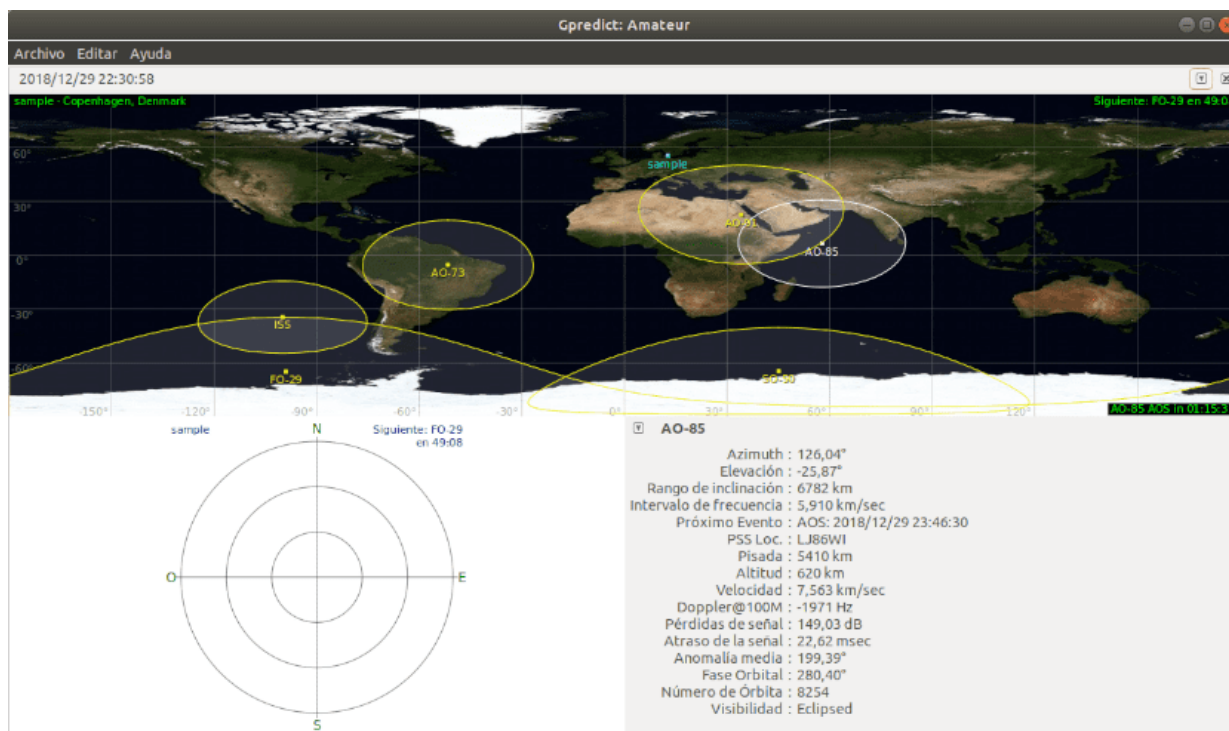


Figura 2-37 Presentación inicial (Fuente: GPredict) [18]

Para todo lo anteriormente mencionado, GPredict se nutre de datos que se descargan y actualizan continuamente obtenidos de CelesTrak.

Resumiendo, las principales funcionalidades que ofrece el programa son:

- **Implementación multiplataforma:** es válido para los sistemas operativos modernos más usados: Mac OS X, Windows, GNU/Linux, BSD... en varias de sus versiones.
- **Predicción de pases y seguimiento de satélites:** en tiempo real (o simulado), de forma rápida y precisa, haciendo uso de los modelos orbitales de la NORAD, el SGP4 y el SGD4.
- **Número de satélites en seguimiento:** no limitado por el software.

2021/01/19 15:20:21						
Satellite ▲	Az	El	Dir	Range	Next Event	Alt
AO-73	204.31°	-78.31°	↑	13085	AOS: 2021/01/19 15:58:27	615
AO-85	134.57°	-21.72°	↓	6353	AOS: 2021/01/19 16:37:43	774
AO-91	76.89°	-62.22°	↓	12120	AOS: 2021/01/19 23:06:50	752
FO-29	240.17°	-29.49°	↓	7693	AOS: 2021/01/19 20:00:36	808
ISS	92.49°	-13.26°	↓	4238	AOS: 2021/01/19 18:23:29	420
SO-50	349.26°	-32.86°	↓	7992	AOS: 2021/01/19 21:29:33	633

Figura 2-38 Lista de satélites visibles (Fuente: GPredict) [18]

- **Cálculos orbitales:** azimut, elevación, velocidad, inclinación o fecha del próximo pase entre otros muchos.

▼	AO-73
Azimuth : 271.36°	
Elevation : -78.22°	
Slant Range : 13101 km	
Range Rate : 0.718 km/sec	
Next Event : AOS: 2021/01/19 15:58:26	
SSP Loc. : BF01FS	
Footprint : 5404 km	
Altitude : 618 km	
Velocity : 7.558 km/sec	
Doppler@100M : -240 Hz	
Sig. Loss : 154.75 dB	
Sig. Delay : 43.70 msec	
Mean Anom. : 211.52°	
Orbit Phase : 297.46°	
Orbit Num. : 38539	
Visibility : Daylight	

Figura 2-39 Cálculos orbitales (Fuente: GPredict) [18]

- **Control de rotores o motores:** permite dar órdenes con la frecuencia deseada a los rotores de la antena para que estén apuntando continuamente en dirección a los satélites, generalmente se hará por medio de un Arduino.
- **Diferentes tipos de presentaciones:** para la mejor comprensión y presentación de los datos.

2.4.3.3 SATPC32

Este software distribuido por la AMSAT es el único de pago que valoraremos en el desarrollo de este proyecto. Por el precio de 45 o 50 €, distinguiendo la tarifa de miembros y la de no miembros, podemos adquirir este programa que permite realizar los cálculos satelitales deseados, siguiendo los modelos orbitales SGP4 y SGD4 de la NORAD, descargando los TLE de CelesTrak.

Una herramienta muy interesante es la modulación del *Uplink* para compensar el efecto Doppler además del control de los rotores para el apuntamiento automático de la antenna. Este programa es válido para prácticamente cualquier radio y antenna.

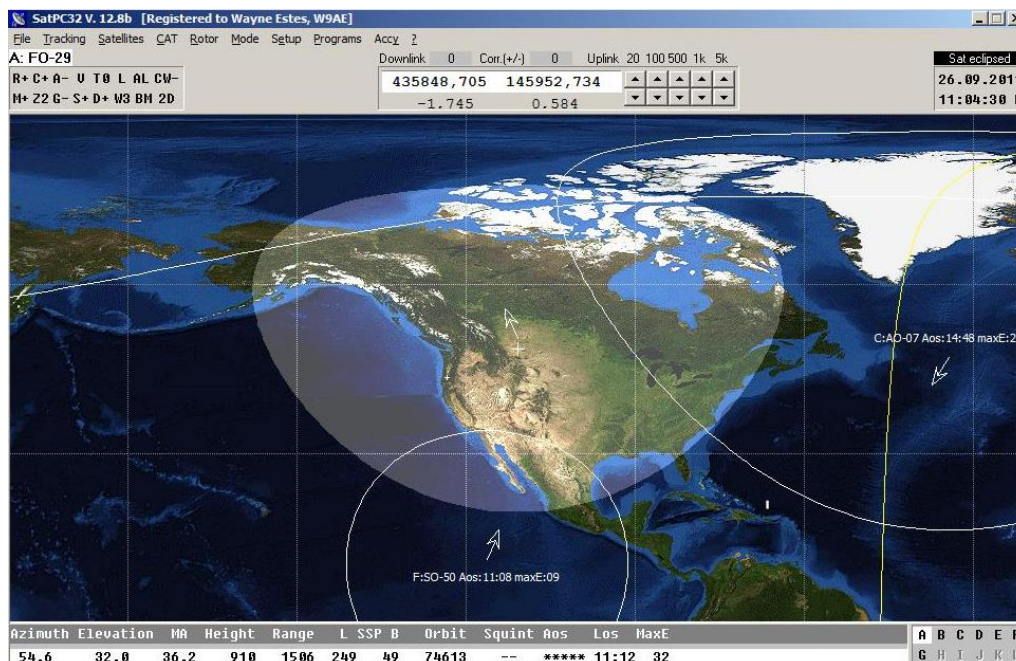


Figura 2-40 Presentación general (Fuente: SATPC32)

De manera breve las funcionalidades que ofrece son:

- **Predicción de pases:** de prácticamente la totalidad de satélites en órbita.
- **Seguimiento de satélites:** con prácticamente cualquier tipo de antenna y radio.
- **Comunicación full-dúplex:** ofrece la posibilidad de recibir y transmitir, permitiendo la descarga de datos y el uso de los satélites como medio de comunicación entre usuarios.
- **Apuntamiento automático:** el programa tiene instalado por defecto el software necesario para dar órdenes a los rotores.

Como conclusión, se puede decir que este es de los programas más completos del mercado y que su único inconveniente es el precio, sin embargo, se ofrece una demo gratuita con prácticamente todas las herramientas a disposición.

3 FUNDAMENTOS TEÓRICOS

3.1 Descripción del apartado

En el presente apartado se desarrollarán con detalle los conceptos teóricos necesarios para comprender el desarrollo del trabajo y las soluciones adaptadas.

Desglosando dicho desarrollo, en una trayectoria que va de lo más general a lo más específico, se comenzará con una breve introducción a la mecánica orbital, hablando de las leyes de Kepler y los 6 elementos orbitales básicos, seguiremos con los cálculos de la elevación y el azimut para, a continuación, finalizar hablando de los formatos de datos satelitales empleados normalmente, desarrollando con más detalle el formato TLE.

3.2 Leyes de Kepler

Las leyes de Kepler del movimiento planetario son las leyes que describen los movimientos de los planetas en el Sistema Solar y que fueron enunciadas por el físico alemán Johannes Kepler.

Estas tres leyes, punto de partida de la mecánica orbital y la astrofísica son:

1. **Ley de las órbitas:** “Todos los planetas se mueven en órbitas elípticas, con el sol en un foco”. A partir de esta ley, definiremos más tarde el concepto de excentricidad
2. **Ley de las áreas:** “Los planetas se mueven con velocidad areolar ⁷ constante. Es decir, el vector posición r de cada planeta con respecto al Sol barre áreas iguales en tiempos iguales”.

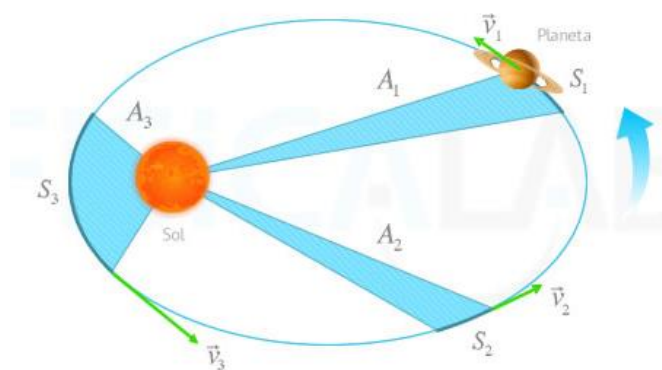


Figura 3-1 2ª Ley de Kepler

⁷ Se define la velocidad areolar v_A como el área barrida por el vector de posición de un cuerpo por unidad de tiempo.

3. **Ley de los periodos:** “El cuadrado del período de cualquier planeta es proporcional al cubo del semieje mayor de su órbita.” (En la siguiente ecuación, T es el período y a, el semieje mayor).

$$\frac{T^2}{a^3} = C = \text{constante}$$

La utilidad de las leyes de Kepler se extiende a los movimientos de los satélites naturales y artificiales, así como a los sistemas estelares y planetas extrasolares. Según lo formulado por Kepler, aunque no tiene en cuenta las interacciones gravitacionales (como efectos perturbadores) de los diversos planetas entre sí.

El problema general de predecir con precisión los movimientos de más de dos cuerpos bajo sus atracciones mutuas es bastante complicado; las soluciones analíticas del problema de los tres cuerpos son imposibles de lograr excepto en algunos casos especiales. Cabe señalar que las leyes de Kepler se aplican no sólo a la gravitación, sino también a todas las demás fuerzas de ley cuadrática inversa y, si se tiene debidamente en cuenta los efectos relativistas y cuánticos, a las fuerzas electromagnéticas dentro del átomo.

A raíz de estas tres leyes fundamentales, Isaac Newton desarrollaría su Ley de Gravitación Universal, derivando a partir de la 2ª Ley fundamentalmente.

3.3 Mecánica orbital

La mecánica orbital, también llamada mecánica de vuelo, es el estudio de los movimientos de satélites artificiales y vehículos espaciales que se mueven bajo la influencia de fuerzas tales como la gravedad, la resistencia atmosférica o el empuje, entre otros.

La raíz de la mecánica orbital se remonta al siglo XVII gracias a los avances y leyes enunciados por estudiosos como Johannes Kepler, del que hemos hablado con anterioridad, o Isaac Newton, que presentó sus leyes del movimiento y formuló su Ley de la Gravitación Universal. Sus aplicaciones de ingeniería de la mecánica orbital incluyen trayectorias de ascenso, reentrada y aterrizaje, cálculos de encuentro y trayectorias lunares e interplanetarias.

Las leyes del movimiento de Newton describen la relación entre el movimiento de una partícula y las fuerzas que actúan sobre ella.

La **primera ley** establece que, “si ninguna fuerza está actuando, un cuerpo en reposo permanecerá en reposo, y un cuerpo en movimiento permanecerá en movimiento en línea recta”. Por lo tanto, si no hay fuerzas actuando, la velocidad (tanto la magnitud como la dirección) permanecerá constante.

La **segunda ley** nos dice que “si se aplica una fuerza habrá un cambio en la velocidad, una aceleración, proporcional a la magnitud de la fuerza y en la dirección en la que se aplica la fuerza”. Esta ley puede ser resumida por la ecuación:

$$F = ma$$

La **tercera ley** establece que, “si el cuerpo 1 ejerce una fuerza sobre el cuerpo 2, entonces el cuerpo 2 ejercerá una fuerza de igual fuerza, pero opuesta en dirección, en el cuerpo 1”. Esta ley se afirma comúnmente, “para cada acción hay una reacción igual y opuesta”.

Finalmente, en su **Ley de Gravitación Universal**, Newton afirma que “dos partículas con masas m_1 y m_2 y separadas por una distancia r se atraen entre sí con fuerzas iguales y opuestas dirigidas a lo largo de la línea que une las partículas. La magnitud común F de las dos fuerzas es:

$$F = G\left(\frac{m_1 m_2}{r^2}\right)$$

Donde G es una constante universal, llamada la constante de gravitación. A partir de esta ley se derivarán todas las demás leyes empleadas en mecánica orbital.

Examinemos ahora el caso de dos cuerpos de masas M y m moviéndose en órbitas circulares bajo la influencia de la atracción gravitatoria el uno del otro. El centro de masa de este sistema de dos cuerpos se encuentra a lo largo de la línea que los une en un punto C (de forma tal que $m \cdot r = M \cdot R$).

El cuerpo mayor, de masa M , se mueve en una órbita de radio constante R y el cuerpo pequeño, de masa m , en una órbita de radio constante r , ambos con la misma velocidad angular. Para que esto suceda, la fuerza gravitatoria que actúa sobre cada cuerpo debe proporcionar la aceleración centrípeta necesaria. Dado que estas fuerzas gravitacionales son un simple par acción-reacción, las fuerzas centrípetas deben ser iguales pero opuestas en dirección.

Es decir, $m \omega^2 r$ debe ser igual a $M \omega^2 R$. El requisito específico, entonces, es que la fuerza gravitacional que actúa sobre cualquier cuerpo debe ser igual a la fuerza centrípeta necesaria para mantenerlo en movimiento en su órbita circular, es decir:

$$\frac{GMm}{(R + r)^2} = m\omega^2 r$$

A continuación, desarrollaremos brevemente los elementos necesarios para caracterizar la órbita de los cuerpos satelitales.

3.4 Secciones cónicas

Una sección cónica, es una curva formada por pasar un plano a través de un cono circular. La orientación angular del plano con respecto al cono determina si la sección cónica es un círculo, elipse, parábola o hipérbola.

El círculo y la elipse surgen cuando la intersección de cono y plano es una curva acotada. El círculo es un caso especial de la elipse en la que el plano es perpendicular al eje del cono. Si el plano es paralelo a una línea de generador del cono, la cónica se llama parábola. Finalmente, si la intersección es una curva sin límites y el plano no es paralelo a una línea generadora del cono, la figura es una hipérbola. En este último caso el plano intersecará ambas mitades del cono, produciendo dos curvas separadas. Podemos verlo en la Figura 3-2.

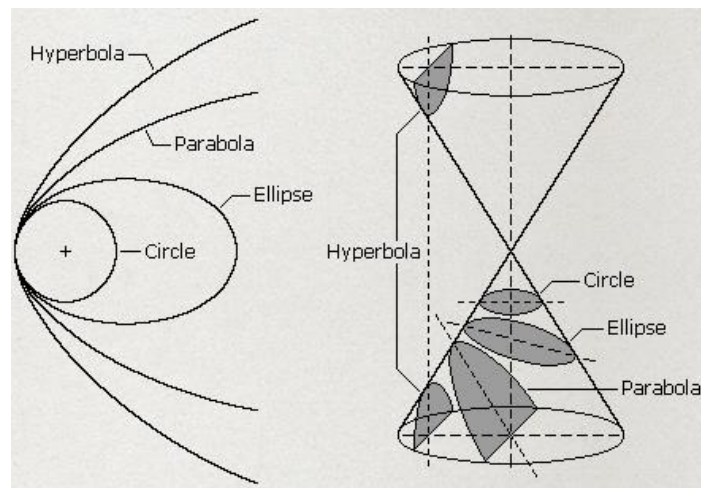


Figura 3-2 Secciones cónicas

De aquí se deriva el término de excentricidad, que se define como un parámetro adimensional que determina la cantidad en que su órbita alrededor de otro cuerpo se desvía respecto a un círculo perfecto (Tabla 3-1).

Sección cónica	Excentricidad
Círculo	0
Elipse	$0 < e < 1$
Parábola	∞
Hipérbola	< 0

Tabla 3-1 Sección cónica- Excentricidad

3.5 Elementos orbitales

Para describir matemáticamente una órbita uno debe definir seis cantidades, llamadas elementos orbitales. Son los siguientes:

1. **Semieje mayor (a)**
2. **Excentricidad (e)**
3. **Inclinación (i)**
4. **Argumento del periastro (ω)**
5. **Tiempo de paso por el periastro (T)**
6. **Longitud del nodo ascendente (Ω)**

Un satélite en órbita sigue una trayectoria ovalada, una sección cónica, conocida como elipse con el cuerpo sobre el que órbita, llamado primario, localizado en uno de los focos.

Una elipse se define como una curva con la siguiente propiedad: para cada punto de una elipse, la suma de sus distancias desde los focos, es constante. Las líneas más largas y cortas que se pueden dibujar a través del centro de una elipse se llaman eje mayor y eje menor, respectivamente.

El **semieje mayor** es la mitad del eje mayor y representa la distancia media de un satélite a su primario.

La **excentricidad**, ya definida anteriormente también se puede extraer de esta explicación como: la distancia entre los focos dividida por la longitud del eje mayor.

La **inclinación** mide la inclinación de la órbita de un cuerpo orbital respecto a la de otro cuerpo celeste. Se expresa como el ángulo entre un plano de referencia, en nuestro caso dicho plano horizontal será el ecuador celeste, y el plano orbital del objeto en órbita. Este ángulo solo tiene valores de 0° a 90° y tiene signo positivo o negativo, lo que distingue si el astro es visible para el observador o no.

El periastro es el punto en una órbita más cercana a la primaria y su opuesto de periastro, el más lejano, se llama apoastro. A partir de estos términos, periastro y apoastro, se denominan otros tales como perihelio y afelio para el Sol, perigeo y apogeo para la Tierra o perilunio y apolunio para la Luna.

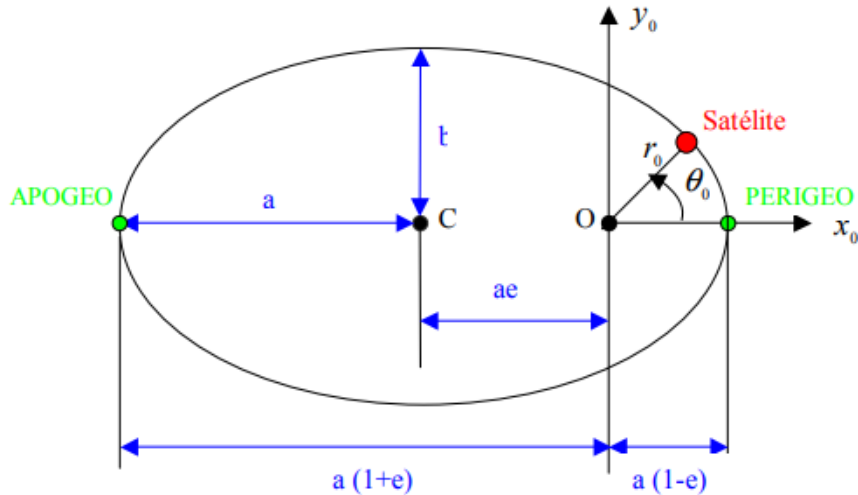


Figura 3-3 Órbita situada en el plano orbital

El **argumento de periastro** es la distancia angular entre el nodo ascendente y el punto de periastro. El ángulo se mide en el plano orbital y en la dirección del movimiento.

El **tiempo de paso del periastro** es el tiempo en que un satélite se mueve a través de su punto de periastro.

Los nodos son los puntos donde una órbita cruza un plano, como un satélite que cruza el plano ecuatorial de la Tierra. Si el satélite cruza el plano de sur a norte, el nodo es el nodo ascendente; si se mueve de norte a sur, es el nodo descendente.

La **longitud del nodo ascendente** es la longitud celestial del nodo. Se define como el ángulo desde una dirección de referencia específica, en nuestro caso el cardinal Norte, llamada origen de longitud, a la dirección del nodo ascendente, medida en un plano de referencia específico, el plano orbital del cuerpo en este caso. El nodo ascendente es el punto donde la órbita del objeto pasa por el plano de referencia. Tiene un valor comprendido entre 0 y 360.

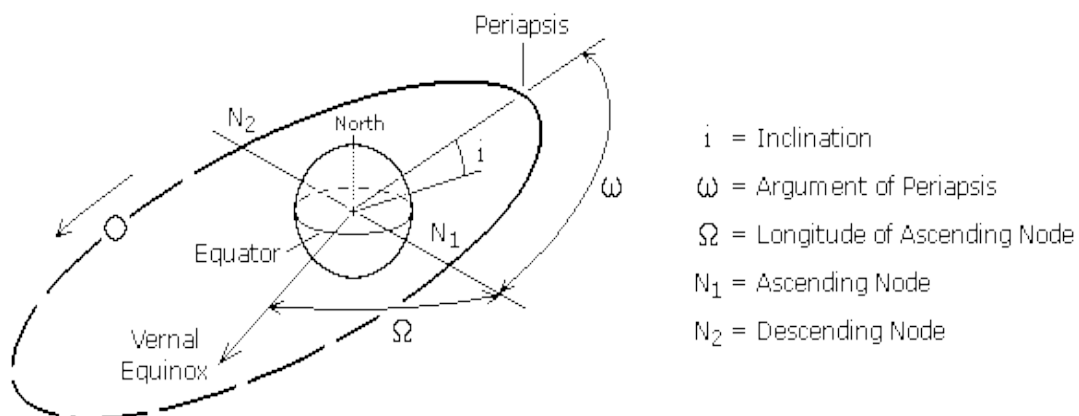


Figura 3-4 Elementos orbitales [37]

Johannes Kepler fue capaz de resolver el problema de relacionar la posición en una órbita con el tiempo transcurrido, o inversamente, cuánto tiempo se tarda en ir de un punto en una órbita a otro.

Para resolver esto, Kepler introdujo la cantidad M , llamada **anomalía media**, que es una fracción de un período orbital que ha transcurrido, expresada como ángulo.

Por definición:

$$M - M_o = n(t - t_o)$$

Donde M_o es la anomalía media en el tiempo inicial y n es el movimiento medio, la velocidad angular de un astro en una órbita elíptica, medido en grados o radianes por día., determinada a partir del semieje mayor de la órbita como sigue:

$$n = \sqrt{\frac{GM}{a^3}}$$

Esta solución dará la posición y la velocidad media.

Sin embargo, las órbitas de los satélites son elípticas con un radio que varía constantemente en la misma órbita.

Debido a que la velocidad del satélite depende de este radio variable, también cambia. Para resolver este problema podemos definir una variable intermedia E , llamada **anomalía excéntrica**, para órbitas elípticas, que viene dada por:

$$\cos(E) = \frac{e + \cos(v)}{1 + e \cos(v)}$$

Donde v es la **anomalía verdadera**, ángulo que forman las líneas foco-satélite y foco-periastro. La anomalía media también se relaciona con la anomalía excéntrica de la siguiente manera:

$$M = E - e \sin(E)$$

De esta manera, podemos apreciar que las ecuaciones anteriores se pueden utilizar para:

- Calcular el tiempo que se necesita para ir de una posición en la órbita a otra.
- Hallar la posición en la órbita después de un período específico de tiempo.

Para finalizar este subapartado, vamos a ver que podemos calcular para cualquier momento de su órbita, la magnitud del vector de posición de cuerpo orbital, su distancia al cuerpo primario, y su ángulo de trayectoria de vuelo, a partir de las siguientes ecuaciones:

$$r = \frac{a(1 - e^2)}{1 + e \cos(v)}$$

$$\phi = \arctan\left(\frac{e \sin(v)}{1 + e \cos(v)}\right)$$

$$v = \sqrt{GM\left(\frac{2}{r} - \frac{1}{a}\right)}$$

Donde ϕ es la **latitud geográfica**, ángulo definido por la intersección del elipsoide normal de referencia a través del punto de interés y el plano ecuatorial verdadero.

3.5.1 Perturbaciones orbitales

Aunque los elementos orbitales discutidos al principio de esta sección proporcionan una excelente referencia para describir órbitas, no nos permiten hacer cálculos exactos si no aproximaciones debido a que hay otras fuerzas que actúan en un satélite que lo perturban lejos de su órbita nominal.

Estas perturbaciones, se pueden clasificar en función de cómo afectan a los elementos keplerianos:

- Las **variaciones seculares** representan una variación lineal en el elemento.
- Las **variaciones de período corto** son periódicas en el elemento con un período menor que el período orbital.
- Las **variaciones de período largo** son aquellas con un período mayor que el período orbital.

Dado que las variaciones seculares tienen efectos a largo plazo, en la predicción de la órbita (los elementos orbitales afectados siguen aumentando o disminuyendo), se examinarán aquí para los satélites en órbita terrestre. Las principales son:

- **Perturbaciones por la acción de un tercer cuerpo:** Las fuerzas gravitacionales del Sol y la Luna causan variaciones periódicas en todos los elementos orbitales, pero sólo la longitud del nodo ascendente, el argumento del perigeo y la anomalía media experimentan variaciones seculares. La variación en la anomalía media es mucho más pequeña que el movimiento medio y tiene poco efecto en la órbita, sin embargo, las variaciones, en la longitud del nodo ascendente y el argumento del perigeo son importantes, especialmente para las órbitas GEO.
- **Perturbaciones debido a que la Tierra no es una esfera perfecta:** al desarrollar las ecuaciones del movimiento de dos cuerpos, hemos asumido que la Tierra era una masa esférica simétrica y homogénea. Sin embargo, la Tierra no es ni homogénea ni esférica. Las características más dominantes son una protuberancia en el ecuador, y aplanamiento en los polos. Para una función potencial de la Tierra, podemos encontrar la aceleración de un satélite tomando el gradiente de la función potencial. La forma más ampliamente utilizada de la función geopotencial depende de la latitud y los coeficientes geopotenciales, J_N , llamados los coeficientes zonales. El potencial generado por la Tierra no esférica causa variaciones periódicas en todos los elementos orbitales. Los efectos dominantes, sin embargo, son variaciones seculares en la longitud del nodo ascendente y el argumento del perigeo debido al achatamiento de la Tierra, representada por el término J_2 en la expansión geopotencial.
- **Perturbaciones por resistencia aerodinámica:** un cuerpo orbital estará sujeto a fuerzas de arrastre cuando se mueva a través de la atmósfera de un planeta. Esta resistencia es mayor durante el lanzamiento y la reentrada, sin embargo, incluso un vehículo espacial en órbita terrestre baja experimenta cierta resistencia a medida que se mueve a través de la delgada atmósfera superior de la Tierra. Con el tiempo, la acción de la resistencia en un vehículo espacial hará que vuelva en espiral a la atmósfera, e incluso, con mucho tiempo, para desintegrarse o quemarse.

- **Perturbaciones por radiación solar:** la presión de radiación solar causa variaciones periódicas en todos los elementos orbitales. En el caso de los satélites LEO, la aceleración debida a la resistencia atmosférica es mayor que la debida a la presión de la radiación solar; para los MEO y GEO, la aceleración debida a la presión de la radiación solar es mayor.

3.6 Coordenadas orbitales

En este subapartado describiremos como definir la trayectoria de un satélite de manera que podamos realizar un seguimiento y apuntamiento preciso.

Para ello, empezaremos hablando de los sistemas de coordenadas, los ángulos de visión, azimut y elevación, definiremos el concepto de punto sub-satélite y terminaremos explicando los cálculos que realiza nuestro sistema para calcular los ángulos de visión y definir los ángulos de visión.

3.6.1 Sistemas de coordenadas

Lo primero es establecer el sistema de coordenadas, las coordenadas celestes, conjunto de valores que dan la capacidad de ubicar un cuerpo en la esfera celeste. Se pueden distinguir según se obtenga la referencia por su origen o plano de referencia. Una primera clasificación las divide en dos grandes grupos, según se trate de coordenadas cartesianas o coordenadas esféricas. En esta memoria emplearemos los dos tipos a la hora de obtener resultados.

- **Coordenadas cartesianas:** se basan en tres ejes (x y z) perpendiculares entre sí que se cruzan en un punto origen (O) que en astronomía suele interesar que sea la Tierra (coordenadas geocéntricas), aunque puede ser el Sol (heliocéntricas).

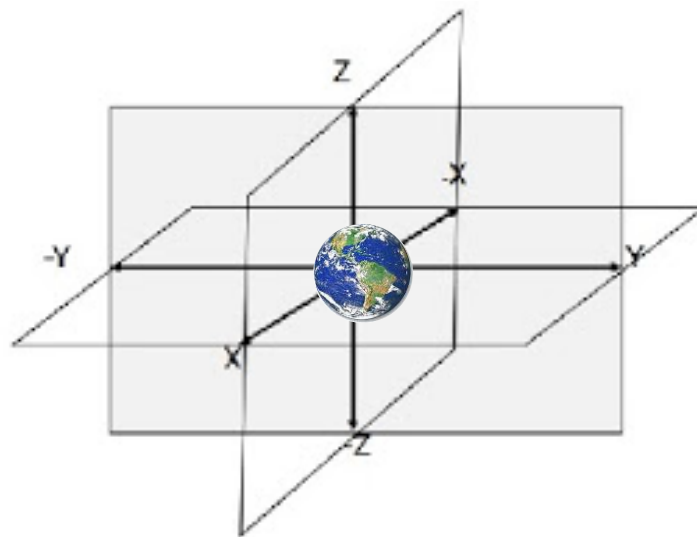


Figura 3-5 Coordenadas cartesianas geocéntricas

- **Coordenadas esféricas o polares:** en este sistema se determina la posición de los astros conociendo dos ángulos: azimut (α) y elevación (β); y una distancia (d).

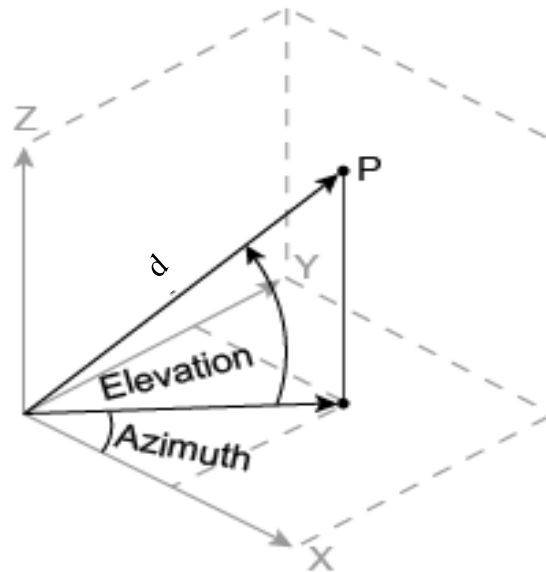


Figura 3-6 Coordenadas polares

En nuestro caso, aunque trabajaremos sobre los dos tipos de sistemas, nos basaremos en un sistema con origen de coordenadas en el observador, la plataforma, con el plano de referencia ecuatorial, donde los polos serán el cénit y el nadir (punto opuesto al cénit) y las referencias cardinales serán los puntos cardinales en el horizonte.

3.6.2 Ángulos de visión

Con las coordenadas polares, definimos también los **ángulos de visión**: el ángulo de azimut y el ángulo de elevación, que serán aquellos a los que debamos apuntar la antena cuando deseemos realizar el seguimiento del satélite.

- **Ángulo de azimut (Az o α):** dirección de un objeto celeste desde el observador, expresada como la distancia angular desde el punto norte del horizonte hasta el punto en que un círculo vertical que pasa por el objeto interseca el horizonte.

- **Ángulo de elevación (El o β):** la elevación es la distancia angular vertical entre un cuerpo celeste (sol, luna) y el horizonte local del observador o, también llamado, el plano local del observador.

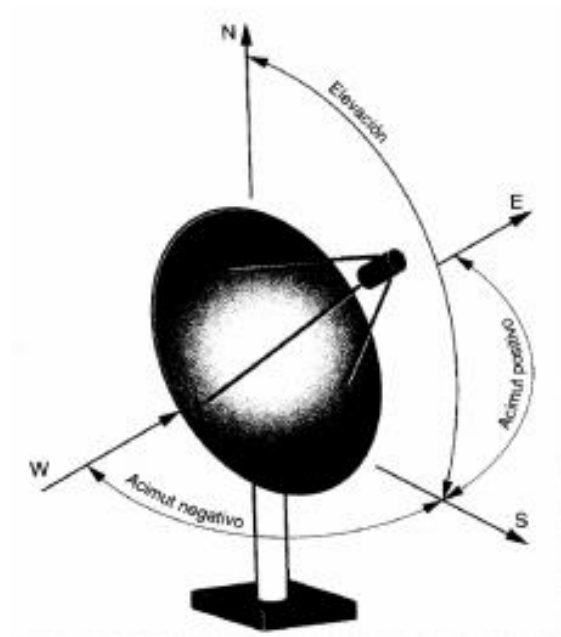


Figura 3-7 Ángulos de visión

3.6.3 Punto sub-satélite

El punto sub-satélite, o trayectoria terrestre, se define como la trayectoria de un avión o un satélite proyectada en la superficie de la Tierra.

En el caso de los satélites, también se le llama “*ground track*”. Se puede considerar como un camino que traza el movimiento de una línea imaginaria a lo largo de la superficie de la Tierra.

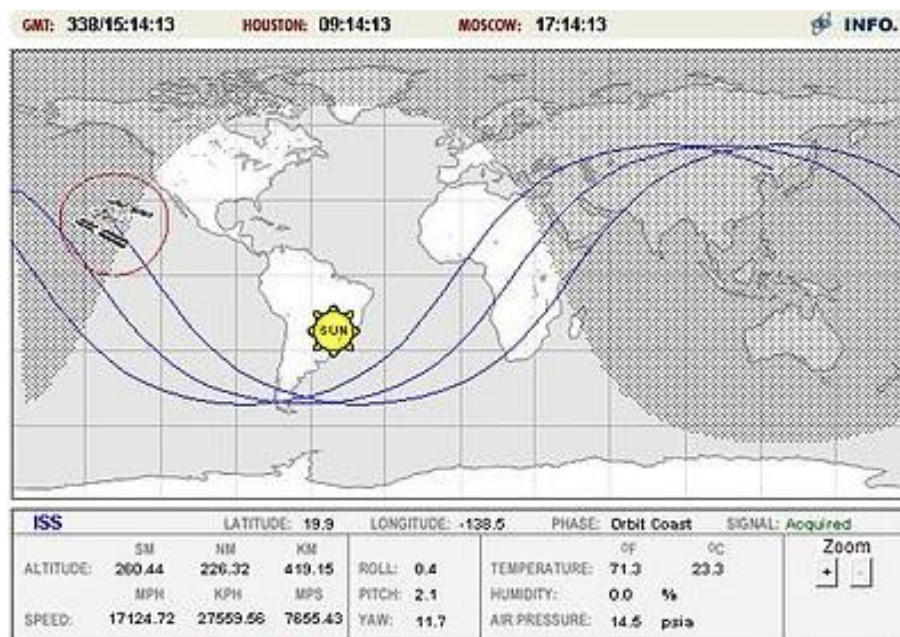


Figura 3-8 Ground track de la ISS (Fuente: Wikimedia Commons)

Podemos obtener las coordenadas de la pista terrestre de la siguiente manera:

$$\text{Latitud} \rightarrow \text{sen}(\phi(t)) = \text{sen}(i)\text{sen}(\omega + v(t))$$

$$\text{Longitud} \rightarrow \lambda(t) = \Omega - (\omega_{\oplus}t + \varphi) + \arctg(\cos(i)\text{tg}(\omega + v(t)))$$

Donde usamos los parámetros:

- Ω : longitud del nodo ascendente
- $\omega_{\oplus}$: velocidad de rotación de la Tierra
- φ : ascensión recta del Meridiano de Greenwich
- i : inclinación
- ω : argumento del periastro

3.6.4 Cálculo del ángulo de elevación

En este apartado procederemos a explicar las fórmulas y ecuaciones necesarias para hallar el ángulo que tenemos que elevar la antena para apuntar al satélite.

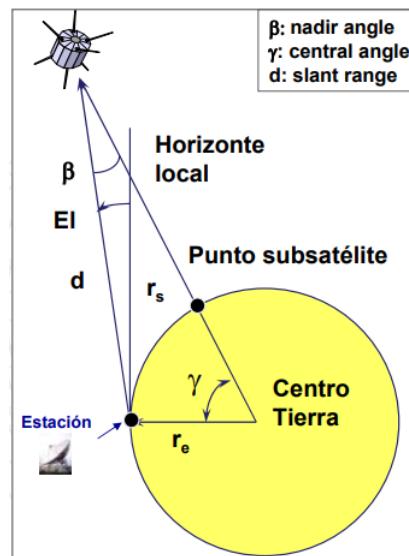


Figura 3-9 Esquema para el cálculo del ángulo de elevación [38]

En la Figura 3-9 tenemos los siguientes vectores que forman un plano:

- r_s : que es la distancia mínima del satélite a la superficie terrestre, es decir, la altura.
- r_e : el radio terrestre en el punto de la estación observadora.
- d : distancia del satélite a la estación observadora

A continuación, tenemos que hallar el ángulo γ , y lo hacemos mediante la siguiente fórmula:

$$\cos(\gamma) = \cos(L_{ae}) \cos(L_{as}) \cos(L_{oe} - L_{os}) + \text{sen}(L_{ae}) \text{sen}(L_{as})$$

Donde γ es el ángulo que forma el plano orbital con la longitud del observador y:

- L_{ae} : latitud de la estación observadora.
- L_{as} : latitud del punto sub-satélite

- **L_{oe}**: longitud de la estación observadora.
- **L_{os}**: longitud del punto sub-satélite.

Después, calculamos la distancia del satélite a la estación:

$$d = r_s \sqrt{1 + \left(\frac{r_e}{r_s}\right)^2 - 2 \frac{r_e}{r_s} \cos(\gamma)}$$

Finalmente, obtendremos el ángulo de elevación del siguiente modo:

$$\cos El = \frac{\sin(\gamma)}{d} = \frac{\sin(\gamma)}{r_s \sqrt{1 + \left(\frac{r_e}{r_s}\right)^2 - 2 \frac{r_e}{r_s} \cos(\gamma)}}$$

3.6.5 Cálculo del ángulo de azimut

Procedemos a desarrollar las fórmulas y ecuaciones necesarias para hallar el ángulo de azimut en dirección al satélite. Lo hallaremos por medio de un triángulo esférico.

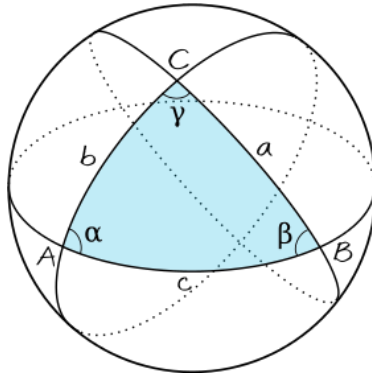


Figura 3-10 Triángulo esférico

Donde conocemos dos lados y un ángulo:

- **A**: situación de la estación observadora
- **B**: situación del punto sub-satélite
- **γ**: ángulo en el polo, el arco de Ecuador comprendido entre sus lados

Podemos dividir la forma de hallar el ángulo en el polo en los siguientes casos:

1. $\gamma = |L_A - L_B|$ si $\rightarrow |L_A - L| \leq 180^\circ$
2. $\gamma = |360^\circ - |L_A - L_B||$ si $\rightarrow |L_A - L_B| \geq 180^\circ$

Habría que particularizar dos casos, si ambos puntos están en el mismo hemisferio, norte o sur, o si ambos están en el mismo. En cualquiera de los dos casos, necesitamos calcular los ángulos α y β , que dependerán de la situación de A y B (Figura 3-10).

Si A (situación de la estación) está más al Norte que B (punto sub-satélite):

$$\tan\left(\frac{1}{2}(\alpha - \beta)\right) = \frac{\cot\left(\frac{\gamma}{2}\right) \sin\left(\frac{1}{2}(L_B - L_A)\right)}{\cos\left(\frac{1}{2}(L_B + L_A)\right)}$$

Por el contrario, si B está más al Norte que A:

$$\tan\left(\frac{1}{2}(\alpha + \beta)\right) = \frac{\cot\left(\frac{\gamma}{2}\right) \cos\left(\frac{1}{2}(L_B - L_A)\right)}{\sin\left(\frac{1}{2}(L_B + L_A)\right)}$$

Una vez calculadas las tangentes, podemos obtener fácilmente los ángulos α y β :

$$\alpha = \frac{1}{2}(\alpha + \beta) - \frac{1}{2}(\beta - \alpha)$$

$$\beta = \frac{1}{2}(\alpha + \beta) + \frac{1}{2}(\beta - \alpha)$$

Finalmente, el Azimut quedará definido de la siguiente forma:

Relación latitudes	Relación longitudes	Azimut
$I_A > I_B$	$\Delta L = L_A - L_B > 0$	$360 - \beta$ (1)
		$180 + \beta$ (2)
$I_A < I_B$	$\Delta L = L_A - L_B > 0$	$360 - \alpha$ (1)
		$180 + \alpha$ (2)
$I_A > I_B$	$\Delta L = L_A - L_B < 0$	β (1)
		$180 - \beta$ (2)
$I_A < I_B$	$\Delta L = L_A - L_B < 0$	α (1)
		$180 - \alpha$ (2)

Tabla 3-2 Resumen para el cálculo del Azimut

(1)→ si A o B están en el hemisferio Sur

(2)→ si ni A ni B están en el hemisferio Sur

3.6.6 Visibilidad

Definimos visibilidad como el ángulo a partir del cual un satélite forma una línea recta con la estación observadora sin que haya ningún obstáculo de por medio.

En el caso de que no haya obstáculos visuales, por ejemplo, navegando en el océano, se dará siempre que el ángulo de elevación sea mayor que cero.

3.6.7 Ángulos de cobertura

El área de cobertura del satélite se define como una región de la Tierra donde el satélite se ve en un ángulo de elevación mínimo predefinido, es decir es visible.

El área de cobertura del satélite en la Tierra depende de los parámetros orbitales. La comunicación bajo ángulos de baja elevación puede verse obstaculizada por barreras naturales, por eso ofrecen más cobertura los satélites GEO que los LEO. La zona en la que un satélite es capaz de proyectar su transmisión se conoce como huella.

3.7 Fuentes de datos orbitales

3.7.1 Formato de datos orbitales de la Norad: TLE

El formato más usado por todos los programas y aplicaciones para el seguimiento de satélites y la predicción de pases es el formato elaborado por la NORAD (*North American Aerospace Defense Command*), que es también usado por la NASA (*National Aeronautics and Space Administration*): el TLE (formato de dos líneas, en inglés *Two-Line Element Set*).

Éste consiste en dos filas de una longitud determinada, concretamente 69 caracteres, con una estructura rígida que nos proporciona la información sobre la identidad del satélite, características de su lanzamiento y, lo que es más importantemente, sobre su kepleriano o mecánica orbital, velocidad, azimut, elevación, etc. Este formato es únicamente válido para los modelos SGP4 y SGD4⁸, que solo tienen en cuenta las perturbaciones a la órbita provocadas por las fuerzas gravitacionales de la Tierra, la Luna, el Sol y los otros planetas del Sistema Solar.

Estos modelos permiten conocer con un error relativamente pequeño, 1 km respecto al epoch⁹, los datos necesarios para la predicción y seguimiento de satélites. Para más precisión existen los modelos SGP8 y SGD8, que tienen en cuenta además el deterioro orbital, proceso lento por el que los astros tienden a acercarse, reduciendo el radio orbital, debido a las fuerzas gravitacionales. Sin embargo, no son necesarios ya que el error es pequeño y la información se proporciona de forma continua con lo que la precisión del modelo SGP4 es bastante alta.

La estructura de los TLE, escrita en caracteres alfanuméricos, consta de:

- Línea 0: el nombre del satélite en 24 caracteres.
- Líneas 1 y 2: líneas de 69 caracteres que ofrecen la información de lanzamiento, referencia al epoch y datos de mecánica orbital.

⁸ Modelos orbitales de perturbaciones simplificadas para órbitas cercanas, SGP4, y búsqueda profunda SGD4.

⁹ En astronomía, hace referencia a una fecha precisa respecto a la cual se calculan las coordenadas celestes teniendo en cuenta la rotación de la Tierra respecto a su eje y al Sol, pero también las variaciones provocadas por los movimientos de precesión y nutación.

Línea 1	
Caracteres	Descripción
1	N.º de línea
03-07	N.º de satélite
08	Clasificación de seguridad
10-11	Últimas 2 cifras del año de lanzamiento
12-14	N.º de lanzamiento del año
15-17	N.º Pieza lanzada
19-20	Año del Epoch
21-32	Día del Epoch
34-43	Coeficiente balístico
45-52	Deterioro orbital
54-61	Arrastre o fricción de fluido
63	Tipo de efemérides
65-68	N.º de elemento
69	Checksum

Tabla 3-3 Línea 1 del TLE

De esta primera línea obtenemos principalmente los datos necesarios para identificar al satélite, con su número identificativo, y datos sobre su lanzamiento, fecha, número de artefactos lanzados, número de lanzamientos del año; y más importantemente se fija el epoch, referencia temporal imprescindible para la determinación de resultados.

Línea 2	
Caracteres	Descripción
1	N.º de línea
03-07	N.º de satélite
09-16	Inclinación
18-25	Longitud del nodo ascendiente (Ω)
27-33	Excentricidad
35-42	Argumento del perigeo (ω)
44-51	Anomalía media
53-63	Movimiento medio
64-68	N.º revolución al epoch
69	Checksum

Tabla 3-4 Línea 2 del TLE

De esta segunda línea obtenemos los datos astronómicos preceptivos para la obtención del azimut y la elevación de la antena en demora al satélite y para su seguimiento, así como otros como su velocidad angular media o la excentricidad de su órbita.

De esta forma, un ejemplo de TLE (de un satélite de la *National Oceanic and Atmospheric Administration*, NOAA, por sus siglas en inglés) lo podemos ver en la Figura 3-11 Ejemplo de

NOAA 14

1 23455U 94089A 97320.90946019 .00000140 00000-0 10191-3 0 2621

2 23455 99.0090 272.6745 0008546 223.1686 136.8816 14.11711747148495

TLE (Fuente: CelesTrak).

Así pues, a partir de estas dos líneas de datos resulta sencillo obtener los datos necesarios para desarrollar los cálculos preceptivos para las aplicaciones que se pretenden desarrollar. Con motivo de facilitar la comprensión el trabajo y para ofrecer una versión general, se pasa a explicar una serie de conceptos astronómicos y otros datos extraídos de los TLE.

3.7.1.1 Conceptos inferidos de la primera línea

Figura 3-11 Ejemplo de TLE (Fuente: CelesTrak) [3]

Tras esta breve introducción a los distintos orígenes de coordenadas, vamos a continuar desarrollando los principales conceptos más avanzados inferidos de la primera línea, algunos de ellos ya se han mencionado anteriormente y los obviaremos en este subapartado, otros, aunque hayan sido mencionados también, los definiremos para facilitar su comprensión, de los elementos de dos líneas como son:

- **Epoch (época):** en astronomía, un *epoch* es un momento en el tiempo que se utiliza como punto de referencia para alguna cantidad astronómica variable en el tiempo, como las coordenadas celestes o los elementos orbitales elípticos de un cuerpo celeste, ya que están sujetos a perturbaciones y varían con el tiempo. Estas características astronómicas variables pueden incluir, por ejemplo, la longitud media o la anomalía media de un cuerpo, entre otras muchas. Por lo general se toma el 1 de enero de 2000 a las 12:00 en Greenwich como referencia.

- **Movimiento medio:** en mecánica orbital, el movimiento medio (representado por n) es la velocidad angular requerida para que un cuerpo complete una órbita, asumiendo una velocidad constante en una órbita circular que se completa al mismo tiempo que la órbita real del cuerpo, elíptica y de velocidad variable.

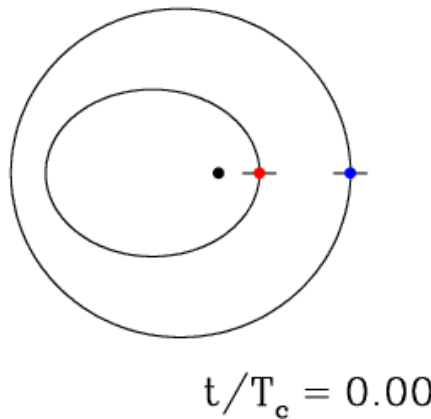


Figura 3-12 Movimiento medio [24]

El movimiento medio se utiliza como una aproximación de la velocidad orbital real con el objetivo de hacer un cálculo aproximado de la posición del cuerpo en su órbita, por ejemplo, a partir de un conjunto de elementos orbitales, como haremos en este trabajo. Para cálculos más precisos esta posición media se refinará mediante la ecuación de Kepler para producir la posición verdadera, usando de base modelos orbitales más complejo.

- **Deterioro orbital:** proceso por el cual la altura de la órbita de un objeto espacial respecto a otro, disminuye con el paso del tiempo. Este fenómeno es provocado principalmente por la fricción producida con la atmósfera o la interacción con fuerzas gravitacionales de otros astros, aunque en este caso no se tendrá en cuenta su influencia. Su cálculo corresponde a la segunda derivada del movimiento medio.
- **Arrastre:** en aerodinámica, la resistencia aerodinámica es la fuerza de arrastre del fluido que actúa sobre cualquier cuerpo sólido en movimiento en la dirección del flujo de flujo libre del fluido.

3.7.1.2 Parámetros orbitales de la segunda línea

Después de detallar los términos y parámetros que proporciona la primera línea de datos vamos a proceder a realizar lo mismo con la segunda línea.

Aunque ya hemos definido con anterioridad la mayoría de conceptos inferidos de la 2ª línea vamos a ahondar en algunos de sus conceptos.

Es importante, además, destacar que mientras que la primera línea abarcaba los datos necesarios para reconocer el satélite, su misión o lanzamiento, en esta 2ª línea se agrupan los datos que permitirán los cálculos propiamente dichos.

De este modo, los principales conceptos que vamos a desarrollar son:

- **Anomalía media:** se define como una fracción, expresada en forma de ángulo, de un período orbital. También se puede ver como el ángulo que forma con el eje de la elipse un astro que se mueve con velocidad uniforme describiendo una circunferencia perfecta y en la que coincide el eje principal de la elipse con su diámetro. Se designa por M .

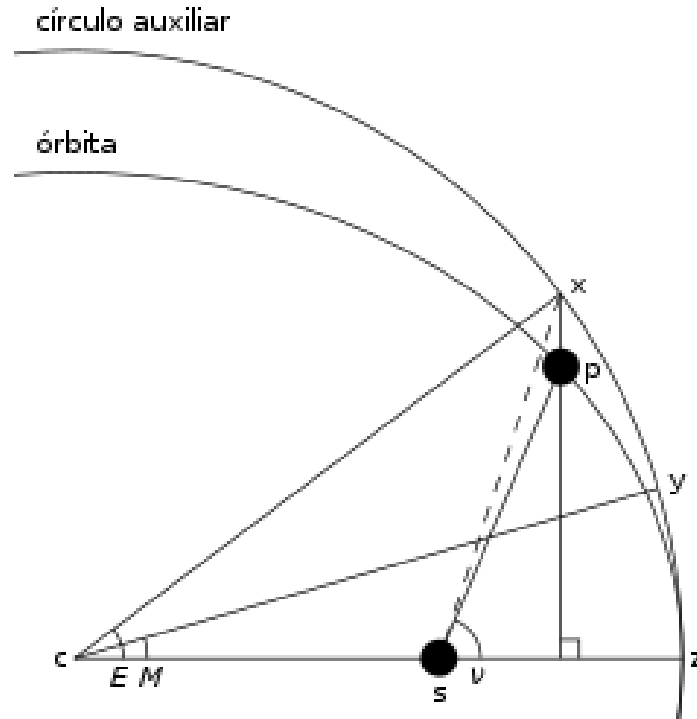


Figura 3-13 Representación de la anomalía media
(Fuente: WikiMedia Commons)

4 DESARROLLO

4.1 Requisitos funcionales

Este proyecto se ha compuesto de dos fases de trabajo diferenciadas, por un lado, el desarrollo del software: escritura del programa, obtención de datos, cálculo y análisis y envío/recepción de datos al hardware; y, por otro lado, el desarrollo de la plataforma mecánica, estudio del diseño, cargas y resistencia, acople de las piezas y acople entre hardware y software.

En este apartado, se definen los requisitos funcionales del proyecto, desde un punto de vista de hardware primero y desde el punto de vista del software después, para proseguir con el desarrollo del diseño mecánico y los requisitos específicos del sistema y, después, el software necesario para satisfacer las exigencias marcadas con los componentes físicos de los que disponemos.

El objetivo del trabajo es el de desarrollar una plataforma mecánica de seguimiento de satélites. Para ello las características principales en cuanto a hardware de las que queremos dotar a nuestro sistema son:

- **Ligereza:** todo el sistema debe ser fácilmente transportable y no debe ser pesado.
- **Resistencia:** debe ser capaz de soportar el peso de los elementos estructurales, los servomotores y la antena, y sus momentos derivados, lo que se traduce en ser capaz de soportar hasta 3 kg y 20 Nm, respectivamente.
- **Portabilidad:** nuestro sistema debe ser fácilmente desmontable, transportable y montado con rapidez para darle una mayor utilidad.

Mientras que, si hablamos de software, queremos que nuestro sistema se defina por las siguientes características:

- **Sencillez:** nuestro sistema debe ser fácilmente manejable y comprensible por cualquier usuario, incluso sin necesidad de conocimientos previos.
- **Configurabilidad:** nuestro proyecto debe admitir diferentes modos de funcionamiento y formas de trabajo.
- **Portabilidad:** nuestro software debe ser fácilmente transportable y ejecutado desde cualquier equipo sin necesidad de instalación previa.
- **Independiente:** nuestro sistema no puede depender de la red para su funcionamiento, para ello tiene que poder disponer de datos para realizar los cálculos deseados hasta sin conexión.

- **Fuente de datos:** debe poder nutrirse de los datos orbitales en el formato especificado en el trabajo, formato TLE.
- **Modularidad:** para permitir su rápido desarrollo, revisión y corrección de errores y añadirle nuevas funcionalidades, nuestro programa debe ser de estructura modular.

De esta manera, pasamos a definir el diseño mecánico y de software para satisfacer las necesidades planteadas en el presente subapartado.

4.2 Diseño mecánico

Es este apartado procedemos a definir el diseño mecánico de nuestro trabajo en base a los requisitos y características desarrolladas en el anterior subapartado.

4.2.1 Diseño conceptual mecánico

En el presente subapartado, procederemos a explicar el desarrollo general del diseño mecánico elaborado para cumplir las funciones deseadas. Los principales componentes mecánicos de nuestra plataforma de seguimiento y apuntamiento de satélites son:

- Trípode
- Servomotores
- Antena
- Placa Arduino
- Agarres y sujeciones
- Fuente de alimentación y cableado

A continuación, iremos desarrollando cada uno de los elementos analizando sus características principales y como nos procurarán los requisitos técnicos exigidos.

4.2.1.1 Trípode

Principal elemento estructural, compuesto de aluminio, soportara el peso conjunto del resto de elementos.



**Figura 4-1 Trípode
empleado en el proyecto**

El trípode elegido [10] tiene un peso de 1,8 kg y es fácilmente plegable. Por especificaciones del vendedor, sabemos que puede aguantar diferentes pesos según el estado de extensión de los

pies, soportando pesos de 4 kg en el punto en el que está más extendido, apertura máxima de los pies y del monopie, y hasta 12 kg en la situación en la que sus pies están más recogidos.

Para mayor seguridad, haremos los cálculos para el punto en el que la carga que puede soportar es mínima, esto es 4 kg. De esta forma, sobredimensionaremos el proyecto, garantizando la estabilidad y fiabilidad del conjunto.

4.2.1.2 Servomotores

Los servomotores son los elementos encargados de realizar el seguimiento y apuntamiento de la antena, orientándola hacia los ángulos de visión, mencionados anteriormente en este trabajo, de elevación y azimut.



Figura 4-2 Servomotor SG12-70 [16]

Un servomotor es un tipo de motor, con un propósito diferente a aquellos que tienen la capacidad de estar permanentemente girando, gobernado electrónicamente, por lo que se le puede considerar un dispositivo actuador, es decir, cuando reciban una señal eléctrica determinada realizarán el movimiento conforme.

Los servomotores, a diferencia de otros tipos de motores, no son capaces de ofrecer revoluciones continuas, en su caso nosotros le ordenaremos el ángulo deseado y él se posicionará en ese ángulo. Cabe destacar también, que normalmente tienen unos topes internos para delimitar las revoluciones que pueden dar, en nuestro caso en un servo de 640°, por lo que podrá dar cerca de dos revoluciones completas.

Los componentes principales de un servomotor son:

- **Topes:** para limitar mecánicamente el giro del motor.
- **Mecanismo de engranajes reductores:** en esta clase de motores, se prioriza la fuerza a la velocidad de giro por medio de una serie de engranajes reductores. En nuestro caso, tenemos una reducción de 7:1, lo que significa que nuestro motor tendrá una velocidad 7 veces menor pero un torque 7 veces mayor.
- **Controlador electrónico:** recibe la señal de giro electrónica, la interpreta y lleva el eje a la posición ordenada.
- **Motor eléctrico:** realiza el movimiento ordenado en base a la señal electrónica.

- **Potenciómetro:** internamente, analiza la señal electrónica recibida para realizar la función de sensor de ángulo de giro.

En este trabajo usaremos dos servos, uno para controlar el ángulo de azimut y otro para controlar el ángulo de elevación. Serán los encargados de soportar los esfuerzos flectores provocados por el movimiento de la antena.

En los Anexos I y II de esta memoria se pueden observar los planos y especificaciones del vendedor del servomotor empleado en este proyecto.

4.2.1.3 Antena

La antena es el elemento que posicionaremos por medio de los servomotores.

Para el trabajo que nos ocupa, hemos empleado una antena de tipo Yagi, de polarización vertical o horizontal, según el montaje, bibanda (UHF/VHF) Maldol hs-fox727(Figura 4-3).



Figura 4-3 Antena Maldol hs-fox727

Esta antena trabaja centrada en dos frecuencias principales, 144/430 MHz, en las bandas de UHF y VHF con ganancias de 9,5 dBi y 11,5 dBi respectivamente. Tiene una relación delante/atrás de 12 dB y una potencia máxima de 50 W, aunque a lo largo de este proyecto solo se utilizará en recepción. Esta ambivalencia se consigue por medio de 5 elementos directores cortos para UHF, de 0,7 m, y 3 elementos directores largos para VHF, de 1,044 m.

Respecto a sus aspectos mecánicos, tiene un peso de 850 gr y una longitud del *boom* de 1,13m con el centro de gravedad aproximadamente en la sujeción.

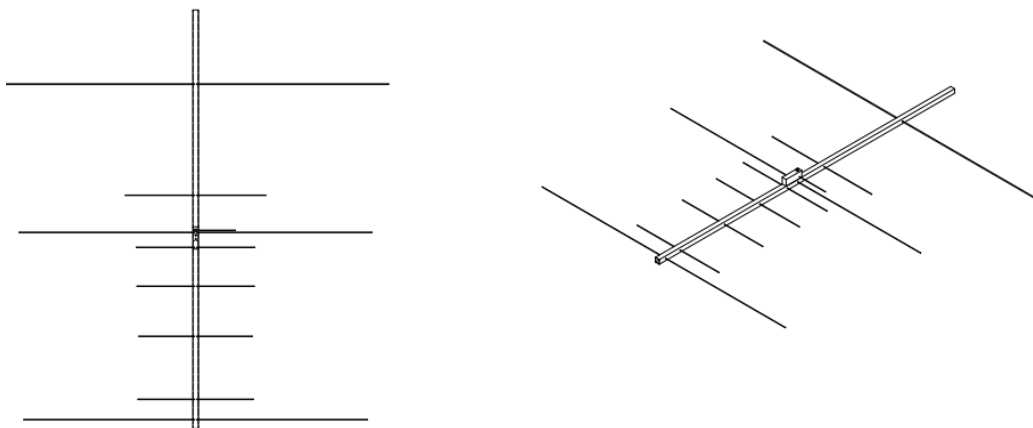


Figura 4-4 Diseño CAD de la antena

4.2.1.4 Placa Arduino

Un Arduino (Figura 4-5) es una plataforma electrónica que combina funciones de hardware y software con un lenguaje de programación basado en programas de código abierto, enfocada en ser fácil de usar.



Figura 4-5 Placa Arduino Uno (Fuente: Arduino) [12]

Físicamente, las placas Arduino son capaces de leer entradas de todo tipo (luz en un sensor, pulsar botón, o un mensaje de Twitter, por ejemplo) y convertirlo en una salida como, por ejemplo: activar un motor, encender un LED o publicar algo en línea.

Para ello se utiliza el lenguaje de programación Arduino, basado en el cableado, y el software Arduino (IDE), basado en procesamiento.

4.2.1.5 Agarres y sujeciones

Son los elementos que unen el resto de piezas entre sí, son una parte crítica de este trabajo ya que han de asegurar la estabilidad del conjunto para no perder precisión en el apuntamiento y seguimiento.



Figura 4-6 Conjunto de agarres y sujeciones del proyecto

En el montaje de este proyecto hemos usado:

- **Canal en U de 5 agujeros:** con una altura y base de 1,5", con un peso de 50 gr. A él se hace firme el servo de azimuth por medio de uniones roscadas y se asienta sobre el trípode.
- **Soporte en C:** de una altura de 1,7" y un peso de 30 gr. A él se hace firme el servo que controla la elevación y se sustenta sobre el mástil.
- **Soporte a 90°:** de una altura de 1,5" y un peso de 13 gr. Se sustenta sobre el servo de elevación por medio de uniones roscadas y lo une con la antena por el otro lado.

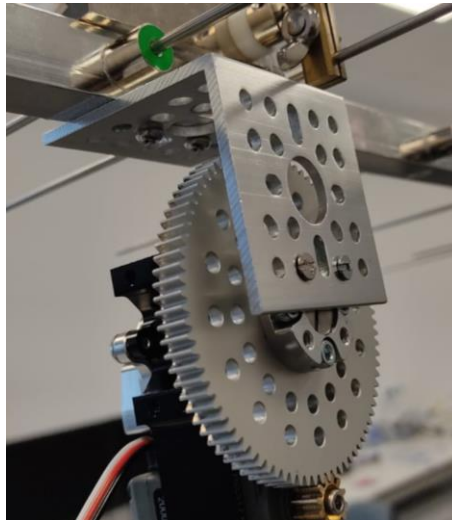


Figura 4-7 Fotografía del soporte a 90° en el montaje

- **Fijación circular:** de 1" con un peso de 12gr y un diámetro de 1" (x2). Sirven para enlazar el mástil con el resto de elementos, uno va al servo de azimuth y otro al soporte en C.

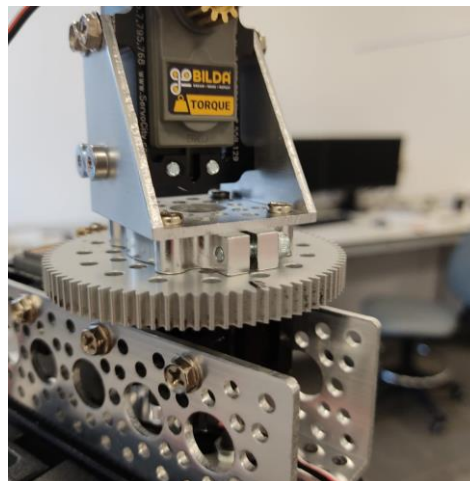


Figura 4-8 Fotografía del soporte en U, el soporte en C y la fijación circular en el montaje

Habrà que tener en cuenta los esfuerzos que soportan y la forma en la que se realizan las uniones: uniones roscadas, apriete o apoyo.

En el Anexo III: Planos y especificaciones de los elementos estructurales pueden encontrarse los planos y especificaciones del vendedor de cada uno de los elementos estructurales.

4.2.1.6 Fuente de alimentación y cableado

Para un mayor rendimiento del sistema, hemos decidido aportar fuentes de alimentación independientes al Arduino Uno, que se alimentará por medio de la entrada USB al ordenador directamente, y a los servomotores, que tendrán

una fuente de alimentación independiente de 6 voltios que toma de la red eléctrica directamente (Figura 4-9).



**Figura 4-9 Fuente de alimentación
de 6V**

Además, tenemos el cableado que va de la placa Arduino al servomotor. Se trata de un cableado de 3 colores:

- **Negro:** tierra, negativo.
- **Rojo:** alimentación, positivo.
- **Blanco:** señales de control. Pin 6 y 9, elevación y azimut respectivamente.

4.2.2 Integración de componentes mecánicos

Teniendo en cuenta los elementos anteriormente mencionados, el conjunto mecánico quedará compuesto de la siguiente forma (Figura 4-10).



Figura 4-10 Montaje del sistema

4.2.3 Límites mecánicos

En este apartado se harán los cálculos necesarios para definir los límites mecánicos de nuestro sistema. Para ello, haremos primero un estudio de los pesos que ha de soportar nuestro trípode (teniendo en cuenta que su resistencia nominal es de 4kg en el caso más desfavorable) y haremos después un estudio de los momentos provocados por el giro de la antena en los planos de elevación y azimut.

4.2.3.1 Estudio de fallo a compresión/tracción

El peso que debe soportar nuestro sistema es:

$$\begin{aligned}
 P_{TOTAL} &= P_{ANTENA} + 2 * P_{SERVO} + P_{ESTRUCTURA}; \\
 P_{TOTAL} &= 850 + 2 * 200 + 100 \\
 P_{TOTAL} &= 1350gr
 \end{aligned}$$

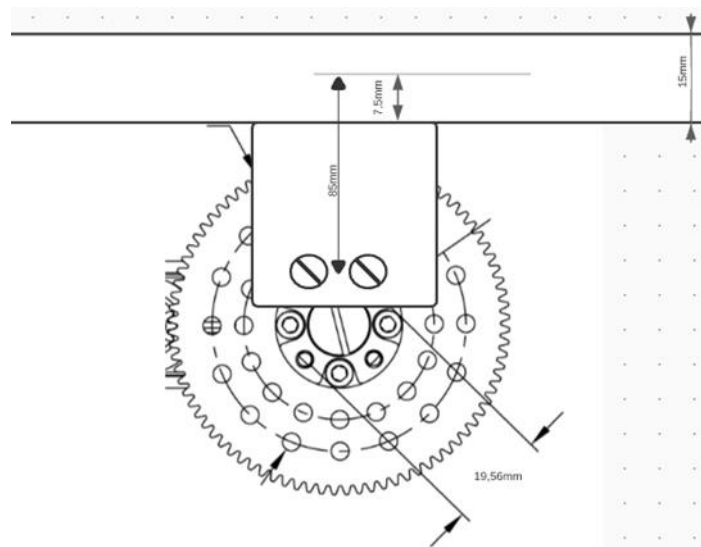


Figura 4-11 Alzado del anclaje de la antena al servo

Por tanto, como nuestro sistema aguanta hasta 4kg (>1,35kg) queda demostrado que nuestro sistema no fallará a compresión.

4.2.3.2 Estudio de fallo por flexión

A continuación, haremos el estudio de resistencia a flexión de nuestro sistema. Para ello, tenemos que estudiar el peso desplazado respecto al centro de gravedad del sistema. El punto crítico será el anclaje de la antena al servo de elevación pues será donde se concentren los esfuerzos más grandes y la unión entre el servo de elevación y el servo de azimut.

En la Figura 4-12 y la Figura 4-11 podemos observar la representación del problema.

A través del estudio realizado en la aplicación de CAD Inventor hemos comprobado que el anclaje de la antena está situado en su centro de gravedad, por lo tanto, la antena estará equilibrada en el eje longitudinal cuando el ángulo de elevación sea igual o menor que 45°.

De esta forma, se derivan las dos posiciones críticas que tendrá que soportar nuestro sistema:

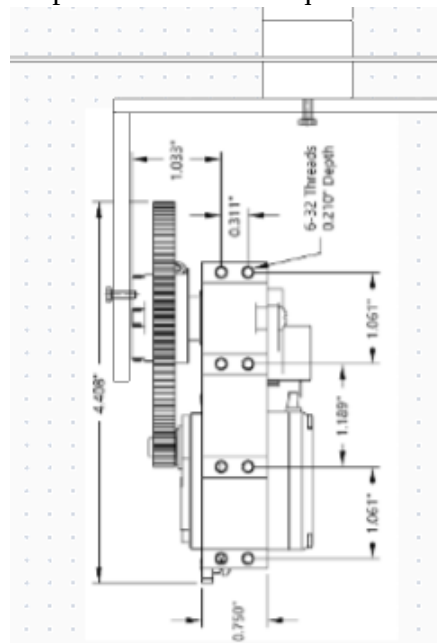


Figura 4-12 Perfil del anclaje de la antena al servo

1. **Antena situada con un ángulo de elevación de 0°:** al no estar alineado el eje del sistema con el centro de gravedad (está situado a una distancia de 90mm) de la antena siempre estará presente este esfuerzo flector.

$$\begin{aligned}M_{Flector1} &= P_{Antena} * d_{ejey} ; \\M_{Flector1} &= 850gr * 9cm ; \\M_{Flector1} &= 7650gr.cm = 7,65kg.cm\end{aligned}$$

Tenemos que comprobar a continuación el momento que pueden aguantar tanto los tornillos.

2. **Antena situada con un ángulo de elevación de 90°:** en este caso todo el peso de la antena estará desplazado respecto al eje del sistema y por tanto generará un momento respecto a él. De forma análoga, tenemos que:

$$\begin{aligned}M_{Flector2} &= P_{Antena} * d_{ejex} ; \\M_{Flector2} &= 850gr * 8,5cm ; \\M_{Flector2} &= 7225gr.cm = 7,225kg.cm\end{aligned}$$

Por especificaciones del vendedor, sabemos que nuestros servos, cuando están alimentados con una tensión de 6V, son capaces de mover cargas de hasta 15,12kg.cm (>7,225kg.cm). Por lo tanto, apreciamos que están sobredimensionados.

4.2.4 Dimensionamiento de componentes

Una vez comprobado que los servos aguantarán sin fallar la carga de la antena debemos realizar la elección de los elementos de unión necesarios para soportar las cargas de la estructura.

No utilizaremos en este proyecto tornillos de calidades menores de 4.6, ya que, al no estar experimentalmente testados, no son válidos para nuestra aplicación. En la siguiente tabla se relacionan unos valores orientativos de resistencia de los tornillos según su calidad.

Valores nominales del límite elástico f_{yb} y de la resistencia a tracción última f_{ub} de tornillos							
Tipo de tornillo	4.6	4.8	5.6	5.8	6.8	8.8	10.9
f_{yb} (N/mm ²)	240	320	300	400	480	640	900
f_{ub} (N/mm ²)	400	400	500	500	600	800	1000

Tabla 4-1 Valores nominales de límite elástico según la calidad del tornillo (Fuente: ingmecanica) [40]

Vamos a empezar a dimensionar el tornillo estudiando los esfuerzos a aplastamiento y cortadura del tornillo. La fuerza cortante que soporta un tornillo de calidad superior a 4.6 es:

$$F_{v,Rd} = \frac{0,6 * f_{ub} * A_s}{\gamma_{Mb}}$$

En donde:

- A_s : es el área resistente del tornillo.
- f_{ub} : es la tensión última a tracción del tornillo, extraída de la normativa en vigor.
- γ_{Mb} : es un coeficiente de seguridad, tomaremos 1,25.

Como además el diámetro de nuestros tornillos viene condicionado por los componentes utilizados, necesitamos un tornillo de diámetro igual a 6mm. Por tanto:

$$F_{v,Rd} = \frac{0,6 * f_{ub} * \pi * (3 * 10^{-3})^2}{1,25}$$

Para que soporte nuestras especificaciones $F_{v,Rd} > 7,65kg.cm$. Así, finalmente:

$$765 * 9,81 = \frac{0,6 * f_{ub} * \pi * (3)^2}{1,25};$$

$$f_{ub} = \frac{1,25 * 765 * 9,81}{0,6 * \pi * (3)^2};$$

$$f_{ub} = 553N/mm^2$$

También cumplirá en el otro sentido de la fuerza (M flector igual a 7,225 kg.cm).

Por lo tanto, necesitamos unos tornillos que tengan una mayor fuerza última a tracción. Hemos decidido escoger tornillos de métrica M3, de paso 0,5 y longitud 11mm, ya que cumplen con los requerimientos deseados, tiene medidas que encajan en nuestro sistema en cuanto a longitud se refiere y son un tamaño de rosca métrica normalizado, lo que facilitará la tarea de buscar respuesta.

4.2.5 Conclusión al estudio del comportamiento mecánico del sistema

Tal y como hemos calculado, nuestro sistema soportará con seguridad y estabilidad la estructura de la plataforma de seguimiento y no se producirán fallos ni a flexión ni a compresión.

Se observa que el primer elemento de la estructura que será limitante son los servos. El momento máximo que pueden soportar es de 15,12kg.cm por lo que, sin cambiar nuestra estructura, se podría reemplazar la antena por otra de hasta 1,77kg, o por otras antenas más pesadas si se reduce la distancia al eje.

4.3 Diseño del hardware

4.3.1 Elección del controlador. Arduino desde el punto de vista del hardware.

Hemos elegido este controlador, Arduino Uno, ya que se caracteriza por una experiencia de usuario sencilla y accesible, es por ello que se usa a menudo en miles de proyectos y aplicaciones diferentes. El software Arduino es fácil de usar para principiantes, pero lo suficientemente flexible para usuarios avanzados. Se puede ejecutar tanto en Mac, como Windows o Linux.

Sus usos van desde su utilización para construir instrumentos científicos de bajo costo, o para empezar con la programación, la robótica y, más recientemente, el internet de las cosas, *machine learning* y *big data*. Sus principales ventajas son:

- **Bajo coste:** las placas Arduino son relativamente baratas en comparación con otras plataformas de microcontroladores.
- **Multiplataforma:** puede correrse en diferentes sistemas operativos como Mac, Windows o Linux.
- **Ligero y fácilmente transportable**
- **Entorno de programación simple y claro:** el software Arduino (IDE) es fácil de usar para principiantes, pero lo suficientemente flexible para que los usuarios avanzados también se aprovechen de él.

- **Hardware de código abierto y extensible:** los planos de las placas de Arduino se publican bajo una licencia *Creative Commons*, para que los diseñadores de circuitos experimentados puedan hacer su propia versión del módulo, ampliándolo y mejorándolo.
- **Software de código abierto:** basado en licencias *Creative Commons*, los programadores comparten sus programas para que cada usuario lo pueda descargar y adaptar a sus necesidades

4.3.2 Control de servomotores

En el presente subapartado entraremos más en detalle en cómo se realiza el control de giro de los servomotores por medio de las señales electrónicas.

El motor se controla con una señal eléctrica PWM (*pulse-width modulation*, en inglés modulación por ancho de pulso), ya sea analógica o digital, que traduce un pulso, dentro de un umbral con un tiempo mínimo y máximo de pulso, de una duración determinada de tiempo en una determinada cantidad de movimiento (Figura 4-13).

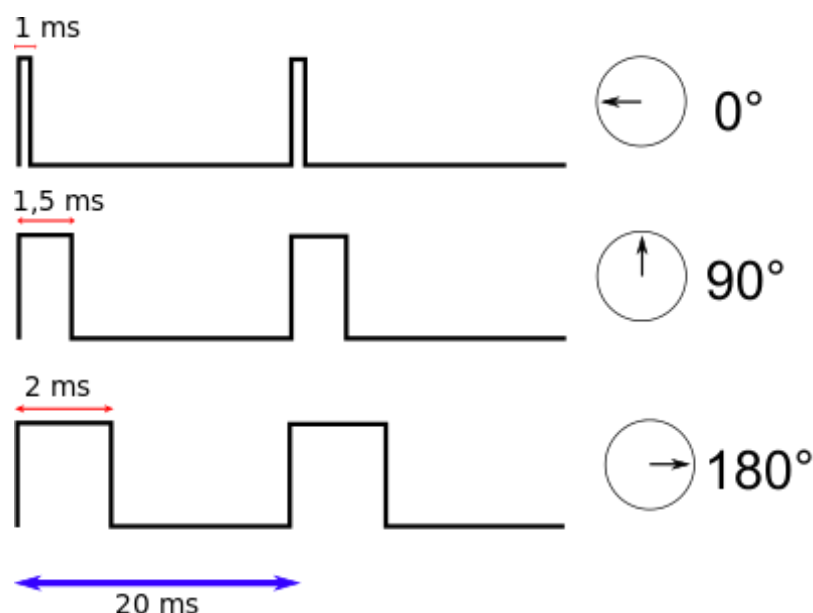


Figura 4-13 Ejemplo de señales PWM (Fuente: Wikimedia Commons)

Los pasos que sigue el sistema son:

En primer lugar, el controlador electrónico recibe por el cable de la señal de control, de los pines correspondientes, las órdenes de posicionamiento, por medio de señales PWM, que envía la placa Arduino.

Después, el potenciómetro interno compara la posición actual con la ordenada, midiendo el voltaje correspondiente al ángulo en el que está ahora el servo y lo compara con la nueva orden recibida para desplazarlo el error, la diferencia entre ambos voltajes. Si el error es igual a cero, significará que se ha ordenado la posición actual y que, por tanto, el servo no recibirá tensión. Por el contrario, si el error es distinto de cero, se envía al servo la tensión correspondiente al desplazamiento deseado.

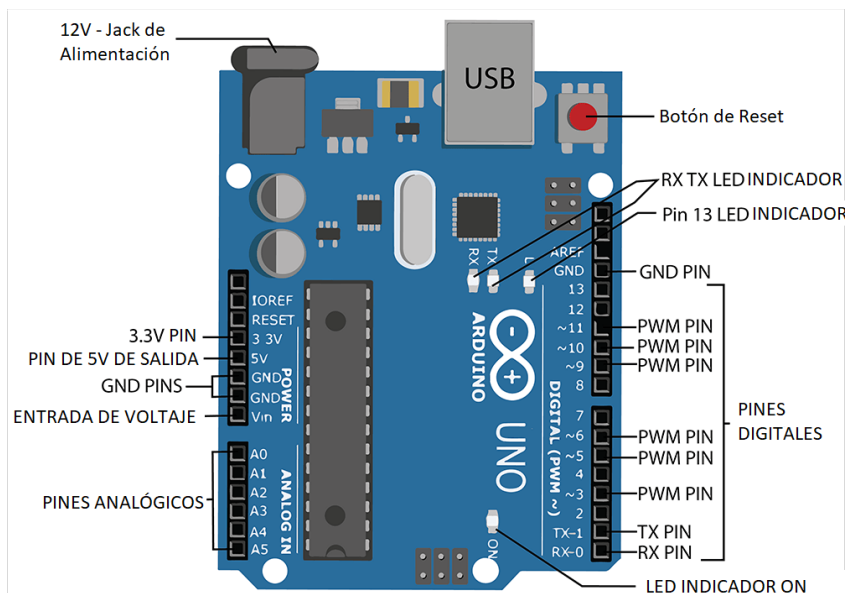


Figura 4-14 Descripción de elementos de la placa Arduino Uno
(Fuente: Arduino) [12]

Cabe destacar, que la placa Arduino tiene un número de pines determinados para el control del servo PWM, en concreto seis salidas PWM (son los pines: 4,5,6,9,10 y 11).

Para mayor facilidad, existe una librería diseñada para el control de los servomotores y disponible bajo licencia de código abierto. Por lo tanto, tendremos que tener en cuenta utilizar esas salidas para nuestro proyecto (en el caso que nos ocupa usamos los pines 6 y 9).

4.3.3 Calibración de los servomotores

Es imprescindible realizar una buena calibración de los servomotores para determinar con exactitud el rango PWM real de cada servomotor y mapearlo en consecuencia para que cada señal PWM de una determinada duración corresponda a una determinada cantidad de giro.

Según especificaciones de diseño del vendedor se tiene que los servos tienen una capacidad de 640° y un rango PWM de 500 a 2500µs. Sin embargo, haciendo la calibración se descubre que tienen una capacidad real de giro de aproximadamente 630° y un rango real de PWM de 500 a 2225µs.

A raíz de esto y de la estructura del montaje, se decide definir los siguientes límites:

- **Servo de Azimut:** recorrido de 0 a 360° correspondiente al rango PWM (500-2225μs)
- **Servo de Elevación:** recorrido de 0 a 90° correspondiente al rango PWM (500-2225us)

4.4 Diseño del software

En el presente apartado desarrollaremos el diseño conceptual del diseño del software y de cómo se acopla y controla el hardware. Para ello, hablaremos del entorno de desarrollo usado, para pasar a volver a hablar de la placa Arduino, esta vez desde un punto de vista puramente de programación y finalizaremos con la explicación de la librería de control de rotores y antenas Hamlib.

4.4.1 Entorno de desarrollo

Python fue diseñado en la década de los ochenta por Guido van Rossum.

Es un lenguaje de programación, orientado a objetos y de alto nivel. Sus estructuras de datos integradas de alto nivel, lo hacen muy atractivo para el desarrollo rápido de aplicaciones, así como para su uso como lenguaje de scripts o como enlace para conectar componentes existentes. Con una sintaxis simple y fácil de aprender, Python facilita la legibilidad y el entendimiento, reduciendo la complejidad de cara al usuario.

Una de las grandes ventajas que tiene respecto a otros entornos es su modularidad. Python admite módulos y paquetes, lo que fomenta la reutilización del código y aporta gran variedad de herramientas a cualquier usuario. El intérprete de Python y la extensa biblioteca estándar están disponibles, tanto en formato fuente como binario, gratuitamente para todas las plataformas principales y pueden distribuirse libremente.



Figura 4-15 Logo Python (Fuente: Python.org) [26]

Otra gran funcionalidad que tiene y lo diferencia de sus competidores es que, dado que no hay ningún paso de compilación, el ciclo de edición-prueba-depuración es increíblemente rápido. Depurar los programas de Python es fácil: un error o una entrada incorrecta nunca causarán una falla de segmentación. En cambio, cuando el intérprete descubre un error, genera una excepción. Cuando el programa no detecta la excepción, el intérprete imprime un seguimiento de la pila. Un depurador a nivel de fuente permite la inspección de variables locales y globales, la evaluación de expresiones arbitrarias, el establecimiento de puntos de interrupción, el paso a través del código una línea a la vez, etc. El depurador está escrito en Python, atestiguando el poder introspectivo de Python.

Por otro lado, a menudo la forma más rápida de depurar un programa es agregar algunas declaraciones de impresión a la fuente: el ciclo rápido de edición-prueba-depuración hace que este enfoque simple sea muy efectivo.

Asimismo, otra característica que lo convierte en un sistema tan apreciado, es el índice de paquetes de Python (PyPI). Es un repositorio de software para el lenguaje de programación Python que ayuda al usuario a encontrar e instalar software desarrollado y compartido por la comunidad de Python.

4.4.1.1 Herramientas de Python

A continuación, explicaremos brevemente dos herramientas ofrecidas libre y gratuitamente por Python y que hemos empleado a lo largo de este trabajo: Anaconda y Spyder.

4.4.1.1.1 Anaconda

Anaconda es una distribución de los lenguajes de programación Python y R para computación, que tiene como objetivo simplificar la administración y la implementación de paquetes de Python para aplicaciones científicas.



Figura 4-16 Logo de Anaconda (Fuente: Anaconda)
[27]

La distribución incluye paquetes de ciencia de datos adecuados para Windows, Linux y macOS. Es desarrollado y mantenido por Anaconda, Inc., creada por Peter Wang y Travis Oliphant en 2012. Como producto de Anaconda, Inc., también se conoce como *Anaconda Distribution* o *Anaconda Individual Edition*, mientras que otros productos de la compañía, como son *Anaconda Team Edition* o *Anaconda Enterprise Edition*, no son gratuitos Anaconda sí lo es.

Las versiones de paquetes en Anaconda son administradas por el sistema de administración de paquetes conda. Conda es un administrador de paquetes que se gestó como un paquete de código abierto separado, pero que a la postre resultó siendo útil por sí solo y para otras funciones más allá de Python.

En el caso de este proyecto lo hemos usado para crear un entorno virtual, herramienta muy útil que permite crear un directorio con diferentes versiones de los mismos paquetes e instalar paquetes únicamente a él y que está muy enfocado a la realización de proyectos y para instalar distintas librerías necesarias para realizar los cálculos (*Ephem*, *Skyfield*, *Numpy*...). Este proceso de instalar paquetes además se realiza de forma muy sencilla al disponer del repositorio PyPI ya que solo habrá que escribir el comando de terminal *'pip install'* seguido del nombre del paquete para que comience el proceso.

4.4.1.1.2 Spyder

El proyecto Spyder se autodefine así: “Spyder es un entorno científico de código abierto y gratuito escrito en Python, y diseñado por y para científicos, ingenieros y analistas de datos. Presenta una combinación única de funciones avanzadas de edición, análisis, depuración y creación de perfiles de una herramienta de desarrollo integral con la exploración de datos, ejecución interactiva, inspección profunda y hermosas capacidades de visualización de un paquete científico” [11].



Figura 4-17 Logo de Spyder (Fuente: Spyder) [11]

Es el interfaz diseñado para programar en este trabajo que, asociado a nuestro entorno virtual, trabajará en la versión deseada de Python y de nuestros paquetes (permitiéndonos trabajar en diferentes versiones de Python con diferentes entornos). A continuación, explicaremos brevemente su estructura y la interfaz en sí. Cuenta, principalmente, de las siguientes partes:

- **Editor de texto:** trabaja de manera eficiente en varios idiomas incluyendo un navegador de funciones y clases, herramientas de análisis de código, finalización automática de código, división horizontal o vertical y definición de acceso. En la Figura 4-18 Captura de pantalla en la parte izquierda de la pantalla.
- **Consola *iPython*:** es la encargada de ejecutar el programa. Nos ofrece una capacidad muy interesante como es el hecho de poder reescribir el código mientras se ejecuta ya que permite enmendar errores rápidamente. También nos permite ejecutar programas por bloques y por líneas de código, lo que aumenta la rapidez a la hora de programar. En la Figura 4-18 en la esquina inferior derecha.
- **Explorador de variables:** esta parte nos permite interactuar y modificar las variables del programa paralelamente a su ejecución, trazar histogramas, gráficas o series temporales, trabajar con matrices *Numpy*, explorar objetos anidados y entre otras funcionalidades. En la Figura 4-18 en la esquina superior derecha.
- **Debugger:** esta herramienta permite analizar el código automáticamente en busca de errores de sintaxis o argumentos inválidos para facilitar la experiencia al usuario.

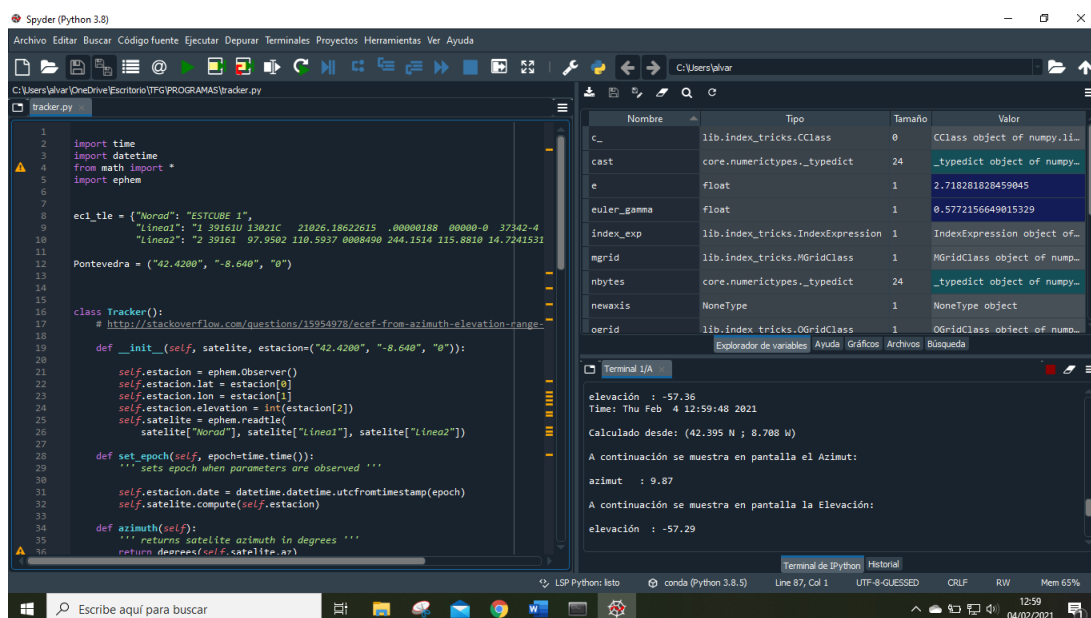


Figura 4-18 Captura de pantalla de Spyder

4.4.1.2 Elección de Python como lenguaje de programación

En este apartado se justifica el uso del lenguaje de programación Python para este proyecto. Python a menudo se compara con otros lenguajes interpretados como Java, C++, Perl o Tcl, entre otros muchos.

Podemos ver como las cualidades, anteriormente mencionadas, de Python han provocado que sea un lenguaje muy apreciado y su popularidad haya crecido rápidamente en la última década, véase (Figura 4-19).

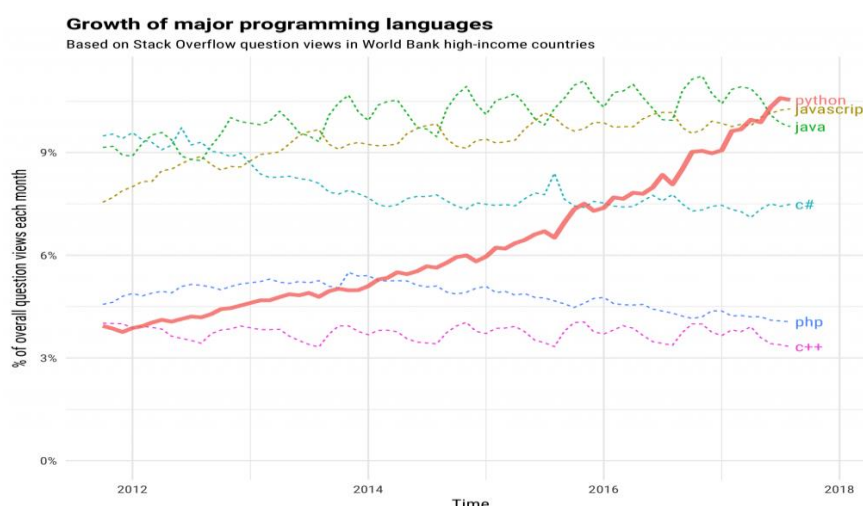


Figura 4-19 Crecimiento de los principales lenguajes de programación (Fuente: StackOverflow) [28]

En esta sección compararemos brevemente Python con dos de los lenguajes más populares en el mundo de la programación, Java y C++, anteriormente citados, haciendo una breve comparativa para explicar porque nos hemos decantado por el lenguaje usado en este proyecto.

Aunque “*a priori*” puede resultar complejo establecer las ventajas y desventajas de un lenguaje sobre otro sí que hay ciertos parámetros que podemos comparar tales como su sintaxis, su rendimiento o el tipo de tipeo.

4.4.1.2.1 Comparativa de Python con Java

Java es un lenguaje de programación orientado a objetos, basado en clases y de propósito general diseñado para tener menores dependencias de implementación. Es una plataforma informática para el desarrollo de aplicaciones. Por tanto, Java es rápido, seguro y fiable. Es ampliamente utilizado para desarrollar aplicaciones Java en computadoras portátiles, centros de datos, consolas de juegos, supercomputadoras científicas, teléfonos celulares, etc.

Si comparamos los programas por su sintaxis apreciamos que, debido a que Python es un lenguaje interpretado, su sintaxis es más concisa, en términos cuantitativos un programa en Python suele ser de 3 a 5 veces más corto que uno de Java, lo que facilita la puesta en marcha y prueba de programas de forma rápida y sencilla. En Python se puede ingresar líneas directamente en la terminal, mientras que Java necesita compilar todo el programa para ejecutarlo.

Si hablamos de rendimiento Java ofrece un mayor nivel de rendimiento con características como la liberación de memoria, además Python necesita ser interpretado, por lo que es más lento. Por lo tanto, en términos de potencia se puede afirmar que Java es más adecuado para grandes softwares que Python.

Sin embargo, la principal diferencia entre ambos programas radica en la clase de tipeo que utiliza cada uno, mientras que en Python se usa tipeo dinámico, lo que permite a los usuarios el cambio de los tipos de variables, lo que hace que sea más sencillo a la hora de escribir, revisar

y corregir el código, en Java se usa tipo estático, por lo que usuario se ve obligado a elegir un tipo de variable, que no se puede cambiar más adelante en el lenguaje de programación, aunque eso también puede ser percibido como una ventaja ya que puede servir para minimizar la posibilidad de fallos no detectados.

También es importante destacar que ambos programas poseen detrás una comunidad de desarrolladores que comparten sus proyectos y ofrecen su colaboración. En este sentido, ambos son un buen recurso gracias a la actividad de sus comunidades, lo que permite además el acceso a infinidad de librerías con usos muy variados.

Finalmente, si comparamos los diferentes ámbitos y usos que los desarrolladores les dan hoy en día. Python suele ser más empleado en áreas de inteligencia artificial, *big data* o *machine learning*, gracias a su tipo dinámico, mientras que Java se suele usar en aplicaciones de escritorio o móviles y bases de datos.



Figura 4-20 Comparativa Python-Java

Como conclusión, debido a que ningún programa supera al otro significativamente a la hora de desarrollar pequeñas aplicaciones como la de esta memoria, nos inclinamos por Python por su sencillez, lo que facilita la curva de aprendizaje para nuevos programadores y la escritura y revisión del código, y por disponer detrás de una comunidad muy activa de desarrolladores de software de código libre, aunque también hubiese sido perfectamente viable realizarlo en Java.

4.4.1.2.2 Comparativa de Python con C++

Otro de los lenguajes de programación con los que podemos establecer una comparativa es C++, de los más populares y apreciados por los programadores y desarrolladores.

C++ es un lenguaje de programación de propósito general de alto nivel creado como una extensión del lenguaje de programación C. El lenguaje se ha expandido significativamente con el tiempo, y el C++ moderno tiene características funcionales, genéricas y orientadas a objetos, además de facilidades para la manipulación de memoria de bajo nivel.

La principal diferencia entre ambos es la misma que la existente entre Java y Python, mientras que C++ es de tipo estático, Python es de tipo dinámico, con las ventajas y desventajas ya mencionadas en el apartado anterior. Otra diferencia clave, es la compilación mientras que en C++ se necesitará que los programas se compilen en Python no hará falta ya que es un lenguaje interpretado.

En cuanto al propio lenguaje se puede decir que C ++ una sintaxis relativamente difícil y poco amigable en la fase de aprendizaje. No es tan sencillo escribir el código C ++ y que Python es fácil de escribir y tiene una sintaxis clara. Por lo tanto, escribir programas Python es mucho más fácil en comparación con C ++, sobre todo para nuevos programadores.

Otra diferencia significativa es que C ++ no es portátil, es decir, necesitamos recompilar el código en cada plataforma diferente. C ++ es principalmente "Escribir una vez, compilar en cualquier lugar". Python es portátil. También es multiplataforma y podemos ejecutar programas en cualquier plataforma, sigue el principio WORA (*Write Once Run Anywhere*, en inglés escríbelo una vez y ejecútalo en todas partes).

Además, en C ++, la gestión de la memoria es manual, es decir, no admite la recolección automática de recolector de basura, hace una peor gestión de la memoria del equipo. Python, por otro lado, tiene una función de recolección automática de basura. Su gestión de memoria está controlada por el sistema, lo que provoca un mejor aprovechamiento de la memoria.

Sin embargo, en lo que si gana la partida C++ es en lo que a rendimiento y velocidad de procesamiento se refiere. Los programas de C++ se ejecutan más rápido, por ello son muy apreciados en plataformas de videojuegos, y hacen un uso más eficiente de los recursos del sistema. Por lo tanto, Python pierde en esta característica respecto a ambos lenguajes de programación.



Figura 4-21 Comparativa Python- C++

Como conclusión, Python es un lenguaje que ofrece usos más variables en diferentes ámbitos, además posee una sintaxis más sencilla y más amigable para nuevos programadores. Además, el hecho de que sea un lenguaje interpretado y pueda usarse en varias plataformas lo hace muy cómodo. En lo que si se pierde respecto a C++ es en velocidad de ejecución y en rendimiento de los recursos del equipo, aunque como este programa no va a requerir de gran cantidad de recursos nos inclinamos por Python.

4.4.2 Arduino desde el punto de vista de software

4.4.2.1 Arduino IDE

Después de analizar la placa desde el punto de vista del hardware se pasa a analizar la placa desde el punto de vista del software, para ello hablaremos de la IDE de programación de Arduino.

Podemos definir un IDE como un entorno de programación que se empaqueta en forma de archivo ejecutable de ejecución. Es decir, consiste en un programa que aglutina:

- **Editor de código**
- **Compilador y depurador**
- **Interfaz gráfica (GUI)**
- **Herramientas para cargar los programas en la placa**

Los programas se escriben en el IDE de Arduino, se verifican y se compilan y después se cargan en la placa para usarlos. El IDE de Arduino se puede descargar de forma gratuita en la web [12] y permite dos modos de trabajo:

- **Modo de trabajo en línea:** si se dispone de una conexión fiable y rápida a Internet se recomienda utilizar el IDE *online* (Arduino Web Editor, [13]). Nos ofrece la posibilidad de guardar nuestros programas y borradores en la nube y poder acceder a ellos desde cualquier lugar, así como hacer una copia de seguridad. Además, el IDE *online* se actualiza de forma automática por lo que siempre trabajará con la versión más reciente.
- **Modo de trabajo off-line:** si no se dispone a conexión a Internet o se prefiere trabajar en nuestro equipo, es posible descargar la aplicación de escritorio IDE Arduino.

4.4.2.2 Lenguaje de programación de Arduino

El lenguaje de programación de Arduino está basado en uno que ya hemos mencionado anteriormente, el lenguaje de programación C++. Lo que nos permite usar muchos comandos de C++ en programación de Arduino.

El lenguaje de Arduino se puede dividir en tres partes principales:

- **Funciones:** para controlar la placa Arduino y realizar cálculos. Son de tipos muy diversos como: matemáticas (*abs* o *constrain*), digitales (*digitalRead*, *digitalWrite* o *pinMode*), analógicas, interruptoras, de tiempo, de caracteres...
- **Variables:** tipos de datos y constantes. Admite y distingue gran variedad de tipos de datos: array, bool, boolean, byte, char, double, float...
- **Estructura:** los elementos de estructuración de C++. Por ejemplo: loop, bucle while o for, break, continue...

El resultado es un lenguaje de programación sencillo para nuevos usuarios, con una sintaxis simple y una gran variedad de funcionalidades disponibles para usuarios más expertos.

4.4.3 Hamlib

En este apartado trataremos la librería Hamlib [14]. Es un proyecto de código libre y gratuito que está enfocado en el control de antenas, su frecuencia y su orientación.

Hamlib (*Ham Radio Control Libraries*), es un esfuerzo de desarrollo que persigue proporcionar una interfaz consistente y sencilla para todos aquellos programadores que quieren incorporar el control de radio y posicionamiento en sus programas.

No es una aplicación de usuario completa, más bien, es una capa de software destinada a hacer el control de varias radios y otros hardware mucho más fácil. De este modo, permitirá a los autores de programas tales como programas de registro, programas de comunicaciones digitales, o aquellos que deseen desarrollar el último software de control de radio para

concentrarse en la interfaz de usuario y la función básica del programa en lugar de control de radio.

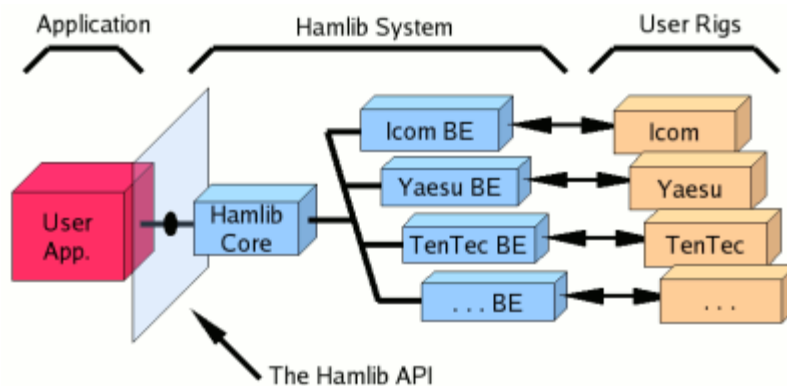


Figura 4-22 Esquema de funcionamiento de Hamlib [14]

Podemos observar en la Figura 4-22 Esquema de funcionamiento de Hamlib que la función que persigue Hamlib es la de dotar al usuario de las herramientas necesarias para controlar la frecuencia y las órdenes de posicionamiento de la antena, y es para ello para lo que lo utilizaremos.

Una característica significativa de Hamlib es que aglutina dos tipos de programa, los programas cliente y programas iguales a los clientes pero que funcionan como servidor, en forma de Daemon. Así, es un compendio de los siguientes programas:

- **Ampctl**: control de amplificadores de radioseñal
- **Ampctld**: servidor Daemon para control de amplificadores de radioseñal
- **Rigctl**: control de transceptores y receptores de radioseñales
- **Rigctld**: servidor Daemon para control de transceptores y receptores de radioseñales
- **Rotctl**: control de rotación de antenas
- **Rotctld**: servidor Daemon para control de rotación de antenas
- **Rigctlcom**: simulador de la plataforma TS-2000 vía puerto COM
- **Rigsmtr**: para controlar una radio para medir el valor del S-Meter (*signal meter* o *signal strength meter*) frente al azimut de la antena. Es decir, calcula el nivel relativo de potencia de un giro en decibelios.
- **Rigswr**: utiliza Hamlib para controlar una radio para medir VSWR (*Voltage Standing Wave Ratio*) en un rango de frecuencia.
- **Rigmem**: copia de seguridad y restaurar la memoria de los transceptores y receptores de radio.

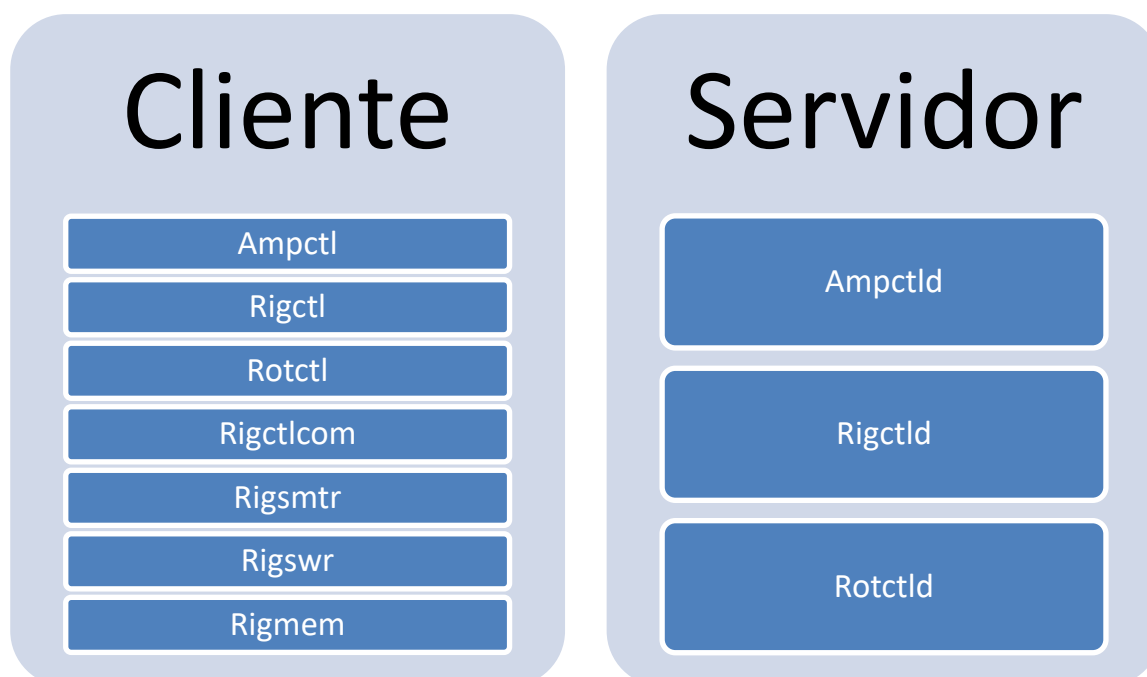


Figura 4-23 Programas de la librería Hamlib

En este proyecto usaremos el servidor Daemon de control de rotores rotctld.

A continuación, vamos a explicar la estructura del programa, utilidades y posición y función en nuestro proyecto.

4.4.3.1 Introducción a rotctld

El programa rotctld es un servidor Daemon ¹⁰de control de rotores que maneja las solicitudes de clientes a través de sockets TCP. Esto permite que varios programas de usuario compartan un rotor. Además, se pueden controlar múltiples rotores simultáneamente en diferentes puertos TCP mediante el uso de múltiples procesos rotctld.

Este programa se enfoca para que sea especialmente útil para programadores que desarrollen sus propios proyectos en funciones de cliente y que utilicen lenguajes como Perl, Python, PHP y otros.

Rotctld se comunica con un cliente a través de un socket TCP usando comandos de texto compartidos con rotctl. El protocolo es simple, los comandos se envían a rotctld en una línea y rotctld responde a los comandos *"get"* con los valores solicitados, uno por línea, cuando tiene éxito, en caso contrario enviará una cadena *'RPRT X'* donde *'X'* será un número negativo indicando el código de error. Los comandos que no devuelven valores se responden con una línea *"RPRT X"*, donde *'X'* es *'0'* cuando la solicitud tiene éxito, de lo contrario es un número de negativo que indica el código de error.

¹⁰ Un servidor Daemon es un trabajo por lotes asociado con un tipo de servidor particular. Sólo hay un daemon de servidor para cada uno de los diferentes tipos de servidor (como base de datos, impresión de red y señal). Cada tipo de servidor tiene una relación de uno a muchos entre servidor y los trabajos de servidor reales; un Daemon permite potencialmente tener muchos trabajos de servidor asociados. El servidor daemon permite a las aplicaciones cliente iniciar comunicaciones con un servidor host que está utilizando soporte de comunicaciones sockets. El daemon del servidor hace esto manejando y enrutando las peticiones de conexión entrantes. Una vez que el cliente establece comunicaciones con el trabajo del servidor, no hay más asociación entre el cliente y el daemon del servidor durante la duración de ese trabajo del servidor. Los subsistemas deben estar activos para utilizar el servidor o servidor de archivos

A continuación, se relacionan los posibles códigos de error que puede devolver rotctld:

CODIGO DE ERROR	NATURALEZA DEL FALLO
RPRT 0	Operación realizada sin error.
RPRT -1	Parámetro inválido
RPRT -2	Error en la configuración
RPRT -3	Fallo por memoria
RPRT -4	Rebasado el tiempo de espera
RPRT -5	Error interno de Hamlib
RPRT -6	Error del protocolo
RPRT -7	Colisión en el bus

Tabla 4-2 Códigos de error de rotctld

4.4.3.2 Inicialización de rotctld

En este proyecto ejecutaremos el servidor desde la terminal de Windows CMD. Para ello debemos darle una serie de parámetros para su inicialización con una estructura muy específica, o de lo contrario tomará los valores por defecto, lo que puede suponer problemas e ineficiencias de diversos tipos.

Por ejemplo, si no le indicamos el puerto IP al que escuchar, lo que hace por defecto es escuchar a todos, perdiendo rendimiento y seguridad. Los principales parámetros que debemos mandar a rotctld son:

- **-m:** selección del modelo de rotor usado, con su protocolo de comunicación asociado. En nuestro caso, será el m 202, que indica el protocolo *EasyCommII*.
- **-r:** selecciona el puerto físico al que nos conectamos. Depende del ordenador, pero en nuestro caso es habitualmente el 'COM 3'.
- **-s:** determina la *baudrate* de transmisión de datos. Por defecto, emplea la máxima del rotor, lo que puede suponer si hay componentes intermedios, como en nuestro caso la placa Arduino.
- **-T:** determina el puerto IP por el que se escucha. Nosotros usamos habitualmente el *localhost* y el de por defecto es todos.
- **-t:** determina el socket TCP por el que se establece la conexión.
- **-v:** nivel de verbosidad aceptado, es decir, el detalle en el que queremos que se realicen los diagnósticos.

Existen más parámetros de inicialización que se pueden consultar en la web, pero no entraremos en ellos pues no son necesarios para este trabajo.

4.4.3.3 Comandos de rotctld

Rotctld comparte los comandos con su homólogo cliente rotctl. Los principales son:

- **P:** para definir la posición a los servomotores. Será de la siguiente manera (“P XXX, XX, XX”) donde el primer número será el ángulo de azimut y el siguiente será el de elevación.
- **p:** sirve para preguntar al rotor en que posición se encuentra. Nos devuelve una cadena igual que la anterior.
- **S:** sirve para detener el movimiento del servo mientras se está realizando, se puede parar uno, el otro o ambos.
- **R:** orden para resetear los servomotores.
- **?:** comando para solicitar el mensaje de ayuda cargado en el programa.

4.5 Esquema global de funcionamiento

A continuación, se muestran de forma muy reducida los esquemas básicos de funcionamiento del sistema en general, primero, y del software posteriormente.

El propósito de este apartado es que sirva como introducción a los posteriores, en los que se detallarán las funciones principales del sistema.

4.5.1 Funcionamiento general del sistema

En este subapartado explicaremos el esquema general de funcionamiento y el orden de actuación de los diferentes módulos del programa.

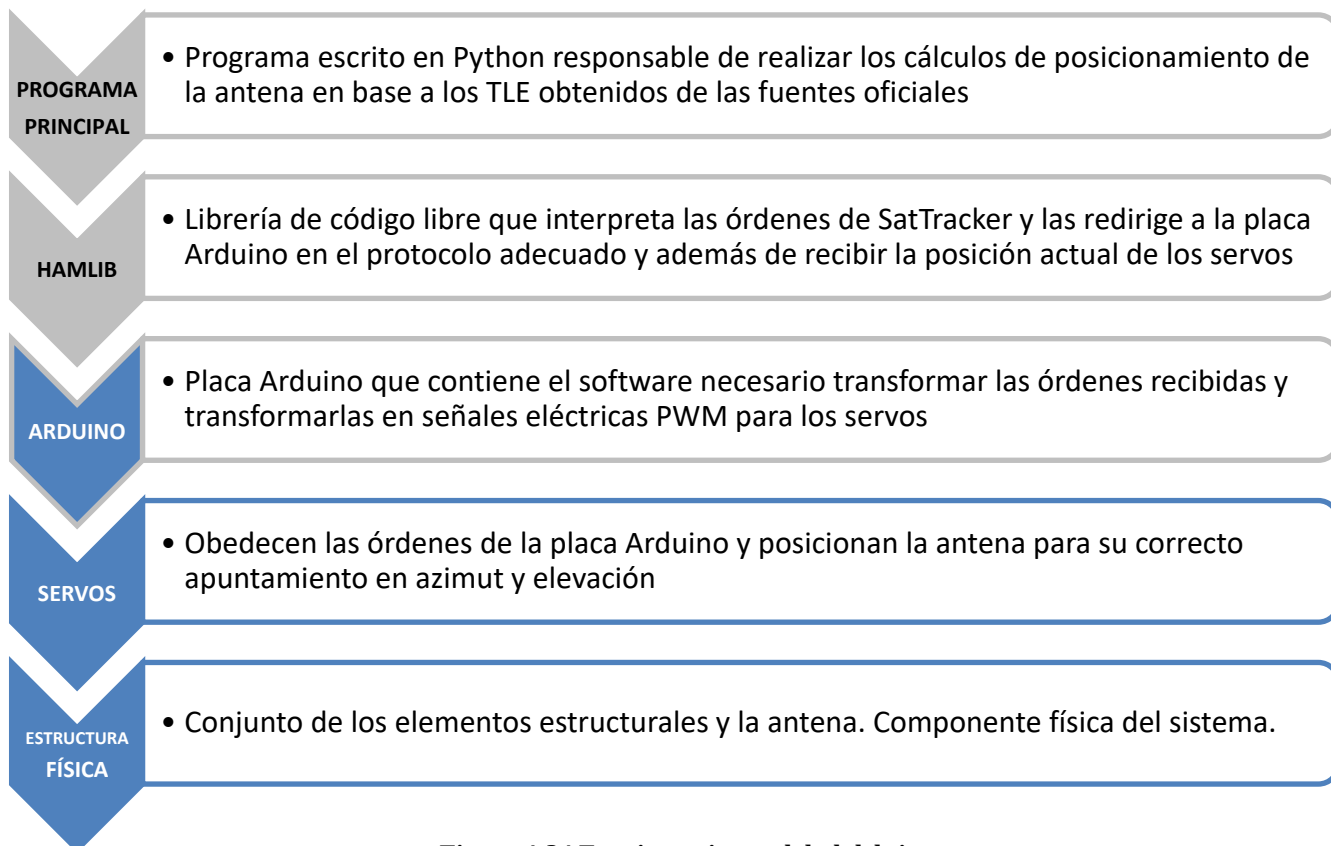


Figura 4-24 Funcionamiento global del sistema

En la Figura 4-24, se pueden ver los componentes principales del sistema con el siguiente código de colores, en gris el software y en azul el hardware.

4.5.2 Comportamiento general del software

Sin entrar aún en detalle, en este subapartado se resumen brevemente las funciones y partes de responsabilidad de cada una de las piezas del software.

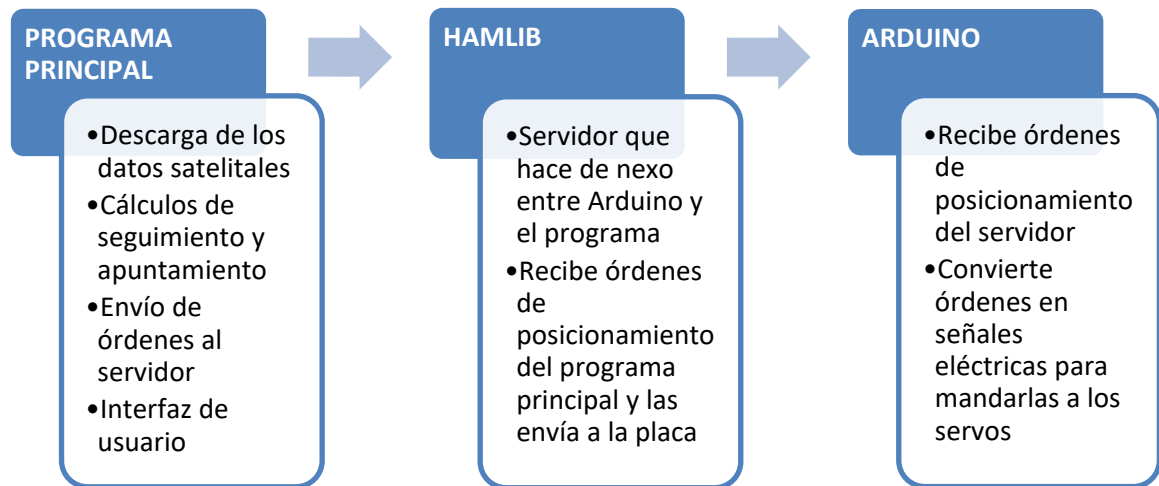


Figura 4-25 Esquema básico de funcionamiento del software

4.5.3 Diseño conceptual

Procedemos a explicar los principales componentes principales del código de nuestro programa principal. Primero explicaremos, de forma breve y concisa, las librerías de Python utilizadas, para, a continuación, entrar a relatar el programa principal, después los diferentes módulos por orden de funcionamiento en el software y finalizando con un diagrama de flujo del sistema.

4.5.3.1 Librerías de Python utilizadas

En este subapartado explicaremos las principales librerías de las que hemos hecho uso en el desarrollo de nuestro software. Nos ofrecen una gran cantidad de funciones de distinta índole y en distintos ámbitos dotándonos de multitud de recursos.

Las librerías usadas en este programa son:

- **Urllib:** es un módulo de Python que se puede utilizar para abrir URLs. Define funciones y clases para ayudar en acciones de peticiones URL. También puedes acceder y recuperar datos de Internet como XML, HTML, JSON... Además, permite para trabajar con estos datos obtenidos directamente.
- **Math:** este módulo proporciona acceso a las funciones matemáticas necesarias para operar con números reales. Aunque estas funciones no se pueden usar con números complejos, sí que estarán disponibles en el módulo *cmath*.
- **Time:** este módulo proporciona varias funciones relacionadas con el tiempo. Para funciones relacionadas, existen también los módulos *datetime* y *calendar*.
- **Socket:** este módulo proporciona las herramientas necesarias para trabajar con *sockets* en los protocolos TCP y UDP. Está disponible en todos los sistemas Unix modernos, Windows y MacOS.

- **Select:** este módulo proporciona acceso a las funciones específicas de monitorización de las entradas y salidas.
- **Keyboard:** nos proporciona las herramientas necesarias para trabajar todo lo relacionado con el teclado: captura de teclas, registro de teclas de acceso rápido, registro de teclas pulsadas...
- **Json:** nos proporciona las herramientas necesarias para trabajar con datos de estructura JSON.
- **Requests:** librería desarrollada para todas las funciones relacionadas con HTTP
- **Ephem:** la librería PyEphem proporciona las herramientas necesarias para realizar cálculos astronómicos avanzados como computación de órbitas de planetas y satélites, cálculo de equinoccios y solsticios... y mucho más. Su autor, Elwood Charles Downey, ha concedido generosamente permiso para que PyEphem se construya sobre su trabajo.

4.5.3.2 Programa principal. Interfaz de usuario (Python)

En este subapartado explicaremos el funcionamiento del programa principal (Menu.py), del que se pueden consultar el código en el Anexo V: Menú.py.

Primero se empiezan importando el resto de módulos del programa, para posteriormente, comprobar la conexión de Internet, para discernir si puede descargar los datos más actualizados o no.

Posteriormente, nos pregunta que elijamos entre tres opciones:

1. **Elegir el host y puerto** introduciéndolo en pantalla
2. **Elegir el host y puerto por defecto** (*localhost*: 127.0.0.1 y puerto: 4533)
3. **Entrar en modo simulación**, que permite pasar al menú principal sin necesidad de tener el servidor activo, facilitando enormemente la revisión del código e introducción de modificaciones.

Desde ahí pasa al bucle principal, en el que se le ofrecen 4 opciones al usuario:

1. **Elegir el satélite** del que se quieran realizar los cálculos, entra en un bucle del que no permite salir hasta haber ingresado una opción válida de la lista de satélites disponibles, y lo guarda en memoria. Además, imprime en pantalla los datos TLE que se van a usar de ese satélite.
2. **Realizar el seguimiento del satélite.** En esta parte el programa llama al calculador para obtener los ángulos de azimut y elevación con la frecuencia de refresco deseada y envía los datos al servidor, comprobando si el envío se realiza satisfactoriamente. Cabe destacar, que si el usuario no ha elegido ningún satélite el programa le obligará a hacerlo antes de realizar el seguimiento.
3. **Posicionar manualmente la antena.** Nos pedirá el ángulo de azimut primero y el de elevación después.
4. **Salir** y cerrar el programa.

Así se termina la explicación del programa principal que, gracias a la definición en módulos y métodos del programa, es fácilmente comprensible y con un código corto.

Posteriormente, procedemos a explicar los diferentes módulos a los que llama el programa principal.

4.5.3.3 Software de comprobación de conexión a Internet y descarga de TLE (Python)

Este programa está encargado en comprobar la conexión a Internet y leer los archivos de los datos TLE, de la página CelesTrak [3], y guardarlos en memoria.

En primer lugar, el programa importa las librerías *requests* y *urllib* para darnos las herramientas para trabajar con datos en línea. A continuación, define la función *Connection* que comprueba si el usuario dispone de acceso a Internet realizando una consulta a Google.

Si dispone de internet avisa al usuario y descarga los últimos datos TLE disponibles, en un fichero de texto llamado 'TLE.txt', sobrescribiendo el archivo que haya en los datos locales o creando uno nuevo. Si no se dispone de Internet, el programa avisa al usuario y lee el fichero de datos locales 'TLE.txt'.

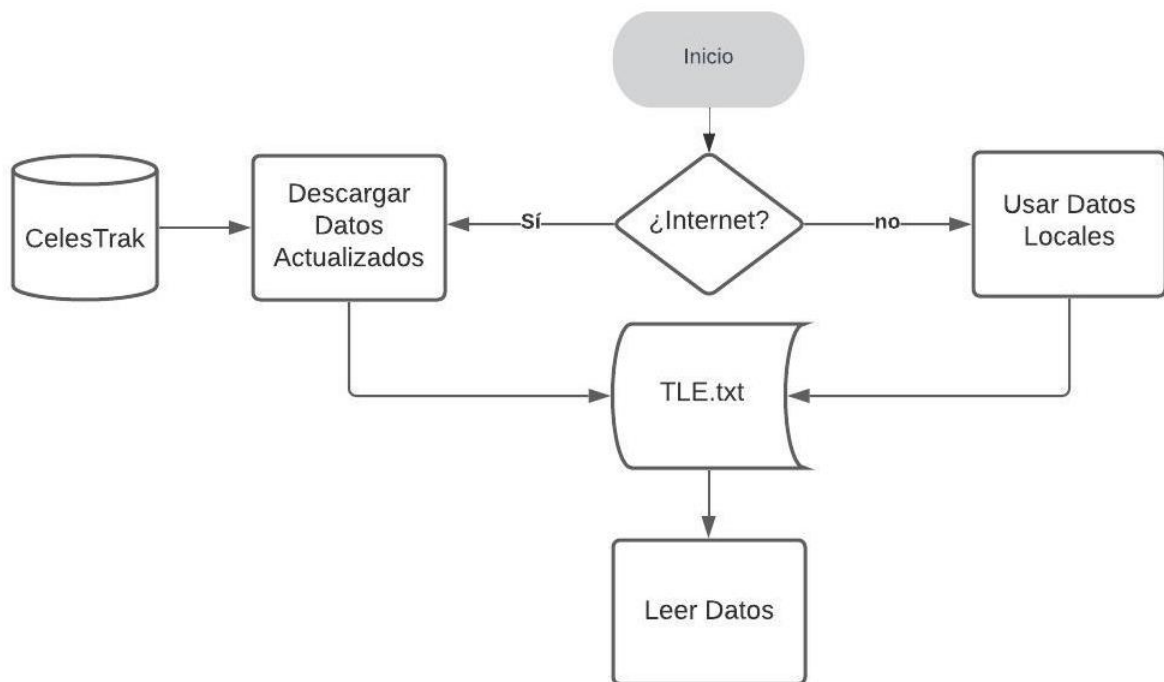


Figura 4-26 Diagrama de flujo del módulo Connection.py

Cabe destacar que, en ambos casos, el programa lee el mismo fichero de texto y que la única diferencia será si ha podido descargar los datos más actualizados o no. También es importante reseñar, que el usuario no tiene necesidad de interactuar con esta parte del programa. Queda resumido en el siguiente diagrama de flujo.

4.5.3.4 Software de conexión TCP con el servidor (Python)

Este programa está encargado de establecer la conexión por TCP con el servidor para enviar las órdenes de posicionamiento y recibir el *feedback* del servidor o, en su defecto, entrar en modo simulación. En él se incluye, además, la parte del código que nos permite posicionar la antena manualmente.

El programa comienza importando las librerías *socket*, *time* y *select*. Para facilitar la estructura en módulos del programa. Esta clase empieza definiendo los valores por defecto de dirección IP '127.0.0.1' y el puerto socket '4533' en la función *init*.

A continuación, en la clase *initialise* se entra en el bucle que le da al usuario las opciones de usar la dirección IP y puerto por defecto, introducirlos por teclado o entrar en modo simulación, sin conectarse al servidor. Posteriormente, llega la función *Send_Data* que envía la variable 'message' codificándola en formato ascii y recibe la respuesta del servidor, detectando posibles errores.

Finalmente, cabe destacar la función *connect*, que es la encargada de realizar la conexión, y reintentarla hasta 3 veces si no lo consigue, y la función *Manual_Mode*, que le da la opción al usuario de introducir manualmente el azimuth y la elevación de la antena deseados.

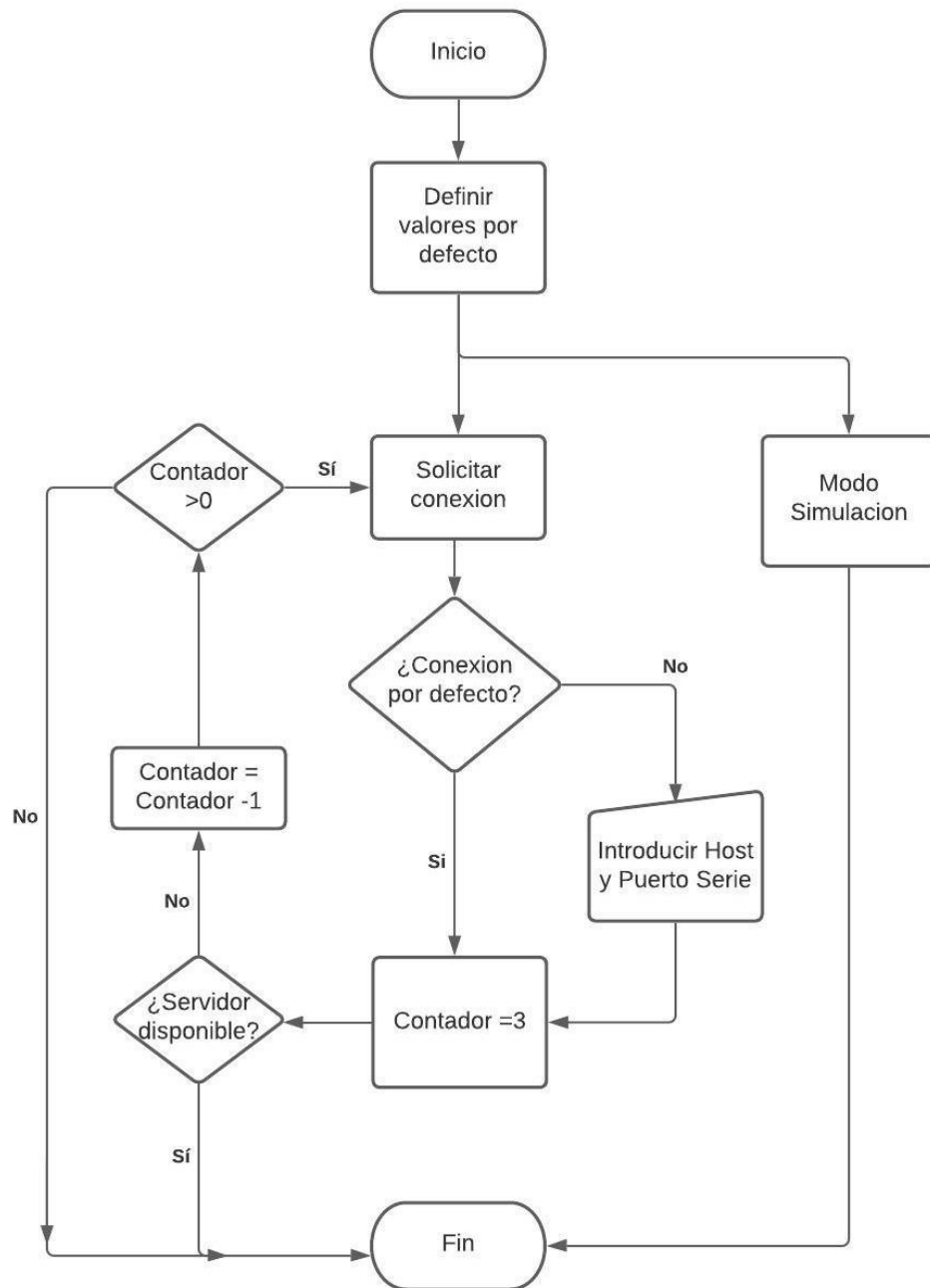


Figura 4-27 Diagrama de flujo de Connection.py

Anexo IX: Client.py

4.5.3.5 Software de lectura de TLE y selección de satélite (Python)

Este software es el encargado de leer los datos TLE de CelesTrak y almacenarlos en forma de biblioteca JSON.

Comienza importando las librerías *json* y *os.path* y, a continuación, define la clase *Satellite*, con el formato característico de los datos TLE, nombre, línea 1 y línea 2.

Después, lee el archivo local TLE.txt con un proceso iterativo de tres en tres filas y almacena cada grupo de tres líneas en una biblioteca json.

Finalmente, le pide al usuario que seleccione entre uno de los satélites introduciendo su nombre por teclado, obligándole a introducir un nombre válido.

Se puede consultar el código en el Anexo VII: Choose_satellite.py.

4.5.3.6 Software de cálculo y seguimiento (Python)

Este módulo, definido como una clase, es el responsable de realizar los cálculos de los ángulos de visión, azimut y elevación, desde el punto de referencia deseado, coordenadas de Pontevedra.

Empieza importando las librerías *time*, *datetime*, *ephem* y el contenido de la librería *math*.

A continuación, se define la clase Tracker en la que se crea el objeto satélite, cuerpo al que referenciar los TLE leídos, estación, la situación del observador, con su longitud, latitud y elevación sobre el nivel del mar.

A partir de ahora se definen una serie de funciones que devuelven los valores de: época, azimut, elevación, latitud y longitud de la pista terrestre y distancia al satélite referida a la estación.

A continuación, se realiza el cambio de sistema de coordenadas (*aer2ecef()*) se transforma las coordenadas esféricas locales en geocéntricas para realizar los cálculos preceptivos para obtener los ángulos de visión. Después, se recuperan las coordenadas (x, y, z) y se vuelve a realizar el cambio a coordenadas polares(*ecef_coordinates()*).

Finalmente, entramos en la función Calculator, que asocia la observación a la época y devuelve el azimut y la elevación del satélite redondeado a dos decimales.

4.5.3.7 Diagrama de flujo del programa principal

En este subapartado, se procede a resumir el funcionamiento del software a través de un diagrama de flujo (Figura 4-28).

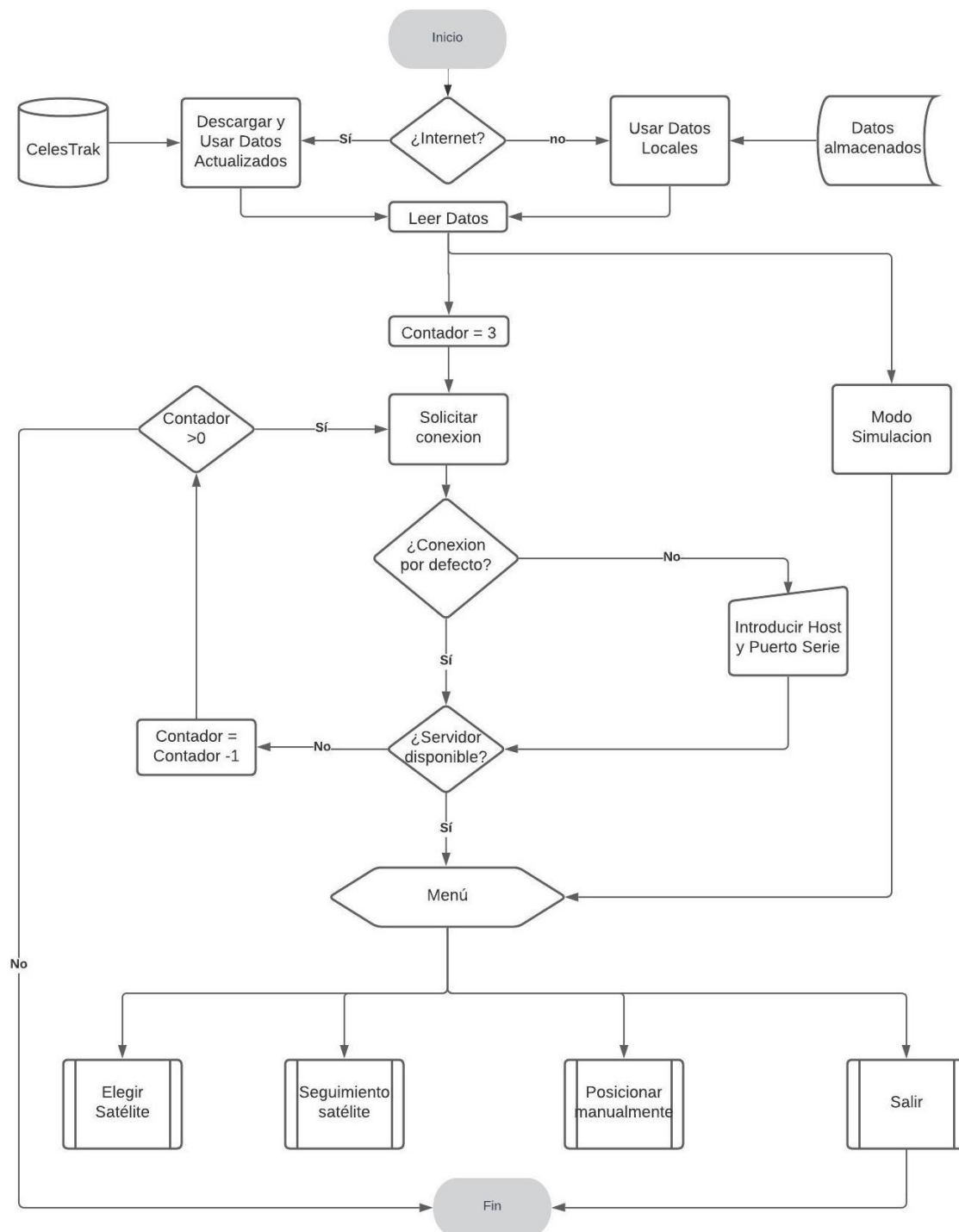


Figura 4-28 Diagrama de flujo del programa principal

4.5.3.8 Software de posicionamiento (Arduino)

Este programa es el que, cargado en la placa Arduino, recibe las órdenes a través del servidor de Hamlib y manda el posicionamiento a los servomotores. Está hecho basado en el programa de licencia abierta SatTrackerII [15].

Incluye las siguientes librerías de Arduino:

- **Servo:** permite a las placas Arduino controlar una gran variedad de servomotores. Puede controlar un gran número de servos, haciendo un uso correcto de los temporizadores: la biblioteca puede controlar hasta 12 servos usando sólo 1 temporizador.
- **EEPROM:** el microcontrolador de la placa basada en Arduino una memoria cuyos valores se mantienen cuando la placa se apaga. Esta librería permite leer y escribir dichos valores.

El programa consta de 6 módulos:

1. **SatTrackerII:** es el módulo principal del programa. En él se definen los parámetros comunes a todo el programa (ángulos máximos y mínimos, rango de PWM, los pines de la placa a los que se conectarán los servos, el 6 y el 9, las posiciones de parada y arranque de los servos, el *baudrate*... Además, define las funciones necesarias para leer la posición actual de los servos y compararla con las órdenes.
2. **EasyCommII:** contiene la lógica real para analizar los comandos definidos por el protocolo *EasyComm II*. Será posible implementar un protocolo o funciones de seguimiento diferentes modificando sólo este archivo.
3. **EEPROM:** lo que este módulo hace es proporcionar funciones para almacenar los valores de los parámetros del sistema en un almacenamiento permanente, de modo que cuando el sistema pierde la alimentación, estos valores no se pierden. Se incluyen los ángulos máximos y mínimos, rango PWM y tiempo de espera.
4. **Help:** ayudará al usuario mostrando todos los comandos soportados, así como los valores actuales de todos los parámetros importantes del sistema.
5. **Response:** es un conjunto de funciones comúnmente utilizadas para enviar respuesta textual al programa de seguimiento. Las funciones simplemente responden con el valor de un parámetro específico del sistema como los ángulos azimut y elevación actuales y los parámetros del sistema.
6. **Servoctrl:** este módulo agrupa las funciones de control de los servos en un solo archivo, para que, si tiene que utilizar un servo diferente, todo lo que se necesita es modificar este archivo.

Con esto quedan definidos todos los elementos que conforman nuestro software.

4.6 Integración y pruebas

En este apartado desarrollaremos las pruebas de nuestro proyecto. En primer lugar, desarrollaremos las pruebas para comprobar el funcionamiento de cada módulo por separado. En segundo lugar, estudiaremos las pruebas extremo-extremo del sistema como conjunto.

4.6.1 Pruebas unitarias

4.6.1.1 Descarga de TLE

Para esta prueba vamos a comprobar que el programa descarga los TLE adecuadamente y los lee en su formato característico para utilizarlos en nuestro software. Para ello vamos a seguir el siguiente proceso:

1. Accedemos a la página web de Celestrak para ver el fichero con los TLE más actualizado (<https://celestrak.com/NORAD/elements/weather.txt>), utilizaremos concretamente el referido a los satélites de meteorología.

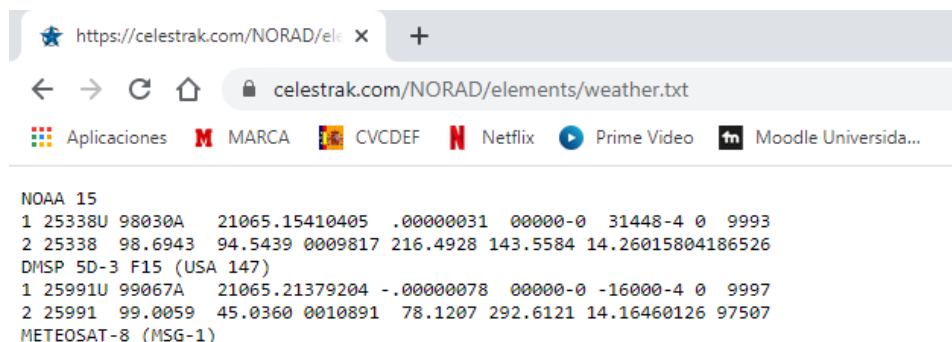


Figura 4-29 Captura de pantalla de Celestrak (16:43 del 6 de marzo)

2. A continuación, vamos a arrancar el programa y seleccionar un satélite cualquiera, tomaremos el NOAA 15 para el ejemplo, y comparar los datos TLE del programa con los de la web, si coinciden quedará demostrado que la descarga se ha realizado correctamente.

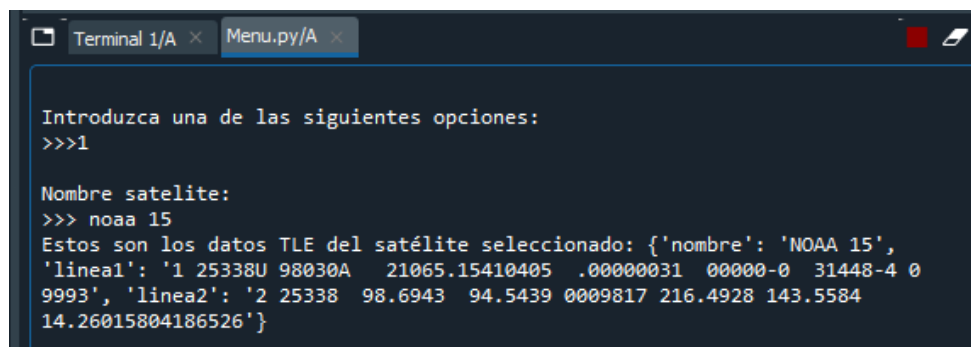


Figura 4-30 Captura de pantalla del programa (16:43 del 6 de marzo)

3. A modo de comprobación complementaria, vamos a comprobar en las propiedades del fichero de texto en el que guardamos los datos su fecha de antigüedad.

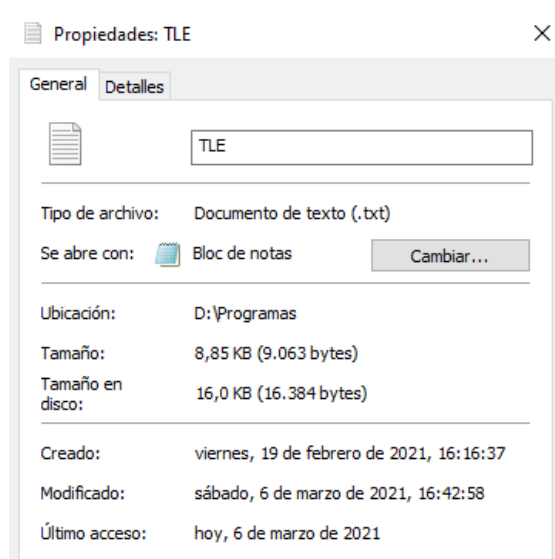


Figura 4-31 Captura de pantalla del fichero TLE.txt

4.6.1.2 Cálculo de apuntamiento

Para realizar este experimento vamos a establecer la comparación de los cálculos de nuestro satélite con los ofrecidos por el programa GPredict, del que hemos hablado con anterioridad, ya que es una fuente fiable y contrastada de datos. Para el ejemplo hemos decidido utilizar el satélite de radioaficionado AO-27.

En la siguiente imagen, vemos a la izquierda de la pantalla los ángulos calculados por el programa Tracker y a la derecha los cálculos del programa GPredict.

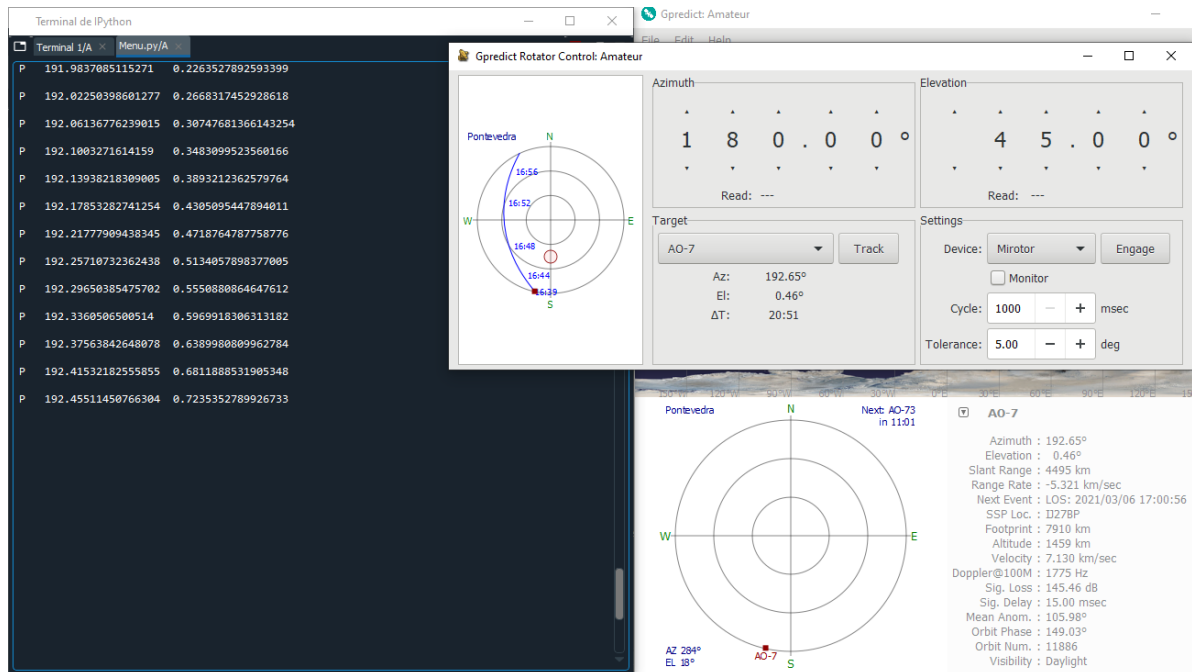


Figura 4-32 Comparativa de los cálculos de Tracker.py y GPredict

Apreciamos que, con un ligero error, de $0,20^\circ$ en azimuth y $0,26^\circ$ en elevación nuestro programa realiza un seguimiento fiable. Este error puede ser debido a diferencias en el instante en el que se realiza el cálculo o a discordancias en la posición respecto a la que se calcula.

4.6.1.3 Comunicación Hamlib

Para este experimento vamos a proceder a enviar los datos calculados por el programa Tracker y comprobar que la respuesta es el código 'RPRT 0' lo que nuestro programa asumirá como una transmisión exitosa, e imprimirá en pantalla "Ok".

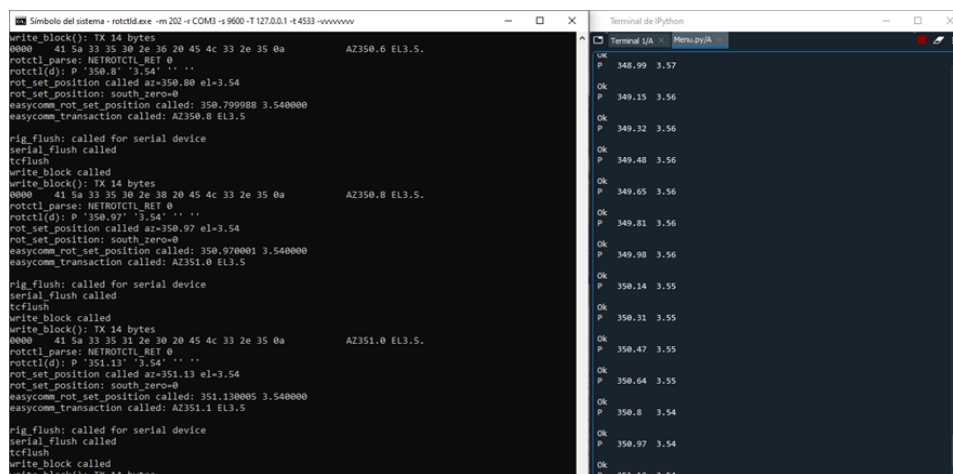


Figura 4-33 Prueba de comunicación con Hamlib

Hemos comprobado que las órdenes se transmiten correctamente al servidor que nos devuelve el código correspondiente al éxito.

4.6.1.4 Pruebas del posicionador mecánico

En este experimento se pretende comprobar que se puede realizar un control preciso y exacto de los servomotores, así como de comprobar la precisión que alcanza el programa.

En una primera prueba, hemos optado por usar el mismo programa que utilizamos en la calibración del servo, hemos ido modificando la señal microsegundo a microsegundo extrayendo los siguientes datos:

- El rango PWM real de los dos servos es 500-2225 microsegundos, corrigiendo al nominal (500-2500) ofrecido por el vendedor.
- El recorrido real de los servos es de unos 630° y no de 640°, datos nominales.
- El rango PWM del servo de azimut correspondiente a una revolución completa es de 500-1595 μ s, mientras que el rango PWM para una revolución completa del servo de elevación es de 500-1578 μ s.
- La sensibilidad del servo a los cambios de señal alcanza hasta los 3 μ s, es decir, percibe la señal 500, 501 y 502 μ s como iguales y no se moverá hasta recibir una señal de duración 503 μ s.

En una segunda prueba, ya con el programa SatTrackerII, hemos decidido mapear el comportamiento de los servos ante las señales PWM recibidas, llegando a las siguientes soluciones:

- El servo de azimut tendrá un recorrido de 0 a 360° correspondiente al rango PWM 500-2225 μ s. Es decir, el rango superior y la revolución extra no serán utilizados.
- El servo de elevación tendrá un recorrido de 0 a 90°, ya que no necesita más porque son los valores entre los que estará comprendido en cualquier caso la elevación, correspondiente al margen PWM 500-2225 μ s, por razones de montaje se opta por desplazar del origen esos valores.

En una tercera prueba hemos decidido comprobar el correcto funcionamiento del control, así como el correcto posicionamiento obteniendo resultados satisfactorios. De esta tercera prueba concluimos, además, que la máxima precisión que nos ofrece SatTrackerII es de 3 décimas de grado, que será nuestro grado de precisión máximo.

Finalmente, hemos comprobado que desde el programa principal se pueda realizar el posicionamiento manual de la antena obteniendo resultados satisfactorios igualmente.

4.6.2 Pruebas de integración

En el presente apartado desarrollaremos las pruebas realizadas para comprobar el funcionamiento del sistema como conjunto.

4.6.2.1 Prueba extremo-extremo del software

Vamos a comprobar el funcionamiento del sistema de forma conjunta. Para ello:

1. Conectamos ambos servos y la placa Arduino a la alimentación eléctrica de 6V y conectamos las señales de control de los servos a los pines correspondientes.
2. Arrancamos el servidor Daemon de Hamlib.
3. Arrancamos el programa principal.

En primer lugar, se realiza la verificación de la conexión a datos en línea y la descarga de datos más actualizado. Hemos decidido hacerlo conectados a Internet para tener los cálculos más fiables, pero podría haberse realizado igualmente sin conexión.

En segundo lugar, arrancamos el programa principal y seleccionamos el puerto y host por defecto y comprobamos que tanto en Hamlib como en el software principal aparezca el mensaje de conexión abierta.

En tercer lugar, comprobamos que podemos mover la antena manualmente, mandándole órdenes de azimuth y elevación desde el software principal. También, le mandamos las órdenes en forma de cálculos de seguimiento de un satélite, comprobando que se realizan ambas sin problemas.

Queda demostrado que la conexión extremo-extremo se realiza satisfactoriamente.

4.6.2.2 Prueba final: seguimiento de un pase

Realizamos el mismo procedimiento que el anterior hasta el punto en el que tenemos la conexión abierta. A partir de ahí, seleccionaremos un satélite y abriremos el programa GPredict para comparar los resultados.



Figura 4-34 Antena realizando un pase

Con el programa principal arrancado y mandando las órdenes a la plataforma y el programa GPredict realizando los cálculos y el seguimiento de manera virtual comprobamos que los ángulos ordenados sean iguales y se realiza el seguimiento satisfactoriamente.

Así, hemos demostrado que el programa es capaz de realizar el seguimiento de manera precisa y fiable.

5 RESULTADOS

*La mayoría de las personas gastan más tiempo y energía
en rondar los problemas que en tratar de resolverlos
-Henry Ford*

5.1 Descripción del apartado

En el presente apartado se muestran los resultados obtenidos en algunas de las pruebas realizadas.

5.2 Ejemplo de un pase de un satélite NOAA

Para este experimento vamos a intentar recuperar los datos de un satélite, para ello haremos el seguimiento y apuntamiento con nuestra plataforma, y descargaremos el Downlink con un software añadido, el SDR-Sharp, que activa la recepción en la antena y nos permite sintonizarlo en la frecuencia del satélite deseado.

En primer lugar , se debe realizar un estudio cuidadoso de la elección del satélite para planificar el pase y configurar el software para su frecuencia de Downlink. Para este ejemplo , hemos decidido seleccionar el satélite NOAA 19 pues tiene una órbita polar por lo que su recorrido será el mejor posible sobre nosotros y transmiten a altas potencias por lo que será más fácil captar la señal.



Figura 5-1 NOAA-19 (Fuente: Lockheed Martin)

El NOAA-19 es un satélite meteorológico de la NOAA que describe una órbita polar de Sur a Norte, a unos 850km de altura, con un periodo orbital de aproximadamente 1 hora y 40 minutos. Transmite con polarización centrado en la frecuencia 137.100MHz con un ancho de banda de 36kHz.

Es importante destacar que habrá que tener en cuenta el desplazamiento de la señal provocado por el efecto Doppler, en nuestro caso se corregirá mediante el uso del software GPredict, que será el encargado para este caso de mandar las órdenes de posicionamiento a la antena y hacer los cálculos para corregir de forma automática el desplazamiento Doppler.

Con el uso de ambos softwares de forma simultánea se ha podido explotar el pase del NOAA realizando un apuntamiento preciso en tiempo real, corregido por efecto Doppler en tiempo real también, y descargando su transmisión.

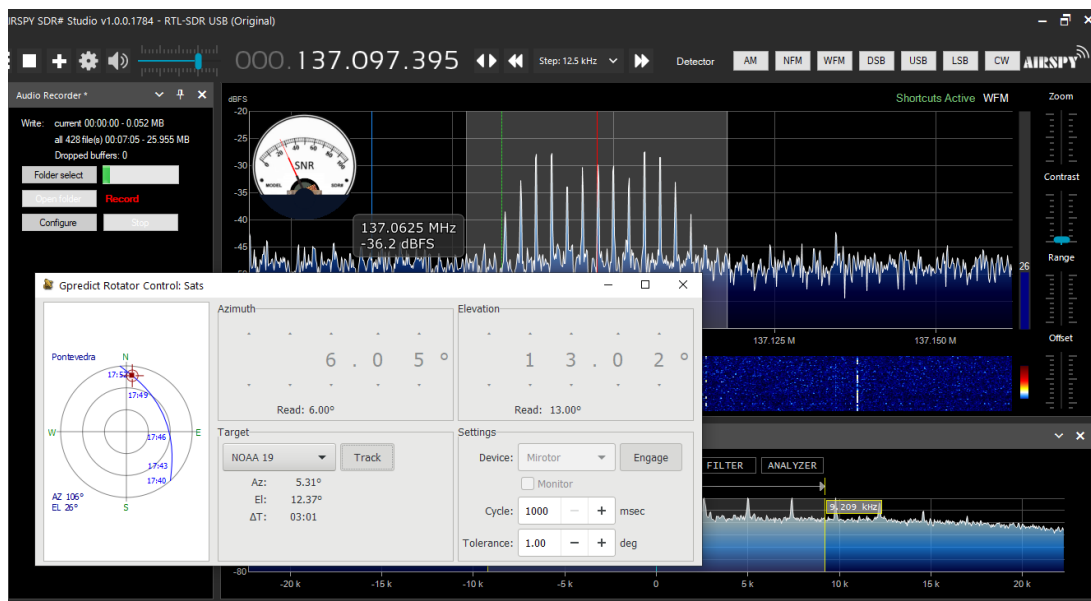


Figura 5-2 Prueba de pase con los softwares GPredict y SDR-Sharp

Para explotar aún más el objetivo de este experimento se puede descargar el programa WXtollmg que se encarga de decodificar los datos y analizarlos para, como ejemplo para este caso, obtener las fotografías e imágenes capturadas por este satélite.

Aunque se ha intentado realizar en esta prueba, debido a no tener visibilidad al ser de noche, las imágenes analizadas del pase no tienen valor.

También se pretendía haber realizado un experimento similar con la ISS sin embargo, no ha sido posible debido a averías internas del satélite.



Figura 5-3 Pase del NOAA-19

6 CONCLUSIONES

6.1 Descripción del apartado

En este apartado se hará una síntesis del sistema conseguido y el grado de satisfacción de los objetivos impuestos.

6.2 Plataforma

Se han desarrollado concurrentemente el software y el hardware.

El software consta de una serie de programas, por el procedimiento explicado con anterioridad, comprueba si dispone de acceso a Internet o no para utilizar los datos más actualizados si es posible, establece una comunicación por socket con un servidor, permitiendo alternativamente un modo de simulación, que a través de una placa Arduino controla dos servomotores que son los encargados de sustentar y posicionar nuestro sistema de antenas.

El hardware consta de los elementos mencionados, trípode, servos, antenas y alimentación que ha habido que unir por medio de diferentes agarres y sujeciones. A lo largo de este proceso, se ha realizado el calibrado y mapeo de las distintas posiciones de los servos en función de la señal eléctrica enviada y se ha realizado un estudio de las cargas y fuerzas que puede soportar nuestro sistema y el dimensionamiento de los distintos componentes.

6.3 Conclusión

De este modo, se ha conseguido desarrollar y montar una plataforma mecánica ligera que realiza el seguimiento y cálculo de órbitas de satélites de forma precisa y fiable por medio de un software de desarrollo físico-matemático. Este sistema es, además, independiente de la conexión a Internet. En cuanto a los elementos se ha equipado con una antena directiva que trabaja en las bandas UHF y VHF y con una serie de materiales ligeros y duraderos.

En definitiva, se puede decir que se han desarrollado los objetivos marcados al principio de esta memoria con un elevado grado de exactitud.

En el siguiente capítulo, se discutirá sobre posibles líneas futuras para seguir ampliando las capacidades de este producto.

7 LÍNEAS FUTURAS

En el presente capítulo se describen una serie de complementos, mejoras o nuevas funcionalidades que se pueden estudiar para aportarlas a este proyecto.

7.1 Hardware

Las principales líneas de mejora de este proyecto a nivel de hardware deben estar enfocadas en aumentar la rigidez y la plataforma de la estructura, usando un soporte más fuerte que el actual. También se deberá trabajar en el aislamiento mecánico y térmico de los componentes para aumentar su durabilidad y que pueda trabajar en ambientes hostiles. Además, su capacidad de integración en todo tipo de vehículos como carros o buques significaría un salto de calidad a este ingenio.

Se puede estudiar, también, el añadir un sistema de orientación para la plataforma, como una aguja magnética integrada, para facilitar las operaciones de orientación previas a su despliegue.

Equipar al sistema con servos más potentes y veloces permitiría una mayor variedad en las antenas que el sistema puede soportar, aumentando el espectro de posibilidades y permitiendo planificar enlaces en bandas más concretas.

Sería conveniente, equipar nuestro sistema con un sistema de transmisión de datos para no limitarnos solo a la recepción haría que aumentasen exponencialmente sus utilidades, al dotarnos de un sistema de comunicaciones satelitales barato, robusto y sencillo.

Por último, sería de gran utilidad acoplar al sistema una batería portátil de tamaño reducido para no tener que depender de ninguna red eléctrica.

7.2 Software

En cuanto el nivel de software se refiere la principal mejora que se debería aplicar es la integración de un método de posicionamiento GPS para que nuestra estación tenga una entrada sobre su ubicación precisa y en tiempo real, facilitando enormemente su despliegue.

Además, relacionado con una de las mejoras a nivel de hardware anteriormente mencionadas, se debería estudiar la posibilidad de añadirle un sistema IMU (*Inertial Measure Unit*), sistemas muy ligeros y de bajo coste que permiten, entre otras muchas cosas, cálculos de velocidad y aceleración, de orientación o medición de temperaturas, de esta forma el satélite podría tener una referencia de donde se encuentra el norte verdadero y auto posicionarse.

Añadirle al software la capacidad de predecir, almacenar y mostrar en pantalla los pases futuros de los cuerpos satelitales para mejorar la experiencia del usuario también podría ser un proyecto recomendable.

Otra mejora digna de estudio sería la de hacer posible que se realice el cálculo de la órbita de varios satélites al mismo tiempo para, bien controlar varias antenas concurrentemente, o bien poder saltar de un satélite con muy poca dificultad.

7.3 Conclusiones

Existen varias líneas de investigación posible para mejorar este trabajo, en cuanto a hardware, principalmente en cuanto a estabilidad, rigidez y durabilidad del montaje, la potencia y precisión de los servos o la posibilidad de transmisión, y software, aportar una entrada GPS para que la plataforma disponga de una entrada respecto a la que hacer los cálculos y darle la capacidad de auto orientarse o la multifunción.

Sin embargo, no conviene olvidar que se debe preservar el enfoque del estudio que es el de hacer una plataforma ligera y transportable por lo que estas mejoras no deberían suponer un aumento significativo del peso.

8 BIBLIOGRAFÍA

En este capítulo se desarrollarán todas las referencias empleadas a la hora de elaborar este trabajo entre los que se incluyen libros, páginas web o artículos de revista.

«Satbeams,» [En línea]. Available: <https://www.satbeams.com/>. [Último acceso: 24 Febrero 1] 2021].

Escuela Naval Militar, *Comunicaciones satélite en el TEAR. Conceptos SOTM y SATQH*, 2018-2] 2019.

D. T. Kelso, «CelesTrak,» 2021. [En línea]. Available: <https://celestrak.com/>. [Último acceso: 3] 20 Enero 2021].

N2YO, «N2YO.com,» N2YO, 2021. [En línea]. Available: <https://www.n2yo.com/>. [Último 4] acceso: 21 Enero 2021].

Asociación de Radioaficionados RU, «AMSAT-UK,» [En línea]. Available: <https://amsat-uk.org/>. [Último acceso: Marzo 2021].

«AMSAT Satellites pass prediction,» [En línea]. Available: <http://lu7abf.com.ar/pass.htm>. 6]

Free Space Foundation, «SatNOGS,» [En línea]. Available: <https://satnogs.org/>. [Último acceso: 7] 27 Febrero 2021].

Vito Technology, «Satellite Tracker,» [En línea]. Available: 8] <https://vitotechnology.com/apps/satellite-tracker>.

S. Stoeffl, «Orbitron - Satellite Tracking System,» [En línea]. Available: <http://www.stoff.pl/>. 9]

«Amazon,» Newer, [En línea]. Available: https://www.amazon.es/Neewer-Port%C3%A1til-Aleaci%C3%B3n-Giratorio-Videoc%C3%A1mara/dp/B01N6JCW8F?ref_=ast_sto_dp&th=1. 10]

Spyder Inc, «Spyder.org,» [En línea]. Available: <https://www.spyder-ide.org/>. 11]

«arduino.cc,» [En línea]. Available: <https://www.arduino.cc/>.

12

]

«Arduino Creator,» [En línea]. Available:
13 <https://login.arduino.cc/login?state=g6Fo2SB2dFRkWVhmS25UY2s3OHZXQ2FYejJqaXNmbFJQMGI3b6N0aWTZIExFdkRjYlJ0Vlk4b2hvbW54a3JoWVdnQVRZaGd5ZWxGo2NpZNkgcnI2TFNPU3o1N2g3ZnZzSGRSWFhQa1pHM0VSb1hUZlo&client=rr6LSOSz57h7fvsHdRXXPkZG3ERoXTfZ&protocol=oauth2&authorizeT>.

F. Singleton y S. Fillode, «SourceForge,» 2020. [En línea]. Available:
14 <https://sourceforge.net/p/hamlib/code/ci/master/tree/>.

]

U. Ghodke, «Radio Amateur Diary, K6VUG,» 7 Julio 2020. [En línea]. Available:
15 <https://sites.google.com/site/k6vugdiary/satellite-tracking-system/satellite-tracker-controller>.

]

«ServoCity,» [En línea]. Available: <https://www.servocity.com/sg12-series-servo-gearbox-7-1-16-ratio-640-rotation-2450-oz-in-1-12-sec-60/>.

]

«STATISTA,» STATISTA, [En línea]. Available:
17 <https://es.statista.com/grafico/18491/satelites-que-se-encuentran-orbitando-alrededor-de-la-tierra/>.
] [Último acceso: 21 Enero 2021].

GPredict, «GPredict,» 2021. [En línea]. Available: <http://gpredict.oz9aec.net/features.php>.

18

]

N. V. Patel, «MIT Technology Review,» MIT, 26 Febrero 2020. [En línea]. Available:
19 <https://www.technologyreview.es/>.

]

D. Salces, «Muy Computer,» 30 Noviembre 2020. [En línea]. Available:
20 <https://www.muycomputer.com/2020/11/30/basura-espacial-esa-agencia-europea/>. [Último acceso:
] 26 Enero 2021].

bsomervi, kamalmostafa, mdblack98, n0nb, «Source Forge,» 16 Julio 2000. [En línea].
21 Available: <https://sourceforge.net/projects/hamlib/>. [Último acceso: 28 Enero 2021].

]

B. C. Rhodes, «PyEphem Homepage,» 2020. [En línea]. Available:
22 <https://rhodesmill.org/pyephem/>. [Último acceso: 28 Enero 2021].

]

F. Stra, «TLE-tools,» 2019. [En línea]. Available: <https://tletools.readthedocs.io/en/latest/#>.
23 [Último acceso: Enero 2021].

]

UC Santa Barbara Physics Department, [En línea]. Available:
24 <http://web.physics.ucsb.edu/~mhlee/resonances.html>. [Último acceso: 3 Febrero 2021].

]

M. Leomanni, «ResearchGate,» Diciembre 2017. [En línea]. Available:
25 https://www.researchgate.net/figure/Classical-orbital-elements_fig1_315734460.
]

Python, «Python.org,» [En línea]. Available: <https://www.python.org/>.
26
]

Anaconda Inc, «Anaconda,» [En línea]. Available: <https://www.anaconda.com/#>.
27
]

D. Robinson, «The Overflow,» StackOverflow, 6 Septiembre 2017. [En línea]. Available:
28 <https://stackoverflow.blog/2017/09/06/incredible-growth-python/>.
]

«European Spatial Agency,» Unión Europea, [En línea]. Available: <https://www.esa.int/>.
29 [Último acceso: 23 Febrero 2021].
]

«INMARSAT,» [En línea]. Available: <https://www.inmarsat.com/en/index.html>. [Último
30 acceso: 24 Febrero 2021].
]

«Hispasat,» [En línea]. Available: <https://www.hispasat.com/es>. [Último acceso: 24 Febrero
31 2021].
]

«Rheinmetall Defence,» [En línea]. Available:
32 [https://www.rheinmetall.ca/en/rheinmetall_canada/publicrelations_1/news_1/2014_10_07_sotm.p](https://www.rheinmetall.ca/en/rheinmetall_canada/publicrelations_1/news_1/2014_10_07_sotm.php)
] hp. [Último acceso: 25 Febrero 2021].

Indra, «SATQH,» [En línea]. Available:
33 https://www.indracompany.com/sites/default/files/indra_satcom_atqh_en_baja.pdf. [Último
] acceso: 25 Febrero 2021].

B. Elbert, The Satellite Communication Ground Segment and Earth Station, 2014.
34
]

«Armada Española,» [En línea]. Available:
35 [https://armada.defensa.gob.es/ArmadaPortal/page/Portal/ArmadaEspañola/iniciohome/prefLang-](https://armada.defensa.gob.es/ArmadaPortal/page/Portal/ArmadaEspañola/iniciohome/prefLang-es/)
] es/. [Último acceso: 27 Febrero 2021].

Orbitack, [En línea]. Available:
36 <https://play.google.com/store/apps/details?id=com.southernstars.orbitrack&hl=es&gl=US>.
]

R. A. Braeunig. [En línea]. Available: <http://www.braeunig.us/space/orbmech.htm>. [Último
37 acceso: 28 Febrero 2021].
]

Dpto. de Señales, Sistemas y Radiocomunicaciones. Universidad Politécnica de Madrid,
38 2009/2010. [En línea]. Available: [https://list.gr.ssr.upm.es/docencia/grado/csat/material/CSAT09-](https://list.gr.ssr.upm.es/docencia/grado/csat/material/CSAT09-2-MecanicaOrbital_Parte2.pdf)
] 2-MecanicaOrbital_Parte2.pdf. [Último acceso: 01 Marzo 2021].

C. M. Corbera y C. M. Jiménez, Astronomía y Navegación, San Fernando: Librería San José, 39 1968.

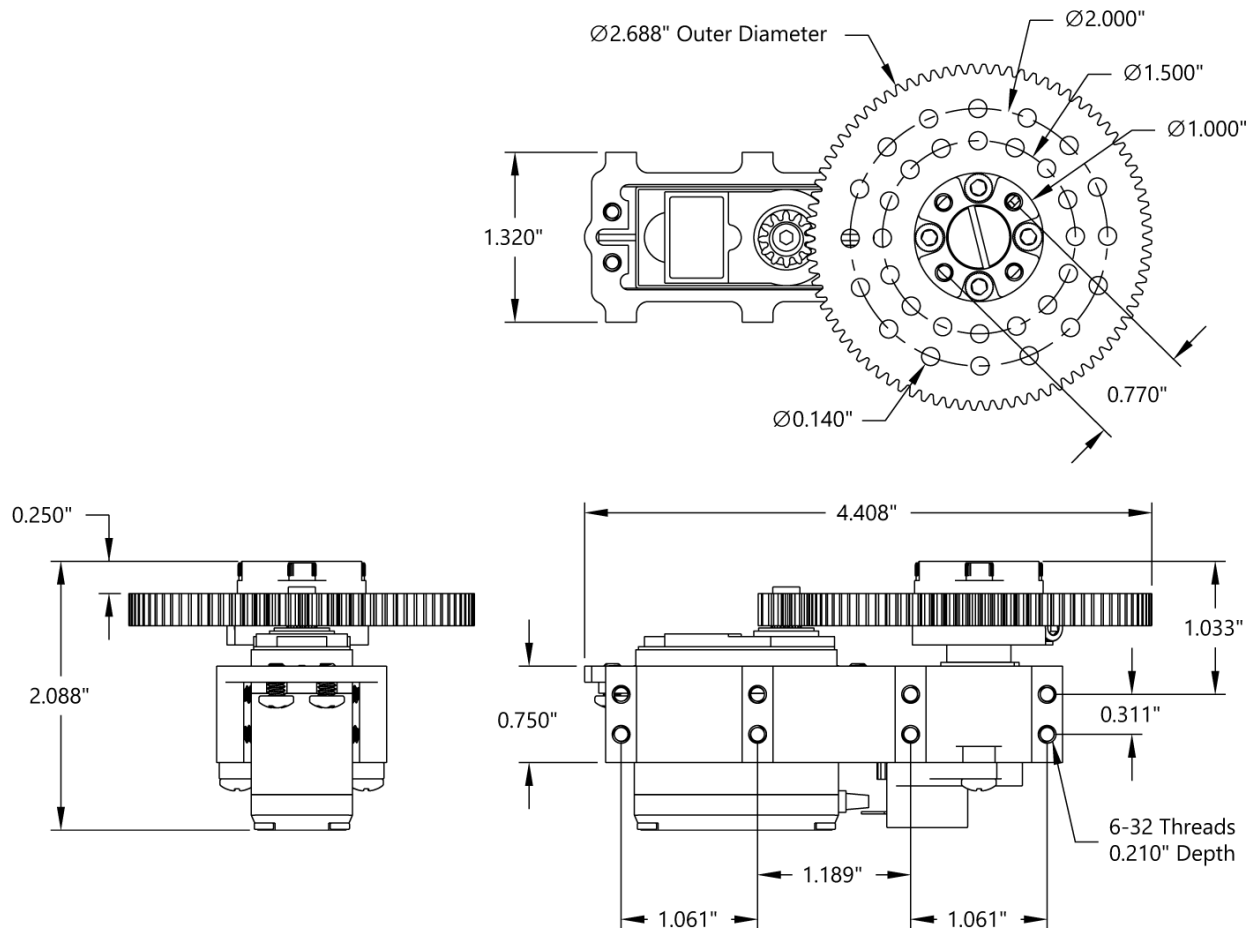
]

«ingmecanica,» [En línea]. Available:
40 <https://ingemecanica.com/tutorialsemanal/tutorialn32.html>. [Último acceso: 7 Marzo 2021].

]

ANEXO I: PLANOS DEL SERVOMOTOR SG12-70

En el presente anexo se muestran los planos de los servomotores usados en este trabajo ofrecidos por la empresa vendedora. [16]



ANEXO II: ESPECIFICACIONES DEL SERVOMOTOR SG12-70

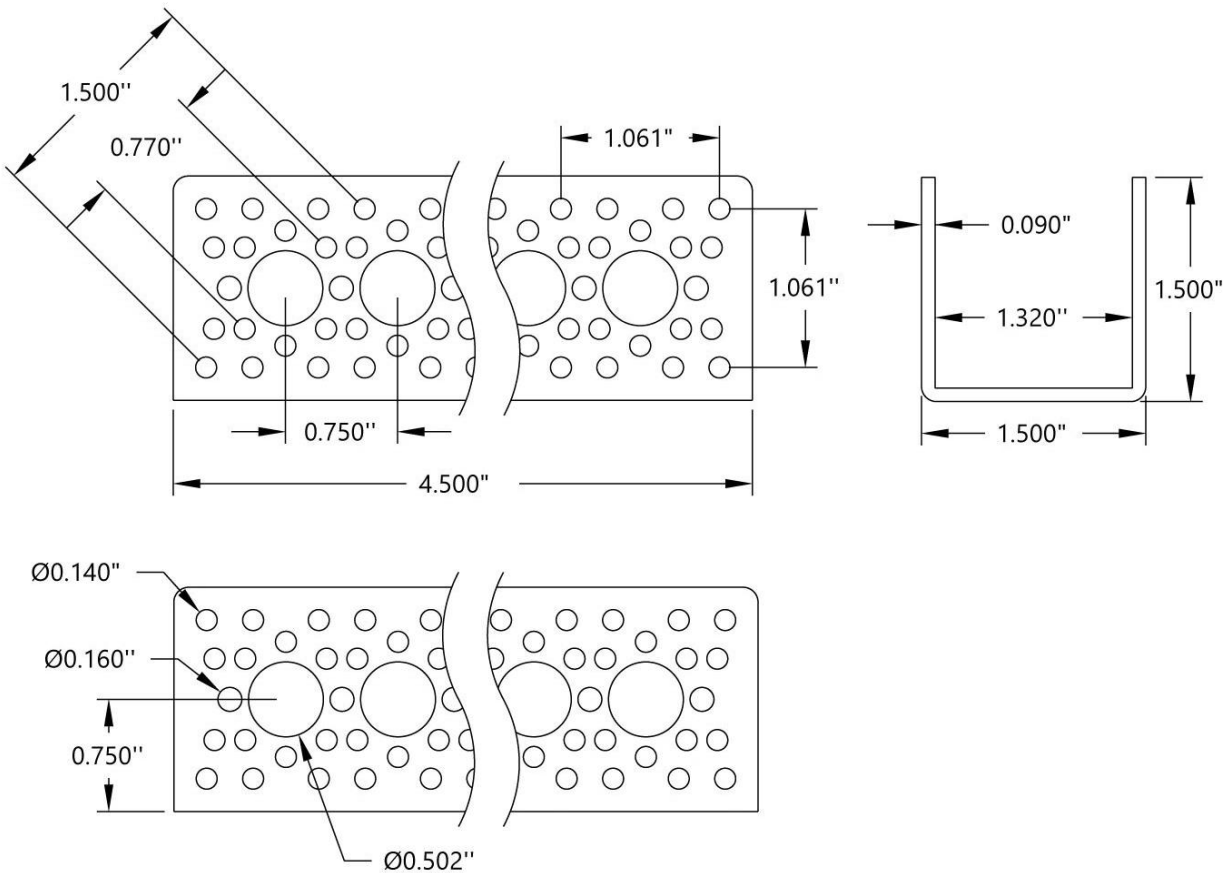
En el presente anexo se muestra la tabla de especificaciones de los servomotores empleados en este trabajo ofrecido por la empresa vendedora [16]

Weight	6.9 oz (194g)
Gear Ratio	7:1
Voltage Range	4.8V - 7.4V
No-Load Speed (4.8V)	1.75 sec/60°
No-Load Speed (6.0V)	1.4 sec/60°
No-Load Speed (7.4V)	1.12 sec/60°
Stall Torque (4.8V)	1680 oz-in (121.0 kg.cm)
Stall Torque (6.0V)	2100 oz-in (151.2 kg.cm)
Stall Torque (7.4V)	2450 oz-in (176.4 kg.cm)
No-Load Current (4.8V)	150mA
No-Load Current (6.0V)	160mA
No-Load Current (7.4V)	200mA
Stall Current (4.8V)	2,000mA
Stall Current (6.0V)	2,500mA
Stall Current (7.4V)	3,000mA
Max PWM Range	500-2500µsec
Travel per µsec	0.321°/µsec
Max Rotation	642°
Pulse Amplitude	3-5V
Deadband Width	4µsec

ANEXO III: PLANOS Y ESPECIFICACIONES DE LOS ELEMENTOS ESTRUCTURALES

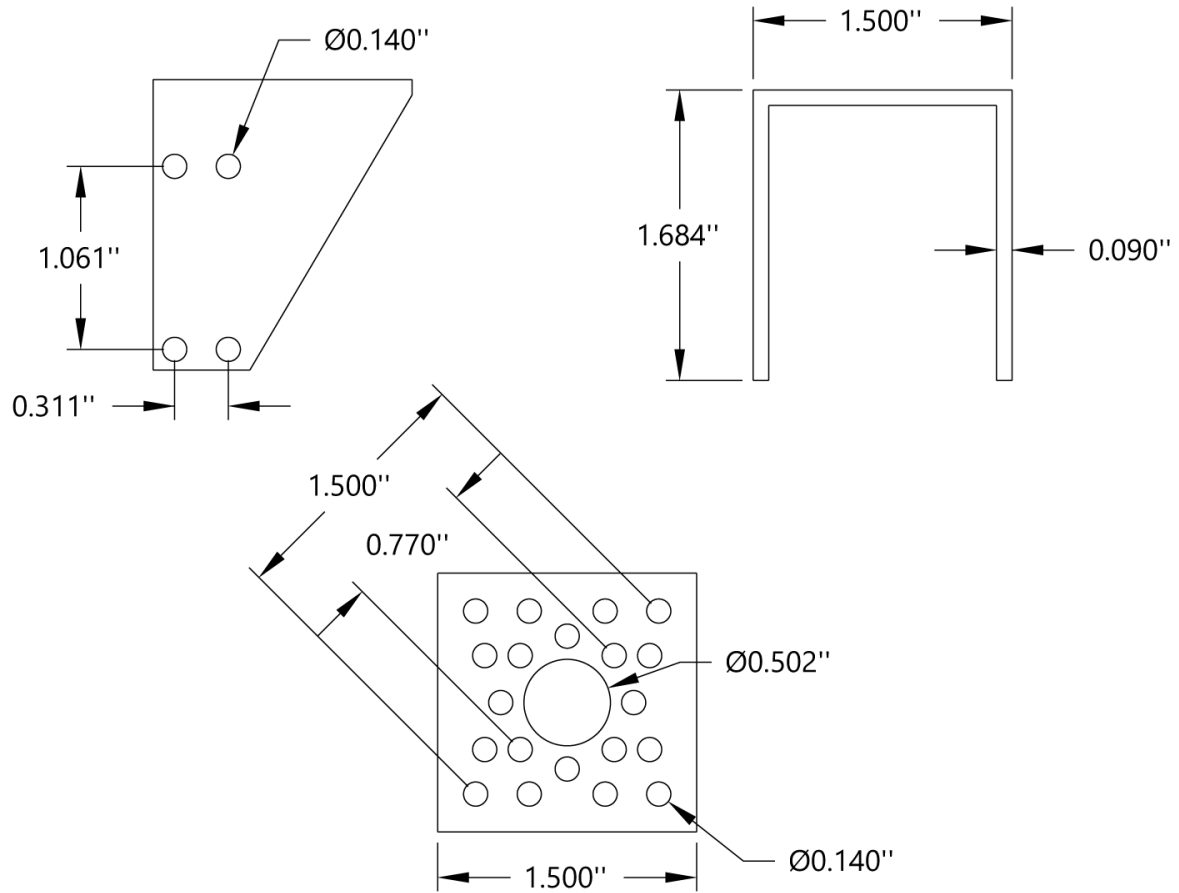
En el presente anexo se presentarán los planos y especificaciones de los principales elementos estructurales del sistema, ofrecidos por la empresa vendedora. [16]

1. Canal en U de 5 agujeros



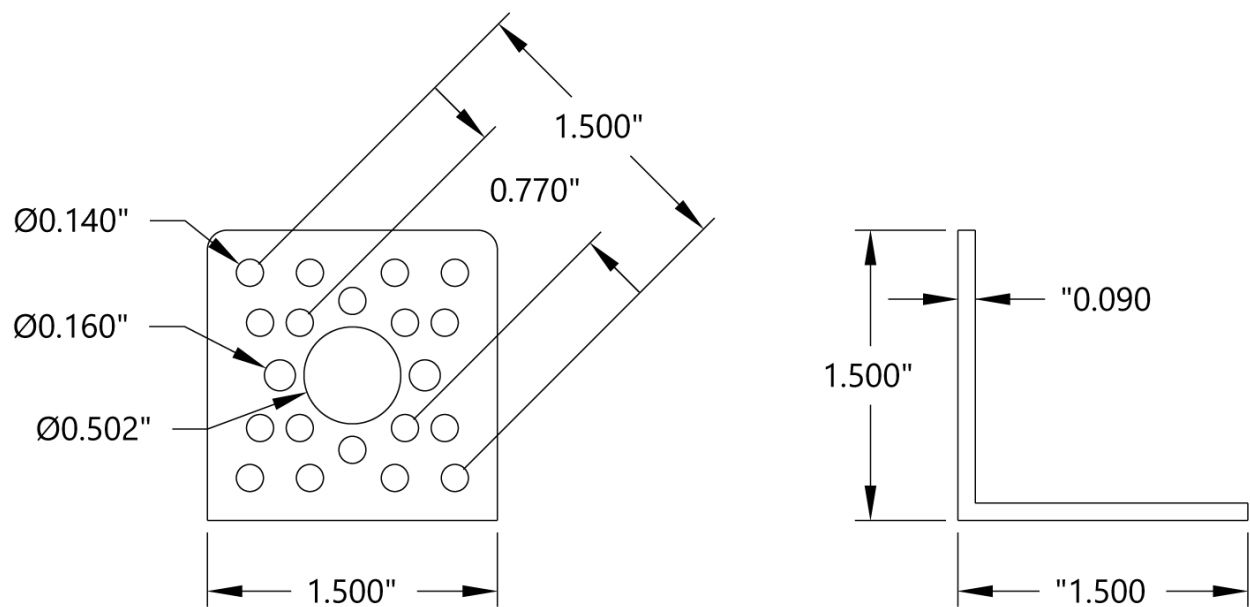
Material	Aluminum
Weight	1.86 oz (53g)
Finish	Clear Anodized
IP	Patented

2. Soporte en C



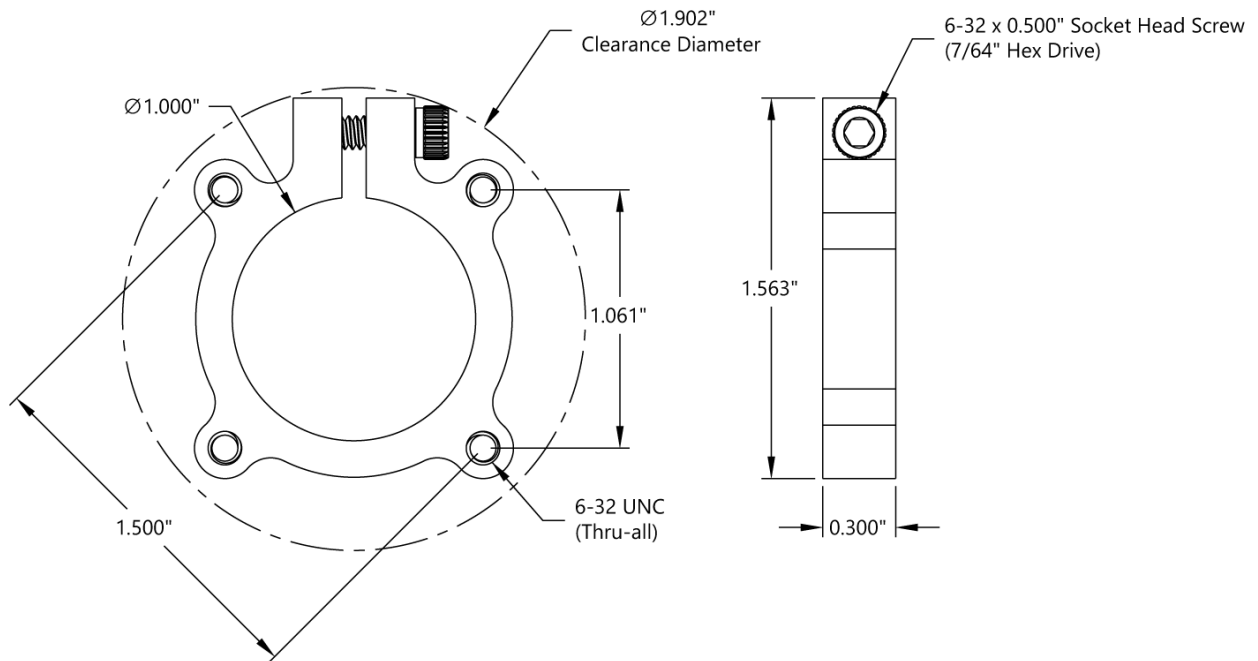
Material	Aluminum
Weight	0.69 oz (20g)
Finish	Clear Anodized
IP	Patented

3. Soporte a 90°



Length	1.50"
Material	Aluminum
Weight	0.47 oz (13g)
IP	Patented

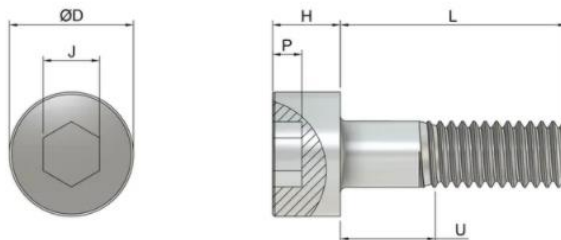
4. Fijación circular de 1"



Weight	0.41 oz (11.7g)
Material	Aluminum
Finish	Clear Anodized

5. Tornillos 6-32 x 0.4375" (7/16)

6-32 X 7/16 Inch UNC Cap Head Screws (ANSI B18.3) - Stainless Steel (A2) Technical Specification



Finish	Natural
Fully Threaded	Yes
General Tolerances	+/- 0.13mm
Head Diameter (D)	0.226 inch
Head Length (H)	0.138 inch
Hexagon A/F (J)	7/64 inch
Hexagon Depth (P)	0.064 inch
Length (L)	7/16 inch
Manufacturing Standard	ANSI B18.3
Material	Stainless Steel (A2)
Material Specification	Stainless Steel (A2)
Screw Thread Pitch	32 TPI
Thread Length (Max)	0.4375 inch
Thread Size (T)	6-32 UNC

ANEXO IV: ESPECIFICACIONES DE LA PLACA ARDUINO UNO

En este anexo se muestran las especificaciones de la placa Arduino Uno, ofrecidas por el vendedor. [12]

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

ANEXO V: MENÚ.PY

```
import Selection, Connection, Client, Choose_Satellite, Tracker
import keyboard, time

Connection.Connection()
cliente = Client.Client()
#Al poner simulation=True establece el modo simulacion por defecto
#Si lo dejo vacio establece por defecto simulacion=False
cliente.initialise()
connected=cliente.Connect()

if (not connected):
    print("Error")

else:

    salir = False
    opcion = 0
    while not salir:
        #Impresión de las opciones en pantalla
        print ("1. Seleccionar el satélite\n")
        print ("2. Seguimiento del satélite\n")
        print ("3. Posicionar estación manualmente\n")
        print ("4. Salir\n")
        print ("Elija una opcion:\n")
        opcion = Selection.pedirNumeroEntero()
        if opcion == 1:

            satellite = Choose_Satellite.Choose_Satellite()
            print("Estos son los datos TLE del satélite seleccionado: %s
\n" % satellite)
            tracker = Tracker.Tracker(satellite)

        elif opcion == 2:

            try:
                visible = tracker.Visible()

                if visible == False:
                    print("Satélite no visible")

                if visible==True:
                    tracker
                    print ("\nComienzo seguimiento\n")
                    while 1:
                        mensaje = tracker.Calculator()
                        cliente.Send_Data(mensaje)
                        time.sleep(1)
```

```
if keyboard.is_pressed('esc') or keyboard.is_pressed('q'):
    print("Ha pulsado la tecla para parar")
    break

except NameError:
    print("Debe definir antes un satellite")
    satellite = Choose_Satellite.Choose_Satellite()
    print("Estos son los datos TLE del satélite seleccionado:
%s \n" % satellite)
    tracker = Tracker.Tracker(satellite)

elif opcion == 3:
    print("\nPosicionar estación\n")
    cliente.Manual_Mode()
elif opcion == 4:
    salir = True
else:
    print ("Error, introduzca una opción válida")
print ("Fin")

cliente.Disconnect()
```


ANEXO VI: CONNECTION.PY

```
import requests, urllib.request
Url = 'https://celestrak.com/NORAD/elements/weather.txt'

def Connection():

    #Este programa comprueba si tenemos acceso a Internet o no.
    #Si tenemos, actualiza los datos, si no, no hace nada más que
    avisarnos que no tenemos
    try:
        print("Comprobando conexión a Internet...")
        requests.get('https://www.google.com/', timeout=5)

    except (requests.ConnectionError, requests.Timeout):
        print("Sin conexión a internet.")
        print("\nUsamos datos locales\n")

    else:
        print("Con conexión a internet.\n")
        print("Usamos datos en línea\n")
        print("Descargando...\n")
        TLE = urllib.request.urlopen(Url)
        Datos = TLE.read().decode('utf-8')
        #file = open("C:/Users/TFG/Desktop/Programas/TLE.txt", "w")
        file = open("D:\\Programas\\TLE.txt", "w")
        file.write(Datos)
        file.close()
        print("\nUsamos datos en línea\n")
```

ANEXO VII: CHOOSE_SATELLITE.PY

```
#Este programa lee los TLE de un archivo que haya en el ordenador.
#Si anteriormente, ha habido conexión a Internet el archivo estará
actualizado.

import json
from os.path import isfile
class Satelite:
    """ Crea objeto satelite """
    def __init__(self, nombre, linea1, linea2):
        self.nombre = nombre
        self.linea1 = linea1
        self.linea2 = linea2
def Choose_Satellite():
    #Proceso para leer datos locales de TLE.txt
    #Archivo = 'c:\\Users\\TFG\\Desktop\\Programas\\TLE.txt'
    Archivo = 'D:\\Programas\\TLE.txt'
    # Controla si el archivo existe
    if isfile(Archivo):
        with open(Archivo, 'r') as f:
            Datos = [linea for linea in f.readlines() if linea != "\n"]
    else:
        print("Archivo inexistente")
    #rstrip() elimina los espacios en blanco que hay al final de
    algunas líneas
    lines = [line.rstrip() for line in Datos]
    satelites=[]
    # Indices
    x, y, z = 0, 1, 2
    # Itera cada 3 elementos
    for i in range(int(len(lines)/3)):
        # Creo objeto y lo agrego a lista
        satelites.append(Satelite(lines[x], lines[y], lines[z]))
        # Incremento indices
        x += 3; y += 3; z += 3
    # Convierte objeto satelite en json
    datosJson = [json.dumps(s. dict ) for s in satelites]
```

```
# Analiza json
analizado = [json.loads(datosJson[i]) for i in
range(len(datosJson))]

resultados = 0
while resultados == 0:

    # Busca satellite por nombre
    query = input("Nombre satellite:\n>>> ")

    for sat in analizado:

        if sat['nombre'].lower() == query.lower():
            #print(sat)
            resultados +=1
            break

    if analizado.index(sat)+1==len(analizado) and resultados ==
0:
        print("No hay satellites con ese nombre.\nPruebe uno de
los siguientes:\n")
        nombres=[sat["nombre"] for sat in analizado]
        print(*nombres, sep="\n")

    return sat
```

ANEXO VIII: TRACKER.PY

```
import time, datetime, ephem, from math import *
#Pontevedra = ("42.4200", "-8.640", "0")
class Tracker():
    def __init__(self, satellite, estacion=("42.4200", "-8.640", "0"),
visible=True):
        self.estacion = ephem.Observer()
        self.estacion.lat = estacion[0]
        self.estacion.lon = estacion[1]
        self.estacion.elevation = int(estacion[2])
        self.satellite = ephem.readtle(satellite["nombre"],
satellite["linea1"], satellite["linea2"])
        self.visible= visible
    def set_epoch(self, epoch=time.time()):
        ''' sets epoch when parameters are observed '''
        self.estacion.date = datetime.datetime.utcnow().timestamp(epoch)
        self.satellite.compute(self.estacion)
    def azimuth(self):
        ''' returns satellite azimuth in degrees '''
        return degrees(self.satellite.az)
    def elevation(self):
        ''' returns satellite elevation in degrees '''
        return degrees(self.satellite.alt)
    def latitude(self):
        ''' returns satellite latitude in degrees '''
        return degrees(self.satellite.sublat)
    def longitude(self):
        ''' returns satellite longitude in degrees '''
        return degrees(self.satellite.sublong)
    def range(self):
        ''' returns satellite range in meters '''
        return self.satellite.range
    def aer2ecef(self, azimuthDeg, elevationDeg, slantRange, obs_lat,
obs_long, obs_alt):
        # site ecef in meters
        sitex, sitey, sitez = llh2ecef(obs_lat, obs_long, obs_alt)
```

```
# some needed calculations
senlat = sin(radians(obs_lat))
senlon = sin(radians(obs_long))
coslat = cos(radians(obs_lat))
coslon = cos(radians(obs_long))
azRad = radians(azimuthDeg)
elRad = radians(elevationDeg)
# az,el,range to sez conversion
SUR = -slantRange * cos(elRad) * cos(azRad)
ESTE = slantRange * cos(elRad) * sin(azRad)
CENIT = slantRange * sin(elRad)
x = (senlat * coslon * SUR) + (-senlon * ESTE) + \
    (coslat * coslon * CENIT) + sitex
y = (senlat * senlon * SUR) + (coslon * ESTE) + \
    (coslat * senlon * CENIT) + sitey
z = (-coslat * SUR) + (senlat * CENIT) + sitez
return x, y, z

def ecef_coordinates(self):
    ''' returns satellite earth centered cartesian coordinates'''
    x, y, z = self._aer2ecef(self.azimuth(), self.elevation(),
self.range(), float(
        self.estacion.lat), float(self.estacion.lon),
self.estacion.elevation)
    return x, y, z

def Visible(self):
    self.set_epoch(time.time())
    if self.elevation() < 0:
        self.visible = False
    else:
        self.visible = True
    return self.visible

def Calculator(self):
    self.set_epoch(time.time())
    AZ = str(round(self.azimuth(),2))
    if self.elevation() < 0:
        print("El satélite no está visible")
```

```
        self.visible = False
        return self.visible
    else:
        EL = str(round(self.elevation(),2))
        mensaje = ("P\t" + AZ + "\t" + EL + "\n\r")
        return mensaje
```

ANEXO IX: CLIENT.PY

```
import socket, time, select
import Selection
class Client():

    def __init__(self,simulation=False):

        self.host= '127.0.0.1'
        self.port= 4533
        self.s= 0
        self.simulation= simulation
    def initialise(self):

        salir = False
        opcion = 0

        while not salir:

            print("1. Seleccionar el puerto y host\n")
            print("2. Elegir puerto y host por defecto\n")
            print("3. Continuar en modo simulacion\n")
            print ("Eliga una opcion:\n")

            opcion = Selection.pedirNumeroEntero()

            if opcion == 1:
                self.host = input("Introduzca el host:\n>>>")
                self.port = int(input("Introduzca el puerto
serie:\n>>>"))

                salir = True

            elif opcion == 2:
```

```
        print("Usando valores por defecto...")
        print("Direccion IP: %s, puerto: %d"
%(self.host,self.port))
        salir = True

    elif opcion == 3:
        print("Entramos en modo simulacion")
        self.simulation = True
        return self.simulation

    else:
        print ("Error, introduzca una opción válida")

def Send_Data(self,message):

    if self.simulation:
        print(message)
        return

    FORMAT = 'ascii'
    try:

        # message sent to server
        self.s.send(message.encode(FORMAT))
        print(message)

        infds, outfds, errfds = select.select([self.s], [self.s],
[], 5)

        if len(infds) != 0:
            data = self.s.recv(1024)
            respuesta = str(data.decode(FORMAT))

            if len(data) != 0:
                if respuesta == 'RPRT 0\n':
                    print("Ok")
```



```
        else:
            print("Ha habido fallo en la conexión")
    else:
        print("Ha habido fallo en la conexión")

except BrokenPipeError():
    print("Ha habido un error en la conexion")

def Connect(self) :

    if self.simulation:

        print("Conectado modo simulacion")
        return True

    count = 3
    connected = False
    self.s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    while count:
        address = ((self.host, self.port))
        print("Se va a establecer la conexión...\n")

        if self.s.connect_ex(address):
            print("Servidor no activo ...")
            time.sleep(5)
            count -= 1

        else:
            print("Conexion establecida")
            connected = True
            break
    return connected

def Manual_Mode(self):
    FORMAT = 'ascii'
```

```

        cerrar = False
    while not cerrar:
        AZ = input("Introduzca el Azimut deseado:\n>>>")
        EL = input("Introduzca la Elevación deseada:\n>>>")
        message = ("P\t" + AZ + "\t" + EL + "\n\r")
        self.s.send(message.encode(FORMAT))
        print(message)
        infds, outfds, errfds = select.select([self.s], [self.s],
[], 5)

        if len(infds) != 0:
            data = self.s.recv(1024)
            respuesta = str(data.decode(FORMAT))
            if len(data) != 0:
                if respuesta == 'RPRT 0\n':
                    print("Ok")
                else:
                    print("Ha habido fallo en la conexión")
            else:
                print("Ha habido fallo en la conexión")
        query = input("¿Continuar?\nY/N")

        if query == ('Yes', 'yes', 'Y', 'y'):
            cerrar=cerrar
        else:
            cerrar=True

    def Disconnect(self):

        if (not self.simulation):
            print("Se va a cerrar la conexion.")
            self.s.close()

```

ANEXO X: CARGAS DE PRUEBAS PARA TORNILLOS. NORMATIVA ISO

En el presente anexo se muestran las cargas nominales para los tornillos de rosca gruesa según normativa ISO (898-1).

Rosca ^a <i>d</i>	Área de esfuerzo nominal ^b $A_{t, \text{nom}}$ mm ²	Clase de calidad									
		4.6	4.8	5.6	5.8	6.8	8.8	9.8	10.9	12.9/12.9	
Carga de prueba, F_p ($A_{t, \text{nom}} \times S_{p, \text{nom}}$), N											
M3	5,03	1 130	1 560	1 410	1 910	2 210	2 920	3 270	4 180	4 880	
M3,5	6,78	1 530	2 100	1 900	2 580	2 980	3 940	4 410	5 630	6 580	
M4	8,78	1 980	2 720	2 460	3 340	3 860	5 100	5 710	7 290	8 520	
M5	14,2	3 200	4 400	3 980	5 400	6 250	8 230	9 230	11 800	13 800	
M6	20,1	4 520	6 230	5 630	7 640	8 840	11 600	13 100	16 700	19 500	
M7	28,9	6 500	8 960	8 090	11 000	12 700	16 800	18 800	24 000	28 000	
M8	36,6	8 240 ^c	11 400	10 200 ^c	13 900	16 100	21 200 ^c	23 800	30 400 ^c	35 500	
M10	58	13 000 ^c	18 000	16 200 ^c	22 000	25 500	33 700 ^c	37 700	48 100 ^c	56 300	
M12	84,3	19 000	26 100	23 600	32 000	37 100	48 900 ^d	54 800	70 000	81 800	
M14	115	25 900	35 600	32 200	43 700	50 600	66 700 ^d	74 800	95 500	112 000	
M16	157	35 300	48 700	44 000	59 700	69 100	91 000 ^d	102 000	130 000	152 000	
M18	192	43 200	59 500	53 800	73 000	84 500	115 000	—	159 000	186 000	
M20	245	55 100	76 000	68 600	93 100	108 000	147 000	—	203 000	238 000	
M22	303	68 200	93 900	84 800	115 000	133 000	182 000	—	252 000	294 000	
M24	353	79 400	109 000	98 800	134 000	155 000	212 000	—	293 000	342 000	
M27	459	103 000	142 000	128 000	174 000	202 000	275 000	—	381 000	445 000	
M30	561	126 000	174 000	157 000	213 000	247 000	337 000	—	466 000	544 000	
M33	694	156 000	215 000	194 000	264 000	305 000	416 000	—	576 000	673 000	
M36	817	184 000	253 000	229 000	310 000	359 000	490 000	—	678 000	792 000	
M39	976	220 000	303 000	273 000	371 000	429 000	586 000	—	810 000	947 000	

^a Cuando en una designación de rosca no se indica el paso de la misma, se especifica el paso grueso.

^b Para calcular $A_{t, \text{nom}}$, véase el apartado 9.1.6.1.

^c Para elementos de fijación con tolerancia en la rosca ópt de acuerdo con la Norma ISO 965-4 sometidos a galvanización en caliente, se aplican valores más reducidos, de acuerdo con la norma ISO 10684:2004, véase el anexo A.

^d Para los elementos de fijación estructurales, 50 700 N (para M12), 68 800 N (para M14) y 94 500 N (para M16).

^a Cuando en una designación de rosca no se indica el paso de la misma, se especifica el paso grueso.

^b Para calcular $A_{t, \text{nom}}$, véase el apartado 9.1.6.1.

^c Para elementos de fijación con tolerancia en la rosca óz de acuerdo con la Norma ISO 965-4 sometidos a galvanización en caliente, se aplican valores más reducidos, de acuerdo con la norma ISO 10684-2004, véase el anexo A.

^d Para los elementos de fijación estructurales, 50 700 N (para M12), 68 800 N (para M14) y 94 500 N (para M16).