

**Centro de Investigación Científica y de Educación  
Superior de Ensenada, Baja California**



**Maestría en Ciencias  
en Electrónica y Telecomunicaciones  
con orientación en Telecomunicaciones**

---

**Análisis y evaluación de algoritmos de compresión  
de datos para la descarga de imágenes en banda  
UHF a bordo de nanosatélites**

Tesis

para cubrir parcialmente los requisitos necesarios para obtener el grado de  
Maestro en Ciencias

Presenta:

**Kathya Vega Pérez**

Ensenada, Baja California, México

2021

Tesis defendida por

**Kathya Vega Pérez**

y aprobada por el siguiente Comité

---

Dr. Miguel Ángel Alonso Arevalo  
Director de tesis

Dra. Eloisa del Carmen García Canseco

Dr. Roberto Conte Galván

Dr. Jonas de Dios De Basabe Delgado

MC. Enrique Guerrero Arbona



---

Dra. María del Carmen Maya Sánchez  
Coordinador del Posgrado en Electrónica y Telecomunicaciones

---

Dr. Pedro Negrete Regagnon  
Director de Estudios de Posgrado

Kathya Vega Pérez © 2021

*Queda prohibida la reproducción parcial o total de esta obra sin el permiso formal y explícito del autor y director de la tesis*

Resumen de la tesis que presenta Kathya Vega Pérez como requisito parcial para la obtención del grado de Maestro en Ciencias en Electrónica y Telecomunicaciones con orientación en Telecomunicaciones.

## **Análisis y evaluación de algoritmos de compresión de datos para la descarga de imágenes en banda UHF a bordo de nanosatélites**

Resumen aprobado por:

---

Dr. Miguel Ángel Alonso Arevalo  
Director de tesis

En la actualidad la tecnología espacial para satélites pequeños ha tenido un progreso significativo en el ámbito académico, por ejemplo, el estándar CubeSat fue desarrollado para centros de investigación con el fin de promover diferentes proyectos científicos de exploración espacial. Este trabajo de investigación se enfoca en el análisis y evaluación mediante simulaciones numéricas, de la robustez y fiabilidad de 4 formatos de compresión de datos para la descarga de imágenes a bordo de nanosatélites en órbita baja. En este trabajo se considera que se utiliza un enlace descendente en la banda de 435-438 MHz. Esto puede ser de gran utilidad, principalmente para los países emergentes y en vías de desarrollo que cada vez toman más interés en la construcción de satélites pequeños, pues esta tecnología es una opción atractiva que puede contribuir a la expansión del capital intelectual de una nación. Para el propósito de este trabajo de tesis es importante utilizar formatos de compresión con pérdida y que cuenten con código fuente abierto con una licencia de uso libre para fines no comerciales. Entre muchas opciones analizadas del estado del arte en compresión de imágenes, se eligieron los formatos: JPEG, JPEG 2000, JPEG XR y WebP. Estos formatos se sometieron a pruebas de calidad objetiva, calidad subjetiva, tasa de compresión, complejidad numérica y robustez a la pérdida de bits, obteniendo resultados diversos en cada una de estas evaluaciones, por lo que se hacen sugerencias para elegir el mejor formato de compresión de acuerdo a las necesidades del enlace satelital.

**Palabras clave: CubeSat, formatos de compresión de imágenes, JPEG, JP2, JXR, WebP, enlace satelital.**

Abstract of the thesis presented by Kathya Vega Pérez as a partial requirement to obtain the Master of Science degree in Electronics and Telecommunications with orientation in Telecommunications.

### **Analysis and evaluation of data compression algorithms for downloading images in the UHF band on board satellites**

Abstract approved by:

---

Dr. Miguel Ángel Alonso Arevalo  
Thesis Director

Currently, space technology for small satellites has had significant progress in the academic field; for example, research centers developed the CubeSat standard to promote different scientific space exploration projects. This research focuses on analyzing and evaluating the robustness and reliability of 4 data compression formats for image downloading onboard low orbiting nanosatellites through numerical simulations. In this work, we consider using a downlink in the 435-438 MHz band. This research subject can be of great interest, mainly for emerging and developing countries that are interested in building small satellites. This technology is an attractive option that can contribute to the expansion of a nation's intellectual capital. In this thesis work, it is essential to use lossy compression formats with open source code or a free license for non-commercial use. Among many options analyzed from state of the art in image compression, the following formats were chosen: JPEG, JPEG 2000, JPEG XR, and WebP. These formats were tested for objective quality, subjective quality, compression rate, numerical complexity, and robustness to bit errors, obtaining different results in each of these evaluations. Finally, suggestions and remarks are made to choose the best compression format according to the needs of the satellite link.

**Keywords: CubeSat, image compression formats, JPG, JP2, JXR, WebP, satellite link.**

## **Dedicatoria**

***A mi familia***

## **Agradecimientos**

Al Centro de Investigación Científica y de Educación Superior de Ensenada por darme la oportunidad de realizar mis estudios de maestría y brindarme los conocimientos necesarios para este proyecto.

A mi director de tesis, Dr, Miguel Ángel Alonso Arevalo por su apoyo brindado, tiempo y dedicación en la realización de este trabajo. También quiero agradecer su interés en mi bienestar personal, por aceptarme y darme la oportunidad de terminar mi maestría cuando yo creía todo perdido. Pero más importante aún, por contribuir en que regresara a mí el gusto por la investigación y abrir las puertas a una línea de investigación que jamás pensé llegaría a ser tan interesante.

A los miembros del comité de tesis: la Dra. Eloisa del Carmen García Canseco, el Dr. Roberto Conte Galván, el MC. Enrique Guerrero Arbona y el Dr. Jonás de Dios De Basabe Delgado por sus aportaciones en la realización de este proyecto y que fueron de valiosa importancia.

A los miembros del CPP por su apoyo para continuar mis estudios de maestría y ayudarme a encontrar soluciones a los problemas presentados durante la realización de ésta.

Al departamento de Bienestar Estudiantil, la Psic. Yahaira Castañeda por su orientación y apoyo en momentos de estrés y ansiedad.

A mis padres, por el apoyo en la realización de este proyecto a pesar del tiempo y la distancia. Al resto de mi familia, mis hermanos Yes, Karely y Jr. y mis sobrinos, Nicole y Alex Jr. por traer alegría a mi vida en mis visitas a San Quintín.

A mis compañeros de clases, especialmente a Alexis por ayudarme con los pequeños problemas que se presentaban al escribir esta tesis, su apoyo en clases y acompañarme en todas mis aventuras durante este tiempo, incluso las veces que decidimos dejar las tareas de último momento.

A Eduardo por siempre apoyarme en todo lo que hago, por animarme a seguir adelante incluso cuando yo sentía que ya no podía seguir, por su compañía en los días de estudio y tareas, pero sobre todo por siempre creer en mí.

A todos los investigadores, estudiantes y personal del Centro de Investigación Científica y de Educación Superior de Ensenada por su enseñanza académica.

Finalmente agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo económico brindado para la realización de mis estudios de maestría.

# Tabla de contenido

	Página
Resumen en español .....	ii
Resumen en inglés .....	iii
Dedicatoria .....	iv
Agradecimientos .....	v
Lista de figuras .....	x
Lista de tablas .....	xiii
<b>Capítulo 1. Introducción</b>	
1.1. Antecedentes .....	1
1.2. Justificación .....	2
1.3. Objetivos .....	3
1.3.1. Objetivo general .....	3
1.3.2. Objetivos específicos .....	3
1.4. Organización de la Tesis .....	3
<b>Capítulo 2. Antecedentes sobre satélites</b>	
2.1. Satélites pequeños .....	5
2.2. Clasificación de los satélites pequeños .....	6
2.2.1. Clasificación de acuerdo a su peso. ....	6
2.2.2. Clasificación de acuerdo a su posición orbital .....	6
2.3. Estándar CubeSat .....	8
2.4. Bandas ISM .....	10
<b>Capítulo 3. Fundamentos de compresión de imágenes</b>	
3.1. Imagen digital .....	14
3.1.1. Formación de una imagen .....	15
3.1.2. Formación de una imagen en el ojo humano .....	16
3.2. Tipos de imágenes .....	17
3.3. Compresión de imágenes .....	18
3.3.1. Compresión con pérdida .....	19
3.3.1.1. Técnicas de codificación con predicción .....	20
3.3.1.2. Técnicas de codificación por transformada .....	20
3.3.1.3. Técnicas de codificación fractal .....	21
3.4. Selección de formatos de compresión .....	21
3.4.1. JPEG (JPG) .....	23
3.4.2. JPEG 2000 (JP2) .....	25
3.4.3. JPEG XR (JXR) .....	27
3.4.4. WebP .....	28
3.5. Base de datos de imágenes .....	30
3.6. Métricas de Calidad .....	31
3.6.1. Métrica de calidad objetiva .....	31

## Tabla de contenido (continuación)

3.6.2. Métrica de calidad subjetiva . . . . .	33
3.7. Complejidad numérica . . . . .	34
3.8. Tasa de Compresión . . . . .	34
3.9. Robustez a errores de bits . . . . .	35

### Capítulo 4. Evaluación del desempeño de los formatos de compresión

4.1. Selección de formatos de compresión . . . . .	37
4.2. Pruebas de calidad objetiva . . . . .	38
4.3. Análisis de resultados para métricas de calidad objetiva . . . . .	45
4.3.1. PSNR . . . . .	45
4.3.2. Tamaño promedio y tasa de compresión . . . . .	45
4.3.3. SNR . . . . .	46
4.3.4. SSIM . . . . .	46
4.4. Pruebas de calidad subjetiva . . . . .	46
4.5. Análisis de resultados para métricas de calidad subjetiva . . . . .	48
4.6. Pruebas de complejidad numérica . . . . .	51
4.6.1. Prueba de complejidad numérica utilizando XnView ejecutándose bajo Windows. . . . .	51
4.6.2. Prueba de complejidad numérica ejecutándose bajo el sistema operativo Linux. . . . .	52
4.6.3. Prueba de complejidad numérica ejecutándose bajo Raspberry Pi 4. . . . .	53
4.7. Pruebas de tasa de compresión . . . . .	54

### Capítulo 5. Robustez a la transmisión a través del canal satelital

5.1. Sistema de Telemetría y Control . . . . .	56
5.1.1. Consideraciones generale de diseño . . . . .	56
5.1.2. Cálculo del efecto Doppler . . . . .	59
5.1.3. Cálculo de las pérdidas por propagación en el espacio libre . . . . .	60
5.2. Tasa de error de bits (BER) . . . . .	61
5.3. Modelo de canal satelital . . . . .	62
5.3.1. Transmisor satelital de enlace descendente . . . . .	63
5.3.2. Ruta de enlace descendente . . . . .	63
5.3.3. Receptor de enlace descendente de estación terrestre . . . . .	64
5.4. Robustez a la pérdida de bits . . . . .	65
5.4.1. Análisis de resultados para robustez a la pérdida de bits . . . . .	67
5.4.1.1. JPG . . . . .	67
5.4.1.2. JP2 . . . . .	70
5.4.1.3. JXR . . . . .	70
5.4.1.4. WebP . . . . .	71
5.4.1.5. PSNR promedio . . . . .	73
5.5. Probabilidad de archivo dañado . . . . .	74
5.5.1. JPG . . . . .	74
5.5.2. JP2 . . . . .	74

## Tabla de contenido (continuación)

5.5.3. JXR . . . . .	75
5.5.4. WebP . . . . .	76
5.6. Evaluación con archivos del mismo tamaño (50 KB) . . . . .	76
<b>Capítulo 6. Conclusiones y trabajo futuro</b>	
6.1. Conclusiones . . . . .	79
6.2. Trabajo a futuro . . . . .	80
<b>Literatura citada</b> . . . . .	81

## Lista de figuras

Figura	Página
1. Ejemplos reales para la clasificación de los satélites según su masa. . . . .	7
2. Estándar Cubesat, tomado de (Chin <i>et al.</i> , 2008). . . . .	9
3. Diagrama del ojo humano (Basado en González <i>et al.</i> (2002)). . . . .	16
4. Diferencias entre imágenes tipo Raster y Vector, tomado de Carey (2017). . . . .	17
5. Ejemplos de imágenes en la base de datos. . . . .	30
6. Ajustes de calidad de compresión disponibles en XnView MP. . . . .	38
7. Ejemplo de compresión con distintos valores de calidad en XnView, utilizando el formato JPG. . . . .	39
8. Valores promedio de PSNR, SNR Y SSIM para cada uno de los formatos con calidad 100. . . . .	40
9. Ajuste polinomial para PSNR en JPG. . . . .	40
10. Ajuste polinomial para PSNR en JP2. . . . .	41
11. Ajuste polinomial para PSNR en JXR. . . . .	41
12. Ajuste polinomial para PSNR en WebP. . . . .	42
13. Conversión de PSNR a MOS, tomado de Khan <i>et al.</i> (2008). . . . .	43
14. Tabla comparativa de los parámetros de calidad para distintos formatos de compresión con un PSNR promedio de 38 dB. . . . .	44
15. Tabla comparativa de los parámetros de calidad para distintos formatos de compresión con un PSNR promedio de 30 dB. . . . .	44
16. Nombre aleatorio asignado a cada imagen con PSNR=38 dB. . . . .	47
17. Nombre aleatorio asignado a cada imagen con PSNR=30 dB. . . . .	47
18. Vista de la hoja de respuestas a llenar por cada observador. . . . .	48
19. Formato con mejor calidad subjetiva para PSNR=38 dB. . . . .	49
20. Formato con mejor calidad subjetiva para PSNR=30 dB. . . . .	49
21. Resultados de la prueba de complejidad numérica utilizando XnView MP bajo Windows. . . . .	52
22. Resultados de la prueba de complejidad numérica utilizando ImageMagick bajo computadora con sistema operativo Linux. . . . .	53
23. Resultados de la prueba de complejidad numérica utilizando ImageMagick bajo Raspberry Pi. . . . .	54
24. Resultados de la prueba de tasa de compresión con PSNR= 38 dB. . . . .	55
25. Resultados de la prueba de tasa de compresión con PSNR= 30 dB. . . . .	55

## Lista de figuras (continuación)

Figura	Página
26. definición de la geometría del satélite para una orbita baja, tomado de Pacheco <i>et al.</i> (2009) . . . . .	57
27. Geometría del efecto Doppler para una orbita baja, tomado de Pacheco <i>et al.</i> (2009) . . . . .	60
28. Diagrama a bloques del modelo de canal satelital. . . . .	62
29. Modelo de canal satelital en Simulink. . . . .	65
30. Imagen original antes de hacer la alteración de bits. . . . .	67
31. Ejemplos de imágenes reconstruidas para un $BER = 1 \times 10^{-3}$ . . . . .	68
32. Ejemplos de imágenes reconstruidas para un $BER = 1 \times 10^{-4}$ . . . . .	68
33. Ejemplos de imágenes reconstruidas para un $BER = 1 \times 10^{-5}$ . . . . .	69
34. Ejemplos de imágenes reconstruidas para un $BER = 1 \times 10^{-6}$ . . . . .	69
35. PSNR resultante por imagen para diferentes valores de BER en el formato JPG. La línea azul representa el PSNR de referencia y la línea naranja el PSNR resultante en cada una de las 133 imágenes de la base de datos. . .	70
36. PSNR resultante por imagen para diferentes valores de BER en el formato JP2. La línea azul representa el PSNR de referencia y la línea naranja el PSNR resultante en cada una de las 133 imágenes de la base de datos. . .	71
37. PSNR resultante por imagen para diferentes valores de BER en el formato JXR. La línea azul representa el PSNR de referencia y la línea naranja el PSNR resultante en cada una de las 133 imágenes de la base de datos. . .	72
38. PSNR resultante por imagen para diferentes valores de BER en el formato WebP. La línea azul representa el PSNR de referencia y la línea naranja el PSNR resultante en cada una de las 133 imágenes de la base de datos. . .	72
39. PSNR promedio en función del BER para los 4 formatos de compresión evaluados. . . . .	73
40. Probabilidad de archivo dañado (imposible de abrir) en el formato JPG para distintos valores de BER. . . . .	75
41. Probabilidad de archivo dañado (imposible de abrir) en el formato JP2 para distintos valores de BER. . . . .	75
42. Probabilidad de archivo dañado (imposible de abrir) en el formato JXR para distintos valores de BER. . . . .	76
43. Probabilidad de archivo dañado (imposible de abrir) en el formato WebP para distintos valores de BER. . . . .	77

44. PSNR promedio en función del BER para archivos de 50 KB de tamaño en los 4 formatos de compresión evaluados. . . . . 78

## Lista de tablas

Tabla	Página
1. Clasificación de los satélites de acuerdo a su masa (tomado de (Armenta <i>et al.</i> , 2016)). . . . .	6
2. Bandas de frecuencia atribuidas al servicio de radioaficionados por satélite (tomado de (Armenta <i>et al.</i> , 2016)). . . . .	12

## Capítulo 1. Introducción

---

Los satélites son una parte fundamental de la vida moderna y contribuyen al desarrollo tecnológico de los países. Sirven para distintos propósitos, siendo los más comunes: monitoreo de cambio climático, agricultura, telemedicina, sistemas de posicionamiento global (GPS), navegación y comunicaciones. Esto hace muy importante el desarrollo de esta tecnología en nuestro país.

Un satélite de telecomunicaciones artificial es un repetidor de comunicaciones de uso espacial desarrollado a través de un proceso de diseño, construcción y transporte hasta su órbita final. Estos se pueden clasificar de distintas formas, la más común es de acuerdo a su peso. Como se presentará a detalle más adelante, un satélite pequeño tipo CubeSat se define como un satélite de forma cúbica de 100 mm por lado y un peso no mayor de 1.33 kg (Lee *et al.*, 2014).

### 1.1. Antecedentes

Tradicionalmente el desarrollo de los satélites ha estado dominado principalmente por las grandes potencias como son: Estados Unidos, Rusia, los miembros de la Unión Europea, Japón, Canadá, China y la India. Esto se debe al gran costo que implica fabricar y lanzar un satélite al espacio. Los satélites pequeños son menos costosos de desarrollar y construir que los satélites convencionales y las naves espaciales de tamaño completo. Sin embargo, los líderes en la creación y lanzamiento de CubeSat siguen siendo las grandes potencias. Hasta 2018 habían sido lanzados 836 CubeSats, de los cuales 643 fueron lanzados por Estados Unidos (Villela *et al.*, 2019). Un número cada vez mayor de lanzamientos espaciales privados, comerciales y públicos (tanto militares como civiles) transportan cada vez más pequeñas cargas útiles “secundarias” en órbita a un costo mucho menor que las misiones especiales que requieren los satélites convencionales. Hasta enero de 2021, el número de nanosatélites lanzados al espacio era de 1357, de los cuales 1009 pertenecían a Estados Unidos (Kulu, 2021).

Los países emergentes y en vías de desarrollo a menudo sufren de problemas de necesidades básicas, como son: nutrición, atención médica adecuada, educación efectiva, estabilidad económica y seguridad. Por consiguiente, los gastos en ciencia espacial y tecnología de satélites en esos países pueden parecer inadecuados debido a la

necesidad de cubrir principalmente los programas sociales a corto plazo. Sin embargo, la prosperidad económica a largo plazo de estos países depende en parte del capital intelectual, cuyo avance requiere capacitación científica, así como el uso y el desarrollo de nuevas tecnologías. El CubeSat es un avance tecnológico reciente, por lo que puede contribuir a la expansión del capital intelectual de una nación emergente. La tecnología CubeSat es una opción atractiva para el avance de nuevos conceptos de ingeniería en el entorno único del espacio ultraterrestre y el estudio de estos fenómenos científicos, además de ser una alternativa bastante económica (Woellert *et al.*, 2011).

Hay diversos trabajos de investigación donde se hace el análisis del canal de comunicación entre satélites (Popescu, 2017), en donde se obtiene el presupuesto de energía requerido por el subsistema de radiocomunicaciones, que permite a un CubeSat intercambiar información con estaciones terrenas y/u otras CubeSat en órbita. Este artículo describe cómo los parámetros orbitales de la trayectoria de CubeSat determinan la distancia del enlace de la estación terrestre y presentan presupuestos de energía tanto para el enlace ascendente como descendente que incluyen tasas de datos alcanzables y márgenes de enlace (Popescu, 2017). Sin embargo, hasta este momento no hemos encontrado trabajos en literatura especializada que se enfoquen en el análisis del comportamiento de diferentes esquemas de compresión de imágenes para la descarga de fotografías de la superficie terrestre desde CubeSat. Por esta razón consideramos que el trabajo desarrollado en este proyecto podría ser de gran utilidad para la comunidad que trabaja en el desarrollo de CubeSats.

## **1.2. Justificación**

Hoy en día los avances en las tecnologías espaciales, tales como el diseño de satélites pequeños, están abriendo muchas puertas para la investigación científica. El estándar CubeSat es una parte importante de estos avances, pues disminuye los costos significativamente (precios accesibles para universidades y/o centros científicos) (Aréchiga, 2015).

En esta tesis analizamos y evaluamos mediante simulaciones numéricas la robustez de esquemas de compresión para la descarga de imágenes a bordo de nanosatélites en órbita baja que utilizan un enlace descendente en la banda de 435-438 MHz. Hasta

el momento de la redacción de este documento, no hemos encontrado trabajos que analicen de manera exhaustiva el comportamiento de diferentes esquemas de compresión de imágenes para la descarga desde un CubeSat. Este proyecto podría ser de gran utilidad para los países emergentes y en vías de desarrollo, que cada vez toman más interés en la construcción de satélites pequeños, pues esta tecnología es una opción atractiva que puede contribuir a la expansión del capital intelectual de una nación.

### **1.3. Objetivos**

#### **1.3.1. Objetivo general**

Analizar y evaluar mediante simulaciones numéricas la robustez y fiabilidad de esquemas de compresión de datos para la descarga de imágenes a bordo de nanosatélites en órbita baja, utilizando un enlace descendente en la banda de 435-438 MHz.

#### **1.3.2. Objetivos específicos**

- Crear una base de datos de imágenes terrestres adquiridas a bordo de satélites de órbita baja.
- Realizar una búsqueda de los formatos de compresión de imágenes con y sin pérdida, que sean más adecuados a bordo de un nanosatélite de órbita baja.
- Utilizar un modelo de canal satelital que incluya: línea de vista con atenuación, efecto Doppler y parámetros de la órbita del nanosatélite. Además simular un esquema de modulación utilizando GMSK a 9600 bps.
- Evaluar el desempeño de los formatos de compresión usando el modelo de canal satelital propuesto, en función de la tasa de compresión y robustez a los errores provocados durante la transmisión.

### **1.4. Organización de la Tesis**

A continuación se muestra un pequeño resumen del contenido de cada capítulo.

En el *Capítulo 2* se presenta una introducción a satélites, enfocándonos en su clasificación y en el estándar CubeSat, también se presentan las bandas de radiofrecuencia utilizadas en satélites pequeños.

En el *Capítulo 3* se describen algunos conceptos básicos sobre imágenes enfocados a la compresión de imágenes, tipos de formatos de compresión, métricas de calidad de imágenes subjetivas y objetivas, entre otros. Por otra parte se presentan los formatos de compresión de imágenes a evaluar en este trabajo, la base de datos de imágenes utilizada y las métricas para evaluar dichos formatos.

El *Capítulo 4* presenta la evaluación del desempeño de los formatos de compresión de imágenes, esta evaluación está compuesta por pruebas de calidad objetiva, calidad subjetiva, complejidad numérica del formato y tasa de compresión.

Las pruebas de robustez en la transmisión del canal satelital se presentan en el *Capítulo 5*, este capítulo incluye el modelo de un enlace de satelital y las pruebas finales de robustez a la pérdida de bits.

La discusión y conclusiones de la tesis, así como las recomendaciones sugeridas para un trabajo futuro se exponen en el *Capítulo 6*.

## Capítulo 2. Antecedentes sobre satélites

---

La definición clásica de un satélite natural es la de un cuerpo que orbita alrededor de un cuerpo de mayor tamaño a cierta distancia, ayudado solo por la fuerza gravitacional que existe entre los dos. Un buen ejemplo es la Luna, que ha estado orbitando la Tierra durante millones de años (Armenta *et al.*, 2016).

Un satélite de comunicaciones artificial es un repetidor espacial para comunicaciones desarrollado a través de un proceso de diseño, construcción y transporte a su órbita final. Los lanzamientos de satélites se logran mediante el uso de un cohete, cuyas rutas de lanzamiento y órbitas finales están determinadas por una parte de la física llamada Mecánica Orbital.

### 2.1. Satélites pequeños

En los últimos años diversas instituciones educativas, privadas y gubernamentales han mostrado un interés creciente en el desarrollo de los llamados satélites pequeños, esto con la finalidad de poder acceder al espacio con un costo apropiado para llevar a cabo proyectos educativos y tecnológicos, así como experimentos científicos (Medina, 2015). La disminución en los costos de componentes y piezas esenciales para su construcción, la miniaturización de los mismos y la reducción del tiempo de desarrollo, son características que favorecen la aproximación de la tecnología satelital hacia grupos de trabajo que históricamente, por falta de infraestructura y recursos económicos, se mantenían al margen de ella (Toorian *et al.*, 2008).

Los satélites pequeños, incluidos los CubeSats, han generado una importante actividad comercial, como por ejemplo la colaboración entre la empresa Pumpkin, Inc., y la Universidad de Stanford, en California, E.U.A., que conduce al desarrollo del Kit Cubesat. El éxito comercial sin duda se ha visto favorecido por el acceso de dominio público al estándar CubeSat (Woellert *et al.*, 2011).

La reducción de los costos para participar en actividades espaciales ha estimulado programas de desarrollo de satélites pequeños en todo el mundo por parte de los gobiernos, la industria y, en particular, las instituciones académicas de un número creciente de países con capacidades tecnológicas avanzadas y emergentes, incluyen-

do Argelia, Argentina, Brasil, Colombia, Egipto, Indonesia, Irán, Israel, Malasia, México, Nigeria, Pakistán, Sudáfrica, Corea del Sur, Turquía y Venezuela (Wood y Weigel, 2009).

## 2.2. Clasificación de los satélites pequeños

Los satélites de comunicación suelen clasificarse por peso, aunque también pueden clasificarse por tamaño, uso y aplicación, o área de cobertura.

### 2.2.1. Clasificación de acuerdo a su peso.

Una de las clasificaciones más aceptadas, y la cual incluye el propulsor, fue realizada por el Centro Espacial de Surrey de la Universidad de Surrey (Rycroft y Crosby, 2013) (ver Tabla 1). Según esta clasificación, los satélites pequeños pueden considerarse satélites con una masa inferior a 500 kg. En la figura ref fig: Classsatelite, se presentan imágenes de ejemplos de la vida real para clasificar los satélites según su masa.

**Tabla 1.** Clasificación de los satélites de acuerdo a su masa (tomado de (Armenta *et al.*, 2016)).

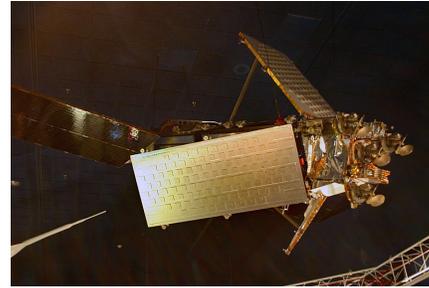
Tipo	Masa
Grandes	>1000 Kg
Medianos	500 a 1000 kg
Mini	100 a 500 kg
Micro	10 a 100 kg
Nano	1 a 10 kg
Pico	0.1 a 1.0 kg
Femto	< 0.1 kg

### 2.2.2. Clasificación de acuerdo a su posición orbital

Otro factor con el que se clasifican los satélites pequeños es su órbita. En el caso de los satélites pequeños, y debido a sus limitaciones en potencia dado su tamaño, la gran mayoría se encuentra en órbitas bajas.



**(a) Satellite grande: Telescopio Hubble (Tomado de Nasa (2008)).**



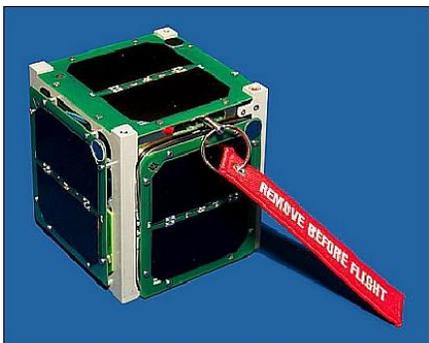
**(b) Mini-satelite: Iridium (Tomado de Long (2006)).**



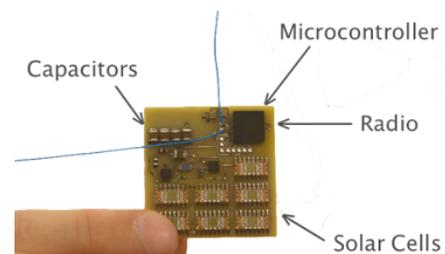
**(c) Micro-satelite: Sputnik-1 (Tomado de Richards (2006)).**



**(d) Nano-satelite: Vanguard-1 (Tomado de Rich (2014)).**



**(e) Pico-satelite: SwampSat (Tomado de eoPortal (2010)).**



**(f) Femto-satelite: Satellite-on-a-chip (Tomado de Gilster (2014)).**

**Figura 1.** Ejemplos reales para la clasificación de los satélites según su masa.

Los satélites pueden ser geoestacionarios (GEOS) y no geoestacionarios (NGEOS), según la relación entre el período orbital y el período de rotación de la Tierra. Se denominan órbitas geoestacionarias (GEO, geosynchronous orbit por sus siglas en inglés) porque la distancia entre ellas y la Tierra (35,758 kilómetros sobre el nivel del mar sobre el ecuador) les permite orbitar la Tierra con la misma velocidad angular que el período de rotación, es decir, una revolución cada 24 horas. Las órbitas no geoestacionarias (NGEO) pueden estar a distintas alturas de la Tierra, donde las más comunes son las órbitas bajas (LEO, low Earth orbit por sus siglas en inglés), y las órbitas medias (MEO, medium Earth orbit por sus siglas en inglés). Generalmente, los satélites pequeños están en órbitas NGEO, la mayoría de estos satélites se encuentran en la órbita LEO, que es mucho más baja que la órbita GEO y solo algunos de ellos están en órbita MEO (Armenta *et al.*, 2016).

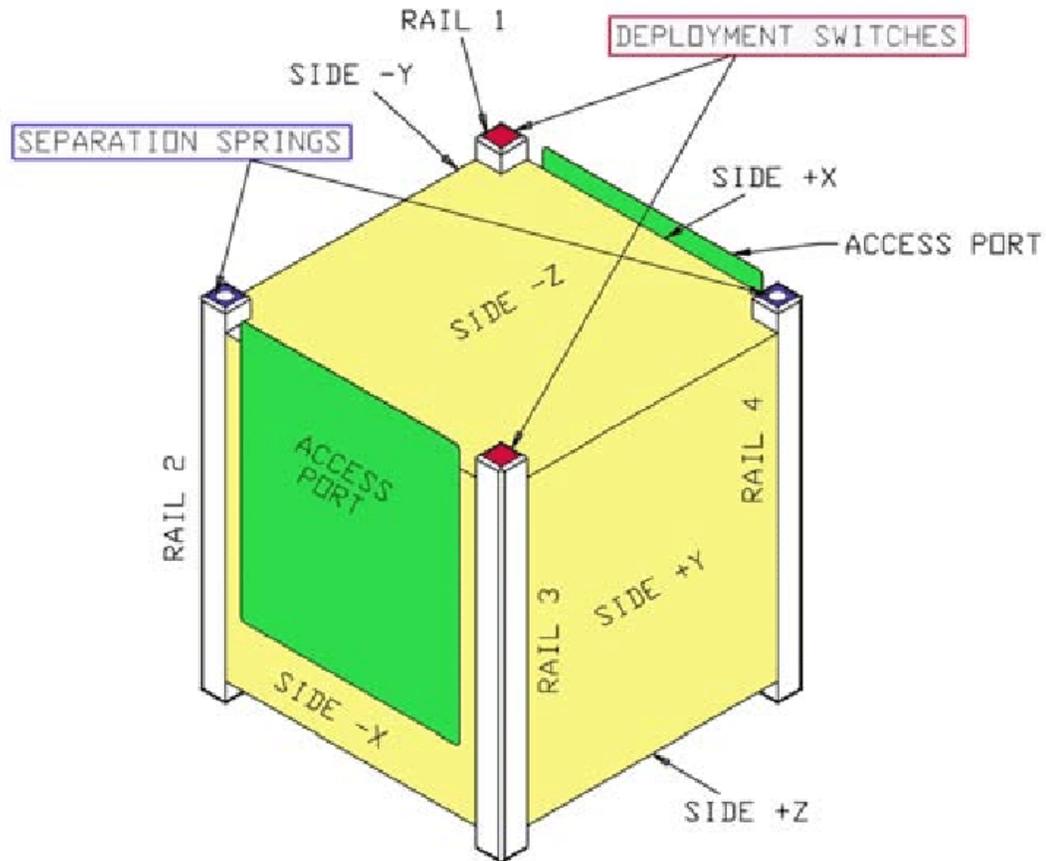
Los satélites de órbita terrestre baja (LEO) son satélites ubicados entre 200 y 1000 kilómetros sobre la superficie de la Tierra. En órbita, los satélites pierden velocidad debido a la fricción con las capas superiores de la atmósfera, motivo de su corta vida útil, ejemplo de esto son los nanosatélites tipo CubeSat. Cuando se requiere una vida más larga, están equipados con sistemas de corrección orbital para mantenerse a la altitud correcta, como los satélites iridium o globalsat.

Los satélites de órbita terrestre media (MEO) son satélites ubicados entre la órbita LEO y la órbita GEO. Estas órbitas se encuentran entre la órbita baja (4000 km) y el límite de la órbita geoestacionaria (35.000 km). La altitud orbital utilizada por MEOS está entre 18.000 y 24.000 km sobre la superficie de la Tierra.

### **2.3. Estándar CubeSat**

Las universidades fueron pioneras en el desarrollo de los satélites más pequeños, las categorías “nano” y “pico” a las que pertenecen los CubeSats. El desarrollo convencional de satélites requiere mucho capital, conocimientos y grandes equipos profesionales, lo que limita gravemente la participación de los estudiantes de ciencias e ingeniería. Reconociendo esto como particularmente problemático para los estudiantes de ingeniería aeroespacial, Jordi Puig-Suari del Instituto Politécnico de California (Cal Poly) y Robert Twiggs de la Universidad de Stanford, introdujeron las especificaciones

del CubeSat. El CubeSat técnicamente puede ser tanto un pico-satélite o un nano-satélite dependiendo de su masa; se tiene su tamaño estándar de 1U (una unidad) el cual es un cubo de 10 cm de lado y con una masa no mayor a 1.33 kg. Actualmente hay más de 100 universidades, centros de investigación, compañías privadas que están desarrollando CubeSats con distintos tipos de cargas útiles (Aréchiga, 2015).



**Figura 2.** Estándar Cubesat, tomado de (Chin *et al.*, 2008).

El Cal Poly estandarizó un sistema de despliegue de CubeSat llamándolo P-POD (Poly Picosatellite Orbital Deployer por sus siglas en inglés). Es capaz de llevar 3 CubeSat de 1U, y sirve de interfaz entre los CubeSat y el vehículo lanzador. El vehículo lanzador, que en su mayoría son cohetes, llevan como cargas secundarias a los CubeSat dentro de los P-POD, esto hace que sea más barato el financiamiento para la puesta en órbita. Algunas compañías y universidades han desarrollado sus propios POD (Picosatellite Orbital Deployer por sus siglas en inglés), los objetivos de los distintos POD son los de proteger al CubeSat durante el lanzamiento y también de proteger al vehículo lanzador de alguna falla del CubeSat (Chin *et al.*, 2008).

Los requerimientos técnicos se pueden obtener en Munakata *et al.* (2009), algunos de los principales puntos que se deben de cumplir para 1U son:

- Un cubo de 10 cm por lado.
- Peso no mayor a 1.33 kg.
- La energía acumulada no debe ser mayor a 100 Watt-hora.
- El satélite debe de estar apagado mientras se encuentra dentro del P-POD.
- Los sistemas de despliegue que tenga el satélite deben de ser activados 30 minutos después del lanzamiento.

Estas limitaciones tienen un gran impacto a la hora de diseñar los CubeSat, una restricción muy delicada es la que se tiene a la hora de generar energía para poder abastecer al satélite principalmente al subsistema de comunicación (es el subsistema que consume mayor energía a la hora de transmitir), ya que se cuenta con una pequeña área para las celdas solares para poder cargar a las baterías. Otro problema que se tiene con el reducido tamaño del CubeSat es que prácticamente no se tiene redundancia en ninguno de los subsistemas. Descrito lo anterior es que se deben de tomar las decisiones para determinar los tipos de enlaces de comunicación, capacidades de procesamiento y definir cargas útiles adecuadas para cada satélite.

Los CubeSats son puestos, comúnmente, en la órbita baja terrestre (LEO, por sus siglas en inglés), que comprende cualquier órbita menor a los 1,000 km de altitud, además siguen una trayectoria polar con un periodo orbital de aproximadamente 90 minutos si se encuentran a una altitud entre los 400 y 600 km (Medina, 2015). El tiempo de funcionamiento o vida útil de este tipo de instrumentos fluctúa entre unos cuantos días hasta casi cinco años, el promedio de duración es de ocho meses, manifestando con esto que son de poca duración (Bouwmeester y Guo, 2010).

#### **2.4. Bandas ISM**

Las bandas de radio ISM son partes del espectro de radio reservadas internacionalmente para fines industriales, científicos y médicos (ISM, por sus siglas en inglés) dis-

tintos de las telecomunicaciones. Los ejemplos de aplicaciones para el uso de frecuencia de radio (RF) en estas bandas incluyen radiofrecuencia, proceso de calentamiento, hornos de microondas, y algunas bandas también son utilizadas para el servicio de radioaficionados. Las potentes emisiones de estos dispositivos pueden crear interferencias electromagnéticas e interrumpir la comunicación por radio utilizando la misma frecuencia, por lo que estos dispositivos están limitados a determinadas bandas de frecuencias. En general, los equipos de comunicaciones que operan en estas bandas deben tolerar cualquier interferencia generada por las aplicaciones ISM y los usuarios no tienen protección reglamentaria contra el funcionamiento del dispositivo ISM (ITU, 2012).

La radioafición es el uso del espectro de radiofrecuencia con fines de intercambio no comercial de mensajes, experimentación inalámbrica, autoformación, recreación privada, radiosport, concursos y comunicaciones de emergencia. El término “aficionado” se utiliza para especificar “una persona debidamente autorizada interesada en la práctica radioeléctrica con un fin puramente personal y sin interés monetario” (Rinaldo, 1995).

La asignación de frecuencias de radioaficionado la realizan las autoridades nacionales de telecomunicaciones. A nivel mundial, la Unión Internacional de Telecomunicaciones (UIT) supervisa la cantidad de espectro de radio que se reserva para las transmisiones de radioaficionados. Las estaciones de aficionados individuales son libres de usar cualquier frecuencia dentro de los rangos de frecuencia autorizados; las bandas autorizadas pueden variar según la clase de licencia de la estación.

Por lo general, las comunicaciones entre la Tierra y los satélites pequeños, se realizan mediante los subsistemas de Telemetría, Telecomando y Control (TTC) que usualmente operan en las bandas de VHF y UHF, sin excluir la posibilidad del uso de otras bandas. En la mayoría de los proyectos académicos y educativos se utilizan frecuencias atribuidas al servicio de aficionados por satélite, debido a que por lo general, hay radioaficionados involucrados en dichos proyectos. En este sentido, en las bandas de VHF y UHF las bandas de 144 a 146 MHz y 435 a 438 MHz están atribuidas al servicio de aficionados por satélite (Armenta *et al.*, 2016). En la Tabla 2 se muestran las bandas atribuidas al servicio de aficionados por satélite.

**Tabla 2.** Bandas de frecuencia atribuidas al servicio de radioaficionados por satélite (tomado de (Armenta *et al.*, 2016)).

	Banda	Tipo de servicio	Comentarios
MHz	28.0 a 29.7	Primario	
	144 a 146	Primario	
	435 a 438	Secundario	
	1260 a 1270	Secundario	Segmento tierra a espacio
	2400 a 2450	Secundario	
	3400 a 3410	Secundario	En algunas regiones
	5650 a 5670	Secundario	Segmento tierra a espacio
	5830 a 5850	Secundario	Segmento espacio a tierra
GHz	10.45 a 10.50	Secundario	
	24 a 24.05	Primario	
	47 a 47.20	Primario	
	76.00 a 77.5	Secundario	
	77.5 a 78.00	Primario	
	78.00 a 81.00	Secundario	
	134.00 a 136.00	Primario	
	136.00 a 141.00	Secundario	
	241.00 a 248.00	Secundario	
	248.00 a 250.00	Primario	

Existen otras bandas de frecuencia que están destinadas para radioaficionados como por ejemplo la banda de VHF de 50 a 54 MHz, pero sería inviable utilizar esta banda debido a que por el tamaño de un CubeSat se complica implementar una antena de 6m de longitud necesaria para dicha banda de frecuencia.

Debido a la cantidad de satélites pequeños no geoestacionarios que orbitan la Tierra, existen bandas de frecuencia que empiezan a mostrar signos de saturación, principalmente algunas bandas atribuidas al servicio de aficionados por satélite. Si el proyecto contempla bandas que muestran condiciones de saturación, se puede correr el riesgo de que la IARU (International Amateur Radio Union) o la UIT no recomienden el uso de dichas frecuencias, obligando a que el proyecto tenga que reemplazar los equipos de radio, lo cual puede representar un gasto y tiempo adicional en el proceso de gestionar la concesión de recurso orbital.

Un ejemplo claro de esta situación es la banda en VHF de 144 a 146 MHz, la cual está atribuida a título primario al servicio de aficionados por satélite. Esta banda se encuentra saturada debido a que la mayoría de los satélites pequeños la utilizan para telemetría y telecomando. A raíz de esto, en 2014, la IARU comunicó que a partir de julio de 2014 dejaría de recibir solicitudes de coordinación para satélites experimentales

que utilicen estas frecuencias (IARU, 2013).

Los CubeSats suelen utilizar frecuencias en las bandas VHF y UHF para establecer enlaces de comunicación con estaciones terrestres, centrándose en las frecuencias de radioaficionados en las bandas VHF de 2 m (144 MHz a 148 MHz) y las bandas UHF de 70 cm (420 MHz a 450 MHz) (Muri y McNair, 2012). Para minimizar el valor de la pérdida de ruta de propagación y reducir los requerimientos de potencia para el transmisor CubeSat, las bandas de 2 m se utilizan para el enlace ascendente (estación terrestre a CubeSat), mientras que las bandas de 70 cm se utilizan para el enlace descendente (estación CubeSat a tierra) (Popescu, 2017), por esta razón se utilizará la banda de 420 MHz a 450 MHz para llevar a cabo la simulación de descarga de telemetría y datos en este trabajo de tesis.

## Capítulo 3. Fundamentos de compresión de imágenes

---

### 3.1. Imagen digital

Una imagen puede definirse como una función  $f(x, y)$ , donde  $x$  y  $y$  son coordenadas espaciales (planas), y la amplitud de  $f$  en cualquier par de coordenadas  $(x, y)$  se llama el nivel de intensidad o de gris de la imagen en ese punto. Cuando  $x$  y  $y$ , los valores de intensidad de  $f$  son todas cantidades finitas y discretas, llamamos a la imagen una imagen digital (González *et al.*, 2002). El campo de procesamiento de imágenes digitales se refiere al procesamiento de imágenes digitales por medio de un ordenador digital. Una imagen digital se compone de un número finito de elementos, cada uno de los cuales tiene una ubicación y un valor particular. Estos elementos se llaman píxeles.

Una imagen contiene uno o más canales de color que definen la intensidad o el color en una ubicación de píxel particular  $(m, n)$ . En el caso más simple, cada ubicación de píxel solo contiene un único valor numérico que representa el nivel de señal en ese punto de la imagen. La conversión de este conjunto de números a una imagen real (mostrada) se logra a través de un mapa de color. Un mapa de color asigna un tono de color específico a cada nivel numérico de la imagen para dar una representación visual de los datos. El mapa de colores más común es la escala de grises, que asigna todos los tonos de gris del negro (cero) al blanco (máximo) según el nivel de señal.

La escala de grises es particularmente adecuada para las imágenes de intensidad, es decir, las imágenes que expresan sólo la intensidad de la señal como un solo valor en cada punto de la región. En ciertos casos, puede ser mejor mostrar imágenes de intensidad usando un mapa de color falso. Uno de los principales motivos del uso de la pantalla de color falso se basa en el hecho de que el sistema visual humano sólo es sensible a aproximadamente 40 tonos de gris en la gama de negro a blanco, mientras que nuestra sensibilidad al color es mucho más fina. El color falso también puede servir para acentuar o delinear ciertas características o estructuras, haciéndolas más fáciles de identificar para el observador humano.

Además de imágenes de escala de grises donde tenemos un solo valor numérico en cada ubicación de píxel, también tenemos imágenes de color verdadero donde el espectro completo de colores se puede representar como un vector triplete, típicamente los componentes rojo, verde y azul (R, G, B, por sus siglas en inglés) en cada ubicación

de píxel. Aquí, el color se representa como una combinación lineal de los colores o valores básicos y la imagen puede considerarse como compuesta de tres planos 2D (Annadurai, 2007).

### 3.1.1. Formación de una imagen

El proceso de formación de imágenes se puede resumir como un pequeño número de elementos clave. En general, una imagen digital  $s$  se puede formalizar como un modelo matemático que comprende una representación funcional del objeto (o escena) (la función de objeto  $o$ ) y la del proceso de captura (la función de dispersión de puntos (PSF)  $p$ ). Además, la imagen contendrá ruido aditivo  $n$ . Estos son esencialmente combinados de la siguiente manera para formar una imagen.

$$\text{Imagen} = \text{PSF} * \text{función de objeto} + \text{ruido}$$

$$s = p * o + n$$

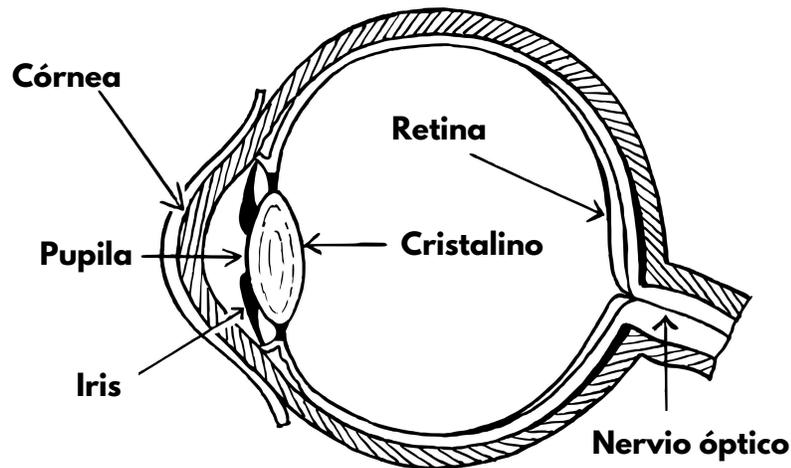
En este proceso tenemos varios elementos clave:

- PSF. Esto describe la forma en que la información sobre la función objeto se extiende como resultado del registro de los datos. Es una característica del instrumento de imagen (i.e. cámara) y es una función determinista (que opera en presencia de ruido).
- Función del objeto. Esto describe el objeto (o escena) que se está fotografiando (su superficie o estructura interna, por ejemplo) y la forma en que la luz se refleja desde esa estructura hasta el instrumento de formación de imágenes.
- Ruido. Esta es una función no determinista que, en el mejor de los casos, sólo puede describirse en términos de alguna distribución estadística del ruido (p. ej., gaussiana). El ruido es una función estocástica que es una consecuencia de todas las perturbaciones externas no deseadas que ocurren durante el registro de los datos de la imagen.
- Operador de convolución. Una operación matemática que convoluciona una función con otra.

Aquí, la función de la luz reflejada desde el objeto/escena (función objeto) se transforma en la representación de datos de imagen por convolución con el PSF. Esta función caracteriza el proceso de formación (o captura) de imágenes el cual se ve afectado por el ruido (Solomon y Breckon, 2011).

### 3.1.2. Formación de una imagen en el ojo humano

Aunque el campo del procesamiento digital de imágenes está construido sobre una base de formulaciones matemáticas y probabilísticas, la intuición y el análisis humanos juegan un papel central en la elección de una técnica frente a otra, y esta elección a menudo se basa en juicios subjetivos y visuales.



**Figura 3.** Diagrama del ojo humano (Basado en González *et al.* (2002)).

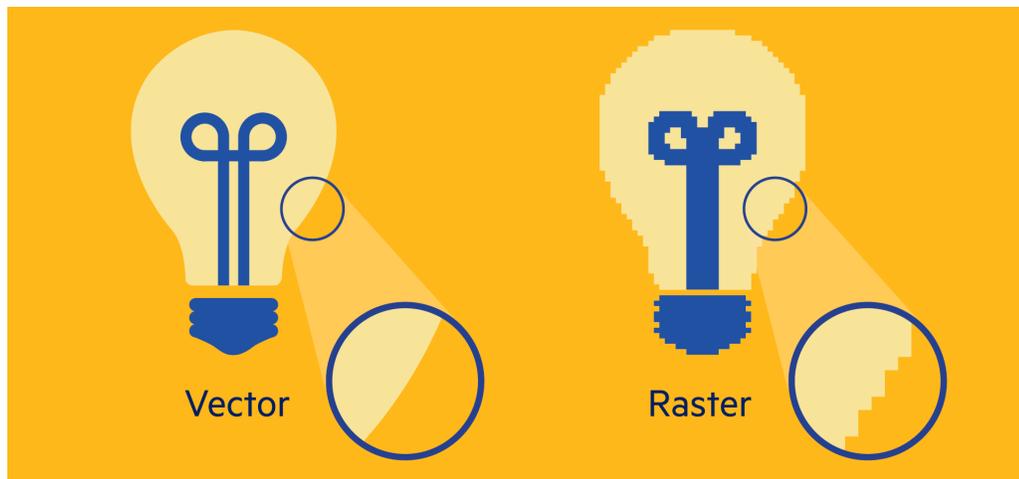
En una cámara fotográfica ordinaria, el objetivo (o lente) tiene una longitud focal fija, y el enfoque a varias distancias se logra variando la distancia entre el objetivo y el plano de imagen, donde se encuentra la película (o el chip de imagen en el caso de una cámara digital). En el ojo humano, lo contrario es cierto; la distancia entre el cristalino y la región de imagen (la retina) es fija, y la longitud focal necesaria para lograr el enfoque adecuado se obtiene variando la forma del cristalino. Las fibras en el cuerpo ciliar logran esto, aplanando o engrosando la lente para objetos distantes o cercanos, respectivamente. La distancia entre el centro de la lente y la retina a lo largo del eje visual es de aproximadamente 17 mm. El rango de longitudes focales es de aproximadamente 14 mm a 17 mm, esto último ocurre cuando el ojo se relaja y se

enfoca a distancias mayores de aproximadamente 3 m (González *et al.*, 2002).

### 3.2. Tipos de imágenes

Hay dos tipos principales de archivos de imagen: raster (también conocido como mapa de bits) y vectorial. Las imágenes raster se crean con software basado en píxeles o se capturan con una cámara o escáner. Son más comunes en general como jpg, gif, png, y son ampliamente utilizados en la web. Los gráficos vectoriales son formas matemáticas definidas creadas con software vectorial y no son tan comunes; se utilizan en ingeniería, animación 3D y diseño gráfico para procesos que reproducen una imagen en un objeto como el grabado y el corte de plantillas.

Cuando usas un programa de raster pintas una imagen y es similar a sumergir un pincel en pintura y pintar. Puede mezclar colores para suavizar la transición de un color a otro. Cuando se utiliza un programa vectorial se dibuja el contorno de las formas y es similar a la creación de una imagen con azulejos de todas las formas y tamaños diferentes, un claro ejemplo se muestra en la Figura 4.



**Figura 4.** Diferencias entre imágenes tipo Raster y Vector, tomado de Carey (2017).

Se pueden hacer muchas imágenes con programas raster o vectoriales y se ven exactamente iguales en ambos programas. Las imágenes con una gradación sutil de un color a otro son las imágenes que se verán más diferentes ya que los programas vectoriales necesitan crear una forma separada para cada tono de color.

Algunos programas vectoriales tienen la capacidad de crear gradientes de color

dentro de una sola forma, pero en realidad son efectos raster. Un gráfico vectorial con gradientes contiene elementos vectoriales y raster y no será adecuado para procesos que requieran un arte vectorial 100% verdadero, como el corte o grabado. Las fotografías son imágenes raster y son probablemente el mejor ejemplo de imágenes completamente hechas de mezclas de color (o mezclas de sombra en el caso de las fotografías en blanco y negro) y esas imágenes se ven muy diferentes cuando se dibujan en formato vectorial (Gomez, 2021).

### 3.3. Compresión de imágenes

Comprimir es el proceso mediante el cual se consigue representar cierto volumen de información usando menor cantidad de datos. Este tipo de representaciones son útiles porque permiten ahorrar recursos de almacenamiento y de transmisión. Los compresores de datos son codificadores universales, capaces de comprimir cualquier secuencia de datos si en ella se detecta redundancia estadística. La compresión de imágenes es un proceso semejante, pero los datos comprimidos siempre son representaciones digitales de señales bidimensionales. Los compresores de imágenes explotan, además de la redundancia estadística, la redundancia espacial. De esta forma, los niveles de compresión que se pueden conseguir aumentan (Torres, 2011).

Las técnicas de compresión de imágenes se basan en algoritmos que pueden desarrollarse tanto en hardware como en software. El rendimiento de un algoritmo de compresión de imágenes se mide en función de sus características de compresión de datos, por la distorsión inducida en las imágenes y por la complejidad de la implantación de este algoritmo. Como se indica en Skodras *et al.* (2001), un sistema de compresión puede modelarse como una secuencia de tres operaciones:

- Transformación (mapeo): Efectúa una traslación del conjunto de píxeles hacia otro dominio, donde el cuantificador y el codificador pueden utilizarse eficazmente. Es una operación reversible.
- Cuantificación: Efectúa la operación de clasificar los datos transformados en clases, para obtener un número más pequeño de valores. Es una operación irreversible.

- **Codificación:** La información ya comprimida se agrega a un tren binario que contiene ya sea la información que se debe transmitir o los códigos de protección y de identificación de las imágenes. Es una operación reversible. Una manera eficiente de lograr una compresión de imagen consiste en la codificación de una transformada de la misma, en lugar de la propia imagen. El objetivo de la transformación es obtener un conjunto de coeficientes en el plano transformado que posean una correlación menor que la existente en el plano de la imagen (redundancia entre píxeles).

Dar una descripción detallada sobre las técnicas y algoritmos de compresión de imágenes está fuera del alcance de este trabajo. En caso de querer más información sobre compresión de imágenes, se invita a revisar las siguientes referencias: González *et al.* (2002), Dhawan (2011), Pancholi *et al.* (2014), Wei (2008), Blanes *et al.* (2014), entre otras.

Los algoritmos de compresión de imágenes se dividen en dos categorías:

- La compresión con pérdidas (técnicas “lossy”), consigue su propósito eliminando alguna información de la imagen, produciendo por tanto una pérdida de calidad en la misma.
- Las técnicas de compresión sin pérdida (técnicas “lossless”), reducen el tamaño conservando toda la información original de la imagen, y por tanto no hay reducción de calidad de la imagen.

### **3.3.1. Compresión con pérdida**

Los algoritmos de compresión con pérdida introducen cierta distorsión entre la imagen reconstruida y la original, ya que no toda la información contenida en la imagen original está codificada en el archivo comprimido. Esto permite lograr ganancias de compresión significativas. El proceso de compresión con pérdida plantea la cuestión de garantizar que la calidad de la imagen reconstruida sigue siendo adecuada para el uso previsto. Por lo tanto sólo se toleran pequeñas cantidades de distorsión; sin embargo, esto permite obtener relaciones de compresión mucho más altas, convirtiéndose así en un enfoque cada vez más popular en las misiones de teleobservación.

La cantidad tolerada de distorsión puede depender de varios factores. En general, si la distorsión es pequeña con respecto al ruido inherente de adquisición de imagen, entonces su efecto probablemente será insignificante. Sin embargo, en algunas aplicaciones específicas se encuentra que incluso niveles de ruido más altos son aceptables, y no reducirá seriamente el rendimiento de la aplicación (Pancholi *et al.*, 2014).

Las técnicas de codificación de imágenes digitales con pérdida de información se clasifican en función de la aproximación que realizan al problema de la reducción de la redundancia psicovisual, en tres grupos:

- Técnicas de codificación con predicción.
- Técnicas de codificación por transformada.
- Técnicas de codificación fractal.

#### **3.3.1.1. Técnicas de codificación con predicción**

Las técnicas de codificación con predicción operan directamente sobre los píxeles de una imagen y se llaman, por tanto, métodos del dominio espacial. Entre las técnicas de codificación con predicción destacan la Modulación delta, los predictores óptimos y la cuantificación de Lloyd-Max. Podemos encontrar más información de cada una de las técnicas de codificación en Fernández (1999).

#### **3.3.1.2. Técnicas de codificación por transformada**

En este apartado se consideran las técnicas de compresión basadas en la modificación de la transformada de una imagen. En las técnicas de codificación de transformadas, se usa una transformación reversible (ondeletas o Fourier, por ejemplo), para mapear la imagen en un conjunto de coeficientes transformados, los cuales son cuantificados y codificados. Para la mayoría de las imágenes reales, la mayor parte de los coeficientes obtenidos poseen magnitudes muy pequeñas, que pueden ser cuantificadas de manera gruesa o descartados directamente con una pequeña distorsión final de la imagen. El decodificador de un sistema de codificación de transformada

realiza la secuencia de pasos inversa a la realizada por el codificador, a excepción de la función de cuantificación. La elección de una transformada particular depende de la cantidad de error de reconstrucción permisible para una aplicación concreta y de los recursos computacionales disponibles. La compresión se realiza durante el paso de cuantificación, no durante la transformación. Las transformaciones más comúnmente utilizadas para la compresión de imágenes digitales son las de Fourier (DFT), Hartley (DHT), Karhunen-Loève (KLT), coseno discreto (DCT), Walsh-Hadamard (WHT), Haar, Gabor (DGT) y Ondeletas (DWT), entre otras (Dhawan, 2011).

### **3.3.1.3. Técnicas de codificación fractal**

La compresión fractal utiliza los principios matemáticos de la geometría fractal para identificar patrones redundantes y repetidos dentro de las imágenes. Estos patrones pueden ser identificados mediante el uso de transformaciones geométricas, tales como escalado y rotación, sobre elementos de la imagen. Una vez identificado, un patrón repetido sólo necesita almacenarse una vez, junto con la información sobre su ubicación en la imagen y las transformaciones necesarias en cada caso. La compresión fractal hace un uso extremadamente intensivo del ordenador, aunque la descompresión es mucho más rápida. Es una técnica con pérdidas que puede lograr grandes tasas de compresión. A diferencia de otros métodos con pérdidas, una compresión más alta no produce la pixelación de la imagen, y aunque todavía se pierde información, esto tiende a ser menos evidente. La compresión fractal funciona mejor con imágenes complejas y altas profundidades de color (Fisher, 1994).

### **3.4. Selección de formatos de compresión**

A pesar de las muchas ventajas de la representación digital de señales en comparación con la contraparte analógica, necesitan un gran número de bits para el almacenamiento y la transmisión. La transmisión de estas señales digitales a través de canales de comunicación de ancho de banda limitado es incluso un reto mayor y a veces imposible en su forma bruta. Aunque el costo del almacenamiento ha disminuido drásticamente en la última década debido a los avances significativos en microelectró-

nica y tecnología de almacenamiento, la necesidad de aplicaciones de almacenamiento y procesamiento de datos está creciendo para superar este logro. Curiosamente, la mayoría de las señales sensoriales tales como imagen fija, video y voz generalmente contienen cantidades significativas de información superflua y redundante en su representación canónica en lo que respecta al sistema perceptivo humano. Por sistema perceptivo humano nos referimos a nuestros ojos y oídos. Por ejemplo, los píxeles vecinos en la región lisa de una imagen natural son muy similares y una pequeña variación en los valores de los píxeles vecinos no son perceptibles para el ojo humano. Este hecho nos dice que hay datos en las señales audiovisuales que no pueden ser percibidos por el sistema perceptivo humano. Llamamos a esto redundancia perceptiva. La compresión de datos es la técnica para disminuir las redundancias en la representación de datos con el fin de aminorar los requisitos de almacenamiento de datos y por lo tanto los costos de comunicación. Reducir las necesidades de almacenamiento equivale a aumentar la capacidad del medio de almacenamiento y, por tanto, el ancho de banda de las comunicaciones. Así, el desarrollo de técnicas de compresión eficientes seguirá siendo un desafío de diseño para futuros sistemas de comunicación y aplicaciones multimedia avanzadas (Acharya y Tsai, 2005).

El objetivo principal de este proyecto es analizar y evaluar mediante simulaciones numéricas el desempeño, la robustez y fiabilidad de distintos formatos de compresión de datos para la descarga de imágenes a bordo de nanosatélites en órbita baja, por lo que es importante decidir cuáles formatos serán evaluados. Para esto, es necesario que tales formatos cumplan con las siguientes características:

1. Deben ser formatos de compresión con pérdida que permita lograr una mayor compresión.
2. Es un requisito que los formatos sean de código fuente abierto, formato abierto y/o que cuenten con una licencia de uso libre para fines no comerciales.
3. Deben existir bibliotecas que soporten los formatos utilizados de manera eficiente bajo Linux embebido.
4. Tales formatos deben poder ser manejados a través de scripts en python y/o línea de comandos mediante bibliotecas open source de procesamiento de imágenes bajo Linux.

En un estudio preliminar los siguientes formatos de compresión de imagen cumplen con estas características y son candidatos a ser evaluados:

- JPEG (propuesto en 1992).
- JPEG XR (propuesto en 2009).
- JPEG XL (propuesto en 2017).
- JPEG 2000 (propuesto en 2000).
- BPG (Better Portable Graphics, propuesto en 2014).
- WEBP (propuesto por Google en 2014).
- FLIF (propuesto en 2016).

Después de hacer un análisis de estos formatos de compresión, decidimos utilizar JPEG, JPEG 2000, WebP y JPEG XR porque además de cumplir con las características antes mencionadas, estos formatos permiten hacer un ajuste en la calidad resultante al realizar la compresión. En el caso de FLIF, el formato dejó de recibir soporte y sus creadores comenzaron a trabajar en el proyecto de JPEG XL. Por su parte, JPEG XL ha sido diseñado para convertirse en el remplazo universal de formato de compresión de imágenes de tipo raster, sin embargo, el formato actualmente está en proceso de definición y no existían bibliotecas de software gratuitas que lo soporten por completo durante el desarrollo de este trabajo de tesis.

A continuación se presenta información fundamental sobre los formatos elegidos a evaluar.

### **3.4.1. JPEG (JPG)**

JPEG es el primer estándar internacional de compresión de imágenes para imágenes fijas de tono continuo, tanto en escala de grises como en color. JPEG es el acrónimo de Joint Photographic Experts Group. Este estándar de compresión de imágenes es el resultado de los esfuerzos de colaboración de la Unión Internacional de Telecomunicaciones (UIT), la Organización Internacional de Normalización (ISO) y la Comisión Electrotécnica Internacional (CEI). El estándar JPEG se conoce oficialmente como ISO/IEC IS

(International Standard) 10918-1: Compresión digital y codificación de imágenes fijas de tono continuo, requerimientos y recomendaciones. (ITU-T, 1992). El objetivo de este estándar es soportar una variedad de aplicaciones para la compresión de imágenes fijas de tono continuo de la mayoría de tamaños de imagen en cualquier espacio de color con el fin de lograr un rendimiento de compresión en o cerca del estado de última generación con relaciones de compresión ajustables por el usuario y una calidad de reconstrucción que puede ir de buena a excelente. Otro objetivo de esta norma es que tendría una complejidad computacional manejable para una implementación práctica generalizada (Acharya y Tsai, 2005).

JPEG es una técnica de compresión con pérdidas que proporciona muy buenas tasas de compresión con imágenes complejas de 24 bits (color verdadero). Logra su efecto eliminando datos de imagen imperceptibles al ojo humano, utilizando la transformada de coseno discreto (DCT), seguida de la codificación Huffman para lograr una compresión aún mayor (Torres, 2011).

Desde su creación en 1992, el objetivo de JPEG ha sido desarrollar un método de compresión de imágenes en tono continuo que cumpla los siguientes requisitos (Wallace, 1992):

- Estar en el estado del arte en la época en que fue propuesto, con respecto a la tasa de compresión y la fidelidad de la imagen que acompaña, en una amplia gama de calificaciones de calidad de imagen, y especialmente en la gama donde la fidelidad visual al original se caracteriza como “muy bueno” a “excelente”; también, el codificador debe ser parametrizable, de modo que la aplicación (o el usuario) pueda establecer la compensación de compresión/calidad deseada.
- Ser aplicable a prácticamente cualquier tipo de imagen de fuente digital de tono continuo (p. ej. no limitarse a imágenes de determinadas dimensiones, espacios de color, relaciones de aspecto de píxeles, etc.) y no limitarse a clases de imágenes con restricciones de contenido de escena, como complejidad, gama de colores o propiedades estadísticas.
- Tener una complejidad computacional manejable, para hacer implementaciones de software factibles con un rendimiento viable en una amplia gama de CPUs, así como implementaciones de hardware con costos viables para aplicaciones que

requieren un alto rendimiento.

### **3.4.2. JPEG 2000 (JP2)**

JPEG2000 es un estándar internacional para la compresión de imágenes, desarrollado conjuntamente por la Organización Internacional de Normalización (ISO) y la Comisión Electrotécnica Internacional (CEI) y también recomendado por la Unión Internacional de Telecomunicaciones (UIT). JPEG 2000 ha sido publicado como una norma ISO, ISO/IEC 15444-1(ISO, 2019). La filosofía subyacente detrás del desarrollo del estándar JPEG 2000 era comprimir una imagen una vez y decodificar el flujo de bits comprimido de muchas maneras para cumplir con diferentes requisitos de aplicaciones. La directriz sobre requisitos de la norma JPEG 2000 establecía algunas características deseadas que debían ser apoyadas por la norma para permitir su uso en diferentes áreas de aplicaciones.

El principal inconveniente del estándar JPEG 2000 en comparación con el original JPEG es que el algoritmo de codificación es mucho más complejo y las necesidades computacionales son mucho mayores. Además, la computación a nivel de bits puede restringir el buen rendimiento computacional con una plataforma de computación de propósito general. La compresión JPEG 2000 es más de 30 veces más compleja en comparación con el JPEG original (Acharya y Tsai, 2005).

El estándar JPEG 2000 proporciona un conjunto de características que son de importancia para muchas aplicaciones de alta gama y emergentes. Aborda áreas donde los estándares anteriores no producen la mejor calidad o rendimiento y proporciona capacidades a mercados que actualmente no utilizan compresión. Los mercados y aplicaciones mejor servidos por el estándar JPEG 2000 son Internet, impresión, escaneo, fotografía digital, teledetección, imágenes médicas, bibliotecas/archivos digitales y comercio electrónico. Cada área de aplicación impone algunos requisitos que el estándar, hasta cierto grado, debe cumplir. Algunas de las características más importantes que este estándar debería poseer son las siguientes (Skodras *et al.*, 2001):

- Rendimiento superior a tasa de bits bajas: Este estándar debería ofrecer un rendimiento superior a los estándares actuales a tasas de bits bajas (por ejemplo,

por debajo de 0.25 b/p (bits por píxel) para imágenes de escala de grises altamente detalladas). Este rendimiento de baja velocidad de bits significativamente mejorado debe lograrse sin sacrificar el rendimiento en el resto del espectro de distorsión de velocidad. La transmisión de imágenes en red y la teledetección son algunas de las aplicaciones que necesitan esta característica.

- **Compresión de tono continuo y de binivel:** se desea tener un estándar de codificación que sea capaz de comprimir tanto imágenes de tono continuo como de binivel. De ser factible, esta norma debería esforzarse por lograrlo con recursos del sistema similares. El sistema debe comprimir y descomprimir imágenes con varios rangos dinámicos (por ejemplo, de 1 a 16 bits) para cada componente de color. Ejemplos de aplicaciones que pueden utilizar esta característica incluyen documentos compuestos con imágenes y texto, imágenes médicas con superposiciones de anotaciones, e imágenes gráficas y generadas por computadora con regiones binarias y cercanas a ellas, planos alfa y de transparencia.
- **Compresión sin pérdida y con pérdida:** Se desea proporcionar compresión sin pérdida de forma natural en el curso de la decodificación progresiva. Ejemplos de aplicaciones que pueden utilizar esta característica incluyen imágenes médicas, donde la pérdida no siempre se tolera; aplicaciones de archivo de imágenes, donde la más alta calidad es vital para la conservación, pero no es necesario para la visualización; aplicaciones de red que suministran dispositivos con diferentes capacidades y recursos; e imágenes de preimpresión. También se desea que el estándar debe tener la propiedad de crear flujo de bits integrado y permitir la pérdida progresiva a la acumulación sin pérdida.
- **Transmisión progresiva por precisión y resolución de píxeles:** La transmisión progresiva que permite reconstruir imágenes con mayor precisión de píxeles o resolución espacial es esencial para muchas aplicaciones, como la navegación web, el archivo de imágenes y la impresión.
- **Codificación de la región de interés (ROI):** A menudo hay partes de una imagen que son de mayor importancia que otras. Esta característica permite a los usuarios definir ciertas ROIs en la imagen a codificar y transmitir en una mejor calidad y menos distorsión que el resto de la imagen.
- **Arquitectura abierta:** Es deseable permitir la arquitectura abierta para optimizar

el sistema para diferentes tipos de imágenes y aplicaciones. Con esta característica, sólo se necesita un decodificador para implementar el conjunto de herramientas principal y el analizador que entiende el flujo de código.

- **Robustez a errores de bit:** Es deseable considerar robustez a errores de bit mientras se diseña el flujo de código. Una aplicación donde esto es importante es la transmisión a través de canales de comunicación inalámbricos. Partes del flujo de código pueden ser más importantes que otras para determinar la calidad de imagen decodificada. Un diseño adecuado del flujo de código puede ayudar a los sistemas de corrección de errores posteriores a aliviar fallos de decodificación catastróficos.
- **Protección de la imagen:** La protección de una imagen digital se puede lograr mediante diferentes enfoques, como el marcado por agua, el etiquetado, el estampado o el cifrado. Los archivos de imagen JPEG 2000 deben tener provisiones para tales posibilidades.

La introducción del estándar de compresión JPEG 2000 significó que por primera vez la transformada de ondeleta discreta (DWT, por sus siglas en inglés) se utilizó para la descomposición y reconstrucción de imágenes junto con un esquema de codificación eficiente. El uso de DWT implica el uso de la codificación de subbandas en la que la imagen se descompone iterativamente en bandas de alta y baja frecuencia. Por lo tanto, se necesitan pares de filtros tanto en las etapas de análisis como de síntesis (Zhu, 2002).

### **3.4.3. JPEG XR (JXR)**

JPEG XR es un estándar de codificación de imágenes del comité JPEG que fue creado y adoptado por Microsoft. Se dirige principalmente a la representación de imágenes fijas en tono continuo, como las imágenes fotográficas, y logra una alta calidad de imagen, a la par con JPEG 2000, mientras que requiere recursos computacionales y capacidad de almacenamiento bajos. Además, responde eficazmente a las necesidades de las nuevas aplicaciones de imágenes de alto rango dinámico al incluir soporte para una amplia gama de formatos de representación de imágenes (Dufaux *et al.*, 2009).

JPEG XR es muy parecido al JPEG tradicional, un diseño basado en bloques que sigue los principios de diseño tradicionales que se encuentran en muchos esquemas de compresión de imágenes. Mientras que JPEG utiliza una DCT clásica de 8x8 píxeles y JPEG 2000 usa una transformada ondeletas, JPEG XR emplea una transformación de bloque superpuesto ortogonal de 4x4 píxeles. La relación entre la posición de un coeficiente dentro de la matriz 4x4 y la frecuencia espacial que representa difiere, sin embargo, de la disposición de la DCT (Richter, 2008).

JPEG XR está diseñado para un uso amplio en una amplia gama de aplicaciones de manejo de imágenes digitales y fotografía digital. Las principales áreas de aplicación incluyen las siguientes:

- Tecnologías de adquisición de imágenes robustas y de alta fidelidad, como una amplia gama de aplicaciones de cámaras. Utilizando JPEG XR, se puede conservar una representación más precisa del rango completo de la fidelidad de la señal de imagen capturada, evitar el cuello de botella introducido por el antiguo estándar de referencia JPEG y evitar la alta demanda de almacenamiento y las dificultades de interoperabilidad asociadas con los formatos de imagen cruda de la cámara.
- Flujos de trabajo de imágenes de alto rango dinámico, incluyendo paquetes de software de edición y tuberías de procesamiento de imágenes de alta capacidad. La capacidad de JPEG XR para retener un rango dinámico extendido con un alto grado de fidelidad de la señal puede ayudar a evitar la pérdida de calidad en entornos de flujo de trabajo de extremo a extremo.
- Entornos de procesamiento de señales restringidos computacionalmente, como aplicaciones móviles e integradas. Como las demandas computacionales asociadas con las imágenes JPEG XR son sustancialmente menores que con otros sistemas de codificación como JPEG 2000 que también proporcionan una alta capacidad de calidad.

#### **3.4.4. WebP**

WebP es una tecnología de compresión de imágenes propuesta por Google. Se deriva del formato de codificación de vídeo VP8; también admite compresiones con

pérdida y sin pérdida. En comparación con los formatos de imagen populares, WebP tiene un volumen menor que JPG y PNG en las mismas condiciones de calidad de imagen (Si y Shen, 2016).

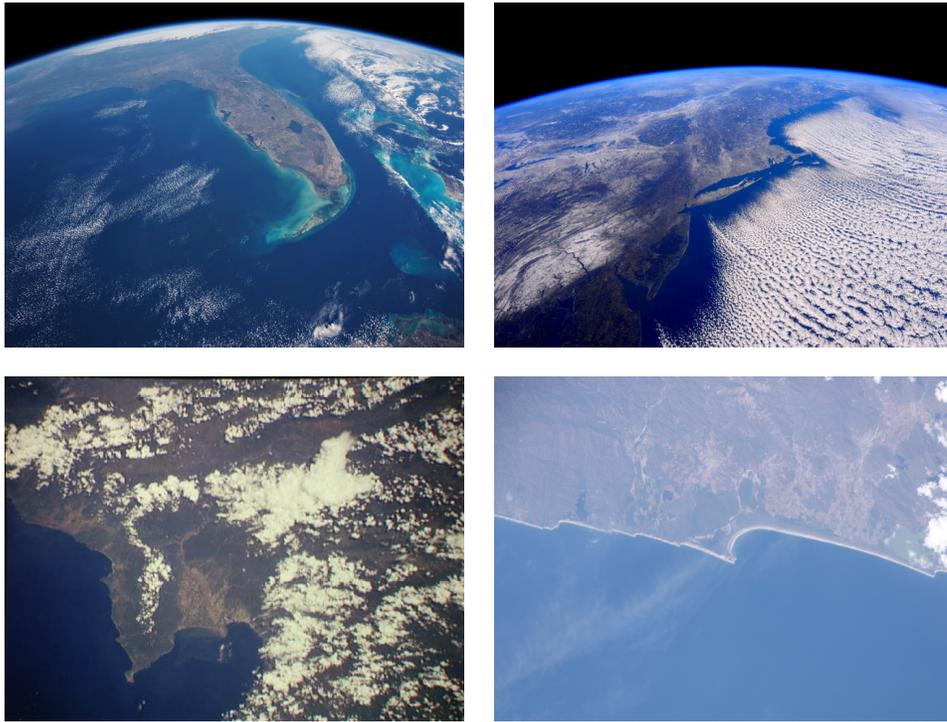
Al igual que muchos esquemas modernos de compresión de video, VP8 se basa en la descomposición de marcos en bloques cuadrados de píxeles, la predicción de tales bloques utilizando bloques previamente construidos, y el ajuste de tales predicciones (así como la síntesis de bloques impredecibles) utilizando una DCT. En un caso especial, sin embargo, VP8 utiliza una transformada de Walsh-Hadamard (WHT, por sus siglas en inglés) en lugar de un DCT.

Ocasionalmente, a velocidades de datos muy bajas, un sistema de compresión puede decidir reducir la resolución de la señal de entrada para facilitar una compresión eficiente. El formato de datos VP8 admite esto a través de la ampliación opcional de su búfer de reconstrucción interna antes de la salida. Esta ampliación restaura los fotogramas de video a su resolución original. En otras palabras, el sistema de compresión/descompresión se puede ver como una “caja negra”, donde la entrada y la salida están siempre a una resolución dada. El compresor puede decidir “hacer trampa” y procesar la señal a una resolución más baja. En ese caso, el descompresor necesita la capacidad de restaurar la señal a su resolución original.

Internamente, VP8 descompone cada marco de salida en una matriz de macrobloques. Un macrobloque es una matriz cuadrada de píxeles cuyas dimensiones Y son 16x16 y cuyas dimensiones U y V son 8x8. Los datos a nivel de macrobloque en un marco comprimido se producen (y deben procesarse) en un orden raster similar al de los píxeles que componen el marco (Bankoski *et al.*, 2011).

Utilizando WebP, los desarrolladores web pueden crear imágenes más pequeñas y ricas que hacen que la web sea más rápida. Las imágenes WebP sin pérdidas son 26% más pequeñas en tamaño en comparación con PNG. Las imágenes con pérdida WebP son un 25-34% más pequeñas que las imágenes JPEG comparables con un índice de calidad SSIM equivalente. WebP sin pérdida soporta transparencia (también conocido como canal alfa) a un costo de sólo 22% de bytes adicionales. Para los casos en que la compresión RGB con pérdida es aceptable, la WebP con pérdida también es compatible con transparencia, por lo general proporcionando archivos de tamaño 3 veces más

pequeños en comparación con PNG (Alakuijala y Rabaud, 2017).



**Figura 5.** Ejemplos de imágenes en la base de datos.

### 3.5. Base de datos de imágenes

Para realizar la evaluación de los formatos de compresión de imágenes se utilizó un lote de 133 imágenes como las mostradas en la Figura 5. Para obtener estas imágenes, a dos estudiantes de la carrera de Ingeniería Electrónica de la UABC que realizaron su servicio social en el CICESE, se les solicitó hacer una búsqueda en internet de imágenes tomadas por satélites pequeños a aproximadamente 500 km de altura. Muchas de las imágenes que forman la base de datos fueron tomadas por astronautas a bordo de la Estación Espacial Internacional. Se buscó que las imágenes fueran de muy buena calidad y de gran tamaño. Todas las imágenes originales seleccionadas tenían una resolución superior a 1000x1000 píxeles y fueron reescaladas a tener una resolución horizontal de 640 píxeles. La resolución vertical de la imagen fue seleccionada automáticamente durante el proceso de reescalamiento, de manera que no se alterara de ninguna forma la razón de aspecto de la imagen original. Se eligió como formato de salida PNG ya que es un formato sin pérdida.

### 3.6. Métricas de Calidad

Para hacer una comparación justa entre los formatos de compresión de imágenes a utilizar, es indispensable que la calidad de las imágenes sea muy similar entre cada uno de los formatos, por lo que se utilizaron distintas métricas de calidad para alcanzar este objetivo.

Las métricas de calidad son particularmente importantes para los algoritmos de compresión de imágenes con pérdida porque los datos reconstruidos difieren de los originales. Por ejemplo, si no hay diferencia perceptible entre los datos reconstruidos y los originales, se puede afirmar que el algoritmo de compresión alcanza una calidad muy alta. A la diferencia entre los datos reconstruidos y los originales se le llama distorsión. Uno espera tener una mayor calidad de los datos reconstruidos, si la distorsión es menor. Las medidas de calidad pueden ser muy subjetivas basadas en la percepción humana, pero pueden definirse objetivamente utilizando evaluaciones matemáticas o estadísticas (Eckert y Bradley, 1998).

#### 3.6.1. Métrica de calidad objetiva

No hay una métrica universalmente aceptada para la calidad objetiva de los algoritmos de compresión de datos. Para la medición objetiva, las métricas de calidad más utilizadas son el error cuadrático medio (RMSE, por sus siglas en inglés), la relación señal-ruido (SNR, por sus siglas en inglés), el error absoluto medio (MAE, por sus siglas en inglés), la relación pico de Señal a Ruido (PSNR, por sus siglas en inglés) y el método del índice de similitud estructural (SSIM, por sus siglas en inglés) (Acharya y Tsai, 2005). Si  $I$  es una imagen  $M \times N$  y  $\bar{I}$  es la imagen reconstruida correspondiente después de la compresión y descompresión, RMSE se calcula por:

$$RMSE = \sqrt{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [I(i, j) - \bar{I}(i, j)]^2}$$

donde  $i, j$  se refiere a la posición del píxel en la imagen.

El SNR en unidad de decibelios (dB) se expresa como:

$$SNR = 20 \log_{10} \left( \frac{\sqrt{\sum_{i=1}^M \sum_{j=1}^M I^2(i, j)}}{RMSE} \right) = 10 \log_{10} \left( \frac{\sum_{i=1}^{MN} \sum_{j=1}^2 I(i, j)}{\sum_{i=1}^M \sum_{j=1}^M [I(i, j) - \bar{I}(i, j)]^2} \right).$$

En caso de una imagen codificada usando 8 bits por píxel, el PSNR correspondiente en dB se calcula como:

$$PSNR = 20 \log_{10} \left( \frac{255}{RMSE} \right)$$

Cabe señalar que un RMSE inferior (o equivalente, SNR o PSNR superior) no indica necesariamente una calidad subjetiva superior. Estas métricas de error objetivo no siempre se correlacionan bien con las métricas de calidad subjetiva. Hay muchos casos donde la PSNR de una imagen reconstruida puede ser razonablemente alta, pero la calidad subjetiva es realmente mala cuando es visualizada por los ojos humanos. Esta situación particular se presentará en la Sección 4.4. Por lo tanto, la elección de las métricas objetivas o subjetivas para evaluar un algoritmo de compresión y descompresión a menudo dependen de los criterios de aplicación (Acharya y Tsai, 2005).

El SSIM es un modelo basado en la percepción. En este método, la degradación de la imagen se considera como el cambio de percepción en la información estructural. También se relaciona con otros hechos basados en la percepción como el enmascaramiento de luminancia, el enmascaramiento de contraste, etc. El término *información estructural* enfatiza sobre los píxeles fuertemente interdependientes o los píxeles cerrados espacialmente. Estos píxeles fuertemente interdependientes refieren alguna información más importante sobre los objetos visuales en el dominio de la imagen. Enmascaramiento de luminancia es un término donde la parte de distorsión de una imagen es menos visible en los bordes de una imagen. Por otro lado, *enmascaramiento de contraste* es un término donde las distorsiones también son menos visibles en la textura de una imagen (Sara et al., 2019). SSIM estima la calidad percibida de imágenes y videos. Mide la similitud entre dos imágenes: la original y la recuperada.

Hay una versión avanzada de SSIM llamada Multi Scale Structural Similarity Index Method (MS-SSIM, por sus siglas en inglés) que evalúa varias imágenes de similitud

estructural a diferentes escalas de imágenes (Wang *et al.*, 2003). En MS-SSIM, dos imágenes se comparan con la escala del mismo tamaño y resoluciones. Al igual que el SSIM, el cambio de luminancia, contraste y estructura se considera para calcular la similitud estructural de múltiples escalas entre dos imágenes (Dosselmann y Yang, 2011). A veces da un mejor rendimiento sobre SSIM en diferentes bases de datos de imágenes y videos subjetivos.

Otra versión de SSIM, llamada SSIM de tres componentes (3-SSIM, por sus siglas en inglés) que opera considerando el hecho de que el sistema visual humano observa las diferencias con mayor precisión en las regiones texturizadas que las regiones lisas. Este modelo SSIM de 3 componentes fue propuesto por Ran y Farvardin (Li y Bovik, 2009) donde una imagen se desintegra en tres propiedades importantes como el borde, la textura y la región lisa. La métrica resultante se calcula como un promedio ponderado de similitud estructural para estas tres categorías. Las estimaciones de medición de peso propuestas son 0.5 para los bordes, 0.25 para la textura y 0.25 para las regiones lisas. También se puede mencionar que una medición de peso 1/0/0 influye en los resultados para estar más cerca de las calificaciones subjetivas. Esto puede implicar que no hay texturas ni regiones lisas, sino que las regiones de borde juegan un papel dominante en la percepción de la calidad de imagen.

### **3.6.2. Métrica de calidad subjetiva**

A menudo la métrica de calidad subjetiva se define como la puntuación media de los observadores (MOS, por sus siglas en inglés) (Khan *et al.*, 2008). A veces, también se llama puntuación media de opinión. Hay diferentes formas estadísticas de calcular la MOS. Una de las maneras más simples consiste en elegir al azar un número estadísticamente significativo de observadores para evaluar la calidad visual de las imágenes reconstruidas. Todas las imágenes son comprimidas y descomprimidas por el mismo algoritmo. Cada observador asigna una puntuación numérica a cada imagen reconstruida basada en su percepción de la calidad de la imagen, digamos dentro de un rango de 1-5 para describir la calidad de la imagen siendo 5 la calidad más alta y siendo 1 la peor calidad. El promedio de las puntuaciones asignadas por todos los observadores a las imágenes reconstruidas se denomina puntuación media del observador (MOS) y puede considerarse como una métrica subjetiva viable si todos los

observadores evalúan las imágenes bajo la misma condición de visualización (Acharya y Tsai, 2005). Las técnicas de medición de la MOS bien podrían ser distintas para diferentes datos perceptivos. La metodología para evaluar la calidad subjetiva de una imagen fija podría ser completamente diferente para datos de video o voz. MOS se calcula sobre la base de la calidad percibida de los datos reconstruidos por un número estadísticamente significativo de observadores humanos.

Además de las métricas de calidad mencionadas, es necesario evaluar otros parámetros que son importantes para lograr una comparación completa entre los formatos de compresión, estos parámetros son: tasa de compresión, complejidad numérica del formato de compresión y robustez a la pérdida de bits.

### **3.7. Complejidad numérica**

La complejidad computacional o complejidad numérica de un algoritmo es la cantidad de recursos necesarios para ejecutarlo, principalmente se refiere a los requisitos de tiempo y memoria. En este trabajo se presta atención a la complejidad numérica de cada uno de los formatos de compresión, especialmente al tiempo de ejecución de cada uno de ellos, pues en ocasiones específicas puede ser de vital importancia descargar imágenes satélites en el menor tiempo posible, por ejemplo cuando se tiene un tiempo reducido de línea de vista entre el satélite y la estación terrestre.

### **3.8. Tasa de Compresión**

Con el incremento en el volumen de datos de imagen, el uso de la compresión con pérdida está ganando cada vez más apoyo. Explorar el nivel de compresión aceptable ha sido un tema importante en la compresión de imágenes con pérdida. La tasa de compresión se define como la relación entre el número de bits en una imagen original y el número de bits en su versión comprimida (Kim *et al.*, 2008). La tasa de compresión es indispensable y la medida más práctica para determinar y comunicar el nivel de compresión.

### **3.9. Robustez a errores de bits**

Debido a que el objetivo de este proyecto es evaluar distintos formatos de compresión de datos para la descarga de datos a través de canales de comunicación inalámbricos, es necesario considerar la robustez a errores de bit en cada uno de los formatos, pues esto puede ayudar a los sistemas de corrección de errores posteriores a aliviar fallos de decodificación catastróficos.

## Capítulo 4. Evaluación del desempeño de los formatos de compresión

---

En este capítulo se presentan las pruebas realizadas para evaluar los formatos de compresión, para esto se utilizaron diversos programas y librerías que nos permiten manejar y convertir todos los formatos de compresión a evaluar, tales como:

- **MATLAB.** Es un lenguaje de alto rendimiento para la computación técnica. Integra computación, visualización y programación en un entorno fácil de usar donde los problemas y soluciones se expresan en notación matemática familiar (Matlab, 2012). En entornos universitarios, es la herramienta instructiva estándar para cursos introductorios y avanzados en matemáticas, ingeniería y ciencias. En la industria, MATLAB es la herramienta elegida para la investigación, el desarrollo y el análisis de alta productividad.
- **Simulink.** Es un programa complementario de MATLAB, es un sistema interactivo para simular sistemas dinámicos no lineales. Es un programa gráfico dirigido por el ratón de computadora (mouse) que permite modelar un sistema dibujando un diagrama de bloques en la pantalla y manipulándolo dinámicamente. Puede funcionar con un sistema lineal, no lineal, de tiempo continuo, tiempo discreto, multivariable y multitasa.
- **ImageMagick.** Es un software libre que se entrega como una distribución binaria lista para funcionar, o como código fuente que puede usar, copiar, modificar y distribuir en aplicaciones tanto abiertas como propietarias. Se distribuye bajo una licencia derivada de Apache 2.0. ImageMagick lee y escribe más de 200 formatos de imagen; también utiliza múltiples hilos computacionales para aumentar el rendimiento y puede leer, procesar o escribir tamaños de imagen de imagen de mega-, giga- o tera-píxeles. La versión actual es ImageMagick 7.1.0-5. Se ejecuta en Linux, Windows, Mac Os X, iOS, Android OS, y otros (Still, 2006).
- **XnView MP.** Es un organizador de imágenes y administrador de archivos de propósito general usado para visualizar, convertir, organizar y editar imágenes, así como para la administración de archivos de propósito general. XnView puede leer más de 500 formatos de imágenes, algunos archivos de audio y de video, y

escribir en 50 formatos de imagen. Viene con herramientas integradas para inspección hexadecimal, renombrado de archivos y captura de pantalla. Se licencia como freeware para uso privado, educacional y sin ánimo de lucro. Para otros usos, se licencia como software comercial (XnView, 2021).

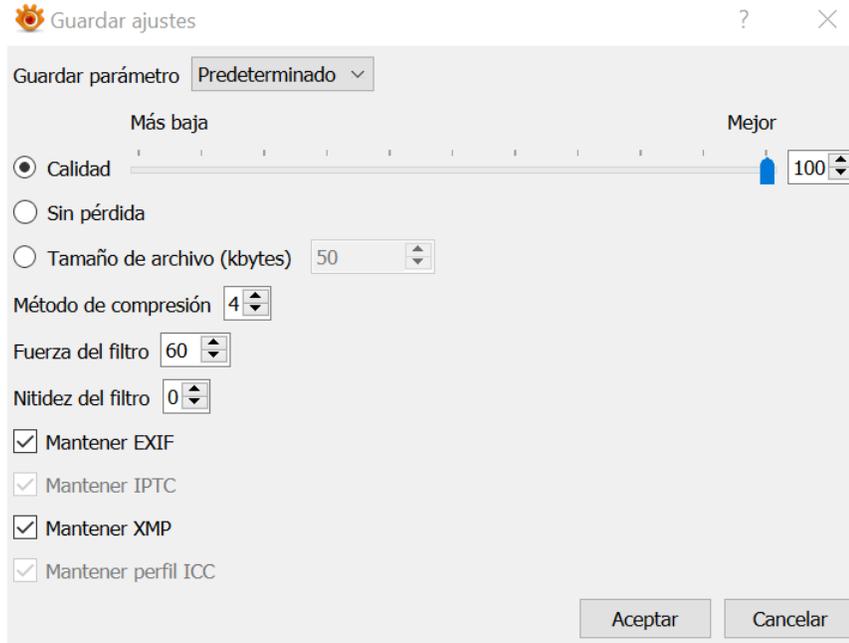
- *nconvert*. Es un procesador de imagen de código cerrado que admite múltiples formatos de imagen. Maneja archivos del formato Portable Network Graphics (PNG) con la biblioteca de código abierto enlazada estáticamente *libpng* (Hu *et al.*, 2019).
- *Jxrlib*. Es una biblioteca de JPEG XR Image Codec publicada por Microsoft bajo licencia BSD-2-Clause. El formato JPEG XR, que soporta el códec, está diseñado para la fotografía digital de consumo y profesional.
- *LibWebp*. Es una implementación de referencia para las aplicaciones de codificación y decodificador de imágenes desarrollado por Google. La herramienta de comandos *cwebp* codifica imágenes en formato JPEG, PNG o TIFF a WebP, mientras que *dwebp* las decodifica de nuevo a PNG.

A continuación se presentan las diversas pruebas realizadas a cada uno de los formatos de compresión.

#### **4.1. Selección de formatos de compresión**

Utilizando XnView, se hizo un análisis de los ajustes que se permiten en los formatos de compresión. Entre estos ajustes se encuentran calidad, tamaño del archivo comprimido, tasa de compresión, método de compresión, entre otros. Como ya lo mencionamos en la Sección 4.1, decidimos evaluar 4 formatos de compresión de imágenes que cuentan parámetros de ajustes similares para la compresión de imágenes. Entre estos se encuentra un parámetro llamado Calidad, este ajuste permite seleccionar la calidad deseada en la compresión y tiene valores que van desde 0 a 100 como se muestra en la Figura 6. Los formatos seleccionados y que cuentan con este ajuste son: JPEG, JPEG 2000, JPEG XR y WebP, por lo que son los formatos que se utilizarán. Este parámetro de calidad es independiente para cada uno de los formatos, en la Figura 7 se muestra un ejemplo de como va cambiando la calidad de una imagen originalmente

en formato PNG al convertirla a formato JPEG usando XnView con diferentes valores de calidad.



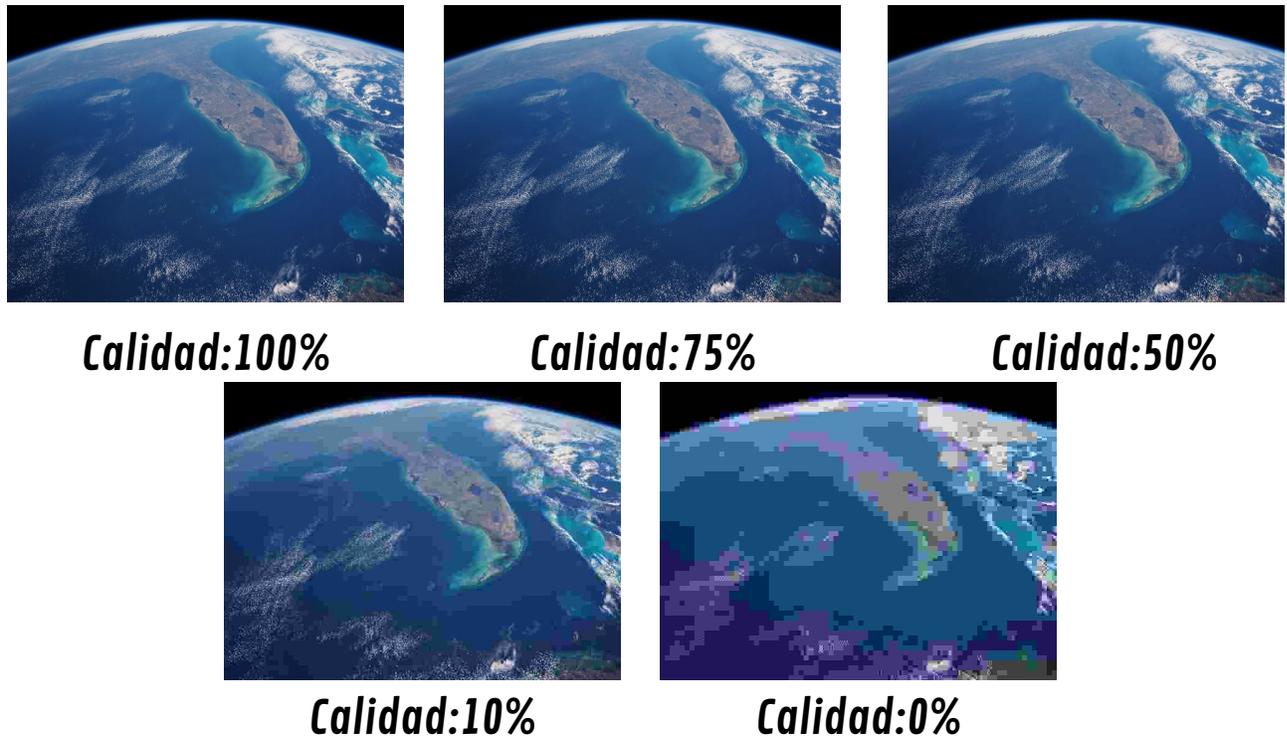
**Figura 6.** Ajustes de calidad de compresión disponibles en XnView MP.

Una vez que los formatos fueron elegidos, se escribió un programa bajo MATLAB para obtener la relación señal a ruido pico (PSNR), la relación señal a ruido (SNR) y el índice de similitud estructural (SSIM) como métricas de calidad objetivas para asegurar que las todas las imágenes contaban con una calidad similar. Estas métricas de calidad se definieron en la sección 3.6.1.

Para hacer este análisis se utilizó un lote de 133 imágenes como las mostradas en la Figura 5. Todas las imágenes seleccionadas originales tenían una resolución superior a 1000x1000 píxeles y fueron reescaladas a tener una resolución horizontal de 640 píxeles. La resolución vertical de la imagen fue seleccionada automáticamente durante el proceso de reescalamiento, de manera que no se alterara de ninguna forma la razón de aspecto de la imagen original. Se eligió como formato de salida PNG ya que es un formato de compresión sin pérdida.

#### 4.2. Pruebas de calidad objetiva

Como primera prueba, las imágenes fueron convertidas a cada uno de los 4 formatos elegidos utilizando XnView y ajustando la calidad al 100, después se hizo un



**Figura 7.** Ejemplo de compresión con distintos valores de calidad en XnView, utilizando el formato JPG.

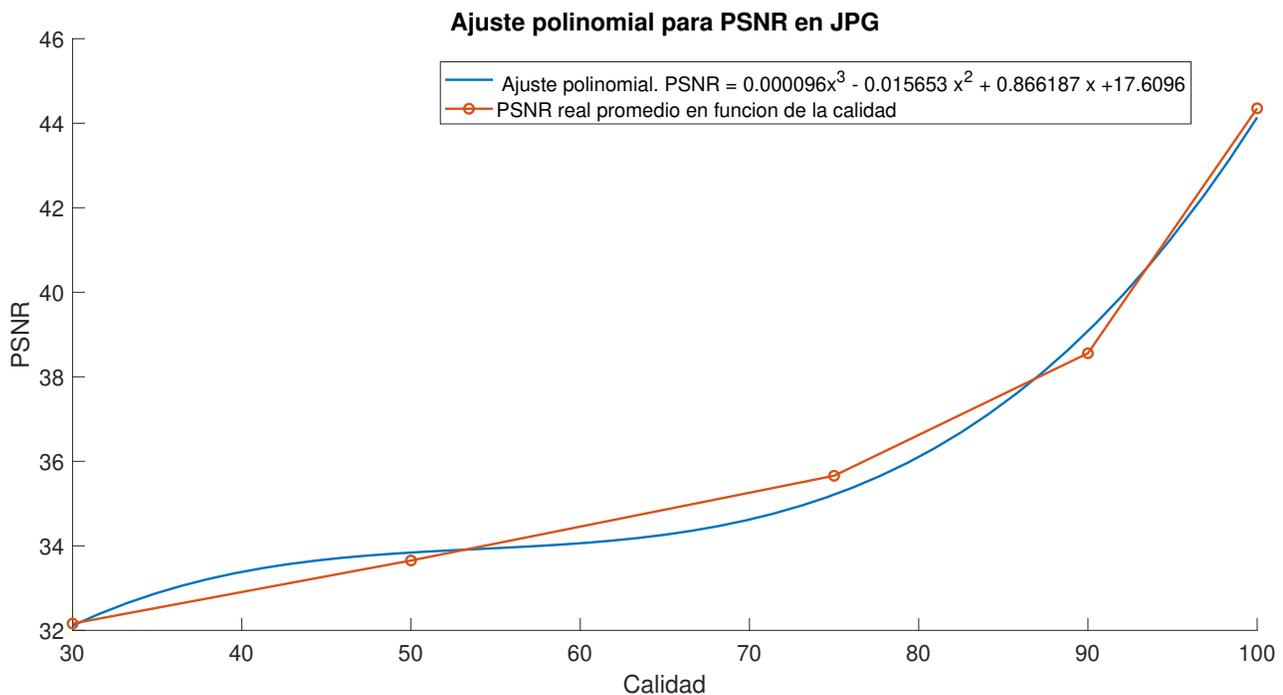
análisis de su PSNR, SNR y SSIM promedio. Los resultados se presentan en la Figura 8. Como se puede ver, utilizando la calidad de 100, cada uno de los formatos cuenta con valores muy distintos para su PSNR y SNR, por lo que se necesitó hacer un ajuste polinomial para cada uno de estos formatos con el objetivo de obtener una ecuación que nos permita estimar los valores promedio de PSNR, SNR Y SSIM a partir de una selección específica de calidad al momento de hacer la conversión de imágenes.

Para realizar el ajuste polinomial, se utilizaron los ajustes de calidad 100, 90, 75, 50 y 30, para cada uno de los 4 formatos de compresión utilizados. Una vez que se tuvieron las imágenes comprimidas a estas calidades, se obtuvo su PSNR promedio, SNR promedio, SSIM promedio y desviación estándar de cada uno de estos parámetros. Con estos datos se realizó un ajuste polinomial de 3er orden para cada uno de los formatos de compresión. En las Figuras 9, 10, 11 y 12 se muestran las gráficas obtenidas para el PSNR promedio en cada uno de los 4 formatos de compresión. Una vez obtenido este ajuste y su ecuación correspondiente, fue posible ajustar todos los formatos a una calidad similar, dependiendo de la métrica de calidad utilizada.

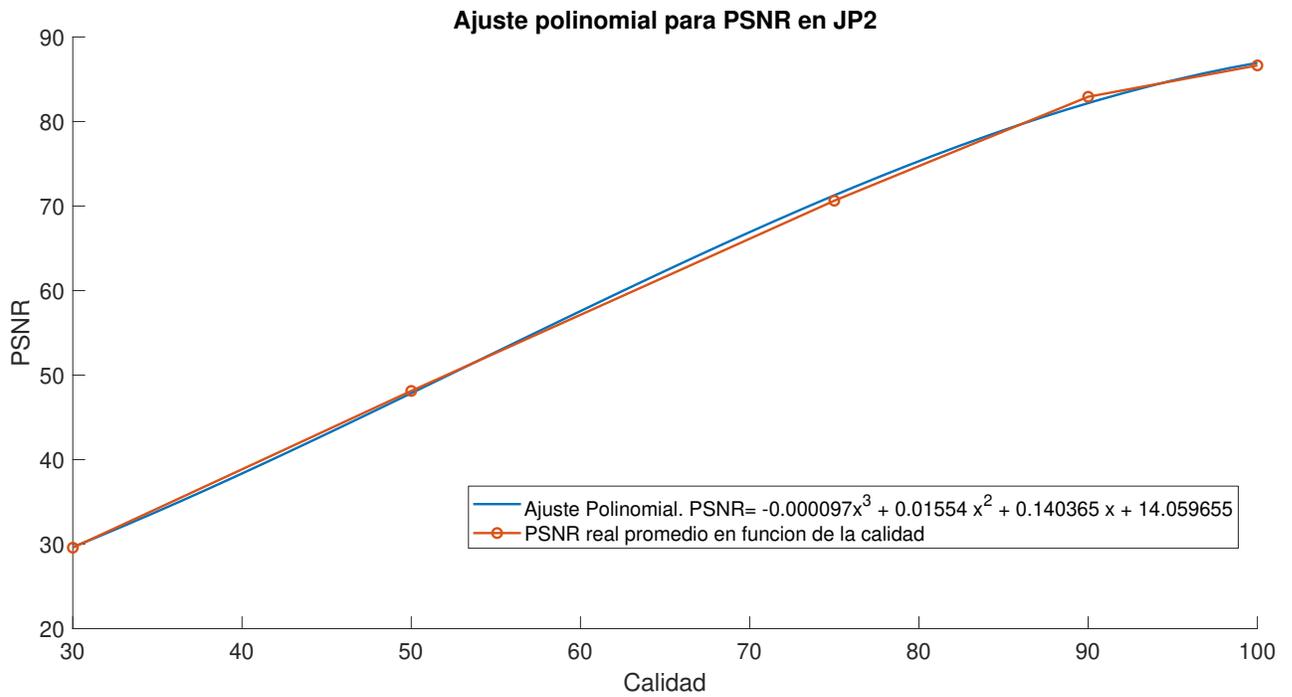
Existen básicamente dos enfoques para medir la calidad de una imagen, medidas

FORMATO	PSNR PROMEDIO (dB)	SNR PROMEDIO (dB)	SSIM PROMEDIO
<b>JPEG</b> <i>100 Calidad</i>	<b>44.3524</b> +/- 3.8084	<b>37.7702</b> +/- 3.9643	<b>0.9952</b> +/- 0.0038
<b>JP2</b> <i>100 Calidad</i>	<b>86.6385</b> +/- 4.9061	<b>80.0562</b> +/- 5.3337	<b>1 +/-</b> <b>0.000004201</b>
<b>JXR</b> <i>100 Calidad</i>	<b>55.074</b> +/- 0.8793	<b>48.4918</b> +/- 1.5929	<b>0.9995 +/-</b> <b>0.00038559</b>
<b>WebP</b> <i>100 Calidad</i>	<b>42.6182</b> +/- 2.8772	<b>36.036</b> +/- 3.1402	<b>0.9927 +/-</b> <b>0.0046</b>

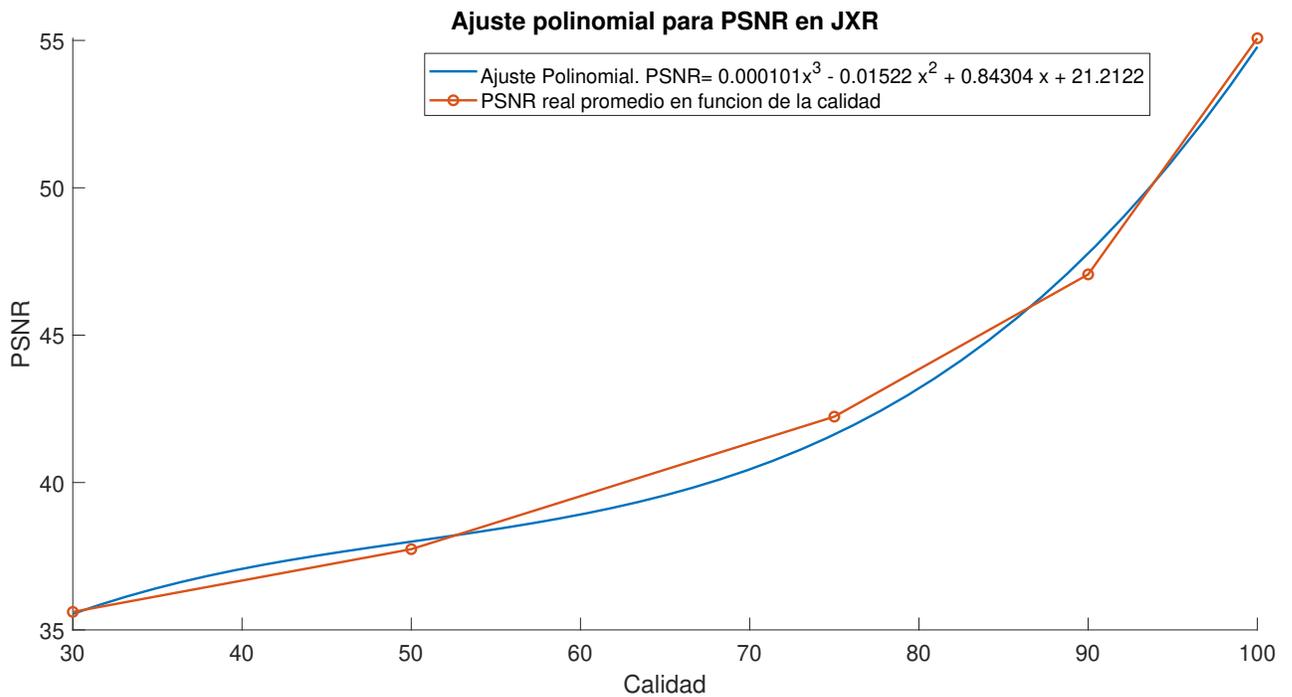
**Figura 8.** Valores promedio de PSNR, SNR Y SSIM para cada uno de los formatos con calidad 100.



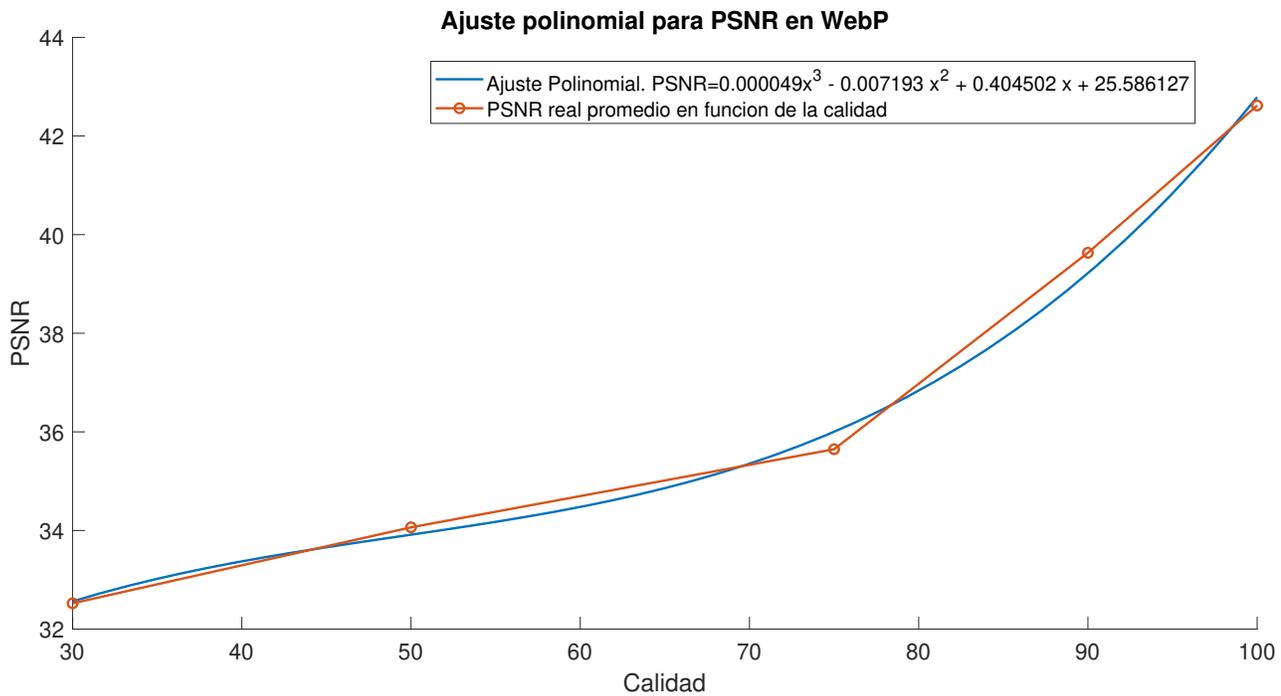
**Figura 9.** Ajuste polinomial para PSNR en JPG.



**Figura 10.** Ajuste polinomial para PSNR en JP2.



**Figura 11.** Ajuste polinomial para PSNR en JXR.



**Figura 12.** Ajuste polinomial para PSNR en WebP.

de calidad subjetivas y medidas de calidad objetivas. Las métricas de calidad subjetivas siempre captan el factor crucial, la impresión del usuario viendo la imagen. Sin embargo estas métricas son muy costosas y altamente consumidoras de tiempo. La impresión de calidad para un humano suele darse en una escala de 5 (mejor) a 1 (peor) como se muestra en la Figura 13. Esta escala se denomina puntuación de opinión media (MOS) (Khan *et al.*, 2008).

Muchas tareas en la industria y la investigación requieren métodos automatizados para evaluar la calidad de las imágenes. Las pruebas subjetivas costosas y complejas a menudo no pueden permitirse. Por lo tanto, se han desarrollado métricas objetivas para emular la impresión de calidad del sistema visual humano (HVS). El método más extendido es el cálculo de la relación de señal a ruido pico (PSNR) imagen por imagen (Klaue *et al.*, 2003).

Basándonos en estos estudios y en la Figura 13, decidimos utilizar un PSNR promedio de 38 dB para simular una calidad excelente en la escala MOS y posteriormente hicimos este mismo procedimiento para obtener imágenes con un PSNR promedio de 30 dB, correspondiente a una calidad aceptable.

Una vez definidos los valores de PSNR promedio a utilizar, se hizo el cálculo para

<b>PSNR (dB)</b>	<b>MOS</b>	<b>Calidad</b>	<b>Deterioro</b>
<b>&gt;37</b>	<b>5</b>	<b>Excelente</b>	<b>imperceptible</b>
<b>31-36.9</b>	<b>4</b>	<b>Buena</b>	<i>Perceptible, pero no molesto</i>
<b>25-30.9</b>	<b>3</b>	<b>Aceptable</b>	<i>Ligeramente molesto</i>
<b>20-24.5</b>	<b>2</b>	<b>Escasa</b>	<b>Molesto</b>
<b>&lt;19.9</b>	<b>1</b>	<b>Mala</b>	<b>Muy molesto</b>

**Figura 13.** Conversión de PSNR a MOS, tomado de Khan *et al.* (2008).

obtener la calidad de compresión necesaria para cada formato utilizando las ecuaciones obtenidas de las Figuras 9, 10, 11 y 12. Los resultados para 38 dB fueron los siguientes:

- JPEG: 87 % de calidad.
- JP2: 40 % de calidad.
- JXR: 50 % de calidad.
- WEBP: 85 % de calidad.

Sin embargo, al realizar la compresión de los archivos a estas calidades y posteriormente su análisis de las métricas de calidad, se observó que el PSNR promedio de estos formatos variaba entre 37.4 dB y 39 dB, por lo que se realizó un ajuste fino variando muy lentamente el factor de calidad para que el PSNR promedio de los formatos fuera lo más cercano a 38 dB. El mismo procedimiento se realizó para ajustar el PSNR promedio de las imágenes a 30 dB.

Teniendo todas nuestras imágenes comprimidas a los formatos utilizados y con el PSNR definido, se hizo un análisis de las métricas objetivas utilizadas en este trabajo y se agregó el tamaño promedio de los archivos comprimidos así como su tasa de compresión para hacer un análisis más a fondo de estos formatos. Los resultados se

FORMATO	PSNR PROMEDIO (dB)	TAMAÑO DE ARCHIVO PROMEDIO	TASA DE COMP. PROMEDIO	SNR PROMEDIO dB	SSIM PROMEDIO
<b>JPEG</b> <i>89 Calidad</i>	<b>38.2257</b> +/- 2.9012	<b>68.556 KB</b> +/- 21.2575 KB	<b>0.8371</b> +/-0.0272	<b>31.6434</b> +/- 3.1577	<b>0.9817</b> +/- 0.0102
<b>JP2</b> <i>39 Calidad</i>	<b>38.5672</b> +/- 0.1875	<b>87.382 KB</b> +/- 60.9642 KB	<b>0.8065</b> +/- 0.1089	<b>31.9849</b> +/- 1.5875	<b>0.9814</b> +/- 0.0095
<b>JXR</b> <i>53 Calidad</i>	<b>38.2236</b> +/- 1.7052	<b>43.388 KB</b> +/- 20.4149 KB	<b>0.8995</b> +/- 0.0331	<b>31.6414</b> +/- 2.1808	<b>0.9800</b> +/- 0.0098
<b>WebP</b> <i>86 Calidad</i>	<b>38.2999</b> +/- 2.0146	<b>44.849 KB</b> +/- 20.3685 KB	<b>0.8963</b> +/- 0.0315	<b>31.7177</b> +/- 2.4503	<b>0.9822</b> +/- 0.0086

**Figura 14.** Tabla comparativa de los parámetros de calidad para distintos formatos de compresión con un PSNR promedio de 38 dB.

FORMATO	PSNR PROMEDIO (dB)	TAMAÑO DE ARCHIVO PROMEDIO	TASA DE COMP. PROMEDIO	SNR PROMEDIO dB	SSIM PROMEDIO
<b>JPEG</b> <i>17 Calidad</i>	<b>30.3490</b> +/- 2.9012	<b>17.495 KB</b> +/- 4.8789 KB	<b>0.9578</b> +/- 0.0080	<b>23.7667</b> +/- 3.0203	<b>0.9092</b> +/- 0.0423
<b>JP2</b> <i>31 Calidad</i>	<b>30.6073</b> +/- 0.3417	<b>22.03 KB</b> +/- 25.9748 KB	<b>0.9537</b> +/- 0.0468	<b>24.0250</b> +/- 1.5805	<b>0.9241</b> +/- 0.0319
<b>JXR</b> <i>0 Calidad</i>	<b>31.5183</b> +/- 2.5118	<b>10.5107 KB</b> +/- 5.6196 KB	<b>0.9756</b> +/- 0.0097	<b>24.9360</b> +/- 2.8927	<b>0.9257</b> +/- 0.0379
<b>WebP</b> <i>10 Calidad</i>	<b>30.6329</b> +/- 2.6872	<b>7.4609 KB</b> +/- 3.9028 KB	<b>0.9827</b> +/- 0.0065	<b>24.0506</b> +/- 3.0792	<b>0.9125</b> +/- 0.0444

**Figura 15.** Tabla comparativa de los parámetros de calidad para distintos formatos de compresión con un PSNR promedio de 30 dB.

muestran en las Figuras 14 y 15.

### **4.3. Análisis de resultados para métricas de calidad objetiva**

En base a los valores obtenidos en las Figuras 14 y 15, se realizó un análisis para cada uno de sus parámetros.

#### **4.3.1. PSNR**

El PSNR promedio en los distintos formatos de imagen se ajustó de tal manera en que fuera lo más cercano a 38 dB y 30 dB respectivamente, sin embargo podemos observar que la desviación estándar en cada caso es muy diferente, mientras con JP2 podemos asegurar que el PSNR promedio se encuentra dentro de los valores seleccionados debido a que su desviación estándar es menor a 0.5 dB, no tenemos la misma certeza para el caso de JPG, pues con su desviación estándar de 2.9 dB no es posible asegurar que el PSNR promedio se encuentra dentro de los valores seleccionados. JXR y WebP tienen su desviación estándar entre 1.5 dB y 2.5 dB, por lo tanto también presentan cierta incertidumbre en el PSNR promedio seleccionado.

#### **4.3.2. Tamaño promedio y tasa de compresión**

Uno de los parámetros que más varía entre los formatos es el tamaño promedio de los archivos, que va desde 43 KB en el caso de JXR hasta 87.4 KB para JP2, ambos analizando la tabla de calidad Excelente (PSNR=38 dB). Este es un factor muy importante, debido a que tener imágenes de menor tamaño, nos permite aprovechar de mejor manera el ancho de banda en las comunicaciones. En este aspecto, debemos tomar en cuenta no solo el PSNR para definir la calidad de la imagen, sino también el tamaño promedio o la tasa de compresión promedio de estas imágenes. Un aspecto interesante de analizar es el caso particular del comportamiento de JP2, pues su desviación estándar en el tamaño promedio de los archivos es muy grande comparado con los demás formatos, por lo que se hizo un análisis más detallado observando el caso particular de cada imagen y se observó que en las imágenes altamente uniformes (por ejemplo mucho mar o mucho terreno desértico) la tasa de compresión es

muy elevada. En cambio si aparecen muchos detalles tipo nubes pequeñas esparcidas y terreno variado, la tasa de compresión puede caer incluso a la mitad, por lo que se concluyó que la variación tan grande en el tamaño de los archivos de JP2 está relacionada con la cantidad de colores y detalles presentes en la imagen. Es importante mencionar que este aspecto observado ocurre en todos los formatos de compresión, pero se ve amplificado en el formato de compresión de imágenes JP2.

### **4.3.3. SNR**

El comportamiento del SNR fue muy similar al del PSNR, solo que los valores rondan entre 31 dB para el caso de calidad excelente y 24 dB en el caso de calidad aceptable. El formato JPG fue el que tuvo mayor desviación estándar, al igual que con el PSNR. Algo interesante se dio en la desviación estándar, mientras que para los formatos JPG, JXR y WebP la desviación fue muy similar a la de PSNR, en el caso de JP2 de calidad excelente (PSNR=38 dB) su valor subió de 0.18 dB a 1.6 dB.

### **4.3.4. SSIM**

El SSIM se mantuvo en valores muy parecidos para todos los formatos de compresión, estando casi en el valor máximo (SSIM=0.98) para el caso de calidad excelente (PSNR=38 dB) con una desviación estándar muy pequeña. En el caso de calidad aceptable (PSNR=30 dB) el SSIM también fue muy similar entre todos los formatos con un valor aproximado de 0.915.

## **4.4. Pruebas de calidad subjetiva**

En conjunto con las pruebas de calidad objetiva, se realizó una pequeña evaluación subjetiva entre estudiantes del CICESE para comprobar si efectivamente al tener el mismo PSNR entre los diferentes formatos de compresión, tenemos una calidad subjetiva similar entre cada uno de ellos.

La prueba consistió en tomar 5 imágenes como las mostradas en la Figura 5, cada una de ellas tiene aspectos distintos como por ejemplo, detalles, colores, cantidad de

nubes o agua en la imagen, etc. Estas imágenes fueron convertidas a los formatos JPG, JP2, JXR y WebP con cada una de las calidades correspondientes de las Figuras 14 y 15, posteriormente se les asignó un nombre aleatorio. En las Figuras 16 y 17 se muestra el nombre asignado a cada imagen.

### PSNR =38dB

IMAGEN 6	IMAGEN 17	IMAGEN 29	IMAGEN 43	IMAGEN 72
A1 usar JP2	B1 usar JXR	C1 usar JP2	D1 usar JPG	E1 usar WBP
A2 usar WBP	B2 usar JP2	C2 usar JPG	D2 usar JXR	E2 usar JP2
A3 usar JPG	B3 usar JPG	C3 usar WBP	D3 usar JP2	E3 usar JXR
A4 usar JXR	B4 usar WBP	C4 usar JXR	D4 usar WBP	E4 usar JPG

Figura 16. Nombre aleatorio asignado a cada imagen con PSNR=38 dB.

### PSNR =30dB

IMAGEN 6	IMAGEN 17	IMAGEN 29	IMAGEN 43	IMAGEN 72
A1 usar JXR	B1 usar JXR	C1 usar JXR	D1 usar JP2	E1 usar JPG
A2 usar WBP	B2 usar WBP	C2 usar JPG	D2 usar WBP	E2 usar JXR
A3 usar JPG	B3 usar JPG	C3 usar JP2	D3 usar JPG	E3 usar WBP
A4 usar JP2	B4 usar JP2	C4 usar WBP	D4 usar JXR	E4 usar JP2

Figura 17. Nombre aleatorio asignado a cada imagen con PSNR=30 dB.

Teniendo las imágenes renombradas, el siguiente paso fue hacer una conversión de estas imágenes a formato PNG, esto con el propósito de que fuera imposible para el observador identificar qué formato de compresión estaba analizando. Posteriormente se mandaron las imágenes a los observadores, junto con una hoja de respuestas en la que se les indicaba qué debían analizar las imágenes y ordenarlas de mayor a menor calidad según su preferencia y anotar sus respuestas de acuerdo a un ejemplo mostrado. En la Figura 18 se muestra dicha hoja de respuestas enviada a cada observador.

En total se recibieron 8 respuestas y se hizo un análisis para observar cuántas coincidencias hubo entre los observadores y cuál formato fue el mejor calificado en cada caso. Para llevar a cabo este análisis se hizo una ponderación a cada una de las respuestas de los usuarios dando un valor de 4 puntos cada vez que un formato era

Evaluación Parte 1 usando PSNR de 38 dB					Evaluación Parte 2 usando PSNR de 30 dB					
Mayor calidad			Menor Calidad							
					Mayor calidad			Menor Calidad		
Imagen A										
Imagen B						Imagen B				
Imagen C						Imagen C				

**Figura 18.** Vista de la hoja de respuestas a llenar por cada observador.

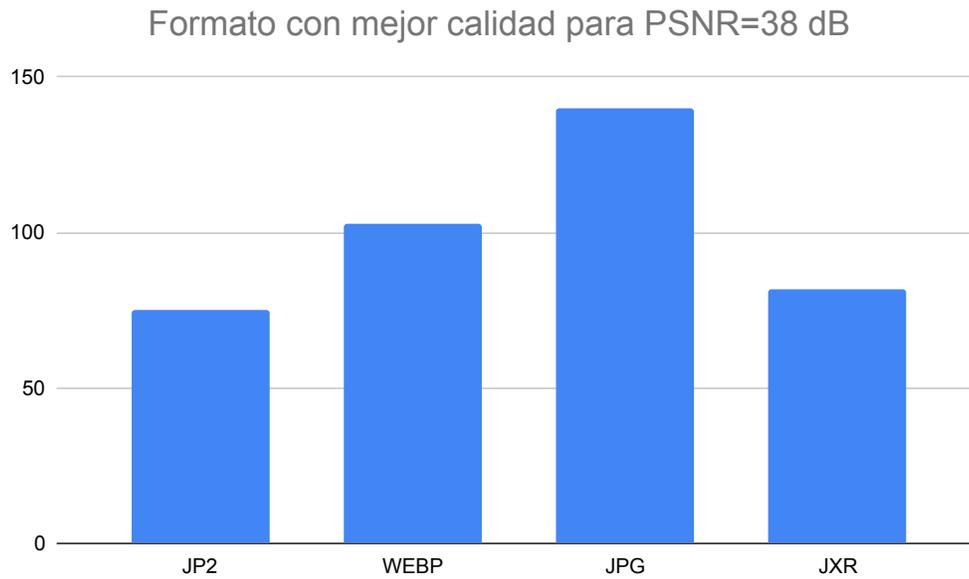
elegido como el de mayor calidad, 3 puntos cada vez que era elegido como el de una calidad media-alta, 2 puntos en el caso de que el formato se elegía de calidad media-baja y 1 punto cada vez que era elegido como el formato de peor calidad. El formato de compresión con más puntos se define como el de mayor calidad y el formato con menos puntos como el de peor calidad. Los resultados para PSNR=38 dB se muestran en la Figura 19 y para PSNR=30 dB son mostrados en la Figura 20. Con esto concluyeron las pruebas subjetivas de calidad para los formatos de compresión de imágenes a utilizar.

#### 4.5. Análisis de resultados para métricas de calidad subjetiva

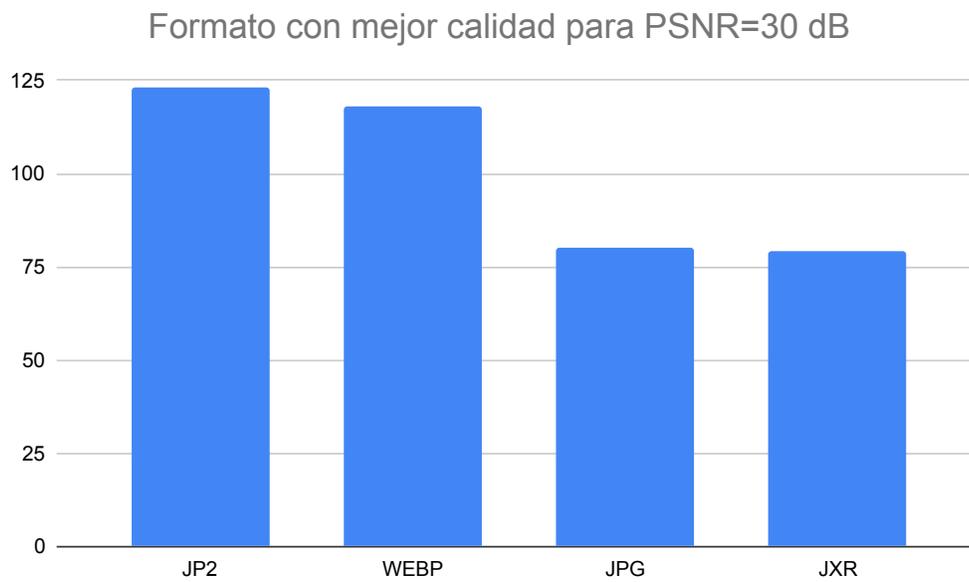
Se analizaron los resultados obtenidos por los observadores y registrados en las Figuras 19 y 20. De esto se pueden hacer las siguientes observaciones.

- Para el caso de calidad excelente (PSNR=38 dB):

En este caso las opiniones de los observadores fueron muy dispersas indicándonos que era muy difícil para ellos decidir cuál imagen tenía menor calidad, por lo que se puede concluir que todas tenían una calidad muy similar. Hubo más coincidencias al definir cuál era la imagen con peor calidad y en 3 de 5 casos se definió la imagen de formato JP2 como la de menor calidad. En el caso de la ima-



**Figura 19.** Formato con mejor calidad subjetiva para PSNR=38 dB.



**Figura 20.** Formato con mejor calidad subjetiva para PSNR=30 dB.

gen de mayor calidad hubo variación en las respuestas, pero en todos los casos el formato con más votos fue JPG, por lo que se concluye que es el formato que presenta mejor calidad subjetiva.

- Para el caso de calidad aceptable (PSNR=30 dB):

En este caso los observadores reportaron que fue más fácil observar las diferencias entre cada imagen y así tomar la decisión de cuál consideran de mejor o peor calidad.

Al definir la imagen con mayor calidad hubo más variación entre las respuestas de los observadores, pero se eligió a JP2 como la imagen con mejor calidad por la mayoría de los observadores, caso contrario a las imágenes catalogadas de calidad excelente, donde JP2 fue seleccionado por los observadores como el formato con peor calidad. Al definir la imagen de peor calidad, la mayoría de los observadores coincidió en su decisión, sin embargo en cada uno de los casos esta decisión correspondía a un formato distinto. Esto nos indica que no hay un solo formato “perdedor” en cuestión de calidad subjetiva, pues dependiendo de los detalles de la imagen se puede observar un formato u otro como el de peor calidad.

Podemos concluir que todos los formatos tienen en general una calidad muy similar y que no es posible tomar una decisión en cuanto a cuál es el de mejor o peor calidad, pues depende mucho de cada imagen y sus detalles. Por lo tanto es posible utilizar estos formatos de compresión de imágenes para nuestro proyecto porque cumplen con la necesidad de tener imágenes con una calidad lo más similar posible.

Esta evaluación se hizo con el fin de observar si al tener un PSNR igual en todos los formatos de compresión, se tiene la misma calidad subjetiva. Fue una prueba rápida y por cuestiones de tiempo y recursos no se hizo en condiciones controladas (misma iluminación, resolución y tamaño de pantalla, etc), por lo que los resultados no son completamente confiables pero nos dan una buena aproximación para el propósito de este proyecto.

#### **4.6. Pruebas de complejidad numérica**

Después de investigar en la literatura la complejidad numérica de los formatos de compresión a evaluar, se hizo una prueba experimental evaluando la complejidad numérica en función del tiempo necesario para convertir el lote de 133 archivos PNG a los formatos de compresión: JPEG, JPEG 2000, JPEG XR y WebP. Cabe señalar que esta no es una prueba completamente confiable porque el tiempo que se tarda en comprimir las imágenes depende mucho de la implementación que se está utilizando. Nosotros no programamos estos formatos y cada biblioteca fue programada muy seguramente por una persona distinta pero podemos hacer una aproximación suponiendo que son programas igual de optimizados.

Esta prueba experimental se llevó a cabo en 3 computadoras con diferentes características por lo que los resultados se presentarán en las siguientes 3 secciones.

##### **4.6.1. Prueba de complejidad numérica utilizando XnView ejecutándose bajo Windows.**

Para definir la complejidad numérica de cada uno de los formatos de compresión hicimos una prueba experimental del tiempo en que tarda el XnView en convertir los 133 archivos a cada uno de estos formatos con la configuración establecida para que los archivos resultantes en cada uno de los formatos tengan un PSNR=38 dB, repetimos esta prueba 10 veces para cada formato. La computadora utilizada para esta parte de la prueba experimental fue una Asus ZenBook Flip 14 UX461UA Signature Edition con las siguientes especificaciones:

- Procesador: intel(R) Core(TM) i7-8550U CPU @ 1.80 GHZ 1.99 GHz.  
RAM: 16 GB. Max clock speed 4 GHz.  
Windows 10 Home. Version 21H1

Los resultados se presentan en la Figura 21. El formato que tardó más tiempo fue JP2 y por lo tanto consideramos que es el más complejo, lo que concuerda con lo encontrado en la literatura. Mientras que JPG y JXR fueron los más rápidos.

<b>FORMATO</b>	<b>TIEMPO TOTAL</b>	<b>TIEMPO PROMEDIO</b>	<b>DESVIACION ESTANDAR</b>
<b>JPG</b>	<b>26 segundos</b>	<b>2.6000</b>	<b>0.5164</b>
<b>JP2</b>	<b>169 segundos</b>	<b>16.9000</b>	<b>0.3162</b>
<b>JXR</b>	<b>28 segundos</b>	<b>2.8000</b>	<b>0.4216</b>
<b>WebP</b>	<b>67 segundos</b>	<b>6.7000</b>	<b>0.4830</b>

**Figura 21.** Resultados de la prueba de complejidad numérica utilizando XnView MP bajo Windows.

#### **4.6.2. Prueba de complejidad numérica ejecutándose bajo el sistema operativo Linux.**

Para definir la complejidad de cada uno de los formatos de compresión hicimos una prueba experimental del tiempo en que tarda una computadora con sistema operativo Linux en convertir los 133 archivos a cada uno de estos formatos con la configuración establecida para que los archivos resultantes en cada uno de los formatos tengan un PSNR=38dB, repetimos esta prueba 20 veces para cada formato. Las características bajo las que se realizó esta prueba son las siguientes:

- Computadora: Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz Max clock speed 4.1 GHz, 32 GB RAM, Nvidia GeForce RTX 2060 Super  
Linux benderPC 5.4.0-81-generic #91-Ubuntu SMP Thu Jul 15 19:09:17 UTC 2021 x86\_64 x86\_64 x86\_64 GNU/Linux

En este caso el software utilizado fue ImageMagick 7.1.0-5. Los resultados se presentan en la Figura 22. Bajo estas condiciones fue se obtuvieron los tiempos de ejecución más bajos pero de nuevo se siguió la misma tendencia que en la prueba anterior, siendo JPEG 2000 el formato más complejo y JPEG JXR el formato más rápido con tiem-

<b>FORMATO</b>	<b>TIEMPO TOTAL</b>	<b>TIEMPO PROMEDIO</b>	<b>DESVIACION ESTANDAR</b>
<b>JPG</b>	<b>118.3477 segundos</b>	<b>5.9174</b>	<b>0.0274</b>
<b>JP2</b>	<b>276.9353 segundos</b>	<b>13.8468</b>	<b>0.0393</b>
<b>JXR</b>	<b>29.7939 segundos</b>	<b>1.4897</b>	<b>0.0047</b>
<b>WebP</b>	<b>128.6656 segundos</b>	<b>6.4333</b>	<b>0.0068</b>

**Figura 22.** Resultados de la prueba de complejidad numérica utilizando ImageMagick bajo computadora con sistema operativo Linux.

pos de ejecución de aproximadamente 1.5 segundos.

#### **4.6.3. Prueba de complejidad numérica ejecutándose bajo Raspberry Pi 4.**

Para definir la complejidad de cada uno de estos formatos hicimos una prueba experimental del tiempo en que tarda una Raspberry Pi 4 en convertir los 133 archivos a cada uno de estos formatos con la configuración establecida para que los archivos resultantes en cada uno de los formatos tengan un PSNR=38dB, repetimos esta prueba 20 veces para cada formato. Las características bajo las que se realizó esta prueba son las siguientes:

- Raspberry Pi 4  
4 GB de RAM @ 1.5GHz  
Linux raspberrypi 5.10.11-v7l+ #1399 SMP Thu Jan 28 12:09:48 GMT 2021 armv7l  
GNU/Linux.

En este caso el software utilizado fue ImageMagick 7.1.0-5 al igual que con el sistema operativo Linux. Los resultados de esta prueba se presentan en la Figura 23. Se puede observar como los tiempos de ejecución fueron significativamente mayores a los obtenidos en la prueba utilizando sistema operativo Windows y Linux, sin embargo los datos obtenidos concuerdan en el orden de complejidad de los formatos numéricos

de la literatura, pues nuevamente el formato con mayor tiempo de ejecución fue JPEG 2000 y el formato menos complejo JXR.

<b>FORMATO</b>	<b>TIEMPO TOTAL</b>	<b>TIEMPO PROMEDIO</b>	<b>DESVIACION ESTANDAR</b>
<b>JPG</b>	<b>208.1400 segundos</b>	<b>10.4070</b>	<b>0.3148</b>
<b>JP2</b>	<b>964.8002 segundos</b>	<b>48.2400</b>	<b>0.1386</b>
<b>JXR</b>	<b>152.0583 segundos</b>	<b>7.6029</b>	<b>0.3771</b>
<b>WebP</b>	<b>624.5951 segundos</b>	<b>31.2298</b>	<b>0.2125</b>

**Figura 23.** Resultados de la prueba de complejidad numérica utilizando ImageMagick bajo Raspberry Pi.

Algo importante a señalar es que el procesador de la Raspberry Pi es de 64 bits, pero el sistema operativo es Raspbian GNU/Linux 10 de 32 bits, por lo que es muy probable que los tiempos de ejecución puedan mejorar al utilizar un sistema operativo que explote mejor los recursos de hardware del sistema.

#### **4.7. Pruebas de tasa de compresión**

Junto con las pruebas de calidad objetiva, se obtuvieron la tasa de compresión y el tamaño promedio de los archivos resultantes, los resultados se muestran en las Figuras 24 y 25.

Como se puede observar los formatos con mayor tasa de compresión son JXR y WebP. Con PSNR=38 dB el formato JXR cuenta con una tasa de compresión mayor, mientras que en la Figura 25 el formato con mayor tasa de compresión es el WebP, esto se debe a que en este caso el PSNR de JXR=31 dB, 1 dB mayor que en los otros formatos, pues el ajuste de calidad se encuentra en el 0 y ya no fue posible bajar su PSNR a 30dB utilizando esta técnica, por lo que su tasa de compresión se ve un poco afectada. El formato que cuenta con una tasa de compresión menor es el JP2, las razones de este comportamiento se analizaron en la sección 4.3.2.

FORMATO	TASA DE COMP. PROMEDIO
<b>JPEG</b> <i>89 Calidad</i>	<b>0.8371</b> <b>+/-0.0272</b>
<b>JP2</b> <i>39 Calidad</i>	<b>0.8065</b> <b>+/- 0.1089</b>
<b>JXR</b> <i>53 Calidad</i>	<b>0.8995</b> <b>+/- 0.0331</b>
<b>WebP</b> <i>53 Calidad</i>	<b>0.8963</b> <b>+/- 0.0315</b>

**Figura 24.** Resultados de la prueba de tasa de compresión con PSNR= 38 dB.

FORMATO	TASA DE COMP. PROMEDIO
<b>JPEG</b> <i>17 Calidad</i>	<b>0.9578</b> <b>+/- 0.0080</b>
<b>JP2</b> <i>31 Calidad</i>	<b>0.9537</b> <b>+/- 0.0468</b>
<b>JXR</b> <i>0 Calidad</i>	<b>0.9756</b> <b>+/- 0.0097</b>
<b>WebP</b> <i>10 Calidad</i>	<b>0.9827</b> <b>+/- 0.0065</b>

**Figura 25.** Resultados de la prueba de tasa de compresión con PSNR= 30 dB.

## Capítulo 5. Robustez a la transmisión a través del canal satelital

---

### 5.1. Sistema de Telemetría y Control

La recolección de datos y de la información de funcionamiento que presentan los instrumentos y sistemas de un vehículo espacial, son actividades que determinan el éxito de la misión de cualquier vehículo espacial.

El subsistema de telemetría, telecomando y control (TT&C) se encarga de proveer a la estación terrena la información necesaria para conocer el estado del satélite y poder controlar a distancia el mismo. La telemetría es la transmisión hacia la Tierra de la información recabada por los diferentes sensores que lleva a bordo el satélite (e.g. voltajes, corrientes y temperatura de los componentes a bordo, carga restante en las baterías, etc.). El telecomando y control se refiere al envío de señales de la estación terrena al satélite para que este realice determinadas funciones (e.g. orientarse en cierta posición o iniciar/finalizar ciertas tareas) (Alonso *et al.*, 2012)

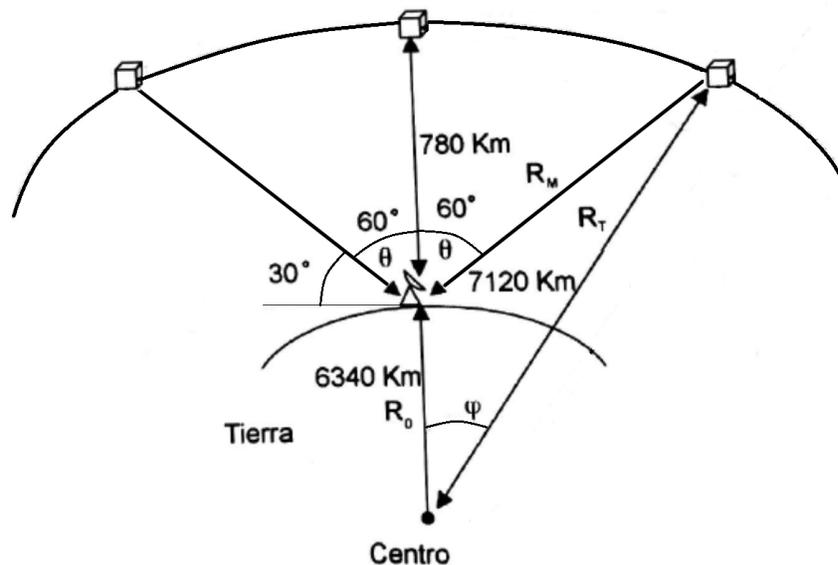
Para satélites pequeños, tradicionalmente este sistema se diseña para operar usando las frecuencias asignadas para los servicios de radio amateur que están contenidas en las bandas de VHF (144-146 MHz) y UHF (430-440 MHz). Sin embargo recientemente algunas misiones comienzan a incluir algunas otras bandas sobre todo la banda S (2.3-2.45 GHz) lo que ha llevado a comenzar a hablar de sistemas de banda angosta y banda ancha usando para este último caso la banda S.

#### 5.1.1. Consideraciones generale de diseño

El sistema de telemetría y comando de una misión espacial está diseñado para satisfacer las características dadas por un microsatélite operando a órbita baja por lo que tiene un constante movimiento con respecto a la tierra, razón por la cual aparece en el horizonte, pasa sobre la estación terrena y vuelve a desaparecer. Si consideramos una altura típica de órbita de 780 Km, el periodo orbital es de alrededor de 100 minutos por cada vuelta a la tierra. Sin embargo, dado que la tierra gira sobre su eje polar, el satélite nunca pasa dos veces seguidas sobre el mismo punto, por lo que solo

cada determinado número de vueltas vuelve a aparecer sobre la estación terrena. Esto permite un tiempo de visibilidad del satélite de entre 4 y 10 minutos, dependiendo de su trayectoria sobre la estación terrena lo cual implica que solamente se tiene ese tiempo para contactar al satélite, subir y bajar información y esperar de nuevo a que vuelva a pasar (Pacheco *et al.*, 2009).

Además, la distancia del satélite varía entre un máximo y un mínimo, esto implica una longitud de arco (distancia recorrida) de 2370 Km a una velocidad instantánea (tangencial) de casi 27000 Km/h, lo cual genera a su vez un corrimiento de frecuencia por efecto Doppler. En la Figura 26 se muestran los parámetros básicos para la geometría del satélite. La forma de establecer dichas geometrías se basa en los cálculos siguientes:



**Figura 26.** definición de la geometría del satélite para una órbita baja, tomado de Pacheco *et al.* (2009)

Primeramente se establece el cálculo de las distancias, para ello consideramos un ángulo de visibilidad de  $60^\circ$  por lo que:

$$\theta = 60^\circ$$

$$\text{Angulo elev. min. } \delta = 30^\circ$$

$$\text{Altura } h = 780\text{Km}$$

$$R_s = 780\text{Km}$$

$$R_t = R_0 + R_s = 7120\text{Km}$$

Para calcular la distancia  $R_M$ , primero por ley de senos:

$$\frac{R_0}{\text{Sen}\alpha} = \frac{R_T}{\text{Sen}\beta} = \frac{R_M}{\text{Sen}\varphi}$$

y dado que

$$\alpha + \beta + \varphi = 180^\circ$$

y

$$\beta = 90 + 30 = 120^\circ$$

entonces

$$\text{sen}\alpha = \frac{R_0}{R_T} \text{sen}\beta = \frac{6340}{7120} \text{sen}120^\circ$$

$$\alpha = \text{sen}^{-1}(0.771) = 50.46^\circ$$

entonces

$$\varphi = 180 - 50.46 - 120 = 9.54^\circ$$

Por lo que se tiene

$$\begin{aligned} R_M^2 &= R_T^2 + R_0^2 - 2 \cdot R_0 \cdot R_T \cdot \text{Cos}\varphi \\ &= 18\,569\,819 \text{ Km}^2 \Rightarrow R_M = 1362.7 \text{ Km} \end{aligned}$$

A continuación la estimación de los parámetros dinámicos del satélite nos dará el tiempo de línea de vista.

Altura del Satélite = 780Km

Elev. min. = 30°, máx = 90°

Dist. ET - SAT :

Min. = 780Km

Max. = 1362.7Km

Dist. de la Trayectoria de Línea de Vista :

$S = r\theta$  donde:  $\theta$  en radianes

$$\theta = 2\varphi = 2(9.54) = 19.08^\circ$$

Por lo que :

$$S = (7120)(0.333) = 2370.85\text{Km}$$

Para calcular la velocidad instantánea del satélite:

$$v = \left(\frac{GM}{R_T}\right)^{1/2} = \sqrt{\frac{(6.67 \times 10^{-11})(5.98 \times 10^{24})}{(7.12 \times 10^6)}} = 7484 \text{ m/s} = 26942\text{Km/h}$$

El periodo orbital estará dado por :

$$T_{ORB} = \frac{2\pi(R_T)^{3/2}}{(GM)^{1/2}} = \frac{2\pi(7.12 \times 10^6)(7.12 \times 10^6)^{1/2}}{(6.67 \times 10^{-11})^{1/2}} = 5977\text{s} = 99.6 \text{ min}$$

Tiempo en Línea de Vista :

$$v = \frac{d}{t} \Rightarrow t = \frac{d}{v} = \frac{S}{v} = \frac{2370.85\text{Km}}{26942\text{Km/h}} = 0.088\text{h} = 5.28 \text{ min}$$

### 5.1.2. Cálculo del efecto Doppler

Con base en los parámetros de comportamiento en órbita, se puede calcular el efecto Doppler mediante el siguiente procedimiento (Mendieta *et al.*, 2010). La Figura 27 muestra la geometría del efecto Doppler.

Altura del Satélite = 780Km

Elev. min. = 30°

Dist. ET -SAT Min. = 780Km

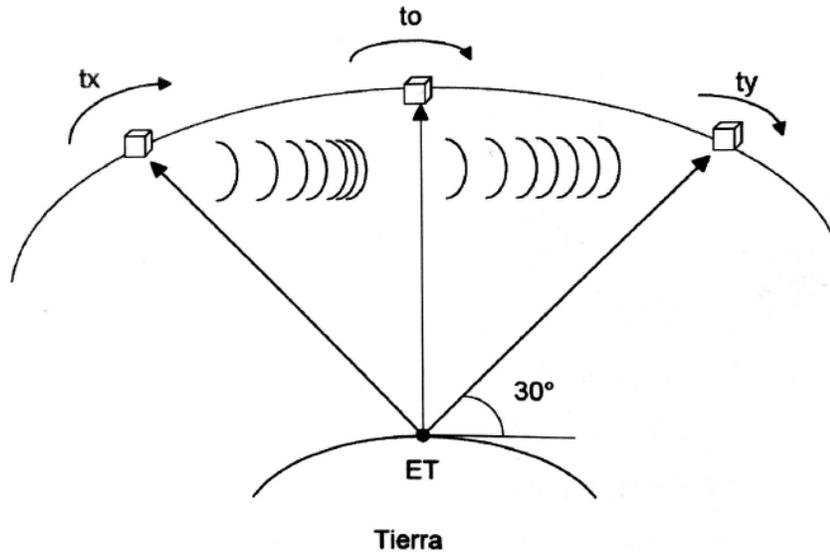
Radio del Satélite = 780 + 6340 = 7120Km

Velocidad del Satélite  $v = 26942\text{Km/h}$

Velocidad Relativa en la ET =  $v_r$

$$v_r = v \frac{R_T}{R_O} = (7484 \text{ m/s}) \frac{6.34 \times 10^6 \text{ m}}{7.12 \times 10^6 \text{ m}} = 6664 \text{ m/s}$$

$\Delta f$  = Corrimiento de Frecuencia Debido al Doppler



**Figura 27.** Geometría del efecto Doppler para una órbita baja, tomado de Pacheco *et al.* (2009)

$$\Delta f = \frac{v_r}{c} f_0 = \frac{6664}{3 \times 10^8} (435 \times 10^6) = 96628 \text{ Hz} = 96.628 \text{ KHz}$$

Al resultado obtenido se le agregan 226 Hz por efecto de rotación de la Tierra. Entonces:

$$\Delta f = 96.854 \text{ KHz}$$

Por lo que el Corrimiento Total sera :

$$f \pm \Delta f = 435 \text{ MHz} \pm 96.854 \text{ KHz}$$

### 5.1.3. Cálculo de las pérdidas por propagación en el espacio libre

El modelo de pérdida por trayectoria en el espacio libre es usado para predecir la intensidad del nivel de recepción cuando el transmisor y receptor tienen una trayectoria de línea de vista clara, sin obstrucciones entre ellos.

La atenuación en espacio libre es directamente proporcional al cuadrado de la distancia y la frecuencia, la pérdida por espacio libre representa la mayor parte de la

atenuación total causada por efectos de propagación de la onda electromagnética.

De acuerdo a la recomendación ITU P.525 ITU (2019), cuando se trata de un enlace punto a punto, es preferible calcular la pérdida de espacio libre considerando dos antenas isótropas, denominada también pérdida básica de transmisión en el espacio libre, esto se calcula de la siguiente manera:

$$L_{bf} = 20 \log \left( \frac{4\pi * d * f}{c} \right) dB = 20 \log \left( \frac{4\pi * 780 \times 10^3 * 435 \times 10^6}{3 \times 10^8} \right) dB = 143.05 dB$$

donde:

$L_{bf}$ : pérdida básica de transmisión en el espacio libre (dB)

d: distancia

f: frecuencia

c: velocidad de la luz

## 5.2. Tasa de error de bits (BER)

Los sistemas modernos de comunicaciones digitales necesitan de mediciones de rendimiento de extremo a extremo. La medida de ese rendimiento suele ser la tasa de error de bits (BER), que cuantifica la fiabilidad de todo el sistema de radio desde “bits de entrada” hasta “bits de salida”, incluyendo la electrónica, las antenas y la ruta de la señal en el medio. El BER es un concepto simple, su definición es simplemente:

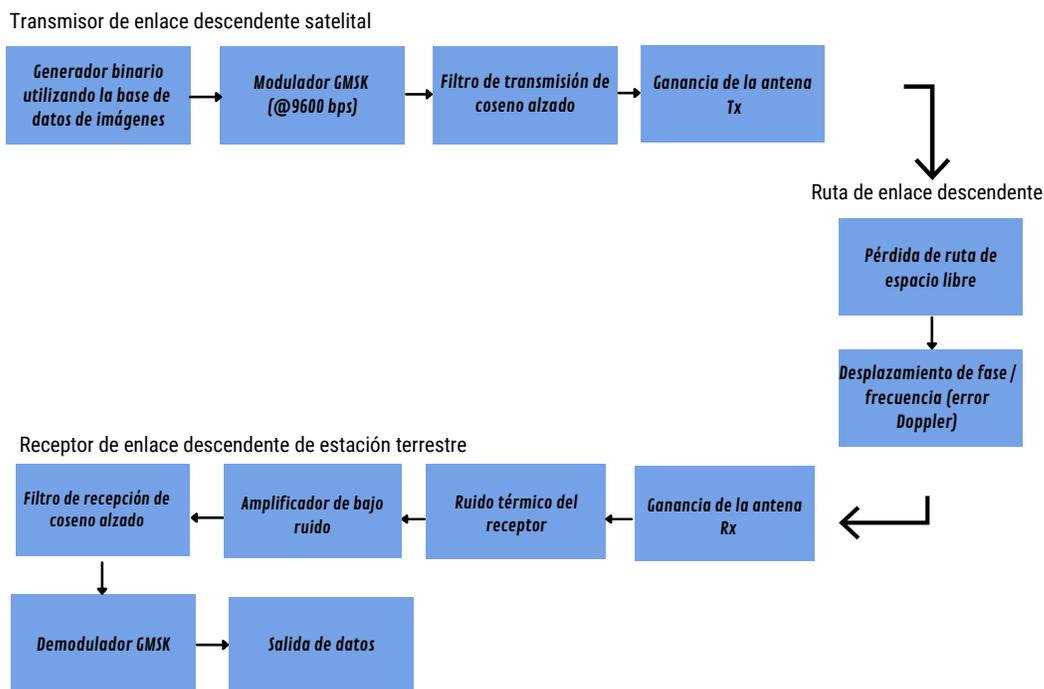
$$BER = \text{Número de bits con errores} / \text{Número total de bits transmitidos}$$

Con una señal fuerte y una trayectoria de señal no perturbada, este número es tan pequeño como para ser insignificante. Se vuelve significativo cuando deseamos mantener una relación señal-ruido suficiente en presencia de transmisión imperfecta a través de circuitos electrónicos (amplificadores, filtros, mezcladores y convertidores digitales/analógicos) y el medio de propagación (por ejemplo, la ruta de radio o la fibra óptica) (Breed, 2003).

### 5.3. Modelo de canal satelital

Utilizando SIMULINK, se creó un modelo de un enlace satelital de RF utilizando los bloques de Communications Toolbox para simular la comunicación entre un CubeSat y la estación terrena, el modelo está basado en las características de un radio comercial PULSAR-TMTC de CLYDE space (Clydespace, 2020) y se hizo con las consideraciones de diseño presentadas en las secciones anteriores de este capítulo.

Este modelo esta formado por distintos bloques, como se muestra en la figura 28, el modelo consta de un transmisor, una ruta y un receptor satelital de enlace descendente. A continuación se presenta una explicación breve de cada uno de los bloques del modelo y sus parámetros.



**Figura 28.** Diagrama a bloques del modelo de canal satelital.

### 5.3.1. Transmisor satelital de enlace descendente

- Generador binario utilizando la base de datos de imágenes. Utilizando la base de datos de 133 imágenes, se crea en MATLAB un flujo de datos binario para cada imagen a transmitir y se envía a la entrada del modelo.
- Modulador GMSK a 9600 bps. Para transmitir el flujo binario de datos sobre el canal de comunicación, necesitamos generar una señal que represente dicho flujo y que se ajuste o adapte a las características del canal. El objetivo de la modulación es transferir de una manera adecuada los datos de la fuente a través de un canal de comunicaciones.

En este modelo se usa modulación GMSK debido a que el radio que estamos tomando de referencia para la simulación es el radio comercial PULSAR-TMTC de Clydespace (2020). Este radio admite modulación AFSK a 1200 bps y GMSK a 9600 bps.

GMSK es un esquema de modulación derivado de MSK, el cual consiste en una modulación en frecuencia modificada; tiene la particularidad de que la señal modulada no contiene discontinuidades en la fase, y la frecuencia cambia en el cruce por cero cuando cambian los datos de entrada (Stremler, 1982).

- Filtro de transmisión de coseno alzado: muestra y da forma a la señal modulada utilizando la forma de pulso de coseno alzado. Este tipo de filtro ayuda a reducir al mínimo la interferencia entre símbolos (ISI), en este modelo estamos simulando un filtro tipo raíz cuadrada con un factor de rolloff de 0.3.
- Ganancia de la antena Tx: Este bloque aplica la ganancia de la antena del transmisor. En este bloque suponemos que la antena del CubeSat tiene una ganancia de 0dBic.

### 5.3.2. Ruta de enlace descendente

- Pérdida por trayectoria de espacio libre: atenúa la señal mediante la pérdida de espacio libre. El modelo de pérdida por trayectoria en el espacio libre es usado para predecir la intensidad del nivel de recepción cuando el transmisor y receptor tienen una trayectoria de línea de vista clara, sin obstrucciones entre ellos. En

este bloque se utiliza la atenuación obtenida en la sección 5.1.3 para hacer la simulación.

- Desplazamiento de fase / frecuencia (error Doppler): gira la señal para modelar el error Doppler en el enlace. Este bloque utiliza el valor obtenido en la sección 5.1.2 para hacer la simulación.

### **5.3.3. Receptor de enlace descendente de estación terrestre**

- Ganancia de la antena Rx: aplica la ganancia de la antena del receptor. En esta simulación se utiliza la antena EB-432 EGGBEATER de M2 Factory Direct (2021), cuenta con una ganancia de 5.5 dBic, es una ganancia alta porque la antena está diseñada para un ambiente de mucho ruido.
- Ruido térmico del receptor (temperatura del sistema del receptor de satélite): agrega ruido blanco gaussiano que representa la temperatura efectiva del sistema del receptor. En la simulación este parámetro se puede ajustar para 4 casos específicos, los cuales son: 0k (sin ruido), 20 K (nivel de ruido muy bajo), 290 K (nivel de ruido típico) y 500 K (nivel de ruido muy alto).
- Amplificador de bajo ruido: aplica una ganancia de amplificador de bajo ruido.
- Filtro de recepción de coseno alzado: aplica un filtro adaptado a la señal modulada utilizando la forma de pulso de coseno alzado de raíz cuadrada con las mismas características que en la transmisión.
- Demodulador GMSK: Demodula el flujo de datos y reconstruye las imágenes.

Este modelo se utilizó para transmitir distintas imágenes en los 4 formatos a evaluar y en todos se obtuvieron resultados similares, en la Figura 29 se muestra el modelo en Simulink. La transmisión de la imagen fue exitosa en todos los casos en que el efecto Doppler era considerado nulo, al aplicar el efecto Doppler calculado en la sección 5.1.2 nuestro modelo presentaba demasiados errores y no era posible reconstruir la imagen recibida. Esto se debe a que el modelo no cuenta con un bloque de corrección del efecto Doppler, pues se utiliza una modulación GMSK que es nueva en estos sistemas de comunicaciones, y Simulink no cuenta con un bloque preestablecido para este tipo de modulación, por cuestiones del tiempo disponible para finalizar la tesis no fue

posible crear este bloque por lo que la evaluación de robustez a la pérdida de bits se realizó con otra metodología, la cual se presenta a continuación.

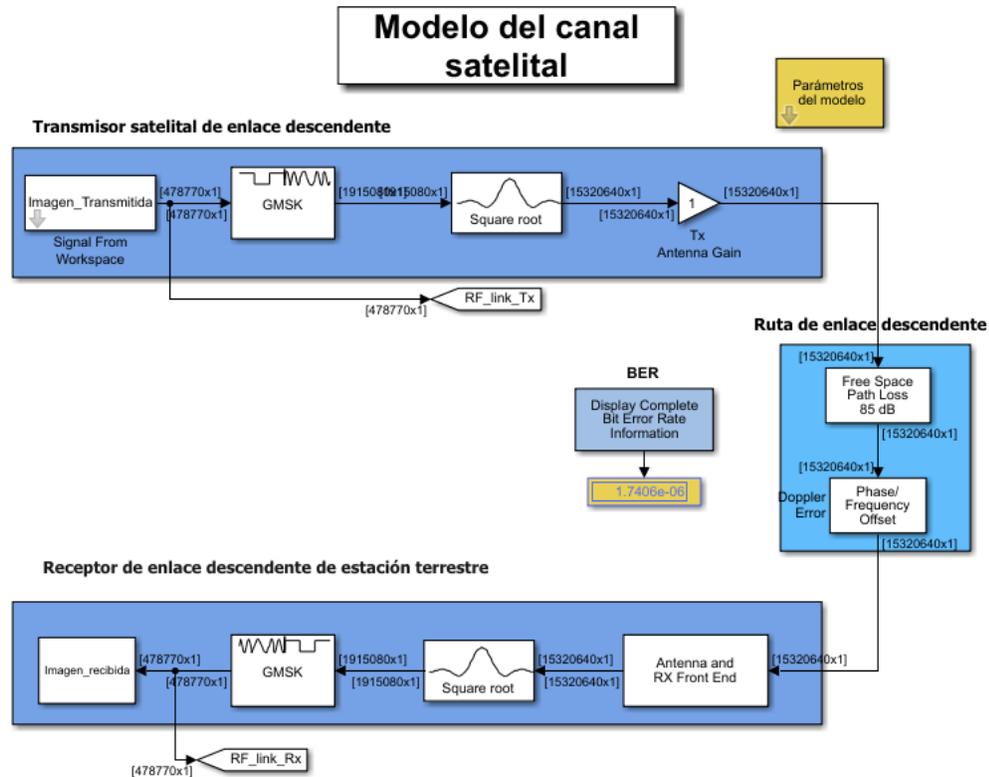


Figura 29. Modelo de canal satelital en Simulink.

#### 5.4. Robustez a la pérdida de bits

Como se mencionó en la sección 5.2, el concepto básico detrás de la tasa de error de bits es bastante sencillo. Se envía un flujo de datos, en nuestro caso imágenes, a través del canal de comunicaciones, para nosotros un enlace de radio satelital, y el flujo de datos resultante se compara con el original. Los cambios se anotan como errores de datos y se registran. Con esta información se puede determinar la tasa de error de bits. Dadas las dificultades encontradas para evaluar la robustez a los errores de bits para cada códec mediante un modelo de canal satelital, se optó por desarrollar un algoritmo que altera de manera aleatoria, pero controlada, los bits en un flujo de datos que simula la transmisión satelital. En nuestro caso, se procedió controlando directamente la tasa de error de bit (BER) en un vector de datos binarios formado por las imágenes en cada uno de los cuatro formatos, la manera en que se procedió se explica a continuación.

El procedimiento para alterar una imagen es el siguiente. Los bits que forman la imagen son alineados y replicados en un solo vector, llamado  $\mathbf{x}$ . Los bits se replican tantas veces como sea necesario para obtener un total de al menos  $T_{\text{bits}} = 10^8$  bits. Específicamente, para una imagen que contiene  $I_{\text{bitsize}}$  bits, la longitud  $x_\ell$  del vector  $\mathbf{x}$  está dada por:

$$x_\ell = \left\lceil \frac{T_{\text{bits}}}{I_{\text{bitsize}}} \right\rceil \times I_{\text{bitsize}},$$

donde  $\lceil \cdot \rceil$  se refiere al redondeo al entero superior próximo. De manera más precisa, el vector  $\mathbf{x}$  contiene los bits de exactamente  $M = \left\lceil \frac{T_{\text{bits}}}{I_{\text{bitsize}}} \right\rceil$  imágenes. Después se procedió a seleccionar aleatoriamente  $N$  bits tomados de una distribución uniforme en el rango  $[1 \ x_\ell]$ . Estos bits fueron alterados de manera que si el bit tenía el valor de 1, fue cambiado por un 0 y viceversa. Después, el vector  $\mathbf{x}$  es dividido en  $M$  secciones que corresponden a cada una de las imágenes que lo forman. Estas imágenes, formadas por bits que han sido alterados de manera intencional, son almacenadas en la computadora y analizadas siguiendo el procedimiento que se explicará en la siguiente sección. En este trabajo, decidimos investigar exhaustivamente la robustez de los formatos de imágenes a la pérdida de bits, por lo que el número de bits alterados se varía de la siguiente manera. En función de los siguientes valores de BER  $T_{\text{BER}} = [1 \times 10^{-3}, 5 \times 10^{-4}, 1 \times 10^{-4}, 5 \times 10^{-5}, 1 \times 10^{-5}, 5 \times 10^{-6}, 1 \times 10^{-6}, 5 \times 10^{-7}, 1 \times 10^{-7}, 1 \times 10^{-8}]$ , se calcula el número de bits a alterar de la siguiente manera  $N(k) = \lceil T_{\text{BER}}(k) \cdot x_\ell \rceil$  donde  $k = 1, \dots, 10$ . Este procedimiento se repite para cada una de las 133 imágenes de la base de datos, para cada uno de los 4 formatos y finalmente para los 10 valores del BER elegidos. Por lo que el procedimiento para analizar la robustez de los formatos se repite en total  $133 \times 4 \times 10 = 5,320$  veces. Los valores de BER bastante altos, por ejemplo  $1 \times 10^{-3}$ , permiten analizar escenarios de comunicación bastante desfavorables donde la transmisión satelital tiene una pérdida de bits considerable. Por el contrario, valores bajos del BER, por ejemplo  $1 \times 10^{-8}$ , permiten analizar escenarios de comunicación con condiciones cercanas a las ideales. Para cada una de las 5,320 imágenes analizadas se calcula el PSNR con respecto a la imagen original (en formato PNG) que se utiliza como referencia. En la siguiente sección se presentan los resultados obtenidos así como varios ejemplos de las imágenes resultantes después de haber sido alteradas por la pérdida de bits.

### 5.4.1. Análisis de resultados para robustez a la pérdida de bits

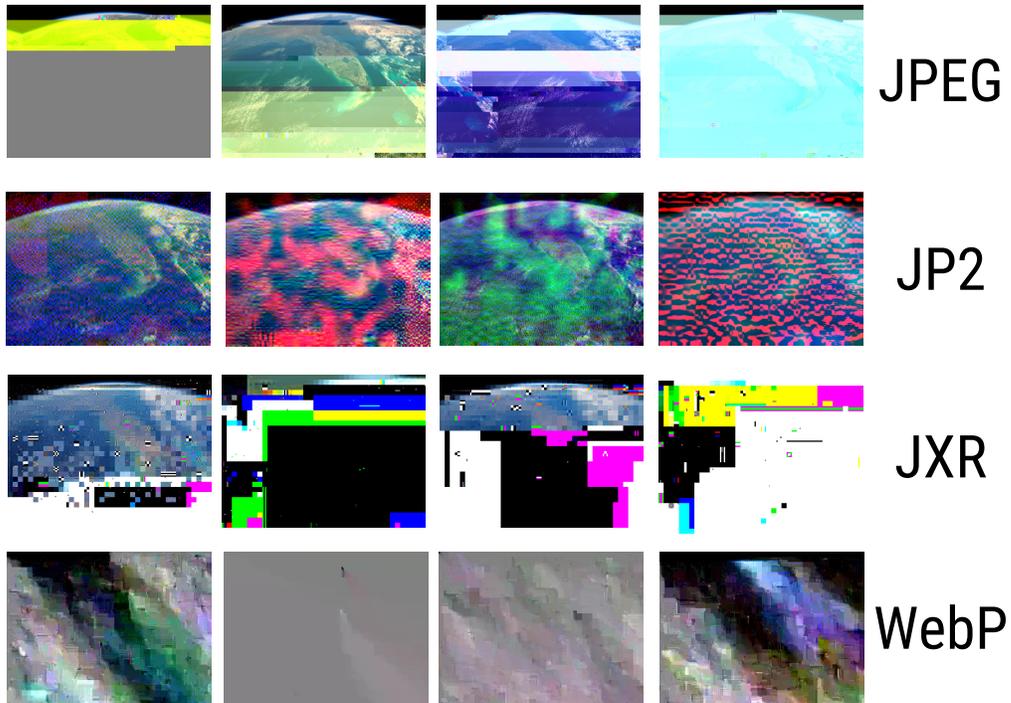
A continuación se presenta un análisis de resultados de la robustez a la pérdida de bits, para cada uno de los formatos de compresión de imágenes evaluados. En las Figuras 31 a la 34 se presentan ejemplos de imágenes reconstruidas para cada uno de los formatos de compresión evaluados con diferentes valores de BER. En todos los casos, la imagen original antes de alterar los bits es la presentada en la Figura 30. La intención de mostrar estas imágenes es permitir al lector darse una idea del tipo de degradación que se pueden obtener como resultado de la pérdida de bits durante el proceso de transmisión.



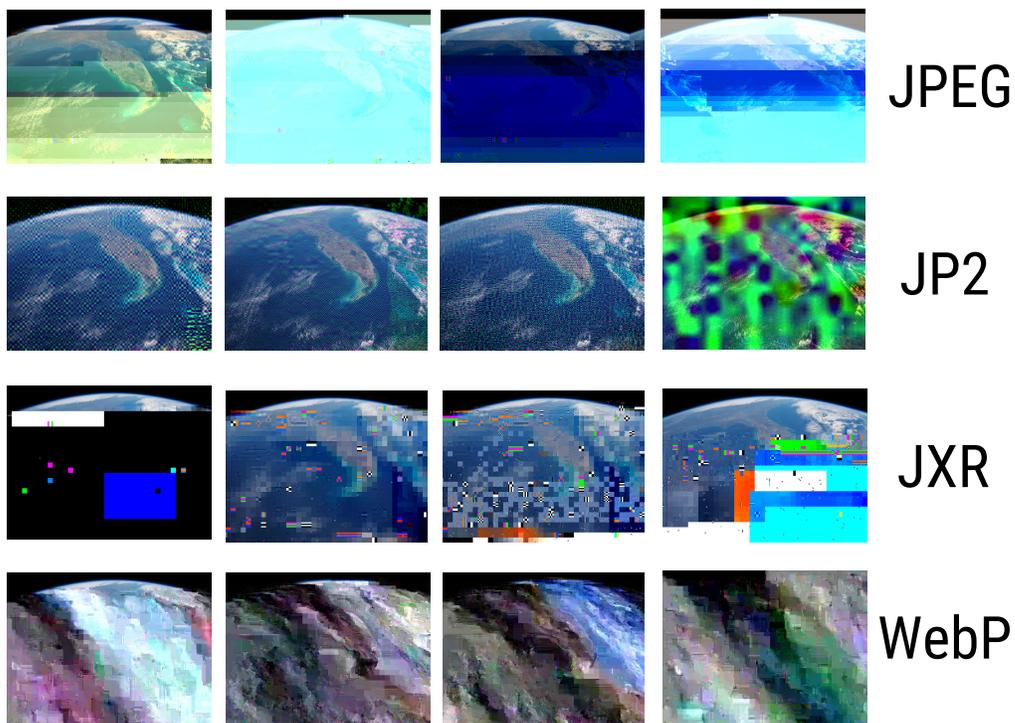
**Figura 30.** Imagen original antes de hacer la alteración de bits.

#### 5.4.1.1. JPG

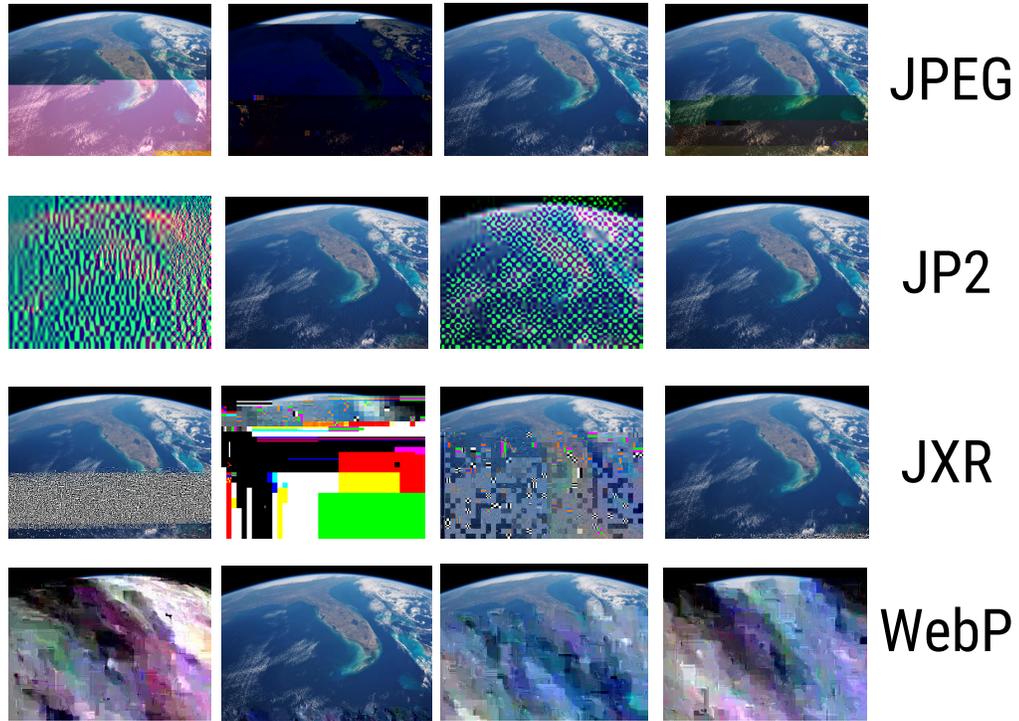
En la Figura 35 se presentan las gráficas de PSNR promedio por número de imagen para diferentes valores de BER. Se puede ver que a medida que el BER disminuye su valor, el PSNR resultante es cada vez más cercano al PSNR de referencia. Para valores de  $BER = 5 \times 10^{-7}$  o menores, el PSNR resultante está muy cerca del PSNR de referencia. Para valores grandes de BER, las imágenes de salida por lo general presentan una parte de la imagen muy similar a la original y la otra parte completamente irreparable o pérdida de colores en diferentes partes de la imagen como se observa en la Figura 31. En la Figura 34 vemos que la imagen reconstruida es muy similar a la original, pero en algunos casos aún presenta cambios de color.



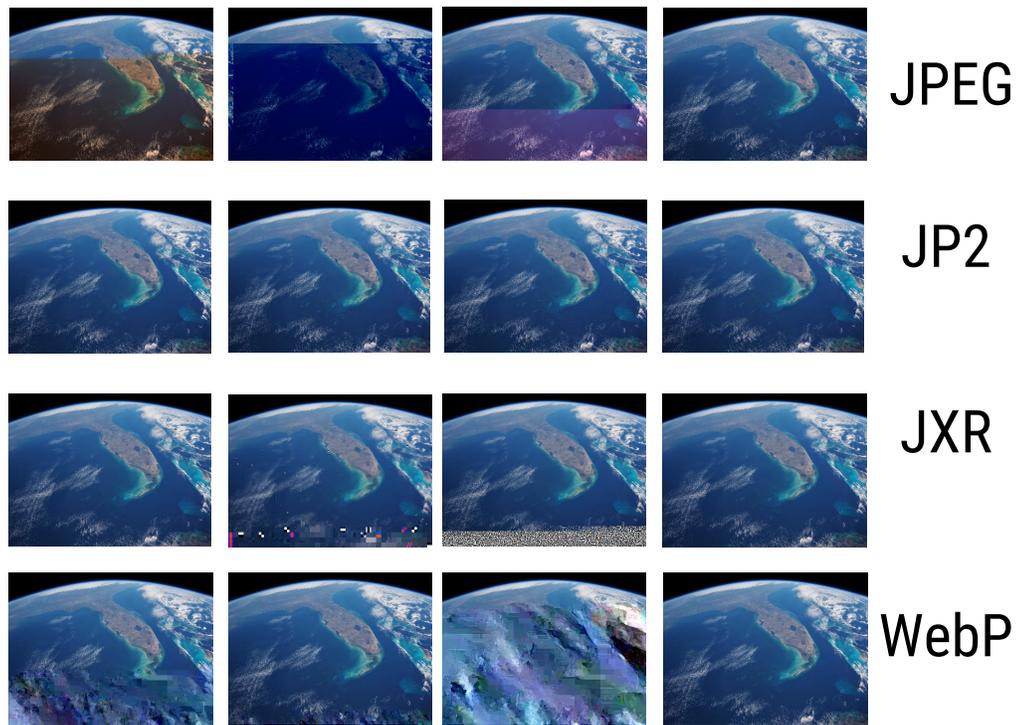
**Figura 31.** Ejemplos de imágenes reconstruidas para un  $BER = 1 \times 10^{-3}$ .



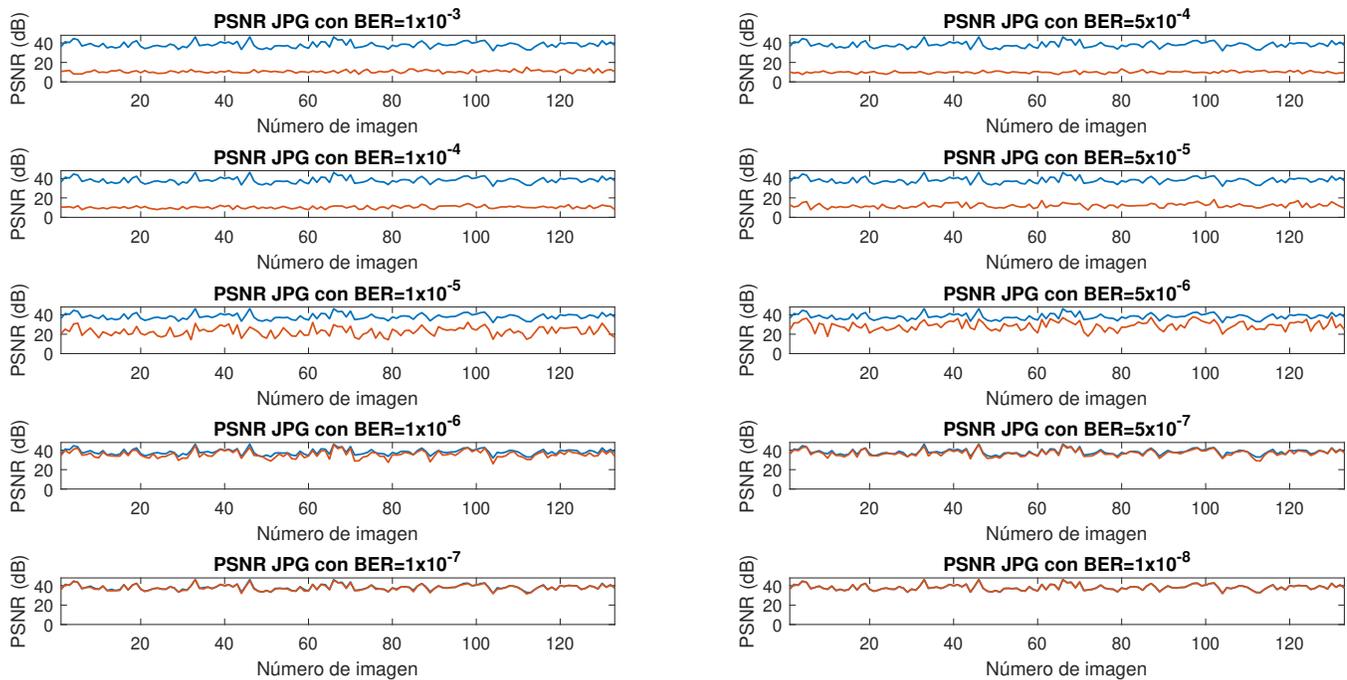
**Figura 32.** Ejemplos de imágenes reconstruidas para un  $BER = 1 \times 10^{-4}$ .



**Figura 33.** Ejemplos de imágenes reconstruidas para un  $BER = 1 \times 10^{-5}$ .



**Figura 34.** Ejemplos de imágenes reconstruidas para un  $BER = 1 \times 10^{-6}$ .



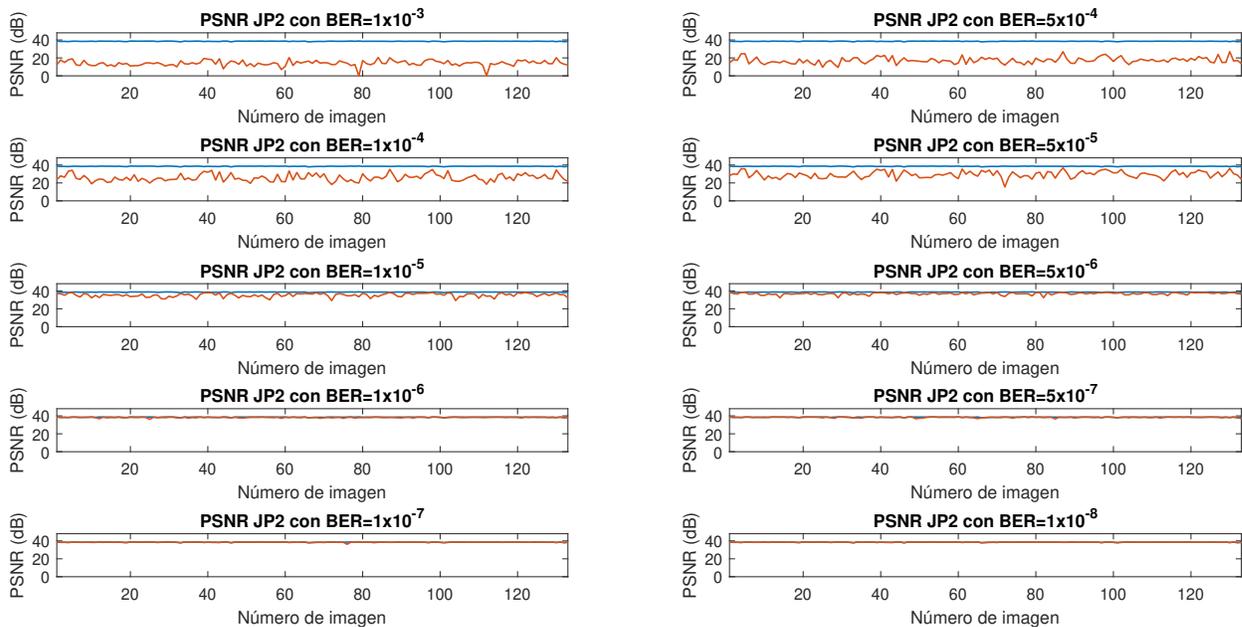
**Figura 35.** PSNR resultante por imagen para diferentes valores de BER en el formato JPG. La línea azul representa el PSNR de referencia y la línea naranja el PSNR resultante en cada una de las 133 imágenes de la base de datos.

#### 5.4.1.2. JP2

Este formato es el que tiene un mejor comportamiento en su PSNR, pues tiende a subir muy rápido incluso con valores de BER grandes, como podemos ver en la Figura 36, a partir de  $BER = 1 \times 10^{-5}$  su PSNR resultante ya está muy cercano al PSNR de referencia. Las imágenes reconstruidas después de alterar los bits tienen por lo general tienen errores en los colores o presentan rayas o cuadros de diferentes tonalidades a las de la imagen original como se ve en las Figuras 31 y 32. Este formato es el que presenta mejor PSNR en función del BER, por lo que las imágenes reconstruidas presentan una mejor calidad y con un  $BER = 1 \times 10^{-6}$  son casi iguales a la imagen original como se ve en la Figura 34.

#### 5.4.1.3. JXR

Este formato presenta un comportamiento parecido a JPG. En cuanto a su PSNR es el segundo mejor formato, por debajo de JP2, pero cuenta con la ventaja de tener los archivos más pequeños. Como se ve en la Figura 37 a partir de un  $BER = 1 \times 10^{-6}$  el PSNR promedio se acerca mucho al PSNR de referencia. Las imágenes de salida tienen un comportamiento muy similar a las de JPG pero más notorio pues presenta

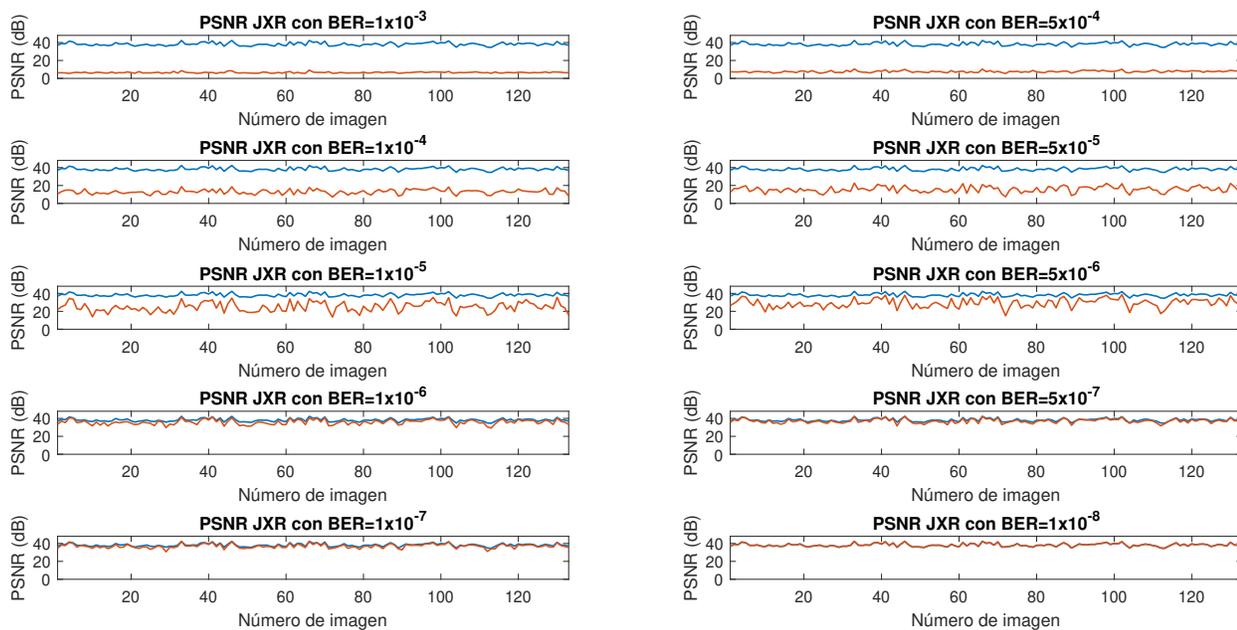


**Figura 36.** PSNR resultante por imagen para diferentes valores de BER en el formato JP2. La línea azul representa el PSNR de referencia y la línea naranja el PSNR resultante en cada una de las 133 imágenes de la base de datos.

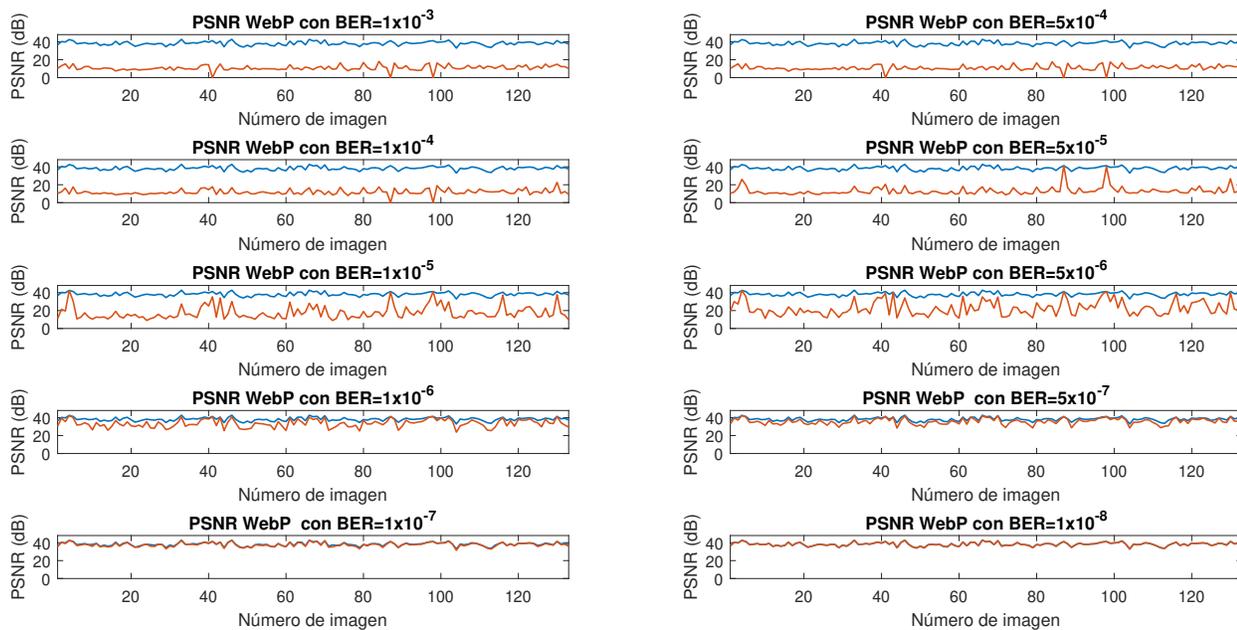
cuadros de diferentes colores que abarcan toda la imagen como se observa en la Figura 31. A partir de un  $BER = 1 \times 10^{-6}$  las imágenes de JXR tienen un buen PSNR y han sido recuperadas casi completamente, solo presentan algunos errores en las orillas o pequeños cambios de color como podemos apreciar en la Figura 34.

#### 5.4.1.4. WebP

Este formato es el que presenta peor desempeño en PSNR, pues como se observa en la Figura 38 con  $BER = 1 \times 10^{-5}$  se tiene un PSNR promedio de 17 dB, mientras que los demás formatos tienen un PSNR mayor a 21 dB en este valor de BER. Analizando las imágenes obtenidas después de modificar los bits, vemos que el PSNR no es una métrica completamente confiable, pues los archivos recibidos de webP con  $BER = 1 \times 10^{-4}$  son muy difíciles de interpretar como se observa en la Figura 32. La mayoría de veces la presenta como una mancha borrosa con los colores de la imagen original difuminados, haciendo imposible que nos demos una idea de cómo es la imagen original, mientras que en los otros formatos las imágenes se presentan con partes muy parecidas a la imagen original y pequeñas partes completamente perdidas.



**Figura 37.** PSNR resultante por imagen para diferentes valores de BER en el formato JXR. La línea azul representa el PSNR de referencia y la línea naranja el PSNR resultante en cada una de las 133 imágenes de la base de datos.

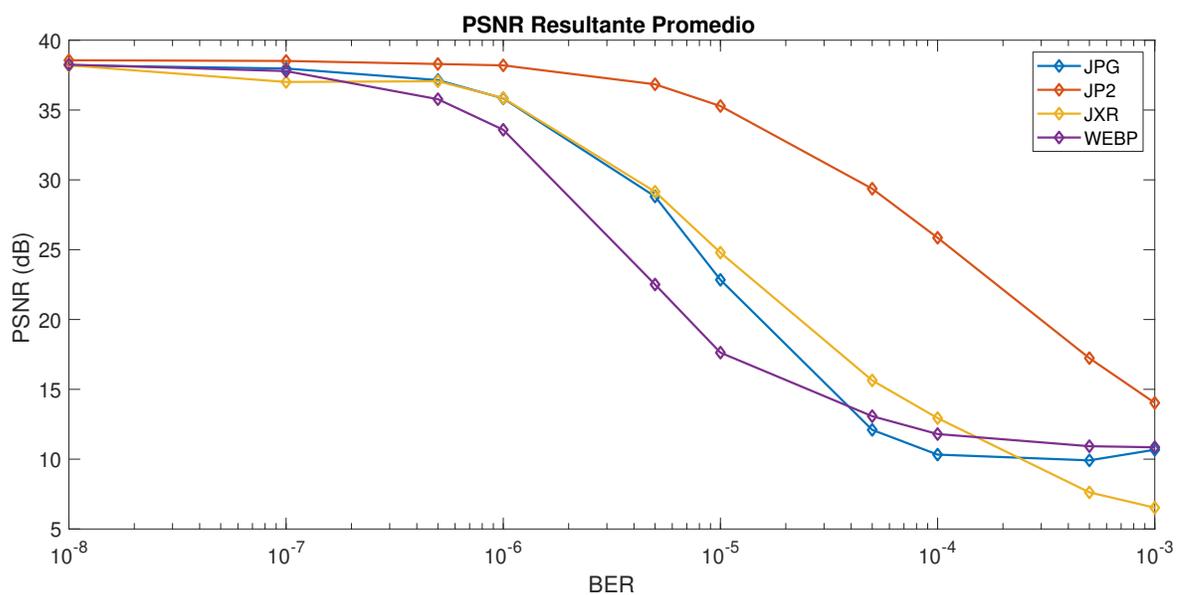


**Figura 38.** PSNR resultante por imagen para diferentes valores de BER en el formato WebP. La línea azul representa el PSNR de referencia y la línea naranja el PSNR resultante en cada una de las 133 imágenes de la base de datos.

### 5.4.1.5. PSNR promedio

En la Figura 39 se presenta el PSNR promedio de cada uno de los 4 formatos de compresión en función del BER. Como se puede ver, el formato JP2 tiene una gran ventaja con respecto a los demás, pues desde valores de BER muy grandes ( $BER = 1 \times 10^{-4}$ ) presenta un PSNR promedio de 25 dB, correspondiente a una calidad aceptable en la escala de MOS presentada en la Figura 13, mientras que con el mismo valor de BER, los demás formatos presentan un PSNR por debajo de 15 dB, lo que se traduce como una calidad mala en las escala de MOS. Sin embargo, de pruebas pasadas obtuvimos que JP2 es el formato con los archivos más grandes y de mayor complejidad numérica, lo que podría representar una desventaja frente a formatos como JXR que es el que tiene los archivos más pequeños y también es el formato más sencillo en cuestión de complejidad numérica.

De la Figura 39 también podemos concluir que JPG y JXR tienen un comportamiento muy similar a partir de un  $BER = 5 \times 10^{-5}$  y para valores de BER menores. El formato WebP es el que presenta un peor comportamiento entre  $BER = 5 \times 10^{-5}$  y  $BER = 5 \times 10^{-7}$  por lo que al tener un sistema con un BER entre estos valores, se recomienda no utilizar formato WebP.



**Figura 39.** PSNR promedio en función del BER para los 4 formatos de compresión evaluados.

## 5.5. Probabilidad de archivo dañado

Un factor muy importante a la hora de evaluar la robustez a la pérdida de bits en nuestro sistema es la probabilidad de que durante la transmisión el archivo haya sido dañado incluso al punto de no puede ser abierto por las aplicaciones de visualización de imágenes. Es decir, con esto nos referimos a la probabilidad de que un archivo se encuentre demasiado dañado por lo que es imposible reconstruirlo y por lo tanto no se puede hacer una evaluación de su PSNR. Para obtener este parámetro en cada formato de compresión, se dividió el total de archivos dañados entre el total de archivos evaluados para cada imagen de la base de datos y se sacó el promedio, esto se hizo para cada valor de BER. También se obtuvo la desviación estándar para cada caso.

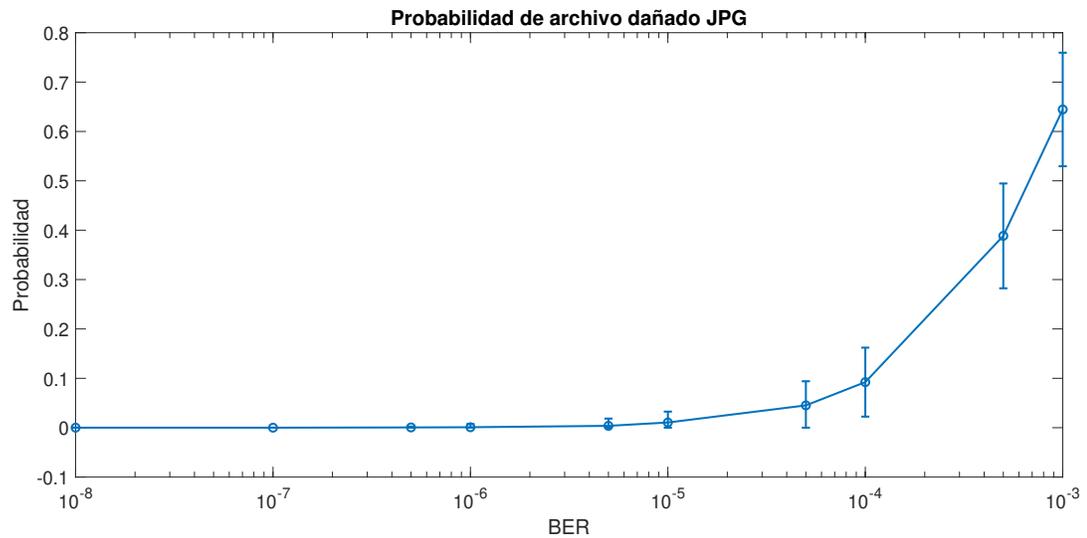
A continuación se presenta un análisis de este parámetro de cada uno de los formatos evaluados

### 5.5.1. JPG

Este formato de compresión es el que presenta mayor probabilidad de archivo dañado como se muestra en la Figura 40. Con un valor de 0.65 y una desviación de 0.11 para  $BER = 1 \times 10^{-3}$ , esto quiere decir que en este valor de BER aproximadamente el 65% de los archivos es imposible de reconstruir, en la transmisión de imágenes en un enlace satelital esto no es lo esperado pues son muy pocas las veces que podríamos recuperar una imagen, pero debemos tomar en cuenta que este valor de BER es muy alto por lo que es de esperar esta probabilidad de archivo dañado. Podemos ver que a partir de un  $BER = 5 \times 10^{-5}$  esta probabilidad se vuelve muy baja por lo que es de esperarse que la mayoría de archivos se puedan reconstruir.

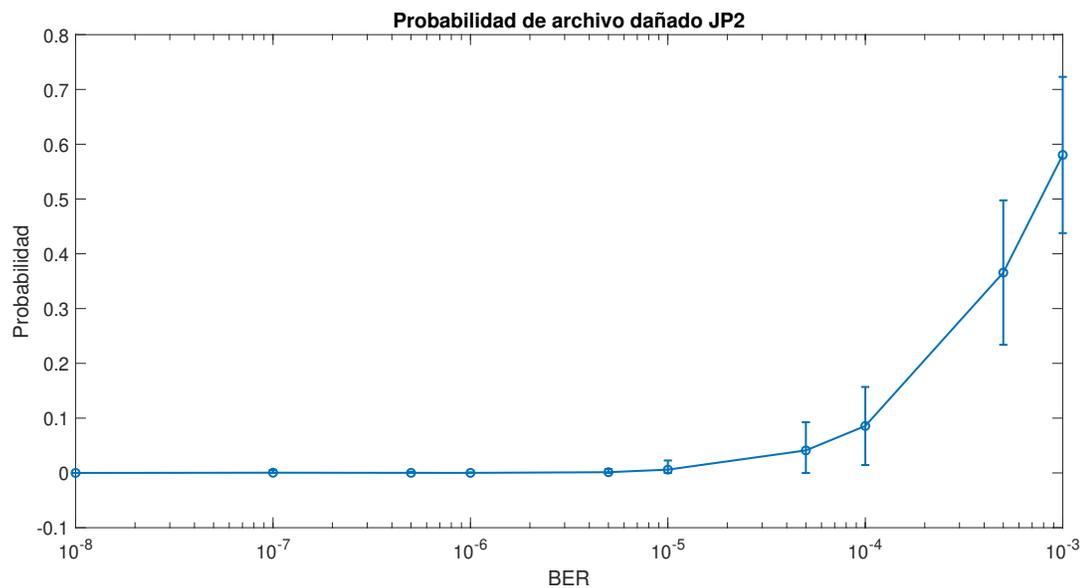
### 5.5.2. JP2

En la Figura 41 se muestran los resultados para el formato JP2. Se tiene un comportamiento muy similar a JPG, en este caso para  $BER = 1 \times 10^{-3}$  la probabilidad de archivo dañado es de 0.58 con una desviación de 0.14. Aunque estos valores son menores a JPG, siguen siendo altos y no es viable utilizar este formato para valores



**Figura 40.** Probabilidad de archivo dañado (imposible de abrir) en el formato JPG para distintos valores de BER.

grandes de BER. A partir de un  $BER = 5 \times 10^{-5}$  la probabilidad de archivo dañado es muy baja con tendencia a cero mientras mas disminuye el BER.

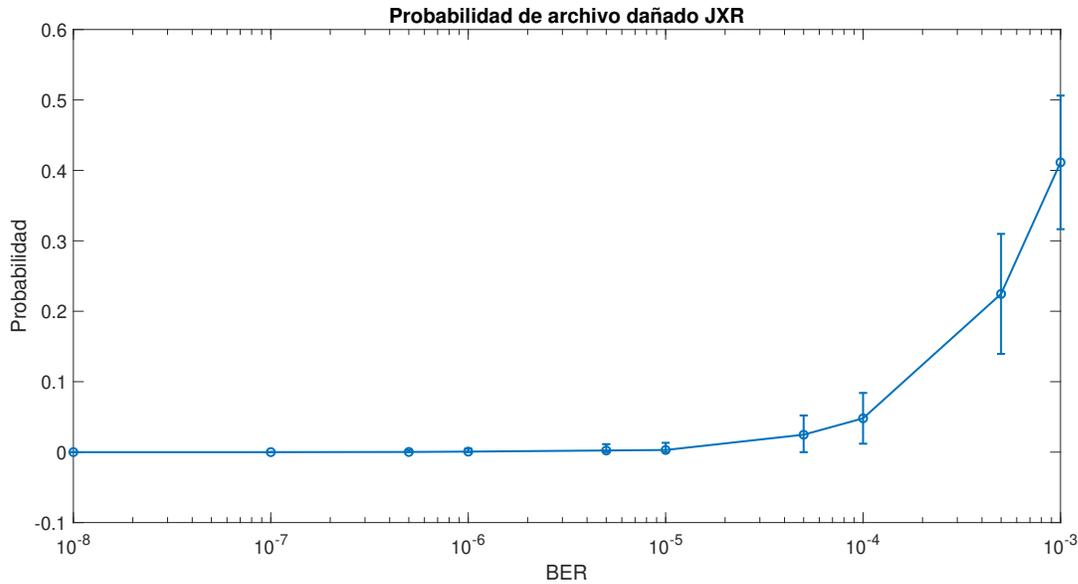


**Figura 41.** Probabilidad de archivo dañado (imposible de abrir) en el formato JP2 para distintos valores de BER.

### 5.5.3. JXR

Este formato de compresión es el que presenta la probabilidad de archivo dañado mostrada en la Figura 42. Tiene un valor de 0.41 y una desviación de 0.09 para  $BER = 1 \times 10^{-3}$ , su comportamiento es similar al caso de JPG y JP2 pero con un mejor desempeño, pues a partir de un  $BER = 1 \times 10^{-4}$  la probabilidad de archivo dañado es

menor al 5%. Para un valor de  $BER = 1 \times 10^{-5}$  en adelante esta probabilidad es casi cero.



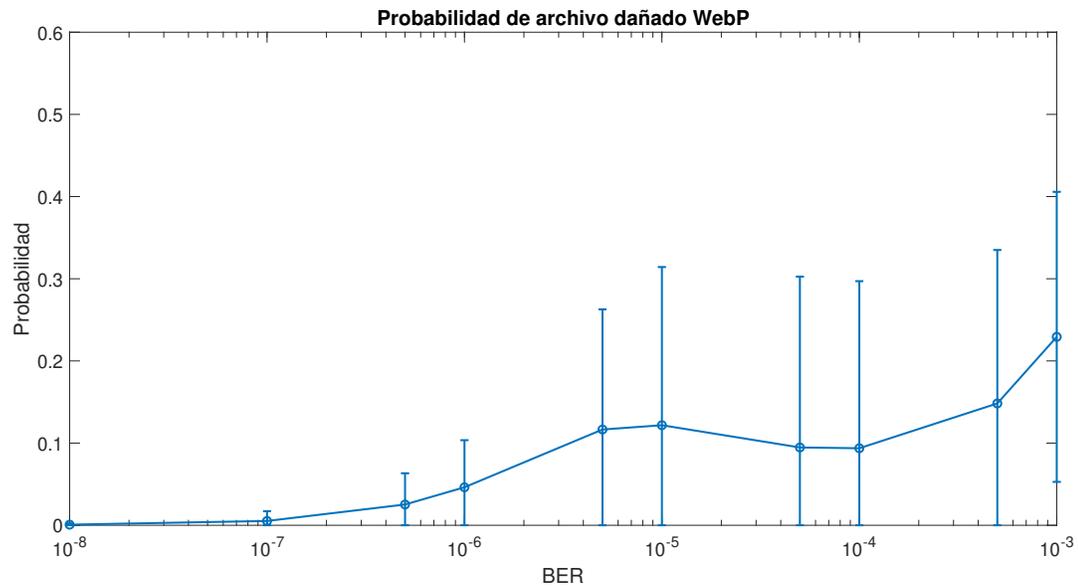
**Figura 42.** Probabilidad de archivo dañado (imposible de abrir) en el formato JXR para distintos valores de BER.

#### 5.5.4. WebP

Como se muestra en la Figura 43, este formato de compresión es el que presenta una probabilidad de archivo dañado promedio más baja (0.23 para  $BER = 1 \times 10^{-3}$ ), pero también presenta la desviación estándar más grande (0.17 para  $BER = 1 \times 10^{-3}$ ), esta desviación continúa con un valor grande (Aproximadamente 0.17) hasta que se tiene un  $BER = 1 \times 10^{-6}$  o mejor. Esto nos indica que aunque en promedio se tiene una probabilidad de archivo dañado baja, en ciertos archivos se tiene una probabilidad alta, esto sucede principalmente cuando los archivos tienen un tamaño menor a 20 KB, lo cual es el caso para algunos archivos en formato WebP.

#### 5.6. Evaluación con archivos del mismo tamaño (50 KB)

La última evaluación que se llevó a cabo fue repetir lo presentado en la sección 5.4, pero esta vez utilizando un lote de imágenes que intencionalmente han sido com-

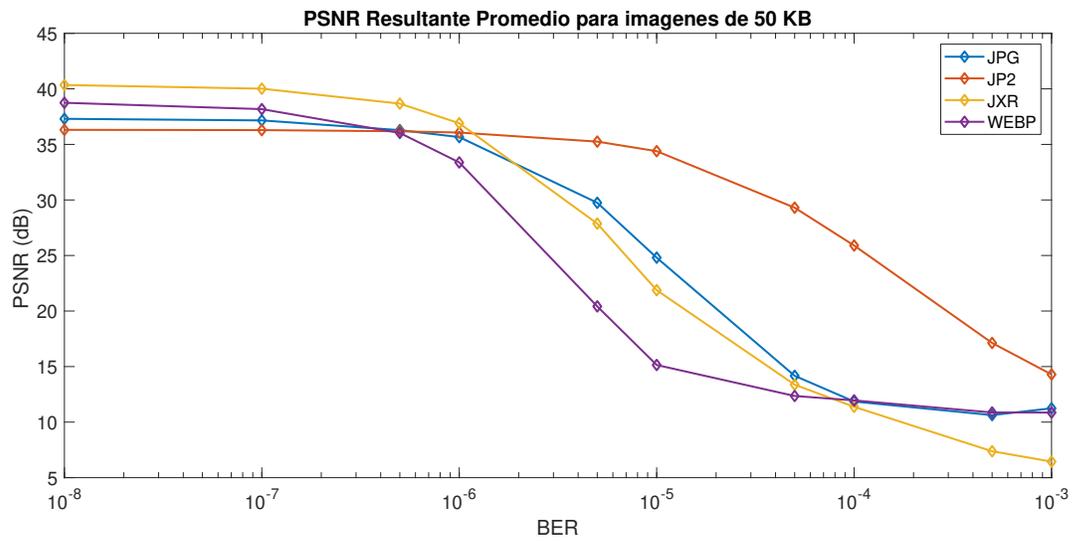


**Figura 43.** Probabilidad de archivo dañado (imposible de abrir) en el formato WebP para distintos valores de BER.

primidas para tener el mismo tamaño, sin importar el formato de compresión. Esto debido a que en la evaluación anterior se observó que el tamaño del archivo juega un papel importante en el hecho de que la imagen se pueda recuperar de forma correcta. Se decidió utilizar imágenes con un tamaño de 50 KB. Los resultados de su PSNR promedio se muestran en la Figura 44.

Como se puede ver, el comportamiento de cada uno de los formatos es el mismo al presentado en la Figura 39, con la diferencia de que el PSNR de referencia es diferente para cada uno de los formatos (esto se debe a la tasa de compresión diferente que presenta cada uno).

En este caso podemos observar que al tener archivos del mismo tamaño, el formato de compresión JP2 presenta un mejor desempeño al tener valores de BER entre  $1 \times 10^{-3}$  y  $1 \times 10^{-6}$ , mientras que el formato JXR tiene mejores resultados desde  $BER = 1 \times 10^{-6}$  hasta  $BER = 1 \times 10^{-8}$ .



**Figura 44.** PSNR promedio en función del BER para archivos de 50 KB de tamaño en los 4 formatos de compresión evaluados.

## Capítulo 6. Conclusiones y trabajo futuro

---

### 6.1. Conclusiones

Al terminar de evaluar los 4 formatos de compresión de imágenes a través de todas las pruebas que se establecieron al inicio de este trabajo, no podemos declarar un absoluto ganador. Sin embargo, sí podemos definir los formatos que mejor trabajan en condiciones específicas.

Al hablar de tasa de compresión, quedó definido que los formatos que mejor desempeño tienen son JXR y WebP, pues podemos lograr que tengan los archivos más pequeños aún manteniendo el  $PSNR \approx 38$  dB que usamos de referencia en todas las pruebas por ser equivalente a un  $MOS=5$  (imagen con excelente calidad).

En cuanto a complejidad numérica tenemos que, de nuevo, JXR es uno de los formatos que requieren menos poder computacional junto con JPG, teniendo un tiempo de ejecución de hasta un 85 % más rápido que JP2, el cual resultó ser el formato más complejo.

Las pruebas de calidad subjetiva parecen indicar que el mejor formato es JPG. Sin embargo, la prueba arroja resultados similares, ya que los 4 formatos se podrían apreciar con una calidad muy similar en nuestros resultados. Esta similitud nos lleva a considerar que la calidad perceptual de estos 4 formatos no es un factor muy relevante para definir cuál de ellos es el mejor. Además, esta evaluación se hizo con el principal objetivo de observar si al tener un PSNR igual en todos los formatos de compresión, se tiene la misma calidad subjetiva. Sin embargo, por cuestiones de tiempo y complicaciones relacionados a la pandemia de covid-19, la prueba subjetiva no tuvo la formalidad para ser considerada como definitiva. Se realizó una prueba rápida y por cuestiones de tiempo y recursos no se hizo en condiciones controladas (en las mismas condiciones de iluminación, usando una pantalla calibrada y la misma en todos los casos, etc). Por esta razón, consideramos que los resultados no son completamente confiables, pero nos dan una buena aproximación para el propósito de este trabajo.

En la robustez a la pérdida de datos, se obtuvo que en general JP2 es el formato con mejor desempeño, seguido de JXR. De esto podemos concluir que si la prioridad al descargar datos es tener un formato muy robusto a la pérdida de bits, sin importar

mucho los otros factores presentados aquí, la opción ideal sería utilizar el formato JP2. Pero si tomamos en cuenta todos los factores que se estudiaron, JXR es el que presenta mejor desempeño en la mayoría de las pruebas, lo que lo hace ser la mejor opción en general para la descarga de imágenes.

El formato más antiguo de los analizados es JPG, originalmente propuesto hace casi 30 años. Aunque durante este tiempo han surgido varios otros estándares y formatos de compresión de imágenes de mejor desempeño y mejores prestaciones, es importante señalar que JPG continua siendo bastante competitivo, como lo demuestran las pruebas realizadas. Además, sigue teniendo mucha más aceptación en el manejo de imágenes raster para la electrónica de consumo, con respecto a los otros formatos más recientes que se han analizado.

## **6.2. Trabajo a futuro**

- En este trabajo se evaluaron 4 formatos de compresión de código fuente abierto o de uso libre para fines no comerciales, que son utilizados actualmente en las comunicaciones o que son novedosos y podrían presentar ventajas sobre los utilizados normalmente. El formato JPEG XL ha sido diseñado para convertirse en el remplazo universal de formato de compresión de imágenes de tipo raster, sin embargo, el formato actualmente está en proceso de definición y no existían bibliotecas de software gratuitas que lo soporten por completo durante el desarrollo de este trabajo de tesis. Por esta razón como trabajo futuro se propone realizar la evaluación incluyendo el formato de compresión JPEG XL una vez que se cuente con bibliotecas que soporten este novedoso formato.
- Retomar el modelo del canal en simulink e incluir la corrección del efecto Doppler para hacer la simulación y obtener una simulación mas apegada a la realidad que permita evaluar la robustez a la pérdida de bits.
- Implementar el protocolo AX.25 en el modelo satelital y la transmisión de datos en paquetes, ya que este protocolo es globalmente usado por los usuarios radioaficionados por satélite y por casi todas las estaciones terrenas.
- Implementar en el modelo satelital algún tipo de codificación de canal para la detección y corrección de errores producidos en el canal de comunicación.

## Literatura citada

- Acharya, T. y Tsai, P.-S. (2005). *JPEG2000 standard for image compression: concepts, algorithms and VLSI architectures*. John Wiley & Sons.
- Alakuijala, J. y Rabaud, V. (2017). Lossless and transparency encoding in webp. Consultado en 2021 de: <https://developers.google.com/svgo/>.
- Alonso, M., García, E., Pacheco, E., y Mendieta, F. (2012). Plataforma educativa para el desarrollo de satélites pequeños basada en software y hardware libre. *Foro de Investigación y Cuerpos Académicos, FICA-2012*, **3**: 07.
- Annadurai, S. (2007). *Fundamentals of digital image processing*. Pearson Education India.
- Antonini, M., Barlaud, M., Mathieu, P., y Daubechies, I. (1992). Image coding using wavelet transform. *IEEE Transactions on image processing*, **1**(2): 205–220.
- Arias, M. y Aguado, F. (2016). Small satellite link budget calculation. *Inf. téc. Universidade de Vigo*.
- Armenta, A., Serrano, A., Conte, R., y V., R. (2016). *Guía de orientación regulatoria para satélites pequeños no sujetos a coordinación*. Fondo Sectorial de Investigación, Desarrollo Tecnológico e Innovación en Actividades Espaciales CONACYT - AEM. p. 116.
- Aréchiga, R. O. (2015). *Diseño y evaluación de un sistema de telemetría para un nanosatélite*. Tesis de maestría, Centro de Investigación Científica y de Educación Superior de Ensenada.
- Bankoski, J., Koleszar, J., Quillio, L., Salonen, J., Wilkins, P., y Xu, Y. (2011). Vp8 data format and decoding guide. En: *RFC 6386*.
- Barnsley, M. F. y Hurd, L. P. (1993). *Fractal image compression*, Vol. 366. AK peters Wellesley.
- Blanes, I., Magli, E., y Serra-Sagrasta, J. (2014). A tutorial on image compression for optical space imaging systems. *IEEE Geoscience and Remote Sensing Magazine*, **2**(3): 8–26.
- Bouwmeester, J. y Guo, J. (2010). Survey of worldwide pico-and nanosatellite missions, distributions and subsystem technology. *Acta Astronautica*, **67**(7-8): 854–862.
- Breed, G. (2003). Bit error rate: Fundamental concepts and measurement issues. *High Frequency Electronics*, **2**(1): 46–47.
- Carey, C. (2017). What is the difference between vector and raster graphics? Consultado en 2021 de: <https://www.deverdesigns.com/what-is-the-difference-between-vector-and-raster-graphics/>.
- Chin, A., Coelho, R., Nugent, R., Munakata, R., y Puig-Suari, J. (2008). Cubesat: the pico-satellite standard for research and education. En: *AIAA Space 2008 Conference & Exposition*. p. 7734.
- Clemence, A. J. (2008). *Error resilient JPEG2000 transmission over wireless fading channels*. Tesis de maestría, University of Southern Queensland Faculty of Engineering and Surveying.

- Clydespace (2020). Pulsar tmtc. Consultado en 2021 de: <https://www.aac-clyde.space/satellite-bits/communications>.
- De Simone, F., Goldmann, L., Baroncini, V., y Ebrahimi, T. (2009). Subjective evaluation of jpeg xr image compression. En: *Applications of Digital Image Processing XXXII*. International Society for Optics and Photonics, Vol. 7443, p. 74430L.
- Dhawan, S. (2011). A review of image compression and comparison of its algorithms. *International Journal of electronics & Communication technology*, **2**(1): 22–26.
- Direct, M. F. (2021). Eb-432 eggbeater antenna. Consultado en 2021 de: <https://www.m2inc.com/FGEB432RK70CM>.
- Dosselmann, R. y Yang, X. D. (2011). A comprehensive assessment of the structural similarity index. *Signal, Image and Video Processing*, **5**(1): 81–91.
- Dufaux, F., Sullivan, G. J., y Ebrahimi, T. (2009). The jpeg xr image coding standard [standards in a nutshell]. *IEEE Signal Processing Magazine*, **26**(6): 195–204.
- Eckert, M. P. y Bradley, A. P. (1998). Perceptual quality metrics applied to still image compression. *Signal processing*, **70**(3): 177–200.
- eoPortal (2010). Swampsat. Consultado en 2021 de: <https://directory.eoportal.org/web/eoportal/satellite-missions/s/swampsat>.
- Fernández, J. (1999). *Estudio comparativo de las técnicas de procesamiento digital de imágenes*. Tesis de maestría, Universitat Politècnica de Madrid.
- Fisher, Y. (1994). Fractal image compression. *Fractals*, **2**(03): 347–361.
- Gaysin, A., Fadeev, V., y Hennhöfer, M. (2017). Survey of modulation and coding schemes for application in cubesat systems. En: *2017 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SINKHROINFO)*. IEEE, pp. 1–7.
- Gilster, P. (2014). Sprites: A chip-sized spacecraft solution. Consultado en 2021 de: <https://www.centauri-dreams.org/2014/07/17/sprites-a-chip-sized-spacecraft-solution/>.
- Gomez, G. (2021). Raster vs vector. Consultado en 2021 de: [http://vector-conversions.com/vectorizing/raster\\_vs\\_vector.html](http://vector-conversions.com/vectorizing/raster_vs_vector.html).
- González, R. C., Woods, R. E., et al. (2002). *Digital image processing*. Prentice hall Upper Saddle River, NJ.
- Hu, Y., Wang, H., Zhang, Y., Li, B., y Gu, D. (2019). A semantics-based hybrid approach on binary code similarity comparison. *IEEE Transactions on Software Engineering*.
- IARU (2013). Satellite frequency coordination in the two-metre band. Consultado en 2021 de: [https://www.iaru.org/wp-content/uploads/2019/12/satellite\\_frequency\\_coordination\\_in\\_the\\_two.pdf](https://www.iaru.org/wp-content/uploads/2019/12/satellite_frequency_coordination_in_the_two.pdf).
- ISO, I. O. f. S. (2019). Iso/iec 15444-1:2019 information technology — jpeg 2000 image coding system. *International Organization for Standardization*, **4**: 234.
- ITU (2012). Article 5 frequency allocations. *Section IV–Table of Frequency Allocations*. Consultado en 2021 de: <https://life.itu.int/radioclub/rr/art05.htm>.

- ITU (2019). P.525-4 (08/2019), cálculo de la atenuación en el espacio libre. Consultado en 2021 de: <https://www.itu.int/rec/R-REC-P.525/es>.
- ITU-T, T. (1992). Information technology-digital compression and coding of continuous-tone still images-requirements and guidelines. *International Telecommunication Union. CCITT recommendation*, **81**: 09.
- Khan, A., Sun, L., y Ifeachor, E. (2008). An anfis-based hybrid video quality prediction model for video streaming over wireless networks. En: *2008 The Second International Conference on Next Generation Mobile Applications, Services, and Technologies*. IEEE, pp. 357–362.
- Kim, K. J., Kim, B., Choi, S. W., Kim, Y. H., Hahn, S., Kim, T. J., Cha, S. J., Bajpai, V., y Lee, K. H. (2008). Definition of compression ratio: difference between two commercial jpeg2000 program libraries. *Telemedicine and e-Health*, **14**(4): 350–354.
- Klaue, J., Rathke, B., y Wolisz, A. (2003). Evalvid—a framework for video transmission and quality evaluation. En: *International conference on modelling techniques and tools for computer performance evaluation*. Springer, pp. 255–272.
- Kulu, E. (2021). Nanosats database. Consultado en 2021 de: <https://www.nanosats.eu/>.
- Lee, S., Hutputanasin, A., Toorian, A., Lan, W., Munakata, R., Carnahan, J., Pignatelli, D., y Mehrparvar, A. (2014). Cubesat design specification, rev. 13: The cubesat program. *San Luis Obispo, California Polytechnic State University*.
- Lewis, A. S. y Knowles, G. (1992). Image compression using the 2-d wavelet transform. *IEEE Transactions on image Processing*, **1**(2): 244–250.
- Li, C. y Bovik, A. C. (2009). Three-component weighted structural similarity index. En: *Image quality and system performance VI*. International Society for Optics and Photonics, Vol. 7242, p. 72420Q.
- Lian, L. y Shilei, W. (2012). Webp: A new image compression format based on vp8 encoding. *Microcontrollers & Embedded Systems*, **3**.
- Long, E. (2006). Motorola satellite for iridium. Consultado en 2021 de: <http://airandspace.si.edu/explore-and-learn/multimedia/detail.cfm?id=1016>.
- Matlab, S. (2012). Matlab. *The MathWorks, Inc.* Consultado en 2021 de: <https://www.mathworks.com/products/matlab.html>.
- Medina, A. G. (2015). *Desarrollo de un entrenador experimental con dos ruedas de reacción para un CubeSat educativo*. Tesis de maestría, Centro de Investigación Científica y de Educación Superior de Ensenada.
- Mendieta, F., Mendoza, B., y E., P. (2010). *Foro y Mesa de Trabajo con el Tema: Investigación Científica y Tecnológica*. Agencia espacial Mexicana.
- Mrak, S. et al. (2004). Reliability of objective picture quality measures. *Journal of Electrical Engineering*, **55**(1-2): 3–10.
- Munakata, R. et al. (2009). Cubesat design specification rev. 12. *The CubeSat Program, California Polytechnic State University*, **1**.

- Muri, P. y McNair, J. (2012). A survey of communication sub-systems for intersatellite linked systems and cubesat missions. *J. Commun.*, **7**(4): 290–308.
- NASA (2008). Hubble space telescope. Consultado en 2021 de: [https://www.nasa.gov/sites/default/files/thumbnails/image/hubble\\_2.jpg](https://www.nasa.gov/sites/default/files/thumbnails/image/hubble_2.jpg).
- Pacheco, E., Mendieta, F., Vicente-Vivas, E., y Conte, R. (2009). Satellite and space communications research in Mexico: Contributions to a national program. En: *AIAA SPACE 2009 Conference & Exposition*. p. 6410.
- Pancholi, B., Shah, R., y Modi, M. (2014). Tutorial review on existing image compression techniques. *Int. J. Eng. Comput. Sci.*, **3**(8): 7882–7889.
- Popescu, O. (2017). Power budgets for cubesat radios to support ground communications and inter-satellite links. *Ieee Access*, **5**: 12618–12625.
- Rich, D. y Estrom, T. (2014). “futurists” and the future of solar energy. Consultado en 2021 de: <https://solartribune.com/what-futurists-say-about-solar/>.
- Richards, D. y Kolundzija, N. (2006). Sputnik. Consultado en 2021 de: <https://jasnavelickovic.com/sputnik.htm>.
- Richter, T. (2008). Visual quality improvement techniques of hdphoto/jpeg-xr. En: *2008 15th IEEE International Conference on Image Processing*. IEEE, pp. 2888–2891.
- Rinaldo, P. (1995). *Guía internacional del radioaficionado*. Marcombo.
- Rycroft, M. J. y Crosby, N. (2013). *Smaller Satellites: Bigger Business?: Concepts, Applications and Markets for Micro/Nanosatellites in a New Information World*, Vol. 6. Springer Science & Business Media.
- Saeed, N., Elzanaty, A., Almorad, H., Dahrouj, H., Al-Naffouri, T. Y., y Alouini, M.-S. (2020). Cubesat communications: Recent advances and future challenges. *IEEE Communications Surveys & Tutorials*, **22**(3): 1839–1862.
- Sara, U., Akter, M., y Uddin, M. S. (2019). Image quality assessment through fsim, ssim, mse and psnr—a comparative study. *Journal of Computer and Communications*, **7**(3): 8–18.
- Si, Z. y Shen, K. (2016). Research on the webp image format. En: *Advanced graphic communications, packaging technology and materials*. Springer, pp. 271–277.
- Skodras, A., Christopoulos, C., y Ebrahimi, T. (2001). The jpeg 2000 still image compression standard. *IEEE Signal processing magazine*, **18**(5): 36–58.
- Solomon, C. y Breckon, T. (2011). *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*. John Wiley & Sons.
- Still, M. (2006). *The definitive guide to ImageMagick*. Apress.
- Stremmer, F. G. (1982). *Introduction to communication systems*. Addison Wesley Publishing Company.
- Toorian, A., Diaz, K., y Lee, S. (2008). The cubesat approach to space access. En: *2008 IEEE Aerospace Conference*. IEEE, pp. 1–14.

- Torres, V. R. (2011). *Compresión y restauración de imágenes por técnicas no lineales*. Tesis de doctorado, Universitat de València. Facultat de Matemàtiques.
- Villela, T., Costa, C. A., Brandão, A. M., Bueno, F. T., y Leonardi, R. (2019). Towards the thousandth cubesat: A statistical overview. *International Journal of Aerospace Engineering*, **2019**.
- Wallace, G. K. (1992). The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, **38**(1): xviii–xxxiv.
- Wang, Z., Simoncelli, E. P., y Bovik, A. C. (2003). Multiscale structural similarity for image quality assessment. En: *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. Ieee, Vol. 2, pp. 1398–1402.
- Wei, W.-Y. (2008). An introduction to image compression. *Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan, ROC*, p. 29.
- Woellert, K., Ehrenfreund, P., Ricco, A. J., y Hertzfeld, H. (2011). Cubesats: Cost-effective science and technology platforms for emerging and developing nations. *Advances in Space Research*, **47**(4): 663–684.
- Wood, D. y Weigel, A. (2009). The evolution of satellite programs in developing countries. En: *Proceedings of the 60th International Astronautical Congress, Daejeon, South Korea*. pp. 12–16.
- Wood, D. y Weigel, A. (2012). Charting the evolution of satellite programs in developing countries—the space technology ladder. *Space Policy*, **28**(1): 15–24.
- XnView (2021). Xnview mp. Consultado en 2021 de: <https://www.xnview.com/en/xnviewmp/>.
- Zhu, J. (2002). Image compression using wavelets and jpeg2000: a tutorial. *Electronics & Communication Engineering Journal*, **14**(3): 112–121.