MASTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIONES

TRABAJO FIN DE MASTER

Desarrollo de una herramienta para la ocultación y *spoofing* de señales GPS

Autor: Arturo Amando Herreras Redondo

Director: Javier Matanza Domingo

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Desarrollo de una herramienta para la ocultación y *spoofing* de señales GPS

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2024/25 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Arturo Herreras Fecha: 17/06/2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Firmado por MATANZA DOMINGO JAVIER - ***4397** el día 13/06/2025 con un certificado emitido por AC FNMT Usuarios

Fdo.: Javier Matanza Domingo Fecha: 17/06/2025

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Arturo Amando Herreras Redondo

DECLARA ser el titular de los derechos de propiedad intelectual de la obra: Desarrollo de una herramienta para la ocultación y *spoofing* de señales GPS, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.
b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
e) Asignar por defecto a estos trabajos una licencia Creative Commons.
f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente)*.

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

a) Que la Universidad identifique claramente su nombre como autor de la misma
b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
c) Solicitar la retirada de la obra del repositorio por causa justificada.
d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e

intereses a causa de la cesión.

d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 17 de junio de 2025.

ACEPTA

Fdo Arturo Herreras

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

MASTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIONES

TRABAJO FIN DE MASTER

Desarrollo de una herramienta para la ocultación y *spoofing* de señales GPS

Autor: Arturo Amando Herreras Redondo

Director: Javier Matanza Domingo

Madrid

Resumen del Trabajo de Fin de Máster.

R.1 INTRODUCCIÓN, MOTIVACIÓN Y OBJETIVOS.

Objetivo general del Proyecto:

El propósito central de este Trabajo de Fin de Máster es doble: por un lado, demostrar experimentalmente la fragilidad del sistema civil GPS frente a interferencias intencionadas; por otro, ofrecer una plataforma reproducible, económica y accesible que permita su estudio en entornos académicos. Esta aproximación no se limita a una exploración teórica, sino que proporciona una implementación concreta y funcional con hardware de bajo coste y herramientas *open-source.* De este modo, el proyecto se configura como una herramienta versátil tanto para la docencia como para la investigación en el ámbito de la seguridad GNSS, permitiendo la comprensión práctica de los riesgos y la evaluación de posibles soluciones.

Motivación dual del proyecto

La motivación que impulsa este trabajo se articula en torno a dos dimensiones complementarias.

1. En primer lugar, desde el punto de vista formativo, se busca suplir la carencia de entornos experimentales accesibles en el campo de la seguridad GNSS. La mayoría de los estudios se limitan a un enfoque teórico, mientras que este proyecto permite observar en tiempo real el efecto de ataques de *jamming* y *spoofing* sobre receptores reales, lo que facilita una experiencia educativa tangible y completa.

2. En segundo lugar, existe una motivación de carácter socio-tecnológico, derivada de la creciente dependencia del GPS en sectores críticos como la energía, las finanzas, el transporte y las telecomunicaciones. A pesar de esta dependencia, las señales civiles de GPS carecen de mecanismos de autenticación, lo que las hace vulnerables a interferencias generadas incluso con equipos de bajo coste. La facilidad con la que puede replicarse un entorno de ataque básico actúa como una llamada de atención

hacia reguladores, desarrolladores y fabricantes, incentivando la adopción de tecnologías más robustas, como OSNMA o la fusión con sensores inerciales. En este contexto, el proyecto actúa como un puente entre el ámbito académico y el industrial, alineándose con objetivos de desarrollo sostenible vinculados a la educación de calidad (ODS 4) y a la innovación tecnológica (ODS 9).

Relevancia crítica del GPS

Desde su apertura al uso civil en 1993, el GPS ha evolucionado hasta convertirse en una infraestructura esencial y silenciosa que sustenta buena parte de la economía digital y de las operaciones cotidianas. Las telecomunicaciones móviles, la sincronización de redes eléctricas, las transacciones financieras en tiempo real o la navegación aérea y marítima se basan en la señal GPS para obtener tiempo y posición con una precisión extrema. La señal L1 C/A, utilizada mayoritariamente por los dispositivos civiles, opera en la banda L1 (1.575 MHz) con una potencia extremadamente baja (≈ –160 dBW), lo que la hace especialmente sensible a interferencias locales.

Diversos informes internacionales, incluidos los de la Comisión Europea y la OTAN, han identificado al GPS como un *"single point of failure"* para infraestructuras críticas, advirtiendo de que una interrupción prolongada podría acarrear consecuencias económicas de miles de millones de euros. Este escenario otorga al presente trabajo una dimensión estratégica, al proponer un entorno de pruebas que permite estudiar, reproducir y comprender los riesgos inherentes a esta dependencia tecnológica.

Definición de jamming y spoofing

Los ataques de *jamming* y *spoofing* representan dos formas distintas de interferencia sobre el sistema GPS, cada una con implicaciones específicas. El *jamming* consiste en emitir señales de ruido o portadoras interferentes que elevan el nivel del ruido de fondo en el receptor GNSS, impidiendo la correcta adquisición y seguimiento de las señales auténticas. Esto genera una pérdida de servicio, caracterizada por la caída brusca del parámetro C/N_0 y la transición del receptor a modo "*No Fix*".

Por su parte, el *spoofing* busca suplantar las señales GPS legítimas mediante la transmisión de réplicas estructuralmente válidas pero con información manipulada. Estas señales, que emulan el formato C/A y la navegación, pueden desplazar la posición, el tiempo o la velocidad estimados por el receptor sin activar alarmas internas, lo que las convierte en una amenaza más sutil y difícil de detectar.

En este trabajo, ambos ataques han sido implementados: el *jamming* mediante la generación de ruido gaussiano en SDR, y el *spoofing* mediante la reproducción de tramas IQ sintéticas con gps-sdr-sim, logrando desplazar la solución de posicionamiento del receptor en más de 100 metros de forma gradual y controlada.

R.2 METODOLOGÍA, RECURSOS Y HERRAMIENTAS DESARROLLADAS.

Hardware y software utilizados:

El entorno experimental se ha construido con hardware accesible y herramientas de código abierto. El transmisor utilizado es un PlutoSDR (ADALM-PLUTO), conectado mediante atenuadores SMA a una antena *patch* pasiva. La generación y control de las señales se realizan desde un portátil con Ubuntu 22.04.

El software empleado incluye GNU Radio 3.10 para el diseño de flujos de señal en *jamming*, gps-sdr-sim para la generación de señales *spoofing*, y GNSS-SDR como receptor software para registrar parámetros como el C/N_0 y la pérdida de *tracking*. Todo el sistema está diseñado para ser replicable con un presupuesto reducido y sin necesidad de hardware especializado.

Fases del desarrollo:

El trabajo se organizó en varias fases: primero se caracterizó el receptor en condiciones normales; luego se diseñó un entorno controlado para evitar emisiones al exterior. A continuación, se implementaron los ataques de *jamming* y *spoofing*, observando sus efectos sobre la señal. Finalmente, se analizaron los resultados registrados con GNSS-SDR.

Herramienta de *jamming*:

La herramienta de *jamming* utiliza el PlutoSDR como generador de ruido gaussiano de banda ancha centrado en 1575,42 MHz. La señal se genera en GNU Radio y se transmite mediante librerías libiio. La potencia se ajusta con atenuadores, permitiendo elevar el nivel de ruido y colapsar los correladores del receptor.

La interferencia produce una caída brusca del C/N_0 y provoca la pérdida del *fix*. Una vez cesa la emisión, el receptor recupera la señal tras 10 a 15 segundos.

Herramienta de *spoofing:*

El sistema de *spoofing* se basa en gps-sdr-sim, que genera un archivo IQ con datos de navegación válidos para una trayectoria circular de unos 150 metros. Este archivo se transmite en bucle desde el PlutoSDR, con una ganancia ligeramente superior a la señal real, lo que permite una captura progresiva.

Una vez tomada la señal falsa, el receptor calcula posiciones erróneas siguiendo la trayectoria sintética, sin levantar alertas. El proceso mimetiza la señal auténtica y permite desplazar al receptor más de 100 metros sin interrupciones ni síntomas de fallo.

R.3 CONCLUSIONES Y PRÓXIMOS PASOS.

Contribución a los Objetivos de Desarrollo Sostenible:

El trabajo se alinea principalmente con el ODS 4 (educación de calidad), al proporcionar un entorno práctico para el aprendizaje activo en seguridad GNSS, y con el ODS 9 (industria, innovación e infraestructura), al fomentar el desarrollo de sistemas resilientes desde una perspectiva abierta y experimental. También contribuye al ODS 10 (reducción de desigualdades), al democratizar el acceso a herramientas avanzadas sin necesidad de recursos elevados.

Conclusiones principales:

El proyecto ha demostrado de forma práctica y reproducible la vulnerabilidad del sistema GPS civil frente a interferencias activas. Tanto el *jamming* como el *spoofing* han podido implementarse con medios asequibles y en un entorno controlado, logrando efectos significativos sobre el receptor sin necesidad de equipamiento especializado. Esta constatación subraya la fragilidad de una infraestructura crítica ampliamente extendida, cuya señal puede ser anulada o suplantada incluso desde instalaciones domésticas.

Además del logro técnico, el entorno diseñado permite una comprensión empírica de los riesgos GNSS, abriendo la puerta a su utilización en contextos formativos o investigativos. El carácter documentado y replicable del sistema representa un aporte tangible al estudio académico de las amenazas que afectan al posicionamiento por satélite.

Próximos pasos:

Entre las líneas de trabajo futuras destacan varias mejoras técnicas: la automatización completa del entorno mediante contenedores, la incorporación de relojes disciplinados para mejorar la estabilidad, y la adaptación a bandas multiconstelación. También se contempla el desarrollo de una versión remota o en la nube que permita realizar experimentos sin necesidad de hardware local.

Desde el punto de vista educativo, el sistema puede evolucionar como herramienta docente en laboratorios GNSS, así como para la evaluación práctica de contramedidas y estrategias de mitigación. Esta plataforma versátil podría integrarse en programas de formación en ciberseguridad o ingeniería de comunicaciones críticas.

Master's Thesis Summary.

R.1 INTRODUCTION, MOTIVATION AND OBJECTIVES.

Main objective of the project:

The core purpose of this Master's Thesis is twofold: on one hand, to experimentally demonstrate the vulnerability of the civil GPS system to intentional interference; on the other, to provide a reproducible, affordable, and accessible platform that enables its study in academic environments. This approach goes beyond theoretical exploration by delivering a concrete, functional implementation using low-cost hardware and open-source tools. Thus, the project serves as a versatile instrument for both teaching and research in GNSS security, fostering practical understanding of the risks and evaluation of possible countermeasures.

Dual motivation of the project:

The motivation behind this work is structured around two complementary dimensions:

1. From an educational standpoint, the aim is to address the lack of accessible experimental environments in GNSS security. Most studies are theoretical, whereas this project allows for real-time observation of jamming and spoofing attacks on actual receivers, facilitating a tangible and comprehensive learning experience.

2. From a socio-technological perspective, the motivation stems from the growing reliance on GPS in critical sectors such as energy, finance, transportation, and telecommunications. Despite this dependence, civil GPS signals lack authentication mechanisms, making them vulnerable to interference even from low-cost equipment. The ease with which a basic attack setup can be replicated acts as a wake-up call to regulators, developers, and manufacturers, encouraging the adoption of more robust technologies like OSNMA or integration with inertial sensors. In this context, the project acts as a bridge between academia and industry, aligning with Sustainable

Development Goals related to quality education (SDG 4) and technological innovation (SDG 9).

Critical relevance of GPS:

Since its release for civilian use in 1993, GPS has evolved into a silent, essential infrastructure that underpins much of the digital economy and daily operations. Mobile communications, electric grid synchronization, real-time financial transactions, and air or maritime navigation rely on GPS to obtain highly precise time and location. The L1 C/A signal, used predominantly by civilian devices, operates in the L1 band (1,575 MHz) at extremely low power levels (≈ –160 dBW), making it particularly vulnerable to local interference.

Multiple international reports, including from the European Commission and NATO, have identified GPS as a "single point of failure" for critical infrastructures, warning that prolonged outages could have economic consequences in the billions of euros. This scenario gives the present work a strategic dimension by proposing a testbed to study, reproduce, and understand the risks inherent in this technological dependence.

Definition of jamming and spoofing:

Jamming and spoofing are two distinct forms of GPS interference, each with specific implications. Jamming involves emitting noise or interfering signals that raise the receiver's noise floor, preventing the correct acquisition and tracking of authentic signals. This leads to a loss of service, marked by a sudden drop in the C/N_0 parameter and the receiver switching to "No Fix" mode.

Spoofing, in contrast, aims to impersonate legitimate GPS signals by transmitting structurally valid but manipulated replicas. These signals, which emulate the C/A format and navigation data, can mislead the receiver into calculating incorrect position, time, or velocity without triggering internal alarms, making spoofing a more subtle and difficult threat to detect.

In this project, both types of attacks were implemented: jamming through the generation of

Gaussian noise via SDR, and spoofing through the replay of synthetic IQ frames generated with gps-sdr-sim, successfully displacing the receiver's estimated position by over 100 meters in a gradual and controlled manner.

R.2 METHODOLOGY, RESOURCES AND DEVELOPED TOOLS.

Hardware and software used:

The experimental setup was built using accessible hardware and open-source tools. The transmitter is a PlutoSDR (ADALM-PLUTO), connected via SMA attenuators to a passive patch antenna. Signal generation and control are managed from a laptop running Ubuntu 22.04.

The software stack includes GNU Radio 3.10 for designing jamming signal flows, gps-sdr-sim for spoofing signal generation, and GNSS-SDR as the software receiver to log parameters like C/N_0 and tracking losses. The entire system is designed to be replicable on a low budget without specialized hardware.

Development phases:

The project was structured in several phases: initially, the receiver was characterized under normal conditions. Then, a controlled environment was designed to prevent external emissions. Subsequently, jamming and spoofing attacks were implemented and their effects on the signal observed. Finally, results were analyzed using GNSS-SDR logs.

Jamming tool:

The jamming tool uses the PlutoSDR to generate wideband Gaussian noise centered at 1575.42 MHz. The signal is created in GNU Radio and transmitted via libiio libraries. Output power is controlled with attenuators, raising the noise floor and saturating the receiver's correlators.

The interference causes a sharp drop in C/N_0 and a loss of GPS fix. Once the transmission stops, the receiver takes 10 to 15 seconds to reacquire the signal.

Spoofing tool:

The spoofing system is based on gps-sdr-sim, which generates an IQ file with valid navigation data following a circular trajectory of about 150 meters. This file is transmitted in a loop from the PlutoSDR with a slightly higher gain than the real signal, allowing for progressive signal capture.

Once the spoofed signal is acquired, the receiver calculates false positions along the synthetic trajectory without raising alarms. The process mimics an authentic signal and can displace the receiver by over 100 meters without visible signs of failure.

R.3 CONCLUSIONS AND NEXT STEPS.

Contribution to the Sustainable Development Goals (SDGs):

The project aligns primarily with SDG 4 (Quality Education), by providing a hands-on learning tool for GNSS security; and with SDG 9 (Industry, Innovation and Infrastructure), by promoting the development of resilient systems from an open and experimental perspective. It also contributes to SDG 10 (Reduced Inequalities), by democratizing access to advanced tools without requiring significant resources.

Main conclusions:

The project has practically and reproducibly demonstrated the vulnerability of the civil GPS system to active interference. Both jamming and spoofing were successfully implemented with affordable means in a controlled environment, achieving significant impact on the receiver without the need for specialized equipment. This confirms the fragility of a widely used critical infrastructure whose signal can be neutralized or spoofed even from a domestic setup.

Beyond the technical achievement, the designed environment enables empirical understanding of GNSS threats, making it valuable for educational and research applications. Its well-documented and replicable design represents a meaningful contribution to academic studies on satellite-based positioning security.

Next steps:

Future improvements include full automation of the setup using containers, incorporation of disciplined oscillators for better signal stability, and adaptation to multi-constellation bands. Development of a remote or cloud-based version is also being considered to allow experiments without local hardware.

From an educational perspective, the system could evolve into a teaching tool for GNSS labs, as well as a platform for testing mitigation strategies and countermeasures. It could be integrated into cybersecurity or critical communications training programs.

Índice de la memoria

Capítulo 1 – Introducción. ***14***

1.1 Motivación del proyecto. 14

1.1.1 Demostrar la fragilidad del GPS. *14*

1.1.2 Entorno replicable para fines académicos. *15*

1.2 Objetivos y alcance. 16

1.2.1 Objetivo del proyecto. *17*

1.2.2 Alcance, entregables y limitaciones. *18*

1.3 Contexto y relevancia del GPS. 19

1.3.1 Evolución histórica y adopción global. *19*

1.3.2 Presencia en la vida cotidiana. *21*

1.3.3 Relevancia crítica para infraestructuras y economía. *22*

1.4 Fundamentos de funcionamiento del GPS. 24

1.4.1 Estructura del sistema. *24*

1.4.2 Señal L1 C/A: modulación, potencia y mensaje. *26*

1.4.3 Datos de navegación: efemérides y almanaque. *27*

1.4.4 Adquisición y seguimiento. *28*

1.4.5 Cálculo de pseudodistancias y solución 3D+T. *29*

1.4.6 Sincronización temporal: GPS Time, UTC y PPS. *30*

1.4.7 Vulnerabilidades inherentes. *30*

Capítulo 2 – Estado del arte. ***32***

2.1 Estado del arte de las interferencias de GPS. 32

2.1.1 Estado del arte para las técnicas de jamming. *32*

2.1.2 Estado del arte para las técnicas de spoofing. *34*

2.2 Estado del arte de la defensa y detección de interferencias. 38

Capítulo 3 - Metodología y recursos ***41***

3.1 Forma de trabajo y gestión. 41

3.2 Fases de evolución del trabajo. 43

3.3 Hardware empleado. 45

3.4 Software empleado. 48

Capítulo 4 - Desarrollo del sistema de interferencias por Jamming ***50***

4.1 Revisión de técnicas de Jamming. 50

4.2 Arquitectura del jammer. 52

4.3 Herramientas de software y hardware elegidas. 54
4.4 Implementación. 57
4.4.1 Puesta a punto de hardware y software. 57
4.4.2 Diseño del flujo GNU radio. 60
4.5 Verificación de resultados. 62
4.5.1 Confirmación espectral de la emisión. 63
4.5.2 Impacto real en un dispositivo móvil 64

***Capítulo 5 - Desarrollo del sistema de interferencias por Spoofing.* 68**
5.1 Revisión de técnicas de Spoofing. 68
5.2 Arquitectura del spoofer. 70
5.3 Herramientas de hardware y software elegidas. 72
5.4 Desarrollo de los subsistemas. 75
5.4.1 Generador de señal -gps-sdr-sim. 75
5.4.2 Receptor software -gnss-sdr. 83
5.5 Integración de los subsistemas para formar el spoofer. 91
5.5.1 Flujo operativo integrado. 91
5.5.2 Parámetros que deben permanecer alineados. 93
5.5.3 Consejos prácticos y problemas habituales. 95
5.6 Verificación de resultados. 97

***Capítulo 6 - Próximos pasos y líneas de trabajo.* 102**
6.1 Mejoras del entorno a corto plazo. 102
6.2 Mejoras del entorno a largo plazo. 103
6.3 Aplicación docente y educativa. 105
6.4 Plataforma para evaluar contramedidas. 106

***Capítulo 7 - Contribución a los objetivos de desarrollo sostenible (ODS).* 108**

***Capítulo 8 - Conclusiones.* 110**
8.1 Fragilidad del GPS demostrada. 110
8.2 Resumen de logros. 111
8.3 Impacto económico. 113

Índice de figuras

Figura 1 Primer satélite GPS lanzado en 1978. 20
Figura 2 Ilustración de la red de satélites que forman el GPS. 25
Figura 3 Ilustración del sistema de codificación para GPS. 26
Figura 4 Ilustración de la aplicación de los códigos PRN. 28
Figura 5 Ilustración del funcionamiento de la técnica meaconing. 35
Figura 6 Aterrizaje de emergencia en Newark, provocado por un jammer. 37
Figura 7 Ejemplo de funcionamiento de antena CRPA 39
Figura 8 SDR, antenas y conectores utilizados. 47
Figura 9 Montaje del jammer. 53
Figura 10 PlutoSDR en Administración de Dispositivos. 58
Figura 11 Bloques para PlutoSDR dentro de GNU Radio. 59
Figura 12 Esquema GNU Radio para el jamming. 60
Figura 13 Espectro antes de la emisión de ruido. 63
Figura 14 Espectro una vez se está emitiendo ruido en la frecuencia GPS. 64
Figura 15 Fix con GPS Data y modo avión cuando no hay jamming. 65
Figura 16 Fix con GPS Data y modo avión con jamming. 66
Figura 17 Apple Mapas incapaz de triangular la posición debido al jamming. 67
Figura 18 Montaje del spoofer. 71
Figura 19 HackRF y PlutoSDR utilizados en el proyecto. 73
Figura 20 Comprobación de la correcta instalación de gps-sdr-sim. 78
Figura 21 Ejemplo de ejecución de gps-sdr-sim de forma digital. 79
Figura 22 Fórmula para el cálculo del tamaño del fichero de datos sintéticos. 79
Figura 23 Transmisión de los datos a través del PlutoSDR con plutoplayer. 81
Figura 24 Comprobación de la conexión del PlutoSDR con el PC. 82
Figura 25 Recurso oficial de la NASA para obtener efemérides por día. 95
Figura 26 Inicialización de la recepción de GNSS-SDR a través del HackRF. 98
Figura 27 Comienzo de la identificación y enganche de los canales GPS. 99
Figura 28 GNSS-SDR comienza a obtener los mensajes NAV del spoofing. 100
Figura 29 GNSS-SDR obtiene la PVT emitida por gps-sdr-sim. 101

CAPÍTULO 1 – INTRODUCCIÓN.

Esta sección introductoria se estructura en torno a cuatro ejes principales. En primer lugar, se presenta la doble motivación que impulsa el proyecto. A continuación, se exponen los objetivos y el alcance del trabajo. Finalmente, se contextualiza la relevancia del sistema *Global Positioning System* (GPS) en la actualidad y se describen brevemente los fundamentos de su funcionamiento, con el fin de facilitar la comprensión de las interferencias que serán objeto de estudio.

1.1 MOTIVACIÓN DEL PROYECTO.

El proyecto se fundamenta en una motivación doble: por un lado, demostrar de forma práctica la vulnerabilidad de las señales GPS frente a interferencias intencionadas; por otro, proporcionar un entorno experimental reproducible y asequible que permita estudiar estas amenazas desde una perspectiva académica.

1.1.1 DEMOSTRAR LA FRAGILIDAD DEL GPS.

La señal GPS presenta una fragilidad inherente que la convierte en un objetivo vulnerable frente a interferencias, tanto accidentales como intencionadas. Esta debilidad deriva, en primer lugar, de la extremadamente baja potencia con la que las señales llegan a la superficie terrestre, típicamente en el orden de −160 dBW. En tales condiciones, cualquier fuente de ruido electromagnético próxima —incluso de bajo coste y potencia limitada— puede bastar para anular completamente la recepción, dejando al receptor sin acceso a información de posicionamiento o sincronización.

Este carácter crítico se agrava por la dependencia masiva de infraestructuras clave —banca, redes eléctricas, aviación, logística y telecomunicaciones— de los servicios de sincronización y localización proporcionados por el GPS. La pérdida o corrupción de esta

señal no solo afecta a terminales individuales, sino que puede desencadenar efectos en cascada con impacto sistémico, dado que el GPS actúa como un punto único de fallo global.

La amenaza no es meramente teórica. Existen múltiples precedentes documentados que evidencian la capacidad operativa actual de ataques contra GPS. Entre los casos más notorios se encuentra la captura del dron estadounidense RQ-170 por parte de Irán en 2011, atribuida a un ataque de *spoofing*. Más recientemente, en 2022, numerosos buques en el mar Negro reportaron ubicaciones imposibles, como resultado de una campaña de suplantación de señal ampliamente difundida. Estos eventos subrayan la urgencia de abordar la seguridad del sistema.

Una de las principales debilidades técnicas que facilita estas interferencias es la ausencia de mecanismos de autenticación en la señal civil L1 C/A (*Coarse Adquisition*), que sigue siendo la más utilizada. Esta carencia impide que los receptores distingan entre señales legítimas y transmisiones falsas generadas con mayor potencia, abriendo la puerta tanto a la denegación de servicio (*jamming*) como a la suplantación de información (*spoofing*).

A partir de esta constatación, la motivación principal de este proyecto es demostrar de forma práctica y rigurosa hasta qué punto resulta sencillo vulnerar la integridad de los sistemas GPS civiles mediante interferencias activas. Se pretende así contribuir al conocimiento de esta problemática desde un enfoque experimental, subrayando la necesidad de incorporar mecanismos de autenticación y detección robustos en futuras implementaciones del sistema.

1.1.2 ENTORNO REPLICABLE PARA FINES ACADÉMICOS.

La segunda motivación principal de este trabajo es la posibilidad de que la memoria funcione como una guía práctica y accesible que permita a otros investigadores, docentes o estudiantes replicar el entorno experimental desarrollado. Esta dimensión divulgativa y metodológica del proyecto responde a la necesidad de contar con plataformas abiertas, de bajo coste y fácilmente reproducibles que faciliten el estudio académico de las interferencias sobre señales *Global Navigation Satellite System* (GGNSS), así como la evaluación de contramedidas en contextos reales.

Frente a la dependencia de simuladores comerciales, que suelen implicar inversiones elevadas y restricciones de acceso, el entorno propuesto se caracteriza por su portabilidad, sencillez y economía. Todos los elementos utilizados son asequibles y de uso común en el ámbito de la radio definida por software (*Software Defined Radio*, SDR), y las herramientas empleadas son de código abierto. Esta elección no solo reduce la barrera de entrada para quienes deseen explorar estas temáticas, sino que promueve un enfoque colaborativo en el que los avances pueden compartirse, reproducirse y ampliarse fácilmente.

El hecho de que este sistema pueda montarse rápidamente en un entorno académico sin necesidad de hardware especializado, utilizando simplemente un ordenador portátil y unos pocos dispositivos SDR, tiene implicaciones directas para la docencia en seguridad GNSS. Permite ilustrar de forma tangible conceptos que de otro modo resultarían abstractos, y ofrece un banco de pruebas ideal para desarrollar, evaluar y comparar diferentes estrategias de protección frente a ataques de *jamming* y *spoofing*. Además, la documentación detallada del proceso favorece la transparencia científica y la repetibilidad de los ensayos, dos principios fundamentales en cualquier investigación rigurosa.

En definitiva, este enfoque busca contribuir a una democratización del estudio de las amenazas GNSS, ampliando el acceso a recursos experimentales que tradicionalmente han estado restringidos a centros con elevado presupuesto. El valor añadido de este proyecto no reside únicamente en las soluciones técnicas que se presentan, sino en el potencial que ofrece para formar, investigar y desarrollar nuevas defensas desde una base empírica y replicable.

1.2 OBJETIVOS Y ALCANCE.

En este apartado se exponen los objetivos del proyecto, tanto desde una perspectiva general como en términos concretos y medibles. Además, se detalla el alcance del trabajo y las principales limitaciones, muchas de ellas autoimpuestas por motivos éticos, técnicos o de

seguridad. Finalmente, se enumeran los entregables clave que resumen los resultados obtenidos.

1.2.1 OBJETIVO DEL PROYECTO.

El presente trabajo tiene como objetivo principal demostrar, de forma experimental, la fragilidad del sistema civil GPS frente a interferencias intencionadas, utilizando para ello un entorno accesible, económico y replicable. Esta aproximación no solo pone de manifiesto los riesgos inherentes a una tecnología crítica en múltiples sectores, sino que también proporciona una herramienta concreta para su análisis académico. La finalidad última es ofrecer una plataforma práctica que permita comprender, enseñar y mitigar las amenazas GNSS desde una perspectiva empírica y fundamentada.

Con el fin de acotar y hacer operativos los fines del proyecto, se han definido los siguientes objetivos específicos:

- O1 – Reproducir en un entorno doméstico un ataque de *jamming* capaz de interrumpir el posicionamiento (*fix*) del receptor, demostrando así el impacto inmediato que puede tener una señal de interferencia incluso a baja potencia.
- O2 – Ejecutar un ataque de *spoofing* que consiga introducir un error de posicionamiento sin que el receptor emita alertas ni detecte anomalías, lo que permite evaluar la eficacia del engaño bajo condiciones controladas.
- O3 – Documentar de manera completa todos los *scripts*, configuraciones y pasos necesarios para repetir los ataques de *jamming* y *spoofing*, con el objetivo de asegurar la reproducibilidad del experimento en entornos académicos o de investigación.
- O4 – Elaborar un informe técnico detallado en el que se exponga de forma clara la metodología utilizada, los resultados obtenidos y una serie de buenas prácticas orientadas a la mejora de la seguridad en entornos GNSS vulnerables.

1.2.2 ALCANCE, ENTREGABLES Y LIMITACIONES.

El alcance del trabajo abarca la generación, transmisión y monitorización de señales GNSS en la banda civil L1 C/A, en un entorno de pruebas estrictamente controlado. Las pruebas se realizan en interiores, como aulas o laboratorios, con una cobertura espacial inferior a los 10 metros y utilizando antenas pasivas que limitan la radiación. Las potencias de emisión se mantienen por debajo de los −30 dBm EIRP (*Effective Isotropic Radiated Power*), garantizando el cumplimiento de los márgenes legales y evitando cualquier interferencia externa no deseada. [1]

Las herramientas empleadas en el desarrollo del entorno incluyen dispositivos de radio definida por software como PlutoSDR y HackRF One, así como un conjunto de aplicaciones *open-source* tales como GNSS-SDR, gps-sdr-sim, GNU Radio y *scripts* en Python. Dado que estos elementos serán descritos con mayor profundidad en la sección metodológica, en este apartado solo se hace referencia a su papel general como soporte del sistema experimental. [2]

Las limitaciones del proyecto, definidas para asegurar un desarrollo seguro, legal y centrado, se resumen a continuación:

- Solo se trabaja con señales de la banda civil L1 C/A, quedando excluidas otras bandas como L2 o L5, así como las señales cifradas o de acceso restringido de cualquier constelación.
- No se consideran ni implementan contramedidas avanzadas, técnicas de mitigación activas ni el uso de hardware criptográfico o profesional destinado a la protección frente a interferencias.
- Todas las pruebas se limitan a entornos interiores y controlados; no se realiza ninguna emisión al aire libre, y se evita cualquier configuración que requiera licencia de transmisión o que pudiera afectar a receptores no participantes.

- La duración de las interferencias está limitada a un máximo de 60 segundos por ensayo, excluyendo de forma explícita cualquier forma de *jamming* o *spoofing* continuo o prolongado.

Como entregables principales del proyecto se incluyen esta memoria técnica, que detalla de forma estructurada y comprensible el proceso de implementación del entorno experimental, así como una serie de vídeos demostrativos que ilustran los ataques realizados y sus efectos sobre los receptores GPS. Además, se acompaña el material con el código necesario para la configuración de los distintos dispositivos y herramientas software utilizadas, con el fin de garantizar la reproducibilidad completa del sistema y facilitar su uso por parte de terceros.

Estas restricciones permiten acotar claramente el ámbito del proyecto, facilitando su replicabilidad en contextos académicos y garantizando un desarrollo responsable y alineado con los principios de la investigación experimental.

1.3 CONTEXTO Y RELEVANCIA DEL GPS.

El GPS ha pasado de ser un sistema militar para convertirse en un pilar invisible de la vida moderna. Su uso masivo en sectores como el transporte, la energía o las telecomunicaciones lo convierte en un protocolo crítico cuya integridad es vital para la sociedad.

1.3.1 EVOLUCIÓN HISTÓRICA Y ADOPCIÓN GLOBAL.

El GPS surgió en el marco de la Guerra Fría como un proyecto estratégico del Departamento de Defensa de los Estados Unidos, bajo el nombre de NAVSTAR (*Navigation System with Time And Ranging).* Su objetivo inicial era proporcionar a las fuerzas armadas —especialmente a la aviación y la marina— una capacidad de navegación precisa, continua y global. El primer satélite fue lanzado en 1978, marcando el inicio de una constelación que, con el tiempo, permitiría la cobertura planetaria; y lo podemos ver en la Figura 1. Durante sus primeras dos décadas de funcionamiento, la señal civil se ofrecía con una precisión deliberadamente degradada mediante la técnica de *Selective Availability*, lo que reservaba la

máxima exactitud para el uso militar. Este esquema cambió radicalmente el 1 de mayo del año 2000, cuando el presidente Bill Clinton ordenó la desactivación de dicha degradación. A partir de ese momento, la precisión de los receptores civiles mejoró de manera inmediata y significativa, pasando de errores del orden de decenas de metros a solo unos pocos, lo que desencadenó una auténtica revolución tecnológica y social.

Figura 1 Primer satélite GPS lanzado en 1978.

En menos de una década, el GPS pasó de requerir dispositivos voluminosos y especializados a estar integrado en chips de tamaño milimétrico, aptos para su inclusión en productos de consumo masivo. Un hito representativo fue el lanzamiento del iPhone 3G en 2008, que popularizó el uso del GPS en teléfonos inteligentes. Actualmente, es habitual encontrar receptores GNSS —que combinan señales de GPS con otros sistemas como Galileo (Europa), GLONASS (Rusia) o BeiDou (China)— en una amplia gama de dispositivos: teléfonos móviles, relojes inteligentes, vehículos, cámaras, drones y aplicaciones industriales o científicas. Este crecimiento ha sido posible gracias a factores clave como la gratuidad del servicio, financiado en su totalidad por el gobierno estadounidense, su cobertura global ininterrumpida y la interoperabilidad con otros sistemas de navegación por

satélite. Según estimaciones, en 2025 habrá más de seis mil millones de receptores GNSS activos en todo el mundo, una cifra que duplica la población global y refleja el grado de integración de esta tecnología en la vida cotidiana.

1.3.2 PRESENCIA EN LA VIDA COTIDIANA.

Aunque muchas personas asocian el GPS únicamente con las aplicaciones de navegación en el móvil, como el clásico "punto azul" que indica la ubicación en tiempo real, su presencia en la vida cotidiana es mucho más profunda y variada. En el sector del transporte, por ejemplo, los servicios de movilidad a demanda —como taxis digitales o vehículos compartidos— dependen del GPS para asignar conductores a usuarios, calcular rutas óptimas y ajustar tiempos de llegada con notable precisión. Igualmente, las empresas logísticas emplean datos GNSS para optimizar rutas de reparto, reducir el consumo de combustible y mejorar la trazabilidad de sus envíos.

En el ámbito del deporte y la salud, aplicaciones de ejercicio físico monitorizan recorridos y distancias con resoluciones que pueden llegar a escasos metros, lo que permite al usuario evaluar su rendimiento con un alto grado de fiabilidad. Del mismo modo, los drones recreativos y semiprofesionales se apoyan en datos de efemérides GNSS para mantener trayectorias estables, ejecutar vuelos programados o regresar de forma automática al punto de despegue, todo ello sin intervención humana directa.

En el sector agrícola, el desarrollo de la agricultura de precisión ha supuesto un salto cualitativo gracias al uso de técnicas como el posicionamiento cinemático en tiempo real (RTK, *Real-time kinematic*), que mejora la precisión del GPS hasta el rango de centímetros. Esto permite sembrar, abonar o cosechar con una eficiencia inédita, reduciendo tanto el uso de fertilizantes como el gasto de combustible y el impacto ambiental.

Incluso en entornos domésticos, el GPS se ha integrado en soluciones cotidianas: etiquetas inteligentes que combinan Bluetooth y GNSS permiten localizar objetos como llaves, mochilas o incluso mascotas con gran rapidez. En todos estos casos, la señal L1 civil, aunque invisible y silenciosa, actúa como una infraestructura crítica que permite el funcionamiento

fluido de múltiples servicios y dispositivos que conforman nuestra vida diaria. La ubicuidad del GPS lo convierte así en un pilar esencial de la sociedad digital contemporánea.

1.3.3 RELEVANCIA CRÍTICA PARA INFRAESTRUCTURAS Y ECONOMÍA.

La red de satélites GPS forma hoy una infraestructura silenciosa, omnipresente y crítica que sostiene la sincronización de numerosos sistemas invisibles para el ciudadano común. Su señal gratuita, continua y global no solo permite determinar ubicaciones con una precisión métrica, sino que proporciona un pulso temporal de precisión nanosegundos que sirve de referencia para todo el ecosistema digital moderno. Esta sincronización invisible se ha convertido, sin apenas percepción social, en el latido común que ordena desde las comunicaciones móviles hasta las operaciones bursátiles, la gestión de redes eléctricas o la seguridad del transporte aéreo.

Uno de los sectores más dependientes es el de las telecomunicaciones. Cada estación base de una red 4G o 5G sincroniza sus tramas y realiza *hand-offs* basándose en el pulso por segundo (PPS, *Pulse per second*) derivado del GPS. Un desfase superior a unos pocos microsegundos puede provocar cortes de llamada, pérdida de celdas y una degradación significativa en la capacidad de toda la red. Esta dependencia es tal que, sin GPS, la red entra rápidamente en colapso operativo.

En el ámbito financiero, la normativa europea MiFID II exige que todas las transacciones electrónicas se marquen con una precisión inferior a 100 microsegundos respecto a UTC (*Universal Time Coordinated)*. La forma más económica de cumplir este requisito consiste en instalar una antena GPS en los centros de datos. La pérdida o alteración de esta referencia temporal no solo comprometería la trazabilidad legal de las operaciones, sino que podría suspender automáticamente la cotización de valores y generar pérdidas millonarias por interrupciones en los sistemas de trading algorítmico. [3].

La infraestructura eléctrica también está profundamente anclada al GPS. Las unidades de medida fasorial (PMU, *Phasor Measurement Unit*), instaladas en nodos clave de las redes de alta tensión, emplean su señal temporal para medir fase y frecuencia con alta resolución.

Esta sincronización es esencial para detectar oscilaciones peligrosas, coordinar respuestas ante fallos y prevenir apagones en cascada. Una desincronización por interferencia o sabotaje pone en riesgo la estabilidad de todo el sistema energético.

En el transporte, especialmente en la aviación, la dependencia es crítica. Los sistemas de navegación basada en prestaciones (PBN, *Performance-Based Navigation*) y de vigilancia automática dependiente (ADS-B, *Automatic Dependent Surveillance-Broadcast*) requieren la señal GPS para mantener las separaciones seguras entre aeronaves. La Administración Federal de Aviación (FAA) mantiene aún una red de radioayudas VOR (*Very High Frequency Omnidirectional Range)* como respaldo, precisamente por el temor a una pérdida súbita de señal GNSS. Este temor ya tiene base empírica: un incidente reciente de interferencia en el entorno de Kaliningrado afectó a más de 1.600 vuelos, y en marzo de 2024 se registraron más de 41.000 eventos de *jamming* o *spoofing* a nivel mundial, con un promedio de 700 incidentes diarios. [4]

Las consecuencias económicas de un fallo del sistema GPS han sido objeto de diversos estudios prospectivos. En Estados Unidos, el Departamento de Comercio estima que un apagón total durante la temporada agrícola supondría una pérdida de hasta 1.000 millones de dólares diarios en el PIB (Producto Interior Bruto) nacional. En Reino Unido, un corte de cinco días afectando a todas las constelaciones GNSS implicaría daños directos y pérdidas económicas superiores a 5.200 millones de libras. A escala global, un apagón durante más de una semana podría situarse en el rango del billón de dólares. [5][6]

Este panorama pone de manifiesto una paradoja inquietante: mientras la economía digital y las infraestructuras críticas confían ciegamente en una señal extremadamente vulnerable, apenas existen mecanismos de respaldo efectivos. El GPS se ha convertido en un "*single point of failure"* de alcance sistémico. Por ello, este trabajo de fin de máster persigue no solo evidenciar esa fragilidad mediante experimentación práctica, sino también sentar las bases para un entorno educativo y replicable donde se puedan evaluar contramedidas de forma segura y asequible. Demostrar la gravedad potencial de una interferencia intencionada o

accidental no es solo una contribución académica, sino una llamada urgente a reforzar la resiliencia tecnológica de nuestras infraestructuras vitales.

1.4 FUNDAMENTOS DE FUNCIONAMIENTO DEL GPS.

Comprender los principios técnicos que rigen el funcionamiento del GPS es esencial para valorar su vulnerabilidad. Este apartado expone los aspectos clave del sistema, desde la estructura de su señal hasta el proceso de posicionamiento y sincronización.

1.4.1 ESTRUCTURA DEL SISTEMA.

El GPS se articula en torno a tres segmentos fundamentales que interactúan de forma continua para proporcionar información precisa de localización y tiempo. Estos segmentos son el espacial, el de control y el de usuario.

El segmento espacial está compuesto por una constelación de aproximadamente 31 satélites distribuidos en órbitas medias terrestres (MEO, *Medium Earth Orbit*), a una altitud de unos 20.200 kilómetros, como podemos ver en la ilustración de la Figura 2. Cada uno de estos satélites está equipado con relojes atómicos de alta precisión, cuya estabilidad alcanza el orden de 10^-14, y transmite señales de navegación en múltiples frecuencias. La más relevante para aplicaciones civiles es la portadora L1 (1575,42 MHz), que incluye el mensaje de navegación necesario para el posicionamiento. [7]

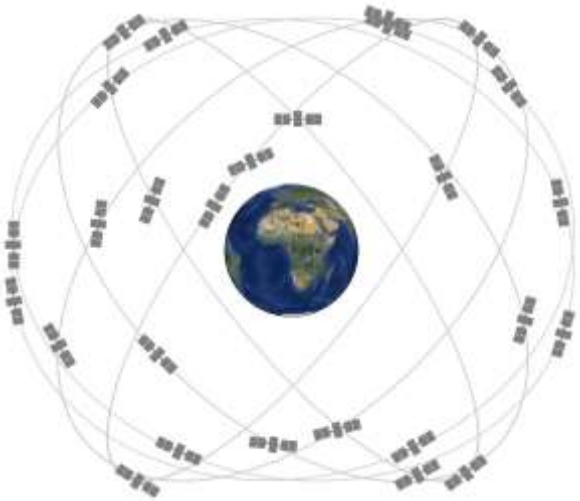

Figura 2 Ilustración de la red de satélites que forman el GPS.

El segmento de control, por su parte, está conformado por una red global de estaciones terrestres que incluyen estaciones monitoras, estaciones de subida y un centro de control maestro ubicado en Colorado (EE.UU.). Estas instalaciones se encargan de calcular las efemérides de los satélites y de corregir regularmente sus relojes, garantizando así la integridad y exactitud del sistema. Las actualizaciones de tiempo y órbita se realizan de manera periódica, cada pocas horas, para minimizar errores y mantener la coherencia del servicio.

Por último, el segmento de usuario engloba todos aquellos receptores capaces de captar la señal L1 y decodificar el mensaje de navegación. Estos dispositivos, presentes en teléfonos móviles, vehículos, aeronaves y una gran variedad de aplicaciones industriales y científicas, procesan las señales recibidas de múltiples satélites para calcular su posición y sincronización temporal mediante un proceso de trilateración.

La comprensión de la interacción entre estos tres segmentos resulta esencial en el contexto del presente proyecto, ya que los ataques de interferencia propuestos —*jamming* y *spoofing*— se sitúan precisamente en el vínculo entre el segmento espacial y el segmento de usuario. Esto implica que, aunque los satélites y el control continúan operando correctamente, los receptores pueden ser engañados o inutilizados si las señales que reciben

son alteradas o bloqueadas intencionadamente, poniendo de manifiesto una de las vulnerabilidades clave del sistema GPS.

1.4.2 SEÑAL L1 C/A: MODULACIÓN, POTENCIA Y MENSAJE.

La señal L1 C/A constituye el canal principal de comunicación entre los satélites GPS y los receptores civiles. Emitida en la frecuencia de 1575,42 MHz, esta señal utiliza una modulación BPSK-R(1) (*Binary Phase Shift Keying, Rate 1*), que incorpora un código de acceso C/A de 1023 chips, transmitido a una velocidad de 1,023 Mchip/s. Este código se repite cada milisegundo, permitiendo la sincronización y la identificación del satélite emisor. Vemos un ejemplo de este proceso de codificación en la Figura 3. [8]

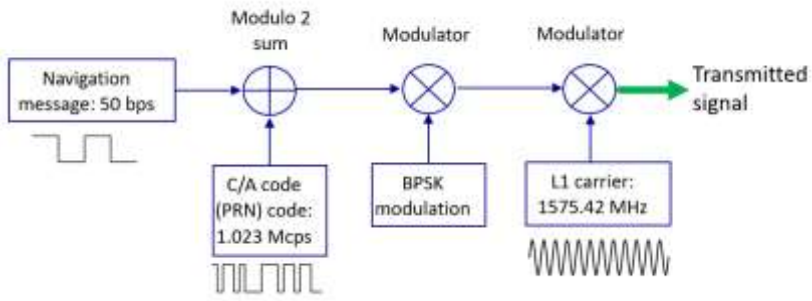

Figura 3 Ilustración del sistema de codificación para GPS.

Superpuesto a este código de propagación se encuentra el mensaje de navegación, transmitido a una tasa de 50 bits por segundo. Dicho mensaje contiene información crítica como las efemérides del satélite, los parámetros de corrección del reloj y datos sobre la constelación, necesarios para el cálculo preciso de la posición.

Una de las características más relevantes de esta señal es su extremadamente baja potencia al llegar al receptor. En condiciones típicas, el nivel de potencia recibido en la antena de un teléfono móvil se sitúa en torno a los −158 dBW, lo que implica una relación señal-ruido del orden de −15 dB respecto al umbral térmico. Esta situación convierte la detección de la señal

en un desafío técnico considerable, que solo puede resolverse mediante técnicas de correlación propias de los sistemas de espectro ensanchado.

Esta debilidad intrínseca, sin embargo, representa también una oportunidad para ataques de *spoofing*. Un emisor SDR con una potencia de salida del orden de −110 dBW a apenas tres metros del receptor puede superar fácilmente la señal legítima, introduciendo una falsa señal GPS que será procesada con preferencia. Este reducido margen de diferencia entre la señal real y la interferente explica por qué es posible suplantar la señal de navegación con equipos de coste reducido, lo que pone de relieve una vulnerabilidad crítica del sistema.

1.4.3 DATOS DE NAVEGACIÓN: EFEMÉRIDES Y ALMANAQUE.

Los datos de navegación transmitidos en la señal L1 C/A del sistema GPS constituyen el núcleo informativo que permite a los receptores calcular su posición y sincronización temporal. Estos datos se emiten a una velocidad de 50 bits por segundo y se estructuran en sub-tramas de 30 segundos, cada una compuesta por 1500 bits. Cada conjunto de cinco sub-tramas conforma un marco de datos de 150 segundos que se repite continuamente.

Las sub-tramas 1 a 3 contienen las efemérides, es decir, los parámetros orbitales precisos de cada satélite codificados según el modelo kepleriano. Estos datos permiten calcular con gran exactitud la posición instantánea del satélite, con un error típico inferior a 2 metros, siempre que las efemérides sean recientes, dado que su validez práctica se limita a un intervalo de unas cuatro horas desde su emisión.

Por su parte, las sub-tramas 4 y 5 albergan el almanaque, que ofrece una descripción resumida de la constelación completa de satélites. Aunque menos preciso que las efemérides, el almanaque proporciona información general sobre la posición y el estado de cada satélite, lo cual resulta útil en la fase inicial de adquisición del sistema.

En el contexto del presente proyecto, la manipulación de estos datos resulta esencial para recrear o suplantar señales GPS de forma coherente. Para ello, se extrae diariamente un fichero RINEX (*Receiver Independent Exchange Format*), que contiene las efemérides

actualizadas de los satélites. Este fichero se convierte al formato específico que emplea la herramienta gps-sdr-sim, permitiendo la generación de una señal GPS simulada que se alinea temporalmente con una hora GPS ficticia, asegurando así la consistencia y credibilidad de la señal emitida desde el SDR.

1.4.4 ADQUISICIÓN Y SEGUIMIENTO.

El proceso de adquisición y seguimiento es fundamental para que un receptor GPS determine con precisión la señal de cada satélite. Durante la fase de adquisición, el receptor analiza múltiples secuencias de códigos de acceso (PRN, *Pseudo-Random Noise*) correspondientes a los satélites activos. Para cada PRN, el receptor examina un espacio tridimensional definido por la variación del código (los 1023 chips del código C/A) y la desviación Doppler, que puede oscilar típicamente entre ±5 kHz debido al movimiento relativo entre el satélite y el receptor. Vemos un esquema extendido con los códigos PRN en la Figura 4.

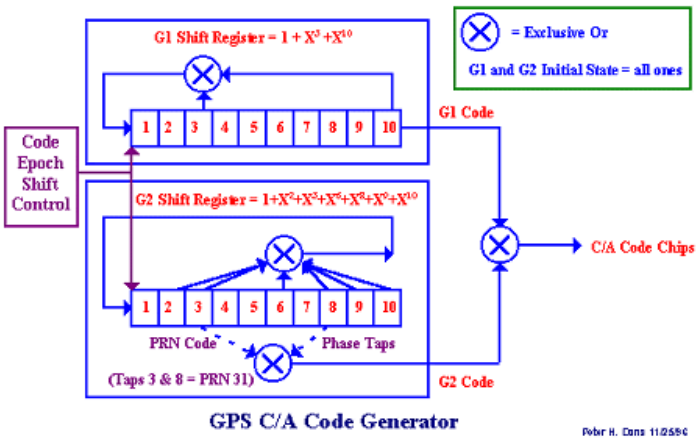

Figura 4 Ilustración de la aplicación de los códigos PRN.

El objetivo de esta búsqueda es encontrar el máximo de correlación entre la señal recibida y una réplica generada localmente, lo que indica que el receptor ha encontrado la señal del satélite correspondiente. Una vez identificados estos picos de correlación, el sistema pasa de la fase de adquisición a la de seguimiento, activando dos lazos de control: el DLL (*Delay Lock Loop*) y el PLL (*Phase Lock Loop*). El DLL se encarga de alinear con precisión el

código C/A, garantizando que la correlación se mantenga en el tiempo, mientras que el PLL ajusta la fase de la portadora para lograr una sincronización continua y precisa con la señal recibida.

Un ataque de *spoofing* eficaz debe tener en cuenta este funcionamiento. Para lograr que el receptor comience a seguir una señal falsa sin interrupciones perceptibles, el *spoofer* debe emitir una réplica de la señal GPS legítima que presente un desplazamiento Doppler creíble, coherente con la dinámica esperada del satélite. Además, la potencia de esta señal falsificada debe superar a la original por un margen de aproximadamente 2 a 3 dB. Este pequeño incremento es suficiente para que los lazos DLL y PLL del receptor comiencen a seguir la señal espuria, desplazándose suavemente desde la señal genuina sin que el usuario perciba la transición.

1.4.5 CÁLCULO DE PSEUDODISTANCIAS Y SOLUCIÓN 3D+T.

El posicionamiento GPS se basa en el cálculo de las pseudodistancias entre el receptor y al menos cuatro satélites visibles. Cada pseudodistancia ρ_i representa el tiempo de propagación de la señal multiplicado por la velocidad de la luz, e incluye tanto la distancia geométrica real R_i como el error de sincronización del reloj del receptor ($\delta t \cdot c$). Esto da lugar a un sistema de cuatro ecuaciones con cuatro incógnitas: latitud, longitud, altitud y el desfase temporal del reloj interno.

Para resolver este sistema, el receptor emplea un ajuste por mínimos cuadrados que minimiza el error entre las distancias medidas y las calculadas, obteniendo así una estimación óptima de su posición y del desfase temporal.

Precisamente este mecanismo permite la introducción de errores controlados si se manipulan adecuadamente los datos de entrada. Alterar las marcas temporales Z-count dentro de la sub-

trama de navegación, o introducir un retardo progresivo en el código C/A, puede inducir al receptor a calcular una posición o un tiempo erróneos.

1.4.6 SINCRONIZACIÓN TEMPORAL: GPS TIME, UTC Y PPS.

La sincronización temporal proporcionada por el sistema GPS es uno de sus servicios más críticos, especialmente para aplicaciones que requieren una elevada precisión temporal, como las redes de telecomunicaciones, los sistemas financieros o los entornos industriales automatizados. El tiempo GPS (GPS Time) se originó el 6 de enero de 1980 y, a diferencia del UTC, no contempla la introducción de segundos intercalares. Esta decisión simplifica el mantenimiento de una escala de tiempo continua, aunque introduce una diferencia creciente con respecto al UTC, que se corrige mediante la difusión periódica de la relación GPS-UTC cada 12,5 minutos dentro del mensaje de navegación.

A partir de esta información, el receptor GPS genera un PPS altamente preciso, cuya incertidumbre típica se sitúa por debajo de los 40 nanosegundos. Este pulso es utilizado como referencia temporal en múltiples sistemas: redes móviles para la sincronización de celdas, infraestructuras de mercados financieros conforme a la normativa MiFID II, y redes SCADA en automatización industrial.

Dada esta dependencia, incluso desviaciones mínimas en la señal temporal pueden tener consecuencias significativas. Un retardo de tan solo 1 microsegundo, equivalente a un desplazamiento de 300 metros en términos de código C/A, ya supone una violación de la precisión exigida por regulaciones como MiFID II y puede provocar una desalineación crítica.

1.4.7 VULNERABILIDADES INHERENTES.

La señal civil del sistema GPS presenta una serie de vulnerabilidades técnicas que la hacen especialmente susceptible a interferencias intencionadas. Una de las más relevantes es la ausencia de mecanismos de autenticación en las señales transmitidas. Esto significa que los

receptores no tienen forma de verificar si una señal recibida proviene efectivamente de un satélite legítimo o si ha sido generada de forma fraudulenta por un emisor externo.

Además, la señal L1 C/A llega a los receptores con una relación señal-ruido (SNR, *Signal to Noise Ratio*) negativa, es decir, por debajo del nivel de ruido térmico. Esto obliga a los dispositivos a recurrir a procesos de correlación muy sensibles para identificar y seguir la señal. En estas condiciones, el receptor adopta una lógica de aceptación basada en el primer código PRN que supere un umbral mínimo de correlación y plausibilidad, bajo la premisa de que el entorno radioeléctrico no ha sido comprometido. [9]

CAPÍTULO 2 – ESTADO DEL ARTE.

2.1 ESTADO DEL ARTE DE LAS INTERFERENCIAS DE GPS.

En esta sección se presenta una revisión del estado del arte en materia de interferencias intencionadas sobre sistemas GNSS, con especial atención a las técnicas de *jamming* y *spoofing*. Se describen las modalidades más representativas de cada tipo de ataque, sus principios de funcionamiento, y sus principales parámetros característicos, como coste, alcance, complejidad técnica, magnitud del error inducido y facilidad de detección.

2.1.1 ESTADO DEL ARTE PARA LAS TÉCNICAS DE JAMMING.

El *jamming* es una técnica de interferencia deliberada cuyo objetivo es impedir que los receptores GNSS, como los del sistema GPS, puedan adquirir o seguir las señales satelitales necesarias para determinar su posición y sincronización. Esta obstrucción se logra mediante la emisión de señales radioeléctricas que degradan la SNR por debajo del umbral operativo del receptor, anulando así su funcionalidad.

Las técnicas actuales de *jamming* empleadas para interferir con señales GNSS, como las del sistema GPS, se dividen en varias categorías según el tipo de señal interferente y su grado de sofisticación. Estas técnicas varían considerablemente en términos de coste, complejidad de implementación, alcance efectivo y facilidad de detección.

• **Ruido de banda ancha (*barrage*):** Esta técnica se basa en la emisión de un ruido gaussiano que ocupa toda la banda L1 utilizada por el sistema GPS. Su finalidad es degradar la SNR de tal forma que el receptor no pueda adquirir ni seguir las señales de los satélites, quedando inutilizado para proporcionar información de posicionamiento. Se caracteriza por su sencillez de implementación, bajo coste (habitualmente entre 100 y 300 €), y una potencia de salida que puede variar entre 0 y 20 dBm. Su radio de acción depende del entorno y del tipo de antena empleada, alcanzando desde los 100 metros hasta varios kilómetros si se usa una antena direccional. La detección de este tipo de interferencia resulta sencilla, ya que

produce un aumento uniforme del nivel de ruido en todo el espectro afectado, perceptible como una "alfombra" plana en los analizadores de espectro.

• **Portadora continua (CW, *Continuous Wave* / tonal):** Consiste en la transmisión de una o varias señales sinusoidales muy estrechas en frecuencia, que saturan selectivamente determinadas celdas de la transformada rápida de Fourier (*Fast Fourier Transform,* FFT) del receptor GNSS. Esto puede impedir la detección de las señales legítimas si coinciden en frecuencia, o bien si se encuentran dentro del margen de ±500 Hz, dado que los lazos de adquisición del receptor quedan ciegos ante estos picos interferentes. Su coste típico se sitúa entre 150 y 400 €, con potencias de emisión que no suelen superar los 5 dBm. El alcance efectivo varía entre los 10 y 200 metros. Aunque su complejidad técnica es baja, su detección requiere un análisis espectral algo más cuidadoso, ya que se manifiesta como picos fijos en el espectro, cuya intensidad depende de la proximidad y potencia del emisor.

• ***Jamming* por barrido o *chirp*:** En esta variante, el emisor barre frecuencialmente la banda objetivo (por ejemplo, a razón de 1 MHz cada 4 microsegundos), generando una señal cuya frecuencia cambia continuamente. Esta estrategia dificulta la acción de los filtros *notch* que algunos receptores emplean para mitigar interferencias de frecuencia fija. El coste de estos sistemas suele oscilar entre 200 y 600 €, y sus potencias típicas se sitúan entre 0 y 10 dBm. Pueden alcanzar radios de acción de entre 50 metros y 2 kilómetros, dependiendo del entorno y de la ganancia de la antena. Su detección es moderadamente compleja, pero puede realizarse mediante el análisis de espectrogramas (*waterfall*), donde esta señal deja un trazo característico en forma de línea diagonal que recorre el espectro.

• ***Jamming* consciente del protocolo (*protocol-aware*):** Esta técnica representa el nivel más alto de sofisticación. El emisor genera una señal que replica el PRN del GPS, utilizando modulación DSSS (*Direct Sequence Spread Spectrum*) para simular una señal auténtica. El receptor percibe esta señal como válida, ya que los niveles de C/N_0 (es lo mismo que decir SNR) aparentan estar dentro de lo normal, pero el proceso de correlación no logra converger, impidiendo así la adquisición sin que el receptor emita ninguna alerta evidente. La implementación de esta técnica requiere equipamiento avanzado, como SDRs de laboratorio

y software especializado, con costes que pueden situarse entre los 5.000 y los 30.000 €. Las potencias de emisión suelen ser inferiores a 0 dBm, pero pueden alcanzar radios de actuación superiores a los 5 kilómetros gracias a la naturaleza "aparentemente válida" de la señal emitida. Su detección es sumamente compleja y exige un análisis detallado de los discriminadores internos del receptor.

2.1.2 ESTADO DEL ARTE PARA LAS TÉCNICAS DE SPOOFING.

El *spoofing* es una técnica de ataque que consiste en engañar al receptor GNSS mediante la transmisión de señales falsas que imitan las auténticas señales satelitales. A diferencia del *jamming*, que busca bloquear la recepción, el *spoofing* pretende hacer que el receptor calcule una posición, velocidad o tiempo erróneos, manteniendo aparentemente una operación normal. Las técnicas actuales de *spoofing* se pueden clasificar en función de su grado de sofisticación, coste y precisión, afectando a distintos parámetros del receptor como las pseudodistancias, la trayectoria calculada o la sincronización temporal.

• ***Meaconing*** **(re-radiado diferido):** Esta técnica captura la señal legítima mediante una antena, la amplifica en torno a 3 dB y la reemite tras un breve retardo, del orden de nanosegundos o milisegundos. El receptor tiende a "engancharse" a la señal más potente, que en este caso es la copia artificial, provocando un sesgo fijo en todas las pseudodistancias. El resultado es una posición errónea, típicamente dentro de un error inferior a 500 metros. Se trata de una técnica de complejidad baja, accesible mediante grabadores-reproductores GNSS comerciales con un coste entre 3,000 y 8,000 €, y un radio de actuación de hasta 1 km. La detección es moderada, pues el salto súbito en la ganancia automática (AGC, *Automatic Gain Control*) puede delatar la irrupción de la señal espuria. Para una mayor comprensión, se puede ver en la Figura 5 una ilustración del funcionamiento del *meaconing*.

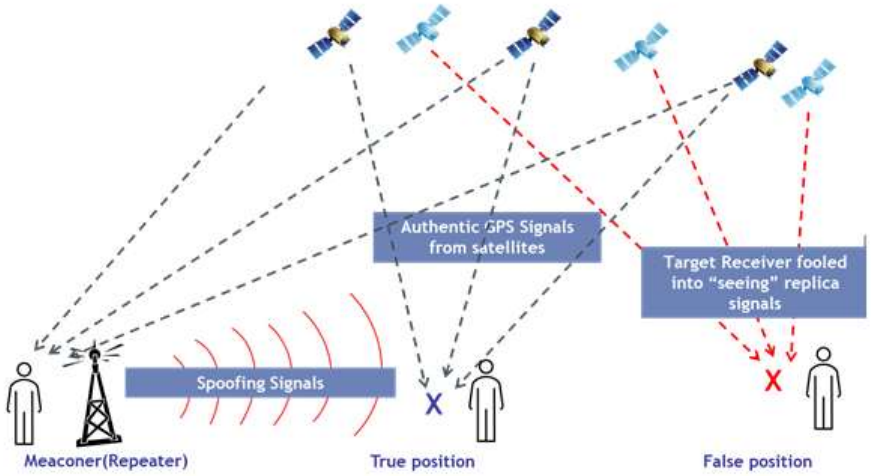

Figura 5 Ilustración del funcionamiento de la técnica meaconing.

• ***Carry-off* coherente:** En este enfoque, se genera una réplica de la señal legítima con sincronización perfecta y una potencia apenas 2–3 dB superior, suficiente para que el receptor cambie su seguimiento (*handover*) sin percibir la transición. A partir de ahí, el atacante arrastra suavemente al receptor desplazando la señal falsificada milímetro a milímetro, logrando alterar su trayectoria sin levantar alertas. La velocidad de desplazamiento suele mantenerse por debajo de 1 m/s para conservar la coherencia interna del sistema. Esta técnica presenta una complejidad media, requiere un generador vectorial de señales y software de guiado especializado, con un coste total estimado entre 10,000 y 25,000 €. Puede alcanzar distancias de entre 100 metros y 5 kilómetros, y su detección es baja debido a la continuidad y plausibilidad de la señal manipulada.

• ***Spoofing* generativo multicanal en tiempo real:** Los simuladores profesionales de constelaciones GNSS, como el Spirent GSS7000, son capaces de sintetizar en tiempo real todas las señales de los satélites visibles, controlando con precisión fases, efemérides y potencias. Esto permite construir escenarios altamente realistas, incluyendo eventos de pérdida de señal o transiciones dinámicas. Su precisión puede alcanzar el rango de los centímetros, y su alcance supera los 5 km si se emplea un amplificador externo. Sin embargo, su coste es extremadamente alto, superando los 90,000 € en equipos nuevos, o 4,500 € por

semana en régimen de alquiler. La complejidad de operación es elevada y su detección resulta muy difícil, incluso con sistemas avanzados.

• **Manipulación del mensaje de navegación:** Esta técnica no altera la fase ni el código de la señal, sino que introduce información errónea en el mensaje de navegación, como efemérides falsas o modificaciones del contador temporal. El receptor mantiene el seguimiento de la portadora, pero su cálculo de tiempo deriva progresivamente, generando errores de sincronización del orden de microsegundos por segundo. Aunque esta técnica requiere una manipulación precisa de las subtramas, su coste es comparable al del *spoofing* generativo multicanal. La detección es moderada, siendo posible mediante algoritmos de integridad como RAIM (*Receiver Autonomous Integrity Monitoring*), que pueden advertir inconsistencias en la geometría o en los residuos de pseudodistancia. [10]

2.1.3 EJEMPLOS ILUSTRATIVOS.

Con el fin de ilustrar la aplicabilidad real de las técnicas descritas y su impacto en entornos operativos, se presentan a continuación varios ejemplos representativos documentados en diferentes regiones del mundo. Estos casos permiten contextualizar las amenazas GNSS no sólo como hipótesis experimentales, sino como incidentes con consecuencias tangibles para infraestructuras críticas y seguridad operativa.

• Newark, Estados Unidos (2012–2013) – *Jamming* de barrido:

Un incidente prolongado en el aeropuerto internacional de Newark evidenció el poder disruptivo de un *jammer* de bajo coste. Durante meses, un conductor de camiones que transitaba cerca de la pista empleó un dispositivo de *jamming* por barrido valorado en apenas 150 €, lo que provocó interferencias sistemáticas en el sistema de aterrizaje asistido por satélite (GBAS, *Ground Based Augmentation System*) del aeropuerto, causando atterajes de emergencia como el que vemos en la Figura 6. La Administración Federal de Comunicaciones (FCC) rastreó la fuente y multó al responsable con 31,875$, subrayando la facilidad con la que se puede comprometer una instalación crítica utilizando tecnología accesible. [11]

Figura 6 Aterrizaje de emergencia en Newark, provocado por un jammer.

• Mar Negro / Sochi (2016–2022) – *Spoofing* tipo *carry-off:*

Diversos buques comerciales reportaron posiciones imposibles situadas hasta 32 kilómetros tierra adentro, revelando una alteración deliberada de las señales GNSS en el área costera de Sochi. El análisis posterior de la organización C4ADS (*Center for Advanced Defense Studies*) identificó un patrón de *spoofing* gradual consistente con la técnica de *carry-off*, en la que los receptores son arrastrados lentamente hacia ubicaciones falsas sin generar alertas internas. Este caso ejemplifica cómo el *spoofing* puede ser empleado de forma sistemática y sostenida para distorsionar la percepción espacial de múltiples objetivos de forma simultánea. [12]

• Región Báltica (2024) – *Jamming protocol-aware* a gran escala:

Entre agosto de 2023 y abril de 2024, más de 46,000 vuelos en Europa del Norte registraron interrupciones o alertas relacionadas con el sistema GNSS, afectando especialmente al Báltico. Las investigaciones apuntan a emisiones deliberadas desde estaciones en la región de Kaliningrado, compatibles con *jamming* consciente del protocolo. Este tipo de interferencia es difícil de detectar y neutralizar, y obligó en múltiples ocasiones a modificar rutas aéreas y reintroducir sistemas tradicionales de navegación. El incidente confirma la creciente escala y sofisticación de los ataques GNSS en contextos geopolíticamente tensos.

2.2 ESTADO DEL ARTE DE LA DEFENSA Y DETECCIÓN DE INTERFERENCIAS.

La detección y mitigación de interferencias GNSS ha cobrado una importancia creciente ante la proliferación de técnicas de *jamming* y *spoofing*. Diversas estrategias tecnológicas han sido desarrolladas para aumentar la resiliencia de los receptores, desde enfoques puramente algorítmicos hasta soluciones hardware de alta gama. A continuación se detallan las principales técnicas actuales de defensa, considerando su eficacia, coste, complejidad y estado de implementación.

• Recepción multiconstelación y multifrecuencia con RAIM:

El uso de receptores capaces de procesar señales de múltiples constelaciones (como GPS, Galileo, GLONASS y BeiDou) y varias bandas de frecuencia (L1, L5, etc.) es una de las defensas más efectivas y económicamente viables. Esta diversidad permite aplicar algoritmos de integridad como RAIM, que detectan inconsistencias entre pseudodistancias y descartan señales anómalas. Además de reducir la vulnerabilidad frente al *jamming* por diversificación espectral, proporciona una protección sólida frente al *spoofing* mediante la identificación de discrepancias geométricas. Los receptores compatibles, como los u-blox M10 o F9P, están disponibles por precios entre 80 y 150 €, y ofrecen mejoras de hasta +10 dB en el margen de señal, con tasas de detección de *spoofing* cercanas al 95 % en condiciones reales.

• Autenticación del mensaje de navegación (OSNMA, CHIMERA):

La incorporación de firmas digitales en los mensajes de navegación, como propone Galileo OSNMA (*Open Service Navigation Message Authentication*) o el futuro CHIMERA en GPS L1, permite verificar la autenticidad de los datos recibidos. Esto bloquea ataques basados en manipulación del mensaje o *meaconing*, al invalidar señales que no puedan ser autenticadas criptográficamente. No protege contra *jamming*, pero representa un gran avance en la defensa frente al *spoofing*. Su coste es nulo si el receptor ya es compatible, requiriendo

únicamente una actualización de firmware. Su eficacia supera el 99 % contra ataques de tipo *meaconing* y modificación de efemérides. Aunque en fase pre-operacional en 2024, su adopción se espera generalizada en los próximos años.

• Antenas adaptativas (CRPA / *beamforming*):

Las antenas con capacidad de formación de haces (*beamforming*), como las CRPA (*Controlled Reception Pattern Antennas*), utilizan múltiples elementos para crear zonas de ganancia nula en la dirección de la interferencia, como podemos ver en la Figura 7. Son extremadamente eficaces contra *jammers* localizados, con mejoras de hasta +40 dB en la relación señal/interferencia. Sin embargo, su coste es elevado (entre 10,000 y 30,000 €), y su implementación requiere electrónica avanzada de control y procesamiento de señales RF (Radiofrecuencia). Por ello, su uso queda reservado a entornos de alta seguridad como el militar, industrial o aeroespacial.

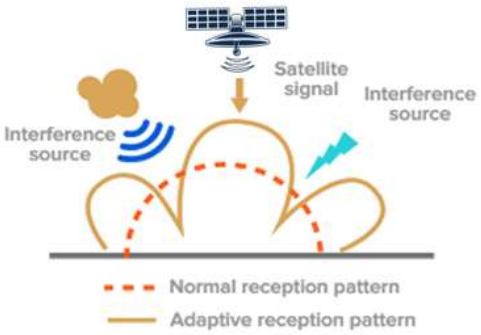

Figura 7 Ejemplo de funcionamiento de antena CRPA

• Monitoreo interno de métricas GNSS (AGC, C/N₀, Doppler, discriminadores):

Muchos receptores ya incluyen algoritmos de detección que supervisan métricas internas como la AGC, la C/N₀, las velocidades Doppler estimadas o los discriminadores de seguimiento. Al detectar desviaciones inusuales en estos parámetros, el sistema puede advertir de posibles ataques sin necesidad de hardware adicional. Esta técnica, integrada en el firmware, permite detectar tanto *jamming* como *spoofing* con una fiabilidad cercana al 94

% en condiciones reales. Su implementación es sencilla y ya está presente en sectores como la aviación civil o la automoción.

• Fusión con sistemas inerciales y relojes de alta estabilidad:

Las unidades de medida inercial (IMU, *Inertial Measurement Unit*), junto con relojes de alta precisión como los OCXO (*Oven-Controlled Crystal Oscillator*) o de rubidio, permiten mantener tanto la posición como la sincronización temporal durante interrupciones breves del GNSS. Esta combinación proporciona una continuidad frente a *jamming* y una detección por incoherencia frente a *spoofing*. El coste de una IMU MEMS ronda los 200–500 €, mientras que los sistemas más avanzados con giroscopios y osciladores de rubidio pueden superar los 20.000 €. Su deriva es mínima: menos de 2 metros en 30 segundos en posicionamiento, y por debajo de 10 nanosegundos en una hora para la sincronización. Su uso está consolidado en aplicaciones exigentes como aviación, trenes o cartografía de precisión.

• Sistemas alternativos de posicionamiento y tiempo (eLoran, R-Mode, PTP):

Las tecnologías no basadas en satélites, como eLoran, R-Mode (basado en señales marítimas) o PTP (*Precision Time Protocol* por red), ofrecen fuentes totalmente independientes para la obtención de posición o tiempo. Aunque no están ampliamente disponibles, representan la única solución real para lograr independencia completa del GNSS. Su coste individual es asumible (200–400 € por receptor), pero su despliegue a escala nacional puede superar los 300 millones de euros. Estos sistemas están en fase piloto en países como Reino Unido y dentro de proyectos europeos coordinados, y su madurez tecnológica está en proceso de consolidación.

CAPÍTULO 3 - METODOLOGÍA Y RECURSOS

Este capítulo recoge la metodología seguida para el desarrollo del proyecto, detallando tanto la forma de trabajo adoptada como los recursos técnicos empleados. Se presentan las estrategias organizativas que han permitido estructurar el trabajo de manera eficiente, así como las fases clave del proceso de desarrollo, desde los primeros ensayos hasta la implementación final de los sistemas de interferencia. Asimismo, se describen en detalle los componentes de hardware y software utilizados, con el fin de proporcionar una visión completa de los medios que han hecho posible la ejecución del proyecto.

3.1 FORMA DE TRABAJO Y GESTIÓN.

En esta sección se describe el enfoque adoptado para la organización y ejecución del trabajo a lo largo del proyecto.

• **Comunicación y coordinación con el tutor:** La interacción con el tutor académico del proyecto se mantuvo de forma continuada a lo largo de todo el desarrollo, garantizando una supervisión adecuada y un seguimiento riguroso de los avances. Esta coordinación se articuló principalmente a través de reuniones periódicas celebradas mediante la plataforma Microsoft Teams. En estos encuentros, que tenían una frecuencia variable en función de la fase del proyecto, se presentaban los progresos obtenidos, se proponían posibles líneas de actuación futura y se tomaban decisiones clave sobre aspectos técnicos y metodológicos. Además, se utilizó el canal de mensajería instantánea integrado en Teams para resolver cuestiones puntuales de forma ágil, especialmente aquellas relacionadas con configuraciones de hardware, interpretación de resultados experimentales o cuestiones de enfoque general.

• **Condiciones del entorno de trabajo:** El proyecto se desarrolló íntegramente en un entorno doméstico, haciendo uso de un equipo informático personal y adaptando el espacio disponible para las tareas de experimentación y análisis. Este entorno, si bien presentaba ciertas limitaciones en comparación con un laboratorio universitario, ofrecía la flexibilidad y continuidad necesarias para trabajar de forma sostenida y autónoma. El hardware

específico requerido —principalmente dispositivos SDR y antenas— fue adquirido con apoyo institucional, lo que permitió dotar al entorno de las capacidades técnicas necesarias. Por el contrario, el software utilizado fue obtenido por iniciativa propia, recurriendo exclusivamente a soluciones de código abierto como GNU Radio, GNSS-SDR o gps-sdr-sim, todas ellas ampliamente documentadas por sus comunidades de usuarios y desarrolladores.

• **Metodología iterativa de trabajo:** A lo largo del proyecto se aplicó un enfoque de trabajo iterativo, especialmente útil en procesos experimentales donde la validación empírica y el ajuste continuo son fundamentales. Para cada una de las técnicas de interferencia estudiadas, se estableció un ciclo sistemático compuesto por cuatro fases: en primer lugar, una etapa de recopilación de información técnica y comprensión del fenómeno a replicar; a continuación, una primera prueba experimental realizada en condiciones controladas, que permitía evaluar la viabilidad del enfoque; posteriormente, un análisis detallado de los resultados obtenidos, con especial atención a los comportamientos anómalos o desviaciones respecto a lo esperado; y por último, una fase de reajuste, en la que se decidía si repetir la prueba con modificaciones o adoptar una estrategia alternativa. Este método, basado en la retroalimentación constante, resultó clave para identificar errores, mejorar progresivamente los resultados y adaptar el diseño del sistema a las limitaciones reales del entorno.

• **Documentación y gestión de archivos:** Uno de los pilares fundamentales para garantizar la trazabilidad del trabajo realizado fue el mantenimiento de una documentación interna rigurosa y estructurada. Esta labor se llevó a cabo mediante un sistema jerárquico de carpetas en Google Drive, donde se organizaban los distintos componentes del proyecto: scripts de configuración y control, capturas de espectro, resultados gráficos y numéricos, logs de ejecución y anotaciones técnicas personales. Esta documentación, aunque no fue compartida directamente con el tutor, sirvió como base para el seguimiento del progreso, la detección de patrones relevantes y la recuperación de configuraciones previas en caso de error. Además, facilitó la posterior elaboración de la memoria técnica, permitiendo una reconstrucción precisa y ordenada del proceso experimental.

3.2 FASES DE EVOLUCIÓN DEL TRABAJO.

Esta sección describe, de forma estructurada, las distintas etapas que han marcado el avance del proyecto, desde el estudio preliminar hasta la validación experimental de los resultados.

1. Estudio teórico inicial.

El punto de partida consistió en una revisión detallada de los fundamentos técnicos del sistema GPS, con especial atención a la estructura de la señal civil L1 C/A, los conceptos de efemérides y almanaques, y los formatos estándar como RINEX. Este estudio permitió comprender el flujo de datos desde el satélite hasta el receptor y sentar las bases necesarias para manipular o emular señales GNSS. Paralelamente, se analizó el funcionamiento de herramientas clave como GNSS-SDR, explorando su arquitectura modular y su proceso de adquisición y seguimiento, con el objetivo de evaluar su aplicabilidad como receptor de referencia y como entorno de experimentación.

2. Investigación de técnicas existentes.

Una vez afianzados los conceptos básicos, se procedió a investigar las principales técnicas de *jamming* y *spoofing* descritas en la literatura y aplicadas en entornos reales. Este análisis abarcó tanto aspectos teóricos —tipologías, mecanismos, efectos esperados— como prácticos —costes, dificultad de implementación, requisitos de hardware—, con el fin de seleccionar aquellas técnicas que pudieran ser replicadas dentro de las limitaciones del entorno experimental propuesto. Se consultaron fuentes académicas, informes técnicos y experiencias documentadas por la comunidad de radio definida por software.

3. Selección de hardware y software.

Sobre la base del estudio anterior, se definió el conjunto de herramientas necesarias para implementar los distintos escenarios de interferencia. La elección de los dispositivos SDR se centró en dos modelos accesibles y versátiles: PlutoSDR y HackRF One, que ofrecen un equilibrio adecuado entre prestaciones, facilidad de uso y coste. En cuanto al software, se optó por entornos de código abierto como GNSS-SDR, GNU Radio y gps-sdr-sim,

seleccionados por su compatibilidad con Linux, su comunidad activa y su capacidad para generar, procesar y analizar señales GNSS. Esta etapa incluyó la instalación, configuración y validación inicial de cada herramienta.

4. Implementación de escenarios de *jamming*.

En esta fase se desarrollaron múltiples configuraciones de *jamming*, basadas en señales interferentes como ruido blanco (*barrage*), portadoras continuas (CW) y *chirps* de barrido frecuencial. A través de GNU Radio se generaron las señales, que luego se emitieron con los SDR, ajustando parámetros como el ancho de banda, la frecuencia central y la potencia de salida. El comportamiento del receptor GNSS se observó ante cada tipo de interferencia, registrando métricas como la pérdida de *fix*, el incremento del HDOP (*Horizontal Dilution of Precision*) o la desaparición de satélites visibles. Estos ensayos permitieron identificar los tipos de *jamming* más eficaces y seguros para su uso en entornos controlados.

5. Implementación de escenarios de *spoofing*.

Con el foco puesto en la generación de señales falsas creíbles, se utilizaron efemérides reales extraídas en formato RINEX para alimentar la herramienta gps-sdr-sim. Este software permitió generar archivos de señal que simulan constelaciones GNSS, incluyendo modificaciones deliberadas en la trayectoria o el tiempo. Las señales generadas se emitieron con el HackRF, y se monitorizó la respuesta del receptor GNSS para evaluar si se producía un cambio de posición controlado y progresivo. Se experimentó también con la sincronización y la variación de potencia para optimizar el efecto sin generar alertas en el receptor.

6. Obtención de métricas y validación de resultados.

En la fase final del desarrollo se procedió a la validación funcional de los sistemas de interferencia implementados. El objetivo principal fue comprobar que tanto el *jamming* como el *spoofing* se ejecutaban correctamente y lograban sus respectivos efectos: anular la recepción de señal en el primer caso, y proporcionar una localización falsa en el segundo. Para ello, se realizaron múltiples ensayos controlados que permitieron verificar, en

condiciones reproducibles, que los ataques interferían eficazmente con el posicionamiento GPS. Esta validación se documentó de manera exhaustiva, tanto mediante material técnico como a través de recursos visuales. Se generaron vídeos ilustrativos que muestran el comportamiento de los receptores durante los ataques, así como capturas de pantalla y registros de software que respaldan de forma empírica la funcionalidad del entorno construido. La documentación obtenida en esta etapa constituye la base para la presentación de las demostraciones prácticas del proyecto y respalda su carácter replicable y educativo.

7. Redacción y revisión.

En paralelo a las fases finales del desarrollo experimental, se inició la redacción estructurada de la memoria del proyecto. Este proceso no solo consistió en describir retrospectivamente las tareas realizadas, sino que también sirvió como ejercicio de análisis y depuración. Se documentaron en detalle la metodología seguida, los resultados obtenidos y las limitaciones encontradas, con el fin de garantizar la claridad expositiva y la reproducibilidad de los experimentos. Asimismo, se revisaron y ajustaron múltiples secciones de forma iterativa, integrando coherentemente los datos generados con el marco teórico previamente estudiado.

3.3 HARDWARE EMPLEADO.

En este apartado se enumeran y describen los dispositivos y componentes físicos utilizados en el desarrollo del proyecto. La información presentada se limita a especificaciones técnicas y características funcionales de cada elemento, sin entrar en los criterios de selección, que se detallarán posteriormente en el apartado correspondiente al diseño del sistema.

• **Portátil HP con sistema operativo Ubuntu 22.04:** El equipo principal utilizado como estación de control es un portátil HP equipado con un procesador Intel Core i5-1135G7 de cuatro núcleos, con frecuencia base de 2.4 GHz y modo turbo de hasta 4.2 GHz. Dispone de 8 GB de memoria RAM (*Random Access* Memory) y una unidad de almacenamiento de 512 GB, lo que permite una ejecución fluida de procesos intensivos en lectura y escritura. Las interfaces USB (*Universal Serial Bus*) disponibles son de tipo 3.2 Gen 1, lo que facilita la comunicación rápida con los dispositivos SDR. El equipo ejecuta en tiempo real la

herramienta GNSS-SDR a una tasa de muestreo de 2.6 MSPS, además de compilar y lanzar *scripts* personalizados para la generación y análisis de señales.

• **Portátil MSI con procesador AMD Ryzen 7:** Como segundo equipo de trabajo se empleó un portátil MSI equipado con un procesador AMD Ryzen 7 5800H de ocho núcleos y dieciséis hilos, con una frecuencia base de 3.2 GHz y modo turbo de hasta 4.4 GHz. El sistema cuenta con 16 GB de memoria RAM y una unidad de 1 TB, lo que proporciona un alto rendimiento tanto en operaciones de cómputo intensivo como en accesos rápidos al sistema de archivos. Las interfaces USB disponibles incluyen puertos 3.2 Gen 1 y Gen 2, lo que garantiza una conectividad robusta y de baja latencia con los dispositivos SDR. Este equipo se utilizó principalmente para tareas de simulación y procesamiento de señales, ejecutando herramientas como gps-sdr-sim, GNU Radio y scripts en Python, con capacidad suficiente para trabajar con tasas de muestreo elevadas y garantizar la estabilidad del sistema durante pruebas prolongadas.

• **ADI ADALM-Pluto (PlutoSDR rev. C):** Este transceptor SDR incorpora el chip AD9363, que opera en un rango de frecuencias comprendido entre 325 MHz y 3.8 GHz, con resolución de 12 bits en los canales I/Q (*In-phase/Quadrature)* y una tasa máxima de muestreo de 61.44 MSPS. Su potencia de salida máxima se sitúa en torno a +7 dBm. Utiliza una interfaz USB 2.0 gestionada mediante la biblioteca libiio y cuenta con un reloj interno de 40 MHz. En el contexto del proyecto, se emplea para emitir señales falsas GNSS en la frecuencia L1 (1575.42 MHz), a partir de archivos IQ generados previamente. [13]

• **HackRF One:** Este dispositivo SDR opera en modo semidúplex dentro de un amplio rango de frecuencias, desde 1 MHz hasta 6 GHz. Maneja señales I/Q de 8 bits y permite una tasa de muestreo de hasta 20 MSPS. La potencia de salida puede alcanzar aproximadamente +15 dBm. El firmware instalado corresponde a la versión 2018-01-1. Cuenta con capacidad de filtrado *baseband* seleccionable y conexión mediante USB 2.0. Su función principal en el proyecto es doble: por un lado, generar interferencias de tipo *jamming*; por otro, actuar como analizador espectral para observar el entorno de pruebas. [14]

• **Antena *patch* GPS L1 pasiva:** Se trata de una antena específica para la banda L1 del GPS, con frecuencia central de 1575.42 MHz ± 2 MHz. Ofrece una ganancia aproximada de 3 dBi, presenta polarización circular derecha (RHCP, *Right-Hand Circularly Polarized*) y una impedancia nominal de 50 ohmios. Al no incluir amplificación activa, permite recibir tanto señales legítimas como señales interferentes de baja potencia sin riesgo de saturar el receptor, lo que resulta especialmente útil en entornos de prueba a corta distancia.

• **Latiguillos RG-316 SMA (SubMiniature Version A)↔SMA de 15 cm y adaptadores recto/acodado:** Los latiguillos coaxiales empleados para el conexionado de los dispositivos SDR y la antena están fabricados con cable RG-316, con conectores SMA en ambos extremos. Presentan una pérdida típica de aproximadamente 0.4 dB por cada 30 cm a una frecuencia de 1.6 GHz. Los adaptadores rectos y acodados facilitan la conexión mecánica en espacios reducidos, asegurando al mismo tiempo una baja pérdida de señal y buena integridad eléctrica.

Podemos encontrar varios de los elementos descritos, como los SDRs o las antenas y conectores, en la Figura 8.

Figura 8 SDR, antenas y conectores utilizados.

3.4 SOFTWARE EMPLEADO.

Este apartado recoge las herramientas software utilizadas a lo largo del proyecto, describiendo sus principales características técnicas y funcionalidades. No se detallan aquí los motivos de su elección, que serán tratados en secciones posteriores dedicadas al diseño experimental y a la toma de decisiones técnicas.

• **GNSS-SDR v0.0.20:** Receptor GNSS definido por software, escrito en C++ y estructurado en bloques modulares. Se emplearon bloques específicos como GPS_L1_CA_PCPS_Acquisition, DLL/PLL Tracking y PVT (*Position, Velocity and Time)* para extraer posición, velocidad y tiempo a partir de señales en la banda L1. La configuración utilizada (archivo hackrf.conf) especificaba una fuente OsmoSDR con frecuencia de muestreo de 2.6 MSPS y formato de entrada ishort. El sistema genera salidas en formato NMEA-0183 y ficheros CSV (*Comma Separated Values*) para análisis detallado de la señal y seguimiento por canal.

• **gps-sdr-sim (commit 28ca29a):** Herramienta de generación de señales GNSS simuladas, escrita en C. Produce archivos IQ intercalados de 16 bits a 2.6 MSPS a partir de efemérides reales en formato RINEX. Permite definir trayectorias simuladas o posiciones fijas, así como ajustar manualmente potencia, desplazamiento Doppler y retardos. La salida principal (gpssim.bin) es utilizada como base para la transmisión de señales falsificadas.

• **plutoplayer:** Programa que transmite de forma continua un archivo IQ al PlutoSDR utilizando la interfaz libiio. Se configuró con frecuencia local de 1575.42 MHz, tasa de muestreo de 2.6 MSPS y amplitud relativa de −10 dBFS. Su función es convertir gpssim.bin en señal RF real en tiempo real, reproduciendo fielmente los datos generados por gps-sdr-sim. [15]

• **GNU Radio 3.10:** Entorno gráfico de procesamiento digital de señales (DSP, *Digital Signal Processing*) basado en bloques. Se emplearon bloques como *Noise Source*, *Signal Source* y *Frequency Mod* para generar señales de interferencia: ruido blanco de banda ancha, CW y *chirps* con barrido de frecuencia. Las salidas se dirigían bien al bloque osmocom Sink

(para transmisión por HackRF), o bien a QT GUI FFT para visualización espectral en tiempo real. [16]

• **SoapySDR + libosmosdr:** Conjunto de bibliotecas que actúan como capa de abstracción entre software y hardware SDR. Permiten a herramientas como GNSS-SDR y GNU Radio comunicarse indistintamente con dispositivos PlutoSDR o HackRF, garantizando la portabilidad de los flujos de trabajo y simplificando la configuración. [17][18]

• **Python 3.10 + Jupyter:** Lenguaje de programación y entorno interactivo empleados para el análisis de datos extraídos de los CSV de GNSS-SDR. Se desarrollaron scripts que calculan la C/N_0, derivan pseudodistancias y generan gráficas con bibliotecas como matplotlib y pandas. Estos análisis permitieron interpretar los efectos de las interferencias de forma cuantitativa.

• **Herramientas auxiliares y de documentación:** Se emplearon *scripts* en *bash* para automatizar tareas recurrentes como la descarga de efemérides (get_eph.sh), la generación de archivos IQ (make_bin.sh) o la ejecución de pruebas de *spoofing* (run_spoof.sh), incluyendo el registro automático de resultados. Asimismo, se utilizaron procesadores de texto como LibreOffice Writer o MS Word para la redacción de la memoria, la elaboración de figuras, tablas y anexos, y la edición general del documento técnico.

CAPÍTULO 4 - DESARROLLO DEL SISTEMA DE INTERFERENCIAS POR *JAMMING*.

En este capítulo se detalla el desarrollo completo del sistema de interferencias por *jamming* implementado en el marco del proyecto. El recorrido comienza con una revisión comparada de las principales técnicas de *jamming* conocidas, que permite contextualizar la elección metodológica adoptada. A continuación, se describe la arquitectura final del sistema, incluyendo su topología física y los criterios que han guiado su diseño. En el tercer bloque se analizan los componentes concretos —tanto hardware como software— empleados para materializar la solución, justificando su selección en función de sus características técnicas y su adecuación al entorno experimental. Finalmente, se exponen el procedimiento de implementación, el modo de operación del sistema y los resultados obtenidos, con el fin de evaluar su eficacia y validar el cumplimiento de los objetivos propuestos.

4.1 REVISIÓN DE TÉCNICAS DE JAMMING.

Las técnicas de interferencia por *jamming* sobre señales GNSS, y en particular sobre la señal L1 C/A del sistema GPS, han evolucionado para adaptarse a distintas finalidades, niveles de sofisticación y restricciones técnicas. En este apartado se revisan las principales estrategias existentes, resaltando sus mecanismos de funcionamiento y justificando los motivos por los cuales no han sido adoptadas en el presente proyecto. La única técnica finalmente empleada —el ruido de banda ancha— se menciona al final de manera sucinta, remitiendo al lector al apartado siguiente para un análisis más detallado.

• **Portadora continua (CW / tonal):** Esta técnica consiste en emitir una o varias señales sinusoidales de frecuencia fija que interfieren selectivamente las celdas de adquisición del receptor GPS, especialmente aquellas utilizadas por los algoritmos FFT. Su bajo ancho de banda y la potencia concentrada hacen que resulte eficaz con niveles de emisión muy reducidos.

No obstante, su implementación exige una sintonización precisa para afectar eficazmente las frecuencias utilizadas por los satélites visibles, lo que complica su operativa en escenarios dinámicos. Además, su efecto puede resultar menos ilustrativo desde un punto de vista didáctico, ya que no impide totalmente la recepción, sino que dificulta la adquisición, generando una interferencia parcial poco evidente para usuarios no especializados.

• ***Chirp* o barrido de frecuencia:** En esta variante se emite un ruido cuyo espectro se desplaza de forma continua a través de la banda GPS, habitualmente en intervalos del orden de microsegundos. Esta variación constante permite evadir filtros *notch* y complica la mitigación basada en supresión espectral.

Sin embargo, su desarrollo práctico requiere una modulación dinámica del generador de señal, así como un diseño cuidadoso para mantener una tasa de barrido coherente. Estas exigencias aumentan considerablemente la complejidad de implementación respecto a otras opciones, especialmente en entornos experimentales con medios limitados. Además, su análisis en tiempo real resulta más difícil de visualizar e interpretar, lo que resta valor educativo al experimento.

• ***Protocol-aware jamming:*** Este enfoque emula el patrón de espectro ensanchado del código C/A utilizando técnicas de modulación DSSS, con el objetivo de sabotear el proceso de correlación en el receptor sin elevar ostensiblemente el nivel de ruido. Se trata de una técnica muy sofisticada y efectiva, que puede anular completamente la adquisición sin generar señales de alerta.

No obstante, su implementación exige una capacidad de generación a nivel de chip, lo que obliga a disponer de un segundo SDR o una FPGA (*Field Programmable Gate Array*) especializada. Estos requisitos contradicen uno de los principios clave del proyecto: desarrollar un entorno de ataque que sea accesible, portátil y de bajo coste, empleando exclusivamente un único SDR y herramientas de código abierto.

• **Ruido de banda ancha (*barrage noise*):** Esta técnica consiste en la emisión continua de una señal de ruido que cubre toda la banda L1 GPS, generando una envolvente gaussiana

sobre el espectro de interés. Su principal ventaja radica en la simplicidad de implementación: basta con una fuente de ruido, un SDR y un software que permita controlar la frecuencia y ganancia.

Esta ha sido la técnica seleccionada para el presente proyecto por su idoneidad en contextos educativos y su capacidad para demostrar, de manera directa y contundente, la fragilidad inherente de las señales GPS civiles. La justificación detallada de esta elección se desarrolla en el siguiente apartado.

4.2 ARQUITECTURA DEL JAMMER.

La técnica empleada en este proyecto, conocida como *barrage noise* o ruido de banda ancha, consiste en la generación y transmisión de una señal gaussiana que ocupa de forma continua toda la banda L1 del sistema GPS (1575,42 MHz). Su principal ventaja radica en su sencillez de implementación y en su elevada eficacia para degradar la C/N_0 de los receptores, anulando la capacidad de adquisición y seguimiento de las señales satelitales. A diferencia de otras estrategias más complejas, esta técnica no requiere sintonización fina, replicación espectral ni generación a nivel de chip, lo que permite su implementación íntegra con un único dispositivo SDR y herramientas de software libre. Esta accesibilidad la convierte en la opción ideal para un entorno académico y divulgativo, pues permite evidenciar de manera directa la fragilidad del GPS civil sin necesidad de equipamiento especializado. Asimismo, su espectro plano y continuo facilita la visualización e interpretación de los resultados en tiempo real, favoreciendo la comprensión de los fenómenos interferentes por parte del usuario.

La arquitectura general del sistema de *jamming* se ha concebido con criterios de simplicidad, portabilidad y eficiencia. En términos físicos, el montaje se basa en un ordenador portátil conectado por USB-C a un PlutoSDR, que actúa simultáneamente como transmisor y receptor. A este SDR se le acopla una antena GPS pasiva tipo *patch* mediante dos latiguillos RG-316 y un conector en "T" que permite unir los puertos de transmisión (TX) y recepción (RX). Esta configuración permite no solo la emisión del ruido interferente, sino también la monitorización simultánea del espectro resultante, sin necesidad de instrumentación externa.

El conjunto se presenta como una estación compacta, fácilmente desplegable en entornos controlados, que cumple con los requisitos del proyecto en cuanto a bajo coste, replicabilidad y mínima complejidad operativa. Podemos ver el montaje en el entorno doméstico en la Figura 9.

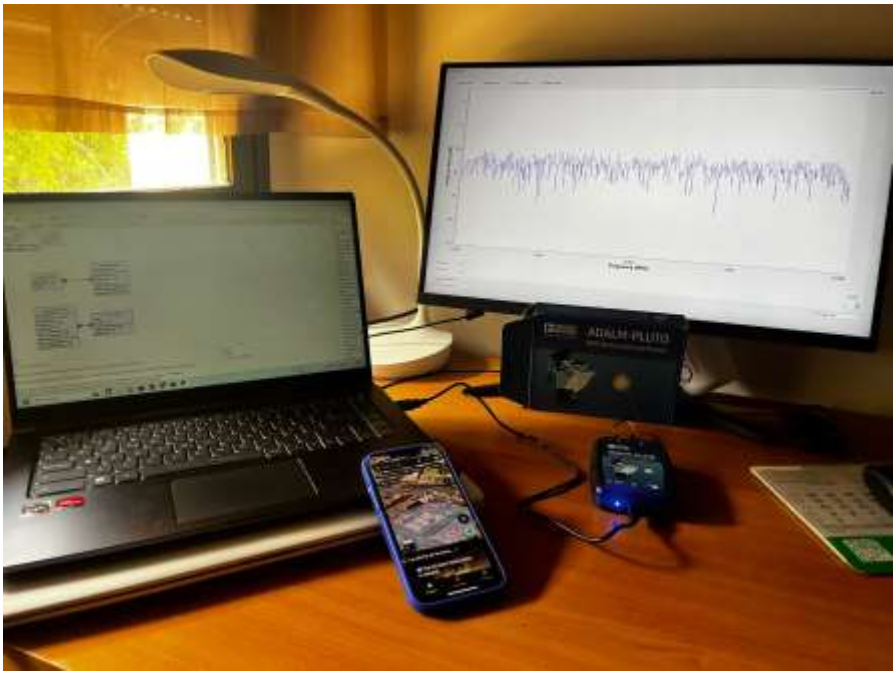

Figura 9 Montaje del jammer.

Una vez establecida la arquitectura, la operativa se articula en torno al control y supervisión desde el propio portátil. A través de él se ejecuta el entorno de generación y emisión de la señal de *jamming*, al tiempo que se visualiza en vivo el espectro transmitido, lo que permite ajustar y validar el correcto funcionamiento del sistema. Paralelamente, se emplean dispositivos iOS para comprobar los efectos del ataque sobre receptores GNSS reales. En este contexto, se han utilizado dos aplicaciones complementarias: Apple Maps, que permite verificar de forma visual e intuitiva la pérdida de posicionamiento o *fix*; y GPSData, una herramienta que ofrece información técnica más detallada, como el número de satélites visibles, los niveles de C/N_0 o el estado interno del receptor, facilitando así una evaluación más precisa del impacto de la interferencia. Esta integración de generación, monitorización

y validación en una única estación portátil proporciona una plataforma didáctica robusta para la demostración práctica de los efectos del *jamming* sobre el GPS.

Con el fin de garantizar la seguridad y el cumplimiento normativo durante el desarrollo del sistema de *jamming*, se han adoptado diversas medidas de precaución. Todas las pruebas se han realizado en un entorno interior controlado, utilizando una antena pasiva sin amplificación (sin LNA, *Low Noise Amplifier*) y limitando intencionadamente la potencia de transmisión mediante atenuación y control software. Además, en determinados ensayos se ha empleado una caja metálica como jaula de Faraday improvisada, lo que ha permitido contener la radiación y evitar posibles interferencias fuera del espacio de prueba. Estas medidas han asegurado que el experimento se mantuviera dentro de los márgenes legales y no supusiera ningún riesgo para dispositivos externos.

4.3 HERRAMIENTAS DE SOFTWARE Y HARDWARE ELEGIDAS.

En este apartado se describen en detalle las especificaciones técnicas de cada componente, tanto de hardware como de software, que conforman el sistema de *jamming* implementado. Además, se ha justificado la elección de cada uno de ellos, atendiendo a criterios de compatibilidad, sencillez de uso, coste y adecuación a los objetivos de reproducibilidad y bajo presupuesto del proyecto.

• **PlutoSDR rev C:** Este dispositivo de radio definida por software opera en un rango de frecuencias comprendido entre 325 MHz y 3.8 GHz, con una resolución de 12 bits y una velocidad máxima de muestreo de 61.44 MS/s. Su potencia máxima de salida alcanza los +7 dBm, y dispone de capacidad full-duplex en una banda de hasta 20 MHz. Integra una interfaz USB 2.0 de control basada en la biblioteca libiio, lo que permite una configuración flexible y acceso desde múltiples plataformas, incluido Windows.

Se ha seleccionado este modelo por su combinación óptima entre prestaciones, coste y compatibilidad. El PlutoSDR resulta ideal para experimentos de *jamming* en interiores, gracias a su potencia de salida suficiente para afectar a receptores cercanos sin necesidad de amplificación adicional. Además, su formato compacto, el soporte directo en GNU Radio y

la integración con herramientas como libiio simplifican notablemente el flujo de trabajo, permitiendo una configuración rápida y reproducible del entorno de interferencia.

• **Antena *patch* GPS L1:** Esta antena opera centrada en la frecuencia de 1575 ± 2 MHz y presenta una ganancia de aproximadamente 3 dBi. Su polarización circular derecha RHCP está alineada con la de los satélites GPS, lo que asegura una recepción y transmisión más eficientes. Al ser pasiva, no introduce componentes activos que puedan amplificar o distorsionar la señal.

Se ha escogido esta antena por su adecuación específica al entorno GPS civil. Al utilizar la misma polarización que los satélites, se garantiza un acoplamiento efectivo sin necesidad de altos niveles de potencia. Además, su carácter pasivo y su ganancia moderada permiten mantener controlada la potencia total irradiada, evitando saturaciones en los receptores y garantizando un comportamiento predecible durante los ensayos.

• **Portátil MSI GF63 (Ryzen 7 5800H, 16 GB RAM):** El equipo utilizado cuenta con un procesador de alto rendimiento de la serie Ryzen 7, acompañado de 16 GB de memoria RAM. Esta configuración ofrece recursos de cómputo más que suficientes para ejecutar flujos DSP en tiempo real con GNU Radio a tasas de muestreo de hasta 10 MS/s, incluso en sistemas operativos como Windows 11.

Se ha optado por este portátil debido a su capacidad de procesamiento, necesaria para la generación continua de señales de *jamming* y su visualización en vivo. Además, su sistema operativo y compatibilidad con el entorno de drivers requerido por PlutoSDR garantizan una ejecución fluida del proyecto, sin cuellos de botella ni requerimientos adicionales de hardware especializado.

• **GNU Radio 3.10:** Esta herramienta de desarrollo de sistemas de procesamiento de señal permite construir flujos DSP mediante bloques visuales conectados entre sí. La versión 3.10 ofrece compatibilidad nativa con PlutoSDR a través de bloques específicos, y permite tanto la generación de señales como su análisis espectral en tiempo real.

Su elección responde a su potencia y flexibilidad como entorno de diseño. GNU Radio permite modificar el flujo de señal en cuestión de minutos, lo que es fundamental en un entorno experimental iterativo. Además, su carácter *open source* y su amplia comunidad de usuarios lo convierten en una solución ideal para proyectos educativos y reproducibles.

• **libiio + driver de PlutoSDR para Windows:** El conjunto formado por la biblioteca libiio y el controlador oficial para Windows permite una interfaz de alto nivel con el dispositivo SDR. A través de esta configuración se pueden ajustar parámetros clave como la frecuencia local, la ganancia y la tasa de muestreo desde aplicaciones como GNU Radio, sin necesidad de programación de bajo nivel.

Esta combinación ha sido elegida por su sencillez y robustez. Al permitir una configuración directa y fiable del PlutoSDR desde el entorno gráfico de GNU Radio, se reduce la complejidad del sistema y se minimizan posibles errores de configuración. Esta integración ha resultado esencial para mantener la fluidez del proceso experimental.

• **Aplicaciones móviles: Apple Maps, GPS Data:** Durante las pruebas, se han utilizado dos aplicaciones complementarias. Apple Maps proporciona una evaluación rápida e intuitiva de la pérdida de posicionamiento al observar si desaparece el *fix*. Por su parte, GPS Data permite obtener métricas detalladas como el número de satélites visibles, los niveles de C/N_0 y el estado del receptor.

Se ha recurrido a estas aplicaciones por su accesibilidad y valor práctico. Apple Maps permite comprobar rápidamente si el *jamming* ha resultado efectivo desde el punto de vista del usuario final, mientras que GPS Data aporta una capa adicional de validación técnica, permitiendo cuantificar el grado de interferencia con mayor precisión.

• **Cables y adaptadores SMA (RG-316, 15 cm):** Los latiguillos empleados son del tipo RG-316, con una longitud reducida de 15 cm y conectores SMA acodados. Este tipo de cable presenta bajas pérdidas en frecuencias del orden de 1.5 GHz, y su flexibilidad facilita el montaje en espacios reducidos, como un escritorio de laboratorio.

Se han seleccionado estos cables por su equilibrio entre rendimiento eléctrico y manejabilidad física. La longitud mínima y los conectores acodados permiten mantener una topología compacta y ordenada, lo que contribuye a la estabilidad de las conexiones durante las pruebas y reduce interferencias indeseadas o pérdidas por acoplamiento.

4.4 IMPLEMENTACIÓN.

En este apartado se detalla el proceso necesario para montar de forma replicable el entorno experimental de *jamming* GPS. Se describe, en primer lugar, la puesta a punto del hardware y del software, incluyendo la detección y configuración del SDR, así como la preparación del entorno de desarrollo. A continuación, se expone el diseño del flujo en GNU Radio, donde se definen los bloques y parámetros clave que permiten generar y controlar la señal de interferencia de forma precisa y reproducible.

4.4.1 PUESTA A PUNTO DE HARDWARE Y SOFTWARE.

Conexión y detección básica del PlutoSDR: El primer paso en la puesta a punto del sistema consiste en establecer una conexión adecuada entre el ordenador y el PlutoSDR. Al conectar el dispositivo mediante un cable USB-C, el sistema operativo Windows lo detecta inicialmente como una unidad de almacenamiento externa. Esta unidad aparece, por ejemplo, con la letra *E:* y contiene archivos de configuración del dispositivo. Para habilitar su uso como radio definida por software, se ejecuta el instalador oficial de Analog Devices (*PlutoSDR-Drivers.exe*), que instala el controlador WinUSB y crea una interfaz de red virtual con la dirección IP 192.168.2.1. La correcta detección puede verificarse mediante el Administrador de dispositivos, donde el Pluto debe figurar bajo la categoría “Universal Serial Bus devices”, como vemos en la Figura 10.

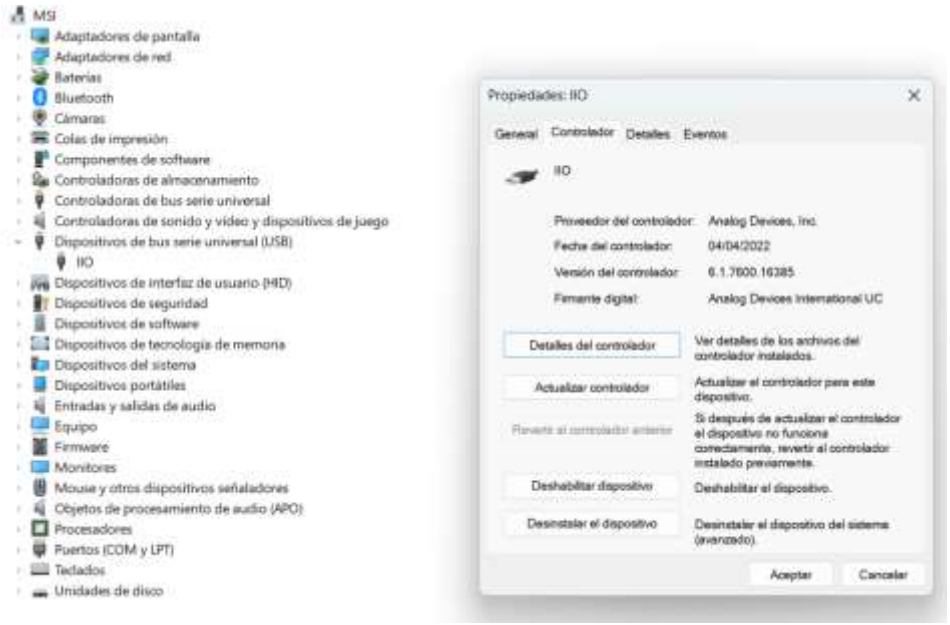

Figura 10 PlutoSDR en Administración de Dispositivos.

Actualización de firmware: Con el objetivo de garantizar la compatibilidad con las herramientas más recientes, se procede a la actualización del firmware del PlutoSDR. Se descarga el paquete oficial (por ejemplo, plutosdr-fw-v0.36.zip) desde el repositorio de GitHub de Analog Devices. Los archivos principales del firmware (.pluto.dfu y .info) se copian directamente a la unidad correspondiente al PlutoSDR. El dispositivo detecta automáticamente los archivos y reinicia el proceso de actualización, flasheando la nueva versión. La verificación de la instalación se puede realizar desde PowerShell con el comando *iio_info -u ip:192.168.2.1*, lo que permite comprobar la versión instalada, la frecuencia del reloj interno (40 MHz) y los parámetros de muestreo habilitados.

Instalación de las librerías de comunicación: Para que el PlutoSDR pueda ser controlado desde aplicaciones externas, es necesario instalar la biblioteca libiio en su versión para Windows. Se copian los archivos correspondientes (*runtime* + DLL) a una carpeta del sistema (p. ej., C:\Program Files\libiio) y se añade dicha ruta a la variable de entorno *PATH*. Además, se crea un archivo config.txt en esa misma carpeta con la línea ip:192.168.2.1, lo que permite que las aplicaciones reconozcan automáticamente el dispositivo sin necesidad de especificar su dirección IP (*Internet Protocol*) de forma explícita.

Entorno GNU Radio: La plataforma principal utilizada para el desarrollo del sistema de *jamming* es GNU Radio. Para su instalación se emplea el paquete PothosSDR (versión recomendada: *2023-05-Nightly*), que incluye GNU Radio 3.10 junto con SoapySDR y los drivers necesarios. Tras la instalación, se actualizan los paquetes gnuradio-iio y gr-osmosdr a través de conda, asegurando así la compatibilidad con la versión más reciente de libiio. La correcta integración se comprueba abriendo GNURadio Companion (GRC) y verificando que, en el menú Preferences > Core > Industrial I/O, aparecen los bloques "PlutoSDR Source" y "PlutoSDR Sink", como vemos en la Figura 11.

Figura 11 Bloques para PlutoSDR dentro de GNU Radio.

Entorno gráfico y dependencias auxiliares: Algunos elementos visuales de GNU Radio, como los bloques QT GUI, requieren la instalación de bibliotecas gráficas adicionales. En este caso, se instala el *runtime* de Qt versión 5.15.2 y el redistribuible de Visual C++ 2019, necesarios para el correcto funcionamiento de ciertos módulos (por ejemplo, gr-iio). Además, se configuran variables de entorno como *QT_PLUGIN_PATH* o *GRC_BLOCKS_PATH* para garantizar que todos los bloques y complementos estén accesibles desde GRC.

Gracias a este conjunto de pasos secuenciales, se parte de un sistema sin preparación específica y se alcanza un entorno completamente funcional, en el que el PlutoSDR está plenamente integrado con GNU Radio bajo Windows, listo para transmitir y recibir señales

GNSS a tasas de hasta 10 MS/s. Este proceso garantiza una base estable sobre la que implementar, probar y ajustar el sistema de *jamming* desarrollado en este proyecto.

4.4.2 DISEÑO DEL FLUJO GNU RADIO.

Una vez completada la instalación del entorno y reconocidos todos los dispositivos, se procede al diseño del flujo de trabajo en GNU Radio. El objetivo de este flujo es generar una señal de ruido gaussiano, transmitirla sobre la banda GPS L1 a través del PlutoSDR, y visualizar en tiempo real su impacto en el espectro, incluyendo la transición desde un entorno limpio hasta uno interferido. A continuación, se detallan los bloques utilizados, su configuración y la justificación de su elección, como vemos en la Figura 12; y posteriormente analizamos cada elemento de forma dedicada.

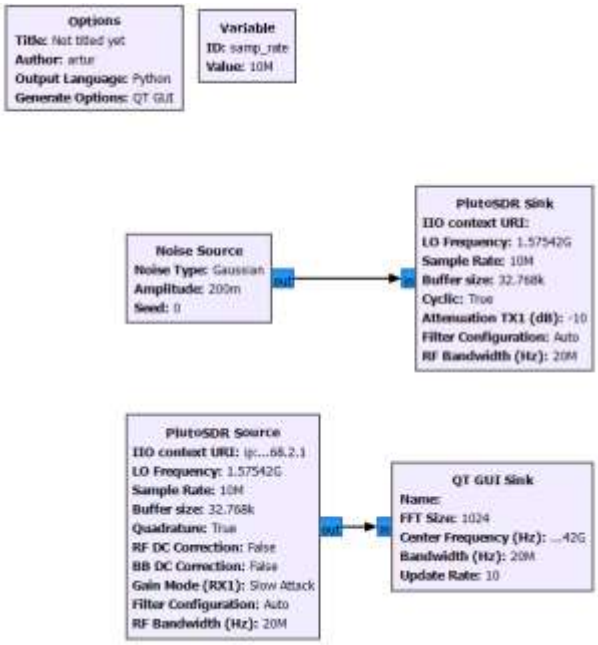

Figura 12 Esquema GNU Radio para el jamming.

- **Variable** *samp_rate = 10e6.*
  - Función: define una tasa de muestreo común para todos los bloques de transmisión y recepción.

- Justificación: garantiza coherencia entre los módulos del flujo y simplifica el ajuste de parámetros. El valor de 10 MS/s permite cubrir toda la banda L1 GPS (≈ 2 MHz útiles + márgenes Doppler y de diseño).

- **Noise Source.**
  - Tipo: Gaussiano.
  - Amplitud: 0,20 Vpp, que corresponde aproximadamente a –20 Dbfs (decibelios respecto al nivel máximo de la cadena digial).
  - Función: genera la señal de *jamming* tipo *barrage noise*.
  - Justificación: esta amplitud ha sido determinada experimentalmente como el mínimo necesario para provocar pérdida de *fix* en menos de 5 segundos, sin llegar a saturar el *front-end* RF de los receptores móviles cercanos.
- **PlutoSDR Sink (Transmisión)**
  - LO Frequency: 1 575.42 MHz (centro de la banda GPS L1).
  - Sample Rate: 10 MS/s.
  - RF Bandwidth: 20 MHz (valor máximo permitido por el filtro del AD9363).
  - Attenuation TX1: –10 dB, que da una potencia de salida ≈ –3 dBm.
  - Cyclic: True, el buffer de ruido se repite de forma indefinida.
  - Función: transmite continuamente la señal de ruido sobre la banda GPS.
  - Justificación: esta configuración permite cubrir toda la banda GPS civil con una señal constante, eficaz pero contenida en potencia para garantizar la seguridad del experimento.
- **PlutoSDR Source (Recepción)**
  - LO Frequency: 1 575.42 MHz (coincide con la de transmisión).
  - Sample Rate: 10 MS/s.
  - Gain Mode RX1: Slow Attack, el AGC responde progresivamente al incremento de potencia.
  - RF Bandwidth: 20 MHz.
  - Función: monitoriza en tiempo real la señal emitida y el entorno circundante.

- Justificación: este bloque permite verificar visualmente el efecto del *jamming* en el espectro, así como comprobar la densidad de potencia recibida con el mismo SDR, evitando la necesidad de instrumentación adicional.

- **QT GUI FFT Sink.**
  - FFT Size: 1024, lo que da una resolución espectral ≈ 9.7 kHz/bin.
  - Center Frequency: 1 575.42 MHz.
  - Bandwidth: 20 MHz.
  - Update Rate: 10 fps.
  - Función: muestra la evolución del espectro antes, durante y después de la emisión.
  - Justificación: permite observar cómo desaparecen las señales discretas de los satélites GPS (picos) y son reemplazadas por una "alfombra" de ruido continua cuando el *jamming* está activo.
- **Conexiones entre bloques**
  - Noise Source → PlutoSDR Sink
  - PlutoSDR Source → QT GUI FFT Sink
  - Justificación: este esquema simple y directo mantiene el flujo estable, minimiza latencias y permite validar la interferencia generada en tiempo real.

Este diseño modular y ajustado permite replicar el entorno experimental con facilidad y asegura que los efectos observados puedan atribuirse directamente a las señales generadas, sin ambigüedades ni interferencias externas.

4.5 VERIFICACIÓN DE RESULTADOS.

Una vez implementado el sistema, este apartado se centra en comprobar su funcionamiento efectivo. En primer lugar, se analiza el espectro para confirmar que se está emitiendo potencia de forma continua y dentro de la banda GPS prevista. Posteriormente, se presentan evidencias del impacto real de la interferencia sobre receptores GNSS, tanto mediante métricas técnicas como a través de aplicaciones gráficas de uso cotidiano.

4.5.1 CONFIRMACIÓN ESPECTRAL DE LA EMISIÓN.

Para verificar que el sistema implementado emite una señal interferente con las características deseadas, se ha recurrido a una inspección espectral directa mediante el propio PlutoSDR operando en modo receptor. La configuración del bloque PlutoSDR Source se mantuvo idéntica a la de transmisión, con una frecuencia central de 1575,42 MHz y un ancho de banda de 20 MHz. Esta simetría permite capturar de forma precisa el comportamiento del canal en las mismas condiciones en las que se está emitiendo. La señal recibida fue representada en tiempo real mediante el bloque QT GUI FFT, lo que facilitó la comparación visual del espectro antes y después de activar el *jamming*.

La Figura 13 muestra el espectro captado en condiciones normales, sin interferencia activa. En ella se observa un ruido de fondo estable en torno a los –70 dB, sobre el cual emergen varios picos estrechos con una separación característica, que corresponden a las señales emitidas por los satélites GPS. Uno de estos picos alcanza los –39 dB, lo que indica una recepción nítida. Asimismo, se distingue claramente el patrón de modulación de espectro ensanchado que caracteriza a la señal L1 C/A, reflejando la estructura inherente del canal GNSS.

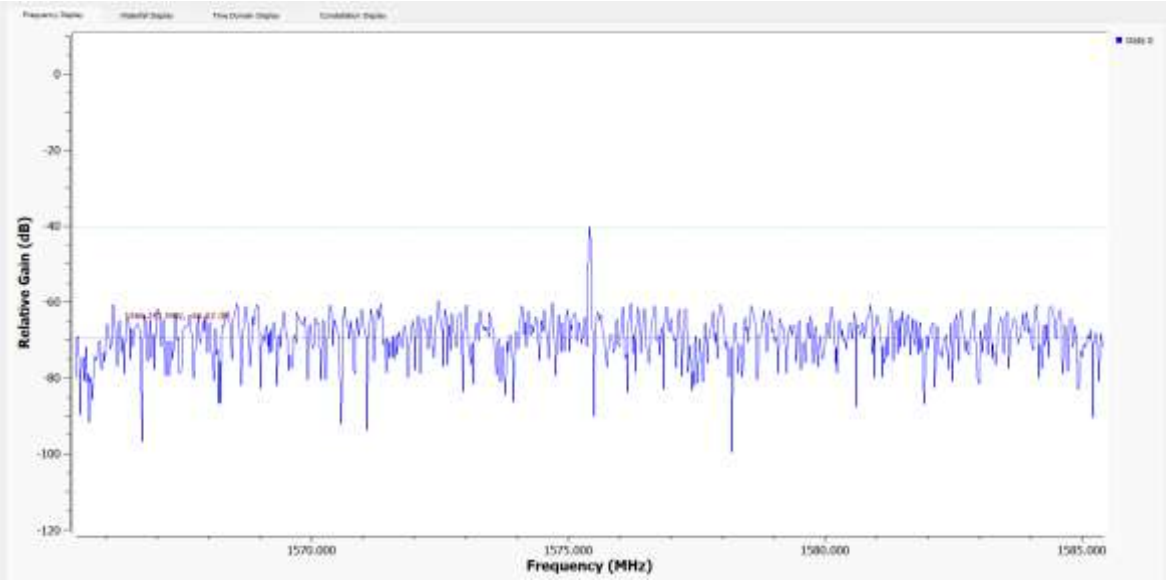

Figura 13 Espectro antes de la emisión de ruido.

En contraste, la Figura 14 presenta el espectro tras activar la emisión del ruido gaussiano. El piso de ruido se eleva uniformemente hasta los –50 dB a lo largo de toda la ventana de observación, lo que indica una inyección continua de energía a lo largo del espectro. Los picos que anteriormente correspondían a los satélites desaparecen completamente, quedando sumergidos más de 25 dB por debajo del nuevo nivel medio. La pendiente del espectro se mantiene prácticamente plana, lo que confirma que la interferencia generada es efectivamente de tipo banda ancha y no una portadora tonal localizada.

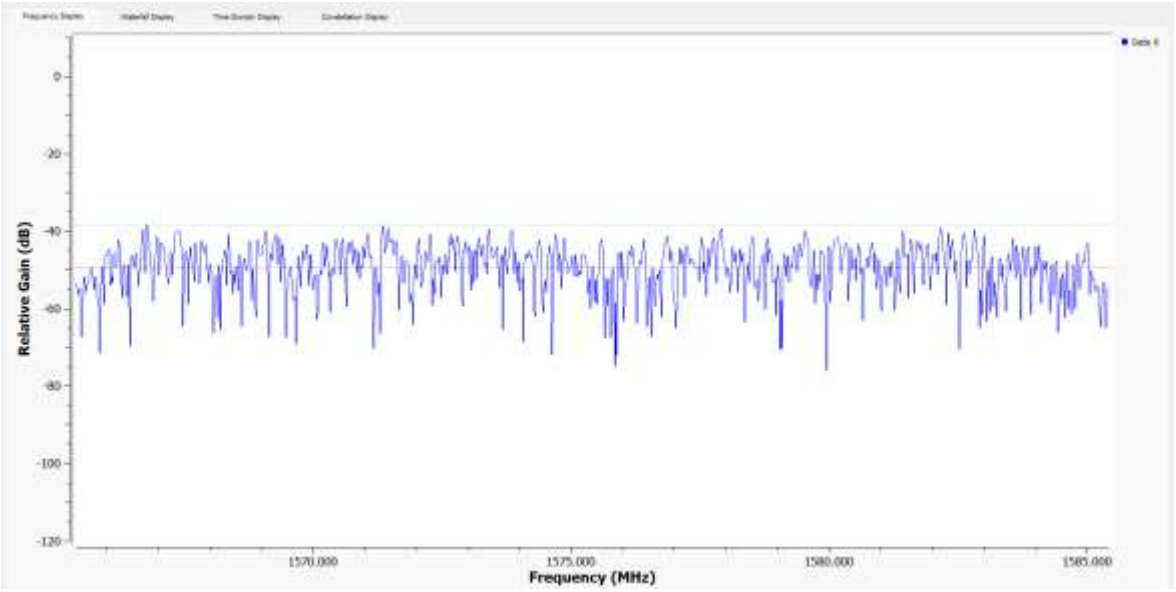

Figura 14 Espectro una vez se está emitiendo ruido en la frecuenci GPS.

En conjunto, esta comparación espectral demuestra que el sistema de *jamming* desarrollado cumple con su función: ocupa de forma eficaz y homogénea toda la banda L1 del GPS, elevando el nivel de ruido e impidiendo la detección de las señales satelitales por parte de receptores GNSS estándar. Esta confirmación es esencial para validar la operatividad del flujo de GNU Radio y la correcta configuración del PlutoSDR como emisor interferente.

4.5.2 IMPACTO REAL EN UN DISPOSITIVO MÓVIL.

Para evaluar el impacto real del sistema de *jamming* sobre un receptor GNSS comercial, se ha utilizado la aplicación GPS Data instalada en un smartphone iOS situado a

aproximadamente un metro de la antena emisora. A fin de aislar los efectos de la interferencia únicamente sobre la señal GPS, el dispositivo se mantuvo en modo avión durante toda la prueba, asegurando que no recibiera datos de posicionamiento asistido (A-GPS, *Assisted GPS*) ni información procedente de redes móviles o Wi-Fi. Esta herramienta proporciona métricas detalladas sobre la calidad del posicionamiento, incluyendo la precisión estimada en los ejes vertical y horizontal, el tiempo necesario para obtener el *fix* y el instante del último posicionamiento válido. Se capturaron dos estados representativos: uno sin emisión interferente (transmisor apagado) y otro durante la activación del *jamming*. Los resultados se resumen en las Figuras 15 y 16, respectivamente.

La Figura 15 refleja el comportamiento del sistema en condiciones normales. El terminal obtiene un *fix* casi inmediato, en menos de un segundo, y mantiene niveles de precisión característicos de un entorno exterior despejado: 12 metros en el eje horizontal y 4 metros en el vertical. La hora del último *fix* se actualiza de forma continua, lo que indica que el receptor mantiene la solución de posicionamiento de manera estable y sin interrupciones.

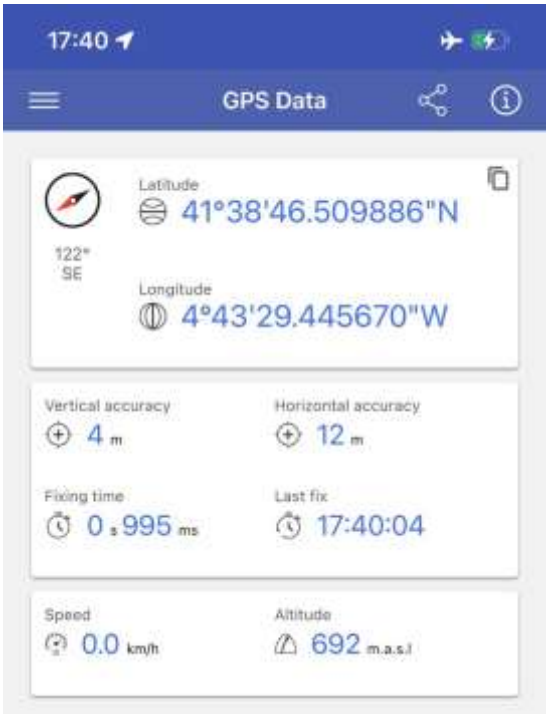

Figura 15 Fix con GPS Data y modo avión cuando no hay jamming.

Tras activar la emisión de ruido (Figura 16), se observan cambios significativos en todos los indicadores clave. El tiempo necesario para obtener un nuevo *fix* se eleva a más de 14 segundos, lo que representa una multiplicación por catorce respecto a la condición base. La precisión se degrada de forma sustancial: el error horizontal aumenta de 12 a 49 metros, y el vertical de 4 a 44 metros. Además, el campo "Last fix" deja de actualizarse con normalidad, lo que sugiere que el receptor pierde repetidamente la solución de posicionamiento, forzando intentos intermitentes de readquisición.

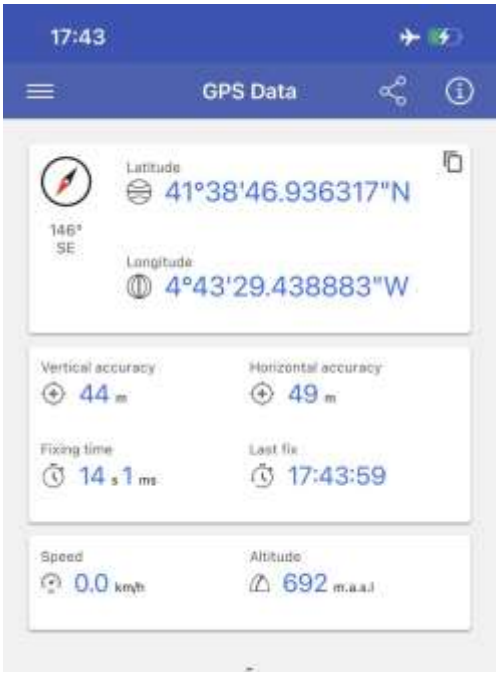

Figura 16 Fix con GPS Data y modo avión con jamming

Cabe destacar que el hecho de que el teléfono logre obtener un *fix* eventual durante el *jamming* no implica que el sistema GNSS esté operando correctamente. En condiciones de interferencia intensa, los receptores tienden a forzar una solución incompleta o degradada basada en un número reducido de satélites o en datos marginales. Esto se traduce en una precisión extremadamente baja y en un posicionamiento inestable, como se evidencia en la oscilación de los valores y la latencia en la obtención del *fix*. Por tanto, estos resultados no solo confirman la emisión efectiva de la interferencia, sino que demuestran su impacto

tangible sobre la capacidad del receptor para determinar su posición de forma fiable y continua.

Verificación gráfica mediante Apple Maps: Como complemento visual a las pruebas técnicas anteriores, se utilizó la aplicación Mapas de Apple en un dispositivo iOS también configurado en modo avión, lo que garantiza que cualquier intento de localización se base únicamente en la recepción de señales GNSS. Durante la prueba con el *jammer* activado, la aplicación no consigue triangular la posición del usuario, como se aprecia en la Figura 17: la pantalla permanece estática con el mensaje "El modo Avión está activado" y el indicador circular de carga (ruleta gris) se mantiene indefinidamente. Esta situación es consecuencia directa de la interferencia: al elevar el nivel de ruido en la banda L1, el dispositivo es incapaz de recibir las señales de los satélites con la calidad necesaria para calcular una posición válida. Esta prueba constituye una forma más intuitiva y cercana al usuario final de constatar la eficacia del *jamming*, ya que evidencia la pérdida del servicio GPS en una aplicación de uso cotidiano.

Figura 17 Apple Mapas incapaz de triangular la posición debido al jamming.

CAPÍTULO 5 - DESARROLLO DEL SISTEMA DE INTERFERENCIAS POR *SPOOFING*.

En este capítulo se describe el diseño, implementación y validación del sistema de *spoofing* desarrollado en el marco del presente TFM. Se revisan inicialmente las distintas técnicas consideradas y se justifica la elección final. A continuación, se detallan los componentes hardware y software empleados, la arquitectura del entorno experimental, y la forma de operación. Finalmente, se presentan los resultados preliminares obtenidos durante las pruebas.

5.1 REVISIÓN DE TÉCNICAS DE SPOOFING.

Este apartado recoge las principales técnicas de *spoofing* analizadas a lo largo del proyecto. Para cada una se describe su funcionamiento, se evalúan sus limitaciones y se justifica su descarte. La última técnica, seleccionada como base para el desarrollo, se introduce brevemente antes de abordarla en detalle en el apartado siguiente.

• ***Meaconing*** **(re-radiado diferido):** Consiste en capturar la señal GPS legítima mediante una antena receptora, amplificarla ligeramente (en torno a +3 dB) y reemitirla tras un retardo fijo que puede variar entre nanosegundos y milisegundos. Este retardo común a todas las señales provoca un alargamiento homogéneo de las pseudodistancias, resultando en un desplazamiento de posición del receptor que, aunque detecta una señal aparentemente válida, queda localizado erróneamente a varios cientos de metros de su ubicación real. Se trata de una técnica simple, que no requiere conocimientos profundos sobre las señales GNSS, y que puede producir errores de localización de hasta 500 metros dentro de una "burbuja" de cobertura próxima al kilómetro.

A pesar de su simplicidad y bajo riesgo de detección, esta técnica presenta limitaciones importantes en el contexto del presente TFM. Su ejecución requiere un sistema de grabación y reproducción GNSS de coste elevado, además de dos cadenas completas de

radiofrecuencia, lo cual contradice el objetivo económico del proyecto. Además, el retardo fijo introducido no permite representar de manera controlada un desplazamiento progresivo, por lo que su valor didáctico es limitado.

• ***Carry-off* coherente:** Esta técnica se basa en generar una réplica de la señal GPS real en fase, pero con una ganancia levemente superior (2–3 dB), logrando así que el receptor realice un traspaso de seguimiento (*handover*) sin advertir la transición. Una vez que se ha capturado al receptor, se desplaza progresivamente la señal falsificada mediante pequeñas modificaciones del código, normalmente inferiores a 1 metro por segundo, "arrastrando" al receptor a lo largo de una trayectoria falsa pero coherente.

Aunque su capacidad de engaño es notable y su detección compleja, el elevado coste del equipamiento necesario (generador vectorial y software especializado) y la dificultad de ajustar la rampa de desplazamiento con precisión la hacen poco adecuada para un entorno doméstico o educativo. Además, el arrastre gradual dificulta una demostración clara y rápida del efecto, algo esencial para fines docentes.

• **Manipulación del mensaje de navegación:** A diferencia de otras técnicas, esta modalidad no modifica la fase ni el código de la señal, sino que altera la información contenida en el mensaje de navegación, introduciendo efemérides falsas o manipulando los campos de sincronización temporal (como el Z-count). El receptor sigue la portadora original sin interrupciones, pero los errores en el mensaje provocan cálculos incorrectos de posición o tiempo, con derivadas acumulativas de hasta cientos de metros.

Se descarta por su elevada complejidad técnica, ya que requiere un conocimiento detallado del protocolo de navegación GPS y un control preciso de la codificación de los datos. Además, su impacto se centra más en la sincronización temporal que en el posicionamiento, alejándose de los objetivos didácticos del proyecto. Al tratarse de un fenómeno poco visible en tiempo real, su efecto resulta difícil de mostrar en una demostración sencilla.

• **Simulación generativa tiempo-real multicanal:** Emplea simuladores profesionales como Spirent o Racelogic, capaces de generar señales GNSS sintéticas en tiempo real, incluyendo

todas las portadoras y efemérides, con trayectorias programables y ajustes precisos por satélite. Estas soluciones permiten construir escenarios de *spoofing* extremadamente realistas, con errores milimétricos y cobertura de varios kilómetros.

Esta técnica, a pesar de ser la más completa y efectiva desde un punto de vista técnico, queda completamente fuera del alcance del presente trabajo debido a su coste prohibitivo y requerimientos logísticos. Su implementación contradice la premisa fundamental del proyecto, que es desarrollar una plataforma educativa y asequible, accesible por menos de 250 euros y replicable en un entorno doméstico.

• **Simulación offline con gps-sdr-sim y SDR doméstico:** Se genera un fichero IQ que contiene una señal GPS sintética basada en efemérides reales, utilizando la herramienta de código abierto gps-sdr-sim. Este archivo es reproducido desde un dispositivo SDR económico (como el PlutoSDR), que emite la señal falsificada con una potencia ligeramente superior a la legítima. De esta forma, el receptor es inducido a aceptar la señal falsa y calcular una posición errónea.

Esta técnica cumple plenamente los dos objetivos principales del proyecto: demostrar de forma práctica la vulnerabilidad del GPS con un equipo de coste inferior a 250 euros, y proporcionar una solución abierta y fácilmente replicable, con scripts personalizables que permiten modificar la trayectoria deseada. El desarrollo detallado de esta técnica se aborda en el siguiente apartado.

5.2 ARQUITECTURA DEL SPOOFER.

La técnica de *spoofing* seleccionada para el presente proyecto se basa en la simulación total generada por software. Este enfoque consiste en sintetizar localmente una réplica completa de la señal GPS civil (L1 C/A), codificada con efemérides reales pero correspondiente a una trayectoria espacial ficticia. El fichero generado, que contiene la señal en formato I/Q, se emite mediante un transmisor SDR con una ganancia ligeramente superior (alrededor de +3 dB) respecto a la señal auténtica de los satélites. Esta diferencia de potencia es suficiente para que el receptor realice una transición automática hacia la señal espuria, sin requerir

arrastre progresivo ni manipulación dinámica. La constelación falsa se encuentra ya alineada en tiempo y fase gracias a la precisión del generador gps-sdr-sim, lo que garantiza una transición instantánea, reproducible y fácilmente demostrable.

La arquitectura del sistema diseñado integra dos dispositivos SDR conectados a un ordenador portátil con sistema operativo Linux (Ubuntu 22.04), tal y como vemos en la Figura 18. El módulo de transmisión se basa en un PlutoSDR, que emite la señal falsa generada previamente con gps-sdr-sim. Esta señal se transmite a través de una antena pasiva con RHCP, replicando el patrón de los satélites GPS. La potencia de emisión se ha ajustado para situarse en torno a −15 dBm EIRP, lo que crea una "burbuja" de cobertura limitada (3–5 metros), suficiente para simular el entorno de ataque. En paralelo, se ha utilizado un HackRF One como receptor de validación, conectado al mismo portátil o a otro independiente, cuya función es verificar el resultado del ataque mediante la demodulación de la señal recibida. A través de GNSS-SDR, configurado con un archivo específico (hackrf.conf), se capturan los datos I/Q y se procesan con adquisición multicanal, permitiendo la reconstrucción completa de la solución PVT en el punto simulado.

Figura 18 Montaje del spoofer.

La operación se ha realizado de forma manual desde el entorno Linux, sin procesos automatizados. Cada paso —desde la descarga de efemérides hasta la conversión, emisión y posterior validación— ha sido ejecutado individualmente para asegurar el control completo del procedimiento y facilitar el análisis de cada fase. La señal sintética se genera con gps-sdr-sim a partir de efemérides recientes y una trayectoria definida por el usuario, y se transmite mediante PlutoSDR. Al mismo tiempo, la recepción con HackRF y el procesamiento mediante GNSS-SDR permiten confirmar el éxito del engaño al observar cómo el receptor testigo converge en las coordenadas falsas. Este enfoque manual permite una mayor flexibilidad y comprensión detallada del proceso, especialmente valiosa en contextos formativos.

Finalmente, cabe destacar que el entorno propuesto ofrece un alto potencial de escalabilidad y adaptación. Aunque el presente diseño se ha limitado a un escenario de *spoofing* básico, la misma arquitectura permite extenderse a pruebas más avanzadas, como trayectorias dinámicas o escenarios multiconstelación. Asimismo, el uso de herramientas de código abierto y hardware ampliamente disponible facilita su adaptación a distintas configuraciones y entornos. Esta versatilidad convierte al sistema no solo en una herramienta de demostración puntual, sino en una plataforma base sobre la que construir futuras investigaciones o aplicaciones docentes en el ámbito de la seguridad GNSS.

5.3 HERRAMIENTAS DE HARDWARE Y SOFTWARE ELEGIDAS.

En este apartado se enumeran las herramientas y dispositivos empleados específicamente en el sistema de *spoofing*, distintos a los utilizados en el módulo de *jamming*. Se incluyen tanto componentes de hardware como aplicaciones de software, destacando sus características clave y los motivos por los que se han considerado adecuados para este entorno.

• **HackRF One:** Este dispositivo SDR cubre un rango de frecuencias extremadamente amplio, de 1 MHz a 6 GHz, y permite la captura o transmisión de señales en formato I/Q de 8 bits, con una tasa de muestreo de hasta 20 MS/s. Su potencia máxima de transmisión se sitúa en torno a +15 dBm, aunque en este proyecto se ha empleado exclusivamente en modo

recepción. Con un coste aproximado de 320 €, representa una de las opciones más asequibles en su categoría. Lo podemos ver con más detalle, junto con el PlutoSDR previamente explicado, en la Figura 19.

Figura 19 HackRF y PlutoSDR utilizados en el proyecto.

Se ha seleccionado por su versatilidad y compatibilidad con librerías como OsmoSDR y libiio, que permiten su integración directa con herramientas como GNSS-SDR. En el marco de este proyecto, ha actuado como receptor flexible para validar la señal falsificada, permitiendo capturas continuas de IQ sin necesidad de equipos de laboratorio, lo que refuerza la motivación de emplear un entorno replicable y económico.

• **GNSS-SDR v0.0.20:** Se trata de un receptor GNSS de código abierto escrito en C++, capaz de configurar flujos de procesamiento completamente personalizados, desde la adquisición hasta el seguimiento (DLL/PLL) y la obtención de la solución PVT final mediante integración con RTKLIB. Su uso exige una carga computacional significativa: en este proyecto, el sistema ha operado al 100 % de uso de CPU (Ryzen 7 5800H) durante la decodificación de señales a 2,6 MS/s en 16 canales simultáneos.

Su elección responde a la necesidad de contar con un software transparente y reproducible, capaz de procesar la señal de *spoofing* como si fuera real. El hecho de que la solución PVT se obtenga con éxito utilizando GNSS-SDR confirma la eficacia del engaño a nivel funcional, demostrando así la validez del ataque dentro de los objetivos académicos del proyecto.

• **gps-sdr-sim (commit 28ca29a):** Esta herramienta genera señales GPS sintéticas en formato IQ a partir de efemérides reales en formato RINEX y una trayectoria definida por el usuario en KML (*Keyhole Markup Language*). Ofrece soporte tanto para PlutoSDR (2,6 MS/s, 8 bits) como para HackRF (2,0 MS/s, 16 bits), permitiendo adaptar la salida al SDR disponible.

Su principal ventaja es su carácter abierto y completamente editable: permite generar constelaciones completas con una única instrucción, modificando tanto el contenido orbital como la posición del receptor objetivo. Esta flexibilidad elimina la necesidad de recurrir a simuladores profesionales con costes de cinco cifras, alineándose perfectamente con el objetivo de crear una plataforma de bajo coste y fácilmente replicable.

• **Antena *patch* GPS L1 pasiva (Tx spoofing):** La antena empleada para la transmisión de la señal simulada es una *patch* pasiva centrada en la frecuencia de 1575 ± 2 MHz, con una ganancia aproximada de 3 dBi. Su RHCP está alineada con la utilizada por los satélites GPS, lo que optimiza la eficiencia en la transmisión de la señal interferente. Al tratarse de una antena pasiva, no incorpora elementos activos como LNA, lo que garantiza que la señal generada por el SDR se emita sin alteraciones adicionales, manteniendo así un mayor control sobre la potencia y la forma espectral de la transmisión.

Esta elección permite trabajar de forma segura en entornos interiores, manteniendo las emisiones dentro de los márgenes permitidos y evitando interferencias accidentales.

• **Portátil HP 14" (Ubuntu 22.04):** El sistema se ha ejecutado en un equipo portátil equipado con procesador Ryzen 5 5500U, 16 GB de memoria RAM, disco de 1 TB y puertos USB 3.0. Este hardware ha demostrado ser suficiente para compilar gps-sdr-sim y GNSS-SDR en

tiempos inferiores a 2 minutos, permitiendo sesiones de trabajo prolongadas sin problemas térmicos.

Su elección se ha basado en criterios de portabilidad y rendimiento. Con un peso inferior a 1,5 kg y un consumo energético moderado, este equipo ha permitido trasladar el entorno de pruebas a diferentes ubicaciones, reforzando el carácter práctico y formativo del sistema desarrollado.

5.4 DESARROLLO DE LOS SUBSISTEMAS.

En este apartado se detalla el desarrollo individual de los distintos componentes que conforman el sistema de *spoofing*. Primero se aborda la instalación y prueba de cada módulo por separado, antes de integrarlos en la arquitectura final. Este enfoque permite una validación incremental y facilita la resolución de errores.

5.4.1 GENERADOR DE SEÑAL -GPS-SDR-SIM.

La herramienta gps-sdr-sim es un generador de señales GNSS en formato I/Q, de código abierto, creada en 2016 por el desarrollador japonés Takuji Ebinuma, conocido en la comunidad como osqzss. Es una aplicación escrita en lenguaje C bajo licencia MIT (*Massachusetts Institute of Technology*), lo que permite su libre uso, modificación y distribución. Originalmente concebida como una herramienta experimental para investigar el comportamiento de receptores GPS, su utilidad se ha extendido con el tiempo a múltiples entornos académicos y de desarrollo por su equilibrio entre simplicidad, potencia y fiabilidad. Su código fuente está disponible públicamente, lo que ha favorecido su adopción en proyectos de seguridad, sincronización, validación de algoritmos de navegación y, como en este caso, generación controlada de interferencias.

Desde un punto de vista técnico, gps-sdr-sim implementa la ecuación de rango GPS y el generador de código C/A correspondiente a los 32 PRN definidos por la especificación del sistema GPS en la banda civil L1 (1575,42 MHz). Su motor de simulación admite efemérides

reales en formato RINEX, lo que permite generar señales que replican con precisión la configuración orbital de la constelación en un instante determinado. A partir de esta información, la herramienta genera un flujo binario en el que se alternan muestras I y Q de 8 o 16 bits, con frecuencias de muestreo configurables que van desde 1 hasta 26 MS/s. Este flujo representa la señal modulada tal como la "vería" un receptor GPS real, y puede ser reproducido directamente desde un SDR para simular la presencia de satélites.

En el marco del presente proyecto, gps-sdr-sim constituye el núcleo funcional del sistema de *spoofing*. Su papel consiste en sintetizar una constelación completa de señales falsas que serán posteriormente emitidas por el PlutoSDR. A diferencia de simuladores comerciales que requieren hardware especializado de elevado coste, esta herramienta permite generar señales coherentes y configurables mediante comandos simples desde un terminal Linux. Las coordenadas falsas pueden establecerse como un punto estático o como una trayectoria dinámica definida por un fichero KML, lo que otorga al sistema una flexibilidad notable con un coste mínimo. En definitiva, gps-sdr-sim permite materializar de forma directa y accesible los objetivos del TFM: construir una plataforma de ataque por *spoofing* basada en hardware doméstico, reproducible, y orientada a la demostración práctica de vulnerabilidades GNSS. [19]

5.4.1.1 INSTALACIÓN Y USO DIGITAL PURO.

Este subapartado describe el proceso de instalación de la herramienta gps-sdr-sim en un entorno Linux, así como su uso en modo digital puro. El objetivo es generar señales GPS sintéticas en formato I/Q sin necesidad de emplear hardware SDR en esta fase inicial.

Instalación de gps-sdr-sim en entorno Linux.

La herramienta gps-sdr-sim ha sido diseñada para su uso en sistemas tipo Unix, y su instalación en Ubuntu 22.04 es directa siempre que se disponga de las dependencias adecuadas. A continuación, se detalla el proceso completo, incluyendo la obtención del código fuente, su compilación y una verificación básica del funcionamiento.

1. Instalación de dependencias: Antes de compilar el código, es necesario asegurarse de que el sistema dispone de los paquetes esenciales para la construcción de software y las bibliotecas requeridas por gps-sdr-sim:

```
sudo apt update

sudo apt install build-essential git libfftw3-dev libusb-1.0-0-dev zlib1g-dev
```

   Este paso instala herramientas de desarrollo como el compilador gcc, la utilidad make, y bibliotecas como FFTW3 (para el manejo de transformadas rápidas de Fourier), libusb (interacción con dispositivos USB), y zlib (compresión y manejo de archivos).

2. Descarga del repositorio y compilación: Una vez instaladas las dependencias, se clona el repositorio oficial del proyecto y se compila el ejecutable:

```
git clone https://github.com/osqzss/gps-sdr-sim.git

cd gps-sdr-sim

make
```

   El proceso de compilación es rápido, y genera un ejecutable llamado gps-sdr-sim en el mismo directorio. Pueden aparecer advertencias relacionadas con las variables r_ref o r_xyz, pero estas no afectan al funcionamiento del programa y pueden ignorarse con seguridad.

3. Verificación de la instalación: Para comprobar que la compilación ha sido exitosa y que el binario es funcional, se recomienda ejecutar la ayuda integrada del programa:

```
./gps-sdr-sim -h
```

   Este comando muestra una lista completa de opciones disponibles, indicando que el ejecutable está correctamente construido y listo para su uso, como podemos ver en la Figura 20.

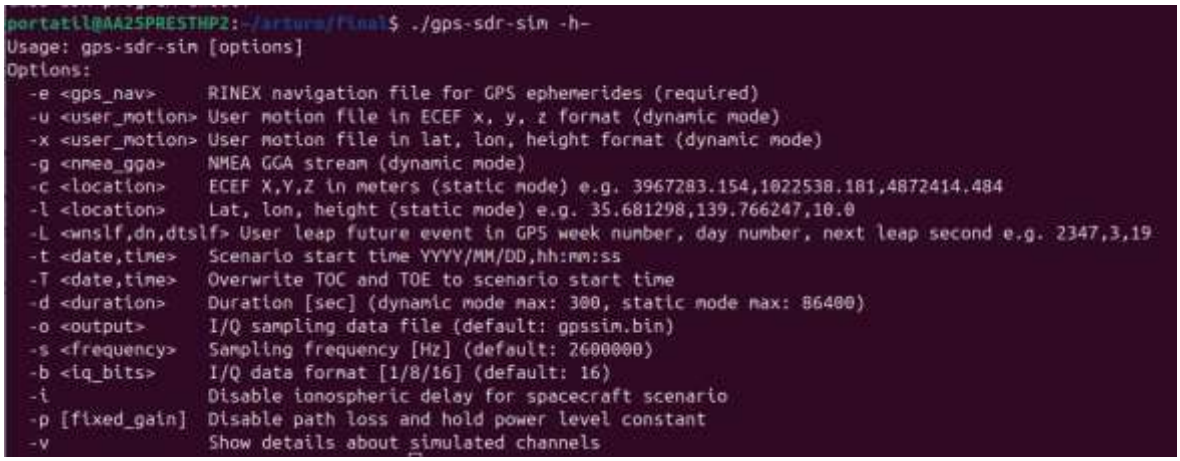

Figura 20 Comprobación de la correcta instalación de gps-sdr-sim.

Uso básico de gps-sdr-sim: generación de señales I/Q.

Una vez instalado, el uso principal de gps-sdr-sim consiste en generar un archivo binario en formato I/Q que simula la señal de una constelación GPS en la banda L1. Este archivo puede ser reproducido por un SDR para emular una señal real. A continuación, se describe un ejemplo típico de generación y se analizan sus parámetros.

3. Comando de generación básico:

```
./gps-sdr-sim  -e brdc0010.22n -l 40.4300523,-3.712636,40  -d 300  -s 2600000
-b 8 -o gpssim8b.bin
```

Este comando produce un archivo gpssim8b.bin que representa 300 segundos de señal GPS centrada en las coordenadas de ICAI, con una frecuencia de muestreo de 2,6 MS/s y resolución de 8 bits por muestra. A continuación, se describen los parámetros utilizados:

- -e brdc0010.22n: Archivo RINEX con efemérides y almanaque. Es obligatorio para generar una constelación coherente con una fecha real.
- -l 40.4300523, -3.712636, 40*:* Coordenadas geográficas (latitud, longitud, altitud en metros) que definen la ubicación del receptor simulado.
- -d 300: Duración de la señal simulada en segundos. Puede alcanzar hasta 86 400 s (24 h).

- -f 2600000: Frecuencia de muestreo, expresada en muestras por segundo. Debe coincidir con la que admite el SDR emisor.
- -b 8: Resolución en bits de cada muestra. La opción de 8 bits reduce el tamaño del fichero, lo que resulta adecuado para SDRs como el PlutoSDR.
- -o gpssim8b.bin: Nombre del archivo de salida. Este contendrá la señal I/Q lista para ser emitida.

Podemos ver el resultado de la ejecución de este comando en la Figura 21, y la generación del fichero de datos como se ha explicado.

Figura 21 Ejemplo de ejecución de gps-sdr-sim de forma digital.

4. Tamaño y validación del archivo: El tamaño del fichero generado depende directamente de la frecuencia de muestreo, la duración y la resolución. En el ejemplo mostrado (2,6 MS/s, 8 bits, 300 s), el archivo final tiene un tamaño aproximado de 1,56 GB, calculado como se indica en la Figura 22:

$$\text{Tamaño} \approx 2{,}6\,\text{MS/s} \times 300\,\text{s} \times 2\,\text{canales} \times 1\,\text{byte} = 1{,}56\,\text{GB}$$

Figura 22 Fórmula para el cálculo del tamaño del fichero de datos sintéticos.

Para una validación rápida, se puede comprobar que el fichero contiene datos no nulos con:

```
xxd -l 32 gpssim8b.bin
```

También puede visualizarse su estructura o distribución espectral con herramientas adicionales como gnss_plot.py, si se dispone de ellas.

5.4.1.2 TRANSMISIÓN DEL BINARIO CON PLUTOSDR.

Una vez generado el fichero binario con gps-sdr-sim, es necesario reproducirlo a través de un transmisor SDR para que la señal falsificada sea emitida y pueda ser captada por los receptores objetivo. En este proyecto se ha empleado el dispositivo PlutoSDR, desarrollado por Analog Devices, que combina facilidad de uso, tamaño compacto y una buena relación coste-prestaciones. La transmisión del archivo I/Q se realiza utilizando el módulo plutoplayer, incluido en el repositorio de gps-sdr-sim.

Antes de proceder con la transmisión, es imprescindible asegurarse de que el entorno de ejecución cumple una serie de requisitos previos. En primer lugar, el sistema debe disponer de los drivers oficiales de Analog Devices, que permiten reconocer al PlutoSDR como un dispositivo accesible mediante la dirección *ip:192.168.2.1.* Además, debe estar instalada la librería libiio en su versión 0.36 o superior, ya que es la encargada de gestionar la comunicación entre el sistema operativo y el SDR. Estas condiciones son necesarias tanto en sistemas Linux como Windows.

Para compilar plutoplayer, se accede al subdirectorio player del repositorio de gps-sdr-sim y se ejecuta el siguiente comando:

```
cd gps-sdr-sim/player

make
```

Esta orden genera el binario plutoplayer, que puede copiarse al sistema para facilitar su ejecución desde cualquier ubicación:

```
sudo cp plutoplayer /usr/local/bin
```

El uso de plutoplayer consiste en transmitir el archivo I/Q previamente generado, controlando parámetros clave como la amplitud de salida, el factor de compensación de picos (*back-off*) y la tasa de muestreo. Un ejemplo típico de ejecución sería:

```
plutoplayer -t gpssim8b.bin -a -10.0 -b 3.0 -s 2600000
```

Cada parámetro tiene una función concreta:

- -t gpssim8b.bin: indica el fichero binario que contiene la señal I/Q a transmitir. Una vez completada su reproducción, el archivo se repite en bucle.

- -a -10.0*:* fija la amplitud de la señal en dBFS. Un valor de –10 dBFS asegura que la señal se emita con una potencia moderada, alrededor de –3 dBm tras el ajuste interno del PlutoSDR, suficiente para crear una "burbuja" segura en interiores.

- -b 3.0: aplica un margen de reducción de pico (*crest factor*) para evitar saturaciones o distorsiones en la señal emitida. Un valor de 3 dB resulta adecuado para señales generadas con gps-sdr-sim.

- -s 2600000: define la frecuencia de muestreo, que debe coincidir con la del fichero generado para evitar errores de sincronización. Si la cabecera del fichero contiene este valor, la opción puede omitirse.

Podemos ver la ejecución del comando explicado y cómo comienza la transmisión de los datos a través del PlutoSDR en la Figura 23.

Figura 23 Transmisión de los datos a través del PlutoSDR con plutoplayer.

El propio plutoplayer configura automáticamente el oscilador local (LO, *Local Oscillator*) del PlutoSDR a 1575,42 MHz, frecuencia correspondiente al canal L1 del sistema GPS. Este valor puede verificarse con el siguiente comando:

```
iio_attr -u ip:192.168.2.1 -d ad9361-phy | grep lo
```

Para confirmar que el PlutoSDR está correctamente enlazado con el sistema y operativo, se puede usar el siguiente comando (y observamos su salida en la Figura 24):

```
iio_info -u ip:192.168.2.1
```

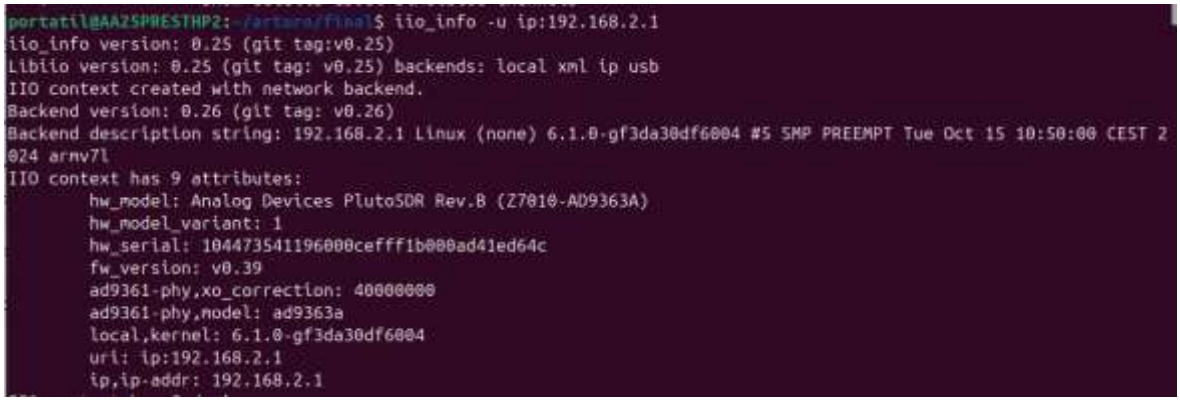

Figura 24 Comprobación de la conexión del PlutoSDR con el PC.

En entornos gráficos como GNU Radio, es posible observar el espectro de la señal emitida añadiendo un bloque PlutoSDR Source sintonizado a 1575,42 MHz, acompañado de un visor FFT mediante QT GUI FFT, lo cual permite visualizar de manera directa la emisión generada.

En términos prácticos, se han observado ciertos efectos de deriva de frecuencia (*drift*) cuando la transmisión se prolonga más de cinco minutos. Aunque no afectan significativamente a las pruebas a corto plazo, se recomienda reiniciar el proceso cada cierto tiempo o, si se desea mayor estabilidad, externalizar el reloj de referencia del PlutoSDR (40 MHz) mediante la entrada REF_CLK, opción que se ha considerado para futuras extensiones del entorno.

Finalmente, para casos en los que se requiera simular un desplazamiento dinámico en la posición, gps-sdr-sim permite generar trayectorias a partir de archivos CSV o KML. En estos casos, basta con generar el fichero IQ correspondiente e introducirlo en plutoplayer siguiendo el mismo procedimiento descrito.

5.4.2 RECEPTOR SOFTWARE -GNSS-SDR.

GNSS-SDR es un receptor GNSS definido por software desarrollado por el grupo de investigación gr-gnss de la Universitat Politècnica de Catalunya. Publicado bajo licencia GPLv3, se trata de una herramienta modular y extensible que permite procesar señales de navegación por satélite utilizando exclusivamente software y hardware generalista. Su núcleo está implementado en C++, y se apoya en GNU Radio para la ejecución de tareas de DSP, como la adquisición y el seguimiento de portadoras, la decodificación del mensaje de navegación y la reconstrucción de la solución PVT.

El sistema está diseñado para ser altamente configurable: cada etapa del receptor —desde la fuente de datos hasta los algoritmos de navegación— puede definirse a través de ficheros de configuración en formato INI (*Initialization File*), lo que permite adaptar el receptor a múltiples escenarios de prueba. GNSS-SDR admite tanto entrada en tiempo real desde un dispositivo SDR como el procesamiento en modo "*replay*" a partir de archivos I/Q previamente grabados. Es compatible con varias constelaciones, incluyendo GPS, Galileo, GLONASS y BeiDou, y soporta múltiples canales en paralelo. Para su funcionamiento, requiere una arquitectura x86-64 con soporte SSE3, al menos 4 GB de memoria RAM, y bibliotecas como Boost, FFTW3, Volk, GFlags y GLog.

En el presente TFM, GNSS-SDR se ha empleado con una doble finalidad. Por un lado, permite validar fuera de línea los archivos binarios generados con gps-sdr-sim, actuando como receptor software que interpreta la señal falsificada y reconstruye una solución PVT coherente con la trayectoria simulada. Por otro, ha sido utilizado como “receptor víctima” en los experimentos prácticos de *spoofing*, donde la señal espuria es emitida mediante PlutoSDR y capturada por un HackRF One. De esta forma, se demuestra empíricamente que el sistema es capaz de engañar a un receptor genérico sin requerir modificaciones en su configuración interna. [20]

5.4.2.1 INSTALACIÓN Y USO DIGITAL PURO.

GNSS-SDR es un receptor GNSS definido por software que, además de operar en tiempo real con dispositivos SDR, permite funcionar en modo digital puro, es decir, procesando directamente archivos I/Q previamente grabados. Esta funcionalidad resulta esencial para validar señales generadas de forma sintética, como las producidas con gps-sdr-sim, sin necesidad de realizar transmisiones radioeléctricas. A continuación, se detalla el proceso de instalación de GNSS-SDR en un sistema Ubuntu 22.04, junto con un ejemplo completo de configuración y ejecución en modo "archivo".

Instalación de GNSS-SDR en Ubuntu 22.04.

El primer paso consiste en preparar el entorno de desarrollo con todas las dependencias necesarias. GNSS-SDR se basa en múltiples librerías externas para tareas de procesamiento de señal, visualización, configuración y gestión del flujo de datos.

```
sudo apt update

sudo apt install git cmake gfortran \

                 libboost-all-dev libfftw3-dev libvolk2-dev \

                 libgflags-dev libgoogle-glog-dev \

                 libgnuradio-dev libuhd-dev gr-osmosdr
```

Este comando instala herramientas de compilación (cmake, gfortran), bibliotecas fundamentales como Boost y FFTW3, la librería VOLK para optimización SIMD, los sistemas de logging GFlags y GLog, y soporte para radios como HackRF o Pluto a través de gr-osmosdr y libuhd.

Una vez instaladas las dependencias, se clona el repositorio oficial del proyecto y se compila el código:

```
git clone https://github.com/gnss-sdr/gnss-sdr.git

cd gnss-sdr && mkdir build && cd build

cmake .. -DUSE_OSMOSDR=ON -DENABLE_GPSD=OFF -DCMAKE_BUILD_TYPE=Release

make -j$(nproc)
```

```
sudo make install
```

La opción USE_OSMOSDR=ON habilita el soporte para dispositivos SDR, aunque no se utilizará en esta fase. Por su parte, ENABLE_GPSD=OFF evita dependencias innecesarias con servicios GPS en segundo plano. La compilación puede tardar varios minutos dependiendo del sistema (≈10 minutos en un Ryzen 7 moderno).

Para maximizar el rendimiento, especialmente en operaciones de seguimiento, se recomienda ejecutar el perfilado de VOLK, que selecciona los *kernels* SIMD óptimos según la arquitectura del procesador [21]:

```
volk_profile
```

Configuración del sistema para uso digital.

Para ejecutar GNSS-SDR en modo digital, es necesario preparar un fichero de configuración en formato INI. A continuación se muestra y comenta el archivo static_file.conf, que permite procesar un archivo I/Q generado por gps-sdr-sim con una frecuencia de muestreo de 2,6 MS/s y resolución de 8 bits:

```
[GNSS-SDR]

GNSS-SDR.internal_fs_sps     = 2600000

GNSS-SDR.dump_filename       = static_log.dat
```

Esta sección define parámetros globales: internal_fs_sp*s* indica la frecuencia de muestreo esperada, mientras que dump_filename establece el nombre del archivo donde se volcarán los resultados binarios del procesamiento.

```
[SignalSource]

SignalSource.implementation  = File_Signal_Source

SignalSource.filename        = /home/user/gpssim8b.bin

SignalSource.item_type       = ishort

SignalSource.sampling_frequency = 2600000

SignalSource.repeat          = true
```

La fuente de señal es un archivo binario (File_Signal_Source), en este caso el generado por gps-sdr-sim. Dado que las muestras se generaron en 8 bits sin signo, GNSS-SDR espera el tipo *ishort* (enteros con signo de 16 bits) para convertirlas adecuadamente en el siguiente bloque. El archivo se repite en bucle para asegurar una alimentación continua al sistema.

```
[DataTypeAdapter]

DataTypeAdapter.implementation = Ishort_To_Complex

[InputFilter]

InputFilter.implementation     = Pass_Through

[Resampler]

Resampler.implementation       = Pass_Through
```

Estos bloques preparan la señal para el procesamiento: Ishort_To_Complex convierte las muestras enteras en valores complejos; los filtros y *resamplers* se configuran como paso directo dado que no se requiere ajuste espectral ni temporal adicional en este modo.

```
[Channels]

Channels_1C.count            = 8

Channels.in_acquisition      = 8

Channel.signal               = 1C
```

Aquí se definen los canales de adquisición y seguimiento. En este caso, se procesan 8 canales simultáneamente, suficiente para obtener un *fix* 3D completo con GPS L1 C/A.

```
[Acquisition_1C]

Acquisition_1C.implementation = GPS_L1_CA_PCPS_Acquisition

Acquisition_1C.pfa             = 1e-4

Acquisition_1C.item_type       = gr_complex
```

La adquisición se realiza mediante el método PCPS (*Parallel Code Phase Search),* eficiente y robusto para señales estáticas. El parámetro pfa (probabilidad de falsa alarma) se fija en 1e-4 para equilibrar sensibilidad y fiabilidad.

```
[Tracking_1C]

Tracking_1C.implementation    = GPS_L1_CA_DLL_PLL_Tracking

Tracking_1C.pll_bw_hz         = 15.0

Tracking_1C.dll_bw_hz         = 2.0
```

La etapa de seguimiento activa lazos PLL y DLL con anchos de banda moderados, adecuados para mantener la traza sobre señales sintéticas sin introducir ruido excesivo.

```
[PVT]

PVT.implementation            = RTKLIB_PVT

PVT.max_gdop                  = 8.0
```

La solución PVT se resuelve utilizando el motor RTKLIB, bien establecido en la comunidad GNSS. Se impone un umbral de máxima DOP geométrica (GDOP, *Geomtric Dillusion of Position*) de 8.0, descartando soluciones degradadas por mala geometría satelital.

Ejecución del receptor GNSS-SDR en modo archivo.

Una vez configurado el archivo INI, GNSS-SDR puede ejecutarse con el siguiente comando:

```
gnss-sdr --config_file=static_file.conf --log_dir=logs_static
```

La opción --config_file especifica el fichero de configuración, mientras que --log_dir define el directorio donde se almacenarán los resultados, incluyendo los registros de seguimiento, observaciones, y el archivo PVT final. Adicionalmente, se puede incluir la opción --sid=$PRN para observar la evolución de un satélite concreto durante el proceso.

Esta ejecución permite verificar, en entorno puramente digital, si la señal generada con gps-sdr-sim es lo suficientemente coherente como para ser procesada exitosamente. La correcta obtención de una solución PVT ficticia demuestra la efectividad del binario como vector de *spoofing* antes de ser emitido físicamente por un SDR.

5.4.2.2 RECEPCIÓN EN TIEMPO REAL CON HACKRF ONE.

Una vez validado el funcionamiento de GNSS-SDR en entorno digital, el siguiente paso consiste en integrarlo con un dispositivo SDR que actúe como receptor de la señal falsificada emitida por el PlutoSDR. Para ello, se ha utilizado el HackRF One, un transceptor de bajo coste y gran flexibilidad, que permite capturar señales en tiempo real directamente desde el entorno radioeléctrico. Este subsistema permite verificar experimentalmente que la señal espuria es procesada con éxito por un receptor genérico, reproduciendo así un ataque de spoofing completo.

Preparación del sistema y requisitos específicos.

Antes de iniciar la captura, es imprescindible comprobar que el HackRF se encuentra en condiciones operativas adecuadas. Se recomienda instalar el firmware oficial de enero de 2018, que garantiza la estabilidad del flujo de datos. La herramienta hackrf_info permite verificar tanto la versión del firmware como el rango de frecuencias y tasas de muestreo soportadas. Se debe confirmar que el dispositivo admite una tasa de al menos 2,6 MS/s en la recepción, valor necesario para reproducir fielmente la señal GPS generada por gps-sdr-sim. Además, se recomienda añadir una regla udev para garantizar el acceso sin privilegios:

```
echo 'SUBSYSTEM=="usb", ATTR{idVendor}=="1d50", MODE="0666"' | \

     sudo tee /etc/udev/rules.d/52-hackrf.rules

sudo udevadm control --reload-rules
```

Configuración del fichero hackrf.conf para captura real.

El archivo de configuración para este modo es similar al utilizado en entorno digital, pero con cambios clave en el bloque de fuente de señal y omisión del adaptador de tipo, ya que las muestras llegan ya en formato complejo (gr_complex). A continuación se describen las secciones relevantes:

```
[GNSS-SDR]

GNSS-SDR.internal_fs_sps = 2600000
```

Se mantiene la frecuencia de muestreo interna en 2,6 MS/s para garantizar compatibilidad con el flujo transmitido.

```
[SignalSource]

SignalSource.implementation   = Osmosdr_Signal_Source

SignalSource.item_type        = gr_complex

SignalSource.sampling_frequency = 2600000

SignalSource.center_frequency = 1575420000

SignalSource.gain             = 14

SignalSource.freq_corr        = 0
```

En este caso, la fuente es el dispositivo HackRF a través de la interfaz OsmoSDR. La frecuencia central se establece en 1575,42 MHz, correspondiente a la portadora L1 del GPS. La ganancia se ajusta empíricamente a 14 dB, lo que permite captar correctamente la señal emitida por el PlutoSDR sin saturar el conversor analógico-digital del HackRF. En caso de observar desviaciones sistemáticas en frecuencia, se puede modificar el parámetro freq_corr para compensar errores del oscilador interno del dispositivo, típicamente del orden de ±1 ppm.

Las demás secciones (*Channels, Acquisition, Tracking*, PVT) permanecen sin cambios respecto a la configuración digital. El número de canales, los métodos de adquisición y seguimiento, y el motor de resolución PVT siguen siendo los mismos:

```
[Channels]

Channels_1C.count = 8

Channels.in_acquisition = 8

[Acquisition_1C]

Acquisition_1C.implementation = GPS_L1_CA_PCPS_Acquisition

Acquisition_1C.pfa = 1e-4

[Tracking_1C]
```

```
Tracking_1C.implementation = GPS_L1_CA_DLL_PLL_Tracking

Tracking_1C.dll_bw_hz = 2.0

Tracking_1C.pll_bw_hz = 15.0

[PVT]

PVT.implementation = RTKLIB_PVT
```

Ejecución del sistema en tiempo real.

Una vez definido el fichero hackrf.conf, se puede lanzar GNSS-SDR con el siguiente comando:

```
gnss-sdr --config_file=hackrf.conf --log_dir=logs_rt
```

Durante la ejecución, la consola mostrará mensajes de adquisición y seguimiento, entre los cuales destaca el inicio del tracking para cada PRN:

```
Tracking of GPS L1 C/A signal started on channel 0 for satellite PRN XX
```

La correcta obtención de una solución PVT con coordenadas falsas confirmará que el *spoofing* ha sido exitoso. Se puede validar este resultado consultando los archivos .nmea generados en el directorio logs_rt, donde se observará una posición desviada al menos 100 metros respecto a la ubicación real del sistema.

Consideraciones prácticas y ajustes.

Durante las pruebas, pueden producirse *overruns* (marcados con "O" en consola), que indican pérdida de datos por saturación del *buffer* USB. En ese caso, se recomienda reducir la frecuencia de muestreo a 2,0 MS/s y regenerar el fichero I/Q con esa tasa. Alternativamente, para sesiones prolongadas, se puede mejorar la estabilidad del LO del HackRF mediante:

```
hackrf_clock --si5351c
```

Esta orden activa la corrección automática de frecuencia del chip Si5351C, contribuyendo a una mejor alineación entre la señal emitida y captada en escenarios de duración extendida.

5.5 INTEGRACIÓN DE LOS SUBSISTEMAS PARA FORMAR EL SPOOFER.

Una vez desarrollados y validados de forma independiente los dos componentes principales del sistema de *spoofing* —la generación de señales sintéticas con gps-sdr-sim y su recepción y análisis mediante GNSS-SDR—, este apartado describe cómo integrarlos de forma coordinada con los dispositivos SDR. Se establece un flujo de trabajo reproducible y seguro que permite ejecutar un ataque de *spoofing* en condiciones controladas. Asimismo, se detallan las precauciones necesarias para garantizar la coherencia temporal, espectral y de configuración entre todos los elementos implicados.

5.5.1 FLUJO OPERATIVO INTEGRADO.

Una vez descritos todos los subsistemas por separado, este apartado presenta el flujo operativo completo para ejecutar un experimento de *spoofing*. Se detallan los pasos necesarios desde la preparación inicial hasta la recopilación de evidencias finales.

1 • Descarga de efemérides GNSS.

Para comenzar una sesión de *spoofing*, es fundamental obtener las efemérides del día desde una fuente oficial como CDDIS (*Crustal Dynamics Data Information System*) o ESA-GNSS (*European Space Agency GNSS*). El fichero, normalmente identificado como BRDC (*Broadcast Daily Ephemeris*) y en formato RINEX comprimido, debe ser descomprimido antes de su uso. Contar con datos orbitales recientes evita que los receptores detecten inconsistencias en los mensajes de navegación o activen advertencias sobre la antigüedad de los datos, lo cual comprometería la credibilidad del experimento.

2 • Generación del fichero binario I/Q.

La generación de la señal sintética se realiza con gps-sdr-sim, definiendo las coordenadas deseadas (latitud, longitud, altitud) o, si se desea simular movimiento, proporcionando un archivo de trayectoria en formato CSV. Es esencial fijar la frecuencia de muestreo a 2,6 MS/s, idéntica a la que se utilizará en el sistema de transmisión, y optar por una resolución

de 8 bits por muestra para optimizar el tamaño del archivo y la eficiencia del enlace USB. Una vez generado el binario, se recomienda verificar que su tamaño sea coherente con la duración programada y que contenga datos válidos, utilizando herramientas como xxd para confirmar la ausencia de ceros planos.

3 • Preparación de los dispositivos SDR.

Antes de la transmisión, se deben preparar correctamente tanto el PlutoSDR como el HackRF One. El PlutoSDR debe estar conectado por USB 2.0 y visible como dispositivo de red en la dirección ip:192.168.2.1, con *firmware* igual o superior a la versión 0.36, y su frecuencia de muestreo configurada a 2,6 MS/s. Por su parte, el HackRF One debe contar con el *firmware* oficial 2018-01-1 y las reglas udev aplicadas para garantizar el acceso sin restricciones. También es necesario configurar su frecuencia de muestreo a 2,6 MS/s, establecer una ganancia de recepción inicial de 14 dB, y fijar la frecuencia central en 1575,42 MHz, que corresponde a la portadora GPS L1.

4 • Lanzamiento de la transmisión.

La transmisión se inicia ejecutando plutoplayer con el fichero binario generado previamente. Es recomendable establecer la amplitud en torno a –10 dBFS y aplicar un *back-off* de 3 dB para evitar el recorte de picos (*clipping*) y preservar la fidelidad espectral de la señal. En cuanto se lanza el comando, el PlutoSDR comienza a emitir de forma continua la constelación GPS falsificada con una potencia suficiente para dominar sobre la señal auténtica, generando una burbuja local de *spoofing* de unos pocos metros de radio.

5 • Inicio de la recepción y análisis.

En paralelo, se ejecuta GNSS-SDR con el fichero de configuración adaptado al HackRF, asegurando que tanto la frecuencia de muestreo como la frecuencia central coincidan con las utilizadas durante la generación y transmisión del binario. En pocos segundos, deberían comenzar a aparecer en consola los mensajes de adquisición de señal, indicando que el receptor ha captado correctamente los satélites simulados. Si el ataque ha sido efectivo, la

solución PVT del receptor convergerá a las coordenadas falsas programadas, lo que se puede verificar fácilmente comparando la salida NMEA con la ubicación real.

6 • Captura de evidencias.

Para documentar el experimento, se emplean exclusivamente los registros generados por GNSS-SDR, que constituyen en sí mismos la evidencia técnica del funcionamiento del sistema. Entre estos se incluyen el fichero channel_tracking.csv, que muestra el seguimiento de los canales GNSS, así como el archivo .nmea, donde se recogen las soluciones de posición calculadas durante la sesión. Adicionalmente, se considera suficiente la información mostrada en tiempo real por pantalla a través de la salida estándar del sistema, sin necesidad de recurrir a receptores externos ni a herramientas adicionales de visualización. Esta aproximación simplifica el proceso de validación y asegura la trazabilidad completa desde un único entorno de ejecución.

7 • Finalización y archivo de sesión.

Una vez concluida la prueba, es importante detener correctamente los procesos en curso. Esto implica cerrar plutoplayer, ya sea desde la terminal o finalizando su PID (*Process Identifier*), y cerrar GNSS-SDR. Todos los archivos generados deben trasladarse a una carpeta identificada con la fecha del experimento, donde se incluya una nota con detalles como la potencia de transmisión estimada y la distancia entre el SDR emisor y el receptor víctima. Esta documentación permitirá contextualizar los resultados y garantizar la trazabilidad del experimento para futuras referencias.

5.5.2 PARÁMETROS QUE DEBEN PERMANECER ALINEADOS.

Para garantizar la coherencia entre transmisión y recepción durante el experimento, es esencial que ciertos parámetros se mantengan perfectamente sincronizados en todos los componentes del sistema. A continuación se describen los más críticos:

• **Frecuencia de muestreo:** Debe fijarse un valor idéntico de 2.600.000 muestras por segundo en todos los elementos implicados: en el parámetro *-f* de gps-sdr-sim, en el PlutoSDR durante la transmisión, en el HackRF para la recepción, y en el campo internal_fs_sps del fichero de configuración de GNSS-SDR. Una divergencia en este valor obliga a volver a muestrear la señal, lo que puede introducir errores de fase y pérdida de sincronización entre código y portadora.

• **Centro de frecuencia:** El valor de 1.575.420.000 Hz debe ser coherente en tres puntos clave: la cabecera del fichero binario generado, el LO del PlutoSDR durante la transmisión, y el campo SignalSource.center_frequency en el fichero de configuración del HackRF. Cualquier modificación en uno de estos valores debe replicarse en los demás para evitar desplazamientos espectrales que impidan la adquisición.

• **Nivel de potencia:** El parámetro -a de plutoplayer regula la amplitud digital de la señal emitida. Esta debe situarse alrededor de 3 dB por encima del nivel de C/N_0 que puede observar el HackRF, suficiente para que el receptor objetivo realice el *handover*. Si la potencia es excesiva, puede saturar los receptores móviles; si es insuficiente, el receptor puede continuar fijado a la señal auténtica.

• **Efemérides vigentes:** Es crucial utilizar el archivo BRDC correspondiente al día exacto del experimento. En casos en que la sesión se prolongue más allá de medianoche en tiempo GPS, puede ser necesario regenerar el binario con las efemérides del día siguiente para evitar inconsistencias y alertas de integridad. Las efemérides se pueden encontrar fácilmente en el recurso oficial de la NASA, en https://cddis.nasa.gov/archive/gnss/data/daily/, como podemos ver en la Figura 25.

Figura 25 Recurso oficial de la NASA para obtener efemérides por día.

• **Ancho de banda de los lazos DLL/PLL (GNSS-SDR):** Se recomienda emplear un ancho de 2 Hz para el DLL y 15 Hz para el PLL. Este ajuste representa un equilibrio eficaz entre estabilidad y agilidad de seguimiento bajo condiciones de alta SNR, evitando tanto la pérdida de traza como el exceso de *jitter* en la solución PVT.

5.5.3 CONSEJOS PRÁCTICOS Y PROBLEMAS HABITUALES.

A lo largo del desarrollo y prueba del sistema de *spoofing*, es común enfrentarse a una serie de incidencias prácticas que, si no se abordan con antelación, pueden comprometer la eficacia del experimento o dificultar la interpretación de los resultados. Este apartado recoge algunos de los problemas más habituales observados durante las sesiones experimentales, junto con recomendaciones para mitigarlos, basadas en la experiencia directa del autor y en buenas prácticas ampliamente compartidas en la comunidad GNSS-SDR.

Uno de los aspectos más críticos es la sincronización temporal. Antes de generar el fichero binario con gps-sdr-sim, es altamente recomendable alinear el reloj del sistema con un servidor NTP (*Network Time Protocol*) confiable. Aunque la herramienta construye internamente una constelación coherente, cualquier desfase de más de un segundo entre el tiempo de generación y el de recepción puede provocar alertas de integridad temporal en los receptores, reduciendo la credibilidad del ataque o impidiendo la adquisición.

Otro factor a tener en cuenta es la deriva del oscilador interno del PlutoSDR. Como dispositivo económico, este SDR utiliza un oscilador de cristal con una estabilidad limitada (~1 ppm), lo que puede derivar en desplazamientos de frecuencia detectables si la sesión supera los cinco minutos. En estos casos, puede optarse por reiniciar periódicamente la transmisión para reestablecer la coherencia espectral, o bien emplear un oscilador de referencia externo (TCXO, *Temperature Compensated Crystal Oscillator*) conectado al pin REF_CLK del dispositivo.

Durante la captura de datos con el HackRF, es posible que se produzcan *overruns*, evidenciados por mensajes "OOOO" en la consola. Estos indican que el sistema no ha procesado los datos a tiempo y ha perdido muestras. Para mitigar este problema, se recomienda reducir la ganancia de recepción o disminuir la tasa de muestreo a 2 MS/s, asegurándose de generar un nuevo archivo IQ con esa misma configuración. Estas medidas aligeran la carga del subsistema USB y del procesador, favoreciendo una recepción continua y estable.

También pueden producirse *underruns* en la transmisión con PlutoSDR, especialmente cuando se utiliza un puerto USB 2.0. Estos errores ocurren cuando el búfer del transmisor se vacía por falta de datos, debido a ráfagas de CPU o latencias en el sistema operativo. Una solución práctica consiste en ejecutar plutoplayer con la opción --force_bulk, que fuerza el uso de transferencias USB más robustas. Además, el uso de un cable USB corto y de buena calidad puede mejorar significativamente la estabilidad del enlace.

En cuanto al nivel de potencia de transmisión, conviene evitar tanto valores excesivos como insuficientes. Un nivel demasiado alto puede saturar el *front-end* del receptor, lo que en el caso de un dispositivo móvil puede traducirse no en una pérdida de señal simulada, sino en una desconexión total del GPS. El objetivo debe ser lograr que la señal falsificada se imponga a la real sin causar saturación; esto se refleja típicamente en un C/N_0 simulado de entre 18 y 20 dB-Hz, valor que puede verificarse con aplicaciones móviles de diagnóstico GNSS.

Por último, un consejo organizativo de gran utilidad es mantener una nomenclatura sistemática de los archivos generados. Dado que se pueden realizar múltiples pruebas en un mismo día con distintas configuraciones, es recomendable nombrar cada binario IQ incluyendo la fecha, la frecuencia de muestreo y la resolución, por ejemplo: "gps_20250527_2.6Msps_8b.bin". Esta práctica reduce significativamente la probabilidad de errores por confusión de archivos y facilita la trazabilidad de cada experimento.

Estas consideraciones operativas, aunque en apariencia menores, resultan determinantes para la ejecución exitosa y reproducible del experimento de *spoofing*. Su aplicación contribuye no solo a mejorar la estabilidad técnica del sistema, sino también a reforzar su valor como herramienta didáctica y de investigación.

5.6 VERIFICACIÓN DE RESULTADOS.

Una vez completada la integración de todos los elementos descritos en los apartados anteriores, se puede proceder a la ejecución práctica del ataque por *spoofing*. Este paso marca la culminación del diseño del sistema y la validación de su operatividad en condiciones controladas. Para llevarlo a cabo, se sigue el procedimiento ya detallado: el dispositivo Pluto SDR se configura como emisor de la señal GNSS falsificada, previamente generada con la herramienta GPS-SDR-SIM a partir de datos de efemérides reales. Simultáneamente, el receptor GNSS SDR, basado en el dispositivo HackRF One, se pone en escucha con la configuración específica descrita anteriormente, preparada para capturar y analizar la señal emitida.

Esta disposición experimental permite recrear de forma fiel un entorno de *spoofing*, en el que un receptor recibe señales coherentes pero manipuladas, generadas artificialmente para inducir errores en su posicionamiento. La sincronización entre ambos SDR, junto con el ajuste fino de parámetros como la potencia relativa de emisión, el timing y la trayectoria simulada, garantiza que la transición entre la señal legítima y la falsa sea suave y no detectable por mecanismos convencionales del receptor.

Como vemos en la Figura 26, el sistema inicia correctamente el receptor GNSS-SDR utilizando el fichero de configuración destinado al HackRF. En la traza se observa que el *front-end* identificado es Osmosdr_Signal_Source, lo que confirma que se está captando señal en tiempo real desde el hardware. Asimismo, el parámetro "RF Channels: 1" indica que se está gestionando un único flujo IQ, tal y como se ha definido en la arquitectura del sistema. El nivel de *log* configurado es "info", lo cual garantiza un equilibrio adecuado entre trazabilidad y rendimiento. Esta captura constituye el punto de partida del proceso de *spoofing*, al evidenciar que el subsistema receptor-víctima se encuentra operativo, correctamente enlazado con el SDR y libre de errores críticos de librerías, sincronización o ganancia.

Figura 26 Inicialización de la recepción de GNSS-SDR a través del HackRF.

En la Figura 27 se recoge el momento en que GNSS-SDR comienza a rastrear activamente la señal GPS falsificada. El sistema asigna con éxito ocho canales a distintos PRN en menos de un segundo, lo que demuestra una fase de adquisición eficaz mediante el método PCPS. Además, la variable "Current receiver time" avanza de forma coherente, sin interrupciones ni reinicios, lo que refleja la estabilidad del seguimiento. La ausencia de eventos de pérdida de señal o readquisición confirma que los lazos de control DLL y PLL se han fijado de forma robusta sobre la señal inyectada. Esta trazabilidad sin fisuras es una clara manifestación de que el *spoofing* ha comenzado con éxito y que el receptor está siendo guiado exclusivamente por la señal transmitida desde el Pluto SDR.

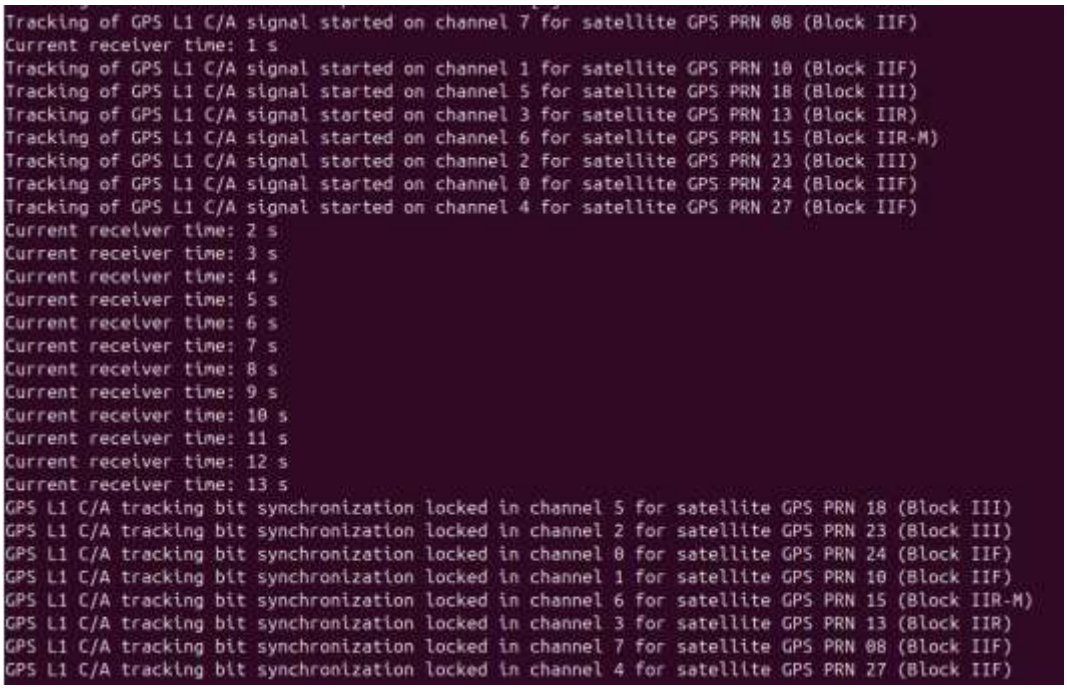

Figura 27 Comienzo de la identificación y enganche de los canales GPS.

La Figura 28 muestra la recepción y decodificación completa de las subtramas 4 y 5 del mensaje de navegación para todos los satélites en seguimiento. En este punto, el receptor no solo está rastreando las portadoras, sino que ya interpreta correctamente los datos NAV transmitidos por la señal espuria. Se destacan niveles de relación C/N_0 excepcionalmente altos, comprendidos entre 46 y 59 dB-Hz, lo cual es significativamente superior al rango habitual en entornos interiores (≈ 35–40 dB-Hz). Este parámetro refuerza la hipótesis de *spoofing*, ya que indica una potencia de señal artificialmente elevada, como es característico cuando se transmite desde una fuente SDR próxima al receptor. Además, se actualiza el fichero RINEX de navegación, reflejando que el sistema considera válidos estos datos y los incorpora como referencia para la solución de posicionamiento.

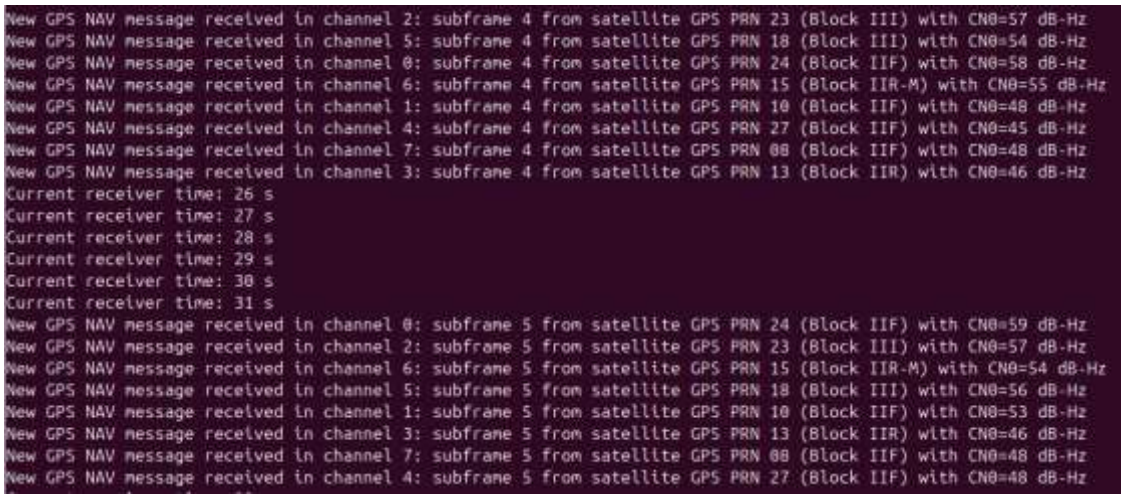

Figura 28 GNSS-SDR comienza a obtener los mensajes NAV del spoofing.

En la Figura 29 se representa la primera solución PVT obtenida por el sistema. Se observa que, apenas 24 segundos después del inicio del proceso, el receptor determina una ubicación con coordenadas latitud 40.758 y longitud –73.986, lo que corresponde a Times Square, Nueva York. Este resultado prueba de manera concluyente la eficacia del ataque por *spoofing*: el receptor, situado físicamente en una sala de ensayo, ha sido inducido a calcular una posición completamente falsa. La solución se basa en cinco satélites, lo cual proporciona una geometría suficientemente robusta. Además, las velocidades medidas se sitúan en torno a 0 m/s, coherentes con un receptor estático, y la altura se mantiene dentro del rango esperado dadas las diferencias entre geoide y elipsoide. La estabilidad de esta solución a lo largo del tiempo refuerza la validez del ataque y sugiere una simulación completa y coherente de la constelación GNSS.

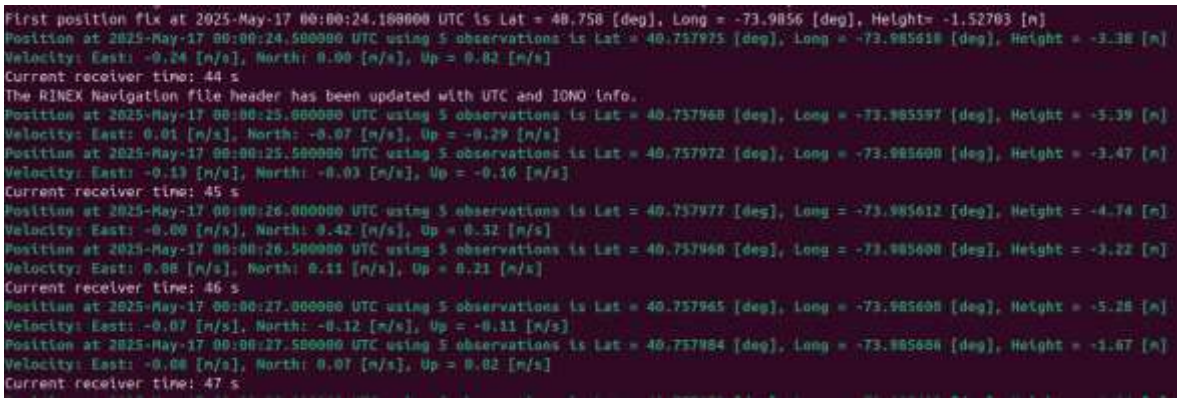

Figura 29 GNSS-SDR obtiene la PVT emitida por gps-sdr-sim.

CAPÍTULO 6 - PRÓXIMOS PASOS Y LÍNEAS DE TRABAJO.

Tras el desarrollo del entorno experimental y la validación de las técnicas de *jamming* y *spoofing*, se abre la puerta a múltiples líneas de evolución y aplicación. Este capítulo explora las posibilidades futuras tanto desde un enfoque técnico como pedagógico, proponiendo mejoras concretas que refuercen la funcionalidad del sistema y amplíen su alcance. Se abordan iniciativas viables a corto plazo, como la automatización del entorno o la integración docente, junto con planteamientos más ambiciosos a medio y largo plazo, como la evaluación de contramedidas o la extensión multibanda. El objetivo es establecer una hoja de ruta realista que permita transformar este banco de pruebas en una plataforma versátil y sostenible.

6.1 MEJORAS DEL ENTORNO A CORTO PLAZO.

El entorno actual ha demostrado su eficacia como banco de pruebas accesible y funcional, pero presenta márgenes claros de mejora. En este apartado se identifican y agrupan diversas propuestas técnicas orientadas a optimizar su despliegue, precisión, escalabilidad y usabilidad.

• **Automatización y despliegue:** La consolidación del entorno en un único script ejecutable (run.sh), complementado por un Makefile, permitiría automatizar la configuración del reloj, el lanzamiento de gps-sdr-sim, la inicialización de GNSS-SDR y la grabación simultánea de datos. Asimismo, el desarrollo de imágenes Docker o Podman con GNSS-SDR ya compilado facilitaría la portabilidad del sistema y su implementación tanto en ordenadores personales como en plataformas embebidas como Raspberry Pi 4. Como refuerzo, la integración de flujos de integración continua mediante GitHub Actions permitiría generar binarios y realizar pruebas básicas (*smoke tests*) de forma automática tras cada confirmación en el repositorio.

• **Calidad de señal:** Para mejorar la estabilidad de frecuencia y reducir el *jitter*, se plantea la inclusión de un TCXO de 0,5 ppm o incluso un oscilador OCXO (*Oven Controlled Crystal Oscillator*) sincronizado por GPS (*GPS-disciplined*). Este último permitiría aprovechar la entrada REF_CLK del dispositivo PlutoSDR. En paralelo, el uso de atenuadores fijos junto con medidores de potencia USB permitiría calibrar con mayor precisión los niveles de potencia en dBm reales. Además, la incorporación de un filtro SAW específico para la banda L1 y un preselector conmutado para L1/L5 contribuiría a mejorar la relación señal/ruido y a reducir interferencias fuera de banda.

• **Escalabilidad del hardware:** Se propone encapsular los componentes en un *shield* metálico con conectores accesibles desde el panel frontal, configurando así un kit compacto y funcional tipo "todo-en-uno". Este podría complementarse con un sistema de *front-end* doble que utilice simultáneamente dos SDR (por ejemplo, PlutoSDR y HackRF o LimeSDR), permitiendo operaciones de captura y emisión multibanda en paralelo. Por último, la sincronización temporal entre equipos mediante un pulso por segundo externo habilitaría escenarios experimentales más complejos, con múltiples dispositivos operando de forma coherente.

• **Usabilidad:** Para facilitar su adopción y uso en entornos académicos, se contempla el desarrollo de un panel web ligero, basado en Flask, que permita visualizar en tiempo real gráficos de relación C/N_0 y ejecutar funciones clave mediante botones predefinidos como "JAM" o "SPOOF". Esta interfaz se complementaría con notebooks de Jupyter preconfigurados que permitirían replicar las demostraciones en menos de 10 minutos, ofreciendo una experiencia reproducible y accesible para usuarios sin conocimientos avanzados.

6.2 MEJORAS DEL ENTORNO A LARGO PLAZO.

Las siguientes líneas de trabajo están concebidas como extensiones de mayor complejidad técnica y planificación, orientadas a expandir significativamente las capacidades del entorno. Su implementación requeriría recursos adicionales, pero abriría posibilidades muy valiosas

para la investigación, la docencia avanzada y el análisis de escenarios reales de seguridad GNSS.

Una primera propuesta consiste en el desarrollo de un laboratorio remoto accesible vía WebRTC, en el que el PlutoSDR permanezca conectado a un servidor accesible por red. Esta infraestructura permitiría a estudiantes y usuarios autorizados ejecutar pruebas de interferencia de forma remota, en cualquier momento del día, maximizando la disponibilidad del entorno y facilitando la experimentación continua sin restricciones geográficas.

Desde el punto de vista técnico, una de las mejoras más relevantes sería la ampliación del sistema a señales multiconstelación y multifrecuencia. En particular, se plantea incorporar soporte para Galileo E1/E5 y GPS L5 mediante dispositivos como el LimeSDR Mini 2.0. Esta evolución permitiría estudiar la respuesta de receptores frente a interferencias que afectan simultáneamente a múltiples bandas o constelaciones, elevando el realismo y el valor experimental del banco de pruebas.

Asimismo, se propone desarrollar una arquitectura de "ataque en la nube", basada en contenedores con aceleración por GPU (*Graphic Processing Unit*) capaces de generar señales IQ en tiempo real. Estas señales podrían ser transmitidas vía RTP (*Real-Time Transport Protocol*) a un emisor remoto, eliminando la dependencia del hardware local y permitiendo centralizar la generación de señales en entornos controlados o de alta capacidad computacional.

Otra vía de exploración consiste en realizar campañas de campo controladas utilizando una caja de Faraday portátil, alimentada por baterías, que permita desplegar el sistema en exteriores sin riesgo de emisiones no deseadas. Esta configuración permitiría registrar de forma sistemática el espectro electromagnético durante los experimentos, generando conjuntos de datos abiertos para su análisis académico o divulgación científica.

Por último, se plantea la investigación sobre técnicas de *spoofing* dirigidas a vehículos no tripulados, como drones. La incorporación de un módulo de transmisión a 433 MHz permitiría inyectar señales falsificadas directamente en el autopiloto del UAV (*Unmanned*

Aerial Vehicle), evaluando su comportamiento ante rutas manipuladas o escenarios de evasión. Además, este entorno serviría para estudiar nuevas técnicas de autenticación GNSS, como OSNMA o CHIMERA, analizando tanto su resiliencia ante ataques como posibles vectores de bypass en contextos avanzados. [22]

6.3 APLICACIÓN DOCENTE Y EDUCATIVA.

El entorno experimental desarrollado en este proyecto presenta un notable potencial como herramienta docente en titulaciones relacionadas con las telecomunicaciones, la ingeniería electrónica y la ciberseguridad. Su carácter reproducible, modular y de bajo coste lo convierte en un recurso idóneo para ilustrar de forma práctica conceptos que a menudo se abordan únicamente desde una perspectiva teórica. Al permitir observar de primera mano los efectos de interferencias intencionadas sobre sistemas GNSS reales, el entorno fomenta una comprensión profunda y crítica de los principios de posicionamiento, sincronización y robustez de señales satelitales.

A nivel metodológico, la estructura del entorno se presta a una integración flexible en asignaturas de grado y máster. Puede utilizarse como base para prácticas de laboratorio, proyectos individuales o trabajo en grupo, y también como soporte para trabajos fin de grado o de máster centrados en ciberdefensa, sistemas embebidos o radiofrecuencia. Además, al emplear herramientas libres como GNU Radio, GNSS-SDR o gps-sdr-sim, los estudiantes se familiarizan con ecosistemas software profesionales de uso creciente en la industria y la investigación. Esta experiencia práctica no solo refuerza su formación técnica, sino que también estimula su capacidad de análisis, resolución de problemas y trabajo autónomo.

Una ventaja especialmente destacable del entorno es su transparencia funcional, derivada del uso de herramientas abiertas y técnicas en lugar de soluciones comerciales cerradas. Gracias a aplicaciones como gps-sdr-sim y GNSS-SDR, es posible visualizar y comprender en detalle tanto el proceso de generación de los mensajes de navegación como el mecanismo de recepción, correlación y cálculo de posición. Esta visibilidad total del flujo de datos —desde los bits del mensaje hasta la solución final 3D+T— convierte cada práctica en una

experiencia formativa completa, donde los estudiantes pueden intervenir, modificar y analizar con precisión cada etapa del sistema.

A continuación se describen algunos ejemplos concretos de aplicación docente que pueden implementarse con el entorno descrito:

• **Práctica sobre degradación de C/N₀ y pérdida de *fix*:** El entorno permite introducir interferencias graduales mientras se monitoriza la relación señal/ruido en tiempo real. Los estudiantes pueden observar cómo se eleva el HDOP y en qué momento el receptor pierde el posicionamiento, lo que facilita la comprensión del umbral operativo de los sistemas GNSS.

• **Simulación controlada de *spoofing*:** A través de un cuaderno Jupyter guiado, los alumnos pueden generar una señal falsa que arrastra lentamente la posición estimada. Comparando la trayectoria real y la manipulada, se analizan los efectos del *spoofing* y se discute la falta de autenticación en la señal civil L1.

• **Detección de interferencias mediante análisis de métricas GNSS:** Se proponen ejercicios de identificación de eventos de *jamming* o *spoofing* basados en la observación de parámetros internos como AGC, C/N₀ o desplazamiento Doppler. Esta práctica introduce al alumno en el diagnóstico forense de anomalías GNSS.

• **Talleres de divulgación o jornadas técnicas:** Gracias a su portabilidad y facilidad de despliegue, el entorno es ideal para demostraciones en tiempo real durante ferias o actividades de divulgación. Se puede mostrar visualmente cómo funciona el GPS y cómo puede ser alterado, promoviendo la concienciación sobre su fragilidad y la necesidad de protección.

6.4 PLATAFORMA PARA EVALUAR CONTRAMEDIDAS.

El entorno desarrollado no solo permite replicar interferencias GNSS de forma controlada, sino que también ofrece una base sólida para probar mecanismos de defensa frente a *jamming*

y *spoofing*. Su modularidad y acceso al flujo interno de señales lo convierten en una plataforma ideal para validar algoritmos de detección, estrategias de mitigación o mejoras en la recepción GNSS. Al operar en un entorno cerrado y reproducible, los ensayos pueden repetirse con precisión, permitiendo comparar de forma objetiva distintas contramedidas.

Además, al tratarse de un sistema completamente abierto, es posible inyectar datos externos, monitorizar métricas internas y adaptar el comportamiento del receptor en tiempo real, lo que facilita el estudio experimental de nuevas técnicas de protección.

Ejemplos concretos de aplicación:

• **Test de algoritmos de detección por C/N₀ y AGC:** Validación de umbrales y lógica de alerta ante escenarios de interferencia progresiva.

• **Comparativa entre receptores multiconstelación:** Evaluación del efecto protector de Galileo + GPS frente a ataques L1 exclusivos.

• **Integración de filtros *notch* dinámicos:** Medición de su eficacia ante *jamming* tonal y barrido.

• **Simulación de OSNMA/CHIMERA mediante señal sintética:** Ensayos de robustez frente a *spoofing* de navegación.

CAPÍTULO 7 - CONTRIBUCIÓN A LOS OBJETIVOS DE DESARROLLO SOSTENIBLE (ODS).

El presente trabajo se alinea con varios Objetivos de Desarrollo Sostenible definidos por Naciones Unidas, al contribuir tanto en el ámbito educativo como en la resiliencia tecnológica y la equidad en el acceso a recursos de experimentación avanzada. A continuación se detallan los más relevantes:

• **ODS 4 – Educación de calidad:** El entorno es abierto, económico y totalmente documentado, lo que permite reproducir ataques de *jamming* y *spoofing* en un entorno docente. Esta capacidad práctica fortalece la enseñanza de conceptos avanzados en ingeniería y ciberseguridad, especialmente en áreas como radiofrecuencia, sincronización temporal y navegación satelital. Además, facilita el aprendizaje basado en la experimentación directa, promoviendo vocaciones STEM (*Science, Technology, Engineering and Mathematics*) y fortaleciendo competencias clave en el alumnado.

• **ODS 9 – Industria, innovación e infraestructuras:** El proyecto permite demostrar de forma práctica la fragilidad del sistema GPS y ensayar contramedidas de bajo coste. Esto contribuye directamente a mejorar la resiliencia de infraestructuras críticas como redes de telecomunicaciones, sistemas de transporte o servicios energéticos. Al mismo tiempo, fomenta el desarrollo de soluciones tecnológicas innovadoras en torno a la detección y mitigación de interferencias, abriendo líneas de colaboración entre academia, industria y administración.

• **ODS 10 – Reducción de las desigualdades:** El bajo presupuesto requerido para reproducir el entorno (< 250 €) hace accesible una capacidad experimental que antes estaba reservada a centros con equipos superiores a los 10,000 €. Esto reduce la brecha tecnológica entre instituciones con distintos niveles de recursos, permitiendo que universidades de países en desarrollo accedan a una plataforma de investigación avanzada. Asimismo, se promueve una participación más equitativa en proyectos internacionales y redes de colaboración científica.

• **ODS 11 – Ciudades y comunidades sostenibles:** El uso del entorno para analizar y prevenir interferencias GNSS contribuye a la seguridad de sistemas que sustentan la movilidad inteligente, la logística urbana y los servicios de emergencia. Al garantizar la continuidad de estos servicios ante eventos disruptivos, se mejora la resiliencia de las ciudades frente a amenazas tecnológicas. El trabajo impulsa además la creación de protocolos de respuesta local ante ataques de señal, fortaleciendo la sostenibilidad operativa de los entornos urbanos.

• **ODS 16 – Paz, justicia e instituciones sólidas:** Al visibilizar la amenaza que suponen las interferencias GNSS y ofrecer un entorno seguro para estudiarlas, el proyecto se convierte en una herramienta estratégica para reguladores, cuerpos de seguridad y responsables de políticas públicas. Permite diseñar normativas más robustas, evaluar escenarios de ataque controlados y formar a personal especializado en ciberdefensa GNSS. En última instancia, refuerza la capacidad institucional para proteger infraestructuras clave frente a usos maliciosos. [23]

CAPÍTULO 8 - CONCLUSIONES.

Este capítulo recoge las principales conclusiones extraídas del trabajo realizado, centrándose en tres ejes fundamentales: la fragilidad estructural del sistema GPS, el grado de cumplimiento de los objetivos planteados y el impacto económico y tecnológico del entorno desarrollado. A través de pruebas prácticas, se ha evidenciado que incluso con medios limitados es posible comprometer el funcionamiento del GPS civil, lo que subraya la necesidad de revisar su arquitectura actual. Asimismo, se valoran los logros alcanzados en términos de replicabilidad, accesibilidad y utilidad formativa.

En estrecha relación con estas conclusiones, el capítulo 6 ha recogido una serie de propuestas de mejora y líneas de trabajo futuro que derivan directamente de los resultados obtenidos. Estas propuestas, que abarcan desde mejoras técnicas hasta aplicaciones docentes y estrategias de protección, constituyen uno de los aportes más relevantes del proyecto. Por ello, se remite al lector interesado a dicho apartado para evitar repeticiones innecesarias.

8.1 FRAGILIDAD DEL GPS DEMOSTRADA.

Uno de los principales objetivos del presente trabajo ha sido demostrar de forma empírica la fragilidad del sistema GPS civil frente a interferencias activas, tanto de *jamming* como de *spoofing*. Esta fragilidad, ya intuida por la baja potencia de la señal L1 C/A y la ausencia de mecanismos de autenticación, adquiere una dimensión crítica si se considera el papel esencial que el GPS desempeña en sectores clave como las telecomunicaciones, la energía, la banca o la logística. La dependencia de esta señal para la sincronización y el posicionamiento de infraestructuras vitales convierte cualquier vulnerabilidad técnica en una amenaza potencial de gran escala, capaz de desencadenar fallos en cascada con consecuencias sistémicas. Por ello, confirmar esta debilidad mediante ensayos prácticos no solo responde a un interés académico, sino que adquiere un valor estratégico en el ámbito de la seguridad tecnológica.

A lo largo del proyecto se ha logrado reproducir con éxito interferencias significativas utilizando medios extremadamente reducidos. Se han generado ataques de *jamming* capaces de interrumpir el posicionamiento de un receptor en menos de cinco segundos y escenarios de *spoofing* que introducen errores de localización superiores a los 100 metros sin activar ningún sistema de alerta en el receptor. Estas pruebas se han llevado a cabo en un entorno controlado, con un presupuesto total inferior a 250 euros, empleando dispositivos SDR y software libre. Además, se ha constatado que basta un incremento de unos pocos dB respecto a la señal legítima para suplantarla con éxito, lo que confirma la escasa protección que ofrece el sistema actual frente a interferencias externas. Estos resultados, en línea con incidentes reales como el caso del dron RQ-170 en 2011 o los episodios de *spoofing* marítimo en 2022, consolidan la evidencia de que la señal civil GPS puede ser manipulada o anulada con facilidad, incluso por actores con recursos limitados. El entorno reproducible desarrollado en este trabajo permite, además, observar esta fragilidad con gran precisión, evidenciando de forma directa las carencias estructurales del sistema.

En este sentido, el trabajo debe interpretarse como una llamada de atención ante una vulnerabilidad latente que sigue sin ser abordada de forma sistemática. La facilidad con la que se pueden replicar estos ataques pone de relieve la necesidad urgente de introducir mecanismos de autenticación en las señales civiles, así como de promover el uso de receptores con algoritmos de detección robusta frente a anomalías. Más allá del valor experimental del entorno creado, este proyecto aspira a contribuir al debate sobre la resiliencia de las infraestructuras GNSS y a incentivar el desarrollo de tecnologías y políticas públicas que refuercen la seguridad de un sistema cuya fiabilidad ya no puede darse por supuesta.

8.2 RESUMEN DE LOGROS.

En este apartado se examina el grado de cumplimiento de los objetivos definidos al inicio del proyecto, evaluando su consecución y aportando consideraciones relevantes sobre los resultados obtenidos.

• **O1 – Reproducir un ataque de *jamming* efectivo en entorno doméstico:** Objetivo plenamente alcanzado. Se ha implementado un sistema de *jamming* de banda ancha utilizando un PlutoSDR, logrando interrumpir la señal GNSS y provocar la pérdida de *fix* en menos de cinco segundos. Este comportamiento ha sido validado mediante varias aplicaciones GNSS comerciales y análisis visual de los parámetros de posicionamiento (HDOP, C/N_0), confirmando su efectividad en condiciones reales controladas.

• **O2 – Ejecutar un ataque de *spoofing* con error significativo sin alertas:** Objetivo también alcanzado. Se ha conseguido realizar un ataque de tipo "*carry-off*", en el que el receptor fue arrastrado suavemente hasta posiciones falsas con errores superiores a los 100 metros, sin generar *flags* de integridad ni mensajes de error en la interfaz del receptor. La transición se efectuó con un incremento controlado de potencia (≈ +3 dB), lo que permitió un *handover* sin disrupciones, replicando un escenario realista de *spoofing* avanzado.

• **O3 – Documentar el entorno para su réplica completa:** Objetivo completamente logrado. Se ha desarrollado una guía paso a paso detallada. Esta documentación permite replicar las pruebas de *jamming* y *spoofing* desde cero, facilitando su uso en contextos docentes o de investigación.

• **O4 – Elaborar materiales técnicos y de divulgación:** Objetivo cumplido. Se ha entregado una memoria técnica completa que detalla todo el proceso de diseño, implementación y validación, además de un vídeo demostrativos donde se realizan ambas interferencias.. Ambos materiales se incluyen como anexos y están disponibles para consulta y difusión académica.

• **Logro adicional – Entorno *low-cost*, *open-source* y portátil:** Aunque no estaba formulado explícitamente como objetivo, se ha conseguido desarrollar un banco de pruebas funcional con un coste total inferior a los 250 €, empleando únicamente software libre y hardware comercial asequible. Esta característica añade un valor significativo al proyecto, al democratizar el acceso a capacidades experimentales avanzadas.

8.3 IMPACTO ECONÓMICO.

Este apartado tiene como objetivo valorar las implicaciones económicas del entorno experimental desarrollado, tanto en términos de ahorro directo como de potencial de escalabilidad y uso futuro. Se considera no solo la comparativa con soluciones comerciales existentes, sino también su aplicabilidad en contextos educativos, industriales y de investigación, así como su capacidad para reducir pérdidas asociadas a interrupciones GNSS. A continuación, se presentan los principales vectores de impacto económico identificados:

Comparativa de costes.

El coste total del entorno propuesto, incluyendo el hardware necesario y el uso de software libre, se sitúa por debajo de los 250 euros. En contraste, los simuladores comerciales de señales GNSS, como los ofrecidos por Spirent o Racelogic, oscilan entre los 10.000 y los 30.000 euros, dependiendo de la configuración y los módulos adquiridos. Esto supone una reducción del coste superior al 98 %, permitiendo a instituciones sin grandes presupuestos acceder a capacidades técnicas que antes estaban fuera de su alcance. Esta diferencia de escala transforma radicalmente el acceso a la experimentación GNSS.

Ahorro en formación e I+D.

El entorno permite montar un laboratorio "*in-house*" para el estudio de interferencias GNSS, evitando así el alquiler de cámaras anecoicas, generadores de señal de alta gama o servicios de simulación externos. Un curso completo de prácticas con equipamiento similar alquilado podría tener un coste estimado superior a los 5.000 € anuales por grupo. En cambio, el sistema propuesto permite realizar prácticas recurrentes y ajustadas al currículum con una inversión inicial mínima, favoreciendo la sostenibilidad de programas formativos en universidades públicas y centros tecnológicos emergentes.

Modelos de explotación.

El proyecto abre la puerta a distintas formas de explotación económica. Por ejemplo, se podría comercializar un kit educativo preensamblado con el hardware necesario,

acompañado de una plataforma formativa online, con un precio objetivo en torno a 400–600 €. Alternativamente, se contempla la idea de ofrecer un servicio en la nube que permita generar señales GNSS en tiempo real desde contenedores Docker con aceleración por GPU, con acceso al hardware SDR en remoto bajo un modelo de alquiler por horas. También sería viable ofrecer servicios de consultoría técnica para evaluar la resiliencia GNSS en infraestructuras críticas, con tarifas competitivas frente a soluciones especializadas.

Valor intangible y costes evitados.

La señal GNSS actúa como infraestructura invisible en sectores clave como la logística, la aviación o la agricultura de precisión. Según estudios del Departamento de Comercio de EE. UU., cada hora de interrupción completa del GPS puede suponer pérdidas superiores a 1.000 millones de dólares solo en el sector agrícola. En Reino Unido, un apagón GNSS de cinco días podría costar más de 5.000 millones de libras. En este contexto, este proyecto contribuye a la reducción del riesgo asociado a dichas pérdidas, al permitir formar técnicos cualificados, evaluar escenarios de ataque y anticipar fallos potenciales antes de que se materialicen.

Escalabilidad y crecimiento potencial.

La arquitectura modular del entorno permite su ampliación mediante componentes adicionales, como módulos multibanda (L1/L5), compatibilidad con señales autenticadas como OSNMA o integración en redes colaborativas de sensores. Además, puede diferenciarse entre licencias académicas, de bajo coste para enseñanza e investigación, y licencias empresariales destinadas a entornos industriales o comerciales. Este modelo escalable permite adaptarse a distintos mercados sin comprometer la accesibilidad ni la sostenibilidad económica del proyecto.

BIBLIOGRAFÍA

[1] International Telecommunication Union. (2020). *Radio Regulations: Volume 1, Article 5 – Frequency Allocations* (2020 ed.). ITU. https://www.itu.int/pub/R-REG-RR

[2] European Telecommunications Standards Institute. (2023). *EN 302 645 V1.1.1: Global Navigation Satellite Systems (GNSS) Repeaters.* ETSI. https://docdb.cept.org/document/28547

[3] European Parliament & Council. (2024). *Directive 2014/65/EU on markets in financial instruments (MiFID II) — consolidated text 28-03-2024.* EUR-Lex. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32014L0065

[4] European Union Aviation Safety Agency. (2024). *Safety Information Bulletin SIB 2024-01 R1: Increased GNSS Interference and Spoofing.* EASA. https://www.caa.gov.cz/wp-content/uploads/2024/01/EASA_SIB_2024_01_R1.pdf

[5] RTI International. (2019). *Economic Benefits of the Global Positioning System (GPS).* U.S. Department of Commerce. https://www.rti.org/sites/default/files/gps_finalreport.pdf

[6] London Economics. (2017). *Economic impact to the UK of a disruption to GNSS.* UK Space Agency. https://www.gov.uk/government/publications/report-the-economic-impact-on-the-uk-of-a-disruption-to-gnss

[7] International Civil Aviation Organization. (2023). *Global Navigation Satellite System (GNSS) Manual* (Doc 9849, 3.ª ed.). ICAO. https://store.icao.int/en/global-navigation-satellite-system-gnss-manual-doc-9849

[8] U.S. Space Force. (2022). *Navstar GPS Space Segment / Navigation User Interfaces (IS-GPS-200N).* GPS Directorate. https://www.gps.gov/technical/icwg/IS-GPS-200N.pdf

[9] Humphreys, T. E. (2012). Civil GPS spoofing vulnerability assessment. *GPS World.* https://web.archive.org/web/20240210092257/https://www.gpsworld.com/drone-hack

[10] Psiaki, M. L., & Humphreys, T. E. (2016). GNSS spoofing and detection. *Proceedings of the IEEE, 104*(6), 1258-1270. https://doi.org/10.1109/JPROC.2016.2526658

[11] Federal Communications Commission. (2013). *Notice of Apparent Liability for Forfeiture: $31 875 GPS-jammer case (DA-14-1505).* FCC. https://www.fcc.gov/document/32k-penalty-proposed-use-gps-jammer-individual

[12] Center for Advanced Defense Studies. (2019). *Above Us Only Stars: Exposing GPS Spoofing in Russia and Syria.* C4ADS. https://c4ads.org/reports/above-us-only-stars

[13] Analog Devices. (2024). *ADALM-PLUTO Active Learning Module – Overview & User Guide.* Analog Devices Wiki. https://wiki.analog.com/university/tools/pluto

[14] Great Scott Gadgets. (2024). *HackRF One hardware specifications (rev. r10).* GitHub. https://github.com/greatscottgadgets/hackrf/wiki

[15] The Qt Company. (2023). *Qt 5.15.2 LTS – Release Notes.* https://wiki.qt.io/Qt_5.15_Release

[16] GNU Radio Project. (2025). *GNU Radio 3.10 Documentation – Main Page.* https://wiki.gnuradio.org

[17] Pothosware. (2023, May 1). *PothosSDR Windows Installer (Release 2023-05).* GitHub. https://github.com/pothosware/PothosSDR

[18] Pothosware. (2021). *SoapySDR 0.8.0 – API documentation.* https://pothosware.github.io/SoapySDR/

[19] Ebinuma, T., & Contributors. (2025). *gps-sdr-sim (Version 2.0) [Software]*. GitHub. https://github.com/osqzss/gps-sdr-sim

[20] GNSS-SDR Contributors. (2025). *GNSS-SDR (Version 0.0.20) [Software]*. GitHub. https://github.com/gnss-sdr/gnss-sdr

[21] OLK Developers. (2025). *VOLK – Vector-Optimized Library of Kernels (v3.0)*. GitHub. https://github.com/gnuradio/volk

[22] Wang, J., Li, D., Ning, J., Lin, J., Wang, Z., & Duo, B. (2025). *An improved UAV countermeasure strategy based on GPS spoofing*. In Z. H. Zhu et al. (Eds.), *Proceedings of the International Conference on Autonomous Unmanned Aerial Systems (ICAUAS 2024)* (pp. 370–378). Springer. https://doi.org/10.1007/978-981-96-3240-4_41

[23] United Nations. (2015). *Transforming our world: The 2030 agenda for sustainable development*. United Nations. https://sdgs.un.org/2030agenda