



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**  
**PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA**  
**INGENIERÍA ELÉCTRICA - TELECOMUNICACIONES**

**DISEÑO DE UNA RADIO DEFINIDA POR SOFTWARE CON POTENCIALES  
APLICACIONES EN UN PROYECTO SATELITAL CUBESAT MEXICANO**

**TESIS**

**QUE PARA OPTAR POR EL GRADO DE  
MAESTRO EN INGENIERÍA EN TELECOMUNICACIONES**

**PRESENTA:**

**ING. ANTONIO DE JESÚS SERRANO ARELLANO**

**TUTOR PRINCIPAL:**

**DR. JOSÉ ALBERTO RAMÍREZ AGUILAR**

**Facultad de Ingeniería**

**COMITÉ TUTOR:**

**DRA. FÁTIMA MOUMTADI - Facultad de Ingeniería**

**DR. VÍCTOR RANGEL LICEA - Facultad de Ingeniería**

**DR. SALVADOR LANDEROS AYALA - Facultad de Ingeniería**

**DR. RAMÓN GUTIÉRREZ CASTREJÓN – Instituto de Ingeniería**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



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

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

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

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

**JURADO ASIGNADO:**

Presidente: **DRA. FÁTIMA MOUMTADI**

Secretario: **DR. VÍCTOR RANGEL LICEA**

Vocal: **DR. JOSÉ ALBERTO RAMÍREZ AGUILAR**

1<sup>er</sup>. Suplente: **DR. SALVADOR LANDEROS AYALA**

2<sup>do</sup>. Suplente: **DR. RAMÓN GUTIÉRREZ CASTREJÓN**

Lugar donde se realizó la tesis:  
POSGRADO DE INGENIERÍA, CIUDAD UNIVERSITARIA

**TUTOR DE TESIS:**

**DR. JOSÉ ALBERTO RAMÍREZ AGUILAR**

-----  
**FIRMA**





*A la juventud espacial mexicana*



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



## AGRADECIMIENTOS

Ahora y siempre, a mis padres, Reyna y Jesús, por siempre ser, estar y amar. Este y cada logro es completamente suyo. GRACIAS!

A Felicia, por siempre creer y nunca soltarme.

A Anaclara, por todo tu amor, pasión, cariño, amistad y compartir. Por el presente.

A la Universidad Nacional Autónoma de México, por mostrarme los caminos e impulsar mis sueños.

Al Programa de Maestría y Doctorado en Ingeniería.

Al CONACYT, por brindarme la oportunidad de estudiar este posgrado gracias a la beca 847422/634585.

A mi asesor y tutor principal de este proyecto, el Dr. Alberto Ramírez Aguilar, quien me ha motivado, guiado y acompañado hacia la búsqueda de un mejor México a través de la tecnología espacial.

Al Departamento de Telecomunicaciones que, gracias al Dr. Víctor Rangel, me facilitó el dispositivo HackRF sin el cual no hubiese sido posible realizar este trabajo de investigación.

A mis sinodales, Dra. Fátima Moumtadi, Dr. Víctor Rangel, Dr. Salvador Landeros y Dr. Ramón Gutiérrez, por su lectura, crítica y comentarios.

A mis profesores, especialmente al Dr. José María Matías, por todo su apoyo y consejo desde el inicio de este ciclo y al Dr. Salvador Landeros, por compartir conmigo sus enseñanzas, experiencia y pasión por la tecnología satelital.

Al Mtro. en Ing. Amado Gutiérrez, por ayudarme a responder muchas de las preguntas e ideas aquí vertidas.

A mis grandes amigos, Marco Carbajal y Adrián Juárez, porque sin su apoyo y sin su brindar este trabajo no hubiese sido posible.

A Gibrán, quien me regaló su tiempo y su consejo cuando más dudas tenía.

A mis compañeros y amigos de la Maestría, con quienes compartí el reto.

Al Dr. Alejandro Farah y a mis amigos de Axolotl AAFI: Elena, Raziel, Josué, Omar, Edwin, Sergio, Marco y Adrián. A su lado cumplí uno de mis sueños. A su lado deseo cumplir muchos más. Ad Astra & See you in Texas.

A la Mamá Osa. Le agradezco cada instante con el corazón.

A mi familia Serrano Arias y Arellano Garay, quienes me han abrazado desde pequeño.

A mi tía Norma y mi tío Jorge, por compartirme su calidez y cariño cada vez. Los admiro tanto.

A mis amigos y confidentes. Las palabras sobran. Alva, Mayra, Milano, Quino, Edgar, Sergio. Les amo.

A mi estimado Maestro José Luis García, quien me otorgó el regalo de los CubeSat.



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



## ABSTRACT

In this thesis work, we study the main characteristics and advantages of the telecommunications systems implemented through Software Defined Radios. The DVB-S2 standard (Digital Video Broadcasting Standard Second Generation) is described and broken down into its different blocks and the use and convergence of both technologies in satellite projects are exposed, particularly in CubeSat platforms focused on terrestrial observation missions.

Subsequently, we present the requirements and objectives of a possible 3U CubeSat satellite mission that transmits information to the Ground Station (GS) located in the High Technology Unit of the Faculty of Engineering of the UNAM, Querétaro, Mexico. A communications subsystem is established as the main focus, which will consist of an SDR module implementing the DVB-S2 standard, capable of transmitting images to the GS with the different ModCod (Modulation & Coding) provided by the standard. Faced with this challenge, an orbital analysis is carried out using STK (System Tool Kit) simulating the orbit of the International Space Station (ISS) at an altitude of 406 km and a Sun Synchronous Orbit (SSO) at 600 km. With the simulation data, the Link Budget is generated using Matlab, considering losses due to free space, atmospheric absorption and rain, and the SNR (Signal to Noise Ratio) reached for each given range value is calculated. Using the data provided by the ETSI standard, the SNR values are established to switch from one ModCod to another and thus guarantee the transmission between the space and the terrestrial segment. Likewise, it is developed a theoretical estimate of the amount of information capable of being transmitted during one year in both orbits and with the established mission conditions.

Then, using GNU Radio in Ubuntu, simulations of the DVB-S2 transmission stage are run in CCM mode (Constant Coding & Modulation), managing to observe its frequency spectrum and constellations generated by QPSK, 8PSK, 16APSK and 32APSK modulations which are manually switched. This transmission block is adapted to the HackRF One transceiver which functions as a low-cost front-end, and allows to establish a test table of a functional SDR which generates a radio frequency (RF) signal that is propagated and received by a commercial satellite receiver of the GTMedia brand located in the same work area.

Once the implementation is done, a comparative analysis of the different transceivers, FPGAs, front ends and integrated SDRs currently available on the market is developed. This analysis allows us to establish the combination of the optimal devices for the design of the nanosatellite SDR transmitter, which seeks to optimize essential parameters such as mass, dimensions, power consumption and processing capacity. In this way, the main differences between solutions based on COTS devices and those already integrated and qualified to operate in space environments are found and an SDR transmitter design is provided.

Finally, the use of an FPGA (Field Programmable Gate Array) and a Front-End COTS is proposed as future work to make the transmission block reduced, portable and capable of being integrated into a 3U CubeSat.

**Keywords:** CubeSat, Software Defined Radio, Adaptive Modulation, DVB-S2, Configurable Radio, Front End, FPGA, Satellite, Space Technology, GNU, Earth Observation.



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



## RESUMEN

En el presente trabajo se estudian las características y ventajas principales de los sistemas de telecomunicaciones implementados a través de Radios Definidos por Software (SDR, Software Defined Radio). Se describe y desglosa el estándar DVB-S2 (Digital Video Broadcasting Standard Second Generation) en sus diferentes bloques y se expone la utilización y convergencia de ambas tecnologías en proyectos satelitales, particularmente en plataformas CubeSat enfocadas a misiones de observación terrestre.

Posteriormente, se plantean los requerimientos y objetivos de una posible misión satelital CubeSat de 3U que transmita la información a la estación terrena ubicada en la Unidad de Alta Tecnología de la Facultad de Ingeniería de la UNAM, campus Juriquilla en Querétaro, México. Se establece como enfoque principal un subsistema de comunicaciones el cual constará de un módulo SDR implementando el estándar DVB-S2, capaz de transmitir imágenes hacia tierra con los diferentes ModCod (Modulation & Coding) brindados por el estándar. Ante tal reto, se lleva a cabo un análisis orbital empleando el programa STK (System Tool Kit) simulando la órbita de la Estación Espacial Internacional (ISS, International Space Station) de 406 km de altura y otra heliosíncrona (SSO, Sun Synchronous Orbit) a 600 km. Con los datos de la simulación se genera el cálculo de enlace empleando Matlab (licencia institucional UNAM), considerando pérdidas por espacio libre, por absorción atmosférica y por lluvia, y se calcula el SNR (Signal to Noise Ratio) alcanzado para cada valor de rango dado. Utilizando los datos brindados por el estándar ETSI se establecen los valores de SNR para conmutar de un ModCod a otro y así garantizar la transmisión entre el segmento espacial y el terrestre. Así mismo, se hace un estimado teórico de la cantidad de información capaz de transmitirse durante un año en ambas órbitas y con las condiciones de misión establecidas.

Después, utilizando GNU Radio en Ubuntu, se corren simulaciones basadas en software libre de la etapa de transmisión de DVB-S2, en modo CCM (Constant Coding & Modulation), logrando observar su espectro en frecuencia y las constelaciones generadas por las modulaciones QPSK, 8PSK, 16APSK y 32APSK conmutados de manera manual. A este bloque de transmisión se le adapta el transceiver HackRF One el cual funciona como front-end de bajo costo, lo cual permite establecer una mesa de pruebas de un SDR funcional el cual genera una señal de radiofrecuencia (RF) la cual es propagada y recibida por un receptor satelital comercial de la marca GTMedia ubicado en la misma área de trabajo.

Una vez realizada la implementación, se desarrolla un análisis comparativo de los diferentes transceivers, FPGAs, front ends y SDR integrados disponibles en el mercado actualmente. De este análisis se consiguen establecer la combinación de dispositivos óptimos para el diseño del transmisor SDR nanosatelital, en el cual se busca optimizar parámetros esenciales tales como masa, dimensiones, consumo de potencia y capacidad de procesamiento. De esta forma se encuentran las principales diferencias entre soluciones basadas en dispositivos COTS y aquellas ya integradas y calificadas para operar en ambientes espaciales y se brinda un diseño de transmisor SDR.

Finalmente, se propone, como trabajo a futuro, el uso de un FPGA (Fiel Programmable Gate Array) y un Front-End tipo COTS (Commercial Off-The-Shelf) para lograr que el bloque de transmisión sea reducido, portable y capaz de ser integrado en un CubeSat de 3U.

**Palabras clave:** CubeSat, Radio Definida por Software, Modulaci3n Adaptativa, DVB-S2, Radio configurable, Front End, FPGA, Sat3elite, Tecnolog3a Espacial, GNU, Observaci3n Terrestre.



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



## I. ANTECEDENTES

La tecnología espacial ha crecido y evolucionado de manera constante desde hace más de 60 años, tomando como punto de partida la puesta en órbita del primer satélite artificial en 1957, el Sputnik 1. De manera particular, la tecnología satelital, se ha posicionado como una herramienta de gran utilidad en diversos campos de la ciencia, la economía y la sociedad. Desde análisis meteorológicos, prevención de catástrofes naturales, demostración de nuevas tecnologías y observación de la Tierra, hasta plataformas de telecomunicaciones para fines particulares y gubernamentales, los satélites son una pieza clave en el desarrollo de los países. Sin embargo, todas sus ventajas y alcances se habían visto un tanto limitadas por el costo de fabricación y lanzamiento que estos conllevan principalmente para países emergentes en el sector espacial como México y otros países de América Latina.

En 1999, la Universidad de Stanford junto con la Universidad de Santa Clara propusieron un estándar de nanosatélite que podía pesar hasta 1.33 kg, y cuyas dimensiones son 10 cm por lado, con la idea de hacer lanzamientos múltiples de estos albergándolos como carga secundaria en un cohete, con lo que se lograría un precio muy accesible para sus lanzamientos; dicho estándar es conocido como CubeSat [A\_1]. Estos proyectos, comparados con las tradicionales misiones satelitales multimillonarias, tienen el gran potencial de brindar una experiencia educativa a sus participantes y de implementar práctica y satisfactoriamente misiones relacionadas con la ciencia, la industria, y actualmente otras áreas como el arte, a un mucho menor costo. Los experimentos con CubeSats son usualmente nuevos y únicos y su periodo de realización va de los 9 a los 24 meses [A\_2] lo cual ha permitido que diferentes universidades, gobiernos y empresas alrededor del mundo los desarrollen y pongan en órbita, a tal punto que hoy en día encabezan una revolución satelital a nivel global.

Como en todos los sistemas satelitales, el subsistema de telecomunicaciones es una parte esencial de los proyectos CubeSat, permitiendo la transmisión de información y telemetría desde el espacio hacia Tierra y viceversa. El inicio de la tecnología CubeSat basó el desarrollo de sus sistemas de comunicaciones en el uso de radios convencionales, configurados para trabajar a determinada frecuencia de operación bajo ciertas características particulares asociadas a los elementos electrónicos y físicos que los componen. Con el avance de la tecnología se ha desarrollado una nueva alternativa llamada Radio Definida por Software (SDR; Software Defined Radio), los cuales combinan el uso de módulos basados en hardware, enfoque clásico, con módulos configurables y diseñados con base en dispositivos reprogramables capaces de trabajar a diferentes frecuencias de operación y con diferentes esquemas de modulación sin tener las limitantes físicas de sus antecesores. Este nuevo enfoque es una respuesta a la demanda de flexibilidad que la tecnología ha impuesto en torno a los sistemas de comunicaciones de manera general: uso de diferentes frecuencias, esquemas de modulación y codificación variables, aplicación de diversos protocolos, variación en las áreas de cobertura, etc. En el campo de los satélites CubeSat, la Radio Definida por Software ha ido tomando mayor impulso los últimos años mostrándose como una alternativa importante en el desarrollo de las comunicaciones nanosatelitales de la actualidad [A\_3], permitiendo el uso de múltiples bandas de frecuencia, técnicas de filtrado y esquemas de modulación sin requerir ningún cambio importante en su hardware, prácticamente todos estos ajustes se logran modificando el software del sistema. Adicional a esto, estas características pueden ser manipuladas en órbita a través de comandos enviados desde Tierra, lo cual será de enorme utilidad dentro de las grandes constelaciones nanosatelitales.

En otro orden de ideas, conforme pasa el tiempo, la tecnología avanza y el acceso a la información se acelera de manera exponencial. Las aplicaciones tecnológicas requieren de mayores capacidades para funcionar y la sociedad de la información



así lo demanda. En 1990, una red celular de 64 kbps era suficiente para cumplir con las demandas de los usuarios. Hoy en día la tecnología 5G aporta velocidades de transmisión arriba de 1 Gbps y se espera que esta tasa siga aumentando con las siguientes generaciones móviles [A\_4]. Lo mismo sucede en el terreno de la tecnología satelital, sobre todo en el campo de las comunicaciones. El primer estándar de televisión satelital, el DVB-S aportaba una tasa de datos de hasta 54 Mbps, los últimos avances del estándar, DVB-S2X nos permiten un flujo de información de hasta 118 Mbps [A\_5].

Lo mismo sucede en todos los niveles de la tecnología satelital. Las misiones y objetivos de los satélites LEO y MEO (Medium Earth Orbit) son cada vez más complejas y con requerimientos de velocidad de información más altos. Además, si bien los beneficios que los proyectos CubeSat aportan son numerosos, su tiempo de vida por lo general va de los 2 meses hasta los 2 años debido a que reingresan a la atmósfera y se desintegran, o a que alguno de sus subsistemas están diseñados con esta restricción. Con esta información se puede entender la importancia de tener enlaces de comunicaciones optimizados entre los proyectos CubeSat y las estaciones en Tierra. La necesidad de aumentar las velocidades de transmisión se hace evidente sobre todo en aquellos proyectos con aplicaciones relacionadas con la observación terrestre.

De este modo es que diferentes aplicaciones satelitales han optado por utilizar diferentes técnicas y esquemas de optimización de la transmisión de la información, una de ellas es la Codificación y Modulación Adaptativa (ACM, Adaptive Coding & Modulation) utilizada por primera vez en el estándar de Televisión Satelital DVB-S2 y actualmente utilizado en aplicaciones tales como LTE, WiMAX, WIFI, y las siguientes generaciones de televisión vía satélite. ACM brinda diferentes opciones de modulación así como tasas de codificación que pueden variar en tiempo real dependiendo del estado del enlace satelital. Toma como referencia el SNR o el  $E_s/N_0$  (Energy per Symbol to Noise Power Spectral Density Ratio) requerido para obtener un valor mínimo de BER (Bit Error Rate), el número de bits recibidos de forma incorrecta respecto al total de bits enviados durante un intervalo especificado de tiempo, en un enlace satelital establecido con Tierra. Mientras mejor sea el SNR mayor será el orden de la modulación y menor la tasa de codificación lo cual se traduce en mayor velocidad de transmisión de información, en caso contrario se utilizará una modulación y tasa de codificación más robusta para garantizar el valor mínimo de BER y con ello la transmisión de la información de manera correcta y sin errores [A\_6]. La técnica ACM es utilizada con diferentes esquemas de modulación: M-QAM, M-APSK, M-PSK, y con diferentes técnicas de codificación: Trellis, Reed Solomon, BCH (Bose–Chaudhuri–Hocquenghem), LDPC (Low Density Parity Check), todo esto dependiendo de la aplicación a implementar.

Es por esto que en este proyecto de investigación se propone el combinar estos tres factores de sumo impacto en el desarrollo de las comunicaciones satelitales: el estándar CubeSat, los sistemas SDR y el estándar DVB-S2, pues se espera que el conocimiento adquirido o los avances que se realicen en cualquiera de los tres rubros sean de gran importancia para el desarrollo de la tecnología satelital mexicana o para el desarrollo de los proyectos satelitales universitarios.



## II. PLANTEAMIENTO DEL PROBLEMA

México es uno de los países en desarrollo más importantes del mundo. Según el pronóstico para el 2020 de la OECD, se ubicará en el puesto 11 del ranking de las mayores economías del mundo y en el continente americano como la tercera economía, por debajo de Estados Unidos y Brasil [P\_1]. Su participación dentro del sector espacial comenzó en 1957, año en el cual se lanzó por primera vez en la historia del país un cohete de 1.7 metros con fines científicos. A esto le siguieron otros proyectos de coherencia impulsados por la Secretaría de Comunicaciones y Transporte en el año 1959, la instalación de una estación de rastreo en Sonora en 1960, colaborando con la NASA, y la creación de la Comisión Nacional del Espacio Exterior en 1962. Transmitió los juegos olímpicos de 1968 vía satélite. Ha utilizado la tecnología satelital para servicio doméstico desde 1970 adquiriendo 11 satélites a lo largo de su historia. Cuenta con tres centros de control satelital ubicados en la CDMX, Hidalgo y Sonora. Su Agencia Espacial, la cual cuenta con un presupuesto de 125 millones de dólares anuales, inició labores en el 2010 y fue país sede del Congreso Nacional de Astronáutica de 2017 [P\_2]. Además cuenta con diversas instituciones académicas, observatorios astronómicos, instituciones y laboratorios vinculados con el estudio del espacio.

Sin embargo, sus contribuciones al desarrollo de la tecnología satelital son escasas. El primer satélite integrado por estudiantes de ingeniería mexicanos y puesto en órbita fue el UNAMSAT-B en 1996, un paquete de ingeniería avanzada de Amsat-Na Microsat que la UNAM consigue al firmar un convenio con la Organización de Radioaficionados (AMSAT) [P\_3]. Si bien el diseño del satélite no es nacional, el desarrollo del proyecto, su adaptación al objetivo de la misión, su integración y su administración fueron llevadas a cabo por la UNAM. A lo largo de los años, han nacido diversas iniciativas satelitales en nuestro país de las cuales la única puesta en órbita, en junio del 2019, es el proyecto CubeSat Painani-I perteneciente a la SEDENA y desarrollado por el CICESE. Otro proyecto CubeSat con altas probabilidades de ser puesto en órbita en los próximos meses es el AzTechSat-1, desarrollado por la Universidad Popular Autónoma del Estado de Puebla (UPAEP).

Actualmente, y desde un inicio, los sistemas satelitales que posee el gobierno mexicano han sido adquiridos a empresas extranjeras generando una dependencia tecnológica de las mismas. México ha invertido cientos de millones de dólares en los últimos 30 años en la compra de estos sistemas. Once satélites como se mencionó anteriormente. Del mismo modo, México gasta 4 mil millones de dólares anualmente en la adquisición de imágenes y otros servicios satelitales [P\_4].

Uno de los grandes problemas del desarrollo satelital de nuestro país es la no culminación de proyectos espaciales, la falta de generación de recursos económicos e intelectuales dentro del mismo y la dependencia tecnológica que trae como consecuencia para México. En este contexto, la investigación y el desarrollo de proyectos universitarios enfocados a las tecnologías espaciales y satelitales, como el que se plantea en el presente proyecto, son una parte importante de la solución, sumándose a otros esfuerzos dentro de la triple hélice: industria, academia y gobierno, en pos de revertir esta tendencia.

Es de esta manera que se plantea la siguiente pregunta como móvil principal para este proyecto de investigación enfocado al campo de las telecomunicaciones:

¿De qué modo se pueden brindar mayores beneficios al avance satelital mexicano a través de un proyecto universitario relacionado con las telecomunicaciones satelitales?



De esta forma, se ha tomado la decisión de enfocar el presente trabajo de investigación en el empleo de tres factores tecnológicos que están impactando el mundo de los proyectos satelitales en los últimos años: el estándar CubeSat, las plataformas de Radio Definida por Software y los esquemas de Codificación y Modulación Adaptativos (ACM).

Por un lado, los proyectos CubeSat, nanosatélites cuya masa varía entre 1 y 5 kg, son uno de los facilitadores tecnológicos más importantes y accesibles de la actualidad para países con economías en desarrollo y principalmente para las universidades. Sus costos y tiempo de desarrollo son relativamente pequeños y su proceso de elaboración y de regulación es similar al de los satélites de mayor tamaño. Así mismo, los beneficios que se generan gracias a su desarrollo son vastos, con la capacidad de repercutir directamente a nivel institución e incluso a nivel nacional. Además, se muestran como uno de los principales atractivos para los gobiernos y las empresas en el ámbito espacial internacional de la actualidad y del futuro.

Por otro lado, las plataformas SDR se muestran como una de las innovaciones tecnológicas más importantes dentro del sector de las comunicaciones. Su gran flexibilidad representa una gran ventaja y de manera paulatina son más las tecnologías que hacen uso de estas, tanto en tierra como en el espacio. Además, presentan un gran potencial a futuro: la posibilidad de rediseñar y adaptar a conveniencia y en tiempo real todo el sistema de comunicaciones. Esto es particularmente importante en el caso de las diversas constelaciones CubeSat que se planean desplegar en órbita LEO en los próximos años, ya que la posibilidad de transitar de una banda de frecuencias a otra será vital para el correcto funcionamiento de los satélites y para disminuir la probabilidad de interferencia que este despliegue conlleva.

Así mismo, los esquemas adaptativos son clave en el desarrollo de los sistemas de alto rendimiento. Estos utilizan diferentes modulaciones y codificaciones de manera conmutada permitiendo optimizar la velocidad de transmisión de los sistemas aumentando la capacidad de los mismos de manera considerable. La necesidad de transmitir información con mayores velocidades es una constante en aplicaciones terrestres y en órbita por lo cual los sistemas ACM están siendo cada vez más utilizados.

Es por eso que en el presente trabajo se propone generar un proyecto en el que converjan estos tres factores, el cual permitirá desarrollar un módulo de pruebas basado en el estándar DVB-S2 e implementado en tecnología SDR. Una vez desarrollado este módulo propondremos el diseño de un sistema de comunicaciones completo y funcional con potencial aplicación en proyectos satelitales mexicanos tipo CubeSat. La miniaturización e integración final del sistema de comunicaciones se establece como labor posterior al desarrollo del trabajo de investigación.



### III. JUSTIFICACIÓN

La propuesta de este trabajo de investigación se relaciona de diferentes maneras con los objetivos y prioridades establecidas por diversas autoridades y representantes del sector espacial mexicano. En primer lugar, la Agencia Espacial Mexicana, principal ente gubernamental relacionado con el desarrollo tecnológico espacial del país, establece en su Plan de Orbita 2.0 publicado en 2017 el proyecto de “Desarrollar una estrategia de diseño, manufactura y puesta en servicio de satélites pequeños” como parte de sus objetivos a corto plazo [J\_1]. Este proyecto se enfoca al desarrollo de satélites CubeSat de 3U. Respecto a las instituciones académicas del país, son diferentes y numerosos los proyectos que actualmente se encuentran enfocados al desarrollo de satélites pequeños (menos de 100 kg) particularmente del tipo CubeSat. En la siguiente tabla se muestra una relación de estas instituciones con sus respectivos proyectos:

**Tabla J\_1. Proyectos satelitales mexicanos en la actualidad.**

Institución	Departamento	Proyecto	Tipo	Masa	Aplicación	Órbita	Fecha tentativa de culminación
AEM y Universidad del Ejército y Fuerza Aérea	CICESE	Painani-I [J_2]	CubeSat 3U	4 kg	Observación de la Tierra	LEO	En órbita (Junio 2019)
Universidad Popular Autónoma del Estado de Puebla y NASA	Facultad de Electrónica	AzTechSat -1 [J_3]	CubeSat 1U	1 kg	Intercomunicación Satelital	LEO (450 km)	2019
UNAM	Instituto de Ciencias Nucleares – LINX	Nano-Connect [J_4]	CubeSat 1U	1 kg	Intercomunicación Satelital	LEO (450 km)	2020
UNAM	Instituto de Ciencias Nucleares – LINX	Nano-SiPM [J_4]	CubeSat 3U	4 kg	Experimentos con Física de Partículas	LEO	2020
UNAM y Colectivo Espacial Mexicano	Unidad de Alta Tecnología	KuautliSAT el Ulises 2.0 [J_5]	TubeSat	1 kg	Observación Terrestre	LEO	2020
Universidad Panamericana y MIT	---	Pakal (Misión Colibrí)	CubeSat 3U	3kg	Análisis Densidad Atmosférica	LEO	2021
UNAM e Instituto de Aviación de Moscú	Unidad de Alta Tecnología	Cóndor [J_6]	Microsatélite	<100 kg	Precursores sísmicos	LEO	Sin definir
UNAM y MIT	Unidad de Alta Tecnología	Quetzal [J_7]	Microsatélite	<100 kg	Análisis de Contaminantes Atmosféricos	LEO (800 km)	Sin definir
INAOE y Colectivo Espacial Mexicano	---	Ulises 1 [J_8]	TubeSat	1kg	Experimental (artístico)	LEO (450 km)	Sin definir
AEM y Universidad del Ejército y Fuerza Aérea	---	Painani-II [J_9]	CubeSat 3U	4 kg	Sin información	Sin información	Sin información
UNAM	Instituto de Geografía	Sin definir	CubeSat 3U	4 kg	Pruebas de Estabilización	LEO	Sin definir
UNAM	Programa Espacial Universitario	Sin Definir	CubeSat 3U	4 kg	Observación Terrestre	LEO	Aún no iniciado

Además, dentro de la UNAM se estableció la creación del Programa Espacial Universitario (PEU) a finales del 2017, “el cual funge como una entidad encargada de coordinar capacidades y esfuerzos de la UNAM para el desarrollo de tecnología espacial y la formación multidisciplinaria de profesionales de las ciencias y humanidades en el área espacial, con



el fin de fomentar la creación de proyectos y la realización de estudios que impulsen el crecimiento de la ciencia, la tecnología y la industria espacial en México, así como de la difusión del conocimiento espacial [J\_10]”.

Los resultados de la investigación están proyectados para su utilización en los proyectos espaciales que se llevan a cabo en la Unidad de Alta Tecnología de la UNAM, potencialmente en un proyecto CubeSat desarrollado en colaboración con otras entidades. En segundo término, cualquier institución perteneciente a la UNAM relacionada con el desarrollo satelital podrá verse beneficiada con estos resultados. Finalmente, se busca que cualquier institución dentro del país pueda hacer uso de los resultados obtenidos, particularmente aquellas que estén desarrollando sistemas de comunicaciones con base en SDRs o esquemas ACM, incluyendo proyectos relacionados con las telecomunicaciones terrestres.



## IV. OBJETIVOS

### OBJETIVOS GENERALES

Realizar un trabajo de investigación en torno a las Radios Definidas por Software, el estándar DVB-S2 y a las plataformas satelitales CubeSat, todos estos impulsores tecnológicos dentro del campo de las comunicaciones satelitales.

Plantear los requerimientos y características principales del módulo transmisor de una misión satelital tipo CubeSat cuya principal aplicación sea la observación terrestre y que considere los aspectos tecnológicos mencionados. Con base en este planteamiento, desarrollar el análisis orbital y de comunicaciones del satélite para caracterizar el enlace de comunicaciones entre este y el segmento terrestre.

Establecer el diseño de un sistema transmisor SDR que considere el estándar de comunicaciones DVB-S2 y que pueda ser potencialmente implementado en una misión satelital mexicana tipo CubeSat con aplicaciones de observación terrestre.

### OBJETIVOS ESPECÍFICOS

- Estudiar el funcionamiento de las plataformas basadas en Radios Definidos por Software, sus ventajas y las técnicas para desarrollarlos o implementarlos.
- Conocer el funcionamiento, desempeño y beneficios del uso del estándar DVB-S2.
- Conocer las características principales de los nanosatélites bajo el estándar CubeSat.
- Investigar los proyectos CubeSats puestos en órbita con sistemas de comunicaciones basados en SDR.
- Analizar el desempeño de un módulo ACM en órbita mediante el uso de enlaces satelitales simulados.
- Simular y estudiar las órbitas de la misión planteada y caracterizar el sistema de comunicaciones con la información generada.
- Desarrollar el cálculo de enlace entre los segmentos espacial y terrestre.
- Simular el estándar DVB-S2 en GNU Radio.
- Implementar un SDR al adaptar un front end comercial a la plataforma de software para lograr la transmisión de información entre este y un receptor satelital tipo COTS.
- Establecer el diseño de un sistema de transmisión CubeSat, completo y funcional, considerando el estándar DVB-S2 y la implementación de un SDR.
- Proponer el uso de un FPGA o un SDR comercial para implementar el sistema de transmisión satelital considerando las dimensiones, precio y consumo de potencia adecuados para el proyecto CubeSat planteado.



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



## V. CONTRIBUCIÓN

Con el desarrollo del trabajo de investigación, se generará información sobre:

1. Importancia y beneficios del desarrollo de proyectos satelitales bajo el estándar CubeSat.
2. Requerimientos específicos del subsistema de comunicaciones de un proyecto CubeSat.
3. Características principales, aplicaciones y ventajas de las plataformas basadas en SDR.
4. Ventajas del empleo de dispositivos FPGAs dentro de un subsistema de comunicaciones.
5. Características de los enlaces satelitales entre un satélite en órbita LEO y la estación de control satelital “ECXSAT” ubicada en la Unidad de Alta Tecnología (UAT), Juriquilla.
6. Utilización de front ends en conjunto con herramientas de procesamiento de señal tales como GNU Radio.
7. Ventajas y desventajas de la utilización de COTS y sistemas con cualificación espacial para un proyecto satelital.

Con esta información y conocimiento adquiridos se busca:

1. Contar con los elementos teóricos necesarios para estructurar una propuesta de desarrollo de sistemas de comunicaciones que brinde diversos beneficios y que facilite el desarrollo de la tecnología satelital en las universidades mexicanas.
2. Reiterar la importancia del uso de las tecnologías ACM y SDR en proyectos satelitales de órbita baja impulsando el uso y aplicación de las mismas dentro de los proyectos satelitales que se llevan a cabo en la UAT de la Facultad de Ingeniería de la UNAM.
3. Dejar las bases para generar en el corto o mediano plazo subsistemas de comunicaciones de a bordo para nanosatélites tipo Tube/CubeSat completamente integrados y funcionales, listos para iniciar la etapa de pruebas de precertificación espacial para poder validar y/o garantizar su correcta funcionalidad en órbita.
4. Generar colaboraciones con otras instituciones y grupos de trabajo, dentro y fuera de la UNAM, que se enfoquen en el desarrollo de sistemas de comunicaciones SDR.



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



## VI. METODOLOGÍA

La metodología se realizó bajo el siguiente orden de ideas:

- Estudio del estado del Arte y Marco Teórico (SDR, DVB-S2 y proyectos CubeSat)
- Simulación orbital en STK
- Cálculos de enlace de comunicaciones
- Simulación del estándar DVB-S2 en GNU Radio
- Implementación de un SDR transmisor empleando GNU Radio y Front End tipo COTS
- Transmisión y recepción de video
- Estudio de FPGAs, Front Ends y SDRs: Características y selección
- Comparativa COTS vs Dispositivos con cualificación espacial
- Análisis de resultados
- Propuesta de diseño de un transmisor SDR nanosatelital

A continuación se describe la metodología empleada en cada una de las etapas del proyecto.

### Marco teórico (Capítulo 1)

La primera etapa de desarrollo es la etapa de investigación o marco teórico, en la cual el enfoque es la tecnología CubeSat, las plataformas SDR, y el estándar DVB-S2.

Primeramente, se considera el Estado del Arte en el cual se mencionan los diferentes proyectos y trabajos de investigación, tanto nacionales como extranjeros, enfocados en las SDR nanosatelitales y en el estándar DVB-S2.

Posteriormente, se desarrolla el marco teórico, el cual se basa en la descripción de las características principales de los sistemas de comunicaciones basados en SDRs, sus ventajas y desventajas así como la descripción bloque por bloque del estándar DVB-S2. De igual modo, se propone la revisión de los proyectos CubeSat puestos en órbita hasta el momento cuyo sistema de comunicaciones se fundamente en el uso de SDRs. Adicional a esto, se hace una revisión de algunos proyectos nanosatelitales de observación terrestre que cuentan con módulos DVB-S2 y se obtienen conocimientos sobre los requerimientos de telecomunicaciones y características de estos módulos, particularmente la demanda de la cantidad de datos asociados a la descarga de imágenes satelitales.

### Simulación orbital en STK (Capítulo 2)

Con base en el marco teórico, se plantean las características y requerimientos de una hipotética misión satelital CubeSat. Con esta información se realiza un análisis orbital gracias al cual se obtiene información sobre la dinámica existente entre un satélite en órbita LEO y la Estación Terrena (ET) ubicada en la UAT, Juriquilla. Para realizar esta simulación se emplea la herramienta System Tool Kit (STK) del proveedor AGI el cual es un simulador de dinámicas orbitales que nos permitirá obtener, entre otros datos:

- Periodo orbital
- Número de avistamientos por día (satélite – Estación Terrena)
- Duración de cada avistamiento
- Rango en km entre el satélite y la Estación Terrena.

Las simulaciones se realizan para una órbita ISS (400 km, 51.64°) y una órbita heliosíncrona (600km, -98°), y considerando que el nanosatélite tipo CubeSat manejará una potencia de transmisión de 0.5 y 1 W. El tiempo de simulación se establece de enero a diciembre de 2020 con una frecuencia de muestreo de 1 segundo.



### **Cálculo de enlace de comunicaciones (Capítulo 3)**

Se considera la metodología de Cálculo de Enlace dada por Neri Vela y Landeros Ayala. Factores como atenuación en el espacio libre, atenuación por lluvia, absorción atmosférica y conectores son utilizados. El objetivo es obtener el valor de la SNR respecto al tiempo con base en los valores de rango obtenidos con STK. Una vez obtenido el valor de SNR se asocia a los diferentes umbrales de conmutación para DVB-S2 y se obtiene el valor de tasa de datos teóricos para el enlace de bajada, con lo cual es posible establecer si la misión satelital planteada es viable para una aplicación de observación terrestre en términos de la capacidad del canal de comunicaciones.

### **Simulación e implementación del estándar DVB-S2 en GNU Radio (Capítulo 4)**

Se simula e implementa un SDR transmisor bajo el estándar DVB-S2. Para ello se emplean las librerías precargadas en GNU Radio para el procesamiento de la señal en DVB-S2 (en un CPU con sistema operativo Linux) y un front end tipo COTS de la marca HackRF para el acondicionamiento de la señal de RF. Como etapa receptora se hace uso de un receptor satelital tipo COTS. Una vez implementado este transmisor, visto como una mesa de pruebas, se procede a caracterizarlo y analizarlo con apoyo de un analizador de espectros. De este modo se hace una comparativa teórica vs experimental del desempeño del estándar y de la capacidad del canal de comunicaciones.

### **Estudio de FPGAs, Front Ends y SDRs: Características y selección (Capítulo 5)**

En esta etapa se describen las características, ventajas y desventajas principales de dispositivos FPGAs, sistemas Front End y SDR integrados. Se realiza una comparativa con las diferentes marcas en el mercado y se seleccionan aquellos componentes que cumplan con los requerimientos de la misión satelital planteada. Los puntos importantes a considerar son tecnología de desarrollo, capacidad de procesamiento, potencia consumida, dimensiones, masa, precio, tolerancia a la radiación y complejidad de integración. Finalmente, con base en los resultados obtenidos se proponen los elementos clave para el diseño de un transmisor SDR con la capacidad de implementar el estándar DVB-S2 en un proyecto CubeSat.



# ÍNDICE

## DISEÑO DE UNA RADIO DEFINIDA POR SOFTWARE CON POTENCIALES APLICACIONES EN UN PROYECTO SATELITAL CUBESAT MEXICANO

ABSTRACT	11
RESUMEN	13
I. ANTECEDENTES	15
II. PLANTEAMIENTO DEL PROBLEMA	17
III. JUSTIFICACIÓN	19
IV. OBJETIVOS	21
V. CONTRIBUCIÓN	23
VI. METODOLOGÍA	25
ACRÓNIMOS	31
LISTA DE FIGURAS	33
LISTA DE TABLAS	37
<b>CAPÍTULO 1. MARCO TEÓRICO</b>	<b>41</b>
<b>1.1 ESTADO DEL ARTE</b>	<b>41</b>
<b>1.2 RADIO DEFINIDA POR SOFTWARE</b>	<b>43</b>
1.2.1. Diagrama de bloques de un SDR	43
1.2.2. Dispositivos programables	46
1.2.3. Beneficios de los SDR	49
1.2.4. Principales aplicaciones	50
1.2.5. Ambientes de desarrollo para SDR	51
<b>1.3 ESTÁNDAR DVB-S2</b>	<b>53</b>
1.3.1. Características principales de DVB-S2	55
1.3.2. Diagrama de Bloques DVB-S2	57
1.3.2.1. <i>Mode adaptation</i>	58
1.3.2.2. <i>Stream Adaptation</i>	64
1.3.2.3. <i>FEC Encoding</i>	66
1.3.2.4. <i>Mapping</i>	68
1.3.2.5. <i>Physical Layer Framing</i>	71
1.3.2.6. <i>Modulación</i>	76
<b>1.4. SATÉLITES CUBESAT</b>	<b>77</b>
1.4.1. Proyectos CubeSat desarrollados	78
1.4.2. Proyectos CubeSat con SDR	82
1.4.3. Proyectos CubeSat con DVB-S2	83
1.4.3.1. <i>Diseño del transmisor</i>	84
1.4.3.2. <i>Diseño del receptor</i>	85
1.4.3.3. <i>Carga útil</i>	85
<b>CAPÍTULO 2. ANÁLISIS ORBITAL</b>	<b>89</b>
<b>2.1 GENERALIDADES DE LA MISIÓN A ANALIZAR</b>	<b>89</b>
<b>2.2 SIMULACIÓN ORBITAL CON STK</b>	<b>92</b>
2.2.1. Apertura STK	92
2.2.2. Creación del escenario	93
2.2.3. Generación de reportes	102



<b>2.3 ANÁLISIS DE RESULTADOS</b>	108
2.3.1. Gráficas de accesos	109
2.3.2. Gráficas de rangos	111
<b>CAPÍTULO 3. ANÁLISIS DE COMUNICACIONES</b>	115
<b>3.1 DIFERENCIAS ENTRE BER, Eb/N0, Es/N0 y SNR</b>	115
3.1.1. Análisis BER vs Eb/No	118
3.1.2. DVB-S2 en Simulink	119
<b>3.2 VALORES DE CONMUTACIÓN PARA ACM</b>	122
<b>3.3 CÁLCULO DE ENLACE</b>	125
<b>3.4 ANÁLISIS DE RESULTADOS</b>	132
3.4.1. Tasa de datos	136
3.4.2. Estimación de la capacidad del canal DVB-S2	138
<b>CAPÍTULO 4. IMPLEMENTACIÓN DEL ESTÁNDAR DVB-S2 EN UN SDR</b>	145
<b>4.1 SIMULACIÓN DE DVB-S2 EN GNU RADIO</b>	146
<b>4.2 INTEGRACIÓN DE LA MESA DE PRUEBAS</b>	156
4.2.1. Front End HackRf One	157
4.2.1.1. Diagrama de bloques HackRF One	158
4.2.1.2. Configuración HackRF One	160
4.2.2. Configuración de GNU Radio y HackRF One	164
4.2.2.1. Prueba de recepción en FM	167
4.2.3. Receptor Satelital	170
4.2.4. Antena receptora y transmisora	171
<b>4.3. PRUEBAS DE TRANSMISIÓN Y RECEPCIÓN</b>	173
<b>4.4. DESEMPEÑO DEL SDR IMPLEMENTADO</b>	180
4.4.1. Configuración final establecida	180
4.4.2. Estimación del SNR experimental	181
4.4.3. Análisis del espectro generado	187
<b>CAPÍTULO 5. DISEÑO DE UN SISTEMA DE COMUNICACIONES SDR UNIFICADO</b>	195
<b>5.1. POTENCIAL IMPLEMENTACIÓN EN FPGA</b>	196
5.1.1. Lenguajes de programación en los FPGA	197
5.1.2. Comparativa entre dispositivos FPGA	199
5.1.3. Entornos de diseño y simulación para FPGA	202
5.1.3.1. Entornos de simulación	202
5.1.3.2. Entornos de implementación	204
<b>5.2. FRONT ENDS COMERCIALES</b>	206
<b>5.3 SDR INTEGRADOS</b>	215
<b>5.4. DISEÑO DEL SDR PROPUESTO</b>	225
5.4.1. Diagrama del SDR propuesto	239
<b>ANÁLISIS FINAL DE RESULTADOS</b>	245
<b>CONCLUSIONES</b>	249
<b>TRABAJO A FUTURO</b>	252
<b>REFERENCIAS</b>	255
<b>ANEXO A. MATLAB</b>	263
A1. Análisis BER vs Eb/no	263
A2. Cálculo de Enlace	264



**Programa de Maestría y Doctorado en Ingeniería**

A3. Conmutación y tasa de datos ACM	.....	266
A4. Tasa de datos CCM	.....	266
A5. Velocidad de transmisión DVB-S2 vs CCM	.....	267
<b>ANEXO B. GNU RADIO</b>	.....	269
<b>ANEXO C. HACKRF</b>	.....	271



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



# ACRÓNIMOS

<b>1U</b>	1 unidad	<b>LDPC</b>	Low Density Parity Check
<b>3U</b>	3 unidades	<b>LEO</b>	Low Earth Orbit
<b>ACM</b>	Adaptive Coding & Modulation	<b>LNA</b>	Low Noise Amplifier
<b>ADC</b>	Analog- Digital Converter	<b>LNB</b>	Low Noise Block
<b>ADS-B</b>	Automatic Dependent Surveillance Broadcast	<b>LTE</b>	Long Term Evolution
<b>AER</b>	Azimuth-Elevation-Range	<b>LSB</b>	Less Significant Bit
<b>AIS</b>	Automatic Identification System	<b>LUT</b>	Look Up Table
<b>AMSAT</b>	Amateur Satellite	<b>M2M</b>	Machine-to-machine
<b>ASIC</b>	Application Specific Integrated Circuit	<b>mA</b>	Miliampers
<b>APSK</b>	Amplitude-Phase Shift Keying	<b>MB</b>	Megabyte
<b>AWGN</b>	Additive White Gaussian Noise	<b>Mbps</b>	Megabits per second
<b>BB</b>	Banda Base	<b>MEO</b>	Medium Earth Orbit
<b>BBFRAME</b>	Base Band Frame	<b>ModCod</b>	Modulation & Coding
<b>BCH</b>	Bose-Chaudhuri-Hocquenghem	<b>MPEG-2</b>	Moving Pictures Experts Group-2
<b>BER</b>	Bit Error Rate	<b>MSB</b>	Most significant Bit
<b>BW</b>	Bandwidth	<b>Msp</b>	Mega símbolos por segundo
<b>CCD</b>	Charge-Coupled-Device	<b>NASA</b>	National Aeronautics and Space Agency
<b>CCM</b>	Constant Coding & Modulation	<b>OECD</b>	Organization for economic co-operation and development
<b>CCIR</b>	Comité Consultivo Internacional de Radiocomunicaciones	<b>OSMoCom</b>	Open Source Mobile Communications
<b>CNES</b>	Centro Nacional de Estudios Espaciales	<b>PER</b>	Packet Error Rate
<b>COTS</b>	Commercial off-the-shelf	<b>PIRE</b>	Potencia Isotrópica Radiada Efectiva
<b>CPLD</b>	Complex Programmable Logic Device	<b>PRBS</b>	Pseudo Random Binary Sequence
<b>CR<sub>BCH</sub></b>	Code Rate de BCH	<b>PSK</b>	Phase Shift Keying
<b>CR<sub>LDPC</sub></b>	Code Rate de LDPC	<b>Ptx</b>	Potencia de transmisión
<b>DAC</b>	Digital-Analog Converter	<b>QAM</b>	Quadrature Amplitude Modulation
<b>dB</b>	Decibel	<b>Rbps</b>	Tasa de Datos
<b>DDC</b>	Digital Down Converter	<b>RF</b>	Radio Frequency
<b>DFL</b>	Data Field Length	<b>RFNoc</b>	RF Network on Chip
<b>DLR</b>	Centro Aeroespacial Alemán	<b>RP-SMA</b>	Reverse Polarity – SMA
<b>DSP</b>	Digital Signal Processor	<b>RTL</b>	Register Transfer Language
<b>DUC</b>	Digital Up Converter	<b>RX</b>	Receptor/Recepción
<b>DVB-S2</b>	Digital Video Broadcasting Standard 2	<b>SDR</b>	Software Defined Radio
<b>Eb/No</b>	Energy per bit to Spectral Noise Density Ratio	<b>SMA</b>	SubMiniature version A
<b>Es/No</b>	Energy per symbol to Spectral Noise Density Ratio	<b>SNR</b>	Signal to Noise Ratio
<b>ESA</b>	European Space Agency	<b>SoC</b>	System on Chip
<b>ET</b>	Estación Terrena	<b>SPMD</b>	Single Programm Multiple Data
<b>ETSI</b>	European Telecommunications Standard Institute	<b>SSO</b>	Sun Synchronous Orbit
<b>FEC</b>	Forward Error Correction	<b>STK</b>	System Tool Kit
<b>FIR</b>	Finite Impulse Response	<b>TX</b>	Transmisor/Transmisión
<b>FPGA</b>	Field Programmable Gate Array	<b>TRX</b>	Transceiver
<b>GSM</b>	Global System for Mobile Communications	<b>UHF</b>	Ultra High Frequency
<b>GB</b>	Gigabyte	<b>UHD</b>	USRP Hardware Driver
<b>Gbps</b>	Gigabits per second	<b>UIT</b>	Unión Internacional de Telecomunicaciones
<b>GEO</b>	Geostationary Orbit	<b>UPL</b>	User Packet Length
<b>GPP</b>	General Purpose Processor	<b>USD</b>	United States Dollar
<b>GPU</b>	Graphic Processing Unit	<b>USRP</b>	Universal Software Radio Peripheral
<b>HDL</b>	Hardware Description Language	<b>V2V</b>	Vehicle to vehicle
<b>HTS</b>	High Throughput satellite	<b>VCM</b>	Variable Coding & Modulation
<b>Hz</b>	Hertz	<b>VHDL</b>	Very High-Speed Integrated Circuit HDL
<b>IEEE</b>	Institute of Electrical and Electronics Engineers	<b>VHF</b>	Very High Frequency
<b>IF</b>	Intermediate Frequency	<b>W</b>	Watts
<b>IoT</b>	Internet of Things		
<b>ISRO</b>	Indian Space Research Organisation		
<b>ISS</b>	International Space Station		
<b>JAXA</b>	Japan Aerospace Exploration Agency		



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



# LISTA DE FIGURAS

Figura 1.1 Diagrama de bloques de un transmisor SDR	44
Figura 1.2 Diagrama de bloques de un receptor SDR	44
Figura 1.3. Interfaz gráfica de GNU Radio	51
Figura 1.4. Crecimiento de la capacidad de transmisión satelital	53
Figura 1.5. Es/No [dB] vs BER para diferentes modulaciones comúnmente utilizadas en ACM	54
Figura 1.6. Modo de operación CCM para DVB-S (QPSK 2/3) y para DVB-S2 (QPSK 4/5) para radiodifusión	56
Figura 1.7. Modo de operación VCM para radiodifusión	56
Figura 1.8. Diagrama de bloques funcional del estándar DVB-S2	57
Figura 1.9. Frame generado a la salida del subsistema Mode Adaptation de DVB-S2	58
Figura 1.10. Implementación del codificador CRC-8	61
Figura 1.11. BBHEADER y DATA FIELD a la salida del subsistema MODE ADAPTATION	62
Figura 1.12. BBFRAME a la salida de Stream Adaptation	65
Figura 1.13. Secuenciador PRBS para aplicar la etapa de scrambling al BBFRAME	65
Figura 1.14. Formato del FECFRAME antes del proceso de Interleaving (Normal $n_{ldpc}=64,800$ bits; Short $n_{ldpc}=16,200$ bits)	66
Figura 1.15. Bloque de Bit Interleaving para 8PSK. Escritura por columna, lectura por filas.	67
Figura 1.16. Mapeo de bits en la constelación QPSK	68
Figura 1.17. Mapeo de bits en la constelación 8PSK	69
Figura 1.18. Constelación 16APSK	69
Figura 1.19. Constelación 32APSK	70
Figura 1.20. Formato del Physical Layer Frame (PLFRAME)	72
Figura 1.21. Generador del código PLS	73
Figura 1.22 Matriz Generadora del campo MODCOD dentro del PLHEADER	74
Figura 1.23. PL Scrambling	75
Figura 1.24. Generador del código de aleatorización ( $n=0$ ).	75
Figura 1.25. CubeSats de 1U, 2U y 3U	77
Figura 1.26. CubeSat 3U desarrollado por Planet Labs	78
Figura 1.27. CubeSats puestos en órbita alrededor del mundo	79
Figura 1.28. Histórico 2013-2018 de lanzamientos nanosatelitales y pronóstico al 2023	79
Figura 1.29. Número de compañías enfocadas a la tecnología nanosatelital de 1990 a 2019	81
Figura 1.30. Empleo de las diferentes bandas satelitales en proyectos CubeSat	82
Figura 1.31. Plataforma de pruebas SCan desarrollado por la NASA	83
Figura 1.32. Diagrama de bloques del transmisor de un CubeSat de Planet Labs	84
Figura 1.33. Diagrama de bloques de recepción de Planet Labs	85
Figura 1.34. Imágenes de Planet en sus diferentes bandas espectrales	86
Figura 1.35. Fotografía tomada por un satélite de la constelación Planet Labs	87
Figura 2.0. Cuadro Nacional de Atribución de Frecuencias. 2025 -2300 MHz	91
Figura 2.1. System Tool Kit v11.5.0	92
Figura 2.2. Cuadro de bienvenida a STK	93
Figura 2.3. Configuración del periodo de la misión en STK	93
Figura 2.4. Inserción de objeto en STK	94
Figura 2.5. Configuración de ubicación de la ET de la UAT, Juriquilla	95
Figura 2.6. Mapa 2D de la ET de la UAT, Juriquilla	95
Figura 2.7. Inserción del objeto tipo sensor para la ET	96
Figura 2.8. Definición del tipo de sensor	96
Figura 2.9. Estación terrena de Juriquilla configurada en STK	97
Figura 2.10. Inserción de objeto tipo satélite	97
Figura 2.11. Orbit Wizard ISS	98



Figura 2.12. Órbita de la ISS configurada en STK	98
Figura 2.13. Orbit Wizard SSO	99
Figura 2.14. Satélite CubeSat de 3U en órbita SSO configurado en STK	99
Figura 2.15. Mapa 2D del escenario configurado en STK	100
Figura 2.16. Mapa 3D del escenario configurado en STK	101
Figura 2.17. Mapa 3D del escenario configurado en STK	101
Figura 2.18. Mapa 3D del escenario configurado en STK	101
Figura 2.19 Selección de opción Access	102
Figura 2.20. Configuración de la ventana Access para la órbita SSO	103
Figura 2.21. Report Manager para generar reporte AER de la órbita SSO	103
Figura 2.22 Reporte generado en STK	104
Figura 2.23. Report Manager para generar reporte AER de la órbita ISS	104
Figura 2.24 Datos de acceso de la órbita ISS	105
Figura 2.25 Datos de acceso de la órbita SSO	105
Figura 2.26. Estadísticas globales del acceso ECXSAT-To-ISS_Orbit	106
Figura 2.27. Estadísticas globales del acceso ECXSAT-To-SSO_Orbit	106
Figura 2.28. Reporte AER correspondiente a la órbita ISS exportado en .csv	107
Figura 2.29. Accesos registrados por las órbitas ISS y SSO de enero a diciembre de 2020	109
Figura 2.30. Accesos registrados por las órbitas ISS y SSO correspondientes a enero 2020	109
Figura 2.31. Histograma de los tiempos de acceso para la órbita ISS	110
Figura 2.32. Histograma de los tiempos de acceso para la órbita SSO	110
Figura 2.33. Histograma de los rangos para la órbita ISS	111
Figura 2.34. Histograma de los rangos para la órbita SSO	112
Figura 2.35. Gráfica Rango vs Número de muestra de las órbitas SSO e ISS	113
Figura 3.1. BER vs Eb/No de las modulaciones QPSK y 8PSK	118
Figura 3.2. Diagrama de bloques DVB-S2 implementado en Simulink	119
Figura 3.3. Selección de la modulación y codificación en Simulink	120
Figura 3.4. Constelaciones I-Q QPSK y 8PSK generadas en Simulink	120
Figura 3.5. Gráfica BER vs Es/No para diferentes ModCods obtenidos de Simulink	121
Figura 3.6. Valores de Es/No ideal para los diferentes ModCod establecidos por la ETSI	122
Figura 3.7. Atenuación producida por la absorción atmosférica	126
Figura 3.8. Mapa CCIR de regiones de intensidad de lluvia	128
Figura 3.9. Gráfica de los valores de SNR para la órbita ISS y SSO con $P_{tx}=0.5W$ y $1W$	132
Figura 3.10. Histograma de los valores de SNR para la órbita ISS con $P_{tx}=0.5W$	133
Figura 3.11. Histograma de los valores de SNR para la órbita ISS con $P_{tx}=1W$	133
Figura 3.12. Histograma de los valores de SNR para la órbita SSO con $P_{tx}=0.5W$	134
Figura 3.13. Histograma de los valores de SNR para la órbita SSO con $P_{tx}=1W$	134
Figura 3.14. Gráfica de los valores de SNR para la órbita ISS	135
Figura 3.15. Gráfica de los valores de SNR para la órbita SSO	136
Figura 3.16 Tasa de datos alcanzada por los diferentes ModCod en DVB-S2	137
Figura 3.17. Histograma de los ModCod para la órbita ISS   $P_{tx}=0.5W$	140
Figura 3.18. Histograma de los ModCod para la órbita ISS   $P_{tx}=1W$	140
Figura 3.19. Histograma de los ModCod para la órbita SSO   $P_{tx}=0.5W$	141
Figura 3.20. Histograma de los ModCod para la órbita SSO   $P_{tx}=1W$	141
Figura 3.21. Tasa de datos CCM vs ACM para la órbita ISS	142
Figura 4.1. Sistema Operativo Ubuntu 18.04.2 LTS	146
Figura 4.2. GNU Radio Companion 3.7.13.4	146
Figura 4.3. Librerías enfocadas a la Televisión Digital en GNU Radio	147
Figura 4.4. Diagrama de bloques del transmisor DVB-S2 en GNU Radio	148
Figura 4.5. Bloque File Source y configuración general	149
Figura 4.6. Bloque BBheader con configuración de FEC 9/10	149
Figura 4.7. Bloque BBscrambler. FECFRAME normal y FEC rate 9/10	150



Figura 4.8. Bloque BCH Encoder configurado a tamaño Normal con un FEC de 9/10	151
Figura 4.9. Polinomios BCH	151
Figura 4.10. LDPC Encoder	152
Figura 4.11 Bloque de Interleaving	152
Figura 4.12 Modulador DVB-S2 configurado a un ModCod de 16APSK 9/10	153
Figura 4.13. Physical Layer Frame. 16APSK 9/10 con portadoras piloto activadas	153
Figura 4.14. FFT Filter	154
Figura 4.15. Espectro en frecuencia y constelación de la señal DVB-S2 a la salida del modulador	154
Figura 4.16. Espectro en frecuencia de la señal DVB-S2 con roll off de 0.20	155
Figura 4.17. Máscara del espectro de la señal esperado a la salida del modulador DVB-S2 para un factor de roll off de 0.35	155
Figura 4.18. Mesa de pruebas propuesta para el transmisor DVB-S2 basado en SDR	156
Figura 4.19. HackRF One	157
Figura 4.20. PCB y elementos electrónicos del HackRF One	159
Figura 4.21. Diagrama de bloques del HackRf One	159
Figura 4.22. Resultado del comando hackrf_info	161
Figura 4.23. Opciones disponibles para hackrf_transfer	162
Figura 4.24. hackrft_transfer en terminal	162
Figura 4.25. Resultados de la prueba de recepción con hackrf_transfer	163
Figura 4.26. Resultados de la prueba de transmisión con hackrf_transfer	164
Figura 4.27. Bloques Osmocom source y sink cargados en GNU Radio	165
Figura 4.28. Bloques osmocom Source y osmocom Sink	165
Figura 4.29 Parámetros de configuración de osmocom Source (izquierda) y osmocom Sink (derecha)	165
Figura 4.30. Receptor FM implementado en GNU Radio	167
Figura 4.31. Parámetros configurados de osmocom Source para el receptor FM	168
Figura 4.32. Espectro de la señal FM recibida en GNU Radio	168
Figura 4.33. Integración de GNU Radio con HackRF One implementando un receptor FM	169
Figura 4.34. HackRF One en modo de recepción	169
Figura 4.35. Receptor satelital GTMedia V9	170
Figura 4.36. Layout del receptor satelital GTMedia V9	171
Figura 4.37. Antena dipolo G401-1	171
Figura 4.38 Patrón de radiación de la antena dipolo G401-1	172
Figura 4.39. Adaptador coaxial F-RPSMA utilizado en el receptor satelital	172
Figura 4.40 Configuración Osmocom Sink para la prueba de transmisión de DVB-S2 empleando HackRF	173
Figura 4.41 Mapa de constelación I-Q de la señal 16APSK generada en GNU Radio	174
Figura 4.42 Espectro en frecuencia de la señal de 1.28 GHz generada por GNU Radio	174
Figura 4.43 HackRF con indicador de TX activado	175
Figura 4.44. Espectro de la señal DVB-S2 recibida a una frecuencia de 1.28 GHz	175
Figura 4.45. Receptor satelital GTmedia V9 super con antena dipolo	176
Figura 4.46 Configuración en receptor satelital previo a la prueba DVB-S2	176
Figura 4.47 Búsqueda de la señal transmitida en el receptor satelital	177
Figura 4.48 Canales encontrados por el receptor satelital	177
Figura 4.49. Intensidad y calidad de la señal detectada por el receptor satelital	178
Figura 4.50. Archivo de video transmitido desde la mesa de pruebas y visualizado en el receptor satelital	178
Figura 4.51. Constelaciones generadas desde GNU Radio para las modulaciones QPSK, 8PSK, 16APSK y 32APSK	179
Figura 4.52 Configuración de la mesa de pruebas para la estimación del SNR	181
Figura 4.53 Potencia de ruido medido en el analizador de espectros	181
Figura 4.54. Gráfica de la potencia mínima vs potencia de recepción sin errores para la mesa de pruebas	184



Figura 4.55. Gráfica de SNR ETSI vs SNR experimental obtenido de la mesa de pruebas	185
Figura 4.56. Gráfica comparativa entre los valores de SNR ETSI, SNR experimental mínimo y sin errores)	185
Figura 4.57. Definición de los valores de la máscara espectral del estándar DVB-S2	187
Figura 4.58. Espectro de la señal   Transmisión desactivada	189
Figura 4.59. Espectro de la señal   QPSK 1/4 y QPSK 9/10	190
Figura 4.60. Espectro de la señal   8PSK 3/5 y 8PSK 9/10	190
Figura 4.61. Espectro de la señal   16APSK 2/3 y 16APSK 9/10	190
Figura 4.62. Espectro de la señal   32APSK 3/4 y 32APSK 9/10	191
Figura 4.63. Espectro de la señal DVB-S2   Modcod: QPSK 1/4   Ganancia FI=45 dB	192
Figura 5.1. Estructura de un FPGA	196
Figura 5.2. Tarjeta de desarrollo Basys-3 de Xilinx	201
Figura 5.3. Entorno gráfico de LabView	203
Figura 5.4 Entorno de desarrollo de Vivado	204
Figura 5.5. Simulador de la herramienta Vivado	204
Figura 5.6. Entorno gráfico de Quartus Prime	205
Figura 5.7 Front End modelo RTL-SDR	208
Figura 5.8 Front End FUNcube Pro+	209
Figura 5.9. Front End HackRF	209
Figura 5.10. Front End Blade RF 2.0	210
Figura 5.11. Front End ETTUS B200	210
Figura 5.12. Esquema de un CubeSat 3U	211
Figura 5.13. Front End Ettus USRP B205 mini-i	212
Figura 5.14. Front End Lime SDR	213
Figura 5.14 SDR integrado modelo $\mu$ SDR-C del proveedor Space Micro	217
Figura 5.15. USRP E320 de Ettus Research	218
Figura 5.16 Esquema del proyecto Gom-X3 de GomSpace	219
Figura 5.17 Nanocom SDR de GomSpace	222
Figura 5.18. Comparativa radial de los SDR Integrados	222
Figura 5.19 Diagrama de bloques del USRP E320 de Ettus Research	223
Figura 5.20 Diagrama de bloques del NanoCom SDR de GomSpace	224
Figura 5.21 Comparativa en tamaño del HackRF, LimeSDR, E320 y NanoCom SDR	228
Figura 5.22 Gráfica comparativa de las 4 configuraciones de SDR propuestas	229
Figura 5.23 Gráfica radial comparativa de las configuraciones de SDR propuestas	230
Figura 5.24 a) Estructura CubeSat 3U b) Antena parche c) Amplificador de potencia d) Cámara	232
Figura 5.25 SDR propuesto   Configuración 1: FPGA + HackRF	233
Figura 5.26 SDR propuesto   Configuración 2: FPGA + LimeSDR	234
Figura 5.27 SDR propuesto   Configuración 3: Ettus E320	235
Figura 5.28 SDR propuesto   Configuración 4: NanoCom SDR	236
Figura 5.29. Perspectivas frontal y lateral de las configuraciones propuestas	237
Figura 5.30 Gráfica radial comparativa de los SDR basados en el LimeSDR y NanoCom	238
Figura 5.31 Diagrama del SDR propuesto (FPGA Cyclone IV + LimeSDR)	241
Figura 5.32. Propuesta final de CubeSat 3U con SDR y cámara integrados	242



# LISTA DE TABLAS

Tabla J.1. Proyectos satelitales mexicanos en la actualidad	19
Tabla 1.1 Comparativa de dispositivos programables para SDRs	48
Tabla 1.2 Ventajas y desventajas de los SDR y los radios tradicionales	50
Tabla 1.3. Parámetros típicos para eficiencia espectral y $E_s/N_0$ para cada tipo de FEC y modulación para DVB-S2	57
Tabla 1.4. Valores tomados por INPUT INTERFACE	59
Tabla 1.5. Valores de MODCOD correspondientes al Comando ACM	60
Tabla 1.6. Parámetros de codificación (normal FECFRAME=64800)	62
Tabla 1.7. Valores de MATYPE	63
Tabla 1.8. Parámetros de codificación (FECFRAME Normal; $n_{ldpc}=64,800$ bits)	66
Tabla 1.9. Parámetros de codificación (FECFRAME corto; $n_{ldpc}=16,200$ bits)	67
Tabla 1.10. Estructura del Bit Interleaver	68
Tabla 1.11. Valores de $\gamma$ óptimos para 16APSK	70
Tabla 1.12. Valores óptimos de $\gamma_1$ y $\gamma_2$ para 32APSK	71
Tabla 1.13 Valores dados para el XFECFRAME	71
Tabla 1.14 Etiquetas asignadas a cada ModCod en el encabezado PLHEADER	73
Tabla 1.15. Ejemplos de proyectos CubeSat con aplicaciones comerciales y tecnológicas	80
Tabla 1.16. Compañías que ofrecen servicios en base a plataformas CubeSat	81
Tabla 1.17. Comparativa de las órbitas ISS y SSO para aplicaciones de observación terrestre	84
Tabla 1.18 Características de las fotografías capturas por lo constelación Planet Labs	87
Tabla 2.1. Consideraciones del Segmento Espacial	90
Tabla 2.2 Consideraciones del Segmento Terrestre	90
Tabla 2.3. Estadísticas globales de los reportes obtenidos para las órbitas ISS y SSO	108
Tabla 3.1. Valores de Bit Error Rate	116
Tabla 3.2. Valores de Modcod vs $E_s/N_0$ bajo las referencias ETSI y Simulink	123
Tabla 3.3. Valores de conmutación Modcod vs SNR	124
Tabla 3.4. Intensidad de lluvia excedida	127
Tabla 3.5. Coeficientes de atenuación específica	128
Tabla 3.6. Resultados obtenidos del cálculo de enlace descendente	131



Tabla 3.7. Capacidad del canal para las órbitas ISS y SSO durante el 2020	138
Tabla 3.8. Resumen de resultados del análisis orbital y de comunicaciones para la Órbita ISS	143
Tabla 3.9. Resumen de resultados del análisis orbital y de comunicaciones para la Órbita SSO	143
Tabla 4.1. Descripción de los parámetros de configuración de los bloques Osmocom	166
Tabla 4.2. Características de la antena dipolo G401-1	172
Tabla 4.3. SNR experimental y potencias de recepción de la mesa de pruebas	183
Tabla 4.4. Tasa de datos y capacidad del canal teóricos y experimentales	186
Tabla 4.5 Definición de los valores la máscara espectral para $f_N=2.5\text{MHz}$ y $f=1.28\text{GHz}$	188
Tabla 4.6. Marcadores configurados en el analizador de espectros	189
Tabla 4.7. Valores de potencia relativa para los diferentes ModCod	191
Tabla 4.8. Valores de potencia de las señales espurias	192
Tabla 5.1. Comparativa de FPGAs comerciales	200
Tabla 5.2. LUTs utilizados en un FPGA para bloques de DVB-S2	201
Tabla 5.3. Transceivers comerciales	206
Tabla 5.4. Front End comerciales	208
Tabla 5.5. Front End comerciales clasificados	214
Tabla 5.6. Características principales de los SDR integrados	216
Tabla 5.7 Características principales de los Soc de la familia Zynq 7000 de Xilinx	217
Tabla 5.8. Características principales de los SDR integrados categorizados	221
Tabla 5.9 Elementos seleccionados para conformar el SDR	225
Tabla 5.10 Configuraciones SDR propuestas	225
Tabla 5.11 Diseño de SDR   Configuración 1	226
Tabla 5.12 Diseño de SDR   Configuración 2	226
Tabla 5.13 Diseño de SDR   Configuración 3	227
Tabla 5.14 Diseño de SDR   Configuración 4	227
Tabla 5.15 Elementos propuestos de la plataforma base	231
Tabla 5.16 Parámetros de la solución final de SDR	242



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



## CAPÍTULO 1. MARCO TEÓRICO

En este primer capítulo se establecen las bases teóricas de la investigación las cuales se enfocan en cada uno de los factores tecnológicos que dan su relevancia al proyecto: los sistemas SDR, el estándar DVB-S2 y la tecnología CubeSat.

Como primer acercamiento en el Estado del Arte se enlistan algunos de los proyectos de investigación relacionados con los SDR y la tecnología satelital, tanto nacionales como extranjeros, que funcionan como referencia o guía para el desarrollo del trabajo actual.

Después se muestra un marco teórico de los proyectos SDR donde su descripción, componentes, ventajas y desventajas son indicadas. Respecto al estándar DVB-S2, se muestra el diagrama a bloques del mismo y se describen de manera detallada cada uno de sus elementos. Por último, se hace un acercamiento a la tecnología CubeSat gracias al cual es posible observar sus generalidades, su evolución desde su origen en el 2003 y su relación con proyectos de comunicaciones SDR y ACM. Es así como se enlistan las características principales y técnicas de una serie de CubeSats utilizados para la observación terrestre y se indican las características principales de las imágenes capturadas por estas constelaciones. Esta información es posteriormente empleada para brindar las bases de la misión satelital analizada durante el resto de la investigación.

### 1.1 ESTADO DEL ARTE

El inicio de este proyecto de investigación se realizó en torno al análisis y revisión de diversos artículos, libros y trabajos de tesis que constituyen el estado del arte de los sistemas de comunicaciones SDR. A continuación se enlistan algunas de las fuentes más importantes que otorgan forma al trabajo de tesis actual y que fueron desarrolladas para proyectos SDR, ACM y sistemas de comunicaciones satelitales.

Primeramente, en [1\_1] se muestra el desarrollo de un módulo capaz de trabajar con las modulaciones QPSK, 8PSK y 16QAM, el cual está particularmente enfocado a un proyecto microsatelital, por lo que también se realiza un análisis orbital empleando la plataforma STK. El desarrollo del modulador se hace en VHDL para la tarjeta Basys 3 de Xilinx, haciendo énfasis en la implementación de un módulo de tolerancia a fallas para proveer al sistema de protección basada en software contra la radiación.

En [1\_2] se realiza el desarrollo de un sistema SDR de transmisión y recepción FSK donde el procesamiento de la señal es llevada a cabo en GNU Radio, y la manipulación y acondicionamiento de la señal en RF se hace empleando un Front End modelo USRP N210 de la marca Ettus Research.

En relación a un proyecto nanosatelital en órbita se revisa [1\_3] correspondiente al CubeSat Aerocube para el cual se desarrolló un SDR transmisor con una frecuencia de operación de 915 MHz y que es capaz de transmitir bajo las modulaciones BPSK, QPSK y 8PSK a una tasa de símbolos de 1MSps. Para la implementación del sistema se emplea el SoC (System On Chip) Zynq 7020 de Xilinx y un transceiver Lime Micro LMS6002D.

Una de las herramientas particularmente estudiadas durante esta primera etapa del proyecto fue Matlab Simulink. En [1\_4] es posible visualizar el paso a paso del diseño e implementación de un sistema de transmisión ADS-B (Automatic Dependent Surveillance – Broadcast) empleando la interfaz gráfica de Simulink para el armado de los bloques de



procesamiento los cuales son convertidos a código VHDL empleando el módulo HDL Encoder propio de Simulink y posteriormente compilados en el SoC Zynq 7000. El transceiver considerado es el AD9361 de Agile.

En [1\_5] es posible encontrar un análisis comparativo realizado por la Universidad de Surrey de diferentes Front End tipo COTS disponibles en el mercado y de algunos sistemas SDR con enfoque satelital, así como los lineamientos básicos para dimensionar un sistema de comunicaciones SDR.

Respecto al estándar DVB-S2, en [1\_6] se consideran diferentes simulaciones y análisis del estándar empleando nuevamente Matlab Simulink con las modulaciones QPSK, 8PSK, 16APSK y 32 APSK. Las simulaciones se centran en la relación entre la SNR y el BER para cada una de las modulaciones dadas por el estándar.

Finalmente, en [1\_7] se hace el desarrollo en VHDL de un sistema de transmisión y recepción capaz de operar a 100 MHz utilizando el FPGA Spartan-6 LX45 de Altera.



## 1.2 RADIO DEFINIDA POR SOFTWARE

Según el SDR Forum, en colaboración con el grupo P1900.1 de la IEEE, un SDR es “una radio en la cual algunas o todas sus funciones dadas por los elementos de su capa física son definidas por software” [1\_8]. Los SDR son radios que combinan el uso de módulos basados en hardware con módulos configurables y diseñados con base en dispositivos reprogramables capaces de trabajar a diferentes frecuencias de operación y con diferentes esquemas de modulación sin tener las limitantes físicas de sus antecesores. Este nuevo enfoque es una respuesta a la demanda de flexibilidad que la tecnología ha impuesto en torno a los sistemas de comunicaciones de manera general: uso de diferentes frecuencias, esquemas de modulación y codificación variables, aplicación de diversos protocolos, variación en las áreas de cobertura, etc. El concepto SDR fue acuñado por Joseph Mitola en 1991 después de desarrollar una estación base para telefonía GSM al combinar un receptor digital de la marca Ferdensi con un transmisor de la marca E-Systems, lo cual representaría el primer transceiver SDR de la historia.

### 1.2.1 Diagrama de bloques de un SDR

Los SDR deben su gran flexibilidad y capacidad de configuración debido al hecho de que el procesamiento de las señales es realizado de forma digital. Esto es, la selección de canal, modulación, demodulación, codificación, decodificación, entre otros procesos en banda base son realizados en software, el cual a su vez corre dentro de microprocesadores y hardware programable.

En general, tomando como referencia el bloque de transmisión de un SDR, se genera una señal en banda base la cual es procesada (codificada y modulada al menos) y llevada a una frecuencia intermedia para finalmente ser trasladada a una radiofrecuencia (RF) que es propagada hacia el espacio libre haciendo uso de una antena. En el bloque de recepción, la señal de RF capturada por la antena es amplificada en un rango determinado de frecuencias, para luego ser convertida a una señal analógica de frecuencia intermedia. Un convertidor analógico-digital (ADC) digitaliza y muestrea esta señal, la cual es inyectada a un mezclador encargado de trasladarla a una frecuencia en banda base. La siguiente etapa consta de un filtro FIR (Finite Impulse Response) que permite el paso de una sola señal y limita su ancho de banda. En conjunto, el mezclador junto con el filtro, funcionan como un convertidor digital que antecede al bloque de procesamiento en banda base. En este último bloque se realizan procedimientos tales como la demodulación y decodificación. Este bloque es normalmente implementado por un dispositivo programable como [1\_9]:

- DSP (Digital Signal Processor)
- GPU (Graphics Processing Unit)
- GPP (General Purpose microProcessors)
- FPGA (Field Programmable Gate Array)

Este procedimiento se resume en el diagrama de bloques de un SDR mostrado en las figura 1.1 y 1.2, conformado por:

- Procesador en Banda Base
- Front End Digital



- DAC/ADC
- Front End Analógico
- Antena

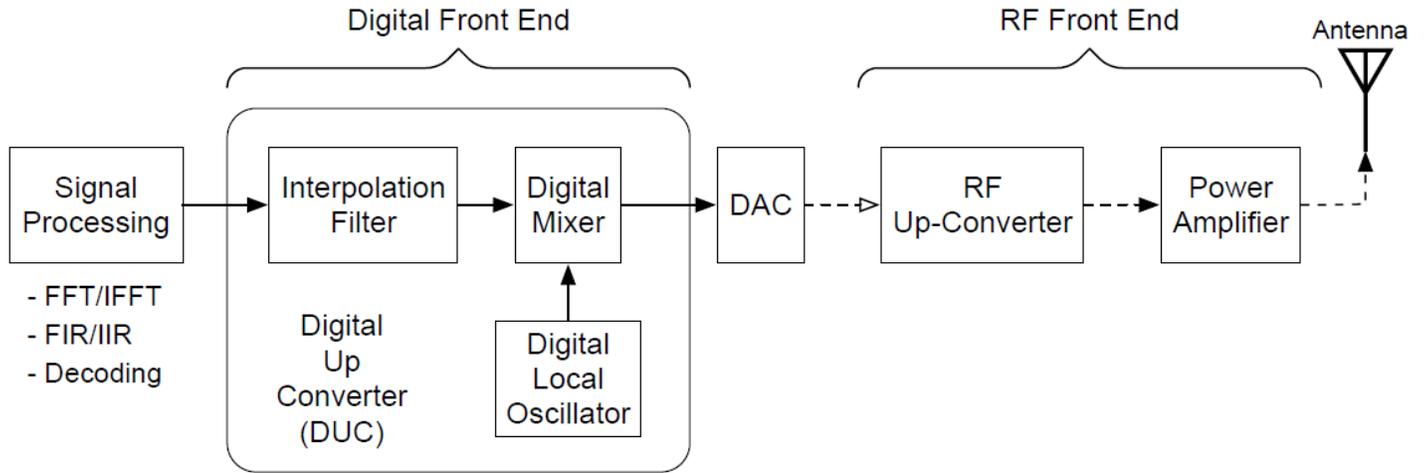


Figura 1.1 Diagrama de bloques de un transmisor SDR [1\_10].

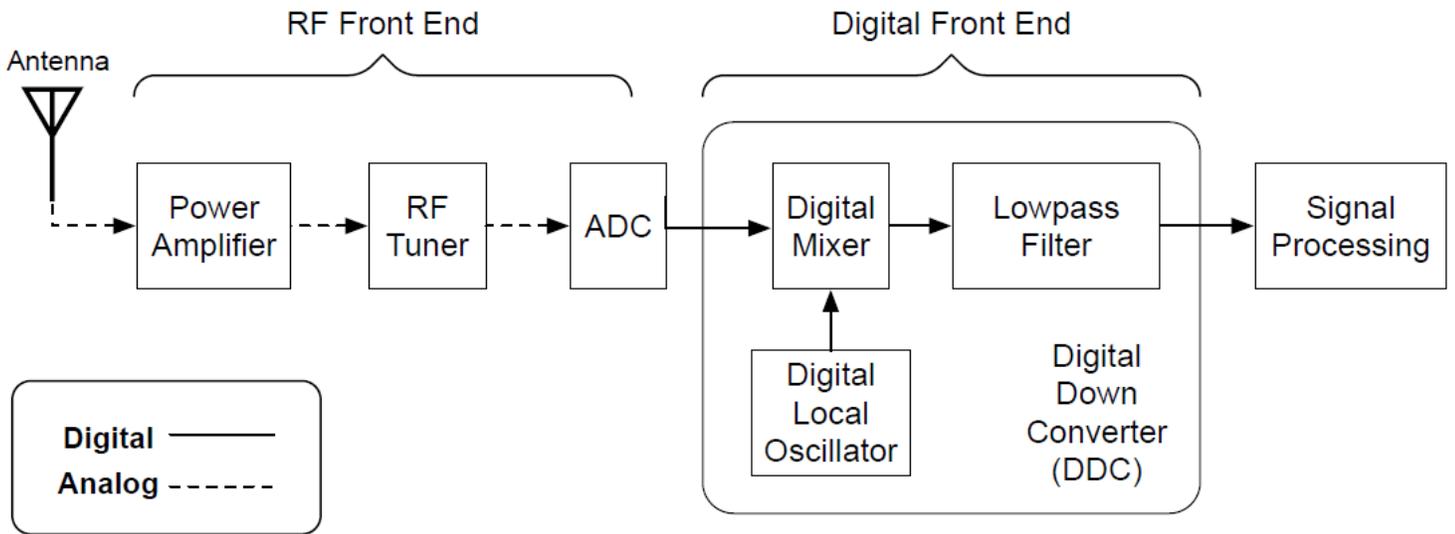


Figura 1.2 Diagrama de bloques de un receptor SDR [1\_10].



A continuación se describirá brevemente cada uno de estos elementos:

### 1) Antena:

De manera ideal, las antenas deben de cumplir con una lista de requerimientos mínimos para poder operar con las características de un SDR tales como la flexibilidad de sintonizar diferentes bandas de frecuencia, reconfiguración de su patrón de radiación y rechazo a la interferencia.

Los SDR utilizan normalmente una antena con un amplio rango de operación o, en su defecto, múltiples antenas que permitan dicha cobertura [1\_11]. Un rango de frecuencias que abarquen por lo menos cinco octavas es requerido así como una ganancia de al menos 0 dBi.

### 2) Front End Analógico (RF):

Consta de un circuito, basado en elementos analógicos, capaz de manipular señales, tanto en la recepción como en la transmisión, en diferentes bandas de radiofrecuencia. Para ello tiene la capacidad de transitar de señales de frecuencia intermedia a señales de RF y viceversa. Su proceso de operación puede dividirse en dos bloques: transmisión y recepción.

En la transmisión, una señal digital es convertida a una señal analógica por el Convertidor Digital Analógico (DAC), con la cual alimenta el Front End Analógico. Esta señal analógica es mezclada y convertida a una señal de RF para posteriormente ser amplificada por un amplificador de potencia y ser llevada a la antena.

En el bloque de recepción, la antena captura la señal de RF proveniente del espacio libre. La entrada de la antena es conectada al Front End utilizando un acoplador para garantizar la óptima transferencia de la señal hacia un Amplificador de Bajo ruido (LNA, Low Noise Amplifier) el cual amplifica las señales recibidas y al mismo tiempo minimiza el nivel de ruido de la señal. La señal amplificada es mezclada con una señal generada por un oscilador local para ser transformada a una frecuencia intermedia y posteriormente es llevada al convertidor ADC.

### 3) ADC y DAC

El DAC es responsable de transformar las muestras digitales entregadas por el Front End Digital en una señal analógica. Su uso es esencial en el bloque transmisor de un SDR.

Por otro lado, el ADC se emplea en el bloque receptor del SDR siendo responsable de convertir señales continuas en el tiempo a señales binarias discretas. Los parámetros importantes para caracterizar el comportamiento de este convertidor son [1\_12]:

- Relación Señal a Ruido (SNR) a su salida.
- Resolución (número de bits que entrega por cada muestra procesada).
- Rango Dinámico Libre de Espurio (SFDR). La relación entre la intensidad de la frecuencia fundamental de la señal de información y la intensidad de la armónica más cercana, asociada al componente de ruido.
- Disipación de Potencia.



#### 4) Front End Digital

Este bloque se compone principalmente de dos bloques: el Convertidor Digital de Subida (DUC, Digital Up Converter) y el Convertidor Digital de Bajada (DDC, Digital Down Converter).

El DUC se utiliza en el bloque de transmisión. Su función es trasladar una señal en banda base a una señal de Frecuencia Intermedia. Esta señal de FI es posteriormente alimentada al DAC.

El DDC es la contra parte del DUC y es empleado en el bloque de recepción del SDR. El ADC convierte una señal de FI en muestras digitales las cuales son llevadas al DDC. Este convertidor consta de un mezclador digital y un oscilador con los cuales logra generar una señal digital en banda base. Finalmente, la señal en banda base es entregada al bloque de Procesamiento en Banda Base [1\_13].

#### 5) Procesamiento en Banda Base

En este bloque se llevan a cabo funciones relacionadas con el procesamiento de la señal tales como codificación, modulación, interleaving, scrambling y sus respectivos procesos inversos. En términos de codificación algunos ejemplos implementados en SDR son códigos convolucionales, turbo códigos, y codificación LDPC (Low Density Parity Check). En cuanto a las modulaciones se pueden encontrar implementaciones sencillas como BPSK (Binary Phase Shift Keying) hasta complejas utilizadas en tecnología LTE como 256-QAM.

Cuando se habla de SDR se puede asegurar que el Procesamiento en Banda Base es el eje principal del sistema. De manera general este procesamiento es implementado en dispositivos programables como se enlistó anteriormente. Parte vital de la implementación es el software, el cual provee de sus funcionalidades y características de procesamiento al SDR. El software puede ser desarrollado en diferentes ambientes de programación entre los que se encuentran GNU Radio, Matlab Simulink, Labview, Vivado, entre otros.

### 1.2.2. Dispositivos programables

Un dispositivo programable es una solución al planteamiento del procesamiento digital de las señales dentro una Radio Definida por Software. Existen diferentes dispositivos capaces de cumplir con el requerimiento de procesamiento impuesto por un SDR, cada una de estos con sus propias características, aplicaciones y particularidades. Estos dispositivos son tan importantes como aquellos que conforman el segmento analógico de un SDR.

Algunas de las consideraciones más importantes al momento de seleccionar un dispositivo programable son su velocidad de procesamiento, costo, consumo de potencia y la capacidad de reconfiguración, sin embargo existen otros factores importantes a considerar tales como tiempo y herramientas de desarrollo, licenciamiento y complejidad de programación.

Como se mencionó anteriormente los dispositivos programables utilizados en la implementación de un SDR son:

- DSP
- GPP
- GPU
- FPGA



En los siguientes párrafos se realiza una breve descripción de cada uno de estos para, posteriormente, presentar una tabla comparativa con el resumen de sus características de desempeño principales.

- **Digital Signal Processors (DSPs)**

Los DSPs son la primera tecnología de procesamiento con la que se desarrollaron los primeros SDR. Tienen la ventaja de contar con una completa flexibilidad, diversas posibilidades de aplicación, y un alto grado de adopción por parte de los desarrolladores y programadores. Son, además, dispositivos fabricados en gran escala con lo cual su costo es reducido.

De manera particular, presentan ciertas limitaciones al momento de optimizar procesos por lo cual son más bien utilizados como dispositivos de prototipado durante las fases iniciales de los proyectos o como solución final una vez que el diseño ha sido definido en su totalidad. Son utilizados para operaciones muy particulares al ser altamente eficientes al momento de llevar a cabo operaciones matemáticas.

Su arquitectura está especialmente diseñada para soportar los requerimientos operacionales del procesamiento digital de señales. Actualmente, al no poder soportar operaciones en tiempo real, son normalmente utilizados para apoyar las primeras etapas de procesamiento o almacenamiento de dispositivos más complejos como los FPGAs [1\_14].

- **General Purpose Processors (GPP)**

Se trata de procesadores con arquitecturas de 32 y 64 bits, tales como Intel y AMD, usualmente empleados en CPUs. Han sido optimizados para soportar una alta gama de aplicaciones por lo cual son altamente flexibles y poseen una alta velocidad de procesamiento [1\_15].

Funcionan como circuitos digitales sincronizados por un reloj interno y basados en el uso de registros. Son capaces de procesar información empleando un sistema de representación binaria. Pueden ser usados para múltiples propósitos y aplicaciones, eliminando la necesidad de diseñar circuitos específicos para cada aplicación, permitiendo la reducción de costos al implementar una determinada solución. Son, generalmente, una plataforma utilizada en la academia y la investigación debido a su gran flexibilidad, capacidad, y facilidad de programación, los cuales son aspectos importantes al momento de trabajar con SDRs. Los GPP han ido avanzando en su desempeño durante los últimos años gracias al desarrollo de la tecnología CMOS y al aumento en su capacidad de procesamiento impulsado por el uso de multinúcleos en un mismo procesador.

Dado que los GPP procesan la información de manera secuencial, presentan limitaciones al utilizarse en aplicaciones en tiempo real con una alta tasa de datos. Debido a esto es que se propone el uso de los GPUs explicados a continuación.

- **Graphics Processing Units (GPU)**

Los GPUs son procesadores especialmente diseñados para llevar a cabo tareas relacionadas con gráficos y procesar largos bloques de información recibida en tiempo real de forma paralela. Esto hace que tengan una serie de ventajas ante aplicaciones relacionadas con el procesamiento de video. En particular, los GPUs emplean el concepto llamado Single Program Multiple Data (SPMD) que permite la ejecución del mismo programa a múltiples líneas de instrucciones a la vez.



Este tipo de procesadores es compatible con arquitecturas paralelas lo cual les permite procesar vectores y gráficos. Han sido optimizados para crear y almacenar imágenes en un buffer para su posterior uso. Sus principales fabricantes son AMD y nVIDIA.

Sin embargo, el uso de GPUs en los SDR conlleva diversas complicaciones incluyendo complejidad en la arquitectura de la solución, lenguajes de programación de bajo nivel, y su consumo de potencia es mayor al de otras plataformas empleadas por SDRs. Los SDR utilizan los GPUs para operaciones aritméticas de punto flotante realizadas a altas velocidades y en paralelo con otros procesamientos [1\_16].

- **Field Programmable Gate Arrays (FPGA):**

Los FPGAs son chips que pueden ser configurados después de haber sido manufacturados. Se conforman de bloques lógicos programables, tales como bloques lógicos, de memoria y multiplicadores que son acompañados de trayectorias o regiones de operación, también configurables. Estos circuitos son capaces de implementar cualquier diseño o función y de modificarla cuando así se requiera [1\_17]. A pesar de que los FPGAs consuman mayor potencia y tengan mayores dimensiones que otros dispositivos programables, sus características de programación son la razón por la cual su adopción ha ido incrementando en un gran número de aplicaciones. Su tiempo de reconfiguración está dentro del orden de los milisegundos, permitiendo a los SDR conmutar entre diferentes modos y protocolos de manera eficiente. Otra gran diferencia es su tiempo y costo de fabricación, dado que los FPGAs pueden ser rápidamente reprogramados, su costo puede llegar a ser menor de los 1,000 USD y sus tiempos de fabricación son más acelerados. Su mayor desempeño a altas velocidades, su bajo consumo de potencia y portabilidad, comparado con los GPPs y DSPs, hacen que los FPGAs sean una de las soluciones más versátiles e importantes para el desarrollo de los SDRs. Son configurados usando lenguaje descriptivo como VHDL o Verilog. Sus principales compañías de fabricación son Xilinx y Altera.

En la tabla 1.1 se muestra una comparativa entre las características principales de estos dispositivos programables:

**Tabla 1.1 Comparativa de dispositivos programables para SDRs [1\_15].**

Característica / Dispositivo	DSP	GPP	GPU	FPGA
Capacidad DSP	Alta	Media	Alta	Alta
Aplicaciones Generales	Baja	Alta	Baja	Media
Flexibilidad	Baja	Alta	Media	Alta
Ejecución	Parcialmente Paralela	Secuencial	Paralela	Paralela
Tasa de datos	Medio	Bajo	Alto	Alta
Eficiencia energética	Alta	Media	Baja	Media
Proveedores	Texas Instruments	Intel, AMD	nVidia, AMD	Xilinx, Altera
Lenguajes de programación	C, Ensamblador	C, Java	CUDA, C	Verilog, VHDL
Tamaño	Pequeño	Moderado	Muy grande	Grande
Masa	>20g	>100g	>800g	>200g
Costo	>500 USD	>1,000 USD	>500 USD	>300 USD



### 1.2.3. Beneficios de los SDR

A continuación se enlistan algunos beneficios del uso de SDRs desde tres diferentes enfoques: Manufactura, proveedores y usuarios finales.

- **Manufactura**

Los SDR representan una familia de productos de radio que son implementados utilizando una arquitectura en común, permitiendo así que nuevos productos sean introducidos de manera más rápida y eficaz en el mercado.

El hecho de que haya una base de software permite la reutilización del mismo en las diferentes y nuevas versiones de los productos lo cual reduce significativamente los costos de desarrollo.

La capacidad de manipular y reconfigurar un SDR a distancia, incluso cuando ya está en funcionamiento, permite el arreglo de errores de fabricación tales como “bugs”. Esto reduce el tiempo y costo asociados al mantenimiento de los productos.

- **Proveedores**

Nuevas actualizaciones y mejoras pueden ser adicionadas a la infraestructura existente sin tener que invertir o modificar de manera importante las redes ya existentes.

El uso de una plataforma de radio en común compatible con diversos mercados y tecnologías reduce significativamente los gastos de soporte y operación de las redes de comunicaciones.

Descargas de software a distancia que permiten el incremento en la capacidad de los equipos y carga de actualizaciones lo cual puede tener un impacto directo a nivel económico.

- **Usuarios finales**

La existencia de redes de comunicaciones capaces de brindar diferentes soluciones y estándares de comunicación a la vez, hace posible que los usuarios puedan comunicarse cuando, donde y como lo requieran sin limitación de una tecnología en particular.

En la tabla 1.2 se enlistan las principales ventajas y desventajas de los SDR y los radios tradicionales.



Tabla 1.2 Ventajas y desventajas de los SDR y los radios tradicionales [1\_5].

Radios Tradicionales	Software Defined Radio
<p><b>Ventajas:</b></p> <ul style="list-style-type: none"><li>• Selección de componentes como el procesador y convertidores ADC/DAC son menos críticos.</li><li>• Económicos y con desarrollos ya disponibles.</li><li>• Bajo consumo de potencia.</li></ul>	<p><b>Ventajas:</b></p> <ul style="list-style-type: none"><li>• Diseño flexible. Múltiples bandas y modos de operación.</li><li>• Basados en software reconfigurable.</li><li>• Permite mejoras y actualizaciones en cualquier momento.</li></ul>
<p><b>Desventajas:</b></p> <ul style="list-style-type: none"><li>• Diseño de comunicación fija. Banda y modos de operación únicos.</li><li>• Alta complejidad en hardware</li><li>• Basados en componentes analógicos.</li><li>• Desempeño disminuye conforme los componentes cumplen su ciclo de vida.</li></ul>	<p><b>Desventajas:</b></p> <ul style="list-style-type: none"><li>• Alta complejidad en software</li><li>• Vulnerables en su software ante amenazas y hackeos.</li><li>• Procesadores y convertidores ADC/DAC de alta capacidad son requeridos.</li><li>• Alto consumo de potencia.</li></ul>

### 1.2.4. Principales aplicaciones

Como ya se ha mencionado, la tecnología SDR presenta múltiples ventajas frente a las tecnologías de radio tradicionales, lo cual permite el desarrollo de dispositivos multifuncionales capaces de operar en múltiples bandas. Esto se relaciona directamente con diversos factores claves que han hecho a los SDR tecnologías con una demanda en aumento durante los últimos años. Entre estos factores se encuentran la interoperabilidad entre redes, adaptabilidad inmediata a nuevas actualizaciones y protocolos, y bajos costos de desarrollo e implementación.

De acuerdo con la firma especializada de análisis de mercado “Markets & Markets”, el mercado de la tecnología SDR presentó un valor de 20 mil millones de dólares en 2017 y tendrá un valor de 30 mil millones de dólares para el año 2022 [1\_18].

Dentro de las principales aplicaciones de los SDR se encuentran:

- Aplicaciones militares como despliegue de redes a gran escala
- Sector público y prevención de desastres
- Estaciones Base para telefonía móvil
- Comunicaciones espaciales
- Sensores aplicados a los Sistemas de Navegación Satelital
- Comunicación vehículo-vehículo (V2V)
- Internet de las cosas (IoT)



### 1.2.5. Ambientes de desarrollo para SDR

#### GNU Radio

El proyecto GNU radio se inició en 2001 y fue fundado por Eric Blossom con el objetivo de desarrollar un marco de trabajo para software radio. Se trata de una herramienta de software libre y de código abierto, constituida por un conjunto de archivos y librerías que proporcionan bloques de procesamiento de señales, permitiendo así el diseño y la simulación de sistemas basados en software radio [1\_19].

Se basa en el uso de bloques gráficos para el diseño y armado de sistemas de comunicaciones. Es compatible con sistemas operativos tipo Linux y, con la adaptación de Front Ends analógicos, puede funcionar como un sistema SDR completo. Es mayormente utilizado en la academia y la investigación para correr simulaciones o implementar sistemas de comunicaciones básicos de manera rápida y sencilla. Incluye diferentes tipos de bloques tales como decodificadores, demoduladores y filtros. Esta herramienta de software puede ser utilizada con hardware externo adicional como puede ser el USRP, RTL2832, OsmosSDR, HackRf, entre otros, brindando la posibilidad de implementar físicamente un sistema basado en software radio, o bien, puede ser utilizada en un entorno de simulación.

Una de las características más importantes de GNU Radio es su permisividad de definir y agregar nuevos bloques a los ya preestablecidos, permitiendo mejoras en el diseño e implementación de los sistemas de comunicación, todo esto utilizando lenguajes de alto nivel como C++ o Python.

El funcionamiento de GNU radio se puede concebir como un diagrama a bloques, donde los nodos simbolizan los bloques de procesamiento de señal, y la interconexión entre ellos determinará el camino que seguirá la señal comenzando en una fuente y terminando en un destino. La figura 1.3 representa el funcionamiento de GNU radio en una aplicación de recepción de radio FM en la que se dispone de una fuente receptora de datos basado en un RTL-SDR, un mezclador, un filtro y finalmente una conversión digital-analógica que está lista para ser llevada y reproducida por un transductor de audio.

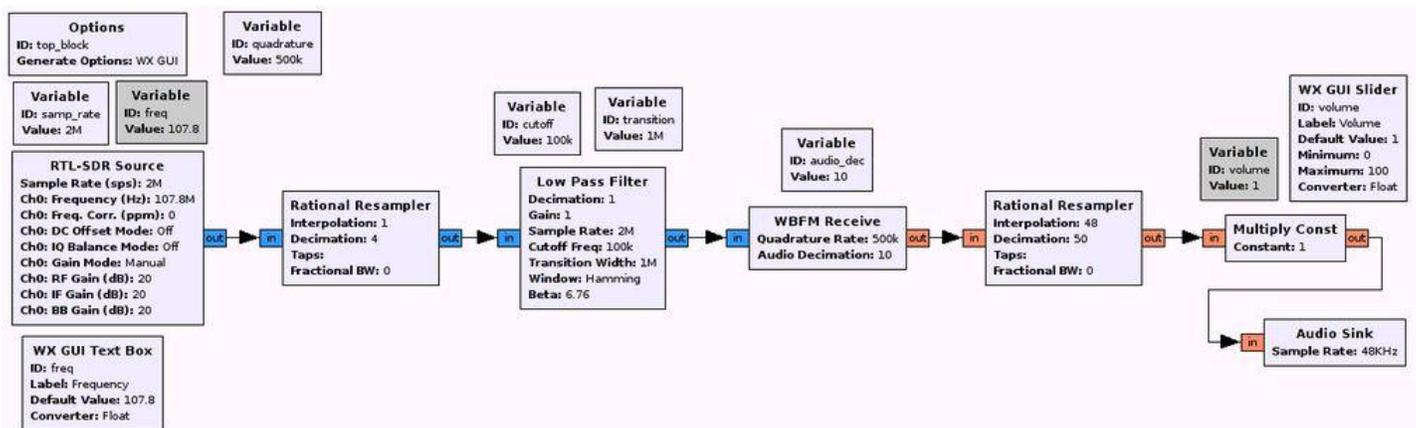


Figura 1.3. Interfaz gráfica de GNU Radio [1\_19].



### **Matlab Simulink**

Una de las soluciones más populares para el diseño de sistemas de comunicación es Simulink dado que ofrece diversas herramientas gráficas de diseño enfocadas en sistemas de comunicaciones y procesamiento de señales. Sin embargo, para poder convertir estos diseños gráficos en códigos adaptables para los diferentes microprocesadores disponibles se requiere el uso de coders que hacen la función de traductor entre un lenguaje gráfico y un lenguaje de alto nivel como puede ser C++ o de nivel medio como VHDL o Verilog compatibles con FPGAs. Por lo general estos coders requieren de la disponibilidad de diversas licencias lo cual puede incrementar el costo de desarrollo de los proyectos.

### **Labview**

Se trata de un ambiente de desarrollo brindado por National Instruments que ofrece un ambiente de programación visual para aplicaciones de automatización y control utilizado tanto por la industria como por la academia. Es muy similar a GNU Radio y Simulink, dado que el diseño de sistemas puede ser realizado conectando diferentes bloques dependiendo de la función que se desee realizar. También ofrece compatibilidad con USRP lo cual permite el rápido prototipado de sistemas de comunicaciones y simulaciones. De igual modo, permite el adicionar nuevos bloques y funciones a sus librerías con apoyo de C o Matlab.



### 1.3 ESTÁNDAR DVB-S2

Conforme pasa el tiempo, la tecnología avanza y la demanda de información se incrementa de manera exponencial. Las aplicaciones tecnológicas requieren de mayores capacidades para funcionar y la sociedad de la información así lo requiere. En 1990 una red celular de 64 kbps era suficiente para cumplir con las demandas de los usuarios. Hoy en día, la tecnología 5G aporta velocidades de transmisión arriba de 1 Gbps y se espera que esta tasa siga aumentando con las siguientes generaciones móviles [1\_20]. Lo mismo sucede en el terreno de la tecnología satelital, sobre todo en el campo de las telecomunicaciones. El primer estándar de televisión satelital, el DVB-S desarrollado en 1997, aportaba una tasa de datos de hasta 54 Mbps, los últimos avances del estándar, establecidos en 2014 bajo el estándar DVB-S2X, permiten un flujo de información de hasta 118 Mbps [1\_21]. Según Northern Sky Research, se pronostica que la capacidad de los satélites de comunicaciones geoestacionarios rebasen los 1000 Gbps en un plazo no mayor al 2025: los llamados High Throughput Satellites (HTS). Ver fig. 1.4.

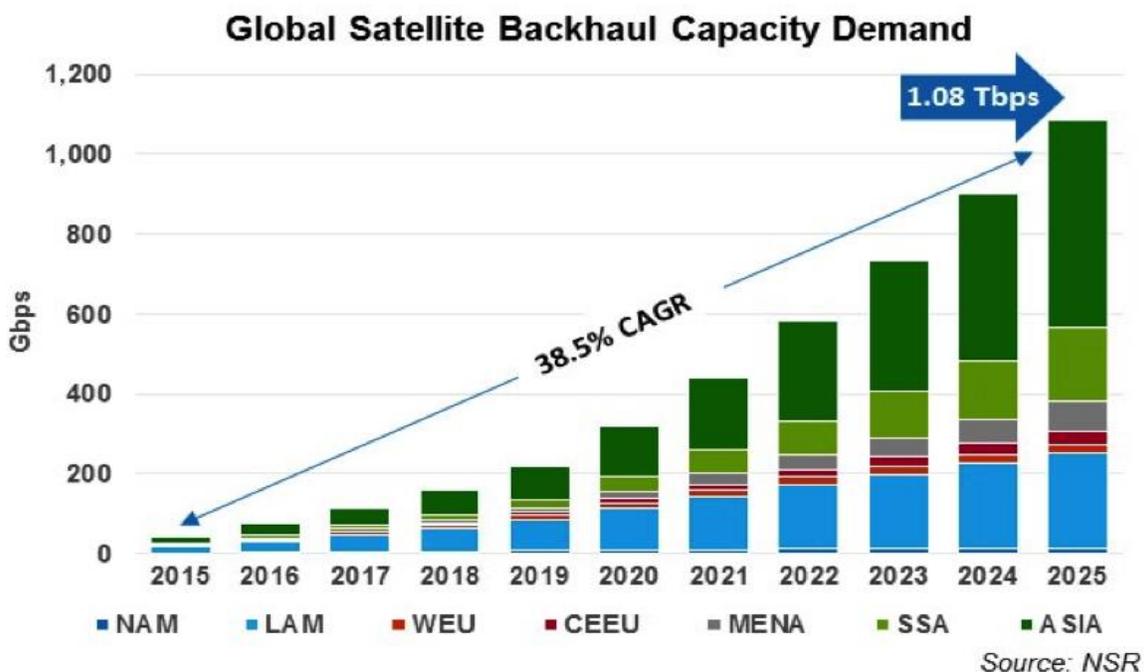


Figura 1.4. Crecimiento de la capacidad de transmisión satelital [1\_22].

Esta misma tendencia se observa en todos los niveles de la tecnología satelital. Las misiones y objetivos de los satélites LEO y MEO son cada vez más complejas y con requerimientos de velocidad de información más altos. Por ejemplo, el proyecto CubeSat LandMapper-BC de 6U, utiliza una cámara con resolución de 22 metros y es capaz de tomar 400 fotografías diariamente lo cual hace que genere alrededor de 1.2 TBytes de información cada día. Su enlace de bajada está diseñado para operar a una velocidad de 170 Mbps en banda Ka [1\_23]. Es decir que se requiere de una comunicación ininterrumpida de aproximadamente 16 horas para obtener toda la información que genera por día. En el caso particular de los CubeSats, la gran mayoría de estos son lanzados en órbitas bajas de entre 300 y 800 km. Esto provoca que giren en torno al planeta en diversas ocasiones durante el día dificultando así que se tengan una comunicación en línea de vista permanente con una estación terrena determinada. Hablando de un satélite orbitando a la misma altura que la estación espacial internacional, 408 km, podemos afirmar que su periodo orbital es de 93 minutos aproximadamente [1\_24] y que está en



línea de vista ( $10^\circ$  por encima del horizonte) con la estación de control satelital “ECXSAT” ubicada en Juriquilla, Querétaro, de 4 a 7 veces por semana durante un promedio de 2 minutos y 54 segundos por pase. Además, si bien los beneficios que los proyectos CubeSat aportan son numerosos, su tiempo de vida por lo general va de los 2 meses hasta los 2 años debido a que reingresan a la atmósfera y se desintegran o que los tiempos de vida de sus subsistemas son reducidos. Con esta información se puede entender la importancia de tener enlaces de comunicaciones optimizados entre los proyectos CubeSat y las estaciones en Tierra. La necesidad de aumentar las velocidades de transmisión se hace evidente sobre todo en aquellos proyectos con aplicaciones relacionadas con la observación terrestre.

De este modo, es que diferentes aplicaciones satelitales han optado por utilizar diferentes técnicas y esquemas de optimización de la transmisión de la información, una de ellas es la Codificación y Modulación Adaptativa (ACM) utilizada por primera vez en el estándar de Televisión Satelital DVB-S2 y actualmente utilizado en aplicaciones tales como LTE, WiMAX, WIFI, y las siguientes generaciones de televisión vía satélite. ACM brinda diferentes opciones de modulación así como tasas de codificación que pueden variar en tiempo real dependiendo del estado del enlace satelital. Toma como referencia el SNR (Signal to Noise Ratio) o el  $E_s/N_0$  (Energy per Symbol to Spectral Noise Density Ratio) requerido para obtener un valor mínimo de BER (Bit Error Rate), el número de bits recibidos de forma incorrecta respecto al total de bits enviados durante un intervalo especificado de tiempo, en un enlace satelital establecido con Tierra. Mientras mejor sea el SNR mayor será el orden de la modulación y menor la tasa de codificación lo cual se traduce en mayor velocidad de transmisión de información, en caso contrario se utilizará una modulación y tasa de codificación más robusta para garantizar el valor mínimo de BER y con ello la transmisión de la información de manera correcta y sin errores [1\_25]. La técnica ACM es utilizada con diferentes esquemas de modulación: M-QAM, M-APSK, M-PSK, y con diferentes técnicas de codificación: Trellis, Reed Solomon, BCH (Bose–Chaudhuri–Hocquenghem), LDPC, todo esto dependiendo de la aplicación a implementar. En la figura 1.5 se muestra la relación entre los valores de BER y  $E_s/N_0$  teórico para diferentes modulaciones empleadas en los esquemas ACM.

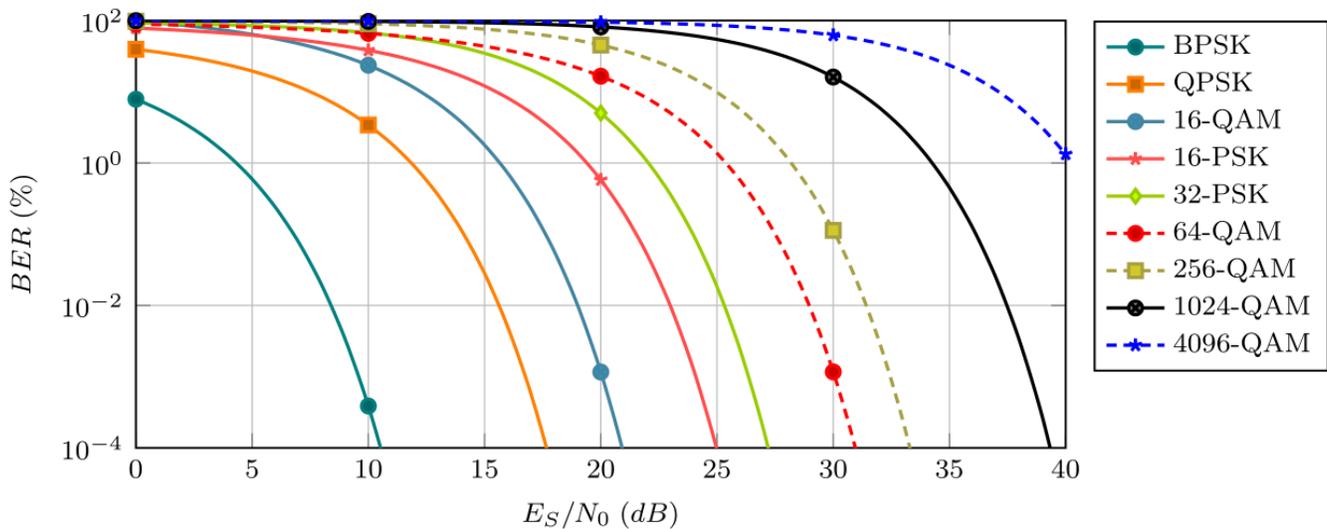


Figura 1.5.  $E_s/N_0$  [dB] vs BER para diferentes modulaciones comúnmente utilizadas en ACM [1\_25].



En este trabajo de investigación se hace un especial énfasis en el estándar DVB-S2 dado que es el que ha sido adoptado por diferentes proyectos nanosatelitales de observación terrestre en la última década y porque los elementos COTS tales como los receptores satelitales tienen una gran difusión en el mercado. De manera muy particular, el enfoque en el estándar DVB-S2X se omite debido a que las modulaciones implementadas 64, 128 y 256 APSK requieren el uso de terminales terrestres mucho más complejas y potencias de transmisión elevadas, lo cual no empata con los requerimientos optimizados de los proyectos nanosatelitales.

En los siguientes subcapítulos describimos de manera general las características principales y funcionamiento del estándar DVB-S2.

### 1.3.1 Características principales de DVB-S2

DVB-S2 es la segunda generación de especificaciones para aplicaciones satelitales de banda ancha, desarrollada por el grupo DVB (Digital Video Broadcasting) en el año 2003 y actualmente coordinado por ETSI (European Telecommunications Standard Institute) bajo el estándar EN 302 307-1 [1\_26].

El sistema está estructurado de forma tal, que es un conjunto de herramientas para permitir el uso satelital de las siguientes aplicaciones:

- Difusión de televisión con sonido.
- Interactividad (acceso a Internet).
- Servicios profesionales, enlaces de televisión y recopilación digital de noticias por satélite.
- Transmisión de imágenes y datos.

Todo esto se ha determinado basándose en los siguientes pilares fundamentales:

- Mejor rendimiento de la transmisión al acercarse al límite de Shannon.
- Múltiples tramas (streams) de entrada (MPEG-2, IP)
- Diversos formatos: MPEG-2 para SDTV y HDTV, AVC/H.264, VC9.
- Cada stream puede protegerse de forma distinta con modulaciones y codificaciones variables (ModCod).
- Complejidad razonable en el receptor.
- Compatibilidad con DVB-S (modulaciones jerárquicas).

Esquemas de transmisión respecto a DVB-S:

- Existen nuevos esquemas de codificación de canal: LDPC + BCH
- Se anexan nuevas modulaciones: QPSK, 8PSK, 16APSK, 32APSK
- Se encuentra a 0.7-1 dB del límite de Shannon.
- Factores de roll-off de 0.2, 0.25 y 0.35

Modos:

- **CCM** (Constant Coding And Modulation): Codificación y modulación constante, pudiendo ser simple o multistream, todas las tramas usan los mismos parámetros de transmisión desde que inicia hasta que termina el enlace. Ver fig. 1.6.
- **VCM** (Variable Coding And Modulation): Transmisiones y servicios que se transmiten con ModCod diferentes hacia las diversas estaciones terrenas a las que se da servicio. Ver fig. 1.7.
- **ACM** (Adaptive Coding And Modulation): Cada trama posee sus propios parámetros de funcionamiento, los mismos que van variando según las condiciones del estado de recepción de cada terminal. Generalmente variará por la condición climática. Para que ACM tenga un funcionamiento eficaz debe usar un canal de retorno.

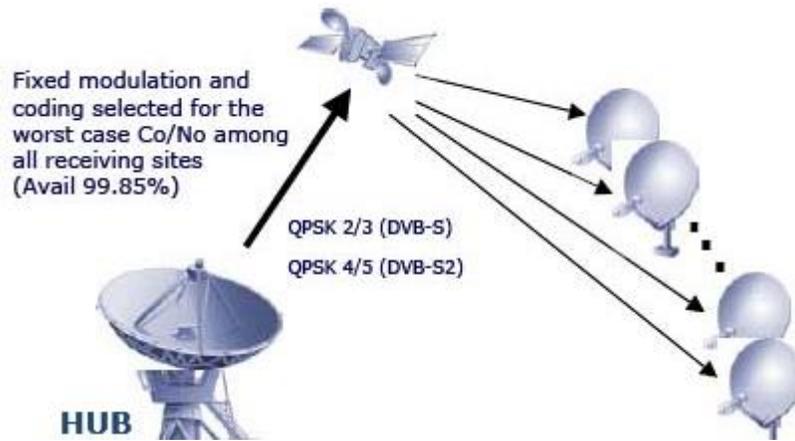


Figura 1.6. Modo de operación CCM para DVB-S (QPSK 2/3) y para DVB-S2 (QPSK 4/5) para radiodifusión [1\_27].

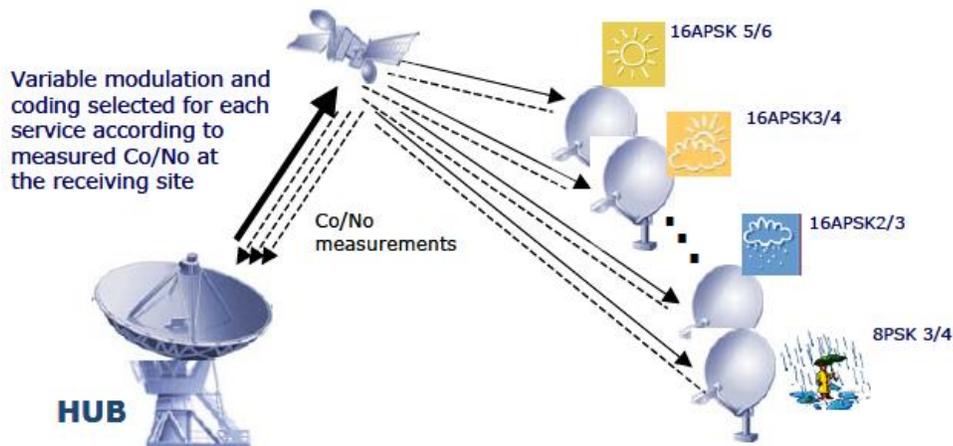


Figura 1.7. Modo de operación VCM para radiodiffusion [1\_27].

En la tabla 1.3 se puede apreciar la eficiencia espectral para cada ModCod (tipo de modulación y tipo de codificación), además del umbral  $E_s/N_0$  respectivo para cada uno de ellos. Estos valores son dados por la ETSI para un Packet Error Rate de  $10^{-7}$ .

Tabla 1.3. Parámetros típicos para eficiencia espectral y Es/No para cada tipo de FEC y modulación para DVB-S2 [1\_26].

MODCOD	Eficiencia Espectral [Bits/Sec/Hz]	Ideal Es/No (dB) FECFRAME Length = 64800	MODCOD	Eficiencia Espectral [Bits/Sec/Hz]	Ideal Es/No (dB) FECFRAME Length = 64800
QPSK 1/4	0,490243	-2,35	8PSK 5/6	2,478562	9,35
QPSK 1/3	0,656448	-1,24	8PSK 8/9	2,646012	10,69
QPSK 2/5	0,789412	-0,3	8PSK 9/10	2,679207	10,98
QPSK 1/2	0,988858	1	16APSK 2/3	2,637201	8,97
QPSK 3/5	1,188304	2,23	16APSK 3/4	2,966728	10,21
QPSK 2/3	1,322253	3,1	16APSK 4/5	3,165623	11,03
QPSK 3/4	1,487473	4,03	16APSK 5/6	3,300184	11,61
QPSK 4/5	1,587196	4,68	16APSK 8/9	3,523143	12,89
QPSK 5/6	1,654663	5,18	16APSK 9/10	3,567342	13,13
QPSK 8/9	1,766451	6,2	32APSK 3/4	3,703295	12,73
QPSK 9/10	1,788612	6,42	32APSK 4/5	3,951571	13,64
8PSK 3/5	1,779991	5,5	32APSK 5/6	4,11954	14,28
8PSK 2/3	1,980636	6,62	32APSK 8/9	4,397854	15,69
8PSK 3/4	2,228124	7,91	32APSK 9/10	4,453027	16,05

### 1.3.2 Diagrama de Bloques DVB-S2

Como se ha mencionado anteriormente, el estándar DVB-S2 es coordinado por la ETSI con apoyo de una serie de requerimientos y normas enlistadas en el documento ETSI EN 307-1 V1.4.1. “Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 1: DVB-S2.” En este se especifican las funciones y características principales de cada uno de los bloques que conforman el estándar. Dichos bloques se muestran en el diagrama de bloques general de la fig. 1.8.

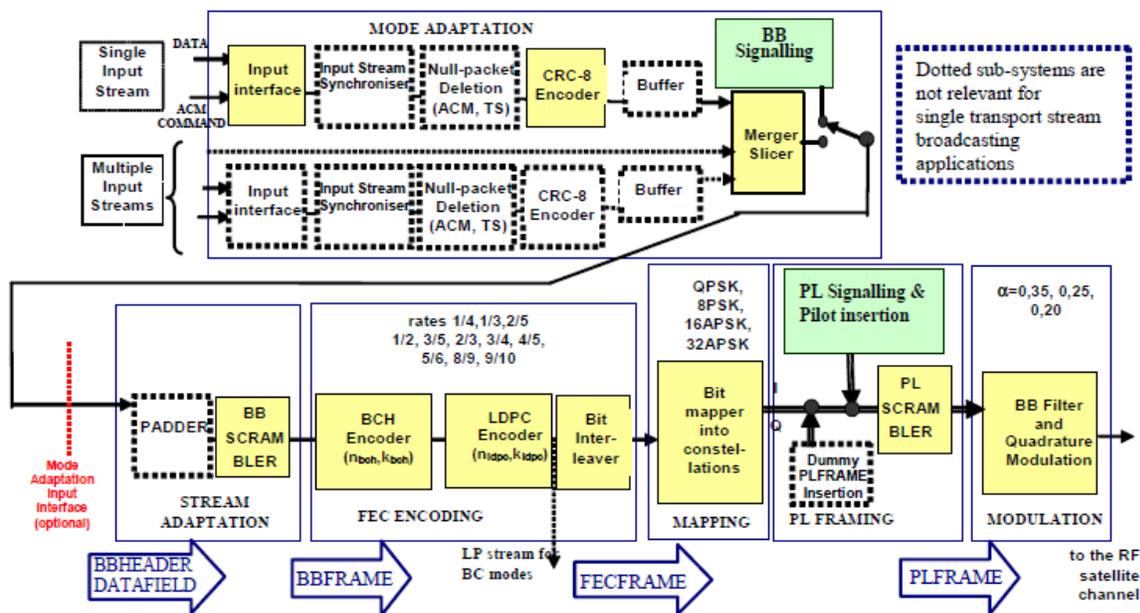


Figura 1.8. Diagrama de bloques funcional del estándar DVB-S2 [1\_26].



Como se puede apreciar el estándar consta de más de 17 bloques embebidos en 6 subsistemas diferentes:

- 1) Mode adaptation
- 2) Stream Adaptation
- 3) FEC Encoding
- 4) Mapping
- 5) PL Framing
- 6) Modulation

A continuación se describe el funcionamiento general de cada subsistema y se hace una breve descripción de cada bloque funcional del estándar con base en lo descrito en el documento de ETSI antes mencionado.

### 1.3.2.1 MODE ADAPTATION

Este subsistema está encargado de los bloques Input Interfacing, Input Stream Synchronization, Null-packet deletion, CRC-8 encoding para la detección de errores, Input stream Merging e Input stream Slicing donde se generan los frames denominados DATA FIELDS. Finalmente, la señalización para el procesamiento en banda base es añadido para especificar al receptor el Modo de Adaptación seleccionado.

Podemos entender como Input Stream a la información entrante en banda base que es entregada por las fuentes de información que anteceden al estándar. Estos flujos de información pueden ser datos, audio, imagen o video.

Básicamente en este subsistema se origina un encabezado llamado BBHEADER donde se especifica el tipo de información contenida en el paquete, el modo de adaptación (CCM, VCM o ACM) adoptado, así como información de sincronización. El encabezado junto con el bloque de información es posteriormente llevado a los bloques de Scrambling, Encoding y Mapping.

En la fig. 1.9 se puede apreciar la composición del BBHEADER y el DATA FIELD originado en esta etapa del estándar.

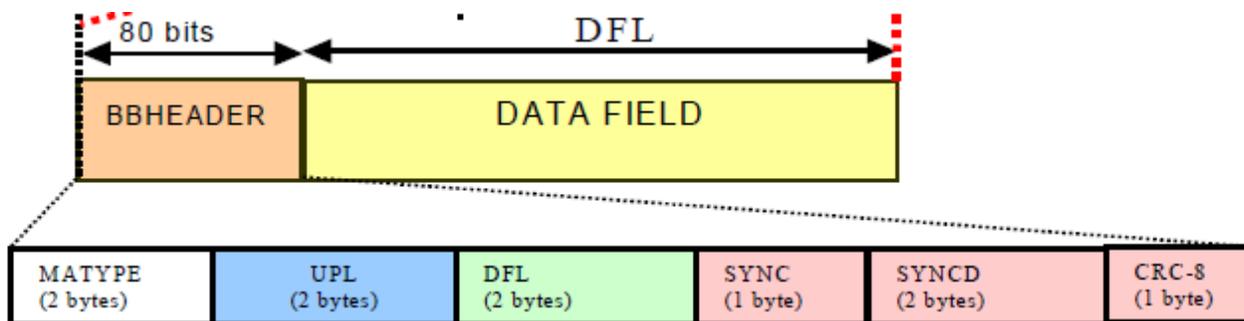


Figura 1.9. Frame generado a la salida del subsistema Mode Adaptation de DVB-S2 [1\_26].



- **Input Interface**

El bloque Input Interface mapea las señales eléctricas a su entrada en señales digitales. El primer bit recibido es el Bit Más Significativo (Most Significant Bit, MSB). Se puede también denominar como bloque de Interfaz de Entrada y tiene la tarea de recibir los flujos de información mostrados en la Tabla 1.4 para su posterior procesamiento o aplicación.

**Tabla 1.4. Valores tomados por INPUT INTERFACE [1\_26].**

Tipo de Información	Procedencia	Características	Función
<b>Transport Stream</b>	Multiplexor MPEG	1 paquete MPEG = UPL = 188x8bits UPL: User Packet Length	Información a transmitir en formato MPEG
<b>Generic Stream</b>	Fuente de información genérica	UPL = 0D	Información a transmitir proveniente de fuente de información genérica (video, audio, datos).
<b>Comando ACM</b>	Unidad de Control	MODCOD (5 bits). TYPE (2 bits). CVALID (Command Valid). SEND (deliver Data).	Específica los parámetros de modulación que deberá adoptar este paquete de información.

El Stream de Transporte deberá estar conformado por User Packets (UP) cada uno con una longitud constante UPL de 188 x 8 bits correspondientes a un paquete MPEG. El primer byte es un byte de sincronía con valor 47<sub>HEX</sub>.

El Stream Genérico consta de un flujo de información constante separado en bloques de hasta 64 kbits. En este caso el valor de UPL es de 0<sub>D</sub> lo cual significa que se trata de un stream continuo.

El Comando de ACM es una información de señalización proveniente de una unidad de control externa. Este comando contiene la información del tipo de Modulación y Codificación a ser empleado en la transmisión del siguiente paquete, así como su tipo (corto o largo).

El Comando de ACM se compone de la siguiente información:

- MODCOD (5 bits).
- TYPE (2 bits).
- CVALID (Command Valid).
- SEND (deliver Data).



MODCOD corresponde a un valor de 5 bits, el cual especifica la tasa de codificación  $C = [1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 4/5, 5/6, 8/9, 9/10]$  y la modulación de acuerdo con su eficiencia espectral  $MOD = [2, 3, 4, 5]$ .

En la Tabla 1.5. Se muestra la relación de los MODCOD a emplear y el valor de 5 bits que le corresponde a cada configuración.

**Tabla 1.5. Valores de MODCOD correspondientes al Comando ACM [1\_26].**

Mode	MOD COD	Mode	MOD COD	Mode	MOD COD	Mode	MOD COD
QPSK 1/4	1 <sub>D</sub>	QPSK 5/6	9 <sub>D</sub>	8PSK 9/10	17 <sub>D</sub>	32APSK 4/5	25 <sub>D</sub>
QPSK 1/3	2 <sub>D</sub>	QPSK 8/9	10 <sub>D</sub>	16APSK 2/3	18 <sub>D</sub>	32APSK 5/6	26 <sub>D</sub>
QPSK 2/5	3 <sub>D</sub>	QPSK 9/10	11 <sub>D</sub>	16APSK 3/4	19 <sub>D</sub>	32APSK 8/9	27 <sub>D</sub>
QPSK 1/2	4 <sub>D</sub>	8PSK 3/5	12 <sub>D</sub>	16APSK 4/5	20 <sub>D</sub>	32APSK 9/10	28 <sub>D</sub>
QPSK 3/5	5 <sub>D</sub>	8PSK 2/3	13 <sub>D</sub>	16APSK 5/6	21 <sub>D</sub>	Reserved	29 <sub>D</sub>
QPSK 2/3	6 <sub>D</sub>	8PSK 3/4	14 <sub>D</sub>	16APSK 8/9	22 <sub>D</sub>	Reserved	30 <sub>D</sub>
QPSK 3/4	7 <sub>D</sub>	8PSK 5/6	15 <sub>D</sub>	16APSK 9/10	23 <sub>D</sub>	Reserved	31 <sub>D</sub>
QPSK 4/5	8 <sub>D</sub>	8PSK 8/9	16 <sub>D</sub>	32APSK 3/4	24 <sub>D</sub>	DUMMY PLFRAME	0 <sub>D</sub>

El MSB del campo TYPE identifica el tamaño que tomará del FECFRAME durante la etapa de Codificación. Las opciones posibles son 0 = normal: 64 800 bits; 1 = short: 16 200 bits. El bit menos significativo establece si se incluirán o no señales piloto en el paquete de información (0 = no pilots, 1 = pilots).

Los campos CVALID y SEND pueden tomar los valores de activo o inactivo. Cuando ambos se encuentran etiquetados como activo se da la instrucción de que el paquete de información ya debe de ser enviado y transmitido. En caso de que los bits de información no sean los suficientes se deberá de recurrir al padding para insertar bits de valor 0 al paquete de datos y así llenar el paquete a transmitir.

- **Input Stream Synchronizer**

El procesamiento de datos en el modulador DVB-S2 puede producir un retardo de transmisión variable en la información del usuario. El Input Stream Synchronizer es capaz de garantizar una velocidad de bits constante (Constant-Bit-Rate, CBR) y un retardo de transmisión constante para paquetes generados en el bloque previo de Input Stream.

- **Null packet deletion**

Cuando se da una entrada de datos del tipo Transport Stream, en este bloque se identificarán los paquetes nulos MPEG (PID = 8191D) y serán removidos. Esto permite reducir la tasa de información y aumentar la protección contra errores en el modulador. El proceso es llevado a cabo de manera que los paquetes nulos eliminados se puedan volver a insertar en el receptor en el lugar exacto donde estaban originalmente.

- **CRC-8 encoder**

Si UPL = 0D (flujo genérico continuo), este subsistema dará paso al flujo de entrada sin modificaciones.



Si  $UPL \neq 0D$ , entonces el flujo de entrada es una secuencia de paquetes de longitud UPL, precedidos por un byte de sincronización.

La parte útil de la UP (excluyendo el byte de sincronización) se procesará mediante un codificador CRC de 8 bits sistemático. El polinomio generador será:

$$g(X) = (X^5+X^4+X^3+X^2+1)(X^2+X+1)(X+1) = X^8+X^7+X^6+X^4+X^2+1$$

La figura 1.10 muestra una posible implementación del generador CRC mediante un registro de desplazamiento. El registro se inicializará en ceros antes de que el primer bit de cada secuencia entre en el circuito. El CRC-8 calculado reemplazará el byte de sincronización del siguiente UP.

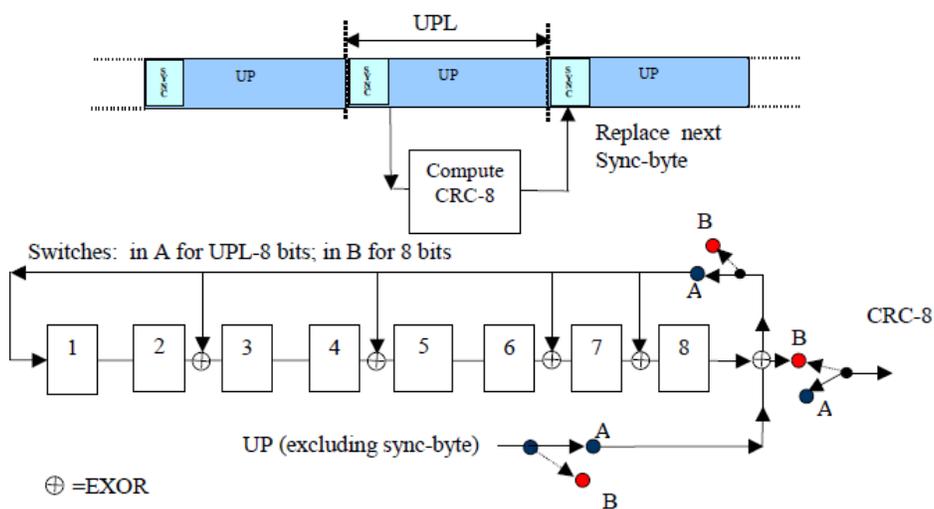


Figura 1.10. Implementación del codificador CRC-8 [1\_26].

- **Merger/Slicer**

La entrada de información al bloque Merger/Slicer (Integrador/Segmentador) se organiza como flujos continuos genéricos o como un flujo de paquetes. La longitud de un UP es de UPL bits (donde  $UPL = 0$  significa una secuencia continua). Los flujos de entrada se almacenarán en un búfer hasta que sean leídos por este bloque.

Primeramente, se realiza una segmentación, ya sea de la entrada continua o la entrada en paquetes, y se genera un frame denominado DATAFIELD, compuesto de DFL bits (DataField Length, Longitud del campo de datos), siendo el rango de valores posibles  $Kbch - (10 \times 8) \geq DFL \geq 0$ . Donde Kbch se refiere al número de bits requeridos antes de aplicar la codificación BCH. Este valor es variable y depende de la tasa de codificación seleccionada conforme a la Tabla 1.6.



Tabla 1.6. Parámetros de codificación (normal FECFRAME=64800) [1\_26].

LDPC code	BCH Uncoded Block $K_{bch}$	BCH coded block $N_{bch}$ LDPC Uncoded Block $k_{ldpc}$	BCH t-error correction	LDPC Coded Block $n_{ldpc}$
1/4	16 008	16 200	12	64 800
1/3	21 408	21 600	12	64 800
2/5	25 728	25 920	12	64 800
1/2	32 208	32 400	12	64 800
3/5	38 688	38 880	12	64 800
2/3	43 040	43 200	10	64 800
3/4	48 408	48 600	12	64 800
4/5	51 648	51 840	12	64 800
5/6	53 840	54 000	10	64 800
8/9	57 472	57 600	8	64 800
9/10	58 192	58 320	8	64 800

Del total de bits disponibles para el DATAFIELD, 80 bits están dedicados al BBHEADER.

Posteriormente, se hace un proceso de integración, en donde los diferentes datos leídos y segmentados deben ahora concatenarse para formar una salida única.

Un bloque de DATA FIELD estará compuesto por bits tomados de un solo puerto de entrada y se transmitirá en un modo de transmisión homogénea (código FEC y modulación). Después de la sustitución del byte de sincronización por CRC-8, es necesario proporcionar al receptor un método para recuperar la sincronización de los paquetes UP (cuando el receptor ya está sincronizado con el DATA FIELD). Por lo tanto, el número de bits desde el inicio del DATA FIELD y el comienzo de la primera UP completa (primer bit del CRC-8) serán detectados y almacenados en el campo SYNC D (es decir, SYNC DISTANCE) del encabezado BBHEADER. Por ejemplo, SYNC D = 0D significa que el primer USER PACKET está alineado con el DATA FIELD.

La figura 1.11 muestra los flujos de entrada continuos y por paquetes, así como los respectivos elementos que conforman el BBHEADER y el DATAFIELD.

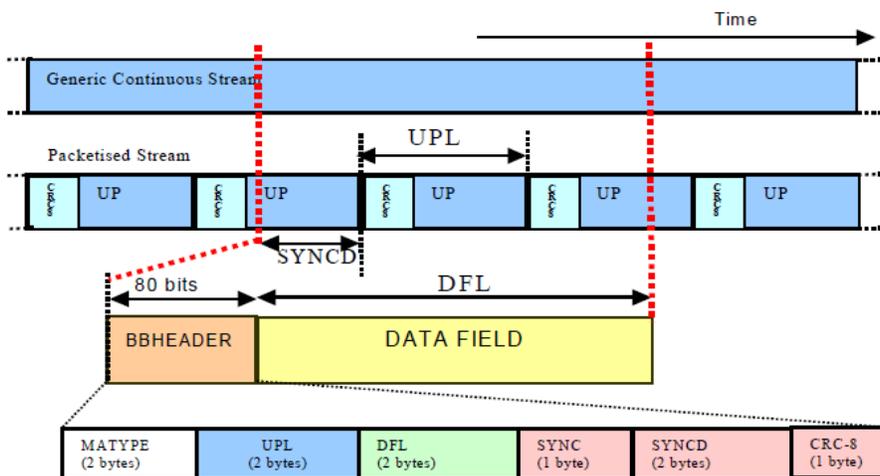


Figura 1.11. BBHEADER y DATA FIELD a la salida del subsistema MODE ADAPTATION [1\_26].



**Inserción del encabezado (BBHEADER)**

En este bloque se genera un encabezado de banda base (BBHEADER) de una longitud fija de 10 bytes. El mismo es insertado delante del DATA FIELD. Su función es especificar los parámetros de codificación y modulación de la trama, y ayudar a la sincronización en el receptor. La pérdida de eficiencia máxima introducida por el BBHEADER es de 0,25% para nldpc = 64 800 y 1% para nldpc = 16 200 asumiendo una tasa de código interno 1/2.

A continuación se describe cada uno de los elementos que conforman el encabezado BBHEADER

**MATYPE**

El campo MATYPE se conforma de 2 bytes y describe el formato de los flujos de entrada, el tipo de MODE ADAPTATION y el factor de Roll-off de la transmisión.

**Primer byte (MATYPE-1) (Ver Tabla 1.7)**

- Campo TS / GS (2 bits): entrada de flujo de transporte o entrada de flujo genérica (empaquetada o continua).
- Campo SIS / MIS (1 bit): Secuencia de entrada única o Secuencia de entrada múltiple.
- Campo CCM / ACM (1 bit): codificación y modulación constantes o codificación y modulación adaptativas (VCM es señalado como ACM).
- ISSYI (Input Stream Synchronization Indicator) (1 bit) : si ISSYI = 1 = activo, el campo ISSY se inserta después de UPs.
- NPD (1 bit): eliminación de paquete nulo activa / no activa.
- RO (2 bits): factor de Roll-off de la transmisión ( $\alpha$ ).

**Tabla 1.7. Valores de MATYPE [1\_26].**

TS/GS	SIS/MIS	CCM/ACM	ISSYI	NPD	RO
11 = Transport 00 = Generic Packetized 01 = Generic continuous 10 = reserved	1 = single 0 = multiple	1 = CCM 0 = ACM	1 = active 0 = not-active	1 = active 0 = not-active	00 = 0,35 01 = 0,25 10 = 0,20 11 = reserved

**Segundo byte (MATYPE-2):**

- Si SIS / MIS = Secuencia de entrada múltiple entonces el segundo byte = Identificador de secuencia de entrada (ISI).

**UPL (2 bytes)**

Longitud del paquete de usuario en bits, en el rango de 0 a 65,535.

EJEMPLO 1: 0000<sub>HEX</sub> = flujo continuo.

EJEMPLO 2: 000A<sub>HEX</sub> = longitud de UP de 10 bits.

EJEMPLO 3: UPL = 188x8<sub>D</sub> para paquetes de flujo de transporte MPEG.



**DFL (2 bytes)**

Longitud del campo de datos en bits, en el rango de 0 a 58,112.

EJEMPLO 4: 000A<sub>HEX</sub> = Longitud del campo de datos de 10 bits.

**SYNC (1 byte)**

Copia del byte de sincronización del paquete de usuario UP.

- Para flujo de paquetes: copia del byte de sincronización
- Para flujos continuos:

SYNC = Entre 00 y B8 reservado para la señalización del protocolo de la capa de transporte

SYNC = entre B9 y FF usuario privado.

EJEMPLO 5: SYNC = 47<sub>HEX</sub> para paquetes de flujo de transporte MPEG.

EJEMPLO 6: SYNC = 00<sub>HEX</sub> cuando el flujo de datos es por paquetes y no existe un byte de sincronización.

**SYNCD (2 bytes)**

- Para flujo de paquetes: distancia en bits desde el comienzo del DATA FIELD y la primera UP tomando como referencia este mismo frame (primer bit del CRC-8).

SYNCD = 65535<sub>D</sub> significa que no se inicia un UP en el DATAFIELD.

- Para transmisiones continuas: SYNCD = 0000 - FFFF reservado para usos futuros.

**CRC-8 (1 byte)**

Código de detección de errores aplicado a los primeros 9 bytes de BBHEADER.

El CRC-8 se computará utilizando el circuito de codificación de la figura 1.10 (interruptor en A para 72 bits, en B para 8 bits).

**1.3.2.2 STREAM ADAPTATION**

El subsistema Stream Adaptation (Adaptación de la Secuencia) proporciona el bloque de Padding (relleno) para completar una longitud constante de Kbch bits para el BBFRAME. El valor de Kbch depende de la tasa de codificación del FEC. El padding se puede aplicar en circunstancias en que los datos de usuario disponibles para la transmisión no sean suficientes para completar un BBFRAME, o cuando se deba asignar un número entero de UP en un BBFRAME.

Posterior al Padding se aplica el bloque de Scrambling (aleatorización) para reducir errores en ráfaga durante la transmisión.

El flujo de entrada de este subsistema es un BBHEADER seguido de un DATA FIELD mientras que la secuencia de salida será un BBFRAME como se muestra en la figura 1.12.

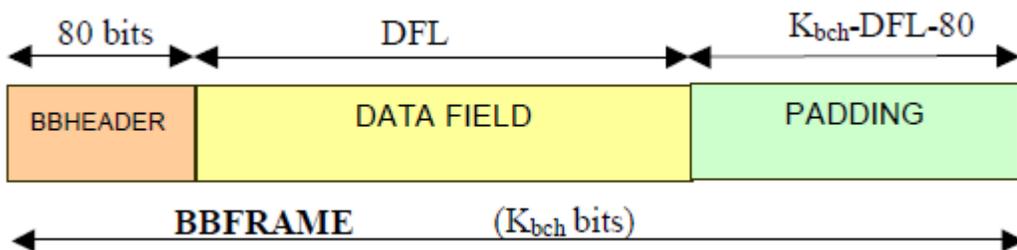


Figura 1.12. BBFRAME a la salida de Stream Adaptation [1\_26].

- **Padding**

En este bloque se añaden  $K_{bch}-DFL-80$  bits de valor cero después del DATA FIELD. El resultado es un bloque denominado BBFRAME el cual deberá de tener una longitud constante de  $K_{bch}$  bits.

- **Scrambling**

El siguiente bloque es el Scrambling en donde el BBFRAME completo es aleatorizado. La secuencia de aleatorización será síncrona con el BBFRAME, comenzando desde el MSB y terminando después de los bits de  $K_{bch}$ . Esta secuencia se genera mediante el registro de desplazamiento de la figura 1.13. El polinomio para el generador de secuencia binaria pseudoaleatoria (PRBS) es:

$$1 + X^{14} + X^{15}$$

La secuencia de inicialización en el registro PRBS será 10010101000000 y se considerará al comienzo de cada BBFRAME.

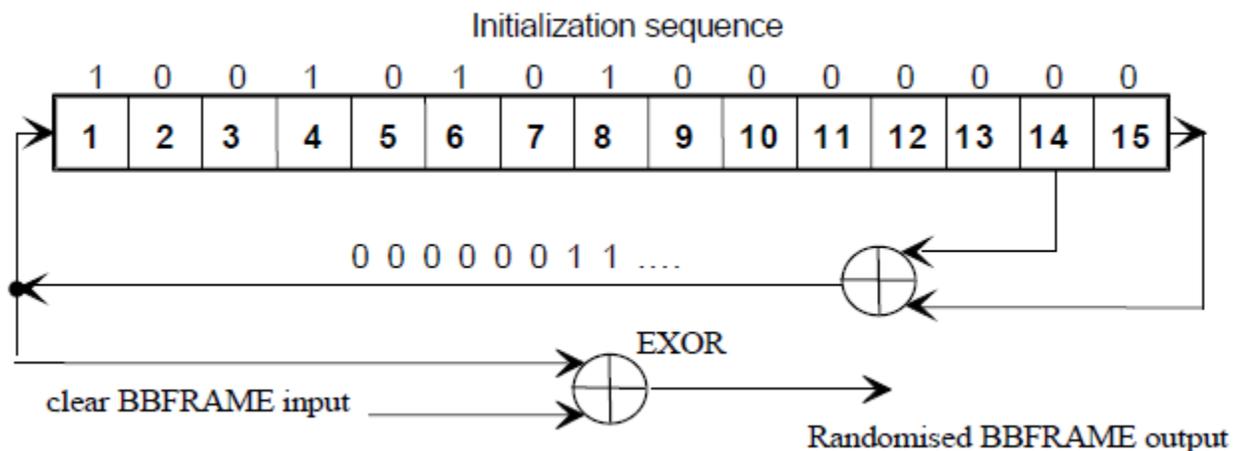


Figura 1.13. Secuenciador PRBS para aplicar la etapa de scrambling al BBFRAME [1\_26].



1.3.2.3 FEC ENCODING

Este subsistema realizará codificación externa (BCH), codificación interna (LDPC) e intercalado de bits al BBFRAME proveniente de la etapa de adaptación. Las tasas de codificación soportadas son 1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 4/5, 5/6, 8/9 y 9/10. El flujo de entrada estará compuesto por un BBFRAME y el flujo de salida se denomina FECFRAME. Cada BBFRAME, de longitud  $K_{bch}$  bits, será procesado por el subsistema de codificación FEC, para generar un FECFRAME de  $n_{ldpc}$  bits. Los bits de verificación de paridad (BCHFEC) del código externo de BCH se agregarán después del BBFRAME, y los bits de verificación de paridad (LDPCFEC) del codificador LDPC interno se agregarán después del campo BCHFEC, como se muestra en la figura 1.14. Dependiendo de la aplicación los bloques a la salida del FEC podrán tener una longitud  $n_{ldpc}$  de 64,800 bits o 16,200 bits.

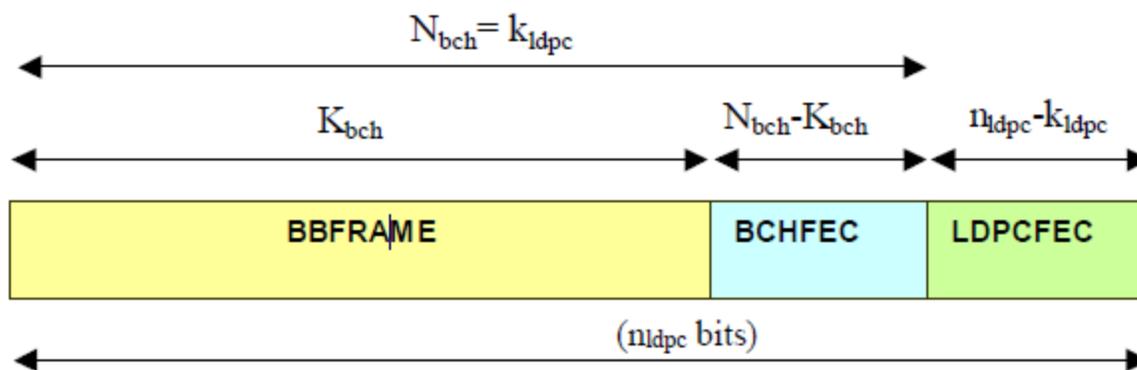


Figura 1.14. Formato del FECFRAME antes del proceso de Interleaving (Normal  $n_{ldpc}=64,800$  bits; Short  $n_{ldpc}= 16,200$  bits) [1\_26].

La Tabla 1.8 establece las longitudes de los bloques de codificación para un FECFRAME normal ( $n_{ldpc} = 64\ 800$  bits) mientras que la tabla 1.9 para un FECFRAME corto ( $n_{ldpc} = 16\ 200$  bits).

Tabla 1.8. Parámetros de codificación (FECFRAME Normal;  $n_{ldpc}=64,800$  bits) [1\_26].

LDPC code	BCH Uncoded Block $K_{bch}$	BCH coded block $N_{bch}$ LDPC Uncoded Block $k_{ldpc}$	BCH t-error correction	LDPC Coded Block $n_{ldpc}$
1/4	16 008	16 200	12	64 800
1/3	21 408	21 600	12	64 800
2/5	25 728	25 920	12	64 800
1/2	32 208	32 400	12	64 800
3/5	38 688	38 880	12	64 800
2/3	43 040	43 200	10	64 800
3/4	48 408	48 600	12	64 800
4/5	51 648	51 840	12	64 800
5/6	53 840	54 000	10	64 800
8/9	57 472	57 600	8	64 800
9/10	58 192	58 320	8	64 800



Tabla 1.9. Parámetros de codificación (FECFRAME corto; nldpc=16,200 bits) [1\_26].

LDPC Code identifier	BCH Uncoded Block $K_{bch}$	BCH coded block $N_{bch}$ LDPC Uncoded Block $k_{ldpc}$	BCH t-error correction	Effective LDPC Rate $k_{ldpc}/16\ 200$	LDPC Coded Block $n_{ldpc}$
1/4	3 072	3 240	12	1/5	16 200
1/3	5 232	5 400	12	1/3	16 200
2/5	6 312	6 480	12	2/5	16 200
1/2	7 032	7 200	12	4/9	16 200
3/5	9 552	9 720	12	3/5	16 200
2/3	10 632	10 800	12	2/3	16 200
3/4	11 712	11 880	12	11/15	16 200
4/5	12 432	12 600	12	7/9	16 200
5/6	13 152	13 320	12	37/45	16 200
8/9	14 232	14 400	12	8/9	16 200
9/10	NA	NA	NA	NA	NA

- Bit Interleaver**

Para los formatos de modulación 8PSK, 16APSK y 32APSK, la salida del codificador LDPC será tomada y se intercalarán sus bits utilizando un bloque de Interleaving o entrelazado. El funcionamiento de este bloque consta en la escritura de los datos en serie ordenados por columnas, mientras que la lectura se realiza en serie, pero por filas (primero se lee el MSB de BBHEADER como se muestra en la figura 1.15). La configuración del Interleaving para cada formato de modulación se especifica en la Tabla 1.10.

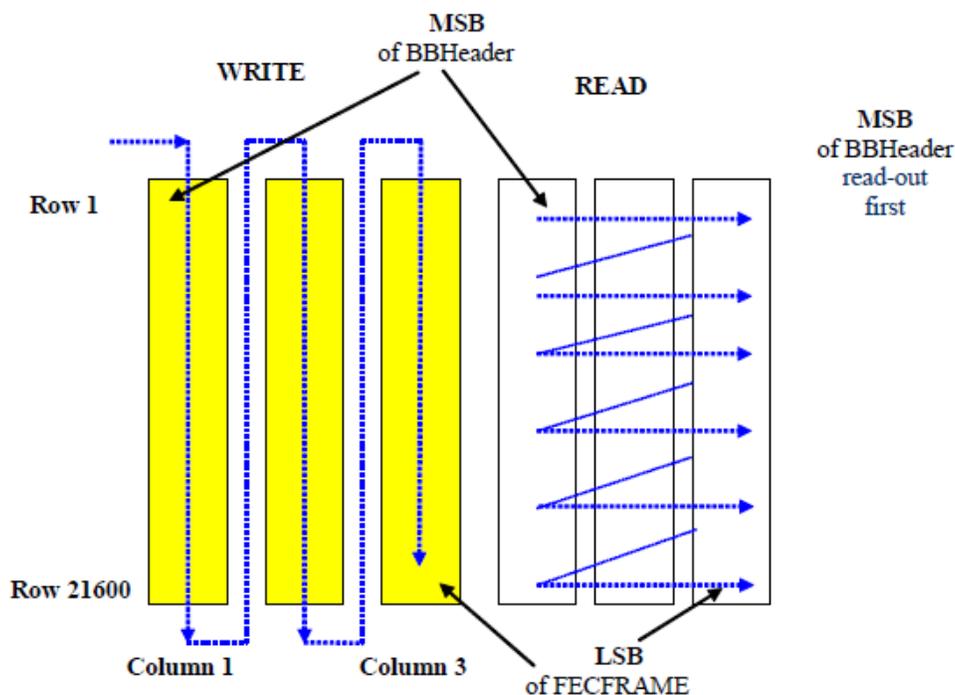


Figura 1.15. Bloque de Bit Interleaving para 8PSK. Escritura por columna, lectura por filas [1\_26].

Tabla 1.10. Estructura del Bit Interleaver [1\_26].

Modulation	Rows (for $n_{Idpc} = 64\ 800$ )	Rows (for $n_{Idpc} = 16\ 200$ )	Columns
8PSK	21 600	5 400	3
16APSK	16 200	4 050	4
32APSK	12 960	3 240	5

### 1.3.2.4 MAPPING

En este subsistema se generan los valores de símbolo asociados a las modulaciones QPSK, 8PSK, 16APSK y 32APSK.

Cada FECFRAME (64,800 bits para FECFRAME normal, o 16,200 bits para FECFRAME corto), será proyectado de manera paralela en un mapa de constelaciones con un número de bits/símbolo ( $\eta_{MOD}$ ) de 2 para QPSK, 3 para 8PSK, 4 para 16APSK y 5 para 32APSK. El MSB del FECFRAME se mapea en el MSB de la primera secuencia paralela. Cada secuencia paralela se mapeará en cada constelación, generando una secuencia I, Q de longitud variable dependiendo de la eficiencia de modulación seleccionada  $\eta_{MOD}$ .

La secuencia de entrada será un FECFRAME, y la secuencia de salida un XFECFRAME (compleX FECFRAME), compuesta por 64 símbolos de  $64,800/\eta_{MOD}$  (normal XFECFRAME) o  $16,200/\eta_{MOD}$  (short XFECFRAME). Cada símbolo de modulación debe ser un vector complejo en el formato I, Q, siendo I el componente en fase y Q el componente en cuadratura o en el formato equivalente  $\rho \exp(j\phi)$  (donde  $\rho$  es el módulo del vector y  $\phi$  su fase).

- **QPSK**

Para el mapeo en QPSK, la asignación de bits en cada símbolo se hace considerando el código Gray, es decir, los símbolos aledaños entre sí sólo difieren en un bit.

Dado que la modulación QPSK engloba 2 bits/símbolo los valores a mapear serán: 00, 10, 11, 01. La energía promedio normalizada por cada símbolo debe ser igual a  $\rho=1$ . El mapeo de bits genera la constelación QPSK mostrada en la figura 1.16.

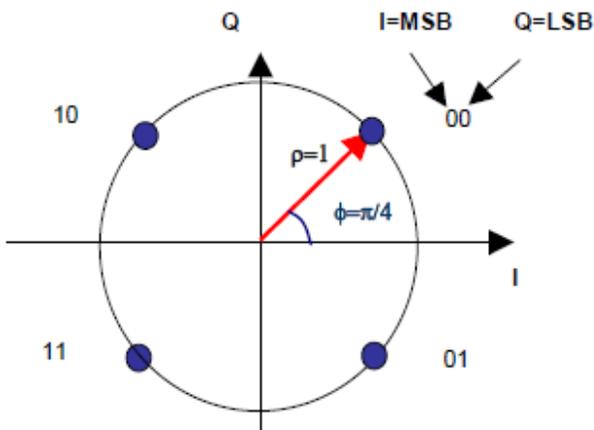


Figura 1.16. Mapeo de bits en la constelación QPSK [1\_26].

Dos bits del FECFRAME son mapeados en un símbolo QPSK. De este modo, los bits  $2n$  y  $2n+1$  determinan el  $n$ ésimo símbolo QPSK, donde  $n=0, 1, 2, \dots, (N/2)-1$  y  $N$  es el tamaño del bloque LDPC codificado, es decir,  $n_{ldpc}$ .

• **8PSK**

Al igual que QPSK, 8PSK utiliza codificación Gray y la energía promedio normalizada de cada símbolo es  $\rho=1$ . 8PSK considera 3 bits/símbolo por lo que los bits  $3n$ ,  $3n+1$  y  $3n+2$  a la salida del bloque de Interleaving determinan el  $n$ ésimo símbolo 8PSK donde  $n=0, 1, 2, \dots, (N/3)-1$  y  $N$  es el tamaño del bloque LDPC codificado. En la figura 1.17 se muestra la constelación 8PSK y el valor de cada símbolo de acuerdo al código Gray.

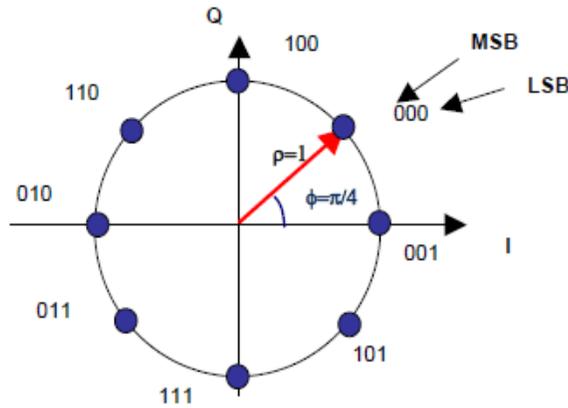


Figura 1.17. Mapeo de bits en la constelación 8PSK [1\_26].

• **16APSK**

La constelación 16APSK se compone de dos anillos concéntricos de 4 y 12 puntos PSK espaciados uniformemente como se muestra en la figura 1.18.

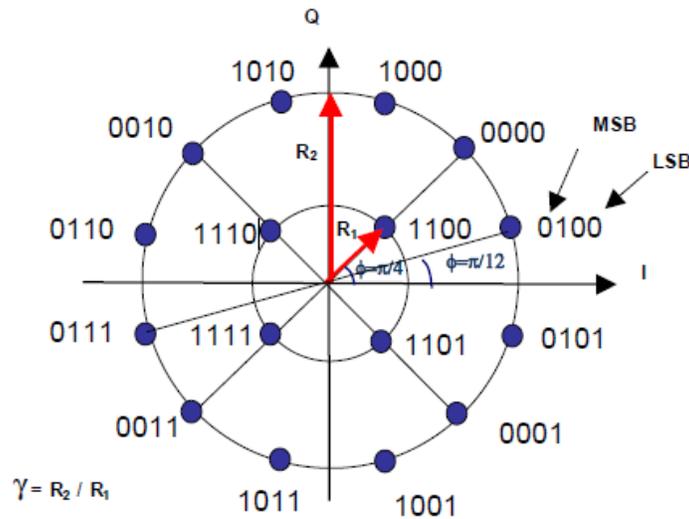


Figura 1.18. Constelación 16APSK [1\_26].



El anillo interior toma un valor de radio  $R_1$  y el anillo exterior de radio  $R_2$ , tal que la relación entre el radio del círculo externo y el radio del círculo interno ( $\gamma = R_2 / R_1$ ) debe cumplir con la tabla 1.11.

**Tabla 1.11. Valores de  $\gamma$  óptimos para 16APSK [1\_26].**

Code rate	Modulation/coding spectral efficiency	$\gamma$
2/3	2,66	3,15
3/4	2,99	2,85
4/5	3,19	2,75
5/6	3,32	2,70
8/9	3,55	2,60
9/10	3,59	2,57

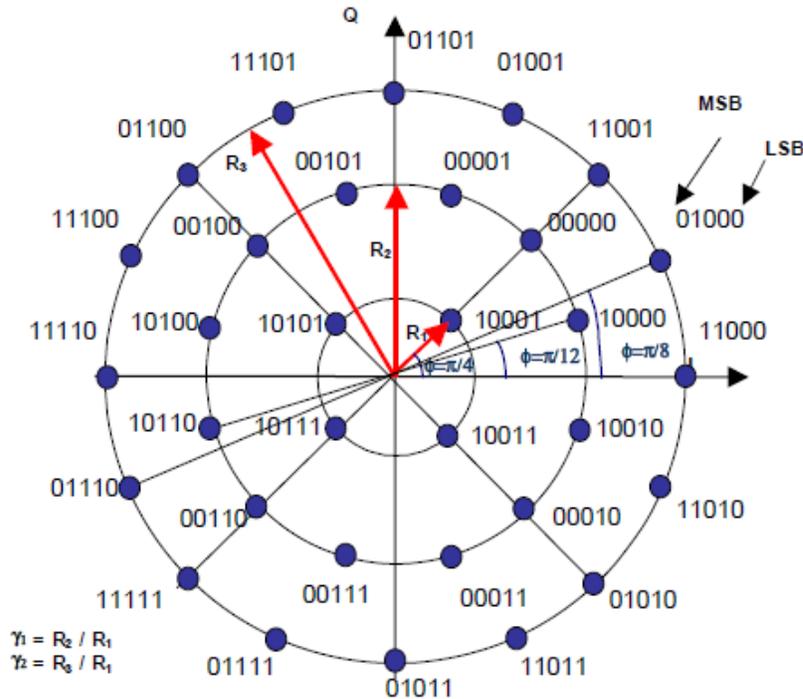
Además, se admiten únicamente dos valores para las amplitudes de la constelación, los cuales permiten la optimización del rendimiento según las características del canal.

- $E = 1$  ( $E$  = energía de símbolo promedio) correspondiente a  $[R_1]^2 + 3[R_2]^2 = 4$ .
- $R_2 = 1$ .

Los bits  $4n, 4n+1, 4n+2$  y  $4n+3$  a la salida del Interleaving determinan el  $n$ ésimo símbolo 16APSK, donde  $n = 0, 1, 2, \dots, (N/4) - 1$  y  $N$  es el tamaño del bloque LDPC codificado.

• **32APSK**

La constelación 32APSK, mostrada en la figura 1.19, se compone de tres anillos concéntricos con 4, 12 y 16 puntos separados de manera uniforme y con un radio de valor  $R_1, R_2$  y  $R_3$  respectivamente.



**Figura 1.19. Constelación 32APSK [1\_26].**



La tabla 1.12 define los valores óptimos de  $\gamma_1=R_2/R_1$  y  $\gamma_2=R_3/R_1$ .

**Tabla 1.12. Valores óptimos de  $\gamma_1$  y  $\gamma_2$  para 32APSK [1\_26].**

Code rate	Modulation/coding spectral efficiency	$\gamma_1$	$\gamma_2$
3/4	3,74	2,84	5,27
4/5	3,99	2,72	4,87
5/6	4,15	2,64	4,64
8/9	4,43	2,54	4,33
9/10	4,49	2,53	4,30

Los valores de los radios de cada anillo deben cumplir con las siguientes condiciones:

- $E=1$  ( $E$ =Energía de símbolo promedio) correspondiente a  $[R_1]^2+ 3[R_2]^2+ 4[R_3]^2 = 8$ .
- $R_3=1$ .

Los bits  $5n, 5n+1, 5n+2, 5n+3$  y  $5n+4$  a la salida del interleaver determinan el enésimo símbolo 32APSK, donde  $n= 0, 1, 2, (N/5)-1$ , y  $N$  es el valor del tamaño del bloque LDPC codificado.

### 1.3.2.5 PHYSICAL LAYER FRAMING

El subsistema PLFraming recibe una trama XFECFRAME a su entrada y genera una trama de capa física (denominado PLFRAME) a su salida mediante la realización de los siguientes procesos:

- Generación de un PLFRAME Dummy cuando ningún XFECFRAME está listo para ser procesado y transmitido.
- La división del XFECFRAME en un número entero  $S$  de slots de longitud constante  $M$  ( $M=90$  símbolos/slot). El valor de  $S$  es asignado de acuerdo a la tabla 1.13.

**Tabla 1.13 Valores dados para el XFECFRAME [1\_26].**

$\eta_{MOD}$ (bit/s/Hz)	$n_{ldpc} = 64\ 800$ (normal frame)		$n_{ldpc} = 16\ 200$ (short frame)	
	$S$	$\eta$ % no-pilot	$S$	$\eta$ % no-pilot
2	360	99,72	90	98,90
3	240	99,59	60	98,36
4	180	99,45	45	97,83
5	144	99,31	36	97,30

- Genera y adiciona un encabezado llamado PLHEADER antes del XFECFRAME para la configuración del receptor. El PLHEADER ocupa exactamente un slot ( $M = 90$  Símbolos).
- Inserción de bloques piloto (para modos que así lo requieren) cada 16 slots. Estos bloques ayudan a la sincronización en el receptor. Un bloque piloto está compuesto por  $P = 36$  símbolos.
- Aleatorización de los símbolos modulados ( $I, Q$ ) mediante un aleatorizador de capa física (scrambling).

La eficiencia del subsistema está dada por  $\eta = 90S/[90(S+1)+ P \text{ int}\{(S-1)/16\}]$ , donde  $P = 36$  y  $\text{int}\{.\}$  representa la función entero la cual trunca la parte decimal del resultado obtenido.



En la figura 1.20 se muestra la composición de un PLFRAME antes de ser aleatorizado.

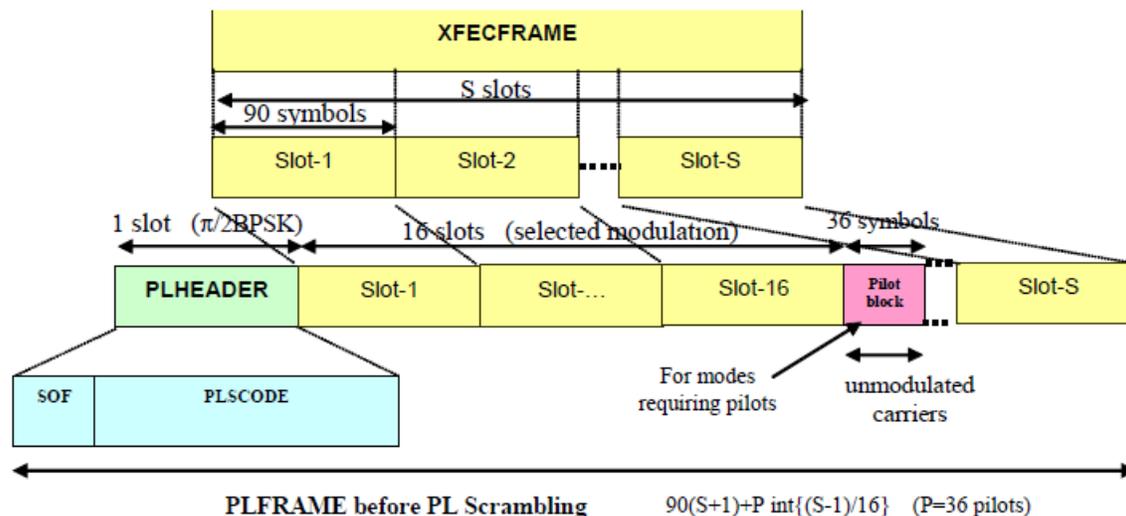


Figura 1.20. Formato del Physical Layer Frame (PLFRAME) [1\_26].

- **Inserción PLFRAME Dummy**

Un PLFRAME Dummy está compuesto de un PLHEADER y de 36 SLOTS de portadoras no moduladas.

$$(I = (1 / \sqrt{2}), Q = (1 / \sqrt{2})).$$

- **PL signaling**

En este bloque se genera el PLHEADER, un encabezado diseñado para la sincronización del receptor y la señalización de la capa física. Después de que el receptor decodifica el PLHEADER, este conoce la duración y la estructura del PLFRAME, el esquema de modulación y codificación del XFEFRAME, y la presencia o ausencia de símbolos piloto.

El PLHEADER (un slot de 90 símbolos) está compuesto de los siguientes campos:

a) Start of Frame (SOF, 26 símbolos) identificando el inicio de Frame.

MODCOD y TYPE definidos de la siguiente manera:

b) MODCOD (5 símbolos), que identifica la modulación XFEFRAME y la tasa de codificación del FEC;

c) TYPE (2 símbolos), que identifica la longitud de FECFRAME (64,800 bits o 16,200 bits) y la presencia o ausencia de pilotos.

d) Código PLS (64 símbolos): el código PLS (Physical Layer Signalling) es un código binario no sistemático de longitud 64 y dimensión 7 con distancia mínima  $d_{min} = 32$ . Es equivalente al primer orden Reed-Muller bajo permutación. Transmite 7 bits para fines de señalización de capa física. Estos 7 bits consisten de dos campos:

El PLHEADER, representado por la secuencia binaria  $(y_1, y_2, \dots, y_{90})$  se modula en 90 símbolos  $\pi/2$  BPSK de acuerdo con la regla:

$$I_{2i-1} = Q_{2i-1} = (1 / \sqrt{2}) (1 - 2y_{2i-1}), I_{2i} = -Q_{2i} = - (1 / \sqrt{2}) (1 - 2y_{2i}) \text{ para } i = 1, 2, \dots, 45$$



**a) SOF field**

El Start of Frame corresponde a la secuencia 18D2E82<sub>HEX</sub> (01-1000-....-0010, en binario), donde el MSB corresponde al valor a la izquierda de la secuencia.

**b) MODCOD field**

Al igual que el BBHEADER el MODCOD será definido por los valores de la tabla 1.14 mostrada aquí nuevamente. Este campo tiene un valor de 5 bits.

**Tabla 1.14. Etiquetas asignadas a cada ModCod en el encabezado PLHEADER [1\_26].**

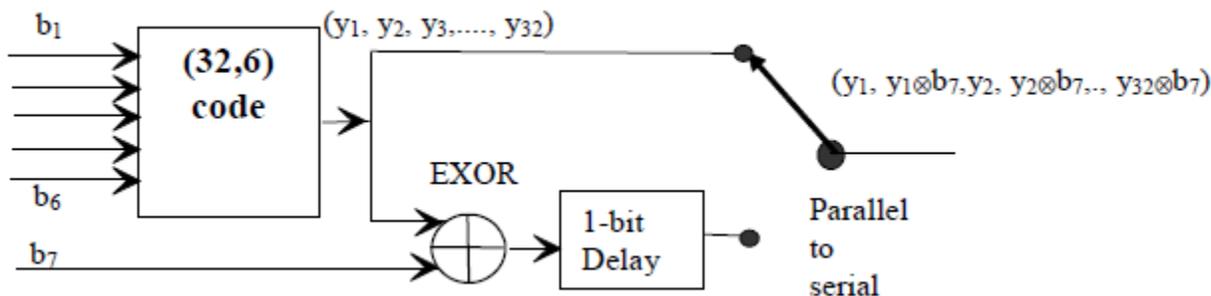
Mode	MOD COD	Mode	MOD COD	Mode	MOD COD	Mode	MOD COD
QPSK 1/4	1 <sub>D</sub>	QPSK 5/6	9 <sub>D</sub>	8PSK 9/10	17 <sub>D</sub>	32APSK 4/5	25 <sub>D</sub>
QPSK 1/3	2 <sub>D</sub>	QPSK 8/9	10 <sub>D</sub>	16APSK 2/3	18 <sub>D</sub>	32APSK 5/6	26 <sub>D</sub>
QPSK 2/5	3 <sub>D</sub>	QPSK 9/10	11 <sub>D</sub>	16APSK 3/4	19 <sub>D</sub>	32APSK 8/9	27 <sub>D</sub>
QPSK 1/2	4 <sub>D</sub>	8PSK 3/5	12 <sub>D</sub>	16APSK 4/5	20 <sub>D</sub>	32APSK 9/10	28 <sub>D</sub>
QPSK 3/5	5 <sub>D</sub>	8PSK 2/3	13 <sub>D</sub>	16APSK 5/6	21 <sub>D</sub>	Reserved	29 <sub>D</sub>
QPSK 2/3	6 <sub>D</sub>	8PSK 3/4	14 <sub>D</sub>	16APSK 8/9	22 <sub>D</sub>	Reserved	30 <sub>D</sub>
QPSK 3/4	7 <sub>D</sub>	8PSK 5/6	15 <sub>D</sub>	16APSK 9/10	23 <sub>D</sub>	Reserved	31 <sub>D</sub>
QPSK 4/5	8 <sub>D</sub>	8PSK 8/9	16 <sub>D</sub>	32APSK 3/4	24 <sub>D</sub>	DUMMY PLFRAME	0 <sub>D</sub>

**c) TYPE field**

El MSB de este elemento identifica el tamaño del FECFRAME (0 = normal: 64,800 bits; 1 = short: 16,200 bits). El LSB señala si existen o no portadoras piloto (0 = no pilots, 1 = pilots).

**d) Código PLS**

Los campos MODCODE y TYPE están codificados con un código (64,7). Dicho código se construye a partir de un código biortogonal (32,6) de acuerdo a la siguiente figura:



**Figura 1.21. Generador del código PLS [1\_26].**



El diseño de este generador garantiza que cada bit impar en el código (64,7) sea siempre igual al anterior o su contrario. El bit b7 determina cuál de las dos opciones es llevada a cabo. Este hecho se puede aprovechar en caso de que se adopte una detección coherente diferencial en el receptor.

El MODCOD y el MSB del campo TYPE se codificarán mediante un código de bloque lineal de longitud 32 con la siguiente matriz generadora.

$$G = \begin{bmatrix} 01010101010101010101010101010101 \\ 00110011001100110011001100110011 \\ 00001111000011110000111100001111 \\ 00000000111111110000000011111111 \\ 00000000000000001111111111111111 \\ 11111111111111111111111111111111 \end{bmatrix}$$

Figura 1.22 Matriz Generadora del campo MODCOD dentro del PLHEADER [1\_26].

El bit más significativo del MODCOD se multiplica con la primera fila de la matriz, el bit siguiente con la segunda fila y así sucesivamente. Los 32 bits codificados se indican como  $(y_1 y_2 \dots y_{32})$ . Cuando el bit menos significativo del campo TYPE es 0, el código PLS final generará  $(y_1 y_1 y_2 y_2 \dots y_{32} y_{32})$  como salida, es decir, se repetirá cada símbolo. Cuando el bit menos significativo del campo TYPE es 1, el código PLS final generará  $y_1 \bar{y}_1 y_2 \bar{y}_2 \dots y_{32} \bar{y}_{32}$  como salida, es decir, el símbolo repetido se complementa de forma binaria.

Finalmente, la salida de 64 bits del código PLS es aleatorizada por la secuencia binaria:

$$0111000110011101100000111100100101010011010000100010110111111010$$

• **Inserción de Pilotos**

Es posible utilizar dos configuraciones del PLFRAME: Sin pilotos y con pilotos. En este último caso, un BLOQUE PILOTO estará compuesto por  $P = 36$  símbolos. Cada símbolo es un símbolo no modulado, identificado por  $I = (1/\sqrt{2})$ ,  $Q = (1/\sqrt{2})$ .

El primer BLOQUE PILOTO se insertará 16 slots después del PLHEADER, el segundo después de 32 slots y así sucesivamente. Si la posición del BLOQUE PILOTO coincide con el comienzo del siguiente SOF, entonces no se transmite el piloto.

La presencia y ausencia de pilotos en VCM y ACM se puede cambiar de un frame a otro.

• **PL scrambler (aleatorización)**

Antes de la modulación, cada PLFRAME, excluyendo el PLHEADER, se aleatoriza para asegurar la dispersión de energía multiplicando las muestras  $(I + jQ)$  por una secuencia de aleatorización compleja  $(C_I + jC_Q)$ :



$$I_{\text{SCRAMBLED}} = [I C_I - Q C_Q]; Q_{\text{SCRAMBLED}} = (I C_Q + Q C_I)$$

La tasa de aleatorización corresponde a la tasa de símbolos I-Q del PLFRAME, por lo que este proceso no tiene impacto en el ancho de banda empleado. La secuencia de aleatorización tiene un período mayor que la duración máxima requerida de aproximadamente 70,000 símbolos.

La secuencia de aleatorización se reinicializará al final de cada PLHEADER. La duración de PLFRAME depende de la modulación seleccionada, por lo tanto, la longitud de la secuencia de aleatorización se truncará a la longitud del PLFRAME en proceso. Ver fig. 1.23.

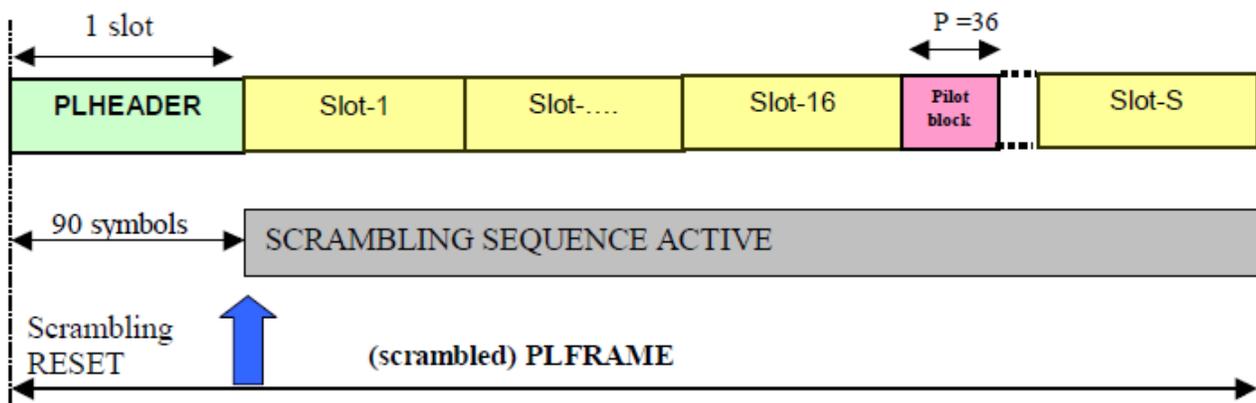


Figura 1.23. PL Scrambling [1\_26].

Las secuencias del código de aleatorización se construirán combinando dos secuencias generadas por medio de dos polinomios generadores de grado 18 en una secuencia compleja. Las secuencias resultantes constituyen segmentos de un conjunto de secuencias de oro (Gold Sequence).

Sean  $x$  y  $y$  las dos secuencias respectivamente. La secuencia  $x$  se construye utilizando el polinomio primitivo  $1 + x^7 + x^{18}$ . La secuencia  $y$  se construye utilizando el polinomio  $1 + y^5 + y^7 + y^{10} + y^{18}$ .

La figura 1.24 muestra el generador del código de aleatorización:

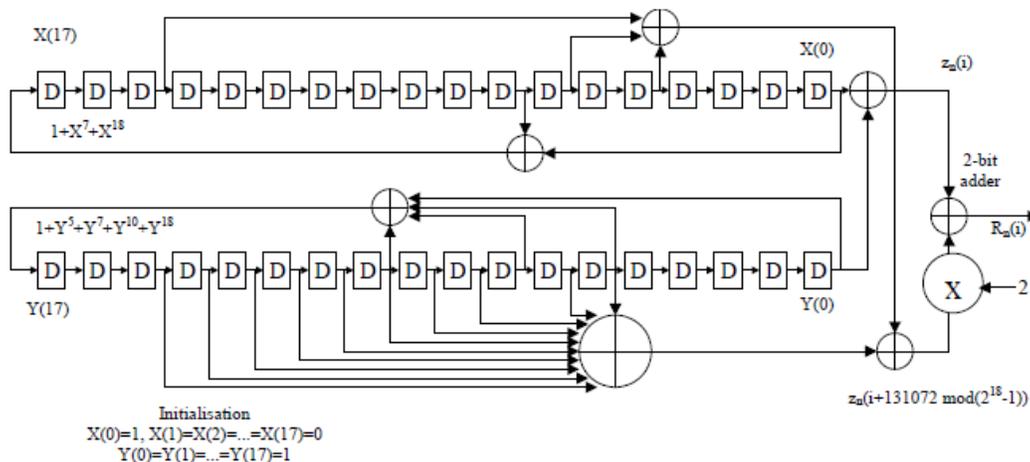


Figura 1.24. Generador del código de aleatorización (n=0) [1\_26].



### 1.3.2.6 MODULACIÓN

Después de la aleatorización, el PLFRAME es llevado a un filtro de coseno elevado (square root raised cosine filter, SRRC). El factor de caída puede tomar los valores de  $\alpha = 0,35, 0,25$  y  $0,20$ , según los requerimientos de la señal.

El SRRC tiene una función de transferencia definida por la siguiente expresión:

$$H(f) = 1; \text{ para } |f| < f_N(1 - \alpha) \quad (1.1)$$

$$H(f) = \left\{ \frac{1}{2} + \frac{1}{2} \sin \frac{\pi}{2f_N} \left[ \frac{f_N - |f|}{\alpha} \right] \right\}^{1/2}; \text{ para } f_N(1 - \alpha) \quad (1.2)$$

$$H(f) = 0; \text{ para } |f| > f_N(1 + \alpha) \quad (1.3)$$

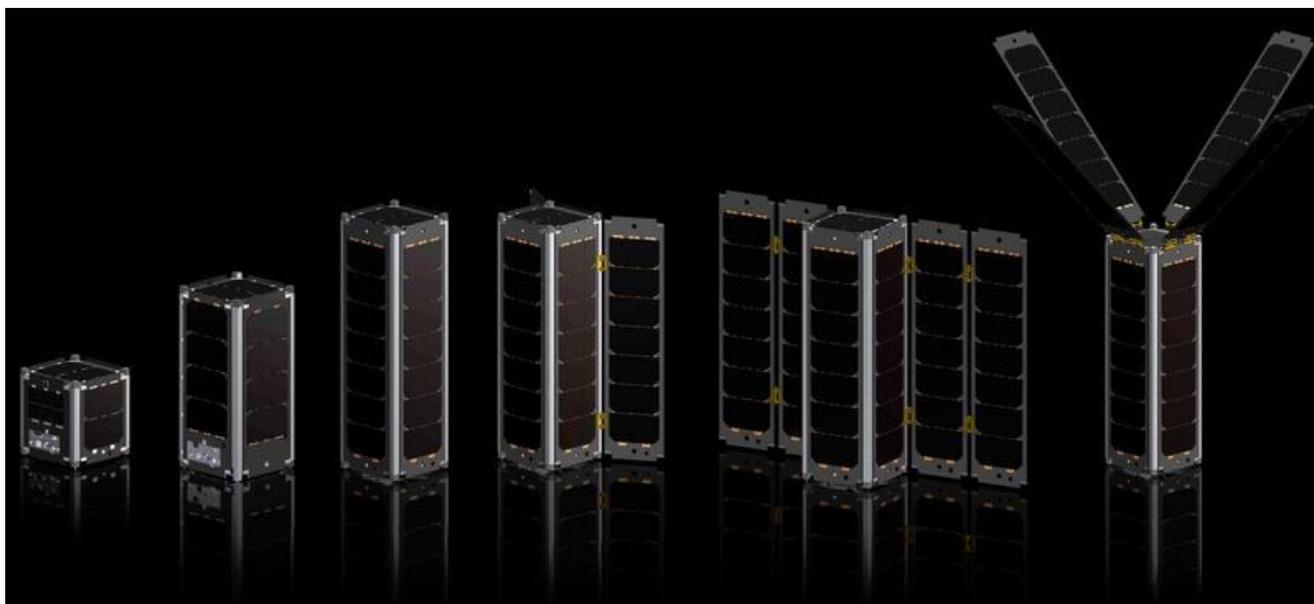
donde  $f_N = \frac{1}{2T_s} = \frac{R_s}{2}$  es la frecuencia de Nyquist y  $\alpha$  es el factor de Roll – off

Finalmente, la modulación en cuadratura se realiza multiplicando las muestras en fase y en cuadratura por  $\sin(2\pi f_0 t)$  y  $\cos(2\pi f_0 t)$ , respectivamente (donde  $f_0$  es la frecuencia portadora). Las dos señales resultantes se suman para obtener la señal de salida del modulador la cual es llevada al front end analógico y posteriormente a la antena transmisora.

## 1.4 SATÉLITES CUBESAT

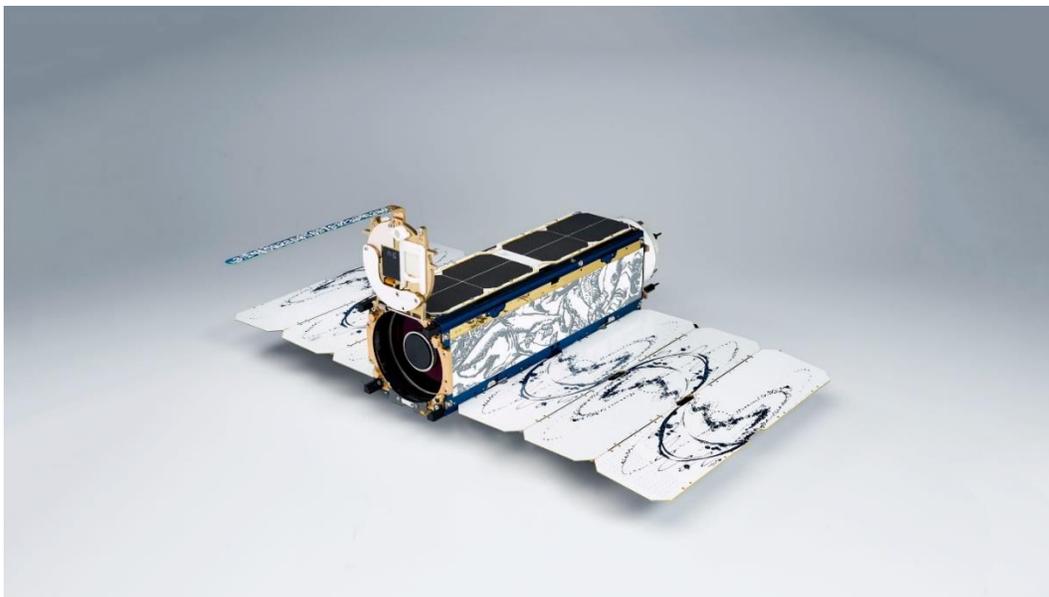
En 1999, la Universidad de Stanford junto con la Universidad de Santa Clara propusieron un estándar de nanosatélite que podía pesar hasta 1.33 Kg con la idea de hacer lanzamientos múltiples de estos, albergándolos como carga secundaria en un cohete, con lo que se lograría un precio muy accesible de estos lanzamientos; dicho estándar es conocido como CubeSat. Estos proyectos, comparados con las tradicionales misiones satelitales multimillonarias, tienen el gran potencial de brindar una experiencia educativa a sus participantes y de implementar práctica y satisfactoriamente misiones relacionadas con la ciencia, la industria, y actualmente otras áreas como el arte, a un mucho menor costo.

A partir del concepto de 1U (una unidad) equivalente a  $10 \times 10 \times 10 \text{ cm}^3$ , surgieron CubeSats de valores escalables de 2U, 3U, 6U, 12U y hasta 16U, de los cuales, el tamaño de 3U (ver fig. 1.25) ha sido el más utilizado por su balance entre capacidad y precio.



**Figura 1.25. CubeSats de 1U, 2U y 3U [1\_28].**

El costo de desarrollo de un CubeSat es, por mucho, menor que el de los grandes satélites y además la madurez de la tecnología CubeSat ha permitido que sus beneficios estén a la par de estos últimos. Por ejemplo, el satélite LandSat 8 cuya misión es la observación terrestre pesa 2780 kg y tuvo un costo de desarrollo de 850 millones de dólares [1\_29], mientras que la primera constelación de 175 CubeSats 3U de la empresa Planet Labs, también destinado a la observación terrestre, tuvo un costo menor a los 100 millones de dólares con una masa de 4 kg por satélite. LandSat 8 requiere de 16 días para cubrir todo el globo terráqueo con sus cámaras [1\_30] mientras que la constelación de Planet Labs toma una imagen de todo el planeta cada día [1\_31]. En la fig. 1.26 se muestra uno de los CubeSat de Planet Labs.



**Figura 1.26. CubeSat 3U desarrollado por Planet Labs [1\_31].**

#### **1.4.1. Proyectos CubeSat desarrollados**

Gracias a las ventajas expuestas, este tipo de proyectos son cada vez más conocidos y numerosos permitiendo que el acceso al espacio sea viable a precios bajos principalmente para países emergentes. Desde el primer lanzamiento CubeSat en 2003 hasta octubre de 2019 han habido poco más de 1100 nanosatélites de este tipo puestos en órbita abarcando un amplio espectro de aplicaciones: educación, arte, comunicaciones, astronomía, observación terrestre, desarrollo tecnológico, medicina, propulsión, aplicaciones militares, agricultura, exploración espacial, etc. [1\_32]. El número de lanzamientos ha ido incrementando año con año y se espera que en los próximos años esta tendencia continúe lo cual destaca la importancia del desarrollo de este tipo de tecnología como llave de acceso al espacio. De este modo, la miniaturización de la tecnología satelital ha permitido a diversos países emergentes tales como Ghana, Vietnam, Rumania, Sudáfrica, Argentina, Colombia, Costa Rica, Guatemala, Perú, Ecuador, entre otros, desarrollar al menos uno de estos proyectos generando altos beneficios tecnológicos, científicos y sociales en ellos, tales como la creación de infraestructura, apoyos gubernamentales, desarrollo de pequeñas empresas y reconocimiento de la comunidad internacional. En la figura 1.27 se muestra el número de proyectos CubeSats puestos en órbita hasta octubre de 2019.

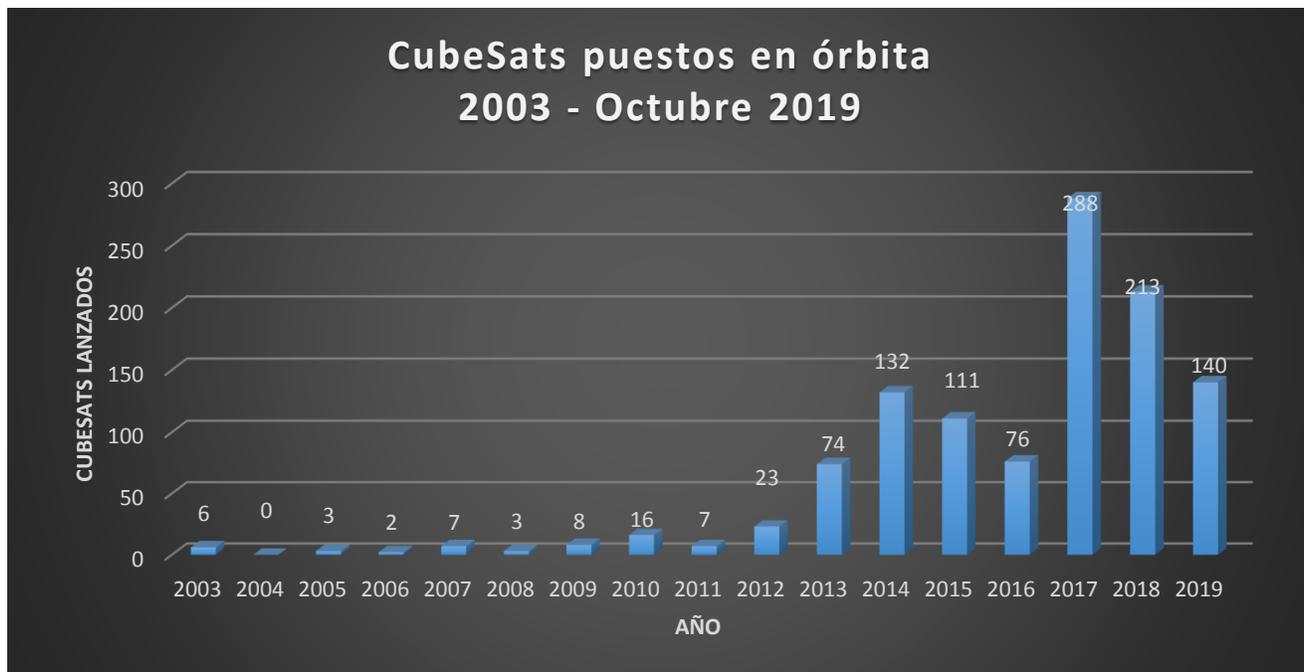


Figura 1.27. CubeSats puestos en órbita alrededor del mundo hasta octubre 2019.

En la figura 1.28 se observa como la tendencia de lanzamientos ha ido en incremento durante los últimos años. Se estima que el comportamiento en el número de lanzamientos siga en aumento y que en los próximos 5 años se pongan en órbita más de 2000 CubeSats a nivel global, es decir, un promedio de 400 satélites anualmente.

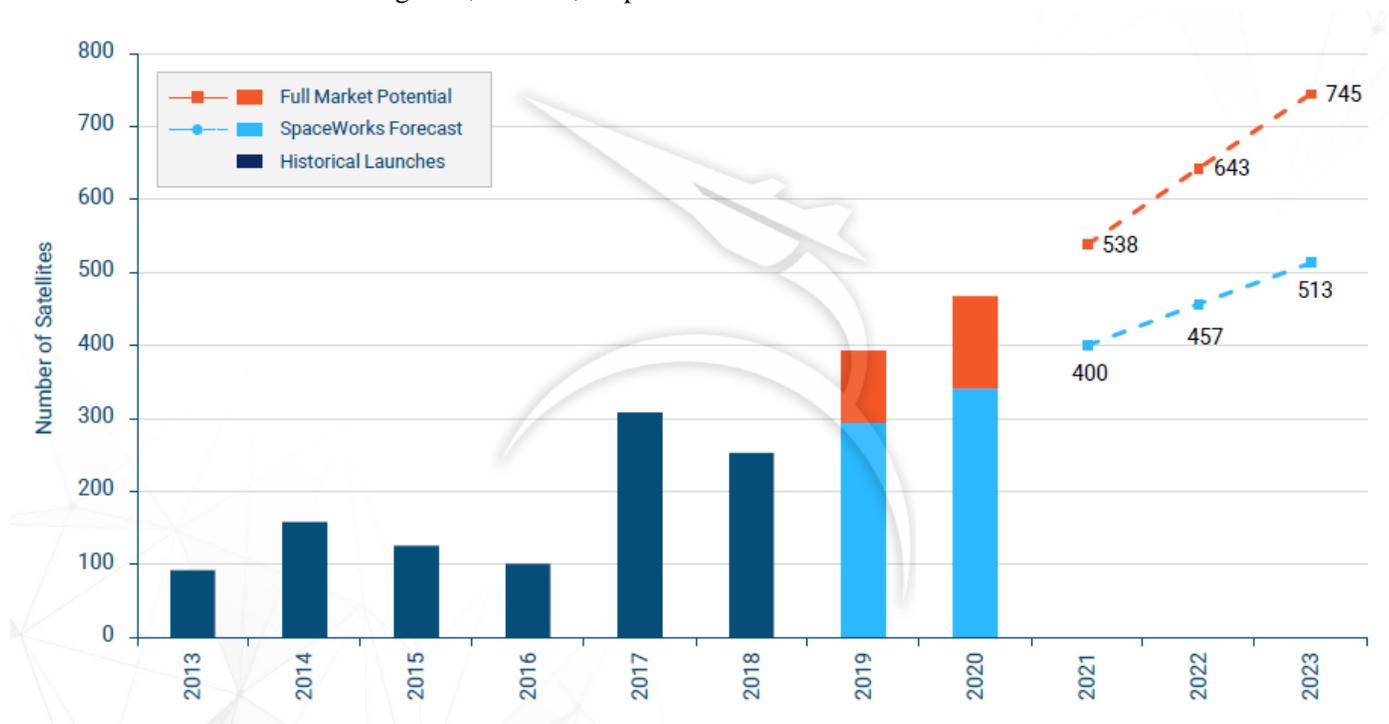


Figura 1.28. Histórico 2013-2018 de lanzamientos nanosatelitales y pronóstico al 2023 [1\_33].



Esto se debe a que la tecnología CubeSat alcanzó un grado de estabilidad y avance que le permitió pasar de ser una plataforma meramente educativa a una plataforma comercial y con aplicaciones tecnológicas y científicas de importancia en cuestión de pocos años. En la tabla 1.15 se muestran sólo algunos ejemplos de proyectos CubeSats con un impacto importante en ámbitos comerciales y tecnológicos durante los últimos años.

**Tabla 1.15. Ejemplos de proyectos CubeSat con aplicaciones comerciales y tecnológicas.**

CubeSat	Empresa/Institución	Tipo	Aplicación	Año de lanzamiento
<b>Flock</b>	Planet Labs (EUA)	Constelación 355 x 3U	Observación Terrestre	2012
<b>exactView</b>	exactEarth (Canadá)	Constelación 60 x 3U	Comunicación Marítima (AIS)	2015
<b>Lemur</b>	Spire (USA)	Constelación 100 x 3U	Comunicación Marítima (AIS) Comunicación Aérea (ADS-B)	2015
<b>Pearls</b>	Sky and Space Global	Constelación 3 x 3U	Machine-to-Machine Internet-of-Things	2019
<b>MarCo</b>	NASA	2 x 3U	Observación de la superficie de Marte	2018
<b>LEDSAT</b>	Universidad de Roma (Italia)	1U	Comunicaciones satelitales con láser	2019
<b>LAICE</b>	Universidad de Illinois (EUA)	6U	Análisis de ondas gravitacionales	2019
<b>LACCE</b>	Universidad de Lousiana (EUA)	3U	Estudio de tormentas solares en órbita geosíncrona	2019
<b>HSKSAT</b>	HSK Amateur Radio Club (Japón)	3U	Comunicaciones de alta velocidad en bandas de radio amateur	2019
<b>Andesite</b>	Universidad de Boston (EUA)	6U	Red inalámbrica intersatelital	2019

Actualmente, considerando toda la gama de ventajas y aplicaciones que los proyectos CubeSat ofrecen, hay un brote de nuevas empresas, pequeñas y medianas, enfocadas a la utilización de este tipo de satélites para proveer diferentes servicios enfocados a la Observación de la Tierra, Comunicaciones y nuevas tendencias tecnológicas tales como Internet of Things (IoT) y comunicaciones Machine-to-Machine (M2M).

En la figura 1.29 se presenta una gráfica mostrando el número de empresas creadas y enfocadas al uso de la tecnología nanosatelital desde 1990 al 2019, donde en la última década se ha presentado un incremento más que considerable en el número de compañías fundadas.

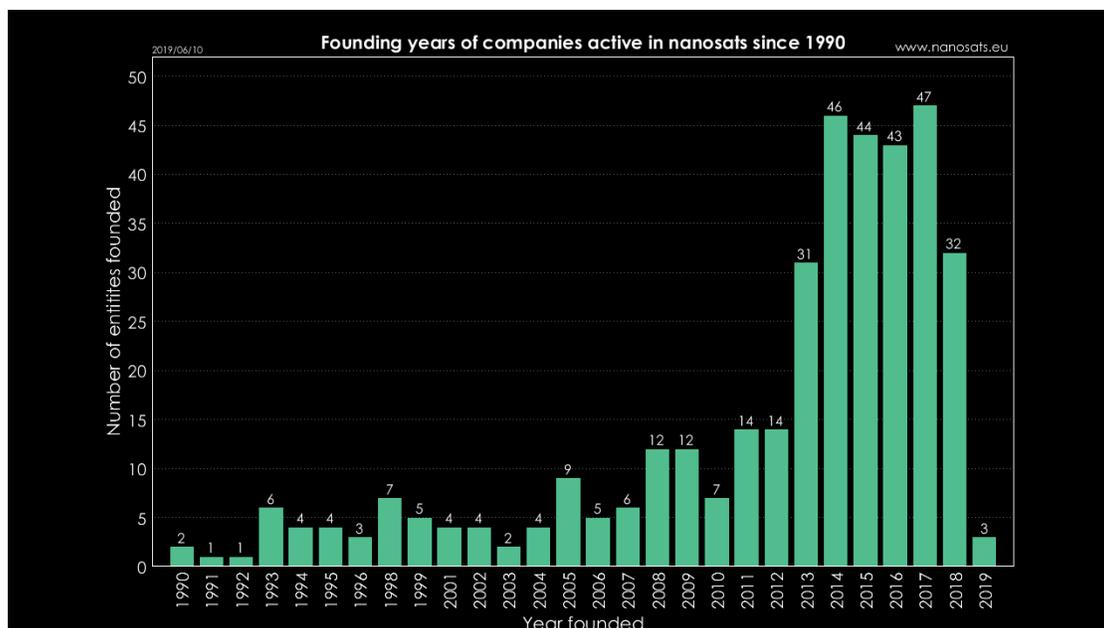


Figura 1.29. Número de compañías enfocadas a la tecnología nanosatelital de 1990 a 2019 [1\_32].

De estas compañías, son varias las que destacan en el ámbito de la tecnología CubeSat y nanosatelital. A continuación se enlista en la tabla 1.16 algunas empresas fundadas en los últimos años y se menciona su principal rubro de acción.

Tabla 1.16. Compañías que ofrecen servicios con base a plataformas CubeSat [1\_33].

Empresa	País	Fundación	Satélites	Servicios	Empleados	Comentarios
Planet Labs	EUA	2010	355 x 3U	Observación de la Tierra	480	Una imagen de la Tierra cada día = 6TB
AstroCast	Suiza	2014	64x3U	IoT/M2M	36	Fundado por el equipo que desarrolló el CubeSat Swisscube
Capella Space	EUA	2016	36	Observación de la Tierra	30	Empleo de Radar de Apertura Sintética
Aerial & Maritime	Dinamarca	2015	8x3U	Monitoreo AIS & ADS-B		SpinOff de GomSpace
Satellogic	Argentina	2010	8	Observación de la Tierra	170	—
HawkEye	EUA	2015	0	Monitoreo marítimo y aéreo	50	—
Spire	EUA	2012	100 x 3U	Monitoreo AIS & ADS-B	150	—
SpaceQuest	EUA	1994	18 x 13kg	Manufactura nanosatelital	10	—
Astro Digital	EUA	2015	6 x 6U	Observación de la Tierra	50	Constelación de 14 x 16U planeada para 2020
Sky and Space Global	Inglaterra	2015	3 x 3U	IoT/M2M	25	Constelación de 200 x 8U para 2020
Kepler Communications	Canadá	2015	2 x 3U	IoT/M2My Monitoreo	25	Operando en banda Ku @ LEO
Hiber	Holanda	2017	2 x 6U	IoT/M2M	20	—
Helios Wire	Canadá	2016	2 x 6U	IoT/M2M	10	30 x 16U en 2020
Blink Astro	EUA	2015	0	IoT/M2M	10	—
Fleet Space	Australia	2015	4 x 3U	IoT/M2M	25	—
Myriota	Australia	2013	1 x 3U	IoT/M2M	10	—



### 1.4.2. Proyectos CubeSat con SDR

Como en todos los sistemas satelitales, el subsistema de comunicaciones es una parte esencial de los proyectos CubeSat, permitiendo la transmisión de información y telemetría desde el espacio hacia Tierra y viceversa. Este tipo de comunicaciones se logra a través del uso de las radiofrecuencias, comprendidas entre los 30 MHz y los 40 GHz catalogadas en diferentes bandas de comunicación satelital [1\_34]:

- Very High Frequency (VHF): 30 a 300 MHz
- Ultra High Frequency (UHF): 300 MHz a 3 GHz
- Banda L: 1 a 2 GHz
- Banda S: 2 a 4 GHz
- Banda C: 4 a 8 GHz
- Banda X: 8 a 12 GHz
- Banda Ku: 12 a 18 GHz
- Banda K: 18 a 27 GHz
- Banda Ka: 27 a 40 GHz
- Óptica (Comunicación Láser): 100 a 800 THz

Como puede apreciarse en la figura 1.30, para los proyectos nanosatelitales de este tipo las bandas más utilizadas han sido las de VHF, UHF y X sin embargo el empleo de las otras bandas como la Ka han ido incrementando permitiendo que los CubeSats funjan como plataformas de prueba de cara a proyectos satelitales de mayor envergadura como los Satélites de Alto Rendimiento (High Throughput satellites, HTS).

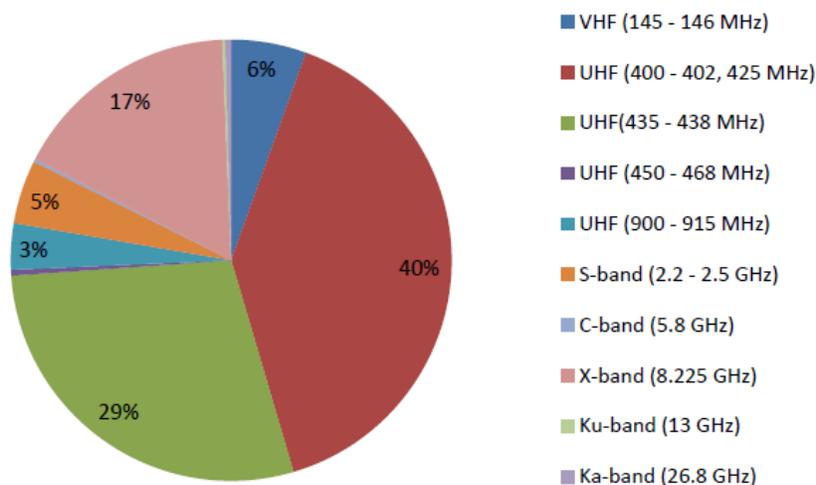


Figura 1.30. Empleo de las diferentes bandas satelitales en proyectos CubeSat [1\_5].

El inicio de la tecnología CubeSat basó el desarrollo de sus sistemas de comunicaciones en el uso de radios convencionales, configurados para trabajar a determinada frecuencia de operación bajo ciertas características particulares asociadas a los elementos electrónicos y físicos que los componen. Con el avance de la tecnología, se han desarrollado nuevas alternativas de comunicación asociadas al uso de los SDR. En el campo de los satélites CubeSat, la Radio Definida por Software ha ido tomando mayor impulso los últimos años mostrándose como una alternativa importante en el desarrollo

de las comunicaciones nanosatelitales de la actualidad [1\_35]. Ejemplo de esto son los proyectos CubeSat GOMX-1, ALL-STAR, AeroCube-7A, Altair-1, LandMapper-BC3, la constelación de CubeSats de Spire, entre otros. Además del proyecto de pruebas SCaN (Space Communication and Navigation) mostrada en la fig. 1.31, una plataforma de la NASA instalada en 2012 en la Estación Espacial Internacional enfocada a la tecnología SDR. Esta consta de tres proyectos SDR utilizados para realizar diferentes experimentos y pruebas de sistemas de comunicaciones en órbita [1\_36].

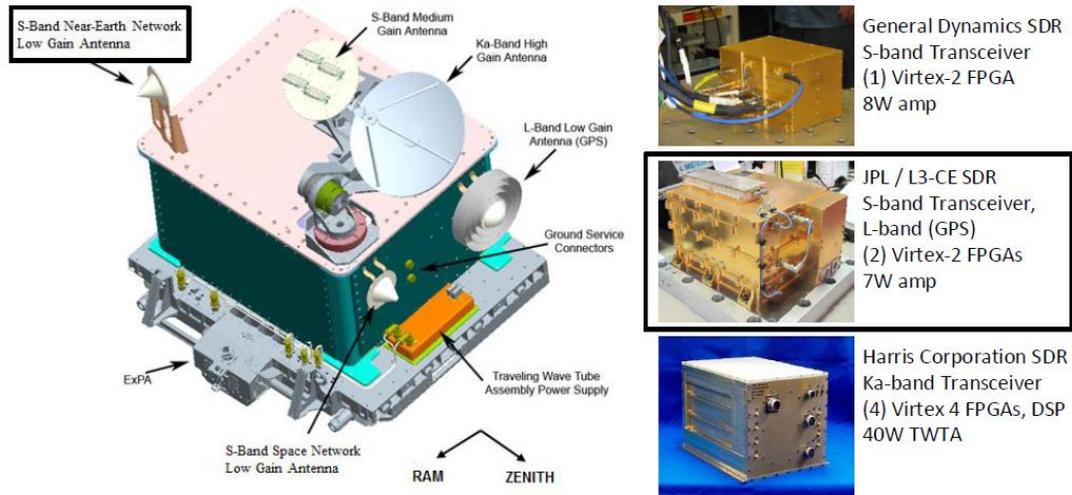


Figura 1.31. Plataforma de pruebas SCaN desarrollado por la NASA [1\_36].

**1.4.3. Proyectos CubeSat con DVB-S2**

Para el caso particular de proyectos CubeSat empleando Adaptive Coding & Modulation se decide analizar la constelación de Planet Labs dado que es la más grande en órbita hasta el momento y basan su capacidad en esta tecnología. Como se menciona anteriormente, Planet ha puesto en órbita más de 350 CubeSats de 3U y es capaz de tomar una fotografía de la superficie total de la Tierra cada día, lo cual representa un flujo de datos arriba de 6 Terabytes. Para permitir tal capacidad de descarga, Planet no sólo ha enfocado recursos a su constelación satelital, también lo ha hecho a su infraestructura en tierra, incluyendo sus diversas estaciones terrenas y centros de operaciones. De este modo, son capaces de alcanzar una velocidad de descarga de 220 Mbps pico equivalente a 160 Mbps promedio en toda su constelación [1\_37].

La constelación activa de Planet consta de aproximadamente 150 CubeSats, llamados Doves, dispuestos en órbitas heliosíncronas (Sun-synchronous Orbit, SSO) y en la órbita de la ISS, los cuales pueden capturar imágenes con una resolución espacial de hasta 3 metros. En la tabla 1.17 se muestra una comparativa entre ambas órbitas con el fin de notar sus semejanzas y diferencias.

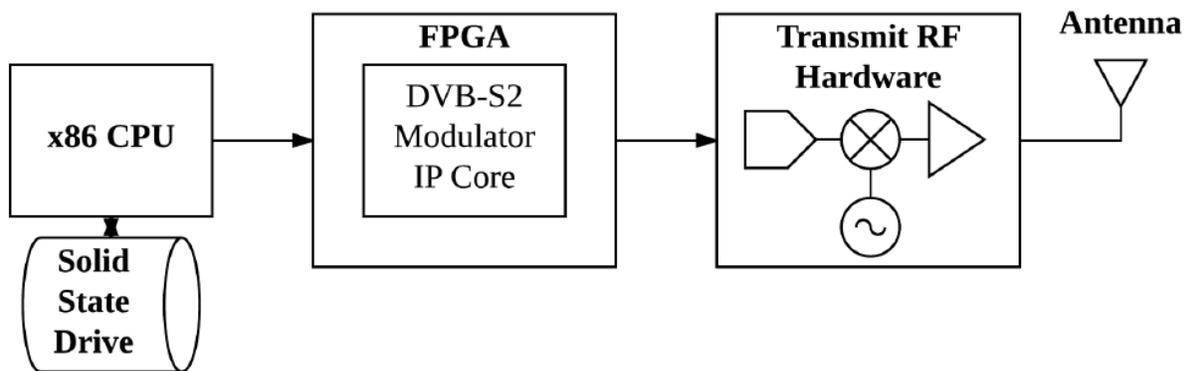
**Tabla 1.17. Comparativa de las órbitas ISS y SSO para aplicaciones de observación terrestre [1\_37].**

Parámetro/órbita	ISS	SSO
<b>Altitud</b>	420 km	475 km
<b>Inclinación</b>	51.6°	98°
<b>Tiempo de vida</b>	1 año	2-3 años
<b>Cobertura</b>	No existe en el Norte de Canadá, Rusia, Antártica y en la parte inferior de Sudamérica	Total, con excepciones en los polos
<b>Resolución Espacial</b>	2.7-3.2 m	3.7-4.9m
<b>Tamaño del frame</b>	20 x 12 km	24 x 16 km
<b>Bandas espectrales</b>	Blue: 455 - 515 nm Green: 500 - 590 nm Red: 590 - 670 nm NIR: 780 - 860 nm	Blue: 455 - 515 nm Green: 500 - 590 nm Red: 590 - 670 nm NIR: 780 - 860 nm

Sus características de transmisión hacia tierra están dadas por un enlace de bajada que opera en la banda X a una frecuencia central de 8150 MHz con una tasa de datos de 70 Mbaud/s , ventajas de modulación y codificación brindadas por el estándar DVB-S2 y una antena helicoidal de alta ganancia entre los 10 y 12 dBi. Las estaciones terrenas de la constelación son antenas parabólicas entre los 4.5 y los 7.6 metros dispuestas en 8 diferentes puntos geográficos.

**1.4.3.1. Diseño del transmisor**

Los sistemas de transmisión de cada satélite Dove constan de un procesador convencional x86, un FPGA donde se encuentra embebido el estándar DVB-S2 y una serie de elementos analógicos que conforman el Front End de Radiofrecuencia. Este arreglo se muestra en la figura 1.32.



**Figura 1.32. Diagrama de bloques del transmisor de un CubeSat de Planet Labs [1\_37].**

Planet desarrolló su propio protocolo de empaquetamiento de frames los cuales son, a su vez, encapsulados en los frames dados por el estándar DVB-S2. La información digital modulada a la salida del FPGA es convertida a una señal

analógica por un DAC y un transmisor superheterodino es empleado para pasar esta señal de banda base a banda X. La etapa final corresponde a un amplificador que brinda una potencia de 2W a la señal, la cual es después transmitida a las estaciones terrenas empleando una antena helicoidal con polarización circular y una ganancia de 12 dBi.

Una de las ventajas más importantes del sistema de transmisión diseñado por Planet es su capacidad de reconfiguración en tiempo real. Parámetros como la frecuencia de operación, tasa de símbolos, ancho de banda, ganancia del amplificador, y otros asociados a los componentes de RF pueden ser modificados cuando sea necesario.

### 1.4.3.2. Diseño del Receptor

Los receptores en tierra se componen de elementos COTS (comercial-off-the-shelf) de bajo costo que operan en la banda L. Un convertidor de bajada, de banda X a banda L, es adicionado a la antena receptora para asegurar la compatibilidad. En el bloque de recepción se aplica un algoritmo de ACM, igualmente desarrollado por Planet, que otorga valores y estadísticas del SNR recibido en el margen que posee el enlace a cada momento. De este modo, una vez cada segundo, es enviada una señal de tierra a cada satélite, para ajustar sus parámetros de modulación y codificación dependiendo de las condiciones en las que se encuentre el enlace.

En la figura 1.33 se aprecia el diagrama de bloques general del esquema de recepción descrito.

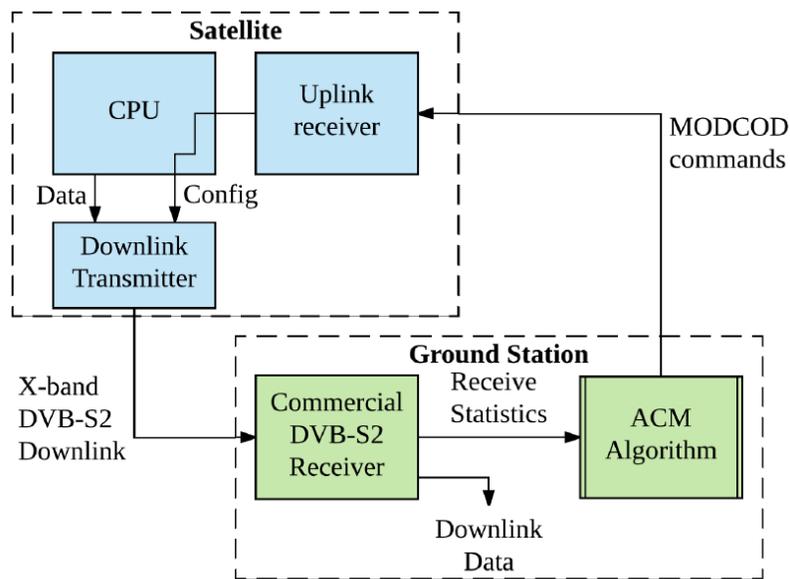


Figura 1.33. Diagrama de bloques de recepción de Planet Labs [1\_37].

### 1.4.3.3. Carga útil

La carga útil de los satélites Dove se compone de un telescopio, una cámara CCD (Charge-Coupled-Device) y un procesador de imágenes. El telescopio es el encargado de capturar los fotones en un rango amplio del espectro y posteriormente son filtrados utilizando un filtro Bayer el cual sólo permite el paso en las bandas RGB (Red-Green-Blue) e infrarrojo. La cámara CCD convierte los fotones filtrados en electrones, que son amplificados con el propósito de producir una señal digital correspondiente a un pixel asociado a cada una de las bandas espectrales permitidas. Finalmente, la

información digital es convertida a imágenes, las cuales son enviadas a una base de datos administrada por el procesador x86 en el transmisor. Ver fig. 1.34.

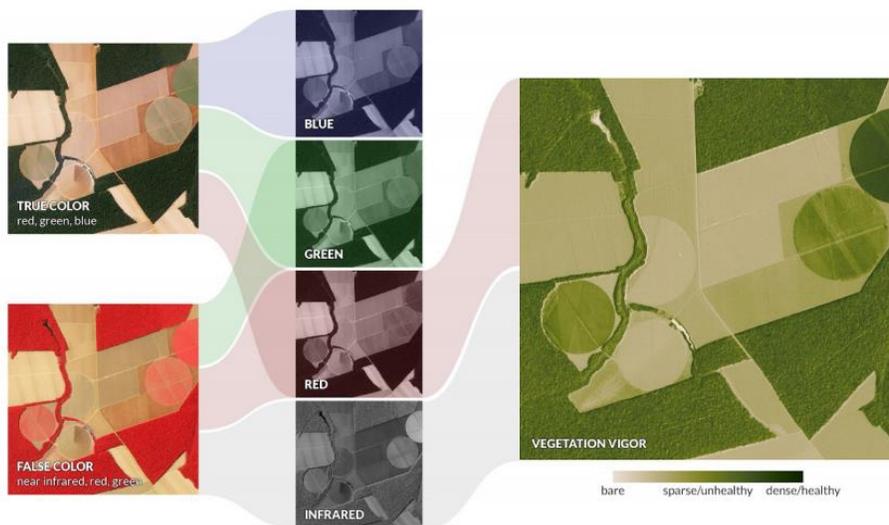


Figura 1.34. Imágenes de Planet en sus diferentes bandas espectrales [1\_31].

Con estas características las imágenes tomadas tienen un tamaño de pixel de 3.7 metros y una profundidad de 12 bits, abarcando aproximadamente un área de 24 x 16 km por fotografía considerando una órbita SSO. En el caso de una órbita ISS se tiene un tamaño de pixel de 3 metros, una profundidad de 12 bits y un área de captura de 20 x 12 km.

La profundidad puede entenderse como la cantidad de detalle que tiene la imagen. De manera concreta se refiere al número de colores posibles de codificar un pixel. Este número de colores se expresa como  $N=2^p$  donde N es el número total de colores posibles y p es la profundidad. De este modo, una profundidad de 12 bits significa que cada pixel puede ser asignado con una de las  $2^{12}$  posibilidades de color disponibles.

Con esta información se puede hacer un estimado del tamaño de cada fotografía empleando la siguiente expresión:

$$\begin{aligned} \text{Tamaño de la fotografía} &= \text{Número total de pixels} \times \text{profundidad} \\ &= \frac{\text{Tamaño del frame}}{\text{Resolución/pixel}} \times \text{profundidad} \end{aligned}$$

Para el caso de la órbita ISS se tiene:

$$\frac{20,000 \times 12,000 \text{ m}^2}{(3 \times 3 \text{ m}^2/\text{pixel})} \times 12 \text{ bits/pixel} = 38.14 \text{ MBytes}$$

Y para la órbita SSO se tiene:

$$\frac{24,000 \times 16,000 \text{ m}^2}{(3.7 \times 3.7 \text{ m}^2/\text{pixel})} \times 12 \text{ bits/pixel} = 40.12 \text{ MBytes}$$

Esta información es resumida en la tabla 1.18 mostrada a continuación.

Tabla 1.18 Características de las fotografías capturas por lo constelación Planet Labs [1\_31].

Parámetro/Órbita	ISS	SSO
Resolución	3 metros/pixel	3.7 metros/pixel
Profundidad	12 bits	12 bits
Tamaño del frame	20 x 12 km	24 x 16 km
Tamaño de la fotografía	38.14 MBytes	40.12 MBytes



Figura 1.35. Fotografía tomada por un satélite de la constelación Planet Labs [1\_31].



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



## CAPÍTULO 2. ANÁLISIS ORBITAL

En este capítulo se presenta el análisis orbital realizado en la versión gratuita del programa System Tool Kit (STK) versión 11.5, desarrollado por la empresa Analytical Graphics Inc [2\_1]. Este software es capaz de realizar simulaciones complejas de sistemas, desarrollados u operando en aire, tierra, mar o en un ambiente espacial, de manera gráfica y en 3D. Fue originalmente diseñado para resolver problemas relacionados con las tecnologías satelitales pero actualmente es utilizado con enfoques aeroespaciales en general.

STK es utilizado por diversos organismos e instituciones tanto gubernamentales como privados, y para el desarrollo de aplicaciones académicas, civiles o militares. Ejemplos de estos entes son diferentes agencias espaciales alrededor del mundo como NASA, ESA, CNES, DLR, JAXA, ISRO, y diferentes empresas como Boeing, Lockheed Martin, Airbus, entre otros.

Para el análisis realizado se consideraron parámetros tales como altitud e inclinación orbital, tipo de satélite, ubicación de la estación terrena y periodo de la misión. Con las simulaciones realizadas fue posible determinar los rangos variables, y el número y duración de los pases entre la estación terrena y el satélite de enero a diciembre de 2020. Esta información es posteriormente empleada para llevar a cabo el Análisis de Comunicaciones descrito en el Capítulo 3 de esta tesis.

### 2.1 GENERALIDADES DE LA MISIÓN A ANALIZAR

Con la información obtenida en el Estado del Arte se establecen los requisitos y objetivos de una misión satelital hipotética:

- **Plataforma:** CubeSat 3U
- **Aplicación:** Observación terrestre
- **Órbitas:** ISS y SSO
- **Plataforma de comunicaciones:** SDR
- **Estándar de comunicaciones (transmisión):** DVB-S2
- **Banda de operación:** Banda S
- **Estación Terrena:** Querétaro, Juriquilla

Se considera la Estación Terrena ubicada en Juriquilla dado que esta es coordinada por la UNAM y la proyección a futuro de este proyecto considera su utilización. Se pueden encontrar más especificaciones de esta Estación Terrena en [2\_2]. Con estos puntos es posible detallar algunos parámetros tanto en el segmento espacial como en el terrestre del sistema de comunicaciones. Estos valores están expresados en las tablas 2.1 y 2.2.



Tabla 2.1. Consideraciones del Segmento Espacial.

Satélite	
Plataforma	CubeSat 3U
Potencia de Transmisión	500 mW y 1W
Frecuencia de Operación	Banda S Up: 2025-2110 MHz Down: 2200-2290 MHz
Órbita	a) 406km, 51.64° (ISS) b) 600km, 98° (heliosíncrona)
ModCod	ACM Mod: PSK y APSK LDPC ½ hasta 9/10 Roll-off: 0.2
Antena	Parche 6 dBi
Ancho de Banda	10 MHz
Aplicaciones	Observación Terrestre
PER	$10^{-7}$

Tabla 2.2 Consideraciones del Segmento Terrestre [2\_2].

Estación Terrena (ECXSAT)	
Ubicación	Querétaro Juriquilla <u>20.704993,</u> <u>-100.450931</u> Altura: 1966m snm
Antena	Parabólica 3.4m 34.58 dBi
Frecuencia de Operación	Banda S Up: 2025-2110 MHz Down: 2200-2290 MHz
Polarización	LHCP y RHCP
Amplificador de Alta Potencia	30W
Ancho de banda de Tx	6MHz

La selección de frecuencias, tanto de subida como de bajada, toma como referencia la asignación de frecuencias dados a la Investigación y Exploración Espacial Espacio-Tierra que el Instituto Federal de Telecomunicaciones indica en el Cuadro Nacional de Atribución de Frecuencias (CNAF) [2\_3]. Ver fig. 2.0.

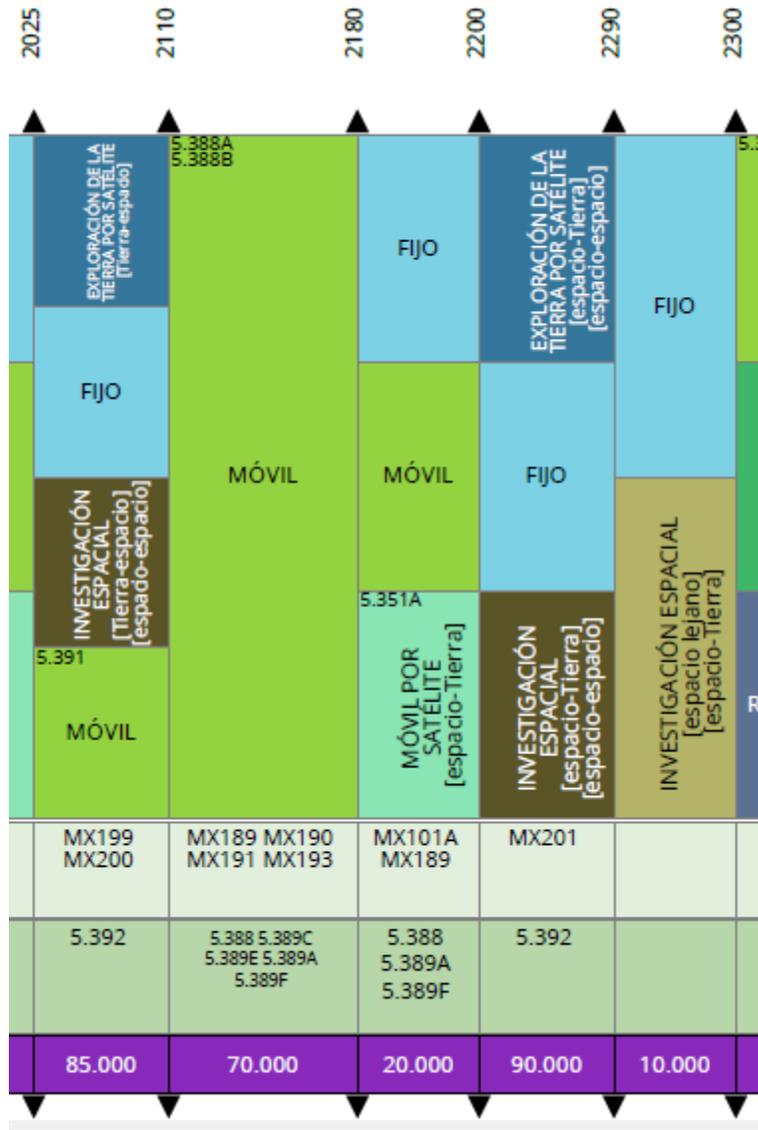


Figura 2.0. Cuadro Nacional de Atribución de Frecuencias. 2025 -2300 MHz [2\_3].

Para el planteamiento de la misión se toman en cuenta dos órbitas: la de la ISS y una heliosíncrona a 600km. Esto se hace debido a las restricciones existentes en la selección de la órbita de operación provocadas por el hecho de que los proyectos CubeSat son cargas secundarias dentro de los vehículos lanzadores. Ambas órbitas propuestas son dos de las más comunes para la puesta en órbita de proyectos CubeSat debido al interés comercial de proyectos que suben como cargas útiles primarias. Finalmente, se proponen analizar dos valores para la potencia de transmisión: 0.5 W y 1 W. Esto con el objetivo de verificar cómo cambia el desempeño del sistema de comunicaciones al variar la potencia de transmisión, y entender qué valor de potencia podría funcionar mejor para los objetivos de la misión.

## 2.2 SIMULACIÓN ORBITAL CON STK

Con las consideraciones generales se crea un escenario de simulación en STK v11.5.0, estableciendo la Estación Terrena (ET) en Juriquilla y analizando dos tipos de órbitas: la de la Estación Espacial Internacional y una órbita heliosíncrona a 600 km de altura. La simulación se realiza para los intervalos del 01/01/2020 al 31/12/2020 con una frecuencia de muestreo de 1 segundo.

Los datos obtenidos de esta simulación son:

- Número de pases satélite – Estación Terrena
- Duración de cada pase
- Rango respecto al tiempo de la línea de vista entre el satélite y la ET.

A continuación se describirá paso a paso la configuración del escenario en STK para posteriormente describir la puesta en marcha de la simulación y el análisis de resultados

### 2.2.1. Apertura STK

1) Se trabaja con la versión 11.5.0 de STK disponible en [2\_3]. Para realizar la descarga es necesario registrarse en la página con un correo institucional y solicitar una licencia gratuita, la cual cuenta con la capacidad de analizar gráficamente y entregar los reportes descritos anteriormente. Las limitantes de la licencia gratuita, en el caso del análisis propuesto, están relacionadas con simulaciones orientadas a las comunicaciones bloqueando reportes tales como Cálculo de Enlace, Bit Error Rate y cálculo de Es/No, los cuales se han desarrollado de manera independiente utilizando Matlab y se describen en el siguiente capítulo.



Figura 2.1. System Tool Kit v11.5.0.



### 2.2.2. Creación del escenario

2) Al abrir STK, el cuadro de bienvenida que se recibe es como el mostrado en la figura 2.2. En este se selecciona la opción “Create a Scenario”.

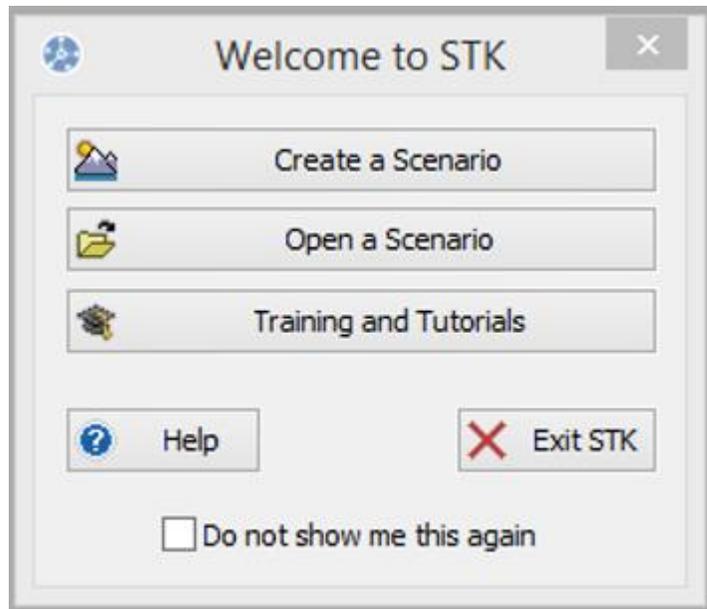


Figura 2.2. Cuadro de bienvenida a STK.

3) A continuación, en la fig. 2.3, se le asigna nombre y descripción al escenario y se configura la fecha y hora de inicio y termino de la simulación. Esta configuración es de suma importancia pues este intervalo de tiempo es el que STK considera para la generación de todos los reportes.

Se establece el periodo de simulación el intervalo entre el 1 de enero de 2020 y el 31 de diciembre de 2020. En términos de proyectos CubeSat, un periodo de simulación de un año es suficiente para poder realizar una proyección de los alcances y restricciones de la misión.

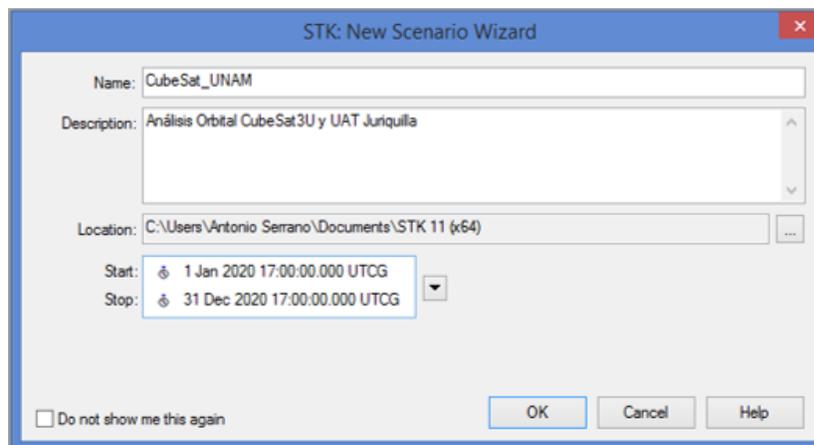


Figura 2.3. Configuración del periodo de la misión en STK.

4) Los pasos siguientes están enfocados a generar los objetos que conformarán el escenario a simular. Para esta misión hay 3 objetos involucrados: Una ET ubicada en Querétaro, un satélite en la órbita de la Estación Espacial Internacional y un satélite en una órbita heliosíncrona.

Para la creación de la ET se comienza estableciendo un centro de operaciones. Para ello, de la barra de menú se elige la opción “**Insert STK Objects**” y en la ventana desplegada se selecciona el objeto “**Facility**”. El método de inserción se establece como “**Define Properties**”. Ver fig. 2.4.

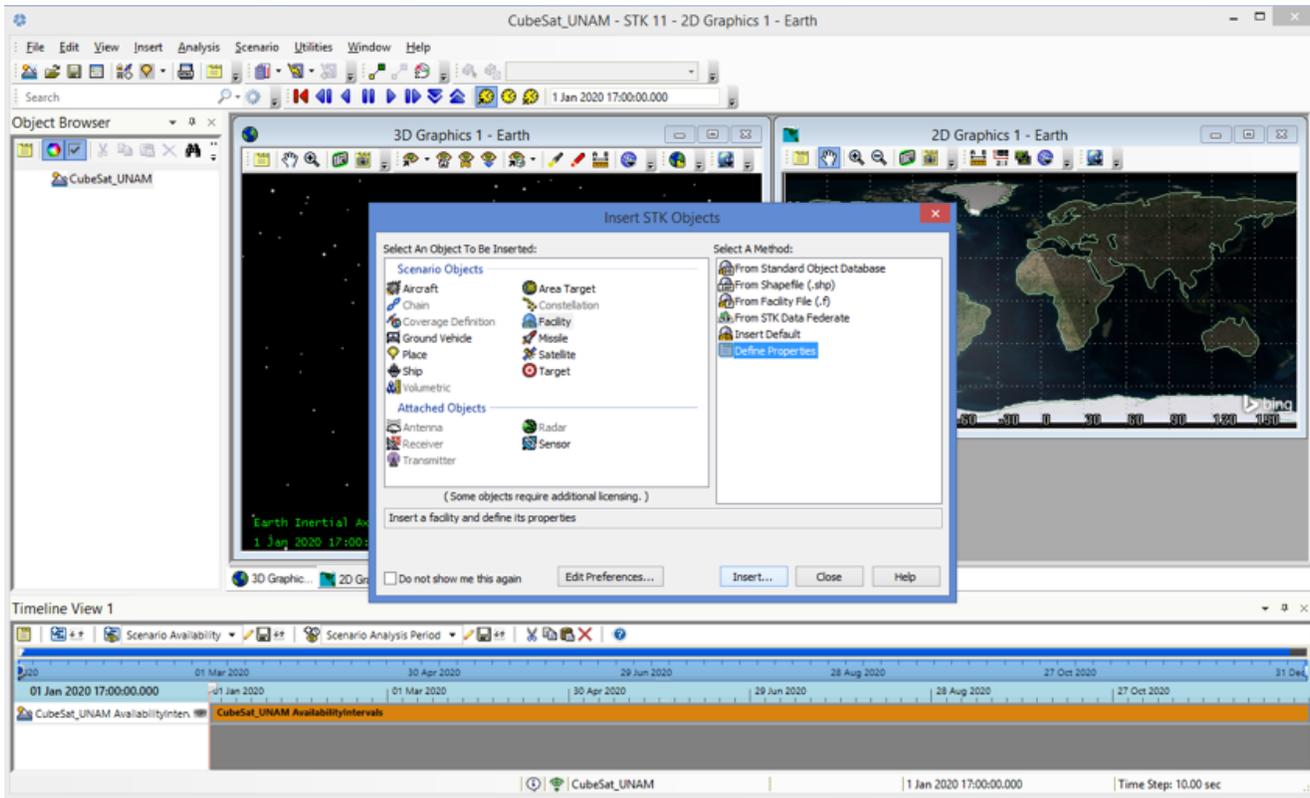


Figura 2.4. Inserción de objeto en STK.

De la ventana desplegada se selecciona la ubicación de la ET con el uso de sus coordenadas y la altura a la que se encuentra por encima del suelo. Para la ET en la Unidad de Alta Tecnología se tienen las coordenadas 20.705,-100.451 y la altura aproximada a la que se encuentra la antena es de 10 metros. Ver fig. 2.5.

Se selecciona la opción de “**Apply**” y con ello habremos establecido el centro de control o de modo simple el edificio en donde se ubicará la antena del escenario simulado.

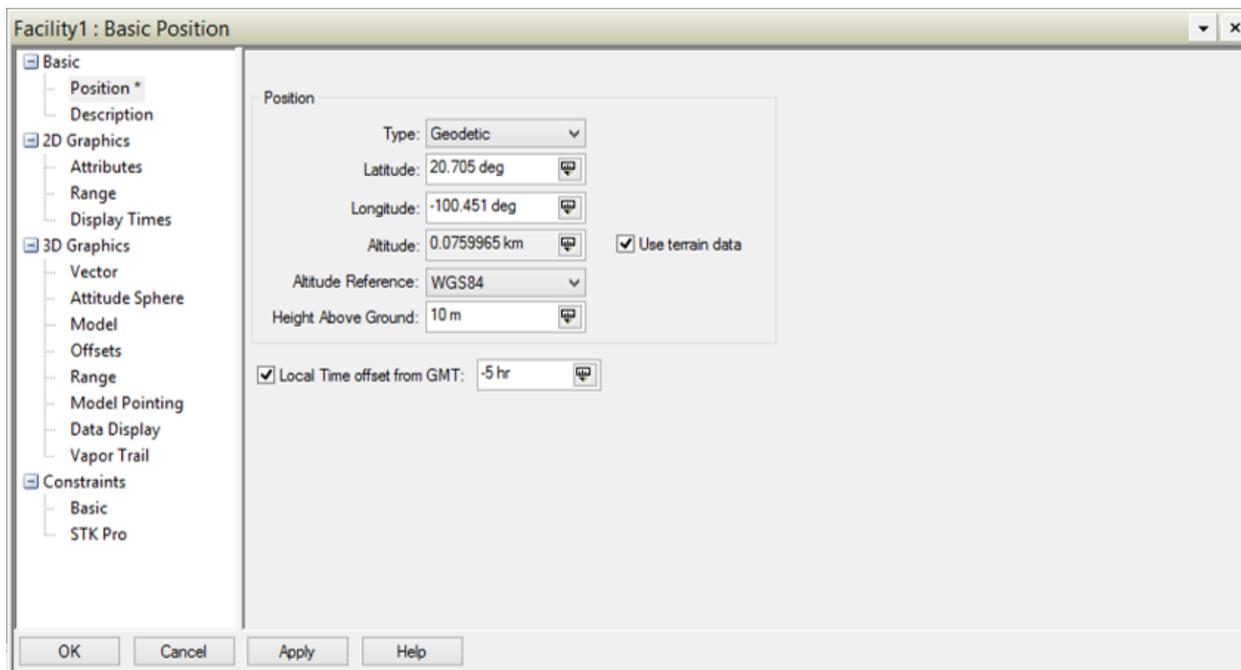


Figura 2.5. Configuración de ubicación de la ET de la UAT, Juriquilla.

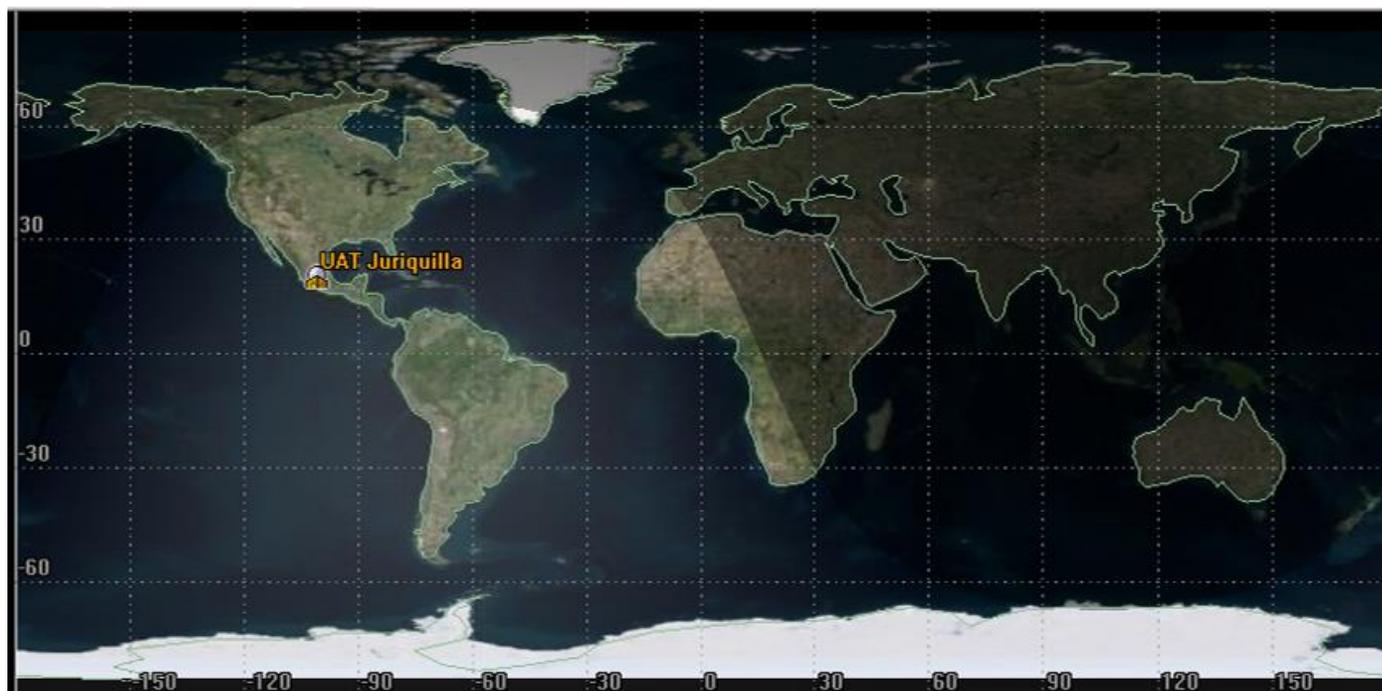


Figura 2.6. Mapa 2D de la ET de la UAT, Juriquilla.

5) Posterior a la inserción de la ET se le debe de asociar una antena a la misma. Para la versión gratuita que se está manejando la opción de añadir Antenas, Receptores, Transmisores o Radars está bloqueada por lo que se utilizará un elemento tipo sensor en su lugar. Este elemento, si bien no es capaz de simular enlaces de comunicaciones, es suficiente para simular una región de cobertura con lo cual podremos calcular los pases y las líneas de vista entre los satélites y la ET.



Para insertar este objeto se selecciona nuevamente la opción “Insert”/”Insert object” y en esta ocasión de la lista de “Attached Objects” se elige “Sensor”. El método de inserción vuelve a ser “Define properties”. Ver fig. 2.7.

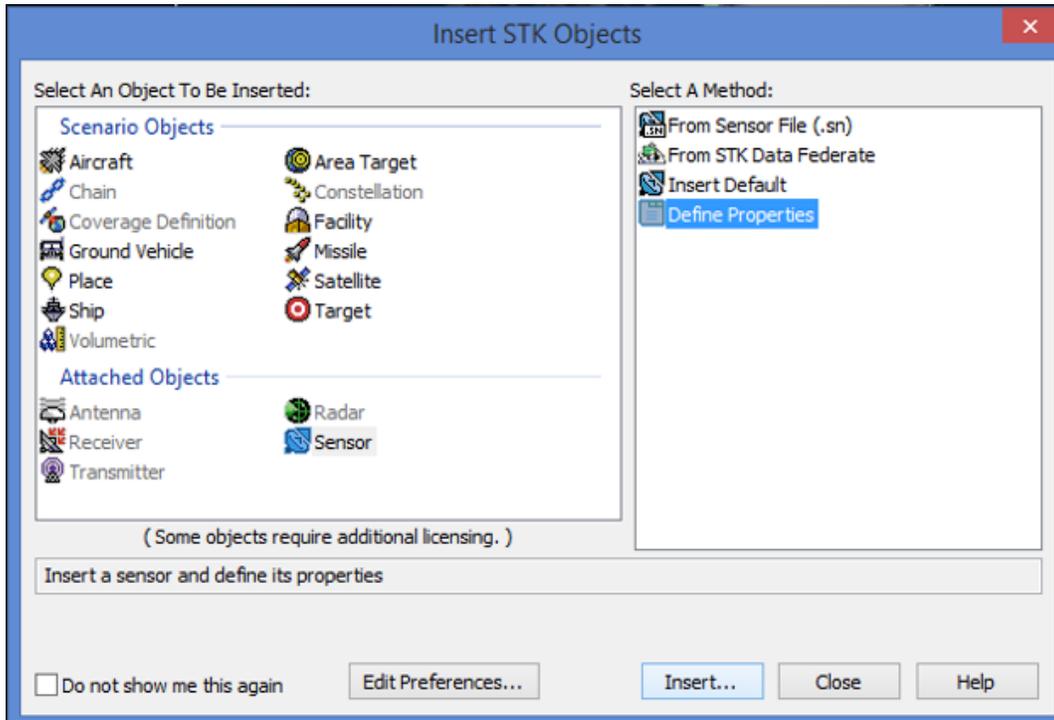


Figura 2.7. Inserción del objeto tipo sensor para la ET.

En la ventana desplegada a continuación, fig. 2.8, se selecciona un sensor de tipo “Simple Conic” y con un cono de aceptación de 85° lo cual se relaciona con el tipo de cobertura y el ángulo de aceptación de la ET.

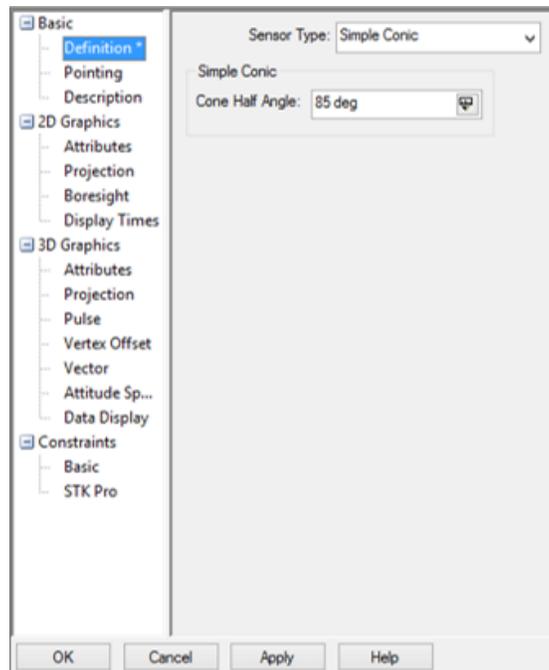


Figura 2.8. Definición del tipo de sensor.

De este modo, estará completamente definida la ET dentro el escenario de simulación como se aprecia en la fig. 2.9.

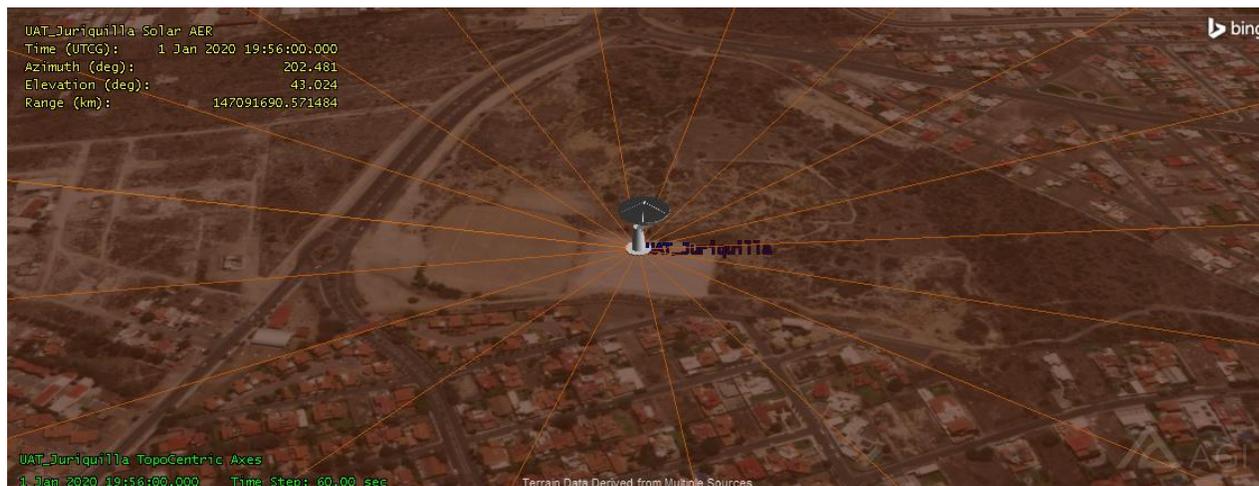


Figura 2.9. Estación terrena de Juriquilla configurada en STK.

6) Se configura ahora el satélite con órbita ISS. Esto es, con una altura orbital de 406km y un ángulo de  $51.64^\circ$ . Para ello se hace uso una vez más de la opción “**Insert STK object**”. En esta ocasión el objeto a insertar será “**Satellite**” y el método de inserción “**Orbit Wizard**”. Ver fig. 2.10.

Con Orbit Wizard es posible configurar los parámetros básicos de una órbita de manera muy sencilla. Con este se puede elegir el tipo de órbita, altura e inclinación. Adicional a esto, da la opción para elegir el render o figura que se empleará para representar visualmente al satélite en la simulación.



Figura 2.10. Inserción de objeto tipo satélite.

Dentro de Orbit Wizard se elige Órbita Circular, 406km, 51.64°.

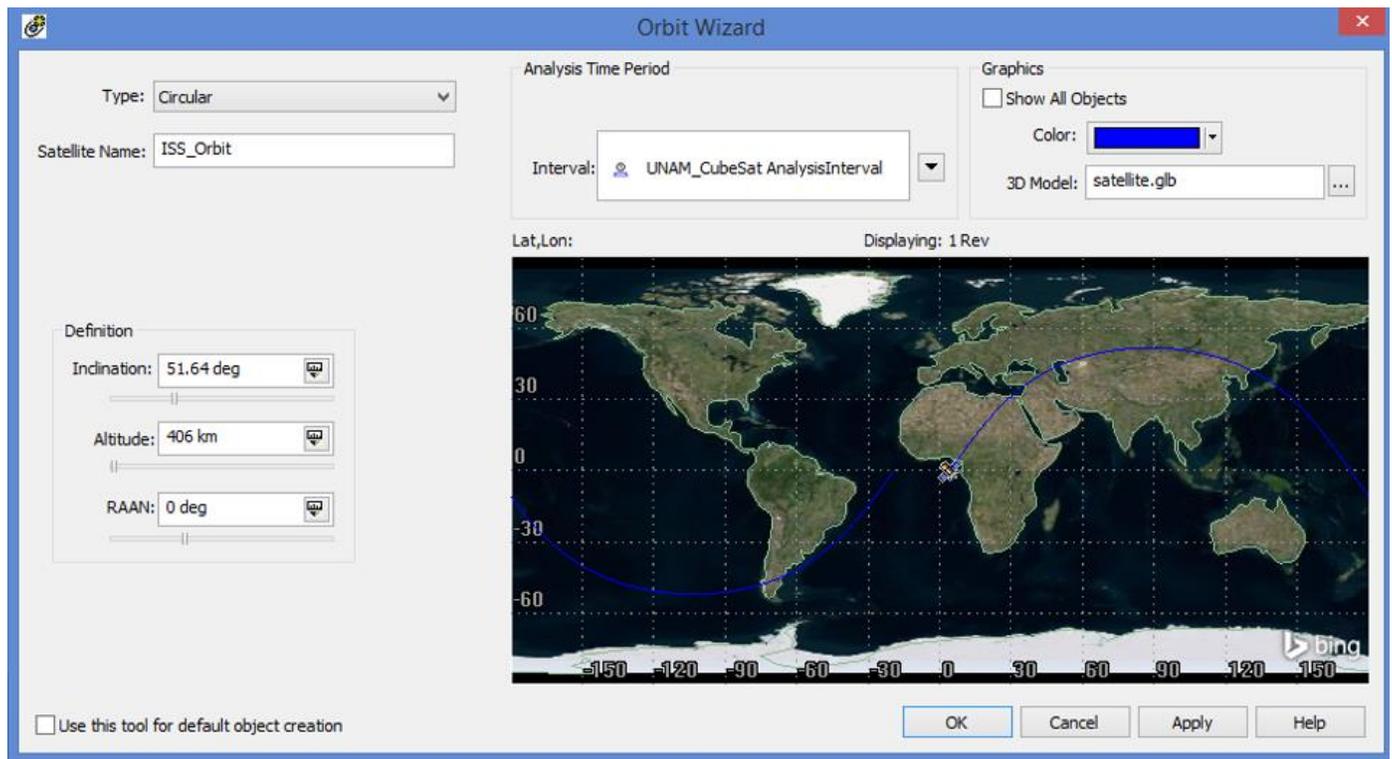


Figura 2.11. Orbit Wizard ISS.

Con ello se habrá configurado correctamente el satélite en una órbita ISS.

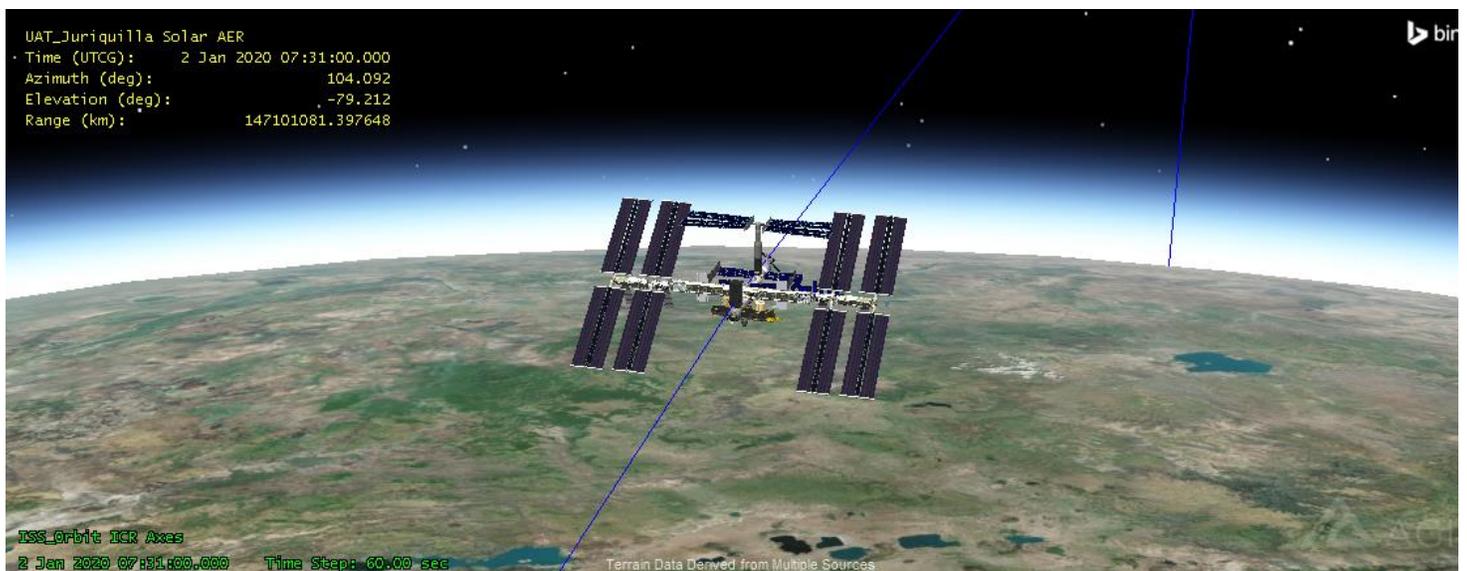


Figura 2.12. Órbita de la ISS configurada en STK.



7) El último objeto a insertar es el satélite en una órbita heliosíncrona con una altura orbital de 600km y un ángulo de 98°. Para ello se sigue el mismo procedimiento de inserción anterior: **Insert/Insert Object/Satellite/Orbit Wizard**.

En Orbit Wizard se establecen los parámetros Sun Synchronous en el tipo de órbita y una altura de 600km. Ver fig. 2.13.

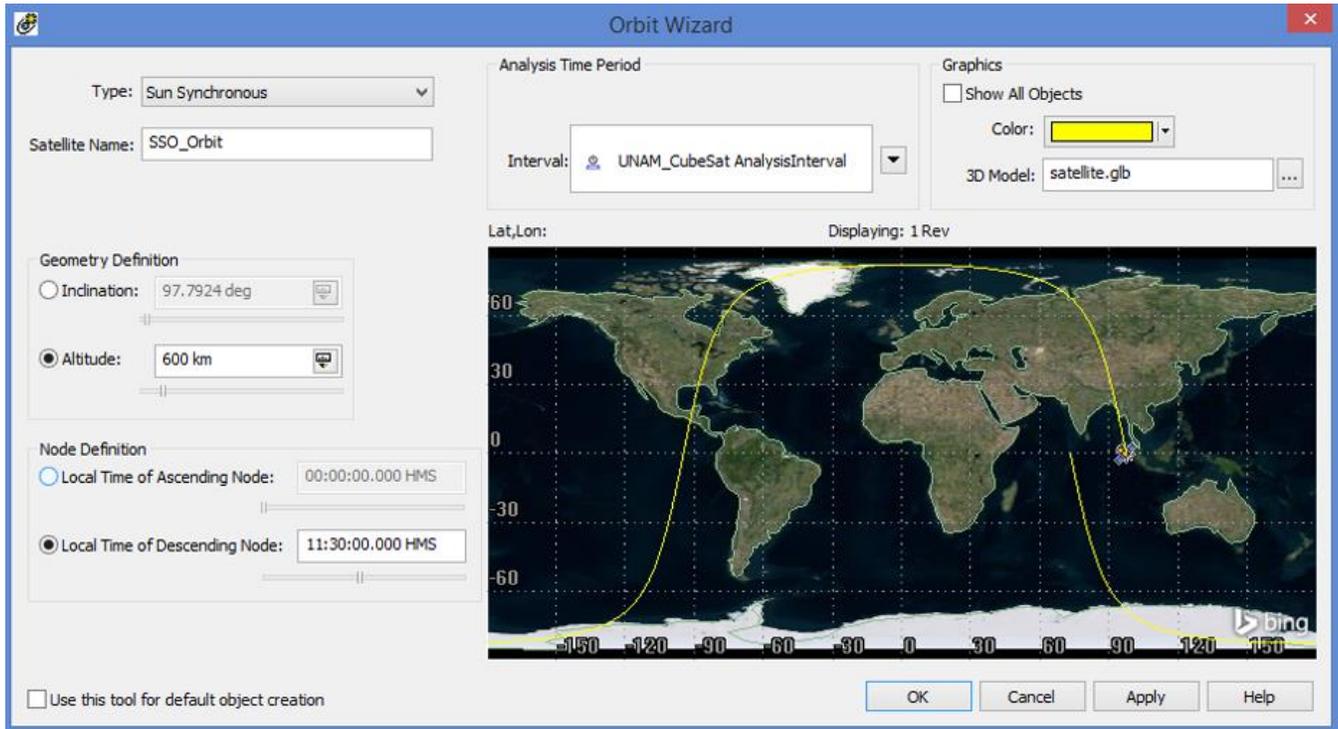


Figura 2.13. Orbit Wizard SSO.

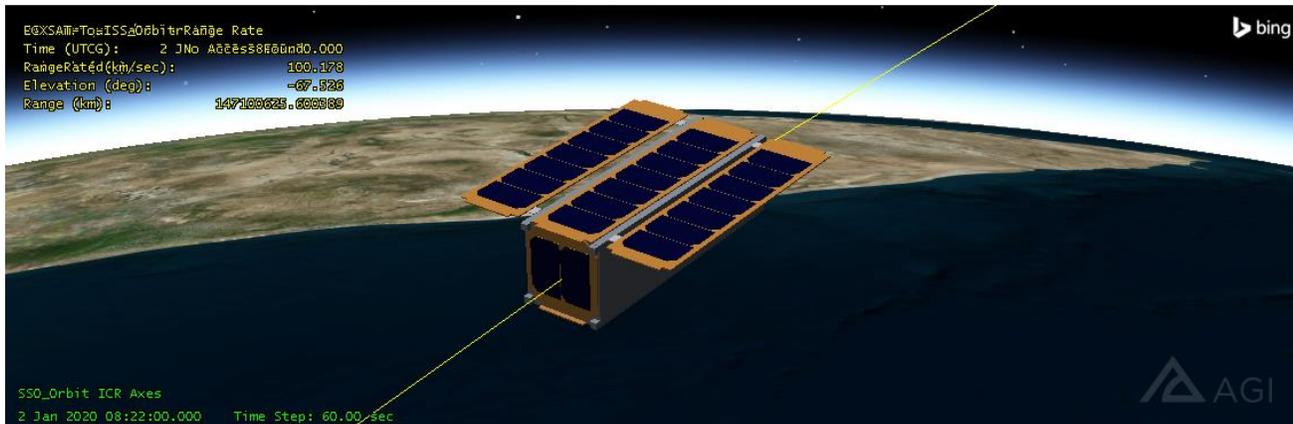


Figura 2.14. Satélite CubeSat de 3U en órbita SSO configurado en STK.

En este momento ya están configurados los objetos dentro del escenario propuesto, al igual que el periodo de simulación.



Se puede visualizar el escenario en los mapas 2D o 3D dados por STK. En el mapa 2D, mostrado en la figura 2.15, es posible observar las trayectorias que siguen las órbitas ISS y SSO por encima de la superficie de la Tierra. Esto es particularmente útil para poder seguir la posición del satélite durante la simulación.

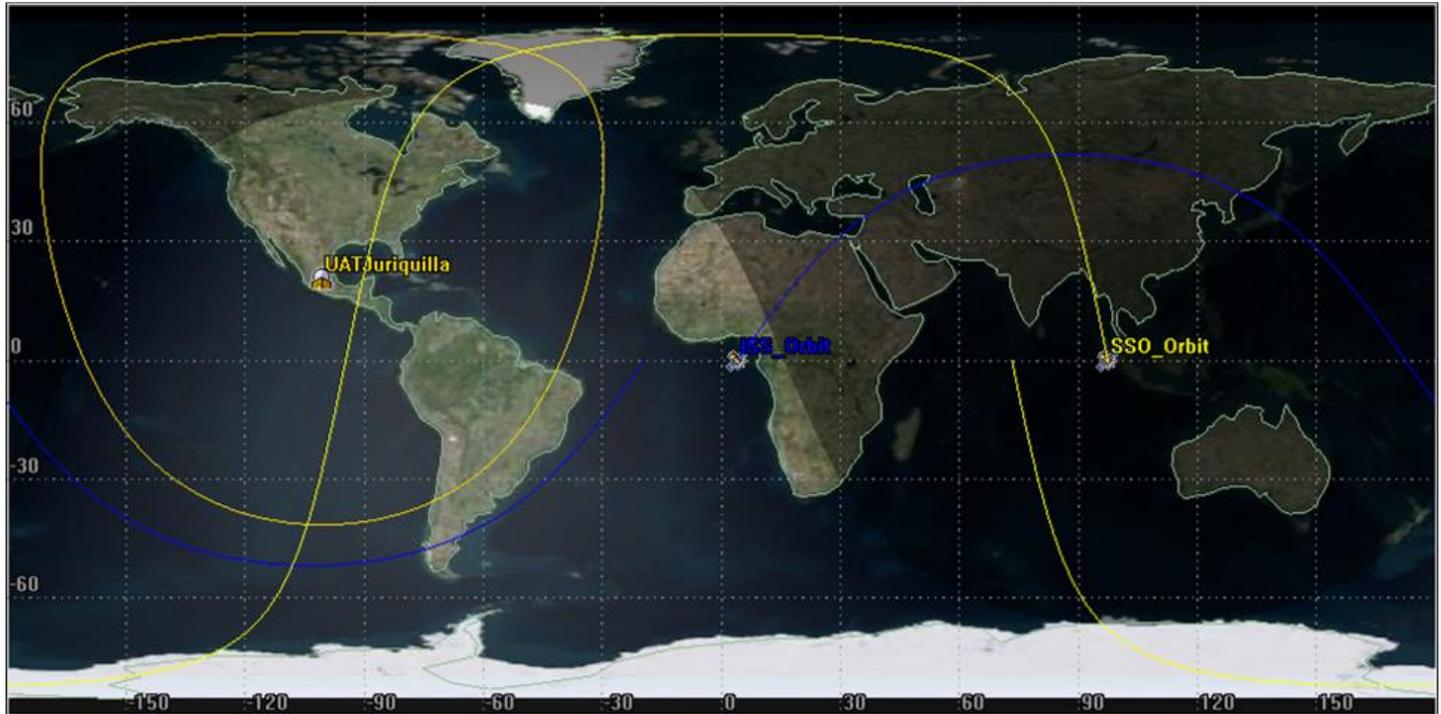


Figura 2.15. Mapa 2D del escenario configurado en STK.

En el mapa 3D de las figuras 2.16, 2.17 y 2.18, se observa, en forma de esfera, la región de cobertura en línea de vista dada por la ET en todas direcciones.

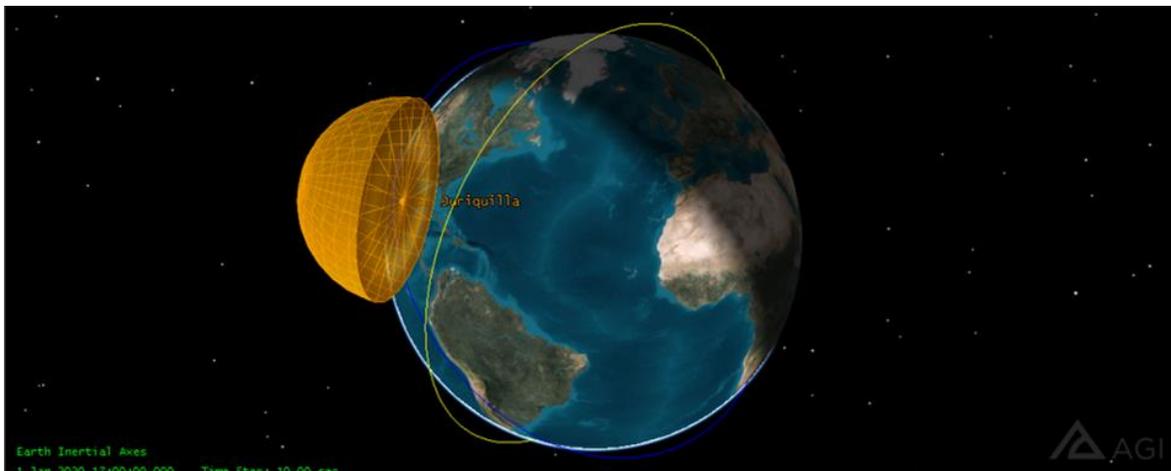


Figura 2.16. Mapa 3D del escenario configurado en STK.

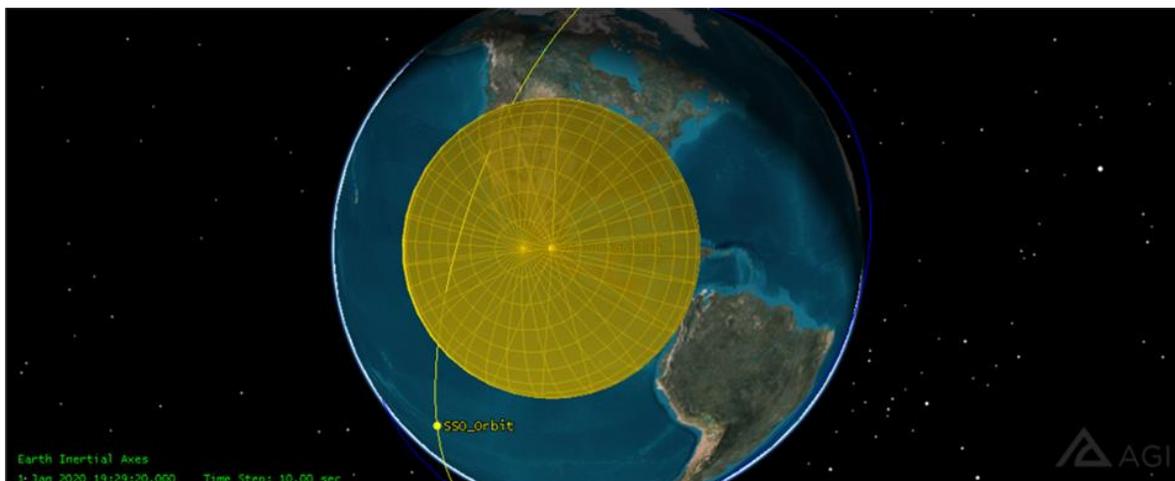


Figura 2.17. Mapa 3D del escenario configurado en STK.

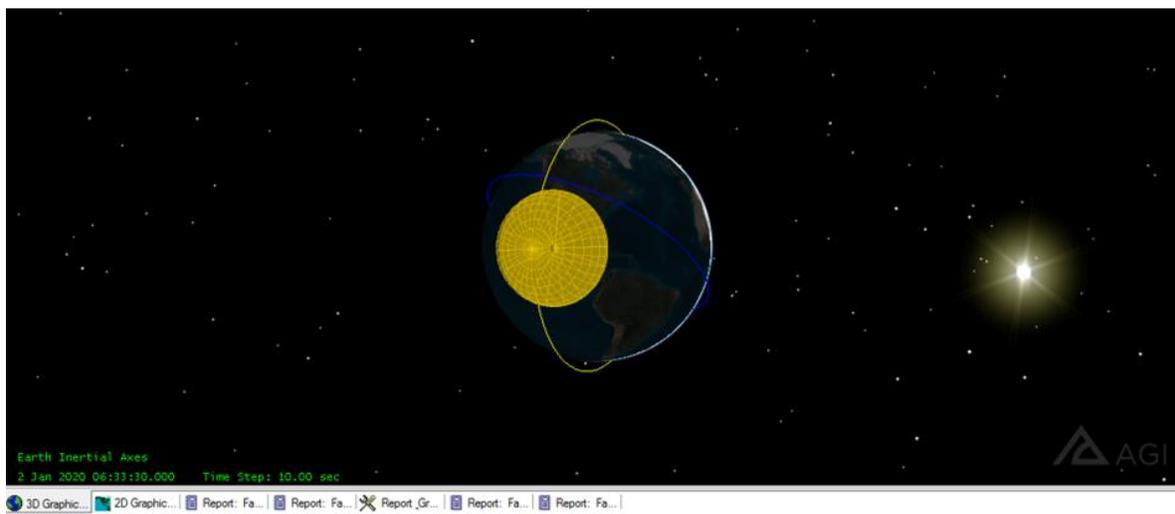


Figura 2.18. Mapa 3D del escenario configurado en STK.



### 2.2.3. Generación de reportes

Existen diversos tipos de reportes disponibles en STK. Para el Análisis de Comunicaciones, los reportes de pases por la ET y de rangos dados en ambas órbitas son de interés.

Para generar los reportes es muy importante considerar el tiempo de muestreo de la simulación. Por default un intervalo de 60 segundos está configurado pero estos saltos son muy pronunciados para detallar la variación de los rangos entre los satélites y la ET respecto del tiempo. En relación a las órbitas de la ISS y SSO, las velocidades a las cuales se desplazan los satélites puede variar entre los 4km/s y los 3km/s, por lo que en un lapso de tiempo de un minuto el rango variará en el orden de los cientos de kilómetros y los cálculos de enlace no podrían mostrar de manera precisa la variación de la potencia de la señal. Es por esto que al momento de correr las simulaciones se procura configurar el tiempo de muestreo en 1 segundo.

De aquí surge otra observación importante. Dado que el intervalo de simulación es igual a un año y el tiempo de muestreo de 1 segundo se tienen más de 31 millones de eventos a considerar (uno por cada segundo transcurrido) por la simulación por lo que el tiempo de procesamiento, dependiendo de las características del ordenador, puede llegar a extenderse por más de 5 horas para analizar ambas órbitas. Una recomendación es hacer un primer análisis con tiempos de muestreo de 60 segundos para un intervalo de tiempo de un par de meses y una vez que se garantiza que todas las configuraciones y resultados son coherentes correr las simulaciones detalladas.

En STK, el reporte que otorga el número y tiempo de pases se conoce como “**Access**” y en el que se encuentran los valores de los rangos es conocido como AER (Azimuth-Elevation-Range).

A continuación se brindan los pasos para generar ambos reportes:

- 1) Seleccionar **Analysis / Access**

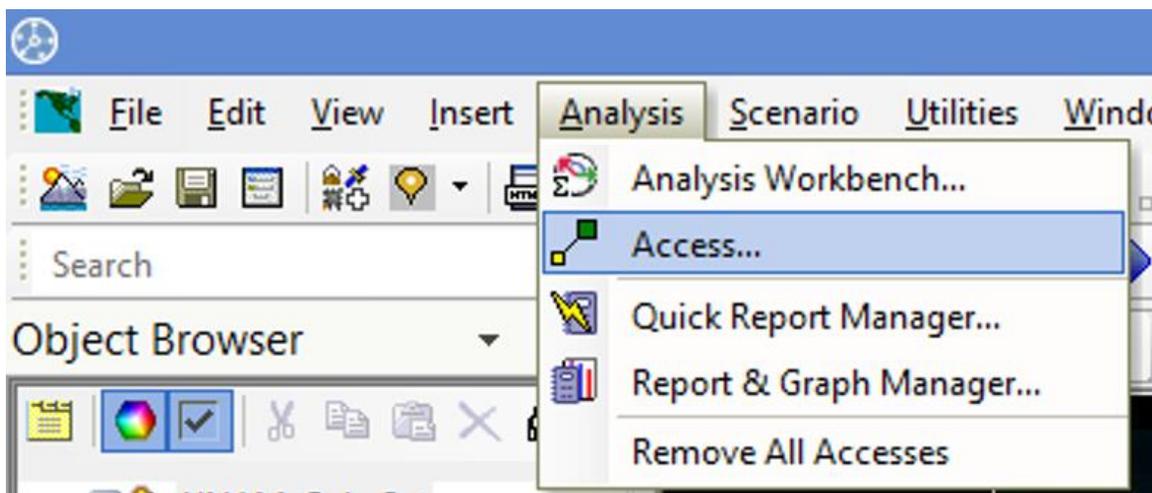


Figura 2.19 Selección de opción Access.



2) En la ventana desplegada (ver fig. 2.20), se selecciona al sensor asociado a la ET, en este caso UAT Juriquilla – ECXSAT como el punto de acceso (Access for). Se selecciona el objeto SSO Orbit y se da click en **“Report & Graph Manager”** en la parte inferior derecha de la ventana.

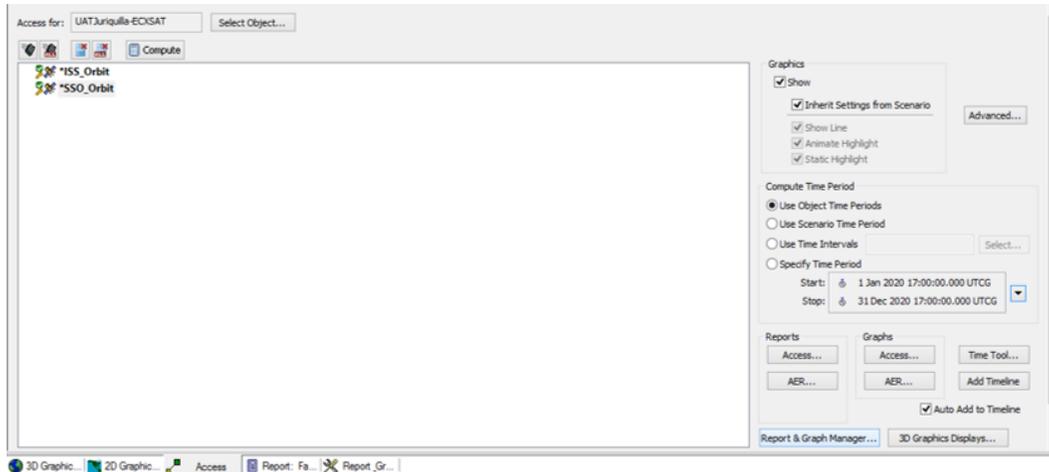


Figura 2.20. Configuración de la ventana Access para la órbita SSO.

3) Dentro del Report Manager en el campo **“Object Type”** se elige **“Access”**. En la lista inferior se elige el elemento asociado a la órbita que se desea analizar. Para el caso de la órbita SSO se selecciona **“Facility UAT Juriquilla ECXSAT to Satellite SSO Orbit”**.

En el campo de **“Time Properties”** se elige la opción **“Specify Time Properties”** donde se verifica que el periodo de la simulación es el correcto y se configura el tiempo de muestreo a 1 segundo en el rubro **“Use step size/time bound”**.

En la lista de reportes ubicada a la derecha de la ventana se selecciona AER y **“Access”** (utilizando el comando Ctrl+click en ambo reportes). Luego **“Generate as: Report/graph”** y se da click en **“Generate...”** como se muestra en la figura 2.21.

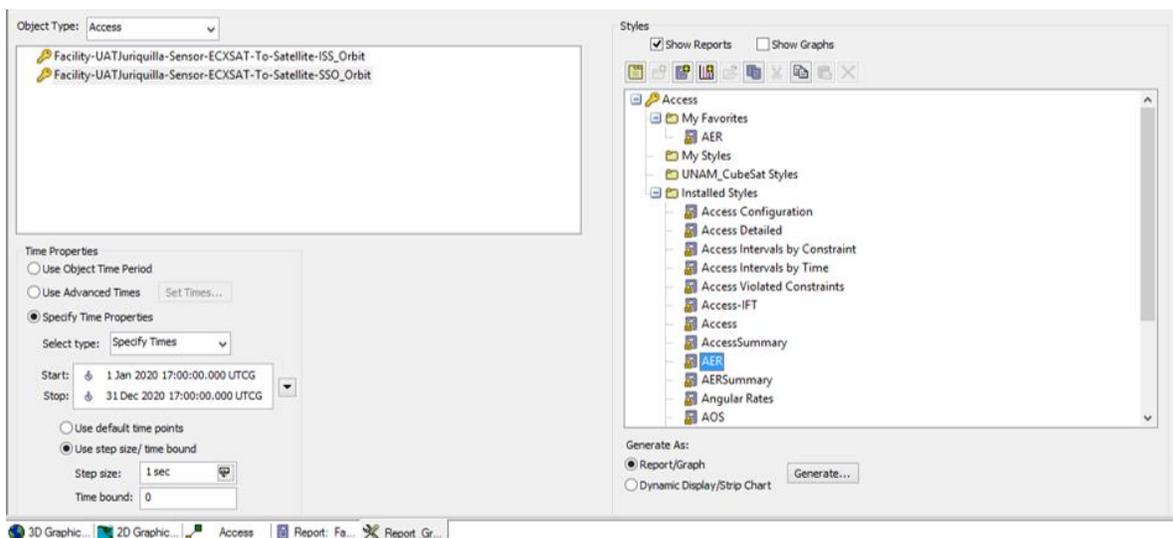


Figura 2.21. Report Manager para generar reporte AER de la órbita SSO



Una vez hecha la simulación es posible seleccionar el icono de “guardar/save as .csv” para poder exportar los reportes generados en un archivo CSV para su posterior análisis. Ver fig. 2.22.

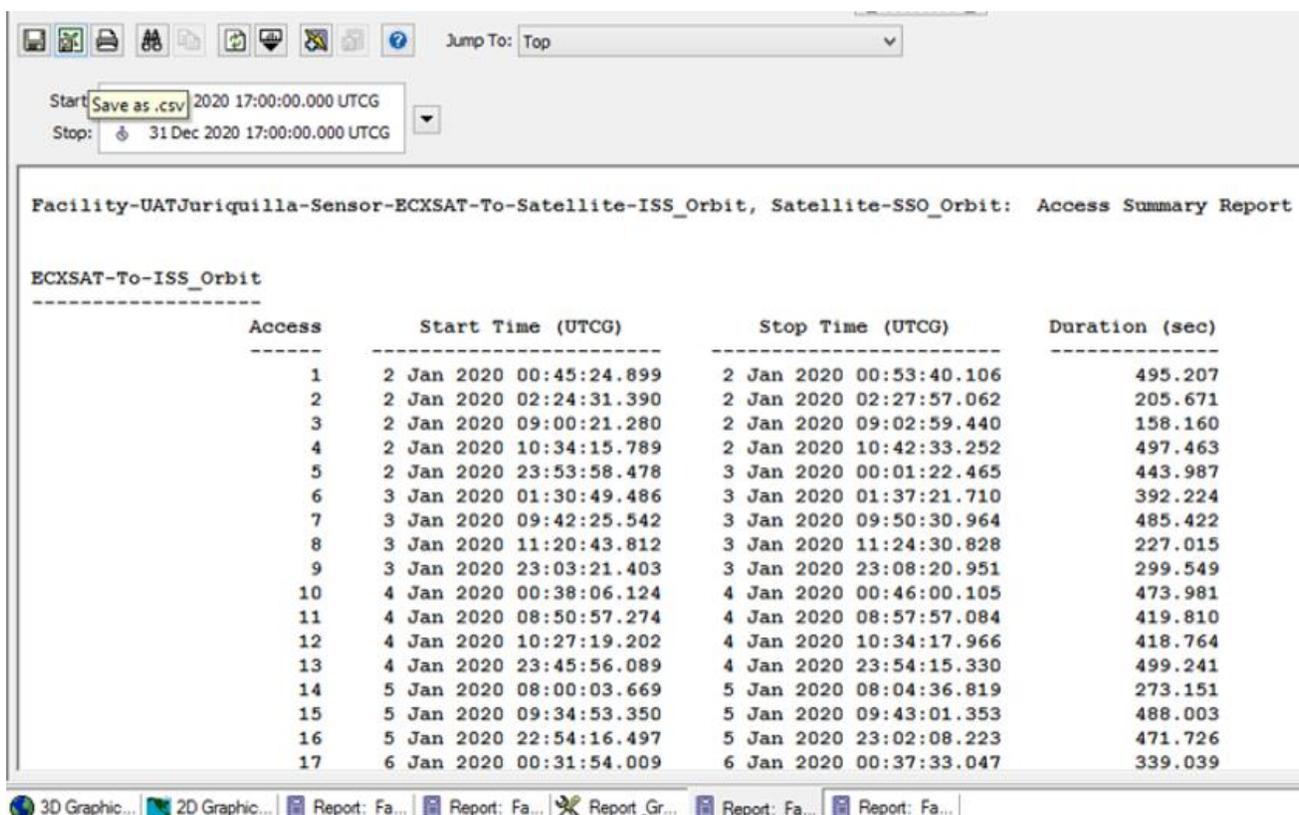


Figura 2.22 Reporte generado en STK.

4) Para generar el reporte de Acceso y rango de la órbita ISS repetimos los pasos del inciso anterior pero seleccionando la órbita ISS en el Report Manager como se muestra en la figura 2.23.

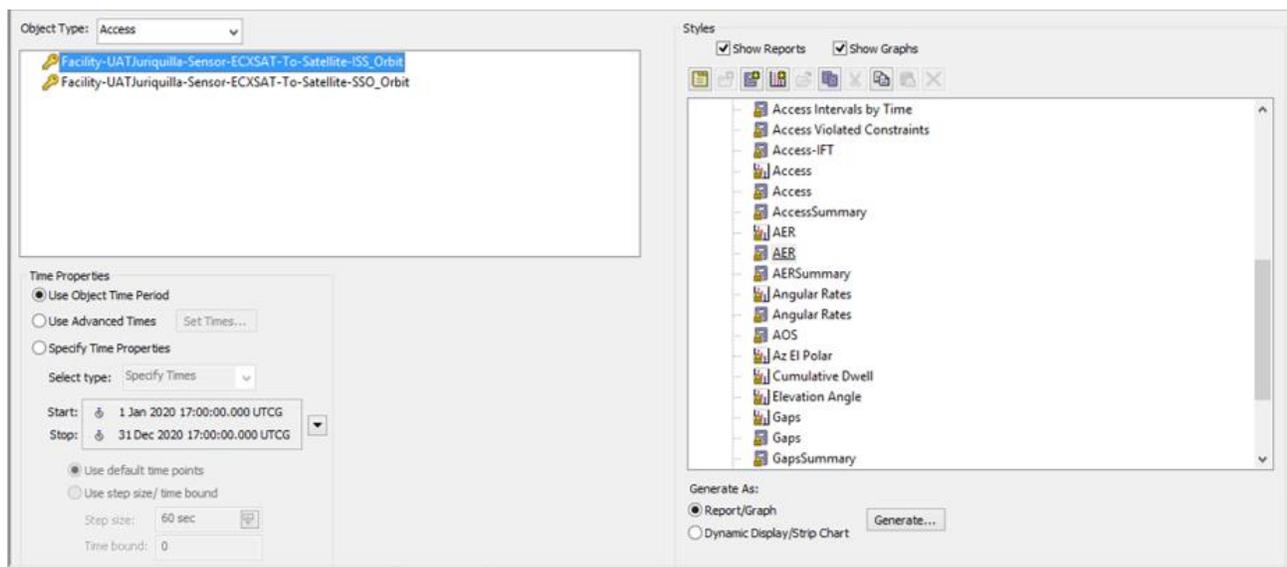


Figura 2.23. Report Manager para generar reporte AER de la órbita ISS.



Programa de Maestría y Doctorado en Ingeniería

De igual modo, se eligen los reportes Access y AER y se generan ambos reportes. Posteriormente, se guardan como archivo CSV.

Los datos de acceso presentados son el número de acceso, la fecha y hora de inicio y fin del acceso y la duración en segundos del mismo. En este caso se generan dos reportes, uno para la órbita de la ISS denominado ECXSAT-To-ISS\_Orbit (fig. 2.24) y otro para la órbita SSO denominado ECXSAT-To-SSO\_Orbit (fig. 2.25).

ECXSAT-To-ISS\_Orbit

Access	Start Time (UTCG)	Stop Time (UTCG)	Duration (sec)
1	2 Jan 2020 00:45:24.899	2 Jan 2020 00:53:40.106	495.207
2	2 Jan 2020 02:24:31.390	2 Jan 2020 02:27:57.062	205.671
3	2 Jan 2020 09:00:21.280	2 Jan 2020 09:02:59.440	158.160
4	2 Jan 2020 10:34:15.789	2 Jan 2020 10:42:33.252	497.463
5	2 Jan 2020 23:53:58.478	3 Jan 2020 00:01:22.465	443.987
6	3 Jan 2020 01:30:49.486	3 Jan 2020 01:37:21.710	392.224
7	3 Jan 2020 09:42:25.542	3 Jan 2020 09:50:30.964	485.422
8	3 Jan 2020 11:20:43.812	3 Jan 2020 11:24:30.828	227.015
9	3 Jan 2020 23:03:21.403	3 Jan 2020 23:08:20.951	299.549
10	4 Jan 2020 00:38:06.124	4 Jan 2020 00:46:00.105	473.981
11	4 Jan 2020 08:50:57.274	4 Jan 2020 08:57:57.084	419.810
12	4 Jan 2020 10:27:19.202	4 Jan 2020 10:34:17.966	418.764
13	4 Jan 2020 23:45:56.089	4 Jan 2020 23:54:15.330	499.241
14	5 Jan 2020 08:00:03.669	5 Jan 2020 08:04:36.819	273.151
15	5 Jan 2020 09:34:53.350	5 Jan 2020 09:43:01.353	488.003
16	5 Jan 2020 22:54:16.497	5 Jan 2020 23:02:08.223	471.726
17	6 Jan 2020 00:31:54.009	6 Jan 2020 00:37:33.047	339.039

Figura 2.24 Datos de acceso de la órbita ISS.

Facility-UATJuriquilla-Sensor-ECXSAT-To-Satellite-SSO\_Orbit: Access Summary Report

ECXSAT-To-SSO\_Orbit

Access	Start Time (UTCG)	Stop Time (UTCG)	Duration (sec)
1	1 Jan 2020 17:38:40.433	1 Jan 2020 17:47:47.679	547.246
2	1 Jan 2020 19:14:48.359	1 Jan 2020 19:22:39.800	471.442
3	2 Jan 2020 05:54:54.972	2 Jan 2020 06:05:15.320	620.348
4	2 Jan 2020 17:50:21.786	2 Jan 2020 18:00:08.001	586.216
5	2 Jan 2020 19:27:27.444	2 Jan 2020 19:33:48.635	381.191
6	3 Jan 2020 04:35:46.577	3 Jan 2020 04:37:55.925	129.348
7	3 Jan 2020 06:06:56.918	3 Jan 2020 06:17:11.657	614.739
8	3 Jan 2020 18:02:10.202	3 Jan 2020 18:12:19.874	609.671
9	3 Jan 2020 19:40:35.404	3 Jan 2020 19:44:26.259	230.856
10	4 Jan 2020 04:45:54.649	4 Jan 2020 04:51:32.545	337.895
11	4 Jan 2020 06:19:06.059	4 Jan 2020 06:29:02.292	596.233
12	4 Jan 2020 18:14:04.566	4 Jan 2020 18:24:24.335	619.769
13	5 Jan 2020 04:56:55.515	5 Jan 2020 05:04:18.783	443.268
14	5 Jan 2020 06:31:23.191	5 Jan 2020 06:40:46.348	563.157
15	5 Jan 2020 18:26:04.362	5 Jan 2020 18:36:21.821	617.460
16	6 Jan 2020 05:08:14.988	6 Jan 2020 05:16:48.637	513.649
17	6 Jan 2020 06:43:49.977	6 Jan 2020 06:52:22.108	512.131

Figura 2.25 Datos de acceso de la órbita SSO.



### Programa de Maestría y Doctorado en Ingeniería

De igual modo, al final de cada reporte se brinda un resumen estadístico de toda la simulación. En este resumen se obtienen datos como los valores mínimo, máximo y promedio de la duración de los pases, así como el tiempo acumulado de todos los pases registrados. Las estadísticas globales de ambas órbitas se muestran en las figuras 2.26 y 2.27.

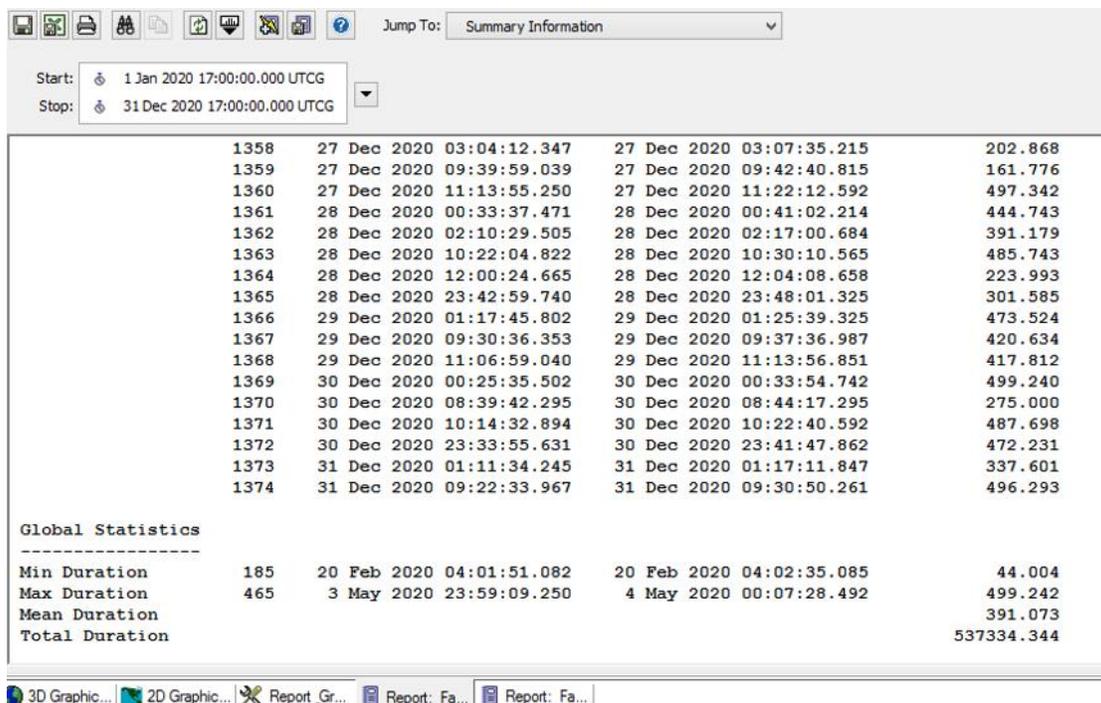


Figura 2.26. Estadísticas globales del acceso ECXSAT-To-ISS\_Orbit.

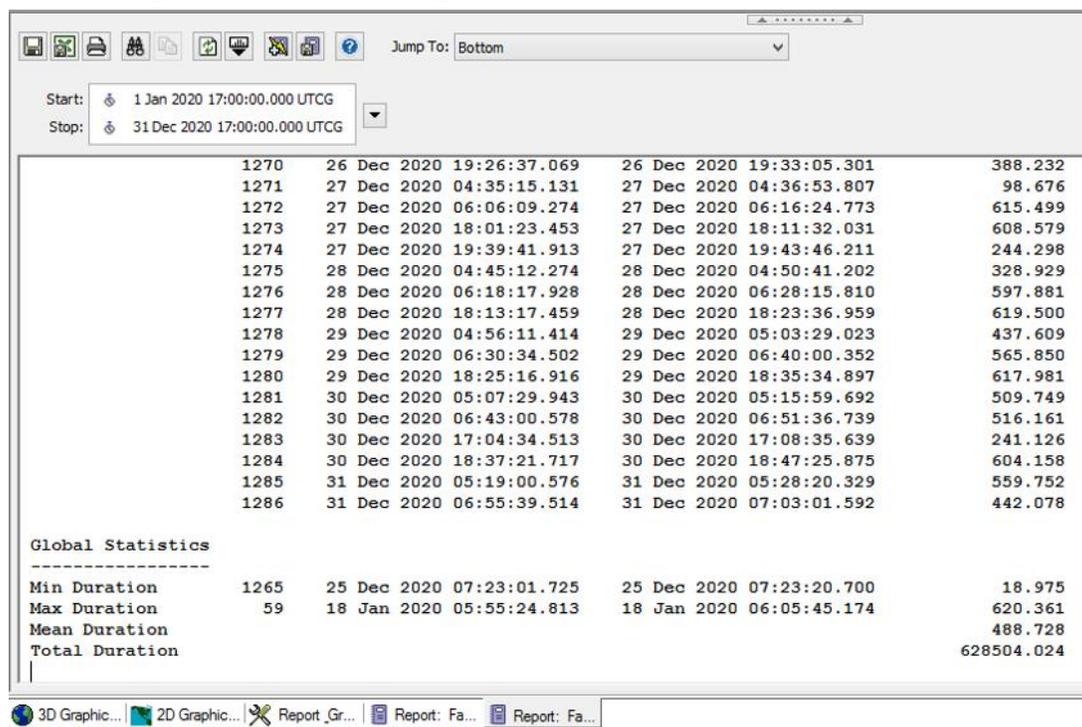


Figura 2.27. Estadísticas globales del acceso ECXSAT-To-SSO\_Orbit.



En cuanto al reporte AER los parámetros obtenidos son tiempo y hora de medición, valor de azimut, ángulo de elevación y rango obtenido para ese instante de tiempo. En la figura 2.28 se muestra el formato en csv generado por el reporte de la órbita ISS. Todos los reportes mencionados pueden exportarse a este tipo de formato lo cual facilita la manipulación y uso de datos para posteriores análisis.

Time (UTCG)	Azimuth (deg)	Elevation (deg)	Range (km)
2 Jan 2020 00:45:24.899	210.926	5	1814.390323
2 Jan 2020 00:45:25.000	210.923	5.009	1813.700973
2 Jan 2020 00:45:26.000	210.896	5.09	1806.8481
2 Jan 2020 00:45:27.000	210.868	5.172	1799.996506
2 Jan 2020 00:45:28.000	210.84	5.254	1793.146214
2 Jan 2020 00:45:29.000	210.812	5.336	1786.297247
2 Jan 2020 00:45:30.000	210.784	5.419	1779.449629

Figura 2.28. Reporte AER correspondiente a la órbita ISS exportado en .csv



### 2.3 ANÁLISIS DE RESULTADOS

Como se comentó previamente se obtienen dos tipos de datos de la simulación en STK: los accesos o pases existentes entre la ET y los satélites y el valor de los rangos de línea de vista. Se comenzará presentado la tabla 2.3 con un resumen de las estadísticas globales de ambos reportes para ambas órbitas.

**Tabla 2.3. Estadísticas globales de los reportes obtenidos para las órbitas ISS y SSO.**

Parámetro	Órbita ISS	Órbita heliosíncrona
Ángulo respecto al Sol	Variable	Constante
Altitud	406 km	600 km
Inclinación	51.6°	-98°
Pases por año	1374	1286
Tiempo de pase (min/máx/avg)	44/499/391 s	19/620/488 s
Tiempo acumulado de pases/año	149.25h	174.58h
Rango (mín/máx/avg)	406/1,834/1,274 km	600/2,342/1,643 km

De las estadísticas globales se observa que la órbita que tiene más pases por encima de la ET en un año es la ISS con 1374 pases, sin embargo el tiempo promedio de los pases registrados por la órbita SSO es mayor por casi 1.5 minutos que los de la órbita ISS. Esto hace que el tiempo de pases acumulados en la órbita ISS sea de 149.25 horas, mientras que en la SSO de 24 horas más.

Esto podría hacer pensar que la órbita más conveniente, en términos de tiempo de acceso, es la SSO sin embargo aún falta considerar los rangos generados y su impacto en la potencia de la señal portadora transmitida a Tierra. Si se observan los valores promedios de los rangos de ambas órbitas se encuentra una diferencia importante de aproximadamente 400km entre estos. La órbita ISS tiene un valor promedio de 1,274 km, mientras que la SSO de 1,643 km. Esta diferencia de rangos, en términos de velocidad de transmisión podría hacer que la órbita ISS resulte con más ventaja respecto a la SSO. Para ello se analiza el ModCod vs SNR en el siguiente capítulo.

A continuación se presentan algunas gráficas generadas para ambos reportes tanto en STK como en Matlab después de haber procesado los reportes obtenidos.

### 2.3.1. Gráficas de Accesos

Un tipo de gráfica generada por STK es el tiempo de acceso mostrada en la figura 2.29. En ella, se observan en rojo y verde la cantidad y duración de cada acceso para ambas órbitas, la ISS y SSO respectivamente. En esta primera gráfica el periodo mostrado es el de todo 2020. Se puede apreciar que la cantidad y periodo de los pases es continuo, es decir, existen pases por encima de la ET durante todo el año para ambas órbitas.

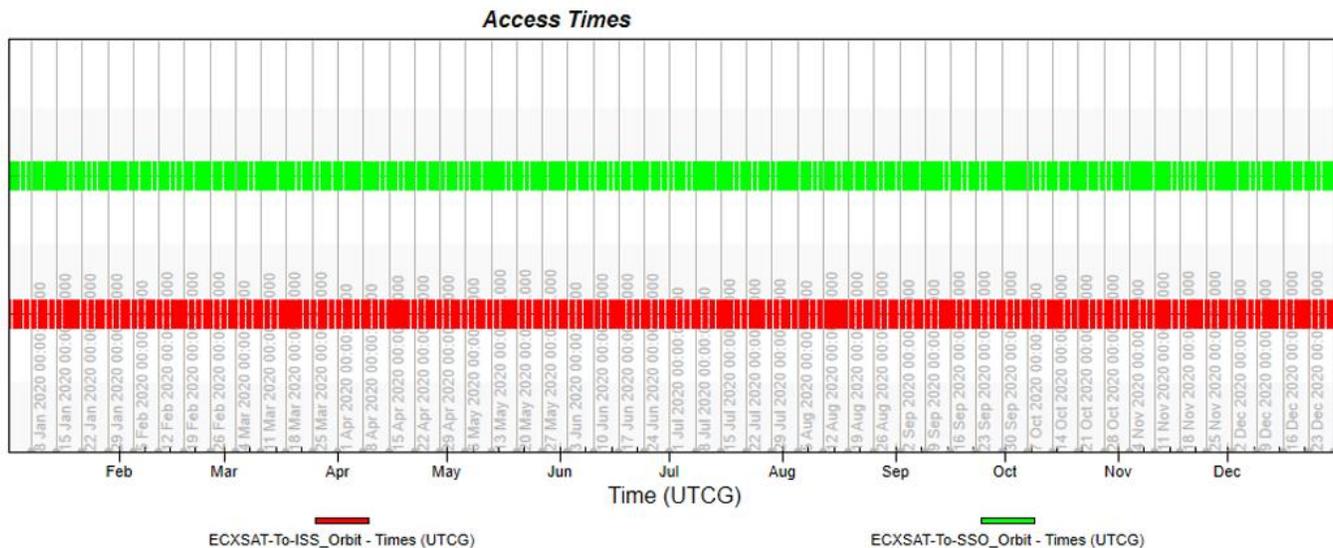


Figura 2.29. Accesos registrados por las órbitas ISS y SSO de enero a diciembre de 2020.

Si se aplica un acercamiento a la misma obtenemos la gráfica de la figura 2.30. Para esta se ha seleccionado el mes de enero de 2020 para ser visualizado. De este modo, se puede apreciar con más detalle el número de pases registrados durante este mes.

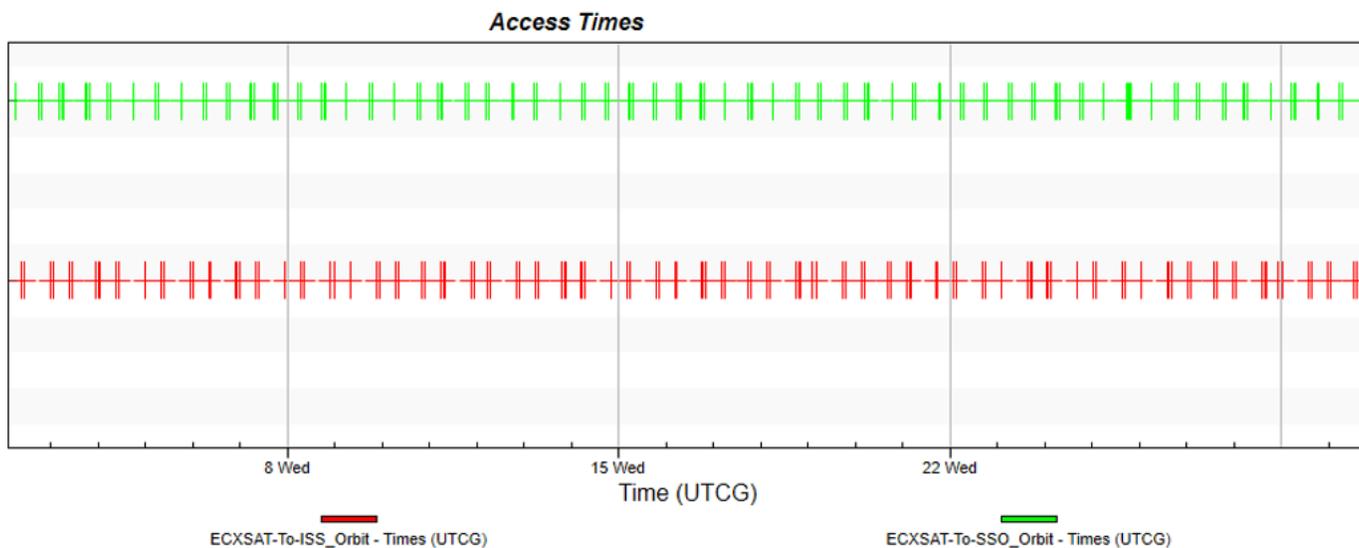


Figura 2.30. Accesos registrados por las órbitas ISS y SSO correspondientes a enero 2020.

Otro tipo de gráfica que puede ser de mucha utilidad es el histograma del tiempo de duración de todos los pases registrados en ambas órbitas durante todo el año 2020. Para generarlo se hace uso del archivo de accesos exportado en formato csv y procesado utilizando Matlab. Con apoyo de los histogramas se tiene información más precisa respecto al



tiempo de duración de los diversos pases de los satélites en las órbitas ISS y SSO y la frecuencia con la que estos suceden. En las figuras 2.31 y 2.32 se observa que el tiempo de duración de los pases por lo general van en incremento hasta llegar a un valor máximo de 500 segundos para la órbita ISS y de 615 segundos para la SSO.

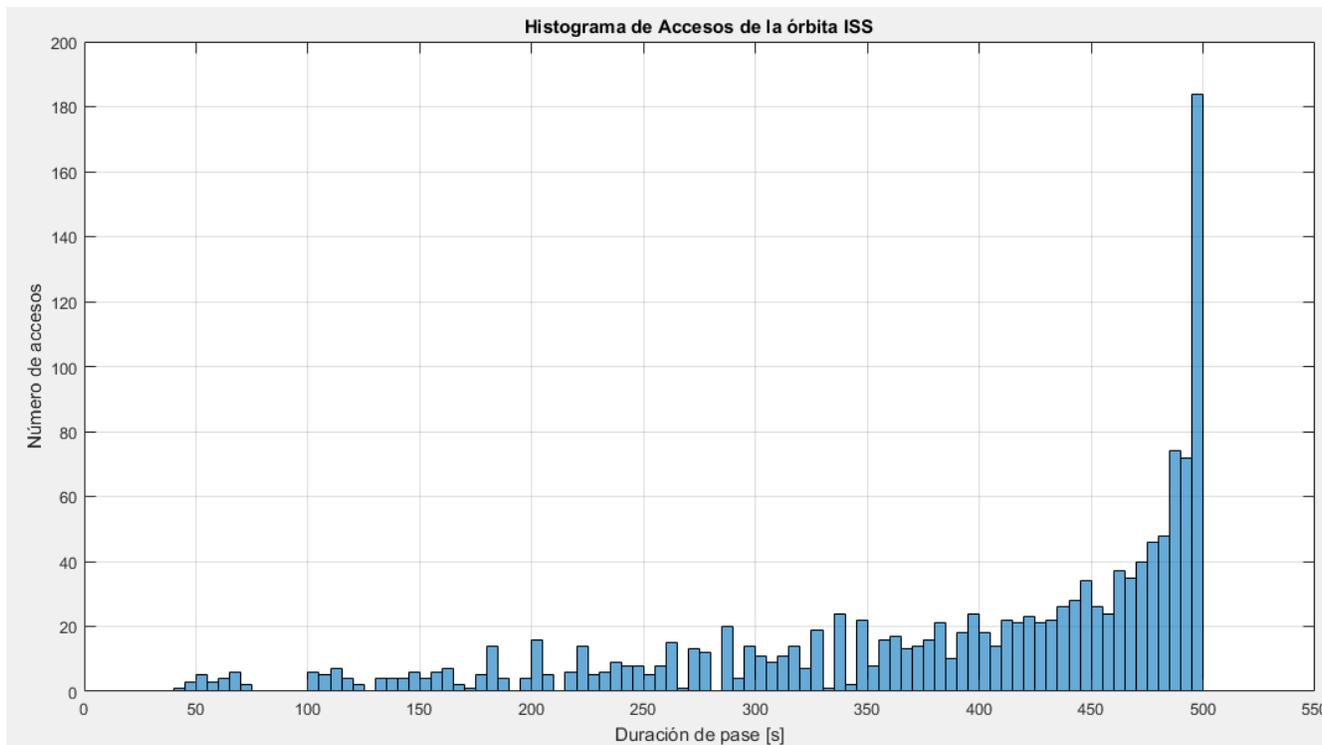


Figura 2.31. Histograma de los tiempos de acceso para la órbita ISS.

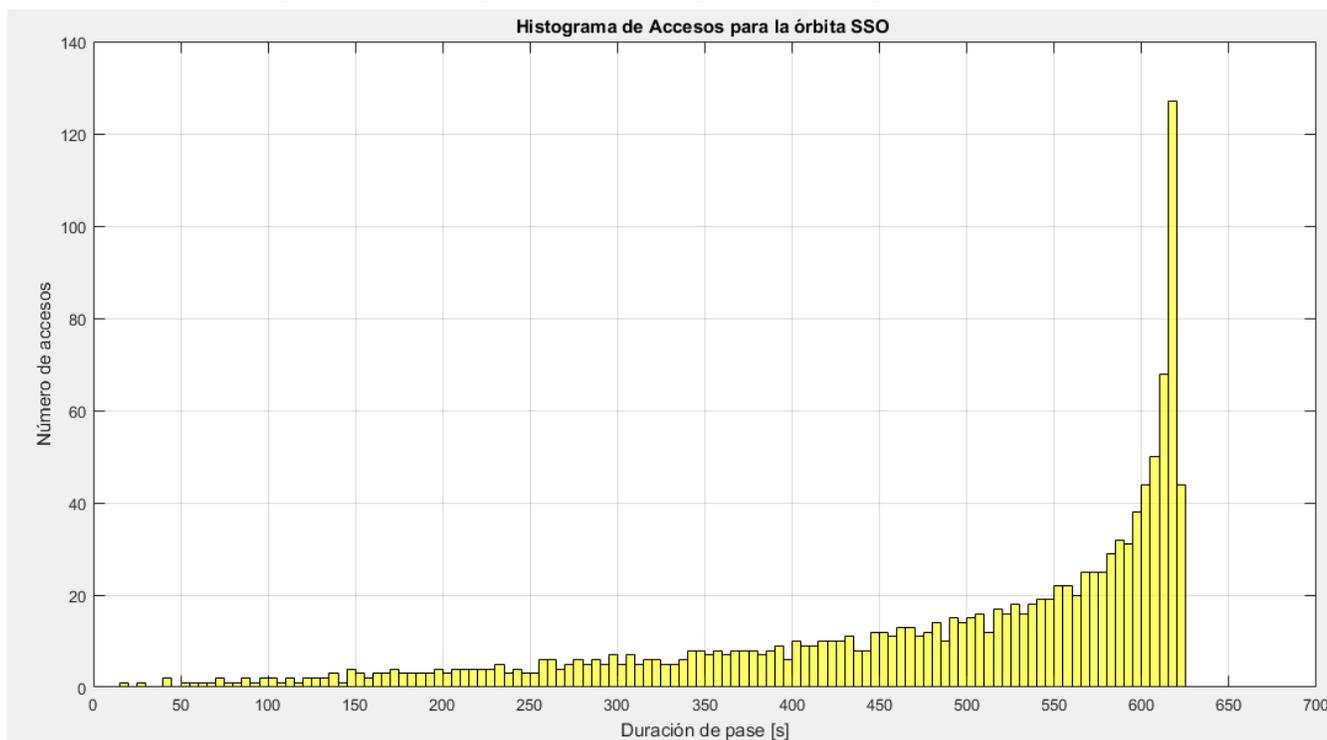


Figura 2.32. Histograma de los tiempos de acceso para la órbita SSO.



Si se posee algún tiempo de duración mínimo para realizar un procedimiento durante la misión satelital, con este análisis se logra verificar la cantidad de pases viables.

Por dar un ejemplo, suponiendo que el tiempo necesario para recibir telemetría del satélite puede ser de 90 segundos con modulación QPSK  $\frac{1}{2}$ , lo cual descartaría los pases en la parte izquierda de la gráfica por no ser lo suficientemente largos.

### 2.3.2 Gráficas de Rangos

Para los rangos obtenidos se realizan, de igual modo, los histogramas para ambas órbitas. Estos son mostrados en las figuras 2.33 y 2.34.

Se observa que la frecuencia de los rangos va en orden ascendente, es decir, los rangos mínimos son los menos frecuentes mientras que los máximos son los más frecuentes. El incremento, de igual modo, es cuasi lineal lo que habla de la excentricidad de la órbita, en ambos casos cercana al cero, por lo que las distancias en el perigeo y el apogeo son prácticamente las mismas.

Para la ISS los rangos máximos rondan los 1800 km con una frecuencia de 12,000 eventos al año. Por su parte, los rangos de la órbita SSO alcanzan un valor máximo de 2,300 km con una frecuencia por debajo de los 12,000 eventos al año. La visualización de ambos histogramas permite considerar con antelación el impacto de las Pérdidas en el Espacio Libre generadas por la propagación de las ondas electromagnéticas.

De manera concreta, considerando la órbita ISS se tiene un rango mínimo de 400 km con 2,000 eventos (la altura orbital en realidad) y uno máximo de 1800 km con 12,000 eventos. El crecimiento en el número de eventos, partiendo de los 400km, aumenta de manera lineal conforme aumenta el valor del rango.

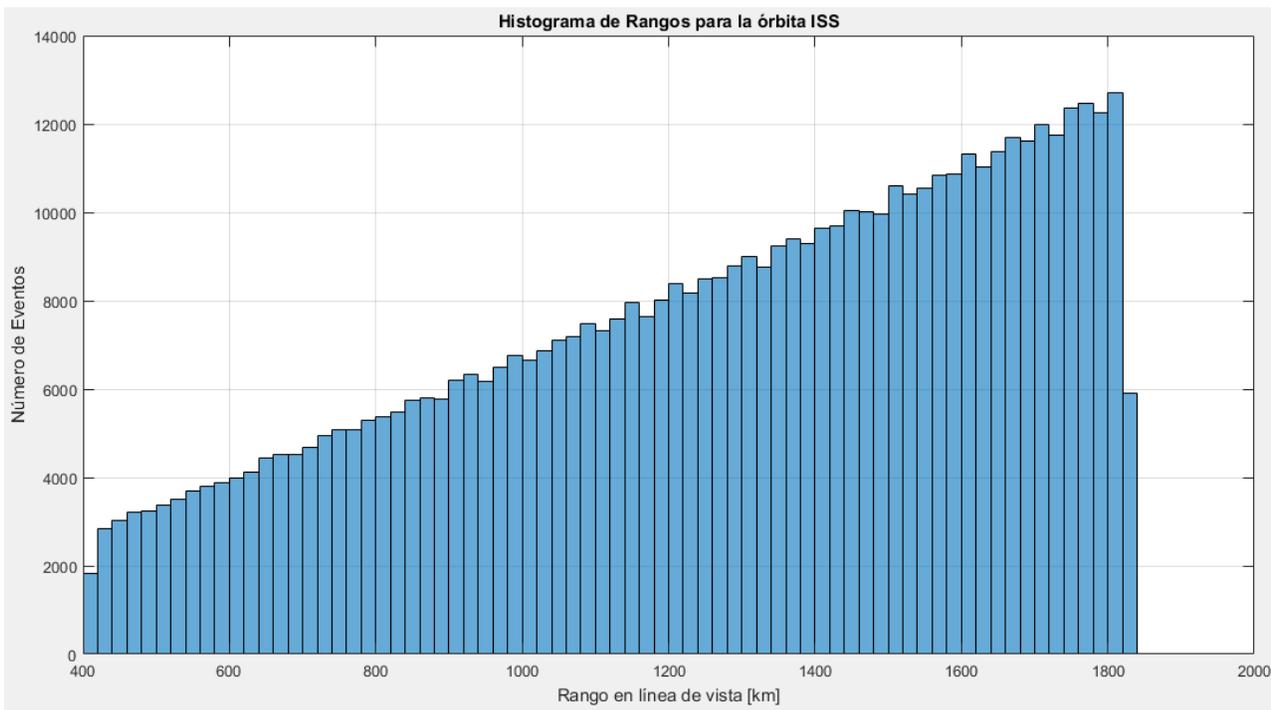
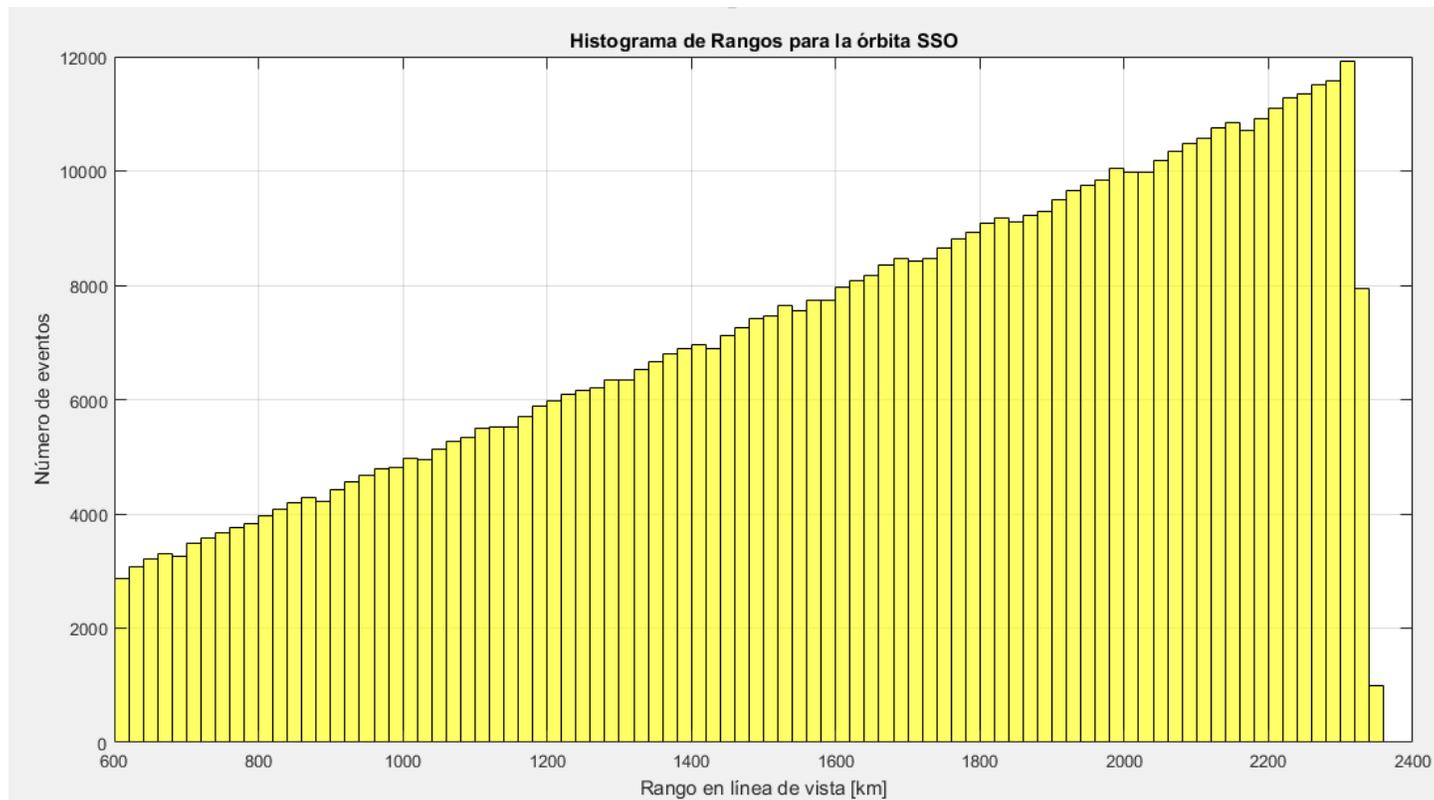


Figura 2.33. Histograma de los rangos para la órbita ISS.

Para la órbita SSO se tiene un rango mínimo de 600 km con 3,000 eventos aproximadamente y uno máximo de 2,300 km con 12,000 eventos. El crecimiento en el número de eventos también aumenta de manera lineal conforme aumenta el valor del rango.



**Figura 2.34. Histograma de los rangos para la órbita SSO.**

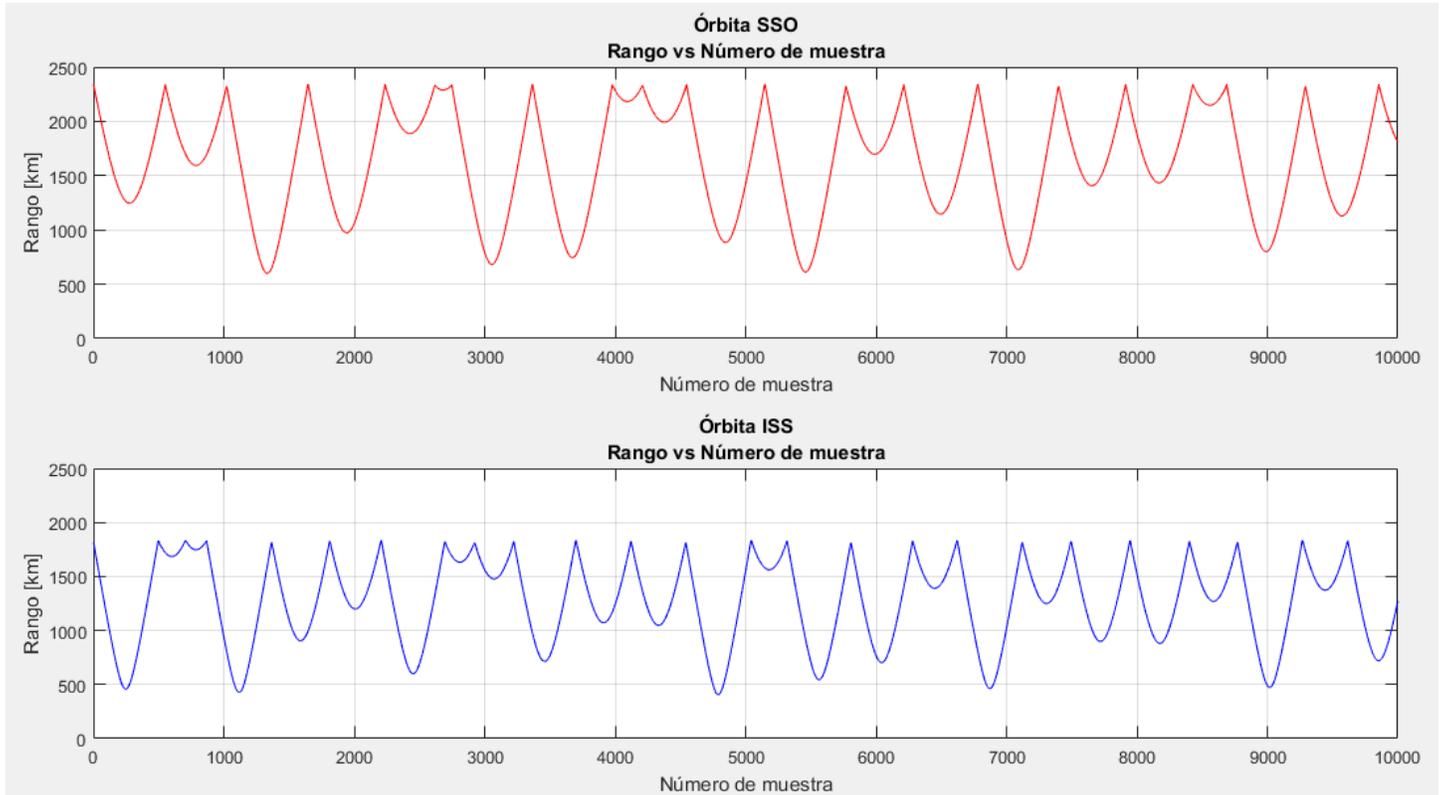
De esto se puede deducir que el comportamiento en el valor del SNR se comportará de manera muy similar al comportamiento de la variación en los rangos, sólo que en sentido inverso debido a que mientras mayor es el rango menor es la potencia de la señal en el receptor.

También se considera el comportamiento de la variación del rango respecto al tiempo, o como se presenta en las figuras 2.5 y 2.36, respecto a una secuencia de muestras.

En ambas gráficas se observan valles y crestas en el valor de los rangos para ambas órbitas. Se puede interpretar una cresta como el punto en el que el satélite entra o sale de línea de vista con la ET y el valle como el punto de máximo acercamiento entre ellos. De las gráficas se comprueba la razón por la que los rangos máximos son más frecuentes pues en un solo pase el satélite debe de adquirir el valor de rango máximo un par de veces.

Para la representación de estas gráficas sólo se considera las primeras 1000 muestras registradas para ambas órbitas. Los eventos, si bien llevan un orden cronológico, no consideran los eventos en los que no hay línea de vista por lo que se encuentra una orden de eventos continuos y sin interrupciones.

Nuevamente se comprueba que la frecuencia de avistamientos de la órbita SSO es menor que la de la ISS pero la duración de los eventos de la órbita SSO son más pronunciados.



**Figura 2.35. Gráfica Rango vs Número de muestra de las órbitas SSO e ISS.**

Con este análisis previo y con los datos obtenidos, particularmente el valor de los rangos, se procede a realizar el análisis de comunicaciones.



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



## CAPÍTULO 3. ANÁLISIS DE COMUNICACIONES

Con los rangos obtenidos con apoyo de STK se realiza un análisis enfocado al sistema de comunicaciones considerando el cálculo de enlace del sistema, así como los valores de SNR, BER, Es/No y umbrales de conmutación ACM. Para estos análisis también se utilizó Matlab como una herramienta complementaria.

El resultado final del análisis es la caracterización del canal de transmisión enfocado a la variación del ModCod seleccionado dependiendo del SNR alcanzada en un instante de tiempo determinado. Este resultado se relaciona directamente con el número de imágenes descargadas en un periodo de misión equivalente a un año, haciendo una comparativa entre el uso de Modulación y Codificación Adaptativa contra un esquema fijo (Constant Coding and Modulation).

El orden de pasos a seguir durante el análisis es el siguiente:

1. Establecer las diferencias entre BER, Eb/No, Es/No y SNR.
2. Definir los valores de conmutación ACM para los diferentes ModCod disponibles.
3. Implementar el procedimiento de Cálculo de Enlace en Matlab.
4. Procesar los rangos obtenidos en STK y asignarles un valor de SNR utilizando el programa de cálculo de enlace anterior.
5. Asociar los valores de SNR a un ModCod determinado con ayuda de los umbrales determinados.
6. Calcular la velocidad de transmisión alcanzada para todos los eventos de línea de vista entre los satélites y la ET.
7. Obtener la capacidad del canal de transmisión utilizando ACM.
8. Obtener la capacidad del canal de transmisión utilizando CCM.
9. Comparar ambos esquemas y analizar los alcances y viabilidad de cada uno de ellos en torno a la misión planteada.

### 3.1. DIFERENCIAS ENTRE BER, Eb/No, Es/No y SNR

El método ACM basa sus beneficios en la capacidad de conmutar entre diferentes órdenes de modulación y codificación dependiendo del nivel de la señal a la entrada del receptor del segmento terrestre. Para establecer los niveles de conmutación de manera óptima diferentes conceptos y variables entran en juego. De estos, los más representativos son:

- BER (Bit Error Rate)
- Eb/No (Energy per bit to Spectral Noise Density Ratio)
- Es/No (Energy per symbol to Spectral Noise Density Ratio)
- SNR (Signal to Noise Ratio)

Todos estos parámetros están relacionados entre sí y son utilizados como punto de referencia en el esquema ACM para determinar qué combinación de modulación y codificación es la mejor.

A continuación se hará una breve descripción de estos.



- **BER**

El Bit Error Rate o la Tasa de Errores de Bit es un parámetro que caracteriza un canal de comunicaciones asignándole un valor que representa cuántos errores se generan en este canal al pasar información por el mismo.

El BER se define como:

$$BER = \frac{\text{Bits con error}}{\text{Bits Totales}} \quad (3.1)$$

Y se trata de un valor adimensional.

Usualmente se le encuentra expresado de la forma  $BER = 10^{-n}$  conforme a la tabla 3.1.

**Tabla 3.1. Valores de Bit Error Rate.**

BER	BER	Bits con error	Bits Totales	% de error
<b>1</b>	1	1	1	100%
<b>10<sup>-1</sup></b>	0.1	1	10	10%
<b>10<sup>-2</sup></b>	0.01	1	100	1%
<b>10<sup>-3</sup></b>	0.001	1	1,000	0.1%
<b>10<sup>-4</sup></b>	0.0001	1	10,000	0.01%
<b>10<sup>-5</sup></b>	0.00001	1	100,000	0.001%
<b>10<sup>-6</sup></b>	0.000001	1	1,000,000	0.0001%
<b>10<sup>-7</sup></b>	0.0000001	1	10,000,000	0.00001%

En comunicaciones, un valor adecuado de BER es  $10^{-6}$  o  $10^{-7}$  pues con estas tasas de error es posible corregir los bits erróneos con las técnicas de corrección de errores.

Para DVB-S2 el BER de referencia es  $10^{-7}$ .

- **SNR,  $E_b/N_0$ ,  $E_s/N_0$**

Para describir estos conceptos se considera la relación entre la potencia de la señal (P) y la potencia de ruido (N) medida en la salida o entrada de un sistema de radiofrecuencia. Dicho esto, se puede establecer la relación P/N y dado que la unidad de potencia son los [watts] que equivale a [joules/s], se entiende que la potencia relaciona energía con el tiempo. En términos de bit o de símbolo se puede proyectar esta idea y tener las siguientes expresiones:

$$P = \frac{E_s}{T_s} = \frac{E_b}{T_b} \quad (3.2)$$

Donde el equivalente a la energía [joules] es la Energía de símbolo [ $E_s$ ] o la Energía de bit [ $E_b$ ] y las unidades de tiempo corresponden al periodo del símbolo [ $T_s$ ] y el periodo del bit [ $T_b$ ].

Con ello, también se puede expresar la tasa de símbolos y la tasa de bits como:



$$R_s = 1/T_s \tag{3.3}$$

$$R_b = 1/T_b \tag{3.4}$$

En relación a la potencia de ruido, este es expresado en el dominio de la frecuencia. La densidad espectral de ruido es constante en todas las frecuencias y es provocada por el ruido térmico presente, principalmente, en los Amplificadores de Bajo Ruido (Low Noise Amplifier). Se representa como:

$$N_0 = kT \text{ [watt/Hz]} \tag{3.5}$$

Donde,

k es la constante de Boltzman=1.38x10<sup>-23</sup> J/K

y T es la temperatura de ruido en [kelvin]

Si el ancho de banda del receptor es BW entonces se tiene:

$$N = N_0(BW) \tag{3.6}$$

y suponiendo que la transmisión es ideal, es decir no hay pérdidas, entonces BW es equivalente a Rs, por lo que

$$N = N_0(BW) = N_0R_s \tag{3.7}$$

Si el ancho de banda del ruido es mayor que el valor óptimo de Rs entonces comenzará a haber pérdidas en el sistema y será necesaria una mayor potencia de la señal para mantener el mismo desempeño.

Igualando las expresiones anteriores se encuentra otra manera de expresar la relación P/N:

$$\frac{P}{N} = \frac{E_s}{N_0} = \frac{E_b R_b}{N_0 R_s} \tag{3.8}$$

El SNR se expresa como:

$$SNR = 10 \log_{10} \frac{P}{N} \text{ [dB]} \tag{3.9}$$

Al SNR también se le conoce como C/N (Carrier to Noise Ratio)

$$SNR = C/N \text{ [dB]} \tag{3.10}$$

Y de manera análoga se expresa la Relación Portadora – densidad espectral de ruido

$$\frac{C}{N} BW = \frac{C}{N_0} \text{ [dB - Hz]} \tag{3.11}$$



### 3.1.1. Análisis BER vs Eb/No

Como se mencionó anteriormente, el método ACM basa sus beneficios en la capacidad de conmutar entre diferentes órdenes de modulación y codificación dependiendo de la SNR recibida en el segmento terrestre y el valor de BER mínimo esperado. Mientras mayor sea la relación SNR, mayor será el orden de la modulación. De lo contrario, mientras el valor SNR disminuya el orden de modulación deberá ser más robusta para garantizar así la comunicación.

Como primer análisis, se plantea una comparativa BER (Bit Error Rate) vs Eb/No en los esquemas de modulación QPSK y 8PSK. Este primer análisis se hace sin considerar ninguna tasa de codificación. Se realiza el cálculo para M=4 y 8, correspondientes a QPSK y 8PSK, desde Matlab y se genera la siguiente gráfica.

Para el cálculo del BER de una modulación M-PSK se cuenta con la siguiente expresión [3\_1]:

$$BER_{M-PSK} = \frac{1}{m} \operatorname{erfc} \left( \sqrt{m \frac{E_b}{N_o}} \sin \frac{\pi}{M} \right) \tag{3.12}$$

Donde,

M: Orden de modulación

m: bits por símbolo

erfc: Función de error complementaria de Gauss

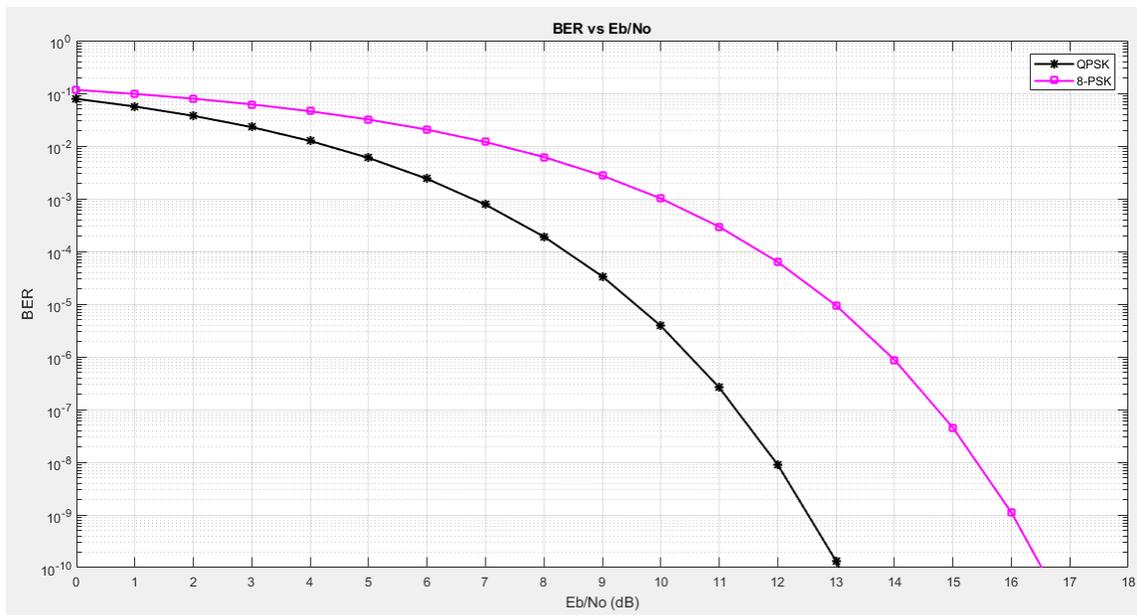


Figura 3.1. BER vs Eb/No de las modulaciones QPSK y 8PSK.

En la figura 3.1 se observa que para un BER de  $10^{-7}$  el Eb/No es aproximadamente de 11 y 15 dB para QPSK y 8PSK respectivamente. Como se esperaba, el valor de Eb/ No relacionado a un valor BER de referencia es mayor conforme el orden de la modulación se incrementa. En este ejemplo hay una diferencia de 4 dB entre un orden de modulación y la

siguiente. De igual modo, si se desean alcanzar valores de BER más cercanos a cero, el nivel de potencia requerido también aumenta. El código generado para desarrollar este cálculo se encuentra en el **Anexo A1**.

### 3.1.2. DVB-S2 en Simulink

El programa Simulink de Matlab tiene dentro de sus librerías una simulación del estándar DVB-S2 con las modulaciones QPSK y 8PSK implementadas con diferentes tasas de codificación. No se implementa un bloque de conmutación pero aun así se puede realizar un estudio de BER vs  $E_s/N_0$  con los resultados de esta simulación. El diagrama de bloques de DVB-S2 en Simulink es mostrado en la figura 3.2.

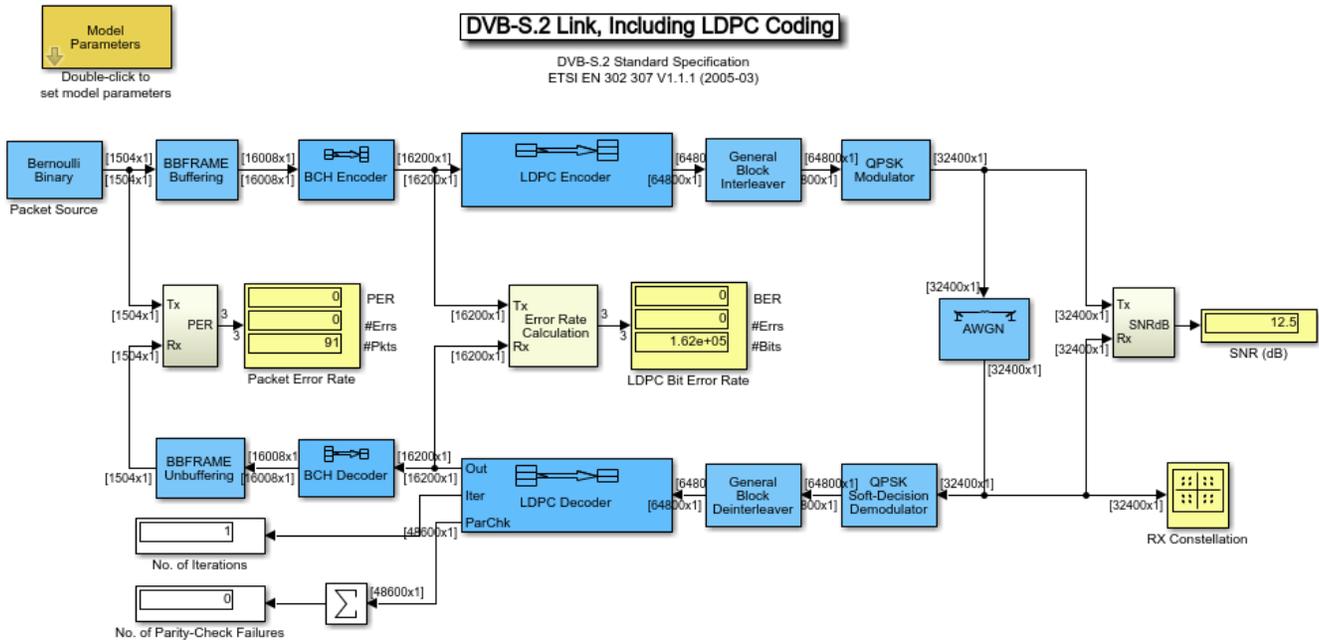


Figura 3.2. Diagrama de bloques DVB-S2 implementado en Simulink.

El diagrama de bloques cuenta con los módulos de Mode Adaptation, FEC Encoding y Modulación del estándar DVB-S2. Se consideran los bloques tanto de transmisión como de recepción. De igual modo, se puede caracterizar el canal con un valor de AWGN (Additive White Gaussian Noise) para simular pérdidas en el espacio libre. Los resultados entregados son los valores de PER (Packet Error Rate) y BER generados al igual que el mapa de constelación correspondiente. La simulación puede configurarse para implementar cualquiera de los ModCod asociados a las modulaciones QPSK y 8PSK y sus respectivas tasas de codificación como se muestra en la figura 3.3.

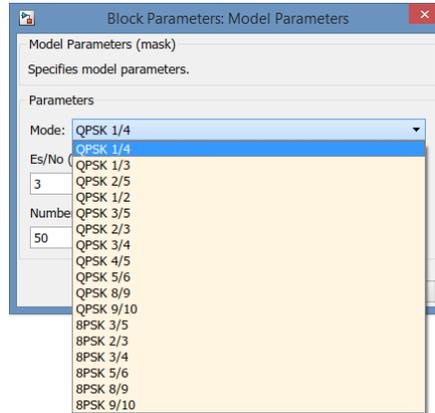


Figura 3.3. Selección de la modulación y codificación en Simulink.

En la figura 3.4 se muestran las constelaciones obtenidas para algunos de los ModCod disponibles:

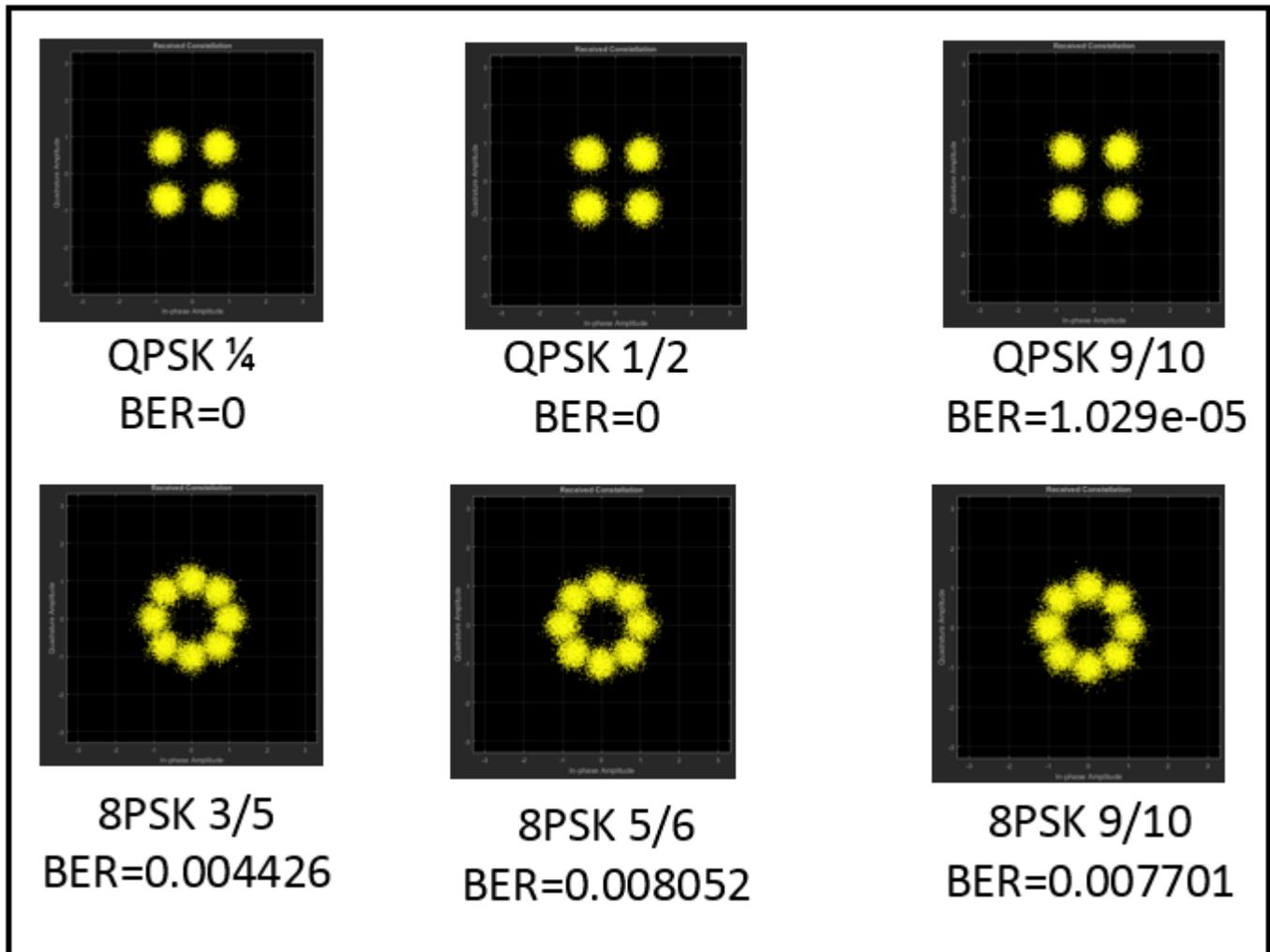
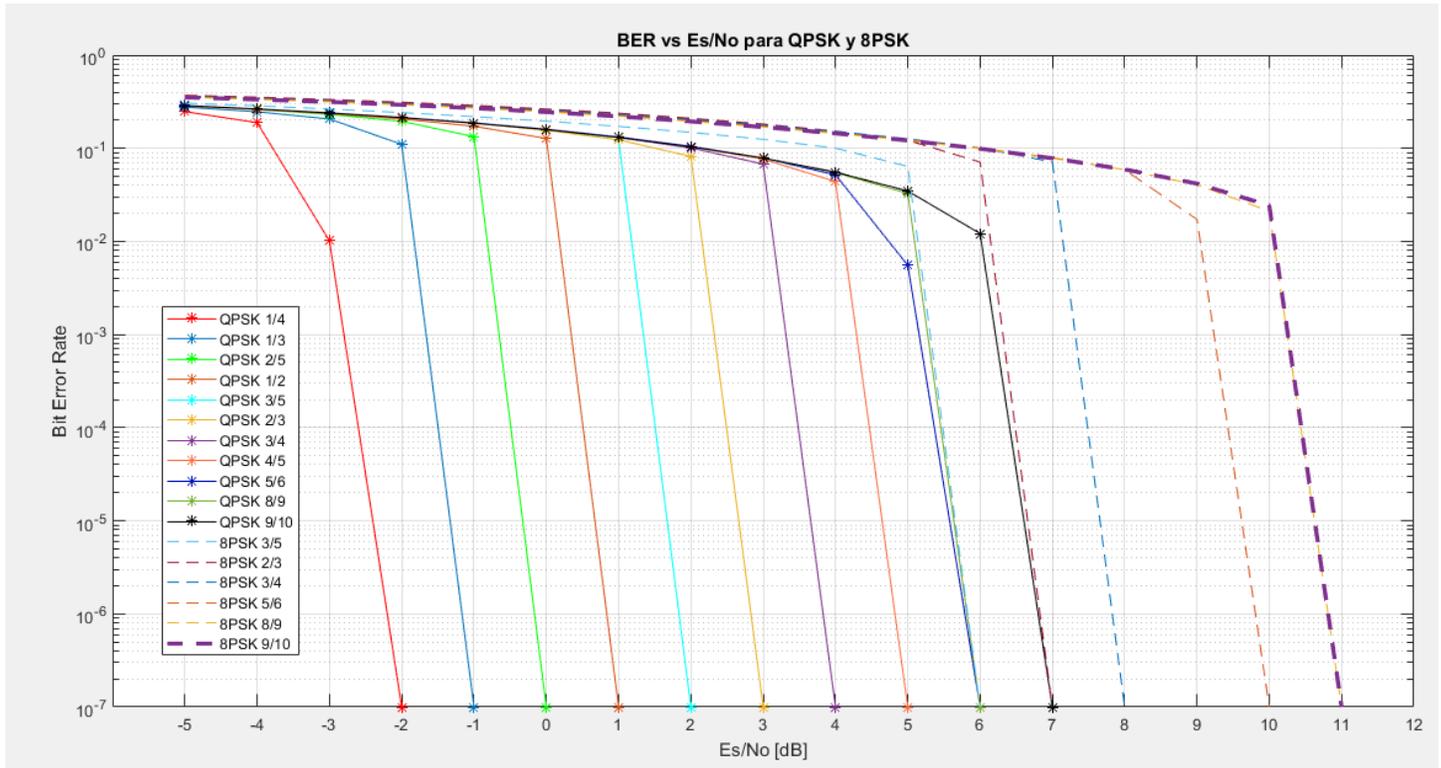


Figura 3.4. Constelaciones I-Q QPSK y 8PSK generadas en Simulink.



Posteriormente, se corren simulaciones para los diferentes ModCod disponibles para valores de  $E_s/N_0$  entre -5 y 12 dB, obteniendo como resultado los BER alcanzados para las diferentes ModCod implementados. Los datos recolectados y el proceso de análisis en Matlab se muestran en el **Anexo A1**.

La figura 3.5 muestra la gráfica BER vs  $E_s/N_0$  obtenida.



**Figura 3.5. Gráfica BER vs  $E_s/N_0$  para diferentes ModCods obtenidos de Simulink.**

Se puede constatar que las modulaciones QPSK requieren un menor  $E_s/N_0$  respecto a las de 8PSK para alcanzar el BER de  $10^{-7}$  deseado. De igual modo, las codificaciones de menor orden requieren una menor energía para ser transmitidas de manera adecuada por el canal de comunicación.

En cuanto a la codificación, se sigue el mismo comportamiento. Codificaciones de tasas menores requieren un menor valor de  $E_s/N_0$  para alcanzar un valor determinado de BER.



### 3.2. VALORES DE CONMUTACIÓN PARA ACM

Los niveles de conmutación para los diferentes ModCod del estándar son establecidos en la norma oficial del DVB-S2 dictada por la ETSI (European Telecommunications Standards Institute). Los umbrales establecidos se hacen tomando como referencia el valor de  $E_s/N_0$  para cada una de las combinaciones de modulación y codificación permitidas por el estándar.

Para generar estos valores se considera un valor de PER mínimo de  $10^{-7}$ , donde PER es la relación del número de paquetes útiles después de aplicar FEC entre el total de paquetes recibidos. En total hay 28 posibilidades distintas que abarcan un valor de  $E_s/N_0$  desde -2.35 dB hasta 16.05dB.

Los umbrales de ACM se presentan en la figura 3.6.

Mode	Spectral efficiency	Ideal $E_s/N_0$ (dB) for FECFRAME length = 64 800
QPSK 1/4	0.490243	-2,35
QPSK 1/3	0.656448	-1,24
QPSK 2/5	0.789412	-0,30
QPSK 1/2	0.988858	1,00
QPSK 3/5	1.188304	2,23
QPSK 2/3	1.322253	3,10
QPSK 3/4	1.487473	4,03
QPSK 4/5	1.587196	4,68
QPSK 5/6	1.654663	5,18
QPSK 8/9	1.766451	6,20
QPSK 9/10	1.788612	6,42
8PSK 3/5	1.779991	5,50
8PSK 2/3	1.980636	6,62
8PSK 3/4	2.228124	7,91
8PSK 5/6	2.478562	9,35
8PSK 8/9	2.646012	10,69
8PSK 9/10	2.679207	10,98
16APSK 2/3	2.637201	8,97
16APSK 3/4	2.966728	10,21
16APSK 4/5	3.165623	11,03
16APSK 5/6	3.300184	11,61
16APSK 8/9	3.523143	12,89
16APSK 9/10	3.567342	13,13
32APSK 3/4	3.703295	12,73
32APSK 4/5	3.951571	13,64
32APSK 5/6	4.119540	14,28
32APSK 8/9	4.397854	15,69
32APSK 9/10	4.453027	16,05

Figura 3.6. Valores de  $E_s/N_0$  ideal para los diferentes ModCod establecidos por la ETSI [1\_26].

Con estos resultados se genera la tabla 3.2 en la cual se aprecia que los valores propuestos por ETSI son cercanos a los valores obtenidos en Simulink.



Tabla 3.2. Valores de Modcod vs Es/No bajo las referencias ETSI y Simulink.

PER 10 <sup>-7</sup>			
Mode	ModCod	Es/No [dB] ETSI	Es/No [dB] Simulink
1	QPSK ¼	-2.35	-2
2	QPSK 1/3	-1.24	-1
3	QPSK 2/5	-0.30	0
4	QPSK ½	1.00	1
5	QPSK 3/5	2.23	2
6	QPSK 2/3	3.10	3
7	QPSK ¾	4.03	4
8	QPSK 4/5	4.68	5
9	QPSK 5/6	5.18	6
10	QPSK 8/9	6.20	6
11	QPSK 9/10	6.42	7
12	8PSK 3/5	5.50	6
13	8PSK 2/3	6.62	7
14	8PSK ¾	7.91	8
15	8PSK 5/6	9.35	10
16	8PSK 8/9	10.69	11
17	8PSK 9/10	10.98	11

De lo anterior podemos concluir que la simulación de DVB-S2 implementada en Simulink arroja una buena aproximación del comportamiento de este estándar a pesar de que no tenga todos sus bloques implementados. Para poder realizar pruebas y generar datos que no requieran tanta precisión es una herramienta funcional.

Para garantizar el cumplimiento con el estándar se ha decidido tomar como umbrales de conmutación los propuestos por ETSI. Debido a que el análisis de cálculo de enlace considerará el SNR como valor entregado se procede a convertir los valores de Es/No de la figura 3.6 en su equivalente en SNR.

El valor de SNR se obtiene conforme a la siguiente relación:

$$SNR = \frac{E_s/N_0}{(1 + \alpha)(CR_{ldpc})(CR_{BCH})} \tag{3.13}$$

En donde CR<sub>bch</sub> (Code Rate de BCH) hace referencia al porcentaje de bits utilizado por la codificación externa BCH donde el porcentaje de bits de codificación es igual a (1-CR<sub>bch</sub>) x100.

CR<sub>ldpc</sub> es la tasa de corrección del código interno LDPC, que va desde ¼ hasta 9/10, dependiendo el ModCod seleccionado.



A es el valor de Roll Off dado por el filtro coseno cuadrado por el cual pasa la señal luego de ser aleatorizada. Este valor se considerará de 0.2 para DVB-S2.

La tabla 3.3. presentada a continuación será la relación final de conmutación en la cual se basarán las simulaciones posteriores y el método ACM del procesamiento en banda base a implementar.

**Tabla 3.3. Valores de conmutación Modcod vs SNR.**

PER 10 <sup>-7</sup>					
Mode	ModCod	CRbch	CRldpc	Es/No [dB] ETSI	SNR[dB]
1	QPSK ¼	0.98815	¼	-2.35	-2.1711
2	QPSK 1/3	0.99111	1/3	-1.24	-1.1456
3	QPSK 2/5	0.99256	2/5	-0.30	-0.2772
4	QPSK ½	0.99407	½	1	0.9239
5	QPSK 3/5	0.99506	3/5	2.23	2.0603
6	QPSK 2/3	0.99630	2/3	3.10	2.8641
7	QPSK ¾	0.99810	¾	4.03	3.7233
8	QPSK 4/5	0.99630	4/5	4.68	4.3238
9	QPSK 5/6	0.99704	5/6	5.18	4.7857
10	8PSK 3/5	0.99506	3/5	5.50	5.0814
11	QPSK 8/9	0.99778	8/9	6.20	5.7281
12	QPSK 9/10	0.99780	9/10	6.42	5.9314
13	8PSK 2/3	0.99630	2/3	6.62	6.1161
14	8PSK ¾	0.99810	¾	7.91	7.3080
15	16APSK 2/3	0.99630	2/3	8.97	8.2873
16	8PSK 5/6	0.99704	5/6	9.35	8.6384
17	16APSK ¾	0.99810	¾	10.21	9.4329
18	8PSK 8/9	0.99778	8/9	10.69	9.8764
19	8PSK 9/10	0.99780	9/10	10.98	10.1443
20	16APSK 4/5	0.99630	4/5	11.03	10.1905
21	16APSK 5/6	0.99704	5/6	11.61	10.7263
22	32APSK ¾	0.99810	¾	12.73	11.7611
23	16APSK 8/9	0.99778	8/9	12.89	11.9089
24	16APSK 9/10	0.99780	9/10	13.13	12.1307
25	32APSK 4/5	0.99630	4/5	13.64	12.6018
26	32APSK 5/6	0.99704	5/6	14.28	13.1931
27	32APSK 8/9	0.99778	8/9	15.69	14.4958
28	32APSK 9/10	0.99780	9/10	16.05	14.8284



### 3.3 CÁLCULO DE ENLACE

El cálculo de enlace es un análisis que considera todas las ganancias y pérdidas que existen entre el transmisor y el receptor en un sistema de comunicaciones. Con base en estos parámetros se realiza un balance para determinar si la señal transmitida es lo suficientemente potente para ser escuchada por el receptor. En este análisis se consideran factores inherentes a los dispositivos de transmisión y recepción, tales como potencia de transmisión, ganancia de las antenas y sensibilidad del receptor, y también factores externos relacionados con el canal de comunicación, por ejemplo, pérdidas en el espacio libre, absorción atmosférica y pérdidas provocadas por lluvia. El resultado final del cálculo de enlace será el margen del enlace que en caso de ser positivo garantizará la comunicación entre los dos puntos en cuestión mientras que si resulta negativo significa que la señal transmitida no es capaz de alcanzar al receptor debido a las diferentes atenuaciones por la que es afectada.

Para realizar el cálculo de enlace del escenario satelital presentado, se emplea la metodología descrita en el libro Comunicaciones por Satélite [3\_2] en donde se puede consultar el procedimiento a detalle. A continuación se presentan los conceptos, expresiones y procedimientos más importantes relacionados con este análisis.

- **Potencia Isotrópica Radiada Efectiva (PIRE)**

La PIRE es la cantidad de potencia que emitiría una antena isotrópica para producir la densidad de potencia observada en la dirección de máxima ganancia de una antena. Se representa por la expresión:

$$\text{PIRE} = 10 \log (P_T) + G_T \quad (3.14)$$

Donde

$P_T$  es la potencia del transmisor en Watts

$G_T$  es la ganancia de la antena transmisora en dBi

- **Atenuación en el Espacio Libre**

Se denomina así a la atenuación de potencia provocada por la distancia recorrida por una señal portadora. Esta disminución de la potencia de la señal es la que más impacta el enlace de comunicaciones satelitales y su valor está en el orden de los 100 a los 200 dB dependiendo de la altura orbital.

Para calcular el valor de esta atenuación se considera la siguiente expresión:

$$L_{el} = \left( \frac{4\pi f R}{c} \right)^2 \quad (3.15)$$

Donde,

$f$  es la frecuencia de la señal portadora en Hz

$R$  es la distancia existente entre el satélite y la ET en metros

$c$  es la velocidad de la luz en m/s

- **Pérdidas por conectores**

Son las atenuaciones generadas por la línea y los conectores que unen la última etapa de amplificación del Front End de RF con la antena en el transmisor. Esta atenuación es dependiente del tipo de cable empleado, la frecuencia de operación, la longitud de la línea y el material con el que están hecho los conectores. Las pérdidas por conectores se encuentran entre los 0.5 dB y 1 dB por cada tramo de guía dentro del sistema de comunicaciones.

- **Pérdidas por absorción atmosférica**

Esta atenuación se debe a la interacción de las ondas electromagnéticas con las moléculas de los gases en la atmósfera. Estas interacciones disminuyen la potencia de la señal portadora conforme esta atraviesa las diferentes capas de la atmósfera, y el comportamiento se intensifica en las frecuencias de resonancia del vapor de agua, oxígeno y bióxido de carbono. Este fenómeno afecta de manera particular a las bandas Ka, Q y V donde los valores de atenuación llegan a alcanzar valores de 0.5 dB/km. Para frecuencias menores a los 10 GHz el valor de la atenuación no sobrepasa los 0.001 dB/km por lo que se puede considerar despreciable.

La figura 3.7 muestra la atenuación provocada por la absorción atmosférica, en dirección al cenit, a una temperatura de 20°C y una densidad de vapor de agua de 7.5 g/m<sup>3</sup>.

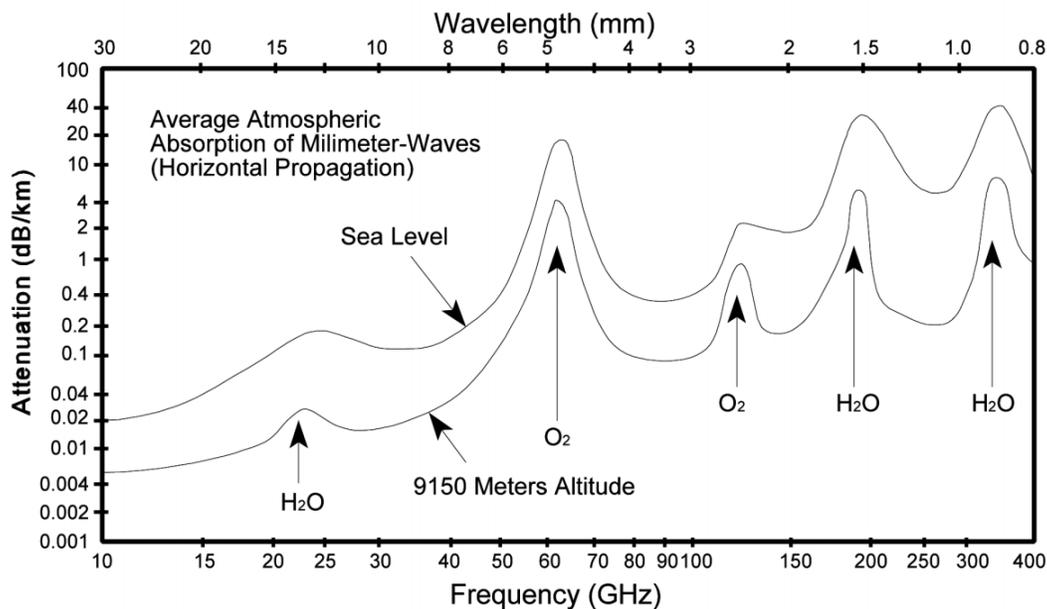


Figura 3.7. Atenuación producida por la absorción atmosférica [3\_2].

- **Atenuación por lluvia**

Las ondas electromagnéticas se atenúan en caso de que interactúen con lluvia. Esto se debe a que las gotas de lluvia absorben energía al ser calentadas por las señales de microondas y conforme la longitud de onda se hace comparable al tamaño de las gotas, el efecto es más severo, causando mayor atenuación en las bandas Ka, Ku y C. Además, la atenuación total dada por este fenómeno depende de la frecuencia de la señal portadora afectada, la intensidad de lluvia y la distancia



que las ondas electromagnéticas deben de recorrer en presencia de lluvia, lo cual a su vez depende de la altura de las nubes respecto al suelo y del ángulo de elevación de la ET.

Para caracterizar la atenuación por lluvia se usar el método dictado por el CCIR perteneciente a la Unión Internacional de Telecomunicaciones. Los pasos se enlistan a continuación:

- a) Calcular la altura  $h_r$  (Altura de las nubes respecto al nivel del mar)

$$h_r = 3 + 0.028L \quad 0^\circ < L < 36^\circ \text{ [km]}$$

$$h_r = 4 - 0.075(L - 36) \quad L > 36 \text{ km}$$

donde  $L$  es la latitud de la ET en grados.

- b) Conocida  $h_r$ , la longitud física del trayecto oblicuo  $d$  por el cual la señal portadora es afectada por la lluvia se obtiene como

$$d = \frac{h_r - h_0}{\sin \theta}$$

Donde

$h_0$  es la altura de la ET sobre el nivel del mar dado en kilómetros

$\theta$  es el ángulo de elevación de la ET

- c) Después se calcula el factor reductor aplicable para una disponibilidad de 99.99%

$$r_{0.01} = \frac{1}{1 + \left( \frac{d}{35e^{-0.015R_{0.01}}} \right) \cos \theta}$$

En donde  $R_{0.01}$  es la intensidad de lluvia en mm/h excedida 0.01% en un año de acuerdo a la tabla 3.4.

**Tabla 3.4. Intensidad de lluvia excedida [3\_2].**

Porcentaje del tiempo excedido en un año (%) y equivalencia en minutos acumulados	Intensidad de lluvia excedida (mm/h)														
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	
0.1	526	2	3	5	8	6	8	12	10	20	12	15	22	35	65
0.01	53	8	12	15	19	22	28	30	32	35	42	60	63	95	145
0.001	5.3	22	32	42	42	70	78	65	83	55	100	150	120	180	250



Donde cada letra corresponde a una región designada por la CCIR del mapa mostrado a continuación:

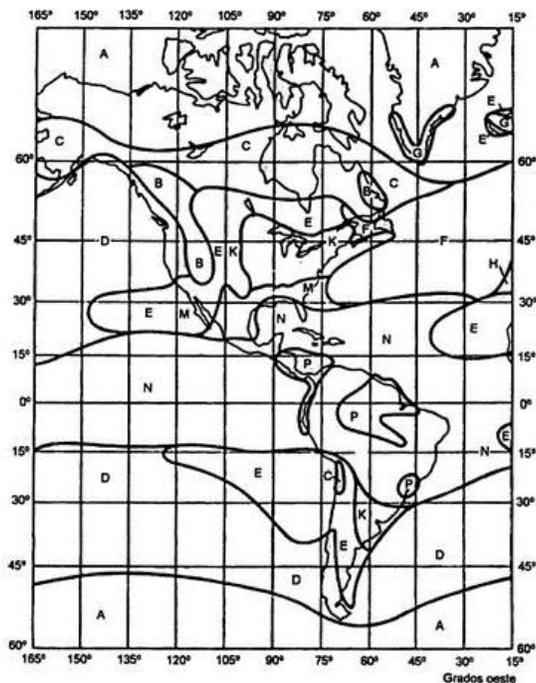


Figura 3.8. Mapa CCIR de regiones de intensidad de lluvia [3\_2].

Para Querétaro, México la región de la CCIR que corresponde es la designada por la letra M y la intensidad de lluvia excedida el 0.01% del tiempo durante un año (equivalente a 52 minutos) es de 63 mm/h.

d) Luego se obtiene el coeficiente de atenuación específica  $\gamma$  dado en dB/km.

$$\gamma = aR_{0.01}^b [dB/km]$$

Los coeficientes a y b se calculan teóricamente y dependen de la propagación electromagnética a través de las gotas de lluvia, dependiendo de la polarización de la señal y de su frecuencia. En la tabla 3.5 se muestran sus valores asociados para las diferentes frecuencias de operación.

Tabla 3.5. Coeficientes de atenuación específica [3\_2].

Frecuencia (GHz)	Ah	av	Bh	bv
1	0.0000387	0.0000352	0.912	0.88
2	0.00065	0.000138	0.963	0.923
4	0.00175	0.000591	1.121	1.075
6	0.00301	0.00155	1.308	1.265
8	0.00454	0.00265	1.332	1.31
10	0.0101	0.00395	1.327	1.264
12	0.0188	0.00887	1.276	1.2
15	0.0367	0.0168	1.217	1.128
20	0.0751	0.0335	1.154	1.065
25	0.124	0.0691	1.099	1.03
30	0.187	0.113	1.061	1



e) Finalmente, la atenuación  $L_{lluvia}$  se calcula como:

$$L_{lluvia} = \gamma(d)(r_{0.01})$$

- **Potencia de ruido térmico**

La potencia de ruido térmico o ruido blanco es un valor con una distribución plana presente en todas las frecuencias, por lo que el valor de potencia es constante para diferentes rangos de anchos de banda.

Se calcula con la expresión:

$$N = kTB \text{ [watts]} \quad (3.16)$$

Donde,

k es la constante de Boltzman =  $1.38 \times 10^{-23}$  J/K

T es la temperatura de ruido equivalente en kelvin

y B es el ancho de banda del sistema en Hz

Si desea expresarse en decibeles la expresión quedaría como  $N[\text{dB}] = 10 \log_{10}(N)$

La temperatura equivalente de ruido se refiere al ruido que la antena de una estación terrena captura a través de los lóbulos principal y secundarios de su patrón de radiación y que es generado por diferentes fuentes. Existen 3 componentes principales de la temperatura de ruido: la temperatura de cielo despejado, la temperatura de lluvia y la temperatura de tierra.

La temperatura de cielo despejado encapsula a los efectos del ruido cósmico, el galáctico, y troposférico. Su valor es de aproximadamente 20K para una frecuencia de 2GHz.

Por su parte, la temperatura de lluvia se define por la siguiente expresión:

$$T_{lluvia} = 280 \left( 1 - \frac{1}{L_{lluvia}} \right) [K] \quad (3.17)$$

Mientras que el valor de la atenuación por lluvia aumenta el valor de la Temperatura de lluvia se aproxima a 280K que es su máximo valor.

La temperatura de Tierra es el ruido proveniente del suelo puede tomar un valor de hasta 50K.

Con estos valores se hace uso de la siguiente expresión para calcular la Temperatura de ruido equivalente:

$$T = T_{cd} + T_{lluvia} + T_{tierra} \quad (3.18)$$

Donde en el peor de los casos tomaría un valor de  $T = 20 + 280 + 50 = 350K$ .



- **Potencia de portadora**

La potencia de portadora se define como:

$$C = \text{PIRE}_{\text{TX}} + G_{\text{RX}} - L_{\text{Total}} \text{ [dB]} \quad (3.19)$$

Donde,

$G_{\text{RX}}$  es la ganancia en la antena receptora expresada en dBi.

Para el caso del enlace de bajada que corresponde al del satélite transmitiendo hacia la ET, el valor de  $\text{PIRE}_{\text{TX}}$  está asociado al satélite y  $G_{\text{RX}}$  a la ET.

- **Relación Señal a Ruido**

El valor de la Relación Señal a Ruido, también conocido como Signal to Noise Ratio (SNR) o Carrier to Noise (C/N) expresa la comparación del valor de potencia de la señal y la potencia de ruido. Si el resultado es un número positivo, tratándose de decibeles, se entiende que la señal de la portadora es lo suficientemente fuerte como para poder ser entregada correctamente en el receptor.

Se expresa como:

$$\text{SNR} = \text{C} - \text{N} \text{ [dB]} \quad (3.20)$$

Dado que el valor de las pérdidas por espacio libre dependen del rango de transmisión y este valor, a su vez, depende de la posición respecto del tiempo del satélite, se puede apreciar que el valor final de SNR será variable. Tendrá valores mínimos cuando el satélite entre y salga en línea de vista desde el horizonte, y tendrá un valor máximo a la mitad de esta trayectoria.

Para poder calcular todos los valores de SNR para cada uno de los rangos obtenidos en STK se desarrolló un código en Matlab que toma en consideración los cálculos descritos anteriormente. El resultado que entrega es precisamente el valor de SNR en decibeles para cada rango dado. Este código se adjunta en el **Anexo A2**. Los parámetros variables a considerar son el rango (determinado por el tipo de órbita) y la potencia de transmisión (0.5W y 1 W).

Los resultados obtenidos se muestran en la tabla 3.6.



Tabla 3.6. Resultados obtenidos del cálculo de enlace descendente.

Enlace de Bajada 2200 MHz	Órbita ISS (406 km)		Órbita heliosíncrona (600km)	
Rango (máx/mín/avg)	1,834/406/1,274 km		2,342/600/1,643 km	
Ganancia de la antena satélite	6 dBi		6 dBi	
Ganancia antena ET	35.28 dBi		35.28 dBi	
Pérdidas por espacio libre [dB] (máx/mín/avg)	164.56/151.47/160.91		166.68/154.86/163.17	
Atenuación por absorción atmosférica	0.3786 dB		0.3786 dB	
Atenuación por lluvia	0.0667 dB		0.0667 dB	
Pérdidas en conectores y cable coaxial	4 dB		4 dB	
Ancho de banda	10 MHz		10 MHz	
Potencia de Ruido [dB]	-140.15		-140.15	
Potencia de Tx	<b>0.5 W</b>	<b>1 W</b>	<b>0.5 W</b>	<b>1 W</b>
PIRE Satélite	3 dB	6 dB	3 dB	6 dB
Potencia portadora [dB] (máx/mín/avg)	-117.64/-130.73/ 127.08	-114.63/-127.72/ 124.07	-121.03/-132.85/ 129.35	-118.02/-129.84/ 126.34
C/N (máx/mín/avg) [dB]	<b>22.50/9.41/ 13.06</b>	<b>25.51/12.42/ 16.07</b>	<b>19.11/7.29/ 10.79</b>	<b>22.12/10.30/ 13.80</b>



### 3.4. ANÁLISIS DE RESULTADOS

Primeramente, en la figura 3.9, se muestran los valores de SNR obtenidos para las diferentes potencias de transmisión y en relación a los diferentes rangos de cada una de las órbitas. Como se esperaba, el valor de SNR es máximo cuando el valor del rango es mínimo y viceversa. Los rangos dados por la órbita ISS cubren valores entre los 400 y 600km que la órbita SSO no considera. De igual modo, los rangos por encima de los 1,800 km y hasta los 2,300 km aproximadamente sólo son generados en la órbita SSO. El valor de SNR entre las dos potencias de transmisión, 0.5 y 1W, varía en un valor constante de 3dB en ambas órbitas.

De igual forma, el valor de SNR no está por debajo de los 6dB por lo que se infiere que es posible aplicar al menos un ModCod de 8PSK de acuerdo a la tabla 3.3.

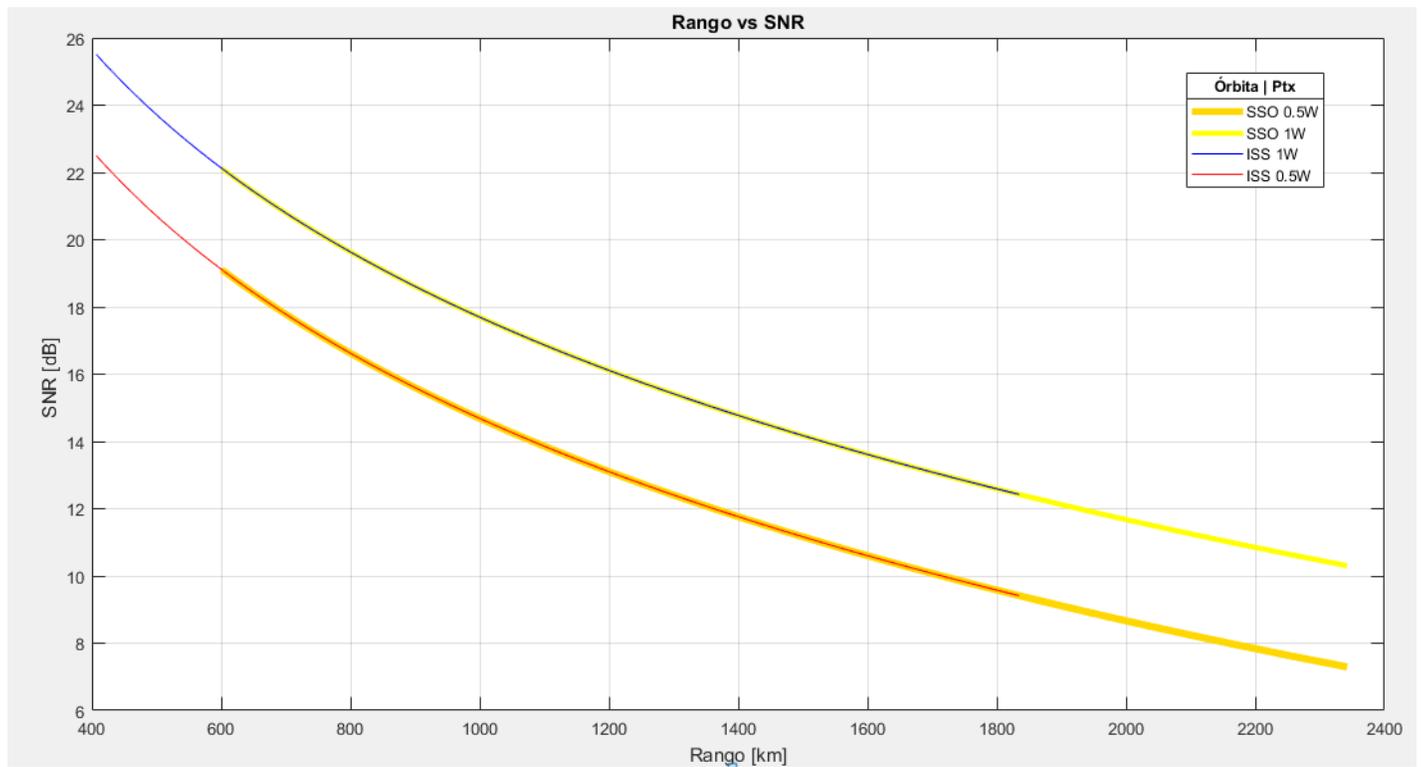


Figura 3.9. Gráfica de los valores de SNR para la órbita ISS y SSO con Ptx=0.5W y 1W.



Posteriormente, se generan los histogramas de los SNR para ambas órbitas con el objetivo de visualizar la distribución de los valores de SNR y el número de eventos de cada uno de estos valores. Ver fig. 3.10, 3.11, 3.12 y 3.13.

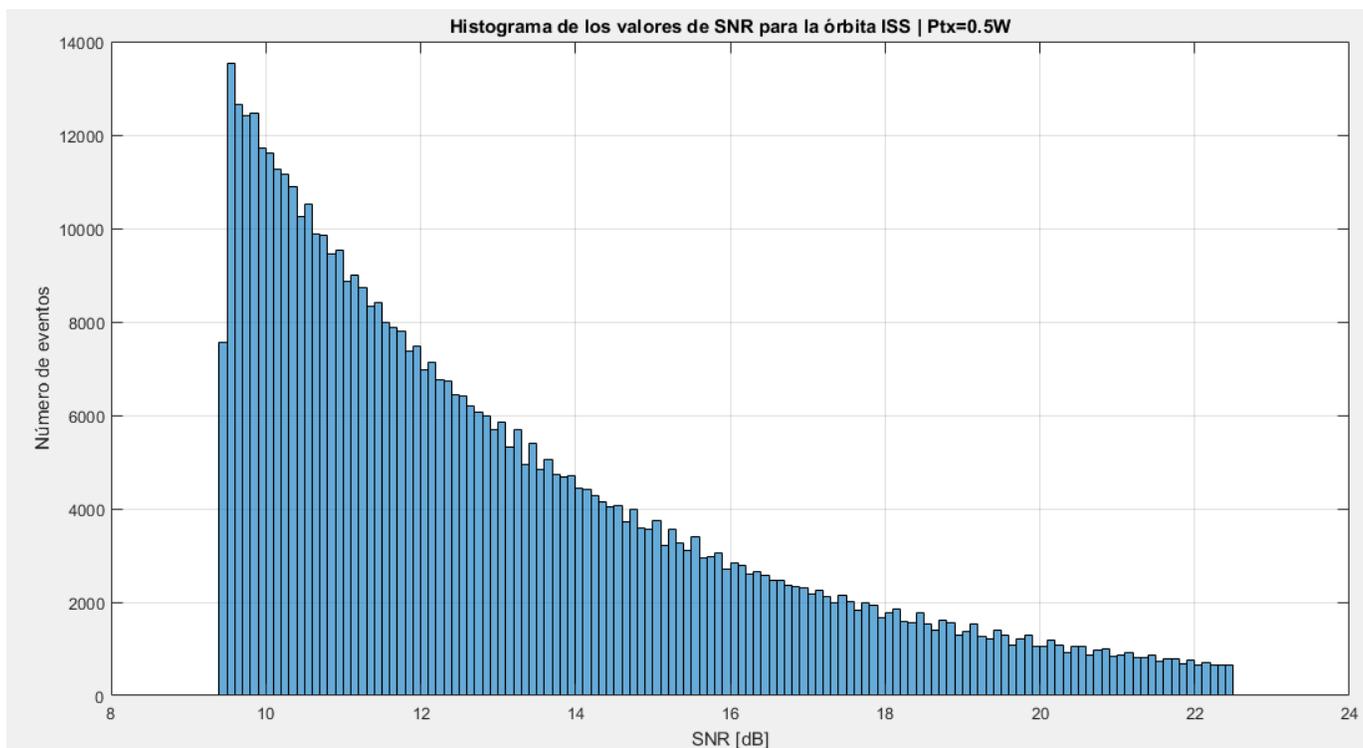


Figura 3.10. Histograma de los valores de SNR para la órbita ISS con  $P_{tx}=0.5W$ .

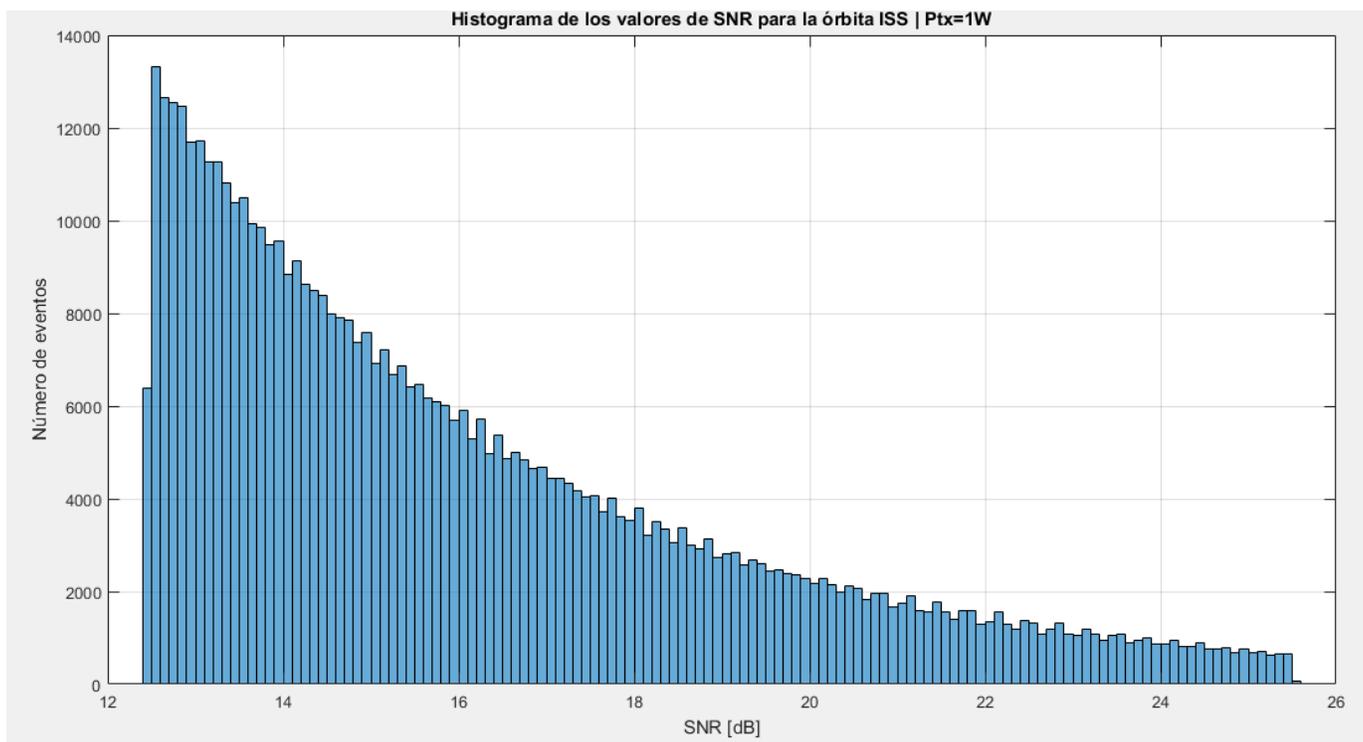


Figura 3.11. Histograma de los valores de SNR para la órbita ISS con  $P_{tx}=1W$ .

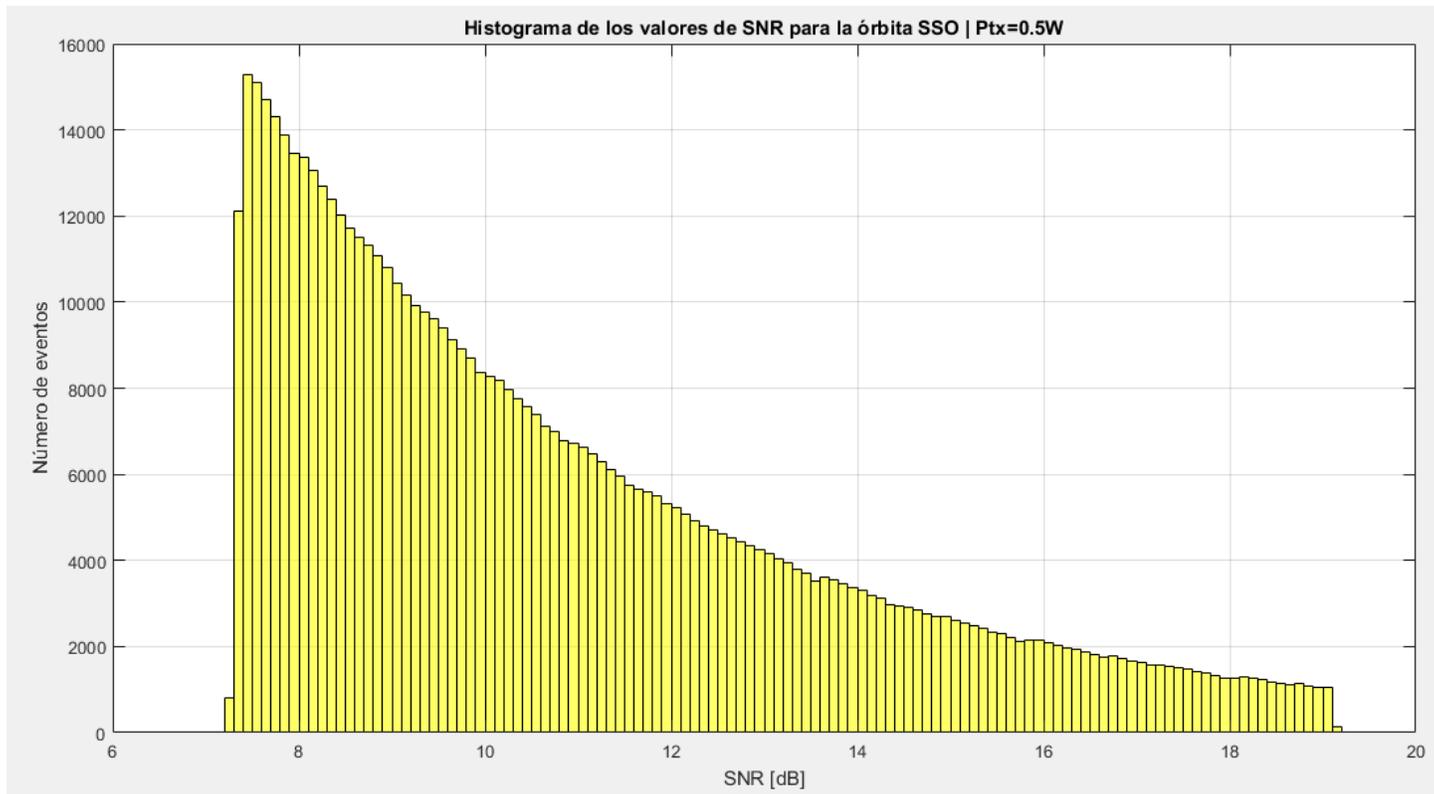


Figura 3.12. Histograma de los valores de SNR para la órbita SSO con  $P_{tx}=0.5W$ .

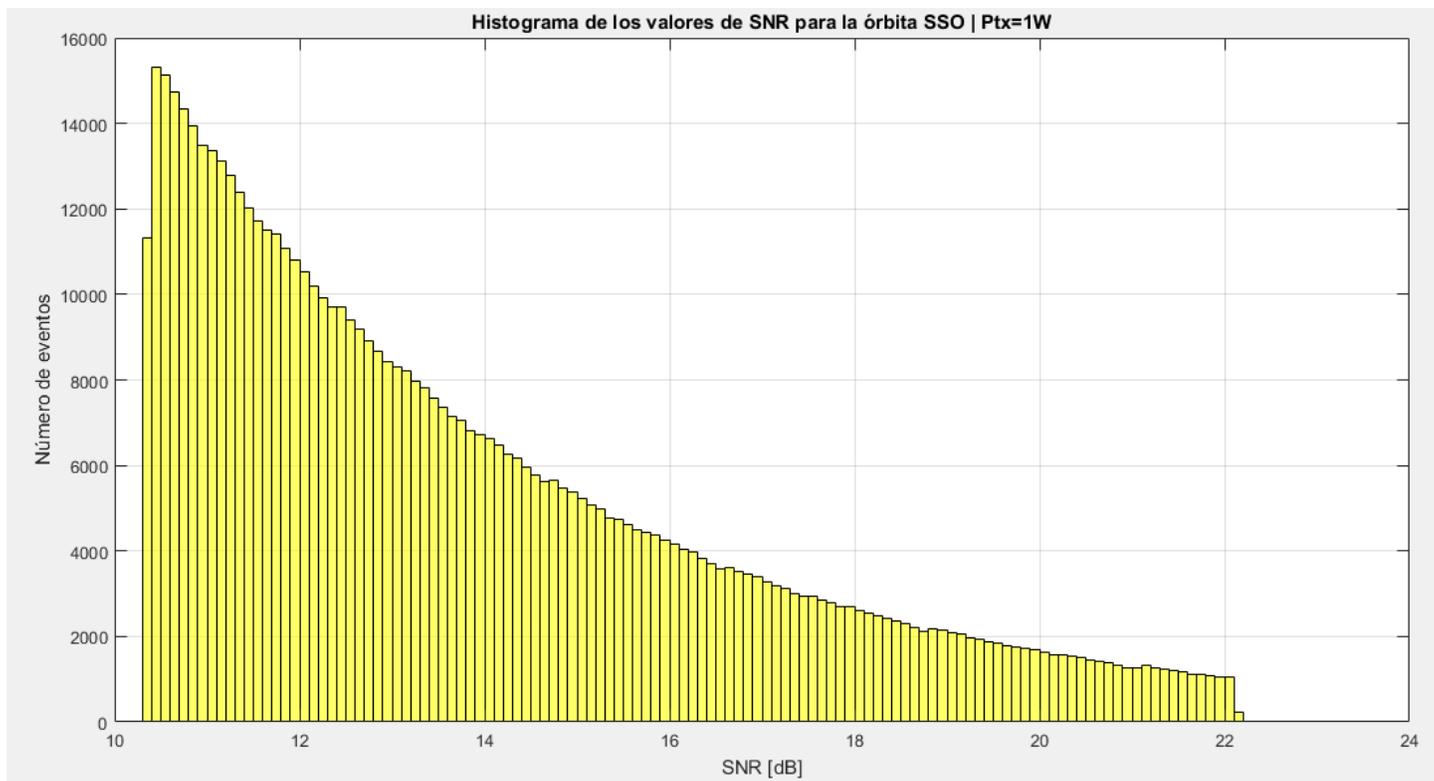


Figura 3.13. Histograma de los valores de SNR para la órbita SSO con  $P_{tx}=1W$ .



En el caso de la órbita ISS con  $P_{tx}$  de 1W el valor más frecuente de SNR es 12.6 dB con 13,330 eventos, mientras que el menos frecuente es 25.6 dB con 73 eventos. Para la órbita SSO el valor más frecuente de SNR es 10.5dB con 15,320 eventos, mientras que el menos frecuente es 22.2 dB con 225 eventos. Para el análisis de potencia de transmisión igual a 0.5W se observa que los valores son reducidos 3dB pero que el comportamiento del histograma es similar al de 1W.

Se observa como ambos histogramas disminuyen la frecuencia de eventos mientras el valor de SNR es mayor. Esto comprueba la relación inversa existente entre el valor de rango y el de SNR.

Luego, se procede a graficar el valor de SNR respecto a la secuencia de valores de los rangos registrados. Esto es, los diferentes valores de rango obtenidos por cada pase mostrado de manera secuencial. De este modo, se podrá observar la variación del SNR respecto al tiempo. Para este análisis se emplean sólo las primeras 10,000 muestras de cada órbita configurada a 1W de  $P_{tx}$  y se generan las gráficas de las figuras 3.14 y 3.15.

Respecto al comportamiento observado en estas gráficas, se encuentra una gran similitud a las gráficas de rango mostradas en la figura 2.34. En este caso, cada pase está representado por una cresta donde el comienzo de la línea de vista entre el satélite y la ET se da cuando el valor de SNR es mínimo. Conforme el satélite continúa su trayectoria y se acerca cada vez más a la ET el rango entre estos disminuye y su SNR aumenta hasta llegar a un valor máximo, momento en el cual el valor de rango es mínimo. Justo después de este pico el SNR comienza a decrementar representando el alejamiento del satélite de la ET hasta llegar a un valor mínimo interrumpido por la pérdida de línea de vista entre satélite y ET. En este punto el pase es terminado y empieza el siguiente. Las gráficas no están referenciadas en el tiempo por lo que se observa un pase seguido de otro de manera ininterrumpida.

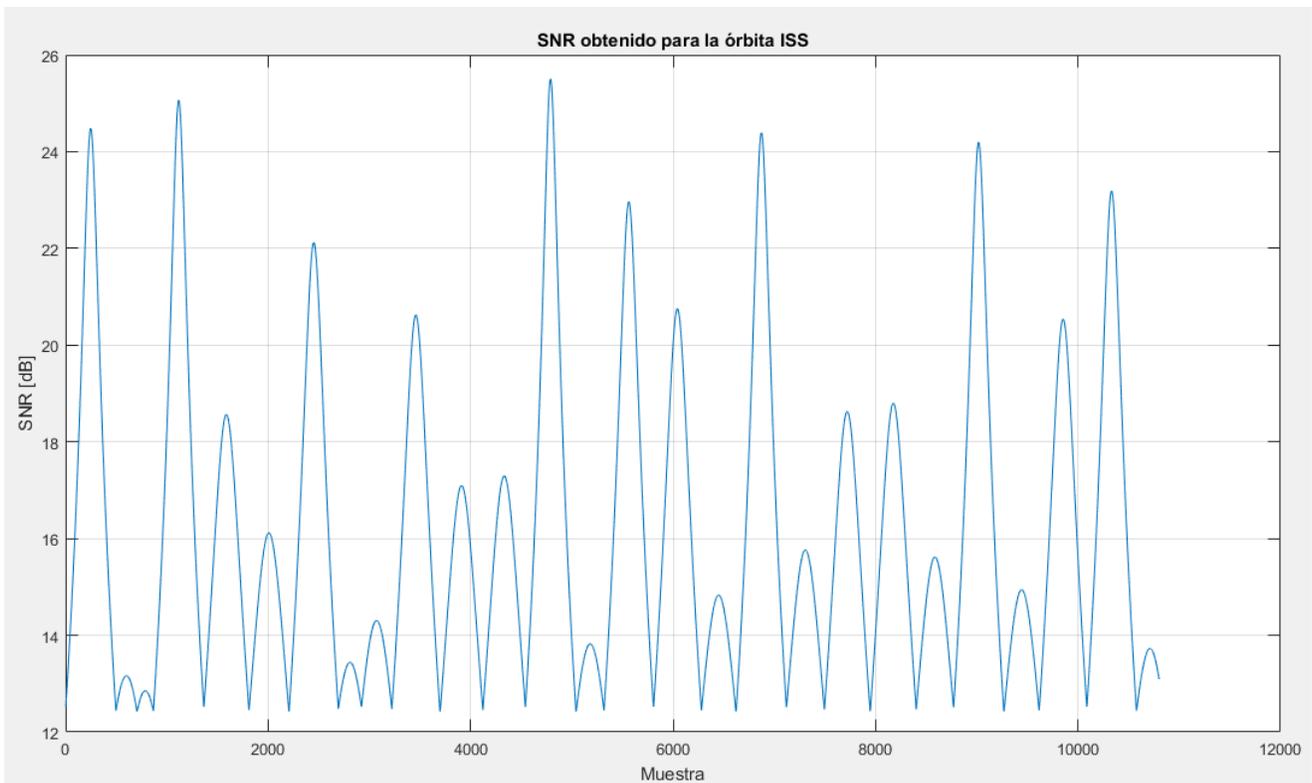


Figura 3.14. Gráfica de los valores de SNR para la órbita ISS.

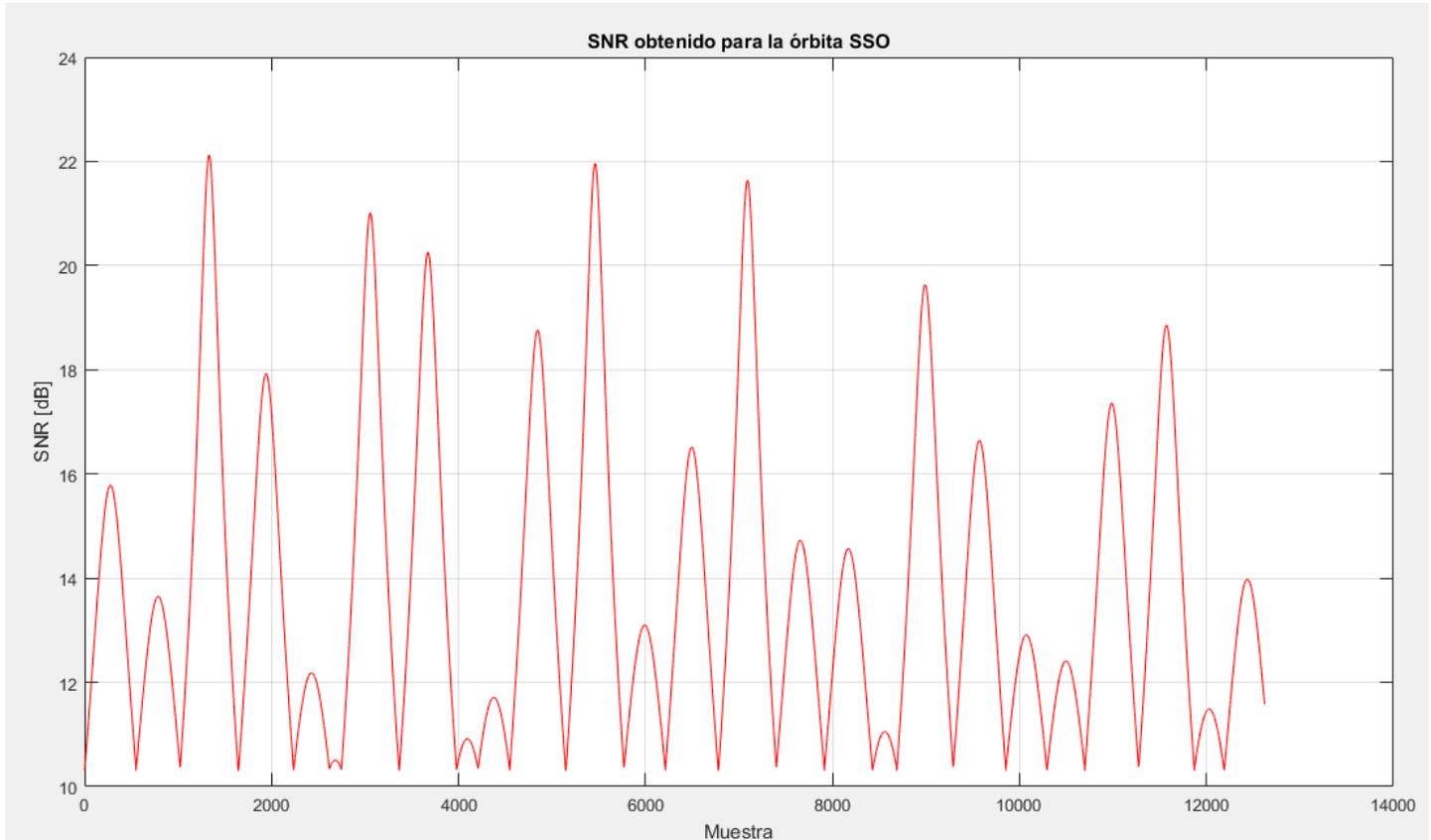


Figura 3.15. Gráfica de los valores de SNR para la órbita SSO.

### 3.4.1. Tasa de datos

Con los valores de SNR disponibles se procede a asignar el ModCod determinado por los umbrales obtenidos en la tabla 3.3. Una vez hecho esto, es posible obtener la tasa de datos generada durante el transcurso de la misión en relación a la variación del ModCod. De este modo, se podrá conocer la capacidad total de transmisión del canal con el esquema ACM.

De igual modo, se calcula la tasa de datos para un esquema CCM fijado en el ModCod QPSK 1/2. Sin importar la variación del SNR la modulación y codificación permanecerán fijas. Con el acumulado de la tasa de datos durante toda la misión se genera la capacidad de canal CCM. Finalmente, estos valores de ACM vs CCM pueden ser analizados y así visualizar las diferencias de capacidad existentes entre ambos esquemas.

Para comenzar, se muestra la expresión con la cual se puede determinar la tasa de datos en [bps] del enlace:

$$R_{bps} = BW \frac{\eta_{espectral}}{1 + \alpha} \tag{3.21}$$

Donde,

BW es el ancho de banda en [Hz]

$\eta_{espectral}$  es la eficiencia espectral

y  $\alpha$  es el factor de roll-off



A su vez, la Eficiencia Espectral se define como:

$$\eta_{espectral} = (M)(CR_{ldpc})(CR_{bch}) \tag{3.22}$$

Donde,

M es el número de bits/símbolo de la modulación

CR<sub>ldpc</sub> es la tasa de codificación LDPC

CR<sub>bch</sub> es la tasa de codificación BCH

Así, para el cálculo de la tasa de datos del esquema CCM con QPSK ½ el cálculo en Matlab sería el siguiente:

```
M=2;%bits por símbolo QPSK=2; 8PSK=3
CR_ldcp=1/2; %LDCP FEC Rates
CR_bch=0.99407; %BCH inflation tomado de ETSI DVB-S2
Eff_esp=M*CR_ldcp*CR_bch; %Eficiencia Espectral
Rbps=(BW*Eff_esp)/(1+RollOff); %Data Rate [bps]
Rbps=8.28 Mbps
```

Para la implementación de la conmutación y el cálculo de la tasa de datos en ACM y CCM se ha desarrollado un código en Matlab incluido en el **Anexo A3 y A4**. Para este cálculo la información presente en la tabla 3.3 es esencial. En la siguiente gráfica (fig. 3.16) se muestran las diferentes tasas de datos asociadas a cada uno de los ModCod correspondientes. En ella se observa que el valor mínimo corresponde al valor de 4Mbps para QPSK ½ y el máximo es de 36Mbps para 32APSK 9/10, es decir, una diferencia de 32 Mbps entre el ModCod de menor y mayor orden.

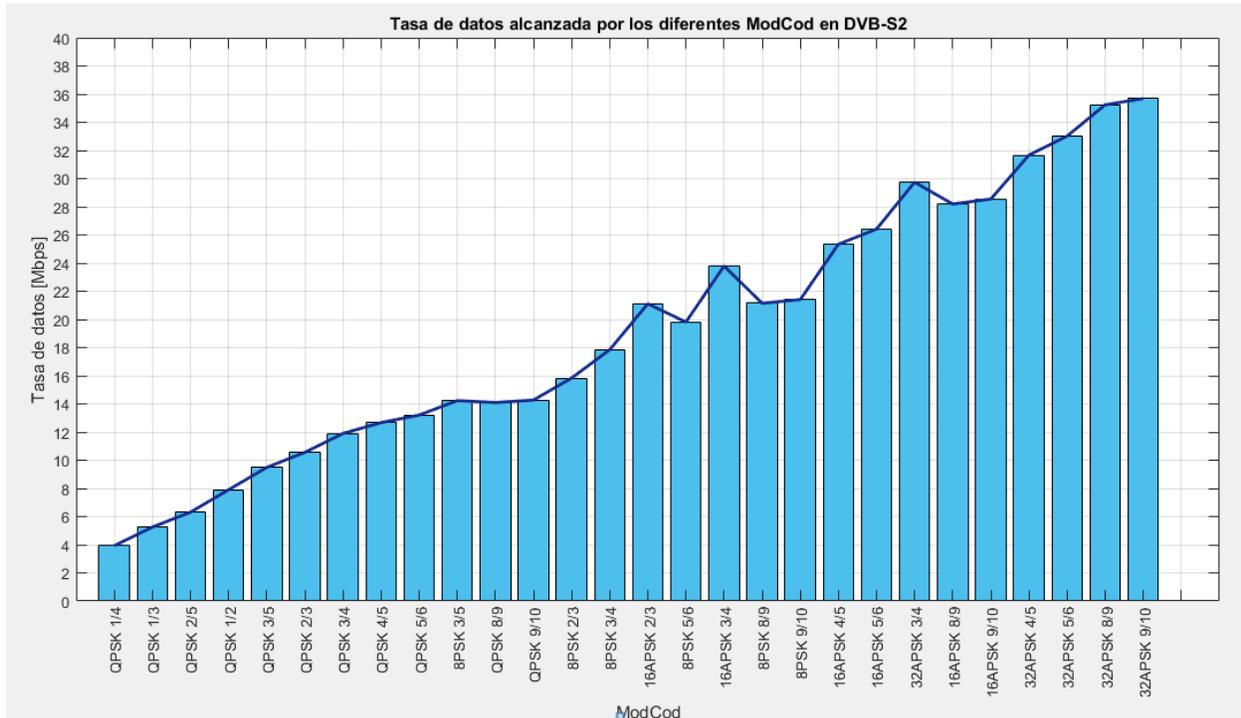


Figura 3.16 Tasa de datos alcanzada por los diferentes ModCod en DVB-S2.



### 3.4.2 Estimación de la capacidad del canal en DVB-S2

Adicionalmente, se ha generado un programa que unifica todos los elementos anteriores: rangos, SNR, conmutación y tasa de datos. Con este, es posible generar el valor de la capacidad del canal durante todo el tiempo de la simulación (enero-diciembre 2020) con un tiempo de muestreo de 1 segundo. Este programa se muestra en el **Anexo A5**.

Una vez corriendo la simulación, los resultados obtenidos son los mostrados en la tabla 3.7.

**Tabla 3.7. Capacidad del canal para las órbitas ISS y SSO durante el 2020.**

Esquema	CCM Rbps					ACM Rbps					
	Mín Mbps	Máx Mbps	Avg Mbps	Acumulado	Fotos/año	Mín Mbps	Máx Mbps	Avg Mbps	Acumulado	Fotos/año	Fotos/día
Órbita ISS 0.5W	7.90	7.90	7.90	520.85 GB	13,984	19.80	35.68	29.50	1945.5 GB	52,234	143
Órbita ISS 1W	7.90	7.90	7.90	520.85 GB	13,984	28.54	35.68	34.24	2257.9 GB	60,622	166
Órbita SSO 0.5W	7.90	7.90	7.90	608.59 GB	15,533	15.83	35.68	24.91	1919.4 GB	48,989	134
Órbita SSO 1W	7.90	7.90	7.90	608.59 GB	15,533	25.33	35.68	31.08	2394.7 GB	61,121	167

La capacidad total del enlace de comunicaciones se cuantifica de dos maneras: GBytes acumulados y número de fotografías, ambos parámetros referenciados al tiempo total de la misión, es decir, un año.

Para el esquema CCM, de manera evidente la tasa de datos se mantiene constante con un valor de 7.90 Mbps con un ModCod fijo de QPSK 1/2. La cantidad de datos acumulados es de 520.85 GB y de 608.95 GB para las órbitas ISS y SSO, respectivamente. Esto se traduce en un valor de 13,984 y 15,533 fotografías descargadas en un año. Cabe mencionar que el tamaño de referencia para una fotografía es el indicado en la tabla 1.18 equivalente a 40.12 MB para la órbita SSO y 38.14 MB para la ISS.

En referencia al esquema ACM, se aprecia como las ventajas de la conmutación son proyectadas en una capacidad del canal mayor al del esquema CCM. En este esquema la tasa de datos no es constante por lo que se tienen valores pico y mínimos de velocidad de transmisión. Para la órbita ISS el valor mínimo es de 19.80 Mbps mientras que para la SSO es de 15.83 Mbps, ambos valores alcanzados con una potencia de transmisión de 0.5W. Ambas órbitas presentan un valor pico de 35.68 Mbps dado por el ModCod 32APSK 9/10.

La capacidad máxima del canal de la ISS es de 2257.9GB lo que representa 60,622 fotografías anuales. La SSO presenta valores de 2394.7 GB y 61,121 fotografías. Ambas cifras alcanzadas con una potencia de transmisión de 1W.

En términos generales, la capacidad de descarga de información de la órbita SSO es mayor que la órbita ISS a pesar de que la tasa de datos promedio sea mayor en la ISS. Esto se debe al mayor tiempo en línea de vista que presenta la órbita SSO a lo largo de un año.



En relación a los dos posibles valores de la potencia de transmisión,  $0.5W$  y  $1W$ , se observa que el impacto generado es una variación en la tasa de datos acumulados y, por ende, en el número de fotografías obtenidas. Sin embargo, esta variación, contrario a lo pensado, es limitada con una variación de sólo 16% para el caso de la órbita ISS, y de 24% para la órbita SSO, respecto al número máximo de fotografías obtenidas. Estos valores señalan la factibilidad de implementar una misión con una potencia de transmisión reducida ( $0.5W$ ) y aun así obtener resultados positivos, siempre y cuando un umbral de 134 fotografías diarias sea considerado como exitoso.

Al obtener los histogramas de los ModCod utilizados por ambas órbitas durante todo el periodo de la misión se obtienen las figuras 3.17 a 3.20. En ellas se aprecia que las modulaciones permitidas por los umbrales de conmutación son 8PSK, 16APSK y 32APSK, y que en ambas órbitas el ModCod más utilizado es 32APSK 9/10.

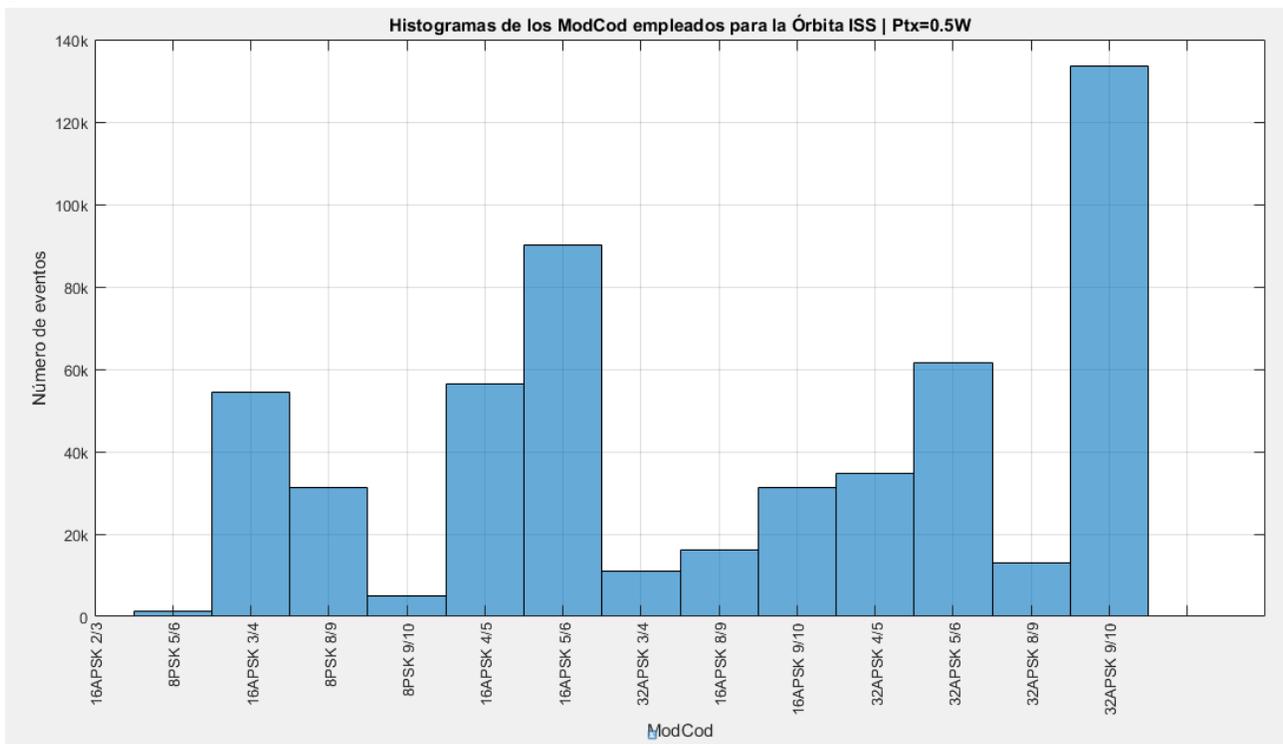


Figura 3.17. Histograma de los ModCod para la órbita ISS | Ptx=0.5W.

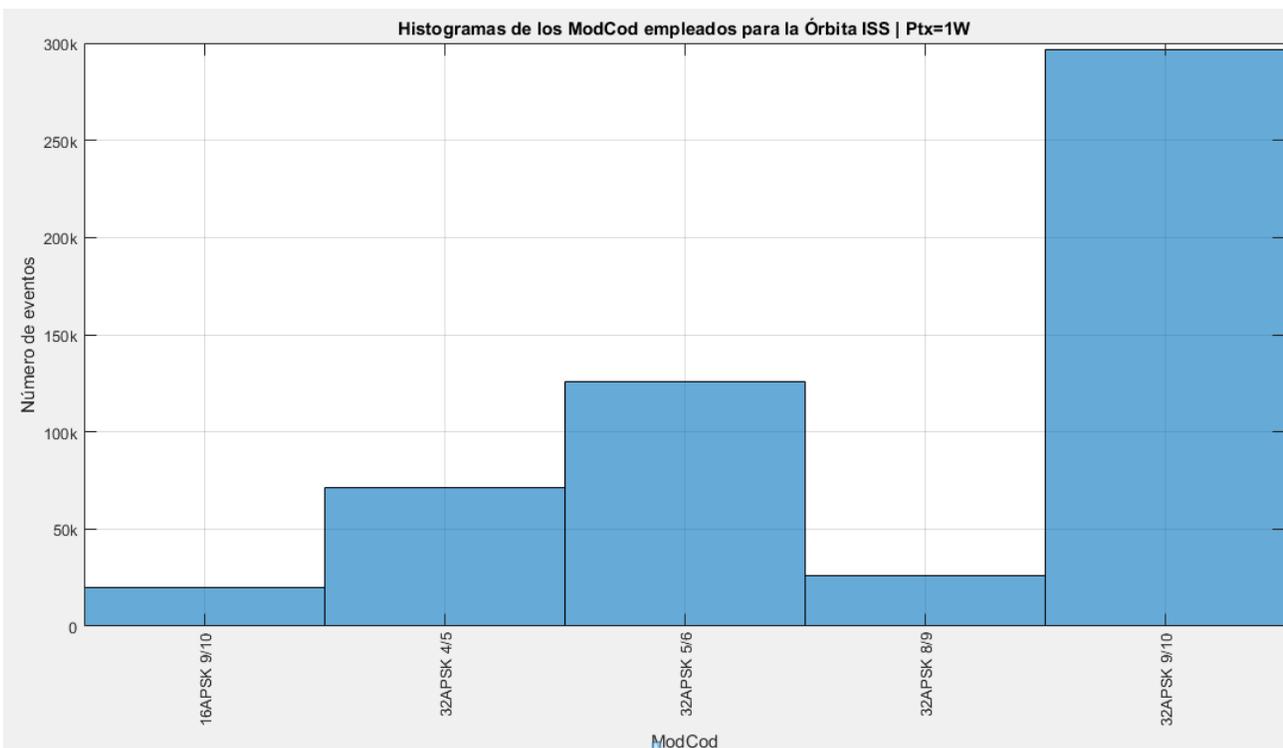


Figura 3.18. Histograma de los ModCod para la órbita ISS | Ptx=1W.

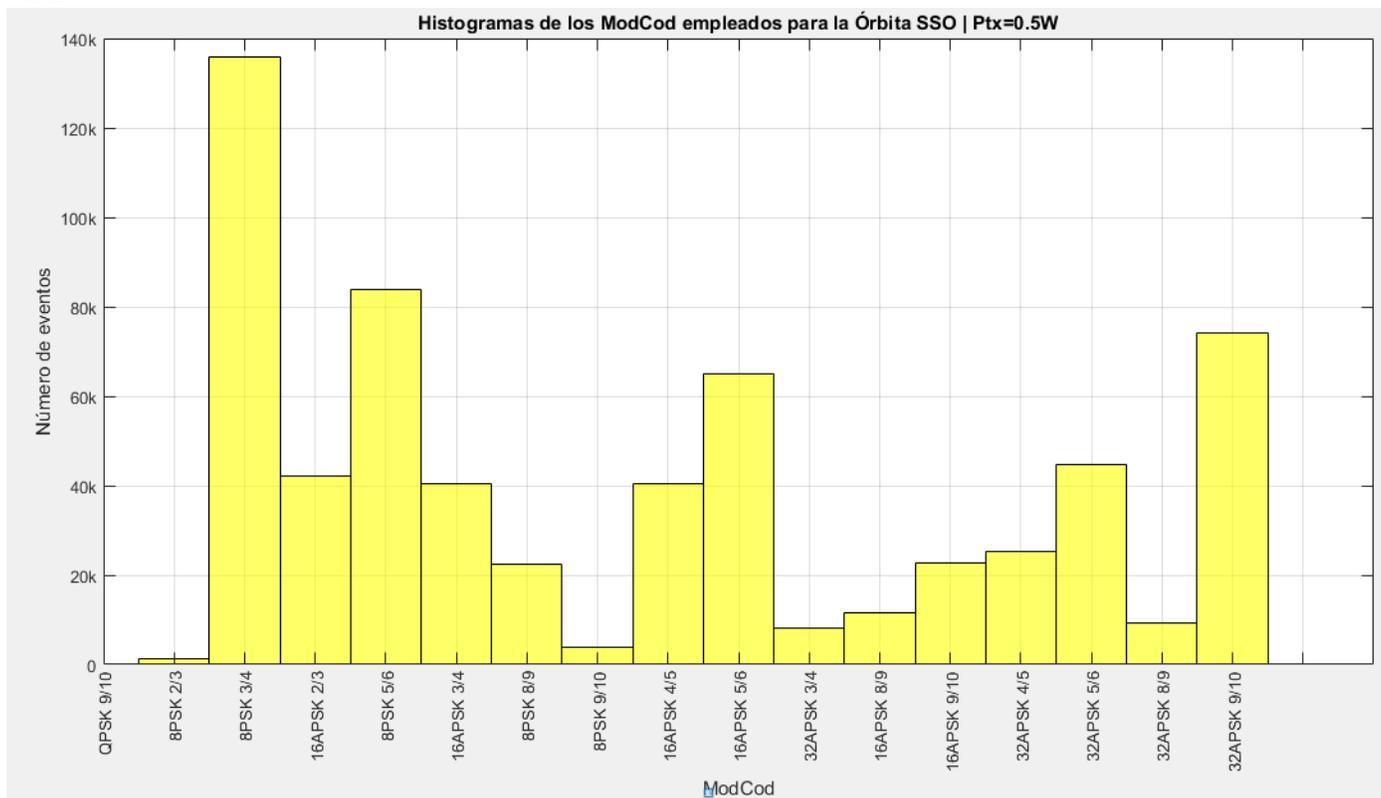


Figura 3.19. Histograma de los ModCod para la órbita SSO | Ptx=0.5W.

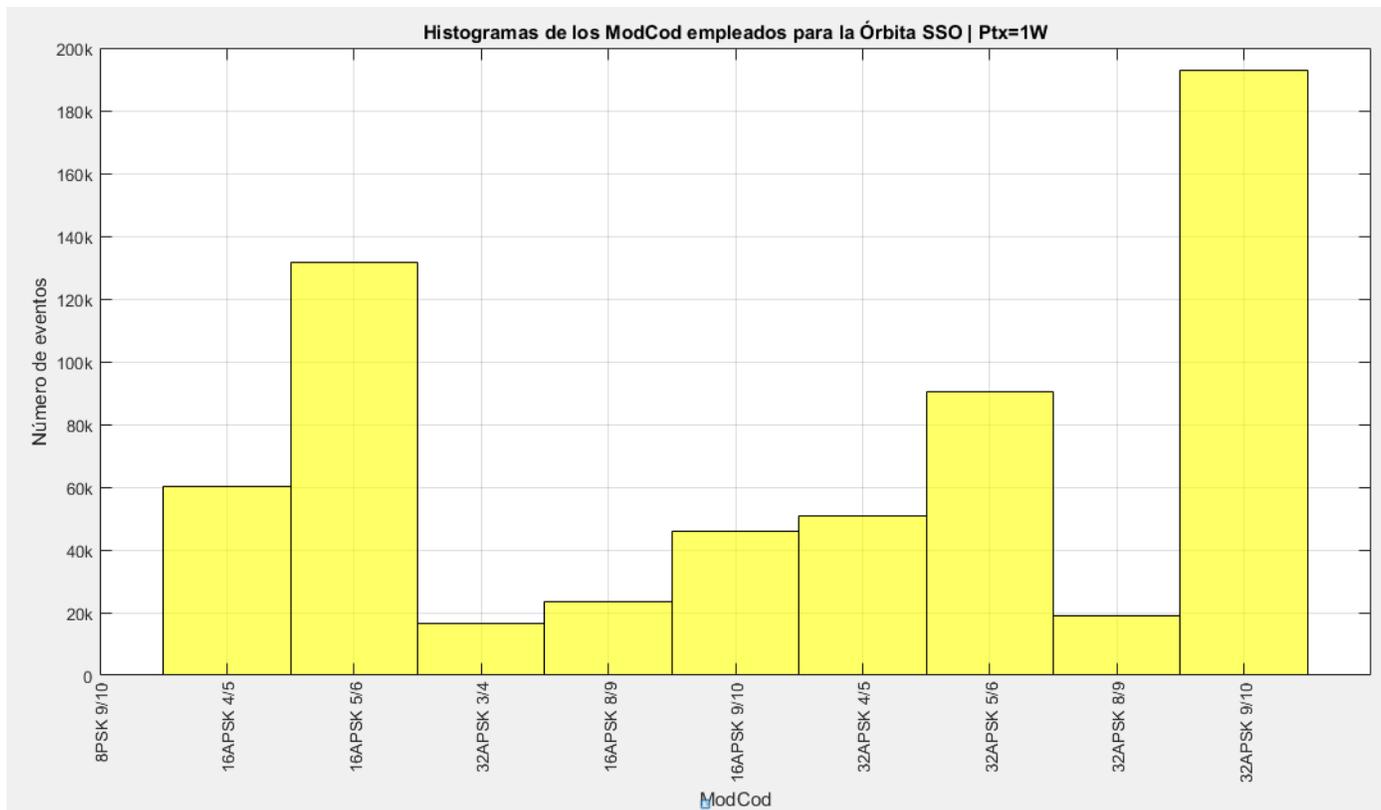


Figura 3.20. Histograma de los ModCod para la órbita SSO | Ptx=1W.



Es evidente que el uso de una potencia de transmisión menor, en este caso 0.5W, significa que el orden de modulación y codificación a implementar muchas veces será de un orden menor comparado con el empleo de una potencia mayor. Es importante señalar que la modulación 8PSK sólo es utilizada cuando la potencia es de 0.5W pues al ser la configuración de 1W el menor de los ModCod es 16APSK 4/5. De igual modo, es importante mencionar que con los parámetros y configuración propuesta el SNR nunca está por debajo de los 6dB por lo que la modulación QPSK de orden menor no resulta necesaria, al menos en la teoría.

En la figura 3.21 se hace una gráfica comparativa de la tasa de datos resultante de los esquemas ACM, en verde, y CCM, en azul. Para el esquema CCM, como ya se había planteado, el comportamiento es constante manteniendo un valor de 7.90 Mbps en todo momento. El esquema ACM muestra variaciones conforme progresa la simulación y, al igual que las gráficas de SNR, presenta valores mínimos al inicio de los avistamientos con 28.54 Mbps y máximos en el punto de máxima cercanía entre los satélites y la ET con 35.68Mbps para la órbita ISS.

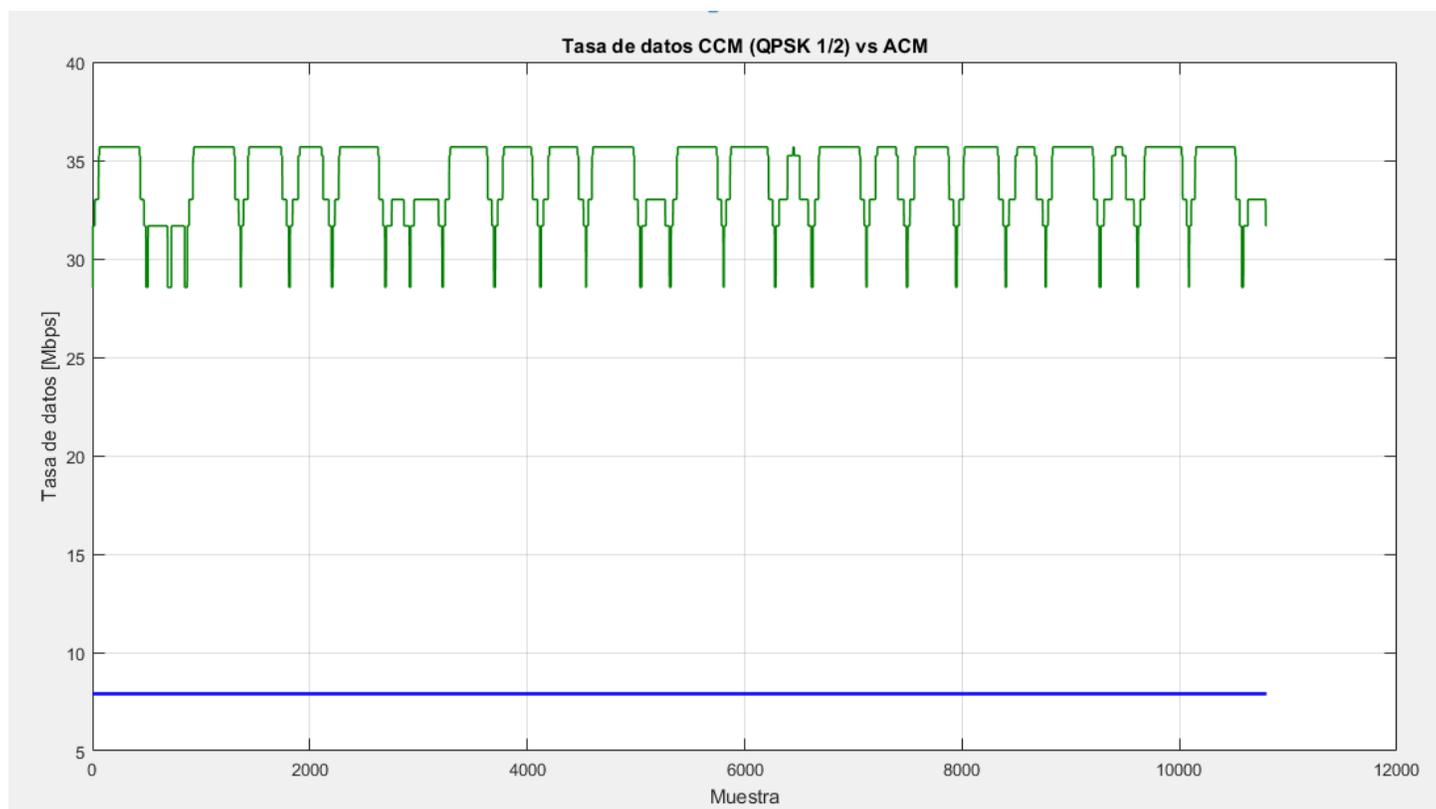


Figura 3.21. Tasa de datos CCM vs ACM para la órbita ISS.



Finalmente, se muestran las tablas 3.8 y 3.9 con un concentrado de toda la información obtenida para cada una de las órbitas propuestas considerando una potencia de transmisión de 1W.

**Tabla 3.8. Resumen de resultados del análisis orbital y de comunicaciones para la Órbita ISS.**

Órbita ISS (406km; 51.64°)   Periodo de simulación: Ene-Dic 2020							
Tiempo de muestreo: 1 segundo							
Valor	Duración de pases		Número de Pases	CCM	Fotografías (38.14 MB)	ACM	Fotografías (38.14 MB)
Máximo	499''	8'19''	5/día	7.90 Mbps	13/pase	35.68 Mbps	58/pase
Mínimo	44''	44''	3/día	7.90 Mbps	1/pase	28.54 Mbps	4/pase
Promedio	391''	6'31''	3.76/día	7.90 Mbps	9/pase	34.24 Mbps	41/pase
Total	537,300''	149.25h	1374/año	520.85 GB	13,984/año	2257.9 GB	60,622/año

**Tabla 3.9. Resumen de resultados del análisis orbital y de comunicaciones para la Órbita SSO.**

Órbita SSO (600km; 98°)   Periodo de simulación: Ene-Dic 2020							
Tiempo de muestreo: 1 segundo							
Valor	Duración de pases		Número de Pases	CCM	Fotografías (40.12 MB)	ACM	Fotografías (40.12 MB)
Máximo	620''	10'20''	5/día	7.90 Mbps	15/pase	35.68 Mbps	68/pase
Mínimo	19''	19''	3/día	7.90 Mbps	0.46/pase	25.33 Mbps	1/pase
Promedio	488''	8'8''	3.52/día	7.90 Mbps	12/pase	31.08 Mbps	47.25/pase
Total	628,488''	174.58h	1286/año	608.59 GB	15,533/año	2394.7 GB	61,121/año

Con base en este análisis y considerando una misión CubeSat de observación terrestre se puede establecer un valor promedio de 134 fotografías descargadas diariamente utilizando una órbita SSO con una potencia de transmisión de 500mW. Si los objetivos de la misión pueden cumplirse tomando en cuenta esta cifra de fotografías descargadas entonces se puede adoptar la solución propuesta.



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



## **Capítulo 4. Implementación del estándar DVB-S2 en un SDR**

Una vez que el análisis orbital y de comunicaciones ha sido realizado se procede a plantear el proceso de implementación del estándar DVB-S2 en un sistema de comunicaciones basado en un transmisor SDR enteramente funcional. Como primera iteración se plantea el desarrollo de una mesa de pruebas conformada por tres elementos principales:

### **1. Procesamiento en banda base:**

Realizado por el programa GNU Radio Companion (GRC) implementado en un CPU. Este elemento es el encargado de procesar el estándar DVB-S2.

### **2. Front End Analógico:**

Basado en un transceiver COTS. En este caso HackRF One es la opción seleccionada. Encargado de transmitir la señal de RF.

### **3. Receptor Satelital:**

Sistema de recepción DVB-S2. Es un dispositivo comercial tipo COTS diseñado para recibir señales del estándar DVB-S2.

En este capítulo el estudio se centrará en el diagrama a bloques de DVB-S2 configurado e implementado en las librerías del ambiente gráfico de GNU Radio Companion. Se describen los módulos que lo componen y se realiza una primera simulación dentro del programa.

Posteriormente, se describen las características y parámetros principales de los elementos de hardware que componen a la mesa de pruebas, tanto en la etapa de transmisión como de recepción. De igual modo, se mencionan los procedimientos de instalación y configuración del HackRF One.

Finalmente, se realizan pruebas de comunicación y desempeño del SDR implementado en la mesa de pruebas con el apoyo de un analizador de espectros y un monitor donde se podrá visualizar la señal DVB-S2 recibida. De este modo, es posible hacer un análisis comparativo entre el desempeño teórico y el experimental del estándar obteniendo la tasa de datos experimental del SDR implementado.



#### 4.1. SIMULACIÓN DE DVB-S2 EN GNU RADIO

Para el desarrollo de las simulaciones se utiliza una computadora Dell con un sistema operativo Ubuntu 18.04.2 LTS de 64 bits, el cual se ha configurado de manera nativa. Esto es importante pues diversos intentos de integración entre GNU Radio corriendo en máquinas virtuales y el HackRF One fueron fallidas. Se cree que la razón de las fallas es la limitada velocidad de transmisión de los puertos USB cuando se emplean máquinas virtuales.



Figura 4.1. Sistema Operativo Ubuntu 18.04.2 LTS.

La versión de GNU Radio Companion (GRC) instalada es la 3.7.13.4. Se hace especial hincapié en las versiones utilizadas pues es reconocida la existencia de diversas incompatibilidades entre las diferentes versiones de los sistemas operativos, los programas de radio simulado y los transceivers adaptados a estos.

El proceso de instalación de GRC en Ubuntu es enlistado en el **Anexo B1**.



Figura 4.2. GNU Radio Companion 3.7.13.4.



Como se explica en el capítulo 1, GRC es una plataforma que funge como la arquitectura de software de los sistemas SDR. Su configuración y desarrollo es sumamente flexible y está basado en software libre por lo que es posible tener acceso al código desarrollado para los diferentes esquemas de comunicación implementados en este.

Durante la primera etapa de desarrollo de este proyecto de investigación se trabajó con Matlab Simulink, sin embargo la restricción de acceso a su código y el licenciamiento requerido para el desarrollo de códigos compatibles con SDRs desde la misma plataforma hizo que no fuera posible continuar trabajando con la misma.

GNU Radio contiene diversas librerías enfocadas a la implementación de sistemas de Televisión Digital tales como: CATV, ATSC, DVB-T, DVB-S, DVB-S2, entre otros. Ver fig. 4.3.

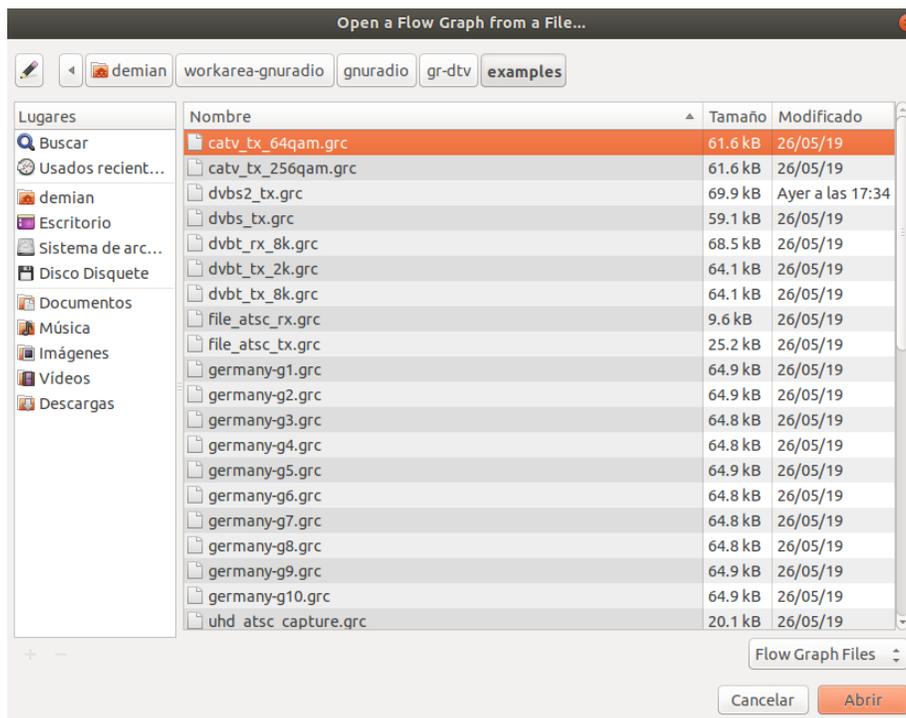


Figura 4.3. Librerías enfocadas a la Televisión Digital en GNU Radio.

Dentro de estas librerías se encuentra la implementación de un transmisor DVB-S2 llamado “dvbs2\_tx.grc”. En la figura 4.4 se muestra la representación gráfica de esta implementación.

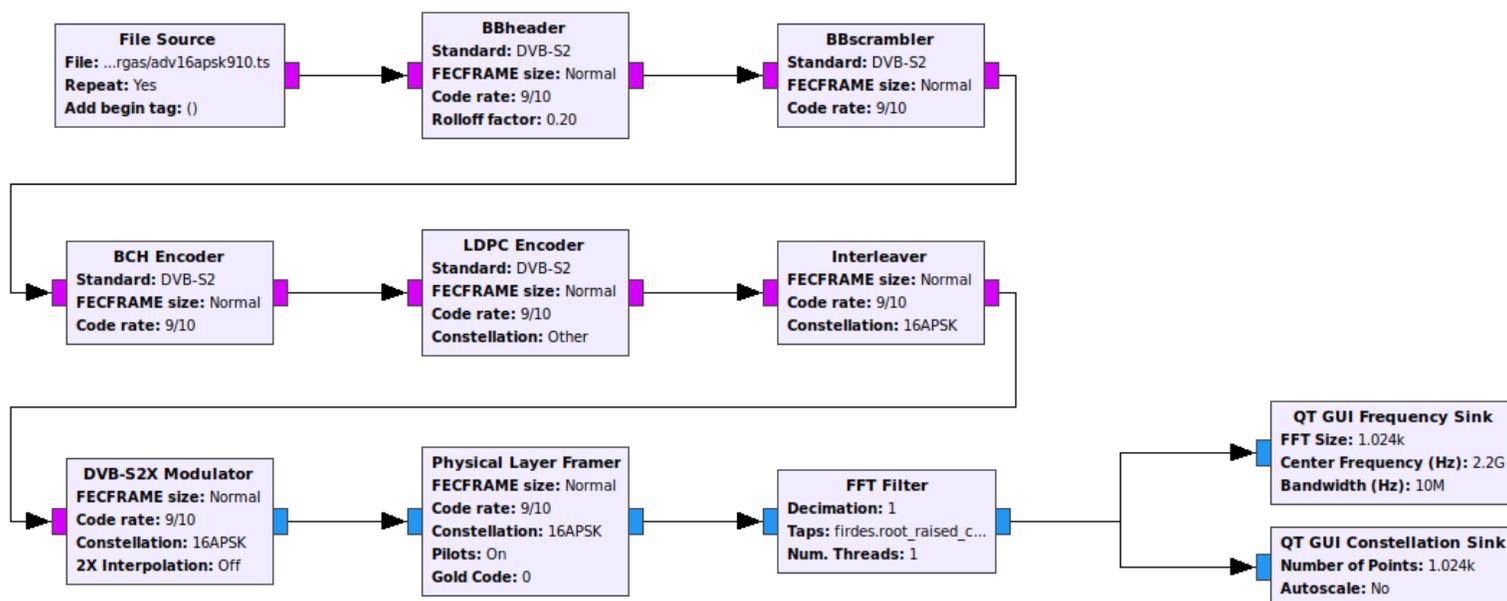


Figura 4.4. Diagrama de bloques del transmisor DVB-S2 en GNU Radio.

Fue desarrollado por Ron Economos [4\_1] y se compone de los bloques BBHeader, BBScrambler, BCH Encoder, LDPC Encoder, Interleaver, Modulación, PL Frame y Filtro lo cual le permite ser compatible con las siguientes características del estándar:

- Modulaciones QPSK, 8PSK, 16APSK y 32 APSK
- Fec coding de  $\frac{1}{4}$ ,  $\frac{1}{3}$ ,  $\frac{2}{5}$ ,  $\frac{2}{3}$ ,  $\frac{3}{4}$ ,  $\frac{4}{5}$ ,  $\frac{5}{6}$ ,  $\frac{8}{9}$  y  $\frac{9}{10}$
- Esquema CCM
- FECFrame normal de 64,800 bits y corto de 16,200 bits
- Transport Stream Simple
- Roll-off de 0.2, 0.25 y 0.35

Sin embargo presenta diversas limitantes entre las que se encuentran:

- Esquema VCM y ACM
- Multiple Transport Streams
- Stream genérico simple y múltiple
- Detección de paquetes nulos
- Inserción de frame dummy

Con esto en mente, la primera iteración de la mesa de pruebas estará únicamente enfocada al esquema CCM.

A continuación se describirán brevemente cada uno de los bloques que componen esta implementación, tomando como referencia el desglose del estándar que se realizó en el Capítulo 1.



- **File Source**

Bloque que alimenta al sistema con una señal de video en formato MPEG-2 lo cual representa el Transport Stream de entrada. Este archivo está previamente codificado a una tasa de símbolos relacionado directamente con el ModCod seleccionado para estar conformado por paquetes de 188 Bytes. En el **Anexo B2** se describen los pasos para dar formato a un archivo MP4 y convertirlo a un archivo MPEG-2 con extensión .ts el cual es compatible con el stream de transporte del estándar.

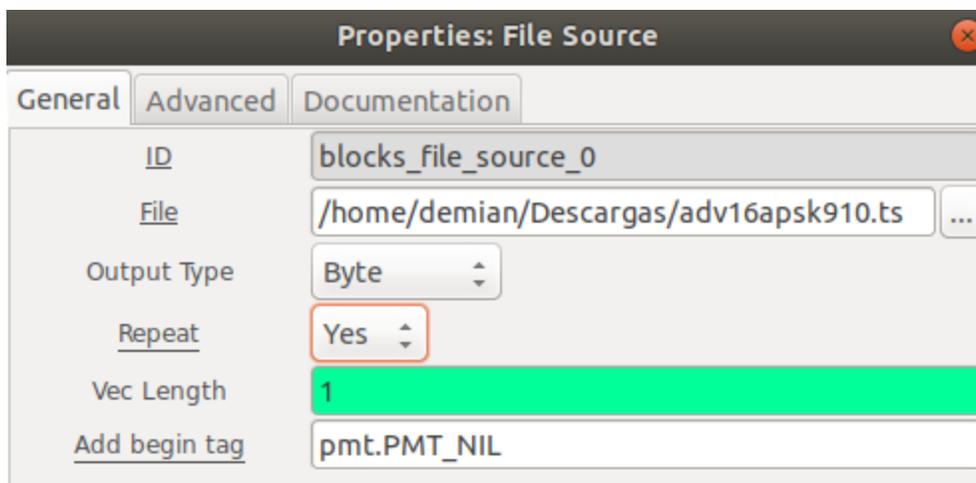
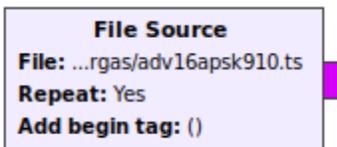
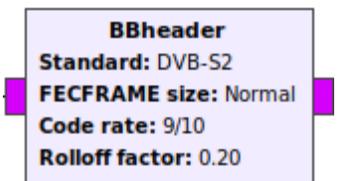


Figura 4.5. Bloque File Source y configuración general.

- **BBheader**

Es el encargado de encapsular los paquetes MPEG-2 provenientes del File Source, añadirles un BBheader y entregar un BBFrame a su salida. Las características y tamaño de este frame van a depender del FEC y el roll off seleccionado, de tal modo que deberá tener Kbch bits de acuerdo a la tabla 1.6.



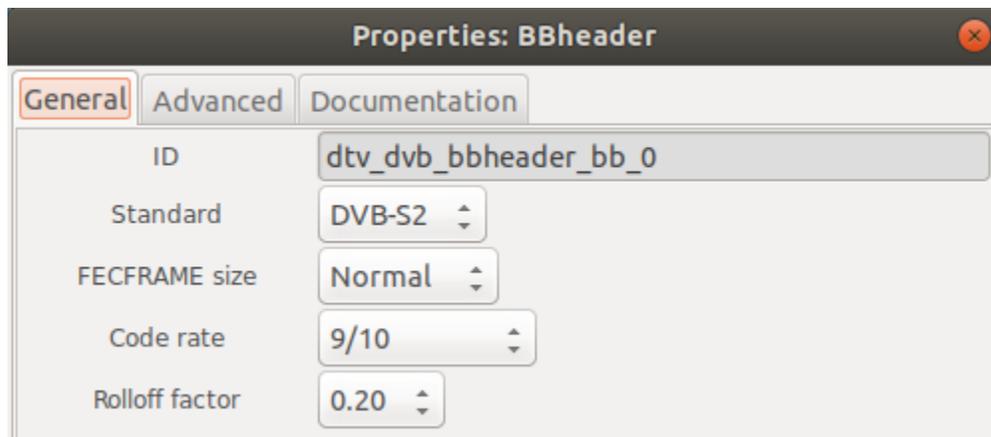


Figura 4.6. Bloque BBheader con configuración de FEC 9/10.

- **BBscrambler**

Aleatoriza los frames provenientes del BBheader considerando el polinomio generador PRBS:  $1+X^{14}+X^{15}$

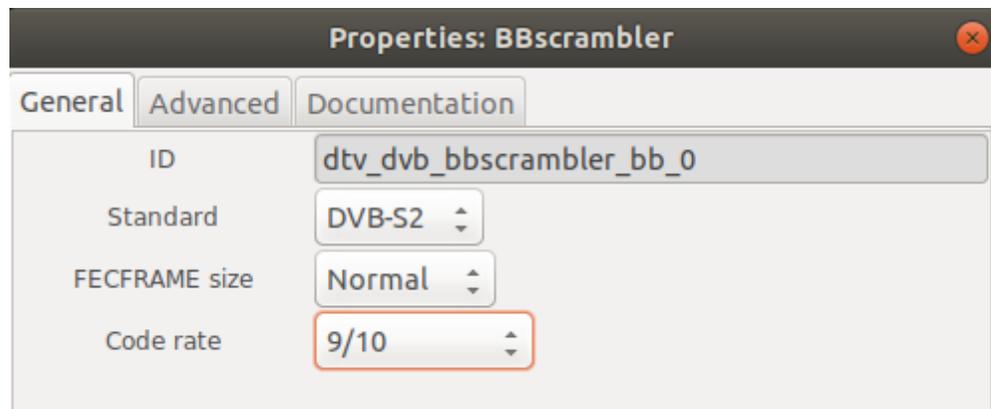
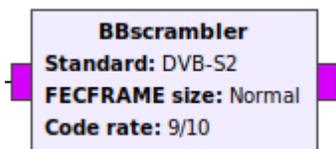


Figura 4.7. Bloque BBscrambler. FECFRAME normal y FEC rate 9/10.

• **BCH Encoder**

Bloque de codificación BCH. Recibe un BBFRAME aleatorizado y le añade un bloque BCHFEC generado por la multiplicación de los polinomios BCH mostrados en la figura 4.9.

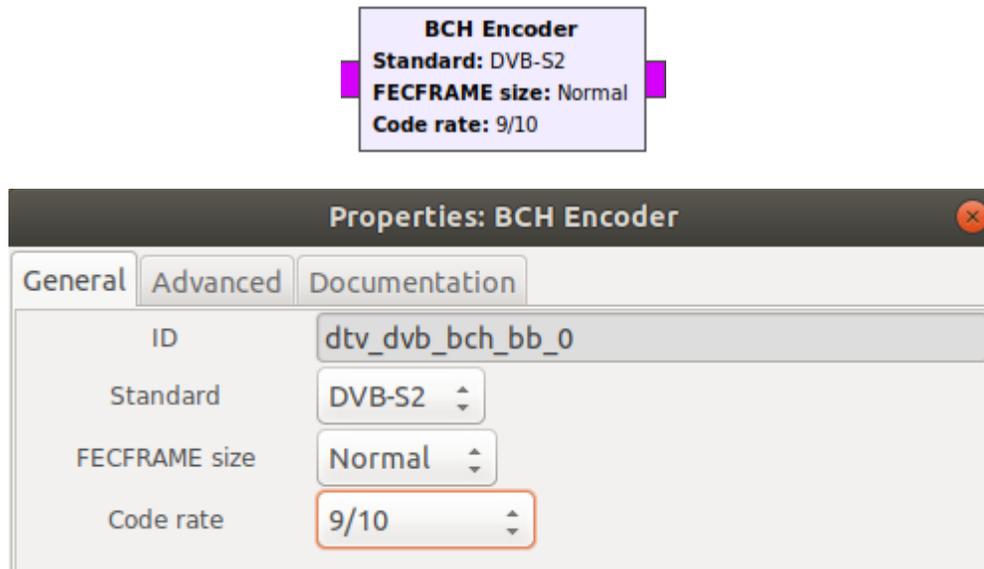


Figura 4.8. Bloque BCH Encoder configurado a tamaño Normal con un FEC de 9/10.

$g_1(x)$	$1+x^2+x^3+x^5+x^{16}$
$g_2(x)$	$1+x+x^4+x^5+x^6+x^8+x^{16}$
$g_3(x)$	$1+x^2+x^3+x^4+x^5+x^7+x^8+x^9+x^{10}+x^{11}+x^{16}$
$g_4(x)$	$1+x^2+x^4+x^6+x^9+x^{11}+x^{12}+x^{14}+x^{16}$
$g_5(x)$	$1+x+x^2+x^3+x^5+x^8+x^9+x^{10}+x^{11}+x^{12}+x^{16}$
$g_6(x)$	$1+x^2+x^4+x^5+x^7+x^8+x^9+x^{10}+x^{12}+x^{13}+x^{14}+x^{15}+x^{16}$
$g_7(x)$	$1+x^2+x^5+x^6+x^8+x^9+x^{10}+x^{11}+x^{13}+x^{15}+x^{16}$
$g_8(x)$	$1+x+x^2+x^5+x^6+x^8+x^9+x^{12}+x^{13}+x^{14}+x^{16}$
$g_9(x)$	$1+x^5+x^7+x^9+x^{10}+x^{11}+x^{16}$
$g_{10}(x)$	$1+x+x^2+x^5+x^7+x^8+x^{10}+x^{12}+x^{13}+x^{14}+x^{16}$
$g_{11}(x)$	$1+x^2+x^3+x^5+x^9+x^{11}+x^{12}+x^{13}+x^{16}$
$g_{12}(x)$	$1+x+x^5+x^6+x^7+x^9+x^{11}+x^{12}+x^{16}$

Figura 4.9. Polinomios BCH [4\_2].



- **LDPC Encoder**

Tiene como entrada un frame BBheader con un bloque BCHFEC añadido, al cual adiciona un bloque LDPCFEC conforme a lo descrito en [4\_2].

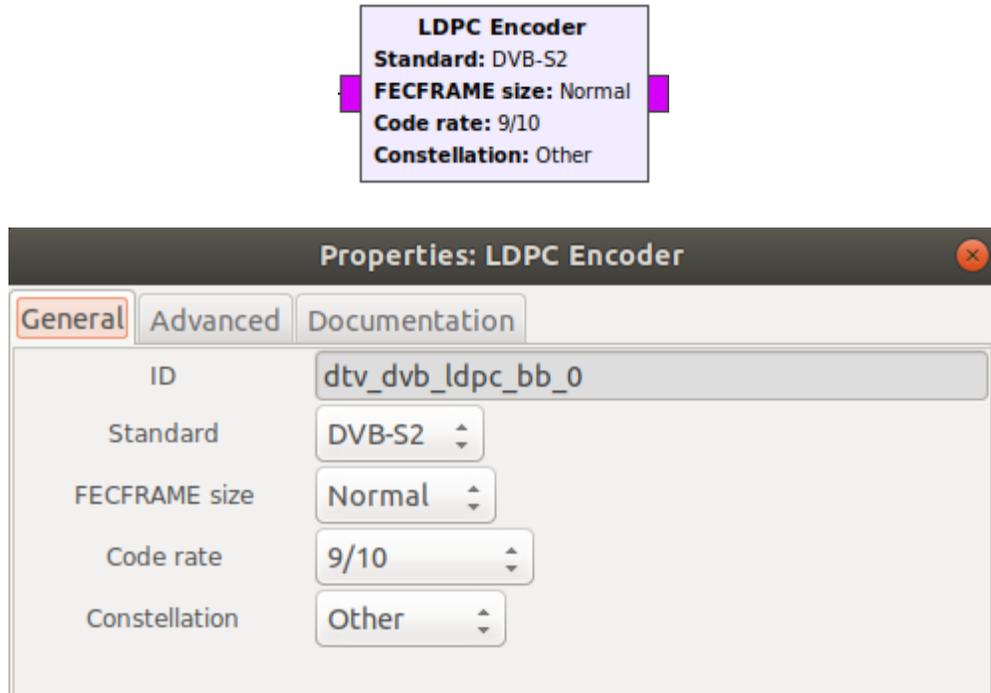


Figura 4.10. LDPC Encoder.

- **Interleaver**

Entrelaza el FECFRAME recibido conforme a la técnica de interleaving descrita en la figura 1.15. Su salida es un frame de 64,800 bits entrelazado.

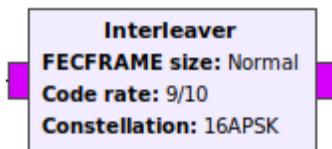


Figura 4.11 Bloque de Interleaving.

- **Modulator**

Procesa el FECFRAME proveniente de la etapa anterior y le asigna valores I-Q dependiendo de la modulación seleccionada. Genera un bloque XFECFRAME.

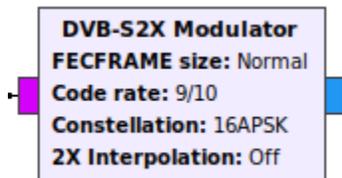


Figura 4.12 Modulador DVB-S2 configurado a un ModCod de 16APSK 9/10.

- **PL Frame**

Del XFECFRAME recibido le asigna un PL header, inserta portadoras piloto moduladas en BPSK y aplica Scrambling al frame. Entrega a su salida un PLFRAME.

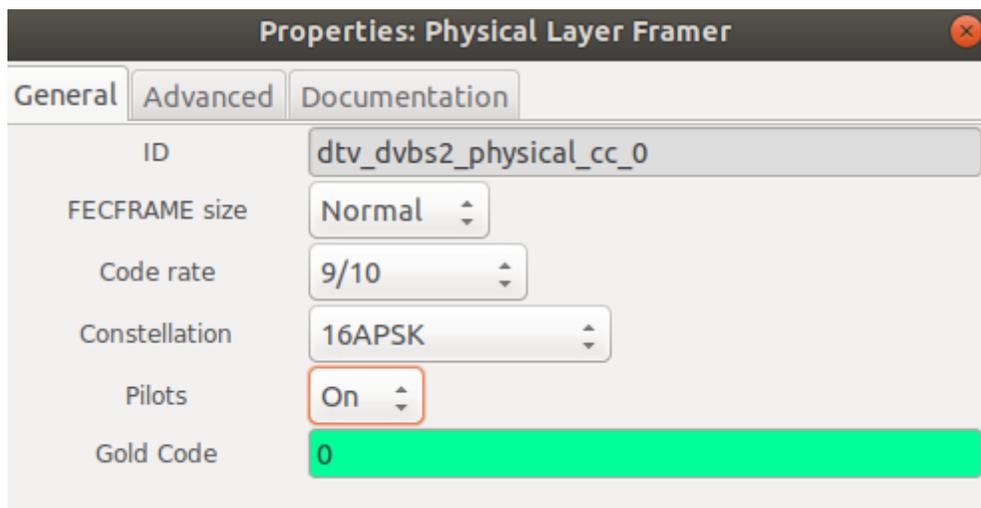
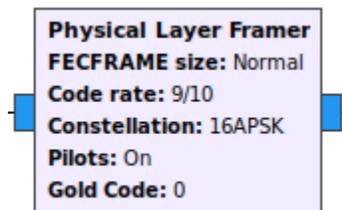
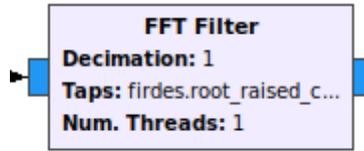


Figura 4.13. Physical Layer Frame. 16APSK 9/10 con portadoras piloto activadas.

- **FFT Filter**

Implementa un filtro coseno invertido aplicado al PLFRAME entrante considerando el factor de Roll Off. También se puede configurar el valor de decimación en frecuencia y el orden del filtro (taps).

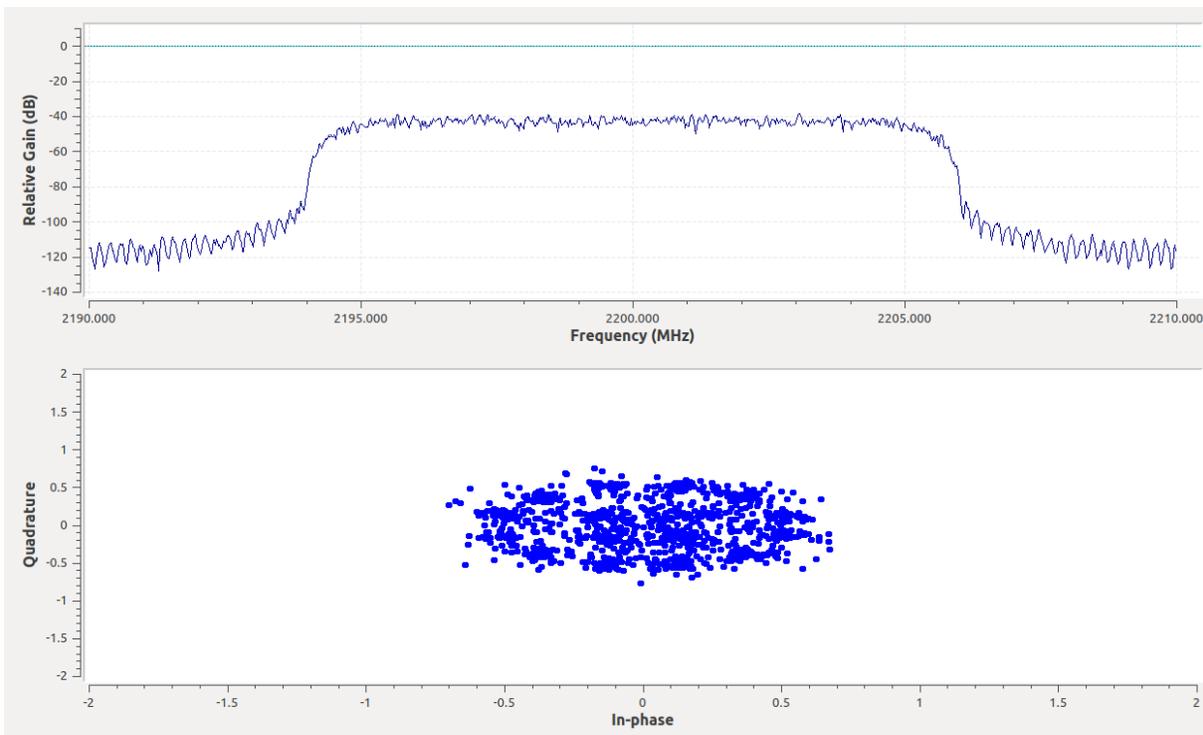


**Figura 4.14. FFT Filter.**

Finalmente, la señal resultante, modulada y filtrada, es llevada a un analizador de frecuencias virtual y a una herramienta gráfica para visualizar la constelación IQ de la señal.

En este primer caso la señal de video es procesada con un ModCod de 16APSK 9/10.

A continuación, figura 4.15, se muestra su espectro en frecuencia y constelación.



**Figura 4.15. Espectro en frecuencia y constelación de la señal DVB-S2 a la salida del modulador.**

Como puede verse en el espectro en frecuencia, fig. 4.16, se genera una señal con frecuencia central de 2.2 GHz y un ancho de banda de 10 MHz. El nivel de la señal es de -40 dB entre los 2195 MHz y 2205 MHz para después tener una atenuación de aproximadamente 50 dB en la frecuencia de 2194 MHz.

En la constelación en cuadratura y en fase se aprecia la disposición 16APSK de la señal con ruido provocado por la respuesta del filtro.

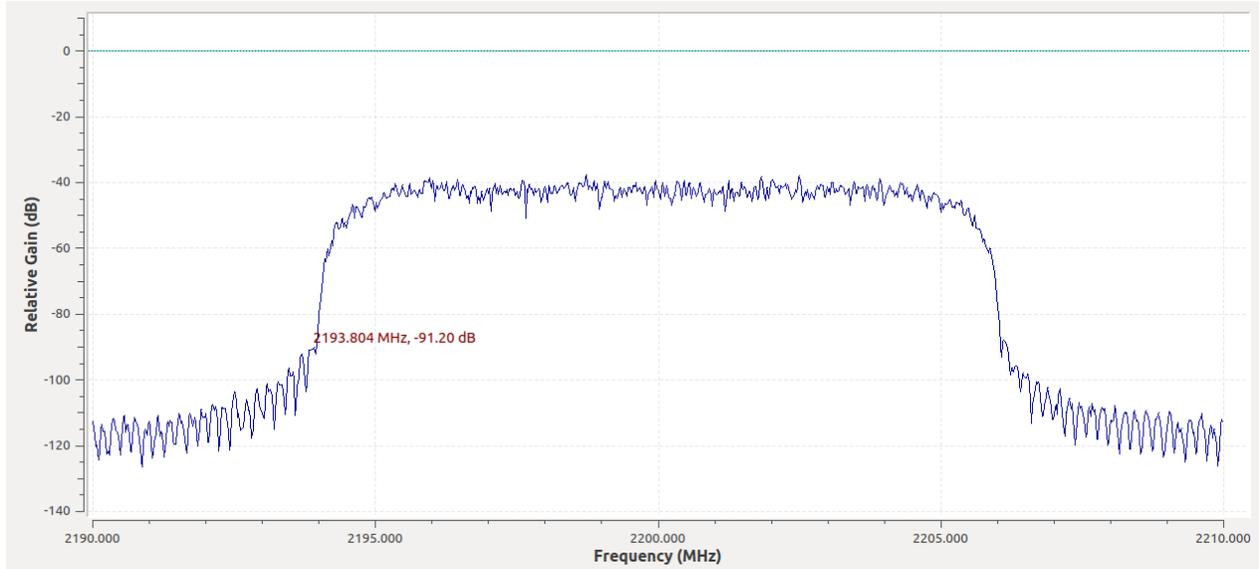


Figura 4.16. Espectro en frecuencia de la señal DVB-S2 con roll off de 0.20.

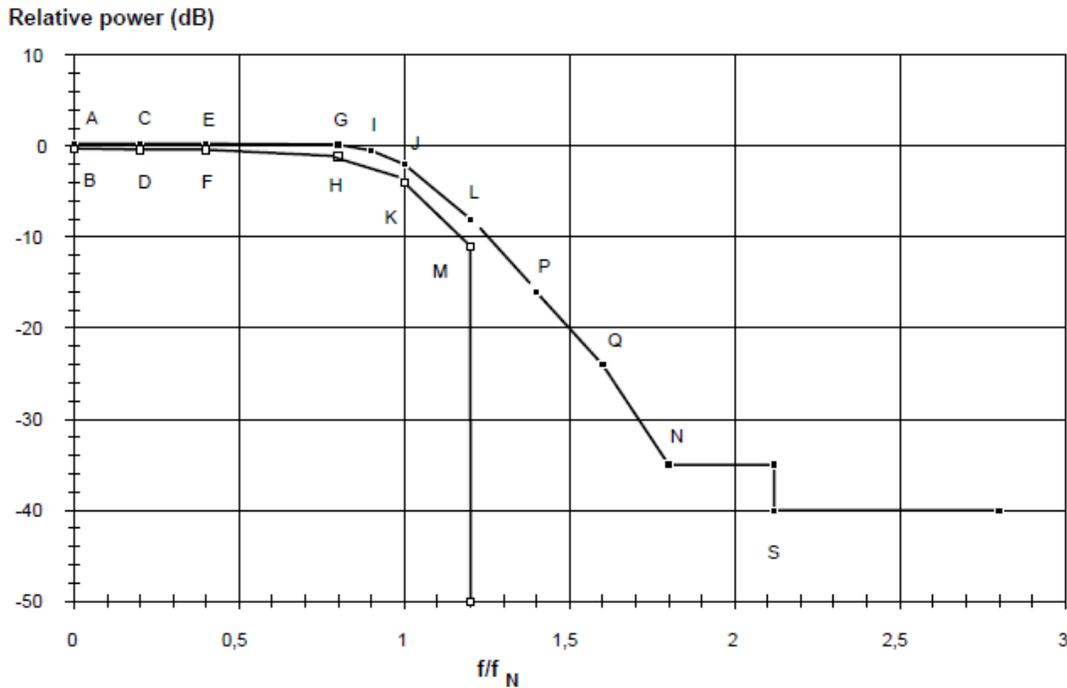


Figura 4.17. Máscara del espectro de la señal esperado a la salida del modulador DVB-S2 para un factor de roll off de 0.35 [4\_2].

Considerando la máscara del espectro de la señal mostrada en la figura 4.17 el nivel de atenuación esperado es justamente de 40 dB a una distancia de  $2f_N$  de la frecuencia central, es decir a los 2015 MHz. Para el caso del espectro generado en 4.16, estas condiciones se cumplen con una tolerancia importante dentro de la simulación por lo que se puede concluir que la señal generada es adecuada para su transmisión. El siguiente paso es adaptar un front end analógico a la etapa de procesamiento en banda base para hacer posible la transmisión de la señal modulada y así implementar la primera iteración del SDR a diseñar.

## 4.2. INTEGRACIÓN DE LA MESA DE PRUEBAS

Para comprobar el funcionamiento del procesamiento de la señal implementada por GNU Radio bajo el estándar DVB-S2 se decide integrar un front end analógico embebido en el dispositivo HackRF One. Con esta integración se busca obtener un transmisor funcional capaz de propagar una señal de televisión DVB-S2 desde una mesa de pruebas. Esta integración es considerada como la primera iteración del SDR propuesto.

Al considerar nuevamente la figura 1.1 se hace una propuesta de implementación para la primera iteración del SDR mostrado en la fig. 4.18.

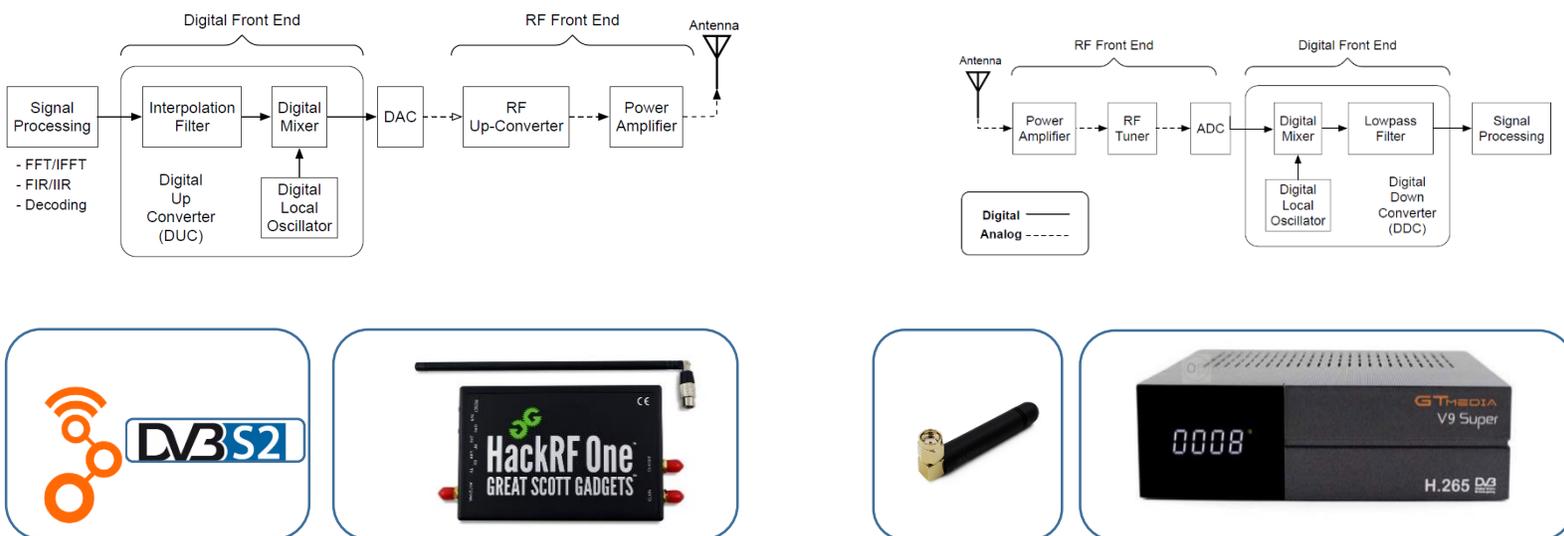


Figura 4.18. Mesa de pruebas propuesta para el transmisor DVB-S2 basado en SDR.

En el segmento de transmisión el bloque de Signal Processing es realizado por la plataforma GNU Radio corriendo en el CPU con el sistema operativo Ubuntu mencionado anteriormente. En este se implementará el estándar DVB-S2.

Para el Digital Front End, así como para el segmento del Front End analógico, se empleará un transceiver HackRF One de la marca Great Scott Gadgets capaz de transmitir frecuencias entre 1 MHz y 6 GHz.

Para el segmento receptor, las funciones del front end analógico y la etapa de procesamiento lo lleva a cabo un receptor satelital comercial de la marca GTMedia modelo V9 Super.

En relación a las antenas de transmisión y recepción se utilizan un par de antenas dipolo modelo G401-1 con una ganancia de 3 dBi.

Esta primera iteración del sistema SDR puede ser entendida como una mesa de pruebas pues carece de portabilidad al ser un CPU el responsable de procesar la banda base de la etapa transmisora.

A continuación se hace una descripción del HackRF One, del receptor satelital y de las antenas empleadas. Así mismo se muestran las etapas de configuración y pruebas de ambos elementos.

### 4.2.1 Front End HackRF One

Como se menciona, para la implementación de la mesa de pruebas, se propone el uso del HackRF One [4\_3] mostrado en la figura 4.19.



**Figura 4.19. HackRF One [4\_3].**

Se trata de un Front End Analógico capaz de transmitir o recibir señales de radio en el rango de 1 MHz a 6 GHz. Está diseñado para permitir la prueba y el desarrollo de la etapa de RF de diversos esquemas de radiocomunicación basados en SDR. Es una plataforma desarrollada con código abierto y puede utilizarse como un periférico USB o programarse para un funcionamiento independiente.

#### Principales características

- Tiene un rango de operación en frecuencia desde 1 MHz hasta 6 GHz.
- Es un transceiver con capacidad de operación half-duplex.
- Tiene una capacidad de muestreo de hasta 20 millones de muestras por segundo en función del tipo de controlador USB 2.0 que incluya el computador al que se conecta.
- Muestreo de las señales con 8 bits en cuadratura, donde 8 bits serán para la componente en fase I y 8 serán para la componente en cuadratura Q.
- Es compatible con los principales programas para SDR tanto en Windows como en Linux, lo que incluye GNU Radio.
- Puede configurar vía software los amplificadores de ganancia, con 3 etapas dedicadas para recepción y 2 etapas para transmisión.



- Puede configurar vía software los filtros de señal en banda base, con un máximo de ancho de banda de señal de 28 MHz, y con una caída de 3 dB hasta 30 MHz.
- Permite controlar vía software la potencia suministrada al puerto de la antena, con hasta 50 mA a 3.3 V.
- Conector de antena SMA hembra.
- Un conector SMA hembra para sincronizar el reloj, tanto a la entrada como a la salida, lo que permite conectar varios HackRF One y hacerlos trabajar conjuntamente, reduciendo los problemas de jitter.
- Uso de la interfaz USB 2.0 High Speed.

### Potencia de transmisión y recepción

La potencia de transmisión máxima del HackRF One varía según la frecuencia de operación:

- **10 MHz a 2150 MHz:** 5 dBm a 15 dBm (3.16 a 31.6 mW)
- **2150 MHz a 2750 MHz:** 13 dBm a 15 dBm (19.9 a 31.6 mW)
- **2750 MHz a 4000 MHz:** 0 dBm a 5 dBm (1mW a 3.16mW)
- **4000 MHz a 6000 MHz:** -10 dBm a 0 dBm (0.1 mW a 1 mW)

En la mayor parte del rango de frecuencias de hasta 4 GHz, la potencia máxima de TX está entre 0 y 10 dBm. El rango de frecuencia con el mejor rendimiento es de 2150 MHz a 2750 MHz.

La potencia máxima de recepción es de -5 dBm.

### Antena [4\_4]

La antena modelo ANT500 de Great Scott Gadgets está incluida con este front end. Es una antena telescópica diseñada para operar en un rango de frecuencias de 75 MHz a 1 GHz. Su longitud total es configurable desde 20 cm a 88 cm. Cuenta con un conector macho SMA, un eje giratorio y un codo ajustable. Su impedancia es de 50 ohms. A pesar de que en la práctica es capaz de recibir señales arriba de 1GHz se prefiere utilizar la antena G401-1 dado a su mejor desempeño y ganancia.

#### 4.2.1.1 Diagrama de bloques HackRF One

En la figura 4.20 se muestra el PCB y los elementos electrónicos que conforman este transceiver donde la entrada del mismo se encuentra en el lado derecho del dispositivo y la salida a la antena de lado izquierdo.

Como se menciona, el HackRF One es un Front End Analógico compuesto por las etapas de potencia, conversión en banda base, filtrado y amplificación.

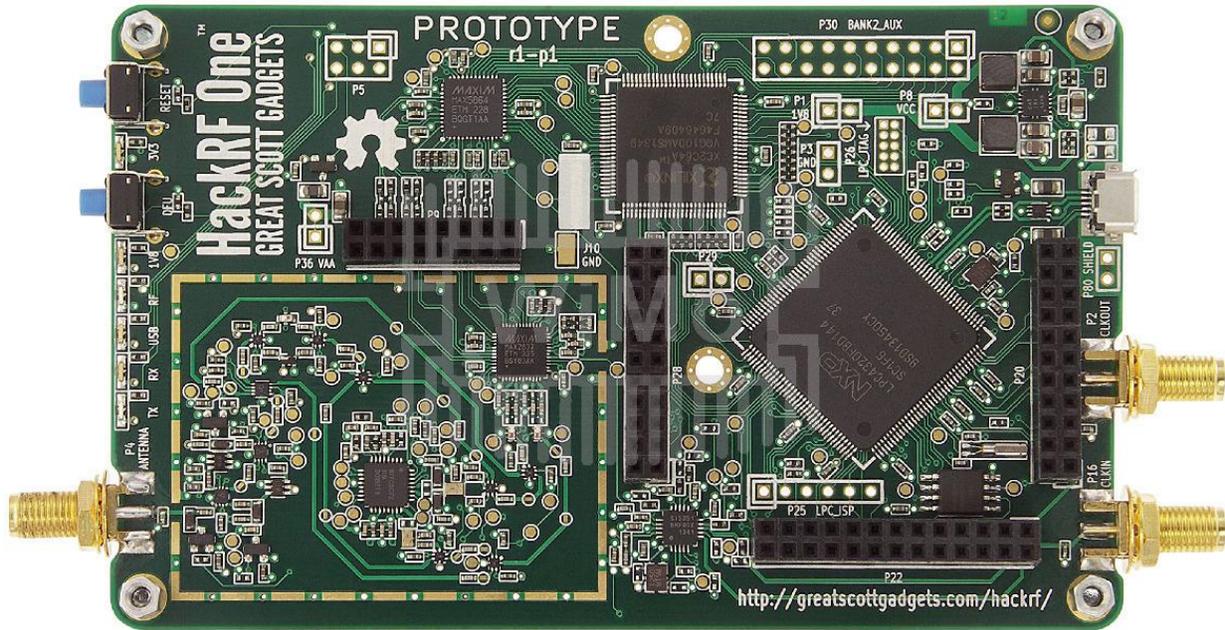


Figura 4.20. PCB y elementos electrónicos del HackRF One [4\_3].

A continuación, fig. 4.21, se muestra el diagrama de bloques del HackRF One y se presenta una breve descripción de sus elementos:

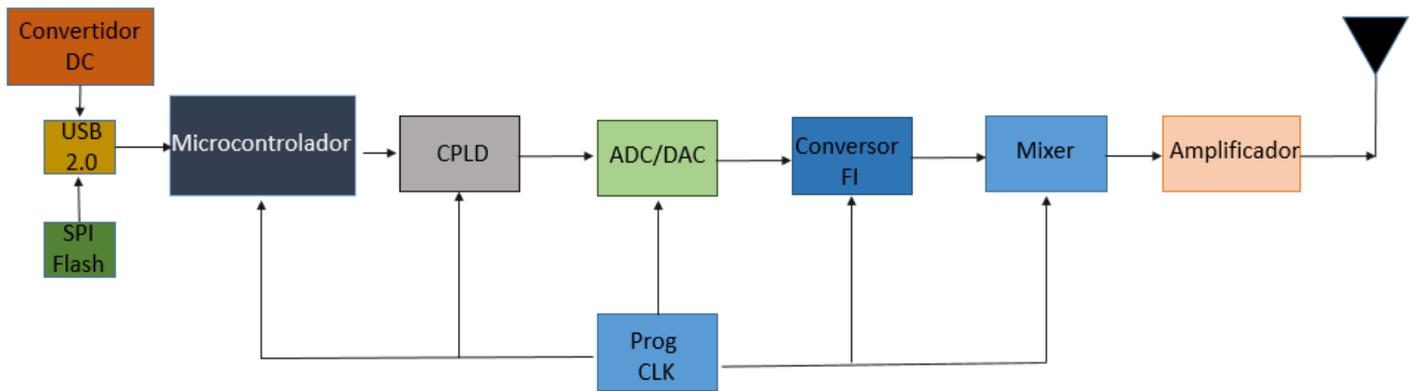


Figura 4.21. Diagrama de bloques del HackRf One [4\_3].

• **USB 2.0**

Es la interfaz encargada de suministrar la potencia al dispositivo HackRF One y de permitir la transmisión/recepción de información entre la computadora y el circuito electrónico del HackRF.

• **Convertidor de DC**

Es un convertidor reductor de voltaje con dos salidas a voltajes de 3.3V y 1.8V que forma la fuente de alimentación del HackRF. Está basado en el chip de Texas Instrument TPS62410.



• **SPI Flash**

Es una memoria flash conectada al microcontrolador ARM, la cual puede ser reconfigurada. Permite la configuración del microcontrolador principal y de la CPLD, cada vez que se inicia el HackRF One.

• **PROG CLK**

Es un circuito integrado SI5351 que genera varios pulsos de reloj de manera simultánea a partir de un único cristal como referencia fundamental.

• **Microcontrolador**

Se trata de un microcontrolador ARM Cortex-M4/M0+ de 2 núcleos LPC43, utilizado para controlar las funciones y procesos básicos del HackRf One.

• **CPLD**

Es un dispositivo lógico programable modelo CoolRunner-II que se usa como interfaz entre el convertidor ADC/DAC y el microcontrolador. Funge como puente lógico entre los protocolos de recepción de datos de ambos bloques.

• **ADC/DAC**

Es un convertidor dual que se compone de un ADC y un DAC MAX5864 de hasta 22 MHz de operación. Tiene capacidad de operación full-duplex, por lo que ambos conversores pueden funcionar simultáneamente.

• **Convertor de Frecuencia Intermedia - BB**

Es un transceiver MAX2837 que puede recibir/transmitir señales desde/hacia la banda de 2.3 a 2.7 GHz directamente desde banda base, por lo que se usa como un convertidor de frecuencia intermedia.

• **Mixer**

Es un mezclador RFFC5072 que se emplea, en el modo de recepción, para la conversión de señales a un rango de frecuencias intermedias de 2.3 y 2.7 GHz usando una portadora local generada en el mezclador. En el modo de transmisión, desplaza la señal entregada por el convertor de FI desde 2.3 a 2.7 GHz hasta cualquier frecuencia distinta entre 1 MHz y 6 GHz, usando una portadora local.

• **Amplificador**

Se trata de un par de amplificadores GaAs MMIC modelo MGA-81563 de Avago. Uno para la recepción y otro para la transmisión. Trabajan en frecuencias de 0.1 a 6 GHz y se utiliza como etapa final de amplificación en el transmisor y como inicial en el receptor. Su punto de compresión a 1dB es de 14.8 dBm@2GHz y su potencia de saturación es de +17 dBm@2GHz.

**4.2.1.2. Configuración HackRF One**

Para la instalación y configuración de HackRF One es necesario descargar dos elementos: HackRF tools y HackRFlib. Ambos pueden ser encontrados en [4\_5].



Los comandos utilizados para la instalación y descarga se muestran en el Anexo C.

Después, se realiza la comprobación de la correcta instalación y detección del HackRF aplicando las siguientes verificaciones:

a) Primeramente, se verifica si el dispositivo es reconocido correctamente por Ubuntu. Esto se logra ingresando en la Terminal el comando <hackrf\_info>, el cual mostrará la versión del hackrf-lib instalada, el número de serie, firmware y PartID del HackRF conectado. Si se recibe la información hackrf\_info versión y firmware versión entonces podemos garantizar que el hardware es reconocido por el sistema.

```
Archivo Editar Ver Buscar Terminal Ayuda
demian@Demian:~$ hackrf_info
hackrf_info version: git-509c8f4
libhackrf version: git-509c8f4 (0.5)
Found HackRF
Index: 0
Serial number: 0000000000000000a06063c82885851b
Board ID Number: 2 (HackRF One)
Firmware Version: git-509c8f4 (API:1.03)
Part ID Number: 0xa000cb3c 0x00504f6d
CPLD checksum: 0x05829db7
demian@Demian:~$
```

Figura 4.22. Resultado del comando hackrf\_info.

Como se puede observar en la figura anterior, el dispositivo ha sido reconocido correctamente con una versión de firmware git-509c8f4 y un serial number a06063c82885851b.

b) En segundo término se comprueba que el dispositivo sea capaz de recibir y transmitir información. Para ello se pueden utilizar diversas funciones disponibles para el HackRF One con el nombre de hackrf\_transfer:

```
-r <filename> # Receive data into file.

-t <filename> # Transmit data from file.

-w # Receive data into file with WAV header and automatic name.

# This is for SDR# compatibility and may not work with other software.

[-f set_freq_hz] # Set Freq in Hz

[-a set_amp] # Set Amp 1=Enable, 0=Disable.

[-l gain_db] # Set lna gain, 0-40dB, 8dB steps
```



```
[-i gain_db] # Set vga(if) gain, 0-62dB, 2dB steps

[-x gain_db] # Set TX vga gain, 0-47dB, 1dB steps

[-s sample_rate_hz] # Set sample rate in Hz (8/10/12.5/16/20MHz)

[-n num_samples] # Number of samples to transfer (default is unlimited).

[-b baseband_filter_bw_hz] # Set baseband filter bandwidth in MHz.

Possible values:
1.75/2.5/3.5/5/5.5/6/7/8/9/10/12/14/15/20/24/28MHz,
default < sample_rate_hz.
```

Figura 4.23. Opciones disponibles para `hackrf_transfer` [4\_5].

```
demian@Demian:~$ hackrf_transfer
specify one of: -t, -c, -r, -w
Usage:
-h # this help
[-d serial_number] # Serial number of desired HackRF.
-r <filename> # Receive data into file (use '-' for stdout).
-t <filename> # Transmit data from file (use '-' for stdin).
-w # Receive data into file with WAV header and automatic name.
  # This is for SDR# compatibility and may not work with other software.
[-f freq_hz] # Frequency in Hz [0MHz to 7250MHz].
[-i if_freq_hz] # Intermediate Frequency (IF) in Hz [2150MHz to 2750MHz].
[-o lo_freq_hz] # Front-end Local Oscillator (LO) frequency in Hz [84MHz to 5400MHz].
[-m image_reject] # Image rejection filter selection, 0=bypass, 1=low pass, 2=high pass.
[-a amp_enable] # RX/TX RF amplifier 1=Enable, 0=Disable.
[-p antenna_enable] # Antenna port power, 1=Enable, 0=Disable.
[-l gain_db] # RX LNA (IF) gain, 0-40dB, 8dB steps
[-g gain_db] # RX VGA (baseband) gain, 0-62dB, 2dB steps
[-x gain_db] # TX VGA (IF) gain, 0-47dB, 1dB steps
[-s sample_rate_hz] # Sample rate in Hz (4/8/10/12.5/16/20MHz, default 10MHz).
[-n num_samples] # Number of samples to transfer (default is unlimited).
[-S buf_size] # Enable receive streaming with buffer size buf_size.
[-c amplitude] # CW signal source mode, amplitude 0-127 (DC value to DAC).
[-R] # Repeat TX mode (default is off)
[-b baseband_filter_bw_hz] # Set baseband filter bandwidth in Hz.
Possible values: 1.75/2.5/3.5/5/5.5/6/7/8/9/10/12/14/15/20/24/28MHz, default <= 0.75 * sample_rate_hz.
[-C ppm] # Set Internal crystal clock error in ppm.
[-H hw_sync_enable] # Synchronise USB transfer using GPIO pins.
```

Figura 4.24. `hackrf_transfer` en terminal.

Primero se prueba la capacidad de recibir del HackRF. Para ello se utiliza la función de recibir la cual es `hackrf_transfer` `-r` y como nombre de archivo donde se guardarán los datos utilizaremos “Null”. Este nombre puede ser el que se desee. La frecuencia de operación que se probará será 2200 MHz. La línea de comando quedaría del siguiente modo:

**Hackrftransfer -r null -f 2200**

Y los resultados obtenidos son los siguientes:



```
demian@Demian:~$ hackrf_transfer -r null -f 2200
call hackrf_set_sample_rate(10000000 Hz/10.000 MHz)
call hackrf_set_hw_sync_mode(0)
call hackrf_set_freq(2200 Hz/0.002 MHz)
Stop with Ctrl-C
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
17.6 MiB / 1.000 sec = 17.6 MiB/second
17.6 MiB / 1.000 sec = 17.6 MiB/second
^Ccaught signal 2
10.2 MiB / 0.513 sec = 19.9 MiB/second

Exiting...
Total time: 13.51505 s
hackrf_stop_rx() done
hackrf_close() done
hackrf_exit() done
fclose(fd) done
exit
```

Figura 4.25. Resultados de la prueba de recepción con `hackrf_transfer`.

Con ello se demuestra que el HackRF One conectado recibe información a una frecuencia de 2.2 GHz y a una velocidad aproximada de 20 MBps.

La prueba opuesta, la de transmisión, se lleva a cabo con el comando `-t` por lo que la línea de comando de prueba es la siguiente:

**Hackrftransfer -t null -f 2200**

El archivo transmitido es el mismo que se acaba de generar. Los resultados obtenidos son los mostrados en la figura 4.26.



```
demian@Demian:~$ hackrf_transfer -t null -f 2200
call hackrf_set_sample_rate(10000000 Hz/10.000 MHz)
call hackrf_set_hw_sync_mode(0)
call hackrf_set_freq(2200 Hz/0.002 MHz)
Stop with Ctrl-C
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
5.5 MiB / 1.000 sec = 5.5 MiB/second

Exiting... hackrf_is_streaming() result: streaming terminated (-1004)
Total time: 14.00169 s
hackrf_stop_tx() done
hackrf_close() done
hackrf_exit() done
fclose(fd) done
exit
```

Figura 4.26. Resultados de la prueba de transmisión con `hackrf_transfer`.

Al igual que la prueba de recepción, la de transmisión ha arrojado resultados positivos y la primera prueba de TX/RX del HackRF One ha sido satisfactoria. En caso de que no exista TX/RX en esta prueba (0.00 MiBps) se recomienda revisar la comunidad de troubleshooting enfocada al HackRF One en [4\_6].

#### 4.2.2. Configuración de GNU Radio y HackRF One

Una vez que el HackRF One está instalado y funcionando, se procede a integrar GNU Radio con este. Para poder recibir/transmitir información entre el HackRF y GNU Radio se requiere de la librería `gr-OsmoSDR` desarrollado por el grupo de trabajo Open Source Mobile Communications [4\_7].

La librería `gr-osmosdr` permite que GNU Radio pueda recibir y enviar información hacia/desde diferentes dispositivos basados en SDR tales como FUNCube Dongle, OsmoSDR, RTL-SDR, MSi2500, SDRplay, SDR-IQ, AirSpy, rad10, HackRF, bladeRF, USRP/UHD, UMtrx, RedPitaya, FreeSRP, entre otros. Esta va a ser el puente entre el HackRF y GNU Radio.

Para validar si se cuenta con la librería instalada se deberá acceder a GNU Radio Companion y buscar en funciones el bloque `osmocom` como se muestra en la figura 4.27.

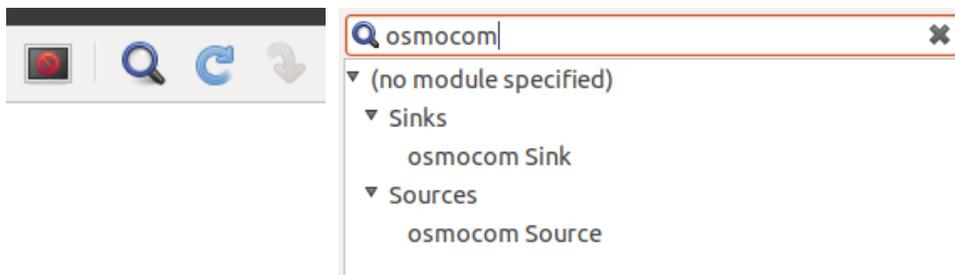


Figura 4.27. Bloques Osmocom source y sink cargados en GNU Radio.

En caso de no encontrarse ningún bloque significa que la librería no ha sido instalada para lo cual se recomienda revisar el proceso de instalación en [4\_8]. El mismo ha sido enlistado en el **Anexo C2**. Los bloques disponibles son osmocom Sink y osmocom Source para transmisión y recepción respectivamente. Ver fig. 4.28.

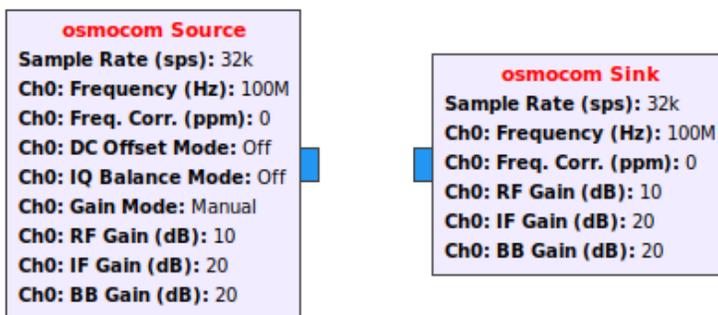


Figura 4.28. Bloques osmocom Source y osmocom Sink.

Las opciones y parámetros de configuración de ambos bloques son mostrados en la figura 4.29:

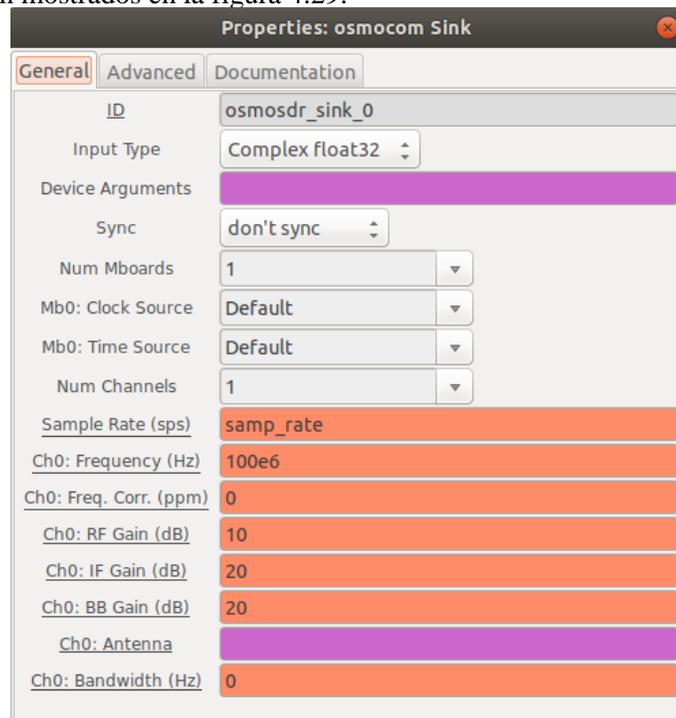
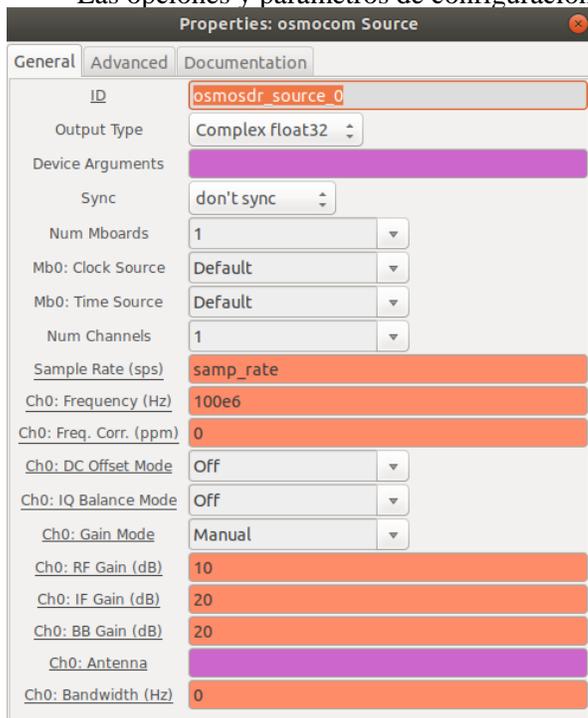


Figura 4.29 Parámetros de configuración de osmocom Source (izquierda) y osmocom Sink (derecha).



En la tabla 4.1. se realiza una descripción general de los parámetros de ambos bloques.

**Tabla 4.1. Descripción de los parámetros de configuración de los bloques Osmocom [4\_7].**

Nombre	Posibles valores	Descripción	
<b>Output Type</b>	Complex Float32	Este parámetro controla el tipo de datos del flujo en GNU Radio.	
<b>Device Arguments</b>	hackrf=0,buffers=32,bias=0,bias_tx=0	Es una cadena delimitada por comas que se utiliza para ubicar dispositivos conectados a la computadora. Si se deja en blanco se utilizará el primer dispositivo encontrado.	
	hackrf=<device-index>		0-based device identifier OR serial number
	bias=0 1		Disable or enable antenna bias voltage in receive mode (source)
	bias_tx=0 1		Disable or enable antenna bias voltage in transmit mode (sink)
	buffers=<number-of-buffers>	Default is 32	
<b>Num Channels:</b>		Selecciona el número total de canales cuando se configuran múltiples dispositivos.	
<b>Sample Rate</b>	[symbols per second]	La frecuencia de muestreo es el número de muestras por segundo que genera el bloque en cada canal.	
<b>Frequency</b>	[Hz]	La frecuencia central que se sintoniza en RF.	
<b>Freq. Corr .</b>	[ppm]	El factor de corrección de frecuencia en partes por millón (ppm). Se establece en 0 si se desconoce.	
<b>DC Offset Mode</b>	Off/Manual/Automatic	Controla el comportamiento de la corrección de offset de DC del hardware. Sólo para dispositivos USRP.	
<b>IQ Balance Mode</b>	Off/Manual/Automatic	Controla el comportamiento de la corrección de desequilibrio IQ del software.	
<b>Gain Mode</b>	Off/Manual/Automatic	Para permitir el control manual de las etapas de ganancia RF / IF / BB, se debe configurar el modo de ganancia manual.	
<b>RF Gain</b>	[dB]	Ganancia total de RF del dispositivo.	
<b>IF Gain</b>	[dB]	Ganancia global de frecuencia intermedia del dispositivo. Disponible para HackRF en modo de recepción y transmisión.	
<b>BB Gain</b>	[dB]	Ganancia general de banda base del dispositivo. Esta configuración está disponible para HackRF en modo de recepción.	
<b>Antenna</b>		Para dispositivos con una sola antena, esto se puede dejar en blanco. De lo contrario, especificar una de las posibles opciones de antena.	
<b>Bandwidth</b>	[Hz]	Configura el filtro paso banda del front end. Para setear el valor por default del filtro este campo deberá ser 0.	

### 4.2.2.1. Prueba de recepción en FM

Para validar el correcto funcionamiento de la integración GNU Radio con el HackRF One se realiza un módulo de recepción de radio FM. El objetivo es recibir una señal de radio comercial en FM utilizando el HackRF, la señal recibida será llevada a GNU Radio con el bloque Osmocom Source y posteriormente es procesada y convertida en audio por GNU Radio.

Para implementar este sistema se elabora el siguiente diagrama de bloques:

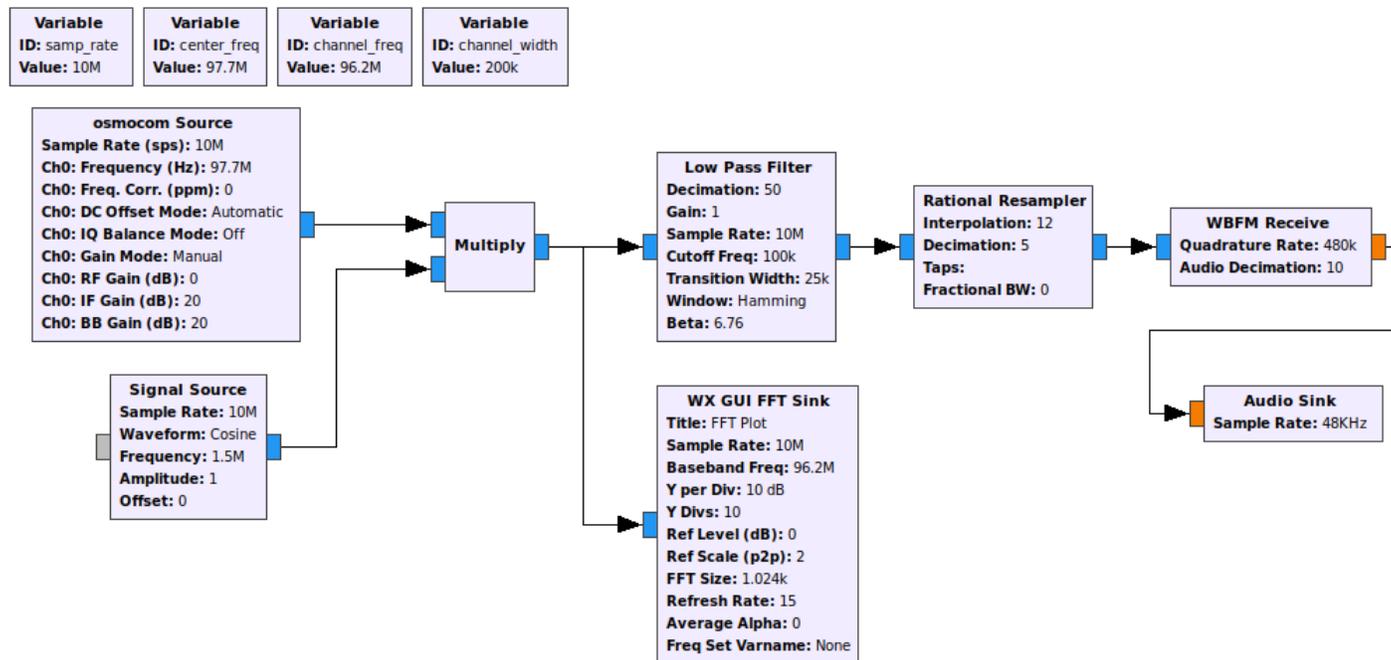


Figura 4.30. Receptor FM implementado en GNU Radio.

En este ejemplo la frecuencia central configurada es de 97.7 MHz la cual corresponde a la estación Grupo Radiocentro. La tasa de muestreo se configura con un valor de 10 MHz y el ancho de canal se establece en 200 kHz.

El sistema receptor inicia en el bloque osmocom Source el cual recibe las señales provenientes del HackRF One en el rango de los 0 a los 6000 MHz. Su configuración es mostrada en la figura 4.31. La señal recibida es multiplicada por una señal coseno de 1.5MHz con el objetivo de demodularla. Enseguida es llevada a un filtro paso bajas con frecuencia de corte de 100 kHz.

Posteriormente, la señal filtrada entra al bloque Rational Resampler donde reduce la razón de muestreo. El bloque WBFM (WideBand Frequency Modulation) recupera la señal de audio enviada a partir de la envolvente compleja de la señal FM recibida. En este bloque también se reduce la tasa de muestreo. Finalmente, el bloque Audio Sink envía la señal de audio a la tarjeta de sonido de la PC.

La configuración del bloque OsmoCom Source se muestra en la figura 4.31.

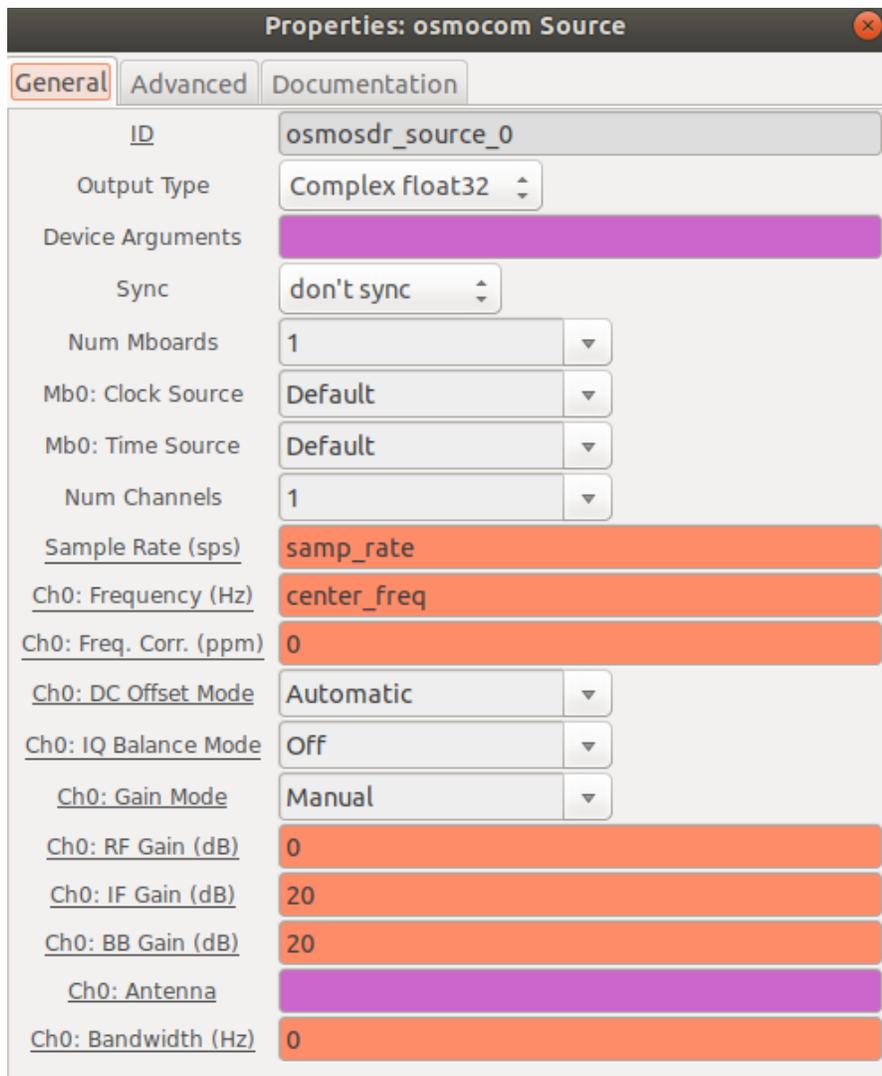


Figura 4.31. Parámetros configurados de osmocom Source para el receptor FM.

Con apoyo del analizador de espectros (FFT Sink) de GNU Radio se puede visualizar la señal FM recibida antes de ser filtrada como se muestra en la figura 4.32.

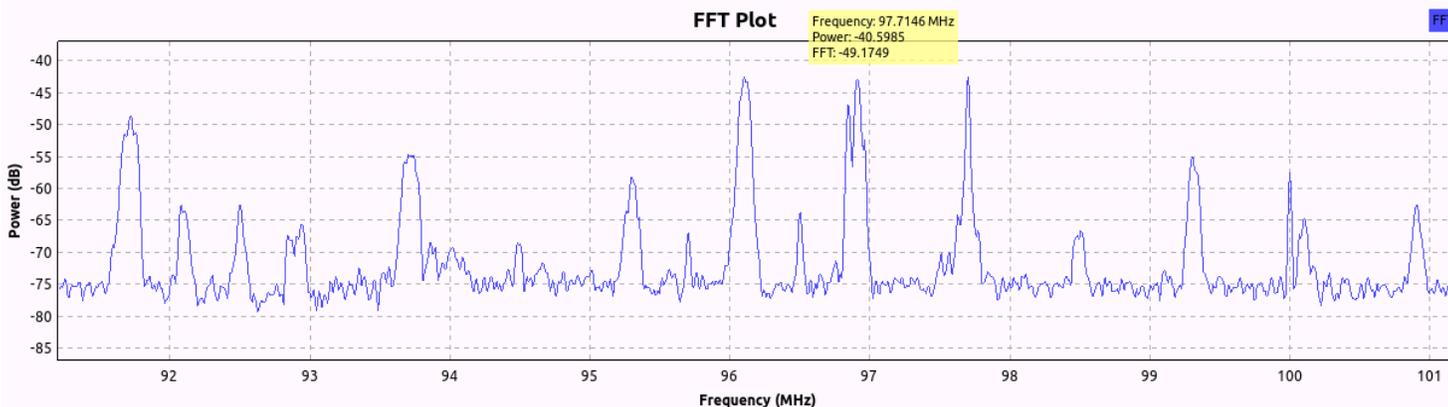


Figura 4.32. Espectro de la señal FM recibida en GNU Radio.



En esta, se observa que la frecuencia central esperada tiene una potencia por encima de los -45 dB. Como se mencionó, HackRF entrega un amplio rango de frecuencias por lo cual es posible visualizar más de una estación de radio previo a la etapa de filtrado.

Finalmente, la estación de radio sintonizada es escuchada a través del sistema de audio de la computadora. En la siguiente figura se muestra la conexión de la computadora donde corre GNU Radio y el HackRF One. De igual modo, en las figuras 4.33 y 4.34 se muestran los led del HackRF donde el indicador amarillo de RX está activado durante el proceso de recepción. Al detenerse el programa este led se desactiva.

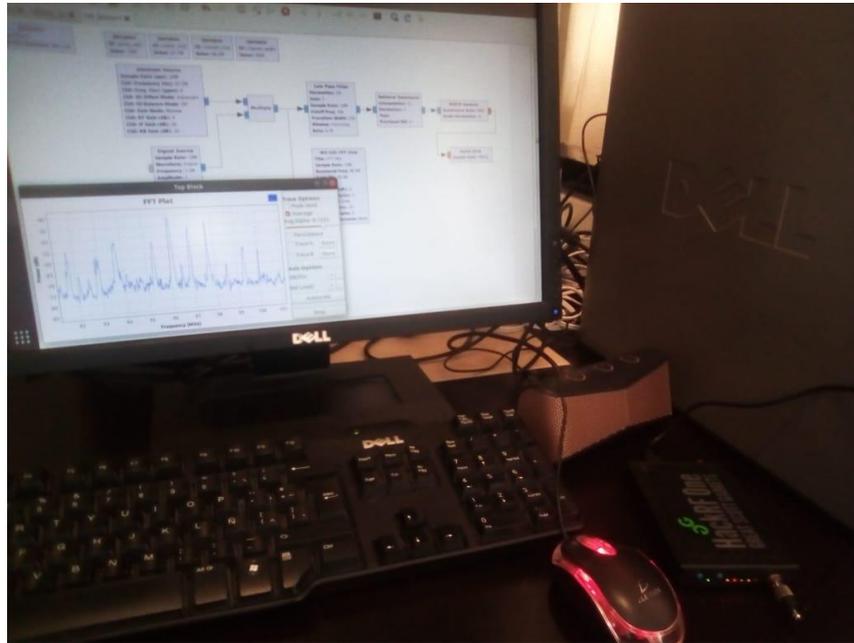


Figura 4.33. Integración de GNU Radio con HackRF One implementando un receptor FM.



Figura 4.34. HackRF One en modo de recepción.

De este modo, se comprueba el funcionamiento del Front End Analógico (HackRF) con el bloque de procesamiento de la señal (GNU Radio) lo cual se puede entender como una validación de un SDR integrado y funcionando de manera exitosa.

#### 4.2.3. Receptor Satelital

Para el bloque de recepción de la mesa de pruebas se ha optado por seleccionar un receptor satelital COTS del fabricante GTMedia (antes Freesat) modelo V9 mostrado en la figura 4.35.



**Figura 4.35. Receptor satelital GTMedia V9.**

Se ha seleccionado esta solución por ser un receptor sumamente accesible, alrededor de los 70 USD, y compatible con el estándar DVB-S2. A continuación se enlistan las características técnicas del mismo:

Estándar:	DVB-S2
Frecuencia de recepción:	950 MHz – 2150 MHz
Impedancia:	75 Ohm
Conector antena:	Coaxial macho Tipo F
Nivel Señal Entrada RF:	-80 / -20 dBm
Alimentación LNB Power:	13 / 18v DC a 400mA max
Demodulación:	QPSK, 8PSK, 16APSK, 32APSK
Formato video:	MKV, MP4, AVI, FLV, F4V, TS, M2TS, MOV, 3GP
Códec Video:	AC-3, MPEG-2, MPEG-4, H.265
Formato imagen:	JPEG, BMP, PNG, GIF

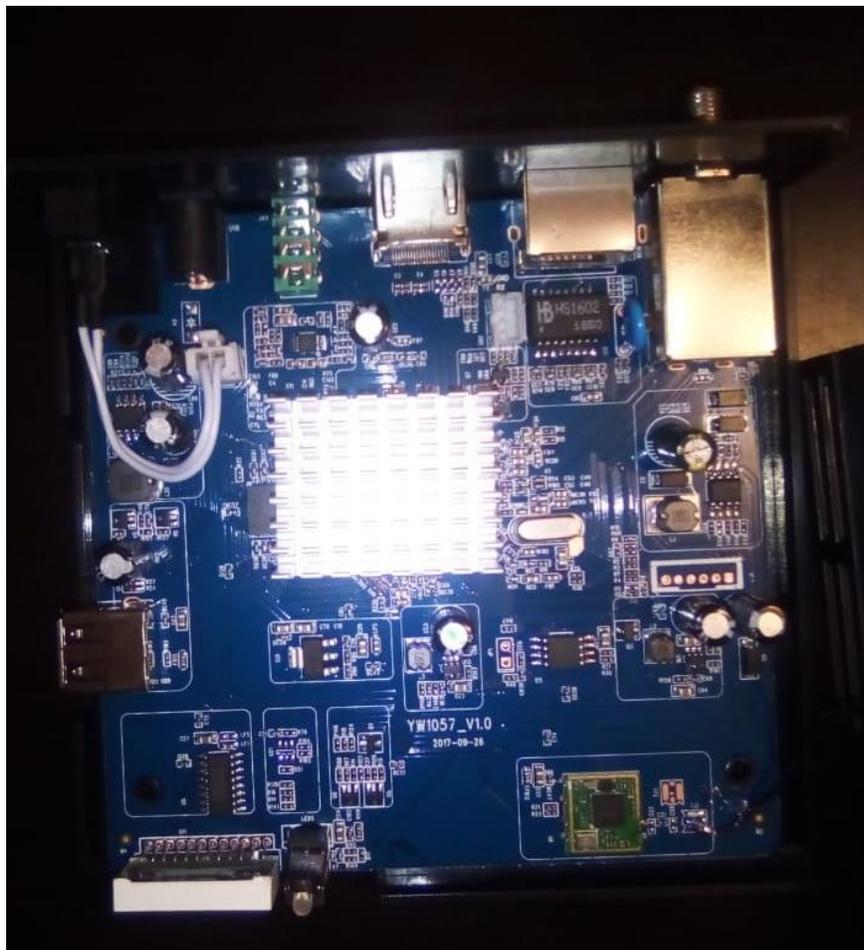


Figura 4.36. Layout del receptor satelital GTMedia V9.

#### 4.2.4. Antena receptora y transmisora

Se ha seleccionado una antena dipolo modelo G401-1, mostrada en la figura 4.37, con un rango de frecuencias de operación que va de los 824 MHz a los 2170 MHz para que conforme el segmento receptor junto con el GTMedia.



Figura 4.37. Antena dipolo G401-1 [4\_9].

Las características más importantes de la antena se enlistan en la Tabla 4.2 y su patrón de radiación se muestra en la figura 4.38.

Tabla 4.2. Características de la antena dipolo G401-1 [4\_9].

Modelo de Antena	Tipo	Frecuencia [MHz]	Ganancia [dBi]	Rango@ (250mW) [km]	Tipo de conector	Masa [g]	Dimensiones [mm]	Precio [USD]
G401-1	Dipolo	900 - 2200	3	45	RPSMA	13g	50	5.10

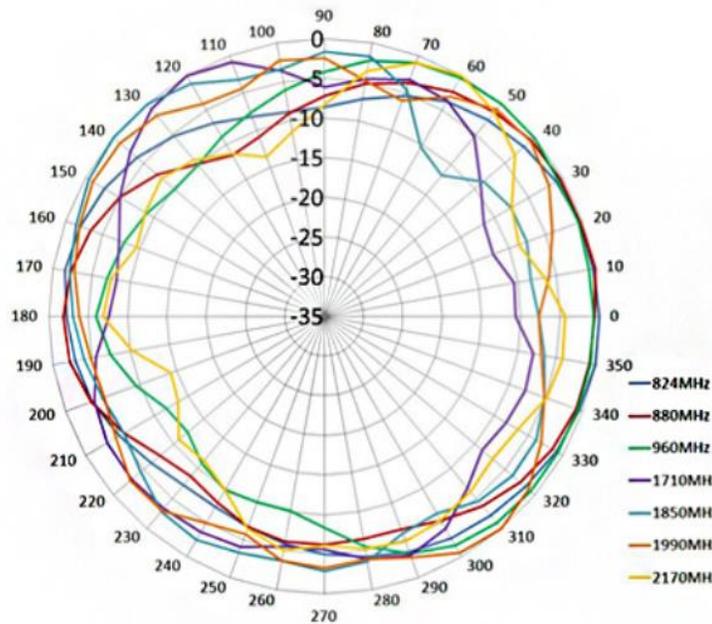


Figura 4.38 Patrón de radiación de la antena dipolo G401-1 [4\_9].

Para poder acoplar la antena dipolo (RPSMA) a la entrada del receptor (Conector F) se utiliza un adaptador coaxial RPSMA - F fabricado por la empresa OneLinkMore Tech como el mostrado en la figura 4.39.



Figura 4.39. Adaptador coaxial OneLinkMore Tech F-RPSMA utilizado en el receptor satelital.



### 4.3. PRUEBAS DE TRANSMISIÓN Y RECEPCIÓN

Una vez integrados los segmentos de transmisión y recepción de la mesa de pruebas se proceden a realizar pruebas de configuración y comunicación entre ambos segmentos. Una de las primeras pruebas realizadas es la transmisión y recepción de video con un ModCod de 16APSK 9/10. La configuración de transmisión y recepción realizada es la siguiente:

En GNU Radio se configura el transmisor DVB-S2 con los siguientes parámetros y conforme a la figura 4.40.

#### (Procesamiento de señal) GNU Radio

- ✓ Frecuencia de operación: 1.28GHz
- ✓ Ancho de banda: 10 MHz
- ✓ Tasa de símbolos: 5Msps
- ✓ Modulación: 16APSK
- ✓ Tasa de Codificación: 9/10
- ✓ Pilotos activados
- ✓ Transmitiendo archivo de video formato MPEG-2 Transport Stream

Para la configuración del HackRF se establece el bloque osocom sink con la siguiente configuración:

- ✓ Ganancia de IF (Front End Tx): 40 dB
- ✓ Ganancia de RF (Front End Tx): 40 dB
- ✓ Ganancia de BB (Front End Rx): -8 dBm

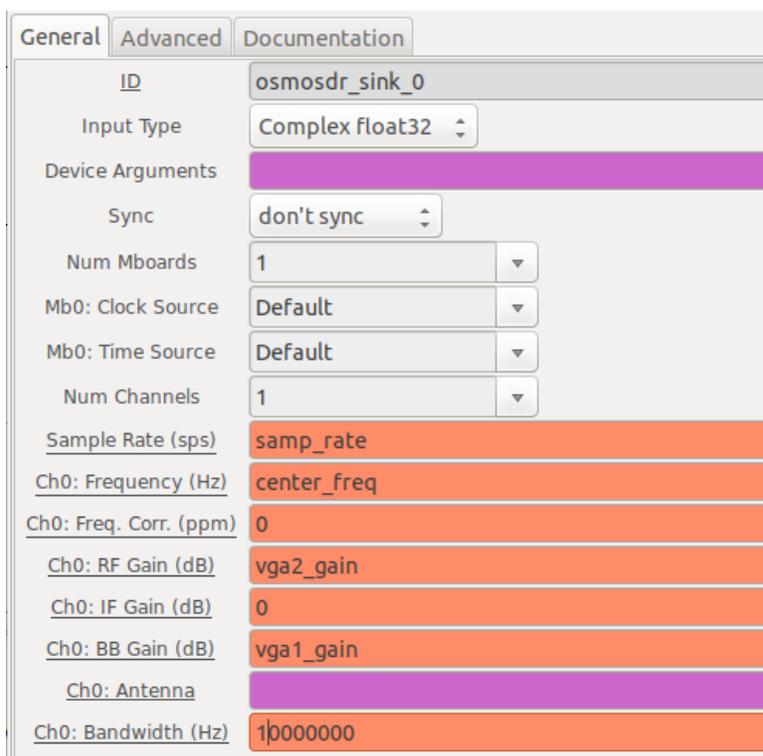


Figura 4.40 Configuración Osmocom Sink para la prueba de transmisión de DVB-S2 empleando HackRF.



Se conecta el HackRF One a la computadora y se activa GNU Radio. Se verifica que se genere y transmita la señal en RF desde el analizador de espectros y mapa de constelación de GNU así como verificando el led indicador de TX en el HackRF. Ver fig. 4.41, 4.42. y 4.43.

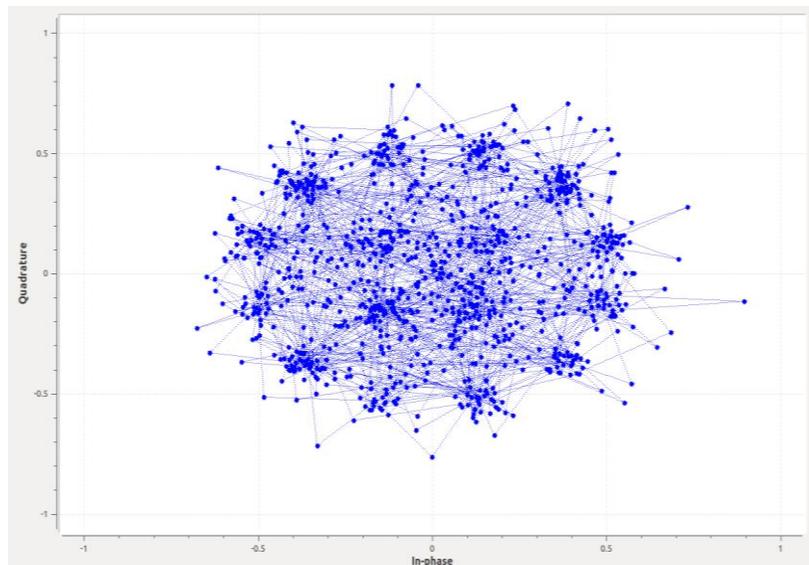


Figura 4.41 Mapa de constelación I-Q de la señal 16APSK generada en GNU Radio.

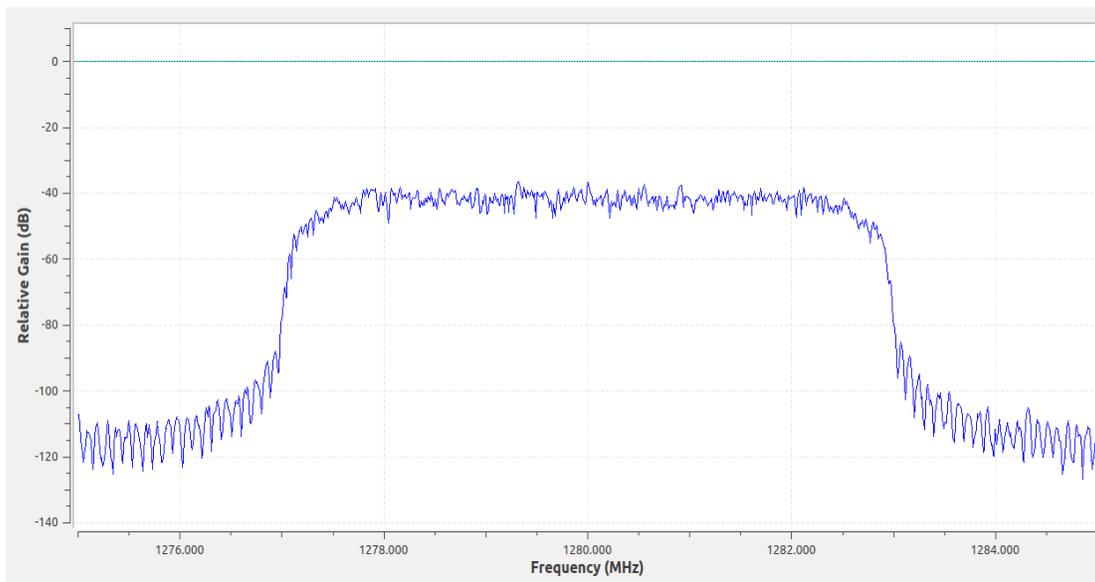


Figura 4.42 Espectro en frecuencia de la señal de 1.28 GHz generada por GNU Radio.



Figura 4.43 HackRF con indicador de TX activado.

Una vez que se comprueba la transmisión de la señal desde la mesa de pruebas se utiliza un analizador de espectros FSH4 de la marca Rohde&Schwarz como receptor al conectar una antena dipolo al mismo. De este modo, se verifica si realmente hay transmisión de una señal de radiofrecuencia por parte del HackRF independientemente de la decodificación y demodulación del receptor satelital y de los gráficos entregados por GNU Radio.

En la figura 4.44 se puede visualizar el espectro recibido por el analizador de espectros con el cual podemos comprobar que la señal, efectivamente, está siendo transmitida en una frecuencia central de 1.28 GHz a una potencia de -72.8 dBm, un valor mayor a la sensibilidad del receptor (-80 dBm).



Figura 4.44. Espectro de la señal DVB-S2 recibida a una frecuencia de 1.28 GHz.

Del lado del receptor satelital se emplea el adaptador tipo F-RPSMA previamente mencionado para poder conectar la antena dipolo.



Figura 4.45. Receptor satelital GTmedia V9 super con antena dipolo.

Para la configuración del receptor satelital se realizan los siguientes ajustes en la configuración de la antena (figura 4.46) donde el parámetro más importante es la selección del LNB en Unicable 10200 configurado en una frecuencia central de 1280 MHZ y se inicia una búsqueda a ciegas mostrada en la figura 4.47.

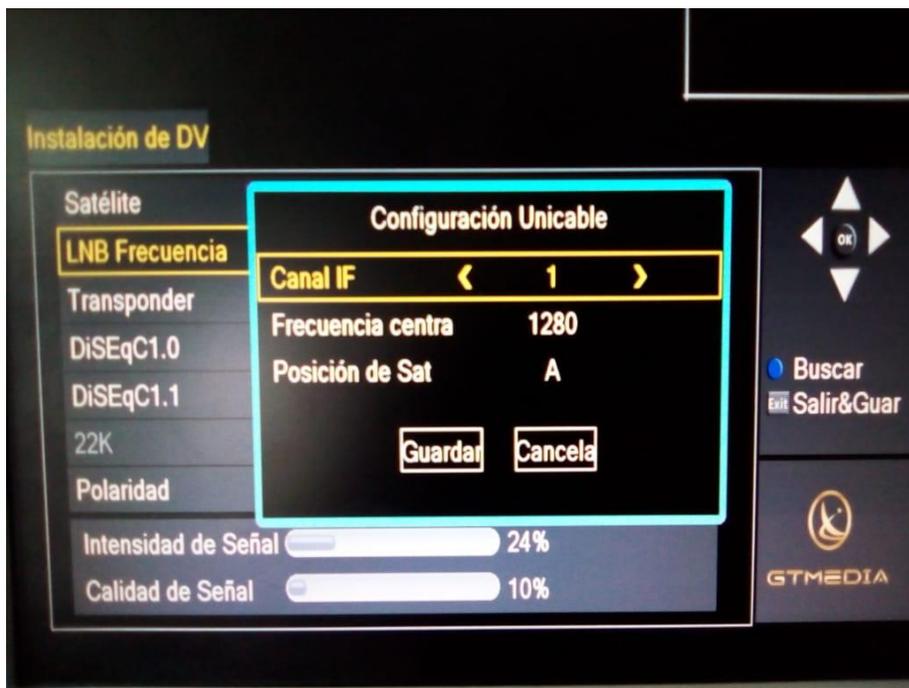


Figura 4.46 Configuración en receptor satelital previo a la prueba DVB-S2.



Figura 4.47 Búsqueda de la señal transmitida en el receptor satelital.

El resultado de la búsqueda usualmente arroja dos canales: uno en polarización horizontal y otro en vertical. Ver fig. 4.48.



Figura 4.48 Canales encontrados por el receptor satelital.

Al encontrarse los canales en el receptor se asocia un valor de tranceiver el cual es asumido por el receptor como la fuente transmisora del canal detectado. De igual modo, se detecta la señal con una intensidad de 56% y una calidad de señal de 90% como se observa en la figura 4.49.



Figura 4.49. Intensidad y calidad de la señal detectada por el receptor satelital.

Posteriormente, es posible ver y escuchar en el receptor satelital el archivo de video precargado en GNURadio y transmitido por la etapa de RF del HackRF como se muestra en la figura 4.50.

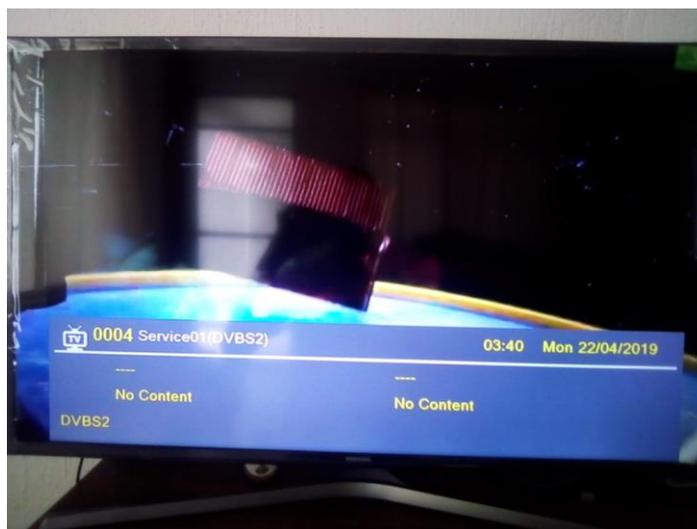
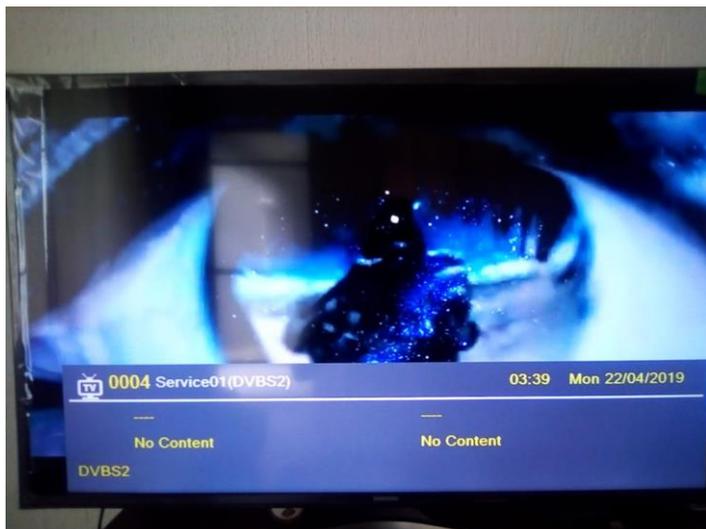
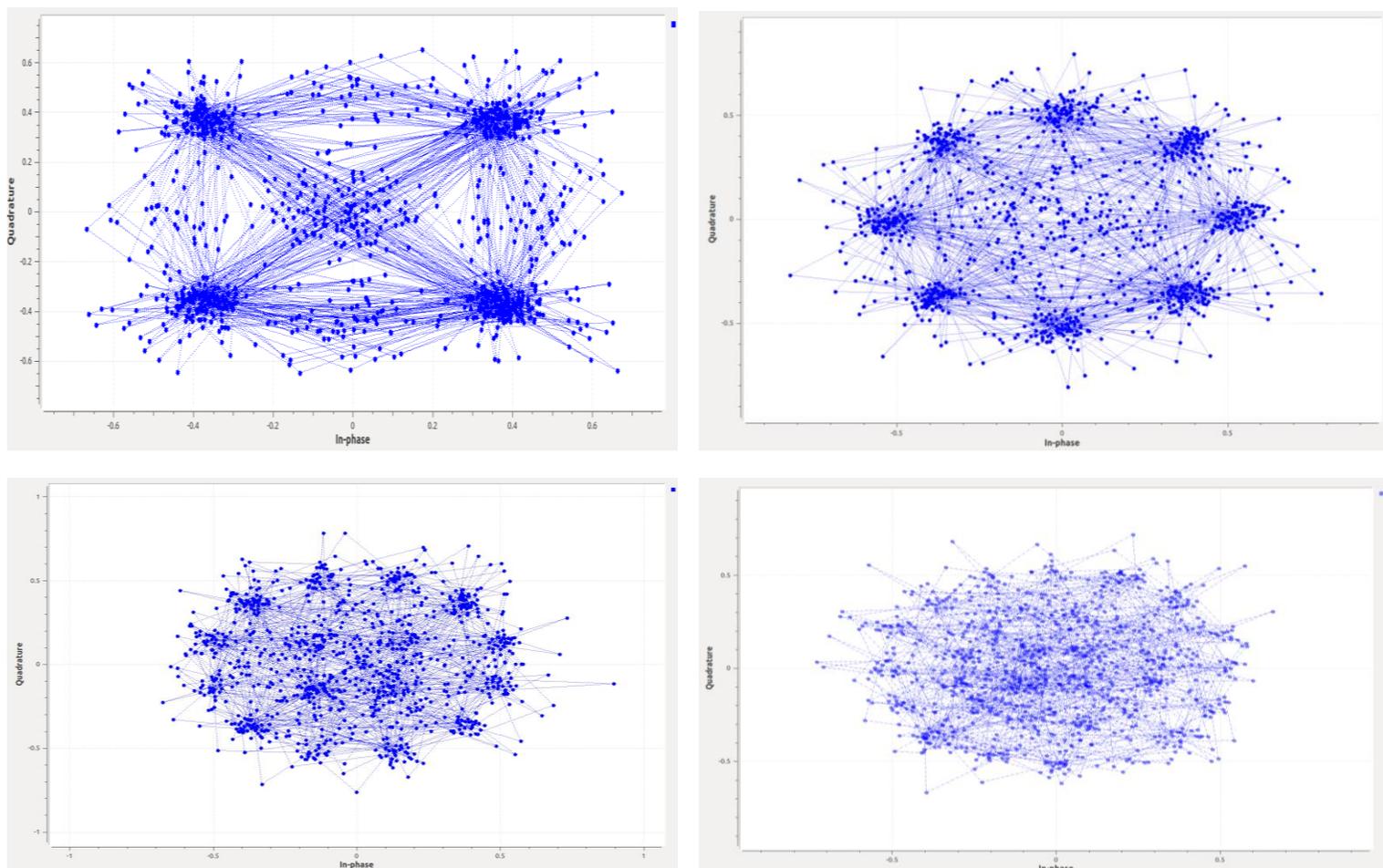


Figura 4.50. Archivo de video transmitido desde la mesa de pruebas y visualizado en el receptor satelital.

De este modo, se logra transmisión y recepción en modo CCM para diferentes modulaciones QPSK, 8PSK, 16APSK y 32APSK. Las constelaciones generadas en GNU Radio de las cuatro modulaciones son mostradas en la figura 4.51.



**Figura 4.51. Constelaciones generadas desde GNU Radio para las modulaciones QPSK, 8PSK, 16APSK y 32APSK.**

Es importante mencionar que para lograr la recepción de la señal DVB-S2 fue necesario llevar a cabo las siguientes acciones:

- Pruebas con configuración de frecuencias centrales en GNU Radio desde los 900 MHz hasta los 2200 MHz en saltos de 100 MHz.
- Configuración de las distintas frecuencias de LNBS disponibles en el menú de configuración del RX GTMedia.
- Las potencias de transmisión en GNU Radio, tanto de RF como de banda base, son establecidas en 0 dB y aumentada en saltos de 5 dB hasta llegar a los 40 dB con el objetivo de alcanzar una potencia en la recepción mayor a los -80 dBm. Pruebas realizadas a una distancia aproximada de 4 metros.
- Se emplean adaptadores coaxiales tipo F – RPSMA Hembra de la marca OneLinkMore. Con esto se busca descartar pérdidas por desacoplamiento.
- Empleo de dos antenas dipolo G401-1 tanto en transmisión como recepción para maximizar la ganancia.
- Uso de un analizador de espectros para visualizar de manera directa la señal generada por GNU Radio.



## 4.4 DESEMPEÑO DEL SDR IMPLEMENTADO

Una vez realizada una primera transmisión y recepción exitosa con la mesa de pruebas propuesta, se ha comprobado que el funcionamiento de un SDR transmisor implementando el estándar DVB-S2 es funcional y posible. Posteriormente, se realizan pruebas y mediciones de potencia de transmisión y recepción para obtener una caracterización del desempeño y funcionamiento de la mesa de pruebas.

A continuación se describen las configuraciones y pruebas realizadas al sistema.

### 4.4.1. Configuración Final Establecida

Después de verificar el funcionamiento y desempeño de diversas y numerosas configuraciones variando parámetros tales como frecuencia de operación, potencias de transmisión, ancho de banda, pilotos activados y desactivados, antenas, canales de recepción, entre otros, se llega al planteamiento de la siguiente configuración para obtener una recepción correcta de los diferentes ModCod establecidos:

#### GNU Radio (Procesamiento de señal)

- ✓ Frecuencia de operación: 1.28GHz
- ✓ Ancho de banda: 10 MHz
- ✓ Tasa de símbolos: 5Msps
- ✓ Modulación: QPSK/8PSK/16APSK/32APSK
- ✓ Tasa de Codificación: 1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 4/5, 5/6, 8/9, 9/10
- ✓ Pilotos activados
- ✓ Transmitiendo archivo de video formato MPEG-2 Transport Stream

#### HackRF OsmoCom (Transmisor)

- ✓ Ganancia de IF (Front End Tx): variable entre 10 y 45 dB dependiendo del ModCod
- ✓ Ganancia de RF (Front End Tx): 10 dB (Aumentar la ganancia de RF arriba de 10 dB no impacta en la potencia de transmisión)
- ✓ Ganancia de BB (Front End Rx): -8 dBm (este valor sólo impacta en modo recepción)

#### GTMedia (Receptor Satelital)

- ✓ LNB: Unicable 10200 (IF 1280 MHz)
- ✓ DiseqC: Desactivados
- ✓ Polarización: Automático
- ✓ Tipo de búsqueda: Búsqueda a ciegas

Como se muestra, la frecuencia de operación seleccionada es de 1.28 GHz la cual es una frecuencia disponible para radio amateur. Se intentaron pruebas en otras frecuencias de operación con resultados no exitosos al momento de recibir la señal. En este sentido el empleo de un convertidor de frecuencia de 2.2GHz a 1.28GHz será necesario para la configuración final de la Estación Terrena si es que se desea realizar con el receptor satelital propuesto.

### 4.4.2. Estimación del SNR experimental

Para realizar la estimación del SNR experimental del sistema se emplea el analizador de espectros mencionado anteriormente, el FSH4 de la marca Rohde&Schwarz. Con este, se implementa la mesa de pruebas observada en la figura 4.52 que consta del segmento transmisor, el segmento receptor y el analizador de espectros.



Figura 4.52 Configuración de la mesa de pruebas para la estimación del SNR.

Primeramente, se hace una aproximación del valor del ruido detectado por el analizador de espectros para la frecuencia de 1.28 GHz. En esta primera medición el transmisor se encuentra desactivado. Como se aprecia en la figura 4.53, el valor obtenido está entre los -94.4 y -93.2 dBm. Es importante considerar que todos los resultados son obtenidos con una distancia de 1 metro en el transmisor y el receptor.

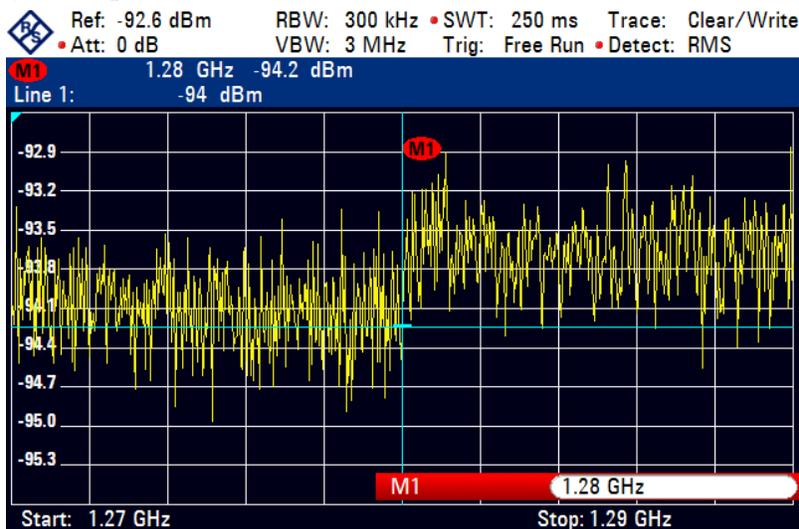


Figura 4.53 Potencia de ruido medido en el analizador de espectros.



Para la determinación del valor de SNR se considera entonces una potencia de ruido de -93 dBm.

En cuanto a la potencia de la señal recibida se consideran dos valores:

**a) Potencia mínima de recepción:**

Es la potencia mínima en dBm detectada por el analizador de espectros a la cual el receptor satelital es capaz de detectar la señal audiovisual proveniente del segmento transmisor. Al ser la potencia mínima se esperan diversos errores en la recepción tales como imagen congelada, pixeleada o incompleta, audio entrecortado, entre otros. Este valor sólo sirve como referencia.

**b) Potencia de recepción sin errores:**

Es la potencia en dBm que se obtiene con base en el valor de la potencia recibida en el analizador de espectros a la cual el receptor satelital es capaz de detectar y mostrar la señal audiovisual proveniente del segmento transmisor sin ningún error. Esto es, el video transmitido es mostrado en su totalidad sin errores, cortes o pausas de audio o video.

Si bien esta estimación es subjetiva, se basa en el hecho de que el valor de SNR recomendado por la ETSI tiene como objetivo alcanzar un valor de PER de  $10^{-7}$ , el cual a su vez garantiza una transmisión con menos de un paquete erróneo durante una hora de transmisión. Es decir, si durante la recepción de la señal DVB-S2 no existe ningún error, se tiene cierta certeza de que el valor del PER es mínimo. Este valor será el utilizado para obtener el SNR experimental.

El video transmitido tiene una duración de 4 minutos y 41 segundos. Tiene un formato MPEG-2 TS con un códec de video H.264 y de audio MPEG-4 AAC. Su tamaño va de los 84 MBytes a los 770 MBytes para los ModCod QPSK  $\frac{1}{4}$  y 32APSK  $\frac{9}{10}$  respectivamente. Para cada ModCod se convierte un video base de formato MP4 a un formato .TS con ayuda del software libre **FFMPEG** el cual es un convertidor de video en diferentes formatos. Para la conversión se considera la tasa de multiplexación del formato MPEG-2 la cual a su vez considera las tasas de codificación y muestreo de la señal a ser transmitida. Es decir, se requiere generar un video específico para cada valor de ModCod y tasa de muestreo dados. En este caso la tasa de muestreo es de 5 Msps para poder obtener el valor de ancho de banda de referencia igual a 10 MHz.

El valor de ambas potencias y el SNR experimental para los 28 ModCod disponibles se muestran en la tabla 4.3.



Tabla 4.3. SNR experimental y potencias de recepción de la mesa de pruebas.

Mode	ModCod	Es/No [dB] ETSI	Throughput Máximo Teórico [Mbps] BW=10MHz	Potencia de RX (mínima)		Potencia de RX (sin errores)		SNR experimental [dB]	SNR ETSI [dB]
				Ganancia FI HackRF [dB]	Analizador Espectros [dBm]	Ganancia FI HackRF [dB]	Analizador Espectros [dBm]		
1	QPSK 1/4	-2.35	3.9266	14	-90.1	16	-88.6	4.4	-2.1711
2	QPSK 1/3	-1.24	5.2511	14	-90.1	16	-88.6	4.4	-1.1456
3	QPSK 2/5	-0.30	6.3105	14	-90.1	16	-88.6	4.4	-0.2772
4	QPSK 1/2	1	7.9002	15	-89.3	16	-88.6	4.4	0.9239
5	QPSK 3/5	2.23	9.4896	16	-88.6	16	-88.6	4.4	2.0603
6	QPSK 2/3	3.10	10.5572	16	-88.6	16	-88.6	4.4	2.8641
7	QPSK 3/4	4.03	11.8983	17	-86.9	17	-86.9	6.1	3.7233
8	QPSK 4/5	4.68	12.6686	17	-86.9	17	-86.9	6.1	4.3238
9	QPSK 5/6	5.18	13.2063	17	-86.9	18	-85.6	7.4	4.7857
10	8PSK 3/5	5.50	14.2344	23	-83.9	27	-80.9	12.1	5.0814
11	QPSK 8/9	6.20	14.0971	18	-85.6	19	-84.8	8.2	5.7281
12	QPSK 9/10	6.42	14.2736	19	-84.8	19	-84.8	8.2	5.9314
13	8PSK 2/3	6.62	15.8358	25	-81.4	31	-78.9	14.1	6.1161
14	8PSK 3/4	7.91	17.8474	27	-80.9	33	-77.7	15.3	7.3080
15	16APSK 2/3	8.97	21.1143	32	-75.3	34	-74.7	18.3	8.2873
16	8PSK 5/6	9.35	19.8094	31	-78.9	33	-77.7	15.3	8.6384
17	16APSK 3/4	10.21	23.7966	34	-74.7	37	-69.9	23.1	9.4329
18	8PSK 8/9	10.69	21.1457	33	-77.7	36	-73.2	19.8	9.8764
19	8PSK 9/10	10.98	21.4105	35	-75.5	37	-68.6	24.4	10.1443
20	16APSK 4/5	11.03	25.3372	36	-70.1	37	-69.9	23.1	10.1905
21	16APSK 5/6	11.61	26.4125	36	-70.1	38	-68.3	24.7	10.7263
22	32APSK 3/4	12.73	29.7457	37	-71.5	40	-69.7	23.3	11.7611
23	16APSK 8/9	12.89	28.1943	38	-68.3	40	-66.2	26.8	11.9089
24	16APSK 9/10	13.13	28.5473	39	-67.2	41	-65.7	27.3	12.1307
25	32APSK 4/5	13.64	31.6715	40	-67.8	41	-67.0	26	12.6018
26	32APSK 5/6	14.28	33.0517	40	-67.8	41	-67.0	26	13.1931
27	32APSK 8/9	15.69	35.2429	41	-66.8	42	-65.7	27.3	14.4958
28	32APSK 9/10	16.05	35.6841	41	-66.8	43	-65.3	27.7	14.8284

En la figura 4.54 se muestran las potencias de recepción medidas para la mesa de pruebas, tanto la mínima como la que no genera errores. Ambas potencias, lo que en realidad caracterizan es el desempeño del receptor, más allá de las características propias del segmento transmisor. En este caso se observa que la diferencia entre la potencia mínima y la potencia sin errores se encuentra entre los 2dBm para las tasas de codificación por debajo de los 3/5 y hasta 5dBm para las



tasas de codificación arriba de este valor. En general, ambas gráficas presentan un comportamiento muy similar al incrementar conforme el orden del ModCod también lo hace.

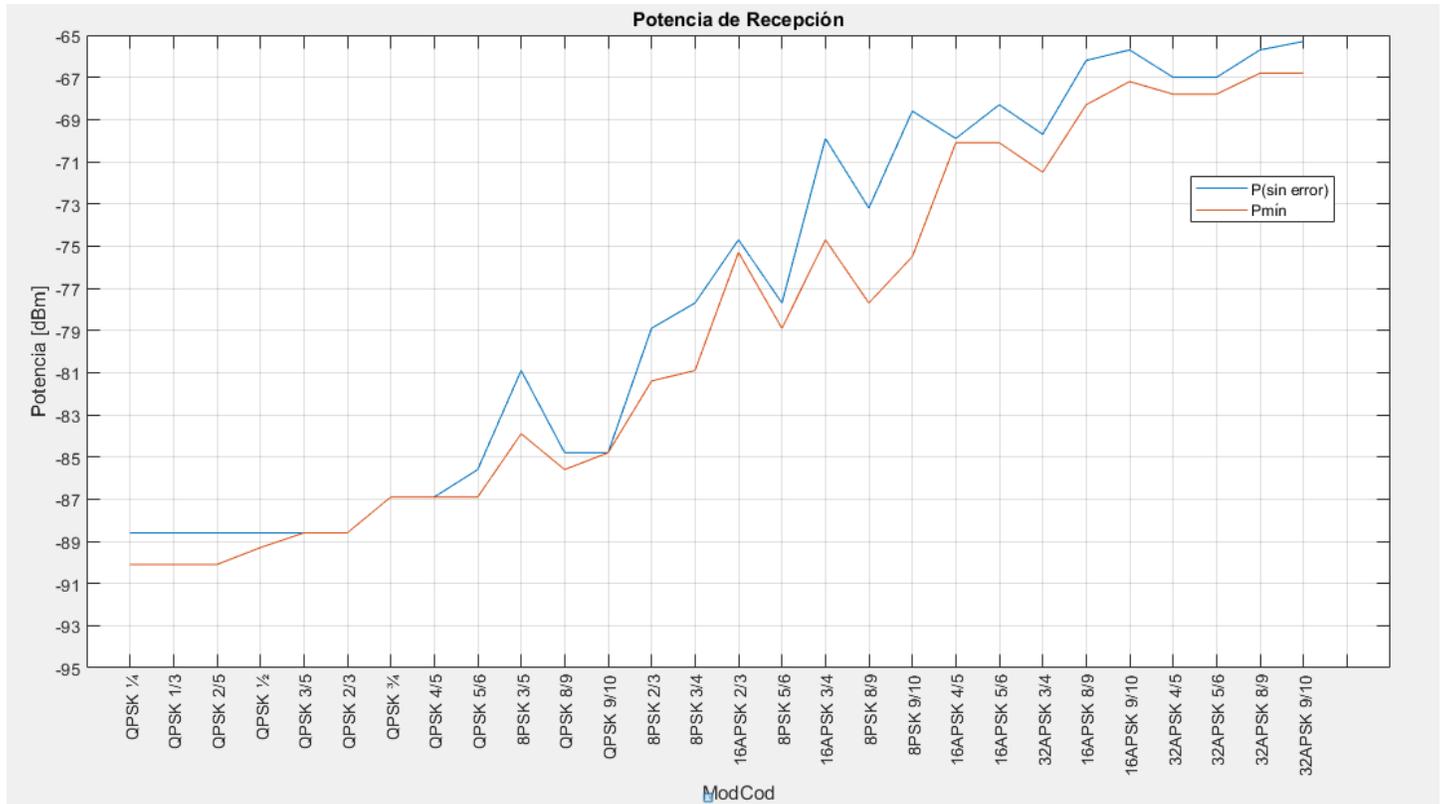


Figura 4.54. Gráfica de la potencia mínima vs potencia de recepción sin errores para la mesa de pruebas.

Por otro lado, el cálculo del SNR experimental se establece del siguiente modo:

$$SNR_{Experimental} = Potencia_{RX Sin Errores} - Potencia de Ruido$$

De los resultados mostrados en la tabla 4.3 se puede apreciar que los valores de SNR experimental distan de los valores recomendados por la ETSI con una diferencia que va de los 1.53 dB para QPSK 2/3 hasta los 15.16dB para 16APSK 9/10. Ambos valores, tanto experimentales como los dados por la ETSI se muestran en la figura 4.55.

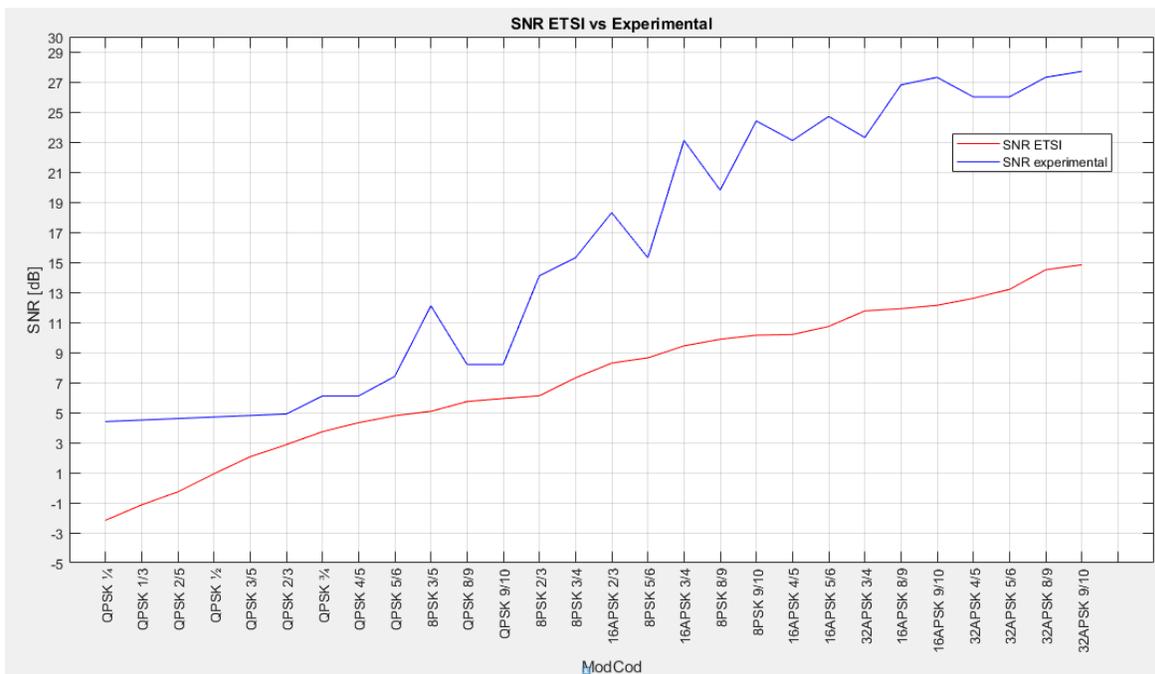


Figura 4.55. Gráfica de SNR ETSI vs SNR experimental obtenido de la mesa de pruebas.

La diferencia entre ambas gráficas es importante a tal grado que ninguno de los valores propuestos por la ETSI se acerca a los obtenidos de manera experimental. La diferencia puede ser originada por dos motivos: el método de estimación del SNR experimental es poco eficaz o el desempeño del receptor satelital no es el óptimo. En este caso se piensa que la segunda opción es la que más afecta el desempeño del sistema pues incluso tomando como referencia los valores de la potencia mínima de recepción los valores entre los SNR de ETSI y experimentales continúan siendo disímiles como se muestra en la figura 4.56.

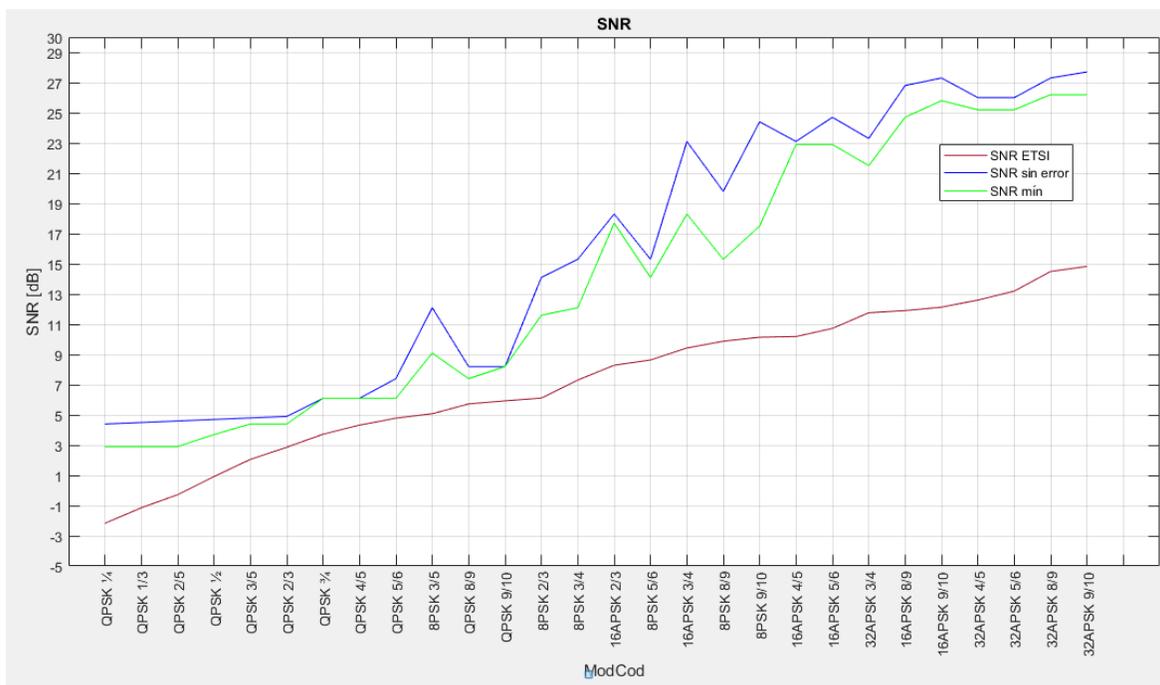


Figura 4.56. Gráfica comparativa entre los valores de SNR ETSI, SNR experimental mínimo y sin errores.



Para lograr apreciar la diferencia en el desempeño entre ambos SNR, tanto los dados por la ETSI como los experimentales, se realiza el análisis de tasa de datos y capacidad del canal realizado en el capítulo anterior. De este modo, se generan los resultados mostrados en la tabla 4.4.

**Tabla 4.4. Tasa de datos y capacidad del canal teóricos y experimentales.**

Esquema	ACM Rbps Teórico (SNR ETSI)					ACM Rbps Mesa Pruebas (SNR experimental)					Delta teo-exp
	Órbita/Rbps	Mín Mbps	Máx Mbps	Avg Mbps	Acumulado	Fotos/año	Mín Mbps	Máx Mbps	Avg Mbps	Acumulado	
<b>Órbita ISS 0.5W</b>	19.80	35.68	29.50	1945.5 GB	52,234	14.27	21.14	15.66	1032.8 GB	27,730	53.0%
<b>Órbita ISS 1W</b>	28.54	35.68	34.24	2257.9 GB	60,622	14.27	29.74	17.73	1169.5 GB	31,398	51.7%
<b>Órbita SSO 0.5W</b>	15.83	35.68	24.91	1919.4 GB	48,989	12.66	19.80	14.67	1130.6 GB	28,857	58.9%
<b>Órbita SSO 1W</b>	25.33	35.68	31.08	2394.7 GB	61,121	14.27	21.14	15.98	1231.4 GB	31,429	51.4%

Con esta información se puede apreciar, en términos generales, que las tasas de datos y la capacidad del canal decremantan prácticamente a la mitad para todos los casos. Esto se logra visualizar de mejor modo al comparar el número de fotos posibles a transmitir de manera anual para cada combinación de órbita y potencia de transmisión. En la columna Delta teo-exp se especifica la diferencia existente entre el valor teórico y el experimental para la capacidad del canal. Esta diferencia se encuentra entre el 51.4 y 58.9%, y puede entenderse como la eficiencia máxima del sistema en relación a la capacidad del canal.



### 4.4.3. Análisis del espectro generado

Posteriormente, se procede a analizar el espectro de las señales transmitidas con el objetivo de obtener más información relacionada con el desempeño del sistema implementado. Para ello se decide tomar como referencia las 4 modulaciones disponibles en el estándar con su mayor y menor tasa de codificación. Para cada uno de estos ModCod se suministra una potencia equivalente a la potencia de recepción sin errores indicada en la tabla 4.3 por lo que la potencia de la señal recibida va en incremento conforme aumenta el orden de modulación.

Para el análisis se revisa nuevamente la máscara del espectro de la señal indicado por la ETSI y mencionado en el subcapítulo anterior.

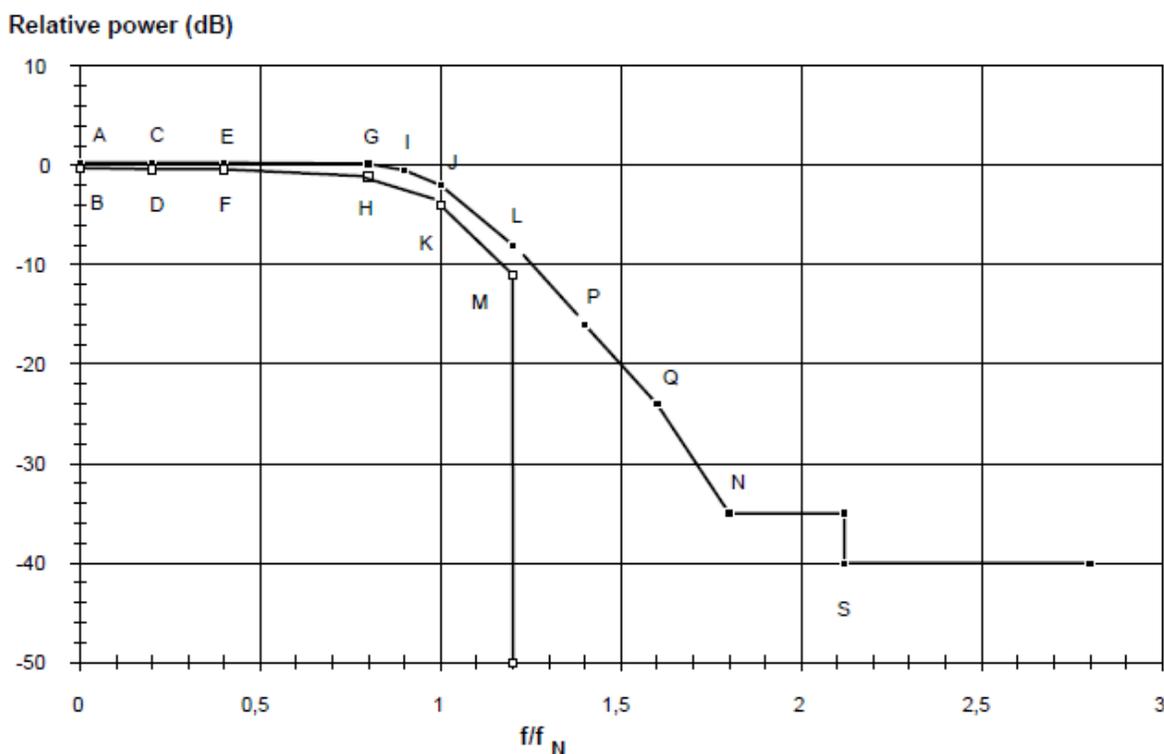


Figura 4.17. Máscara del espectro de la señal esperado a la salida del modulador DVB-S2 para un factor de roll off de 0.35 [4\_2].

Dado que el factor de roll off con el cual se trabaja es de 0.2 se toma como referencia los datos indicados en la figura 4.57 los cuales son tomados de [4\_2] y los cuales definen los valores de cada uno de los puntos que conforman la máscara de la figura 4.17 colocada aquí nuevamente por practicidad.



Point	Frequency for $\alpha = 0,35$	Frequency for $\alpha = 0,25$	Frequency for $\alpha = 0,20$	Relative power (dB)	Group delay
A	0,0 $f_N$	0,0 $f_N$	0,0 $f_N$	+0,25	+0,07 / $f_N$
B	0,0 $f_N$	0,0 $f_N$	0,0 $f_N$	-0,25	-0,07 / $f_N$
C	0,2 $f_N$	0,2 $f_N$	0,2 $f_N$	+0,25	+0,07 / $f_N$
D	0,2 $f_N$	0,2 $f_N$	0,2 $f_N$	-0,40	-0,07 / $f_N$
E	0,4 $f_N$	0,4 $f_N$	0,4 $f_N$	+0,25	+0,07 / $f_N$
F	0,4 $f_N$	0,4 $f_N$	0,4 $f_N$	-0,40	-0,07 / $f_N$
G	0,8 $f_N$	0,86 $f_N$	0,89 $f_N$	+0,15	+0,07 / $f_N$
H	0,8 $f_N$	0,86 $f_N$	0,89 $f_N$	-1,10	-0,07 / $f_N$
I	0,9 $f_N$	0,93 $f_N$	0,94 $f_N$	-0,50	+0,07 / $f_N$
J	1,0 $f_N$	1,0 $f_N$	1,0 $f_N$	-2,00	+0,07 / $f_N$
K	1,0 $f_N$	1,0 $f_N$	1,0 $f_N$	-4,00	-0,07 / $f_N$
L	1,2 $f_N$	1,13 $f_N$	1,11 $f_N$	-8,00	-
M	1,2 $f_N$	1,13 $f_N$	1,11 $f_N$	-11,00	-
N	1,8 $f_N$	1,60 $f_N$	1,5 $f_N$	-35,00	-
P	1,4 $f_N$	1,30 $f_N$	1,23 $f_N$	-16,00	-
Q	1,6 $f_N$	1,45 $f_N$	1,4 $f_N$	-24,00	-
S	2,12 $f_N$	1,83 $f_N$	1,7 $f_N$	-40,00	-

Figura 4.57. Definición de los valores de la máscara espectral del estándar DVB-S2 [4\_2].

De este modo, para un roll off de 0.2, una frecuencia de operación de 1.28 GHz y una tasa de símbolos de 5Mpsps (equivalente al valor de  $R_s=2f_N$ ) se tienen los siguientes datos:

$$f_N = R_s/2 = 5\text{Mpsps}/2 = 2.5\text{MHz} \quad f = 1.28\text{GHz}$$

Con esta información se completa la tabla 4.5. mostrada a continuación.

Tabla 4.5 Definición de los valores la máscara espectral para  $f_N=2.5\text{MHz}$  y  $f=1.28\text{GHz}$ .

Punto	Frecuencia para $\alpha=0.20$	Desplazamiento en frecuencia [MHz] = $\Delta f$	Frecuencia equivalente ( $f + \Delta f$ ) [GHz]	Potencia relativa (dB)
A	0.0 $f_N$	0.0	1.28	+0.25
B	0.0 $f_N$	0.0	1.28	-0.25
C	0.2 $f_N$	0.5	1.285	+0.25
D	0.2 $f_N$	0.5	1.285	-0.40
E	0.4 $f_N$	1	1.281	+0.25
F	0.4 $f_N$	1	1.281	-0.40
G	0.8 $f_N$	2	1.282	+0.15
H	0.8 $f_N$	2	1.282	-1.10
I	0.9 $f_N$	2.25	1.28225	-0.50
J	1.0 $f_N$	2.5	1.2825	-2
K	1.0 $f_N$	2.5	1.2825	-4
L	1.2 $f_N$	3	1.283	-8
M	1.2 $f_N$	3	1.283	-11
N	1.8 $f_N$	4.5	1.2845	-35
P	1.4 $f_N$	3.5	1.2835	-16
Q	1.6 $f_N$	4	1.284	-24
S	2.12 $f_N$	5.3	1.2853	-40



Los valores de los puntos B, H, K y M son resaltados en la tabla pues estos son los puntos de inflexión de la máscara espectral y son los puntos analizados con apoyo del analizador de espectros.

Para ello lo que se obtiene el espectro de las señales de los ModCod mencionados anteriormente. Esto es:

- QPSK 1/4 y 9/10
- 8PSK 3/5 y 9/10
- 16APSK 2/3 y 9/10
- y 32APSK 3/4 y 9/10

Los espectros generados, figuras 4.59, 4.60, 4.61 y 4.62, son tomados directamente del analizador de espectros. En este se configuraron diferentes marcadores para medir el nivel de potencia de la señal, cada uno de estos ubicados en los puntos de interés como se muestra en la tabla 4.6.

Tabla 4.6. Marcadores configurados en el analizador de espectros.

Marcador	Punto	Posición
M1	B	1.28 GHz
M2	H	1.282 GHz
M3	K	1.2825 GHz
M4	M	1.283 GHz
M5	-	1.272 GHz
M6	-	1.287 GHz

Los marcadores M5 y M6 tienen como objetivo medir el valor de las espurias generadas durante la transmisión de la señal. En primer término, en la figura 4.58 se muestra el analizador de espectros con el transmisor desactivado. Como se espera no se recibe ninguna señal.

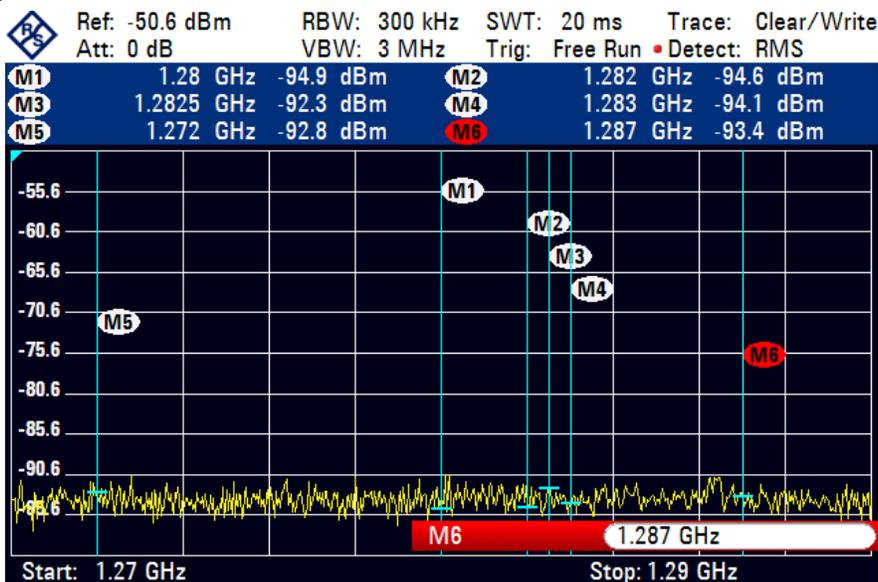


Figura 4.58. Espectro de la señal | Transmisión desactivada.

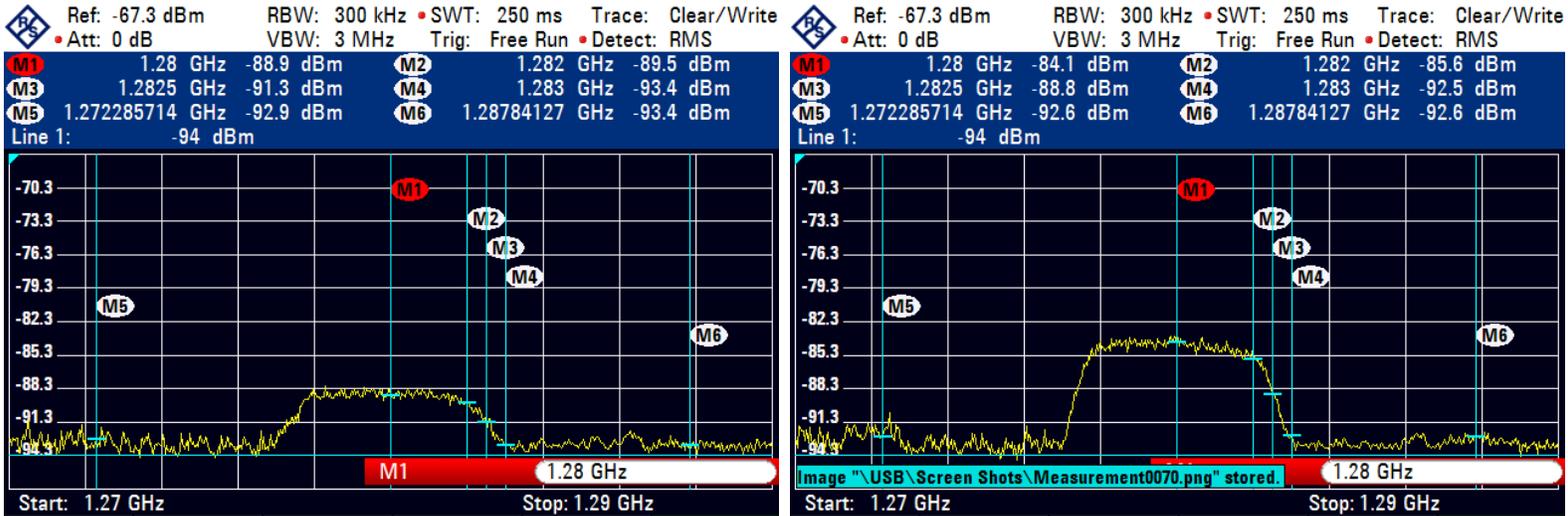


Figura 4.59. Espectro de la señal | QPSK 1/4 (izquierda) y QPSK 9/10 (derecha).

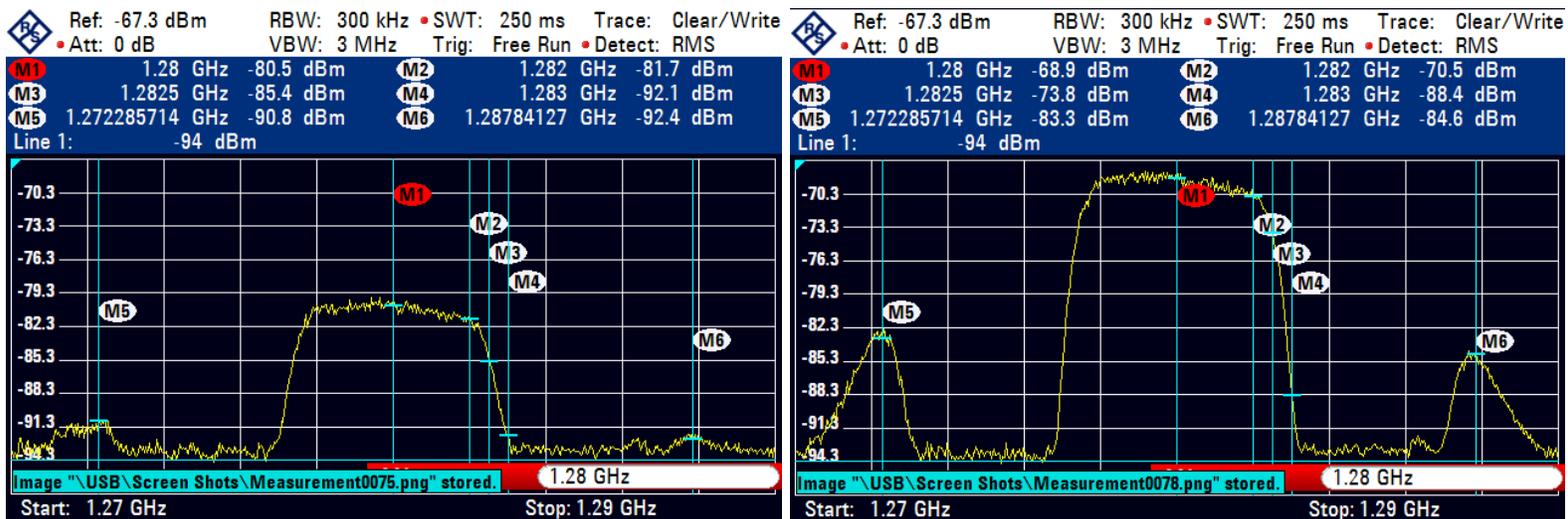


Figura 4.60. Espectro de la señal | 8PSK 3/5 (izquierda) y 8PSK 9/10 (derecha).

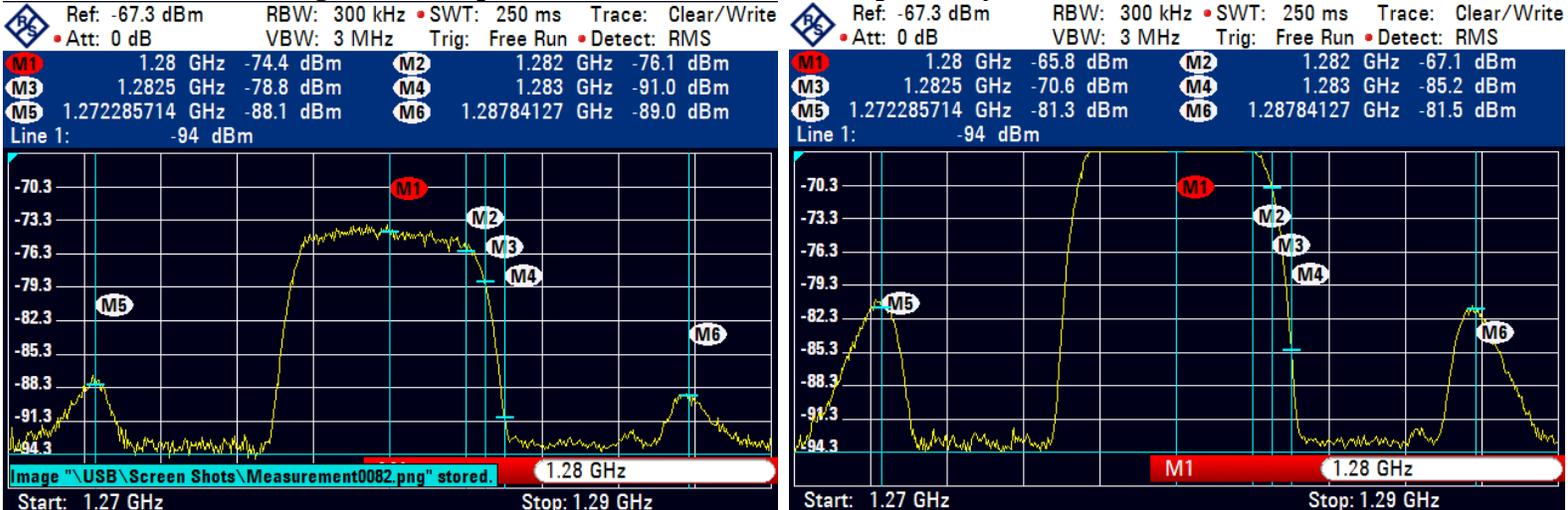


Figura 4.61. Espectro de la señal | 16APSK 2/3 (izquierda) y 16APSK 9/10 (derecha).

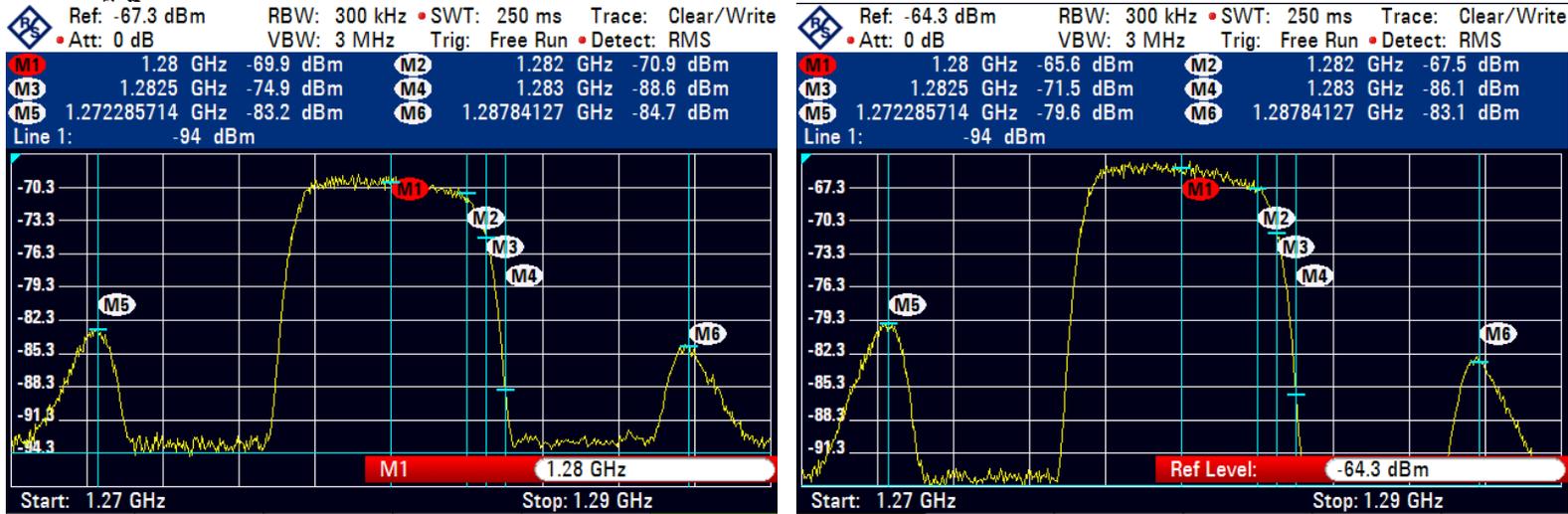


Figura 4.62. Espectro de la señal | 32APSK 3/4 (izquierda) y 32APSK 9/10 (derecha).

En la tabla 4.7 se muestran los datos recopilados de los diferentes espectros para los puntos de interés B, H, K y M. En general se observa que los valores establecidos por la ETSI respecto a los límites de la máscara espectral son cumplidos por excepción de las modulaciones en QPSK subrayadas en amarillo.

Esto demuestra que la señal generada por el segmento transmisor cumple con la normativa establecida por ETSI lo cual puede llegar a ser crucial al momento de trabajar en frecuencias fuera de la radio amateur como lo es la banda de los 2.2 GHz propuesta para el transmisor satelital.

Tabla 4.7. Valores de potencia relativa para los diferentes modcod.

Potencia Relativa (dB)											
Marcador	Punto	Posición [GHz]	Potencia relativa ETSI (dB)	QPSK	QPSK	8PSK	8PSK	16APSK	16APSK	32APSK	32APSK
				¼	9/10	3/5	9/10	2/3	9/10	3/4	9/10
M1	B	1.28	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
M2	H	1.282	-1.10	-0.85	-1.75	-1.45	-1.85	-1.95	-1.55	-1.25	-2.15
M3	K	1.2825	-4	-2.65	-4.95	-5.12	-5.15	-4.65	-5.05	-5.25	-6.15
M4	M	1.283	-11	-4.75	-8.65	-11.85	-19.75	-16.85	-19.65	-18.95	-20.75

Sin embargo, en la tabla 4.8 se muestran valores relacionados con señales espurias generadas por el segmento transmisor, específicamente en las frecuencias de 1.2722 GHz y los 1.2878 GHz, lo cual significa un valor de  $\pm 7.8$  MHz respecto a la frecuencia de operación.

Estas espurias son provocadas por la no linealidad del amplificador de frecuencia intermedia del front end analógico, es decir, la etapa de amplificación del HackRF. Se observa que las espurias tienen una potencia entre los -90.8 dBm, para 8PSK 3/5, y los -79.6 dBm para 32APSK 9/10. En general las espurias rebasan el nivel de ruido, por encima de los -91 dBm, a partir de que la ganancia del amplificador de Frecuencia Intermedia rebasa los 25dB. Considerando esto se puede afirmar que las modulaciones 8PSK, 16APSK y 32APSK presentan espurias, al ser implementadas en la mesa de pruebas, pues todas ellas requieren de un nivel de potencia mayor a los -81 dBm para ser detectadas correctamente en el receptor,



esto, sin considerar los valores obtenidos del SNR experimental lo cual requiere que la señal este muy por encima del nivel de ruido.

Tabla 4.8. Valores de potencia de las señales espurias.

Potencia absoluta (dBm)											
Marcador	Punto	Posición [GHz]	Potencia relativa ETSI (dB)	QPSK	QPSK	8PSK	8PSK	16APSK	16APSK	32APSK	32APSK
				¼	9/10	3/5	9/10	2/3	9/10	3/4	9/10
M5	--	1.2722	--	-92.9	-92.6	-90.8	-83.3	-88.1	-81.3	-83.2	-79.6
M6	--	1.2878	--	-93.4	-92.6	-92.4	-84.6	-89.0	-81.5	-84.7	-83.1

Es por esto que se recomienda considerar un espaciamento de  $\pm 11$  MHz respecto a la frecuencia de operación, ya sea para una señal adicional de transmisión o para cualquier señal en la recepción.

También es importante mencionar que el segmento transmisor es capaz de generar una señal de hasta -30 dBm con una configuración de ganancia para el amplificador de frecuencia intermedia igual a 45 dB, valor límite teórico. Este nivel de potencia es alcanzado para un modo QPSK 1/4. La imagen del espectro de la señal se muestra en la figura 4.63.

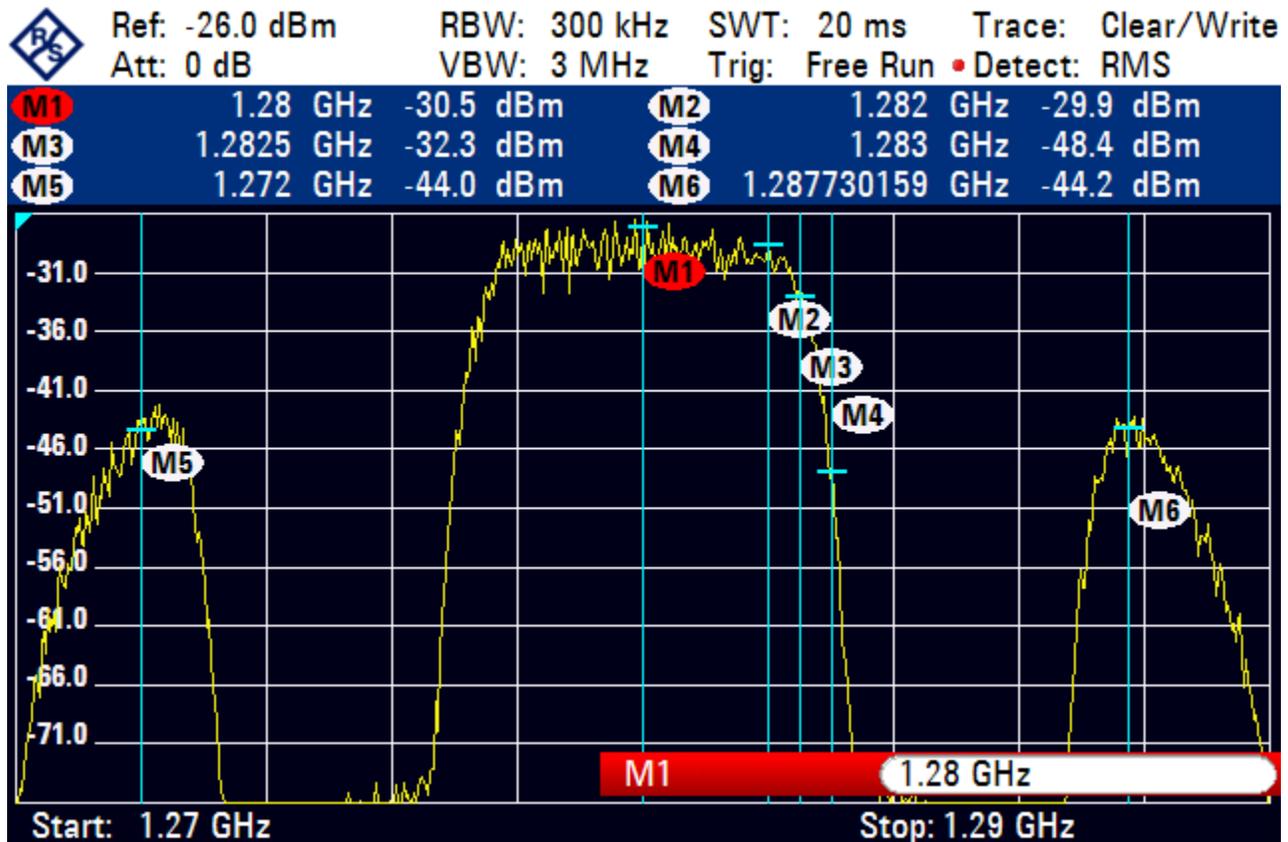


Figura 4.63. Espectro de la señal DVB-S2 | Modcod: QPSK ¼ | Ganancia FI=45 dB.



El valor de las señales espurias ronda los  $-44\text{dBm}$ , un valor de potencia importante que provoca un Rango Dinámico Libre de Espurios (SFDR) de  $14\text{ dBm}$  y que puede causar interferencia en el receptor del satélite en caso de encontrarse próximo a la frecuencia de operación del transmisor.

Finalmente, a manera de resumen y considerando los análisis anteriores se logran entonces puntualizar los siguientes datos:

- El valor de SNR experimental difiere con el valor teórico para todos los ModCod implementados con una diferencia mínima de  $1.53\text{ dB}$  para QPSK 2/3 hasta los  $15.16\text{dB}$  para 16APSK 9/10.
- La eficiencia máxima del sistema transmisor-receptor implementado puede ser de  $58.9\%$  respecto al valor teórico de la capacidad del canal.
- La presencia de señales espurias es originada por el efecto del amplificador de RF del segmento transmisor. Estas señales se encuentran a  $\pm 7.8\text{MHz}$  respecto a la frecuencia de operación. Se recomienda considerar un espaciamiento de  $\pm 11\text{ MHz}$  respecto a la frecuencia de operación
- Los límites de la máscara espectral marcados por la ETSI son cumplidos con excepción de la modulación QPSK y las mencionadas señales espurias.
- La potencia máxima alcanzada por el transmisor basado en el HackRF es de  $-30\text{ dBm}$  con espurias de valor  $-44\text{dBm}$  (SFDR= $14\text{dBm}$ ).



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



## **CAPÍTULO 5. DISEÑO DE UN SISTEMA DE COMUNICACIONES**

### **SDR UNIFICADO**

En este capítulo se presentan una serie de elementos básicos para establecer el diseño de un transmisor SDR funcional, portable y que sea viable para su implementación en un proyecto satelital CubeSat. Hasta ahora se han explorado diferentes soluciones, tanto en software como en hardware, que buscan exponer las ventajas y posibilidades del empleo de la tecnología SDR en los sistemas de comunicaciones, particularmente explorando la aplicación del estándar DVB-S2 con GNU Radio y Front Ends comerciales de fácil adquisición. Sin embargo, la mesa de pruebas implementada, considerada como la primera iteración del sistema transmisor propuesto, si bien demuestra las ventajas del procesamiento de la señal a través de elementos basados en software, el hardware que la compone hace que su portabilidad sea inexistente y, por ende, no viable para una potencial aplicación en un proyecto nanosatelital.

De este modo, se busca proponer la utilización de otros elementos como responsables del procesamiento de la señal, de manera particular el uso de un FPGA, como parte del desarrollo a futuro del proyecto, para la implementación de la segunda iteración del sistema. Hay diversas razones por las cuales se establece esta propuesta pero la principal es la portabilidad y dimensionamiento que el SDR adquiriría. El FPGA realizaría las tareas del bloque de procesamiento, hasta ahora realizadas por GNU Radio funcionando dentro de un CPU, permitiendo reducir enormemente las dimensiones y masa del SDR propuesto.

La implementación del bloque de procesamiento en un FPGA no ha sido realizada durante este proyecto debido a obstáculos relacionados con el licenciamiento de los diferentes ambientes de desarrollo propios de los FPGAs. Por esta razón, durante este capítulo, se proporciona información y un análisis amplio para facilitar la implementación del sistema DVB-S2 generado en GNU Radio dentro de un FPGA y se enlistan algunos ambientes de desarrollo y sus requerimientos mínimos a considerar.

De igual modo, se exploran diversas opciones de Front-ends a utilizar y, en conjunto, se busca establecer el diseño integro de un transmisor SDR a desarrollar como trabajo a futuro.

Se exploran dos propuestas principalmente:

- a) FPGA + Front End COTS (dispositivos independientes)
- b) FPGA & Front End (Solución SDR embebida)

Se realiza un compilado de tablas con las características principales de transceivers, front ends y SDR completamente integrados para visualizar la gama de soluciones disponibles en el mercado.

Posteriormente, se detallarán los diseños propuestos así como las implicaciones técnicas y económicas de cada solución. El objetivo final de este capítulo es brindar los elementos óptimos y establecer la interacción entre ellos para la integración final de un SDR portable en un CubeSat de 3U. Todo esto con base en los productos y soluciones comerciales disponibles actualmente en el mercado. Es decir, el diseño de un SDR transmisor basado en componentes comerciales COTS.

### 5.1. POTENCIAL IMPLEMENTACIÓN EN FPGA

Como se menciona en el Capítulo 1, un FPGA es un semiconductor conformado por diversos bloques lógicos programables (Configurable Logic Blocks, CLBs) que puede ser usado para llevar a cabo desde funciones sencillas hasta emular sistemas complejos. Las principales ventajas de los FPGAs son su viabilidad de reprogramación, sus bajos costos de desarrollo y adquisición y su portabilidad. De este modo, los FPGAs han sido adoptados por diferentes investigadores y desarrolladores dentro de diferentes campos de la industria y la tecnología tales como medicina, comunicaciones, procesamiento de señales, aviónica, espacio, finanzas, ejército, electrónica, entre otras que se benefician de su flexibilidad y capacidad de reconfiguración [5\_1].

Dentro de un FPGA, los diferentes CLBs de los cuales se conforma están acomodados e interconectados entre ellos dentro de una matriz. Estas interconexiones son las que brindan a la matriz su capacidad de reconfiguración pues pueden ser modificadas sin restricción. Contrario a los sistemas basados en ASICs, los cuales poseen un diseño fijo sin posibilidad de ser reconfigurados, los FPGAs pueden ser reprogramados para modificar o mejorar una o varias de las características que brindan al sistema al cual pertenecen [5\_2].

Existen diferentes proveedores de FPGAs los cuales desarrollan sus propias arquitecturas reconfigurables pero en esencia utilizan los mismos componentes básicos: CLBs, Celdas o Elementos Lógicos, buffers (bloques I/O), bloques de RAM y bloques de DSP. En conjunto, logran la ya mencionada reconfigurabilidad y además optimizan la utilización de recursos y la frecuencia de operación del dispositivo.

En la figura 5.1., se aprecia la estructura de un FPGA conformado principalmente por bloques lógicos, bloques Input/Output (I/O), bloques RAM e interconexiones programables.

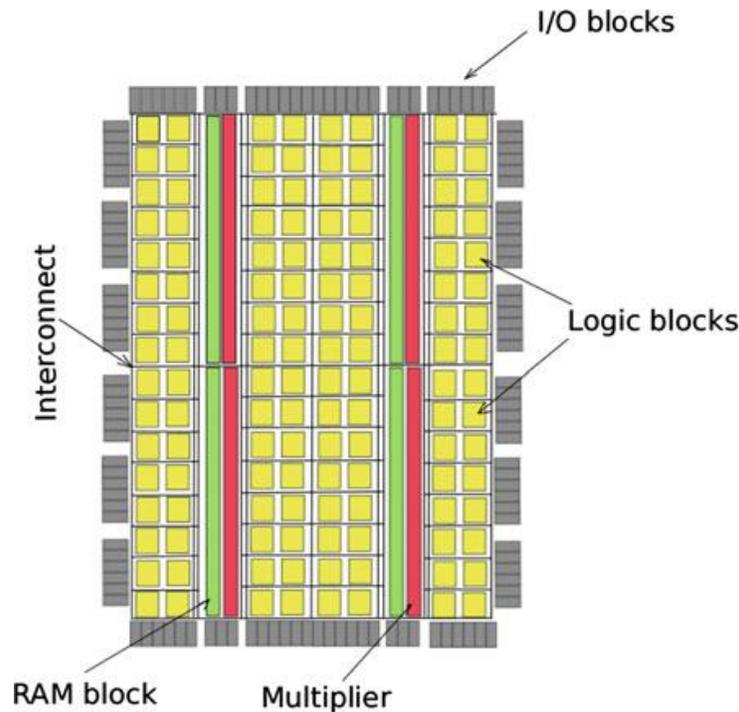


Figura 5.1. Estructura de un FPGA [5\_3].



Los CLBs constituyen el principal recurso lógico para la implementación de circuitos lógicos y combinacionales. Cada bloque lógico contiene 4 segmentos que a su vez están interconectados con los segmentos de los bloques lógicos vecinos. Cada uno de estos segmentos contiene dos LookUp Table (LUT), que es un generador de funciones lógicas basado en el empleo de memoria RAM, dos elementos de almacenamiento, multiplexores y compuertas aritméticas.

Los LUTs son elementos lógicos esenciales para la construcción de los FPGAs. Son generadores de funciones que pueden ser configurados como una memoria RAM distribuida o como un registro de desplazamiento de 16 bits. Cada LUT tiene 4 entradas lógicas y una sola salida. Cualquier operación lógica de tipo booleano que involucre hasta 4 variables puede ser implementada en un LUT. Cuando se precisa de operaciones con más variables se emplean LUTs en cascada y sus salidas son llevadas a elementos de almacenamiento o a multiplexores lógicos. El número de LUTs en un FPGA está directamente relacionado con su capacidad de procesamiento.

Por otro lado, los bloques I/O contienen registros y buffers que proveen de una interfaz programable entre la lógica interna del FPGA y la información entrante. Básicamente son el medio de comunicación del FPGA con los elementos o dispositivos externos.

### **5.1.1. Lenguajes de programación en los FPGA**

Los FPGAs requieren ser programados utilizando un lenguaje de descripción de hardware (Hardware Description Language, HDL), el cual es un lenguaje de computación especializado para describir el comportamiento y la estructura de circuitos digitales. Una de las principales características de estos lenguajes radica en su capacidad para describir en distintos niveles de abstracción cierto diseño de un sistema dado. Los niveles de abstracción se emplean para clasificar modelos HDL según el grado de detalle y precisión de sus descripciones. El lenguaje difiere de los lenguajes de programación convencionales, en que la ejecución de las sentencias no es estrictamente lineal. Un diseño en HDL consiste de una jerarquía de módulos los cuales son definidos con conjuntos de puertos de entrada, salida y bidireccionales. Internamente, contienen una lista de cables y registros y las sentencias concurrentes y secuenciales definen el comportamiento del mismo, describiendo las relaciones entre los puertos, cables y registros. Las sentencias secuenciales son colocadas dentro de un bloque inicio y ejecutadas en orden secuencial, pero todas las sentencias concurrentes y todos los bloques inicio son ejecutadas en paralelo en el diseño.

De entre los diferentes lenguajes desarrollados VHDL y Verilog son los HDLs más difundidos actualmente, donde el primero es comúnmente utilizado por los dispositivos fabricados por Xilinx y el segundo por Altera.

VHDL es el acrónimo de Very High-Speed Integrated Circuit Hardware Description Language, y fue desarrollado alrededor de 1980 a solicitud del Departamento de Defensa de los Estados Unidos. Al principio, el objetivo principal de VHDL era la simulación de circuitos eléctricos; sin embargo, las herramientas para la síntesis e implementación en hardware basadas en el comportamiento de VHDL o los archivos de descripción de estructura se desarrollaron más adelante. Con el uso creciente de VHDL, se generó la necesidad de estandarizarlo por lo que en 1986, el IEEE estandarizó el primer lenguaje de descripción de hardware, VHDL, a través de los estándares 1076 y 1164. Además, VHDL es independiente de la tecnología y del proveedor por lo que los códigos VHDL son adaptables y reutilizables [5\_4].

Por otra parte, Verilog fue inventado por Phil Moorby en 1985. Con el incremento en el éxito de VHDL, Verilog fue transformado en un lenguaje libre y después fue enviado al IEEE que lo convirtió en el estándar IEEE 1364-1995,



habitualmente referido como Verilog 95. Posteriormente, se liberó al público el Standard revisado en Marzo del 2002 conocido como IEEE 1364-200. Esta versión tuvo muchos errores los cuales fueron revisados y corregidos en la versión del 2003, conocida en la IEEE 1364-2001 revisión C [5\_5].

A continuación se enumeran algunas ventajas y desventajas que representa desarrollar circuitos integrados con este tipo de lenguajes [5\_6]:

### **Ventajas**

- ✓ Disponibilidad pública.

VHDL y Verilog son estándares no sometidos a patente o a alguna marca registrada, por lo que cualquier persona, empresa o institución puede utilizarlos sin restricciones. Además, el IEEE los mantiene y documenta, por lo que se tiene la garantía de estabilidad y soporte.

- ✓ Independencia tecnológica de diseño.

Se diseñaron para soportar diversas plataformas tecnológicas, como son los PLD, FPGA y los ASIC, con distinta funcionalidad, a fin de satisfacer las distintas necesidades de diseño.

- ✓ Independencia en tecnología.

Se crearon para que funcione de igual manera en circuitos diseñados con tecnología bipolar, MOS, CMOS, BICMOS, etc.

- ✓ Capacidad descriptiva.

Permiten diseñar en cualquiera de los diferentes niveles de abstracción (algorítmico, RTL y lógico) y combinarlos entre sí para conseguir el diseño final.

- ✓ Independencia de proveedores.

Debido a que son un lenguaje estándar, permiten que las descripciones o modelos generados para un dispositivo de cierta compañía sean compatibles para cualquier otro dispositivo de compañías diferentes.

- ✓ Reutilización del código.

Permiten reutilizar código en diversos diseños, sin importar que hayan sido generados para una plataforma (FPGA, CPLD, ASIC, etc.) y una tecnología (MOS, CMOS, bipolar, etc.) en particular.

### **Desventajas**

- Posible pérdida de la libertad de diseño.

Con la finalidad de facilitar el esfuerzo que requiere el diseño de aplicaciones, algunas compañías de dispositivos programables proveen herramientas de diseño con características adicionales al lenguaje (características no estándar), por lo que la libertad de diseño se ve limitada.



- Lenguaje difícil de aprender.

Debido a que VHDL fue desarrollado por un comité, en el cual se debió satisfacer las diversas opiniones de sus miembros, VHDL presenta alta complejidad, convirtiéndolo en un lenguaje difícil de aprender.

Verilog se diseñó basándose en el lenguaje de programación C, con el fin de que resultara familiar para los diseñadores y así fuese rápidamente aceptado. Por ello, Verilog tiene un preprocesador como C y la mayoría de sus palabras reservadas son similares a las de C por lo que presenta cierta facilidad respecto a VHDL pero en esencia ambos lenguajes son menos amigables que C.

### 5.1.2. Comparativa entre dispositivos FPGA

Como se menciona anteriormente, los proveedores más importantes de FPGAs son Xilinx y Altera. Con el objetivo de establecer recomendaciones para la implementación del módulo DVB-S2 se hace una comparativa de diferentes FPGAs en el mercado. Importante mencionar que se muestran FPGAs diseñados para aplicaciones terrestres. De manera general, los FPGAs de aplicación terrestre podrían ser utilizados para aplicaciones satelitales considerando la exposición a la radiación y el impacto que esta tenga en el desempeño del FPGA. En el siguiente capítulo se hace una breve mención de técnicas a considerar para reducir los efectos de la radiación espacial.

Se hace la diferencia entre FPGAs que están disponibles con tarjetas de desarrollo y los que no. Para la segunda iteración del SDR se recomienda la utilización de una tarjeta de desarrollo pues disminuirá de manera importante la complejidad y tiempo de implementación con la desventaja del aumento en dimensiones, masa y potencia consumida.

Para la integración final del SDR, el empleo de un FPGA independiente a una tarjeta de desarrollo es necesario debido al altísimo consumo de potencia de estas, el cual es prohibitivo para una misión satelital. De igual modo, se recomienda explorar la viabilidad de Front Ends comerciales como los que se enlistan en el siguiente capítulo.

En la tabla 5.1. se muestra una comparativa de los FPGA de última generación disponibles en el mercado.



Tabla 5.1. Comparativa de FPGAs comerciales.

FPGAs (Aplicaciones Terrestres)												
Proveedor	Altera [5_7]	Xilinx [5_8]	Xilinx [5_9]	Xilinx [5_10]	Xilinx [5_11]	Xilinx [5_12]	Xilinx [5_13]	Altera [5_14]	Micro semi [5_15]	Altera [5_14]	Xilinx [5_16]	Xilinx [5_17]
Modelo	Stratix V GT	Virtex UltraScale+ VU9P	Virtex UltraScale	Virtex 7 VX48 5T	Kintex UltraScale+	Kintex UltraScale XCKU-40	Kintex 7 XC7K 325T	Cyclone IV EP4C E115	SF2 (SoC)	Cyclone IV EP4C E40	Artix 7 35T	Spartan 7 50T
Tarjeta de desarrollo	--	VCU118	VCU108	VC707	KCU116	Dev Kit	Genesys 2	--	--	--	Basys-3	Arty S7
Licencia de desarrollo	Quartus Prime	Vivado Voucher Design Suite	Vivado Voucher Design Suite	--	Vivado Voucher Design Suite	Vivado Voucher Design Suite	Vivado Voucher Design Suite	Quartus Prime	--	Quartus Prime	Vivado WebPack License	Vivado WebPack License
Tecnología	14 nm 3D Tri-Gate transistor	16 nm	20 nm	28nm HPL	16 nm	20 nm	28nm HPL	60nm	CRI portfolio	60nm	28 nm HPL	28nm HPL
Frecuencia de reloj (MHz)	875	528	450	1600	528	528	450	500	166	500	450	450
LUTs	234k	2586k	4704k	1,955k	1900k	1451k	203k	114k	146k	40k	33k	32k
Tasa de datos [Gbps] / número de canales	28.05 (4) / 12.5 (32)	30.5	30.5	2,784 (Full Duplex)	28	12.5	10.3	3.125 (3)	5 (16)	3.125 (3)	6.6 (16)	---
Bloques DSP	512	6840	768	3600	1824	5520	840	360	240	90	90	120
Precio [USD]	13k	7k	6k	4k	3k	1k	1K	600	300	200	200	180

Para la selección de la mejor solución se deben de considerar parámetros tales como la capacidad de procesamiento del FPGA (LUTs), su velocidad de procesamiento (frecuencia de reloj), la tasa de datos a la que puede emplearse y su precio. De igual modo, uno de los puntos más importantes al momento de hacer la selección es su licenciamiento. Esto es importante dado que todos los fabricantes de FPGAs cuentan con un software específico para poder diseñar, simular y programar los FPGAs. Las licencias anuales de estos programas (por ejemplo Vivado o Quartus) llegan a tener un precio arriba de los 60,000 MXN por lo que el costo del proyecto podría incrementarse drásticamente si no se tiene en mente este factor. Algunas tarjetas de desarrollo vienen con una licencia gratuita de uso limitado que permite utilizar la utilería básica de cada programa pero con diversas restricciones.

Para iniciar con la selección de un FPGA se propone realizar un estimado de los requerimientos específicos del sistema a implementar. En este caso del transmisor DVB-S2. Los pasos propuestos son los siguientes:

1. Realizar un estimado del número de LUTs empleado por el sistema.
2. Comprobar que la tasa de datos es lo suficientemente alta para cumplir con los requerimientos de comunicación.
3. Establecer si la velocidad de procesamiento es la adecuada para la aplicación deseada.
4. Considerar el licenciamiento incluido al adquirir el FPGA o la tarjeta de desarrollo.
5. Evaluar el precio final.



Para DVB-S2 se comienza realizando un estimado del número de LUTs que podría llegar a requerir el sistema. Para ello se realiza una búsqueda de subsistemas propios de este estándar desarrollados previamente en otros proyectos alrededor del mundo.

De este modo, podemos considerar la información recabada en la tabla 5.2.

**Tabla 5.2. LUTs utilizados en un FPGA para bloques de DVB-S2.**

Bloque	Detalles	FPGA	LUTs utilizados	Referencia
Scrambler	LFSR	Xilinx Spartan 6	31k	[5_18]
Modulación	QPSK y 8PSK	Xilinx Virtex 5 SX50T	22k	[5_19]
Codificador LDPC	DVB-T2	Xilinx Virtex 4	5k	[5_20]
Filtro	(square root raised cosine) roll off: 0.2,0.25,0.35 Order 96	Xilinx Spartan 6	4k	[5_21]

Al considerar el número de LUTs empleados por estos 4 bloques se determina que el número de LUTs disponibles en el FPGA deberá ser mayor a los 60k. Otros bloques que pueden llegar a sumar una cantidad importante de LUTs son el Interleaver, el codificador BCH y el Bit Mapper por lo cual estableceremos el valor requerido arriba de los 100k LUTs.

En cuanto a la frecuencia de reloj se sabe que para diferentes aplicaciones de comunicaciones una velocidad mayor a 250 MHz es recomendada [5\_3]. Por debajo de esta velocidad el desempeño del sistema comienza a ser menor.

Finalmente, el último factor a considerar debería ser el precio en conjunto con el licenciamiento. De este modo, la opción más económica y viable para la implementación del transmisor DVB-S2 sería el FPGA Cyclone IV EP4CE115 de Altera seguido del Kintex 7 XC7K325T de Xilinx. La elección del proveedor dependerá del acceso y la compatibilidad a los ambientes de desarrollo que se tengan disponibles.

Sin embargo, un factor adicional a considerar es la experiencia que el usuario o el equipo de trabajo tiene con dispositivos FPGAs. Como una primera interacción se recomienda la adquisición de una tarjeta de desarrollo económica como la Basys-3 de Xilinx mostrado en la figura 5.2. De este modo, podría comenzar a utilizarse el FPGA para la implementación de sistemas sencillos al mismo tiempo que se adquiere una mayor experiencia que ayudará a una mejor toma de decisiones con implementaciones más robustas o complejas.



**Figura 5.2. Tarjeta de desarrollo Basys-3 de Xilinx [5\_16].**



### 5.1.3. Entornos de diseño y simulación para FPGAs

La simulación es un paso crítico en el diseño de aplicaciones basadas en FPGAs dado que permite al diseñador probar su diseño y ver cómo el código que escribió reacciona ante diferentes entradas. Con una simulación es posible emular todos los estados posibles del diseño para garantizar que todos los escenarios de entrada se manejen adecuadamente. En el caso de los FPGAs existen diferentes herramientas de diseño y simulación que son esenciales en el desarrollo de los proyectos. De igual modo, estas herramientas proporcionan al usuario una mayor accesibilidad al estar basadas en interfaces gráficas y el desarrollo de sistemas empleando bloques y conectores visuales, lo cual es mucho más fácil de manipular que los programas basados en la síntesis de HDL tales como Vivado y Quartus. A continuación se hace una breve mención de las principales herramientas de diseño y simulación.

#### 5.1.3.1. Entornos de Simulación

##### RFNoC (RF Network-on-Chip)

RFNoC es una herramienta de procesamiento, desarrollado por Ettus Research, enfocado en habilitar el procesamiento de FPGAs en dispositivos USRP. Permite el flujo de datos dentro y fuera de un FPGA de manera transparente, lo que da oportunidad al uso continuo del procesamiento de señales embebido en un FPGA propio de los equipos USRP [5\_22].

Los algoritmos de procesamiento de señales están contenidos en módulos conocidos como "Bloques RFNoC", y se proporciona un envoltorio de interfaz para encapsular los bloques entre RFNoC y el propio software de programación del FPGA. Esto funciona como un puente de datos entre ambos entornos de programación. Los componentes internos de un bloque RFNoC son totalmente independientes de cualquier otro bloque y pueden diseñarse con cualquier herramienta que admita interfaces de flujo AXI, incluidos VHDL, Verilog y Xilinx Vivado HLS.

RFNoC está integrado con el software UHD, y todos los dispositivos USRP de la tercera generación en adelante (Serie X300, Serie E300) son compatibles con RFNoC de fábrica. Adicionalmente, RFNoC también es software libre y de código abierto y es compatible con GNU Radio por lo que se convierte en una herramienta sumamente atractiva.

##### LabView FPGA

LabVIEW FPGA, fig. 5.3, es una herramienta de simulación y diseño de sistemas complejos el cual proporciona un entorno de desarrollo integrado en el cual es posible combinar el acceso directo a los puertos I/O de la FPGA con bloques lógicos propios de LabVIEW definidos por el usuario para definir el comportamiento del hardware para aplicaciones como comunicación de protocolos digitales y simulación de hardware-in-the-loop. Aunque LabVIEW FPGA contiene varias rutinas integradas de procesamiento de señales, también es posible integrar código HDL existente. El costo de la licencia es de 61,000 MXN por un año [5\_23].

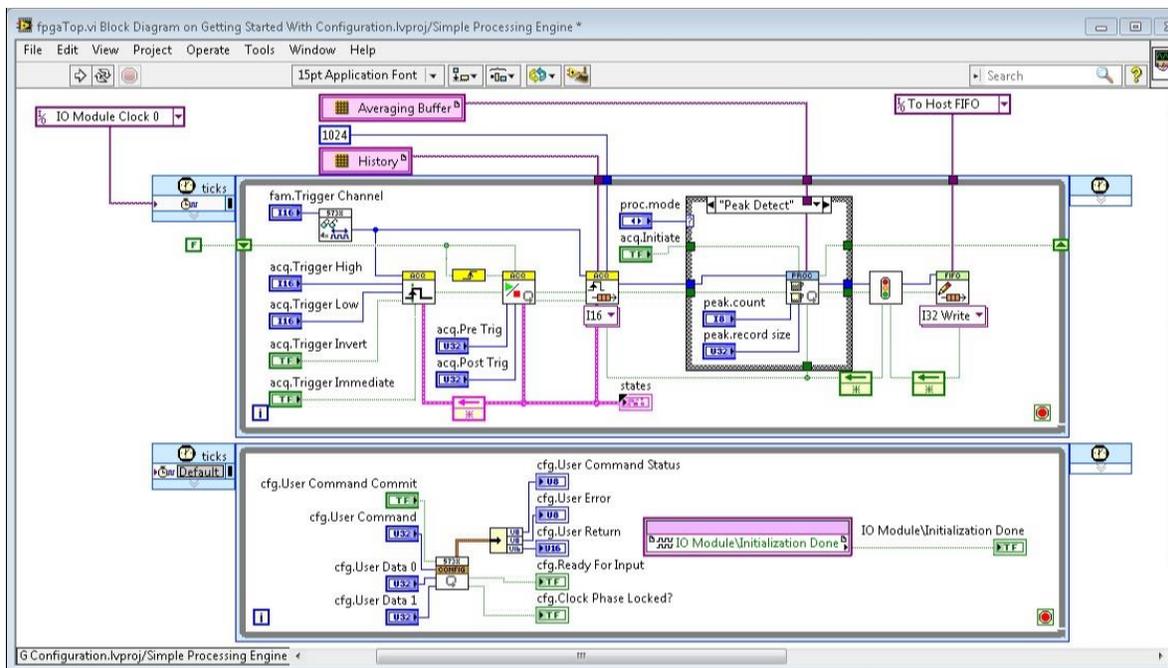


Figura 5.3. Entorno gráfico de LabView [5\_23].

## ModelSim

Modelsim es un programa creado por Mentor Graphics el cual es utilizado para simular diseños basados en VHDL y Verilog. ModelSim es un entorno de simulación HDL en varios idiomas de Mentor Graphics, para la simulación de lenguajes de descripción de hardware como VHDL, Verilog y SystemC, e incluye un depurador C incorporado. Es posible utilizar ModelSim de forma independiente o en combinación con Intel Quartus Prime, o Xilinx Vivado. La simulación se realiza mediante la interfaz gráfica de usuario (GUI) o automáticamente mediante scripts [5\_24].

## Matlab Simulink

Simulink es un entorno de programación gráfico para el modelado, la simulación y el análisis de sistemas dinámicos perteneciente a Matlab, desarrollado también por MathWorks. Dispone de una herramienta gráfica con la que se diseñan diagramas con el uso de librerías en las que se encuentran diversos bloques básicos con los cuales programar. Respecto al diseño de aplicaciones con FPGAs, los principales fabricantes como son Altera y Xilinx ofrecen soluciones que permiten exportar modelos de Simulink a sus dispositivos programables, evitando así el paso manual del modelo desarrollado en Simulink a un lenguaje de descripción de hardware. Además, dichas herramientas también permiten verificar y validar el diseño en la propia FPGA (hardware in the loop) donde se sustituye la parte que va en la FPGA del modelo por la propia FPGA, conectándola al PC para nutrirla de entradas y analizar las salidas.

Esta herramienta no está ligada a ningún fabricante de FPGAs pues es capaz de crear directamente código HDL, tanto Verilog como VHDL, del modelo que se quiera exportar. De esta forma, un modelo desarrollado en Simulink y que sea sintetizable puede exportarse a código HDL y luego sintetizarlo en cualquier fabricante de FPGAs, tanto Altera como Xilinx o cualquier otro. El motor de este tipo de enfoque es el módulo HDL Coder [5\_25].

### 5.1.3.2. Entornos de Implementación

#### Vivado Xilinx

Vivado Design Suite es un software producido por Xilinx para síntesis y análisis de diseños HDL. Incluye un simulador lógico incorporado y presenta síntesis de alto nivel, con una cadena de herramientas que convierte el código C en lógica programable. La herramienta permite a los desarrolladores sintetizar sus diseños (fig. 5.4), realizar análisis en el tiempo, examinar diagramas RTL, simular la reacción de un diseño a diferentes estímulos (fig. 5.5) y configurar el FPGA para su programación. Vivado es un entorno de diseño para productos FPGA de Xilinx, y está estrechamente acoplado a la arquitectura de dichos chips, por lo cual no puede utilizarse con productos FPGA de otros proveedores. Una versión gratuita de WebPACK Edition de Vivado ofrece a los diseñadores una versión limitada del entorno de diseño [5\_26].

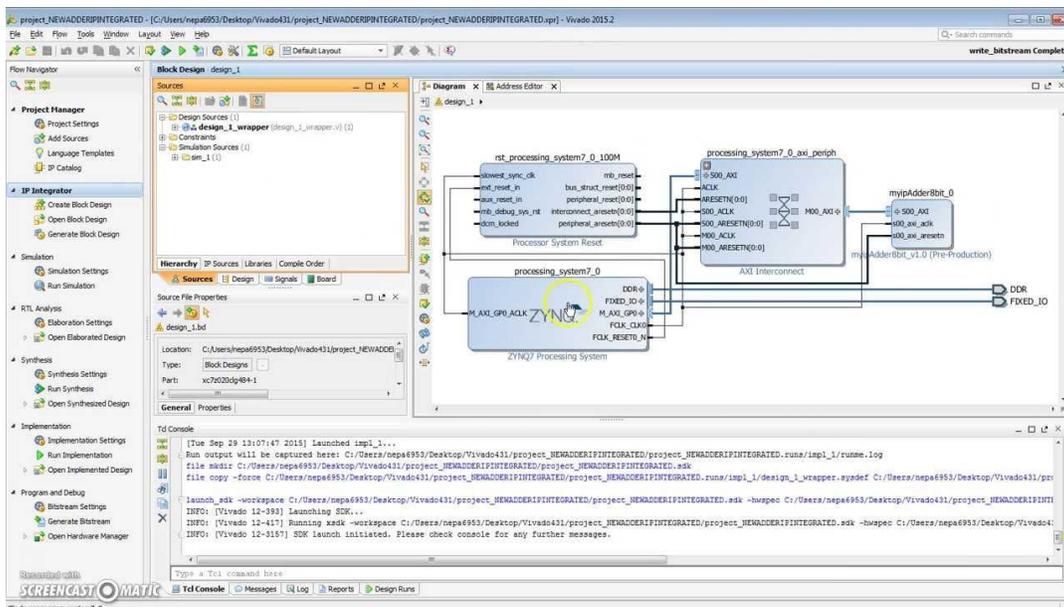


Figura 5.4 Entorno de desarrollo de Vivado [5\_26].



Figura 5.5. Simulador de la herramienta Vivado [5\_26].

### Altera Quartus

Intel Quartus Prime es un software de diseño producido por Intel. Antes de la adquisición de Altera por parte de Intel, la herramienta se llamaba Altera Quartus II. Quartus Prime permite el análisis y la síntesis de diseños HDL, lo que permite al desarrollador compilar sus diseños, realizar análisis en el tiempo, examinar diagramas RTL y simular la respuesta de un diseño a diferentes estímulos. Quartus Prime incluye una implementación de VHDL y Verilog para la descripción del hardware, edición visual de circuitos lógicos y simulación de forma de onda vectorial. Ver fig. 5.6.

La edición Web es una versión gratuita de Quartus II que proporciona compilación y programación para un número limitado de dispositivos Intel FPGA. La familia Cyclone de FPGA de bajo costo es totalmente compatible con esta edición, lo que significa que los pequeños desarrolladores y las instituciones educativas no tienen gastos generales del costo del software de desarrollo. Se requiere registro de licencia para usar la Edición Web de Quartus Prime, que es gratuita y puede renovarse un número ilimitado de veces.

La edición de suscripción también está disponible para descarga gratuita, pero se debe pagar una licencia para usar la funcionalidad completa en el software con un valor aproximado de 60,000 MXN [5\_27].

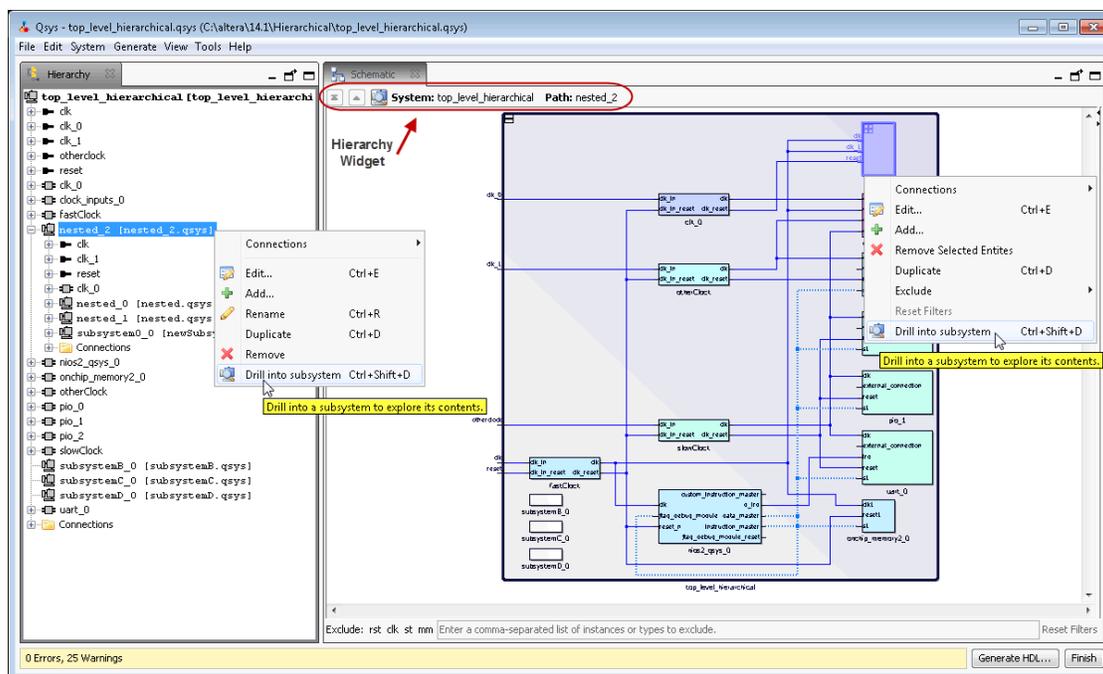


Figura 5.6. Entorno gráfico de Quartus Prime [5\_27].



## 5.2. FRONT ENDS COMERCIALES

Con el objetivo de entregar una propuesta de Front End tipo COTS, se plantea el realizar un primer análisis enfocado en los principales transceivers existentes en el mercado. Se decide partir de esta comparativa ya que los transceptores son uno de los componentes básicos dentro de las etapas de RF y de estos depende, junto con los amplificadores de potencia y filtros, el rango de frecuencias de operación del sistema. De este modo, al tener una noción clara de los alcances y límites de estos, es posible entender de mejor forma los alcances de los Front Ends. La selección de los transceiver busca revisar aquellos que son utilizados en la mayoría de los Front End y se encuentran disponibles en el mercado. Los principales parámetros a comparar son:

- Frecuencia de operación
- Ancho de banda
- Potencia de salida
- Potencia consumida

Los datos recabados son mostrados en la tabla 5.3.

**Tabla 5.3. Transceivers comerciales.**

Transceiver	Proveedor	Frecuencia TX/RX	Ancho de banda	Potencia a Salida	Potencia Máxima Consumida	ADC/DAC	Costo [USD]	Referencia
<b>LMS6002D</b>	Lime Microsystems	300 MHz – 3.8GHz	28 MHz	6 dBm	1.5 W	12 bit	110	[5_28]
<b>LMS7002M</b>	Lime Microsystems	100 kHz- 3.8 GHz	60 MHz	0 dBm	1 W	12 bit	130	[5_28]
<b>LMS8001+</b>	Lime Microsystems	100 kHz – 12 GHz	120 MHz	6 dBm	1.5 W	12 bit	330	[5_28]
<b>AD9364</b>	Analog Devices	70 MHz – 6 GHz	56 MHz	6.5 dBm	3W	12 bit	230	[5_29]
<b>AD9361</b>	Analog Devices	70 MHz – 6 GHz	56 MHz	6.5 dBm	3W	12 bit	300	[5_30]
<b>ADF7242</b>	Analog Devices	2400 – 2483.5 MHz	2.5 MHz	4.8 dBm	3.6 W	5 bit	6	[5_31]
<b>CC2500</b>	Texas Instruments	2400 – 2483.5 MHz	812 kHz	1 dBm	77 mW	--	5	[5_32]

De los datos mostrados se hace evidente que los transceivers ADF7242 y CC2500 son utilizados específicamente para la banda global ISM (Industrial, Scientifics & Medic) de los 2.4 GHz. El resto de los elementos analizados posee rangos de operación amplios que llega a los 6 GHz y hasta los 12 GHz para el caso del LMS8001+, lo cual cumple con el requerimiento de flexibilidad propio de los SDR.

Un segundo parámetro de gran importancia es el ancho de banda que cada uno de estos transceptores otorga siendo el de menor valor el LMS6002D con 28 MHz y el de mayor el LMS8001+ con 120 MHz. Para la misión planteada un valor de 10 MHz debería ser suficiente por lo que cualquiera de estos transceptores cumple con este requerimiento.



Las potencias de salida de los transceivers tienen un máximo de 6.5 dBm para los dispositivos de Analog Devices lo cual implica el uso de un amplificador de potencia que entregué una salida de aproximadamente 27 dBm para así alcanzar la potencia de transmisión mínima establecida de 500mW.

Finalmente, la potencia consumida establece un factor diferenciador, con valores que van de 1W a los 3W para las diferentes opciones disponibles. Este valor será mejor proyectado en el análisis de los Front End cuyo consumo de potencia dependerá principalmente de los convertidores de voltaje que llevan integrados.

Respecto a los Front End, hasta el momento se tuvo la posibilidad de trabajar con el HackRF, sin embargo este no es el único dispositivo disponible en el mercado. Actualmente, existen diversos Front End tipo COTS disponibles. En la tabla 5.4 se presentan los datos de varios de estos dispositivos referentes a los siguientes parámetros:

- **Frecuencia:** Rango de frecuencias de operación.
- **Ancho de banda:** Ancho de banda máximo en transmisión y recepción.
- **ADC/DAC:** Resolución en bits del transceiver embebido
- **Canales Tx/Rx:** Indica el número de canales disponibles en transmisión y recepción
- **Duplex:** Modo de transmisión con el cual es compatible
- **Interfaz:** Interfaz de comunicación entre el front end y el bloque de procesamiento previo
- **Transceiver:** Modelo del transceiver en el front end.
- **Open source:** Indica si es software libre. En algunos casos sólo el firmware es gratuito mientras que la plataforma de desarrollo tiene costo o está bajo licenciamiento.
- **GNU Radio:** Indica si el dispositivo es compatible con GNU Radio.
- **Precisión del oscilador:** Relacionado al oscilador del transceiver indicado.
- **Potencia de Tx:** Potencia final de salida.
- **Consumo de potencia:** Potencia consumida por el dispositivo utilizando un canal de transmisión.
- **Precio:** Precio de adquisición
- **Precio de lanzamiento:** Costo de lanzamiento considerando la masa (20,000 USD/kg)
- **Herencia de vuelo:** Indica si el Front End desempeñó una o varias misiones en el espacio.

**Tabla 5.4. Front End comerciales.**

Parámetro	HackRF One [5_33]	Ettus B200 [5_34]	Ettus B205 mini [5_35]	Blade RF 2.0 micro [5_36]	RTL-SDR [5_37]	LimeSDR [5_38]	FunCube Pro+[5_39]
Frecuencia	1MHz – 6GHz	70MHz – 6GHz	70MHz – 6GHz	47MHz – 6GHz	500kHz – 2.2GHz	100kHz – 3.8GHz	150kHz-1.9GHz
Ancho de banda	20MHz	61.44MHz	56 MHz	56MHz	3.2MHz	61.44MHz	1 MHz
ADC/DAC	8 bits	12 bits	12 bits	12 bits	8 bits	12 bits	16 bits
Tasa de muestreo	20MSPS	61.44MSPS	61.44MSPS	61.44MSPS	3.2MSPS	61.44MSPS	1 MSPS
Canales TX/RX	1/1	1/1	1/1	2/2	0/1	2/2	0/1
Duplex	Half	Full	Full	Full	Simplex RX	Full	Simplex RX
Interfaz	USB 2.0	USB 3.0	USB 3.0	USB 3.0	USB 2.0	USB 3.0	USB 1.0
Transceiver	MAX5864 MAX2837 RFFC5072	AD9364	AD9364	AD9361	RTL2832U	LMS7002M	NA
Open Source	Sí	Firmware	Firmware	Firmware	No	Sí	Firmware
GNU Radio	Sí	Sí	Sí	Sí	No	Sí	No
Precisión del oscilador	+/- 20ppm	+/- 2ppm	+/- 2ppm	+/- 1ppm	No	+/- 4ppm	+/- 1.5ppm
Potencia de TX	+15 dBm@2.4GHz	+10 dBm	+10 dBm	+8 dBm	NA	+10 dBm	NA
Consumo de Potencia	5V, 80mA (400mW)	6V, 400mA (2.4W)	5V, 200mA (1W)	5V, 100mA (500 mW)	4.5V, 280 mA (1.26W)	6V, 100mA (600 mW)	5V
Dimensiones [cm]	12 x 7.5 x 2	9.7 x 15.5 x 1.5	8.33 x 5.1 x 2.4	6.3 x 10.2 x 1.8	9 x 7.1 x 2	10 x 6 x 1	8 x 4 x 1
Masa [g]	180	350	24	90	200	30	30
Precio [USD]	300	700	900	480	30	300	190
Precio Lanzamiento [USD]	3600	7000	480	1800	4000	600	600
Precio Total [USD]	3900	7700	1380	2280	4030	900	790
Herencia de vuelo	No	No	No	No	No	Sí (ISS)	No

De este modo, se hace un análisis comenzando por descartar los Front End que no son aptos para los objetivos de la misión. En este caso se trata del RTL-SDR (fig. 5.7) y el FunCube Pro+ (fig. 5.8) los cuales sólo funcionan como receptor lo que los hace inviable al momento de pensar en implementarlos como un transmisor.



**Figura 5.7 Front End modelo RTL-SDR [5\_37].**



Figura 5.8 Front End FUNcube Pro+ [5\_39].

En relación a la frecuencia de operación se puede establecer que el resto de los dispositivos tienen un desempeño muy similar con un rango que alcanza hasta los 6GHz con excepción del LimeSDR con un límite superior de 3.8 GHz lo cual no impacta en los requerimientos de la misión planteada que establecen una frecuencia de operación de 2.2 GHz.

Al revisar el ancho de banda y la tasa de muestreo destaca el HackRF (fig. 5.9) como el Front End de menor desempeño con sólo 20MHz y 20 MSps. Al igual que con el LimeSDR estos valores no impactan los requerimientos de la misión establecidos en un ancho de banda de 10 MHz. El resto de los dispositivos soportan un ancho de banda arriba de los 56 MHz.

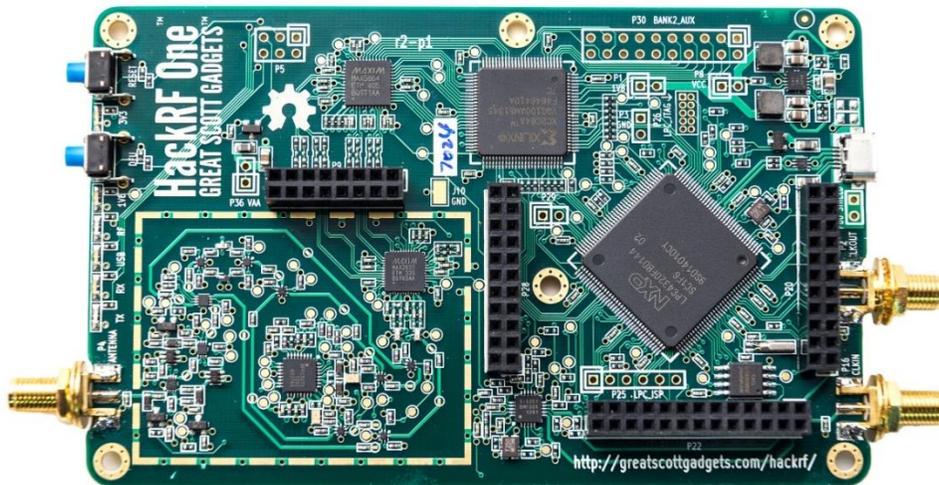
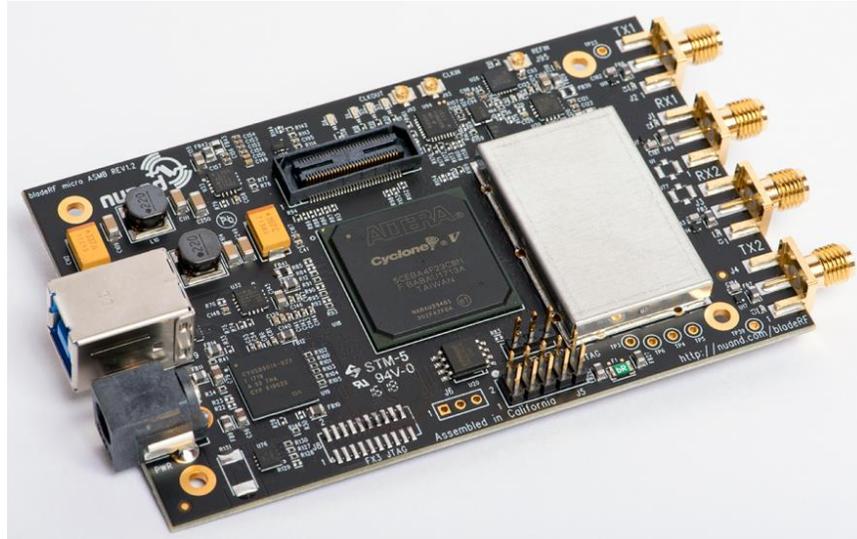


Figura 5.9. Front End HackRF [5\_33].

En cuanto al número de canales y el modo de comunicación los de mejor desempeño son el BladeRF 2.0 (fig. 5.10) y el LimeSDR ya que cuentan con 2 canales para transmisión y dos para recepción, ambos en full dúplex. Los otros Front End están diseñados con un canal de transmisión y uno de recepción también en full dúplex. De nuevo la excepción es el HackRF con un modo half dúplex.



**Figura 5.10. Front End Blade RF 2.0 [5\_36].**

Respecto a la interfaz todos los Front End se comunican usando el protocolo USB con el bloque de procesamiento previo. La mayoría son USB 3.0, interfaz con hasta 4.8 Gbps de velocidad de transferencia de datos, un valor 10 veces mayor al USB 2.0. El único dispositivo con interfaz USB 2.0. es, nuevamente, el HackRF.

El rubro de Open Source cubre por completo al HackRf y al LimeSDR. Esto significa que el acceso a los detalles de su diseño, tanto en hardware como en software, es de uso público. Ambos están basados en Linux y todas las actualizaciones de su firmware son gratuitas. Para los dispositivos Ettus B200 (fig. 5.11), B205 y el Blade RF 2.0 sólo el firmware es código abierto, pero ambas marcas poseen diferentes programas de desarrollo o simulación con algún costo y sin que sean código abierto tales como RfNoc. Así mismo, estos últimos dispositivos no ponen bajo dominio público su diseño electrónico. Todos los Front End sin descartar son compatibles con GNU Radio lo cual es de gran ventaja para aplicaciones en sitio.



**Figura 5.11. Front End ETTUS B200 [5\_34].**

La potencia de transmisión de todos los Front End ronda los 10 dBm (10 mW). Esto significa que el uso de un amplificador de potencia adicional deberá de ser previsto con el objetivo de alcanzar por lo menos una potencia de transmisión equivalente a 500mW (+27 dBm) de acuerdo al análisis de comunicaciones desarrollado en el capítulo 3. Destaca el HackRf presentando una potencia máxima de 15 dBm (30 mW).

En relación al consumo de potencia se observa que el Front End que presenta un menor consumo es el HackRF con 400mW mientras que el Ettus B200 es el de mayor consumo con 2.4W. Se desea mantener el consumo del Front End por debajo de 1W dado que a este valor debe de adicionarse el consumo producido por el FPGA encargado de la implementación del estándar DVB-S2 también considerado parte del subsistema de comunicaciones. Este valor lo cumplen los dispositivos BladeRF, Ettus B205 y el LimeSDR.

Por otro lado, uno de los parámetros más importantes dentro de este análisis comparativo es sin duda las dimensiones y la masa de los dispositivos. Como ya se mencionó anteriormente, el proyecto nanosatelital planteado tiene unas dimensiones definidas por el estándar CubeSat de 10 x 10 x 34 cm y un masa de hasta 4kg. Mientras más reducidas sean las dimensiones de los subsistemas mayor será el espacio destinado a la carga útil del proyecto. De igual modo, si se logra minimizar la masa del satélite al máximo se optimizará el costo final del proyecto pues el costo de lanzamiento depende totalmente de la masa del CubeSat. Actualmente el costo para poner en una órbita SSO un satélite es de 20,000 USD/kg [5\_40]. Este costo es el más bajo del mercado y es provisto por la compañía SpaceX [5\_41].

De este modo, los Front End que son más convenientes, en términos de dimensiones, son aquellos que no superan los 10cm por lado, en este caso el Ettus B205 mini y el LimeSDR. Esto tiene gran relevancia pues en el caso de que las dimensiones de algún dispositivo exceda los 10 cm se hace necesario que la disposición del mismo sea a lo largo del eje Z del CubeSat lo cual implica un acomodo totalmente diferente al normalmente utilizado en este tipo de proyectos el cual es lo largo del eje X, tomando como referencia la figura 5.12.

El seleccionar alguno de los otros Front End con dimensiones superiores a los 10 cm, como el HackRf o el BladeRf 2.0, implica que el diseño interno del CubeSat y el acomodo de las placas del mismo cambie por completo respecto al diseño estandarizado. Esto último no es necesariamente un impedimento pues existen diferentes proyectos que han adoptado este tipo de diseño sin embargo presenta diversas desventajas relacionadas con la adquisición y acomodo de PCBs fuera de los lineamientos marcados por el estándar PC/104 [5\_42], el cual a su vez ha sido adoptado por la comunidad CubeSat en general.

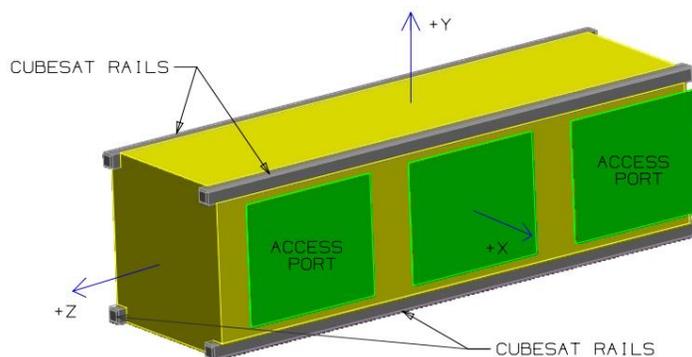


Figura 5.12. Esquema de un CubeSat 3U [5\_43].

De igual modo, los Front End con menor masa son el Ettus B205 mini (fig. 5.13) y el LimeSDR con 24 y 30 gramos respectivamente, esto representa un precio de lanzamiento de 480 y 600 USD. En el polo opuesto se encuentra el Ettus B200 con una masa de 350 gramos, equivalente a una diferencia de 6,520 USD en términos de costo de lanzamiento respecto al Ettus B205.



**Figura 5.13. Front End Ettus USRP B205 mini-i [5\_35].**

Por último, el precio de adquisición de cada dispositivo va de los 300 USD para el HackRF y el LimeSDR a los 900 USD para el Ettus B205 mini. Si bien la diferencia entre ambos precios puede llegar a ser representativa, para términos generales de un proyecto satelital 600 USD no son significativos en comparación con las ventajas que otros parámetros como dimensión, masa y consumo de potencia pueden otorgar.

De lo anterior se puede concluir que las mejores opciones para la misión satelital propuesta son el LimeSDR y el Ettus B205 mini. Ambos cumplen con los requisitos de frecuencia y ancho de banda estipulados, sus dimensiones son reducidas y por debajo de los 10cm, su baja masa optimiza el costo de lanzamiento y su consumo de potencia es de igual modo reducido por lo cual la demanda al subsistema de potencia se verá amortizado.

Las principales diferencias entre ambos son el rango de frecuencia, siendo el Ettus B205 mini el de mayor rango con un límite superior de 6 GHz, y el número de canales disponibles, en este caso el LimeSDR con mayor ventaja al tener 2 canales de transmisión y 2 de recepción.

Como criterio final se decide tomar en consideración la herencia de vuelo. En este rubro solamente el Front End LimeSDR ha sido probado en el espacio, esto dentro del proyecto MarconISSta desarrollado por la Universidad Politécnica de Berlín y el cual tenía el objetivo de monitorear el uso global del espectro radioeléctrico desde la ISS. El proyecto tuvo lugar de agosto del 2018 a febrero del 2019 con resultados exitosos [5\_44].

Tomando todo lo anterior en consideración se considera al LimeSDR como la mejor propuesta de Front End para la misión CubeSat propuesta en este proyecto.

Como resumen del análisis anterior se puede puntualizar que:

1. RTL-SDR y FunCube Pro+ no son aptos al contar sólo con la etapa de recepción.
2. El resto de los Front End cumplen con los requerimientos de frecuencia y ancho de banda establecidos.
3. HackRf, Ettus B200 y BladeRF presentan la desventaja de tener dimensiones mayores a los 10cm.
4. HackRF y Ettus B200 son las dos propuestas de mayor masa, lo cual afecta el costo final del proyecto con un impacto de hasta 6,520 USD.
5. Las dos mejores propuestas, en términos de frecuencia, ancho de banda, dimensiones, masa y costo, son el Ettus B205 y el LimeSDR.
6. La mejor propuesta es el LimeSDR (fig. 5.14) por su mayor cantidad de canales de TX/RX y por su herencia de vuelo en la órbita ISS.



Figura 5.14. Front End Lime SDR [5\_38].

Finalmente, en la tabla 5.5, haciendo uso nuevamente de la tabla 5.4, se han reorganizado las columnas de peor a mejor propuesta y marcado con diferentes colores algunos de los parámetros de los dispositivos estudiados con base en el siguiente criterio:

**Verde:** Mejor valor respecto a los demás dispositivos.

**Amarillo:** Valor que presenta algún inconveniente para la misión planteada.

**Rojo:** Valor prohibitivo. No apto para la misión planteada.



Tabla 5.5. Front End comerciales clasificados.

Parámetro	FunCube Pro +	RTL-SDR	Ettus B200	HackRF One	Blade RF 2.0 micro	Ettus B205 mini	LimeSDR
Frecuencia	150kHz-1.9GHz	500kHz – 2.2GHz	70MHz – 6GHz	1MHz – 6GHz	47MHz – 6GHz	70MHz – 6GHz	100kHz – 3.8GHz
Ancho de banda	1 MHz	3.2MHz	61.44MHz	20MHz	56MHz	56 MHz	61.44MHz
ADC/DAC	16 bits	8 bits	12 bits	8 bits	12 bits	12 bits	12 bits
Tasa de muestreo	1 MSPS	3.2MSPS	61.44MSPS	20MSPS	61.44MSPS	61.44MSPS	61.44MSPS
Canales TX/RX	0/1	0/1	1/1	1/1	2/2	1/1	2/2
Duplex	Simplex RX	Simplex RX	Full	Half	Full	Full	Full
Interfaz	USB 1.0	USB 2.0	USB 3.0	USB 2.0	USB 3.0	USB 3.0	USB 3.0
Transceiver	NA	RTL2832U	AD9364	MAX5864 MAX2837 RFFC5072	AD9361	AD9364	LMS7002M
Open Source	Firmware	No	Firmware	Sí	Firmware	Firmware	Sí
GNU Radio	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Precisión del oscilador	+/- 1.5ppm	No	+/- 2ppm	+/- 20ppm	+/- 1ppm	+/- 2ppm	+/- 4ppm
Potencia de TX	NA	NA	+10 dBm	+15 dBm	+8 dBm	+10 dBm	+10 dBm
Consumo de Potencia	5V	4.5V, 280 mA (1.26W)	6V, 400mA (2.4W)	5V, 80mA (400mW)	5V, 100mA (500 mW)	5V, 200mA (1W)	6V, 100mA (600 mW)
Dimensiones [cm]	8 x 4 x 1	9 x 7.1 x 2	15.5x9.7x1.5	12x7.5x2	10.2x6.3x1.8	8.3x5.1x2.4	10 x 6 x 1
Masa [g]	30	200	350	180	90	24	30
Precio [USD]	190	30	700	300	480	900	300
Precio Lanzamiento [USD]	600	4000	7000	3600	1800	480	600
Precio Total [USD]	790	4030	7700	3900	2280	1380	900
Herencia de Vuelo	No	No	No	No	No	No	Sí (ISS)



### 5.3. SDR INTEGRADOS

En los dos subcapítulos anteriores se establecen las propuestas de FPGA y Front End a integrar para la implementación de un SDR. En este subcapítulo, por otro lado, se exploran las soluciones ya integradas de SDR disponibles actualmente en el mercado. Esto es, soluciones que cuenten con el bloque de procesamiento de la señal y el Front End analógico ya acoplados y embebidos en un mismo bloque.

Para este análisis se propone comparar tres diferentes SDR integrados. Por un lado, el USRP E320 del proveedor Ettus Research diseñado para aplicaciones terrestres, y por otro lado, el NanoCom SDR del proveedor GomeSpace y el  $\mu$ SDR-C de Space Micro, ambos diseñados para tolerar ambientes espaciales. Al igual que en el análisis de los Front End, en este se revisarán parámetros tales como: frecuencia, ancho de banda, resolución ADC/DAC, tasa de muestreo, canales de TX/RX, modo de comunicación, transceiver, potencia de TX, consumo de potencia, dimensiones, masa y precio.

Adicionalmente, se detallarán las características del FPGA empleado para el procesamiento de la señal en banda base, la tolerancia a la radiación y la herencia de vuelo. En la tabla 5.6 se muestran los datos obtenidos de cada uno de estos SDR integrados.



**Tabla 5.6. Características principales de los SDR integrados.**

	Ettus E320 [5_45]	NanoCom SDR [5_46]	μSDR-C [5_47]
Proveedor	Ettus Research	GomSpace	Space Micro
Frecuencia	70MHz – 6GHz	70MHz – 6GHz	150 MHz – 6 GHz
Ancho de banda	56MHz	56MHz	56 MHz
ADC/DAC	12 bits	12 bits	12 bits
Tasa de muestreo	61.44Msps	61.44Msps	61.44 Msps
Canales TX/RX	2/2	2/2	2/2
Duplex	Full	Full	Full
Interfaz	USB 3.0	USB 2.0	--
Transceiver	AD9361	AD9361	AD9361
Potencia de TX	+10 dBm	+10 dBm	+10 dBm
FPGA Baseband	Xilinx Zynq 7045	Xilinx Zynq 7030	Xilinx Zynq 7020
LUTs	218k	78k	53k
Software	UHD, GNU Radio, RFNoc, SDK, Vivado	SDK, Vivado	SDK, Vivado
Consumo de potencia	10 V, 3A (30W)	5.5V, 600mA (3.3W)	8W
Herencia de Vuelo	No	Sí (GomX-3@órbita ISS)	No
Tolerancia Radiación	No	Sí	30 krad SEL: <70 Mev/mg/cm <sup>2</sup> SEU: <1E-3/día SEFI: 0
Dimensiones	17.3 x 10 x 3.6 cm	9.2 x 8.9 x 2.4 cm	10 x 10 x 5 cm
Volumen	622 cm <sup>3</sup>	196 cm <sup>3</sup>	500 cm <sup>3</sup>
Masa	160 g	271 g	600 g
Precio [USD]	4,900	31,100	20,000
Precio Total [USD]	8,100	36,520	32,000

Al analizar los datos recabados es de particular interés notar que las 3 soluciones estudiadas utilizan el mismo transceiver, el AD9361 de Analog Devices. Es por esto que sus características en RF son prácticamente las mismas:

**Ancho de banda:** 56 MHz      **Tasa de muestreo:** 61.44 Msps      **Canales TX/RX:** 2/2  
**Modo:** Full dúplex      **Potencia de Tx:** +10 dBm

La excepción es el rango de frecuencias del  $\mu$ SDR-C (fig. 5.14) el cual inicia en los 150 MHz, mientras que los rangos del E320 y el NanoCom SDR lo hacen en los 70 MHz. Respecto a la frecuencia máxima, todos son capaces de operar hasta los 6 GHz.



Figura 5.14 SDR integrado modelo  $\mu$ SDR-C del proveedor Space Micro [5\_47].

Respecto al dispositivo de procesamiento, los tres utilizan un Zynq 7000 de Xilinx. Este dispositivo es un SoC (System on Chip) que consta de dos elementos principales: un procesador de doble núcleo y un FPGA. Para las soluciones dadas por Ettus y Micro Space se utiliza un Zynq 7045 y 7020 respectivamente mientras que GomSpace emplea la serie 7030.

En la siguiente tabla 5.7 se muestra una comparativa entre ambos modelos.

Tabla 5.7 Características principales de los Soc de la familia Zynq 7000 de Xilinx.

SoC Zynq 7000 [5_48]			
Proveedor	Xilinx	Xilinx	Xilinx
Modelo	Zynq 7020	Zynq 7030	Zynq 7045
Procesador	Dual-core ARM Cortex-A9 MPCore		
Velocidad procesador	866 MHz	1 GHz	1 GHz
FPGA embebido	Artix-7	Kintex-7	Kintex-7
LUTs	53k	78k	218k
Celdas lógicas	85k	125k	350k
Bloques DSP	220	400	900
Precio [USD]	125	300	900

De la tabla, se entiende que la principal diferencia entre ambos modelos es su número de LUTs y de celdas lógicas lo cual impacta directamente en su capacidad de procesamiento. Por un lado, la serie 7020 cuenta con 53k LUTs, la serie 7030 con 78k mientras que la serie 7045 lo hace con 218k. De igual modo, la velocidad del procesador ARM, las celdas lógicas

y los bloques DSP varían dependiendo del modelo. De manera general, la serie 7030 tiene una capacidad de procesamiento 50% mayor a la 7020. La serie 7045 triplica la capacidad de la 7030 y cuadruplica la de la 7020.

Como se mencionó anteriormente, la estimación del número de LUTs para implementar el estándar DVB-S2 en un FPGA es de al menos 60k. De este modo, un número menor podría representar la no viabilidad del SDR para la misión planteada. Si se toma en cuenta este criterio queda claro que el Zynq 7020 queda descartado al no cumplir con los requerimientos de capacidad de procesamiento del sistema. Es por ello que el SDR integrado  $\mu$ SDR-C no puede ser considerado como una propuesta viable a menos que se logre reducir el requerimiento en la capacidad de procesamiento del FPGA por debajo de los 53k LUTs o se desee implementar una misión diferente que cumpla este último punto.

En cuanto a la compatibilidad con software los tres dispositivos son compatibles con Vivado por estar asociados al proveedor Xilinx. De manera particular el USRP E320 es compatible con GNU Radio, RFNoc y Vivado lo cual es de gran ventaja al momento de desarrollar e implementar aplicaciones relacionadas con las comunicaciones como es en este planteamiento de misión. Por su parte, GomSpace y Space Micro proporcionan únicamente un software propietario de Xilinx llamado SDK (Software Development Kit) el cual es compatible con Vivado.

Uno de los grandes inconvenientes de la plataforma de Ettus, mostrada en la figura 5.15, es su consumo de potencia máxima establecida en 30W, la cual presenta una gran diferencia respecto a GomSpace con sólo 3.3W y Space Micro con 8W. En definitiva, al ser un dispositivo para aplicaciones terrestres el E320 no ha sido diseñado para optimizar su consumo en potencia lo cual al momento de trasladarlo a un proyecto espacial resulta prácticamente prohibitivo. Es así como una conclusión importante respecto al SDR E320 se puede plantear: si se desea implementar en el espacio debe de modificarse su consumo de potencia y reducirse a menos de 10W. Al considerar los elementos que conforman al E320 y al NanoCom SDR se tiene que la única gran diferencia es el modelo del Zynq implementado en cada uno. Es decir que la posibilidad de lograr reducir el consumo de potencia de este dispositivo se muestra como alcanzable. En caso contrario, esta solución deberá de ser descartada pues no sería posible cumplir con los objetivos de la misión los cuales tienen como prioridad la observación terrestre y no la demostración del sistema de comunicaciones.



Figura 5.15. USRP E320 de Ettus Research [5\_45].

En relación al hecho de que el E320 sea considerado un producto COTS es inherente que, en principio, no tenga ningún tipo de cualificación para operar en ambientes espaciales, donde una de sus más grandes desventajas en comparación con GomSpace y Space Micros son los mecanismos de tolerancia a la radiación que posee, en este caso, ninguno. Por lo contrario los SDR de GomSpace y Space Micro han sido modificados, ya sea en software o en hardware, para tener cierta tolerancia a la radiación generada en la órbita LEO y garantizar así su correcto funcionamiento, y no sólo eso, sino que el NanoCom SDR ha sido probado en órbita ISS dentro del proyecto GomX-3 con resultados exitosos [5\_49]. Sin embargo, un número considerable de los COTS probados en órbitas bajas no son provistos de ninguna defensa contra la radiación por considerarse proyectos de una vida útil escasa, menor a los dos años, y por la cantidad de radiación presente en estas órbitas la cual permite, utilizando redundancia, la adecuada operación de las computadoras de abordo y demás electrónicos [5\_50].

Los últimos diferenciadores a analizar son las dimensiones, masa y precio de los SDR. Como con el análisis de los Front End, las dimensiones juegan un papel esencial en el diseño de todo el proyecto satelital. En el caso específico de los SDR cualquiera con dimensiones mayores, en alguno de sus lados, a los 10cm establecerá la imposibilidad de ser ubicado y distribuido perpendicular al eje Z lo cual aumentaría la complejidad en la integración final del satélite. En la figura 5.16 se muestra un esquema del proyecto GomX-3 de GomSpace el cual se utiliza como ejemplo de una distribución estándar, con placas paralelas al plano XY del CubeSat 3U.

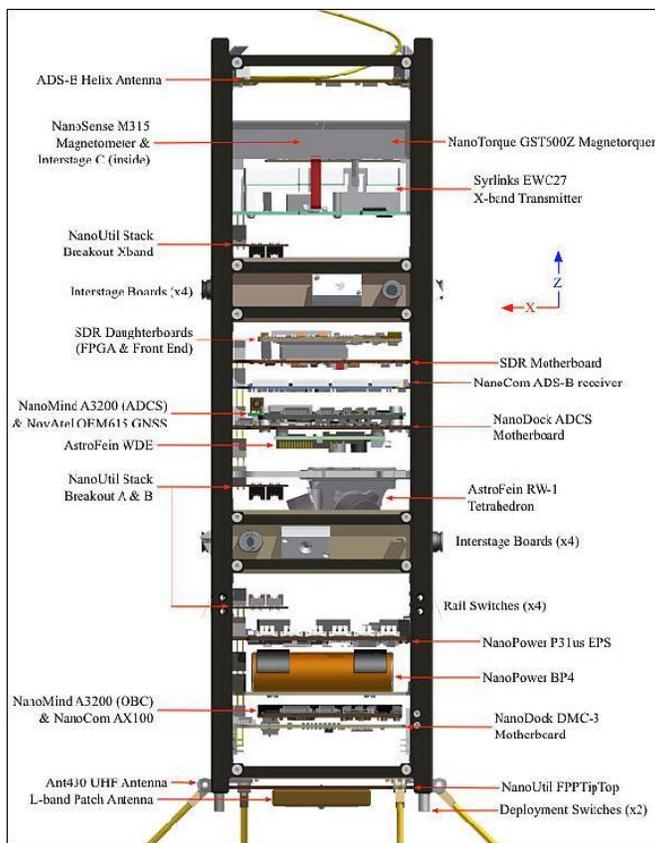


Figura 5.16 Esquema del proyecto Gom-X3 de GomSpace [5\_51].



Este inconveniente recae en el SDR E320 al tener uno de sus lados una longitud de 17.3 cm. En caso de implementarse dentro de un CubeSat 3U este dispositivo deberá ser colocado a lo largo del eje Z del satélite y el diseño y distribución de las placas y demás subsistemas dependerán de la ubicación del SDR. Para el caso del NanoCom SDR y el  $\mu$ SDR-C, ambos tienen dimensiones menores o iguales a los 10cm lo cual no afectaría la distribución horizontal mencionada.

Adicionalmente, mientras mayor sea el volumen del subsistema de comunicaciones menor será el espacio disponible para la carga útil o para algún otro subsistema que requiera utilizar mayor volumen de manera crítica. Redondeando se aprecia que el SDR que ocupa un mayor volumen es el E320 con 0.6U, le sigue el  $\mu$ SDR-C con 0.5U y el dispositivo con menor volumen es el NanoCom con 0.2U.

De manera contra intuitiva, el SDR que presenta menor masa es el E320 con 160 g. A este le sigue el NanoCom con 271g y finalmente el  $\mu$ SDR-C con 600 g. El bajo valor en la masa del E320 es explicado por la falta de recubrimiento o carcasa sobre el mismo, es decir, sólo se considera la PCB y sus electrónicos como se muestra en la figura 5.15. Las masas adicionales de los otros dos dispositivos son dadas por las cubiertas que estos poseen las cuales presentan tolerancia a la radiación. Es importante considerar un aumento en la masa del E320 debido a algún recubrimiento adicionado contra la radiación. Importante volver a mencionar que la masa de cada dispositivo impacta directamente el costo de lanzamiento del satélite considerando el precio de 20,000 USD/kg.

Si se revisa el precio de adquisición de cada dispositivo se aprecia una enorme diferencia entre el E320, un dispositivo COTS, y el NanoCom SDR y el  $\mu$ SDR-C, ambos dispositivos con cualificación espacial. El costo del E320 es de 4,900 USD, por otro lado el del NanoCom es de 31,100 USD y el SDR-C cuesta 20,000 USD. Esto representa una diferencia del 600% y del 400% respecto al E320.

Si ahora se le adiciona el costo de lanzamiento generado por la masa de cada dispositivo se tiene un precio total de 8100, 36520 y 32000 USD para el E320, NanoCom y  $\mu$ SDR-C, respectivamente. Aun agregando un recubrimiento al E320 con una masa estimada de 300g su costo total sería de 14,100 USD, un valor menor al presentado por los otros dos dispositivos

Llegado a este punto, se aprecian dos posibilidades claras de selección dependiendo, básicamente, de los recursos con los que cuenta el proyecto. Por un lado, si lo que se busca es reducir costos al máximo y aumentar la capacidad de procesamiento, la opción del USRP E320 es la mejor. Por otro lado, si la prioridad es garantizar el buen funcionamiento del subsistema y minimizar los riesgos al máximo, al igual que las dimensiones y la potencia consumida, la mejor opción es el NanoCom SDR de GomSpace.

La opción del  $\mu$ SDR-C queda descartada por su baja capacidad de procesamiento, su alta masa y elevado costo final.

Tomando como referencia la tabla 5.6 y agregando las categorías por colores en la siguiente tabla se logra visualizar de mejor modo los puntos fuertes y débiles de cada solución respecto a los objetivos de la misión planteada. Los parámetros de RF se mantienen sin resaltar pues todos presentan las mismas características.



Tabla 5.8. Características principales de los SDR integrados categorizados.

	Ettus E320 [5_45]	NanoCom SDR [5_46]	μSDR-C [5_47]
Proveedor	Ettus Research	GomSpace	Space Micro
Frecuencia	70MHz – 6GHz	70MHz – 6GHz	150 MHz – 6 GHz
Ancho de banda	56MHz	56MHz	56 MHz
ADC/DAC	12 bits	12 bits	12 bits
Tasa de muestreo	61.44Msps	61.44Msps	61.44 Msps
Canales TX/RX	2/2	2/2	2/2
Duplex	Full	Full	Full
Interfaz	USB 3.0	USB 2.0	--
Transceiver	AD9361	AD9361	AD9361
Potencia de TX	+10 dBm	+10 dBm	+10 dBm
FPGA Baseband	Xilinx Zynq 7045	Xilinx Zynq 7030	Xilinx Zynq 7020
LUTs	218k	78k	53k
Software	UHD, GNU Radio, RFNoc, SDK, Vivado	SDK, Vivado	SDK, Vivado
Consumo de potencia	10 V, 3A (30W)	5.5V, 600mA (3.3W)	8W
Herencia de Vuelo	No	Sí (GomX-3@órbita ISS)	No
Tolerancia Radiación	No	20 kRad	30 krad SEL: <70 Mev/mg/cm <sup>2</sup> SEU: <1E-3/día SEFI: 0
Dimensiones	17.3 x 10 x 3.6 cm	9.2 x 8.9 x 2.4 cm	10 x 10 x 5 cm
Volumen	622 cm <sup>3</sup>	196 cm <sup>3</sup>	500 cm <sup>3</sup>
Masa	160 g	271 g	600 g
Precio [USD]	4,900	31,100	20,000
Precio Total [USD]	8,100	36,520	32,000

De esta forma, nuevamente se hace énfasis en las características prohibitivas del μSDR-C: procesamiento, masa y costo final. Para el Ettus E320 este punto de restricción se encuentra en su elevado consumo de potencia y su volumen.

Sin duda, el NanoCom de GomSpace, mostrado en la figura 5.17, es el SDR más optimizado en términos de potencia, dimensión y volumen, además de tener herencia de vuelo. Sin embargo, su costo de adquisición puede llegar a ser prohibitivo sobre todo tratándose de proyectos universitarios pero es el mejor dispositivo a implementar en un proyecto satelital.



Figura 5.17 Nanocom SDR de GomSpace [5\_46].

En la figura 5.18 se muestra un gráfico radial con el resumen de las características analizadas de los 3 dispositivos. Mientras mayor sea el área abarcada por cada parámetro mejor es el desempeño del dispositivo.

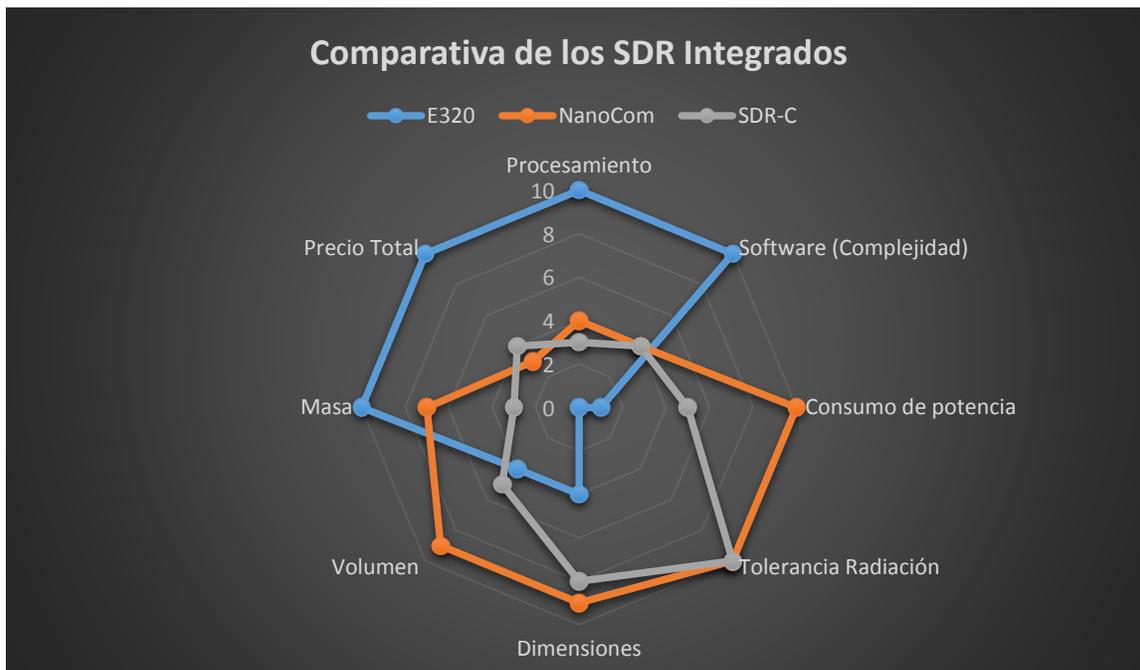


Figura 5.18. Comparativa radial de los SDR Integrados.

Como conclusión de este subcapítulo, se propone la omisión del  $\mu$ SDR-C y la posible utilización del E320 y el NanoCom SDR, dependiendo de los recursos y prioridades del proyecto. Un proyecto con una solvencia importante bien podría costear el Nanocom de GomSpace con lo cual aumentaría la probabilidad de éxito del proyecto al ser un dispositivo diseño y fabricado para operar en un ambiente espacial. Si los recursos económicos son restringidos o se desea utilizar GNU Radio y RFNoC como herramientas de desarrollo, lo cual reduciría enormemente la complejidad de desarrollo, entonces la opción del E320 es la más factible, siempre y cuando se encuentre una solución para reducir el consumo de potencia considerablemente a un valor menor de 10W.

A continuación se muestran los diagramas de bloques de ambos SDR seleccionados con el fin de entender de mejor modo la integración de los elementos que los componen. Ver fig. 5.19 y 5.20.

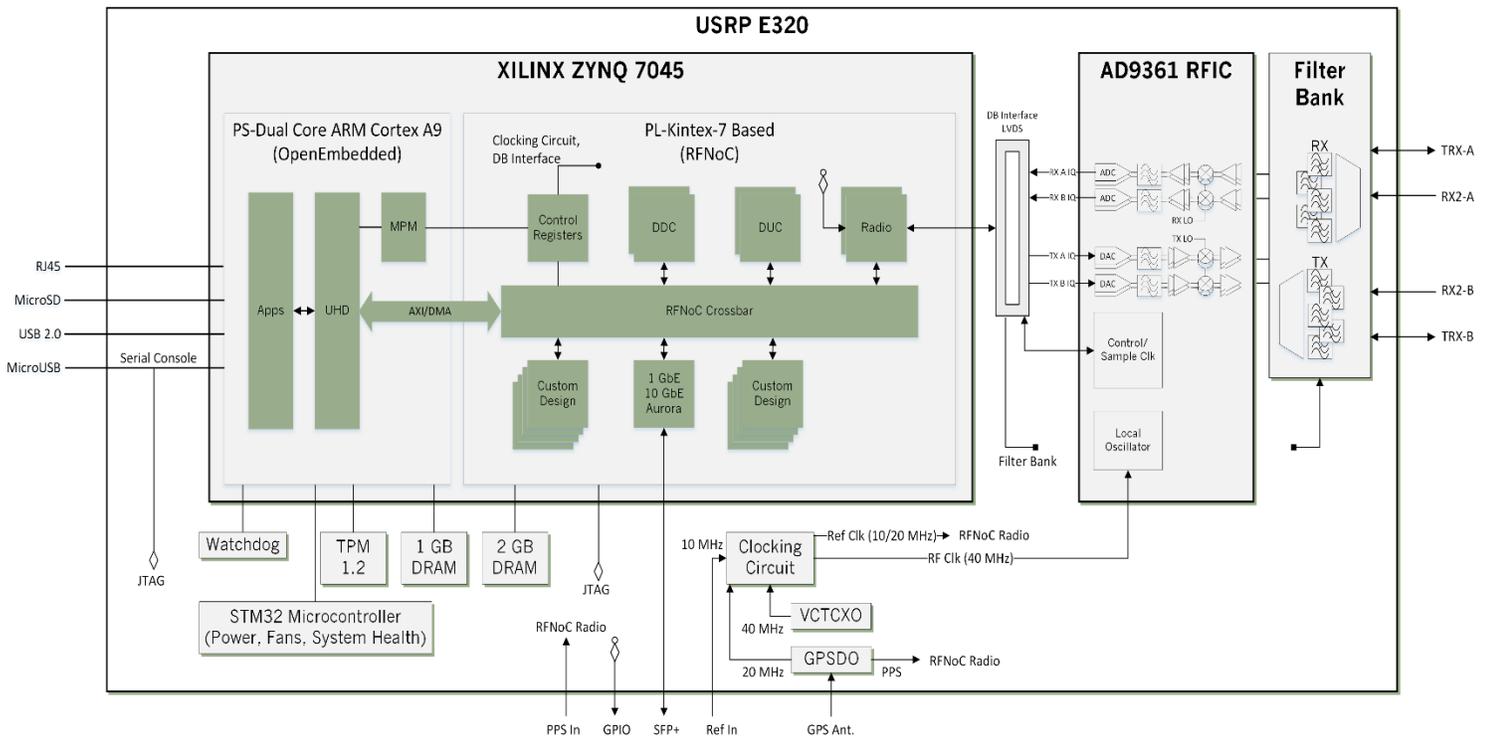


Figura 5.19 Diagrama de bloques del USRP E320 de Ettus Research [5\_45].

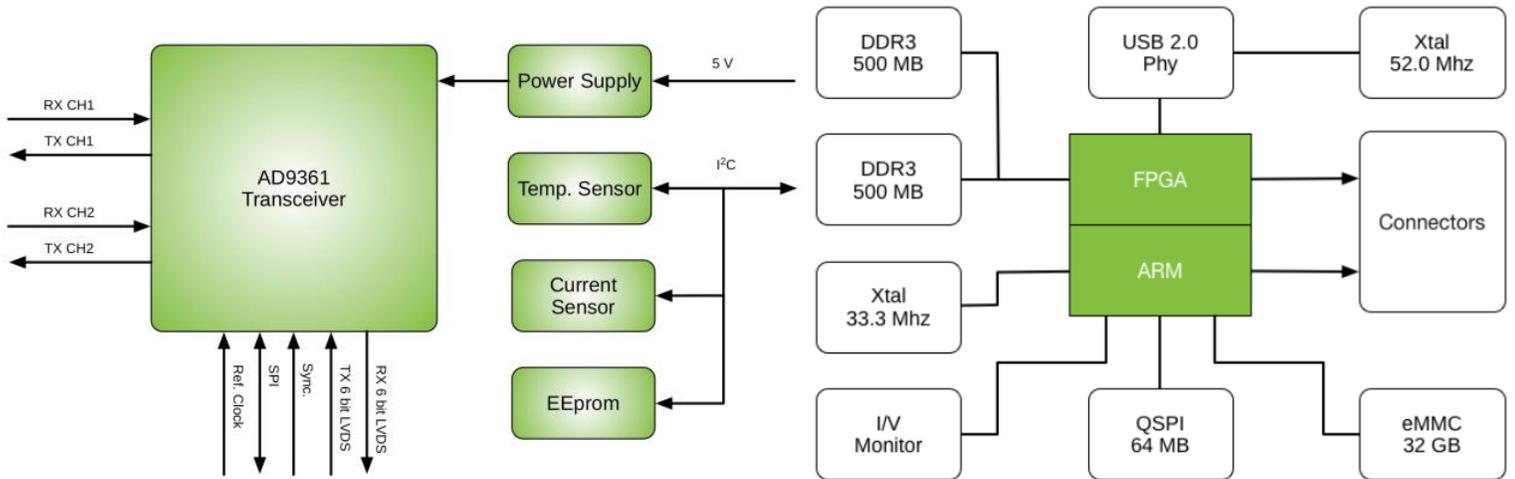


Figura 5.20 Diagrama de bloques del NanoCom SDR de GomSpace [5\_46].

En la siguiente sección se continuará con el análisis de las propuestas seleccionadas hasta el momento con el objetivo de brindar una propuesta final como solución a la implementación de un SDR para la misión satelital de observación terrestre.



#### 5.4. DISEÑO DEL SDR PROPUESTO

Como se menciona al inicio del capítulo se exploran dos posibilidades para basar el diseño del SDR:

- a) Dispositivos independientes (FPGA + Front End COTS)
- b) Solución embebida (SDR Integrado)

Ahora se procede a realizar un último análisis para finalmente definir qué solución es la más conveniente al momento de desear implementarla en la misión CubeSat propuesta. Se procederá, de igual modo, con un análisis de complejidad, compatibilidad con software, potencia, dimensiones y costo, y se contrastaran las ventajas y desventajas de cada solución. También se mostrarán los diseños asistidos por computadora (CAD) de cada propuesta para así constatar de manera visual las implicaciones de optar por una u otra solución.

Del análisis realizado en las secciones previas se concluye que los dispositivos que presentan mayores ventajas son los presentados en la tabla 5.9.

**Tabla 5.9 Elementos seleccionados para conformar el SDR.**

Elemento	Proveedor	Modelo
FPGA	Altera	Cyclone IV EP4CE115
FPGA	Xilinx	Kintex 7 XC7K325T
Front End	Lime Microsystems	LimeSDR
Front End	Great Scott Gadgets	HackRF
SDR Integrado	Ettus Research	USRP E320
SDR Integrado	GomSpace	NanoCom SDR

Se ha decidido agregar el dispositivo HackRF a la lista porque, si bien no presenta las mejores características técnicas y físicas, es el dispositivo al cual se tiene acceso durante el momento de desarrollar este proyecto por lo cual resulta lógico considerarlo para el último análisis planteado.

Considerando esto se establecen 4 configuraciones de SDR diferentes de acuerdo a la tabla 5.10.

**Tabla 5.10 Configuraciones SDR propuestas.**

Configuración	Tipo	Elementos
1	Dispositivos Independientes	FPGA+ HackRF
2	Dispositivos Independientes	FPGA + LimeSDR
3	Solución Embebida	Ettus E320
4	Solución Embebida	NanoCom SDR

A partir de la selección de las configuraciones anteriores se procede a realizar un desglose de los parámetros más representativos de cada una de estas, los cuales serán empleados para establecer la comparativa final entre las 4 soluciones propuestas. De este modo, se muestran las tablas 5.11, 5.12, 5.13 y 5.14 con los valores de las configuraciones 1, 2, 3 y 4 respectivamente.



Programa de Maestría y Doctorado en Ingeniería

Tabla 5.11 Diseño de SDR | Configuración 1.

Configuración 1   Frecuencia: 70 MHz – 6GHz   BW: 56MHz   1 Tx / 1 Rx				
Elemento / Parámetro	FPGA (PCB)	Front End	Total	Margen
Modelo	Cyclone IV	HackRF	--	--
Dimensiones [cm]	9.4 x 9.4 x 2	12 x 7.5 x 2	<b>14 cm (eje Z)</b>	+20 cm (eje Z)
Volumen [cm <sup>3</sup> ]	36	180	<b>216</b>	+3,184 cm <sup>3</sup>
Posición	Horizontal	Vertical	<b>Mixta</b>	--
Masa [g]	10	180	<b>190</b>	+3810 g
Consumo de Potencia [W]	1.5	0.4	<b>1.9</b>	+8.1W
Procesamiento (LUTs)	114k	--	<b>114k</b>	+54k
Programación	VHDL/Verilog	--	--	
Herramienta de diseño	Quartus	--	--	
Herramienta de desarrollo	ModelSim	--	--	
Complejidad	Alta	--	<b>Alta</b>	
Alto	Alto	--	<b>Alto</b>	
Herencia Vuelo	No	No	<b>No</b>	
Tolerancia Radiación	No	No	<b>No</b>	
Precio licencia [USD]	5,000/año	--	--	
Precio adquisición [USD]	600	300	<b>900</b>	
Precio Lanzamiento [USD]	200	3600	<b>3800</b>	
Precio TOTAL [USD]	5,800	3,900	<b>9,700</b>	

Tabla 5.12 Diseño de SDR | Configuración 2.

Configuración 2   Frecuencia: 100 kHz – 3.8GHz   BW: 56MHz   2Tx / 2Rx				
Elemento / Parámetro	FPGA (PCB)	Front End	Total	Margen
Modelo	Cyclone IV	LimeSDR	--	--
Dimensiones [cm]	9.4 x 9.4 x 2	10 x 6 x 1	<b>3 (Eje Z)</b>	+31 cm (Eje Z)
Volumen [cm <sup>3</sup> ]	36	60	<b>96</b>	+3,304 cm <sup>3</sup>
Posición	Horizontal	Horizontal	<b>Horizontal</b>	--
Masa [g]	10	30	<b>40</b>	+3960 g
Consumo de Potencia [W]	1.5	0.6	<b>2.1</b>	+7.9 W
Procesamiento (LUTs)	114k	40k	<b>154k</b>	+94k
Programación	VHDL/Verilog	VHDL/Verilog	--	
Herramienta de diseño	Quartus	Quartus	--	
Herramienta de desarrollo	ModelSim	ModelSim	--	
Complejidad	Alta	--	<b>Alta</b>	
Tiempo Desarrollo	Alto	--	<b>Alto</b>	
Herencia Vuelo	No	Sí	<b>Front End</b>	
Tolerancia Radiación	No	No	<b>No</b>	
Precio licencia [USD]	5,000/año	Open Source	<b>5,000</b>	
Precio adquisición [USD]	600	300	<b>900</b>	
Precio Lanzamiento [USD]	200	600	<b>800</b>	
Precio TOTAL [USD]	5,800	900	<b>6,700</b>	



Programa de Maestría y Doctorado en Ingeniería

Tabla 5.13 Diseño de SDR | Configuración 3.

Configuración 3   Frecuencia: 70 MHz – 6GHz   BW: 56MHz   2Tx / 2Rx		
Elemento / Parámetro	SDR Integrado	Margen
Modelo	USRP E320	--
Dimensiones [cm]	17.3 x 10 x 3.6	+16.7 cm (Eje Z)
Volumen [cm <sup>3</sup> ]	623	+2,777 cm <sup>3</sup>
Posición	Vertical	--
Masa [g]	160	+3,840 g
Consumo de Potencia [W]	30W	-20 W
Procesamiento (LUTs)	218k	+158k
Programación	Phyton, VHDL	
Herramienta de diseño	GNU Radio, RFNoc	
Herramienta de desarrollo	Vivado	
Complejidad	Media	
Herencia Vuelo	No	
Tiempo Desarrollo	Medio	
Tolerancia Radiación	No	
Precio licencia [USD]	3,000	
Precio adquisición [USD]	4,900	
Precio Lanzamiento [USD]	3,200	
Precio TOTAL [USD]	11,100	

Tabla 5.14 Diseño de SDR | Configuración 4.

Configuración 4   Frecuencia: 70 MHz – 6GHz   BW: 56MHz   2Tx / 2Rx		
Elemento / Parámetro	SDR Integrado	Margen
Modelo	NanoCom SDR	--
Dimensiones [cm]	9.2 x 8.9 x 2.4	+31.6 cm (Eje Z)
Volumen [cm <sup>3</sup> ]	196	3,204 cm <sup>3</sup>
Posición	Horizontal	Horizontal
Masa [g]	271	3,729 g
Consumo de Potencia [W]	3.3W	+6.7W
Procesamiento (LUTs)	78k	+18k
Programación	VHDL	
Herramienta de diseño	SDK	
Herramienta de desarrollo	Vivado	
Complejidad	Alta	
Tiempo Desarrollo	Bajo	
Herencia Vuelo	Sí	
Tolerancia Radiación	Sí (20kRad)	
Precio licencia [USD]	3,000	
Precio adquisición [USD]	31,100	
Precio Lanzamiento [USD]	5,420	
Precio TOTAL [USD]	39,520	



Se ha prestado especial atención a los valores de dimensiones, masa, posición, capacidad de procesamiento, complejidad de programación, consumo de potencia, tolerancia a la radiación y precio, dado que estos son los que influyen de manera directa en el desempeño y viabilidad del proyecto nanosatelital.

Con el fin de apreciar de mejor forma las dimensiones y volumen de cada uno de los módulos se muestra la figura 5.21 en la cual se tienen de izquierda a derecha el HackRF (azul fuerte), LimeSDR (rojo), Ettus E320 (azul claro) y el NanoCom SDR (amarillo) de GomSpace.

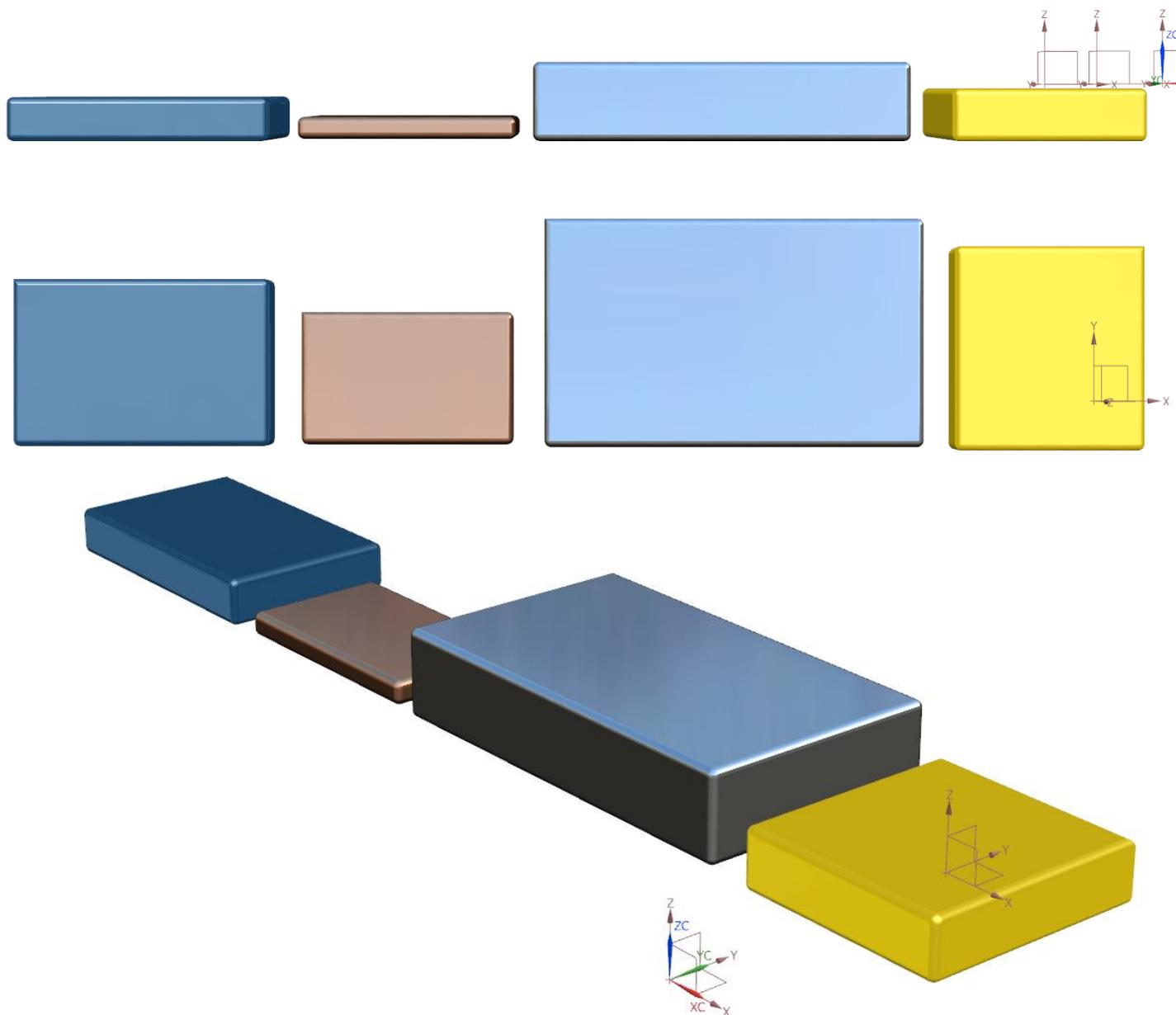


Figura 5.21 Comparativa en tamaño del HackRF, LimeSDR, E320 y NanoCom SDR (Elaboración propia).



En la figura 5.22, se muestra una comparativa entre los parámetros de las 4 configuraciones propuestas. De los valores relacionados con el precio, dimensión, volumen, masa, consumo de potencia y complejidad, se entiende que mientras mayor es su valor menor es su factibilidad para ser empleado en un proyecto CubeSat. Por otro lado, esta relación se invierte en los parámetros referentes al procesamiento y la tolerancia a la radiación, dado que mientras mayor es el valor, más convenientes resultan.

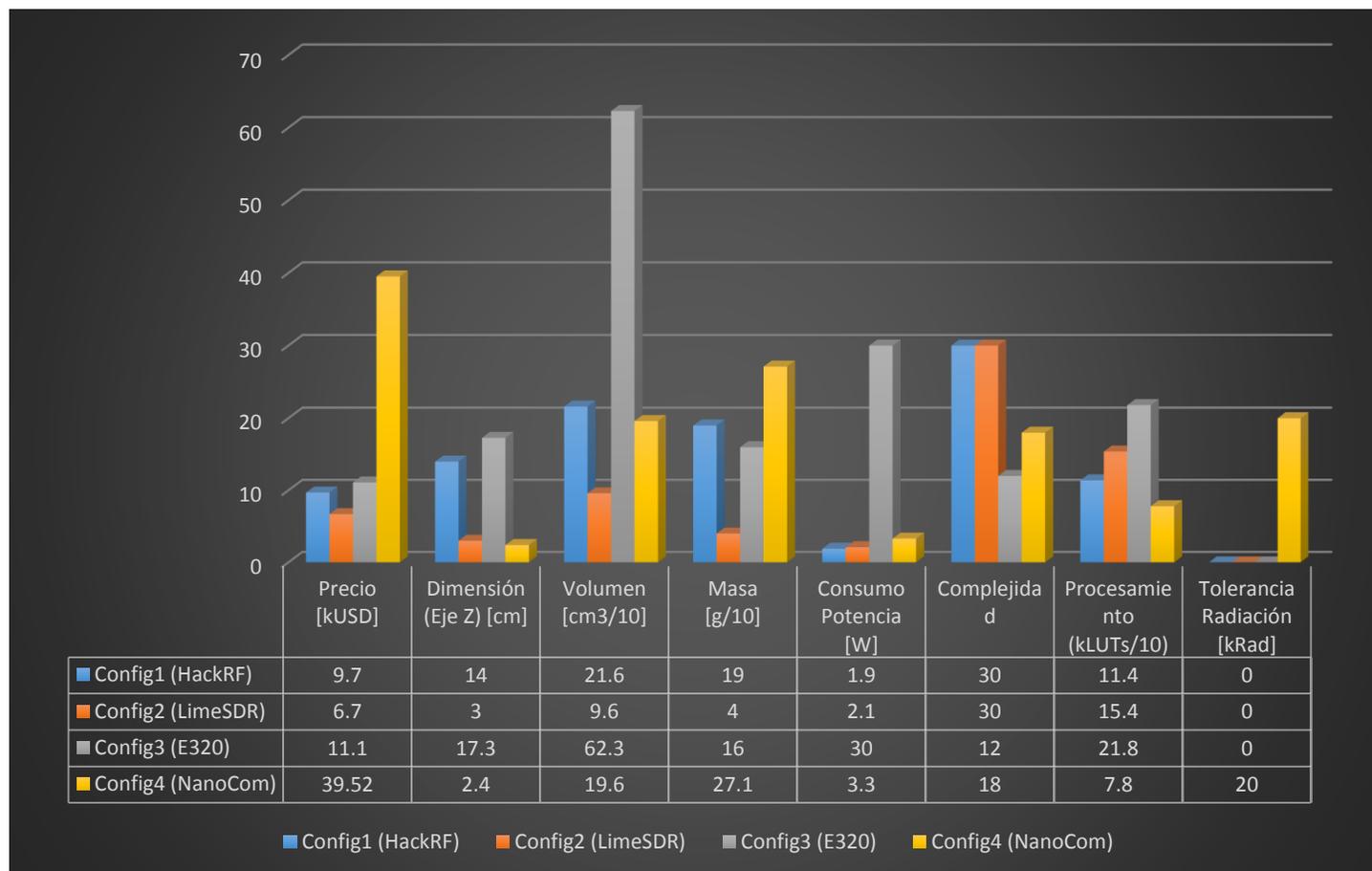
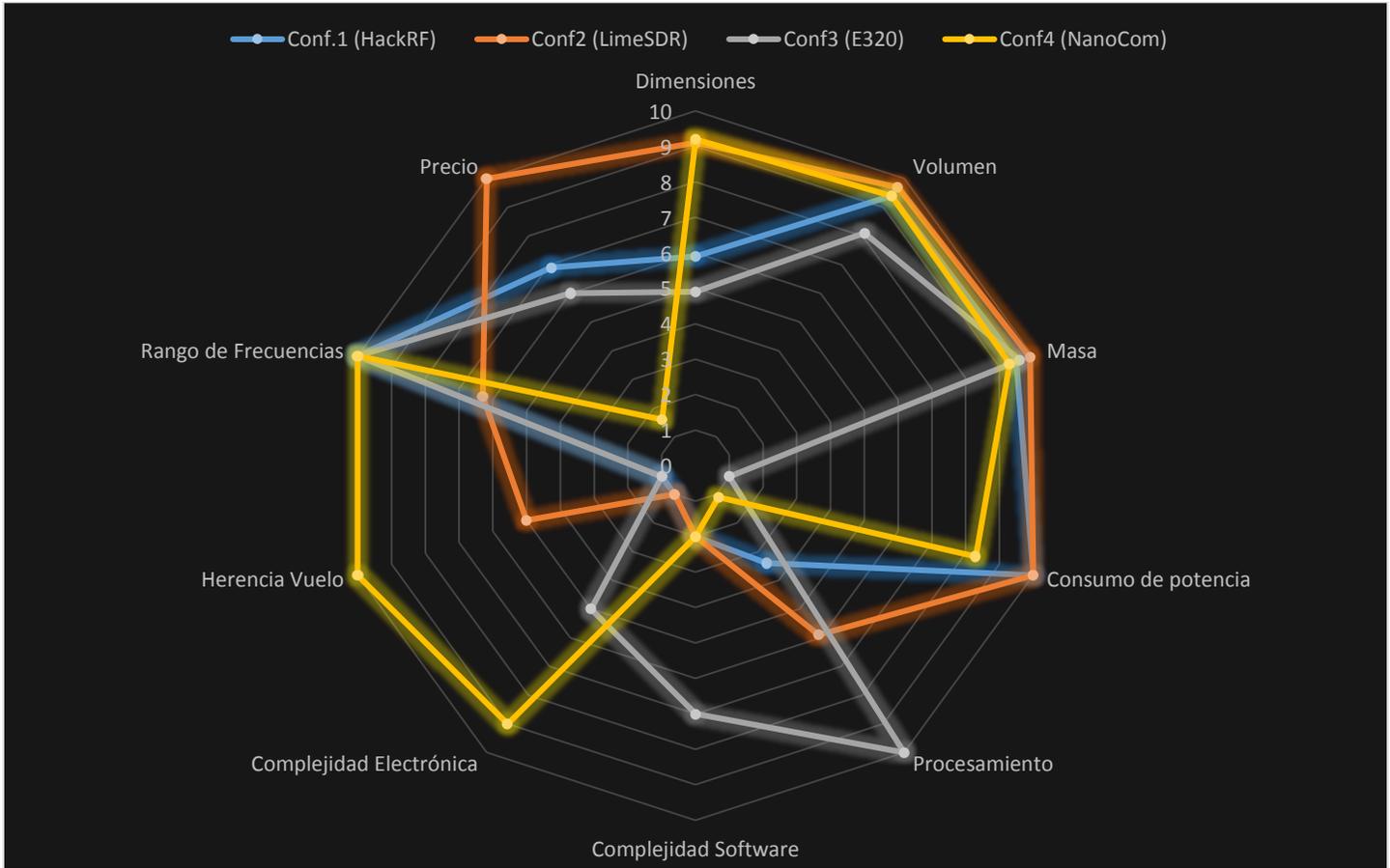


Figura 5.22 Gráfica comparativa de las 4 configuraciones de SDR propuestas.

De esta forma se muestra otro resumen de las ventajas y desventajas de estas configuraciones en la siguiente gráfica radial en la cual se han adaptado los datos para que, en todos los casos, una mayor área signifique una mayor ventaja y factibilidad. Ver fig. 5.23.



**Figura 5.23 Gráfica radial comparativa de las configuraciones de SDR propuestas.**

Es así como se denota que las dos configuraciones que presentan más ventajas son la número 2 y 4, asociadas al LimeSDR y al NanoCom SDR, respectivamente. Es decir, una solución basada en dispositivos COTS independientes y otra en un sistema integrado de categoría espacial.

Por un lado, las mayores ventajas de la configuración basada en el LimeSDR son su bajo precio, dimensiones, volumen, masa, consumo de potencia y alta capacidad de procesamiento, mientras que sus mayores desventajas son su nula tolerancia a la radiación, su menor rango de frecuencias y la gran complejidad electrónica y de software que requiere para ser desarrollado al tratarse de un FPGA y un Front End independientes.

En cuanto a la solución de GomSpace sus principales ventajas son el rango de frecuencia, su herencia de vuelo (tolerancia a la radiación), su optimizada complejidad electrónica al tratarse de un sistema integrado, adicional a los valores estándar de dimensiones, volumen, masa y consumo de potencia propios de un sistema diseñado para un proyecto CubeSat. En cuanto a sus desventajas es posible mencionar su justa capacidad de procesamiento, alta complejidad de software (debida al desarrollo en VHDL) y elevado precio.



Las soluciones basadas en el HackRF y el Ettus E320 son en principio descartadas por sus grandes dimensiones, su alto precio y alto consumo de potencia.

En otro orden de ideas, con el objetivo de tener una mejor comprensión del diseño conceptual del CubeSat y los SDR integrados en el mismo, se han desarrollado diferentes imágenes CAD de las diferentes configuraciones propuestas en la versión para estudiantes del software Solid Edge 3D CAD de Siemens. De este modo, se tendrá mejor idea de la disposición y espacio utilizado por el SDR, y el espacio disponible para el resto de los subsistemas. Para cada una de estas configuraciones de SDR se propone añadir los siguientes elementos sólo como referencia y para completar conceptualmente el subsistema de comunicaciones del nanosatélite:

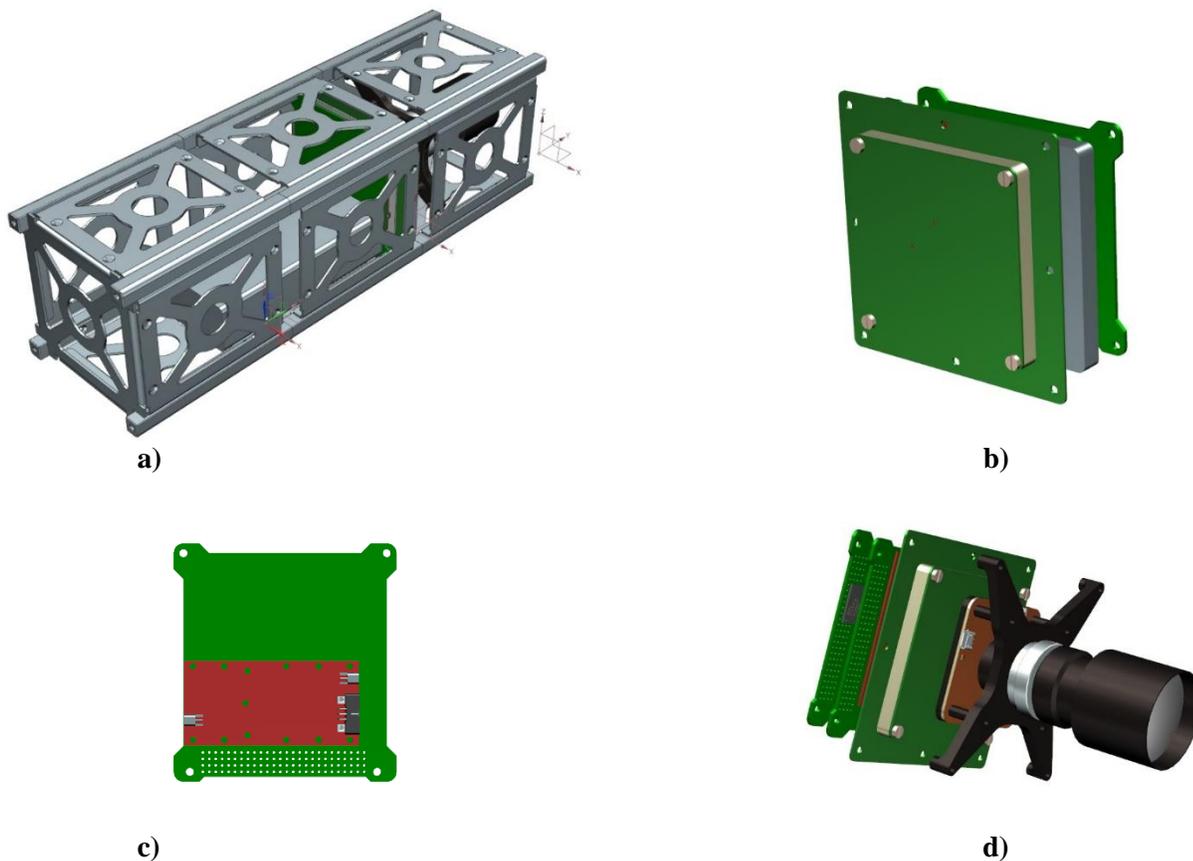
- Estructura CubeSat 3U
- Amplificador de potencia
- Antena
- Cámara

De este modo, cada propuesta consta de una plataforma base conformada por estructura, amplificador de potencia, antena y cámara. Para los elementos que conforman la plataforma base no se realiza ningún análisis pues el enfoque recae en las configuraciones de SDR propuestas. Sin embargo, se propone el uso de los siguientes elementos mostrados en la tabla 5.15 para formar parte de la plataforma base.

**Tabla 5.15 Elementos propuestos de la plataforma base.**

Elemento	Proveedor	Modelo	Características
<b>Estructura CubeSat [5_52]</b>	Pumpkin Inc	3U	34 x 10 x 10 cm 474g 3,100 USD
<b>Amplifier [5_53]</b>	Triad RF Systems	TA1109	6.85 x 3.3 x 0.83 cm 14g Frecuencia: 2 – 2.5 GHz Potencia Salida: 4W Ganancia: 25 dB Voltaje: 12 V Corriente: 0.8 A
<b>Antena [5_54]</b>	GomSpace	Nanocom Ant2000	9.8 x 9.8 x 2 cm 110g 2025 – 2290 MHz Ganancia: 8dBi Polarización circular Amplificador de potencia 2W incluido Consumo: 10W
<b>Cámara [5_55]</b>	GomSpace	Nanocam C1U	9.6 x 9 x 5 cm 277g Sensor CMOS 2048 x 1536 pixeles RGGB 10 bits JPEG, BMP Lentes de 70mm (30m/pix)

La imagen de estos elementos son mostrados en la figura 5.24.



**Figura 5.24 a) Estructura CubeSat 3U b) Antena parche c) Amplificador de potencia d) Cámara.**

A continuación es posible visualizar los diferentes CAD generados para cada una de las configuraciones de SDR propuestas. Para la estructura, cámara, antena, FPGA y placas se han empleado los siguientes CAD: [5\_56], [5\_57], [5\_58] y [5\_59].

Se decide mostrar diferentes perspectivas para tener una mejor visualización de las dimensiones y distribución que conlleva cada una de ellas. Ver fig. 5.25, 5.26, 5.27 y 5.28.

Configuración 1: FPGA + HackRF

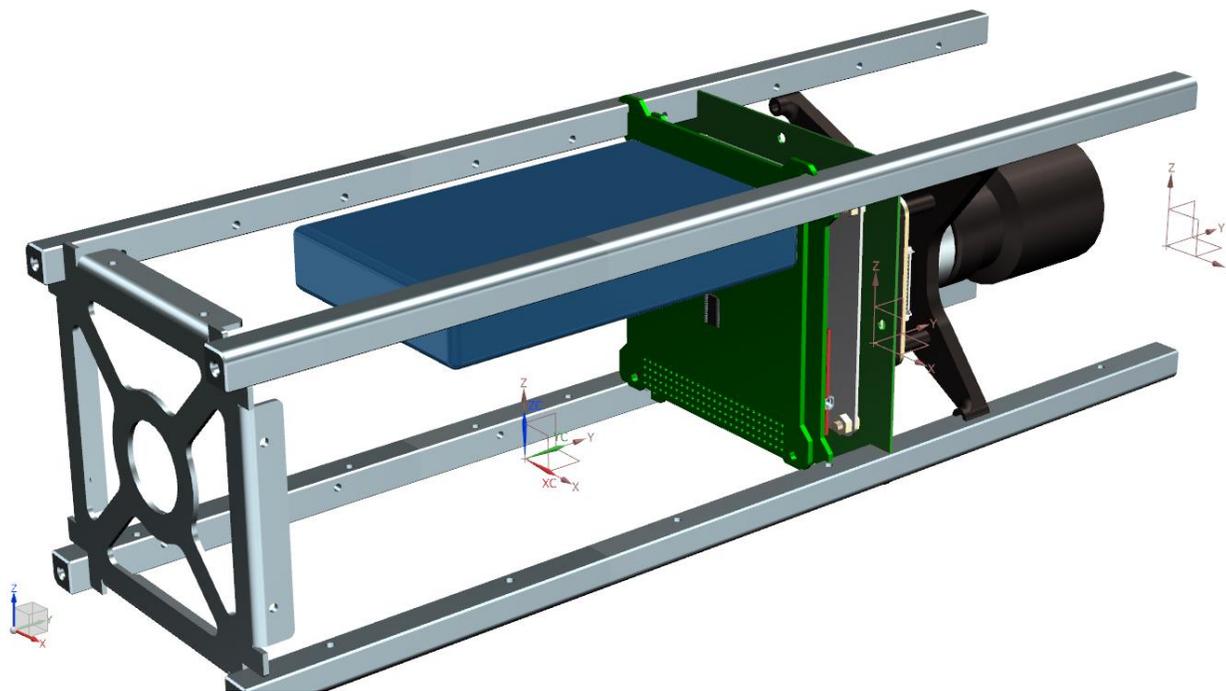


Figura 5.25 SDR propuesto | Configuración 1: FPGA + HackRF.

Configuración 2: LimeSDR

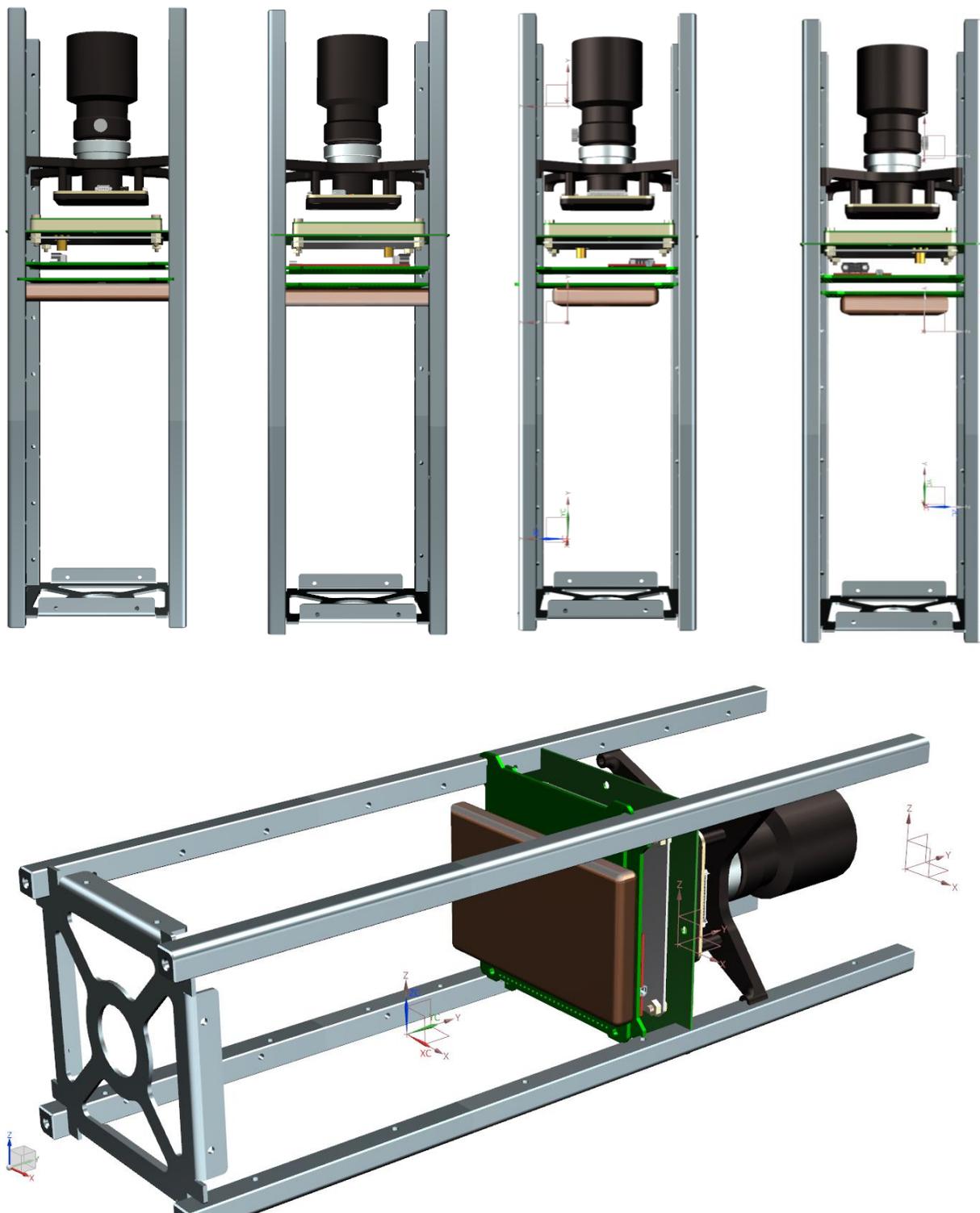


Figura 5.26 SDR propuesto | Configuración 2: FPGA + LimeSDR.

Configuración 3: FPGA + Ettus E320

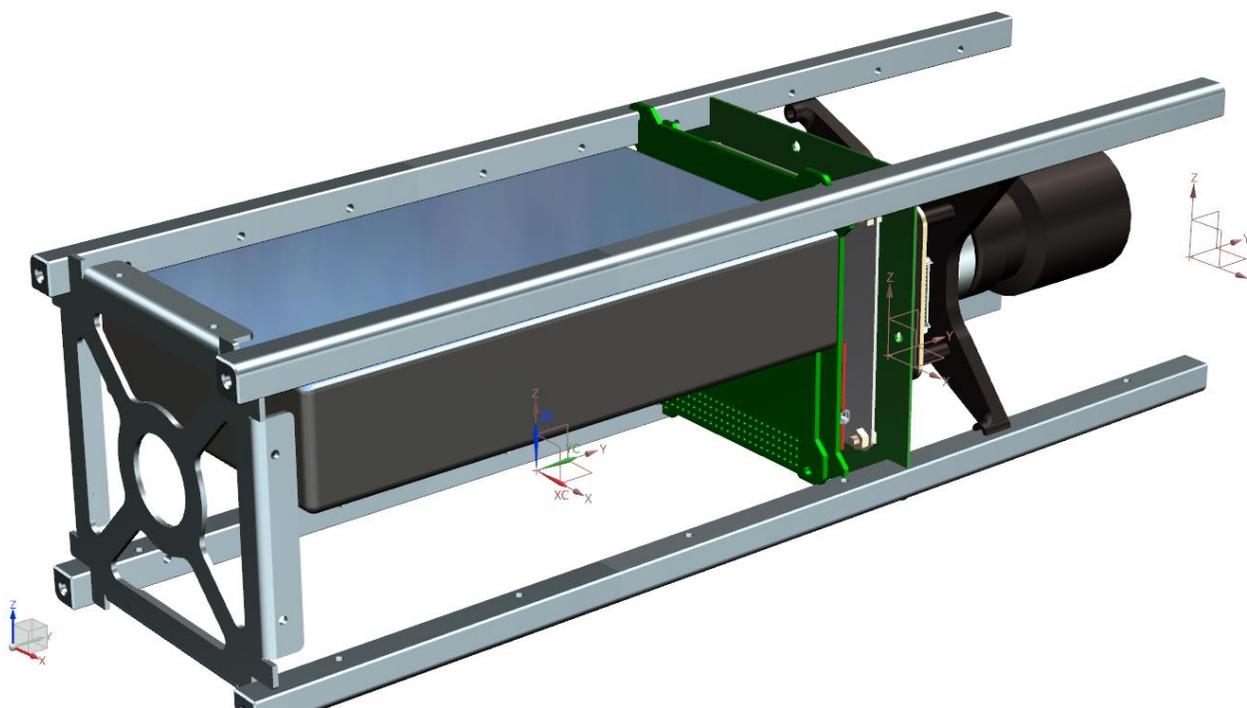


Figura 5.27 SDR propuesto | Configuración 3: Ettus E320.

Configuración 4: NanoCom SDR

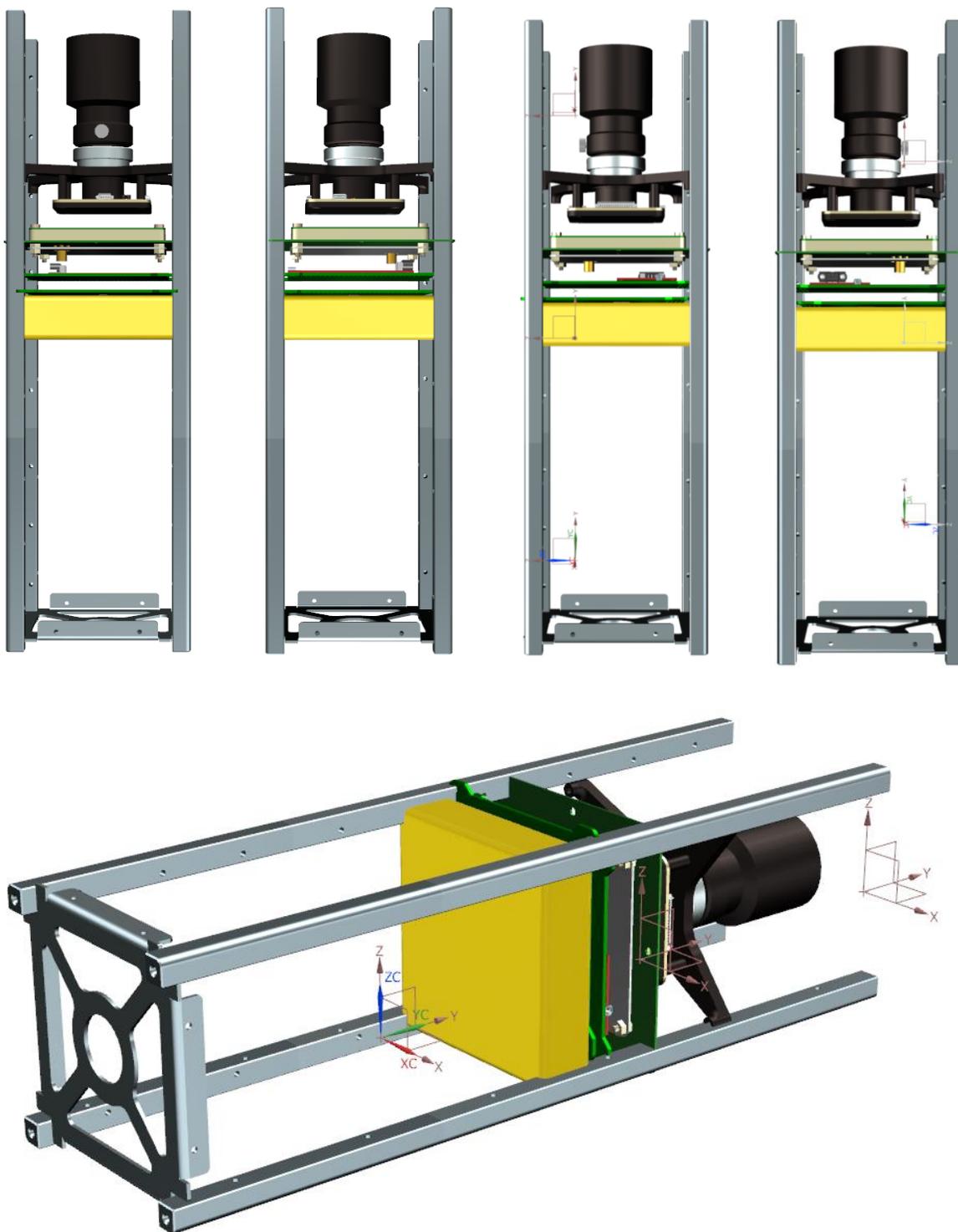


Figura 5.28 SDR propuesto | Configuración 4: NanoCom SDR.

Para tener una comparativa directa de las cuatro configuraciones implementadas dentro de la estructura CubeSat de 3U se retomaron algunas de las perspectivas anteriores y se muestran en la figura 5.29, de izquierda a derecha, el HackRF, LimeSDR, E320 y NanoCom SDR.

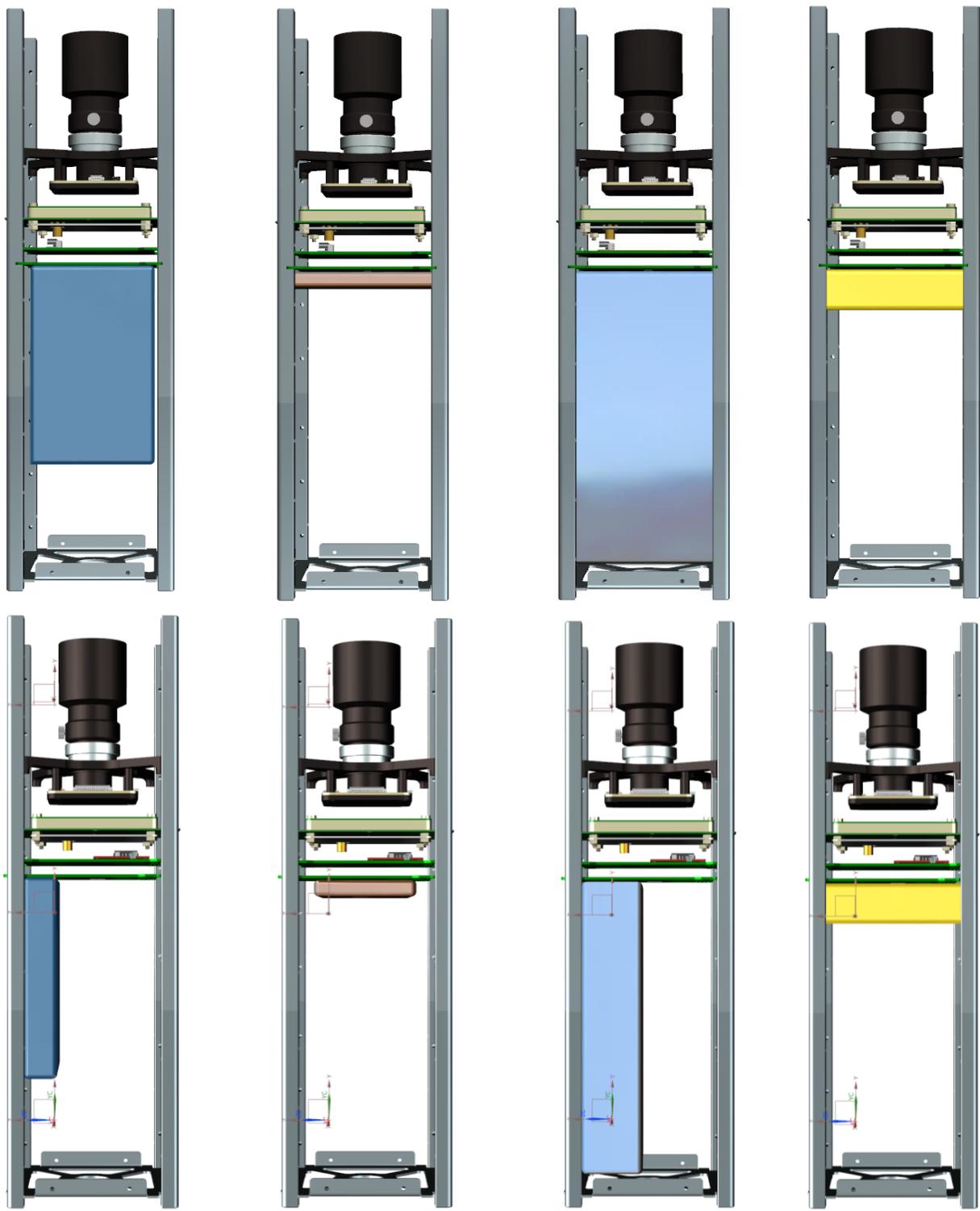


Figura 5.29. Perspectivas frontal y lateral de las configuraciones propuestas.



Con lo cual se reitera el análisis realizado previamente referente a las dimensiones de cada una de las configuraciones. El hecho de que los dispositivos HackRf y Ettus E320 tengan uno de sus lados con un valor mayor a los 10 cm, fuerza la distribución vertical de los mismos impactando directamente en la optimización interna del espacio destinado para el resto de los subsistemas.

El efecto contrario se apreció con el LimeSDR y el NanoCom, donde su distribución en horizontal hace posible la optimización de los espacios.

Considerando todo lo anterior se pueden resumir en la figura 5.30 las ventajas y desventajas de estas dos últimas configuraciones.

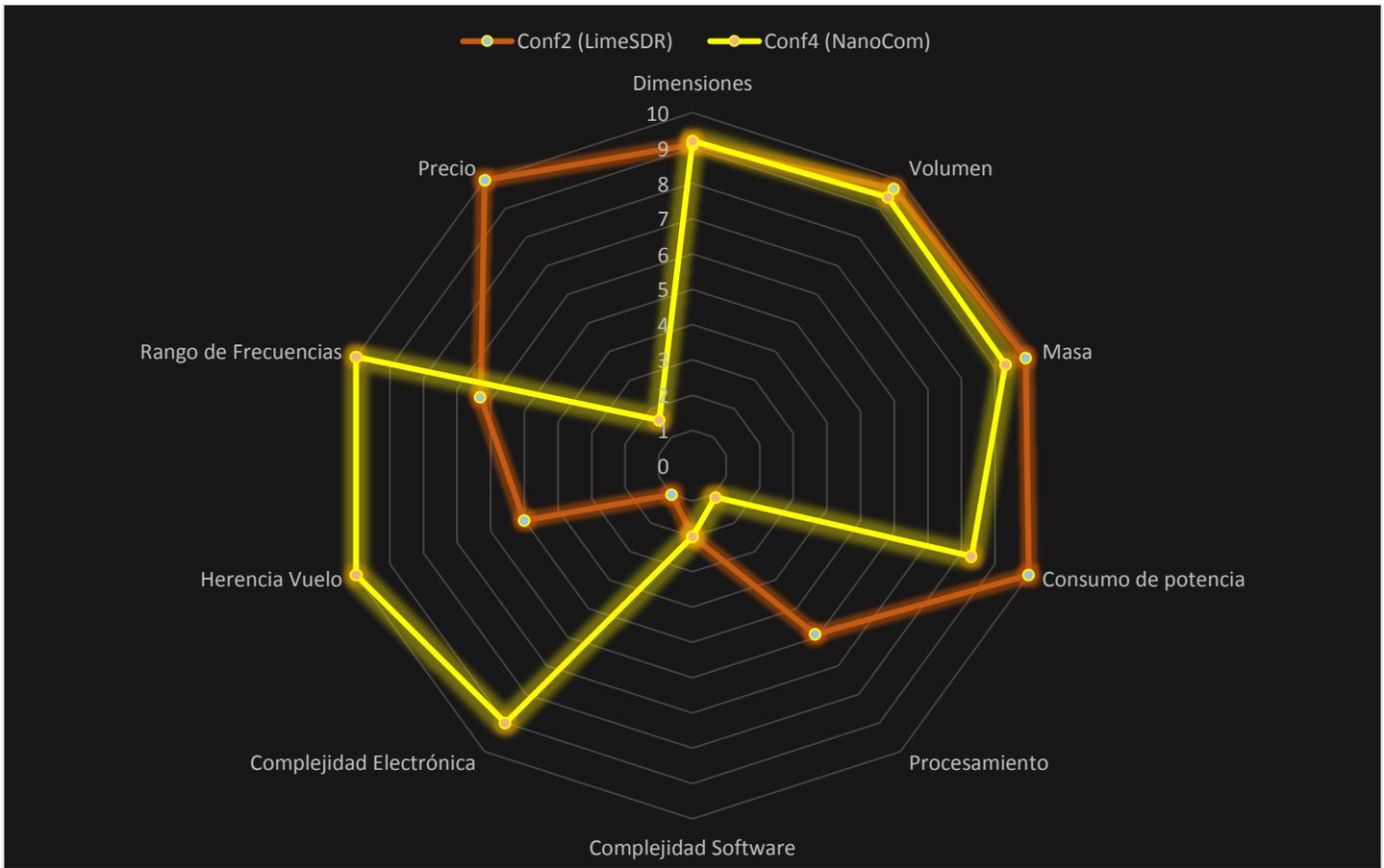


Figura 5.30 Gráfica radial comparativa de los SDR basados en el LimeSDR y NanoCom.

Para concluir el análisis, se decide optar por la configuración 2, basada en el LimeSDR, como la propuesta final a implementar. Esta decisión se hace considerando dos puntos principales:

- **Costo final**

El precio final (adquisición + licenciamiento + lanzamiento) de las propuestas es de 6,700 USD para el LimeSDR y de 39,520 USD para el NanoCom SDR. Esto es una diferencia de casi el 600% entre ambas soluciones.



Dado que el proyecto es, en principio, de índole académica, la disponibilidad de recursos económicos es una de las principales limitantes a enfrentar. Por esta razón una solución de un costo tan elevado como la dada por GomSpace se hace prácticamente inviable.

- **Procesamiento**

Si bien la capacidad de procesamiento del NanoCom SDR es de 78k LUTs, este valor está justo por encima del valor mínimo requerido de 60k. En este sentido y al buscar un margen mayor para el tipo de implementación que se desea realizar (transmisor DVB-S2), la solución más confiable es la dada por la combinación entre el FPGA Cyclone IV y el LimeSDR con un valor de 154k LUTs en total disponibles.

Es importante mencionar que el precio a pagar que conlleva esta elección es la mayor complejidad electrónica y de software en el desarrollo e integración del SDR, lo cual impacta en el tiempo de desarrollo del subsistema y los recursos intelectuales necesarios para llevarlo a cabo. Es decir, será necesaria una integración electrónica entre el FPGA y el Front End, adicional a un desarrollo en VHDL para la etapa de procesamiento lo cual requerirá de la experiencia y dedicación de al menos dos personas durante un tiempo estimado de 12 a 18 meses cada una.

#### 5.4.1 Diagrama del SDR propuesto

Para establecer los detalles de diseño del transmisor SDR se procede a enlistar los elementos que lo conforman así como a mostrar su diagrama de conexiones. Es importante recordar que este diseño está enfocado al desarrollo de un segmento transmisor bajo el estándar DVB-S2.

Como se estableció anteriormente, el SDR se conforma de dos segmentos principales: el bloque de procesamiento en banda base, el Front End Digital y el Front End Analógico. A continuación se describen las funciones de los elementos principales del SDR propuesto.

#### **Bloque de procesamiento**

- **FPGA**

Del proveedor Altera, modelo Cyclone IV EP4CE115. Encargado del procesamiento de la información entregada por la computadora de a bordo, la cual trata de una imagen en formato MPEG-2 y generada por la cámara integrada en la carga útil del CubeSat. El FPGA procesará, modulará, codificará y filtrará la trama de datos recibida y la adaptará para que cumpla con el estándar DVB-S2 en sus diferentes ModCod disponibles. Finalmente, el FPGA entregará una trama a la entrada del Front End la cual se trata de un puerto USB 3.0.

#### **Front End**

- **Controlador USB 3.0**

Cypress FX3 USB 3.0 (CYUSB3014-BZX). Se trata de una interfaz entre el conector USB 3.0 y el FPGA interno del Front End. Tiene como entrada el bloque de procesamiento y su salida se conecta directamente al FPGA interno.



- **FPGA interno**

Altera Cyclone IV EP4CE40. Es el FPGA interno del front end, con 40k LUTs es responsable de inicializar y controlar todos los elementos del mismo. Recibe los datos provenientes del controlador Cypress los cuales son procesados, en caso de estar configurado para ello, y llevados al DAC/ADC del transceiver. Este FPGA también puede ser utilizado para funciones de procesamiento de la señal utilizando el módulo UHD dentro de GNU Radio.

- **RF Transceiver**

Lime Microsystems LMS7002M MIMO. Transceptor con un rango de frecuencias de 0.3 a 3.8 GHz, convertidores ADC y DAC de 12 bits, y con anchos de banda programables desde 1.5 a 28 MHz.

- **Memoria** 256 MBytes DDR2 SDRAM
- **Detector de fase** ADF4002
- **Oscilador** Rakon RPT7050A@30.72MHz

En la figura 5.31 se muestra un diagrama en la que se muestran los elementos primordiales de la solución SDR propuesta. Por un lado, el procesamiento en banda base realizado por el FPGA seleccionado y por otro el Front End de RF embebido en la solución ya provista por el LimeSDR.

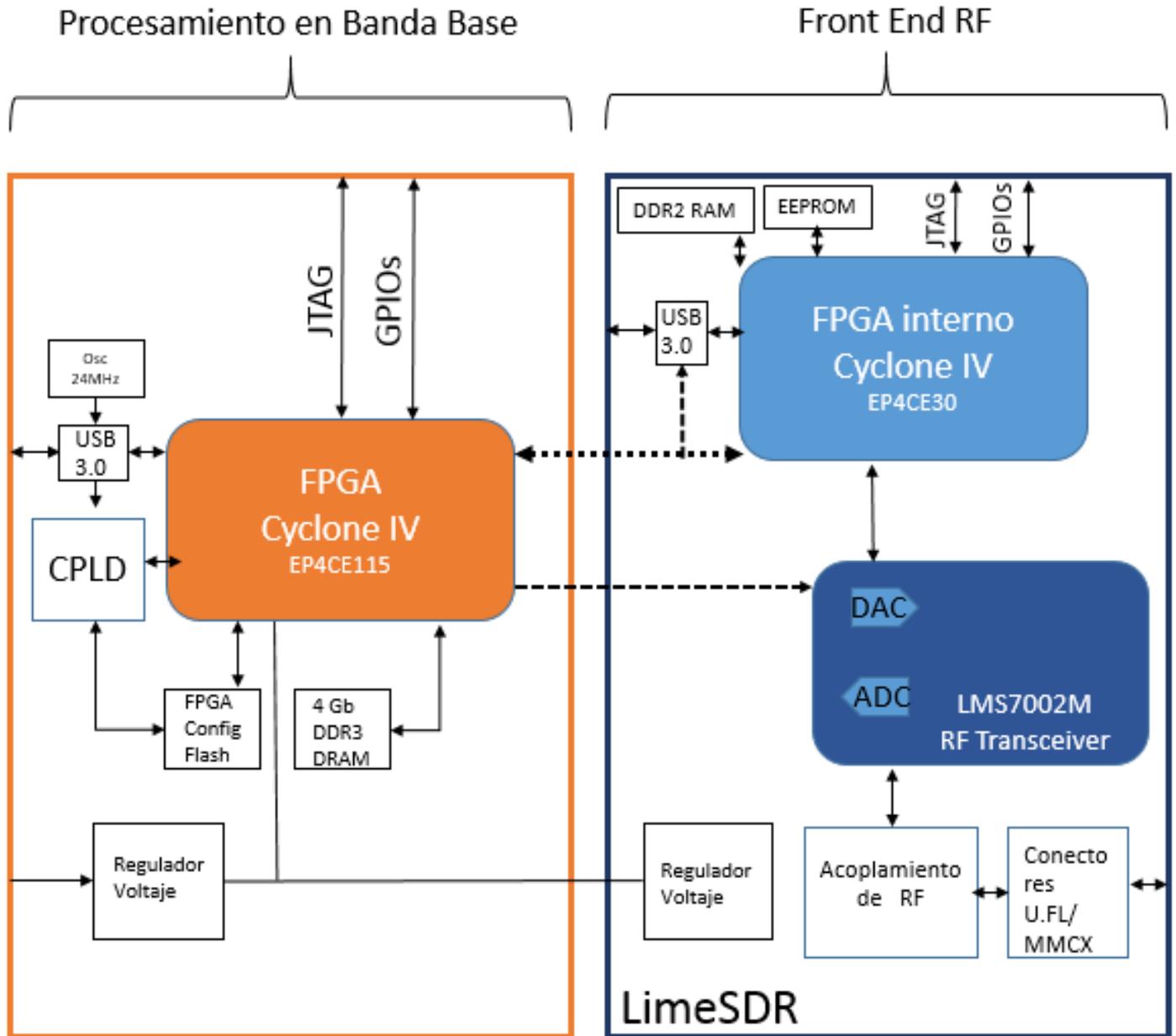


Figura 5.31 Diagrama del SDR propuesto (FPGA Cyclone IV + LimeSDR). Elaboración propia.

Como compilado final se presenta una vez más la tabla 5.12 en la cual se enlistan las características y parámetros del diseño final del transmisor SDR basado en COTS. Como se ha mencionado anteriormente la potencia de salida del front end es de +6 dBm por lo que se deberá adicionar un amplificador de potencia que aumente la señal a un valor de por lo menos 500mW.

Tabla 5.16 Parámetros de la solución final de SDR.

Frecuencia: 100 kHz – 3.8GHz   BW: 56MHz   2Tx / 2Rx   Pout: 6 dBm				
Elemento / Parámetro	FPGA (PCB)	Front End	Total	Margen
Modelo	Cyclone IV	LimeSDR	--	--
Dimensiones [cm]	9.4 x 9.4 x 2	10 x 6 x 1	<b>3 (Eje Z)</b>	+31 cm (Eje Z)
Volumen [cm <sup>3</sup> ]	36	60	<b>96</b>	+3,304 cm <sup>3</sup>
Posición	Horizontal	Horizontal	<b>Horizontal</b>	--
Masa [g]	10	30	<b>40</b>	+3960 g
Consumo de Potencia [W]	1.5	0.6	<b>2.1</b>	+7.9 W
Procesamiento (LUTs)	114k	40k	<b>154k</b>	+94k
Programación	VHDL/Verilog	VHDL/Verilog	--	
Herramienta de diseño	Quartus	Quartus	--	
Herramienta de desarrollo	ModelSim	GNU Radio	--	
Complejidad	Alta	Media	<b>Alta</b>	
Tiempo Desarrollo	Alto	Medio	<b>Alto</b>	
Herencia Vuelo	No	Sí	<b>Front End</b>	
Tolerancia Radiación	No	No	<b>No</b>	
Precio licencia [USD]	5,000/año	Open Source	<b>5,000</b>	
Precio adquisición [USD]	600	300	<b>900</b>	
Precio Lanzamiento [USD]	200	600	<b>800</b>	
Precio TOTAL [USD]	5,800	900	<b>6,700</b>	

Finalmente, en la figura 5.32 se da la imagen final del satélite CubeSat de 3U con el SDR y la carga útil integrados. Los diferentes márgenes de dimensión, volumen, masa, consumo de potencia y procesamiento son establecidos en la tabla 5.16.

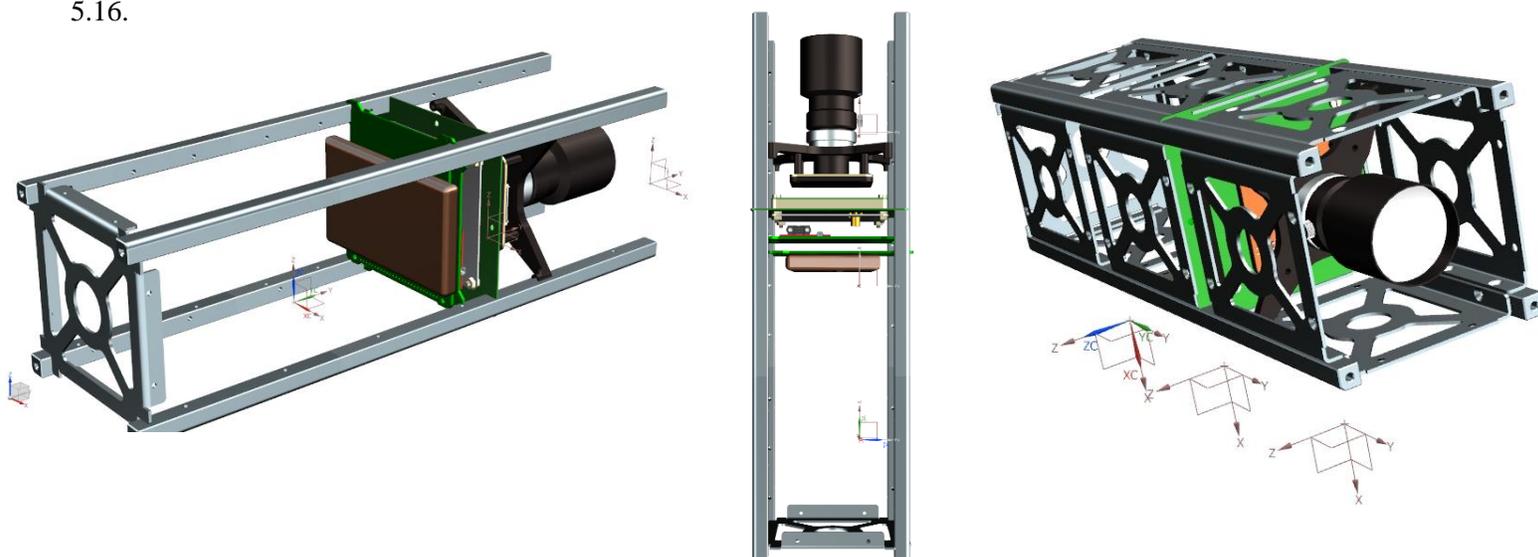


Figura 5.32. Propuesta final de CubeSat 3U con SDR y cámara integrados.



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



## ANÁLISIS FINAL DE RESULTADOS

- El avance de la tecnología satelital hace necesario el incremento de las velocidades de transmisión y la capacidad de una mayor flexibilidad en los subsistemas de comunicaciones, sobre todo en aquellos proyectos con aplicaciones relacionadas con la observación terrestre. Los proyectos y estándares basados en SDRs y esquemas ACM se muestran como una combinación clave en las soluciones tecnológicas del sector satelital en el presente y futuro.
- El estándar CubeSat, un facilitador esencial del acceso al espacio, ha contabilizado más de 950 proyectos en órbita del 2003 al 2018 y se estima que en los próximos 5 años se pongan en órbita más de 2000 CubeSats a nivel global.
- Durante los últimos 10 años, más de 220 empresas enfocadas al desarrollo o administración de la tecnología satelital han sido creadas alrededor del mundo. Esto es una cifra que triplica al número de empresas creadas desde la década de los 60s al 2009.
- Planet Labs, una empresa de observación terrestre basada en una constelación activa de 150 CubeSat de 3U, logra tasas de transmisión de datos de hasta 220 Mbps utilizando esquemas ACM, propio del estándar DVB-S2, implementado en sus transmisores SDR.
- La simulación de DVB-S2 implementada en Matlab Simulink arroja una buena aproximación del comportamiento de este estándar a pesar de que sólo cuenta con las modulaciones QPSK y 8PSK habilitadas. Para poder realizar pruebas y generar datos relacionados con el SNR y el BER con un error de hasta 1 dB es una herramienta funcional.
- Para el planteamiento de la misión satelital estudiada, se consideraron dos órbitas: la ISS (406km, 51.64°) y una SSO (600km, -98°). Del análisis orbital realizado con la herramienta STK, simulando un tiempo de misión de enero a diciembre de 2020, se obtuvo un rango promedio de 1,274 km@ISS y de 1,643 km@SSO.
- Al realizar el cálculo de enlace (frecuencia Tx=2.2 GHz; ET@UAT) y considerando los umbrales de conmutación de ModCod en DVB-S2 dados por la ETSI, se encontró una tasa de datos teórica promedio de 29.50 Mbps@ISS y de 24.91 Mbps@SSO con una potencia de transmisión (Ptx) de 0.5W. Este valor se incrementa en 4.74 Mbps y 6.17 Mbps, respectivamente, considerando una Ptx de 1W. Esto es, a pesar de que la potencia de transmisión sea incrementada un 100%, la tasa de datos sólo incrementa un 16% para la órbita ISS y un 24% para la SSO.
- De lo anterior se estimó un valor teórico de descarga de 143 fotos/día@ISS contra 134 fotos/día@SSO aplicando una Ptx de 0.5W. Para un valor de Ptx de 1W, los valores encontrados son 166 fotos/día@ISS contra 167 fotos/día@SSO. Estos valores significan un incremento del 18% y del 24% en el número de fotografías descargadas en cada órbita respecto al incremento en la Ptx de 0.5 a 1W.
- GNU Radio se muestra como una solución flexible, accesible, amigable y eficaz al ser responsable del bloque de procesamiento de la señal dentro de un sistema de comunicaciones. Sin embargo, la solución brindada no es portable, en términos nanosatelitales, debido a que GNU Radio es procesado en un sistema operativo basado en Linux, lo cual requirió el empleo de un PC o Laptop.



- Fue posible implementar un transmisor SDR funcional y compatible con DVB-S2 a un costo aproximado de 14,000 MXN. Esta implementación, considerada una mesa de pruebas, consta de un bloque de procesamiento basado en GNU Radio, un Front End modelo HackRF, un receptor satelital GTMedia V9 super, un par de antenas dipolo modelo G401-1 y diversos adaptadores coaxiales tipo F-RPSMA.
- El transmisor SDR es capaz de transmitir archivos de video, precargados en formato MPEG-2 .ts, en todos los ModCod dados por el estándar DVB-S2 únicamente bajo el modo CCM. La compatibilidad con el esquema ACM se establece como trabajo a futuro en conjunto con el establecimiento del enlace de retorno. Su desarrollo en esta etapa del proyecto no ha sido posible debido a que diversos requerimientos para llevarlos a cabo fueron encontrados y entendidos hasta muy tarde en el avance del mismo. Dentro de estos requerimientos los principales son: módulo de multiplexación en formato MPEG-2 aplicado trama por trama y en tiempo real a los archivos de video o datos a transmitir y permitir al sistema la transmisión trama a trama de los archivos para escuchar así el enlace de retorno y el ModCod seleccionado.
- Con las pruebas realizadas en la mesa de pruebas, se estimó el SNR experimental haciendo uso de un analizador de espectros modelo FSH4 de la marca Rohde&Schwarz. Se encontró que el valor del SNR experimental obtenido difiere con el valor propuesto por la ETSI para todos los ModCod implementados con una diferencia mínima de 1.53 dB para QPSK 2/3 y una máxima de 15.16 dB para 16APSK 9/10.
- Es así que la eficiencia del sistema SDR implementado va del 51.4% al 58.9% respecto al valor teórico obtenido del número de fotografías descargadas durante un año. De este dato se entiende la importancia de caracterizar el sistema transmisor – receptor desde una etapa temprana del desarrollo del proyecto con el fin de presupuestar la capacidad del sistema y evaluar si este cumple con los objetivos de la misión planteada. En el caso de la misión analizada se sabe que, considerando esta eficiencia, es posible descargar, por lo menos, 27,730 fotos/año@ISS  $P_{tx}=0.5W$ , lo cual significa un promedio de 75 fotos/día. Si esta misión fuese llevada a cabo, este número de fotografías sería suficiente para cubrir un área de 18,000 km<sup>2</sup> cada día, equivalente a 12 veces la superficie de la CDMX.
- El SDR implementado muestra la presencia de señales espurias, las cuales se asume son originadas por el efecto del amplificador de RF del segmento transmisor del HackRF. Estas señales se encuentran a  $\pm 7.8$  MHz respecto a la frecuencia de operación de 1.28 GHz, que es la única frecuencia a la cual el receptor fue capaz de detectar la señal enviada. Es por esto que se recomienda considerar un espaciado de  $\pm 11$  MHz respecto a la frecuencia de operación, ya sea para una señal adicional de transmisión o para cualquier señal a la recepción.
- En caso de utilizar el receptor satelital GTMedia V9 en las siguientes fases del proyecto, el empleo de un convertidor de frecuencia de 2.2GHz a 1.28GHz será necesario para la configuración final de la Estación Terrena.
- Los límites de la máscara espectral marcados por la ETSI son cumplidos por la mesa de pruebas con excepción de la modulación QPSK cuyos valores de potencia exceden a los estipulados de 2.35 a 6.25 dB dependiendo de la tasa de codificación.
- Respecto al transmisor SDR implementado se concluye que tiene diversos puntos de mejora: el valor de SNR puede ser más próximo a los valores teóricos, disminuir la potencia de las señales espurias, verificar que la máscara espectral para QPSK esté dentro de los valores permitidos y desarrollar la compatibilidad con el modo ACM.



- La potencia máxima alcanzada por el HackRF es de -30 dBm con espurias de valor -44dBm. Sin embargo, se encontró que el HackRF no es apto para una misión CubeSat debido a sus dimensiones y masa.
- Para el diseño o selección de componentes que conformarán los subsistemas de una misión nanosatelital los parámetros de masa, dimensiones y potencia consumida son esenciales, adicional a los propios del subsistema correspondiente. En el caso del subsistema de comunicaciones factores como frecuencia de operación, ancho de banda, velocidad y número de canales de transmisión, deben ser siempre considerados.
- Los SDR, así como los transceivers por debajo de los 10 GHz, desarrollados en la actualidad son capaces de operar en un rango de frecuencias que va de 1 MHz hasta los 6 GHz, con anchos de banda de 56 MHz y con DAC/ADC de 8 a 12 bits. En general tienen un canal a la recepción y uno en la transmisión, pero también son comunes los sistemas de 2x2 canales.
- La masa es un factor clave para el desarrollo de proyectos nanosatelitales. El costo de lanzamiento va de los 20,000 USD/kg en adelante por lo que una diferencia pequeña, por ejemplo de 100 gramos, equivaldría a 2,000 USD, importante sobre todo en proyectos universitarios.
- Se concluye que la implementación final del SDR sea realizada con apoyo de FPGAs. En este aspecto se estima que se requiere una capacidad de procesamiento entre 60k y 100k LUTs para embeber el estándar DVB-S2. De las diferentes opciones de FPGA estudiadas se seleccionan los FPGA Cyclone IV EP4CE115 de Altera seguido del Kintex 7 XC7K325T de Xilinx, por ser las más económicas y viables. Uno de los factores claves a considerar al momento de realizar un proyecto con FPGAs son los costos de licenciamiento de las diferentes herramientas de diseño y simulación, los cuales rondan los 60,000 MXN anuales.
- En general todas las soluciones de SDR integradas que fueron analizadas, tanto terrestres como espaciales, son conformadas por un dispositivo Xilinx ZYNQ 7000 (procesador ARM + FPGA Kintex 7) y de un transceiver AD9361. Esto habla de una tendencia en el desarrollo de las comunicaciones basadas en SDR y de la homologación de capacidades de RF. Es decir, los diferenciadores de la tecnología SDR se encuentran ahora en la capacidad de procesamiento, su tamaño y potencias optimizadas y su operación por encima de los 6 GHz. Para una implementación DVB-S2 los Soc Zynq 7030 o Zynq 7045 son recomendables.
- El diseño final del SDR propuesto consta de la integración de dos elementos tipo COTS: un FPGA Cyclone IV de Altera y el front End Lime SDR. Esto gracias a sus reducidas dimensiones, masa, precio y potencia consumida, además de sus capacidades en comunicaciones de hasta 3.8 GHz. Como segunda opción se propone el uso de la solución SDR integrada NanoCom, de GomSpace. Ambas tienen características de masa, dimensiones, volumen y potencias consumidas muy similares y aptas para un proyecto CubeSat. El precio final (adquisición + licenciamiento + lanzamiento) de las propuestas es de 6,700 USD para el LimeSDR y de 39,520 USD para el NanoCom SDR. Esto es una diferencia de casi el 600% entre ambas soluciones.
- La mayor desventaja de la propuesta basada en el LimeSDR es el tiempo y complejidad de desarrollo, tanto electrónico como de software. Por un lado, se debe de adaptar una placa electrónica para permitir el funcionamiento y programación del FPGA y por otro lado se debe de considerar el desarrollo del estándar DVB-S2 en código VHDL, el cual es un lenguaje de complejidad media.



- Todas las soluciones analizadas contaban con una potencia de transmisión no mayor a +10 dBm por lo que el uso de un amplificador de potencia capaz de suministrar una  $P_{tx}$  de al menos 0.5 W será necesario.
- La tolerancia a la radiación es un factor importante en el diseño y desarrollo de los subsistemas nanosatelitales. Se requiere encontrar el balance adecuado al tipo de solución a implementar. Por un lado, las soluciones que constan de recubrimientos físicos aumentan la masa del sistema, lo cual impacta directamente en el precio, y por otro lado las soluciones basadas en redundancia de software aumentarían aún más la complejidad del bloque de procesamiento. Sin masa
- Se considera que, en caso de requerir optimizar recursos económicos o se tenga como objetivo el desarrollo de sistemas electrónicos y de comunicaciones por cuenta propia la solución dada por la integración del FPGA y el LimeSDR es la mejor. Por otro lado, si no hay limitaciones económicas o se desea tener un grado de confiabilidad muy alto, la solución del NanoCom SDR de GomSpace es la más recomendada.



## CONCLUSIONES

Se concluye mencionando que los diferentes objetivos generales planteados al inicio de este proyecto han sido cumplidos satisfactoriamente.

Por un lado, se desarrolló un trabajo de investigación gracias al cual se logró vincular diversos impulsores tecnológicos de las comunicaciones satelitales tales como dispositivos SDR, estándar DVB-S2 y plataformas CubeSat. De esta investigación, es posible la identificación de los métodos para alcanzar altas velocidades de transmisión en proyectos nanosatelitales, donde una de las principales referencias es la constelación CubeSat de Planet Labs, la cual emplea SDRs basados en FPGAs implementando el estándar DVB-S2 en su enlace de bajada y alcanzando así tasas de transmisión de hasta 220 Mbps.

De igual modo, con base en este trabajo de investigación, fue posible establecer las características de comunicaciones principales para una misión satelital tipo CubeSat con aplicaciones de observación terrestre, para la cual se analizaron dos órbitas diferentes: la órbita de la ISS (400km, 51.64°) y una órbita heliosíncrona (600km, -98°), y dos potencias de transmisión posibles: 0.5 y 1W. La misión satelital consideró un módulo transmisor operando a una frecuencia de 2.2 GHz empleando el estándar DVB-S2. Así mismo, la Estación Terrena propuesta es la estación ECXSAT, ubicada en la UAT en Juriquilla, Querétaro.

Con base en la misión satelital planteada, se desarrolló el análisis orbital de la misma empleando el software STK, mientras que el análisis de comunicaciones se realizó a través del cálculo de enlace implementado en Matlab. Se consideraron parámetros tales como el BER, la SNR y los valores de conmutación de DVB-S2 establecidos por la ETSI. De estos análisis y simulaciones se obtuvieron valores de tasa de datos promedio y número de fotografías descargadas equivalentes a 29.50Mbps=143 fotos/día@0.5W y 34.24Mbps=166 fotos/día@1W para la órbita ISS. Para la órbita SSO los valores teóricos encontrados fueron de 24.91Mbps=134 fotos/día@0.5W y 31.08Mbps=167 fotos/día@1W. Con estos datos es posible concluir que la misión satelital de observación terrestre planteada puede ser viable en ambas órbitas e incluso para una potencia de transmisión de 0.5W.

Para la implementación del SDR se optó por utilizar el software GNU Radio y un front end HackRF. De este modo, fue posible establecer un SDR funcional capaz de transmitir archivos de video utilizando los diferentes ModCod del estándar DVB-S2 hacia un receptor satelital tipo COTS. El modo de transmisión implementado fue CCM. Con las pruebas realizadas se aproximaron valores de tasa de transmisión experimentales los cuales resultaron ser inferiores a los valores teóricos. Es así que la eficiencia del sistema SDR implementado va del 51.4% al 58.9% respecto al valor teórico obtenido del número de fotografías descargadas durante un año. De igual forma, la presencia de señales espurias originadas por el HackRF es determinada a una frecuencia de  $\pm 7.8$  MHz respecto a la frecuencia de operación. Este SDR fue realizado utilizando un CPU por lo que se considera como trabajo a futuro implementar el SDR en un FPGA para adquirir portabilidad.

Por otra parte, fue posible establecer el diseño basado en COTS de un sistema transmisor SDR con potenciales aplicaciones nanosatelitales. Para ello se realizó una comparativa de costos, capacidades y dimensiones de diferentes transceivers, FPGAs, front ends y SDR ya integrados. Dispositivos tales como el front end HackRF, el SDR embebido USRP E320 y otros, resultaron no ser aptos para la misión satelital propuesta debido a su baja capacidad de procesamiento o altas dimensiones, costos y consumos de potencia. En relación a la selección del FPGA óptimo para la implementación



en DVB-S2 se obtuvieron 2 opciones viables: Cyclone IV EP4CE115 de Altera y Kintex-7 de Xilinx. Cabe mencionar que el SoC Zynq 7000 (procesador ARM + FPGA Kintex 7) se muestra como una solución importante y ampliamente utilizada en SDR comerciales a nivel global por lo que su utilización es recomendada en el caso de optar por el proveedor Xilinx.

Uno de los aprendizajes más importantes ha sido entender el impacto que tiene la selección adecuada del FPGA desde el inicio del proyecto. En esencia, siempre deberían de considerarse 5 factores clave antes de comenzar a trabajar con un FPGA determinado: capacidad de procesamiento requerido, potencia consumida, software de desarrollo e implementación, características de licenciamiento de los mismos y costo final del dispositivo (adquisición + licenciamiento).

El diseño final del transmisor SDR propuesto considera el empleo del FPGA Cyclone IV EP4CE115 de Altera para el procesamiento de la señal bajo el estándar DVB-S2 y un front end LimeSDR para la etapa de RF, capaz de operar en un rango de los 100 kHz a los 3.8 GHz, ancho de banda de 61.44 MHz y con una potencia de transmisión de hasta +6 dBm en full duplex. En comparación con el Nanocom SDR, la segunda mejor solución encontrada, proveído por la compañía GomSpace, este diseño basado en COTS tiene un costo de 6,900 USD lo cual representa un costo menor en 600%. Sin embargo, se considera que la mayor complejidad electrónica para realizar la integración de los diferentes componentes COTS aumenta el tiempo de desarrollo del SDR de manera considerable. Esta propuesta se muestra como la base de un transmisor SDR, funcionando bajo el estándar DVB-S2, para un proyecto CubeSat 3U enfocado a la Observación Terrestre.

### CONSIDERACIONES PERSONALES

- El desarrollo de tecnología espacial requiere de una alta integración entre los diferentes campos del conocimiento que conforman cada uno de los subsistemas satelitales. Desarrollar un subsistema de comunicaciones requiere de una labor donde interactúan de manera directa y muy cercana la electrónica, la programación y las telecomunicaciones. En este sentido, es necesario colaborar mano a mano con ingenieros e investigadores con experiencia en electrónica y programación, particularmente FPGAs y VHDL. De igual modo, se hace evidente la importancia de acercar a los estudiantes en telecomunicaciones a estos campos del conocimiento, particularmente los sistemas embebidos, su programación y configuración.
- El esquema ACM es, hoy en día, un gran potencializador en las capacidades de los sistemas de comunicaciones, al igual que la tecnología SDR. El que los estudiantes de ingeniería se vean inmersos en estas tecnologías desde las primeras etapas de su formación representará una ventaja enorme para ellos en su desarrollo profesional y en general para los proyectos de comunicaciones desarrollados en la UNAM y en México.
- Herramientas de desarrollo como GNU Radio se muestran como un facilitador importante de cara al entendimiento y aplicación de los SDR. Con estas, se logra optimizar tiempo y recursos por lo que se cree que se deben explorar soluciones que brinden las ventajas de estas herramientas aplicadas directamente a dispositivos programables como los FPGAs. Un ejemplo de esto es GNU Radio, los USRP de Ettus y la interfaz entre ambas: RFNoC. Sin embargo, las grandes dimensiones y altos consumos de potencia de los USRP compatibles con RFNoC hacen imposible pensarlos para soluciones COTS de índole espacial.
- Durante el desarrollo de este proyecto se evidenció la interconexión existente entre las comunicaciones móviles inalámbricas, las comunicaciones espaciales y los beneficios y retos que los SDR tienen para ambas. Es así que se piensa



que este avance tecnológico puede ser mejor aprovechado si se logra generar una colaboración constante y estrecha entre los estudiantes, investigadores y profesionistas en general dedicados a este tipo de comunicaciones.

- Como comentario final, se desea hacer énfasis en las ventajas de adquirir una solución integrada y ya lista para su utilización. A pesar de que un componente o subsistema satelital sea dado por un proveedor externo, los retos y recompensas que conlleva la integración de este elemento en el sistema satelital final, son igual de importantes que el desarrollo de un dispositivo o tecnología de manera independiente y por cuenta propia. El generar experiencia en torno a proyectos integrados, finalizados y puestos en órbita debería ser la más grande de las prioridades del sector espacial en México. Esto es debido a que, aunque sólo se integren los subsistemas, la gestión del proyecto, el desarrollo de infraestructura, la administración de las frecuencias, la cualificación espacial del sistema, entre otros muchos y diversos requerimientos propios de una misión nanosatelital, representan la base y cimientos de una plataforma de desarrollo de proyectos satelitales activa y potencialmente capaz de desarrollar su propia tecnología espacial.



## **TRABAJO A FUTURO**

---

Para el trabajo a futuro se considera importante desarrollar el enlace de retorno, compatible con el estándar DVB-S2, el cual deberá ser capaz de definir el valor de SNR de la señal transmitida a la ET y así hacer que el modo ACM pueda ser utilizado en el sistema. Después de esto será importante generar el código VHDL para DVB-S2, directamente o con apoyo de alguna herramienta de desarrollo como Simulink o GNU Radio, para su posterior implementación en el FPGA seleccionado (Cyclone IV, Kintex 7 o Zynq 7030/7045).

De igual modo, se propone proseguir con el desarrollo del módulo electrónico para el FPGAs seleccionado. El objetivo del módulo es establecer la electrónica que permita la energización del sistema, la administración de memoria, la programación del FPGA, uso de conectores y la transferencia de datos entre este y el bloque de front end.

Finalmente, se recomienda realizar pruebas de caracterización de diferentes receptores satelitales compatibles con el modo ACM del estándar DVB-S2 con el objetivo de aumentar la eficiencia del sistema. Dependiendo de la elección de este receptor estimar si el uso de un convertidor de frecuencia será necesario a la entrada del mismo.



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



## REFERENCIAS

[A\_1] CubeSat Design Specification Rev. 13 (2014, febrero). Sitio web en:

[https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/56e9b62337013b6c063a655a/1458157095454/cds\\_rev13\\_final2.pdf](https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/56e9b62337013b6c063a655a/1458157095454/cds_rev13_final2.pdf)

[A\_2] A. J. Serrano Arellano, “Requerimientos para desarrollar y poner en órbita satélites CubeSat dentro de un entorno universitario”, Tesis de licenciatura, Facultad de Ingeniería, UNAM, CDMX, 2015.

[A\_3] J. R. Wertz, D. F. Everett, and J. J. Puschell, Eds., “Space Mission Engineering: The New SMAD”, Microcosm Press, 2011.

[A\_4] Evolución de la red de comunicación móvil, del 1G al 5G. [Universidad Internacional de Valencia, Online]. Sitio web en: <https://www.universidadviu.es/evolucion-la-red-comunicacion-movil-del-1g-al-5g/>

[A\_5] M. Angelone, “DVB-S2X system performance results for broadcast and unicast broadband networks”, IJSCN, no. 34, pp 387–412, Enero 2016.

[A\_6] H. Bischl, R. Rinaldo, “Adaptive coding and modulation for satellite broadband networks: From theory to practice”, IJSC, no. 28, pp 59 – 111, Marzo 2009.

[P\_1] (2019). Real GDP Forecast de OECD. Sitio web en: <https://data.oecd.org/gdp/real-gdp-forecast.htm3A ETSI DVB-S2>

[P\_2] J.Roch, B. Castañeda, M. Rodríguez. *Plan De Órbita 2.0, Mapa De Ruta Del Sector Espacial Mexicano*. Segunda Edición. CDMX: ProMéxico. 2017.

[P\_3] G. Nuñez Jiménez. (2019, Marzo 9). UNAMSAT. OM-30. Sitio web: <http://archive.is/1hSE#selection-23.0-23.14>

[P\_4] (2018, Marzo 5) .Buscan construir microsatélite mexicano para explorar selvas y bosques. [La jornada, Online]. Sitio web: <http://www.jornada.unam.mx/ultimas/2018/03/05/buscan-construir-microsatelite-mexicano-para-explorar-selvas-y-bosques-5706.html>

[J\_1] J.Roch, B. Castañeda, M. Rodríguez. *Plan De Órbita 2.0, Mapa De Ruta Del Sector Espacial Mexicano*. Segunda Edición. CDMX: ProMéxico. 2017.

[J\_2] Painani-1 Nanosatélite mexicano desplegado en el espacio [cienciamx, Online]. Sitio web: <http://www.cienciamx.com/index.php/tecnologia/nanotecnologia/26142-painani-1-nanosatelite-mexicano-desplegado-en-el-espacio>

[J\_3] (2019). UPAEP pondrá en órbita Nanosatélite con apoyo de NASA y AEM. [Online]. Disponible en: <https://www.upaep.mx>

[J\_4] México en el espacio: nanosatélites. [ICN UNAM, Online]. Sitio web en: <http://epistemias.nucleares.unam.mx/web?name=linx&page=55>

[J\_5] T. Robles. (2015, Diciembre 17). El nanosatélite Ulises 2.0 se acerca al espacio. [Agencia Informativa CONACyT, Online] Sitio web en: <http://www.conacytprensa.mx/index.php/tecnologia/transportes/4756-el-nanosatelite-ulises-2-0-se-acerca-al-espacio>

[J\_6] J.A. Ramírez Aguilar, S. de la Rosa Nieves, E. Sanchez Medina, S. Santillán Gutierrez, C. Romo Fuentes, J.A. Ferrer Pérez, O.M. Brekhov, “Satellite CONDOR UNAM-MAI: Technical scientific cooperation” en 6th International Conference on Recent Advances in Space Technologies (RAST), Junio, 2013.

[J\_7] (2013, Julio 18). Desarrollan En La Unam Microsatélite Para Monitoreo De Contaminación Atmosférica. [DGCS, Online]. Sitio web en: [http://www.dgcs.unam.mx/boletin/bdboletin/2013\\_433.html 18 julio 2013](http://www.dgcs.unam.mx/boletin/bdboletin/2013_433.html 18 julio 2013)

[J\_8] (Febrero 2014). Hoy se presentó en el INAOE el nanosatélite Ulises 1. [Online]. Sitio web en: <http://www.inaoep.mx/noticias/?noticia=121>

[J\_9] (2016, Julio 19). Painani-II. [IARU, Online]. Disponible en: [http://www.amsatuk.me.uk/iaru/finished\\_detail.php?serialnum=508](http://www.amsatuk.me.uk/iaru/finished_detail.php?serialnum=508)



- [J\_10] R. Capilla. (2017, Octubre 9). Conoce el Programa Espacial Universitario de la UNAM. [Agencia Informativa CONACyT, Online] Disponible en: <http://conacytprensa.mx/index.php/sociedad/politica-cientifica/18509-programa-espacial-universitario-unam>
- [1\_1] G. Salazar Diaz, “Desarrollo De Un Modulador Adaptativo Para Un Sistema De Telecomunicación A Bordo De Un Microsatélite”, Tesis de Maestría, Posgrado de Ingeniería, UNAM, CDMX, 2017.
- [1\_2] J.P. Montero Hidalgo, “Implementación de un sistema de comunicaciones basado en Software Radio”, Tesis de Licenciatura, Universidad Autónoma de Madrid, Madrid, 2014.
- [1\_3] E. Grayver, A. Chin. “Software Defined Radio for Small Satellites”. Aerospace Corporation. 2014.
- [1\_4] M. Donovan, A. Cozma, y D. Pu. “Four Quick Steps to Production: Using Model-Based Design for Software-Defined Radio (Part 4). Rapid Prototyping Using the Zynq SDR Kit and Simulink Code Generation Workflow”, Analog Dialogue, vol. 49 , Diciembre 2015.
- [1\_5] Mamatha R. Maheshwarappa, “Software Defined Radio (SDR) Architecture For Concurrent Multi-Satellite Communications” Ph. D., University of Surrey, 2016.
- [1\_6] L. M. Guerra Hidalgo, “Análisis y simulación del estándar DVB-S2 de televisión satelital usando varias modulaciones para los laboratorios de telecomunicaciones”, Tesis de Maestría, Universidad Politecnica Salesiana, Quito, 2017.
- [1\_7] M. P. Lara Mendoza, “Desarrollo De Un Sistema De Recepción-Transmisión Digital De Propósito General En Un FPGA” Tesis de Maestría, UNAM, CDMX, 2015.
- [1\_8] What is Software Defined Radio? [Wireless Innovation Forum].Sitio web: <https://www.wirelessinnovation.org/assets/documents/SoftwareDefinedRadio.pdf>
- [1\_9] Joseph Mitola III. (2000). Software Radio Architecture. Canadá: Wiley-Interscience.
- [1\_10] Akeela, R., & Dezfouli, B. (2018). Software-defined Radios: Architecture, State-of-the-art, and Challenges. *ArXiv*.
- [1\_11] T. J. Roupael, RF and digital signal processing for software-defined radio: a multi-standard multi-mode approach. Newnes, 2009.
- [1\_12] T. Hentschel, M. Henker, and G. Fettweis, “The digital front-end of software radio terminals,” IEEE Personal Communications, vol. 6, no. 4
- [1\_13] M. N. O. Sadiku and C. M. Akujobi, “Software-defined radio: a brief overview,” IEEE Potentials
- [1\_14] P. B. Kenington, *RF and Baseband Techniques for Software Defined Radio*. 1<sup>a</sup> Edición. Massachusetts: ARTECH HOUSE Mobile Communication Series, 2005.
- [1\_15] Oliveri, Steven J. (2011), *Modular FPGA Based software defined radio for CubeSats*, Masters Dissertation, Worcester Polytechnic Institute.
- [1\_16] Plishker, W., Zaki, G. F., Bhattacharyya, S. S., Clancy, C., & Kuykendall, J. *Applying graphics processor acceleration in a software defined radio prototyping environment*, In Rapid System Prototyping (RSP), 2011 22nd IEEE International Symposium.
- [1\_17] S. Kilts, *Advanced FPGA Design Architecture, Implementation, and Optimization*. New Jersey: John Wiley & Sons, Inc, 2007.
- [1\_18] (2019). Software Defined Radio Market worth 30.00 Billion USD by 2022. Mayo 2019, de Markets&Markets Sitio web: <https://www.marketsandmarkets.com/PressReleases/software-defined-radios.asp>



[1\_19] About GNU Radio. What is GNU Radio?. Sitio web en: <https://www.gnuradio.org/about/>

[1\_20] (2016, octubre 12). Evolución de la red de comunicación móvil, del 1G al 5G. [Universidad Internacional de Valencia] Sitio web: <https://www.universidadviu.es/evolucion-la-red-comunicacion-movil-del-1g-al-5g/>

[1\_21] M. Angelone, “DVB-S2X system performance results for broadcast and unicast broadband networks”, IJSCN, no. 34, pp 387–412, Enero 2016.

[1\_22] Global satellite capacity supply and demand, 16th edition. Northern Sky Research. 2016

[1\_23] Landmapper Constellation. [AstroDigital] Sitio web : <https://astrodigital.com/downloads/brochure-astro.pdf>

[1\_24] Where is the International Space Station? [ESA, Online]. Sitio web: [http://www.esa.int/Our\\_Activities/Human\\_Spaceflight/International\\_Space\\_Station/Where\\_is\\_the\\_International\\_Space\\_Station](http://www.esa.int/Our_Activities/Human_Spaceflight/International_Space_Station/Where_is_the_International_Space_Station)

[1\_25] H. Bischl, R. Rinaldo, “Adaptive coding and modulation for satellite broadband networks: From theory to practice”, IJSC, no. 28, pp 59 – 111, Marzo 2009.

[1\_26] ETSI EN 302 307-1 V1.4.1 (2014-07). “Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 1: DVB-S2”. 2014.

[1\_27] D. Breynaeri. “Analysis of the bandwidth efficiency of DVB-S2 in a typical data distribution network.”, CCBN2005, 2005.

[1\_28] Cubesats vector space systems. Spaceflightinsider. Abril, 2016. Sitio web en: <https://www.spaceflightinsider.com/missions/commercial/can-former-spacex-founding-member-enter-newspace-race-vector-space-systems/attachment/cubesats-vector-space-systems-photo-posted-on-spaceflight-insider/>

[1\_29] D. Leone. (2014, Junio 9). NASA Official: A Landsat 8 Clone Would Cost More Than \$650 Million. [SpaceNews, Online]. Sitio web: <http://spacenews.com/40841nasa-official-a-landsat-8-clone-would-cost-more-than-650-million/>

[1\_30] (2019, Marzo 18). Landsat 8. [USGS, Online]. Sitio web: <https://landsat.usgs.gov/landsat-8>

[1\_31] Global Monitoring, Daily Insight. [Planet, Online]. Sitio web: <https://www.planet.com/products/monitoring/>

[1\_32] Nanosatellite & CubeSat Database. [Nanosats Database]. Sitio web: <https://www.nanosats.eu/#database>

[1\_33] Nano/Microsatellite Forecast, 9th edition (2019). Spaceworks.

[1\_34] Bruce Yost (2018, marzo 28). Communications - State of the Art of Small Spacecraft Technology. NASA. Sitio web en en: <https://sst-soa.arc.nasa.gov/09-communications>

[1\_35] J. R. Wertz, D. F. Everett, y J. J. Puschell, Eds., “Space Mission Engineering: The New SMAD”, Microcosm Press, 2011.

[1\_36] (2012, Febrero 9). Unique Testbed Soon Will Be in Space. [NASA, Online]. Sitio web: [https://www.nasa.gov/mission\\_pages/station/research/news/Connect.html](https://www.nasa.gov/mission_pages/station/research/news/Connect.html)

[1\_37] Kiruthika D.&Kingsbury R.. (2015). Dove High Speed Downlin System. 31st Annual AIAA/USU, SSC17-VII-02, 1. Mazo 2019.



[2\_1] Satellite Design and Operations. System Tool Kit. AGI (2019) Sitio web en: <https://www.agi.com/products/satellite-design-and-operations>

[2\_2] M. A. Estrada Arellano, “Estación terrena en banda S y X para telemetría, rastreo y comando de satélites LEO desde la UAT FI UNAM”, Tesis de Licenciatura, UNAM, CDMX, 2017.

[2\_3] Cuadro de atribución de frecuencias. Instituto Federal de Telecomunicaciones. 2019. Sitio web en: <http://cnaf.ift.org.mx/CNAF/Index>

[2\_3] Software Downloads. AGI (2019). Sitio web en: <https://support.agi.com/downloads/>

[3\_1] J. Lassing, E. G. Strom, E. Agrell, & T. Ottosson, "Computation of the Exact Bit-Error Rate of Coherent M-ary PSK With Gray Code Bit Mapping," IEEE Trans. on Commun., vol.51, no.11, pp. 1758-1760, Nov. 2003.

[3\_2] Vela, R. N., & Ayala, S. L. (2015). *Comunicaciones por satélite*. Xalapa, México: Universidad Veracruzana.

[4\_1] A DVB-S2 and DVB-S2X transmitter for GNU Radio. Sitio web en: <https://github.com/drmpeg/gr-dvbs2>

[4\_2] ETSI EN 302 307-1 V1.4.1 (2014-07). “Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 1: DVB-S2”

[4\_3] HackRF One documentation. (2019) Sitio web en: <https://github.com/mossmann/hackrf/wiki/HackRF-One>

[4\_4] ANT 500. Great Scott Gadgets. (2016). Sitio web en: <https://greatscottgadgets.com/ant500/>

[4\_5] HackRF Host. Sitio web en: <https://github.com/mossmann/hackrf/tree/master/host>

[4\_6] HackRF Issues Forum. Sitio web en: <https://github.com/mossmann/hackrf/issues>

[4\_7] Open Source Mobile Communications. Projects. Sitio web en: <https://osmocom.org/projects>

[4\_8] Blankeship, J. (s.f.). Osmocom SDR Driver Install for Linux (2019). Sitio web en: <https://github.com/joeblankeship1/SDR/wiki/Osmocom-SDR-Driver-Install-for-Linux>

[4\_9] Antena 915 MHz. Ebay. 2019. Sitio web en: <https://www.ebay.com/itm/1pc-Lora-wireless-915-Mhz-antenna-915Mhz-antenna-rubber-Aerial-SMA-Male-pin-/262681541805>

[5\_1] E.T. Cuautle. (2016). *Engineering Applications of FPGAs*. Springer.

[5\_2] Xilinx, What is a FPGA? (2019). Sitio web en: <http://www.xilinx.com/fpga/>

[5\_3] W. Wayne (2015). *FPGA-Based System Design*. Pearson Education, Upper Saddle River.

[5\_4] Ferdjallah, M. (2011). *Introduction to Digital Systems: Modeling, Synthesis, and Simulation Using VHDL*. John Wiley & Sons

[5\_5] Cavanagh, J. (2017). *Verilog HDL: Digital Design and Modeling*. CRC Press.

[5\_6] VHDL El arte de programar sistemas digitales. David Maxinez.Jessica Alcalá. Instituto tecnológico y de estudios superiores monterrey. Continental. 2002

[5\_7] Stratix V FPGA – Intel FPGA (2019). Sitio web en: <https://www.intel.com/content/www/us/en/products/programmable/fpga/stratix-v.html>



- [5\_8] Virtex UltraScale+ (2019). Sitio web en: <https://www.xilinx.com/products/silicon-devices/fpga/virtex-ultrascale-plus-vu19p.html>
- [5\_9] Virtex UltraScale (2019). Sitio web en: <https://www.xilinx.com/products/silicon-devices/fpga/virtex-ultrascale.html>
- [5\_10] Virtex 7(2019). Sitio web en: <https://www.xilinx.com/products/silicon-devices/fpga/virtex-7.html#productTable>
- [5\_11] Kintex UltraScale Plus (2019). Sitio web en: <https://www.xilinx.com/products/silicon-devices/fpga/kintex-ultrascale-plus.html>
- [5\_12] Kintex UltraScale (2019). Sitio web en: <https://www.xilinx.com/products/silicon-devices/fpga/kintex-ultrascale.html>
- [5\_13] Kintex-7 (2019). Sitio web en: <https://www.xilinx.com/products/silicon-devices/fpga/kintex-7.html>
- [5\_14] Cyclone IV FPGAs (2019). Sitio web en: <https://www.intel.la/content/www/xl/es/products/programmable/fpga/cyclone-iv.html>
- [5\_15] SmartFusion2 SoC FPGAs (2019). Sitio web en: <https://www.microsemi.com/product-directory/soc-fpgas/1692-smartfusion2>
- [5\_16] Artix-7 FPGA Family (2019). Sitio web en: <https://www.xilinx.com/products/silicon-devices/fpga/artix-7.html>
- [5\_17] Spartan-7 FPGA Family (2019). Sitio web en: <https://www.xilinx.com/products/silicon-devices/fpga/spartan-7.html>
- [5\_18] R.O. Arias Zamora, “Codificador de Televisión Digital Terrestre de la norma DTMB con FPGA”, Tesis de Maestría, Universidad Central “Marta Abreu” de las Villas, Santa Clara, Cuba, 2017.
- [5\_19] C.Timmerman (MIT Lincoln Laboratory); I. Corretjer (Naval Research Laboratory). *Porting Of An Fpga Based High Data Rate Dvb-S2 Modulator*. Proceedings of the SDR 11 Technical Conference and Product Exposition, 2011 Wireless Innovation Forum.
- [5\_20] D. Biadene, “Implementation of DVB-T2 LDPC encoder on FPGA”, Tesis de Licenciatura, Universidad de Padova, Italia, 2011.
- [5\_21] Badave, S. M., & Bhalchandra, A. S. (2012). *Multiplierless FIR Filter Implementation on FPGA*. International Journal of Information and Electronics Engineering, 2(2).
- [5\_22] RFNoC (RF Network on Chip) (2019). Sitio web en: <https://www.ettus.com/sdr-software/rfnoc/>
- [5\_23] Labview National Instruments (2019). Sitio web en: <https://www.ni.com/es-mx/shop/labview.html>
- [5\_24] Modelsim-Intel FPGA Edition Software (2019). Sitio web en: <https://www.intel.la/content/www/xl/es/software/programmable/quartus-prime/model-sim.html>
- [5\_25] HDL Code Generation (2019). Sitio web en: <https://www.mathworks.com/help/comm/hdl-code-generation.html>
- [5\_26] Vivado Design Suite – HLx Editions (2019). Sitio web en: <https://www.xilinx.com/products/design-tools/vivado.html>
- [5\_27] Intel Quartus Prime Pro Edition Software (2019). Sitio web en: [https://www.intel.la/content/www/es\\_XL/programmable/buy/design-software.html](https://www.intel.la/content/www/es_XL/programmable/buy/design-software.html)
- [5\_28] Lime Microsystems Technology (2019). Sitio web en: <https://limemicro.com/technology/>
- [5\_29] RF Agile Transceiver AD9364 (2019). Sitio web en: <https://www.analog.com/media/en/technical-documentation/data-sheets/AD9364.pdf>



- [5\_30] RF Agile Transceiver AD9361 (2019). Sitio web en: <https://www.analog.com/media/en/technical-documentation/data-sheets/AD9361.pdf>
- [5\_31] Analog Devices Transceiver ADF7242 (2019). Sitio web en: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADF7242.pdf>
- [5\_32] Texas Instruments Transceiver CC2500(2019). Sitio web en: <http://www.ti.com/product/CC2500>
- [5\_33] HackRF Home (2019). Sitio web en: <https://github.com/mossmann/hackrf/wiki>
- [5\_34] USRP B200/B210 Bus Series (2019). Sitio web en: [http://www.ettus.com/wp-content/uploads/2019/01/b200-b210\\_spec\\_sheet.pdf](http://www.ettus.com/wp-content/uploads/2019/01/b200-b210_spec_sheet.pdf)
- [5\_35] USRP B205mini-i (2019). Sitio web en: <http://www.ettus.com/all-products/usrp-b205mini-i-board/>
- [5\_36] bladeRF 2.0 micro (2019). Sitio web en: <https://www.nuand.com/bladerf-2-0-micro/>
- [5\_37] About RTL-SDR (2019). Sitio web en: <https://www.rtl-sdr.com/about-rtl-sdr/>
- [5\_38] LimeSDR (2019). Sitio web en: <https://limemicro.com/products/boards/limesdr/>
- [5\_39] The New FunCube Dongle Pro+ (2019). Sitio web en: [http://www.funcubedongle.com/?page\\_id=1073](http://www.funcubedongle.com/?page_id=1073)
- [5\_40] Berger, E. (2019, 5 agosto). SpaceX enters smallsat launch market with a very low price. Ars Technica. Sitio web en: <https://arstechnica.com/science/2019/08/spacex-enters-competition-for-smallsat-launch-contracts-with-a-low-low-price/>
- [5\_41] SmallSat Rideshare Program (2019). Sitio web en: <https://www.spacex.com/SmallSat>
- [5\_42] PC/104 Specification version 2.6 (2008, octubre). Sitio web en: [https://pc104.org/wpcontent/uploads/2015/02/PC104\\_Spec\\_v2\\_6.pdf](https://pc104.org/wpcontent/uploads/2015/02/PC104_Spec_v2_6.pdf)
- [5\_43] CubeSat Design Specification Rev. 13 (2014, febrero). Sitio web en: [https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/56e9b62337013b6c063a655a/1458157095454/cds\\_rev13\\_final2.pdf](https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/56e9b62337013b6c063a655a/1458157095454/cds_rev13_final2.pdf)
- [5\_44] MARconISSta (2019). Sitio web en: <https://marconissta.com/>
- [5\_45] USRP E320 (2019). Sitio web en: <https://kb.ettus.com/E320>
- [5\_46] Datasheet NanoCOM SDR, Rev. 1.0, (2018, Noviembre). Sitio web en: <https://gomspace.com/UserFiles/Subsystems/datasheet/gs-ds-nanocom-sdr-10.pdf>
- [5\_47] Datasheet  $\mu$ SDR-C Space Micro. Rev. 7.2. (2019, Enero). Sitio web en: <http://www.spacemicro.com/assets/datasheets/rf-and-microwave/%C2%B5SDR-C.pdf>
- [5\_48] Zynq-7000 SoC. Product Selection Guide. (2019). Sitio web en: <https://www.xilinx.com/support/documentation/selection-guides/zynq-7000-product-selection-guide.pdf>
- [5\_49] GOMX-3. GomSpace. (2019). Sitio web en: <https://gomspace.com/gomx-3.aspx>
- [5\_50] L.E. Braten, A.N. Skauen & A. Yusuf (2018, Enero). *Nanosatellites in low earth orbits for satellite communications*. Norwegian Defense Research Establishment.



- [5\_51] GOMX-3 eoPortal Directory (2019). Sitio web en: <https://directory.eoportal.org/web/eoportal/satellite-missions/g/gomx-3>
- [5\_52] Pumpkin Chassis Walls (2019). Sitio web en: [https://www.pumpkinspace.com/store/p30/Chassis\\_Walls.html](https://www.pumpkinspace.com/store/p30/Chassis_Walls.html)
- [5\_53] Triad RF Systems TA1109 S-Band CubeSat Amplifier (2019). Sitio web en: <https://triadrf.com/products-2/power-amplifiers/power-amplifiers-standard/ta1109/>
- [5\_54] ANT 2000 Datasheet Rev. 1.1. GomSpace. (2017, Septiembre). Sitio web en: <https://gomspace.com/UserFiles/Subsystems/datasheet/gs-ds-nanocom-ant2000-11.pdf>
- [5\_55] NanoCam C1U. GomSpace. (2019). Sitio web en: [https://gomspace.com/UserFiles/Subsystems/flyer/NanoCamC1U\\_HIGH.pdf](https://gomspace.com/UserFiles/Subsystems/flyer/NanoCamC1U_HIGH.pdf)
- [5\_56] GPS Antenna. GrabCAD Community. (2019). Sitio web en: <https://grabcad.com/library/gps-antenna-with-lna-1>
- [5\_57] CubeSat PCB. GrabCAD Community. (2019). Sitio web en: <https://grabcad.com/library/ocp-cubesat-pcb-1>
- [5\_58] TSSOP-28. GrabCAD Community. (2019). Sitio web en: <https://grabcad.com/library/tssop-28-thin-shrink-small-outline-package-28-leads-body-width-4-4-mm-1>
- [5\_59] FPGA. GrabCAD Community. (2019). Sitio web en: <https://grabcad.com/library/fpga-1>



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



## ANEXO A. MATLAB

### A1. Análisis BER vs Eb/N<sub>0</sub>

```
EbNodB=0:1:20;
EbNo=10.^(EbNodB/10);
%colors={'k-*','g-o','r-h','c-s','m-s','y-*','k-p','b:s','m:d','g:p'};
%4-PSK
m=2; %bits por símbolo
M=2.^m; % Orden de modulación
ber= 1/m*erfc(sqrt(EbNo*m)*sin(pi/M));
BER=log10(ber);
plotHandle=plot(EbNodB,ber,'k-*');
set(plotHandle,'LineWidth',1.5);
hold on
```

```
%8-PSK
m=3;M=2.^m;
ber= 1/m*erfc(sqrt(EbNo*m)*sin(pi/M));
BER=log10(ber);
plotHandle=plot(EbNodB,ber,'m-s');
set(plotHandle,'LineWidth',1.5);
legend('QPSK','8-PSK');
set(gca,'XTick',EbNodB);
set(gca,'YScale','log')
set(gca,'YLim',[10^-10 1])
%set(gca,'YTick',0:10^(-20));
ylabel('BER'); xlabel('Eb/No (dB)');
ylim=(10^-10 1)
title('BER vs Eb/No');
grid on;
```

#### %QPSK BER OBTENIDOS

```
QPSK_1_4=[0.2479,0.1886,0.01014,1e-7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
QPSK_1_3=[0.2771,0.2468,0.2064,0.1103,1e-7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
QPSK_2_5=[0.2854,0.2605,0.2317,0.1948,0.1334,1e-7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
QPSK_1_2=[0.2844,0.2611,0.2354,0.2067,0.1724,0.1276,1e-7,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
QPSK_3_5=[0.2865,0.2634,0.239,0.2132,0.1856,0.1566,0.1239,1e-7,0,0,0,0,0,0,0,0,0,0,0,0,0];
QPSK_2_3=[0.286,0.2632,0.2387,0.2126,0.1853,0.1562,0.1233,0.08172,1e-7,0,0,0,0,0,0,0,0,0,0];
QPSK_3_4=[0.2871,0.264,0.2392,0.2139,0.1866,0.159,0.1305,0.101,0.06765,1e-7,0,0,0,0,0,0,0,0,0];
QPSK_4_5=[0.2875,0.2649,0.2406,0.2146,0.1871,0.1587,0.1315,0.1039,0.07647,0.04396,1e-7,0,0,0,0,0,0,0];
QPSK_5_6=[0.2881,0.2648,0.2402,0.2145,0.1874,0.1602,0.1322,0.1049,0.07859,0.05188,0.005552,1e-7,0,0,0,0,0,0];
QPSK_8_9=[0.287,0.2642,0.2392,0.2134,0.186,0.1586,0.1305,0.1037,0.07825,0.05477,0.03272,1e-7,0,0,0,0,0,0];
QPSK_9_10=[0.2871,0.2643,0.2394,0.2135,0.1861,0.1586,0.1306,0.1038,0.07849,0.05549,0.03477,0.01215,1e-7,0,0,0,0,0];
```

#### %8PSK

```
OPSK_3_5=[0.3057,0.2858,0.2643,0.2419,0.2189,0.1956,0.1719,0.1482,0.1249,0.1003,0.06399,1e-7,0,0,0,0,0];
OPSK_2_3=[0.3713,0.3541,0.3344,0.3125,0.2891,0.2643,0.2386,0.2103,0.1821,0.1533,0.122,0.07107,1e-7,0,0,0,0];
OPSK_3_4=[0.3636,0.3458,0.326,0.3046,0.281,0.2561,0.2307,0.204,0.1773,0.1519,0.1266,0.1008,0.07141,1e-7,0,0,0];
OPSK_5_6=[0.3576,0.3396,0.3201,0.2979,0.2746,0.2501,0.2244,0.1986,0.1724,0.1474,0.123,0.1004,0.07905,0.05883,0.01722,1e-7,0];
OPSK_8_9=[0.3529,0.3348,0.3154,0.2937,0.2709,0.2461,0.2212,0.1952,0.1693,0.1447,0.1214,0.09897,0.07799,0.05893,0.04054,0.02112,1e-7];
OPSK_9_10=[0.3524,0.3343,0.3149,0.2932,0.2701,0.2452,0.2203,0.1945,0.169,0.1446,0.1214,0.09904,0.07822,0.05937,0.04172,0.0243,1e-7];
```



## A2. Cálculo de Enlace

### Función range\_stk

Para esta función se mandan llamar dos archivos: Matlab\_ISS\_Rangos.csv y Matlab\_SSO\_Rangos, los cuales contienen todos los valores de los rangos para ambas órbitas generadas con STK. Los archivos csv deben de contener una sola columna con los valores de los rangos sin encabezados. Ambos archivos deben de estar guardados en la misma carpeta en la que se encuentren las funciones guardadas. En el siguiente enlace se comparten estos dos archivos con los rangos disponibles:

<https://drive.google.com/open?id=1tqmPKxSLOdhPULOLXE8BkXEM9u0qkNHJ>

El propósito de la función es cargar los valores de los rangos al programa principal para su posterior procesamiento con cálculo de enlace.

```
function[Range]=range_stk(x)

if x==1
%Rango Órbita Estación Espacial Internacional tomado de STK [metros]
Range=csvread('Matlab_ISS_Rangos.csv')*1000;

if x==2
%Rango Órbita SSO tomado de STK [metros]
Range=csvread('Matlab_SSO_Rangos.csv')*1000;
```

### Función SNR

Realiza el cálculo de enlace de bajada y regresa el valor de SNR para cada valor de rango dado por la función range\_stk

```
function[CN]=snr(x)

%Función para calcular el SNR [dB] de un rango dado considerando Cálculo de
%Enlace

%Rango
Rm=x; %Rango obtenido de STK para la órbita seleccionada [metros]

Re=6378; %Radio de la Tierra [km]
Req=8500; %Radio equivalente de la Tierra [km]
c=3e8; %Velocidad de la luz [m/s]

%f_up=2030e6; %Frecuencia de subida [Hz]
f=2200e6; %Frecuencia de bajada [Hz]
BW=10e6; %Ancho de banda de la señal [Hz]
tetha=5; %Ángulo de elevación de la Estación Terrena[grados]

lat=20.704993; %Latitud de la Estación Terrena [grados]
long=-100.450931; %Longitud Estación Terrena [grados]
h0=1.966; %Altitud Estación Terrena sobre el nivel del mar [km]
L_cenit=0.033; %Atenuación [dB] por absorción en el cenit@2GHz
R_01=63; %Intensidad de lluvia excedida [mm/h] /Queretaro/Disponibilidad 99.99%
gamma=0.01129; %Coeficiente de Atenuación Específica[dB/km]@2Ghz

%Pérdidas en el espacio libre
L_el_down=20*log10((4*pi*Rm*f/c)); %Decibeles

%Pérdidas por absorción atmosférica
L_aa=L_cenit*cscd(tetha); % [dB]
```



%Pérdidas por efecto de lluvia (CCIR-UIT)

```
if lat<36
    hr=3+0.028*lat; % Altura del nivel del mar a la región de lluvia [km] (CCIR 564-4)
else
    hr=4-0.075*(lat-36); % [km]
end
```

```
if tetha>10
    d=hr-h0/sind(tetha); % [km]
else
    A=2*(hr-h0);
    d=A/sqrt((sind(tetha)^2)+(A/Req)+sind(tetha)); % [km]
end
```

```
r_01=1/(1+(d/(35*exp(-0.015*R_01)*cosd(tetha))))); % Factor Reductor 99.99%
L_lluvia=gamma*d*r_01; % Atenuación por lluvia[dB] @ 2GHz confiabilidad 99.99%
```

%Pérdidas adicionales

```
L_ad=4; %Pérdidas por desapuntamiento y conectores (Estimación)
```

%Pérdidas totales

```
L=L_el_down+L_aa+L_lluvia+L_ad; % [dB]
```

%Potencia de ruido

```
T_cd=20; % Temperatura de cielo despejado [K]
T_tierra=50; % Temperatura de Tierra[K]
%T_lluvia=280*(1-(1/L_lluvia)); % Temperatura de lluvia [K]
```

```
T_down=T_cd+T_tierra; % Temperatura de Ruido [K]
%T_down=T_cd+T_tierra+T_lluvia; % Temperatura de Ruido [K]
k=1.38e-23; % Constante de Boltzman [J/K]
```

```
N0=k*T_down; % Densidad de Ruido subida [J]
N=N0*BW; % Potencia de Ruido térmico subida [Watts]
NdB=10*log10(N);
N0=10*log10(N0); % [dB]
```

%PIRE Satélite

```
Pt=1; % Watt
Pt=10*log10(Pt); % dB
Gt=6; % Ganancia antena satélite dBi (CubeSat 3U de Planet Labs con antena parche)
```

```
PIRE_sat=Pt+Gt;
```

%Portadora

```
D=3.4; % Diámetro antena ET [metros]
eff=0.55; % Eficiencia antena ET
```

```
%Pr=30; % Potencia de transmisión ET[W]; Utilizado sólo en cálculo de subida
%Pr=10*log10(Pr); % Potencia en dB
```

```
Gr=eff*((pi*D*f/c)^2);
Gr=10*log10(Gr); % Ganancia Antena Tx dB
```

```
C=PIRE_sat+Gr-L; % Portadora recibida en la ET [dB]
```

```
CN=C-NdB; % C/N [dB]
end
```



### A3. Conmutación y tasa de datos ACM

```
function[Rbps]=modcod(x)
```

```
%Función de conmutación MODCOD en ACM
```

```
global MODE
```

```
CN=x;  
BW=10e6;  
RollOff=0.2; %Roll Off factor = 20%
```

```
%Se insertan los valores de la tabla 3.3. en el orden en el que se encuentran
```

```
SNR_min=[-2.1711,-1.1456,-  
0.2772,0.9239,2.0603,2.8641,3.7233,4.3238,4.7857,5.0814,5.7281,5.9314,6.1161,7.3080,8.2873,8.6384,9.4329,9.8764,10.1443,10.1905,10.7263,11.7611,11.9089,12.1  
307,12.6018,13.1931,14.4958,14.8284];% Valor de SNR obtenido de los umbrales dados por ETSI  
M=[2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3,4,4,4,4,4,4,5,5,5,5];%bits por símbolo QPSK=2; 8PSK=3; 16APSK=4; 32APSK=5  
CR_ldcp=[1/4,1/3,2/5,1/2,3/5,2/3,3/4,4/5,5/6,3/5,8/9,9/10,2/3,3/4,2/3,5/6,3/4,8/9,9/10,4/5,5/6,3/4,8/9,9/10,4/5,5/6,8/9,9/10]; %LDPC FEC Rates  
CR_bch=[0.98815,0.99111,0.99256,0.99407,0.99506,0.99630,0.99810,0.99630,0.99704,0.99506,0.99778,0.99780,0.99630,0.99810,0.99630,0.99704,0.99810,0.99778,  
0.99780,0.99630,0.99704,0.99810,0.99778,0.99780,0.99630,0.99704,0.99778,0.99780]; %BCH inflation tomado de ETSI DVB-S2  
k_snr=length(SNR_min);  
mode=1;
```

```
for j=1:k_snr;
```

```
if(CN>=SNR_min(j))  
mode=j;  
end  
end
```

```
%k_snr=mode;  
Eff_esp=M(mode)*CR_ldcp(mode)*CR_bch(mode); %Eficiencia Espectral  
Rbps=(BW*Eff_esp)/(1+RollOff); %Data Rate [bps]  
MODE=mode;  
end
```

### A4. Tasa de datos CCM

```
function[Rbps]=modcod_ccm(x)
```

```
%Cálculo del Data Rate considerando ModCod Constante (CCM) con QPSK 1/2
```

```
BW=10e6;  
RollOff=0.2; %Roll Off factor = 20%
```

```
M=2;%bits por símbolo QPSK=2; 8PSK=3  
CR_ldcp=1/2; %LDPC FEC Rates  
CR_bch=0.99407; %BCH inflation tomado de ETSI DVB-S2
```

```
Eff_esp=M*CR_ldcp*CR_bch; %Eficiencia Espectral  
Rbps=(BW*Eff_esp)/(1+RollOff); %Data Rate [bps]  
end
```



## A5. Velocidad de Transmisión DVB-S2 vs CCM

%DVB-S2. Tasa y Cantidad de datos generado para una órbita seleccionada

- %1. Se elige la órbita (ISS=406km, 51.64° o SSO=600km, 98°)
- %2. Se calcula el SNR para los rangos (obtenidos de STK) de dicha órbita.
- %3. Elección del Modcod adecuado en función del SNR y se obtiene el data rate
- %4. Se calcula la cantidad total de datos transmitidos en esa órbita.

%Nota: Los rangos fueron previamente calculados con STK para una simulación %del 01/01/2020 al 31/12/2020 y con una frecuencia de muestreo de 1 segundo.  
%Estación Terrena ECXSAT ubicada en Juriquilla, Querétaro.

```
clc;clear all;close all
```

```
global MODE;
```

```
orbita=1; %1=Órbita ISS; 2=Órbita SSO
```

```
if orbita==1
```

```
foto=0.03724609375; % Valor en GB de 38.14MBytes
```

```
else
```

```
foto=0.0391796875; % Valor en GB de 40.12Mbytes
```

```
end
```

```
Range=range_stk(orbita); %Entrega el valor de los rangos a evaluar para la %órbita seleccionada. Los rangos fueron previamente calculados en STK.
```

```
deltaT=1; %Tasa de muestreo en segundos utilizado en STK
```

```
%SNR [dB]
```

```
CN=snr(Range);
```

```
l=length(CN); %# de valores de SNR generados
```

```
%Conmutación
```

```
%ACM: QPSK, 8PSK, 16APSK y 32APSK; FEC desde 1/4 hasta 9/10.
```

```
for i=1:l;
```

```
x=CN(i);
```

```
Rbps=modcod(x);%Función Modcod. Calcula la tasa de datos en función del SNR obtenido. Utiliza ACM.
```

```
DR(i)=Rbps; %Tasa de datos [bps]
```

```
Data_acm(i)=Rbps*deltaT; %Cantidad de datos [bits]
```

```
ModCod(i)=MODE;
```

```
end
```

```
DR_acm_min=min(DR)*(1/1024)*(1/1024) %Data Rate mínimo para ACM
```

```
DR_acm_peak=max(DR)*(1/1024)*(1/1024) %Data Rate pico para ACM
```

```
DR_acm_avg=mean(DR)*(1/1024)*(1/1024) %Data Rate promedio para ACM
```

```
Data_acm_total=sum(Data_acm);%Datos transmitidos con ACM totales [bits]
```

```
Data_acm_GB=Data_acm_total*(1/8)*(1/1024)*(1/1024)%Datos transmitidos con ACM totales [GBps]
```

```
Fotos_acm=Data_acm_GB/foto %Cantidad de fotos transmitidas 1 foto Planet Labs =40.12MBytes
```

```
%CCM QPSK 1/2
```

```
for i=1:l;
```

```
Rbps=modcod_ccm(i);%Función Modcod. Calcula la tasa de datos en función del SNR obtenido. Utiliza ACM.
```

```
DR_ccm(i)=Rbps; %Tasa de datos [bps]
```

```
Data_ccm(i)=Rbps*deltaT; %Cantidad de datos [bits]
```

```
end
```

```
DR_ccm_peak=DR_ccm*(1/1024)*(1/1024);
```

```
Data_ccm_total=sum(Data_ccm);%Datos transmitidos en CCM totales [bits]
```

```
Data_ccm_GB=Data_ccm_total*(1/8)*(1/1024)*(1/1024)%Datos transmitidos en CCM totales [GBps]
```

```
Fotos_ccm=Data_ccm_GB/foto %Cantidad de fotos transmitidas 1 foto Planet Labs =40.12MBytes
```

```
Data_acm_megas=Data_acm*(1/1024)*(1/1024);
```

```
Data_ccm_megas=Data_ccm*(1/1024)*(1/1024);
```

```
y=50;
```

```
figure(1)
```



```
plot(1:l/y,Data_ccm_megas(1:l/y))
hold on
plot(1:l/y,Data_acm_megas(1:l/y))
ylabel('Tasa de datos [Mbps]');
xlabel('Muestra');
title('Tasa de datos CCM (QPSK 1/2) vs ACM');
grid on;

figure(2)
plot(1:l/12,CN(1:l/12));

if orbita==1
figure(3)
axes1 = axes('Parent',figure);
histogram(ModCod);
title('Histogramas de los ModCod empleados para la Órbita ISS');
ylabel('Número de eventos');
xlabel('ModCod');
grid on;
set(axes1,'XTick',...
[12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30],'XTickLabel',...
{'QPSK 9/10','8PSK 2/3','8PSK 3/4','16APSK 2/3','8PSK 5/6','16APSK 3/4','8PSK 8/9','8PSK 9/10','16APSK 4/5','16APSK 5/6','32APSK 3/4','16APSK 8/9','16APSK 9/10','32APSK 4/5','32APSK 5/6','32APSK 8/9','32APSK 9/10','',''},...
'XTickLabelRotation',90);

end

if orbita==2
figure(4)
axes1 = axes('Parent',figure);
histogram(ModCod,'FaceColor',[1 1 0]);
title('Histogramas de los ModCod empleados para la Órbita SSO');
ylabel('Número de eventos');
xlabel('ModCod');
grid on;
set(axes1,'XTick',...
[12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30],'XTickLabel',...
{'QPSK 9/10','8PSK 2/3','8PSK 3/4','16APSK 2/3','8PSK 5/6','16APSK 3/4','8PSK 8/9','8PSK 9/10','16APSK 4/5','16APSK 5/6','32APSK 3/4','16APSK 8/9','16APSK 9/10','32APSK 4/5','32APSK 5/6','32APSK 8/9','32APSK 9/10','',''},...
'XTickLabelRotation',90,'YTickLabel',...
{'0','20k','40k','60k','80k','100k','120k','140k','160k','180k','200k'});

end
```



## ANEXO B. GNU RADIO

### B1. Instalación y configuración de GNU Radio

Antes de construir e instalar GNU Radio, todas las dependencias deben estar instaladas y actualizadas. Para actualizar desde la línea de comandos en los sistemas Ubuntu ingresar la siguiente instrucción:

```
sudo apt-get update
```

En Ubuntu 18.04 ejecutar::

```
sudo apt-get -y install git swig cmake doxygen build-essential libboost-all-dev libtool libusb-1.0-0 libusb-1.0-0-dev libudev-dev libncurses5-dev libfftw3-bin libfftw3-dev libfftw3-doc libcppunit-1.14-0 libcppunit-dev libcppunit-doc ncurses-bin cpufrequtils python-numpy python-numpy-doc python-numpy-dbg python-scipy python-docutils qt4-bin-dbg qt4-default qt4-doc libqt4-dev libqt4-dev-bin python-qt4 python-qt4-dbg python-qt4-dev python-qt4-doc python-qt4-doc libqwt6abi1 libfftw3-bin libfftw3-dev libfftw3-doc ncurses-bin libncurses5 libncurses5-dev libncurses5-dbg libfontconfig1-dev libxrender-dev libpulse-dev swig g++ automake autoconf libtool python-dev libfftw3-dev libcppunit-dev libboost-all-dev libusb-dev libusb-1.0-0-dev fort77 libsdl1.2-dev python-wxgtk3.0 git libqt4-dev python-numpy ccache python-opengl libgsl-dev python-cheetah python-mako python-lxml doxygen qt4-default qt4-dev-tools libusb-1.0-0-dev libqwtplot3d-qt5-dev pyqt4-dev-tools python-qwt5-qt4 cmake git wget libxi-dev gtk2-engines-pixbuf r-base-dev python-tk liborc-0.4-0 liborc-0.4-dev libasound2-dev python-gtk2 libzmq3-dev libzmq5 python-requests python-sphinx libcomedi-dev python-zmq libqwt-dev libqwt6abi1 python-six libgps-dev libgps23 gpsd gpsd-clients python-gps python-setuptools
```

Después de actualizar se debe de reiniciar el sistema. Una vez reiniciado ingresar los siguientes comandos:

<https://github.com/gnuradio/gnuradio/tree/v3.7.13.4>

```
cd $HOME
mkdir workarea-gnuradio
cd workarea-gnuradio
git clone --recursive https://github.com/gnuradio/gnuradio
cd gnuradio
git checkout v3.7.13.4
git submodule update --init --recursive
mkdir build
cd build
cmake ../
make
make test
sudo make install
sudo ldconfig
gnuradio-config-info --version
gnuradio-config-info --prefix
gnuradio-config-info --enabled-components
gnuradio-companion
```



## B2. Conversión de archivos MP4 a MPEG compatible con el Transport Stream DVB-S2

Para codificar un archivo de video para la transmisión, hay varios pasos que deben seguirse. Primero se debe identificar la tasa de símbolos a la que se deberá codificar el flujo de transporte. Esto depende de la modulación y la tasa de codificación. Ron Economos ha creado una herramienta útil que calcula la tasa de codificación en función de la mayoría de las combinaciones de modulaciones y códigos.

Esta herramienta se puede encontrar en el enlace: <https://github.com/drmpeg/dtv-utils/blob/master/dvbs2rate.c>

Primero, el archive debe ser descargado.

```
wget https://raw.githubusercontent.com/drmpeg/dtv-utils/master/dvbs2rate.c
```

Luego, se compila utilizando el compilador gcc.

```
gcc dvbs2rate.c -o dvbs2rate -lm
```

Lo cual producirá un archivo ejecutable `dvbs2rate`.

Después, se corre el ejecutable para desplegar sus opciones:

```
./dvbs2rate
$ ./dvbs2rate
usage: dvbs2rate -sx <symbol rate>
Options:
  s = short FECFRAME rates
  x = DVB-S2X rates
```

como ejemplo se toma la siguiente configuración:

- Symbol Rate: 5000000
- Constellation: 16 APSK
- Code Rate: 9/10
- Pilots: On

Se debe obtener la tasa de datos de la capa de transporte de `dvbs2rate` como se muestra:

```
./dvbs2rate 5000000
```

Esto mostrará una table asociada al stream de transporte para las diferentes combinaciones de `Constellations`, `Code Rates` y `Pilots On and Off`.

Para `16APSK`, `Pilots On`, y `Code Rate of 9/10`:

```
coderate = 9/10, BCH rate = 8, ts rate = 17413400.455472
```

Este valor será utilizado posteriormente.

Para realizar la conversión de MP4 a MPEG-2 se emplea el programa `ffmpeg`:

```
sudo apt-get install ffmpeg
ffmpeg -i test.mp4 -c:v copy -c:a copy -muxrate 17413400 -f mpegts test.ts
```

Esto creará un archivo de flujo de transporte (.ts) que se puede usar dentro del bloque Fuente del archivo del diagrama de flujo.



## ANEXO C. HackRF

### C1. Comandos para descargar e instalar HackRFlib en Ubuntu 16.04

```
Sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get install git build-essential cmake libusb-1.0-0-dev liblog4cpp5-dev libboost-dev libboost-system-dev libboost-thread-dev
libboost-program-options-dev swig libfftw3-dev
git clone https://github.com/mossmann/hackrf.git
cd ./hackrf/host
mkdir build
cd build
cmake ../-DINSTALL_UDEV_RULES=ON
Make
sudo make install
cd ../../firmware
sudo apt-get install python-yaml
```

```
git submodule init
git submodule update
sudo add-apt-repository ppa:team-gcc-arm-embedded/ppa
sudo apt-get update
sudo apt-get install gcc-arm-embedded
cd libopencm3/
Make
cd ../hackrf_usb
mkdir build
cd build
cmake ..
Make
sudo ldconfig
sudo hackrf_spiflash -w hackrf_usb.bin
# Apretar el botón Reset del HackRF
cd ../.././
# Carpeta Hackrf
sudo hackrf_cpldjtag -x firmware/cpld/sgpio_if/default.xsvf
# Resetear HackRf cuando sea indicado
```

### C2. Instalación de osmoSDR en Ubuntu

En la terminal de Ubuntu ejecutar los siguientes comandos:

```
git clone git://git.osmocom.org/gr-osmosdr
cd gr-osmosdr/
git checkout gr3.6
mkdir build
cd build/
cmake ../
make
sudo make install
sudo ldconfig
cd build/
cmake ../-DENABLE_DOXYGEN=1
make -C docs
```



UNAM

---

**Programa de Maestría y Doctorado en Ingeniería**



UNAM

Programa de Maestría y Doctorado en Ingeniería

