

---

This is the **published version** of the bachelor thesis:

Contreras Alayo, Yimi Anthomi; Vázquez Castro, María Ángeles , dir. Evaluación del efecto de la constelación Starlink sobre la visibilidad y geometría GNSS desde la Tierra. 2025. 145 pag. (Grau en Enginyeria de Sistemes de Telecomunicació)

---

This version is available at <https://ddd.uab.cat/record/318680>

under the terms of the  license



TRABAJO DE FIN DE GRADO

GRADO EN INGENIERÍA DE SISTEMAS DE TELECOMUNICACIÓN

EVALUACIÓN DEL EFECTO DE LA CONSTELACIÓN  
STARLINK SOBRE LA VISIBILIDAD Y GEOMETRÍA  
GNSS DESDE LA TIERRA

Yimi Anthoni Contreras Alayo

DIRECTOR: Ángeles Vázquez-Castro

DEPARTAMENTO DE TELECOMUNICACIÓN E INGENIERÍA DE SISTEMAS

UNIVERSITAT AUTÒNOMA DE BARCELONA

Bellaterra, Julio 16, 2025



# Abstract

The deployment of low Earth orbit (LEO) megaconstellations, such as Starlink, introduces a new paradigm for global navigation, offering improved coverage while raising emerging challenges. In particular, the physical blockage of the line of sight (LoS) to GNSS satellites by non-cooperative LEO constellations may cause geometric degradation that remains largely unquantified.

This work aims to model and systematically assess this impact by evaluating how it varies with orbital density and observer latitude. To that end, a Python-based simulator is developed using real TLE data through the `Skyfield` library. The analysis will be carried out from three geographic locations and under three satellite density scenarios, using key metrics such as the Blocking Index ( $BI_t$ ) and the percentage variation of PDOP ( $\Delta PDOP$ ), under ionosphere-free conditions.

The results are expected to reveal a direct correlation between LEO density and GNSS geometric degradation, as well as the influence of satellite positioning and the relative importance of those obstructed. This study provides quantitative insight into an emerging threat, emphasizing the need for mitigation strategies to preserve positioning integrity in an increasingly congested orbital environment.



# Resumen

El despliegue de megaconstelaciones LEO, como Starlink, representa un nuevo paradigma para la navegación global, combinando oportunidades de mejora con desafíos aún poco cuantificados. En particular, el bloqueo físico de la línea de visión (LoS) hacia satélites GNSS por parte de constelaciones no cooperativas plantea una posible degradación de la geometría de posicionamiento que requiere análisis sistemático.

Este trabajo tiene como objetivo modelar y medir dicho impacto, evaluando cómo varía en función de la densidad orbital y la latitud del observador. Para ello, se desarrolla un simulador en Python basado en datos TLE reales mediante la librería Skyfield. El análisis se realizará desde tres ubicaciones geográficas y bajo tres escenarios de densidad satelital, aplicando métricas como el Índice de Bloqueo ( $BI_t$ ) y la variación porcentual del PDOP ( $\Delta PDOP$ ), en condiciones libres de ionosfera.

Se espera demostrar una correlación directa entre la densidad LEO y la degradación geométrica GNSS, así como la influencia de la posición orbital y la importancia relativa de los satélites bloqueados. Los resultados permitirán aportar evidencia cuantitativa sobre una amenaza emergente, subrayando la necesidad de estrategias de mitigación para preservar la integridad del posicionamiento global en un entorno espacial cada vez más saturado.



# Agradecimientos

Nunca pensé que escribiría un capítulo de agradecimientos si no lo hacía en el primer libro que llegase a publicar. Como dijo una figura importante en mi vida, uno siempre debe ser agradecido con aquellas personas que te han rodeado durante el proceso que culmina en el momento en que presentas este trabajo y te despides de las últimas asignaturas que te quedan. Primero, quiero agradecer a mi tutora, Ángeles, quien aceptó tutorizarme casi al límite de tiempo y me orientó en la formación de las bases de este trabajo. A todos los profesores por cuyas aulas pasé y que me enseñaron más de lo que esperaba en un inicio. A aquellos amigos con los que me tocó debatir sobre mi tema de estudio y con los que pasé tardes en el comedor resolviendo problemas y redactando. A quienes están en el extranjero y llevo siempre en mi corazón. A mis padres, que siempre estuvieron allí para ayudarme a avanzar; a toda la familia que está aquí conmigo y la que se encuentra lejos, en especial a mi abuelo materno, sin sus palabras creo que nunca hubiera llegado a ser la persona que soy ahora. Y para finalizar, quiero agradecer a mi amiga Diana, que soportó cada día mis desvaríos y explicaciones sobre el proyecto y también me apoyó moralmente para que este fuera posible.



# Índice general

<b>Abstract</b>	I
<b>Resumen</b>	III
<b>Agradecimientos</b>	V
<b>Índice de Figuras</b>	XIV
<b>Índice de Tablas</b>	XVI
<b>1. Introducción</b>	1
1.1. Motivación y Objetivos . . . . .	1
1.2. Métricas . . . . .	1
1.3. Metodología . . . . .	2
<b>2. Estado del Arte</b>	5
2.1. Los GNSS y los LEO . . . . .	5
2.2. Estudios Previos . . . . .	6
2.2.1. Mejora geométrica y reducción del tiempo de convergencia en PPP . . . . .	6
2.2.2. Modelado de órbitas LEO mediante parámetros TLE . . . . .	8
2.2.3. Detección de interferencias GNSS desde plataformas LEO . . . . .	9
2.2.4. Evaluación comparativa de modelos PPP mejorados con LEO (LeGNSS) .	11
2.3. Modelos de simulación, métricas evaluativas y formulación estratégica del enfoque	13

---

2.3.1.	Síntesis de conclusiones clave . . . . .	13
2.3.2.	Aspectos no abordados en profundidad . . . . .	15
2.3.3.	Definición de sub-objetivos . . . . .	16
<b>3.</b>	<b>Metodología y Modelado</b>	<b>17</b>
3.1.	Planteamiento general . . . . .	17
3.2.	Flujo de trabajo de la simulación y análisis . . . . .	19
3.3.	Métricas geométricas y de rendimiento . . . . .	21
3.3.1.	Definición del PDOP . . . . .	21
3.3.2.	Variación porcentual del PDOP . . . . .	23
3.3.3.	Probabilidad e índice de bloqueo visual . . . . .	23
3.4.	Implementación para el modelado . . . . .	24
3.4.1.	Selección de puntos de observación . . . . .	24
3.4.2.	Obtención y preprocesamiento de TLEs . . . . .	30
3.4.3.	Simulación orbital GNSS y LEO . . . . .	31
3.4.4.	Detección de coincidencias angulares GNSS–LEO . . . . .	31
3.4.5.	Cálculo de métricas estadísticas . . . . .	33
3.4.6.	Estimación del impacto geométrico . . . . .	34
3.4.7.	Visualización y análisis comparativo . . . . .	35
3.5.	Entorno de desarrollo y herramientas . . . . .	36
<b>4.</b>	<b>Implementación de la simulación GNSS–LEO</b>	<b>39</b>
4.1.	Skyfield . . . . .	39
4.2.	Celestrak . . . . .	40
4.3.	Entorno de desarrollo y ejecución . . . . .	41
4.3.1.	Librerías y complementos . . . . .	42
4.4.	Obtención y procesamiento de datos . . . . .	43
4.4.1.	Carga y clasificación de constelaciones GNSS . . . . .	43

4.4.2. Descarga y filtrado de satélites LEO (Starlink) . . . . .	43
4.4.3. Selección por niveles de densidad . . . . .	45
4.5. Aplicación de observadores y eventos de bloqueo . . . . .	46
4.5.1. Simulación temporal y estructura de datos . . . . .	46
4.5.2. Detección de visibilidad GNSS . . . . .	47
4.5.3. Cálculo de posiciones LEO . . . . .	47
4.5.4. Algoritmo de detección de bloqueo visual . . . . .	47
4.5.5. Resultado y almacenamiento . . . . .	48
4.6. Cálculo de métricas GNSS–LEO a partir de la simulación . . . . .	48
4.6.1. Probabilidad de Bloqueo Individual (PBI) . . . . .	48
4.6.2. Índice Global de Bloqueo Visual (BI) . . . . .	49
4.6.3. Frecuencia Horaria de Eventos de Bloqueo . . . . .	49
4.6.4. Desglose horario por franja . . . . .	50
4.7. Cálculo del PDOP y análisis geométrico . . . . .	50
4.7.1. Estimación del PDOP en instantes representativos . . . . .	50
4.7.2. PDOP completo, filtrado y variación porcentual . . . . .	51
4.7.3. Estadísticas horarias de PDOP . . . . .	52
4.8. Generación de informes y visualizaciones avanzadas . . . . .	52
4.8.1. Informe textual automático . . . . .	52
4.8.2. Visualizer para generar gráficas . . . . .	54
4.8.3. Gráfico de evolución temporal del BI instantáneo ( $BI_t$ ) . . . . .	54
4.8.4. Análisis global: <code>generar_graficas_por_densidad()</code> . . . . .	55
<b>5. Análisis y discusión de resultados</b>	<b>57</b>
5.1. Tromsø . . . . .	57
5.1.1. Resultados y análisis por densidad orbital . . . . .	57
5.1.2. Conclusión de los resultados . . . . .	65
5.2. Quito . . . . .	65

---

5.2.1. Resultados y análisis por densidad orbital . . . . .	65
5.2.2. Conclusión de los resultados . . . . .	74
5.3. Alice Springs . . . . .	74
5.3.1. Resultados y análisis por densidad orbital . . . . .	74
5.3.2. Conclusión de los resultados . . . . .	83
5.4. Resultados por densidad . . . . .	83
5.4.1. Boxplot de PDOP completo vs. filtrado . . . . .	83
5.4.2. Stripplot de variación porcentual de PDOP ( $\Delta$ PDOP) . . . . .	84
5.4.3. Stripplot del BI <sub>t</sub> instantáneo por densidad . . . . .	85
5.4.4. Conclusión de los análisis por densidad . . . . .	86
5.5. Correlaciones clave y comparación con estudios previos . . . . .	86
5.5.1. Relaciones observadas . . . . .	87
5.5.2. Ponderación del Índice de Bloqueo . . . . .	87
5.5.3. Comparación con estudios previos . . . . .	92
5.6. Discusión final . . . . .	93
<b>6. Conclusiones y Líneas Futuras</b>	<b>95</b>
6.1. Conclusiones Principales . . . . .	95
6.2. Proyección a Futuro . . . . .	96
6.3. Limitaciones del Estudio y Futuras Líneas de Investigación . . . . .	97
6.4. Conclusión Final . . . . .	98
<b>A. Código fuente del simulador</b>	<b>99</b>
A.1. Clase principal: <code>GNSSLEOSimulator</code> . . . . .	99
A.2. Módulo de análisis de PDOP . . . . .	110
A.3. Script para visualización de bloqueos . . . . .	113
A.4. Reportes Detallados por Observador . . . . .	118
A.4.1. Observador: Tromsø, Noruega . . . . .	119

A.4.2. Observador: Quito, Ecuador . . . . .	121
A.4.3. Observador: Alice Springs, Australia . . . . .	123



# Índice de figuras

3.1.	Esquema general del proceso metodológico propuesto. . . . .	18
3.2.	Ubicación de las ciudades seleccionadas en proyección Mollweide. . . . .	29
3.3.	Celestrak: punto de partida para la descarga de TLEs. . . . .	30
3.4.	Diagrama de flujo del proceso de simulación orbital y detección de bloqueos. . . .	33
3.5.	Representación de las visualizaciones a generar. . . . .	36
3.6.	Representación del entorno, lenguaje y librerías utilizadas. . . . .	38
4.1.	Ejemplo real de TLE para la constelación Starlink. . . . .	44
5.1.	Skyplot GNSS con densidad LEO baja (Tromsø) . . . . .	58
5.2.	PDOP completo y filtrado con densidad LEO baja . . . . .	58
5.3.	Índice de bloqueo visual (BI) – densidad LEO baja . . . . .	59
5.4.	Skyplot GNSS con densidad LEO media (Tromsø) . . . . .	59
5.5.	PDOP completo y filtrado con densidad LEO media . . . . .	60
5.6.	Índice de bloqueo visual (BI) – densidad LEO media . . . . .	61
5.7.	Skyplot GNSS con densidad LEO completa (Tromsø) . . . . .	62
5.8.	PDOP completo y filtrado con densidad LEO completa . . . . .	62
5.9.	Índice de bloqueo visual (BI) – densidad LEO completa . . . . .	63
5.10.	Skyplot GNSS con densidad LEO baja (Quito) . . . . .	66
5.11.	PDOP completo y filtrado con densidad LEO baja . . . . .	66
5.12.	Índice de bloqueo visual (BI) – densidad LEO baja . . . . .	67

---

5.13. Skyplot GNSS con densidad LEO media (Quito) . . . . .	68
5.14. PDOP completo y filtrado con densidad LEO media . . . . .	69
5.15. Índice de bloqueo visual (BI) – densidad LEO media . . . . .	70
5.16. Skyplot GNSS con densidad LEO completa (Quito) . . . . .	71
5.17. PDOP completo y filtrado con densidad LEO completa . . . . .	71
5.18. Índice de bloqueo visual (BI) – densidad LEO completa . . . . .	72
5.19. Skyplot GNSS con densidad LEO baja (Alice Springs) . . . . .	75
5.20. PDOP completo y filtrado con densidad LEO baja . . . . .	75
5.21. Índice de bloqueo visual (BI) – densidad LEO baja . . . . .	76
5.22. Skyplot GNSS con densidad LEO media (Alice Springs) . . . . .	77
5.23. PDOP completo y filtrado con densidad LEO media . . . . .	78
5.24. Índice de bloqueo visual (BI) – densidad LEO media . . . . .	79
5.25. Skyplot GNSS con densidad LEO completa (Alice Springs) . . . . .	80
5.26. PDOP completo y filtrado con densidad LEO completa . . . . .	80
5.27. Índice de bloqueo visual (BI) – densidad LEO completa . . . . .	81
5.28. Boxplot de PDOP completo y filtrado según la densidad de la constelación LEO. . . . .	84
5.29. Variación porcentual de PDOP ( $\Delta$ PDOP) por densidad y observador. . . . .	85
5.30. Distribución del índice de bloqueo instantáneo ( $BI_t$ ) por densidad de constelación y observador. . . . .	86
5.31. BIP - Densidad completa para Tromsø . . . . .	89
5.32. BIP - Densidad completa para Quito . . . . .	90
5.33. BIP - Densidad completa para Alice Springs . . . . .	91

# Índice de tablas

2.1. Resumen técnico de estudios previos sobre integración LEO–GNSS . . . . .	14
3.1. Interpretación cualitativa de los valores de PDOP. . . . .	22
3.2. Evaluación comparativa de ubicaciones candidatas según criterios técnicos de selección. . . . .	27
3.3. Criterios técnicos de selección de ubicaciones finales para análisis GNSS–LEO . . .	28
4.1. Principales librerías utilizadas en el desarrollo del simulador. . . . .	42
5.1. Registros con valores $\Delta$ PDOP superiores a 1 % . . . . .	60
5.2. Registros con valores $BI_t$ superiores a 1 % . . . . .	61
5.3. Registros con valores $\Delta$ PDOP superiores a 1.2 % . . . . .	63
5.4. Registros con valores $BI_t$ superiores al 2.5 % . . . . .	64
5.5. Comparación de métricas GNSS–LEO para Tromsø según densidad orbital . . . .	64
5.6. Registros con valores $\Delta$ PDOP superiores a 0.7 % . . . . .	67
5.7. Registros con valores $BI_t$ superiores a 2 % . . . . .	68
5.8. Registros con valores $\Delta$ PDOP superiores a 1.5 % . . . . .	69
5.9. Registros con valores $BI_t$ superiores al 3 % . . . . .	70
5.10. Registros con valores $\Delta$ PDOP superiores a 3 % . . . . .	72
5.11. Registros con valores $BI_t$ superiores a 5 % . . . . .	73
5.12. Comparación de métricas GNSS–LEO para Quito según densidad orbital . . . .	73
5.13. Registros con valores $\Delta$ PDOP superiores a 0.75 % . . . . .	76

5.14. Registros con valores $BI_t$ superiores a 1.7 % . . . . .	77
5.15. Registros con valores $\Delta PDOP$ superiores a 1.5 % . . . . .	78
5.16. Registros con valores $BI_t$ superiores a 3 % . . . . .	79
5.17. Registros con valores $\Delta PDOP$ superiores a 2.5 % . . . . .	81
5.18. Registros con valores $BI_t$ superiores a 4.5 % . . . . .	82
5.19. Comparación de métricas GNSS-LEO para Alice Springs según densidad orbital . . . . .	82
5.20. Comparativa entre $BI_t$ y $BIP_t$ en distintos contextos geométricos . . . . .	88
5.21. Registros de Tromsø con $BI_{\text{ponderado}}$ superior al 0.25 . . . . .	89
5.22. Registros de Quito con $BI_{\text{ponderado}}$ superior a 0.9 . . . . .	90
5.23. Registros de Alice Springs con $BI_{\text{ponderado}}$ superior al 0.5 . . . . .	91
5.24. Valores de PDOP promedio en estaciones IGS, con y sin soporte LEO. . . . .	92

# Capítulo 1

## Introducción

### 1.1. Motivación y Objetivos

En los últimos años, el despliegue masivo de satélites en órbita baja (LEO), impulsado por proyectos como Starlink, ha transformado radicalmente el uso del espacio cercano a la Tierra. Estas megaconstelaciones, diseñadas principalmente para ofrecer conectividad global, introducen nuevos desafíos en términos de gestión del tráfico orbital, sostenibilidad y posible interferencia con sistemas satelitales preexistentes.

Uno de los sistemas potencialmente más afectados por esta transformación es el conjunto de Sistemas Globales de Navegación por Satélite (GNSS del inglés *Global Navigation Satellite System*), cuya precisión y fiabilidad dependen en gran medida de una visibilidad directa y constante con múltiples satélites situados en órbitas medias o altas. Aunque existen numerosos estudios sobre el uso de satélites LEO como apoyo a los sistemas GNSS, son menos comunes los que analizan sus posibles efectos adversos, como el bloqueo visual o la degradación de la geometría de posicionamiento.

Por ello, el objetivo principal de este trabajo es analizar si el crecimiento acelerado de las constelaciones de satélites LEO, como es el caso de Starlink, puede comprometer la visibilidad efectiva de los satélites GNSS desde la superficie terrestre. Se plantea la hipótesis de que, ante determinados niveles de densidad orbital, pueden producirse bloqueos visuales que afecten negativamente la geometría del posicionamiento y, por lo tanto, reduzcan la precisión del sistema.

### 1.2. Métricas

Para abordar este análisis, se llevará a cabo una simulación basada en datos TLE reales, centrada en diferentes ubicaciones geográficas, con especial atención a la estimación de la probabilidad de

bloqueo y su relación con el valor del PDOP como indicador clave de degradación geométrica.

- Número de satélites GNSS visibles: cantidad de satélites GNSS detectables desde el punto de observación, por encima de un umbral mínimo de elevación, lo cual permite definir la cobertura efectiva del sistema en condiciones normales y durante eventos de bloqueo.
- Coincidencias angulares GNSS–LEO: eventos en los que un satélite LEO se alinea angularmente con un GNSS desde el punto de vista del observador, lo que puede provocar un bloqueo visual temporal de la señal GNSS.
- Probabilidad de bloqueo visual: porcentaje de tiempo en que un satélite GNSS determinado permanece oculto por un LEO en un intervalo de observación, permitiendo evaluar el impacto en la disponibilidad de la constelación.
- PDOP (*Position Dilution of Precision*): métrica que refleja la calidad geométrica de la distribución de satélites GNSS visibles en el cielo; su análisis en momentos de mayor y menor bloqueo permite observar si se produce una degradación significativa del posicionamiento.
- Skyplots y gráficas de datos: representaciones que ilustran la posición relativa de los satélites en el firmamento y permiten visualizar de forma intuitiva las variaciones de cobertura y los eventos de bloqueo según el lugar y la hora.

Estas métricas no se analizan de forma aislada, sino como un sistema interrelacionado: las coincidencias angulares con satélites LEO pueden reducir el número efectivo de GNSS disponibles, lo que a su vez impacta en la distribución espacial de la constelación y, por ende, en el valor del PDOP, cuya degradación implicaría la pérdida de precisión en el posicionamiento.

### 1.3. Metodología

A continuación, se describe la secuencia metodológica planteada para el desarrollo del proyecto:

- Determinar los satélites GNSS visibles desde cada ubicación seleccionada (ciudad en latitud polar, un punto en el ecuador y una zona rural aislada), utilizando una ventana de observación de 24 horas y un umbral de elevación definido.
- Seleccionar inicialmente una capa orbital baja de Starlink para evaluar su impacto de forma controlada sobre la visibilidad GNSS.
- Detectar coincidencias angulares entre satélites GNSS y LEO a lo largo del día, y calcular a partir de ellas la probabilidad de bloqueo global y por satélite. Se generarán salidas interpretables del tipo:

“Desde el punto de observación de la ciudad  $X$ , hubo un bloqueo GNSS del  $Y\%$ . El satélite más afectado fue  $Z$ , con un  $W\%$  de bloqueo individual.”

- Evaluar el PDOP estimado en distintos momentos representativos del día (correspondientes a condiciones de máximo, mínimo e intermedio nivel de bloqueo) para detectar posibles degradaciones geométricas.
- Visualizar los resultados mediante skyplots y mapas de visibilidad que reflejen las variaciones angulares y espaciales registradas durante la simulación.
- Aumentar progresivamente la densidad orbital de Starlink, incorporando más capas hasta simular un escenario de densidad media y, si es posible, la constelación completa. Esto permitirá comparar el impacto acumulado en distintos niveles de saturación.

Todo el proyecto se llevará a cabo utilizando herramientas de programación científica que permitan simular de forma precisa el comportamiento orbital de los satélites. En particular, se empleará el lenguaje Python junto con la librería Skyfield, especializada en el cálculo astronómico y orbital a partir de los ya mencionados TLE (Two-Line Element sets). Los TLE utilizados corresponderán a datos reales actualizados de la constelación Starlink, extraídos de fuentes públicas como Celestrak o la propia base de datos de SpaceX.

Skyfield permitirá modelar dinámicamente las posiciones tanto de los satélites GNSS como de los LEO, y calcular ángulos de elevación, posiciones relativas y condiciones de visibilidad desde puntos específicos de la superficie terrestre. La implementación del análisis, el cálculo de métricas (visibilidad, coincidencias angulares, PDOP, etc.) y la generación de visualizaciones se realizará íntegramente en Python, utilizando bibliotecas como NumPy, Matplotlib y herramientas geoespaciales para representar los resultados de forma clara y visual.



## Capítulo 2

# Estado del Arte

Este capítulo traza un recorrido por las investigaciones más relevantes sobre la relación entre constelaciones LEO y sistemas GNSS, incluyendo tanto enfoques que exploran su integración como medio para mejorar la precisión del posicionamiento, como estudios que advierten de los riesgos asociados al crecimiento descontrolado del tráfico satelital. A lo largo del análisis, se pondrá especial atención en los modelos, métricas y aproximaciones empleados en la literatura reciente, lo que permitirá delimitar el enfoque técnico de este trabajo y justificar la elección metodológica adoptada.

### 2.1. Los GNSS y los LEO

Los sistemas globales de navegación por satélite (GNSS), como GPS (EE.UU.), GLONASS (Rusia), Galileo (UE) o BeiDou (China), consisten en constelaciones de satélites en órbita media terrestre (MEO) (aprox. 20.000 km de altitud) que transmiten señales de radio-navegación hacia la Tierra. Estos satélites se distribuyen en múltiples planos orbitales para garantizar cobertura global; típicamente hay entre 24 y 30 satélites operativos por sistema, permitiendo que un usuario en tierra reciba al menos cuatro señales para calcular su posición tridimensional.

Las órbitas MEO  $\approx$  20.000 km implican períodos orbitales de  $\approx$  12 horas, de modo que la geometría de visibilidad desde un punto en la Tierra cambia lentamente a lo largo del día. Las señales GNSS en banda L (1-1,5 GHz) llegan a la superficie con potencia muy débil (equivalente al brillo de una bombilla de 50 W vista desde 2000 km) [1], lo que las hace susceptibles a interferencias, pero suficientemente estables para lograr precisiones métricas en receptores de navegación estándar.

En contraste, las constelaciones de satélites en órbita terrestre baja (LEO) operan a altitudes mucho menores (unos 500-1600 km). Ejemplos destacados son las mega-constelaciones como

Starlink de SpaceX (con  $\approx$  550 km de altitud orbital) y OneWeb ( $\approx$  1200 km) [2, 3]. Estos satélites LEO se mueven mucho más rápido (órbitas de  $\approx$  90 minutos) y cubren áreas más pequeñas de la Tierra, por lo que se requieren cientos o miles de satélites para lograr cobertura continua global [4].

SpaceX ha lanzado más de 4.000 satélites Starlink en 2023 y actualmente lleva más de 6.750 satélites en órbita, prestando servicio a millones de clientes alrededor del mundo, y con planes de expandir su constelación a  $\approx$  42.000 unidades [5, 6], distribuidas en decenas de planos orbitales.

OneWeb, por su parte, ha desplegado  $\approx$  630 satélites operativos en órbitas polares a 1200 km, con el objetivo de proporcionar cobertura casi global. En general, un satélite LEO “ve” un área más reducida, pero emite una señal más intensa debido a su proximidad a la Tierra: a 500 km de altitud, la potencia recibida puede ser órdenes de magnitud mayor (hasta  $\approx$  30 dB más) que la de un satélite GNSS en MEO [4, 7], lo cual favorece las comunicaciones y, potencialmente, también la navegación.

Sin embargo, la baja altitud implica que cada satélite permanece visible durante un intervalo de tiempo limitado y a elevaciones relativamente bajas desde el punto de observación en tierra, entrando y saliendo del horizonte rápidamente. Por esta razón, las constelaciones LEO deben contar con un número elevado de satélites para garantizar que, en todo momento, haya varios visibles sobre el horizonte de un usuario (de lo contrario, la señal se interrumpe al ocultarse un satélite).

En resumen, GNSS y LEO difieren en régimen orbital (MEO vs. LEO), número de satélites y propósito principal: los sistemas GNSS proporcionan posicionamiento global con alta estabilidad temporal, mientras que las constelaciones LEO están orientadas típicamente a comunicaciones de baja latencia y alta capacidad, aunque sus características orbitales abren nuevas oportunidades y desafíos en el ámbito de la navegación.

## 2.2. Estudios Previos

Diversos trabajos han investigado la interacción entre constelaciones GNSS y LEO, ya sea aprovechando satélites LEO como complemento para la navegación, o evaluando posibles impactos adversos. Los proyectos analizados y las conclusiones obtenidas se explican a continuación:

### 2.2.1. Mejora geométrica y reducción del tiempo de convergencia en PPP

Un estudio en 2023 demostró que incorporar observaciones de satélites LEO puede acelerar drásticamente el Posicionamiento Puntual Preciso (PPP) de GNSS [8]. El PPP es una técnica de posicionamiento de alta precisión que, a diferencia de los métodos diferenciales como RTK, no

requiere estaciones base locales, sino que utiliza modelos precisos de órbita y reloj satelital junto con observaciones de fase portadora para alcanzar precisiones centimétricas o incluso milimétricas [9]. En este estudio se simuló una constelación global de 180 satélites LEO con órbitas polares e inclinadas para evaluar su contribución al PPP rápido.

El PPP se resolvió en tres modos: solución flotante (sin fijar ambigüedades), PPP-AR (con resolución de ambigüedades enteras) y PPP-RTK (asistido por una red de referencias). La Tabla 1 del estudio [8] resume los modelos y estrategias adoptados para integrar LEO en cada caso, asegurando que el receptor estime posición 3D y, cuando aplique, ambigüedades de fase, reloj receptor y otros parámetros conforme al modelo estándar de observación extendido con LEO.

Las métricas de rendimiento muestran mejoras contundentes al añadir satélites LEO. En términos de geometría, la disponibilidad de  $\sim 5\text{--}11$  satélites LEO visibles por estación (según latitud) mejora la distribución espacial y reduce significativamente la Dilución de Precisión Posicional (PDOP), incluso en estaciones con constelaciones GNSS completas o con geometría desfavorable. En dichos casos, la inclusión de LEO redujo el PDOP y facilitó soluciones PPP más rápidas [8].

Consecuentemente, el tiempo de convergencia del PPP flotante (definido como alcanzar un error  $< 10$  cm en horizontal/vertical) se redujo en aproximadamente un  $\sim 90\%$  al añadir LEO, logrando convergencia en  $\sim 1\text{--}2$  minutos frente a  $> 10$  minutos en sistemas tradicionales. Del mismo modo, en PPP-AR de GPS puro, el tiempo hasta la primera solución (TTFF, por sus siglas en inglés) disminuyó en torno al  $\sim 86\%$ , y un  $\sim 83\%$  en configuraciones multi-GNSS (GPS+Galileo+BeiDou), indicando que las ambigüedades enteras pueden resolverse mucho antes con apoyo LEO [8].

En PPP-RTK, usando redes locales de referencia, se observó también una mejora en la fiabilidad de la solución: en una red de 57 km el porcentaje de soluciones fijadas correctamente subió de 86.8 % a 94.9 %, reduciendo el TTFF en aproximadamente 37 %. En una red de 110 km, la tasa de fijación pasó de 64.0 % a 88.6 %, con un TTFF  $\sim 32\%$  menor. Estas mejoras se atribuyen a que los LEO aportan observaciones redundantes y una geometría mucho más favorable (ángulos de observación variados y rápidos cambios de posición), complementando así a los GNSS tradicionales [8].

Como ventaja adicional, los LEO ayudan a decorrelacionar errores atmosféricos en soluciones avanzadas como PPP-RTK; el estudio resalta que este beneficio aumenta conforme se incrementa la distancia entre estaciones de referencia, debido a que la correlación espacial de los retardos troposféricos se reduce al incorporar satélites en órbitas bajas con geometría diversa [8]. De manera general, este trabajo demuestra claramente que una constelación LEO densa puede mejorar significativamente la robustez geométrica (menor PDOP) y acelerar dramáticamente los tiempos necesarios para obtener soluciones precisas.

No obstante, es importante subrayar que dicho estudio se basa en simulaciones ideales y asu-

me condiciones operativas óptimas. En escenarios reales, implementar tales beneficios requeriría receptores duales o de triple frecuencia con capacidad para rastrear satélites LEO con movimientos rápidos, así como abordar desafíos técnicos adicionales como la gestión precisa de sesgos inter-sistema y la integración efectiva de nuevas señales y códigos [8].

En contraste con dicho enfoque orientado hacia la mejora del posicionamiento mediante LEO, en este proyecto adoptamos una perspectiva distinta: evaluar específicamente los potenciales impactos derivados de la interacción geométrico-visual entre la constelación Starlink (LEO) y las constelaciones GNSS existentes. En este sentido, el estudio citado proporciona una base conceptual y metodológica relevante para entender cómo cambios en la distribución espacial de satélites afectan directamente a métricas geométricas clave, permitiéndonos justificar sólidamente nuestro análisis de impacto visual negativo a través del mismo marco teórico del PDOP, pero desde una perspectiva inversa y novedosa.

### 2.2.2. Modelado de órbitas LEO mediante parámetros TLE

La ausencia de constelaciones LEO operativas dedicadas a navegación ha motivado evaluaciones mediante simulación. Un trabajo de 2024 propone un mapeo de parámetros orbitales TLE (Two Line Element) al formato de efemérides GNSS (RINEX 4) [10]. Este método permite aprovechar los TLE disponibles públicamente (p. ej., de Starlink o satélites de observación terrestre), convirtiéndolos en efemérides compatibles con herramientas GNSS e integrando así fácilmente las órbitas LEO reales en simuladores de navegación.

Aunque los TLE están afinados para la propagación SGP4 y no garantizan precisión orbital a nivel de metros, el estudio verificó con la constelación Spire LEMUR que las discrepancias frente a efemérides precisas (SP3) son del orden de decenas de kilómetros, lo cual se considera aceptable para evaluaciones cualitativas de rendimiento. En otras palabras, pese a la pérdida de precisión fina, las órbitas derivadas de TLE representan de forma realista la posición de cientos de satélites LEO con suficiente fidelidad para estudiar cobertura y geometría [10].

Como validación, se llevó a cabo un análisis de visibilidad y DOP combinando GNSS con la mega-constelación Starlink ( $\approx 5000$  satélites). Los resultados confirman las expectativas: la integración de una constelación LEO masiva incrementa sustancialmente la cantidad de satélites en vista desde tierra en cada instante, mejorando la cobertura global y reduciendo los valores de Dilución de Precisión [10].

En escenarios simulados para Barcelona (latitud media) y Tromsø (alta latitud), la combinación GNSS+Starlink garantizó visibilidad casi continua de un gran número de satélites en todas las elevaciones, algo imposible solo con GNSS de órbita media. Consecuentemente, métricas geométricas como el PDOP se reducen drásticamente al añadir Starlink. El estudio reporta que la fusión de GNSS con  $\sim 5000$  LEO mejora la robustez del posicionamiento, disminuyendo los valores

de DOP y potencialmente acelerando los tiempos de convergencia de soluciones precisas [10].

Si bien no se computó directamente el tiempo de convergencia PPP (al no generarse mediciones de pseudorango en este trabajo), la reducción de DOP y la cobertura omnipresente sugieren que cualquier algoritmo de posicionamiento (PPP, filtrado, etc.) se beneficiaría de convergencias más rápidas y mayores redundancias observacionales. Como aporte metodológico clave, esta referencia proporciona un enfoque práctico para construir una base de datos completa de órbitas LEO reales a partir de TLE transformados en efemérides compatibles con herramientas GNSS [10], lo que facilita simulaciones reproducibles con datos actualizados.

Esta metodología resulta particularmente útil para nuestro proyecto, ya que utilizaremos precisamente TLE reales de la constelación Starlink junto a la librería Skyfield para simular dinámicamente la posición y trayectoria de cada satélite en el cielo. La advertencia sobre la precisión limitada de las órbitas derivadas de TLE (con errores típicos del orden de decenas de kilómetros) representa un riesgo metodológico aceptable en nuestro contexto, pues aunque introduce incertidumbre en la posición exacta de cada satélite LEO, sigue siendo suficiente para evaluar visualmente el impacto geométrico de dichos bloqueos. En otras palabras, el compromiso entre precisión y disponibilidad de datos TLE es razonable y no impide obtener conclusiones robustas sobre patrones generales de visibilidad e interacción orbital.

Este estudio nos brinda un sustento técnico fundamental para integrar datos orbitales reales de Starlink en nuestras simulaciones. A partir de ello, podremos identificar y cuantificar eventos específicos en los que la presencia masiva de satélites LEO no necesariamente mejore, sino que perjudique la cobertura o degrade la calidad geométrica GNSS, particularmente aumentando la Dilución de Precisión Posicional (PDOP) como consecuencia directa de bloqueos visuales desde observadores terrestres fijos [10].

### 2.2.3. Detección de interferencias GNSS desde plataformas LEO

Otro aspecto crítico en la interacción GNSS–LEO es la gestión de interferencias de señal. Investigadores de la Universidad de Bundeswehr (UniBw Múnich) han explorado un concepto innovador: utilizar satélites LEO equipados con receptores especializados para monitorizar e incluso geolocalizar emisiones interferentes que degradan las señales GNSS [11].

Este estudio de ingeniería describe la arquitectura de un sistema espacial de vigilancia de interferencias GNSS, detallando todos los bloques funcionales a bordo del satélite: antenas orientadas a la Tierra, filtros RF, conversores analógico-digitales de banda ancha y almacenamiento de datos brutos. Se definen varios modos operativos para equilibrar la sensibilidad de detección con las limitaciones de energía, memoria y enlace descendente: por ejemplo, un modo de escaneo periódico del espectro GNSS en busca de interferencias (almacenando solo resúmenes o espectros), alternando con un modo de grabación continua de alta resolución temporal cuando se detecta

un evento crítico [11].

Este diseño debe lidiar con restricciones prácticas, como capacidad de almacenamiento limitada a bordo, ancho de banda de descarga de datos y disponibilidad de enlaces inter-satélite (ISL) para coordinar constelaciones de monitoreo [11].

En cuanto a los modelos y algoritmos de detección, el enfoque combina técnicas de análisis en tiempo-frecuencia y métodos de correlación. Para detección a bordo de distintas clases de interferencia, se evalúan algoritmos como la Transformada de Fourier de ventana corta (STFT) y distribuciones tiempo-frecuencia avanzadas (p. ej. Wigner-Ville suavizada), capaces de identificar señales continuas de onda portadora, chirp lineales o múltiples emisores simultáneos [11].

Una vez detectada la presencia de interferencia, la geolocalización del emisor en tierra se plantea mediante una formación de al menos dos satélites LEO capturando la señal interferente de forma sincronizada. Cada satélite mide diferencias de tiempo de llegada (TDoA) y diferencias de frecuencia (desplazamiento Doppler, FDoA) respecto al otro satélite [11]. Estas diferencias se analizan mediante la función de ambigüedad cruzada (CAF, por sus siglas en inglés) para estimar el desfase temporal y Doppler entre ambas recepciones [11].

El estudio explica que en entornos de rápida variación (un satélite LEO se mueve  $\sim 7.5$  km/s respecto a la fuente interferente terrestre), la forma de la CAF se distorsiona: si la separación entre satélites es muy grande, en un intervalo corto la variación relativa de retardo y frecuencia es tan alta que el pico de correlación se atenúa y ensancha, dificultando la estimación precisa [11]. De hecho,模拟aron una señal interferente tipo chirp lineal y observaron que con dos satélites separados 1800 km, una integración de 20 ms sufría una grave degradación en la magnitud del pico correlativo y un ensanchamiento de lóbulos en la CAF [11].

Para cuantificar la precisión alcanzable, los autores derivan la cota inferior de Cramér-Rao (CRLB) para el error de geolocalización en función de TDoA/FDoA medidos, y luego ejecutan simulaciones Monte Carlo. Los resultados muestran que existe una distancia inter-satélites óptima en torno a  $\sim 1200$  km que minimiza el error de localización horizontal. Con  $\sim 1200$  km de separación, la mediana del error fue de apenas  $\sim 14$  m, mientras que con satélites demasiado próximos (100 km) el error crecía por un factor  $> 7$  ( $\sim 100$  m). No se obtuvieron mejoras adicionales al ampliar la separación a 1800 km, corroborando que  $\sim 1200$  km equilibra bien la base de trilateración y la coherencia temporal de la señal [11].

En conjunto, este estudio [11] aporta un modelo detallado de cómo una constelación LEO podría servir como centinela global contra interferencias GNSS, una aplicación de gran relevancia para la integridad de sistemas de navegación. En términos de ventajas, ilustra que los LEO ofrecen una perspectiva única: al estar por encima de la fuente interferente, pueden cubrir amplias regiones terrestres y triangular la ubicación de un jammer con una precisión del orden de decenas de metros, algo difícil de lograr con sensores únicamente en tierra.

Sin embargo, también expone riesgos y desafíos: la alta dinámica relativa LEO-emisor exige algoritmos robustos y rápidos; además, el volumen de datos I/Q a almacenar y enviar a tierra es significativo, requiriendo gestión cuidadosa de memoria y enlaces de comunicaciones.

Aunque nuestro proyecto no se centra específicamente en detección de interferencias deliberadas, la metodología descrita aporta dos elementos de valor crítico: primero, destaca la importancia crucial de la configuración geométrica satélite–observador en términos de sensibilidad y precisión de medidas; segundo, alerta indirectamente sobre el posible impacto involuntario que podría tener una constelación LEO numerosa, como Starlink, en receptores GNSS terrestres, ya sea mediante bloqueos visuales o posibles interferencias involuntarias generadas por señales cercanas en frecuencia o reflexiones multipath orbitales. Por tanto, considerar estos factores en nuestras simulaciones fortalece significativamente la robustez y relevancia del análisis visual y geométrico que planteamos.

#### 2.2.4. Evaluación comparativa de modelos PPP mejorados con LEO (LeGNSS)

Finalmente, en 2024 se publicó un estudio enfocado en evaluar detalladamente distintos modelos de PPP mejorados con LEO, término que los autores denominan *LEO-enhanced GNSS* o LeGNSS [12]. A diferencia de trabajos previos principalmente simulados, aquí se combinaron datos reales multi-GNSS de 8 estaciones MGEX (GPS, Galileo, BDS) con observaciones simuladas de una constelación LEO de prueba, para ejecutar soluciones PPP bajo varias estrategias de procesamiento [12].

Concretamente, se compararon tres modelos de observación: (1) el modelo *ionosphere-free* (IF) clásico, que combina las dos frecuencias para cancelar el primer orden de error ionosférico; (2) un modelo no diferenciado y no combinado (UDUC), donde se utilizan observaciones crudas en múltiples frecuencias estimando los parámetros de ionosfera y sesgos asociados directamente en el filtro; y (3) un modelo *ionosphere-weighted* (IW), que integra ambas aproximaciones tratando la ionósfera como una observación adicional con varianza ponderada, funcionando como solución intermedia [12].

Los hallazgos confirman el potencial del enfoque LeGNSS y muestran que todas las variantes producen mejoras sustanciales, con diferencias menores entre ellas. En primer lugar, la geometría del sistema mejora notablemente: con la adición de satélites LEO, el PDOP promedio se reduce a  $\sim 0.5\text{--}1.2$ , lo que supone una mejora de  $\sim 38\%$  respecto a GPS únicamente, y un 22% adicional frente a la solución multi-GNSS (GPS+Galileo+BDS) sin LEO. Un PDOP tan bajo refleja una configuración casi óptima de satélites alrededor del receptor, gracias a la densidad y movilidad de los LEO.

Esta mejor distribución espacial redundante en mayor precisión posicional: en modo estático, la

incorporación de LEO logró reducir el error RMS en aproximadamente 50 % en Este, 30 % en Norte y 50 % en la componente vertical. En modo cinemático, los beneficios fueron similares en horizontal ( $\sim 35\%$ ) y aún más marcados en altitud, con una reducción de  $\sim 67\%$ , un resultado destacable considerando que la vertical suele ser la componente más débil en GNSS.

Al comparar directamente los tres modelos (IF, UDUC, IW), se observó que su desempeño en precisión final es prácticamente equivalente, lo cual sugiere que la mejora dominante proviene de la presencia de los LEO más que del modelo matemático aplicado. Sin embargo, al analizar el tiempo de convergencia, emergen diferencias relevantes: con LEO, el PPP alcanzó solución estable en  $\sim 1.1\text{--}1.3$  minutos, frente a más de 10 minutos sin LEO, lo que representa una reducción superior al 90 %.

Adicionalmente, el estudio explora la influencia de los sesgos inter-sistema (ISB) al combinar múltiples constelaciones. Aplicando constreñimientos a priori entre GPS, Galileo, BDS y LEO, se logra recortar otro  $\sim 1$  minuto en la convergencia. Esto demuestra que, si se calibran bien las diferencias de hardware y relojes entre sistemas, el filtrado PPP puede aprovechar mejor la redundancia de observaciones.

También aporta un análisis crítico de modelos para LeGNSS. Muestra que incluso el enfoque más simple (ionosphere-free clásico) ofrece resultados prácticamente indistinguibles de modelos más complejos una vez que los LEO están presentes. También destaca la importancia de tratar correctamente los ISB y aprovechar todas las observaciones disponibles. Como desafío técnico, se menciona la necesidad de receptores capaces de rastrear múltiples constelaciones y frecuencias, y de gestionar parámetros adicionales como sesgos de fase y desfases temporales entre satélites GNSS (MEO) y LEO.

Las ventajas evidenciadas —reducción drástica de PDOP, mejora de precisión 3D y convergencia casi instantánea— subrayan por qué el concepto LeGNSS ha ganado tanto interés. Para nuestro proyecto, esta referencia aporta datos cuantitativos clave y validación experimental sólida: al trabajar con TLE reales de Starlink, podemos replicar análisis de PDOP y cobertura similares, y comparar nuestras simulaciones con los valores aquí reportados. Asimismo, la comparación de modelos observacionales nos permite empezar con el enfoque IF simplificado sin comprometer la validez, siempre que consideremos los posibles sesgos entre sistemas.

En conclusión, el estudio presentado anteriormente consolida el potencial de integrar constelaciones LEO para mejorar significativamente la precisión, geometría (PDOP) y convergencia de soluciones GNSS [12]. Confirma que variaciones en la distribución espacial y número de satélites visibles impactan directamente en el PDOP. Nuestro proyecto utiliza precisamente esta relación, pero desde una perspectiva inversa, analizando cómo bloqueos visuales provocados por Starlink reducen puntualmente la cantidad de satélites GNSS disponibles, aumentando así el PDOP y degradando la geometría del posicionamiento. Este estudio ofrece, por tanto, métricas validadas

y una referencia metodológica robusta para sustentar nuestro análisis.

## 2.3. Modelos de simulación, métricas evaluativas y formulación estratégica del enfoque

A continuación, se presenta un análisis de los modelos aplicados en los estudios previos, con énfasis en las conclusiones clave extraídas de dichas investigaciones y en los elementos que sirven de base para el desarrollo del presente proyecto.

Los cuatro estudios analizados constituyen un conjunto coherente que avala el potencial de los satélites LEO como actores fundamentales en el ecosistema GNSS futuro. Aunque cada investigación aborda la integración LEO-GNSS desde ángulos distintos, emergen patrones comunes en cuanto a los beneficios técnicos y las herramientas necesarias para la simulación.

También se identifican ciertos aspectos no enfatizados en sus investigaciones; todas se centran en descubrir efectos positivos de la implementación de las mega-constelaciones, y si bien el panorama luce muy bonito, no se enfocan en los efectos negativos derivados de la proliferación y el aumento de la densidad de los satélites LEO en mega-constelaciones como Starlink, cosa que analizaremos en detalle más adelante.

### 2.3.1. Síntesis de conclusiones clave

Del análisis técnico de los estudios previos se extraen varias conclusiones clave que fortalecen el enfoque metodológico adoptado en este proyecto. Entre ellas, destacan las siguientes:

- Todos los estudios coinciden en que la incorporación de satélites LEO mejora la geometría espacial del posicionamiento (reducción del PDOP), acelera los tiempos de convergencia del PPP y aporta redundancia en las observaciones.
- La convergencia más rápida y estable se explica por una combinación de efectos: mayor diversidad angular, cobertura más densa y dinámica elevada de los LEO, lo que favorece la resolución de ambigüedades y reduce el tiempo hasta la primera solución (TTFF).
- Las tecnologías empleadas incluyen desde simulaciones ideales (sin entorno computacional explícito) hasta herramientas concretas como Skyfield, TLEs públicos, librerías de procesamiento GNSS, modelado espectral avanzado y métodos estadísticos como la cota de Cramér-Rao.
- Se reconoce de forma unánime la necesidad de hardware avanzado (receptores multibanda y multiconstelación), así como del tratamiento adecuado de los sesgos inter-sistema (ISB)

al combinar GNSS tradicionales con nuevas señales LEO.

- Se valida que el modelo de observación simplificado *ionosphere-free* (IF) es suficiente cuando el foco está en la geometría y no en el posicionamiento absoluto de alta precisión, lo que permite aplicar simulaciones eficientes con bajo coste computacional.

La Tabla 2.1 presenta una síntesis técnico-comparativa de los principales estudios revisados, estructurada en torno a tres ejes: los modelos empleados, las tecnologías utilizadas y su integración directa en nuestro trabajo.

Modelos principales	Tecnologías utilizadas	Adaptación e integración en nuestro proyecto
PPP, PPP-AR, PPP-RTK con satélites LEO simulados; estimación de ambigüedades, reloj y posición	Modelado completo de observaciones GNSS+LEO; simulación ideal sin herramienta explícita mencionada	Adoptamos el PDOP como métrica principal y aplicamos el enfoque inverso propuesto para evaluar cómo los bloqueos LEO degradan la geometría GNSS.
Análisis de visibilidad, sky-plots, cálculo de DOP con TLE reales	Skyfield (Python), TLEs desde Celestrak, conversión a RINEX, visualización propia	Usamos Skyfield con TLE reales para generar efemérides y simular visibilidad GNSS-LEO, aceptando su error como válido para análisis geométrico.
Análisis espectral + estimación TDoA/FDoA + geometría de detección desde LEO	STFT, Wigner-Ville, CAF, CRLB; separación satelital óptima simulada	Adoptamos la métrica angular LEO-GNSS $< 1^\circ$ para detectar bloqueos visuales, aplicando criterios de geometría relativa en tiempo real.
Modelos IF, UDUC, IW; comparación en precisión y tiempo de convergencia	Datos reales MGEX, PPP con simulación de LEO, ISB, estimación de errores RMS y PDOP	Validamos el modelo <i>ionosphere-free</i> como adecuado para simular PDOP; nos centramos exclusivamente en la geometría

Tabla 2.1: Resumen técnico de estudios previos sobre integración LEO-GNSS

### 2.3.2. Aspectos no abordados en profundidad

Tal como se argumentó en secciones anteriores, a continuación se identifican los aspectos que no han sido abordados con la profundidad necesaria en la literatura técnica revisada. Reconocer estas lagunas no solo refuerza la justificación del enfoque metodológico adoptado en este trabajo, sino que también permite delimitar con claridad el espacio de innovación y contribución que nuestro proyecto pretende ocupar dentro del ámbito del análisis geométrico de la coexistencia GNSS–LEO.

- Ausencia de análisis de efectos adversos visuales: Ninguno de los estudios revisados evalúa explícitamente los posibles efectos negativos que podría generar una constelación LEO masiva sobre las constelaciones GNSS existentes. En particular, se omite el análisis del fenómeno de *bloqueo visual* u *ocultación angular*, por el cual un satélite LEO, al alinearse en el cielo con uno GNSS, puede impedir momentáneamente su detección por parte de receptores terrestres. Esta omisión es relevante, ya que el fenómeno puede afectar significativamente la geometría de posicionamiento sin implicar interferencia electromagnética directa.
- Enfoque unidireccional orientado a mejoras: La mayoría de las investigaciones se centran exclusivamente en los beneficios que ofrece la integración LEO–GNSS —como mayor cobertura, mejora del PDOP o convergencias más rápidas—, sin considerar escenarios en los que esa misma integración pueda introducir vulnerabilidades geométricas o efectos contraproducentes. Se carece, por tanto, de una evaluación sistemática de posibles *anti-mejoras*, es decir, degradaciones puntuales de la solución GNSS como consecuencia de la superposición visual con satélites LEO.
- Falta de una metodología escalable de análisis por densidad: No se ha propuesto aún una estrategia parametrizada y reproducible para representar distintos niveles de densidad orbital de una mega-constelación. En particular, falta una comparación clara entre tres niveles: (i) baja densidad (una sola capa orbital de Starlink), (ii) densidad media (dos o tres capas combinadas) y (iii) densidad completa (toda la constelación). Esta ausencia limita la capacidad de simular escenarios realistas de evolución futura.
- Débil resolución espacial en simulaciones previas: Las simulaciones geométricas existentes suelen presentar resultados agregados a nivel global o en latitudes representativas, pero sin atención específica a zonas urbanas, rurales o ecuatoriales concretas. Esta generalización reduce la aplicabilidad directa de los estudios a casos de uso reales. Tampoco se proporciona una desagregación horaria o regional clara, lo cual es especialmente relevante para sistemas de navegación terrestre sensibles a obstrucciones locales o patrones atmosféricos específicos.

### 2.3.3. Definición de sub-objetivos

A partir de los aspectos identificados como insuficientemente abordados en la literatura previa, se definen a continuación una serie de sub-objetivos específicos. Estos no solo profundizan en los objetivos generales establecidos en el Capítulo 1, sino que además los concretan operativamente en coherencia con la metodología propuesta. De este modo, se garantiza una alineación precisa entre las necesidades de investigación detectadas y las acciones técnicas que articularán el desarrollo del presente proyecto.

- Definir una métrica angular de coincidencia GNSS–LEO: Se establecerá un umbral angular (por ejemplo,  $< 1^\circ$ ) como criterio para detectar eventos de bloqueo visual, basándose en la geometría relativa y la metodología propuesta por Duetsch [11], quien demostró la importancia de estos ángulos en la triangulación de señales interferentes.
- Estimar el impacto de los bloqueos sobre el PDOP: Siguiendo la metodología del estudio de Garcia Fernandez [10], se utilizarán TLEs reales junto a la librería `Skyfield` para simular la posición orbital de satélites Starlink, identificando eventos de ocultación y midiendo su efecto directo sobre la geometría GNSS (PDOP).
- Comparar múltiples escenarios de densidad LEO: Se implementarán tres niveles de densidad de la constelación LEO (configuración baja, configuración intermedia, constelación completa) y se evaluará cómo cambia el número de satélites GNSS visibles y la calidad geométrica en cada caso.
- Analizar diferencias por ubicación geográfica: Se seleccionarán tres escenarios contrastados: una ciudad en latitud alta, otra en latitud media y una zona rural con baja densidad urbana. Con el fin de comparar cómo la combinación de latitud y entorno urbano-rural influye en la visibilidad de satélites GNSS y en la magnitud de los bloqueos visuales provocados por la constelación Starlink.
- Visualizar los resultados con gráficas y skyplots: Se implementarán visualizaciones dinámicas como skyplots, gráficas de datos que permitirán analizar la variación de los parámetros dependiendo de las horas y densidades.

En conjunto, estos sub-objetivos concretan el enfoque metodológico del proyecto, permitiendo explorar dimensiones aún poco tratadas por la literatura previa y dando continuidad al análisis crítico desarrollado en el apartado anterior.

## Capítulo 3

# Metodología y Modelado

### 3.1. Planteamiento general

El presente capítulo tiene como objetivo desarrollar en profundidad la metodología que sustenta este proyecto, estructurada a partir de los objetivos generales y sub-objetivos específicos definidos previamente en los dos capítulos anteriores. A través de un enfoque sistemático, se busca abordar las lagunas identificadas en la literatura técnica sobre la integración de constelaciones LEO con sistemas GNSS, con especial énfasis en los efectos geométricos derivados de bloqueos visuales temporales.

A partir del análisis crítico de los estudios previos, se constató que la mayoría de investigaciones se centran en los beneficios que las constelaciones LEO pueden aportar a la precisión y convergencia del posicionamiento GNSS. Sin embargo, pocos trabajos han explorado los efectos negativos o contraproducentes derivados del aumento de densidad satelital en el cielo, en particular la posibilidad de que satélites LEO bloquen visualmente las señales GNSS en determinados instantes o ubicaciones. Esta omisión representa una oportunidad metodológica clara que este trabajo pretende abordar.

Con tal propósito, este capítulo articula una propuesta metodológica original y cuantitativa basada en simulaciones orbitales con datos TLE reales de la constelación Starlink. Se modelará el entorno visual del cielo desde diferentes ubicaciones terrestres, evaluando la posibilidad de que los satélites LEO interfieran visualmente con los GNSS, y analizando el impacto que esto tendría sobre la geometría de posicionamiento mediante la métrica PDOP.

En primer lugar, se presentará el modelo conceptual general del sistema GNSS–LEO observado desde tierra, así como los fundamentos geométricos del bloqueo visual. A continuación, se detallarán las métricas de rendimiento utilizadas (PDOP, índice de probabilidad de bloqueo), justificando su elección en función de su relevancia técnica y su uso en estudios previos.

Luego, se expondrá el flujo metodológico completo que se implementará en el entorno de simulación, desde la obtención y procesamiento de los TLE, pasando por el cálculo de efemérides y skyplots, hasta la estimación del PDOP bajo distintas configuraciones de densidad satelital. Se incluirán además criterios específicos para la selección de ubicaciones (latitud, entorno urbano o rural), capas orbitales a simular y escenarios de comparación.

Finalmente, se integrarán estas acciones metodológicas con cada uno de los sub-objetivos definidos, de forma que quede claramente explicitado cómo cada acción contribuye al avance del proyecto. En conjunto, este planteamiento metodológico no solo permite evaluar una dimensión poco explorada de la coexistencia GNSS–LEO, sino que también proporciona una base reproducible y extensible para investigaciones futuras.

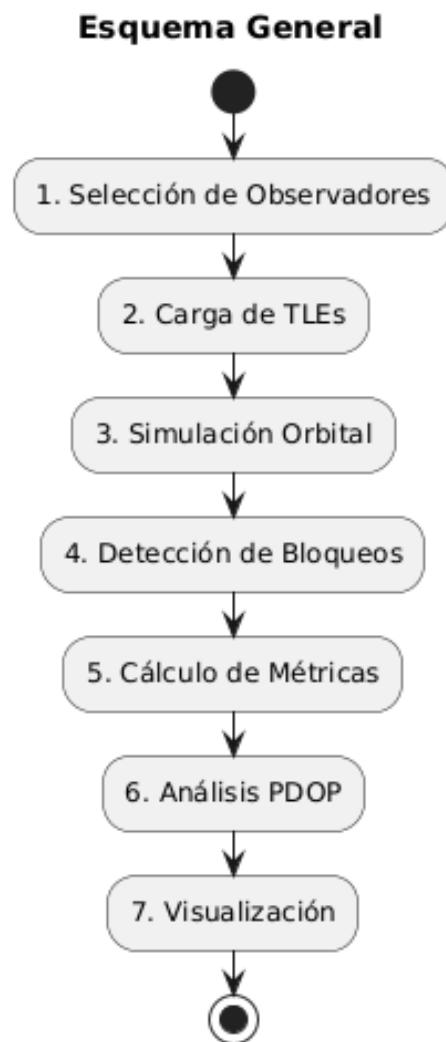


Figura 3.1: Esquema general del proceso metodológico propuesto.

## 3.2. Flujo de trabajo de la simulación y análisis

En esta sección se presenta el flujo de trabajo empleado en el modelo de simulación para evaluar el impacto visual de constelaciones LEO sobre la visibilidad y geometría de satélites GNSS desde ubicaciones terrestres fijas. A continuación, se detalla el proceso técnico adoptado:

1. Selección de puntos de observación: Se eligen ubicaciones geográficas representativas para analizar distintos escenarios de interés. En esta versión inicial se trabajará con las siguientes ciudades:
  - Tromso (latitud alta,  $\approx 69^\circ$  N): ubicación adecuada para analizar condiciones limitadas de visibilidad GNSS en zonas cercanas al polo.
  - Quito (zona ecuatorial, latitud  $\approx 0^\circ$ ): permite análisis directo y frecuente de satélites en órbitas inclinadas.
  - Alice Springs (zona rural austral, latitud  $\approx 23,7^\circ$  S): seleccionada para evaluar condiciones rurales de baja densidad urbana en latitudes medias del hemisferio sur.Estas localizaciones permiten contrastar cómo la geometría y densidad orbital afectan la visibilidad según la latitud y el tipo de entorno; se expondrá en detalle más adelante.
2. Descarga y preprocesado de TLEs reales: Se recopilan elementos orbitales actualizados (TLEs) de satélites Starlink desde la plataforma Celestrak. A partir de estos datos, se generan subconjuntos representativos de distintas densidades orbitales:
  - Densidad baja: subconjunto de  $\approx 500$  satélites.
  - Densidad media: subconjunto de  $\approx 2000$  satélites.
  - Densidad total: constelación completa actual ( $\approx 5000$  satélites).
3. Simulación orbital GNSS y LEO: Utilizando la librería Skyfield en Python, se simula el movimiento orbital de satélites GNSS (GPS, Galileo, GLONASS, BeiDou) y LEO seleccionados durante un intervalo de 24 horas. Para cada instante temporal se realiza lo siguiente:
  - Cálculo de coordenadas topocéntricas (elevación y azimut) desde cada receptor fijo.
  - Aplicación de un filtro de visibilidad GNSS considerando únicamente satélites con elevación  $> 10^\circ$ .
  - Registro de posiciones y trayectorias relativas para análisis posterior.
4. Detección de coincidencias angulares GNSS–LEO: En cada instante y para cada satélite GNSS visible:
  - Se determina el ángulo entre la línea GNSS–receptor y la línea LEO–receptor.

- Se considera bloqueo visual si dicho ángulo es menor a un umbral angular definido, típicamente  $< 1^\circ$ .
  - Se registran estos eventos, generando matrices temporales de bloqueos GNSS–LEO para cada observador.
5. Cálculo de métricas estadísticas: A partir de los datos anteriores se calculan diversas métricas para evaluar el impacto visual, tales como:
- Probabilidad de bloqueo individual por cada satélite GNSS, como el porcentaje del tiempo total durante el cual un satélite específico estuvo bloqueado.
  - Índice global de bloqueo visual, expresado como el porcentaje promedio de satélites GNSS bloqueados en cada instante respecto al total visible.
  - Promedio de bloqueo por constelación GNSS y número promedio de eventos de bloqueo por hora.
6. Estimación del PDOP en momentos clave del día: Se seleccionan instantes representativos para analizar cómo los bloqueos visuales afectan la geometría GNSS. Dichos momentos son:
- Máxima actividad de bloqueos visuales.
  - Mínima actividad de bloqueos visuales.
  - Situación intermedia representativa del comportamiento promedio.
- Para cada uno de estos escenarios se realiza un análisis comparativo del PDOP considerando la presencia y ausencia de bloqueos.
7. Visualización y análisis comparativo: Se realizan visualizaciones gráficas y análisis comparativos mediante:
- Skyplots que indican claramente posiciones de satélites GNSS bloqueados y proximidad de trayectorias LEO.
  - Tablas comparativas de métricas obtenidas según ubicación geográfica y densidad orbital LEO.
  - Gráficas que representan el número de satélites GNSS visibles por hora y la variación porcentual del PDOP durante el periodo simulado.

Cada una de estas etapas será abordada con detalle en las siguientes secciones, proporcionando definiciones explícitas, criterios técnicos precisos, fragmentos de código implementados, resultados intermedios obtenidos, y discusiones técnicas que permitirán justificar las decisiones adoptadas y la validez del análisis realizado.

### 3.3. Métricas geométricas y de rendimiento

#### 3.3.1. Definición del PDOP

La métrica principal para evaluar la calidad geométrica de una solución de posicionamiento tridimensional es el *Position Dilution of Precision* (PDOP). El PDOP pertenece a la familia de los llamados factores de dilución de precisión (DOP, por sus siglas en inglés), que cuantifican el efecto amplificador que tiene la geometría relativa de los satélites sobre los errores de posicionamiento [13, 14]. Estos factores no dependen directamente de la calidad de la señal recibida, sino exclusivamente de la disposición angular de los satélites visibles desde un receptor.

El PDOP representa específicamente cómo se diluyen los errores en las mediciones de pseudodistancia en la estimación de las coordenadas espaciales ( $x, y, z$ ). Una buena distribución de satélites —es decir, cuando están repartidos ampliamente en el cielo— da lugar a valores bajos de PDOP, lo que indica que el sistema es geométricamente robusto. En cambio, una distribución agrupada o coplanar genera un PDOP alto, reflejando mayor sensibilidad a errores.

En términos formales, el PDOP se define como:

$$PDOP = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2} \quad (3.1)$$

Donde  $\sigma_x$ ,  $\sigma_y$  y  $\sigma_z$  representan las desviaciones estándar (o varianzas si se eleva al cuadrado) en las tres dimensiones espaciales. Estas se derivan del análisis matricial del modelo linealizado del sistema GNSS, en el cual se define la matriz de geometría o matriz de diseño  $G$ :

$$\Delta\rho = G \cdot \Delta\mathbf{x} \quad (3.2)$$

Cada fila de la matriz  $G$  corresponde a un satélite visible e incluye los componentes del vector unitario desde el receptor hacia el satélite, más un cuarto término asociado a la corrección del reloj:

$$G = \begin{bmatrix} u_{1x} & u_{1y} & u_{1z} & 1 \\ u_{2x} & u_{2y} & u_{2z} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ u_{nx} & u_{ny} & u_{nz} & 1 \end{bmatrix} \quad (3.3)$$

Donde cada  $u_{ix}, u_{iy}, u_{iz}$  son los componentes del vector unitario  $\hat{\mathbf{r}}_i$  entre el receptor y el satélite  $i$ .

La matriz de covarianza de la solución estimada se obtiene como:

$$Q = (G^T G)^{-1} \quad (3.4)$$

La matriz  $Q \in \mathbb{R}^{4 \times 4}$  refleja la sensibilidad de la estimación de posición respecto a los errores en las mediciones. Sus elementos diagonales contienen las varianzas (no escaladas) de cada parámetro estimado. En particular:

- $Q_{11}, Q_{22}, Q_{33}$ : varianzas en las componentes espaciales  $x, y, z$ .
- $Q_{44}$ : varianza asociada al sesgo temporal del receptor.

A partir de estos elementos, el valor del PDOP se extrae como:

$$PDOP = \sqrt{Q_{11} + Q_{22} + Q_{33}} \quad (3.5)$$

Este valor refleja cuán fuertemente la geometría de los satélites afecta la precisión posicional. A su vez, es común descomponer el PDOP en componentes:

- **HDOP (Horizontal DOP)**: afecta a las coordenadas  $x$  e  $y$  (posición horizontal).
- **VDOP (Vertical DOP)**: afecta a la coordenada  $z$  (altitud).
- **TDOP (Time DOP)**: afecta a la estimación del reloj del receptor.

Así, se puede expresar:

$$PDOP = \sqrt{HDOP^2 + VDOP^2} \quad (3.6)$$

### 3.3.1.1. Interpretación cualitativa del PDOP

El valor numérico del PDOP, si bien es adimensional, tiene una interpretación directa en términos de confiabilidad geométrica [14]. En la práctica, se usa la siguiente escala cualitativa:

PDOP	Calidad geométrica
< 1	Ideal: geometría óptima
1~2	Excelente
2~5	Buena
5~10	Aceptable
> 10	Pobre: geometría no confiable

Tabla 3.1: Interpretación cualitativa de los valores de PDOP.

La importancia del PDOP radica en su capacidad de anticipar la precisión del posicionamiento aún antes de realizar la estimación de coordenadas. Por ello, se emplea ampliamente tanto en

planificación de misiones GNSS como en simulaciones de constelaciones satelitales, incluyendo casos donde se evalúa el efecto de posibles bloqueos visuales, como los generados por satélites LEO, o el impacto de la incorporación de nuevas capas orbitales.

### 3.3.2. Variación porcentual del PDOP

Para cuantificar el impacto geométrico de los bloqueos visuales sobre la calidad del posicionamiento, se evalúa la variación porcentual del *Position Dilution of Precision* (PDOP), una métrica que refleja la distribución espacial de los satélites GNSS en el cielo.

Cuando uno o más satélites quedan bloqueados, la geometría de observación se degrada, lo que puede traducirse en un incremento del PDOP y, por tanto, en una pérdida de precisión. La variación porcentual del PDOP se define como:

$$\Delta PDOP = \frac{PDOP_{\text{bloqueado}} - PDOP_{\text{original}}}{PDOP_{\text{original}}} \times 100 \% \quad (3.7)$$

donde:

- $PDOP_{\text{original}}$  es el valor de PDOP calculado con todos los satélites GNSS visibles.
- $PDOP_{\text{bloqueado}}$  es el valor de PDOP tras eliminar aquellos satélites que se consideran bloqueados.

Esta métrica permite estimar el deterioro de la solución de posicionamiento causado exclusivamente por el bloqueo visual de satélites.

### 3.3.3. Probabilidad e índice de bloqueo visual

La probabilidad de bloqueo visual es una métrica estadística esencial para evaluar hasta qué punto las constelaciones de satélites LEO pueden interferir con la visibilidad de los satélites GNSS desde un receptor terrestre. Esta métrica cuantifica el porcentaje del tiempo durante el cual la línea de visión entre el receptor y un satélite GNSS queda obstruida angularmente por uno o más satélites LEO.

La probabilidad de bloqueo individual para un satélite GNSS  $s_i$  se define como:

$$P_{\text{bloqueo}}(s_i) = \frac{T_{\text{bloqueado}}(s_i)}{T_{\text{total}}} \times 100 \% \quad (3.8)$$

donde:

- $T_{\text{bloqueado}}(s_i)$  es el tiempo acumulado durante el cual el satélite GNSS  $s_i$  está visualmente bloqueado por algún satélite LEO, dentro de un umbral angular inferior a  $1^\circ$ .
- $T_{\text{total}}$  es la duración total del intervalo simulado.

De forma complementaria, se puede evaluar el impacto a nivel del conjunto GNSS mediante el Índice de bloqueo visual (BI), el cual permite cuantificar el grado de afectación del sistema GNSS por bloqueos LEO. Esta métrica puede formularse desde dos perspectivas distintas:

- **Índice de bloqueo visual instantáneo:** se calcula para cada instante  $t$  como el porcentaje de satélites GNSS visibles que se encuentran bloqueados en ese momento:

$$BI_t = \frac{N_{\text{bloqueados},t}}{N_{\text{visibles},t}} \times 100 \% \quad (3.9)$$

- **Índice de bloqueo visual promedio:** representa la media temporal del índice instantáneo a lo largo del intervalo de simulación, y se expresa como:

$$BI = \frac{1}{N_{\text{instantes}}} \sum_{t=1}^{N_{\text{instantes}}} \left( \frac{N_{\text{bloqueados},t}}{N_{\text{visibles},t}} \right) \times 100 \% \quad (3.10)$$

donde:

- $N_{\text{bloqueados},t}$  es el número de satélites GNSS cuya línea de visión está obstruida por un satélite LEO en el instante  $t$ .
- $N_{\text{visibles},t}$  es el número total de satélites GNSS visibles en ese mismo instante.
- $N_{\text{instantes}}$  es el número total de pasos temporales en la simulación.

Estas métricas ofrecen enfoques complementarios: la probabilidad individual evalúa la afectación por satélite, mientras que los índices de bloqueo visual (instantáneo y promedio) reflejan la magnitud global del fenómeno, facilitando comparaciones entre ubicaciones, horarios y configuraciones orbitales LEO.

## 3.4. Implementación para el modelado

### 3.4.1. Selección de puntos de observación

El análisis del impacto de constelaciones LEO sobre sistemas GNSS desde tierra exige seleccionar ubicaciones geográficas que reflejen distintas condiciones orbitales, geométricas y ambientales.

Para ello, se establecieron una serie de criterios técnicos diseñados para guiar la elección de los puntos de observación, asegurando que las ubicaciones seleccionadas sean representativas y útiles desde el punto de vista de la simulación:

- Diversidad latitudinal: cubrir ubicaciones en zonas polares, ecuatoriales y medias para capturar la variación angular de trayectorias LEO sobre la superficie terrestre.
- Tipo de entorno: representar tanto entornos urbanos densos (donde los servicios GNSS tienen alta demanda práctica) como regiones rurales más despejadas (útiles como referencia base y para evaluar visibilidad sin interferencias estructurales).
- Cobertura orbital relevante: asegurar que la ubicación se vea atravesada con frecuencia por trayectorias de satélites LEO de distintas inclinaciones, especialmente polares, inclinadas o ecuatoriales.
- Condiciones visuales: disponer de buena visibilidad del cielo, idealmente sin obstrucciones físicas (montañas, edificios, vegetación densa) ni atmosféricas (niebla, nubes persistentes), lo que permite observar con claridad la interacción geométrica entre satélites LEO y GNSS.
- Relevancia GNSS: presencia documentada de aplicaciones que dependan de los sistemas GNSS en la zona, como transporte, navegación aérea, estudios científicos, meteorología, logística o vigilancia ambiental.

Cabe destacar que la selección final no se limitó a un cumplimiento formal de los cinco criterios mencionados. Además, se buscaron ubicaciones que aportaran contraste entre sí y permitieran analizar situaciones extremas o especialmente sensibles al fenómeno estudiado.

#### 3.4.1.1. Construcción del conjunto preliminar de ubicaciones

Se elaboró una lista preliminar mediante un proceso exploratorio guiado por literatura especializada y documentación técnica en áreas como observación GNSS, propagación satelital y análisis geométrico desde tierra. Las ciudades incluidas fueron clasificadas según su fuente de referencia principal:

- **Uso previo en estudios de propagación, visibilidad o simulación GNSS–LEO:**
  - Tromsø y Kiruna han sido utilizadas por la ESA y EUREF como nodos árticos para campañas de observación ionosférica, auroral y de posicionamiento satelital, debido a su ubicación en latitudes elevadas y su proximidad al óvalo auroral, lo que las convierte en puntos estratégicos para la investigación de fenómenos de propagación en condiciones extremas [15, 16].

- Yakutsk figura en estudios rusos sobre efectos de escintilación y climatología espacial en regiones polares, en los que se ha documentado su papel como emplazamiento representativo del comportamiento del plasma ionosférico en la zona subauroral [17].
- Quito aparece recurrentemente en análisis ecuatoriales de propagación GNSS y variabilidad ionosférica, dado su emplazamiento casi exacto sobre la línea ecuatorial magnética y su relevancia en el estudio de irregularidades tipo burbuja y bandas ecuatoriales [18].

- **Presencia en redes de monitoreo global o agencias espaciales nacionales:**

- Alice Springs forma parte de la red de geodesia espacial australiana operada por Geoscience Australia y CSIRO, siendo un nodo relevante para el seguimiento de satélites, calibración de sensores GNSS y observación geodinámica del hemisferio sur [19].
- La Quiaca se encuentra registrada como estación activa en el sistema de referencia geodésico sudamericano SIRGAS, y ha sido utilizada como punto de control en estudios de deformación cortical y posicionamiento de alta precisión en la región andina [20].
- Libreville y Nairobi han sido incluidas en trabajos sobre contenido total de electrones (TEC) en regiones ecuatoriales, aportando datos relevantes para la modelización de variaciones diurnas y estacionales en zonas de alta actividad ionosférica [21, 22].

- **Condiciones geográficas especiales:**

- Marfa es una zona de baja densidad urbana y cielo despejado, utilizada ocasionalmente en estudios atmosféricos y ópticos, lo que la convierte en un entorno interesante para simulaciones visuales en condiciones de mínima contaminación lumínica y alta visibilidad celeste [23].
- Murmansk ha sido citada en literatura técnica como centro relevante en el desarrollo y despliegue de sistemas de guerra electrónica, incluyendo dispositivos capaces de interferir señales GNSS, lo cual plantea escenarios extremos útiles para evaluar la robustez de algoritmos de mitigación frente a interferencias intencionadas o entornos hostiles.

Estas diez ciudades fueron evaluadas conforme a los cinco criterios técnicos establecidos, tal como se muestra en la Tabla 3.2. Dicha tabla presenta un análisis comparativo detallado de cada ubicación, ordenadas alfabéticamente y valoradas en función de su diversidad latitudinal, tipo de entorno, cobertura orbital relevante, condiciones visuales y relevancia GNSS.

Ciudad	Diversidad latitudinal	Tipo de entorno	Cobertura orbital relevante	Condiciones visuales	Relevancia GNSS	¿Elegida?
Alice Springs (Australia)	Media Sur	Rural	Media	Excelentes	Moderada (comparativa)	Sí
Kiruna (Suecia)	Alta	Urbano	Alta	Buenas	Media	No
La Quiaca (Argentina)	Media Sur	Rural	Media	Variables	Baja	No
Libreville (Gabón)	Ecuatorial	Urbano	Media	Buena	Media	No
Marfa (EE. UU.)	Media Norte	Rural	Media	Buenas	Baja	No
Murmansk (Rusia)	Alta	Urbano	Alta	Variables	Limitada	No
Nairobi (Kenia)	Ecuatorial	Urbano	Alta	Buena	Media	No
Quito (Ecuador)	Ecuatorial	Urbano	Muy alta	Moderadas	Alta (urbana)	Sí
Tromsø (Noruega)	Alta	Urbano	Alta	Buenas	Alta (científica)	Sí
Yakutsk (Siberia)	Alta	Urbano	Alta	Variables	Limitada	No

Tabla 3.2: Evaluación comparativa de ubicaciones candidatas según criterios técnicos de selección.

### 3.4.1.2. Ubicaciones seleccionadas

Tras la evaluación realizada, se seleccionaron tres puntos de observación que cumplen de forma complementaria los cinco criterios técnicos definidos: ubicaciones que, por su diversidad geográfica y características particulares, permiten abordar el análisis desde perspectivas contrastadas y representativas.

- **Tromsø, Noruega** ( $69,6^\circ$  N): ciudad costera ubicada por encima del círculo polar ártico, reconocida por su infraestructura científica en geofísica y meteorología. Alberga el *Tromsø Geophysical Observatory* y estaciones GNSS y SLR de alta precisión [24]. Su latitud extrema la expone a órbitas polares e inclinadas, lo que la convierte en un punto estratégico para observar acumulación de trayectorias LEO y posibles bloqueos visuales.
- **Quito, Ecuador** ( $0,2^\circ$  S): capital ecuatoriana situada sobre el ecuador terrestre, a 2850 metros de altitud. Su posición permite una visibilidad simétrica de órbitas inclinadas, con cruces frecuentes de satélites LEO. Estudios previos sobre propagación, clima espacial y visibilidad GNSS en entornos urbanos ecuatoriales destacan su valor como punto de análisis [25].
- **Alice Springs, Australia** ( $23,7^\circ$  S): ciudad australiana aislada en el desierto, con cielos despejados y condiciones atmosféricas estables. Su baja densidad estructural y la presencia de estaciones de seguimiento como parte de la red geodésica nacional [26] la hacen ideal para analizar la interferencia LEO en un entorno rural controlado.

La siguiente tabla sintetiza los atributos clave de estas tres ubicaciones seleccionadas:

Criterio técnico	Tromsø (Noruega)	Quito (Ecuador)	Alice Springs (Australia)
Diversidad latitudinal	Alta latitud (69.6° N)	Ecuatorial (0.2° S)	Latitud media sur (23.7° S)
Tipo de entorno	Urbano costero ártico	Urbano montañoso	Rural interior continental
Cobertura orbital relevante	Paso frecuente de órbitas polares/inclinadas	Cruce constante de órbitas inclinadas/equatoriales	Cobertura intermedia desde múltiples planos LEO
Condiciones visuales	Baja obstrucción natural, cielo despejado	Elevación variable por entorno andino	Alta visibilidad, mínima infraestructura urbana
Relevancia GNSS	Aplicaciones científicas, navegación polar, clima extremo	Dependencia urbana de GNSS, logística y transporte	Escenario de referencia sin ruido urbano, buena comparación base

Tabla 3.3: Criterios técnicos de selección de ubicaciones finales para análisis GNSS–LEO.

### 3.4.1.3. Razones para el descarte de otras ubicaciones

Tras seleccionar las tres ubicaciones principales, se descartaron otras candidatas que, si bien cumplían parcialmente con los criterios técnicos (ver Tabla 3.2), presentaban limitaciones logísticas, menor relevancia científica o escasa viabilidad para el modelo propuesto. A continuación, se resumen los motivos de exclusión:

- **Kiruna (Suecia):** comparte características clave con Tromsø, como su localización en latitudes árticas y la frecuente cobertura de órbitas LEO de alta inclinación. Sin embargo, a pesar de albergar instalaciones científicas relevantes (como el Instituto Sueco de Física Espacial), Kiruna presenta menor integración en redes de observación geodésica globales (como IGS o EUREF) y un acceso más restringido a datos GNSS en tiempo real.
- **Marfa (EE. UU.):** pequeña localidad rural del suroeste de Texas, conocida por su baja densidad poblacional y cielos despejados, que la convierten en una buena candidata desde el punto de vista observational. No obstante, no cuenta con tradición en estudios satelitales ni con infraestructura GNSS permanente.
- **La Quiaca (Argentina):** situada en el altiplano andino a más de 3400 metros sobre el nivel del mar, ofrece condiciones atmosféricas extremas y una perspectiva interesante para analizar propagación en altura. Sin embargo, su inclusión se vio limitada por la inestabilidad meteorológica, la menor densidad de receptores GNSS disponibles públicamente en la zona, y la falta de documentación técnica detallada. Estas restricciones comprometían

su viabilidad como caso de simulación estable, especialmente para un estudio centrado en disponibilidad geométrica.

- **Murmansk y Yakutsk (Rusia):** ambas ciudades se encuentran en latitudes muy elevadas del hemisferio norte, lo que les confiere una cobertura orbital favorable para el estudio de trayectorias LEO. No obstante, fueron descartadas por presentar acceso restringido a bases de datos internacionales, escasa interoperabilidad con herramientas occidentales como Skyfield o Celestrak, y una limitada producción científica reciente publicada en repositorios accesibles. Además, la inestabilidad política e institucional de la región añade incertidumbre sobre la fiabilidad de los datos.
- **Libreville (Gabón) y Nairobi (Kenia):** ambas capitales africanas fueron consideradas por su valor estratégico al estar situadas en la franja ecuatorial. Sin embargo, factores como el bajo número de estaciones GNSS de acceso abierto, la falta de continuidad en campañas de observación satelital y la limitada documentación en literatura técnica redujeron su idoneidad. En comparación, Quito ofrecía no solo una mejor altitud y entorno montañoso favorable para análisis geométricos, sino también una trayectoria consolidada en estudios GNSS y mayor soporte técnico-institucional.

En resumen, Tromsø, Quito y Alice Springs fueron seleccionadas no solo por cumplir los cinco criterios técnicos definidos (latitud, entorno, cobertura orbital, visibilidad y relevancia GNSS), sino también por representar contextos donde el impacto visual y geométrico de constelaciones LEO sobre sistemas GNSS puede observarse con mayor claridad, comparabilidad y validez práctica.

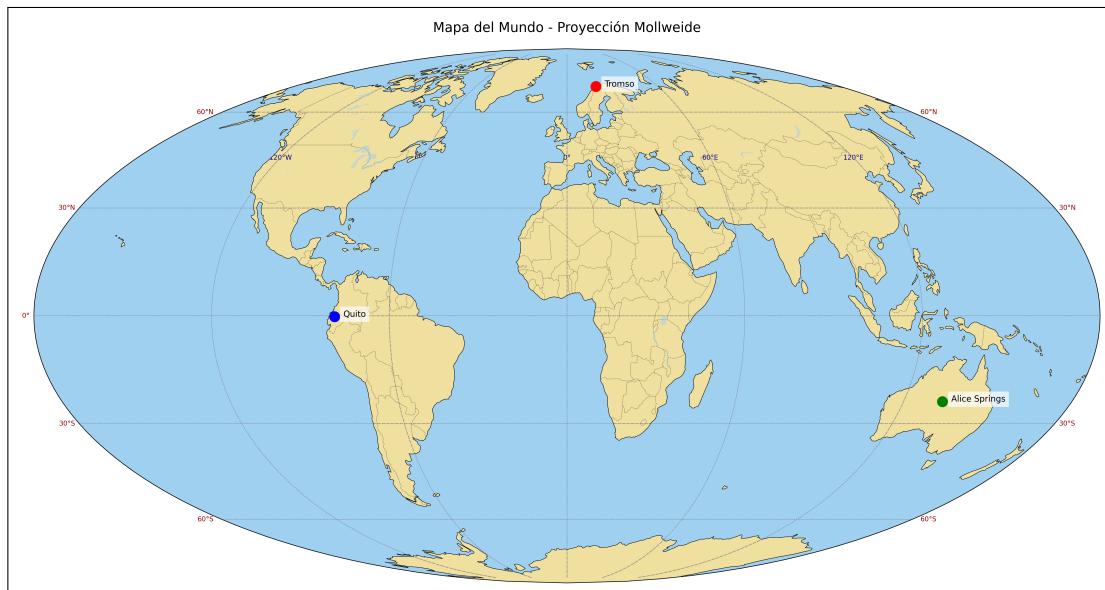


Figura 3.2: Ubicación de las ciudades seleccionadas en proyección Mollweide.

### 3.4.2. Obtención y preprocesamiento de TLEs

Los datos orbitales utilizados para simular la constelación Starlink fueron obtenidos del repositorio público [Celestrak.com](https://celestak.com) [27], concretamente del grupo “Starlink”. Este repositorio proporciona TLEs (*Two-Line Elements*) actualizados diariamente, derivados del seguimiento realizado por NORAD y compilados para uso civil y científico. Los TLEs constituyen una descripción compacta de los parámetros orbitales, compatibles con propagadores estándar como SGP4 y con herramientas como [Skyfield](https://skyfield.readthedocs.io).

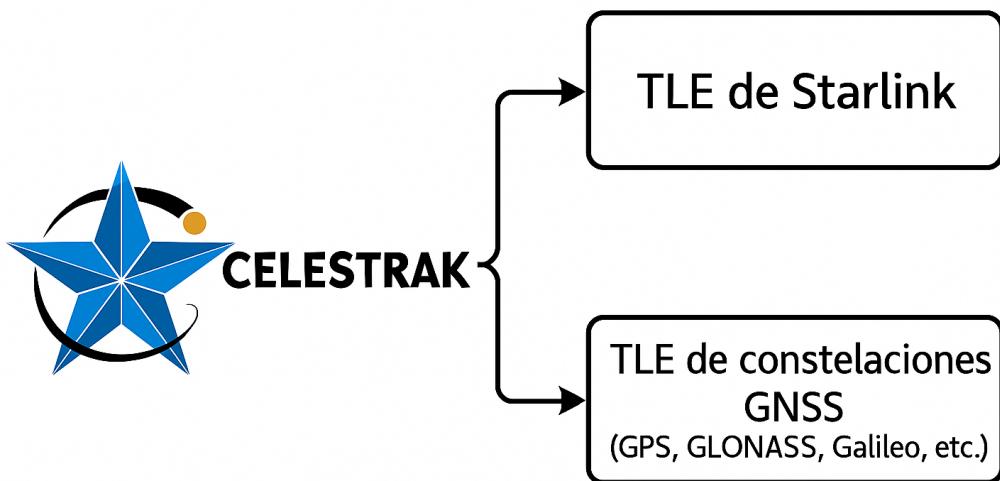


Figura 3.3: Celestrak: punto de partida para la descarga de TLEs.

Con el fin de analizar cómo la congestión satelital afecta a la geometría de visibilidad GNSS, se definieron tres configuraciones representativas de densidad orbital, construidas a partir de subconjuntos reales de satélites Starlink:

- **Baja densidad:** subconjunto reducido de satélites centrado en la capa principal de Starlink (altitud  $\sim 550$  km, inclinación  $\sim 53^\circ$ ). Se seleccionan únicamente aquellos satélites con parámetros dentro de los rangos  $[530, 570]$  km y  $[52^\circ, 54^\circ]$ . Esta configuración simula un despliegue parcial o inicial, útil como caso base de comparación.
- **Densidad media:** conjunto intermedio de aproximadamente 2000 satélites distribuidos en múltiples capas orbitales, priorizando aquellos con TLEs recientes (últimos 7 días) y

órbitas propagables. Representa una constelación parcialmente desplegada y operativa a gran escala, pero no saturada.

- **Densidad alta:** constelación completa disponible al momento de la simulación, con más de 5000 satélites distribuidos en altitudes entre 340 y 570 km, e inclinaciones de 53°, 70° y 97.6°. Esta configuración emula una megaconstelación madura en operación continua.

Para garantizar una comparación rigurosa entre escenarios, todos los TLE fueron descargados el mismo día y propagados a una fecha común de simulación. Se eliminaron elementos defectuosos mediante validación numérica directa: cada TLE fue convertido en un objeto `EarthSatellite` y propagado al instante inicial. Los que arrojaron errores durante la propagación fueron descartados automáticamente.

### 3.4.3. Simulación orbital GNSS y LEO

Para modelar con precisión la dinámica de las constelaciones GNSS y LEO, se desarrolló un simulador personalizado en `Python`, utilizando la librería `Skyfield` [28]. El objetivo de esta etapa es obtener las efemérides orbitales de los satélites visibles desde varios puntos fijos durante un intervalo continuo de 24 horas. El proceso incluye:

- **Simulación de efemérides:** Se calcula la posición de cada satélite con una resolución temporal de 5 minutos. A cada instante, Skyfield transforma las posiciones desde coordenadas ECEF a topocéntricas (azimut y elevación) relativas a cada estación receptora.
- **Filtrado de visibilidad:** Se considera visible un satélite GNSS si su elevación supera los 10°, umbral convencional que garantiza observaciones confiables al evitar errores por refracción atmosférica [14]. Este criterio permite centrarse en configuraciones geométricas útiles para la navegación terrestre.
- **Registro de datos para análisis posterior:** Se almacenan las posiciones topocéntricas y visibilidades horarias de cada satélite GNSS y LEO visible desde cada receptor. Estos datos permiten generar visualizaciones como skyplots y tablas de cobertura, además de servir como base para la detección de coincidencias GNSS–LEO.

Esta etapa proporciona la base para detectar posibles eventos de bloqueo visual y calcular métricas geométricas relevantes, como el PDOP.

### 3.4.4. Detección de coincidencias angulares GNSS–LEO

A partir de los datos de visibilidad obtenidos en la simulación, se implementa un algoritmo que detecta bloqueos visuales por coincidencia angular entre satélites GNSS y LEO. Este análisis se

basa en la geometría relativa entre el observador y los satélites visibles en cada instante.

- **Cálculo del ángulo de separación:** Para cada par GNSS–LEO visible desde una estación receptora, se calcula el ángulo de separación  $\theta$  entre sus vectores unitarios topocéntricos:

$$\theta = \arccos(\hat{\mathbf{r}}_{\text{GNSS}} \cdot \hat{\mathbf{r}}_{\text{LEO}})$$

donde  $\hat{\mathbf{r}}$  representa la dirección angular normalizada desde el receptor hacia cada satélite.

- **Criterio de coincidencia visual:** Se considera que un satélite GNSS está bloqueado por un LEO si  $\theta < 1^\circ$ . Este umbral se toma de la propuesta metodológica de Duetsch [11], quien evidenció que este nivel de coincidencia angular representa una alineación suficientemente precisa como para interferir con la recepción visual de la señal sin necesidad de considerar obstrucción física directa o interferencia electromagnética.
- **Optimización computacional:** Dada la gran cantidad de combinaciones posibles entre satélites GNSS y LEO, se implementaron:
  - Estructuras `KDTree` para búsquedas rápidas de vecinos más cercanos en el espacio angular.
  - Cálculo vectorizado con `NumPy` para acelerar el análisis en cada paso de tiempo.

Aunque no se incorpora paralelización explícita, estas técnicas permiten una ejecución eficiente en escalas grandes de satélites.

- **Almacenamiento de eventos de bloqueo:** Se registran matrices binarias de estado (bloqueado/no bloqueado) para cada satélite GNSS y cada instante de simulación. Estos datos sirven de base para calcular posteriormente métricas como el porcentaje de tiempo bloqueado por satélite, el índice global de bloqueo o el impacto sobre la geometría PDOP.
- **Generación de visualizaciones:** A partir de estos eventos, se podrán generar representaciones como skyplots, mapas de cobertura por hora local o histogramas de coincidencias por ubicación, que facilitarán el análisis comparativo entre escenarios.

El proceso desde la obtención de TLEs hasta la detección de coincidencias, queda representado en el diagrama de flujo de la Figura 3.4:

### Detección de bloqueos GNSS-LEO por coincidencia angular

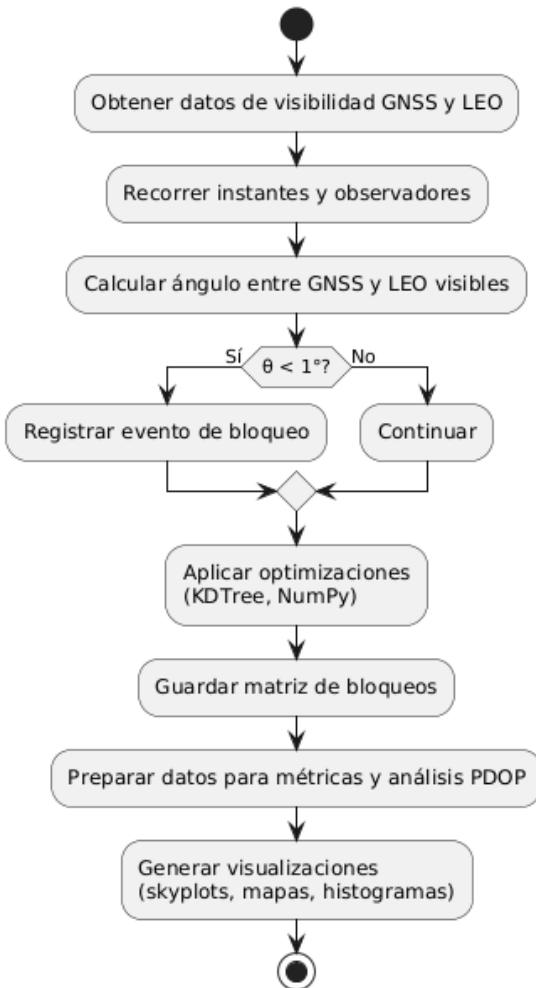


Figura 3.4: Diagrama de flujo del proceso de simulación orbital y detección de bloqueos.

#### 3.4.5. Cálculo de métricas estadísticas

A partir del registro horario de eventos de coincidencia angular entre satélites GNSS y LEO, se definen y calculan una serie de métricas cuantitativas destinadas a evaluar el impacto geométrico de las constelaciones LEO sobre la disponibilidad efectiva de señales GNSS. Estas métricas permiten una interpretación sistemática del fenómeno de bloqueo visual en distintos contextos.

Las métricas se clasifican en tres categorías fundamentales:

- **Probabilidad de Bloqueo Individual (PBI):**

Estima el porcentaje del tiempo simulado en que un satélite GNSS específico fue bloqueado visualmente por algún satélite LEO desde un observador determinado. Esta métrica está

formalizada en la Ecuación (3.8), y permite identificar qué satélites resultan más vulnerables a interferencias recurrentes según su trayectoria relativa frente al plano LEO.

- **Índice Global de Bloqueo Visual (BI):**

Ofrece una visión instantánea del grado de obstrucción de la constelación GNSS. Su forma instantánea se define en la Ecuación (3.9), mientras que su promedio temporal se recoge en la Ecuación (3.10). Esta métrica resume el impacto acumulado de los eventos de obstrucción angular sobre la disponibilidad GNSS en un intervalo dado.

- **Frecuencia Horaria de Eventos de Bloqueo:**

Cuantifica la intensidad temporal de los eventos de coincidencia angular GNSS–LEO. Se define como:

$$\text{Frecuencia}_{\text{bloqueo}} = \frac{N_{\text{bloqueos totales}}}{T_{\text{simulación (horas)}}} \quad (3.11)$$

donde:

- $N_{\text{bloqueos totales}}$  es el número total de eventos de bloqueo visual registrados.
- $T_{\text{simulación (horas)}}$  es la duración de la simulación expresada en horas.

Este parámetro facilita la comparación entre ubicaciones y configuraciones orbitales según la intensidad de coincidencias GNSS–LEO registradas por hora.

#### 3.4.6. Estimación del impacto geométrico

Durante la simulación diaria de cada escenario, se identifican tres instantes representativos por ubicación: el de mayor número de bloqueos GNSS, el de menor, y uno de valor medio. Para cada uno de estos momentos, se evalúa el impacto geométrico mediante el cálculo del PDOP utilizando dos configuraciones de visibilidad:

- **PDOP completo:** incluye todos los satélites GNSS visibles ( $\text{elevación} > 10^\circ$ ), sin filtrar.
- **PDOP filtrado:** excluye aquellos satélites GNSS con coincidencias angulares menores a  $1^\circ$  respecto a algún LEO visible, considerados potencialmente bloqueados.

El cálculo del PDOP se basa en la matriz de diseño generada a partir de las posiciones de los satélites GNSS en cada instante, utilizando funciones propias en Python integradas en el flujo general de simulación. La diferencia entre ambas configuraciones (PDOP completo vs. filtrado) permite cuantificar el deterioro geométrico ( $\Delta\text{PDOP}$ ) inducido por las coincidencias GNSS–LEO.

Esta métrica se obtiene para cada ubicación (Tromsø, Quito y Alice Springs) y para los tres niveles de densidad orbital definidos. Los resultados se analizan de forma comparativa y se representan mediante gráficos y tablas asociadas a cada perfil temporal.

### 3.4.7. Visualización y análisis comparativo

La representación visual de los resultados constituye una parte fundamental del análisis, al facilitar la interpretación del efecto geométrico y temporal de los bloqueos visuales inducidos por la presencia de satélites LEO. El sistema implementado genera una serie de gráficos y métricas que permiten evaluar, de forma integrada, tanto el impacto individual sobre cada observador como las tendencias globales según la densidad de la constelación Starlink:

- **Skyplots dinámicos:** Gráficos polares que muestran la distribución angular de los satélites visibles desde cada punto de observación. Incluyen:
  - Satélites GNSS por constelación, diferenciados por color.
  - Satélites GNSS bloqueados en un instante dado.
  - Satélites LEO visibles.
- **Series temporales de métricas clave:** Gráficos de líneas que representan la evolución por hora de:
  - El porcentaje de observaciones GNSS bloqueadas (**BI**, *Blockage Index*, Índice de bloqueo visual en español).
  - El PDOP completo y el PDOP filtrado (que excluye satélites bloqueados), ambos calculados a lo largo del día.
  - La variación porcentual  $\Delta$ PDOP, representada en gráficos secundarios para cuantificar el impacto de los bloqueos.
- **Heatmaps horarios de PDOP:** Representaciones de calor que permiten comparar visualmente la variación del PDOP (completo y filtrado) a lo largo del día. Estas gráficas son especialmente útiles para detectar patrones recurrentes de degradación geométrica.
- **Comparativas multi-densidad (Starlink):** Para evaluar el efecto de la expansión de la constelación LEO sobre el sistema GNSS, se implementaron gráficos agregados que comparan distintas densidades:
  - *Boxplots* de PDOP completo y filtrado, para `low`, `medium` y `full`.
  - Gráfico de dispersión del  $\Delta$ PDOP promedio en función del número de satélites LEO.
  - Evolución del índice de bloqueo global (BI) según la densidad.
- **Tablas resumen cuantitativas:** Todas las métricas extraídas se almacenan como estructuras pandas `DataFrame`, exportables para análisis posterior. Incluyen:
  - Índice global de bloqueo (BI) por ciudad y densidad.

- Valores de PDOP completo, filtrado y variación porcentual media.
- Satélite GNSS más afectado (por tasa de bloqueo individual).
- Estadísticas por constelación GNSS (GPS, GLONASS, etc.).

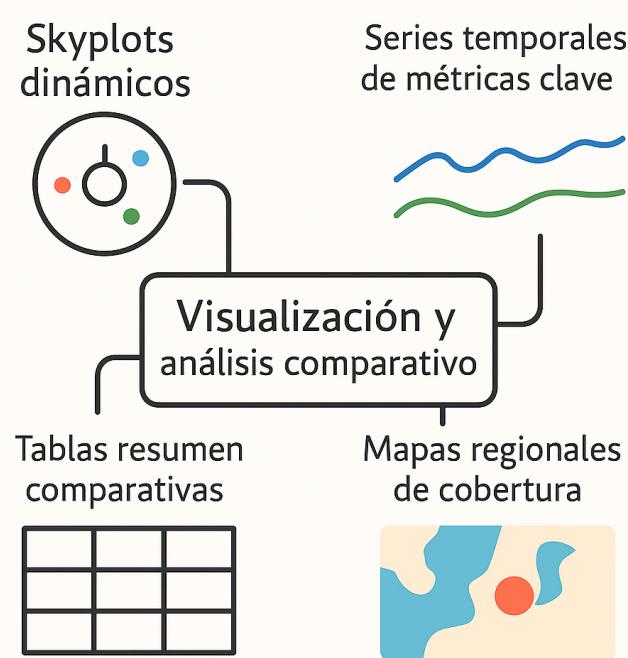


Figura 3.5: Representación de las visualizaciones a generar.

### 3.5. Entorno de desarrollo y herramientas

El proyecto se implementó en un entorno de desarrollo reproducible, basado en herramientas estándar para análisis orbital, programación científica y simulación de eventos visuales GNSS-LEO. A continuación se describen los componentes principales del entorno técnico utilizado:

■ **Entorno de programación:**

- Visual Studio Code como editor principal, con extensiones de soporte para Python, y control de entornos virtuales.
- Gestión de dependencias mediante entorno virtual creado con pipenv.

■ **Lenguaje y librerías clave:**

- Python 3.11 como lenguaje base, por su versatilidad, soporte científico y compatibilidad con librerías especializadas.

- Librerías fundamentales:

- `Skyfield` + archivo `de421.bsp`: para propagación orbital de TLEs mediante modelo SGP4 y cálculo de coordenadas topocéntricas (azimut y elevación).
- `NumPy` y `SciPy`: para operaciones vectorizadas y uso de `KDTree` en la detección angular de bloqueos visuales (umbral 1°).
- `Pandas`: para estructuración y manipulación eficiente de datos tabulares.
- `Matplotlib` y `Seaborn`: para generación de gráficos, skyplots polares y visualización de series métricas temporales.

- Fuentes de datos utilizadas:

- TLEs actualizados desde el portal `Celestrak.com`, descargados de forma automática mediante `urllib.request`.
- Archivos utilizados:
  - `gnss.txt`: contiene TLEs de constelaciones GNSS (GPS, Galileo, BeiDou, GLONASS).
- Efemérides planetarias `de421.bsp` del JPL (*Jet Propulsion Laboratory*), necesarias para computar referencias astronómicas precisas desde la Tierra.

- Estructura del código y organización del simulador:

- El núcleo del sistema está implementado como una clase central `GNSSLEOSimulator`, que organiza las siguientes tareas:
  - Descarga y filtrado de TLEs por altitud e inclinación.
  - Simulación de visibilidad de satélites GNSS y detección de bloqueos por coincidencia angular con LEO.
  - Generación de métricas visuales como el porcentaje de bloqueo (*Blocking Index*) y cálculo del PDOP modificado.
- Funciones auxiliares especializadas:
  - `calculate_pdop()`: cálculo del Dilution of Precision usando vectores unitarios de satélites GNSS visibles no bloqueados.
  - `plot_skyplot()`, `plot_blocking_timeline()`: visualización de trayectorias y métricas por ubicación y momento.

Este capítulo ha definido una metodología clara y reproducible para simular el impacto geométrico de constelaciones LEO sobre señales GNSS, estructurada en torno a ubicaciones seleccionadas, métricas específicas (BI, PDOP) y distintos niveles de densidad orbital. En el siguiente capítulo se aplicará el modelo a través de simulaciones, detallando el proceso.

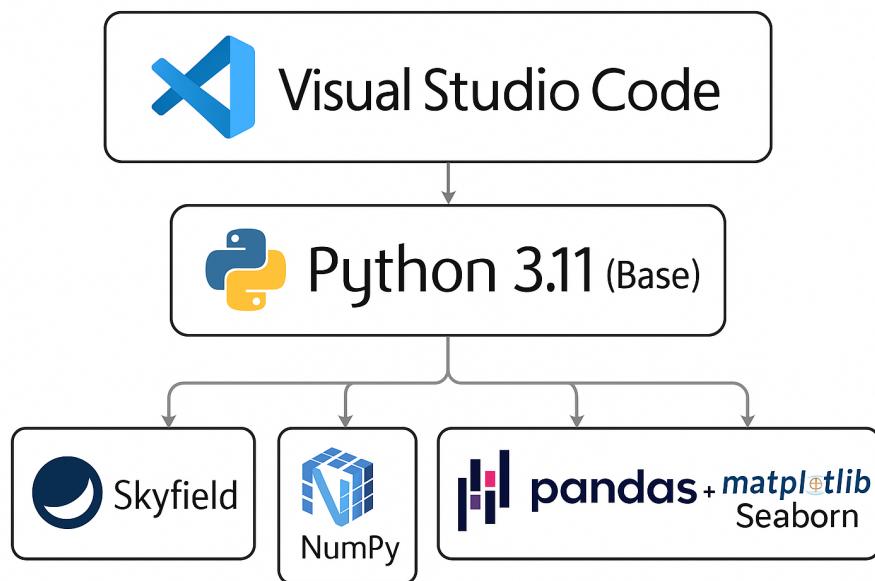


Figura 3.6: Representación del entorno, lenguaje y librerías utilizadas.

## Capítulo 4

# Implementación de la simulación GNSS–LEO

En este capítulo se recorre el proceso completo que permite construir y ejecutar el modelo de simulación, desde la elección del entorno de desarrollo y las librerías clave, hasta la descarga y filtrado de datos orbitales. También se detalla la lógica implementada para representar de forma realista el bloqueo visual entre satélites GNSS y constelaciones LEO.

### 4.1. Skyfield

Skyfield es una librería de código abierto escrita en `Python`, diseñada para calcular posiciones astronómicas y efemérides de alta precisión de cuerpos celestes y satélites artificiales [28]. Fue desarrollada por Brandon Rhodes como sucesora de `PyEphem`, mejorando su precisión numérica y simplificando su interfaz mediante una API moderna y legible.

Su principal ventaja en simulaciones satelitales es la capacidad de propagar elementos orbitales tipo TLE (Two-Line Elements) mediante el modelo estándar SGP4, permitiendo generar posiciones orbitales precisas en distintos sistemas de referencia (ICRF, ECEF, topocéntrico). Para ello, Skyfield opera sobre objetos como `EarthSatellite`, instanciados a partir de TLEs descargados desde plataformas como Celestrak, y proporciona cálculos relativos con respecto a observadores terrestres definidos mediante la clase `Topos`.

En este trabajo, Skyfield permite:

- Simular el movimiento orbital de satélites GNSS (GPS, Galileo, GLONASS, BeiDou) y de satélites LEO como Starlink con una resolución de hasta 5 minutos.
- Obtener coordenadas topocéntricas (azimut, elevación y distancia) mediante métodos sim-

plificados como `(sat - observer).at(t).altaz()`.

- Integrarse con NumPy para vectorizar operaciones orbitales sobre múltiples objetos, y con Pandas para estructurar los datos resultantes de forma tabular.

Es importante destacar que en la implementación actual del simulador no se aplican correcciones por refracción atmosférica ni aberración de la luz, aunque Skyfield permite hacerlo opcionalmente. Esta elección se tomó para mantener consistencia con los datos TLE sin alterar su significado geométrico puro.

Skyfield ha sido ampliamente adoptada en entornos académicos y de investigación. Por ejemplo, se ha utilizado en estudios de propagación orbital para GNSS reflectometría en la NTNU [29], y en trabajos publicados en conferencias del IEEE orientados al análisis de cobertura y coincidencias visuales en constelaciones LEO como Starlink [30, 31].

## 4.2. Celestrak

Celestrak es una plataforma de acceso abierto que proporciona conjuntos TLE (Two-Line Elements) para el seguimiento orbital de satélites artificiales. Fundada por el Dr. T. S. Kelso en 1985, ha sido durante décadas una fuente de referencia para la comunidad científica, organismos espaciales y usuarios del ámbito GNSS [27].

Los TLEs disponibles en Celestrak se actualizan al menos diariamente y se obtienen a partir de observaciones procesadas por el Comando Espacial de los Estados Unidos (USSPACECOM), siguiendo el formato estandarizado por NORAD. Estos elementos definen la órbita de un satélite en un instante de tiempo específico y son compatibles con el modelo de propagación SGP4 utilizado por múltiples herramientas, entre ellas Skyfield.

En el contexto de este proyecto, Celestrak constituye la fuente primaria de datos para todas las constelaciones simuladas:

- Se emplea el archivo consolidado `gnss.txt` para las constelaciones GNSS (GPS, Galileo, GLONASS, BeiDou).
- Para Starlink, se utiliza la URL dinámica `gp.php?GROUP=starlink`, que proporciona el subconjunto actualizado de satélites operativos en la constelación LEO.

El acceso a los TLEs es completamente programático mediante el uso de `urllib.request`, y los datos se integran directamente en objetos `EarthSatellite` sin necesidad de pasos intermedios. Para garantizar coherencia geométrica con los satélites LEO activos, el simulador aplica un filtrado adicional en función de parámetros extraídos del propio TLE:

- Altitud media (estimada por el movimiento medio): entre 400 y 600 km.
- Inclinación orbital: entre 50° y 60°.

Celestrak es citada frecuentemente en investigaciones académicas por su fiabilidad y cobertura. En estudios recientes de impacto visual y geométrico de satélites LEO sobre redes GNSS [30, 31], ha sido la fuente principal de datos para evaluar trayectorias, ángulos de occultación y métricas como la densidad angular o el PDOP.

### 4.3. Entorno de desarrollo y ejecución

El simulador se desarrolla íntegramente en Python 3.11, utilizando Visual Studio Code como entorno principal. Para garantizar la estabilidad del entorno y facilitar la portabilidad del proyecto, se implementa una configuración de entorno virtual con `venv`, que permite aislar las dependencias necesarias para la simulación GNSS-LEO.

A continuación se resumen los pasos para la configuración del entorno:

1. **Verificación de Python:** se comprueba la instalación ejecutando `python -version`. En caso necesario, se descarga desde la página oficial [32].
2. **Creación del entorno virtual:** `python -m venv gnss_env`
3. **Activación del entorno (Windows):** `gnss_env\Scripts\activate`
4. **Instalación de paquetes:** `pip install numpy pandas matplotlib seaborn skyfield scipy tqdm pytz cartopy`
5. **Verificación e instalación reproducible:** `pip list` para confirmar paquetes instalados y `pip freeze >requirements.txt` para generar el archivo de dependencias.

El código fuente se concentra en un único archivo principal:

- `main.py`: contiene todas las clases del simulador, análisis de métricas, visualizaciones y ejecución general.
- `gnss.txt`: archivo con los TLEs de las constelaciones GNSS utilizadas.
- `de421.bsp`: efemérides planetarias requeridas por Skyfield (descargadas automáticamente si no están presentes).

Este diseño modular y reproducible garantiza que el entorno pueda replicarse con facilidad en cualquier sistema compatible con Python 3.11, minimizando errores por conflictos de versión y facilitando el análisis posterior de los resultados simulados.

### 4.3.1. Librerías y complementos

A lo largo del desarrollo se incorporaron distintas librerías, algunas previstas desde el inicio y otras añadidas durante el proceso por su utilidad en tareas concretas. En la tabla siguiente se resumen las más relevantes y su uso dentro del proyecto:

Librería	Uso en el simulador
<code>skyfield</code>	Propagación de TLEs, cálculo de posiciones topocéntricas, altitud, azimut y efemérides.
<code>numpy</code>	Operaciones vectorizadas, normalización de vectores, álgebra lineal y matrices de geometría.
<code>pandas</code>	Gestión de observaciones, estructuración de datos, generación de tablas de resultados.
<code>matplotlib</code>	Visualización de resultados: gráficos de líneas, skyplots, comparativas y cronogramas.
<code>seaborn</code>	Heatmaps estilizados para representar la variación del PDOP por hora y dirección.
<code>scipy.spatial.KDTree</code>	Búsqueda rápida de vecinos en el espacio angular, usada en la detección de bloqueos GNSS–LEO.
<code>tqdm</code>	Barra de progreso para simular visibilidad y calcular PDOPs en escenarios largos.
<code>datetime</code> y <code>pytz</code>	Gestión de fechas UTC, pasos temporales, y sincronización de simulaciones.
<code>urllib.request</code>	Descarga directa de TLEs desde Celestrak.

Tabla 4.1: Principales librerías utilizadas en el desarrollo del simulador.

Durante el desarrollo se evitó la integración de entornos gráficos complejos o dependencias innecesarias. Todo se ejecuta desde consola, lo que simplifica la depuración y facilita su portabilidad.

Con esta estructura, el simulador puede ejecutarse desde un único script y reproducir completamente cada experimento definido en el apartado metodológico. La generación de datos, gráficos y análisis está automatizada, y la integración entre componentes permite realizar simulaciones completas en cuestión de minutos por observador y configuración.

## 4.4. Obtención y procesamiento de datos

Antes de ejecutar cualquier simulación, el sistema debe obtener, filtrar y organizar los datos orbitales de entrada. El proceso se encuentra completamente automatizado en el script principal y se divide en varias etapas bien definidas: descarga de TLEs, clasificación y filtrado.

### 4.4.1. Carga y clasificación de constelaciones GNSS

La función responsable de esta tarea es `load_gnss_constellations()`. Su objetivo es descargar el archivo `gnss.txt` desde Celestrak y organizar los satélites contenidos en él en cuatro listas separadas, correspondientes a los sistemas GPS, Galileo, GLONASS y BeiDou.

El archivo descargado contiene todos los GNSS mezclados. Por ello, el script recorre cada satélite mediante un bucle que analiza su nombre. Para evitar inconsistencias por formato, todos los nombres se transforman primero a mayúsculas. Luego se evalúan distintas condiciones:

- Si el nombre contiene `GPS` o `NAVSTAR`, se clasifica como parte del sistema GPS.
- Si contiene `GALILEO` o `GSAT`, se asocia al sistema Galileo.
- Si el nombre incluye `GLONASS`, `COSMOS`, o empieza por la letra `R` seguida de un número (ej. `R-750`), se asocia a GLONASS.
- Si contiene `BEIDOU`, `COMPASS`, o una cadena que comienza por `BD-`, se identifica como BeiDou.

Cada vez que un satélite es reconocido, se añade a una lista dentro del diccionario interno `self.gnss_sats`, cuya clave es el nombre de la constelación. Si el satélite no puede clasificarse de forma fiable, se omite.

Al finalizar, el sistema imprime un resumen con el número de satélites identificados por cada constelación, permitiendo verificar la carga completa de emisores antes de proceder a la simulación.

### 4.4.2. Descarga y filtrado de satélites LEO (Starlink)

La función correspondiente a esta fase es `load_starlink_satellites(density)`, que se encarga de obtener todos los TLE disponibles del grupo `starlink` desde Celestrak. El archivo resultante se almacena en memoria como una lista de líneas, que el sistema recorre en bloques de tres: una línea de nombre y dos líneas TLE.

Para cada bloque, se intenta construir un objeto `EarthSatellite`. Sin embargo, antes de almacenar definitivamente el satélite, se realiza una verificación basada en sus parámetros orbitales: altitud e inclinación. Ambos se extraen o estiman a partir del contenido textual del TLE.

```
STARLINK-1008
1 44714U 19074B 25176.41164397 .00000009 00000+0 19465-4 0 9994
2 44714 53.0541 261.5847 0001346 94.4534 265.6609 15.06402192309978
STARLINK-1010
1 44716U 19074D 25176.53731297 .00064978 00000+0 11264-2 0 9990
2 44716 53.0540 246.7641 0007593 312.4160 47.6205 15.50276106310340
```

Figura 4.1: Ejemplo real de TLE para la constelación Starlink.

La inclinación orbital se extrae directamente de la segunda línea del TLE, en el campo correspondiente a las posiciones 9–16 del formato estándar. Este valor, en grados, representa el ángulo entre el plano de la órbita y el plano ecuatorial terrestre. Se trata de un valor declarado, que no requiere cálculos adicionales, y es especialmente útil para filtrar satélites en capas específicas (por ejemplo, las de 53°, 70° o 97.6° utilizadas por Starlink).

La altitud orbital, por el contrario, no aparece explícitamente en el TLE. Por ello, el sistema la estima a partir del parámetro de `mean motion` (revoluciones por día), que se encuentra en las posiciones 53–63 de la segunda línea [27]. A partir de ese valor, se calcula el semieje mayor de la órbita utilizando una forma invertida de la tercera ley de Kepler, adaptada a unidades estándar:

$$a = \left( \frac{\mu}{n^2} \right)^{1/3} \quad (4.1)$$

donde  $\mu = 398600,4418 \text{ km}^3/\text{s}^2$  es el parámetro gravitacional estándar de la Tierra, y  $n$  representa el movimiento medio convertido a radianes por segundo.

El procedimiento seguido en el simulador consiste en extraer el valor de `mean motion` ( $n_{\text{rev}}$ ) desde el TLE, expresado en revoluciones por día. Este valor se transforma a radianes por segundo mediante la expresión:

$$n = n_{\text{rev}} \cdot \frac{2\pi}{86400} \quad (4.2)$$

Con el valor de  $n$ , se calcula el semieje mayor orbital  $a$  aplicando la fórmula anterior. Finalmente, se resta el radio medio terrestre ( $R_{\oplus} = 6371 \text{ km}$ ) para obtener la altitud orbital del objeto.

En la implementación definitiva del simulador, todos estos pasos se combinan en una única expresión algebraica compacta:

$$\text{altitud} = \frac{\mu^{1/3}}{\left(\frac{2\pi \cdot n_{\text{rev}}}{86400}\right)^{2/3}} - 6371 \quad (4.3)$$

Esta fórmula permite estimar con precisión suficiente la altitud de satélites en órbitas casi circulares, como los de la constelación Starlink, con un error típico inferior a 1 km. Dado que la mayoría de los objetos considerados en este trabajo presentan baja excentricidad y se mantienen en altitudes LEO estables, este método resulta adecuado para el filtrado orbital requerido por el sistema.

El sistema aplica un filtrado físico que excluye aquellos objetos cuya altitud o inclinación no se ajustan al rango definido para simulaciones LEO-GNSS. Inicialmente se valoró trabajar con rangos muy estrechos. Sin embargo, esta decisión implicaba descartar una parte sustancial de la constelación activa, incluyendo capas operativas con inclinaciones de 43°, 70° o 97.6°.

Por este motivo, se adoptó finalmente un criterio más amplio: altitudes entre 400 y 650 km e inclinaciones entre 40° y 100°. Este rango abarca todas las capas activas de Starlink sin incluir objetos externos, y garantiza mayor realismo y robustez ante futuros cambios en la arquitectura orbital de la red.

Los satélites válidos se almacenan en una lista temporal. Si ocurre algún error al procesar un bloque (por ejemplo, un TLE inválido), el sistema lo ignora silenciosamente y continúa con el resto.

#### 4.4.3. Selección por niveles de densidad

Una vez que el sistema finaliza el filtrado físico de satélites Starlink, la lista resultante puede incluir entre 4000 y 5500 objetos, dependiendo del momento de descarga de los TLEs. Para permitir diferentes configuraciones de carga computacional y precisión geométrica, se implementó un mecanismo de selección por niveles de densidad, definido a través del parámetro de entrada `density`.

Este parámetro se puede configurar con tres valores posibles: "low", "medium" y "full". Su interpretación condiciona el tamaño máximo del subconjunto de satélites que se utilizará en las simulaciones, aplicando una segmentación basada únicamente en la posición de cada objeto en la lista.

- **Densidad baja ("low"):** el sistema selecciona únicamente los primeros 500 satélites tras el filtrado. Estos suelen pertenecer, por orden de aparición en los TLEs de Celestrak, a la capa orbital más numerosa de la constelación (550 km, 53°).
- **Densidad media ("medium"):** se retienen los primeros 2000 satélites válidos. A diferencia

del modo anterior, esta modalidad sí abarca una variedad más amplia de órbitas dentro del espectro LEO de Starlink.

- **Densidad completa ("full"):** no se aplica ningún recorte, y se utiliza la lista completa de satélites que pasaron el filtrado físico.

En la implementación, tras completar el filtrado físico, los satélites válidos se almacenan en una lista intermedia. A partir de ella, y según el valor del parámetro `density`, se aplica una selección determinista por truncado. Para los modos "`low`" y "`medium`", se conservan únicamente los primeros  $N$  elementos de la lista —500 o 2000, respectivamente— mediante una operación directa por corte (`satellites[:N]`). En el caso del modo "`full`", no se realiza ninguna restricción adicional, y se conserva la lista completa.

El conjunto de procedimientos presentados garantiza que, al comenzar la simulación, el sistema disponga de un conjunto limpio, representativo y clasificable de satélites GNSS y LEO. Todas las funciones implicadas están descritas en el apéndice al final del documento, junto con sus respectivas implementaciones y clases detalladas.

## 4.5. Aplicación de observadores y eventos de bloqueo

El simulador permite analizar de forma detallada la visibilidad y los bloqueos visuales entre satélites GNSS y objetos LEO desde los tres observadores fijos seleccionados con anterioridad: Tromsø (Noruega), Quito (Ecuador) y Alice Springs (Australia). Cada uno de ellos se representa mediante coordenadas absolutas, y se añade al sistema mediante la función `add_observer(name, lat, lon, elev=0)`, que crea una instancia de `Topos` (de `Skyfield`) con los valores proporcionados y la almacena en el diccionario interno `self.observers` usando como clave el nombre del observador.

### 4.5.1. Simulación temporal y estructura de datos

El cálculo de visibilidad y eventos se ejecuta con el método `simulate_visibility(observer_name, date, duration_hours=24, time_step_min=5)`, que realiza una simulación durante un período de 24 horas, con intervalos de 5 minutos, a partir de una fecha concreta, en este caso el 21 de Junio de 2025, el último solsticio. Los instantes temporales se almacenan en una lista `times`, que se genera como un conjunto de objetos `Time` de `Skyfield`.

Los resultados de toda la simulación se almacenan en `self.results[observer_name]`, un diccionario con la siguiente estructura:

- `times`: lista de objetos de tiempo simulados.
- `gnss_visible`: diccionario por constelación con todos los GNSS visibles, su elevación, azimut y estado de bloqueo.
- `leo_positions`: lista con las posiciones aparentes de todos los LEO por instante.
- `blocked_gnss`: diccionario por constelación con los nombres de los GNSS bloqueados en cada instante.

#### 4.5.2. Detección de visibilidad GNSS

Para cada instante temporal  $t$ , se recorren todos los satélites GNSS almacenados en `self.gnss_sats`, organizados por constelación. A través del operador `(sat - observer).at(t)`, se calcula su posición relativa respecto al observador. Luego se extraen:

- La elevación y el azimut (`alt, az`) usando `altaz()`.
- La ascensión recta y declinación (`ra, dec`) usando `radec()`.

Solo los satélites con `alt.degrees > 10` se consideran visibles. Los datos se registran como diccionarios individuales y se almacenan agrupados en `visible_gnss` para su análisis posterior.

#### 4.5.3. Cálculo de posiciones LEO

Del mismo modo, cada satélite LEO en `self.leo_sats` se evalúa en ese instante  $t$ , sin aplicar filtrado angular. Se almacenan sus coordenadas aparentes (`ra, dec`), altitud y azimut en una lista `leo_positions`, que servirá para la búsqueda de coincidencias visuales.

#### 4.5.4. Algoritmo de detección de bloqueo visual

El sistema considera que un satélite GNSS está bloqueado si existe un LEO cuya separación angular con él es menor a  $1^\circ$ . Para verificar esto:

- Se convierten las posiciones (`ra, dec`) de GNSS y LEO a vectores de coordenadas.
- Se construye un árbol `KDTree` a partir de `leo_positions`.
- Se calcula, para cada GNSS, la distancia angular al LEO más cercano mediante `tree.query()`.

- Se marcan como bloqueados aquellos cuya distancia sea menor al umbral, definido como:

```
BLOCK_THRESHOLD = np.radians(1.0)
```

Los GNSS bloqueados se añaden a `results['blocked_gnss'][constelación]`.

#### 4.5.5. Resultado y almacenamiento

Por cada paso de tiempo, se actualizan tres estructuras clave dentro de `results`:

- `gnss_visible`: añade un diccionario por satélite con su visibilidad y estado de bloqueo.
- `leo_positions`: añade la lista de LEO proyectados en ese instante.
- `blocked_gnss`: actualiza el conjunto de nombres GNSS bloqueados por constelación.

Una vez completada la simulación, el diccionario `results` se asocia al nombre del observador dentro de `self.results`, permitiendo acceder a los datos desde cualquier otra parte del sistema, incluyendo análisis globales de métricas GNSS y visualización de *skyplots*.

### 4.6. Cálculo de métricas GNSS–LEO a partir de la simulación

A partir de los resultados de visibilidad generados para cada observador, el sistema permite calcular automáticamente las métricas estadísticas definidas en la Sección 3.4.5. Esta funcionalidad está implementada en el método `calculate_metrics(observer_name)`, que opera sobre la estructura `self.results[observer_name]` obtenida durante la simulación. Adicionalmente, se emplea el método complementario `calculate_frequency_metrics(observer_name)` para computar métricas temporales agregadas.

#### 4.6.1. Probabilidad de Bloqueo Individual (PBI)

Para cada constelación GNSS, el sistema recorre todas las observaciones individuales registradas en el diccionario `gnss_visible`. En este proceso:

- Se contabiliza cuántas veces cada satélite fue observado por el observador.
- Se registra el número de instantes en los que dicho satélite fue bloqueado por un satélite LEO.

- A partir de estos conteos se calcula la proporción de observaciones bloqueadas sobre el total, generando así el valor de PBI correspondiente, conforme a la Ecuación (3.8).
- Esta información se almacena en la estructura `satellite_metrics`, junto con su constelación asociada.
- Además, el sistema identifica el satélite con mayor vulnerabilidad (mayor PBI) y lo guarda bajo la clave `most_affected_sat`.

#### 4.6.2. Índice Global de Bloqueo Visual (BI)

El cálculo del *Blocking Index* (BI) se implementa en varios niveles de agregación, a través del método `calculate_metrics(observer_name)`. Este método procesa los resultados de visibilidad de cada observador y devuelve múltiples métricas, entre ellas las relacionadas con el bloqueo visual global del sistema GNSS. En particular, se distinguen dos formas principales:

- **BI instantáneo ( $BI_t$ ):** para cada instante simulado, se calcula la fracción de satélites GNSS visibles que están siendo bloqueados por coincidencias angulares con satélites LEO, como en la ecuación (3.9). Esta métrica puntual se almacena como una lista de pares tiempo–valor en `bi_instantaneo`, y permite generar representaciones temporales del nivel de obstrucción en función de la hora del día.
- **BI promedio:** representa la media de los valores de BI instantáneo a lo largo de toda la simulación. Esta forma agregada se formaliza en la Ecuación (3.10), y proporciona una visión consolidada del grado de obstrucción visual sufrido por el sistema GNSS durante el intervalo completo. El resultado se guarda como `bi_promedio`.

Este enfoque permite diferenciar entre una visión acumulada y una visión dinámica del comportamiento GNSS bajo perturbación visual.

#### 4.6.3. Frecuencia Horaria de Eventos de Bloqueo

Esta métrica cuantifica el número de coincidencias visuales (bloqueos) por hora simulada. El método `calculate_frequency_metrics(observer_name)`:

- Recorre el diccionario `blocked_gnss`, donde se almacena para cada constelación GNSS el listado de satélites afectados en cada instante.
- Suma el número total de eventos registrados.

- Calcula la duración total de la simulación en horas, a partir de la diferencia entre el primer y último instante registrado en `data['times']`.
- Aplica la fórmula de la Ecuación (3.11) y guarda el resultado en `block_frequency`.

Esta métrica resulta útil para comparar distintas ubicaciones, configuraciones orbitales o niveles de densidad.

#### 4.6.4. Desglose horario por franja

El sistema también realiza un análisis por hora del día:

- Agrupa las observaciones GNSS por hora UTC.
- Para cada franja horaria, calcula:
  - Número total de satélites visibles.
  - Número total de satélites bloqueados.
  - Porcentaje de bloqueo en esa franja.
- Estos datos se almacenan en `hourly_stats`, lo cual permite generar gráficos y estudios de variabilidad diurna.

### 4.7. Cálculo del PDOP y análisis geométrico

#### 4.7.1. Estimación del PDOP en instantes representativos

El cálculo puntual del *Position Dilution of Precision* (PDOP), cuya definición formal se expone en la ecuación (3.5), se implementa mediante el método `calculate_pdop()`, incluido en la clase principal del simulador. Esta función:

- Toma como entrada el nombre del observador (`observer_name`) y un índice temporal.
- Recupera todos los satélites GNSS visibles no bloqueados en ese instante de simulación.
- Calcula la matriz de geometría a partir de los vectores unitarios satélite–receptor proyectados en coordenadas ECEF.
- Evalúa la matriz de covarianza  $Q$ , y extrae el PDOP como se describe en la Sección 3.3.1.

Este cálculo se aplica a tres momentos clave por ubicación (máximo, mínimo y valor medio de bloqueo), lo que permite analizar la sensibilidad de la calidad geométrica en función del grado de obstrucción GNSS–LEO.

#### 4.7.2. PDOP completo, filtrado y variación porcentual

Para estimar el impacto geométrico de los bloqueos visuales GNSS–LEO sobre la calidad del posicionamiento, se desarrolla una clase auxiliar específica, `PDOPAnalyzer`, que implementa el cálculo del PDOP bajo dos configuraciones:

- **PDOP completo:** evaluado mediante la función `calculate_full_pdop()`, considera todos los satélites GNSS visibles (sin filtrado por bloqueo).
- **PDOP filtrado:** obtenido con `calculate_pdop()`, incluye únicamente los satélites GNSS no bloqueados, es decir, aquellos sin coincidencias angulares con satélites LEO en el instante correspondiente. La función es la misma mencionada en el punto anterior, debido a que esta ya nos retorna el valor filtrado.

Ambas métricas se aplican de forma iterativa sobre toda la jornada simulada mediante el método `calculate_pdop_variation()`. Esta función centraliza el análisis y devuelve los siguientes vectores de datos:

- `pdop_full`: serie temporal de PDOP completo.
- `pdop_filtered`: serie temporal de PDOP tras el filtrado por bloqueos.
- `delta_pdop_percent`: variación porcentual entre ambas, conforme a la Ecuación (3.7).

La métrica  $\Delta\text{PDOP}$  permite cuantificar el deterioro de la solución geométrica inducido exclusivamente por la pérdida de visibilidad GNSS. Esta comparación es especialmente útil para:

- Identificar ubicaciones geográficas más vulnerables al deterioro.
- Evaluar la sensibilidad de la geometría de observación ante distintos niveles de densidad LEO.
- Justificar la necesidad de filtrado geométrico en simulaciones GNSS–LEO.

#### 4.7.3. Estadísticas horarias de PDOP

El método `calculate_hourly_pdop_stats()` permite analizar el comportamiento horario del PDOP a lo largo del día simulado. Para cada hora UTC, se agregan:

- `avg_pdop_full`: promedio del PDOP completo.
- `avg_pdop_filtered`: promedio del PDOP filtrado.
- `avg_delta_pdop`: variación porcentual media.
- `max_delta_pdop`: máximo deterioro registrado durante esa hora.

Estos indicadores permiten detectar ventanas críticas en las que la geometría GNSS se degrada sustancialmente por la presencia de satélites LEO, facilitando un análisis horario comparativo por ubicación y densidad orbital.

### 4.8. Generación de informes y visualizaciones avanzadas

Una vez calculadas las métricas clave para un observador dado, el sistema proporciona dos herramientas complementarias para la explotación de resultados: el generador de informes textuales y el módulo de visualización gráfica. Ambos están integrados en la arquitectura del simulador y funcionan a partir de los datos almacenados internamente en la variable `self.results`.

#### 4.8.1. Informe textual automático

La función `generate_report(observer_name, pdop_data)` permite generar un resumen escrito de todos los resultados clave para un observador específico. Internamente, esta función:

- Invoca a `calculate_metrics()` para obtener todas las métricas de visibilidad y bloqueo.
- Si no se le pasa explícitamente `pdop_data`, calcula los valores de PDOP usando `PDOPAnalyzer`.
- Obtiene la frecuencia de eventos de bloqueo mediante `calculate_frequency_metrics()`.
- Evalúa el PDOP en tres momentos clave del día (máximo, mínimo y promedio de bloqueo visual).

El resultado es un bloque de texto que resume:

- El porcentaje promedio de bloqueo (`bi_promedio`).
- El satélite GNSS más afectado por bloqueos (con nombre, constelación y PBI).
- Una tabla resumen por constelación (observaciones totales, bloqueadas y porcentaje).
- Un análisis de PDOP en horas clave.
- Métricas agregadas como PDOP completo, filtrado y variación porcentual promedio.

Al finalizar la simulación para una combinación concreta de observador y densidad de constelación LEO, se genera un conjunto de archivos de salida en formato texto plano (`.txt`) que recogen los principales resultados numéricos. Todos estos archivos se almacenan automáticamente en un subdirectorio con la siguiente estructura:

```
./resultados/{observer_name}_{density}/
```

Si dicho directorio no existe, el propio código se encarga de crearlo. Dentro de esta carpeta, se generan los siguientes informes:

- **Informe resumen:** incluye una descripción textual de las principales métricas obtenidas (bloqueo visual, estadísticas por constelación, análisis de PDOP en momentos clave, etc.):

```
reporte_{observer_name}_{density}.txt
```

- **Variación porcentual del PDOP ( $\Delta$ PDOP):** contiene los valores de  $\Delta$ PDOP calculados en cada instante simulado, junto con sus marcas temporales UTC. Este archivo permite representar la evolución temporal de la degradación geométrica provocada por los bloqueos:

```
delta_pdop_{observer_name}_{density}.txt
```

- **Bloqueo Visual Instantáneo ( $BI_t$ ):** almacena la evolución del índice de bloqueo visual en cada instante temporal de la simulación, expresado en porcentaje. Estos datos son útiles para generar gráficos cronológicos y evaluar la dinámica del entorno LEO:

```
bi_instantaneo_{observer_name}_{density}.txt
```

Estos archivos están codificados en UTF-8 y utilizan el tabulador como separador. Gracias a esta estructura, facilitamos el análisis posterior y la comparación entre configuraciones orbitales o ubicaciones geográficas distintas.

#### 4.8.2. Visualizer para generar gráficas

El módulo `Visualizer` contiene funciones diseñadas para representar visualmente los resultados simulados. Su constructor acepta como parámetro una instancia del simulador principal, desde la cual accede a las posiciones GNSS, los ángulos de elevación, los eventos de bloqueo y los PDOP asociados.

Los nombres de los archivos gráficos generados siguen una convención clara según su contenido:

- `skyplot_{observer_name}_{density}_{time_index}.png`: Diagramas polares instantáneos (skyplots) en un instante de tiempo concreto.
- `blocking_timeline_{observer_name}_{density}.png`: Evolución de la probabilidad de bloqueo GNSS por hora del día.
- `pdop_heatmap_{observer_name}_{density}.png`: Heatmap que muestra la variación del PDOP según el azimut y la hora UTC.
- `pdop_comparison_{observer_name}_{density}.png`: Gráfico comparativo entre el PDOP completo y el filtrado, junto con la evolución de  $\Delta$ PDOP.

Cada función de visualización admite un parámetro opcional `save_path`, que permite guardar el gráfico directamente como archivo `.png` con calidad ajustable. Si este parámetro no se especifica, la figura se retorna como objeto `matplotlib.Figure`, pudiendo mostrarse o insertarse en otro flujo visual.

#### 4.8.3. Gráfico de evolución temporal del BI instantáneo ( $BI_t$ )

Para cada observador y configuración de densidad, se genera un gráfico de línea que muestra la evolución temporal del Índice de Bloqueo Visual instantáneo ( $BI_t$ ) a lo largo del día. Este gráfico se construye a partir de la lista de tuplas almacenadas en `instantaneous.bi`, generada por `calculate_metrics()`, y se representa usando `matplotlib`.

- El eje X corresponde a la hora UTC simulada.
- El eje Y indica el valor de  $BI_t$  en porcentaje.
- El gráfico se guarda automáticamente como:

`./resultados/{observer_name}_{density}/bi_timeline_{observer_name}_{density}.png`

Este gráfico permite observar los patrones horarios de obstrucción de visibilidad GNSS en una ubicación específica, y resulta útil para detectar franjas de mayor vulnerabilidad a interferencias visuales por satélites LEO.

#### 4.8.4. Análisis global: generar\_graficas\_por\_densidad()

La función `generar_graficas_por_densidad()` coordina la simulación completa y la generación de visualizaciones globales que comparan el comportamiento de distintos observadores ante tres configuraciones de densidad LEO: `low`, `medium` y `full`. Para cada caso:

1. Se inicializa un objeto `GNSSLEOSimulator` y se cargan las constelaciones GNSS y LEO correspondientes.
2. Se simula la visibilidad para cada observador, guardando:
  - Series de PDOP completo, PDOP filtrado y  $\Delta$ PDOP.
  - Índices  $BI_t$  individuales por instante de tiempo.
3. Se recopilan los datos y se generan tres tipos de gráficas globales:
  - **Boxplot de PDOP completo vs filtrado:** compara la distribución de ambos tipos de PDOP según la densidad de satélites LEO.  
`boxplot_pdop_vs_density.png`
  - **Stripplot de  $\Delta$ PDOP individual:** muestra la variabilidad de deterioro geométrico por observador.  
`delta_pdop_vs_density_individual.png`
  - **Stripplot de  $BI_t$  individual:** permite comparar el comportamiento horario de visibilidad GNSS entre distintas ubicaciones.  
`bi_t_vs_density_individual.png`

Todos los gráficos se almacenan en la carpeta:

```
./resultados/por_densidad/
```

Este enfoque nos proporciona una base sólida para evaluar comparativamente el impacto de distintas densidades de constelaciones LEO sobre la degradación geométrica (PDOP) y la pérdida visual (bloqueos GNSS). A partir de las simulaciones y visualizaciones generadas, el siguiente capítulo analiza en detalle cómo varían estas métricas según la ubicación del observador y la configuración orbital, permitiendo extraer conclusiones sobre la influencia operativa de Starlink sobre la infraestructura GNSS actual.



## Capítulo 5

# Análisis y discusión de resultados

A lo largo de este capítulo se recogen y analizan los resultados de las simulaciones previamente descritas. La evaluación de métricas como PDOP, bloqueo visual o variación porcentual se organiza en dos caminos paralelos: uno enfocado en cada ubicación observadora, y otro centrado en los distintos niveles de densidad orbital. Con el fin de entender, con mayor claridad, hasta qué punto la constelación Starlink puede llegar a alterar la operatividad de los sistemas GNSS.

### 5.1. Tromsø

Es una ciudad situada en el norte de Noruega, localizada aproximadamente a  $69,66^{\circ}$  de latitud norte y  $18,94^{\circ}$  de longitud este. Su ubicación dentro del Círculo Polar Ártico la convierte en un entorno ideal para evaluar el comportamiento geométrico de sistemas GNSS en latitudes elevadas, donde las geometrías satelitales tienden a ser más desfavorables. Además, cuenta con una estación permanente GNSS (TRO1) operada por la Red Permanente Europea (EUREF), que sirve como referencia geodésica para múltiples estudios científicos y aplicaciones de posicionamiento de alta precisión [16].

#### 5.1.1. Resultados y análisis por densidad orbital

La evaluación del impacto de la constelación Starlink sobre la observabilidad GNSS en Tromsø se realizó para tres niveles de densidad orbital según lo planificado: baja (`low`), media (`medium`) y completa (`full`). A continuación, se muestran y analizan las visualizaciones generadas para cada escenario, abordando aspectos como la geometría satelital (`skyplot`), la evolución del PDOP, el índice global de bloqueo (BI) y la variación porcentual de PDOP ( $\Delta\text{PDOP}$ ).

### 5.1.1.1. Densidad baja

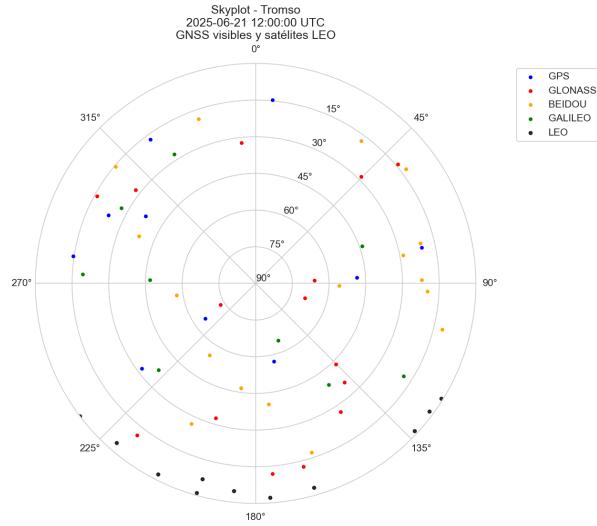


Figura 5.1: Skyplot GNSS con densidad LEO baja (Tromsø)

El skyplot en condiciones de baja densidad (Figura 5.1) ofrece una visión general de los satélites GNSS visibles sobre Tromsø, representando una escena simplificada del entorno satelital. Se observa una distribución razonable de satélites de múltiples constelaciones, pero sin una superposición significativa de trayectorias con los satélites Starlink.

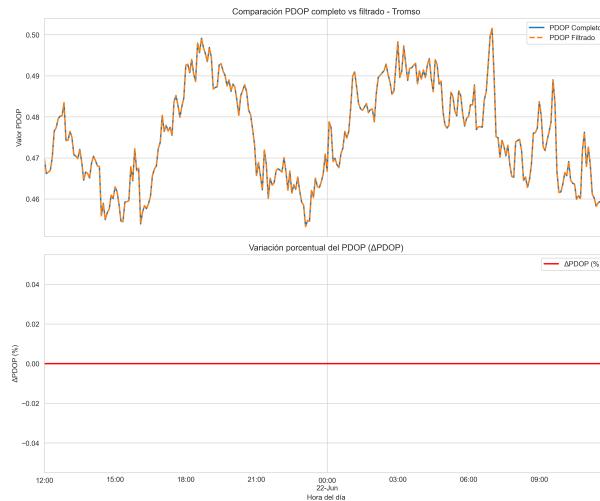


Figura 5.2: PDOP completo y filtrado con densidad LEO baja

La Figura 5.2 muestra la evolución del PDOP completo y filtrado. Ambas curvas permanecen completamente superpuestas durante las 24 horas simuladas, y no se registra ninguna variación porcentual significativa. Esto indica que no se ha producido ningún bloqueo visual por parte de

los satélites LEO durante este período.

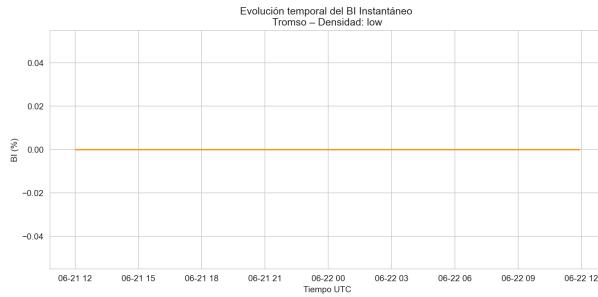


Figura 5.3: Índice de bloqueo visual (BI) – densidad LEO baja

En la Figura 5.3, la evolución del índice de bloqueo visual (BI) se mantiene completamente nula, confirmando que la constelación LEO, en su configuración más reducida, no interfiere con la visibilidad GNSS desde Tromsø.

#### 5.1.1.2. Densidad media

Al aumentar la densidad orbital a un nivel medio, se observan cambios notables. El skyplot de la Figura 5.4 revela una mayor concentración de satélites Starlink, especialmente en la región inferior del horizonte, lo que anticipa posibles bloqueos.

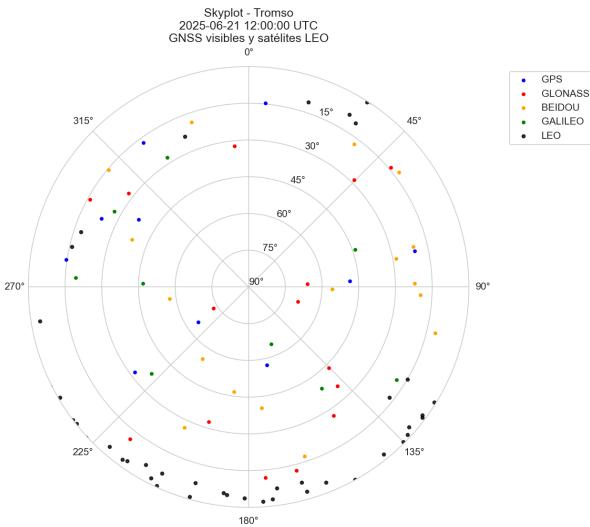


Figura 5.4: Skyplot GNSS con densidad LEO media (Tromsø)

La Figura 5.5 confirma esta expectativa, mostrando divergencias visibles entre el PDOP completo y el filtrado, especialmente durante la madrugada y primeras horas del día. En estos

intervalos, la pérdida de algunos satélites GNSS debido a interferencias visuales por LEO genera un aumento relativo en el PDOP.

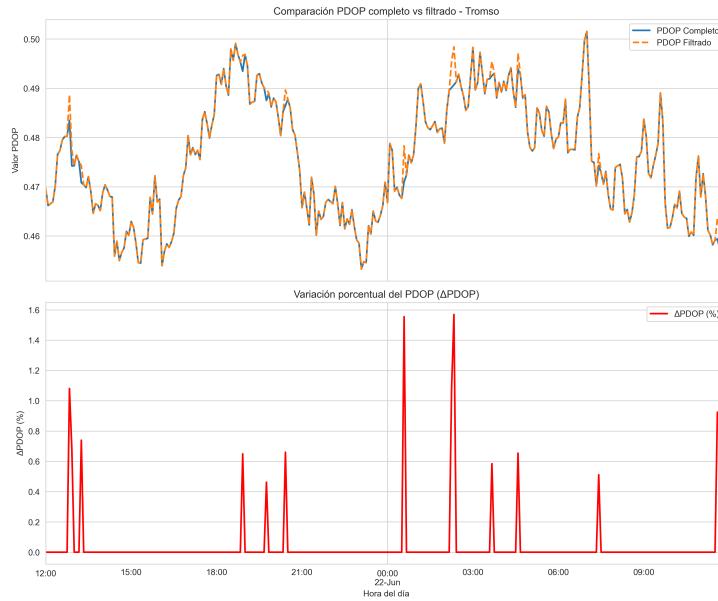


Figura 5.5: PDOP completo y filtrado con densidad LEO media

La Tabla 5.1 recoge los registros con valores de  $\Delta\text{PDOP}$  superiores al 1%. Estos picos se concentran principalmente entre medianoche y el amanecer, lo que sugiere una mayor probabilidad de interferencia visual cuando las órbitas LEO intersectan los ángulos de observación más críticos.

Timestamp (UTC)	$\Delta\text{PDOP} (\%)$
2025-06-21 12:50:00	1.08058
2025-06-22 00:35:00	1.55444
2025-06-22 02:15:00	1.07643
2025-06-22 02:20:00	1.56931

Tabla 5.1: Registros con valores  $\Delta\text{PDOP}$  superiores a 1%

La evolución temporal del índice de bloqueo visual en este escenario, representada en la Figura 5.6, muestra un comportamiento más dinámico. Aunque los valores no son extremos, ciertos valores alcanzan un  $\text{BI}_t$  superior al 2% en varios momentos del día.

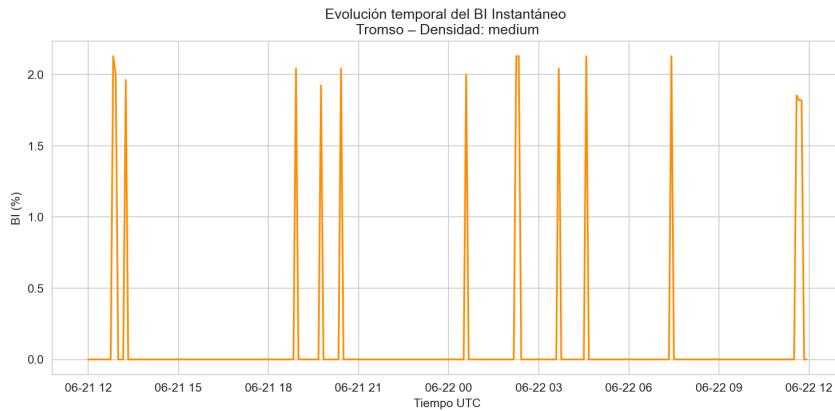


Figura 5.6: Índice de bloqueo visual (BI) – densidad LEO media

La Tabla 5.2 detalla los momentos exactos donde el índice supera el 1%, mostrando una recurrencia de bloqueos leves a moderados a lo largo del día, con énfasis en las franjas horarias comprendidas entre el mediodía y las primeras horas de la madrugada.

Timestamp (UTC)	BI <sub>t</sub> (%)
2025-06-21 12:50:00	2.12766
2025-06-21 12:55:00	2.00000
2025-06-21 13:15:00	1.96078
2025-06-21 18:55:00	2.04082
2025-06-21 19:45:00	1.92308
2025-06-21 20:25:00	2.04082
2025-06-22 00:35:00	2.00000
2025-06-22 02:15:00	2.12766
2025-06-22 02:20:00	2.12766
2025-06-22 03:40:00	2.04082
2025-06-22 04:35:00	2.12766
2025-06-22 07:25:00	2.12766
2025-06-22 11:35:00	1.85185
2025-06-22 11:40:00	1.81818
2025-06-22 11:45:00	1.81818

Tabla 5.2: Registros con valores BI<sub>t</sub> superiores a 1%

En conjunto, el escenario de densidad media revela un impacto ya perceptible sobre la calidad de observación GNSS: se manifiestan eventos de bloqueo visual de forma recurrente, las variaciones de  $\Delta$ PDOP se intensifican, y la disponibilidad geométrica comienza a mostrar signos de degradación.

### 5.1.1.3. Densidad alta

En el escenario de densidad completa, los efectos del bloqueo se acentúan significativamente. El skyplot (Figura 5.7) muestra un cielo densamente poblado por trayectorias LEO, lo que incrementa drásticamente la probabilidad de interferencia.

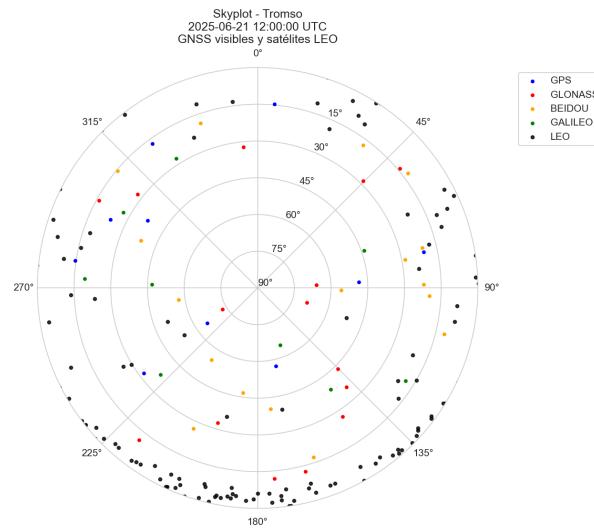


Figura 5.7: Skyplot GNSS con densidad LEO completa (Tromsø)

La Figura 5.8 evidencia una degradación más marcada. La divergencia entre PDOP completo y filtrado es frecuente, y los valores de  $\Delta\text{PDOP}$  superan el 2 % en varios momentos críticos. La pérdida de visibilidad directa a varios satélites GNSS genera una degradación geométrica significativa.

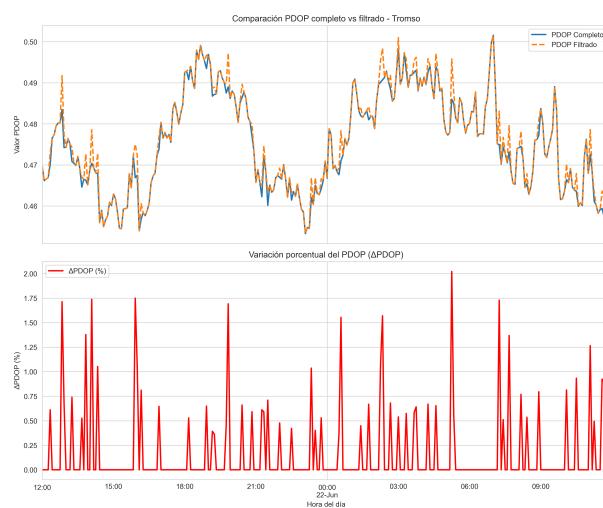


Figura 5.8: PDOP completo y filtrado con densidad LEO completa

La Tabla 5.3 resume los picos más relevantes de  $\Delta\text{PDOP}$ , todos ellos superiores al 1.2 %, con concentraciones visibles en la mañana y madrugada. Estos datos refuerzan la conclusión de que la superposición de trayectorias en zonas críticas reduce efectivamente la calidad de la solución GNSS.

Timestamp (UTC)	$\Delta\text{PDOP} (%)$
2025-06-21 12:50:00	1.71333
2025-06-21 13:50:00	1.37833
2025-06-21 14:05:00	1.73736
2025-06-21 14:20:00	1.05281
2025-06-21 15:55:00	1.74879
2025-06-21 16:00:00	1.05057
2025-06-21 19:50:00	1.69155
2025-06-21 23:20:00	1.03627
2025-06-22 00:35:00	1.55444
2025-06-22 02:15:00	1.07643
2025-06-22 02:20:00	1.56931
2025-06-22 05:15:00	2.02089
2025-06-22 07:15:00	1.72930
2025-06-22 11:05:00	1.26509
2025-06-22 11:45:00	2.00338

Tabla 5.3: Registros con valores  $\Delta\text{PDOP}$  superiores a 1.2 %

En paralelo, el índice de bloqueo visual también refleja un aumento considerable. La Figura 5.9 muestra múltiples tramos del día con BI superior al 2.5 %, indicando bloqueos simultáneos de varios enlaces satelitales.

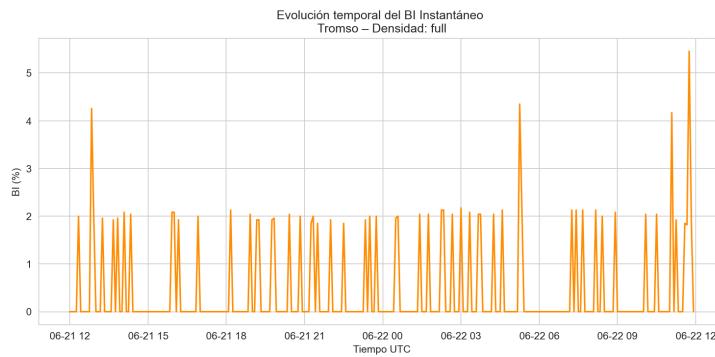


Figura 5.9: Índice de bloqueo visual (BI) – densidad LEO completa

La Tabla 5.4 identifica los momentos en los que el  $\text{BI}_t$  alcanza valores máximos, incluso

superando el 5 %. Esto representa una situación de saturación en ciertas ventanas temporales, donde las condiciones geométricas para GNSS se ven gravemente afectadas.

Timestamp (UTC)	BI <sub>t</sub> (%)
2025-06-21 12:50:00	4.25532
2025-06-22 05:15:00	4.34783
2025-06-22 11:05:00	4.16667
2025-06-22 11:45:00	5.45455

Tabla 5.4: Registros con valores BI<sub>t</sub> superiores al 2.5 %

El escenario de máxima densidad revela un patrón claro de saturación visual. Las trayectorias LEO invaden amplias regiones del cielo, solapándose con múltiples GNSS a lo largo del día. El PDOP filtrado presenta picos pronunciados en torno a mediodía y en la madrugada, reflejando la pérdida temporal de visibilidad en configuraciones clave.

Para facilitar una interpretación global, la Tabla 5.5 resume las métricas clave para las tres configuraciones de densidad. Se observa un crecimiento exponencial del impacto a medida que aumenta la cantidad de satélites LEO, afectando al índice de bloqueo, la frecuencia de eventos y la degradación promedio del PDOP.

Métrica	Baja	Media	Completa
Índice de bloqueo visual promedio (BI) [%]	0.00	0.10	0.43
Frecuencia de bloqueos [eventos/s/h]	0.0	0.6	2.6
Satélite GNSS más afectado	GPS PRN 22 (0.00 %)	BeiDou C06 (1.90 %)	Galileo 21 (2.50 %)
PDOP completo promedio	0.47550	0.47550	0.47550
PDOP filtrado promedio	0.47550	0.47570	0.47631
ΔPDOP promedio [%]	0.00000	0.04329	0.17044
Proporción total de observaciones bloqueadas [%]	0.00	0.11	0.44
Hora de máximo bloqueo	12:00	18:00	15:00
PDOP en la hora de máximo bloqueo	0.47	0.49	0.46

Tabla 5.5: Comparación de métricas GNSS–LEO para Tromsø según densidad orbital

También es importante aclarar ciertas métricas presentadas que podrían inducir a confusión.

La *Hora de máximo bloqueo* indicada en la tabla se refiere al momento del día en el que el satélite GNSS más afectado ha experimentado su mayor nivel de interferencia visual por parte de un satélite LEO, según su probabilidad de bloqueo individual, definida en la Ecuación 3.8. Este valor está expresado en el campo correspondiente junto al identificador del satélite. Las tablas restantes mantienen esta misma lógica de presentación para facilitar la interpretación coherente de los resultados.

### 5.1.2. Conclusión de los resultados

En síntesis, los resultados obtenidos para Tromsø confirman que la ubicación geográfica desempeña un papel crucial en la interacción entre las constelaciones LEO y los sistemas GNSS. Debido a su proximidad a trayectorias orbitales polares, esta ciudad experimenta un impacto más severo a medida que se incrementa la densidad satelital. Mientras que el escenario **low** no presenta efectos significativos sobre la visibilidad ni sobre la geometría de posicionamiento, los escenarios **medium** y, especialmente, **full**, evidencian bloqueos visuales recurrentes, incrementos notables en  $\Delta\text{PDOP}$  y una degradación progresiva de la solución geométrica.

Para facilitar el análisis cuantitativo, se establecieron umbrales mínimos en la presentación de resultados tabulados:  $\Delta\text{PDOP}$  superiores al 1% y valores de  $\text{BI}_t$  mayores al 1% o 2.5%, dependiendo del contexto. Estos umbrales no representan límites operativos absolutos, sino que se utilizan como criterio para filtrar y destacar los picos más significativos en cada escenario, permitiendo así una identificación clara de los momentos de máxima interferencia.

## 5.2. Quito

Quito, capital de Ecuador, se encuentra ubicada cerca del ecuador terrestre, a aproximadamente  $0,23^\circ$  de latitud sur y  $78,52^\circ$  de longitud oeste. Esta localización ecuatorial ofrece condiciones únicas para el análisis de geometrías GNSS, ya que la distribución angular de los satélites suele ser más simétrica. Estudios previos han analizado específicamente el rendimiento GNSS en entornos urbanos de Quito, subrayando los desafíos particulares que presenta esta región para la precisión y disponibilidad de señales [18].

### 5.2.1. Resultados y análisis por densidad orbital

En esta sección se analiza el impacto de la constelación Starlink en la observabilidad GNSS desde Quito. Para ello, se evalúan los tres escenarios de densidad orbital definidos previamente en este estudio: baja, media y alta.

### 5.2.1.1. Densidad baja

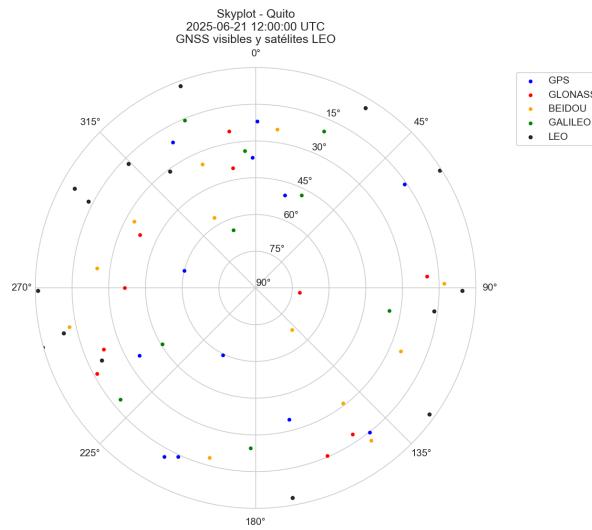


Figura 5.10: Skyplot GNSS con densidad LEO baja (Quito)

La Figura 5.10 presenta el skyplot para el escenario de baja densidad LEO. Se observa una presencia mínima y dispersa de satélites de órbita baja, con una cobertura geométrica dominada por las constelaciones GNSS tradicionales, principalmente GPS y BeiDou.

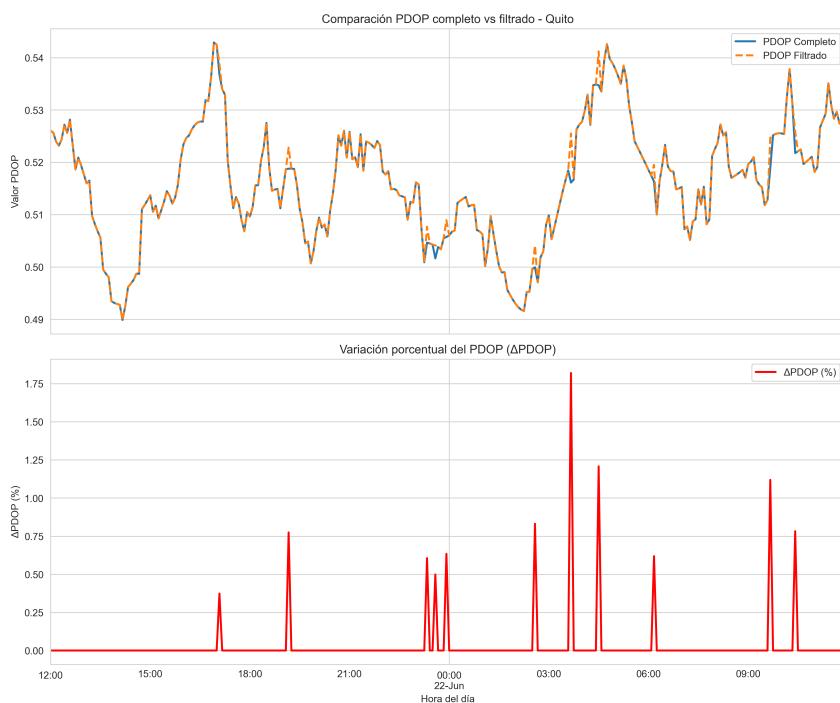


Figura 5.11: PDOP completo y filtrado con densidad LEO baja

La Figura 5.11 compara la evolución del PDOP completo (sin bloqueos) y el PDOP filtrado (considerando bloqueos LEO). Para esta latitud ecuatorial, se aprecian variaciones notables, con picos de degradación ( $\Delta$ PDOP) que ocurren principalmente durante la mañana y cerca del mediodía. El evento más destacable corresponde a un incremento porcentual del PDOP que supera el 1.8 %, como se detalla a continuación.

Timestamp (UTC)	$\Delta$ PDOP (%)
2025-06-21 19:10:00	0.77419
2025-06-22 02:35:00	0.83103
2025-06-22 03:40:00	1.81961
2025-06-22 04:30:00	1.20720
2025-06-22 09:40:00	1.11909
2025-06-22 10:25:00	0.78228

Tabla 5.6: Registros con valores  $\Delta$ PDOP superiores a 0.7 %

La Tabla 5.6 cuantifica los instantes de mayor degradación porcentual del PDOP ( $\Delta$ PDOP). Estos valores confirman que la pérdida de satélites GNSS por bloqueos de la constelación Starlink deteriora la geometría satelital. Eventos con  $\Delta$ PDOP superiores al 1 %, y un máximo de 1.82 %, evidencian este impacto negativo.

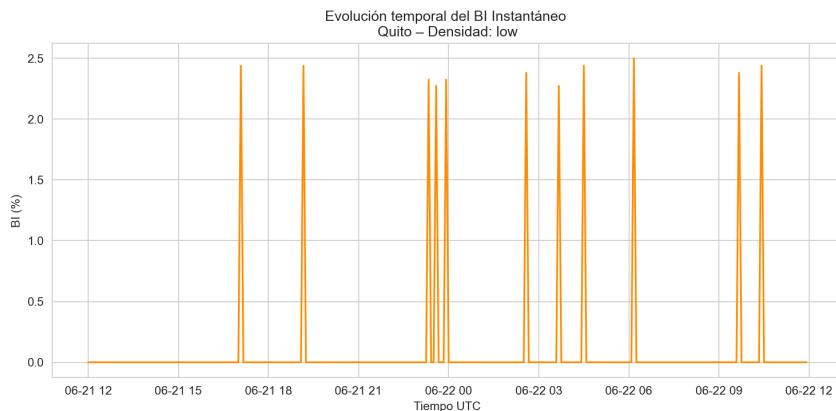


Figura 5.12: Índice de bloqueo visual (BI) – densidad LEO baja

La evolución temporal del Índice de Bloqueo Visual (BI) para el escenario de baja densidad se ilustra en la Figura 5.12. Se observa que los instantes de mayor variación del PDOP tienden a coincidir con picos en el BI, los cuales superan el 2 %. Estos picos representan el porcentaje de satélites GNSS bloqueados simultáneamente, confirmando la causa de la degradación geométrica.

Timestamp (UTC)	BI <sub>t</sub> (%)
2025-06-21 17:05:00	2.43902
2025-06-21 19:10:00	2.43902
2025-06-21 23:20:00	2.32558
2025-06-21 23:35:00	2.27273
2025-06-21 23:55:00	2.32558
2025-06-22 02:35:00	2.38095
2025-06-22 03:40:00	2.27273
2025-06-22 04:30:00	2.43902
2025-06-22 06:10:00	2.50000
2025-06-22 09:40:00	2.38095
2025-06-22 10:25:00	2.43902

Tabla 5.7: Registros con valores BI<sub>t</sub> superiores a 2 %

La Tabla 5.7 detalla los valores máximos del BI instantáneo (BI<sub>t</sub>). Estas cifras, todas superiores al 2 %, corroboran que, incluso en un escenario de baja densidad, se producen momentos en los que un porcentaje significativo de los satélites GNSS visibles son obstruidos por la constelación Starlink.

En resumen, el escenario de baja densidad ya presenta un impacto medible sobre la observabilidad GNSS en una latitud ecuatorial. Aunque el efecto promedio es bajo, se registran eventos puntuales de bloqueo que provocan degradaciones significativas tanto en el PDOP como en el BI, evidenciando un deterioro temporal de la geometría satelital.

### 5.2.1.2. Densidad media

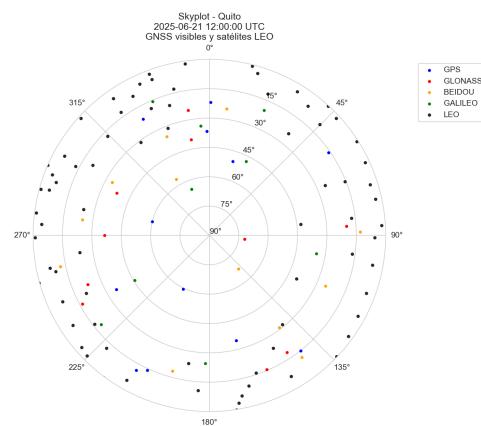


Figura 5.13: Skyplot GNSS con densidad LEO media (Quito)

La Figura 5.13 ilustra el skyplot para el escenario de densidad media. Se observa un incremento notable en la cantidad de satélites Starlink visibles. Sus trayectorias comienzan a superponerse con las de los satélites GNSS, lo que anticipa una mayor probabilidad de eventos de bloqueo en comparación con el escenario de baja densidad.

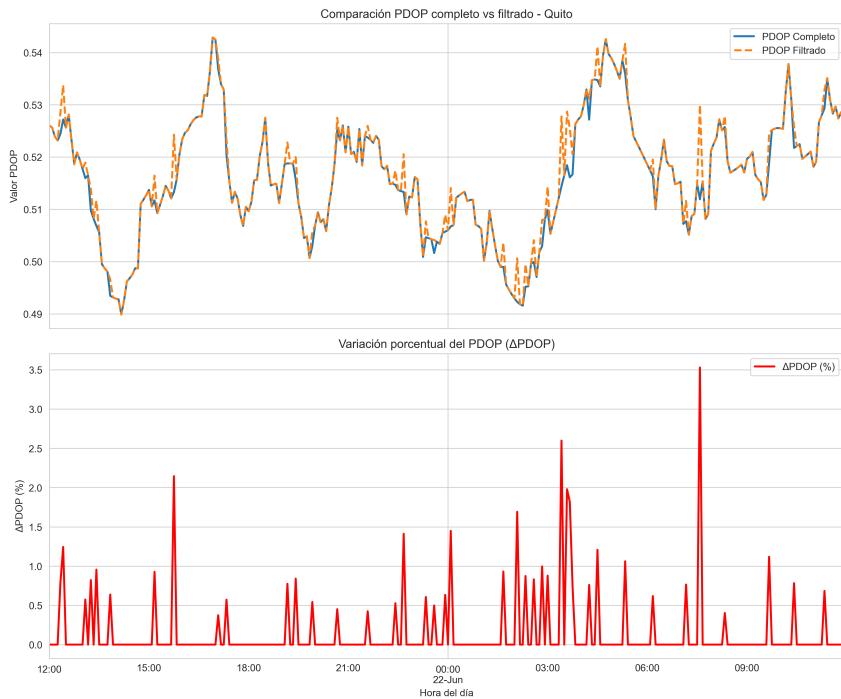


Figura 5.14: PDOP completo y filtrado con densidad LEO media

La evolución del PDOP, mostrada en la Figura 5.14, confirma el impacto del aumento de densidad. Se aprecian divergencias claras y frecuentes entre la curva del PDOP completo y la del filtrado, lo que indica una degradación geométrica recurrente. La gráfica inferior de variación porcentual ( $\Delta$ PDOP) revela múltiples picos que superan el 1.5 %, alcanzando un valor máximo superior al 3.5 % entre las 07:00 y 08:00 UTC.

Timestamp (UTC)	$\Delta$ PDOP (%)
2025-06-21 15:45:00	2.14471
2025-06-22 02:05:00	1.69139
2025-06-22 03:25:00	2.59783
2025-06-22 03:35:00	1.97892
2025-06-22 03:40:00	1.81961
2025-06-22 07:35:00	3.52957

Tabla 5.8: Registros con valores  $\Delta$ PDOP superiores a 1.5 %

La Tabla 5.8 cuantifica los eventos de mayor degradación, confirmando un pico máximo de 3.53 % a las 07:35 UTC. La frecuencia y magnitud de estos valores demuestran que el aumento en la densidad LEO se traduce directamente en un deterioro significativo y medible de la geometría GNSS.

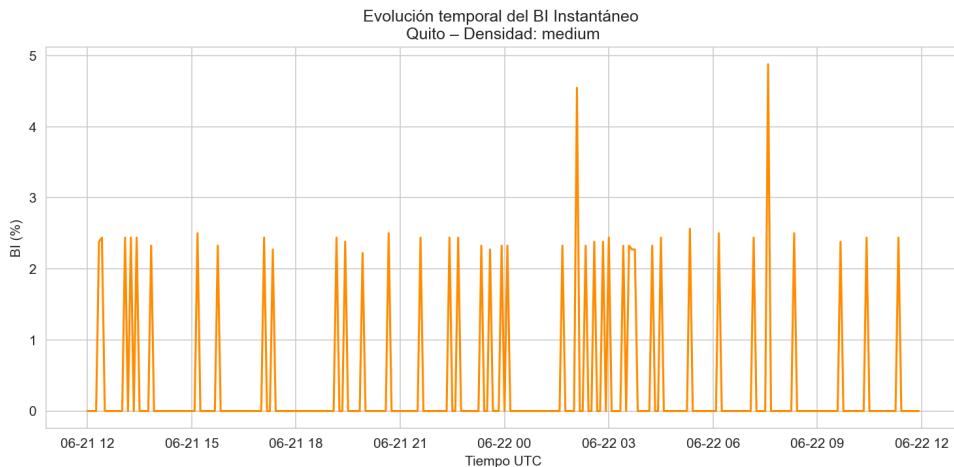


Figura 5.15: Índice de bloqueo visual (BI) – densidad LEO media

El análisis del Índice de Bloqueo Visual (BI), presentado en la Figura 5.15, corrobora la causa de esta degradación. La gráfica muestra picos instantáneos que superan el 4 %, lo que indica que un porcentaje significativo de satélites GNSS son obstruidos simultáneamente. Este comportamiento es consistente con el aumento de la densidad LEO y su impacto en una ubicación ecuatorial.

Timestamp (UTC)	BI <sub>t</sub> (%)
2025-06-22 02:05:00	4.54545
2025-06-22 07:35:00	4.87805

Tabla 5.9: Registros con valores BI<sub>t</sub> superiores al 3 %

La Tabla 5.9 detalla los dos eventos de bloqueo más severos, que alcanzan casi el 5 % y coinciden con momentos de alta degradación del PDOP. En conclusión, el escenario de densidad media evidencia un deterioro notable de la geometría GNSS en Quito. Los picos recurrentes en el  $\Delta$ PDOP, junto con valores elevados de BI, demuestran que el impacto de la constelación Starlink deja de ser marginal y se convierte en un factor relevante para aplicaciones que requieran alta disponibilidad y precisión.

### 5.2.1.3. Densidad alta

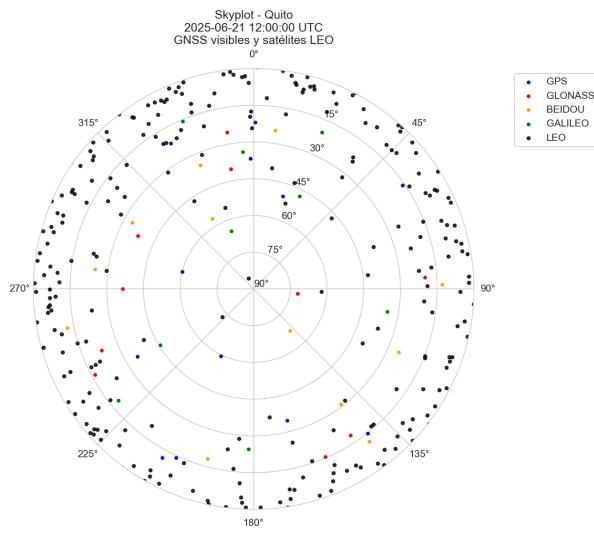


Figura 5.16: Skyplot GNSS con densidad LEO completa (Quito)

El escenario de alta densidad, ilustrado en la Figura 5.16, muestra una saturación visual significativa. Las trayectorias de los satélites Starlink cubren una porción considerable del cielo visible, creando numerosas superposiciones con los enlaces GNSS. Este incremento en la densidad anticipa una frecuencia y magnitud de bloqueos considerablemente mayor, como se analizará a continuación.

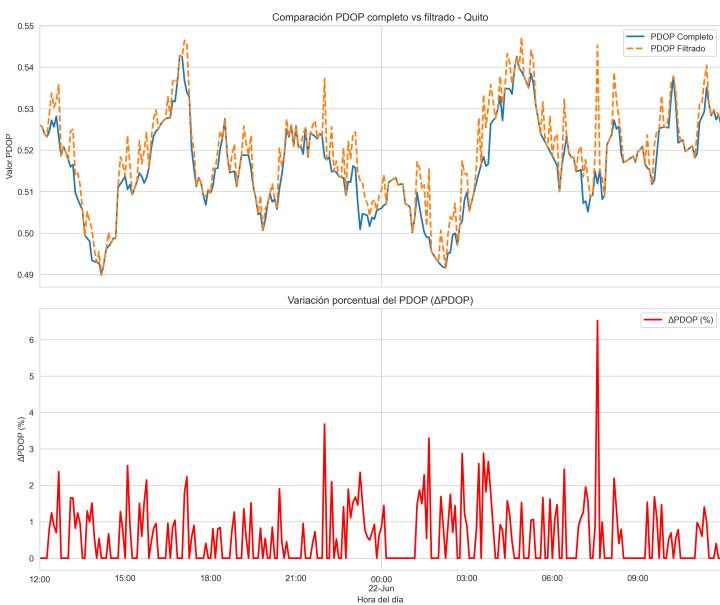


Figura 5.17: PDOP completo y filtrado con densidad LEO completa

La Figura 5.17 confirma el severo impacto de este escenario. La divergencia entre las curvas de PDOP completo y filtrado es constante y pronunciada, lo que evidencia una degradación geométrica persistente. La gráfica de variación porcentual ( $\Delta$ PDOP) revela picos de degradación notables, con tres eventos superando el 3 % y un máximo que excede el 6.5 %, demostrando un deterioro sustancial de la geometría satelital.

Timestamp (UTC)	$\Delta$ PDOP (%)
2025-06-21 22:00:00	3.67913
2025-06-22 01:40:00	3.29557
2025-06-22 07:35:00	6.52349

Tabla 5.10: Registros con valores  $\Delta$ PDOP superiores a 3 %

La Tabla 5.10 cuantifica estos picos, registrando el evento de máxima degradación (6.52 %) a las 07:35 UTC. Esta tendencia se correlaciona directamente con un aumento en los eventos de bloqueo visual, como se analiza a continuación.

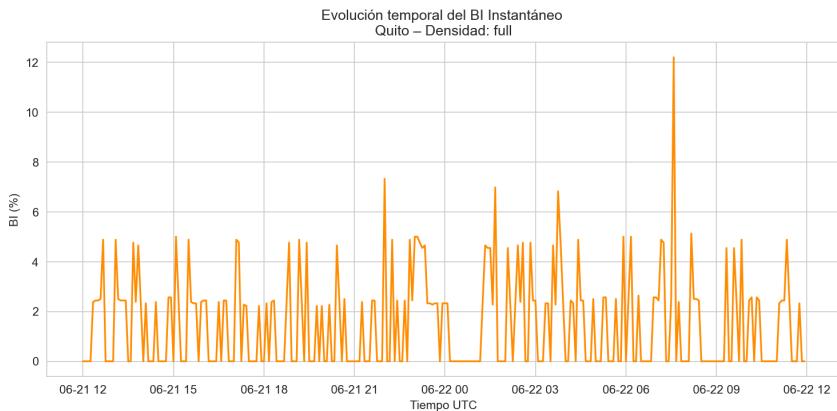


Figura 5.18: Índice de bloqueo visual (BI) – densidad LEO completa

El Índice de Bloqueo Visual en la Figura 5.18 confirma esta relación causa-efecto. La gráfica muestra un BI persistentemente elevado, con múltiples picos que superan el 5 %. La Tabla 5.11 detalla estos eventos, destacando un máximo de 12.2 % a las 07:35 UTC. Este valor, que coincide temporalmente con el pico de  $\Delta$ PDOP, indica que más de uno de cada diez satélites GNSS visibles estaban obstruidos en ese instante, explicando así la severa degradación geométrica observada.

Timestamp (UTC)	BI <sub>t</sub> (%)
2025-06-21 15:05:00	5.00000
2025-06-21 22:00:00	7.31707
2025-06-21 23:00:00	5.00000
2025-06-21 23:05:00	5.00000
2025-06-22 01:40:00	6.97674
2025-06-22 03:45:00	6.81818
2025-06-22 05:55:00	5.00000
2025-06-22 06:10:00	5.00000
2025-06-22 07:35:00	12.19512
2025-06-22 08:10:00	5.12821

Tabla 5.11: Registros con valores BI<sub>t</sub> superiores a 5 %

La Tabla 5.11 detalla los eventos de bloqueo más severos, con valores de BI que superan consistentemente el 5 % y alcanzan un máximo del 12.2 %. Estos picos, concentrados en la madrugada y el amanecer, coinciden temporalmente con las mayores degradaciones del PDOP, confirmando un patrón de interferencia visual severa en el escenario de máxima densidad.

Métrica	Baja	Media	Completa
Índice de bloqueo visual promedio (BI) [%]	0.09	0.36	1.50
Frecuencia de bloqueos [eventos/h]	0.5	1.8	7.5
Satélite GNSS más afectado	BeiDou-3 M23(2.36 %)	BeiDou-3 M23(3.94 %)	GSAT0206(7.84 %)
PDOP completo promedio	0.51605	0.51605	0.51605
PDOP filtrado promedio	0.51621	0.51679	0.51905
ΔPDOP promedio [%]	0.03216	0.14396	0.58313
Proporción total de observaciones bloqueadas [%]	0.09	0.36	1.50
Hora de máximo bloqueo	23:00	23:00	05:00
PDOP en la hora de máximo bloqueo	0.52	0.52	0.54

Tabla 5.12: Comparación de métricas GNSS-LEO para Quito según densidad orbital

Para facilitar una interpretación global, la Tabla 5.12 sintetiza las métricas clave para los tres escenarios de densidad. Los resultados muestran un crecimiento del impacto: a medida que aumenta la densidad LEO, se incrementan de forma acelerada el índice de bloqueo promedio, la frecuencia de eventos y la degradación media del PDOP.

### 5.2.2. Conclusión de los resultados

Los resultados para la ciudad de Quito confirman que, en una ubicación ecuatorial, el impacto de las constelaciones LEO sobre la observabilidad GNSS es significativo incluso a bajas densidades. El escenario **low** ya presenta eventos puntuales de bloqueo, con  $\Delta\text{PDOP}$  superiores al 1% y  $\text{BI}_t$  por encima del 2%. El impacto se intensifica en el modo **medium**, donde los picos de  $\Delta\text{PDOP}$  alcanzan el 3.5% y los bloqueos simultáneos afectan a casi el 5% de los satélites visibles. Finalmente, el escenario **full** evidencia una saturación clara del cielo, con degradaciones del PDOP superiores al 6.5% y un  $\text{BI}_t$  que supera el 12%.

Esta progresión demuestra una correlación directa y severa entre el aumento de la densidad orbital LEO y el deterioro de la solución GNSS. Los hallazgos sugieren que Quito —y por extensión, otras zonas ecuatoriales— requerirán contramedidas tecnológicas robustas para garantizar la integridad de la navegación ante el despliegue masivo de futuras redes LEO.

## 5.3. Alice Springs

Alice Springs es una ciudad ubicada en el Territorio del Norte de Australia, situada aproximadamente a  $-23,70^\circ$  de latitud sur y  $133,88^\circ$  de longitud este. Su localización en una región subtropical del hemisferio sur permite estudiar la influencia de las constelaciones LEO sobre la observación GNSS en latitudes medias del sur, un entorno menos documentado que sus equivalentes del hemisferio norte. La ciudad alberga la *Alice Springs Space Geodesy Facility*, una estación permanente administrada por *Geoscience Australia*, utilizada para aplicaciones científicas avanzadas en geodesia espacial y seguimiento de satélites [19].

### 5.3.1. Resultados y análisis por densidad orbital

Para concluir el análisis por ubicaciones, esta sección evalúa el impacto de la constelación Starlink desde Alice Springs. Al igual que en los casos anteriores, el estudio se realiza para los tres escenarios de densidad orbital definidos.

### 5.3.1.1. Densidad baja

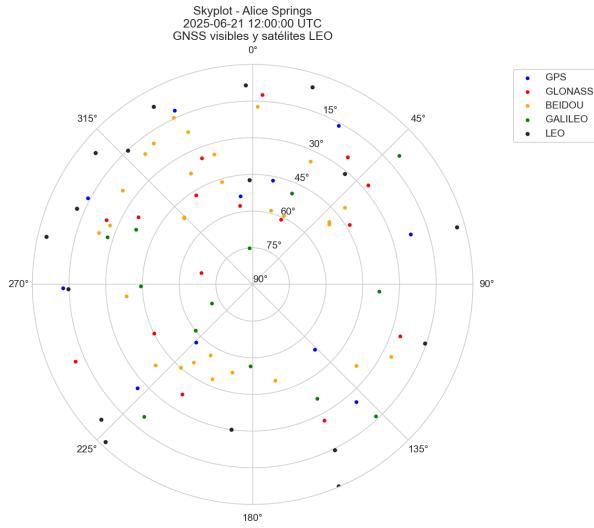


Figura 5.19: Skyplot GNSS con densidad LEO baja (Alice Springs)

El skyplot de la Figura 5.19 ilustra una distribución dispersa de satélites LEO, un comportamiento previsible dada la latitud de Alice Springs. Aunque a primera vista no se aprecian occlusiones significativas, el análisis de la degradación del PDOP confirma su existencia.

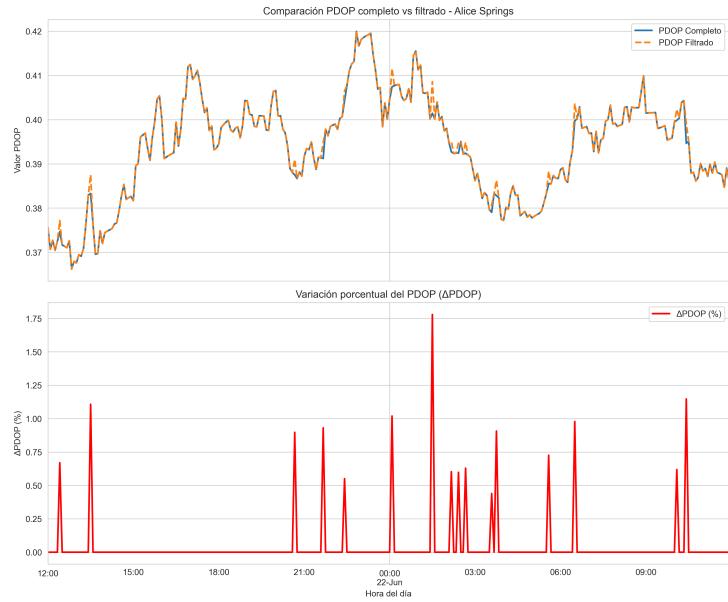


Figura 5.20: PDOP completo y filtrado con densidad LEO baja

La Figura 5.20 evidencia la divergencia entre el PDOP completo (sin bloqueo) y el filtrado (con bloqueo). Los eventos de ocultación se concentran principalmente durante la madrugada

(UTC), con picos de  $\Delta\text{PDOP}$  que alcanzan un máximo de 1.78 %. Resulta destacable que, incluso con una densidad baja, la degradación del PDOP es superior a la registrada en Quito en varios instantes del día. La Tabla 5.13 detalla los valores más significativos.

Timestamp (UTC)	$\Delta\text{PDOP} (\%)$
2025-06-21 13:30:00	1.10713
2025-06-21 20:40:00	0.89740
2025-06-21 21:40:00	0.93092
2025-06-22 00:05:00	1.01958
2025-06-22 01:30:00	1.77957
2025-06-22 06:30:00	0.97800
2025-06-22 10:25:00	1.14724

Tabla 5.13: Registros con valores  $\Delta\text{PDOP}$  superiores a 0.75 %

El siguiente gráfico nos muestra la evolución del BI para esta densidad y latitud, lo que observamos son picos que superan el 1.5 por ciento y tres de ellos llegando aproximadamente al 3 por ciento, con un pico entre las 9 y 12 de la mañana superando este valor, un 3 por ciento de satélites GNSS bloqueados de la constelación cuyos efectos son medibles incluso a baja densidad.

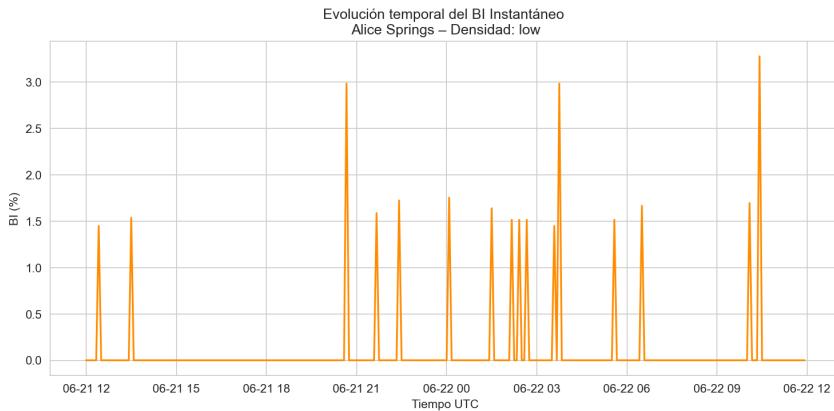


Figura 5.21: Índice de bloqueo visual (BI) – densidad LEO baja

La evolución del Índice de Bloqueo (BI), representada en la Figura 5.21, corrobora estos eventos. Se registran múltiples picos superiores al 1.5 %, y varios de ellos se aproximan al 3 %. Destaca un evento máximo que supera el 3.2 %, indicando que un porcentaje medible de los satélites GNSS visibles fueron bloqueados, incluso en este escenario de baja densidad.

Timestamp (UTC)	BI <sub>t</sub> (%)
2025-06-21 20:40:00	2.98507
2025-06-21 22:25:00	1.72414
2025-06-22 00:05:00	1.75439
2025-06-22 03:45:00	2.98507
2025-06-22 10:25:00	3.27869

Tabla 5.14: Registros con valores BI<sub>t</sub> superiores a 1.7 %

La Tabla 5.14 recoge los registros con mayor BI. El valor máximo, un 3.28 %, se produce a las 10:25 UTC. Al cruzar este dato con la Tabla 5.13, se observa que este instante coincide con un evento de bloqueo que, si bien no generó la máxima degradación de PDOP (1.15 %), sí corresponde al mayor porcentaje de satélites ocultados en la simulación.

En conclusión, para el escenario de baja densidad, el impacto en la visibilidad GNSS desde Alice Springs es notable. El número de eventos de bloqueo es superior al registrado en Tromsø y comparable, o incluso mayor, al observado en Quito bajo las mismas condiciones.

### 5.3.1.2. Densidad media

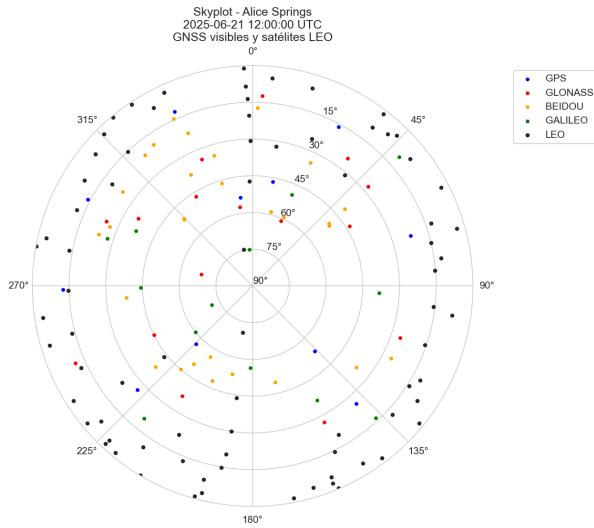


Figura 5.22: Skyplot GNSS con densidad LEO media (Alice Springs)

Con el incremento a una densidad media (2000 satélites), el skyplot de la Figura 5.22 muestra una mayor ocupación del cielo por satélites LEO, lo que anticipa un aumento de los efectos de bloqueo. El siguiente análisis cuantifica dicho comportamiento.

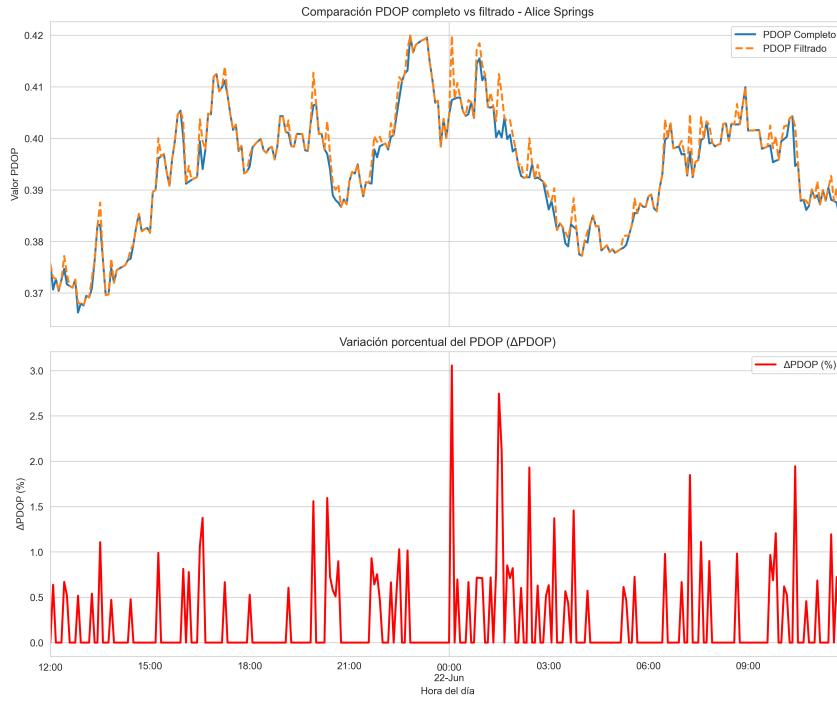


Figura 5.23: PDOP completo y filtrado con densidad LEO media

La Figura 5.23 muestra un aumento tanto en la frecuencia como en la magnitud de la degradación del PDOP. La divergencia entre ambas curvas es constante, destacando un pico pronunciado cercano a las 00:00 UTC, donde la variación porcentual supera el 3 %. La Tabla 5.15 presenta los datos específicos de los eventos más relevantes.

Timestamp (UTC)	$\Delta\text{PDOP} (\%)$
2025-06-21 19:55:00	1.55980
2025-06-21 20:20:00	1.59426
2025-06-22 00:05:00	3.05551
2025-06-22 01:30:00	2.74572
2025-06-22 01:35:00	2.10890
2025-06-22 02:25:00	1.93072
2025-06-22 07:15:00	1.84747
2025-06-22 10:25:00	1.94432

Tabla 5.15: Registros con valores  $\Delta\text{PDOP}$  superiores a 1.5 %

El análisis del BI (Figura 5.24) muestra una clara correlación con la degradación del PDOP. El máximo pico de  $\Delta\text{PDOP}$  (3.06 % a las 00:05 UTC) coincide con un BI del 5.26 %, lo que significa

que el bloqueo de algo más del 5 % de los satélites GNSS visibles provocó una degradación geométrica significativa. El resto de los picos de BI también se corresponden con momentos de alta variación porcentual del PDOP.

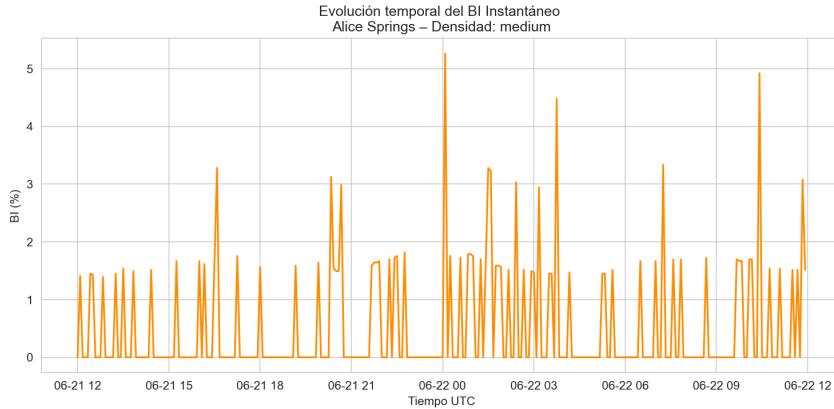


Figura 5.24: Índice de bloqueo visual (BI) – densidad LEO media

La Tabla 5.16 confirma estos hallazgos, mostrando múltiples instantes en los que el BI se aproxima o supera el 5 %.

Timestamp (UTC)	$BI_t$ (%)
2025-06-21 16:35:00	3.27869
2025-06-21 20:20:00	3.12500
2025-06-22 00:05:00	5.26316
2025-06-22 01:30:00	3.27869
2025-06-22 01:35:00	3.22581
2025-06-22 02:25:00	3.03030
2025-06-22 03:45:00	4.47761
2025-06-22 07:15:00	3.33333
2025-06-22 10:25:00	4.91803
2025-06-22 11:50:00	3.07692

Tabla 5.16: Registros con valores  $BI_t$  superiores a 3 %

Con la densidad media, la degradación se vuelve mucho más notoria. Los eventos de bloqueo se intensifican, especialmente durante la madrugada. Se establece una tendencia clara de aumento del impacto, que se espera sea aún mayor en el escenario de máxima densidad.

### 5.3.1.3. Densidad alta

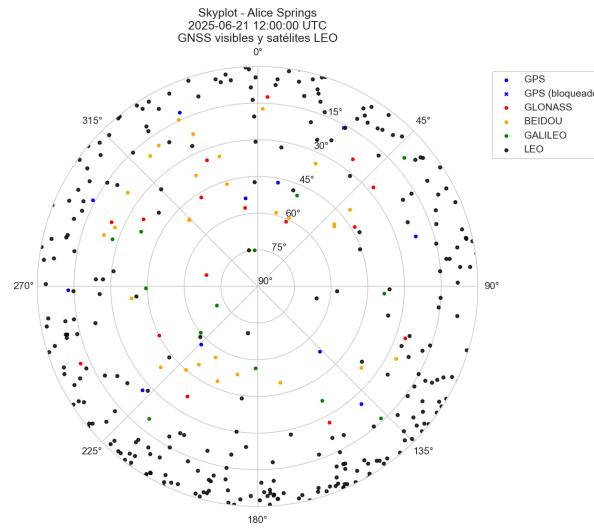


Figura 5.25: Skyplot GNSS con densidad LEO completa (Alice Springs)

El skyplot para la densidad completa presentado en la Figura 5.25, revela una saturación del cielo visible por satélites LEO, con occlusiones directas de satélites GNSS evidentes incluso en una captura estática. La evolución temporal de este impacto se analiza a continuación.

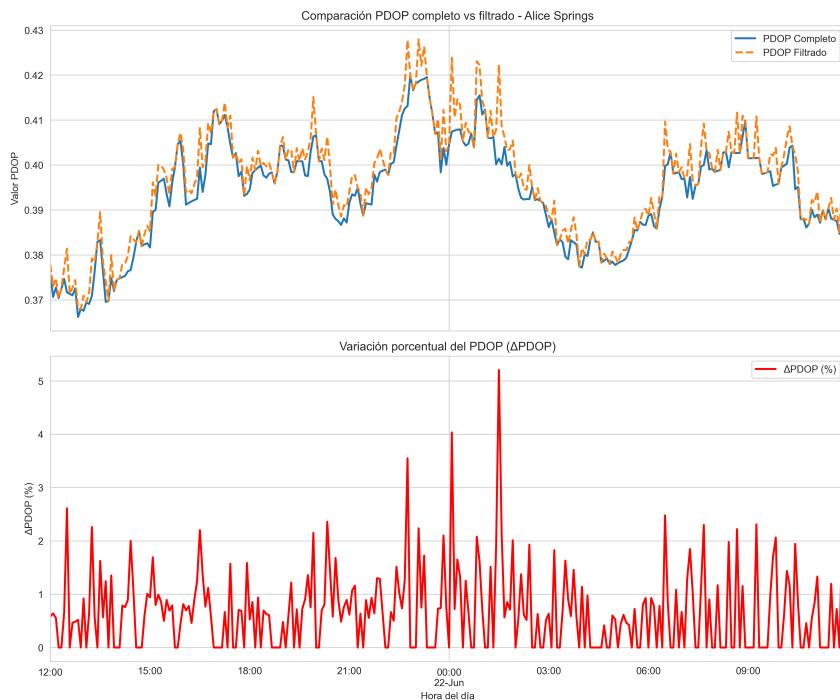


Figura 5.26: PDOP completo y filtrado con densidad LEO completa

En la Figura 5.26, las curvas de PDOP completo y filtrado se mantienen separadas durante casi todo el período de simulación, indicando un bloqueo casi constante. Los picos de  $\Delta$ PDOP son frecuentes y de gran magnitud, con tres eventos superando el 3.5 % entre las 21:00 y las 03:00 UTC. El máximo registrado alcanza una degradación superior al 5.2 %, como detalla la Tabla 5.17.

Timestamp (UTC)	$\Delta$ PDOP (%)
2025-06-21 12:30:00	2.61201
2025-06-21 22:45:00	3.55013
2025-06-22 00:05:00	4.03465
2025-06-22 01:30:00	5.20578

Tabla 5.17: Registros con valores  $\Delta$ PDOP superiores a 2.5 %

El gráfico del BI (Figura 5.27) presenta picos que superan el 7 %. Es interesante notar que no siempre existe una proporcionalidad directa entre el BI y el  $\Delta$ PDOP. Por ejemplo, el pico máximo de  $\Delta$ PDOP (5.21 % a las 01:30 UTC) se corresponde con un BI del 6.56 %, mientras que otros eventos con un BI superior (7.27 % a las 22:45 UTC) generan una degradación del PDOP menor (3.55 %). Esto evidencia que la geometría específica de los satélites bloqueados es un factor crucial.

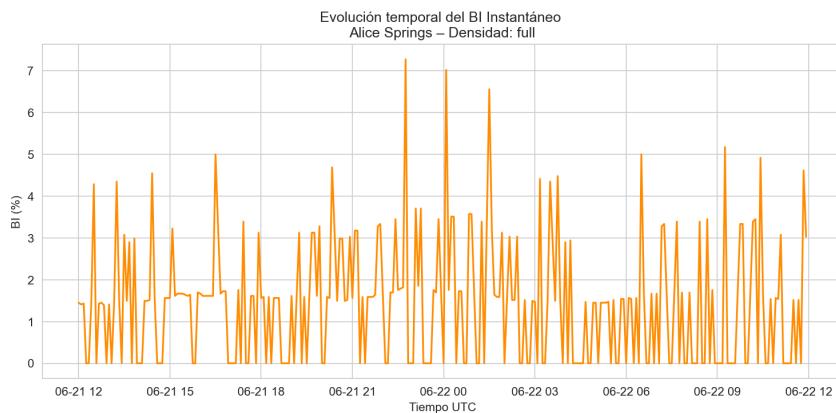


Figura 5.27: Índice de bloqueo visual (BI) – densidad LEO completa

La Tabla 5.18 cuantifica los picos más altos de BI, confirmando la alta frecuencia de eventos con más del 5 % de satélites GNSS bloqueados.

Timestamp (UTC)	BI <sub>t</sub> (%)
2025-06-21 16:30:00	5.00000
2025-06-21 22:45:00	7.27273
2025-06-22 00:05:00	7.01754
2025-06-22 01:30:00	6.55738
2025-06-22 06:30:00	5.00000
2025-06-22 09:15:00	5.17241
2025-06-22 10:25:00	4.91803

Tabla 5.18: Registros con valores BI<sub>t</sub> superiores a 4.5 %

El escenario de máxima densidad confirma que el aumento del número de satélites LEO intensifica de forma directa los valores de  $\Delta$ PDOP y BI<sub>t</sub>. A diferencia de Quito y Tromsø, donde los bloqueos se concentraban en franjas horarias, en Alice Springs el impacto es casi continuo a lo largo del día, lo que subraya la criticidad de su latitud en este tipo de análisis.

Métrica	Baja	Media	Completa
Índice de bloqueo visual promedio (BI) [%]	0.10	0.49	1.43
Frecuencia de bloqueos [eventos/h]	0.8	3.7	10.7
Satélite GNSS más afectado	Galileo 9 (4.55 %)	GLONASS (4.69 %)	GLONASS (7.04 %)
PDOP completo promedio	0.39375	0.39375	0.39375
PDOP filtrado promedio	0.39393	0.39466	0.39642
$\Delta$ PDOP promedio [%]	0.04722	0.23159	0.67603
Proporción total de observaciones bloqueadas [%]	0.11	0.49	1.42
Hora de máximo bloqueo	12:00	00:00	00:00
PDOP en la hora de máximo bloqueo	0.38	0.40	0.40

Tabla 5.19: Comparación de métricas GNSS-LEO para Alice Springs según densidad orbital

De forma análoga a los análisis previos, la Tabla 5.19 sintetiza las métricas clave para Alice Springs. Los datos muestran un crecimiento progresivo y pronunciado del BI promedio, la frecuencia de bloqueos y la degradación del PDOP a medida que aumenta la densidad orbital.

### 5.3.2. Conclusión de los resultados

El análisis para Alice Springs, una ubicación de latitud media, revela un patrón de impacto distintivo. A diferencia de las localizaciones ecuatoriales o polares, la interferencia sobre la observabilidad GNSS se manifiesta de forma casi continua a lo largo del día, incluso a densidades moderadas.

El escenario `low` ya registra una degradación notable, con picos de  $\Delta\text{PDOP}$  que superan el 1.7% y un  $\text{BI}_t$  que alcanza el 3.2%. El impacto se agrava en el modo `medium`, donde los máximos de  $\Delta\text{PDOP}$  sobrepasan el 3% y el  $\text{BI}_t$  llega a un 5.2%, consolidando la presencia de bloqueos sistemáticos. Finalmente, el escenario `full` demuestra la severidad del problema, con una degradación del PDOP que supera el 5.2% y un  $\text{BI}_t$  máximo por encima del 7.2%. En este nivel, el bloqueo deja de ser un evento puntual para convertirse en un fenómeno persistente.

Esta progresión confirma una relación directa y creciente entre la densidad de la constelación LEO y la degradación de los parámetros de posicionamiento. Los hallazgos para Alice Springs son cruciales, pues sugieren que las latitudes medias no solo enfrentan picos de degradación, sino un deterioro de base casi continuo. Esto resulta especialmente problemático para aplicaciones que exigen alta disponibilidad y continuidad.

## 5.4. Resultados por densidad

Este apartado presenta un análisis global comparativo de los resultados obtenidos en función de la densidad de la constelación Starlink utilizada. Para ello se utilizaron los datos de visibilidad y métricas de los tres observadores simulados, con el fin de estudiar tendencias generales que pudieran extrapolarse a otras localizaciones y contextos GNSS. Los resultados se agrupan en tres bloques gráficos: análisis geométrico (PDOP), degradación porcentual ( $\Delta\text{PDOP}$ ) e impacto sobre el índice de bloqueo instantáneo ( $\text{BI}_t$ ).

### 5.4.1. Boxplot de PDOP completo vs. filtrado

La elección del diagrama de caja y bigotes (*Boxplot*) responde a la necesidad de comparar de forma rigurosa las distribuciones completas de los valores de PDOP, en lugar de limitarse a sus promedios. Esta herramienta sintetiza la mediana, la dispersión de los datos y los valores atípicos, ofreciendo una panorámica estadística robusta.

De este modo, es posible visualizar directamente cómo el filtrado de satélites afecta a toda la distribución del PDOP, revelando cambios en su estabilidad y en la aparición de picos de degradación que un simple valor medio ocultaría.

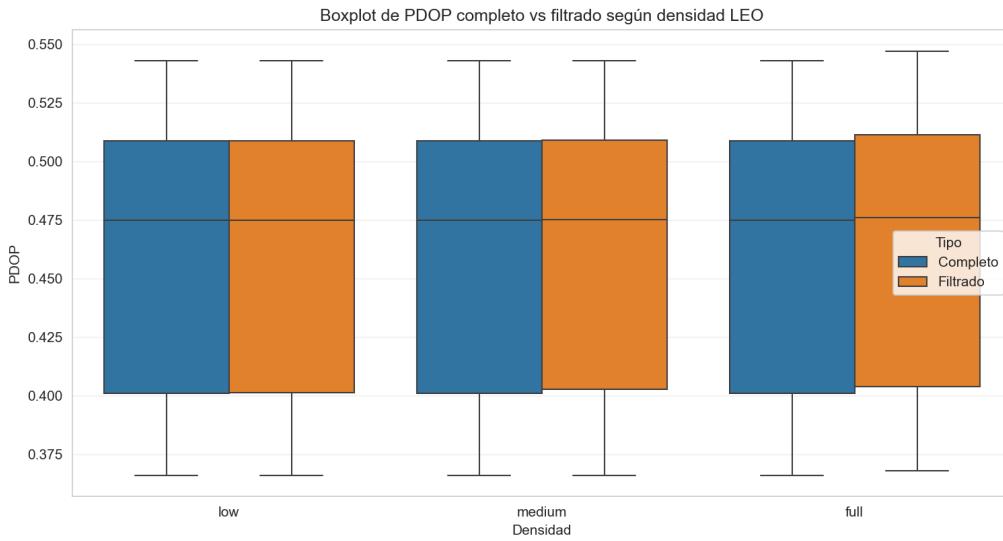


Figura 5.28: Boxplot de PDOP completo y filtrado según la densidad de la constelación LEO.

La Figura 5.28 presenta un análisis comparativo de la distribución del PDOP mediante diagramas de caja. El gráfico contrasta el PDOP completo con el PDOP filtrado para cada uno de los tres niveles de densidad de la constelación. Se observa un aumento progresivo en la mediana y la dispersión del PDOP filtrado a medida que se incrementa la densidad, lo que evidencia una degradación sistemática de la geometría de la solución debido a la oclusión.

Esta divergencia entre ambas métricas se acentúa progresivamente, evidenciando que la frecuencia y severidad de los bloqueos se intensifican con el aumento de satélites.

#### 5.4.2. Stripplot de variación porcentual de PDOP ( $\Delta$ PDOP)

Para analizar la magnitud y distribución de los eventos de degradación individuales, se emplea un gráfico de dispersión categórico (*Stripplot*). A diferencia de los gráficos de resumen como el boxplot, esta visualización muestra cada punto de datos. Su elección es deliberada, ya que permite identificar no solo la tendencia general, sino también la concentración de eventos y, fundamentalmente, la magnitud de los picos máximos de  $\Delta$ PDOP, que son cruciales para entender el impacto real del bloqueo.

De este modo, se obtiene una visión detallada de la frecuencia y severidad de la degradación, revelando patrones y valores extremos que podrían quedar ocultos en una simple agregación estadística.

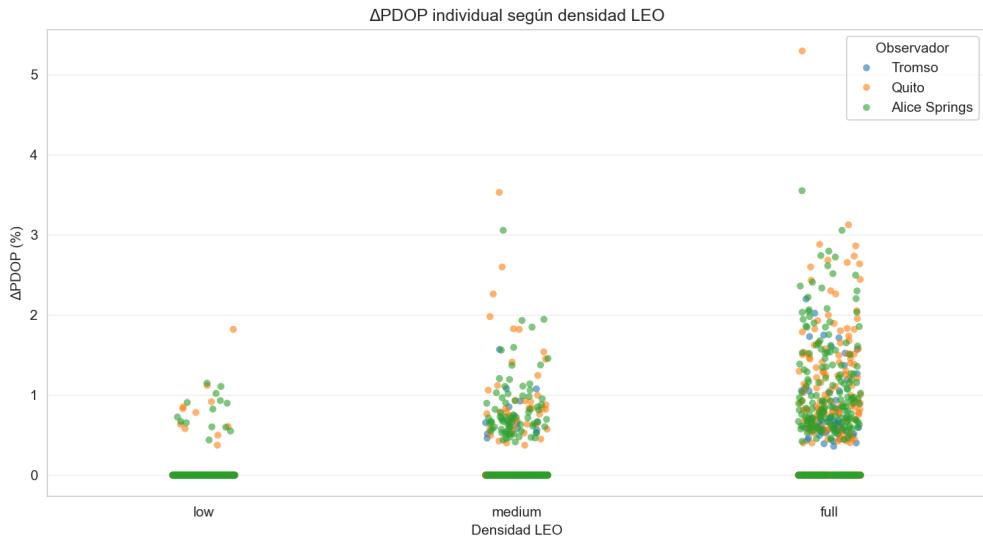


Figura 5.29: Variación porcentual de PDOP ( $\Delta\text{PDOP}$ ) por densidad y observador.

La Figura 5.29 detalla la distribución de la variación porcentual del PDOP para cada observador y nivel de densidad. El gráfico confirma visualmente varios hallazgos clave de los análisis por ciudad: en el escenario de baja densidad, el impacto es prácticamente nulo para Tromsø, mientras que Quito y Alice Springs ya registran eventos de degradación. A medida que la densidad aumenta, se incrementa tanto el número de eventos como la magnitud del  $\Delta\text{PDOP}$  en todas las estaciones.

Esta alta variabilidad y la aparición de picos extremos, especialmente notorios en el escenario de alta densidad, tienen implicaciones técnicas directas. Indican que el impacto del bloqueo no es un fenómeno homogéneo ni fácilmente predecible, lo que representa un desafío para el diseño de receptores robustos.

#### 5.4.3. Stripplot del $\text{BI}_t$ instantáneo por densidad

Para analizar la magnitud del bloqueo físico, se recurre nuevamente al stripplot. Esta visualización es particularmente efectiva para el Índice de Bloqueo instantáneo ( $\text{BI}_t$ ), ya que expone la distribución completa de los eventos de oclusión. En lugar de un valor agregado, permite identificar la frecuencia de bloqueos de baja intensidad y, de manera crítica, la ocurrencia de eventos severos donde un alto porcentaje de los satélites GNSS es ocultado simultáneamente.

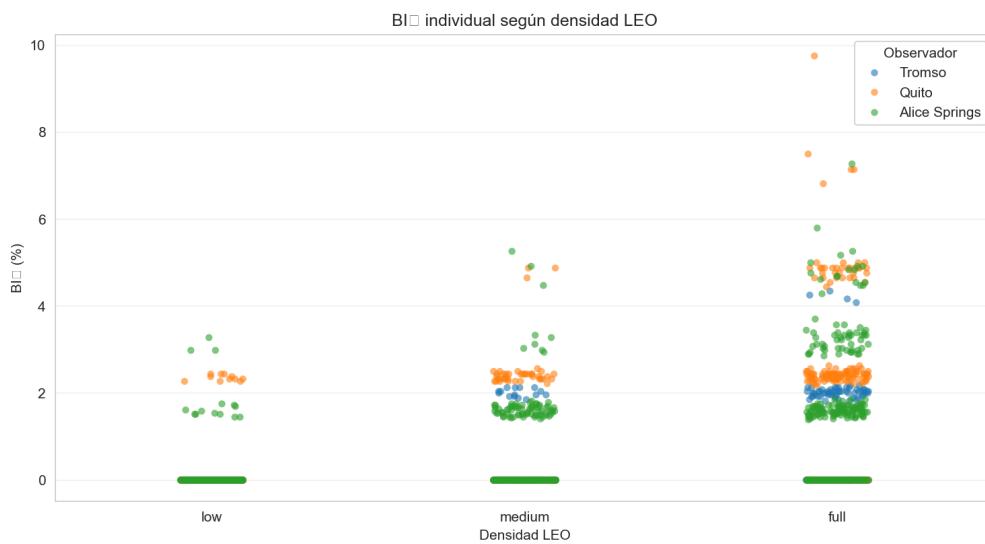


Figura 5.30: Distribución del índice de bloqueo instantáneo ( $BI_t$ ) por densidad de constelación y observador.

La Figura 5.30 ilustra la distribución del  $BI_t$ , confirmando la tendencia creciente del bloqueo con la densidad. Es notable cómo, para los tres observadores, los eventos tienden a concentrarse en valores bajos (generalmente inferiores al 2-3 %), lo que indica una base constante de oclusiones frecuentes pero de baja magnitud. Sin embargo, el gráfico también expone la existencia de picos de bloqueo severo, especialmente pronunciados en Quito y Alice Springs, donde un porcentaje significativo de la constelación visible es ocultado en determinados instantes.

#### 5.4.4. Conclusión de los análisis por densidad

En conjunto, los análisis de esta sección consolidan una conclusión fundamental: el despliegue masivo de constelaciones LEO presenta consecuencias para el ecosistema GNSS. Introducen desafíos significativos en términos de visibilidad efectiva y robustez operativa. El siguiente capítulo se dedicará a discutir estos resultados en mayor profundidad, contrastándolos con la literatura científica y extrayendo las conclusiones finales del estudio.

### 5.5. Correlaciones clave y comparación con estudios previos

En esta sección se profundiza en las relaciones estadísticas más relevantes observadas a lo largo de las simulaciones. El análisis de las métricas de PDOP,  $BI_t$  y  $\Delta PDOP$ , revela patrones consistentes que vinculan la densidad de la constelación Starlink con la degradación geométrica del

posicionamiento GNSS, permitiendo extraer conclusiones de carácter general.

### 5.5.1. Relaciones observadas

- **Densidad vs.  $\Delta\text{PDOP}$ :** Se observa una correlación positiva directa entre la densidad de satélites LEO y la magnitud de la degradación porcentual del PDOP. Al aumentar la densidad, se incrementa tanto el valor promedio como la frecuencia de picos elevados de  $\Delta\text{PDOP}$ . Este efecto es particularmente severo en ubicaciones como Quito y Alice Springs, donde la geometría GNSS base o el tránsito LEO ya presentaban desafíos, registrándose eventos de bloqueo significativos incluso en el escenario de baja densidad.
- **Densidad vs.  $\text{BI}_t$ :** El Índice de Bloqueo ( $\text{BI}_t$ ) aumenta con la densidad de la constelación. Las configuraciones **full** no solo presentan un  $\text{BI}_t$  promedio más alto, sino también picos más frecuentes y de mayor magnitud. Esto es una consecuencia directa de la mayor probabilidad estadística de oclusión: un cielo más poblado de satélites LEO incrementa inevitablemente las oportunidades de interferencia en la línea de visión hacia los satélites GNSS.
- **$\Delta\text{PDOP}$  vs.  $\text{BI}_t$ :** La relación entre estos dos indicadores es de una correlación positiva, pero no estrictamente lineal. Si bien un  $\text{BI}_t$  elevado tiende a producir un  $\Delta\text{PDOP}$  mayor, la magnitud exacta de la degradación depende críticamente de la importancia geométrica de los satélites bloqueados. Como se demostró en el análisis de Alice Springs (Figura 5.27), es posible registrar un  $\text{BI}_t$  muy alto (muchos satélites bloqueados) con un  $\Delta\text{PDOP}$  relativamente bajo. Este comportamiento se observa cuando los satélites bloqueados resultan geométricamente redundantes o representan una proporción reducida en relación con el total de satélites visibles. Por el contrario, el bloqueo de un único satélite en una posición clave puede causar un  $\Delta\text{PDOP}$  desproporcionadamente grande.

### 5.5.2. Ponderación del Índice de Bloqueo

Siguiendo la línea de lo analizado en el apartado anterior, se destaca la métrica del BI y su posible ambigüedad en ciertos escenarios. Según el número de satélites en línea de vista, un mismo valor de BI puede tener implicaciones distintas si se produce con cuatro o con cuarenta satélites visibles.

#### 5.5.2.1. Penalización proporcional al riesgo geométrico

Para corregir esta limitación, se introduce un índice ponderado de bloqueo visual ( $BIP_t$ ), el cual penaliza proporcionalmente los bloqueos ocurridos en condiciones de escasa visibilidad. Una

primera aproximación es una ponderación inversa lineal:

$$\text{BIP}_t = BI_t \cdot \left( \frac{N_{\min}}{N_{\text{visibles},t}} \right) \quad (5.1)$$

donde:

- $BI_t$  es el índice de bloqueo visual instantáneo.
- $N_{\text{visibles},t}$  es el número total de satélites GNSS visibles.
- $N_{\min}$  representa el mínimo deseable para una solución robusta; en este trabajo se fija en  $N_{\min} = 6$ , valor comúnmente adoptado para asegurar redundancia, detección de errores y precisión aceptable en entornos GNSS degradados [33, 34].

#### 5.5.2.2. Ponderación logarítmica con protección de límites

Para mejorar la sensibilidad de esta métrica y evitar penalizaciones excesivas o insignificantes, se adopta una versión logarítmica adaptativa, que introduce los siguientes ajustes:

$$\text{BIP}_t = BI_t \cdot \max \left( 0.1, 1 + \log_{10} \left( \frac{N_{\min}}{N_{\text{visibles},t} - N_{\text{bloqueados},t}} \right) \right) \quad (5.2)$$

donde:

- $N_{\text{residuales},t} = N_{\text{visibles},t} - N_{\text{bloqueados},t}$  representa los satélites funcionales restantes.
- Se impone un mínimo al factor multiplicativo mediante  $\max(0.1, \dots)$  para evitar que  $\text{BIP}_t$  sea menor que un umbral razonable incluso en contextos con decenas de satélites.
- El valor final de  $\text{BIP}_t$  se limita a 100 % como tope superior, evitando sobreestimaciones artificiales.

A continuación se presenta un ejemplo con valores configurados manualmente para comprobar la efectividad de la ecuación 5.2 en distintos contextos:

$N_{\text{vis}}$	$N_{\text{bloq}}$	$BI_t (\%)$	$N_{\text{res}}$	Factor	$\text{BIP}_t (\%)$
4	1	25.00	3	1.30	32.50
12	3	25.00	9	0.82	20.50
41	4	9.76	37	0.21	2.05
65	4	6.15	61	0.13	0.80

Tabla 5.20: Comparativa entre  $BI_t$  y  $\text{BIP}_t$  en distintos contextos geométricos

El BIP permite una interpretación más realista del impacto geométrico de los bloqueos. A igualdad de porcentaje de bloqueo, se otorga mayor severidad a los casos en los que el número de satélites visibles es bajo. Esta métrica mejora la sensibilidad del análisis ante situaciones críticas de disponibilidad GNSS y facilita la comparación entre distintos entornos y condiciones orbitales.

Se realizaron tres nuevas simulaciones en la densidad máxima para los tres observadores y se obtuvieron los siguientes datos:

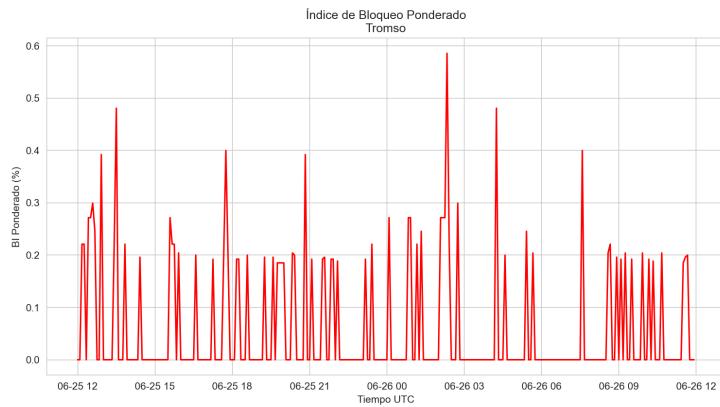


Figura 5.31: BIP - Densidad completa para Tromsø

Timestamp (UTC)	<b>BI<sub>trad</sub></b> (%)	<b>BI<sub>pond</sub></b> (%)
2025-06-25 12:25:00	2.17391	0.27161
2025-06-25 12:30:00	2.17391	0.27161
2025-06-25 12:35:00	2.22222	0.29933
2025-06-25 12:55:00	3.92157	0.39216
2025-06-25 13:30:00	4.16667	0.48081
2025-06-26 00:05:00	2.17391	0.27161
2025-06-26 00:50:00	2.17391	0.27161
2025-06-26 00:55:00	2.17391	0.27161
2025-06-26 02:05:00	2.17391	0.27161
2025-06-26 02:10:00	2.17391	0.27161
2025-06-26 02:15:00	2.17391	0.27161
2025-06-26 02:20:00	4.34783	0.58565
2025-06-26 02:45:00	2.22222	0.29933
2025-06-26 04:15:00	4.16667	0.48081
2025-06-26 07:35:00	4.00000	0.40000

Tabla 5.21: Registros de Tromsø con BI<sub>ponderado</sub> superior al 0.25

La Figura 5.31 y la Tabla 5.21 muestran los resultados para Tromsø. Debido a la alta latitud, la visibilidad de satélites es consistentemente elevada. Como resultado, el factor de ponderación raramente aplica una penalización severa. Los valores de  $BI_{trad}$  del 2-4 % se traducen en un  $BI_{pond}$  que rara vez supera el 0.5. Esto indica que, a pesar de los bloqueos, el riesgo geométrico se mantiene bajo gracias a la gran redundancia de satélites visibles.

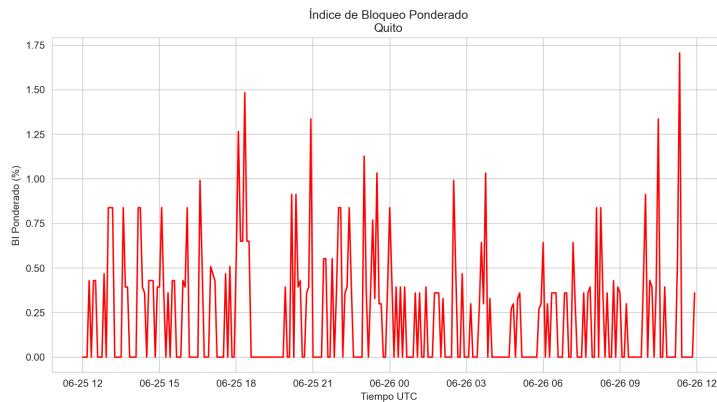


Figura 5.32: BIP - Densidad completa para Quito

Timestamp (UTC)	$BI_{trad}$	$BI_{pond}$
2025-06-25 16:35:00	5.00000	0.99184
2025-06-25 18:05:00	5.40541	1.26531
2025-06-25 18:20:00	5.71429	1.48364
2025-06-25 20:10:00	4.87805	0.91262
2025-06-25 20:20:00	4.87805	0.91262
2025-06-25 20:55:00	7.14286	1.33633
2025-06-25 23:00:00	6.81818	1.12750
2025-06-25 23:30:00	6.66667	1.03268
2025-06-26 02:30:00	5.00000	0.99184
2025-06-26 03:45:00	6.66667	1.03268
2025-06-26 10:00:00	4.87805	0.91262
2025-06-26 10:30:00	7.14286	1.33633
2025-06-26 11:20:00	7.69231	1.70653

Tabla 5.22: Registros de Quito con  $BI_{ponderado}$  superior a 0.9

En contraste, la situación en Quito (Figura 5.32 y Tabla 5.22) es marcadamente diferente. Al ser una ubicación ecuatorial, el número de satélites visibles es, en promedio, inferior al de latitudes altas. La ponderación revela el verdadero impacto de esta menor redundancia: valores de  $BI_{trad}$  de entre 4-7 % generan picos de  $BI_{pond}$  muy superiores, alcanzando valores de hasta 1.7.

Esto demuestra que la pérdida de unos pocos satélites en el ecuador en relación a su cantidad es geométricamente mucho más crítica, un hecho que el BI tradicional no logra capturar con la misma claridad.

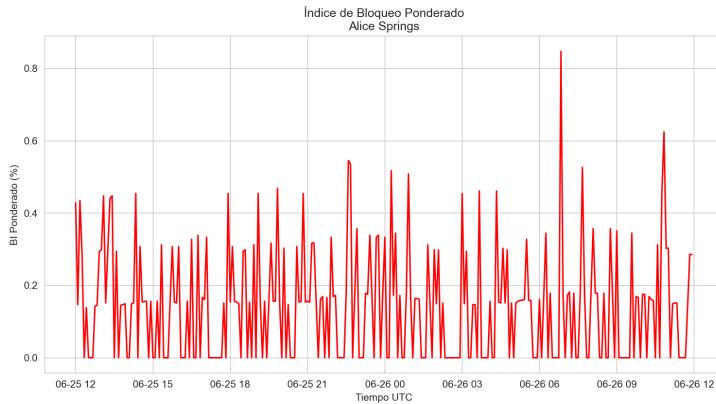


Figura 5.33: BIP - Densidad completa para Alice Springs

Finalmente, Alice Springs (Figura 5.33 y Tabla 5.23), en una latitud media, presenta un comportamiento intermedio. El  $BI_{pond}$  magnifica el impacto del bloqueo, pero de forma menos dramática que en Quito. El umbral de eventos significativos ( $BI_{pond} > 0.4$ ) se sitúa entre el de Tromsø ( $> 0.25$ ) y el de Quito ( $> 0.75$ ), confirmando la dependencia latitudinal del riesgo. El pico máximo registrado, de 0.84, evidencia un riesgo, pero sin alcanzar la severidad observada en el ecuador.

Timestamp (UTC)	$BI_{trad}$ (%)	$BI_{pond}$ (%)
2025-06-25 22:35:00	5.45455	0.54545
2025-06-25 22:40:00	5.35714	0.53571
2025-06-26 00:15:00	5.17241	0.51724
2025-06-26 00:55:00	5.08475	0.50847
2025-06-26 06:50:00	8.47458	0.84746
2025-06-26 07:40:00	5.26316	0.52632
2025-06-26 10:50:00	6.25000	0.62500

Tabla 5.23: Registros de Alice Springs con  $BI_{ponderado}$  superior al 0.5

Los resultados cuantifican cómo las regiones ecuatoriales (Quito), con menor redundancia satelital, son significativamente más vulnerables a la pérdida de unos pocos satélites que las de alta latitud (Tromsø). Por ende, al ponderar cada evento de bloqueo según el riesgo geométrico real asociado al número de satélites restantes, el BIP no solo corrige la ambigüedad inicial, sino que ofrece una evaluación más honesta y precisa del impacto, sentando una base mucho más robusta para comparar la resiliencia GNSS en distintos escenarios operativos.

### 5.5.3. Comparación con estudios previos

Los resultados obtenidos en este trabajo permiten establecer un diálogo directo con la literatura científica. A continuación, se contrastan nuestras conclusiones con las de estudios de referencia, utilizando nuestros propios datos cuantitativos para destacar las perspectivas complementarias que aporta esta investigación.

Uno de los estudios previos comentado en la Sección 2.2.1 sirve como un excelente punto de referencia para el escenario cooperativo. Sus hallazgos, resumidos en la Tabla 5.24, demuestran una reducción sustancial del PDOP (alcanzando valores tan bajos como 0.73) al integrar satélites LEO en la solución de posicionamiento [8]. Los datos de dicho estudio se presentan en la siguiente tabla:

Estación	G	G+L	C	C+L	GREC	GREC+L
THU2	2.16	1.28	2.32	1.33	0.98	0.80
ARUC	1.84	1.23	1.80	1.12	0.96	0.80
SEYG	1.73	1.37	1.22	1.05	0.78	0.73

Tabla 5.24: Valores de PDOP promedio en estaciones IGS, con y sin soporte LEO.

G: GPS, C: BDS, R: GLONASS, E: Galileo, L: LEO.

Los datos tomados del estudio comentado en la Sección 2.2.1 miden una mejora, nuestro trabajo cuantifica la degradación causada por el bloqueo de una constelación LEO no cooperativa. Partiendo de un PDOP base multi-GNSS ya muy bajo (en torno a 0.5), hemos registrado picos de degradación relativa ( $\Delta$ PDOP) que superaron el 5.2% en Alice Springs y el 6.5% en Quito en escenarios de alta densidad. Esta es la aportación clave: nuestro estudio revela el otro lado del despliegue masivo de Starlink para los receptores GNSS que no pueden integrar sus señales, un escenario realista para la vasta mayoría de dispositivos actuales.

De manera similar, los valores de PDOP ideales (entre 0.5 y 1.2) reportados en las simulaciones LEO-GNSS [12] son consistentes con nuestros valores de PDOP base. Sin embargo, nuestro análisis de oclusión demuestra la fragilidad de este escenario ideal. Una degradación relativa del 5-6% sobre un PDOP base de 0.5 es una pérdida de precisión significativa. Por tanto, nuestro trabajo no contradice a al estudio de base utilizado en 2.2.4, sino que le añade una capa de realismo al cuantificar el impacto negativo de una interferencia física que sus modelos no consideran.

Finalmente, nuestros hallazgos sobre la relación entre geometría y precisión se alinean con las tendencias de García-Fernández [10], pero vamos un paso más allá. Nuestro estudio identifica y cuantifica una causa física específica para dicha degradación: el bloqueo visual directo, medido a través del Índice de Bloqueo ( $BI_t$ ). En nuestras simulaciones, este índice alcanzó picos superiores

al 7.2% en Alice Springs y al 12% en Quito, demostrando que una fracción considerable de la constelación GNSS puede ser ocultada simultáneamente. Esto proporciona una métrica tangible para un fenómeno que hasta ahora se discutía de forma más teórica.

## 5.6. Discusión final

Este estudio ha demostrado de manera cuantitativa que el bloqueo físico ejercido por la constelación Starlink constituye una fuente significativa de degradación para la observabilidad GNSS. Lejos de ser un efecto trivial, hemos constatado una relación directa y positiva entre el aumento de la densidad orbital y el deterioro de la solución, medido tanto a través del Índice de Bloqueo ( $BI_t$ ) como de la degradación porcentual del PDOP ( $\Delta PDOP$ ). En los escenarios más densos, se han registrado picos de  $BI_t$  superiores al 12% (Quito) y de  $\Delta PDOP$  por encima del 6.5% (Quito), confirmando que incluso en configuraciones geométricas multi-GNSS muy robustas, la pérdida de precisión es tangible y severa.

El análisis ha revelado, además, que el impacto no es homogéneo. La degradación varía notablemente con la latitud, manifestándose en picos agudos en ubicaciones ecuatoriales y como un fenómeno más persistente y constante en latitudes medias como Alice Springs. Se ha identificado una de las conclusiones más relevantes de este trabajo: la relación entre el bloqueo físico ( $BI_t$ ) y la degradación geométrica ( $\Delta PDOP$ ) no es estrictamente lineal, razón por la que definimos el BIP. La importancia geométrica de los satélites ocultados ha demostrado ser un factor crítico, así como la proporción visible de estos en los instantes analizados. Este hallazgo matiza de forma crucial la visión mayoritaria de la literatura, centrada en los beneficios de un paradigma LEO-GNSS cooperativo, y pone de relieve el riesgo inherente a una coexistencia no coordinada.

Con los hallazgos y correlaciones ya discutidos, este estudio ha sentado las bases para la evaluación final. El capítulo de conclusiones que sigue se dedicará, por tanto, a sintetizar el recorrido completo de esta investigación, desde el planteamiento del problema hasta la cuantificación de sus efectos. Más importante aún, se utilizará la tendencia proporcional observada entre la densidad y la degradación para proyectar las consecuencias de la futura expansión de Starlink más allá de los 6.000 satélites, delineando así las implicaciones a largo plazo de este desafío para la navegación global.



## Capítulo 6

# Conclusiones y Líneas Futuras

El presente trabajo se originó a partir de una brecha identificada en la literatura técnica: la mayoría de los estudios se centran en los beneficios de la integración cooperativa de constelaciones LEO en el posicionamiento GNSS, mientras que el efecto adverso de su interferencia física ha sido escasamente cuantificado. El objetivo principal fue, por tanto, el desarrollo de una metodología sistemática para modelar y medir el impacto del bloqueo visual ejercido por la constelación Starlink.

Para ello, se implementó un simulador en Python basado en la librería Skyfield, que procesa datos TLE reales para propagar las órbitas GNSS y LEO. El análisis se fundamentó en tres ubicaciones geográficas clave y tres escenarios de densidad orbital, permitiendo evaluar el fenómeno bajo un amplio rango de condiciones latitudinales y de congestión satelital. La degradación se cuantificó a través de dos métricas principales: el Índice de Bloqueo instantáneo ( $BI_t$ ) y la variación porcentual del PDOP ( $\Delta PDOP$ ), transformando una hipótesis en un problema medible.

### 6.1. Conclusiones Principales

El análisis cuantitativo de los datos generados por las simulaciones permite extraer las siguientes conclusiones fundamentales:

1. **Se confirma una correlación directa entre densidad y degradación.** Los resultados demuestran de forma inequívoca que un aumento en el número de satélites Starlink se traduce en un incremento de los eventos de bloqueo. Este patrón fue consistente en las tres ubicaciones estudiadas, validando la hipótesis central del trabajo.
2. **El impacto del bloqueo es cuantitativamente significativo.** La degradación no es

marginal. Se registraron picos de  $BI_t$  que superaron el 12% en Quito, indicando que una fracción considerable de la constelación GNSS visible puede ser ocultada simultáneamente. Esto se tradujo en una degradación geométrica severa, con picos de  $\Delta PDOP$  por encima del 6.5 %, incluso partiendo de un escenario multi-GNSS con un PDOP base muy robusto.

3. **El efecto del bloqueo varía con la latitud.** El impacto no es globalmente uniforme. Las ubicaciones ecuatoriales como Quito mostraron una mayor propensión a picos de degradación agudos y de alta magnitud. En contraste, en latitudes medias como Alice Springs, el fenómeno se manifestó como un deterioro más persistente y constante. Tromsø, en alta latitud, demostró ser la más resiliente a bajas densidades, pero su vulnerabilidad creció notablemente en el escenario **full**.
4. **La degradación geométrica no es directamente proporcional al número de satélites bloqueados.** Se ha identificado que la relación entre el  $BI_t$  y el  $\Delta PDOP$  es de una correlación positiva, pero no estrictamente lineal. La importancia geométrica de los satélites ocultados es un factor más determinante que su cantidad. El bloqueo de múltiples satélites redundantes puede no afectar significativamente la solución, mientras que la ocultación de un único satélite en una posición geométrica clave puede causar una degradación desproporcionada.
5. **El índice de bloqueo visual debe contextualizarse según la disponibilidad efectiva.** Para mejorar la sensibilidad de la métrica  $BI_t$  ante situaciones de baja redundancia geométrica, se introduce un índice ponderado de bloqueo visual ( $BIP_t$ ). Esta ponderación permite distinguir de forma más precisa entre escenarios críticos (pocos satélites restantes) y triviales (bloqueos con amplia disponibilidad residual). En los casos analizados, se verifica que el  $BIP_t$  ofrece una representación más realista del riesgo geométrico que el índice tradicional, siendo especialmente útil para comparar entornos con distinta densidad satelital o condiciones extremas de bloqueo.

## 6.2. Proyección a Futuro

Los resultados de este estudio representan una instantánea del impacto con una constelación de aproximadamente 6.000 satélites. Sin embargo, suponiendo que los planes de expansión de Starlink contemplan fases con 12.000 satélites o más, es fundamental proyectar nuestros hallazgos para anticipar la escala del desafío. Para ello, se realiza una extrapolación lineal basada en la tendencia de degradación observada entre los escenarios de densidad media y alta.

El método consiste en calcular primero la tasa de crecimiento de la degradación por satélite

( $R_{\text{métrica}}$ ) a partir de los datos existentes:

$$R_{\text{métrica}} = \frac{\text{Métrica}_{\text{alta}} - \text{Métrica}_{\text{media}}}{N_{\text{sats, alta}} - N_{\text{sats, media}}} \quad (6.1)$$

Luego, se utiliza esta tasa para proyectar el valor de la métrica para un número futuro de satélites:

$$\text{Proyección} = \text{Métrica}_{\text{alta}} + R_{\text{métrica}} \times (N_{\text{sats, proyectado}} - N_{\text{sats, alta}}) \quad (6.2)$$

Aplicando este método a los datos de Alice Springs, resumidos en la Tabla 5.19, y considerando las densidades media ( $\approx 2,000$  sats) y alta ( $\approx 6,000$  sats), se obtiene lo siguiente para una constelación proyectada de 12,000 satélites:

- **Proyección del Índice de Bloqueo (BI) promedio:**

- Datos de entrada:  $\text{BI}_{\text{alta}} = 1,43\%$ ;  $\text{BI}_{\text{media}} = 0,49\%$ .
- Tasa de crecimiento:  $R_{\text{BI}} = \frac{1,43-0,49}{6000-2000} = \frac{0,94}{4000} = 0,000235\%/\text{satélite}$ .
- Valor proyectado:  $\text{Proyección}_{\text{BI}} = 1,43 + 0,000235 \times (12000 - 6000) \approx 2,84\%$ .

- **Proyección de la degradación del PDOP ( $\Delta\text{PDOP}$ ) promedio:**

- Datos de entrada:  $\Delta\text{PDOP}_{\text{alta}} = 0,676\%$ ;  $\Delta\text{PDOP}_{\text{media}} = 0,232\%$ .
- Tasa de crecimiento:  $R_{\Delta\text{PDOP}} = \frac{0,676-0,232}{6000-2000} = \frac{0,444}{4000} = 0,000111\%/\text{satélite}$ .
- Valor proyectado:  $\text{Proyección}_{\Delta\text{PDOP}} = 0,676 + 0,000111 \times (12000 - 6000) \approx 1,34\%$ .

Esta extrapolación sugiere que al duplicar el número de satélites de 6,000 a 12,000, tanto el Índice de Bloqueo promedio como la degradación media del PDOP también se duplicarían aproximadamente. Es crucial subrayar que esta es una estimación de primer orden que asume una relación lineal, sin tener en cuenta la ponderación que se debería realizar. Sin embargo, incluso esta proyección conservadora dibuja un panorama preocupante: un futuro en el que el bloqueo dejaría de ser un evento intermitente para convertirse en una característica persistente del entorno de navegación, amenazando la integridad de aplicaciones críticas.

## 6.3. Limitaciones del Estudio y Futuras Líneas de Investigación

La validez de los resultados de este trabajo debe enmarcarse dentro de ciertas limitaciones metodológicas, las cuales, a su vez, definen las líneas de investigación futura más prometedoras. El modelo de bloqueo empleado fue de naturaleza puramente angular ( $< 1^\circ$ ), sin considerar el tamaño físico de los satélites ni los efectos de la difracción de la señal. Asimismo, la simulación

operó bajo el modelo *ionosphere-free*, excluyendo deliberadamente efectos atmosféricos complejos como la escintilación, que podrían agravar la degradación en escenarios reales. Tampoco se incluyeron modelos de obstrucción terrestre, cuyo impacto se sumaría al del bloqueo orbital.

Estas limitaciones definen de forma natural un camino claro para la investigación futura. En primer lugar, resulta fundamental la validación experimental de los resultados mediante campañas de medición en las ubicaciones estudiadas. En paralelo, el modelo de simulación podría enriquecerse desarrollando modelos de bloqueo de mayor fidelidad que integren parámetros físicos y analizando el impacto acumulado al combinar el bloqueo orbital con modelos de obstrucción terrestre. Finalmente, desde una perspectiva de sistema, emergen dos áreas críticas: el estudio de la resiliencia de los algoritmos de integridad (RAIM) frente a estos nuevos patrones de fallo, y el diseño de estrategias de mitigación proactivas capaces de predecir occlusiones y optimizar la selección de satélites en tiempo real.

## 6.4. Conclusión Final

En conclusión, el presente trabajo ha cumplido con éxito su objetivo principal: demostrar y cuantificar el impacto del bloqueo ejercido por la constelación Starlink sobre la observabilidad GNSS. Mediante un análisis sistemático del Índice de Bloqueo y  $\Delta\text{PDOP}$ , se ha constatado que este fenómeno no es teórico, sino una fuente real de degradación, cuya severidad se incrementa de forma directa con la densidad de la constelación.

La contribución fundamental de este estudio reside, por tanto, en haber aportado métricas cuantitativas a una faceta poco explorada de la coexistencia GNSS-LEO. Los resultados ponen de manifiesto que el despliegue de megaconstelaciones no es inherentemente benigno para la infraestructura GNSS y que la integridad de la navegación global en la próxima década dependerá críticamente del desarrollo proactivo de tecnologías y estrategias de mitigación que aborden este nuevo y creciente desafío.

# Apéndice A

## Código fuente del simulador

### A.1. Clase principal: GNSSLEOSimulator

```
1  class GNSSLEOSimulator:
2      """Simulador de visibilidad GNSS y LEO con análisis de bloqueo y métricas"""
3
4      def __init__(self):
5          """Inicializa el simulador con objetos básicos"""
6          self.ts = load.timescale() # Escala de tiempo para cálculos astronómicos
7          self.eph = load('de421.bsp') # Efemérides planetarias
8          self.gnss_sats = {} # Diccionario para satélites GNSS por constelación
9          self.leo_sats = {} # Lista para satélites LEO
10         self.observers = {} # Puntos de observación terrestres
11         self.results = {} # Almacenamiento de resultados de simulaciones
12
13     def load_gnss_constellations(self):
14         """
15             Carga datos de constelaciones GNSS desde Celestrak.
16
17             Procesa:
18             - GPS (NAVSTAR)
19             - Galileo (GSAT)
20             - GLONASS (COSMOS)
21             - BeiDou (COMPASS)
22         """
23         url = 'https://celesttrak.org/NORAD/elements/gnss.txt'
24         satellites = load.tle_file(url) # Descarga y parsea TLEs
25
26         for sat in satellites:
27             name = sat.name.upper()
28             # Clasificación por nombre/identificador
29             if 'GPS' in name or 'NAVSTAR' in name:
```

```
30         constel = 'GPS'
31     elif 'GALILEO' in name or 'GSAT' in name:
32         constel = 'GALILEO'
33     elif 'GLONASS' in name or 'COSMOS' in name or 'R' in name.split('-')[0]:
34         constel = 'GLONASS'
35     elif 'BEIDOU' in name or 'COMPASS' in name or 'BD' in name.split('-')[0]:
36         constel = 'BEIDOU'
37     else:
38         continue # Ignorar otras constelaciones
39
40     if constel not in self.gnss_sats:
41         self.gnss_sats[constel] = []
42     self.gnss_sats[constel].append(sat)
43
44     # Resumen de carga
45     loaded_constels = ', '.join([f"{k} ({len(v)} sats)" for k, v in self.gnss_sats.
46     items()])
47     print(f"GNSS cargados: {loaded_constels}")
48
49     def load_leo_constellation(self, density='full'):
50         """
51             Carga constelación LEO (Starlink) con control de densidad.
52
53             Args:
54                 density: 'low' (500 sat), 'medium' (2000 sat), 'full' (todos)
55
56             Filtra por:
57                 - Altitud: 400-650 km
58                 - Inclinación: 40-100°
59
60             url = 'https://celestak.org/NORAD/elements/gp.php?GROUP=starlink&FORMAT=tle'
61
62             try:
63                 # Descarga directa de TLEs
64                 response = urlopen(url)
65                 tle_text = response.read().decode('utf-8')
66                 lines = tle_text.strip().split('\n')
67             except Exception as e:
68                 print(f"Error al descargar TLEs de Starlink: {e}")
69             return
70
71             filtered_leo = []
72             for i in range(0, len(lines), 3):
73                 if i+2 >= len(lines):
74                     break
75
76                 name = lines[i].strip()
```



```

120     print(f"Observador añadido: {name} ({lat}°, {lon}°)")
121
122     def simulate_visibility(self, observer_name, date, duration_hours=24, time_step_min
123     =5):
124         """
125             Simula visibilidad de satélites para un observador.
126
127             Args:
128                 observer_name: Nombre del observador registrado
129                 date: Fecha/hora inicial
130                 duration_hours: Duración en horas (default 24)
131                 time_step_min: Intervalo en minutos (default 5)
132
133             Returns:
134                 Diccionario con resultados de visibilidad y bloqueos
135         """
136
137         observer = self.observers[observer_name]
138         start_time = self.ts.utc(date.year, date.month, date.day, date.hour, date.minute,
139         date.second)
140         end_time = start_time + duration_hours/24
141
142         # Generación de pasos de tiempo
143         times = []
144         current = start_time
145         while current.tt < end_time.tt:
146             times.append(current)
147             current += time_step_min / (24*60)
148
149         # Estructura para resultados
150         results = {
151             'times': times,
152             'gnss_visible': defaultdict(list), # GNSS visibles por constelación
153             'leo_positions': [], # Posiciones LEO por tiempo
154             'blocked_gnss': defaultdict(list) # GNSS bloqueados
155         }
156
157         BLOCK_THRESHOLD = np.radians(1.0) # Umbral angular de bloqueo (1°)
158
159         # Simulación por paso de tiempo (con barra de progreso)
160         for t in tqdm(times, desc=f"Simulando {observer_name}"):
161             # Detección de GNSS visibles (>10° elevación)
162             visible_gnss = []
163             for constel, sats in self.gnss_sats.items():
164                 for sat in sats:
165                     pos = (sat - observer).at(t)
166                     alt, az, _ = pos.altaz()
167                     if alt.degrees > 10:

```

```

165         ra_gnss, dec_gnss, _ = pos.radec()
166         visible_gnss.append({
167             'sat': sat,
168             'name': sat.name,
169             'constel': constel,
170             'ra': ra_gnss.radians,
171             'dec': dec_gnss.radians,
172             'alt': alt.degrees,
173             'az': az.degrees
174         })
175
176     # Cálculo de posiciones LEO
177     leo_positions = []
178     for sat in self.leo_sats:
179         pos = (sat - observer).at(t)
180         alt, az, _ = pos.altaz()
181         ra_leo, dec_leo, _ = pos.radec()
182         leo_positions.append({
183             'ra': ra_leo.radians,
184             'dec': dec_leo.radians,
185             'alt': alt.degrees,
186             'az': az.degrees
187         })
188
189     # Detección de bloqueos usando KDTree para eficiencia
190     blocked = set()
191     if visible_gnss and leo_positions:
192         # Preparación de datos para búsqueda espacial
193         gnss_array = np.array([(g['ra'], g['dec']) for g in visible_gnss])
194         leo_array = np.array([(l['ra'], l['dec']) for l in leo_positions])
195
196         tree = KDTree(leo_array)
197         distances, indices = tree.query(gnss_array, k=1)
198
199         # Identificación de GNSS bloqueados
200         for i, dist in enumerate(distances):
201             if dist < BLOCK_THRESHOLD:
202                 blocked.add(visible_gnss[i]['name'])
203                 results['blocked_gnss'][visible_gnss[i]['constel']].append(
204                     visible_gnss[i]['name'])
205
206         # Almacenamiento de resultados
207         for g in visible_gnss:
208             results['gnss_visible'][g['constel']].append({
209                 'time': t,
210                 'name': g['name'],
211                 'alt': g['alt'],
212             })

```

```
211         'az': g['az'],
212         'blocked': g['name'] in blocked
213     })
214
215     results['leo_positions'].append({
216         'time': t,
217         'positions': leo_positions
218     })
219
220     self.results[observer_name] = results
221
222     return results
223
224     def calculate_metrics(self, observer_name):
225         """
226             Calcula métricas estadísticas de bloqueo GNSS.
227
228             Args:
229                 observer_name: Nombre del observador
230
231             Returns:
232                 Diccionario con métricas detalladas
233
234         """
235
236         if observer_name not in self.results:
237             raise ValueError(f"No hay datos para {observer_name}")
238
239         data = self.results[observer_name]
240         metrics = {
241             'total_gnss_obs': 0, # Observaciones GNSS totales
242             'total_blocked_obs': 0, # Observaciones bloqueadas
243             'constel_metrics': {}, # Métricas por constelación
244             'satellite_metrics': {}, # Métricas por satélite
245             'hourly_stats': defaultdict(dict), # Estadísticas horarias
246             'bi_instantaneo': [] # Índice de bloqueo instantáneo
247         }
248
249         # Cálculo por constelación
250         for constel, observations in data['gnss_visible'].items():
251             total = len(observations)
252             blocked = sum(1 for obs in observations if obs['blocked'])
253
254             metrics['constel_metrics'][constel] = {
255                 'total_obs': total,
256                 'blocked_obs': blocked,
257                 'block_percent': (blocked / total) * 100 if total > 0 else 0
258             }
259
260             metrics['total_gnss_obs'] += total
```

```

258     metrics['total_blocked_obs'] += blocked
259
260     # Cálculo por satélite individual
261     sat_counts = defaultdict(lambda: {'total': 0, 'blocked': 0})
262     for obs in observations:
263         sat_counts[obs['name']]['total'] += 1
264         if obs['blocked']:
265             sat_counts[obs['name']]['blocked'] += 1
266
267         for sat, counts in sat_counts.items():
268             metrics['satellite_metrics'][sat] = {
269                 'block_percent': (counts['blocked'] / counts['total']) * 100 if
270             counts['total'] > 0 else 0,
271                 'constel': constel
272             }
273
274     # Cálculo global de bloqueo
275     if metrics['total_gnss_obs'] > 0:
276         global_block_ratio = (metrics['total_blocked_obs'] / metrics['total_gnss_obs'])
277     else:
278         global_block_ratio = 0
279     metrics['global_block_percent'] = global_block_ratio
280
281     # Cálculo horario
282     for i, t in enumerate(data['times']):
283         hour = t.utc_datetime().hour
284         visible = len([obs for constel in data['gnss_visible'].values() for obs in
285         constel if obs['time'] == t])
286         blocked = len([obs for constel in data['gnss_visible'].values() for obs in
287         constel if obs['time'] == t and obs['blocked']])
288
289         if visible > 0:
290             bi_t = (blocked / visible) * 100
291             metrics['hourly_stats'][hour]['block_percent'] = bi_t
292             metrics['hourly_stats'][hour]['visible_sats'] = visible
293             metrics['hourly_stats'][hour]['blocked_sats'] = blocked
294             metrics['bi_instantaneo'].append((t.utc_datetime(), bi_t))
295         else:
296             metrics['bi_instantaneo'].append((t.utc_datetime(), 0))
297
298     # Cálculo preciso de BI promedio
299     bi_values = []
300     for t in data['times']:
301         visible = sum(1 for constel in data['gnss_visible'].values() for obs in
302         constel if obs['time'] == t)

```

```

299         blocked = sum(1 for constel in data['gnss_visible'].values() for obs in
300                         constel if obs['time'] == t and obs['blocked'])
301
302         if visible > 0:
303             bi_values.append(blocked / visible)
304
305     metrics['bi_promedio'] = np.mean(bi_values) * 100 if bi_values else 0
306
307     # Identificación del satélite más afectado
308     if metrics['satellite_metrics']:
309         most_affected = max(metrics['satellite_metrics'].items(),
310                               key=lambda x: x[1]['block_percent'])
311         metrics['most_affected_sat'] = {
312             'name': most_affected[0],
313             'block_percent': most_affected[1]['block_percent'],
314             'constel': most_affected[1]['constel']
315         }
316
317     return metrics
318
319
320     def calculate_frequency_metrics(self, observer_name):
321         """
322             Calcula frecuencia de eventos de bloqueo.
323
324             Returns:
325                 - Total de bloqueos
326                 - Horas de simulación
327                 - Frecuencia (bloqueos/hora)
328
329         """
330
331         data = self.results[observer_name]
332         total_blocks = sum(len(v) for v in data['blocked_gnss'].values())
333         sim_hours = (data['times'][-1] - data['times'][0]) * 24
334
335         return {
336             'total_blocks': total_blocks,
337             'simulation_hours': sim_hours,
338             'block_frequency': total_blocks / sim_hours
339         }
340
341
342     def calculate_pdop(self, observer_name, time_index):
343         """
344             Calcula el Position Dilution of Precision (PDOP) para un instante.
345
346             Args:
347                 observer_name: Nombre del observador
348                 time_index: Índice del tiempo en la simulación

```

```

345
346     Returns:
347         Valor PDOP (infinito si <4 satélites visibles)
348     """
349
350     if observer_name not in self.results:
351         raise ValueError(f"No hay datos para {observer_name}")
352
353     data = self.results[observer_name]
354     if time_index >= len(data['times']):
355         raise ValueError("Índice de tiempo fuera de rango")
356
357     t = data['times'][time_index]
358
359     # Obtención de GNSS visibles no bloqueados
360     visible_sats = []
361     for constel, observations in data['gnss_visible'].items():
362         for obs in observations:
363             if obs['time'] == t and not obs['blocked']:
364                 obs['constel'] = constel
365                 visible_sats.append(obs)
366
367     if len(visible_sats) < 4:
368         return float('inf') # Mínimo 4 satélites para PDOP
369
370     # Cálculo de matriz de geometría
371     observer = self.observers[observer_name]
372     obs_pos = observer.at(t).position.km
373
374     H = []
375     for sat in visible_sats:
376         # Búsqueda del objeto satélite correspondiente
377         for constel_sats in self.gnss_sats[sat['constel']]:
378             if constel_sats.name == sat['name']:
379                 sat_pos = (constel_sats - observer).at(t).position.km
380                 break
381
382         # Vector unitario receptor-satélite
383         rho = sat_pos - obs_pos
384         rho_norm = np.linalg.norm(rho)
385         u = rho / rho_norm
386         H.append(u)
387
388     H = np.array(H)
389     try:
390         Q = np.linalg.inv(H.T @ H) # Matriz de covarianza
391         pdop = np.sqrt(Q[0,0] + Q[1,1] + Q[2,2]) # Cálculo PDOP
392     except:

```

```
392         pdop = float('inf')
393
394     return pdop
395
396
397     def generate_report(self, observer_name, pdop_data=None):
398         """
399             Genera un reporte textual completo con métricas clave.
400
401         Args:
402             observer_name: Nombre del observador
403             pdop_data: Datos PDOP precalculados (opcional)
404
405         Returns:
406             report: Texto del reporte
407             pdop_data: Datos PDOP utilizados
408         """
409
410     metrics = self.calculate_metrics(observer_name)
411
412     if pdop_data is None:
413         pdop_analyzer = PDOPAnalyzer(self)
414         pdop_data = pdop_analyzer.calculate_pdop_variation(observer_name)
415
416     freq_metrics = self.calculate_frequency_metrics(observer_name)
417
418     # Construcción del reporte
419     report = (
420         f"Desde el punto de observación de la ciudad {observer_name}, "
421         f"el Índice de Bloqueo Visual Promedio (BI) fue del {metrics['bi_promedio']:.2f}%. "
422         f"El satélite GNSS más afectado fue {metrics['most_affected_sat']['name']} "
423         f"({metrics['most_affected_sat']['constel']}), con un "
424         f"{metrics['most_affected_sat']['block_percent']:.2f}% de tiempo bloqueado "
425         f"individualmente.\n\n"
426         "Estadísticas por constelación:\n"
427     )
428
429     for constel, stats in metrics['constel_metrics'].items():
430         report += (
431             f"- {constel}: {stats['block_percent']:.2f}% de bloqueo "
432             f"({stats['blocked_obs']}/{stats['total_obs']}) observaciones\n"
433         )
434
435     # Análisis de PDOP en momentos clave
436     times = self.results[observer_name]['times']
437     hourly_block = {h: s['block_percent'] for h, s in metrics['hourly_stats'].items()}
438 }
```

```

436     max_hour = max(hourly_block.items(), key=lambda x: x[1])[0]
437     min_hour = min(hourly_block.items(), key=lambda x: x[1])[0]
438     avg_hour = sorted(hourly_block.items(), key=lambda x: abs(x[1] - metrics['
439         bi_promedio']))[0][0]
440
440     max_idx = next(i for i, t in enumerate(times) if t.utc_datetime().hour ==
441         max_hour)
441     min_idx = next(i for i, t in enumerate(times) if t.utc_datetime().hour ==
442         min_hour)
442     avg_idx = next(i for i, t in enumerate(times) if t.utc_datetime().hour ==
443         avg_hour)
443
444     pdop_max = self.calculate_pdop(observer_name, max_idx)
445     pdop_min = self.calculate_pdop(observer_name, min_idx)
446     pdop_avg = self.calculate_pdop(observer_name, avg_idx)
447
448     report += (
449         "\nAnálisis de PDOP en momentos clave:\n"
450         f"- Hora de máximo bloqueo ({max_hour:02d}:00): PDOP completo = {pdop_max:.2f
451 }\n"
452         f"- Hora de mínimo bloqueo ({min_hour:02d}:00): PDOP completo = {pdop_min:.2f
453 }\n"
454         f"- Hora de bloqueo promedio ({avg_hour:02d}:00): PDOP completo = {pdop_avg
455 :.2f}\n"
456     )
457
458     # Métricas agregadas
459     pdop_promedio = np.nanmean(pdop_data['pdop_full'])
460     pdop_filtrado_promedio = np.nanmean(pdop_data['pdop_filtered'])
461     delta_promedio = np.nanmean(pdop_data['delta_pdop_percent'])
462
463     report += (
464         "\nMÉTRICAS AVANZADAS:\n"
465         f"- Frecuencia de bloqueos: {freq_metrics['block_frequency']:.1f} eventos/
466         hora\n"
467         f"- PDOP completo promedio: {pdop_promedio:.5f}\n"
468         f"- PDOP filtrado promedio: {pdop_filtrado_promedio:.5f}\n"
469         f"- ΔPDOP promedio (porcentual): {delta_promedio:.5f}%\n"
470         f"- Proporción total de observaciones bloqueadas: {metrics['
471         global_block_percent']:.2f}%\n"
472     )
473
474
475     return report, pdop_data

```

Listing A.1: Archivo gnss\_leo\_simulator.py

## A.2. Módulo de análisis de PDOP

```

1 class PDOPAnalyzer:
2     """Analizador de PDOP (Dilución de Precisión en Posición) para evaluar impacto de
3     bloqueos"""
4
5     def __init__(self, simulator):
6         """Inicializa el analizador con referencia al simulador principal"""
7         self.simulator = simulator # Instancia del simulador GNSS-LEO
8
9     def calculate_full_pdop(self, observer_name, time_index):
10        """
11        Calcula el PDOP considerando TODOS los satélites GNSS visibles (incluyendo
12        bloqueados).
13
14        Args:
15            observer_name: Nombre del punto de observación
16            time_index: Índice del tiempo en la simulación
17
18        Returns:
19            Valor PDOP (float) o infinito si no hay suficientes satélites
20        """
21
22        if observer_name not in self.simulator.results:
23            raise ValueError(f"No hay datos para {observer_name}")
24
25        data = self.simulator.results[observer_name]
26        if time_index >= len(data['times']):
27            raise ValueError("Índice de tiempo fuera de rango")
28
29        t = data['times'][time_index]
30        observer = self.simulator.observers[observer_name]
31
32        # Obtención de todos los GNSS visibles (sin filtrar bloqueados)
33        visible_sats = []
34        for constel, observations in data['gnss_visible'].items():
35            for obs in observations:
36                # Comparación precisa de tiempos (evita problemas de redondeo)
37                if abs(obs['time'].tt - t.tt) < 1e-8:
38                    # Búsqueda del objeto satélite correspondiente
39                    for sat in self.simulator.gnss_sats[constel]:
40                        if sat.name == obs['name']:
41                            visible_sats.append(sat)
42                            break
43
44        if len(visible_sats) < 4:
45            return float('inf') # Mínimo 4 satélites para cálculo PDOP

```

```

44     # Construcción de matriz de geometría
45     obs_pos = observer.at(t).position.km    # Posición ECEF del observador
46     H = []
47     for sat in visible_sats:
48         sat_pos = (sat - observer).at(t).position.km    # Posición ECEF del satélite
49         rho = sat_pos - obs_pos    # Vector receptor-satélite
50         u = rho / np.linalg.norm(rho)    # Vector unitario
51         H.append(u)
52
53     H = np.array(H)
54     try:
55         Q = np.linalg.inv(H.T @ H)    # Matriz de covarianza
56         return np.sqrt(Q[0,0] + Q[1,1] + Q[2,2])    # Cálculo PDOP
57     except np.linalg.LinAlgError:
58         return float('inf')    # Fallo en cálculo matricial
59
60
61     def calculate_pdop_variation(self, observer_name):
62         """
63             Calcula la variación porcentual del PDOP ( $\Delta$ PDOP) entre escenarios completo y
64             filtrado.
65
66             Args:
67                 observer_name: Nombre del punto de observación
68
69             Returns:
70                 Diccionario con:
71                     - times: Lista de tiempos
72                     - pdop_full: PDOP con todos los satélites
73                     - pdop_filtered: PDOP solo con satélites no bloqueados
74                     - delta_pdop_percent: Variación porcentual  $[(\text{filtrado}-\text{full})/\text{full}*100]$ 
75         """
76
77         data = self.simulator.results[observer_name]
78         pdop_values = []    # PDOP completo (con bloqueados)
79         pdop_filtered_values = []    # PDOP filtrado (sin bloqueados)
80
81         # Cálculo para cada paso de tiempo (con barra de progreso)
82         for i in tqdm(range(len(data['times'])), desc=f"Calculando PDOPs para {observer_name}"):
83             # PDOP estándar (solo satélites no bloqueados)
84             pdop_filt = self.simulator.calculate_pdop(observer_name, i)
85             pdop_filtered_values.append(pdop_filt)
86
87             # PDOP considerando todos los satélites visibles
88             pdop_full = self.calculate_full_pdop(observer_name, i)
89             pdop_values.append(pdop_full)

```

```
89     # Cálculo de variación porcentual
90     delta_pdop = []
91     for full, filt in zip(pdop_values, pdop_filtered_values):
92         if full == float('inf') or full == 0: # Casos especiales
93             delta_pdop.append(0)
94         else:
95             delta_pdop.append((filt - full)/full * 100) # % de variación
96
97     return {
98         'times': data['times'],
99         'pdop_full': pdop_values,
100        'pdop_filtered': pdop_filtered_values,
101        'delta_pdop_percent': delta_pdop
102    }
103
104 def calculate_hourly_pdop_stats(self, observer_name):
105     """
106     Calcula estadísticas horarias de PDOP (promedios y máximos).
107
108     Args:
109         observer_name: Nombre del punto de observación
110
111     Returns:
112         Diccionario con estadísticas por hora:
113         - avg_pdop_full: PDOP promedio (escenario completo)
114         - avg_pdop_filtered: PDOP promedio (escenario filtrado)
115         - avg_delta_pdop: Variación porcentual promedio
116         - max_delta_pdop: Variación porcentual máxima
117     """
118     pdop_data = self.calculate_pdop_variation(observer_name)
119     hourly_stats = defaultdict(lambda: {
120         'pdop_full': [], # Valores PDOP completos
121         'pdop_filtered': [], # Valores PDOP filtrados
122         'delta_pdop': [] # Valores ΔPDOP
123     })
124
125     # Agrupación por hora
126     for t, full, filt, delta in zip(
127         pdop_data['times'],
128         pdop_data['pdop_full'],
129         pdop_data['pdop_filtered'],
130         pdop_data['delta_pdop_percent']
131     ):
132         hour = t.utcnow().hour
133         hourly_stats[hour]['pdop_full'].append(full)
134         hourly_stats[hour]['pdop_filtered'].append(filt)
135         hourly_stats[hour]['delta_pdop'].append(delta)
```

```

136
137     # Cálculo de estadísticas agregadas por hora
138     for hour in hourly_stats:
139         stats = hourly_stats[hour]
140         # Promedios (ignorando valores infinitos)
141         stats['avg_pdop_full'] = np.nanmean(stats['pdop_full']) if stats['pdop_full']
142         else float('nan')
143         stats['avg_pdop_filtered'] = np.nanmean(stats['pdop_filtered']) if stats['
144         pdop_filtered'] else float('nan')
145         stats['avg_delta_pdop'] = np.nanmean(stats['delta_pdop']) if stats['
146         delta_pdop'] else 0
147         # Máximo delta (variación más severa)
148         stats['max_delta_pdop'] = np.nanmax(stats['delta_pdop']) if stats['delta_pdop
149         '] else 0
150
151     return dict(hourly_stats)

```

Listing A.2: Archivo pdop\_analyzer.py

### A.3. Script para visualización de bloqueos

```

1 class Visualizer:
2     def __init__(self, simulator):
3         self.simulator = simulator # Instancia del simulador que contiene los datos
4         self.earth_radius_km = 6371 # Radio terrestre promedio en kilómetros
5
6     def _azel_to_point(self, observer, az, el, distance=500):
7         """
8             Convierte coordenadas azimut/elevación a coordenadas geográficas.
9
10            Args:
11                observer: Objeto Observer con lat/lon del punto de observación
12                az: Ángulo azimutal en grados
13                el: Ángulo de elevación en grados
14                distance: Distancia proyectada en kilómetros (default 500km)
15
16            Returns:
17                Point: Objeto con coordenadas geográficas resultantes
18        """
19        lat = observer.latitude.degrees # Latitud del observador en grados
20        lon = observer.longitude.degrees # Longitud del observador en grados
21
22        # Conversión de azimut a radianes para cálculos trigonométricos
23        az_rad = np.radians(az)
24
25        # Cálculo de desplazamiento en ejes X (longitud) y Y (latitud)

```

```

26     distance_x = distance * np.sin(az_rad) # Componente este-oeste
27     distance_y = distance * np.cos(az_rad) # Componente norte-sur
28
29     # Conversión aproximada a coordenadas geográficas
30     # 111.32 km ≈ 1 grado de latitud (aproximación)
31     new_lat = lat + (distance_y / 111.32)
32     new_lon = lon + (distance_x / (111.32 * np.cos(np.radians(lat)))))
33
34     return Point(new_lon, new_lat)
35
36 def plot_skyplot(self, observer_name, time_index, save_path=None):
37     """
38         Genera un gráfico polar (skyplot) mostrando satélites visibles en un momento
39         específico.
40
41     Args:
42         observer_name: Nombre del observador/ubicación
43         time_index: Índice del tiempo en la simulación
44         save_path: Ruta para guardar la imagen (None para mostrar en pantalla)
45
46     Returns:
47         Figure object si save_path es None, de lo contrario None
48     """
49
50     if observer_name not in self.simulator.results:
51         raise ValueError(f"No hay datos para {observer_name}")
52
53     data = self.simulator.results[observer_name]
54     t = data['times'][time_index] # Obtiene el tiempo específico
55
56     # Configuración del gráfico polar
57     fig, ax = plt.subplots(subplot_kw={'projection': 'polar'})
58     ax.set_theta_zero_location('N') # Norte en la parte superior
59     ax.set_theta_direction(-1) # Sentido horario
60     ax.set_ylim(0, 90) # Límite de elevación (0-90°)
61     ax.set_yticks(range(0, 90, 15)) # Marcas cada 15°
62     ax.set_yticklabels([f'{90-x}°' for x in range(0, 90, 15)]) # Etiquetas de
63     # elevación
64     ax.grid(True)
65
66     # Graficar satélites GNSS visibles
67     for constel, observations in data['gnss_visible'].items():
68         for obs in observations:
69             if obs['time'] == t:
70                 # Asignación de colores por constelación
71                 color = {
72                     'GPS': 'blue',
73                     'GALILEO': 'green',
74                     'GLONASS': 'red',
75                     'BEIDOU': 'purple',
76                     'QUANTUM': 'orange'
77                 }
78
79                 # Dibujar el satélite en el gráfico
80                 ax.plot(obs['azimuth'], obs['elevation'], color=color[constel])
81
82     # Mostrar el gráfico
83     plt.show()

```

```
71         'GLONASS': 'red',
72         'BEIDOU': 'orange'
73     }.get(constel, 'gray')

74
75     # Marcadores diferentes para satélites bloqueados
76     marker = 'o' if not obs['blocked'] else 'x'
77     size = 8 if not obs['blocked'] else 10

78
79     # Graficar punto en el skyplot
80     ax.scatter(np.radians(obs['az']), 90 - obs['alt'],
81                color=color, marker=marker, s=size,
82                label=f'{constel} {"(bloqueado)" if obs["blocked"] else ""}')
83
84
85     # Graficar satélites LEO (solo los cercanos a GNSS)
86     leo_positions = next(item['positions'] for item in data['leo_positions'] if item[
87     'time'] == t)
88     for leo in leo_positions:
89         ax.scatter(np.radians(leo['az']), 90 - leo['alt'],
90                 color='black', marker='o', s=10, alpha=0.8, label='LEO')

91
92     # Configuración de leyenda y título
93     handles, labels = ax.get_legend_handles_labels()
94     by_label = dict(zip(labels, handles)) # Eliminar duplicados
95     ax.legend(by_label.values(), by_label.keys(), loc='upper right', bbox_to_anchor
96     =(1.3, 1))

97     # Formatear tiempo para el título
98     time_str = t.utc_datetime().strftime('%Y-%m-%d %H:%M:%S UTC')
99     ax.set_title(f'Skyplot - {observer_name}\n{time_str}\nGNSS visibles y satélites
100    LEO')

101
102    # Guardar o mostrar el gráfico
103    if save_path:
104        plt.savefig(save_path, bbox_inches='tight', dpi=150)
105        plt.close()
106        return None
107    else:
108        return fig

109
110
111    def plot_blocking_timeline(self, observer_name, save_path=None):
112        """
113            Grafica el porcentaje de bloqueo GNSS a lo largo del tiempo.
114
115            Args:
116                observer_name: Nombre del observador/ubicación
117                save_path: Ruta para guardar la imagen (None para mostrar en pantalla)
```

```
114
115     Returns:
116         Figure object si save_path es None, de lo contrario None
117         """
118     if observer_name not in self.simulator.results:
119         raise ValueError(f"No hay datos para {observer_name}")
120
121     # Obtener métricas calculadas
122     metrics = self.simulator.calculate_metrics(observer_name)
123     hours = sorted(metrics['hourly_stats'].keys()) # Horas con datos
124     block_percents = [metrics['hourly_stats'][h]['block_percent'] for h in hours] # Porcentajes de bloqueo
125
126     # Configurar gráfico
127     fig, ax = plt.subplots()
128     ax.plot(hours, block_percents, 'b-', marker='o') # Línea azul con marcadores
129     ax.set_xlabel('Hora del día (UTC)')
130     ax.set_ylabel('Porcentaje de bloqueo GNSS')
131     ax.set_title(f'Porcentaje de bloqueo GNSS por hora - {observer_name}')
132     ax.set_xticks(range(24)) # Marcas horarias completas
133     ax.grid(True)
134
135     # Guardar o mostrar
136     if save_path:
137         plt.savefig(save_path, bbox_inches='tight', dpi=150)
138         plt.close()
139         return None
140     else:
141         return fig
142
143 def plot_pdop_heatmap(self, observer_name, save_path=None):
144     """
145     Genera un heatmap mostrando valores de PDOP por hora y azimut.
146
147     Args:
148         observer_name: Nombre del observador/ubicación
149         save_path: Ruta para guardar la imagen (None para mostrar en pantalla)
150         """
151     data = self.simulator.results[observer_name]
152     observer = self.simulator.observers[observer_name]
153
154     # Preparar datos para el heatmap
155     heatmap_data = []
156     for i, time_data in enumerate(data['leo_positions']):
157         t = time_data['time']
158         hour = t.utc_datetime().hour # Extraer hora UTC
```

```
160     # Calcular PDOP para este momento
161     pdop_full = self.simulator.calculate_pdop(observer_name, i)
162
163     # Agregar datos por segmento de azimut (bins de 15°)
164     for az_bin in range(0, 360, 15):
165         heatmap_data.append({
166             'hour': hour,
167             'azimuth_bin': az_bin,
168             'pdop': pdop_full
169         })
170
171     # Crear DataFrame y pivotar para heatmap
172     df = pd.DataFrame(heatmap_data)
173     pivot_df = df.pivot_table(index='hour', columns='azimuth_bin', values='pdop',
174                               aggfunc='mean')
175
176     # Configurar figura
177     fig, ax = plt.subplots(figsize=(15, 8))
178
179     # Generar heatmap con escala de colores invertida (mejor PDOP = más oscuro)
180     sns.heatmap(pivot_df, cmap='viridis_r', ax=ax, cbar_kws={'label': 'PDOP'})
181
182     # Configurar título y etiquetas
183     ax.set_title(f'Heatmap de PDOP por hora y azimut\n{n{observer_name}}', fontsize=14)
184     ax.set_xlabel('Azimut (bin 15°)')
185     ax.set_ylabel('Hora del día (UTC)')
186
187     # Guardar o mostrar
188     if save_path:
189         plt.savefig(save_path, bbox_inches='tight', dpi=300)
190         plt.close()
191     else:
192         plt.show()
193
194     def plot_pdop_comparison(self, observer_name, pdop_data=None, save_path=None):
195         """
196             Compara PDOP completo vs filtrado y muestra la variación porcentual.
197
198             Args:
199                 observer_name: Nombre del observador/ubicación
200                 pdop_data: Datos precalculados (opcional)
201                 save_path: Ruta para guardar la imagen (None para mostrar en pantalla)
202
203             # Obtener datos PDOP si no se proporcionan
204             if pdop_data is None:
205                 pdop_analyzer = PDOPAnalyzer(self.simulator)
206                 pdop_data = pdop_analyzer.calculate_pdop_variation(observer_name)
```

```

206
207     # Preparar DataFrame para plotting
208     times = [t.utc_datetime() for t in pdop_data['times']]
209     df = pd.DataFrame({
210         'Hora': times,
211         'PDOP Completo': pdop_data['pdop_full'],
212         'PDOP Filtrado': pdop_data['pdop_filtered'],
213         'ΔPDOP (%)': pdop_data['delta_pdop_percent']
214     })
215
216     # Crear figura con dos subplots (PDOP y variación)
217     fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(12, 10), sharex=True)
218
219     # Gráfico superior: Comparación PDOP completo vs filtrado
220     df.plot(x='Hora', y=['PDOP Completo', 'PDOP Filtrado'],
221             ax=ax1, style=['-', '--'], linewidth=2)
222     ax1.set_title(f'Comparación PDOP completo vs filtrado - {observer_name}')
223     ax1.set_ylabel('Valor PDOP')
224     ax1.legend()
225     ax1.grid(True)
226
227     # Gráfico inferior: Variación porcentual del PDOP
228     df.plot(x='Hora', y='ΔPDOP (%)', ax=ax2, color='red', linewidth=2)
229     ax2.set_title('Variación porcentual del PDOP (ΔPDOP)')
230     ax2.set_ylabel('ΔPDOP (%)')
231     ax2.set_xlabel('Hora del día')
232     ax2.grid(True)
233
234     # Ajustar layout para evitar solapamientos
235     plt.tight_layout()
236
237     # Guardar o mostrar
238     if save_path:
239         plt.savefig(save_path, bbox_inches='tight', dpi=300)
240         plt.close()
241     else:
242         plt.show()

```

Listing A.3: Archivo graficador\_bloqueos.py

## A.4. Reportes Detallados por Observador

En este apéndice se presentan los reportes de texto completos con las métricas finales generadas por el simulador para cada combinación de observador y densidad orbital.

### A.4.1. Observador: Tromsø, Noruega

```

1 Desde el punto de observación de la ciudad Tromso, el Índice de Bloqueo Visual Promedio (BI) fue del 0.00%. El satélite GNSS más afectado fue GPS BIIR-5 (PRN 22) (GPS), con un 0.00% de tiempo bloqueado individualmente.
2
3 Estadísticas por constelación:
4 - GPS: 0.00% de bloqueo (0/3008 observaciones)
5 - GLONASS: 0.00% de bloqueo (0/3804 observaciones)
6 - BEIDOU: 0.00% de bloqueo (0/4422 observaciones)
7 - GALILEO: 0.00% de bloqueo (0/2991 observaciones)
8
9 Análisis de PDOP en momentos clave:
10 - Hora de máximo bloqueo (12:00): PDOP completo = 0.47
11 - Hora de mínimo bloqueo (12:00): PDOP completo = 0.47
12 - Hora de bloqueo promedio (12:00): PDOP completo = 0.47
13
14 MÉTRICAS AVANZADAS:
15 - Frecuencia de bloqueos: 0.0 eventos/hora
16 - PDOP completo promedio: 0.47550
17 - PDOP filtrado promedio: 0.47550
18 - ΔPDOP promedio (porcentual): 0.00000%
19 - Proporción total de observaciones bloqueadas: 0.00%

```

Listing A.4: Reporte de métricas para Tromsø (Densidad Baja).

```

1 Desde el punto de observación de la ciudad Tromso, el Índice de Bloqueo Visual Promedio (BI) fue del 0.10%. El satélite GNSS más afectado fue BEIDOU-2 IGSO-1 (C06) (BEIDOU), con un 1.90% de tiempo bloqueado individualmente.
2
3 Estadísticas por constelación:
4 - GPS: 0.17% de bloqueo (5/3008 observaciones)
5 - GLONASS: 0.13% de bloqueo (5/3804 observaciones)
6 - BEIDOU: 0.07% de bloqueo (3/4422 observaciones)
7 - GALILEO: 0.07% de bloqueo (2/2991 observaciones)
8
9 Análisis de PDOP en momentos clave:
10 - Hora de máximo bloqueo (18:00): PDOP completo = 0.49
11 - Hora de mínimo bloqueo (13:00): PDOP completo = 0.47
12 - Hora de bloqueo promedio (13:00): PDOP completo = 0.47
13
14 MÉTRICAS AVANZADAS:
15 - Frecuencia de bloqueos: 0.6 eventos/hora
16 - PDOP completo promedio: 0.47550
17 - PDOP filtrado promedio: 0.47570
18 - ΔPDOP promedio (porcentual): 0.04329%
19 - Proporción total de observaciones bloqueadas: 0.11%

```

---

Listing A.5: Reporte de métricas para Tromsø (Densidad Media).

```
1 Desde el punto de observación de la ciudad Tromsø, el Índice de Bloqueo Visual Promedio (BI) fue del 0.43%. El satélite GNSS más afectado fue GSAT0217 (GALILEO 21) (GALILEO), con un 2.50% de tiempo bloqueado individualmente.  
2  
3 Estadísticas por constelación:  
4 - GPS: 0.43% de bloqueo (13/3008 observaciones)  
5 - GLONASS: 0.42% de bloqueo (16/3804 observaciones)  
6 - BEIDOU: 0.50% de bloqueo (22/4422 observaciones)  
7 - GALILEO: 0.37% de bloqueo (11/2991 observaciones)  
8  
9 Análisis de PDOP en momentos clave:  
10 - Hora de máximo bloqueo (15:00): PDOP completo = 0.46  
11 - Hora de mínimo bloqueo (13:00): PDOP completo = 0.47  
12 - Hora de bloqueo promedio (13:00): PDOP completo = 0.47  
13  
14 MÉTRICAS AVANZADAS:  
15 - Frecuencia de bloqueos: 2.6 eventos/hora  
16 - PDOP completo promedio: 0.47550  
17 - PDOP filtrado promedio: 0.47631  
18 - ΔPDOP promedio (porcentual): 0.17044%  
19 - Proporción total de observaciones bloqueadas: 0.44%
```

Listing A.6: Reporte de métricas para Tromsø (Densidad Alta).

#### A.4.2. Observador: Quito, Ecuador

```

1 Desde el punto de observación de la ciudad Quito, el Índice de Bloqueo Visual Promedio (
2 BI) fue del 0.09%. El satélite GNSS más afectado fue BEIDOU-3 M23 (C45) (BEIDOU), con
3 un 2.36% de tiempo bloqueado individualmente.
4
5 Estadísticas por constelación:
6 - GPS: 0.14% de bloqueo (4/2890 observaciones)
7 - GLONASS: 0.04% de bloqueo (1/2788 observaciones)
8 - BEIDOU: 0.16% de bloqueo (5/3182 observaciones)
9 - GALILEO: 0.03% de bloqueo (1/3156 observaciones)
10
11 Análisis de PDOP en momentos clave:
12 - Hora de máximo bloqueo (23:00): PDOP completo = 0.52
13 - Hora de mínimo bloqueo (12:00): PDOP completo = 0.53
14 - Hora de bloqueo promedio (12:00): PDOP completo = 0.53
15
16 MÉTRICAS AVANZADAS:
17 - Frecuencia de bloqueos: 0.5 eventos/hora
18 - PDOP completo promedio: 0.51605
19 - PDOP filtrado promedio: 0.51621
- ΔPDOP promedio (porcentual): 0.03216%
- Proporción total de observaciones bloqueadas: 0.09%

```

Listing A.7: Reporte de métricas para Quito (Densidad Baja).

```

1 Desde el punto de observación de la ciudad Quito, el Índice de Bloqueo Visual Promedio (
2 BI) fue del 0.36%. El satélite GNSS más afectado fue BEIDOU-3 M23 (C45) (BEIDOU), con
3 un 3.94% de tiempo bloqueado individualmente.
4
5 Estadísticas por constelación:
6 - GPS: 0.38% de bloqueo (11/2890 observaciones)
7 - GLONASS: 0.25% de bloqueo (7/2788 observaciones)
8 - BEIDOU: 0.50% de bloqueo (16/3182 observaciones)
9 - GALILEO: 0.29% de bloqueo (9/3156 observaciones)
10
11 Análisis de PDOP en momentos clave:
12 - Hora de máximo bloqueo (23:00): PDOP completo = 0.52
13 - Hora de mínimo bloqueo (12:00): PDOP completo = 0.53
14 - Hora de bloqueo promedio (12:00): PDOP completo = 0.53
15
16 MÉTRICAS AVANZADAS:
17 - Frecuencia de bloqueos: 1.8 eventos/hora
18 - PDOP completo promedio: 0.51605
19 - PDOP filtrado promedio: 0.51679
- ΔPDOP promedio (porcentual): 0.14396%
- Proporción total de observaciones bloqueadas: 0.36%

```

---

Listing A.8: Reporte de métricas para Quito (Densidad Media).

```
1 Desde el punto de observación de la ciudad Quito, el Índice de Bloqueo Visual Promedio (BI) fue del 1.50%. El satélite GNSS más afectado fue GSAT0206 (GALILEO 10) (GALILEO), con un 7.84% de tiempo bloqueado individualmente.  
2  
3 Estadísticas por constelación:  
4 - GPS: 1.49% de bloqueo (43/2890 observaciones)  
5 - GLONASS: 1.18% de bloqueo (33/2788 observaciones)  
6 - BEIDOU: 1.60% de bloqueo (51/3182 observaciones)  
7 - GALILEO: 1.68% de bloqueo (53/3156 observaciones)  
8  
9 Análisis de PDOP en momentos clave:  
10 - Hora de máximo bloqueo (05:00): PDOP completo = 0.54  
11 - Hora de mínimo bloqueo (12:00): PDOP completo = 0.53  
12 - Hora de bloqueo promedio (19:00): PDOP completo = 0.51  
13  
14 MÉTRICAS AVANZADAS:  
15 - Frecuencia de bloqueos: 7.5 eventos/hora  
16 - PDOP completo promedio: 0.51605  
17 - PDOP filtrado promedio: 0.51905  
18 - ΔPDOP promedio (porcentual): 0.58313%  
19 - Proporción total de observaciones bloqueadas: 1.50%
```

Listing A.9: Reporte de métricas para Quito (Densidad Alta).

#### A.4.3. Observador: Alice Springs, Australia

```

1 Desde el punto de observación de la ciudad Alice Springs, el Índice de Bloqueo Visual
2 Promedio (BI) fue del 0.10%. El satélite GNSS más afectado fue GSAT0205 (GALILEO 9) (
3 GALILEO), con un 4.55% de tiempo bloqueado individualmente.
4
5 Estadísticas por constelación:
6 - GPS: 0.15% de bloqueo (4/2683 observaciones)
7 - GLONASS: 0.00% de bloqueo (0/4346 observaciones)
8 - BEIDOU: 0.13% de bloqueo (10/7862 observaciones)
9 - GALILEO: 0.16% de bloqueo (5/3152 observaciones)
10
11 Análisis de PDOP en momentos clave:
12 - Hora de máximo bloqueo (12:00): PDOP completo = 0.38
13 - Hora de mínimo bloqueo (12:00): PDOP completo = 0.38
14 - Hora de bloqueo promedio (12:00): PDOP completo = 0.38
15
16 MÉTRICAS AVANZADAS:
17 - Frecuencia de bloqueos: 0.8 eventos/hora
18 - PDOP completo promedio: 0.39375
19 - PDOP filtrado promedio: 0.39393
- ΔPDOP promedio (porcentual): 0.04722%
- Proporción total de observaciones bloqueadas: 0.11%

```

Listing A.10: Reporte de métricas para Alice Springs (Densidad Baja).

```

1 Desde el punto de observación de la ciudad Alice Springs, el Índice de Bloqueo Visual
2 Promedio (BI) fue del 0.49%. El satélite GNSS más afectado fue IRNSS-1I (GLONASS),
3 con un 4.69% de tiempo bloqueado individualmente.
4
5 Estadísticas por constelación:
6 - GPS: 0.60% de bloqueo (16/2683 observaciones)
7 - GLONASS: 0.30% de bloqueo (13/4346 observaciones)
8 - BEIDOU: 0.56% de bloqueo (44/7862 observaciones)
9 - GALILEO: 0.51% de bloqueo (16/3152 observaciones)
10
11 Análisis de PDOP en momentos clave:
12 - Hora de máximo bloqueo (00:00): PDOP completo = 0.40
13 - Hora de mínimo bloqueo (12:00): PDOP completo = 0.38
14 - Hora de bloqueo promedio (12:00): PDOP completo = 0.38
15
16 MÉTRICAS AVANZADAS:
17 - Frecuencia de bloqueos: 3.7 eventos/hora
18 - PDOP completo promedio: 0.39375
19 - PDOP filtrado promedio: 0.39466
- ΔPDOP promedio (porcentual): 0.23159%
- Proporción total de observaciones bloqueadas: 0.49%

```

---

Listing A.11: Reporte de métricas para Alice Springs (Densidad Media).

```
1 Desde el punto de observación de la ciudad Alice Springs, el Índice de Bloqueo Visual  
2 Promedio (BI) fue del 1.43%. El satélite GNSS más afectado fue IRNSS-1A (GLONASS),  
3 con un 7.04% de tiempo bloqueado individualmente.  
4  
5 Estadísticas por constelación:  
6 - GPS: 1.83% de bloqueo (49/2683 observaciones)  
7 - GLONASS: 1.31% de bloqueo (57/4346 observaciones)  
8 - BEIDOU: 1.22% de bloqueo (96/7862 observaciones)  
9 - GALILEO: 1.74% de bloqueo (55/3152 observaciones)  
10  
11 Análisis de PDOP en momentos clave:  
12 - Hora de máximo bloqueo (00:00): PDOP completo = 0.40  
13 - Hora de mínimo bloqueo (12:00): PDOP completo = 0.38  
14 - Hora de bloqueo promedio (04:00): PDOP completo = 0.38  
15  
16 MÉTRICAS AVANZADAS:  
17 - Frecuencia de bloqueos: 10.7 eventos/hora  
18 - PDOP completo promedio: 0.39375  
19 - PDOP filtrado promedio: 0.39642  
- ΔPDOP promedio (porcentual): 0.67603%  
- Proporción total de observaciones bloqueadas: 1.42%
```

Listing A.12: Reporte de métricas para Alice Springs (Densidad Alta).

# Bibliografía

- [1] T. E. Humphreys, Z. L. Clements, and W. Qin, “Gnss interference: Situational awareness and leo backup,” The University of Texas at Austin, Tech. Rep., 9 2024.
- [2] SpaceX, “Starlink satellite constellation overview,” Online resource, 2023. [Online]. Available: <https://www.starlink.com>
- [3] OneWeb. (2025) OneWeb: Global Connectivity. Accessed: May 15, 2025. [Online]. Available: <https://www.oneweb.world>
- [4] M. Guan, T. Xu, F. Gao, W. Nie, and H. Yang, “Optimal walker constellation design of leo-based global navigation and augmentation system,” *Remote Sensing*, vol. 12, p. 1845, 06 2020.
- [5] T. E. Talbert. (2020) Starlink satellites are disrupting cosmic studies. Accessed: May 15, 2025. [Online]. Available: <https://www.astronomy.com/science/starlink-satellites-disrupt-cosmic-studies>
- [6] SpaceX, “Starlink’s approach to satellite demisability,” [https://www.starlink.com/public-files/Starlink\\_Approach\\_to\\_Satellite\\_Demisability.pdf](https://www.starlink.com/public-files/Starlink_Approach_to_Satellite_Demisability.pdf), 2025, accessed: 2025-05-16.
- [7] U. Weinbach and O. Montenbruck. (2023) Enter leo on the gnss stage: Navigation with starlink satellites. Accessed: May 15, 2025. [Online]. Available: <https://insidegnss.com/enter-leo-on-the-gnss-stage-navigation-with-starlink-satellites>
- [8] J. Hong, R. Tu, P. Zhang, R. Zhang, J. Han, L. Fan, S. Wang, and X. Lu, “GNSS Rapid Precise Point Positioning Enhanced by Low Earth Orbit Satellites,” *Satellite Navigation*, vol. 4, p. 11, 2023.
- [9] European Space Agency, “Precise point positioning,” [https://gssc.esa.int/navipedia/index.php/Precise\\_Point\\_Positioning](https://gssc.esa.int/navipedia/index.php/Precise_Point_Positioning), accedido el 17 de mayo de 2025.
- [10] M. Garcia-Fernandez, “Mapping tle orbital parameters to gnss ephemeris for leo pnt mega-constellation orbit simulations and visibility analysis,” 2024, preprint. [Online]. Available: <https://arxiv.org/abs/2401.17767>
- [11] N. Duetsch, H. Ernest, T. Pany, A. P. Campello, D. Borheck, J. Speidel, and H. Sunay, “Gnss interference detection and geolocation from leo satellites – satellite formation and payload design specific considerations and their impact on the detection sensitivity and geolocation accuracy,” in *Proceedings of the European Navigation Conference (ENC)*. Wiesbaden, Germany: Universität der Bundeswehr München and OHB System AG, 2024, accessed: 2025-05-19. [Online]. Available: <https://athene-forschung.unibw.de/doc/149939/149939.pdf>

- [12] Y. Wang, X. Meng, X. Wang, Y. Li, and Z. Zhang, “Leo enhanced gnss (legnss) precise point positioning with simulated leo and real gnss data,” *Advances in Space Research*, vol. 83, no. 6, pp. 1459–1471, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0273117724005489>
- [13] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*, 2nd ed. Ganga-Jamuna Press, 2006.
- [14] E. D. Kaplan and C. J. Hegarty, *Understanding GPS/GNSS: Principles and Applications*, 3rd ed. Artech House, 2017.
- [15] E. S. Agency, “Esrangle and arctic observations from northern scandinavia,” ESA Earth Observation Portal, Tech. Rep., 2020, accessed: 2025-06-06. [Online]. Available: <https://earth.esa.int/eogateway/stations/esrange>
- [16] E. P. Network, “Tromsø gnss permanent station (tro1),” European Reference Frame (EUREF), Tech. Rep., 2021, accessed: 2025-06-06. [Online]. Available: [http://www.epnchoma.be/station\\_logs/TRO1\\_20210630.log](http://www.epnchoma.be/station_logs/TRO1_20210630.log)
- [17] S. V. Polyakov and A. N. Krasheninnikov, “Ionospheric disturbance detection using gnss signals in eastern siberia,” *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 197, pp. 105–113, 2019.
- [18] M. Romero and D. Jativa, “Gnss performance analysis in equatorial urban areas: A case study in quito, ecuador,” *Revista Politécnica*, vol. 43, no. 1, pp. 55–62, 2019, accessed: 2025-06-06. [Online]. Available: [https://revistapolitecnica.epn.edu.ec/index.php/revista\\_pol/article/view/892](https://revistapolitecnica.epn.edu.ec/index.php/revista_pol/article/view/892)
- [19] G. Australia, “Alice springs space geodesy facility,” Australian Government, Tech. Rep., 2021, accessed: 2025-06-06. [Online]. Available: <https://www.ga.gov.au/about/projects/geodesy/space-geodesy>
- [20] S. Network, “Estación gnss: La quiaca, argentina,” 2020, accessed: 2025-06-06. [Online]. Available: <https://www.sirgas.org/en/red/sirgas-continuously-operating-stations/>
- [21] T. Afullo and A. Malinga, “Tec fluctuation in equatorial africa using gnss measurements,” *Advances in Space Research*, vol. 72, no. 8, pp. 2197–2209, 2023.
- [22] J. O. Oketch and W. M. Mwangi, “Assessment of gnss signal degradation in equatorial urban regions: The case of nairobi,” *IEEE Access*, vol. 11, pp. 59 432–59 442, 2023.
- [23] U. of Texas Observatory, “McDonald observatory site conditions (near marfa, tx),” 2022, accessed: 2025-06-06. [Online]. Available: <https://mcdonaldobservatory.org/about/site>
- [24] E. S. Agency, “Esa ground stations – arctic activities and ionospheric monitoring,” European Space Agency, Tech. Rep., 2021, available at: [https://www.esa.int/Enabling\\_Support/Operations/Earth\\_observation\\_ground\\_stations](https://www.esa.int/Enabling_Support/Operations/Earth_observation_ground_stations).
- [25] R. T. Tsunoda, “On the nature of localized wave structures in the equatorial ionosphere,” *Geophysical Research Letters*, vol. 8, no. 6, pp. 649–652, 1981.
- [26] G. Australia, “National geodesy program: Geodetic stations and observatories,” Australian Government, Tech. Rep., 2022, alice Springs SLR and GNSS infrastructure documentation. Available at: <https://www.ga.gov.au/scientific-topics/positioning-navigation/geodesy/gnss-networks>.

- [27] T. Kelso, “Celestrak: Current norad two-line element sets,” 2023, Último acceso: junio de 2025. [Online]. Available: <https://celestrak.org/NORAD/documentation/tle-fmt.php>
- [28] S. D. Team, “Skyfield: High precision astronomy for python,” Python library documentation, 2024. [Online]. Available: <https://rhodesmill.org/skyfield/>
- [29] N. Department of Engineering Cybernetics, “Satellite orbit propagation using python and skyfield,” <https://www.ntnu.edu/engcyber/skyfield-guide>, 2021, tutorial académico consultado en junio de 2025.
- [30] I. González and V. León, “Impact of large leo constellations on gnss availability: A simulation-based study,” *Journal of Navigation and Satellite Systems*, vol. 12, no. 3, pp. 145–162, 2022.
- [31] M. Lelyveld and A. Cheng, “Visual obstruction analysis of gnss signals by starlink constellations,” in *Proceedings of the IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2023, pp. 343–350.
- [32] Python Software Foundation, “Python programming language – official website,” 2024, accessed: 2025-06-26. [Online]. Available: <https://www.python.org/>
- [33] J. Blanch, T. Walter, P. Enge, and et al., “Raim with integrity and continuity requirements,” *Navigation: Journal of the Institute of Navigation*, vol. 62, no. 4, pp. 267–283, 2015. [Online]. Available: <https://www.ion.org/publications/abstract.cfm?articleID=12945>
- [34] Wikipedia contributors, “Receiver autonomous integrity monitoring — wikipedia, the free encyclopedia,” [https://en.wikipedia.org/wiki/Receiver\\_autonomous\\_integrity\\_monitoring](https://en.wikipedia.org/wiki/Receiver_autonomous_integrity_monitoring), 2024, accessed: 2025-06-29.