

**UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA,
INFORMÁTICA Y MECÁNICA**

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



TESIS

**ANÁLISIS DEL RENDIMIENTO DE UN PROTOTIPO DE
RADIO COGNITIVO SOBRE SDR EN APLICACIONES WSN**

PRESENTADO POR:

Br. WINER ANDREE RIOS RADO

**PARA OPTAR AL TÍTULO PROFESIONAL
DE INGENIERO ELECTRÓNICO**

ASESOR:

Dr. JORGE LUIS ARIZACA CUSICUNA

CUSCO - PERÚ

2026



Universidad Nacional de San Antonio Abad del Cusco

INFORME DE SIMILITUD

(Aprobado por Resolución Nro.CU-321-2025-UNSAAC)

El que suscribe, el Asesor JORGE LUIS ARIZACA CUSICUNA.....
..... quien aplica el software de detección de similitud al
trabajo de investigación/tesis titulada: ANALISIS DEL RENDIMIENTO DE UN
PROTOTIPO DE RADIO COGNITIVO SOBRE SDR EN APLICACIONES
WSN.....

Presentado por: WINER ANDREE RIOS RADO..... DNI N° 71726831.....;
presentado por: DNI N°:
Para optar el título Profesional/Grado Académico de INGENIERO ELECTRONICO.....

Informo que el trabajo de investigación ha sido sometido a revisión por 02 veces, mediante el
Software de Similitud, conforme al Art. 6° del **Reglamento para Uso del Sistema Detección de
Similitud en la UNSAAC** y de la evaluación de originalidad se tiene un porcentaje de 8 %.

Evaluación y acciones del reporte de coincidencia para trabajos de investigación conducentes a grado académico o título profesional, tesis

Porcentaje	Evaluación y Acciones	Marque con una (X)
Del 1 al 10%	No sobrepasa el porcentaje aceptado de similitud.	☒
Del 11 al 30 %	Devolver al usuario para las subsanaciones.	☐
Mayor a 31%	El responsable de la revisión del documento emite un informe al inmediato jerárquico, conforme al reglamento, quien a su vez eleva el informe al Vicerrectorado de Investigación para que tome las acciones correspondientes; Sin perjuicio de las sanciones administrativas que correspondan de acuerdo a Ley.	☐

Por tanto, en mi condición de Asesor, firmo el presente informe en señal de conformidad y **adjunto** las primeras páginas del reporte del Sistema de Detección de Similitud.

Cusco, 26 de JUNIO de 2026


.....
Firma

Post firma Jorge Luis Arizaca Cusicuna.....

Nro. de DNI 42348906.....

ORCID del Asesor 0000-0003-2658-5492.....

Se adjunta:

1. Reporte generado por el Sistema Antiplagio.
2. Enlace del Reporte Generado por el Sistema de Detección de Similitud: **oid:** 272598602774419.....

Winer Rios Rado

Tesis_Radio_Cognitivo_sustentacion.pdf

 Universidad Nacional San Antonio Abad del Cusco

Detalles del documento

Identificador de la entrega

trn:oid:::27259:602774119

Fecha de entrega

25 jun 2026, 11:09 p.m. GMT-5

Fecha de descarga

25 jun 2026, 11:13 p.m. GMT-5

Nombre del archivo

Tesis_Radio_Cognitivo_sustentacion.pdf

Tamaño del archivo

5.8 MB

153 páginas

35.574 palabras

185.122 caracteres

8% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Coincidencias menores (menos de 8 palabras)

Fuentes principales

- 6%  Fuentes de Internet
- 2%  Publicaciones
- 6%  Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alerta de integridad para revisión

-  **Texto oculto**
76 caracteres sospechosos en N.º de páginas
El texto es alterado para mezclarse con el fondo blanco del documento.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

PRESENTACIÓN

SEÑOR: DECANO DE LA FACULTAD DE INGENIERÍA ELECTRÓNICA DE LA UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO

SEÑOR: DIRECTOR DE LA ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

SEÑORES: DOCENTES MIEMBROS DEL JURADO

Con la conformidad del reglamento de grados y títulos de la Escuela Profesional de Ingeniería Electrónica, pongo a consideración el presente trabajo de tesis intitulado: **“ANÁLISIS DEL RENDIMIENTO DE UN PROTOTIPO DE RADIO COGNITIVO SOBRE SDR EN APLICACIONES WSN”** para optar al Título Profesional de **INGENIERO ELECTRÓNICO**.

Atentamente,

Br. Winer Andree Rios Rado

DEDICATORIA

En primer lugar quiero agradecer a mis padres, por haberme guiado, apoyado y cuidado. A mis docentes por todo el conocimiento brindado en estos años de carrera profesional. Deseo agradecer a todo aquel que me impulso a seguir adelante, amigos, compañeros y todo aquel que me dio la mano en este proceso.

Winer Andree Rios Rado

ABREVIATURAS

AGC	Automatic Gain Control / Control Automático de Ganancia
BER	Bit Error Rate / Tasa de Error de Bits
BLE	Bluetooth Low Energy / Bluetooth de Bajo Consumo
BW	Bandwidth / Ancho de Banda
CFAR	Constant False Alarm Rate / Tasa Constante de Falsas Alarmas
CA-CFAR	Cell-Averaging CFAR / CFAR por Promediado de Celdas
CFO	Carrier Frequency Offset / Desplazamiento de Frecuencia de Portadora
CR	Coding Rate / Tasa de Codificación
CRC	Cyclic Redundancy Check / Verificación de Redundancia Cíclica
CRSEQ	Channel Rendezvous Sequence / Secuencia de encuentro de canal
CSS	Chirp Spread Spectrum / Espectro Ensanchado por Chirps
CUT	Cell Under Test / Celda Bajo Prueba
CW	Continuous Wave / Onda Continua
DRSEQ	Deterministic Rendezvous Sequence / Secuencia de encuentro determinista
EBG	Electromagnetic Band Gap / Banda prohibida electromagnética
ED	Energy Detection / Detección de Energía
EMA	Exponential Moving Average / Media Móvil Exponencial
EMA-SNR	Exponential Moving Average Signal-to-Noise Ratio / Promedio Móvil Exponencial de SNR

EPFL	École Polytechnique Fédérale de Lausanne / Escuela Politécnica Federal de Lausana
FCC	Federal Communications Commission / Comisión Federal de Comunicaciones
FEC	Forward Error Correction / Corrección de Errores Directa
FFT	Fast Fourier Transform / Transformada Rápida de Fourier
FSK	Frequency Shift Keying / Modulación por Desplazamiento de Frecuencia
GRC	GNU Radio Companion / Entorno Gráfico de GNU Radio
GSMA	Global System for Mobile Communications Association / Asociación del Sistema Global para Comunicaciones Móviles
IEEE	Institute of Electrical and Electronics Engineers / Instituto de Ingenieros Eléctricos y Electrónicos
IoT	Internet of Things / Internet de las Cosas
ISM	Industrial, Scientific and Medical / Bandas Industriales, Científicas y Médicas
LoRa	Long Range / Largo Alcance
LoRaWAN	LoRa Wide Area Network / Red de Área Amplia LoRa
LPWAN	Low-Power Wide Area Network / Red de Área Amplia de Bajo Consumo
MAC	Medium Access Control / Control de Acceso al Medio
MCA	Morphological Component Analysis / Análisis de componentes morfológicos
MMCA	Multichannel Morphological Component Analysis / Análisis de componentes morfológicos multicanal
OOT	Out-Of-Tree / Módulo Externo en GNU Radio
PER	Packet Error Rate / Tasa de Error de Paquetes
Pfa	Probability of False Alarm / Probabilidad de Falsa Alarma

PHY	Physical Layer / Capa Física
PSD	Power Spectral Density / Densidad Espectral de Potencia
PSK	Phase Shift Keying / Modulación por Desplazamiento de Fase
QoS	Quality of Service / Calidad de Servicio
RF	Radio Frequency / Radiofrecuencia
RSSI	Received Signal Strength Indicator / Indicador de Intensidad de Señal Recibida
Rx	Receiver / Receptor
SDR	Software-Defined Radio / Radio Definida por Software
SF	Spreading Factor / Factor de Ensanchamiento
SNR	Signal-to-Noise Ratio / Relación Señal-Ruido
STO	Symbol Timing Offset / Desplazamiento Temporal de Símbolos
SVM	Support Vector Machine / Máquina de Vectores de Soporte
ToA	Time on Air / Tiempo en el Aire
Tx	Transmitter / Transmisor
UHD	USRP Hardware Driver / Controlador de Hardware USRP
UHF	Ultra High Frequency / Frecuencia Ultra Alta
USRP	Universal Software Radio Peripheral / Periférico de radio de software universal
WCRSN	Cognitive Wireless Sensor Networks / Redes de Sensores Inalámbricos Cognitivos
WSN	Wireless Sensor Network / Red de Sensores Inalámbricos

ÍNDICE

PRESENTACIÓN	II
DEDICATORIA	III
ABREVIATURAS	IV
ÍNDICE	VII
ÍNDICE DE TABLAS	XI
ÍNDICE DE FIGURAS	XII
RESUMEN	XV
ABSTRACT	XVI
1 Generalidades	1
1.1 Planteamiento del problema	1
1.2 Formulación del problema	3
1.2.1 Problema general	3
1.2.2 Problemas específicos	3
1.3 Objetivos	3
1.3.1 Objetivo General	3
1.3.2 Objetivos Específicos	4
1.4 Justificación	4
1.4.1 Justificación tecnológica	4

1.4.2	Justificación social	4
1.4.3	Justificación académica	5
1.5	Alcances	5
1.6	Limitaciones	5
2	Marco Teórico	7
2.1	Antecedentes - Estado del Arte	7
2.2	Bases Teóricas	11
2.2.1	Fundamentos de radio cognitivo	11
2.2.2	Espectro radioeléctrico	14
2.2.3	Radio definida por software (SDR)	16
2.2.4	Entorno GNU Radio y modelo por bloques	17
2.2.5	Tecnologías WSN sobre SDR	18
2.2.6	La Tecnología LoRa	19
2.2.7	Coexistencia de LoRa en Bandas ISM	22
2.2.8	Radio Cognitivo e internet de las cosas	23
2.2.9	Ventajas de la Tecnología de Radio Cognitivo en redes IoT	24
2.2.10	Radio Cognitivo aplicado a LoRa	25
2.2.11	Modelo del ciclo cognitivo aplicado a LoRa sobre SDR	26
2.2.12	Técnicas de detección de espectro	27
2.2.13	Detección de Energía (ED) y sus Desafíos	28
2.2.14	Detección de Tasa Constante de Falsas Alarmas CFAR	29
3	Diseño e implementación de un prototipo de radio cognitivo	31
3.1	Criterios de diseño	32
3.2	Selección e implementación en GNU Radio	33
3.2.1	Implementación de LoRa sobre SDR	34
3.3	Configuraciones para radio cognitivo	39
3.3.1	Cambios estructurales en <code>frame_sync_impl.cc</code>	40
3.3.2	Cambios estructurales en <code>frame_sync_impl.h</code>	42

3.3.3	Cambios estructurales en <code>modulate_impl.cc</code>	43
3.3.4	Cambios estructurales en <code>modulate_impl.h</code>	45
3.4	Descripción de equipamiento	47
3.4.1	Criterios	47
3.4.2	Inventario de equipos	47
3.5	Bloques GNU Radio	51
3.5.1	Configuración y variables globales	51
3.5.2	Transmisor LoRa SDR	51
3.5.3	Receptor LoRa SDR	52
3.5.4	Cálculo SNR	53
3.5.5	Cálculo RSSI	55
3.6	Diseño prototipo de radio cognitivo	58
3.6.1	Bloque ED_CFAR_Scanner	58
3.6.2	Bloque SliceStepper	61
3.6.3	Bloque ChannelPickerAndTune_SNR	62
3.6.4	Bloque SfDeciderMin	65
3.6.5	Bloque QtRangeSF (ajuste de SF)	68
3.6.6	Aplicación de la resintonización (control de UHD desde el <i>flowgraph</i>) .	68
3.6.7	Interacción en el lazo cognitivo	69
4	Pruebas y Resultados	73
4.1	Caracterización del Sistema LoRa Base	73
4.1.1	Rendimiento en Canal Limpio (Sin Interferencia)	73
4.1.2	Rendimiento bajo interferencia controlada	74
4.2	Evaluación del prototipo de radio cognitivo	78
4.2.1	Eventos de adaptación frecuencia-SF	81
4.2.2	Evidencia gráfica de los eventos cognitivos	83
4.3	Prueba de adaptación de robustez bajo canal simulado (AWGN)	87
	DISCUSIONES	93

CONCLUSIONES	97
RECOMENDACIONES	99
BIBLIOGRAFÍA	100
ANEXOS	107

ÍNDICE DE TABLAS

Tabla 1	Parámetros relevantes del enlace según factor de ensanchamiento (SF)	21
Tabla 2	Validación de escalabilidad en sistemas LoRa multiusuario sobre plataformas SDR	33
Tabla 3	SNR mínimos operativos por Spreading Factor	63
Tabla 4	Resumen de SNR, RSSI, y PER por SF en canal limpio.	74
Tabla 5	Resumen de SNR, RSSI y PER con interferencia por SF en el enlace LoRa no cognitivo.	76
Tabla 6	Eventos de adaptación de frecuencia y <i>factor de ensanchamiento</i> en el trans- ceptor LoRa cognitivo bajo interferencia CW.	81
Tabla 7	Resumen de parámetros medidos del sistema cognitivo LoRa bajo interferen- cia Channel Model	88

ÍNDICE DE FIGURAS

Figura 1	Arquitectura Básica del Prototipo de Radio Cognitivo	11
Figura 2	Diagrama de bloques simplificado de la arquitectura de una unidad de radio cognitivo.	13
Figura 3	Huecos en el espectro o espacios en blanco.	14
Figura 4	Espectro Radioeléctrico - Bandas de Frecuencia.	15
Figura 5	Parámetros Regionales LoRa Alliance - Perú	16
Figura 6	Aplicaciones de la Tecnología LoRa	20
Figura 7	Diagrama de implementación del prototipo de Radio Cognitivo.	31
Figura 8	Cadena Tx Lora	35
Figura 9	Cadena Rx Lora	38
Figura 10	USRP 2920	48
Figura 11	Antena VERT900 y atenuadores.	49
Figura 12	ERASynth Micro	49
Figura 13	Estación de trabajo (Linux)	50
Figura 14	Implementación de Hardware	50
Figura 15	Variables Globales	51
Figura 16	Transmisor LoRa	52
Figura 17	Receptor LoRa	53
Figura 18	Estimación SNR	53
Figura 19	Estimación RSSI	56
Figura 20	Sistema LoRa (baseline)	57
Figura 21	Bloque ED_CFAR_Scanner	61

Figura 22 Bloque Slice_Stepper	62
Figura 23 Bloque ChannelPickerAndTune_SNR	65
Figura 24 Bloque SfDeciderMin	67
Figura 25 Bloque QTRangeSF	68
Figura 26 Sistema LoRa con Radio Cognitivo	72
Figura 27 Flujo de pruebas	74
Figura 28 Flujo de pruebas con interferencia.	75
Figura 29 Generador ERASynth Micro configurado para producir interferencia CW co- canal a 915.6 MHz.	75
Figura 30 SNR en función de la potencia de interferencia para diferentes <i>factores de</i> <i>ensanchamiento</i> (SF7–SF12) en el enlace LoRa no cognitivo.	78
Figura 31 RSSI en función de la potencia de interferencia para diferentes <i>factores de</i> <i>ensanchamiento</i> (SF7–SF12) en el enlace LoRa no cognitivo.	79
Figura 32 Captura del enlace LoRa no cognitivo con SF7 en condición de saturación del canal por interferencia CW co-canal a -32 dBm.	79
Figura 33 Captura del enlace LoRa no cognitivo con SF12 en condición de saturación del canal por interferencia CW co-canal a -17 dBm.	80
Figura 34 Flujo de pruebas con radio cognitivo	80
Figura 35 Captura de pantalla del transceptor LoRa cognitivo durante un evento de cambio de canal.	83
Figura 36 Captura de pantalla del transceptor LoRa cognitivo durante un evento de cambio de <i>factor de ensanchamiento</i>	84
Figura 37 Evolución temporal del SNR del enlace LoRa cognitivo.	85
Figura 38 Evolución temporal del RSSI para el enlace LoRa cognitivo.	85
Figura 39 Evolución temporal del canal de frecuencia utilizado por el transceptor LoRa cognitivo.	86
Figura 40 Evolución temporal del <i>Factor de ensanchamiento</i> en el sistema LoRa cog- nitivo.	87
Figura 41 Evolución de SNR con Channel Model	89

Figura 42 Evolución de RSSI con Channel Model	89
Figura 43 Evolución de SF con Channel Model	90
Figura 44 Captura de pantalla del transceptor LoRa cognitivo durante un evento de aumento de SF con Channel Model	91
Figura 45 Captura de pantalla del transceptor LoRa cognitivo durante un evento de descenso de SF con Channel Model	91

RESUMEN

Las redes de sensores inalámbricos basadas en LoRa, ampliamente desplegadas sobre bandas no licenciadas, muestran una degradación de su tasa de error de paquetes (PER) ante interferencia co-canal sostenida. Esta tesis presenta el diseño, implementación y evaluación experimental de un prototipo de Radio Cognitivo sobre una plataforma de Radio Definida por Software (SDR) capaz de mitigar estos efectos. El sistema se construyó sobre el módulo gr-lora_sdr en GNU Radio, cuyo código fuente fue modificado para habilitar la reconfiguración del factor de ensanchamiento; sobre esta base se integraron cinco bloques cognitivos de diseño propio que implementan: detección de energía con umbral adaptativo de tasa de falsa alarma constante (CA-CFAR), barrido escalonado del espectro, decisión de cambio de canal basada en SNR, adaptación del factor de ensanchamiento, y control sincronizado de la sintonización del hardware. La validación experimental se realizó sobre dos equipos USRP-2920 con interferencia controlada generada mediante ERASynth Micro. Las pruebas experimentales con equipos USRP mostraron que bajo interferencia co-canal severa (-29 dBm), el sistema cognitivo mantuvo una Tasa de Error de Paquetes (PER) inferior al 10 %, mientras que el enlace convencional colapsó con un PER superior al 10 %. Los resultados validan que la adaptación dinámica en la capa física incrementa significativamente la robustez de los enlaces LoRa.

Palabras clave: Radio cognitivo (CR), GNU Radio, LoRa, Radio definida por software (SDR), Detección de espectro, Redes de sensores inalámbricos (WSN).

ABSTRACT

LoRa-based wireless sensor networks, widely deployed over unlicensed bands, exhibit a degradation of their packet error rate (PER) under sustained co-channel interference. This thesis presents the design, implementation, and experimental evaluation of a Cognitive Radio prototype on a Software-Defined Radio (SDR) platform capable of mitigating these effects. The system was built on the `gr-lora_sdr` module in GNU Radio, whose source code was modified to enable the reconfiguration of the spreading factor; on this basis, five cognitive blocks of own design were integrated, implementing: energy detection with an adaptive constant false alarm rate threshold (CA-CFAR), stepped spectrum scanning, SNR-based channel switching decision, spreading factor adaptation, and synchronized control of the hardware tuning. The experimental validation was carried out on two USRP-2920 devices with controlled interference generated by an ERASynth Micro. The experimental tests with USRP equipment showed that under severe co-channel interference (-29 dBm), the cognitive system maintained a Packet Error Rate (PER) below 10 %, while the conventional link collapsed with a PER above 10 %. The results validate that dynamic adaptation at the physical layer significantly increases the robustness of LoRa links.

Keywords: Cognitive Radio (CR), GNU Radio, LoRa, Software Defined Radio (SDR), Spectrum Sensing, Wireless Sensor Networks (WSN).

CAPÍTULO I

GENERALIDADES

1.1 Planteamiento del problema

En la última década, el crecimiento exponencial de dispositivos conectados a Internet ha impulsado la necesidad de redes de sensores inalámbricos (WSN) más eficientes y adaptables. El Internet de las Cosas (IoT) ha experimentado un auge sin precedentes, con proyecciones que indican que el número de dispositivos IoT conectados alcanzará los 39.6 mil millones para el 2033 (Statista, 2025). Las diversas tecnologías de comunicación inalámbrica para aplicaciones WSN han emergido como soluciones prometedoras para distintos escenarios de implementación. Sin embargo, estas implementaciones enfrentan desafíos significativos, particularmente en entornos con alta densidad de dispositivos y condiciones de radio variables (Augustin et al., 2016).

En el continente americano, y especialmente en Latinoamérica, la adopción de WSN se ha acelerado en diversos sectores. Un estudio realizado por la GSMA (2024) reveló que la región experimentará un crecimiento anual del 8 % en conexiones IoT entre 2023 y 2030, con proyecciones de alcanzar 125 millones de conexiones para 2030. Este crecimiento ha sido impulsado por aplicaciones en agricultura de precisión, monitoreo ambiental y gestión urbana.

En el contexto nacional de Perú, la adopción de tecnologías IoT y WSN ha mostrado un crecimiento significativo. El informe del Ministerio de Transportes y Comunicaciones (2022) menciona que el país ha experimentado un aumento del 25 % en la implementación de soluciones IoT en los últimos tres años. Sin embargo, la geografía diversa del país, que incluye regiones costeras, andinas y amazónicas, presenta desafíos únicos para la implementación de WSN. Además, la creciente urbanización y la concentración de actividades económicas en ciertas áreas han llevado a una mayor congestión del espectro radioeléctrico (OSIPTel, 2023).

Las redes WSN implementadas mediante plataformas SDR ofrecen ventajas significativas en términos de flexibilidad y adaptabilidad. Sin embargo, en entornos operativos específicos, se han identificado problemas críticos que afectan su rendimiento. La interferencia creciente

debido al incremento de uso del espectro, especialmente en zonas urbanas e industriales, ha llevado a una degradación de la calidad de señal. Un estudio realizado por Santiago J. (2017) en Lima, Perú, encontró que un porcentaje significativo de las transmisiones en ciertas bandas de frecuencia experimentaban altos niveles de interferencia y congestión, las cuales causaban caída de paquetes y una notable pérdida de señal.

Las causas principales de estos problemas son multifacéticas. En primer lugar, el uso ineficiente del espectro radioeléctrico se debe a la falta de adaptabilidad de los sistemas tradicionales de comunicación inalámbrica. La asignación estática del espectro en sistemas WSN limita la eficiencia y la capacidad de adaptación. Axell et al. (2012) y Yucek y Arslan (2009) coinciden en que la ausencia de mecanismos cognitivos reduce la posibilidad de ajustar parámetros frente a cambios en las condiciones del canal.

Estos factores han resultado en un rendimiento subóptimo de las redes WSN, manifestándose en problemas críticos. En entornos con variabilidad ambiental y densidad urbana, las redes de sensores convencionales enfrentan desafíos significativos. Por ejemplo, Domínguez (2019) diseñó una red WSN para monitoreo de temperatura y humedad en un campus universitario en Perú, mostrando cómo condiciones reales del entorno afectan la confiabilidad de la comunicación. Además, Zegarra (2021) encontró que la presencia de vegetación densa en selva baja peruana altera la propagación de señales y exige un diseño específico del espectro. En línea con estudios internacionales, como Axell et al. (2012) y Gurugopinath et al. (2020), estos hallazgos refuerzan la necesidad de mecanismos adaptativos en tiempo real para mantener el rendimiento y la eficiencia espectral en condiciones dinámicas.

Para abordar estos desafíos, se propone un prototipo de radio cognitivo para aplicaciones WSN mediante plataformas SDR. Este enfoque busca mejorar la eficiencia y adaptabilidad de las redes, proporcionando capacidades de detección espectral, toma de decisiones inteligente y adaptación dinámica de parámetros de comunicación. Particularmente, se prioriza el uso de métricas como la Relación Señal/Ruido SNR y el Indicador de Intensidad de Señal Recibida RSSI, fundamentales para evaluar la calidad del canal y guiar la selección óptima de frecuencias o esquemas de modulación en entornos de sensores inalámbricos. Según Lin y Hao (2021), el uso de estrategias cognitivas basadas en estas métricas puede mejorar la confiabilidad del enlace

y reducir la probabilidad de retransmisiones, optimizando así el uso del espectro y el consumo energético en aplicaciones IoT y WSN.

1.2 Formulación del problema

1.2.1 Problema general

¿En qué medida afecta al rendimiento de un sistema de red de sensores inalámbricos (WSN) la implementación de radio cognitivo basado en tecnología de radio definida por software (SDR)?

1.2.2 Problemas específicos

- ¿Qué conceptos de radio cognitivo son aplicables a las redes de sensores inalámbricos WSN?
- ¿Cómo se puede implementar un prototipo de radio cognitivo sobre tecnología de radio definida por software para redes WSN?
- ¿De qué manera se puede configurar y adecuar el prototipo de radio cognitivo para su funcionamiento en aplicaciones de redes de sensores inalámbricos WSN?
- ¿Cómo se puede evaluar el funcionamiento del prototipo de radio cognitivo en un entorno de redes de sensores inalámbricos?
- ¿De qué manera los resultados caracterizan el rendimiento del prototipo de radio cognitivo en entornos WSN?

1.3 Objetivos

1.3.1 Objetivo General

Evaluar el comportamiento del rendimiento de un prototipo de radio cognitivo basado en plataforma de radio definida por software (SDR) para aplicaciones de redes de sensores inalámbricos WSN.

1.3.2 *Objetivos Específicos*

1. Analizar los conceptos de radio cognitivo aplicables a redes de sensores inalámbricos WSN.
2. Diseñar un prototipo de radio cognitivo sobre tecnología de radio definida por software.
3. Configurar el prototipo de radio cognitivo para su funcionamiento en aplicaciones WSN.
4. Realizar pruebas de funcionamiento del prototipo en un entorno de redes de sensores inalámbricos WSN.
5. Analizar los resultados de las pruebas de funcionamiento para caracterizar el rendimiento del prototipo.

1.4 Justificación

1.4.1 *Justificación tecnológica*

El presente estudio propone una solución innovadora basada en tecnología de radio cognitivo, permitiendo un uso más eficiente del espectro radioeléctrico y mejorando la adaptabilidad de las redes en entornos con alta congestión y variabilidad espectral. La implementación de un sistema de radio cognitivo sobre una plataforma SDR optimizará aplicaciones en redes de sensores inalámbricos (WSN), estableciendo una base tecnológica avanzada para sistemas de comunicación más inteligentes y autónomos.

1.4.2 *Justificación social*

Este estudio contribuye al desarrollo de soluciones tecnológicas que optimizan el uso del espectro radioeléctrico para aplicaciones académicas y de investigación. Los resultados servirán como antecedente para trabajos futuros de redes inalámbricas de sensores en la UNSAAC, fomentando la colaboración interdisciplinaria entre estudiantes y docentes interesados en el desarrollo de tecnologías de comunicación inalámbrica avanzadas. La investigación en el campo

de radio cognitivo puede contribuir al desarrollo de tecnologías de comunicación más robustas y confiables, capaces de operar de manera efectiva en entornos dinámicos y cambiantes.

1.4.3 Justificación académica

El estudio contribuye al campo de la ingeniería electrónica y las telecomunicaciones mediante la ampliación del conocimiento en diseño e implementación de sistemas adaptativos basados en SDR para redes de sensores inalámbricos. También fortalece la investigación en radio cognitivo aplicada a la gestión eficiente del espectro, proporcionando un recurso académico valioso para futuros trabajos en entornos de alta interferencia.

1.5 Alcances

1. El estudio abarcará el diseño, implementación y evaluación de un prototipo funcional de radio cognitivo, en el entorno controlado del LIISTTI de la UNSAAC.
2. Se desarrollará un prototipo funcional con capacidades de radio cognitivo mediante plataformas SDR, con pruebas en condiciones simuladas de congestión y variabilidad espectral dentro del laboratorio.
3. Se evaluará el rendimiento del sistema mediante dos parámetros específicos SNR (Signal-to-Noise Ratio) y RSSI (Received Signal Strength Indicator) en el entorno controlado del laboratorio.

1.6 Limitaciones

1. El estudio se limitará al entorno controlado del laboratorio en la UNSAAC, enfocándose exclusivamente en pruebas bajo condiciones específicas, lo que limitará la generalización directa de los resultados.
2. Las simulaciones y pruebas se realizarán utilizando únicamente el equipamiento disponible en el Laboratorio de Investigación e Innovación en Sistemas de Telecomunicaciones

y Tecnologías de la Información de la UNSAAC, el cual incluye equipos USRP 2920, computadora de alta gama, antenas de 900 MHz y otros accesorios.

3. El nivel de pruebas del prototipo se restringe a la simulación de una comunicación entre un nodo y un gateway, para la realización de pruebas de concepto y la demostración de las funcionalidades de radio cognitivo.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes - Estado del Arte

Busacca et al. (2024) se plantearon como objetivo desarrollar una implementación de un transceptor LoRa basado en radio definida por software (SDR) para optimizar la capa física del protocolo LoRa y explorar aplicaciones avanzadas. La metodología empleada incluyó el diseño de un transceptor SDR completamente funcional y su validación a través de experimentos con dispositivos comerciales y un emulador de canal inalámbrico. Los resultados demostraron que SDR-LoRa ofrece un rendimiento comparable a los sistemas LoRa comerciales, con aplicaciones exitosas en localización precisa, cancelación de interferencias y mejora de la fiabilidad del enlace. Se concluyó que SDR-LoRa puede potenciar la investigación en redes de largo alcance y baja potencia, facilitando innovaciones en escenarios de IoT.

Cuña, Durán y Olivera (2014) se plantearon como objetivo estudiar e implementar algoritmos de manejo del espectro en redes cognitivas utilizando software definido por radio (SDR) en plataformas USRP. La metodología consistió en una revisión bibliográfica exhaustiva y la selección de algoritmos como Jump Stay, MCA, MMCA, DRSEQ y CRSEQ, los cuales fueron evaluados a través de simulaciones en MATLAB y pruebas experimentales en hardware. Los resultados mostraron que los algoritmos propuestos lograron tiempos de Rendezvous óptimos bajo condiciones específicas, como sincronización y modelos simétricos y asimétricos. En conclusión, el estudio demostró la viabilidad de implementar algoritmos de encuentro en SDR para mejorar la eficiencia espectral, proporcionando una plataforma modular reutilizable para futuros trabajos en redes cognitivas.

Buitrago Rodil (2018) se planteó como objetivo implementar un esquema de radio cognitivo para redes LoRaWAN con el propósito de reducir la tasa de pérdida de paquetes y optimizar el consumo energético en dispositivos IoT. La metodología empleada consistió en el desarrollo

de un sistema integrado por una SDR, nodos LoRaWAN, un gateway LoRaWAN y un servidor CORE para la toma de decisiones cognitivas. Se implementaron técnicas de sensado del espectro mediante detección de energía y algoritmos de cambio de canal basados en la ocupación espectral. Los resultados mostraron que la inclusión de capacidades cognitivas permitió una reducción del 40 % en la tasa de pérdida de paquetes y una mejora del 25 % en la eficiencia energética de los dispositivos. En conclusión, el estudio demostró que las redes LoRaWAN cognitivas son una solución viable para mejorar la coexistencia espectral y la eficiencia en aplicaciones de IoT.

Salika et al. (2022) se plantearon como objetivo desarrollar un protocolo denominado LoRaCog, que implementa la técnica de radio cognitivo en redes LoRa para mejorar la eficiencia espectral y reducir la congestión en el espectro no licenciado. La metodología consistió en modificar el protocolo LoRaWAN para incluir funciones de sensado del espectro a través de gateways y asignar canales no utilizados a dispositivos finales mediante un servidor de red. Se realizaron simulaciones utilizando el modelo Okumura-Hata para evaluar el desempeño del protocolo mediante la reducción de paquetes rechazados. Los resultados evidenciaron una significativa disminución en la tasa de paquetes rechazados, pasando del 50 % con LoRaWAN al 24 % con LoRaCog cuando se utilizó un canal cognitivo adicional. Asimismo, se evidenció una disminución en la cantidad de gateways necesarios para mantener la cobertura de red, lo que implica una reducción de costos de infraestructura. En conclusión, el protocolo LoRaCog mejora la eficiencia espectral y la calidad del servicio en redes IoT, proporcionando una solución complementaria a LoRaWAN para mitigar la congestión del espectro.

Pérez y Risco (2020) se plantearon como objetivo analizar el funcionamiento de LoRa y LoRaWAN, proponiendo su implementación conjunta con tecnologías emergentes como blockchain para aplicaciones en la agroindustria peruana. La metodología se basó en el análisis y revisión de las características técnicas de LoRa y LoRaWAN, destacando su potencial en términos de bajo consumo energético y amplia cobertura. Los resultados mostraron que la adopción de estas tecnologías puede mejorar la eficiencia de operación, reducir costos y aumentar la

productividad en el sector agroindustrial. Por ejemplo, al integrar sensores para monitorear variables clave como humedad y temperatura en tiempo real, se observó un aumento del 30 % en la eficiencia de los procesos agrícolas. En conclusión, el estudio resalta la importancia de LoRa y LoRaWAN como herramientas clave para el desarrollo de la industria 4.0 en sectores rurales.

Gurugopinath et al. (2020) se plantearon como objetivo realizar un análisis integral sobre los avances recientes en sistemas de IoT basados en radio cognitivo (CR-IoT), explorando su integración con tecnologías emergentes como blockchain y aprendizaje automático. La metodología consistió en un estudio exhaustivo de la literatura, detallando los desafíos técnicos, de seguridad y normativos, y proponiendo arquitecturas de CR-IoT para optimizar la gestión del espectro y la eficiencia de la red. Los resultados indicaron que la implementación de CR-IoT puede reducir la interferencia espectral en un 45 % y mejorar la calidad del servicio en un 35 %. Se concluyó que el CR-IoT es una tecnología prometedora para manejar el creciente número de dispositivos IoT, proporcionando un uso eficiente del espectro y mejorando la fiabilidad de las comunicaciones.

Goswami y Goswami (2020) se plantearon como objetivo diseñar e implementar una antena compacta con ranura fractal para redes de sensores inalámbricos cognitivas (WCRSN), compatible con diversas aplicaciones como LoRa, Zigbee y Bluetooth. La metodología consistió en diseñar una antena de ranura fractal truncada con una estructura de banda de exclusión electromagnética EBG sobre un sustrato FR4 de $12 \times 18 \times 1.6$ mm. El diseño se validó mediante simulaciones y mediciones experimentales para evaluar parámetros como el ancho de banda de impedancia, la ganancia y la eficiencia de radiación. Los resultados mostraron que la antena propuesta cubría un amplio rango de frecuencias de 0.79 a 12 GHz, con una ganancia promedio de 2.59 dBi y una eficiencia de radiación del 79.9 %. En conclusión, la antena diseñada demostró ser altamente efectiva para el escaneo de espectro amplio en entornos de redes cognitivas, mejorando la detección de oportunidades de canal en un 45 %.

Akram et al. (2021) se plantearon como objetivo diseñar y evaluar una arquitectura basada en tecnología LoRa para la administración de residuos sólidos, con énfasis en la eficiencia energética y la estabilidad de la comunicación. La metodología incluyó el diseño de un nodo sensor personalizado y un gateway, utilizando la tecnología LoRa y el simulador FLoRa para analizar parámetros clave como el consumo de energía, el tiempo en aire ToA, la sensibilidad de LoRa y el presupuesto de enlace. Los resultados demostraron que la arquitectura propuesta redujo el consumo energético de los nodos en un 30 % y aumentó la vida útil de la batería en un 40 % en comparación con soluciones convencionales. En conclusión, la integración de nodos personalizados y gateways basados en LoRa proporciona una solución eficiente para la transmisión de datos en tiempo real a servidores en la nube, optimizando la gestión de residuos en entornos urbanos.

Onumanyi et al. (2020) se plantearon como objetivo explorar el potencial de integrar radio cognitivo en redes de área amplia de baja potencia LPWAN para aplicaciones sobre Internet de las cosas (IoT). La metodología consistió en una revisión sistemática de tecnologías LPWAN y su compatibilidad con radio cognitivo, enfocándose en la capa física y en modelos arquitectónicos. Los resultados destacaron que la incorporación de radio cognitivo en LPWAN puede reducir la interferencia espectral en un 45 %, aumentar la eficiencia espectral en un 30 % y prolongar la vida útil de los dispositivos IoT en un 20 % al optimizar el uso del espectro. En conclusión, el estudio demostró que la integración de radio cognitivo en LPWAN no solo mejora la calidad del servicio, sino que también aumenta la capacidad de manejo de datos en redes IoT, estableciendo bases para innovaciones futuras.

García-Requejo et al. (2023) se plantearon como objetivo implementar un sistema de monitoreo fundado en redes de sensores inalámbricos (WSN) utilizando tecnología LoRaWAN para personas con deterioro cognitivo leve. La metodología incluyó el desarrollo de un dispositivo portátil que combina tecnologías GNSS para localización en exteriores, un sistema de localización en interiores basado en infrarrojos, y un acelerómetro para la detección de pasos. El sistema transmite datos mediante LoRaWAN a un servidor, donde se almacenan y visualizan en

una aplicación web, permitiendo el monitoreo de la actividad diaria y la posición del usuario. Los resultados demostraron que el sistema alcanzó una precisión del 92.8 % en la detección de pasos y más del 90 % en la localización de habitaciones en interiores, logrando una cobertura de hasta 12 km en áreas rurales. En conclusión, este sistema mejora la calidad de vida de los pacientes al promover una vida independiente, proporcionando tranquilidad a los cuidadores mediante alertas en tiempo real.

2.2 Bases Teóricas

2.2.1 Fundamentos de radio cognitivo

Radio cognitivo emergió como una respuesta innovadora a los desafíos planteados por la creciente demanda de espectro radioeléctrico y la escasez de una gestión más eficiente de este recurso limitado. Mitola y Maguire (1999) introdujeron por primera vez el término “radio cognitivo” para describir un dispositivo de comunicación inalámbrica capaz de adecuar sus características de transmisión y recepción de manera inteligente, basándose en la percepción de su entorno y en las necesidades del usuario. Haykin (2005) lo definió como una red de comunicaciones inalámbricas inteligentes la cual reconoce la condición de su entorno del cual aprende y se adapta para mejorar la eficiencia espectral.

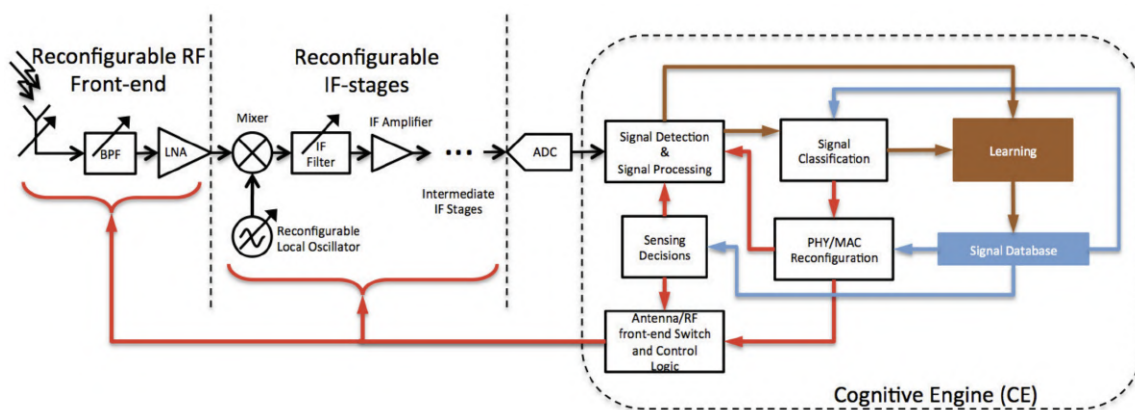


Figura 1
Arquitectura Básica del Prototipo de Radio Cognitivo

Nota. Tomado de “Learning-Aided Sub-Band Selection Algorithms for Spectrum Sensing in Wide-Band Cognitive Radios” (Fig. 1), por Li et al., 2014.

Radio cognitivo surge como respuesta a los modelos de asignación estática del espectro, los cuales han sido ineficientes. FCC (2002) reconoció que gran parte del espectro licenciado estaba infrautilizado y promovió el acceso dinámico como solución. Mediante radio cognitivo, los usuarios secundarios pueden usar estos canales “huecos” cuando los usuarios primarios no los utilizan, garantizando que los usuarios primarios permanezcan libres de interferencia. Este proceso se logra mediante un bucle de operación continua donde los nodos cognitivos:

1. Detectan o perciben el espectro: miden la ocupación del canal en tiempo real (spectrum sensing). Es la tarea de monitorizar el entorno de radiofrecuencia para identificar la presencia de otras transmisiones (interferencias o usuarios primarios licenciados) y localizar agujeros de espectro.
2. Deciden la acción apropiada: analizan la disponibilidad y seleccionan el mejor canal libre. Esta fase implica analizar los datos del sensado para seleccionar el canal de comunicación y los parámetros de transmisión óptimos que cumplan con los objetivos de calidad de servicio (QoS).
3. Adaptan sus parámetros: Se refiere a la capacidad del sistema para reconfigurar sus parámetros de radio frecuencia RF y hardware y ejecutar el cambio al nuevo estado operativo decidido. Este proceso debe ser lo más rápido y transparente posible para el usuario.
4. Distribución de Espectro: Esta es la estrategia o política de alto nivel que gobierna cómo radio cognitivo coexiste con otros usuarios (tanto primarios como secundarios), ya sea mediante evasión de interferencia o acceso oportunista coordinado.

En el diseño de redes cognitivas, Akyildiz et al. (2006) propusieron una arquitectura genérica de redes de nueva generación que integra estas funciones, incluyendo la gestión de espectro, la movilidad de espectro (cambio de canal durante la transmisión) y la distribución de espectro. Estos autores identifican los principales retos de las redes de radio cognitivo, como la necesidad de métodos avanzados en capas físicas y el diseño cross-layer para mantener el rendimiento de las capas superiores ante los cambios dinámicos de canal.

La figura 2 presenta un diagrama de bloques simplificado de la arquitectura de una unidad de radio cognitivo.

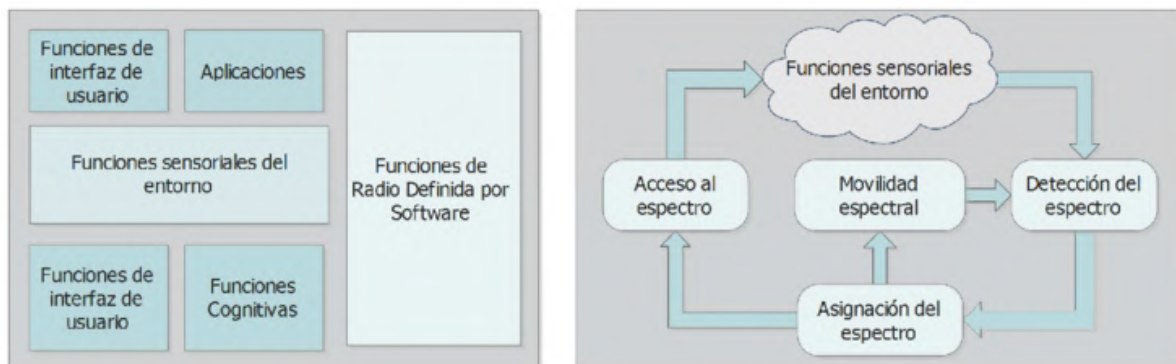


Figura 2
Diagrama de bloques simplificado de la arquitectura de una unidad de radio cognitivo.

Nota. El diagrama muestra los componentes principales de una unidad de radio cognitivo, incluyendo el transceptor RF, la unidad de procesamiento de banda base, el motor cognitivo, el módulo de detección de espectro, la base de conocimiento y la interfaz de políticas. Las flechas indican el flujo de información entre los componentes.

En aplicaciones de redes de sensores inalámbricos (WSN), la toma de decisiones cognitivas debe fundamentarse en métricas de capa física que puedan ser obtenidas de manera directa, confiable y con bajo costo computacional. Entre estas métricas, la Relación Señal–Ruido (SNR) y el Indicador de Intensidad de Señal Recibida (RSSI) destacan como parámetros clave para caracterizar la calidad instantánea del canal de comunicación. La SNR permite cuantificar la degradación del enlace producto del ruido térmico y la interferencia, mientras que el RSSI proporciona una estimación del nivel de potencia recibido, útil para identificar condiciones de atenuación o congestión espectral.

Diversos estudios han demostrado que existe una relación directa entre la SNR y métricas de desempeño como la tasa de error de bits (BER) y la tasa de error de paquetes (PER), especialmente en sistemas de modulación robusta como LoRa. En este contexto, la utilización de SNR y RSSI como variables de entrada para el motor cognitivo permite implementar mecanismos de adaptación dinámica que optimizan el rendimiento del enlace sin requerir información de capas superiores, lo cual resulta adecuado para dispositivos WSN con restricciones energéticas y de procesamiento (Haykin, 2005; Akyildiz et al., 2006; Axell et al., 2012).

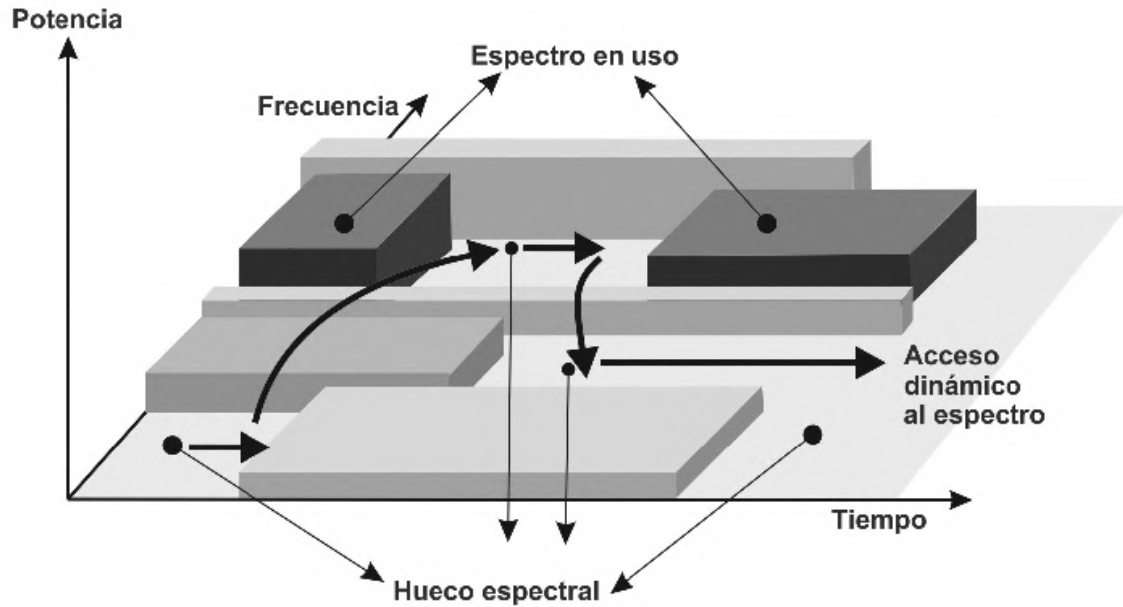


Figura 3
Huecos en el espectro o espacios en blanco.

Nota. Tomado de “NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey” (Fig. 3), por Akyildiz et al. 2006. La figura representa la idea de que el espectro no siempre está totalmente ocupado, y que existen oportunidades (huecos) para usarlo de manera más eficiente.

2.2.2 Espectro radioeléctrico

El espectro radioeléctrico se define como la porción del espectro electromagnético que abarca las frecuencias utilizadas para las comunicaciones inalámbricas, típicamente desde 3 kHz hasta 300 GHz. Cada banda presenta características de propagación distintas, lo que las hace más o menos adecuadas para aplicaciones específicas.

Para fines de regulación y gestión, el espectro se divide principalmente en dos categorías:

- **Bandas Licenciadas:** Son porciones del espectro asignadas o vendidas a operadores específicos (como compañías de telefonía móvil o radiodifusoras) para su uso exclusivo. Esto garantiza una comunicación libre de interferencias, pero tiene un alto costo y es rígidamente regulado.
- **Bandas No Licenciadas:** Son bandas de frecuencia reservadas internacionalmente para

aplicaciones Industriales, Científicas y Médicas (ISM). Estas bandas son de uso libre y gratuito, lo que ha fomentado una explosión de innovación en tecnologías de corto y largo alcance.

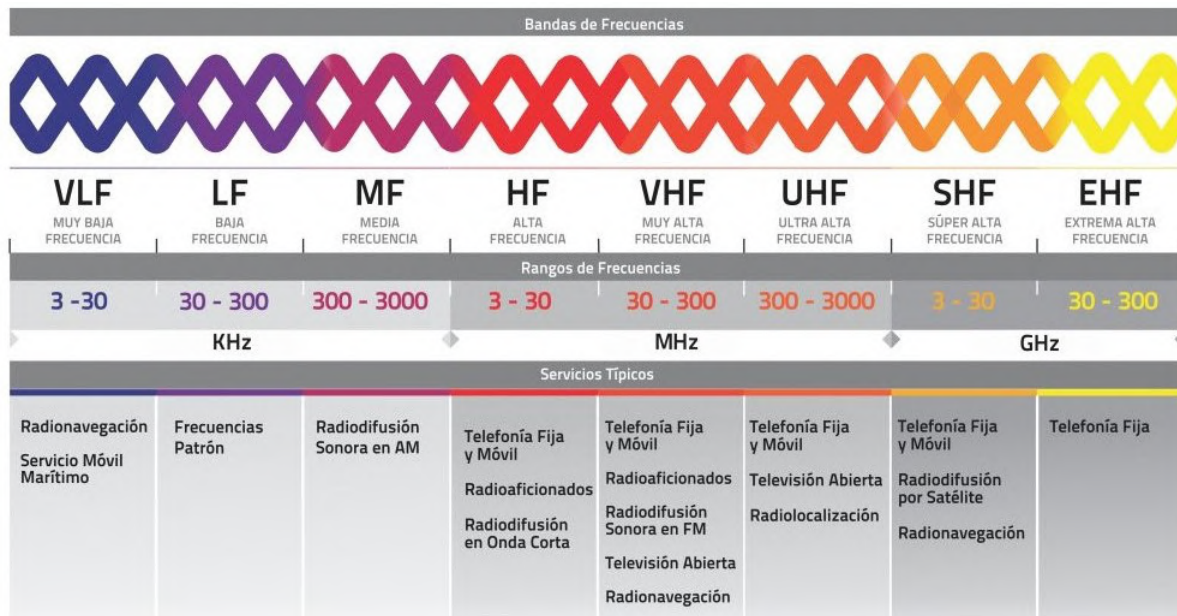


Figura 4
Espectro Radioeléctrico - Bandas de Frecuencia.

La gran mayoría de las tecnologías WSN, incluyendo LoRa, Wi-Fi y Bluetooth, operan en estas bandas ISM para evitar los costos y la complejidad de las licencias. Si bien esto reduce las barreras de entrada, crea el problema central de esta tesis: la congestión del espectro. Al ser un recurso compartido y no coordinado, todos los dispositivos deben competir por el mismo ancho de banda, generando altos niveles de interferencia.

Aunque existen muchas bandas ISM, las más relevantes para aplicaciones WSN y, específicamente, para la tecnología LoRa, se encuentran predominantemente en la banda de Ultra Alta Frecuencia (UHF). Esta banda (300 MHz - 3 GHz) ofrece un equilibrio ideal entre penetración en edificios, alcance físico y tamaño de antena.

Las bandas ISM clave en UHF para LoRa son:

- Banda de 433 MHz: Usada en la Región 1 (Europa), se caracteriza por una excelente propagación y penetración.

- Banda de 868 MHz: Es la banda principal para LoRaWAN en Europa (Región 1), ofreciendo un buen equilibrio entre alcance y tasa de datos.
- Banda de 915 MHz: Es la banda principal para LoRaWAN en las Américas (Región 2). Esta es la banda de operación seleccionada para el prototipo y las pruebas en esta tesis.
- Banda de 2.4 GHz (Global): Aunque no es la banda principal de LoRa para largo alcance, es una banda ISM crucial donde coexisten Wi-Fi, Bluetooth y otras WSN (como Zigbee). Sirve como una fuente principal de interferencia que los sistemas de radio cognitivos deben aprender a gestionar.



RP002-1.0.2 LoRaWAN® Regional Parameters

ISO 3166-1 Country name (Code alpha-2)	Band / channels	Channel Plan	LoRaWAN® Certified devices with Regulatory Type Approval
Peru (PE)	915 - 928 MHz	AU915-928 ³	

Figura 5
Parámetros Regionales LoRa Alliance - Perú

Nota. De LoRaWAN® Regional Parameters RP002-1.0.2, por LoRa Alliance, 2021.

El trabajo de esta tesis se centra, por tanto, en el desafío de la congestión dentro de la banda ISM de 915 MHz. El objetivo del prototipo de radio cognitivo es analizar, detectar y adaptarse inteligentemente a la interferencia generada por otros usuarios en esta banda compartida y no licenciada.

2.2.3 Radio definida por software (SDR)

La radio definida por software (SDR, *Software Defined Radio*) es un paradigma de diseño en el que la mayor parte de las funciones de la cadena de comunicaciones (modulación, demodulación, filtrado, sincronización, estimación de canal) se implementan en software sobre plataformas reconfigurables, en lugar de estar fijadas en hardware específico (Mitola, 2000).

De forma simplificada, un sistema SDR está compuesto por tres bloques principales: *front-end* de RF, conversores A/D y D/A, y procesamiento digital. El *front-end* se encarga de la conversión de frecuencia y la amplificación de la señal para situarla en una banda intermedia adecuada para los conversores; estos últimos fijan el ancho de banda observable y el rango dinámico del sistema. Finalmente, el procesamiento digital se realiza en un computador, FPGA o procesador embebido, donde se implementan en software los algoritmos de comunicación y control del enlace (Haykin, 2005). Esta separación entre parte analógica mínima y parte digital flexible hace que la SDR sea especialmente adecuada para investigación y creación de prototipos, como en la presente tesis.

2.2.4 Entorno GNU Radio y modelo por bloques

GNU Radio es un entorno de desarrollo de código abierto orientado a la implementación de sistemas SDR mediante la interconexión de bloques de procesamiento de señal (Blossom, 2004). Cada bloque realiza una función bien definida (por ejemplo, fuente de datos, filtro, modulador, demodulador, estimador de sincronía) y se conecta con otros a través de puertos de entrada y salida, formando un *flowgraph* o cadena de procesamiento.

En GNU Radio coexisten puertos de flujo, que transportan muestras continuas (por ejemplo, muestras IQ), y puertos de mensajes, que transportan unidades discretas de información o comandos de control. El planificador del entorno se encarga de activar los bloques según la disponibilidad de datos. Esta arquitectura resulta especialmente útil para radio cognitivo: métricas como SNR, RSSI o PER pueden enviarse desde los bloques receptores hacia bloques de decisión, que a su vez modifican en tiempo real parámetros de los bloques transmisores (frecuencia, factor de ensanchamiento, potencia), cerrando el ciclo sensar, decidir y actuar (Haykin, 2005).

En este contexto se sitúa el módulo `gr-lora_sdr`, que implementa la capa física LoRa sobre GNU Radio mediante bloques especializados (codificador Hamming, entrelazador, modulador CSS, sincronizador de trama, demodulador por FFT, etc). La tesis aprovecha esta estructura modular y la extiende con bloques cognitivos adicionales encargados del sensado del canal y de la adaptación de parámetros.

2.2.5 Tecnologías WSN sobre SDR

LoRa (gr-lora_sdr) ofrece un PHY de espectro ensanchado por chirps (CSS) con parámetros reconfigurables (frecuencia, ancho de banda, factor de ensanchamiento y tasa de codificación). Su principal ventaja es la robustez a SNR bajos (típico de WSN de gran alcance); como contrapartida, la tasa es reducida y crece el retardo cuando se eligen SF altos. En GNU Radio la integración es directa y expone métricas (SF, CR, BW, sync word, header/CRC) que facilitan evaluación objetiva.

IEEE 802.15.4 O-QPSK 2.4 GHz (gr-ieee802-15-4) es el punto de referencia para WPAN: capa física PHY a 250 kbps con 16 canales y DSSS, fácil de instrumentar y de enlazar con capas superiores (p. ej., ZigBee/Thread en análisis de trazas). Permite estudiar coexistencia realista en 2.4 GHz (Wi-Fi/BLE). Sus desventajas principales son el menor alcance respecto de sub-GHz y la sensibilidad a congestión en bandas ISM. En CR, el channel hopping y la selección por energía detectada (ED/LQI) son estrategias inmediatas y reproducibles.

DASH7/D7AP (FSK en sub-GHz) se posiciona como alternativa LPWAN ligera con perfiles de 25–200 kHz y opciones de FEC. FSK ofrece cadenas de demodulación simples, buena sensibilidad y tiempos de adquisición cortos; además, el hopping en sub-GHz y la conmutación de datarate/FEC son palancas naturales para políticas de CR. Requiere algo más de trabajo de integración que LoRa/802.15.4, pero rinde bien en escenarios de interferencia angosta (tonos continuos o señales 2-FSK cercanas). Es una “tercera vía” útil para validar que las decisiones del motor cognitivo generalizan a otro PHY.

BLE (PHY 1M/2M) es valioso aunque no sea la WSN de mayor alcance. En GNU Radio suele emplearse para recepción/sniffing y estimación de ocupación de 2.4 GHz gracias a su frequency hopping rápido; con ello se pueden construir mapas de actividad y disparar decisiones de CR (evitar canales ocupados, medir impacto cruzado con 802.15.4). Su limitación radica en que un stack completo TX/MAC es más complejo; sin embargo, como “sensor del espectro” y fuente de interferencia controlada es inmediato y práctico en laboratorio.

IEEE 802.15.4 LECIM (sub-GHz) amplía 802.15.4 hacia mayores alcances y menor consumo usando canales más angostos. Desde la óptica cognitiva, esa granularidad de canalización

habilita decisiones más finas: seleccionar sub-canales con mejor SNR, ajustar datarate y potencia con pasos pequeños y medir el tiempo de resintonización efectivo del SDR. El costo es una menor madurez pública de bloques y más trabajo de integración, suele plantearse como línea de investigación complementaria.

Para esta tesis se eligió `gr-lora_sdr` como base del prototipo de radio cognitivo porque está alineado con nuestro entorno: es un módulo OOT nativo para GNU Radio 3.10 (con bloques GRC) y UHD; expone Tx/Rx completos y parámetros clave (SF, CR, BW, sync word, header/CRC) desde el flowgraph. Robusto a SNR bajos y con cadena PHY completa (sincronización con corrección de STO/CFO, FEC, *whitening*/interleaver, CRC), útil cuando introducimos interferencia y forzamos la resintonización. Código abierto, mantenido y documentado (paper GRCon 2024 y guía EPFL), lo que nos permitió modificar/automatizar parámetros. Los parámetros frecuencia, ancho de banda (BW), factor de ensanchamiento (SF) y tasa de codificación (CR) pueden ajustarse dinámicamente desde bloques/mensajería de GNU Radio, materializando políticas de CR (resintonización de canal, adaptación de SF ante interferencia). Se eligió `gr-lora_sdr` porque minimiza fricción de integración (GRC+UHD), cubre todo el PHY con desempeño a bajo SNR, es compatible con USRP y radios LoRa comerciales para pruebas de campo y/o laboratorio, permite simulación sin hardware y es fácilmente automatizable para decisiones cognitivas.

2.2.6 La Tecnología LoRa

LoRa es una tecnología de comunicación inalámbrica de largo alcance y muy bajo consumo energético. La tecnología LoRa (Long Range) es una técnica de modulación de capa física (PHY) patentada por Semtech, diseñada específicamente para aplicaciones de Redes de Área Amplia y Baja Potencia (LPWAN). Su popularidad en el Internet de las Cosas (IoT) se debe a su capacidad para lograr una comunicación robusta a larga distancia con un consumo de energía extremadamente bajo, permitiendo que los dispositivos de sensores funcionen con baterías durante años.

Fundamentos de la Tecnología LoRa

El núcleo de LoRa es su técnica de modulación: Chirp Spread Spectrum (CSS). A diferencia de otras técnicas de modulación que usan desplazamiento de frecuencia FSK o de fase PSK, CSS codifica la información en “chirps”. Un chirp es una señal de banda ancha cuya frecuencia aumenta (up-chirp) o disminuye (down-chirp) linealmente durante un período de tiempo.

La principal ventaja de CSS es su extrema robustez. Debido a que la energía de la señal se “esparce” por todo el ancho de banda del canal, es altamente resistente a la interferencia de banda estrecha, al desvanecimiento por multitrayectoria y al efecto Doppler. Esto permite que el receptor decodifique señales que están significativamente por debajo del piso de ruido (hasta -20 dB), una hazaña imposible para las modulaciones tradicionales.



Figura 6
Aplicaciones de la Tecnología LoRa

Parámetros Clave de la Capa Física

La robustez y la tasa de datos de LoRa no son fijas; se configuran mediante un conjunto de parámetros ortogonales. La capacidad de un sistema de radio cognitivo para manipular estos parámetros es fundamental para esta tesis.

Factor de ensanchamiento SF: Este es el parámetro más crítico. Define cuántos chirps se utilizan para codificar un solo símbolo de información, variando desde SF 7 (el más rápido) hasta SF 12 (el más robusto). Existe un compromiso directo:

- **Un SF alto** (ej. SF 12): Utiliza más chirps por símbolo. Esto aumenta el “Tiempo en el Aire” (ToA) y, por lo tanto, reduce la tasa de datos, pero incrementa drásticamente la sensibilidad del receptor y el alcance (más robusto).
- **Un SF bajo** (ej. SF 7): Utiliza menos chirps por símbolo. Tiene un ToA corto y una tasa de datos más alta, pero es menos sensible al ruido y tiene menor alcance. Las transmisiones en diferentes SF son cuasi-ortogonales, lo que permite que múltiples señales coexistan en el mismo canal simultáneamente.

Esta relación se resume en la siguiente tabla:

Tabla 1

Parámetros relevantes del enlace según factor de ensanchamiento (SF)

SF	Tiempo de aire (ms)	Tasa de datos (kbps)	SNR mínimo (dB)
7	56	5.469	-7.5
8	103	3.125	-10.0
9	185	1.758	-12.5
10	370	0.977	-15.0
11	741	0.537	-17.5
12	1483	0.293	-20.0

Nota. Adaptado de SX1276/77/78/79 Datasheet (Rev. 7, Tabla de parámetros LoRa, p. 16), por Semtech Corporation, 2020.

Ancho de banda BW: Define el ancho del canal utilizado para la transmisión del chirp (ej. 125 kHz, 250 kHz, 500 kHz). Un ancho de banda mayor permite tasas de datos más altas, pero reduce la sensibilidad del receptor.

Tasa de codificación CR: Define el nivel de corrección de errores hacia adelante FEC aplicado a la carga útil. Un CR más alto (ej. 4/8) agrega más bits de redundancia, mejorando la robustez contra ráfagas de interferencia a costa de una menor tasa de datos efectiva.

Distinción entre LoRa y LoRaWAN

Es académicamente crucial distinguir entre LoRa y LoRaWAN:

- LoRa: Es exclusivamente la capa física PHY, como se describió anteriormente. Es la modulación CSS que permite la comunicación de largo alcance.
- LoRaWAN: Es el protocolo de red de capa de Control de Acceso al Medio (MAC). LoRaWAN define la arquitectura del sistema (una topología de estrella de estrellas), las clases de dispositivos (A, B, C), la seguridad (cifrado AES) y la gestión de la red para que los dispositivos se comuniquen con un servidor de red.

Esta tesis se centra en el análisis de la capa física LoRa, implementada sobre SDR mediante el módulo `gr-lora_sdr`. El objetivo es dotar a esta capa física de capacidades cognitivas para detectar y evadir la interferencia, un problema que la capa MAC de LoRaWAN por sí sola no gestiona dinámicamente.

2.2.7 Coexistencia de LoRa en Bandas ISM

Como se estableció en la sección anterior, las tecnologías WSN como LoRa operan predominantemente en las bandas de frecuencia no licenciadas o libres (ISM) (FCC, 2021). Si bien esto ha permitido un crecimiento exponencial de dispositivos (Statista, 2024), también ha creado el desafío central de esta tesis: la saturación del espectro.

Este crecimiento ha llevado a una mayor congestión del espectro radioeléctrico y a problemas significativos en entornos con alta densidad de dispositivos. Esta saturación resulta en niveles crecientes de interferencia, lo cual degrada la calidad de la señal y la fiabilidad de las redes LoRaWAN (Salika et al., 2022).

Aunque la modulación Chirp Spread Spectrum (CSS) utilizada por LoRa proporciona una elevada robustez frente a interferencias de banda estrecha y condiciones de bajo SNR, esta robustez no implica inmunidad total a la interferencia. En escenarios de coexistencia espectral, particularmente bajo interferencia co-canal continua, la degradación del SNR efectivo puede provocar incrementos significativos en la tasa de error de paquetes, especialmente cuando se utilizan factores de ensanchamiento bajos.

Estudios experimentales han evidenciado que la interferencia co-canal tiene un impacto más severo que la interferencia adyacente en enlaces LoRa, ya que afecta directamente la

correlación de los chirps en el receptor. En este contexto, la adaptación dinámica de parámetros de la capa física, como la frecuencia de operación y el factor de ensanchamiento, emerge como una estrategia efectiva para mitigar los efectos de la interferencia y mejorar la coexistencia en bandas ISM congestionadas (Bor et al., 2016; Croce et al., 2018; Georgiou y Raza, 2017).

Estudios han demostrado que en bandas de frecuencia con alta congestión se experimentan “altos niveles de interferencia y congestión, las cuales causaban caída de paquetes y una notable pérdida de señal” (Santiago J., 2017). El uso ineficiente del espectro, debido a la falta de adaptabilidad de los sistemas tradicionales, es la causa principal de estos problemas.

Los mecanismos de mitigación estandarizados, como el duty cycle (ciclo de trabajo) en LoRaWAN, si bien aseguran un acceso justo al medio, no resuelven activamente las colisiones dinámicas ni se adaptan a un entorno de interferencia que cambia rápidamente (LoRa Alliance, 2019).

Ante este escenario de espectro congestionado, radio cognitivo emerge como el paradigma de solución teórica. Un sistema de CR se define como un sistema de radio “inteligente” que posee la capacidad de detectar o sentir (sense) su entorno operativo para reconfigurarse dinámicamente y evitar la interferencia (Haykin, 2005).

2.2.8 Radio Cognitivo e internet de las cosas

Radio cognitivo ofrece ventajas para las aplicaciones de IoT, especialmente en entornos con muchas cosas conectadas que compiten por espectro. En sistemas de IoT, la gran cantidad de dispositivos (sensores, actuadores) puede congestionar las bandas libres (por ejemplo, banda ISM usada por LoRaWAN, Sigfox, etc.). Al integrar radio cognitivo, un nodo IoT puede operar entre bandas licenciadas y no licenciadas: por defecto usar la banda libre, pero pasar a bandas licenciadas vacantes cuando estén disponibles. Esto reduce la congestión y mejora la eficiencia espectral.

Muchos trabajos exploran este enfoque: Gurugopinath et al. (2020) revisan avances en IoT basado en radio cognitivo, señalando mejoras en el alcance y la fiabilidad del sistema. Zhang et al. (2019) destacan que compartir el espectro permite escalar la cantidad de dispositivos IoT sin degradar el servicio. Además, en aplicaciones IoT críticas (salud, monitoreo ambiental,

industriales), radio cognitivo puede asegurar prioridad cuando el espectro no licenciado falla. Por ejemplo:

Salud remota: Manna y Misra (2019) proponen un gateway para redes biomédicas (WBAN) que usa radio cognitivo para encontrar canales con licencia vacíos, logrando monitoreo continuo de pacientes de forma confiable en entornos congestionados.

Gestión de recursos y medio ambiente: Akram et al. (2021) demuestran mediante sensores LoRa-IoT que radio cognitivo puede mejorar sistemas de monitoreo de residuos sólidos, reduciendo pérdidas de paquete al evitar interferencia en bandas libres saturadas.

Redes inteligentes: Protocolos LPWAN mejorados con radio cognitivo (Salika et al., 2022) permiten cubrir áreas más amplias con menos gateways y menor congestión, siendo útiles para agricultura inteligente y control de infraestructuras.

En general, radio cognitivo puede aliviar la brecha digital rural al permitir que redes IoT capten espectro desocupado y extiendan la cobertura sin necesidad de costosa nueva infraestructura. El protocolo LoRaCog de Salika et al. (2022) es un ejemplo práctico: adapta LoRaWAN para que los dispositivos (nodos finales) puedan sensar el espectro (vía gateway) y cambiar dinámicamente a canales licenciados libres, manteniendo el bajo consumo energético característico de LoRa.

2.2.9 Ventajas de la Tecnología de Radio Cognitivo en redes IoT

Uso eficiente del espectro: La capacidad de los sistemas de radio cognitivo permite a los dispositivos IoT identificar y utilizar dinámicamente las bandas de frecuencia no utilizadas, optimizando el uso del espectro evitando asignar más espectro.

Reducción de interferencias: Al permitir el acceso dinámico al espectro, los sistemas de radio cognitivo mejoran la comunicación en entornos densamente poblados, como fabricas o ciudades inteligentes, minimizando la interferencia entre los dispositivos.

Mejora en la calidad de servicio: La capacidad de adaptar los parámetros de transmisión, como la potencia y la frecuencia permite a los dispositivos cambiar a bandas menos congestionadas, reduciendo la latencia y la pérdida de datos, lo cual es crucial en aplicaciones críticas como salud o transporte autónomo.

Escalabilidad aumentada: La capacidad de gestionar eficientemente el uso del espectro, radio cognitivo facilita la incorporación de un gran número de dispositivos sin degradar el rendimiento, útil en ciudades inteligentes o redes industriales.

Eficiencia energética: Radio cognitivo ayuda a conservar energía al reducir transmisiones innecesarias y mejorar el uso del canal, lo cual es vital en dispositivos alimentados por batería o recolectan energía del entorno.

Mayor seguridad: Técnicas de detección cognitiva pueden identificar accesos no autorizados o usuarios maliciosos en el espectro, y al combinarse con algoritmos de machine learning, se mejora la protección contra ataques y brechas de seguridad.

Confiabilidad reforzada: La habilidad de adaptarse a condiciones cambiantes del espectro mejora la conectividad continua en aplicaciones críticas, como respuesta a emergencias o automatización industrial.

Flexibilidad operativa: radio cognitivo dota a los dispositivos IoT de la capacidad de operar en diversas bandas de espectro, ajustándose en tiempo real a la disponibilidad y condiciones del entorno.

Soporte en entornos densos y dinámicos: La adaptabilidad de radio cognitivo permite mantener un rendimiento óptimo incluso en escenarios con gran variabilidad espectral, como agricultura inteligente o monitoreo ambiental.

Optimización de costos: Al maximizar el uso del espectro existente, se reducen los costos operativos relacionados con la necesidad de licencias o infraestructura adicional, especialmente en áreas urbanas o industriales densas.

2.2.10 Radio Cognitivo aplicado a LoRa

En el contexto específico de la tecnología LoRa, las fases de “Decisión” y “Movilidad” del espectro no se limitan únicamente a la selección de un canal de frecuencia. La modulación Chirp Spread Spectrum (CSS) empleada por LoRa (discutida en la sección 2.2.6) introduce una segunda dimensión fundamental para la adaptación cognitiva: el factor de ensanchamiento (Spreading Factor o SF).

La característica fundamental que hace al SF una variable cognitiva tan poderosa es que

los diferentes SFs utilizados por LoRa (típicamente de 7 a 12) son cuasi-ortogonales entre sí. Esto implica que dos transmisiones en el mismo canal de frecuencia, pero con SFs distintos (ej. SF7 y SF10), no colisionan de forma destructiva y pueden ser demoduladas por un receptor que espere ese SF específico. Esta propiedad ha sido demostrada y caracterizada en múltiples estudios, que si bien no es una ortogonalidad perfecta, permite la coexistencia de múltiples “capas lógicas” en un mismo canal (Croce et al., 2018; Reynders & Pollin, 2017).

2.2.11 Modelo del ciclo cognitivo aplicado a LoRa sobre SDR

El funcionamiento de un sistema de radio cognitivo puede describirse mediante un ciclo cognitivo cerrado que integra las etapas de percepción, decisión y acción, concepto introducido originalmente por Mitola y posteriormente formalizado en múltiples trabajos sobre radio cognitivo (Mitola, 2000; Haykin, 2005). Este ciclo permite que el sistema observe su entorno radioeléctrico, razone sobre la información obtenida y actúe adaptando sus parámetros de transmisión de manera autónoma.

En el contexto de esta tesis, el ciclo cognitivo se implementa a nivel de la capa física LoRa sobre una plataforma de radio definida por software (SDR), siguiendo arquitecturas propuestas para redes cognitivas de nueva generación (Akyildiz et al., 2006). En la etapa de percepción, el sistema realiza un sensado continuo del espectro mediante detección de energía asistida por CFAR, identificando la ocupación espectral y la presencia de interferencia en los canales disponibles. De manera complementaria, se extraen métricas de calidad del enlace, como la relación señal–ruido (SNR) y el indicador de intensidad de señal recibida (RSSI), directamente desde el receptor LoRa.

La etapa de decisión se fundamenta en el análisis de dichas métricas, priorizando parámetros directamente relacionados con el rendimiento de la capa física, tal como se recomienda en sistemas de radio cognitivo orientados a IoT y WSN debido a sus restricciones de complejidad y consumo energético (Gurugopinath et al., 2020; Wang et al., 2018). En esta fase, el motor cognitivo selecciona acciones orientadas a preservar la robustez del enlace, como la elección de canales con mayor SNR o el ajuste del factor de ensanchamiento.

Finalmente, en la etapa de acción, el sistema ejecuta la reconfiguración dinámica del trans-

ceptor mediante la resintonización de frecuencia y la modificación del factor de ensanchamiento, cerrando el ciclo cognitivo. Este enfoque concuerda con modelos de acceso dinámico al espectro, donde la adaptación de parámetros de la capa física constituye el principal mecanismo para mitigar interferencia y mejorar la eficiencia espectral en bandas no licenciadas (Yucek y Arslan, 2009; Axell et al., 2012).

La implementación de este ciclo cognitivo aplicado a LoRa permite materializar, de forma práctica, los principios teóricos del radio cognitivo en un entorno SDR, habilitando un sistema adaptativo capaz de responder en tiempo real a variaciones del entorno radioeléctrico y mejorando el rendimiento del enlace en escenarios de alta congestión espectral.

2.2.12 Técnicas de detección de espectro

La Detección de Espectro es el pilar sobre el que se construye todo el ciclo cognitivo. Su fiabilidad y precisión determinan la eficacia de cualquier decisión posterior. Aunque existen múltiples técnicas, estas se pueden clasificar por su complejidad y su robustez. La capacidad de detectar señales primarias débiles es crítica en radio cognitivo, existen múltiples técnicas:

- Detección de energía: compara la energía recibida con un umbral. Es sencilla (no requiere conocimiento de la señal primaria) pero falla a bajas relaciones señal-ruido.
- Detección ciclostacionaria: explota propiedades periódicas de las señales primarias (modulación, tramas) para diferenciarlas del ruido. Puede discriminar ruido de señales legítimas, pero requiere estructuras modulares conocidas (Gardner, 2006).
- Detección basada en auto vectores: técnicas eigenvalue-free (Zeng & Liang, 2009) analizan las correlaciones del canal con poca información previa, adaptándose a incertidumbre en ruido.
- Transformadas de dominio: como detección por wavelets (Tian & Giannakis, 2006) para escanear bandas anchas e identificar abruptos en la densidad espectral.
- Detección cooperativa: múltiples nodos cognitivos comparten decisiones (votación o fusión estadística) para mejorar la detección en entornos con desvanecimiento (Thilina et

al., 2013).

- Técnicas de aprendizaje automático: algoritmos supervisados o no supervisados (redes neuronales, SVM, K-means, etc.) pueden clasificar presencia primaria basándose en características complejas de las señales (Thilina et al., 2013).

Axell et al. (2012) resaltan que el sensado debe poder operar con señales primarias muy débiles, lo cual es difícil porque existen límites teóricos de detección a bajos SNR. En la práctica, el rendimiento de los detectores (probabilidad de detección vs. de falsa alarma) depende críticamente del tiempo de sensado y la incertidumbre del ruido.

2.2.13 Detección de Energía (ED) y sus Desafíos

El método más fundamental y de menor complejidad computacional es la Detección de Energía (ED). Su principio de funcionamiento es simple: el receptor mide la energía recibida en una banda de frecuencia durante un período de tiempo y compara esta energía con un umbral fijo predefinido. Si la energía supera el umbral, se declara que el canal está “ocupado”; de lo contrario, se declara “libre”.

Las ventajas de la Detección de energía son su simplicidad de implementación y su bajo costo computacional. Además, es un método “ciego” o no coherente, lo que significa que no requiere ningún conocimiento previo sobre las características de la señal del usuario primario o la interferencia (Axell, et al., 2012).

Sin embargo, la Detección de energía posee una desventaja crítica: su rendimiento colapsa bajo condiciones de incertidumbre del ruido (noise uncertainty). El detector de ED simple asume que la potencia del ruido de fondo (el “piso de ruido”) es constante y conocida. En un entorno de RF real, donde el piso de ruido fluctúa constantemente debido a cambios térmicos y a la interferencia lejana, un umbral fijo genera una tasa inaceptable de falsas alarmas (marcar un canal libre como ocupado) o de detecciones fallidas (no ver una interferencia real), haciendo que el sistema sea poco fiable (Yucek & Arslan, 2009).

Existen diversas técnicas de detección de espectro utilizadas en sistemas de radio cognitivo, entre las que destacan la detección de energía, la detección basada en filtros adaptados (matched

filtering) y la detección por características cicloestacionarias (Ye, et al., 2008). Sin embargo, estas técnicas presentan diferencias significativas en términos de complejidad computacional, requerimientos de información previa y viabilidad de implementación en entornos de redes de sensores inalámbricos.

La detección por filtros adaptados y la detección cicloestacionaria ofrecen mayor precisión en la identificación de señales específicas; no obstante, requieren conocimiento previo del formato de la señal o un procesamiento intensivo, lo que incrementa el consumo energético y la latencia. En contraste, la detección de energía se caracteriza por su simplicidad, bajo costo computacional y facilidad de implementación sobre plataformas SDR, convirtiéndola en una alternativa adecuada para aplicaciones WSN. Por estas razones, en esta tesis se adopta la detección de energía como técnica base, complementada con un esquema CFAR para mitigar su sensibilidad a la variación del ruido y mejorar la confiabilidad del sensado espectral (Yucek y Arslan, 2009; Axell et al., 2012).

2.2.14 Detección de Tasa Constante de Falsas Alarmas CFAR

Para resolver la limitación fundamental de la Detección de energía, el prototipo de esta tesis implementa un detector de Tasa Constante de Falsas Alarmas (CFAR). Esta es una técnica avanzada, originada en el procesamiento de señales de radar, diseñada específicamente para mantener una Probabilidad de Falsa Alarma P_{fa} constante y predecible, incluso en entornos con ruido no estacionario (Richards, 2014).

La principal ventaja del uso de técnicas CFAR en sistemas de radio cognitivo radica en su capacidad para ajustar dinámicamente el umbral de detección en función de las condiciones estadísticas del ruido y la interferencia presentes en el entorno. En escenarios de bandas ISM, donde el nivel de ocupación espectral varía de manera impredecible, el empleo de umbrales fijos resulta ineficiente y propenso a elevadas tasas de falsas alarmas o detecciones erróneas.

En particular, el esquema Cell-Averaging CFAR (CA-CFAR) permite estimar el nivel de ruido local a partir de celdas vecinas a la celda bajo prueba, garantizando una probabilidad de falsa alarma aproximadamente constante. Esta característica lo convierte en una solución adecuada para sistemas SDR orientados a WSN, ya que equilibra precisión de detección y complejidad

computacional. En el contexto de esta tesis, CFAR se emplea como mecanismo de sensado espectral adaptativo para identificar canales interferidos y habilitar decisiones cognitivas de cambio de frecuencia de manera confiable y reproducible.

Un detector CFAR no utiliza un umbral fijo. En su lugar, calcula un umbral adaptativo para cada “Celda Bajo Prueba” (Cell Under Test, CUT) que se está analizando. Este umbral se deriva de una estimación del piso de ruido local, que se obtiene promediando la energía en las “celdas de entrenamiento” (Training Cells) adyacentes (Barkat, 1991).

La variante más común, y la implementada en esta tesis, es el CFAR de Promediado de Celdas CA-CFAR. Este algoritmo desliza una ventana sobre la Densidad Espectral de Potencia PSD de la señal. Esta ventana tiene tres componentes:

- Celda Bajo Prueba CUT: El punto de frecuencia que se está decidiendo si está “ocupado” o “libre”.
- Celdas de Guarda (Guard Cells): Un pequeño búfer a cada lado de la CUT para evitar que la propia energía de la señal objetivo contamine la estimación del ruido.
- Celdas de Entrenamiento (Training Cells): Las ventanas a cada lado que se promedian para obtener una estimación en tiempo real del piso de ruido local.

El umbral adaptativo se establece como un factor (determinado por la Pfa deseada) por encima de este piso de ruido estimado. Al recalcular el umbral en cada paso, el detector se adapta automáticamente a las variaciones del ruido de fondo. Esto garantiza un rendimiento de detección robusto y predecible, lo cual es esencial para construir un motor cognitivo fiable (Richards, 2014).

CAPÍTULO III

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE RADIO COGNITIVO

En este capítulo se describe el diseño e implementación del prototipo de radio cognitivo basado en SDR, utilizando la tecnología LoRa, desarrollado en la presente investigación. El objetivo es detallar cómo se integran los elementos de hardware y software para conformar un transceptor LoRa capaz de observar el entorno radioeléctrico, tomar decisiones y adaptar sus parámetros físicos en tiempo de ejecución.

En primer lugar, se presentan los criterios de diseño y las delimitaciones del sistema. A continuación, se describen la plataforma de software basada en GNU Radio y las modificaciones realizadas al módulo *gr-lora_sdr*, así como la descripción del equipamiento. Posteriormente, se explica la implementación del enlace LoRa de referencia o *baseline* (sin capacidades cognitivas) y se introduce el conjunto de bloques desarrollados para dotar al sistema de funciones de sensado espectral, selección de canal y adaptación del factor de ensanchamiento (SF).

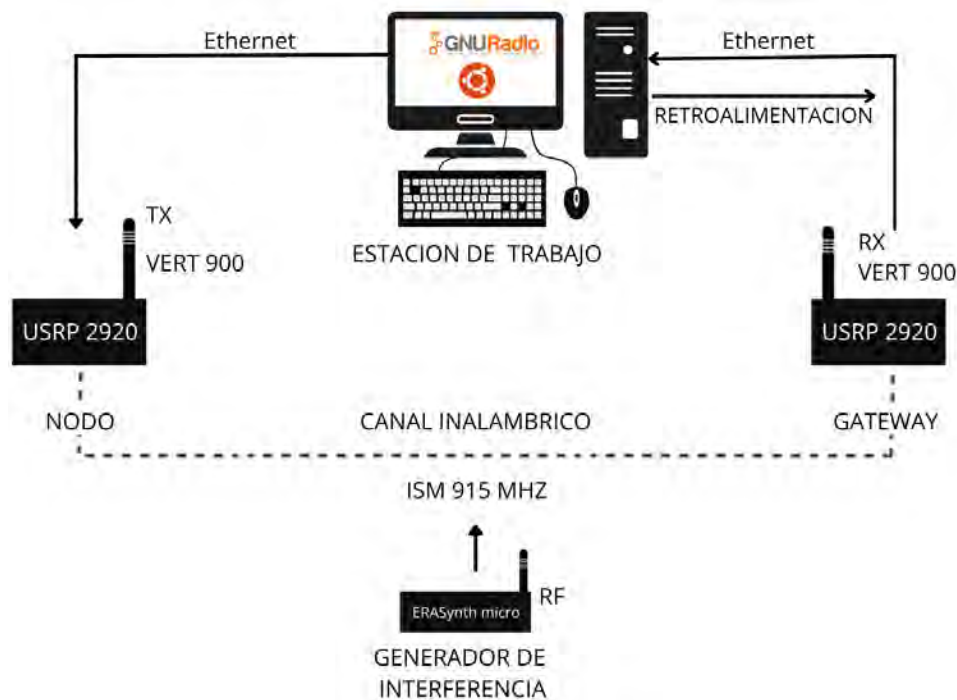


Figura 7
Diagrama de implementación del prototipo de Radio Cognitivo.

3.1 Criterios de diseño

El prototipo de radio cognitivo está fundamentado en la arquitectura SDR (*Software Defined Radio*) para permitir la flexibilidad y reconfiguración dinámica, características esenciales para radio cognitivo.

A nivel de hardware, se utilizaron dos transceptores USRP 2920, los cuales ofrecen capacidades de transmisión y recepción en bandas de frecuencia adecuadas para WSN y experimentación cognitiva. A estos equipos se acoplaron antenas VERT900 para optimizar la eficiencia del enlace. Asimismo, se emplearon atenuadores de precisión de $50\ \Omega$, sumando una atenuación total de 32 dB (uno de 30 dB y otro de 2 dB en serie) para proteger y estabilizar la etapa de recepción en entornos de laboratorio. El entorno de desarrollo se basó en *GNU Radio*, ejecutado en un ordenador con sistema operativo Ubuntu 22.04. Para la validación bajo condiciones hostiles, se utilizó el generador de señales *ERAsynth Micro* con el fin de inyectar interferencia controlada y evaluar el desempeño del prototipo de radio cognitivo.

En cuanto al software, se seleccionó el módulo *gr-lora_sdr* (disponible en repositorio GitHub) para implementar la capa física de LoRa sobre SDR. Esta elección facilita la integración con protocolos de redes WSN de baja potencia y largo alcance, alineándose con los objetivos del prototipo.

La validación experimental del prototipo se restringió a una arquitectura de enlace punto a punto compuesta por un nodo y un gateway. Esta delimitación se fundamenta en la topología de red en estrella inherente al protocolo LoRaWAN, donde no existe comunicación tipo malla entre nodos y cada enlace opera de manera independiente. Dado que el algoritmo cognitivo propuesto ejecuta la toma de decisiones (adaptación de SF y canal de transmisión), demostrar la robustez y la recuperación de la conectividad en un enlace individual constituye la condición necesaria y suficiente para validar la funcionalidad del sistema en una red escalable. Este enfoque, que prioriza la caracterización de la capa física (PHY) sobre la complejidad del tráfico de red, es consistente con los estándares experimentales recientes para implementaciones de LoRa sobre SDR (Busacca et al., 2024; Slabicki et al., 2018). En cuanto a la escalabilidad de la herramienta base, Xhonneux et al. (2020) demostraron que *gr-lora_sdr* permite la decodificación simultánea

de dos usuarios con el mismo SF en escenarios de colisión mediante un detector de máxima fiabilidad implementado sobre GNU Radio. La extensión a redes de mayor densidad ha sido validada sobre la misma PHY de LoRa usando plataformas SDR equivalentes: Wang et al. (2019) reportaron decodificación de hasta tres transmisiones concurrentes con mejora de rendimiento de $3\times$ mediante cancelación de interferencia sucesiva (SIC), y Xia et al. (2022) demostraron decodificación concurrente de hasta 40 nodos LoRa simultáneos en un banco de pruebas de 8 USRPs sincronizados, con una mejora de rendimiento de $4,9\times$. Estos resultados sustentan la viabilidad de escalar el prototipo desarrollado en esta tesis hacia escenarios de red densa WSN en trabajos futuros.

Tabla 2

Validación de escalabilidad en sistemas LoRa multiusuario sobre plataformas SDR

Referencia	Plataforma	Freq. / SF	Nodos	Rendimiento	PRR / SER	Tipo de validación
Xhonneux et al. (2020)	GNU Radio + gr-lora_sdr, USRP (EPFL)	ISM / mismo SF	2	No reportado	Decodificación exitosa a SNR muy bajo	Experimental (hardware real)
Wang et al. (2019)	6 USRP (testbed), GNU Radio, 2 MHz	915 MHz / SF7–SF13	hasta 3	$\sim 3\times$	No reportado numéricamente	Experimental (6 nodos)
Xia et al. (2020)	SDR + GNU Radio	ISM / mismo SF	hasta 3	Creciente con nodos	SER < 0.1 con 3 nodos	Experimental
Tesfay et al. (2020)	Simulación (LoRa PHY, SF=8, BW=250 kHz)	ISM / SF8	múltiples	Mejora significativa de capacidad reportada	BER decreciente con incremento de nodos	Simulación
Xia et al. (2022)	8 USRP N210 sincronizados + gr-lora	ISM / mismo SF	hasta 40	$4.9\times$ sobre estado del arte	PRR creciente con antenas	Experimental (indoor + outdoor 1.08 km ²)

3.2 Selección e implementación en GNU Radio

Para esta tesis se seleccionó el módulo *gr-lora_sdr* como núcleo del prototipo de radio cognitivo debido a su total compatibilidad con el entorno de desarrollo: es un módulo *Out-Of-Tree* (OOT) nativo para GNU Radio 3.10 y el controlador UHD, compatible con los dispositivos USRP-2920. Este módulo expone bloques completos de Transmisión (Tx) y Recepción (Rx), permitiendo el acceso a parámetros clave como SF, CR, BW, *sync word* desde el diagrama de flujo (*flowgraph*).

Una ventaja decisiva de esta implementación es la compatibilidad con hardware comercial: a diferencia de simuladores puros, *gr-lora_sdr* genera formas de onda física y espectralmente idénticas a las de los chips Semtech (e.g., SX1276), permitiendo que el prototipo cognitivo se integre de forma transparente en una WSN real mixta (SDR y nodos comerciales).

El sistema destaca por su robustez frente a niveles bajos de SNR e implementa la cadena PHY completa (sincronización con corrección de STO/CFO, FEC, *whitening/interleaver* y CRC), lo cual es crítico al introducir interferencia intencional y forzar la resintonización de frecuencia. Al ser de código abierto y contar con documentación reciente (como en la conferencia GRCon 2024), permitió la modificación del código fuente para automatizar cambios en el factor de ensanchamiento y la frecuencia portadora dentro del bucle cognitivo.

Los parámetros de frecuencia central, ancho de banda (BW), factor de ensanchamiento (SF) y tasa de codificación (CR) pueden ajustarse dinámicamente mediante el sistema de paso de mensajes de GNU Radio, materializando así las políticas de radio cognitivo, tales como la resintonización de canal o la adaptación de SF ante interferencia. En resumen, *gr-lora_sdr* fue elegido porque minimiza la complejidad de integración, cubre toda la capa física con buen desempeño y permite tanto la simulación sin hardware como la experimentación real con nodos comerciales, facilitando la automatización de las decisiones cognitivas en un entorno escalable.

3.2.1 Implementación de LoRa sobre SDR

La implementación se realizó sobre el sistema operativo Ubuntu 22.04 LTS, con el software GNU Radio 3.10. El módulo *gr-lora_sdr* se puede instalar desde la fuente o directamente como un paquete conda el cual es un gestor de sistemas de código abierto, que permite crear entornos aislados para proyectos en *Linux*.

Clonación y compilación del módulo

Esta instalación genera los binarios, donde se encuentran los bloques de transmisión y recepción LoRa. El repositorio *gr-lora_sdr* fue clonado desde la fuente oficial (EPFL–TCL), que implementa una versión abierta de la capa física de LoRa:

Segmento de código 3.1: Clonación e instalación de gr-lora_sdr

```
git clone https://github.com/tapparelj/gr-lora_sdr.git

cd gr-lora_sdr
mkdir build && cd build
cmake ..
make -j4
sudo make install
sudo ldconfig
```

En la implementación de GNU Radio de las cadenas LoRa Tx y Rx, el usuario puede elegir todos los parámetros de la transmisión, como el factor de ensanchamiento, la tasa de codificación, el ancho de banda, la palabra de sincronización, la presencia de un encabezado explícito y CRC (Tapparel et al., 2024).

La implementación se puede utilizar para obtener resultados de rendimiento experimentales de extremo a extremo de un receptor LoRa SDR con relaciones señal-ruido (SNR) bajas.

En la cadena Tx, la implementación contiene todos los bloques principales del transceptor LoRa: los bloques de inserción de encabezado y CRC, el bloque de *whitening*, el bloque del codificador Hamming, el bloque del entrelazador, el bloque de desmapeo de Gray y el bloque de modulación (Joachim et al., 2020)

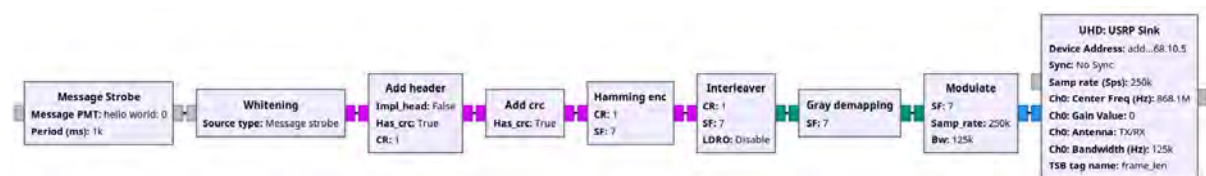


Figura 8
Cadena Tx Lora

Whitening: El bloque *Whitening* aplica un blanqueo de datos: realiza una operación XOR entre los bits de la carga útil y una secuencia pseudoaleatoria conocida. Este proceso elimina sesgos de DC y evita largas secuencias de ceros o unos, lo que mejora el comportamiento del modulador y de los algoritmos de sincronización, sin cambiar la tasa de datos efectiva. En la

implementación de *gr-lora_sdr* el *Whitening* se aplica antes de la codificación *Hamming* y del entrelazado, siguiendo el esquema descrito en el reverse engineering de la PHY LoRa.

Add header El bloque *Add header* inserta el encabezado de la trama LoRa, que contiene información de control necesaria para que el receptor pueda decodificar correctamente el paquete. Entre otros campos, el encabezado codifica:

- Longitud de la carga útil
- Tasa de codificación configurado
- Presencia o no de CRC de payload
- Bits de reserva y de control

En LoRa el encabezado se codifica siempre con el tasa de codificación más robusto (4/8) para maximizar la probabilidad de decodificación correcta.

Add crc Este bloque calcula y añade el CRC de la carga útil (Payload CRC). En la PHY LoRa se utiliza un CRC de 16 bits basado en el polinomio CCITT-16; dicho CRC se calcula sobre los bytes de payload y se agrega al final antes de la codificación y del entrelazado. En el receptor se recalcula el mismo CRC para verificar la integridad de los datos.

Hamming enc El bloque *Hamming enc* implementa el código *Hamming* modificado utilizado por LoRa como FEC (Forward Error Correction). Trabaja a nivel de nibble (4 bits de datos) y genera palabras de longitud mayor según el tasa de codificación (4/5, 4/6, 4/7 o 4/8). Los bits de paridad se calculan mediante combinaciones XOR específicas, como se describe en el análisis de ingeniería inversa de la PHY LoRa. Su objetivo es detectar y corregir errores de bit provocados por el canal antes de que estos afecten a los datos de usuario.

Interleaver El bloque *Interleaver* realiza el entrelazado de las palabras codificadas. En LoRa los bits de cada código *Hamming* se redistribuyen entre varios símbolos LoRa distintos, de manera que los errores producidos por la degradación de un solo símbolo (fading, interferencia) se reparten sobre varios *codewords* en vez de concentrarse en uno solo. Este proceso mejora la

efectividad del FEC, ya que el código *Hamming* está diseñado para corregir errores dispersos y no ráfagas de errores altamente correlacionadas.

Gray demapping En la cadena transmisora este bloque implementa el mapeo Gray de los símbolos LoRa. A partir de los bits entrelazados genera un entero en el rango $[0; 2^{SF} - 1]$ que será el valor de símbolo que modula cada chirp. LoRa utiliza un código Gray “invertido” con un desplazamiento de -1 para minimizar el número de bits que cambian cuando, por efecto de ruido o de errores de sincronización, un símbolo se confunde con su vecino.

Modulate El bloque *Modulate* implementa la modulación LoRa basada en Chirp Spread Spectrum (CSS). Toma como entrada la secuencia de símbolos enteros y genera las correspondientes formas de onda tipo *upchirp* con desplazamiento de frecuencia. Para un ancho de banda BW y un factor de ensanchamiento SF, el período de símbolo es:

$$T_s = \frac{2^{SF}}{BW}$$

y cada símbolo S se codifica como un chirp con un offset de frecuencia proporcional a S.

Los parámetros clave en el bloque son:

- SF (robustez y tasa de datos)
- sample rate (Sps)
- BW (ancho de banda de la señal LoRa)

En el lado del receptor se encuentra el bloque de sincronización de paquetes, que realiza todas las tareas necesarias para la sincronización, como la estimación y corrección de STO y CFO. A continuación, se encuentran el bloque de demodulación, el bloque de mapeo de Gray, el bloque de desentrelazado, el bloque de decodificación de Hamming y el bloque de *dewhitening*, así como un bloque de verificación CRC.

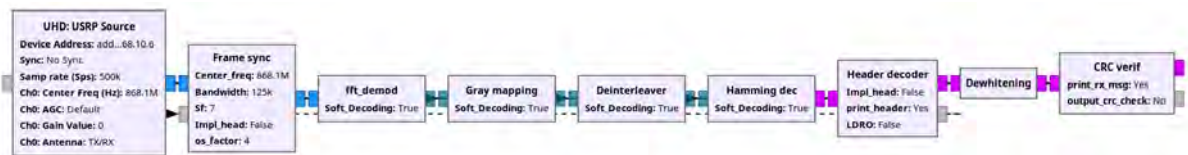


Figura 9
Cadena Rx Lora

Frame sync El bloque Frame sync realiza la sincronización de trama. Detecta el preámbulo LoRa (secuencia de upchirps y downchirps) y estima:

- El instante de inicio de paquete
- Correcciones gruesas de frecuencia y fase
- Alineación de las ventanas de símbolo

Se basa en correlación con chirps de referencia y en la estructura de preámbulo descrita para LoRa (upchirps repetidos, símbolos de sincronización y downchirps).

fft_demod Este bloque implementa la demodulación de los chirps. Para cada símbolo recibido multiplica la señal por el conjugado del chirp de referencia (dechirping) y luego aplica una transformada rápida de Fourier FFT. El índice de la bin de FFT con mayor energía corresponde al valor del símbolo S transmitido. La salida son símbolos discretos (o métricas “blandas” si se usa soft decoding) que representan los enteros entre 0 y $2^{SF} - 1$.

Gray mapping En la recepción, el bloque Gray mapping aplica la conversión inversa del código Gray usado en el transmisor. Convierte cada símbolo entero en el patrón de bits correspondiente, recuperando así las secuencias binarias que alimentarán al desentrelazador. En modo Soft_Decoding el bloque genera probabilidades o LLRs por bit en lugar de decisiones duras, lo que mejora el rendimiento del FEC.

Deinterleaver El bloque Deinterleaver revierte el entrelazado aplicado en transmisión. Reorganiza los bits provenientes de distintos símbolos para reconstruir las palabras de código Hamming originales, siguiendo la misma ley de permutación pero en sentido inverso. Este proceso vuelve a concentrar en cada codeword la información correspondiente a un nibble de datos, permitiendo que el decodificador Hamming actúe correctamente.

Hamming dec El bloque Hamming dec realiza la decodificación y corrección de errores de los códigos Hamming modificados de LoRa. Calcula el síndrome de cada codeword utilizando la matriz de comprobación de paridad adecuada y, en función del resultado, detecta y corrige hasta un bit erróneo por palabra (en CR 4/7 y 4/8) o detecta errores no corregibles (en CR 4/6). Su salida son los nibbles de datos ya libres de la redundancia de FEC.

Header decoder Este bloque interpreta las primeras palabras de datos como cabecera explícita LoRa. Extrae de los bits de cabecera:

- Longitud de la carga útil
- tasa de codificación utilizado
- Presencia de payload CRC

otros campos de control definidos en la especificación. A partir de esa información, el receptor sabe cuántos símbolos debe procesar y cómo decodificar el resto de la trama.

Dewhitening El bloque Dewhitening aplica el proceso inverso al blanqueo: XOR entre los bits recibidos y la misma secuencia pseudoaleatoria utilizada en transmisión. Con ello se recupera la carga útil original, eliminando el efecto del *whitening* aplicado antes de la codificación.

CRC verif Finalmente, el bloque CRC verif recalcula el CRC de la carga útil usando el mismo polinomio CCITT-16 y compara el resultado con el CRC recibido. Si la verificación es correcta, el bloque entrega el mensaje decodificado (por ejemplo, lo imprime en consola cuando `print_rx_msg = Yes`). Si el CRC no coincide, la trama se descarta o se marca como errónea.

3.3 Configuraciones para radio cognitivo

El objetivo central de esta tesis es dotar a un enlace LoRa la capacidad de detectar degradación en el canal y reaccionar de forma autónoma, modificando su frecuencia de operación y su factor de ensanchamiento (SF) sin interrumpir la comunicación. Para lograrlo fue necesario intervenir directamente en el código fuente del módulo `gr-lora_sdr`, ya que sus bloques

originales fueron diseñados para operar con parámetros fijos durante toda la ejecución. Esta sección documenta las modificaciones realizadas en cuatro archivos clave del módulo original: `frame_sync_impl.cc`, `frame_sync_impl.h`, `modulate_impl.cc` y `modulate_impl.h`. Estos cambios transformaron el transceptor LoRa de un sistema estático en uno reconfigurable en tiempo de ejecución, habilitando el ciclo cognitivo *sentir*, *decidir* y *actuar* que se evalúa experimentalmente en el Capítulo 4.

3.3.1 Cambios estructurales en `frame_sync_impl.cc`

El archivo `frame_sync_impl.cc` es el primer eslabón activo de la cadena de recepción LoRa. Su función es detectar el preámbulo de cada trama entrante una secuencia de upchirps y downchirps y a partir de él estimar y corregir los errores de temporización (STO) y de frecuencia (CFO) que introduce el canal. Una vez sincronizado, entrega los símbolos de la carga útil al siguiente bloque para su demodulación.

En su versión original, este bloque es completamente estático el SF, el ancho de banda y las palabras de sincronización se fijan en el constructor y no pueden modificarse durante la ejecución. Esto significa que si el motor cognitivo decide cambiar de SF7 a SF8 como se vio en las pruebas descritas en la sección 4.2 el receptor no sería capaz de sincronizarse con la nueva señal, porque sus chirps de referencia y el plan de FFT seguirían dimensionados para SF7.

Para resolver este problema, se introdujo la función pública `set_sf(int sf)`, la cual constituye la modificación principal de este archivo. Cuando el motor cognitivo invoca esta función con un nuevo valor de SF, el bloque ejecuta una secuencia completa de reconfiguración: actualiza el número de bins (`m_number_of_bins = 1 << sf`), redimensiona todos los búferes internos dependientes de dicho valor incluidos los vectores de referencia `upchirp` y `downchirp`, los búferes de preámbulo y las memorias de corrección de CFO y reconstruye el plan de FFT con el nuevo tamaño mediante `kiss_fft_alloc`. Este procedimiento permite que el receptor quede preparado para sincronizarse con la nueva señal antes de la llegada de la siguiente trama.

La función puede ser invocada de dos maneras complementarias. La primera es asíncrona, a través de un puerto de mensajes registrado como “set”: cuando el bloque `SfDeciderMin` decide aumentar el SF por degradación sostenida del SNR, envía un diccionario `sf: < int >` que

dispara la reconfiguración de inmediato. La segunda es síncrona, mediante stream tags con la clave “new_frame” que viajan incrustadas en el flujo de muestras I/Q: al detectar una de estas etiquetas en general_work, el bloque lee el SF indicado y llama internamente a set_sf antes de procesar la trama correspondiente. Esta segunda vía permite que transmisor y receptor se reconfiguren de forma coordinada, paquete a paquete.

Adicionalmente, se habilitó un puerto de salida de mensajes “snr_out” que publica, tras cada trama decodificada, los metadatos del enlace: SNR estimada, CFO y STO. Esta información es la que alimenta al bloque SNRLogger y, a través de él, a los decisores cognitivos ChannelPickerAndTune_SNR y SfDeciderMin.

Fragmentos de Código de Modificaciones

Segmento de código 3.2: Registro del puerto de mensajes para cambio asíncrono de SF

```
message_port_register_in(pmt::mp("set"));
set_msg_handler(pmt::mp("set"), [this](pmt::pmt_t msg) {
    if (!pmt::is_dict(msg)) return;
    pmt::pmt_t ksf = pmt::intern("sf");
    if (pmt::dict_has_key(msg, ksf)) {
        int sf_new = (int)pmt::to_long(pmt::dict_ref(msg, ksf, pmt::
            from_long(this->m_sf)));
        this->set_sf(sf_new);
        std::cout << "[frame_sync] set.sf -> " << sf_new << std::endl;
    }
});
```

Segmento de código 3.3: Núcleo de la función set_sf, reconfiguración completa del receptor

```
void gr::lora_sdr::frame_sync_impl::set_sf(int sf) {
    if (sf < 7 || sf > 12) return;
    if ((int)m_sf == sf) return;
    m_sf = (uint8_t)sf;
    m_number_of_bins = (uint32_t)(1u << m_sf);
```

```

m_samples_per_symbol = m_number_of_bins * m_os_factor;
m_upchirp.resize(m_number_of_bins);
m_downchirp.resize(m_number_of_bins);

build_ref_chirps(&m_upchirp[0], &m_downchirp[0], m_sf);

kiss_fft_free(m_kiss_fft_cfg);
m_kiss_fft_cfg = kiss_fft_alloc(m_number_of_bins, 0, 0, 0);
set_output_multiple(m_number_of_bins);
}

```

Segmento de código 3.4: Lectura de “stream tags” para reconfiguración sincronizada por paquete

```

std::vector<tag_t> tags;
get_tags_in_window(tags, 0, 0, ninput_items[0],
                  pmt::string_to_symbol("new_frame"));
if (tags.size()) {
    int sf = pmt::to_long(pmt::dict_ref(tags[0].value,
    pmt::string_to_symbol("sf"), err));
    set_sf(sf);
}

```

3.3.2 Cambios estructurales en *frame_sync_impl.h*

El archivo de cabecera *frame_sync_impl.h* define la estructura de la clase: qué variables almacena el bloque y qué métodos expone al exterior. En su versión original, la clase no declaraba ninguna función para modificar el SF en tiempo de ejecución, lo que impedía cualquier intento de reconfiguración externa.

Los cambios realizados en este archivo son el complemento directo de los cambios en el .cc. El más importante es la declaración pública de `void set_sf(int sf) override`, que expone la capacidad de reconfiguración al *flowgraph* y a otros bloques. Sin esta declaración, la función implementada en el .cc sería inaccesible desde fuera del bloque.

Además, se amplió la firma del constructor para aceptar los parámetros `sync_word`, `os_factor` y `preamb_len` como argumentos, eliminando los valores que en la versión original estaban directamente codificados en el `.cc`. También se añadieron las variables miembro correspondientes (`m_sync_words`, `m_os_factor`, `m_preamb_len`) y el identificador del puerto de salida `d_snr_port`, que habilita la publicación de métricas hacia los bloques cognitivos. Finalmente, los múltiples manejadores de mensajes del receptor se consolidaron en `frame_info_handler` y `noise_est_handler`, simplificando la interfaz interna del bloque.

Fragmentos de Código de Modificaciones Clave (Headers)

Segmento de código 3.5: El Constructor (Más Flexible y Parametrizado)

```
public:
    frame_sync_impl(uint32_t center_freq, uint32_t bandwidth, uint8_t sf,
        bool impl_head,
        std::vector<uint16_t> sync_word, uint8_t os_factor, uint16_t preamb_len
    );
```

Segmento de código 3.6: Declaración pública de `set_sf` y puerto de SNR

```
void set_sf(int sf) override;
private:
    pmt::pmt_t d_snr_port;
    uint8_t m_os_factor;
    std::vector<uint16_t> m_sync_words;
    void frame_info_handler(pmt::pmt_t frame_info);
    void noise_est_handler(pmt::pmt_t noise_est);
```

3.3.3 Cambios estructurales en `modulate_impl.cc`

El bloque `modulate` es el equivalente en transmisión de lo que `frame_sync` representa en recepción. Su función es tomar los símbolos enteros de la carga útil y construir la señal de banda

base LoRa completa: primero genera la secuencia de preámbulo (*upchirps*, palabras de sincronización y *downchirps*) y luego modula cada símbolo como un *upchirp* con el desplazamiento de frecuencia correspondiente al valor del símbolo.

En su versión original, este bloque también operaba con parámetros fijos. La longitud del preámbulo estaba codificada como `n_up = 8`, y las palabras de sincronización eran los enteros fijos 8 y 16, escritos directamente en la lógica de `general_work`. Para que el sistema cognitivo pudiera cambiar el SF del transmisor de forma coordinada con el receptor, fue necesario hacer este bloque igualmente reconfigurable.

El cambio principal fue introducir la función `update_var(int new_sf, int new_bw)`, que encapsula toda la lógica de reconfiguración del modulador. Cuando se invoca con un nuevo SF, esta función actualiza las variables derivadas (`m_number_of_bins`, `m_samples_per_symbol`, `m_os_factor`), redimensiona los vectores de chirps de referencia y los regenera llamando a `build_ref_chirps`. Esto garantiza que el modulador produzca señales coherentes con el nuevo SF antes de transmitir la siguiente trama.

Al igual que en el receptor, la reconfiguración puede activarse por dos vías. La vía asíncrona utiliza un nuevo puerto de mensajes “set” que acepta el mismo formato de diccionario `{sf: <int>}` que usa el receptor. La vía síncrona lee *stream tags* con la clave “configuration” incrustadas en el flujo de símbolos de entrada; al detectar una al final de una trama, el modulador actualiza sus parámetros para la siguiente.

Fragmentos de Código de Modificaciones Clave

Segmento de código 3.7: Control Asíncrono por Mensajes (en el constructor)

```
message_port_register_in(pmt::mp("set"));
set_msg_handler(pmt::mp("set"), [this](pmt::pmt_t msg) {
    if (!pmt::is_dict(msg)) return;
    pmt::pmt_t ksf = pmt::intern("sf");
    if (pmt::dict_has_key(msg, ksf)) {
        int sf_new = (int)pmt::to_long(pmt::dict_ref(msg, ksf,
            pmt::from_long(this->m_sf)));
```

```

        this->set_sf(sf_new);
    }
});

```

Segmento de código 3.8: Función `update_var` para la reconfiguración del modulador

```

void modulate_impl::update_var(int new_sf, int new_bw) {
    m_sf = new_sf;
    m_bw = new_bw;
    m_number_of_bins = (uint32_t)(1u << m_sf);
    m_os_factor = m_samp_rate / m_bw;
    m_samples_per_symbol = m_number_of_bins * m_os_factor;
    m_downchirp.resize(m_samples_per_symbol);
    m_upchirp.resize(m_samples_per_symbol);
    build_ref_chirps(&m_upchirp[0], &m_downchirp[0], m_sf, m_os_factor);
    set_output_multiple(m_samples_per_symbol);
}

```

Segmento de código 3.9: Reconfiguración síncrona por “stream tags” al final de cada trama

```

get_tags_in_window(tags, 0, 0, 1, pmt::string_to_symbol("configuration"));
if (tags.size() > 0) {
    int new_sf = pmt::to_long(pmt::dict_ref(tags[0].value,
        pmt::string_to_symbol("sf"), err_sf));
    int new_bw = pmt::to_long(pmt::dict_ref(tags[0].value,
        pmt::string_to_symbol("bw"), err_bw));
    update_var(new_sf, new_bw);
}

```

3.3.4 Cambios estructurales en `modulate_impl.h`

El archivo de cabecera `modulate_impl.h` define la estructura de la clase `modulate_impl`: qué variables almacena el bloque y qué métodos expone al exterior. En su versión original, el

constructor solo aceptaba tres parámetros básicos (*sf*, *samp_rate*, *bw*), y la clase no declaraba ninguna función para modificar el SF ni el BW después de la inicialización, lo que impedía cualquier forma de reconfiguración externa.

Los cambios realizados en este archivo son el complemento directo de los cambios en el .cc descritos en la sección 3.3.3. El más importante es la declaración pública de `void set_sf(int sf) override`, que expone la capacidad de reconfiguración al *flowgraph* y a los bloques decisores. Sin esta declaración en la cabecera, la función implementada en el .cc sería inaccesible desde el motor cognitivo.

Se añadió el prototipo de `void update_var(int new_sf, int new_bw)`, que encapsula la lógica de reconfiguración interna y es invocada por `set_sf` y por el lector de *stream tags* en *general_work*. Finalmente, se incorporaron las nuevas variables miembro `m_sync_words`, `m_inter_frame_padding`, `m_preamb_len` y `m_os_factor`, necesarias para que el modulador pueda recalcular correctamente todos los parámetros derivados ante cualquier cambio de SF o BW en tiempo de ejecución.

Los fragmentos de código más relevantes de estas modificaciones se presentan a continuación.

Fragmentos de Código de Modificaciones Clave (Headers)

Segmento de código 3.10: Constructor modificado — trama completamente parametrizable

```
public:
    modulate_impl(uint8_t sf, uint32_t samp_rate, uint32_t bw,
                  std::vector<uint16_t> sync_words,
                  uint32_t frame_zero_padd,
                  uint16_t preamb_len);
    ~modulate_impl();
```

Segmento de código 3.11: Declaración pública de `set_sf`, `update_var` y nuevas variables miembro

```
void set_sf(int sf) override; //
void update_var(int new_sf, int new_bw); //
```

```
private:
    uint32_t m_number_of_bins; //
    int m_os_factor; //
    std::vector<uint16_t> m_sync_words; //
    uint32_t m_inter_frame_padding; //
    uint16_t m_preamb_len;
```

3.4 Descripción de equipamiento

3.4.1 Criterios

La selección del hardware para este prototipo respondió a cuatro necesidades concretas derivadas de los objetivos de la tesis. En primer lugar, se requería un transceptor con control programático de frecuencia y ganancia, para poder medir con precisión los tiempos de reconfiguración del enlace durante los eventos cognitivos. En segundo lugar, el sistema debía permitir extraer métricas de capa física concretamente SNR y RSSI directamente de las muestras I/Q procesadas en GNU Radio, sin depender de estimaciones externas. En tercer lugar, era necesario contar con un generador de interferencia controlada que permitiera reproducir condiciones de congestión co-canal de forma repetible, condición indispensable para comparar el comportamiento del sistema cognitivo frente al enlace base. Finalmente, todos los equipos debían ser compatibles con el entorno de desarrollo GNU Radio sobre Ubuntu 22.04, dado que la implementación completa (incluyendo los bloques cognitivos) residen en el software.

3.4.2 Inventario de equipos

SDR principal (Tx/Rx): NI USRP-2920. Se emplearon dos unidades USRP-2920, una como nodo transmisor y otra como gateway receptor, constituyendo el núcleo del enlace LoRa experimental. Este equipo opera en el rango de 50 MHz a 2.2 GHz, cubriendo íntegramente la banda ISM de 915 MHz utilizada en para tecnología LoRa en Perú. Sus convertidores analógico-digitales de 14 bits proporcionan un margen dinámico suficiente para decodificar señales LoRa en condiciones de SNR negativo, tal como las que se generaron durante las pruebas de interferencia. Un aspecto importante del diseño experimental fue deshabilitar el Control

Automático de Ganancia (AGC) en ambas unidades, operando con ganancia fija durante todas las pruebas. Esta decisión asegura que las variaciones observadas en el SNR y el RSSI reflejen exclusivamente las condiciones del canal y la acción del interferente, y no compensaciones automáticas del hardware que podrían enmascarar el comportamiento real del sistema cognitivo.

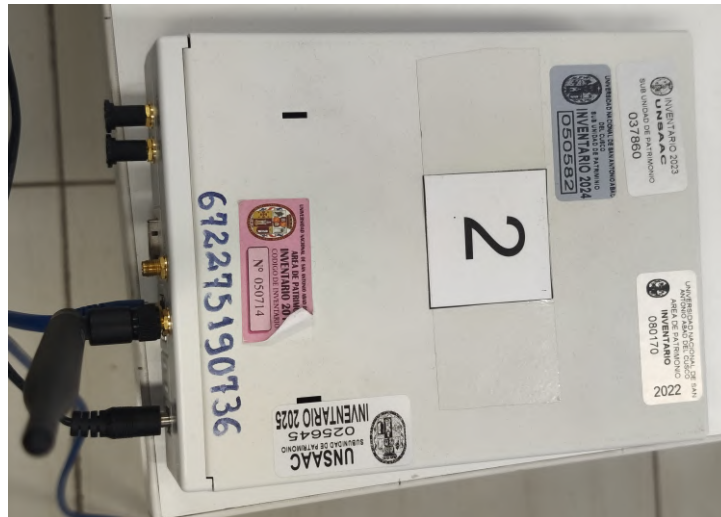


Figura 10
USRP 2920

Antenas omnidireccionales VERT900. El enlace inalámbrico se implementó utilizando un par de antenas VERT900 de polarización vertical, optimizadas para el rango de 824 MHz a 960 MHz, lo que las hace adecuadas para la frecuencia de operación de 915.6 MHz empleada en los experimentos. Se utilizaron unidades idénticas en transmisión y recepción para mantener la coherencia de polarización y minimizar las pérdidas por desajuste.

Atenuadores y accesorios RF. Dado que las pruebas se realizaron en el entorno controlado del laboratorio del LIISTTI, con los transceptores separados por aproximadamente 1.97 metros, fue necesario insertar atenuadores entre la etapa de transmisión y el receptor para proteger el hardware y garantizar mediciones válidas. Sin atenuación, la potencia recibida habría saturado el amplificador de bajo ruido del USRP-2920, ingresando a la región no lineal del receptor y generando distorsión de intermodulación que habría invalidado las lecturas de SNR utilizadas por el motor cognitivo. Para prevenir esto, se conectaron en serie dos atenuadores de 50 Ω que suman 32 dB (30 dB y 2 dB respectivamente), forzando la señal recibida a niveles seguros de operación. Esta configuración permitió realizar las pruebas de forma repetible y segura, sin riesgo de daño al hardware ni de contaminación de las métricas medidas.



Figura 11
Antena VERT900 y atenuadores.

Generador de interferencia: ERASynth Micro. Para simular la congestión espectral co-canal que motiva esta tesis, se utilizó el generador de señales ERASynth Micro. Este dispositivo, basado en una arquitectura PLL fraccional-N de doble lazo, genera tonos de onda continua (CW) con alta pureza espectral, lo que garantiza que la perturbación introducida se concentre en la frecuencia objetivo sin elevar el piso de ruido del resto del espectro. En las pruebas, se configuró para emitir en 915.6 MHz la misma frecuencia central del enlace LoRa y su potencia de salida se incrementó progresivamente en pasos de 3 dB para caracterizar el umbral de colapso del sistema no cognitivo y verificar la respuesta adaptativa del prototipo. La reproducibilidad de esta fuente de interferencia fue fundamental para que los resultados del Capítulo 4 sean comparables entre escenarios.



Figura 12
ERASynth Micro

Estación de trabajo. El procesamiento digital de ambos flujos SDR se realizó en un equipo de cómputo con sistema operativo GNU/Linux (Ubuntu 22.04 LTS), con GNU Radio 3.10 y los controladores UHD instalados. El equipo dispone de puertos Gigabit Ethernet dedicados a cada USRP y memoria RAM suficiente para ejecutar de forma simultánea el detector CA-CFAR, los bloques de FFT del escáner espectral y la cadena de recepción LoRa sin pérdida de muestras.

La estabilidad computacional de esta plataforma fue un requisito indispensable, ya que los eventos cognitivos de cambio de canal y de SF dependen de que el flujo de muestras I/Q no se interrumpa durante la reconfiguración.

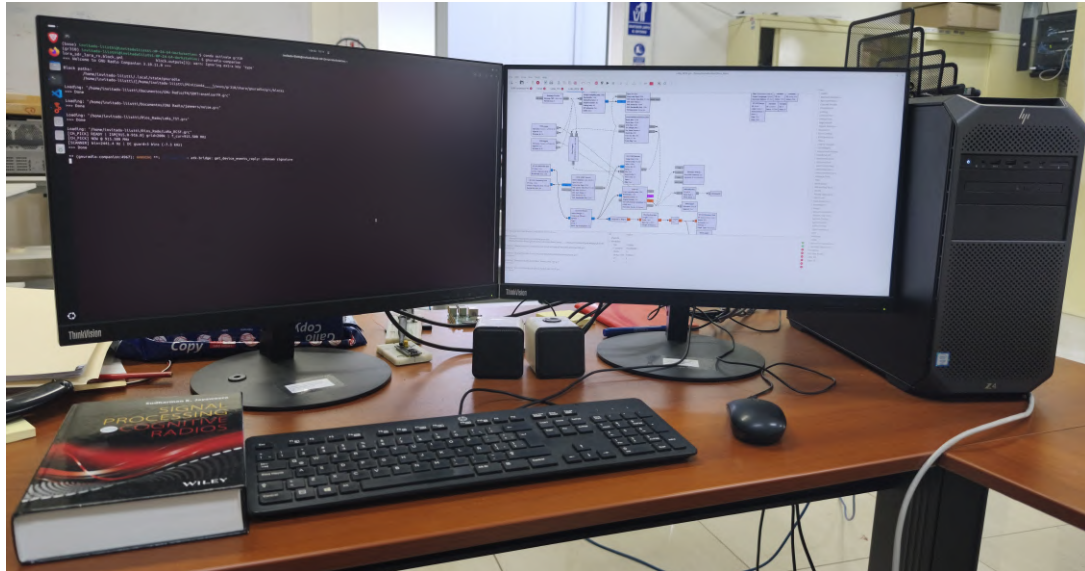


Figura 13
Estación de trabajo (Linux)

La Figura 14 muestra la configuración física completa del banco de pruebas tal como fue montada en el LIISTTI, integrando los equipos descritos anteriormente. Los USRP se ubicaron a 1.97 m entre sí sobre la mesa de trabajo, mientras que el ERASynth Micro se posicionó a 0.87 m del receptor para simular interferencia co-canal en condiciones controladas y reproducibles.

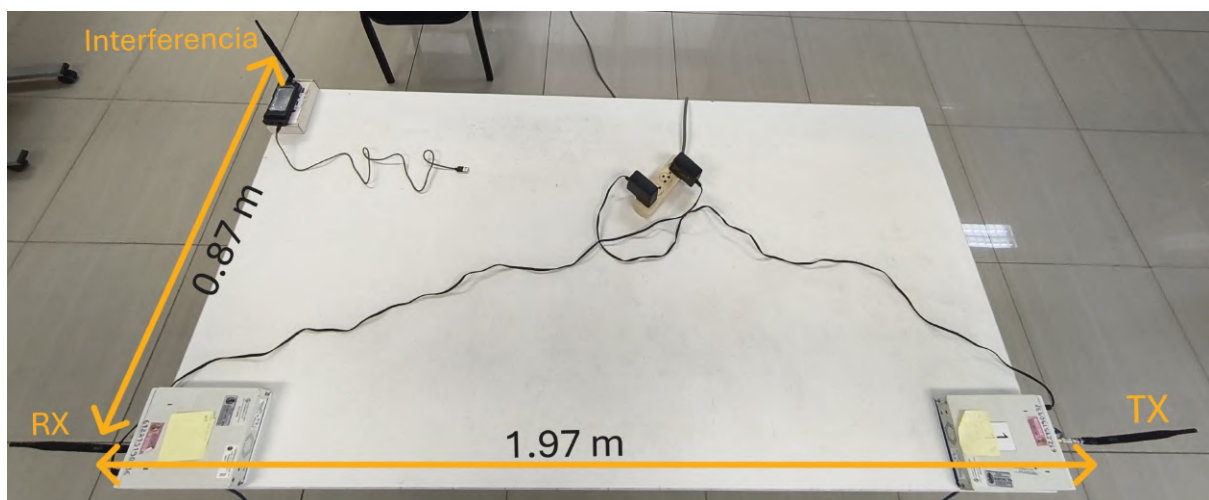


Figura 14
Implementación de Hardware

3.5 Bloques GNU Radio

Esta sección describe el enlace LoRa-SDR base usado como línea de referencia (sin sensado ni decisiones). El diagrama se organiza en cinco partes: la configuración global, la cadena de transmisión, la cadena de recepción, la monitorización y registro de métricas, y visualización.

3.5.1 Configuración y variables globales

La figura 15 muestra cuatro bloques de variables definidos para un flujo de transmisión SDR en GNU Radio, donde cada bloque especifica un parámetro esencial de operación: la tasa de muestreo (*samp_rate*) establecida en 250 kS/s, el ancho de banda (*bw*) de 125 kHz, la frecuencia central (*f_central*) fijada en 915.6 MHz, y el factor de ensanchamiento (*sf*) igual a 7, todos ellos parámetros clave que controlan el comportamiento de la modulación LoRa y la operación de los equipos SDR, permitiendo flexibilidad en los experimentos y una configuración precisa adaptada a las necesidades de la red y del prototipo desarrollado en la tesis.



Figura 15
Variables Globales

3.5.2 Transmisor LoRa SDR

El diagrama representa la cadena de transmisión implementada en GNU Radio para un sistema LoRa sobre SDR, donde el bloque Message Strobe genera mensajes periódicos simulando el tráfico típico en redes de sensores inalámbricos, el bloque LoRa Tx modula estos mensajes usando parámetros adecuados para aplicaciones de largo alcance y bajo consumo (como tasa de muestreo, ancho de banda, factor de expansión y codificación), y finalmente el bloque UHD: USRP Sink transmite la señal modulada empleando un hardware SDR configurado en frecuencia y ganancia específicos, permitiendo la evaluación experimental de tecnologías LoRa

en el contexto de radio cognitivo mediante pruebas controladas y replicables en entornos de laboratorio.

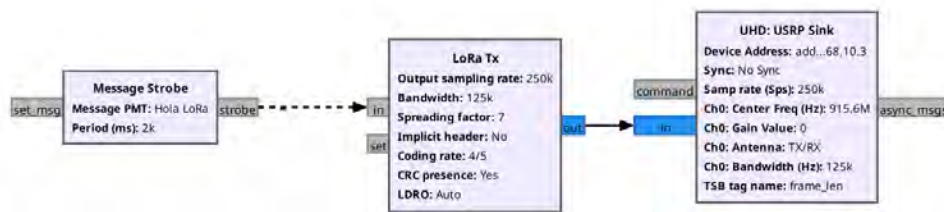


Figura 16
Transmisor LoRa

- **Message Strobe** Este bloque tiene la función de generar mensajes a intervalos regulares definidos por el usuario. En este caso, el mensaje es el string “Hola LoRa” y el período es de 2 segundos (2000 ms).
- **Modulador – LoRa Tx** Este bloque corresponde al transmisor digital compatible con el estándar LoRa, encargado de modular los mensajes que llegan desde el bloque anterior.
- **UHD: USRP Sink** Este bloque es responsable de la transmisión física de la señal modulada usando el hardware USRP, el cual actúa como el transceptor SDR.

3.5.3 Receptor LoRa SDR

El diagrama representa la cadena de recepción implementada en GNU Radio para un sistema LoRa sobre SDR, donde el bloque UHD: USRP Source sintoniza la banda ISM, posteriormente, el bloque LoRa Rx detecta el preámbulo, sincroniza los chirps y demodula la modulación CSS finalmente, las salidas de payload y SNR posibilitan el registro, la evaluación de desempeño (BER/PER) y la reconfiguración dinámica mediante su puerto snr_out, habilitando experimentos reproducibles de radio cognitivo en laboratorio bajo variaciones controladas de canal y parámetros del receptor.

- **UHD: USRP Source:** Captura la señal RF desde la antena y la entrega como I/Q a GNU Radio. Parámetros usados: frecuencia 915.6 MHz, tasa de muestreo 250 kS/s, BW

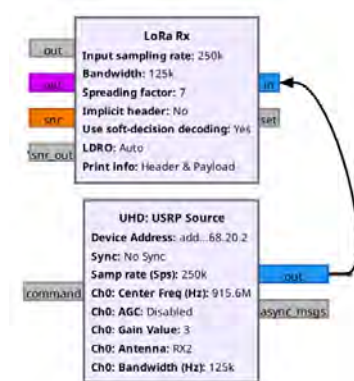


Figura 17
Receptor LoRa

analógico 125 kHz, ganancia fija 3 (AGC off), antena RX2. Sirve como front-end SDR del receptor.

- LoRa Rx: Demodula y decodifica la señal LoRa: detecta preámbulo, sincroniza chirps, aplica CSS con BW=125 kHz y SF=7, usa encabezado explícito y decodificación soft-decision. Salidas: PDUs (payloads) y métrica de SNR; puerto set para reconfiguración en tiempo real.

3.5.4 Cálculo SNR

El diagrama extrae la SNR estimada por el receptor LoRa y la guarda en disco para su análisis estadístico posterior.

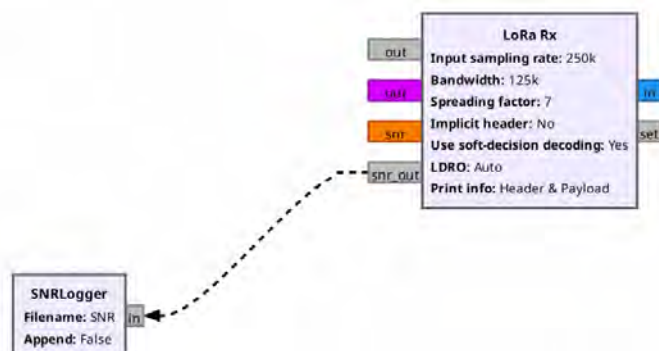


Figura 18
Estimación SNR

Bloque SNRLogger

Registra SNR por paquete, aplica un suavizado temporal y almacena trazas (con SF y marcas de éxito/fracaso). Provee insumos estables para la decisión y para el análisis posterior.

Arquitectura funcional

1. **Extracción:** Lee SNR y metadatos de cada paquete recibido.
2. **Suavizado:** Calcula el promedio móvil exponencial de SNR EMA-SNR con una constante de tiempo configurable.
3. **Registro:** Guarda los valores para el análisis posterior.

Estimación de SNR en gr-lora_sdr

El demodulador de `gr-lora_sdr` estima el SNR *por símbolo* después del *dechirp* y una FFT de tamaño $N = 2^{\text{SF}}$. En ese dominio, la señal útil se concentra en un bin (el “tono” del símbolo) y el ruido queda distribuido en el resto de bins. El SNR se calcula como la razón entre la potencia media del pico (señal) y la potencia media del fondo espectral (ruido).

Pasos para estimación de SNR. Para cada símbolo recibido $x[n]$, $n = 0, \dots, N - 1$:

1. **Dechirp:** se multiplica por el chirp conjugado ideal $c^*[n]$:

$$z[n] = x[n] \cdot c^*[n].$$

2. **FFT por símbolo:**

$$Y[k] = \text{FFT}\{z[n]\}, \quad k = 0, \dots, N - 1.$$

3. **Bin útil:** se identifica el tono como el índice de máxima energía:

$$\hat{k} = \arg \max_k |Y[k]|^2.$$

4. **Potencia de señal (pico + vecinos):** para capturar fuga espectral por pequeños desajustes de frecuencia/tiempo (CFO/STO), se promedia la energía del pico y sus vecinos inmediatos:

$$\mathcal{S} = \{\hat{k} - 1, \hat{k}, \hat{k} + 1\} \pmod{N}, \quad P_s = \frac{1}{|\mathcal{S}|} \sum_{k \in \mathcal{S}} |Y[k]|^2.$$

5. **Potencia de ruido (fondo espectral):** se excluye $\mathcal{S}$ y se promedia el resto de bins:

$$P_n = \frac{1}{N - |\mathcal{S}|} \sum_{k \notin \mathcal{S}} |Y[k]|^2.$$

6. **SNR del símbolo:**

$$\text{SNR} = \frac{P_s}{P_n}, \quad \text{SNR}_{\text{dB}} = 10 \log_{10} \left(\frac{P_s}{P_n} \right).$$

Tras el dechirp, un símbolo LoRa ideal concentra casi toda su energía en un bin de la FFT; el ruido térmico (AWGN) se reparte aproximadamente uniforme en los demás. Así, el cociente entre “pico” y “fondo” aproxima bien el SNR efectivo de ese símbolo. Incluir vecinos ± 1 evita subestimar la señal cuando hay leve CFO/STO o ventana rectangular (sin windowing adicional).

El valor anterior es *por símbolo*. En la práctica se reporta el SNR de la trama como media o mediana de los símbolos útiles; también puede aplicarse un suavizado exponencial EMA para estabilizar lecturas en tiempo real. Si aparece interferencia estrecha en otros bins, ésta eleva P_n y el SNR estimado baja, reflejando correctamente la degradación del enlace.

3.5.5 Cálculo RSSI

La figura 19 muestra el diagrama de flujo diseñado para la estimación del RSSI (Received Signal Strength Indicator) en GNU Radio utilizando SDR y hardware USRP. El proceso comienza con el bloque UHD: USRP Source, que recibe la señal de RF, configurándose con parámetros como tasa de muestreo (250 kS/s), frecuencia central (915,6 MHz), ganancia (3 dB) y ancho de banda (125 kHz). La señal recibida pasa por un filtro pasa bajos que reduce el

ruido fuera de banda y se adapta al ancho de banda útil; luego, la señal compleja se convierte a magnitud al cuadrado para calcular la potencia instantánea. Posteriormente, mediante un promedio móvil de ventana definida, se suavizan variaciones rápidas, tras lo cual el bloque Log10 aplica la escala logarítmica para obtener el nivel en decibelios (dB). Los datos resultantes se muestran visualmente con un bloque QT GUI Number Sink para monitoreo en tiempo real y se almacenan en un archivo CSV utilizando el bloque RSSItoCSV, facilitando el análisis posterior. La cadena implementa un método robusto para la estimación y registro automatizado de la intensidad de señal recibida en sistemas experimentales basados en SDR, lo que resulta esencial para caracterización de canales y calibración en ensayos de redes inalámbricas.

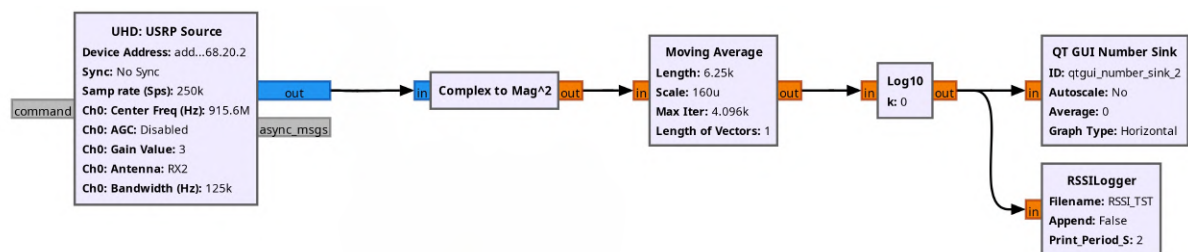


Figura 19
Estimación RSSI

- Complex to Mag² obtiene potencia instantánea.
- Moving Average integra la potencia para estabilizar lecturas.
- Log10 convierte a escala dB.
- RSSItoCSV guarda los valores de rssi en un archivo.

Bloque RSSIToCSV

Calcula el RSSI medio en ventanas temporales y lo exporta a un archivo CSV. Permite observar tendencias de potencia recibida y compararlas con SNR y eventos de resintonización.

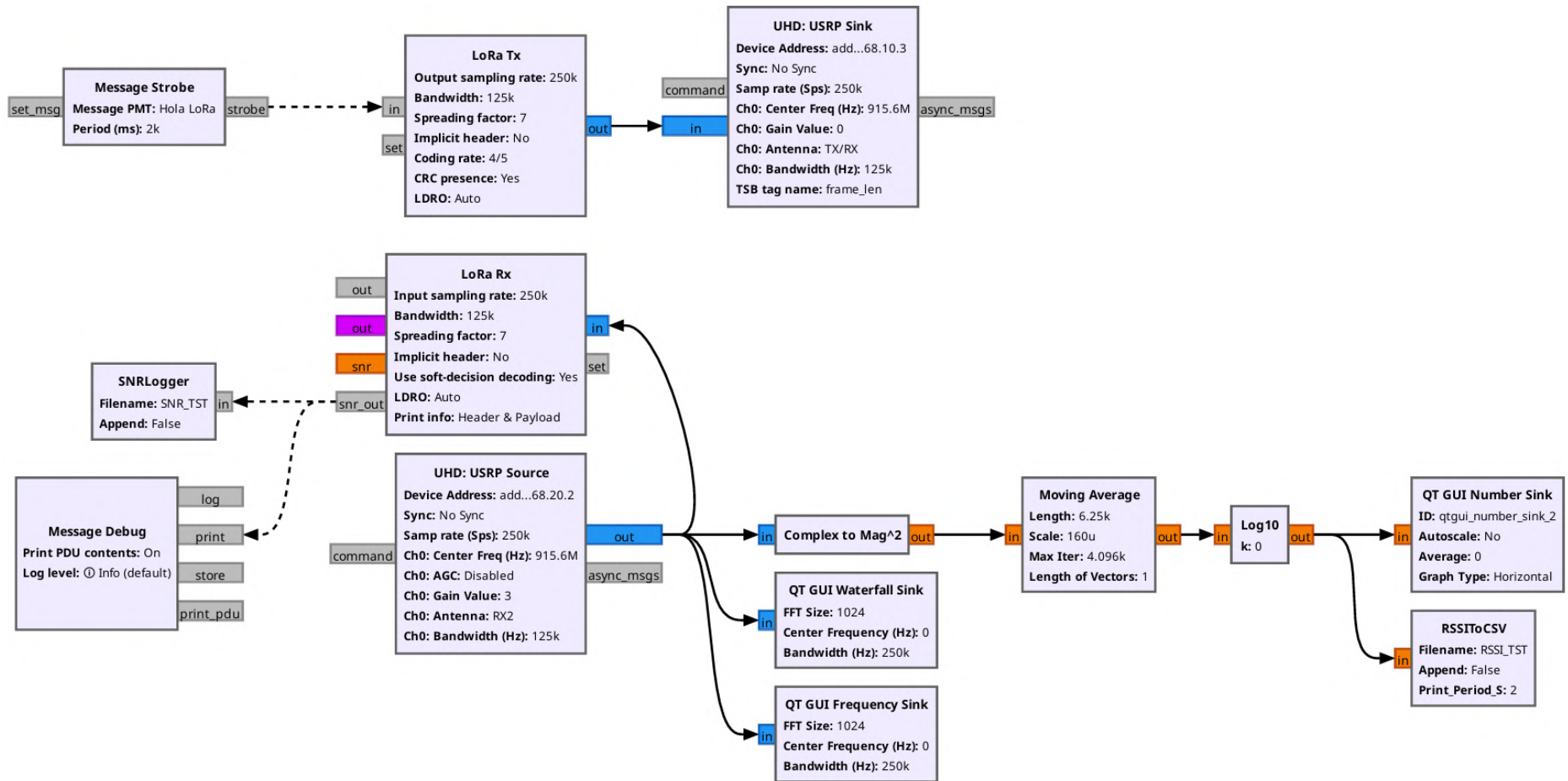


Figura 20
Sistema LoRa (baseline)

3.6 Diseño prototipo de radio cognitivo

El prototipo implementado aplica el ciclo cognitivo sentir, decidir y actuar sobre dos parámetros del enlace LoRa que pueden modificarse en tiempo de ejecución: la frecuencia central del canal y el factor de ensanchamiento (SF). Cada uno de estos parámetros tiene su propio bloque de decisión, y ambos actúan sobre el mismo transceptor físico. El sistema también incorpora un escáner espectral que monitorea continuamente la banda de 915 MHz para identificar canales con interferencia antes de ordenar un cambio. Esta arquitectura se refleja directamente en el diagrama de flujo de GNU Radio de la figura 26, donde se pueden identificar cinco bloques cognitivos desarrollados para esta tesis, más la conexión al hardware USRP que ejecuta las acciones. A continuación se describe cada bloque junto con los valores de configuración utilizados durante las pruebas experimentales.

3.6.1 Bloque *ED_CFAR_Scanner*

Este bloque implementa la fase *sentir* del ciclo cognitivo. Su propósito es analizar el espectro de la banda ISM de 915 MHz y determinar qué frecuencias se encuentran ocupadas por señales interferentes. Para lograrlo, opera en tres etapas secuenciales: estimación de la densidad espectral de potencia (PSD) mediante el método de Welch, aplicación de un detector CA-CFAR sobre la PSD.

Estimación de la PSD — Resolución frecuencial

La señal I/Q del USRP receptor llega a una tasa de muestreo $f_s = 250$ KSps. Cada bloque de procesamiento utiliza $nfft = 4096$ puntos de FFT. Lo que resulta en una resolución frecuencial de cada bin:

$$\Delta f = \frac{f_s}{nfft} = \frac{250000}{4096} \approx 61.0 \text{ Hz/bin} \quad (3.1)$$

Un canal LoRa de $BW = 125$ kHz queda representado por $125\,000/61 \approx 2049$ bins, lo que permite estimar con precisión el espectro de la señal. Una señal CW produce un pico perfectamente localizado en el espectro, distinto del espectro distribuido de la señal LoRa. Esta distinción espectral es la base física que hace posible la detección de interferencia de onda continua.

Método de Welch — Número efectivo de promedios

Antes de la FFT, cada bloque de 4096 muestras es multiplicado por una ventana de Hann, lo que reduce la fuga espectral (*spectral leakage*). Sin el uso de una ventana Hann, los lóbulos secundarios de una señal CW de gran amplitud contaminarían bins adyacentes, pudiendo enmascarar interferentes débiles o generar falsas alarmas en canales vecinos. La PSD se estima mediante el método de Welch (Welch, 1967), que promedia varios periodogramas calculados sobre ventanas solapadas al 50 %. El parámetro `sense_ms = 80` define la ventana de captura disponible por ciclo de sensado, lo que equivale a:

$$N_{\text{muestras}} = \frac{\text{sense_ms } f_s}{1000} = \frac{80 \times 250000}{1000} = 20000 \text{ muestras} \quad (3.2)$$

Con una longitud de salto de

$$\text{hop} = \frac{\text{nfft}}{2} = 2048 \text{ muestras} \quad (3.3)$$

correspondiente al 50 % de solapamiento, el número de periodogramas que caben dentro de las 20 000 muestras es:

$$N_{\text{frames}} = 1 + \left\lfloor \frac{N_{\text{muestras}} - \text{nfft}}{\text{hop}} \right\rfloor = 1 + \left\lfloor \frac{20000 - 4096}{2048} \right\rfloor = 1 + 7 = 8 \quad (3.4)$$

Aunque el parámetro `navg = 12` permite hasta 12 promedios, el valor efectivo está limitado por `sense_ms = 80`, resultando en 8 *frames* promediados. La reducción de varianza del estimador espectral que esto produce, conforme al resultado clásico del método de Welch (Stoica, 2005), es:

$$\text{Var}(\text{PSD}_{\text{Welch}}) = \frac{\text{Var}(\text{PSD}_{\text{periodograma}})}{N_{\text{frames}}} = \frac{\text{Var}}{8} \quad (3.5)$$

lo que implica una reducción de la desviación estándar del piso de ruido en un factor de

$$\sqrt{8} \approx 2.83 \quad (3.6)$$

Este resultado implica que la estimación del nivel de ruido local es 2.83 veces más estable

que la de un periodograma simple, lo que se traduce directamente en una reducción de la tasa de falsas alarmas del detector CFAR.

Detector CA-CFAR

La PSD estimada es procesada por un detector de Tasa Constante de Falsas Alarmas por Promediado de Celdas (CA-CFAR) (Richards, 2014). El umbral de detección se calcula de forma adaptativa para cada Celda Bajo Prueba (CUT), evitando que las variaciones del piso de ruido generen falsas alarmas cuando el nivel de ruido de fondo fluctúa.

El algoritmo define alrededor de cada CUT:

- `guard_cells` = 4 celdas de guarda a cada lado, que excluyen la energía de la señal objetivo del promedio de ruido.
- `train_cells` = 64 celdas de entrenamiento a cada lado, cuyo promedio estima el piso de ruido local.

El número total de celdas de referencia es:

$$N = 2 \times 64 = 128 \quad (3.7)$$

El umbral adaptativo en escala de potencia se define como:

$$T_{\text{umbral}} = \beta \mu_{\text{ruido}} \quad (3.8)$$

donde μ_{ruido} es la potencia media de las 128 celdas de entrenamiento y

$$\beta = P_{\text{fa}}^{-1/N} \quad (3.9)$$

es el factor multiplicador que garantiza la probabilidad de falsa alarma objetivo $p_{\text{fa}} = 0.05$ (5 %). Sustituyendo:

$$\beta = 0.05^{-1/128} = \exp\left(\frac{\ln(20)}{128}\right) = \exp(0.02340) \approx 1.0237 \quad (3.10)$$

En escala logarítmica, el umbral queda:

$$T_{\text{dB}} = \mu_{\text{ruido,dB}} + 10 \log_{10}(\beta) = \mu_{\text{ruido,dB}} + 10 \log_{10}(1.0237) \approx \mu_{\text{ruido,dB}} + 0.10 \text{ dB} \quad (3.11)$$

El umbral se sitúa apenas 0.10 por encima del nivel de ruido estimado. Este valor es matemáticamente correcto para $P_{\text{fa}} = 5\%$ con $N = 128$ celdas. Esto nos da una estimación muy precisa del piso de ruido, la señal solo necesita superar ese piso para ser clasificada como interferencia. Este valor es aceptable para un sistema cognitivo donde una falsa alarma produce un cambio de canal innecesario, mientras que una detección fallida puede provocar la pérdida sostenida de paquetes (Axell, 2012).

La elección de `guard_cells = 4` se basa en el perfil espectral de la señal CW tras la ventana de Hann: el lóbulo principal de la transformada de Hann tiene un ancho de aproximadamente 4 bins a -6 dB (Harris, 1978). Con 4 celdas de guarda a cada lado, la zona de exclusión cubre $4 \times 61 \text{ Hz} \approx 244 \text{ Hz}$ alrededor de la CUT, protegiendo completamente el promedio de ruido de la energía del tono interferente.

La elección de `train_cells = 64` garantiza que la estimación del ruido se realice sobre una ventana espectral de $64 \times 61 \text{ Hz} \approx 3.9$ a cada lado de la CUT, suficientemente amplia para capturar las estadísticas del ruido local y suficientemente estrecha para no solaparse con canales LoRa vecinos, espaciados en al menos 200 kHz.

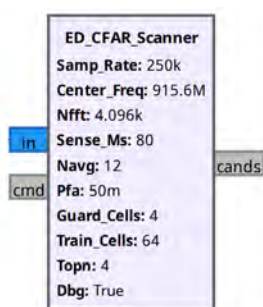


Figura 21
Bloque *ED_CFAR_Scanner*

3.6.2 Bloque *SliceStepper*

El bloque implementa el barrido escalonado del espectro que permite al *ED_CFAR_Scanner* construir un mapa de ocupación de toda la subbanda ISM, en lugar de monitorear una sola

frecuencia fija. Desplaza periódicamente la frecuencia central de escucha a través de posiciones predefinidas, manteniendo al receptor en cada posición el tiempo necesario para completar un ciclo de sensado del escáner en las siguientes frecuencias:

$$915.2 / 915.4 / 915.6 / 915.8 / 916.0 \text{ MHz} \quad (3.12)$$

Estas frecuencias coinciden exactamente con el parámetro `grid_step = 200` kHz cada posición de barrido del `SliceStepper` cubre un canal candidato de decisión, de modo que el mapa de ocupación espectral generado es directamente utilizable por el bloque de decisión sin transformación adicional. El canal de frecuencia inicial del sistema (915.6 MHz) forma parte de las frecuencias, lo que garantiza que el propio canal activo sea evaluado.

En cada posición de barrido, el receptor permanece el tiempo suficiente para que el `ED_CFAR_Scanner` complete un ciclo de sensado: `sense_ms = 80` ms de captura más el tiempo de procesamiento.



Figura 22
Bloque Slice_Stepper

3.6.3 Bloque *ChannelPickerAndTune_SNR*

Este bloque implementa la fase *decidir*. Combina la lista de frecuencias interferidas proveniente del `ED_CFAR_Scanner` y la SNR suavizada del enlace LoRa. Cuando detecta degradación sostenida del enlace, selecciona el canal alternativo más favorable dentro de la lista de frecuencias de la banda ISM y ordena la resintonización del hardware.

Tabla de SNR mínimos operativos

El bloque utiliza internamente una tabla de SNR mínimos operativos para cada SF:

Estos valores difieren de los umbrales teóricos de Semtech de la tabla 1 (-7.5 dB para SF7, -10.0 dB para SF8, etc.) en una diferencia constante de -4.5 dB para todos los SF. Esta dife-

Tabla 3
SNR mínimos operativos por Spreading Factor

SF	SNR _{mín} (dB)
SF7	-12.0
SF8	-15.0
SF9	-18.0
SF10	-20.0
SF11	-23.0
SF12	-27.0

rencia refleja la ganancia adicional que aporta la decodificación *soft-decision* implementada en `gr-lora_sdr` (Sección 3.5.3: “Use soft-decision decoding: Yes”). La decodificación de decisión suave proporciona una ganancia de codificación de 2 dB a 3 dB respecto a la decisión dura en sistemas con codificación FEC, y el resto del margen proviene de las características específicas de la implementación.

Suavizado de la SNR mediante EMA

La SNR estimada por el receptor LoRa fluctúa paquete a paquete incluso en canal estable, por lo que se filtra mediante un Promedio Móvil Exponencial (EMA) de la forma:

$$\text{SNR}_{\text{ema}}[k] = a \text{SNR}_{\text{ema}}[k - 1] + (1 - a) \text{SNR}[k] \quad (3.13)$$

donde

$$a = \exp\left(-\frac{\Delta t}{\tau_{\text{EMA}}}\right) \quad (3.14)$$

Con `ema_tau_ms = 300` ms y el período entre paquetes $\Delta t = 2000$ ms (configurado en *Message Strobe*, Sección 3.5.2), se obtiene:

$$a = \exp\left(-\frac{2000}{300}\right) = \exp(-6.67) \approx 0.00127 \quad (3.15)$$

por lo que

$$\text{SNR}_{\text{ema}}[k] \approx 0.001 \text{SNR}_{\text{ema}}[k - 1] + 0.999 \text{SNR}[k] \approx \text{SNR}[k] \quad (3.16)$$

Dado que $\Delta t \gg \tau_{\text{EMA}}$, la constante de tiempo del filtro es mucho menor que el intervalo

entre muestras. En la práctica, el EMA actúa como un filtro de valores no deseados, si un paquete llega con una SNR anómala por un error de sincronización transitorio, el EMA la absorbe sin afectar la decisión; pero si la degradación es real y persiste durante múltiples paquetes, sera considerado por el contador `snr_need_consec`.

El umbral de SNR que activa la búsqueda de un canal alternativo se calcula como:

$$\gamma_{\text{canal}}(\text{SF}) = \text{SNR}_{\text{mín}}(\text{SF}) + \text{snr_margin_db} \quad (3.17)$$

Con `snr_margin_db` = 3 dB y $\text{SNR}_{\text{mín}}(\text{SF7}) = -12.0\text{dB}$:

$$\gamma_{\text{canal}}(\text{SF7}) = -12.0 + 3.0 = -9.0\text{dB} \quad (3.18)$$

El margen de 3 dB tiene la función de anticipar la acción cognitiva antes de que el enlace alcance su límite real de operabilidad, dejando tiempo para que el sistema detecte la degradación, confirme su persistencia y ejecute la resintonización para evitar pérdidas masivas de paquetes. Esta anticipación es consistente con los datos de la Tabla 6 donde el sistema realizó la resintonización cuando el valor de SNR se aproxima a -9 dB para SF7.

Para evitar resintonizaciones por fluctuaciones momentáneas, el bloque exige que el SNR suavizado sea menor que el umbral SNR

$$\text{SNR}_{\text{ema}} < \gamma_{\text{canal}}(\text{SF}) \quad (3.19)$$

durante `snr_need_consec` = 4 paquetes consecutivos. Con $\Delta t = 2\text{ s}$:

$$t_{\text{confirmacin}} = \text{snr_need_consec} \times \Delta t = 4 \times 2\text{ s} = 8\text{ s} \quad (3.20)$$

Esta ventana de 8 s de degradación sostenida filtra desvanecimientos rápidos y errores de estimación de SNR que podrían durar uno o dos intervalos sin representar un problema estructural del canal.

El parámetro `dwell_ms` = 800 ms define el tiempo mínimo de permanencia en el canal actual antes de volver a evaluar la necesidad de cambio. Como `dwell_ms` = 800 ms < Δt =

2000 ms, el *dwell* expira antes de que llegue el siguiente paquete, lo que equivale a habilitar la evaluación paquete a paquete. El parámetro `post_ms = 800` ms, igual a `dwell_ms` por diseño del código (`self.post_ms = int(dwell_ms)`), bloquea nuevas decisiones durante 800 ms tras un cambio, protegiendo al sistema durante el transitorio inmediato.

Selección del canal alternativo

Los canales candidatos se distribuyen desde `band_min= 915.0` MHz hasta `band_max = 916.0` MHz con una separación de `grid_step = 200` kHz, dando seis posiciones: 915.0, 915.2, 915.4, 915.6, 915.8 y 916.0 MHz. Esta separación de 200 kHz coincide con el espaciado de canal estándar de LoRaWAN para la Región 2, garantizando que ningún par de canales se superponga espectralmente con $BW = 125$ kHz.

Adicionalmente, `MIN_HOP_HZ = 150` kHz previene micro-saltos: el sistema no cambia al canal actual ni a frecuencias a menos de 150 kHz del canal actual, lo que, combinado con la separación de 200 kHz, hace que el salto mínimo sea siempre de un paso completo.

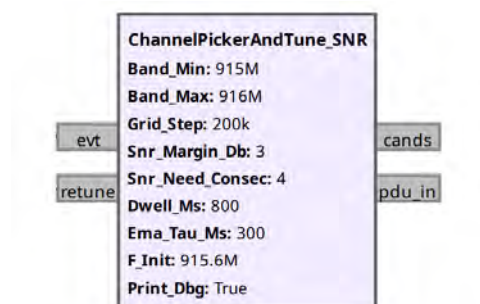


Figura 23

Bloque `ChannelPickerAndTune_SNR`

3.6.4 Bloque `SfDeciderMin`

`SfDeciderMin` implementa la fase decidir, actúa de forma complementaria al bloque `ChannelPickerAndTune_SNR`. Este bloque responde a degradación generalizada del canal, como ruido de banda ancha o interferencia persistente en los canales disponibles aumentando la robustez de la modulación. El bloque prioriza la protección del enlace sobre la recuperación de la tasa de datos.

El bloque `SfDeciderMin` utiliza la misma tabla que `ChannelPickerAndTune_SNR` ver tabla 3. Esta coherencia es importante, si ambos bloques evalúan la degradación mediante

tablas distintas, podrían generar decisiones contradictorias ante una misma condición de canal. A partir de esta tabla se definen dos umbrales para el cambio de SF.

Umbral de deterioro El umbral que propone incrementar el SF hacia $SF + 1$ se define como:

$$\gamma_{\text{malo}}(\text{SF}) = \text{SNR}_{\text{mín}}(\text{SF}) + \text{snr_margin} = \text{SNR}_{\text{mín}}(\text{SF}) + 4.0 \text{ dB} \quad (3.21)$$

Por ejemplo:

$$\gamma_{\text{malo}}(\text{SF7}) = -12.0 + 4.0 = -8.0 \text{ dB} \quad (3.22)$$

$$\gamma_{\text{malo}}(\text{SF8}) = -15.0 + 4.0 = -11.0 \text{ dB} \quad (3.23)$$

Umbral de mejora El umbral que propone disminuir el SF hacia $SF - 1$ se evalúa respecto al SF objetivo:

$$\gamma_{\text{bueno}}(\text{SF}) = \text{SNR}_{\text{mín}}(\text{SF} - 1) + \text{snr_margin} + \text{good_margin} \quad (3.24)$$

Con $\text{snr_margin} = 4 \text{ dB}$ y $\text{good_margin} = 2 \text{ dB}$, cuando el sistema se encuentra en SF8 y evalúa retornar a SF7:

$$\gamma_{\text{bueno}} = -12.0 + 4.0 + 2.0 = -6.0 \text{ dB} \quad (3.25)$$

La zona de histéresis entre ambos umbrales resulta:

$$\gamma_{\text{bueno}} - \gamma_{\text{malo}} = -6.0 - (-8.0) = 2.0 \text{ dB} \quad (3.26)$$

Esta zona de histéresis previene el fenómeno de oscilación (*flapping*), en el cual el sistema alternaría indefinidamente entre SF7 y SF8 bajo condiciones de SNR marginal.

La validez del umbral γ_{malo} puede verificarse con los datos de la Tabla 7 (Sección 4.3). El sistema incrementó el SF de 7 a 8 en el bloque 9 de la prueba AWGN, donde la SNR registrada fue de -9.32 dB . En el bloque anterior (bloque 8), la SNR era de -7.58 dB , valor que aún supera $\gamma_{\text{malo}}(\text{SF7}) = -8.0 \text{ dB}$. Esto indica que el cambio de SF se ejecutó exactamente cuando la SNR descendió por debajo del umbral de -8.0 dB .

La asimetría $\text{need_bad} > \text{need_good}$ responde a criterios operativos. Para la subida de SF se exige mayor tiempo de confirmación debido a que la degradación en una red real puede ser transitoria; actuar prematuramente incrementa el tiempo en aire (ToA). En cambio, para la reducción de SF, el tiempo de confirmación es menor porque recuperar un SF inferior cuando el canal mejora permite aumentar la tasa de datos y liberar tiempo de canal para otros nodos.

Esta decisión es consistente con los principios de adaptación de modulación descritos en la literatura clásica de comunicaciones adaptativas descrito en Goldsmith y Chua (1998)

El parámetro $\text{dwell} = 1200$ ms define el período de estabilización tras un cambio de SF:

$$1200 \text{ ms} < 2000 \text{ ms} \quad (3.27)$$

el dwell expira antes de la llegada del siguiente paquete, permitiendo evaluar la condición del canal en cada paquete recibido una vez transcurrido el tiempo de asentamiento.

EMA de SNR en SfDeciderMin La constante de tiempo del EMA es $\tau = 300$ ms, obteniéndose el mismo comportamiento que en `ChannelPickerAndTune_SNR`. Con $\Delta t = 2000$ ms, el coeficiente de decaimiento resulta:

$$a = \exp\left(-\frac{2000}{300}\right) \approx 0.001 \quad (3.28)$$

Por tanto, cada nuevo valor de SNR prácticamente reemplaza al anterior. En este régimen, el EMA actúa principalmente como protección frente a errores aislados de sincronización y no como un suavizador de largo plazo.

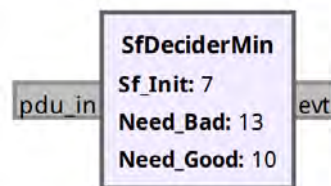


Figura 24
Bloque SfDeciderMin

3.6.5 Bloque *QtRangeSF* (ajuste de SF)

QtRangeSF actúa como punto único de sincronización del valor de SF en el *flowgraph*. Recibe mensajes del tipo $\{ 'sf' : N \}$ por su puerto de entrada (generados por el *SfDeciderMin* cuando decide un cambio) y los propaga a todos los objetos del *flowgraph* que implementen el método `set_sf()`.

La relevancia de este bloque es operativa: si el cambio de SF se enviara directamente al transmisor y al receptor por separado, podría producirse una desincronización temporal en la que uno de los dos extremos del enlace utilizara un SF distinto. *QtRangeSF* evita este riesgo al ser el punto de despacho único del nuevo valor.

El rango $sf_min = 7 / sf_max = 12 / sf_init = 7$ está definido en el *SfDeciderMin*, que es quien restringe los valores posibles; *QtRangeSF* no filtra el valor recibido, por lo que la coherencia del espacio de estados depende íntegramente del bloque generador.



Figura 25
Bloque QTRangeSF

3.6.6 Aplicación de la resintonización (control de UHD desde el flowgraph)

Cuando el *ChannelPickerAndTune_SNR* decide cambiar de canal, genera un mensaje de control con la nueva frecuencia central f_c y lo publica por su puerto `evt`. El bloque raíz del *flowgraph* (`top_block`) recibe este mensaje y lo entrega al parámetro `command` de los bloques UHD Source (Gateway, USRP 192.168.20.2) y UHD Sink (Nodo, USRP 192.168.10.3), actualizando ambos USRP de forma coordinada. La actualización simultánea de ambos extremos del enlace es indispensable: un cambio en solo uno de ellos resultaría en un enlace roto.

Tras el cambio de frecuencia en el hardware, el *flowgraph* espera un tiempo de asentamiento de `settling_ms = 500 ms` antes de reanudar la demodulación y el registro de métricas. Este retardo protege de dos fenómenos: el transitorio del oscilador local del USRP 2920 al bloquearse en la nueva frecuencia mediante su PLL (National Instruments, 2014), y la necesidad del bloque

Frame_sync de re-adquirir la sincronización de trama en el nuevo canal.

Los parámetros $\text{dwell_ms} = 800$ ms y $\text{post_ms} = 800$ ms de la sección 3.6.3 impiden que el sistema evalúe un nuevo cambio durante el período de asentamiento, aunque en la práctica ambos valores son menores que el período de transmisión de 2 s, de modo que la primera evaluación posterior al cambio ocurre con el primer paquete recibido en el nuevo canal, después de que el asentamiento haya concluido.

Secuencia de ejecución

1. ChannelPickerAndTune_SNR detecta degradación sostenida (paquetes consecutivos con $\text{SNR} < \gamma_{\text{canal}}$) y selecciona f_c alternativa de la lista de frecuencias.
2. Publica {"freq": f_c } por el puerto evt.
3. top_block actualiza center_freq en UHD Sink y UHD Source de forma coordinada.
4. Se espera $\text{settling_ms} = 500$ ms.
5. Se reanuda la recepción; el primer paquete recibido en el nuevo canal alimenta el EMA de SNR y reinicia los contadores del ChannelPickerAndTune_SNR.

Si la degradación persiste tras el cambio de canal (condición característica de ruido de banda ancha, analizada en Sección 4.3), el SfDeciderMin actúa de forma independiente: si $\text{SNR}_{\text{ema}} < \gamma_{\text{malo}}(\text{SF})$ durante paquetes consecutivos, incrementa el SF como segunda acción de robustez.

3.6.7 Interacción en el lazo cognitivo

Sentir: El SliceStepper desplaza periódicamente la frecuencia de escucha por las cinco posiciones de la lista de frecuencias (915 MHz a 916 MHz en pasos de 200 kHz). En cada posición, el ED_CFAR_Scanner captura 20 000 muestras (80 ms), promedia 8 periodogramas mediante el método de Welch, aplica el detector CA-CFAR ($N = 128$, $P_{fa} = 5\%$, $\beta \approx 1.024$) y envía al puerto cands la lista de frecuencias.

Medir: En paralelo, el bloque LoRa Rx entrega por su puerto `snr_out` la SNR estimada de cada paquete recibido (cada 2 s). El `SNRLogger` aplica el EMA ($\tau = 300$ ms, $\alpha \approx 0.001$ para $\Delta t = 2$ s) y registra la traza temporal con metadatos de SF y estado de CRC.

El `RSSIToCSV` registra la potencia total recibida, que permite distinguir si la degradación de SNR se debe a interferencia (RSSI sube) o a atenuación de propagación (RSSI baja).

Decidir: El `ChannelPickerAndTune_SNR` combina los candidatos del escáner con la SNR_{ema} . Si

$$SNR_{ema} < \gamma_{canal}(SF) \quad (3.29)$$

durante varios paquetes consecutivos y existe un canal a más de 200 kHz del interferente, ordena la resintonización. El `SfDeciderMin` opera en paralelo: si

$$SNR_{ema} < \gamma_{malo}(SF) \quad (3.30)$$

durante mayor tiempo o el cambio de canal no fue suficiente, propone incrementar SF; si

$$SNR_{ema} > \gamma_{bueno}(SF) \quad (3.31)$$

durante varios paquetes consecutivos, propone reducir SF. Los umbrales de ambos bloques son distintos (−9 dB para `ChannelPicker` frente a −8 dB para `SfDeciderMin` en SF7), lo que puede generar acciones combinadas o secuenciales según la evolución de la interferencia.

Actuar: El mecanismo de la sección 3.6.6 actualiza los bloques UHD del transmisor y el receptor, aplica 500 ms de asentamiento y reanuda la operación.

Si se ha decidido también un cambio de SF, `QtRangeSF` propaga el nuevo valor a la cadena completa. Tras cualquier acción de adaptación, los contadores de histéresis se reinician y el sistema retoma el ciclo de sensado desde el nuevo estado operativo.

Resumen: Las tablas 6 y 7 documentan los parámetros exactos que gobiernan este ciclo. El comportamiento observado en los experimentos (cambios de canal a $SNR \approx -9$ dB para SF7,

cambios de SF a $\text{SNR} \approx -8 \text{ dB}$) es directamente explicable a partir de los valores de la tabla sin necesidad de suposiciones adicionales, lo que confirma la coherencia interna entre el diseño del sistema y los resultados experimentales.

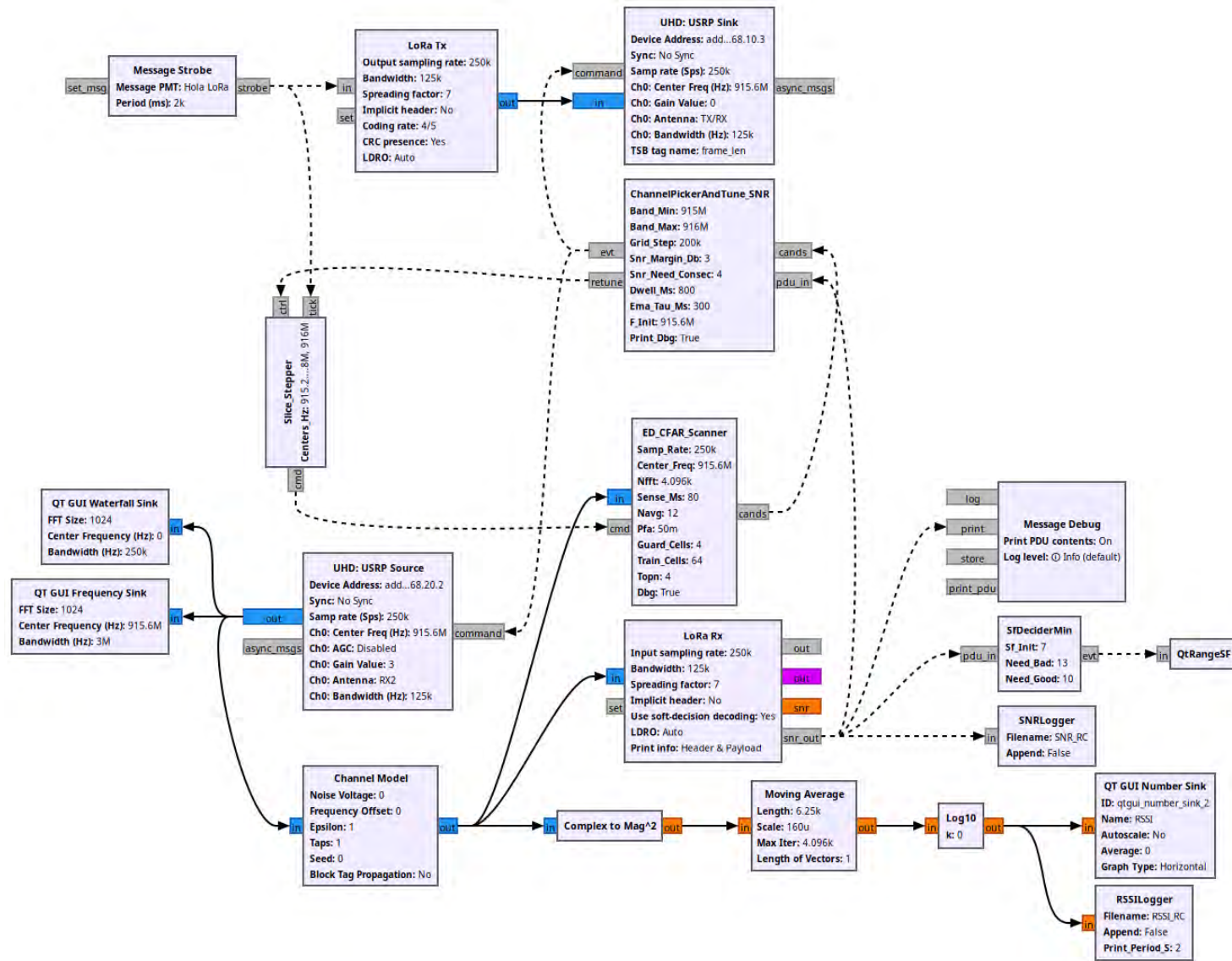


Figura 26
Sistema LoRa con Radio Cognitivo

CAPÍTULO IV

PRUEBAS Y RESULTADOS

En este capítulo se presenta la validación experimental del prototipo de radio cognitivo diseñado e implementado en el capítulo anterior. El objetivo principal de estas pruebas es “Evaluar el comportamiento del rendimiento de un prototipo de radio cognitivo basado en plataforma de radio definida por software (SDR) para aplicaciones de redes de sensores inalámbricos (WSN)”. Para todos los escenarios se adopta como *umbral de fallo operativo* el valor de $PER = 10\%$. La elección de este umbral se justifica en el comportamiento experimental observado: por encima de ese valor el sistema no cognitivo exhibe una degradación abrupta del enlace (Sección 4.1.2), y es consistente con los criterios de saturación de canal y escalabilidad reportados para LoRaWAN por Adelantado et al. (2017) y Bor et al. (2016).

4.1 Caracterización del Sistema LoRa Base

El propósito de este primer escenario de pruebas es establecer una línea base (baseline) de rendimiento. Para ello, se utiliza el “Sistema LoRa Base” (Figura 20), que opera como un transmisor LoRa estándar sin ninguna capacidad cognitiva. Se mide el rendimiento del sistema en dos condiciones: un canal limpio y un canal con interferencia controlada.

4.1.1 Rendimiento en Canal Limpio (Sin Interferencia)

En esta prueba inicial, el sistema se opera en el entorno de laboratorio sin activar el generador de interferencia. El objetivo es medir el rendimiento óptimo del enlace en las condiciones de laboratorio (con atenuación fija), caracterizando el piso de ruido y los valores de SNR y RSSI, según el esquema de la Figura 27.

El sistema transmitió paquetes de forma continua, registrándose las métricas en el receptor. Los resultados mostraron un enlace robusto y estable. Como se detalla en la tabla 4, la media de la SNR registrada fue de 8.373 dB. En estas condiciones, la Tasa de Paquetes Perdidos (PER) fue del 0 %, lo que indica que todos los paquetes enviados fueron recibidos y decodificados co-



Figura 27
Flujo de pruebas

rectamente, validando el funcionamiento del enlace base. Esto indica que el enlace decodifica correctamente la totalidad de las tramas transmitidas y valida el correcto funcionamiento del transceptor, incluso considerando las modificaciones realizadas al código fuente del módulo “gr-lora_sdr” descritas en la Sección 3.3.

Tabla 4
Resumen de SNR, RSSI, y PER por SF en canal limpio.

SF	Muestras	$\overline{\text{SNR}}$ (dB)	$\overline{\text{RSSI}}$ (dBm)	PER (%)
7	105	8.3356	-82.4130	0.0000
8	95	8.6648	-82.3026	0.0000
9	100	8.3192	-81.9351	0.0000
10	106	8.7135	-81.2925	0.0000
11	98	8.2485	-79.7089	0.0000
12	103	7.9625	-77.2005	0.0000

4.1.2 Rendimiento bajo interferencia controlada

Una vez caracterizado el rendimiento en canal limpio, se procedió a evaluar la degradación del sistema en presencia de interferencia co-canal. Para ello se empleó el generador ERASynth Micro como se aprecia en la figura 28, configurado para inyectar una señal *continuous wave*, CW en la misma frecuencia central del canal LoRa (915.6 MHz). La potencia de esta señal interferente se incrementó progresivamente en pasos de 3 dB, mientras se registraban el SNR, el RSSI y el PER para los factores de ensanchamiento SF7 a SF12. En la figura 29 se muestra el generador ERASynth Micro utilizado en las pruebas.

Se aplica el criterio de fallo definido al inicio del capítulo (PER >10 %); por encima de ese umbral el enlace se considera fuera de los márgenes de calidad de servicio (QoS) aceptables para aplicaciones WSN.

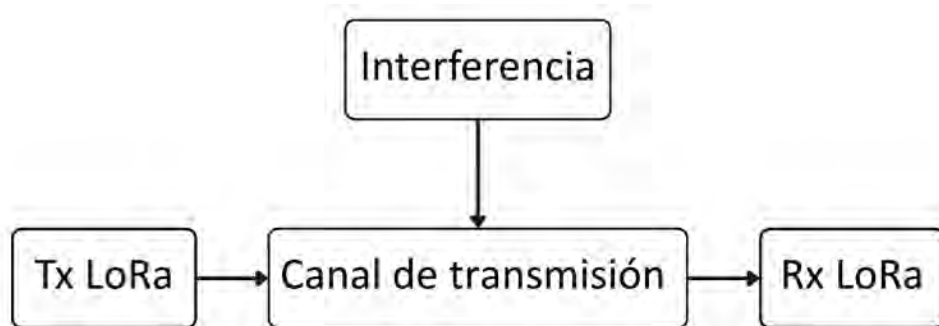


Figura 28
Flujo de pruebas con interferencia.



Figura 29
Generador ERASynth Micro configurado para producir interferencia CW co-canal a 915.6 MHz.

En esta tesis la SNR y el RSSI se definieron como variables de decisión del motor cognitivo, mientras que la PER se adoptó como criterio de validación del rendimiento (QoS). Esta separación responde a la naturaleza de cada métrica y no a una omisión. El RSSI mide la potencia total recibida, en tanto que la SNR cuantifica la calidad del enlace.

Por ello, la SNR y el RSSI se eligieron como entradas de decisión: son observables directos de la capa física, disponibles paquete a paquete, de bajo costo computacional, adecuados para decisiones en tiempo real en dispositivos WSN. De ahí que la arquitectura cognitiva decida sobre SNR/RSSI y se valide sobre PER: la PER es precisamente la magnitud que el lazo busca mantener por debajo del umbral de 10 %. Aunque las decisiones se basen en la SNR que mide constantemente la calidad del enlace.

La Tabla 5 resume los valores promedio de SNR, RSSI y PER obtenidos bajo interferencia para los factores de ensanchamiento SF7 a SF12. Para SF7 se observa que, con una potencia de

interferencia de -32 dBm, el SNR medio tiene un valor de -12.66 dB y el PER alcanza 16.67 %. Al incrementar la interferencia a -29 dBm, el SNR desciende hasta -14.68 dB y el PER se eleva a 75.00 %, lo que representa un enlace prácticamente inutilizable. Este mismo patrón se repite para los demás factores de ensanchamiento, evidenciando además una mayor tolerancia a la interferencia conforme aumenta el valor de SF.

Un patrón en la tabla 5 es el incremento de apenas 3 dB en la potencia de interferencia separa un aumento elevado del PER. Este comportamiento corresponde al efecto umbral característico de la modulación CSS, donde la demodulación funciona con alta probabilidad mientras el SNR efectivo supere el SNR mínimo del SF, y falla por debajo. Es relevante observar también que el SNR medido en el punto de fallo es similar para todos los SF (entre -12 dB y -15 dB aproximadamente); lo que cambia entre SF no es el SNR sino la potencia de interferencia tolerable antes de alcanzarlo, gracias a la ganancia de procesamiento aportada por cada incremento de SF.

Tabla 5
Resumen de SNR, RSSI y PER con interferencia por SF en el enlace LoRa no cognitivo.

SF	Interferencia (dBm)	$\overline{\text{SNR}}$ (dB)	$\overline{\text{RSSI}}$ (dBm)	PER (%)
7	-35	-10.0008	-61.6730	6.67
	-32	-12.6590	-58.9300	16.67
	-29	-14.6817	-57.1310	75.00
8	-32	-12.2010	-59.1500	5.00
	-29	-15.3217	-56.1690	13.33
	-26	-17.8104	-53.2600	78.94
9	-29	-15.2782	-56.0090	3.33
	-26	-18.1684	-53.3264	13.33
	-23	-20.9248	-50.2800	66.67
10	-26	-17.6070	-53.1990	5.00
	-23	-21.0006	-50.2325	8.33
	-20	-23.4119	-47.3523	66.67
11	-23	-21.6777	-50.4820	3.33
	-20	-23.7970	-48.8578	20.00

SF	Interf. (dBm)	$\overline{\text{SNR}}$ (dB)	$\overline{\text{RSSI}}$ (dBm)	PER (%)
	-17	-26.8155	-45.5630	56.67
12	-20	-24.3812	-48.1630	5.00
	-17	-27.2698	-44.7900	16.67
	-14	-29.2736	-41.6790	71.67

La figura 30 muestra la evolución del SNR en función de la potencia de interferencia para los factores de ensanchamiento SF7 a SF12. Se aprecia una degradación prácticamente lineal del SNR conforme aumenta la potencia de interferencia, con trazas casi paralelas y superpuestas para todos los SF. La ventaja del SF alto no radica, por tanto, en mejorar el SNR medido, sino en su capacidad de operar bajo condiciones más hostiles: como se evidencia en la Tabla 5, aunque los valores de SNR son similares entre SFs, el enlace con SF7 falla (PER >10 %) a partir de -29 dBm de interferencia, mientras que el SF12 logra decodificar exitosamente hasta los -14 dBm. La superposición casi total de las curvas para los diferentes SF en la figura 30 refleja que el SNR medido a nivel del receptor LoRa (tras dechirp y FFT) depende de la razón entre la potencia de la señal recibida y la potencia total de ruido más interferencia, y esta razón es independiente del SF utilizado. Esto confirma que el aumento del SF otorga una ganancia de procesamiento que permite mantener el enlace activo incluso a SNR bajos, validando su robustez en la práctica.

La Figura 31 muestra cómo responde el RSSI frente a la señal de interferencia. A diferencia de lo observado en el SNR, aquí se evidencia una progresión lineal creciente: el valor de RSSI medido aumenta de forma directamente proporcional a la potencia de interferencia inyectada. Al igual que en el caso anterior, las curvas de los distintos factores de ensanchamiento se encuentran superpuestas, lo que evidencia que la medición de energía (RSSI) es un proceso independiente del SF seleccionado. Mientras el SNR refleja la degradación de la calidad del enlace, el RSSI refleja el aumento de la potencia de interferencia.

La figura 32 muestra el comportamiento del enlace LoRa no cognitivo con SF7 cuando la potencia de interferencia se aproxima a -32 dBm, con un SNR medio de -12.66 dB. De igual manera, la figura 33 ilustra el comportamiento del enlace LoRa no cognitivo con SF12 bajo una interferencia de -17 dBm y un SNR más bajo, de -27.27 dB. En el panel izquierdo de ambas capturas se observa la salida de consola del receptor: los mensajes `frame_info` reportan

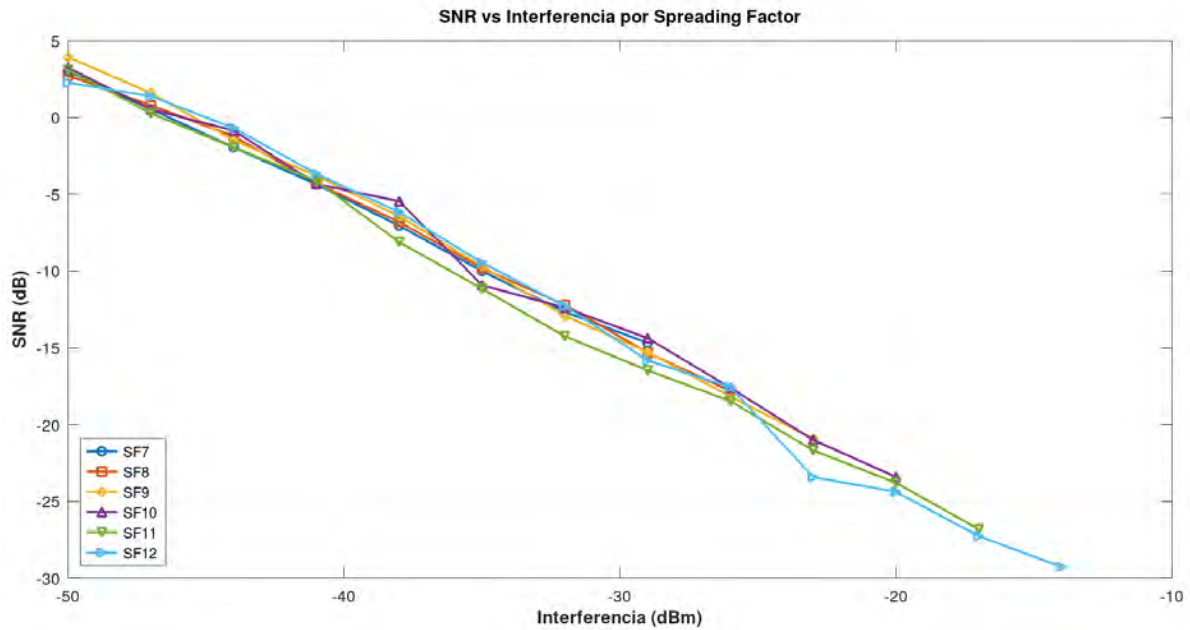


Figura 30

SNR en función de la potencia de interferencia para diferentes factores de ensanchamiento (SF7–SF12) en el enlace LoRa no cognitivo.

el SNR, CFO y STO estimados por el bloque `Frame_sync`, y los mensajes `rx_msg`: ...CRC invalid señalan los paquetes corruptos por interferencia. En el panel derecho, el espectrograma temporal (parte superior) muestra cómo el tono CW aparece como una línea vertical brillante centrada en 0 kHz (relativa a la frecuencia central de 915.6 MHz), y la representación de PSD instantánea (panel inferior derecho) revela el pico interferente sobresaliendo del piso de ruido en más de 30 dB.

Estos resultados demuestran la vulnerabilidad del sistema LoRa estándar (no cognitivo) frente a interferencia co-canal sostenida. El transmisor continúa operando con parámetros fijos aun cuando el canal se encuentra saturado, lo que se traduce en tasas elevadas de pérdida de paquetes. Esta caracterización sirve como línea base para comparar, en la siguiente sección, el desempeño del prototipo de radio cognitivo propuesto.

4.2 Evaluación del prototipo de radio cognitivo

En el escenario anterior se mostró que el sistema LoRa base (no cognitivo) es altamente vulnerable a interferencia co-canal: para SF7 el enlace comienza a fallar cuando la potencia de interferencia CW se sitúa en torno a -32 dBm, con un SNR medio cercano a -12.66 dB y una

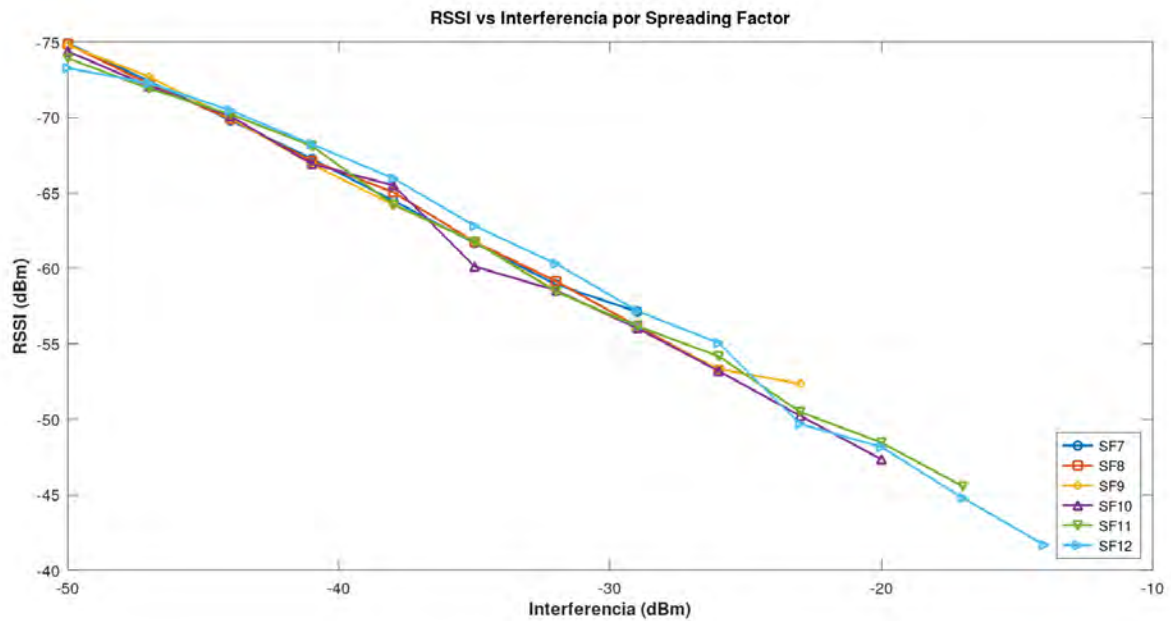


Figura 31

RSSI en función de la potencia de interferencia para diferentes factores de ensanchamiento (SF7–SF12) en el enlace LoRa no cognitivo.

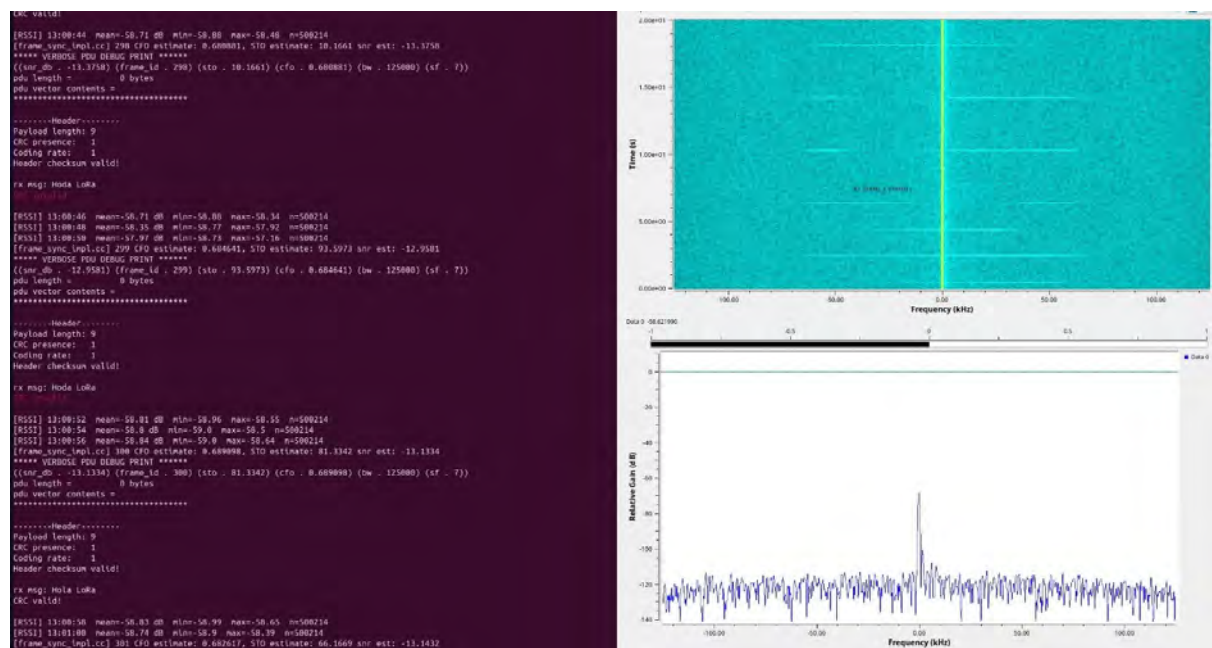


Figura 32

Captura del enlace LoRa no cognitivo con SF7 en condición de saturación del canal por interferencia CW co-canal a -32 dBm.

tasa de paquetes perdidos (PER) de 16.67 %; al aumentar la interferencia a -29 dBm el PER asciende hasta 75 %, lo que representa un colapso práctico de la comunicación como se vio en la tabla 5. Mientras que el valor de RSSI aumentó de manera directamente proporcional al valor de la interferencia.

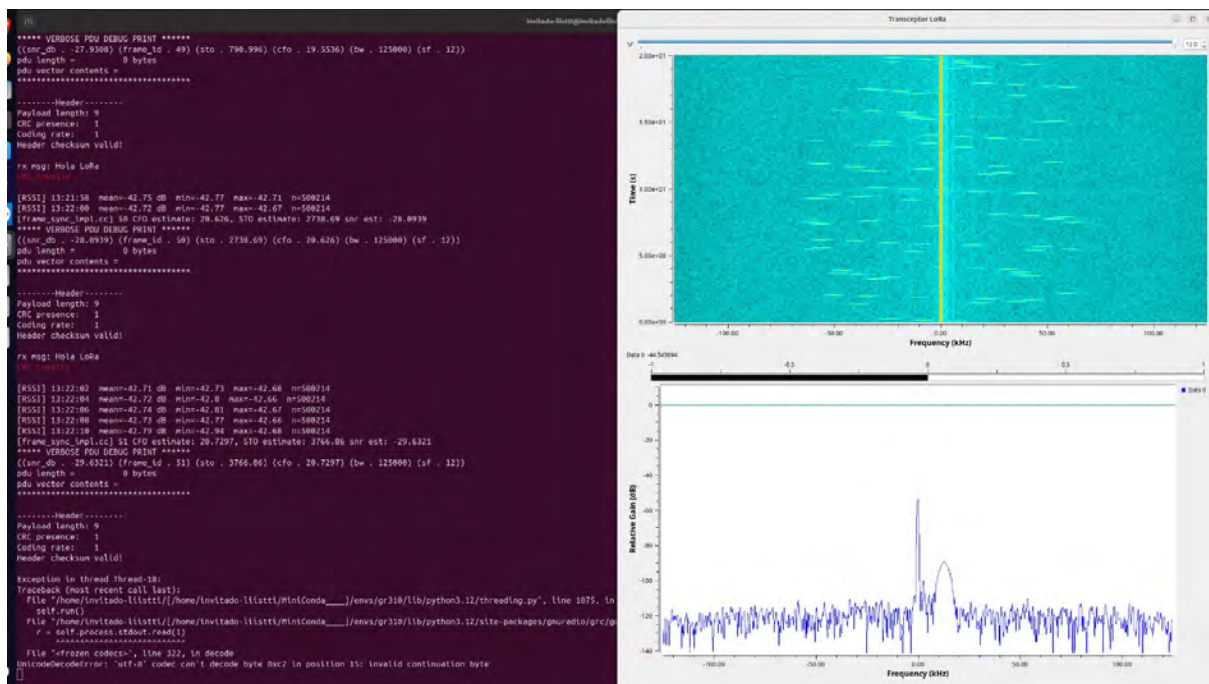


Figura 33
Captura del enlace LoRa no cognitivo con SF12 en condición de saturación del canal por interferencia CW co-canal a -17 dBm.

El objetivo de este segundo escenario es evaluar el rendimiento del sistema LoRa con radio cognitivo, que incorpora la lógica de sensado de espectro y adaptación en tiempo real del SF desarrollada en el Capítulo 3. El prototipo se somete a las mismas condiciones de interferencia (pasos de 3 dBm) que provocaron el fallo del sistema base, con el fin de verificar si las capacidades cognitivas permiten mantener la comunicación y contener el PER por debajo del umbral de 10 %.

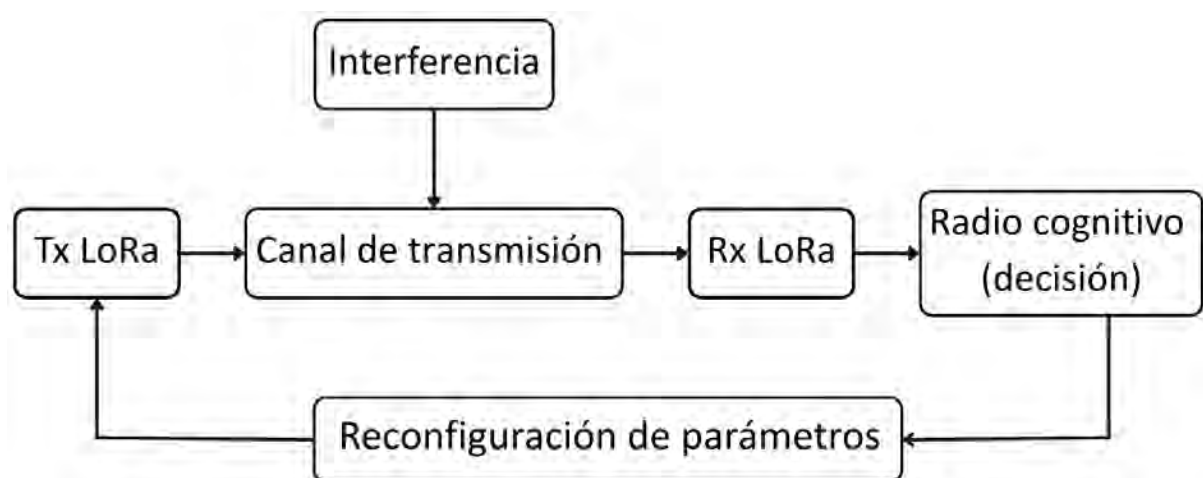


Figura 34
Flujo de pruebas con radio cognitivo

La figura 34 esquematiza el ciclo cognitivo implementado. En la etapa de transmisión, el nodo LoRa emite la señal que atraviesa un canal sometido a interferencia controlada. La característica principal de este diseño es el bucle de realimentación representado por la línea de retorno hacia el transmisor. Este enlace de control cierra el ciclo cognitivo, permitiendo que el sistema “Actúe” tras “Observar” y “Decidir”. Agrupa conceptualmente los cinco bloques cognitivos desarrollados y descritos en la sección 3.6.

4.2.1 Eventos de adaptación frecuencia–SF

La tabla 6 recoge un subconjunto representativo de mediciones que evidencian las decisiones de adaptación tomadas por los bloques cognitivos del prototipo. Esta tabla debe interpretarse como una secuencia temporal: cada fila representa el estado del enlace en un intervalo de tiempo continuo, donde las celdas con “–” en la columna de interferencia identifican periodos inmediatamente posteriores a una resintonización, cuando el sistema acaba de saltar a un canal aún no contaminado.

Tabla 6

Eventos de adaptación de frecuencia y factor de ensanchamiento en el transceptor LoRa cognitivo bajo interferencia CW.

SF	Frecuencia	Interf.	SNR (dB)	RSSI (dBm)	PER (%)
7	915.6 MHz	–50 dBm	1.93	–74.20	0.90
	915.6 MHz	–47 dBm	0.54	–71.99	0.50
	915.6 MHz	–44 dBm	–2.28	–69.01	0.10
	915.6 MHz	–41 dBm	–6.46	–65.76	0.00
	915.6 MHz	–38 dBm	–9.17	–62.76	0.00
	915.0 MHz	–	11.56	–83.07	0.00
	915.0 MHz	–38 dBm	–9.64	–62.45	0.44
	915.2 MHz	–	10.00	–83.14	0.44
8	915.2 MHz	–38 dBm	–8.41	–62.57	1.69
	915.2 MHz	–35 dBm	–12.42	–59.44	3.55
7	915.0 MHz	–	8.45	–83.16	3.45

La trayectoria registrada se compone de cuatro fases distinguibles:

1. **Fase de degradación inicial (filas 1–5):** El sistema opera en la frecuencia original de 915.6 MHz con SF7. A medida que la interferencia CW se incrementa desde –50 dBm hasta –38 dBm, el SNR desciende de un valor inicial de 1.93 dB hasta –9.17 dB. A pesar

de esta severa degradación, el bloque `ChannelPickerAndTune_SNR` no actúa debido a que el SNR aún no alcanza el umbral crítico de fallo configurado (-9 dB), manteniendo un PER prácticamente nulo ($\leq 0.9\%$).

2. **Primer salto de frecuencia (filas 6–7):** Al aproximarse el SNR al umbral crítico en la fila 5, se activa la condición de degradación sostenida. El sistema decide resintonizar la frecuencia central a 915.0 MHz, lo que provoca una recuperación inmediata del SNR a 11.56 dB con PER de 0.00 %. Sin embargo, al desplazarse la interferencia hacia este nuevo canal con una potencia de -38 dBm, el SNR vuelve a degradarse de forma abrupta hasta los -9.64 dB.
3. **Segundo salto y subida de SF (filas 8–10):** Al violarse nuevamente el umbral, el sistema ejecuta un segundo cambio de canal hacia los 915.2 MHz, logrando inicialmente un SNR de 10.00 dB y un PER bajo (0.44 %). No obstante, al persistir y aumentar la interferencia en este tercer canal hasta -35 dBm (provocando un SNR de -12.42 dB), el algoritmo determina que los saltos de frecuencia son insuficientes. En este punto, entra en acción el bloque `SfDeciderMin`, el cual incrementa el *factor de ensanchamiento* de SF7 a SF8. Esta acción incrementa la robustez del enlace y le permite tolerar niveles de ruido que en SF7 habrían significado un mayor PER, conteniendo el PER en un moderado 3.55 %.
4. **Fase de recuperación (fila 11):** Una vez que las condiciones de interferencia crítica cesan y se detecta un canal alternativo limpio disponible (915.0 MHz), el lazo cognitivo demuestra ser reversible: el sistema conmuta de regreso a dicho canal y reduce el SF nuevamente a 7 para recuperar la tasa de transmisión nominal, registrando un PER residual de 3.45 % durante el cambio de parámetros.

En todo este experimento dinámico, la PER máxima observada en el transceptor cognitivo es de apenas 3.55 %, un resultado notablemente inferior a los valores superiores al 10 % que registra el sistema no cognitivo operando bajo las mismas condiciones de interferencia estática.

4.2.2 Evidencia gráfica de los eventos cognitivos

Las figuras 35 y 36 muestran capturas de pantalla del sistema LoRa cognitivo en operación. En la figura 35 se observa una captura integral del transceptor LoRa cognitivo durante un evento de adaptación. En la parte derecha, el espectrograma temporal exhibe la dinámica completa del evento cognitivo: en su mitad inferior (instantes previos a la resintonización) destaca una línea vertical amarilla intensa en 0 kHz, correspondiente al tono CW interferente que dominaba el canal de 915.60 MHz; tras el cambio de frecuencia, dicha línea desaparece y el receptor sintoniza una región del espectro libre de interferencia, donde solo persiste el ruido térmico de fondo y las trazas tenues de las nuevas transmisiones LoRa decodificadas correctamente.

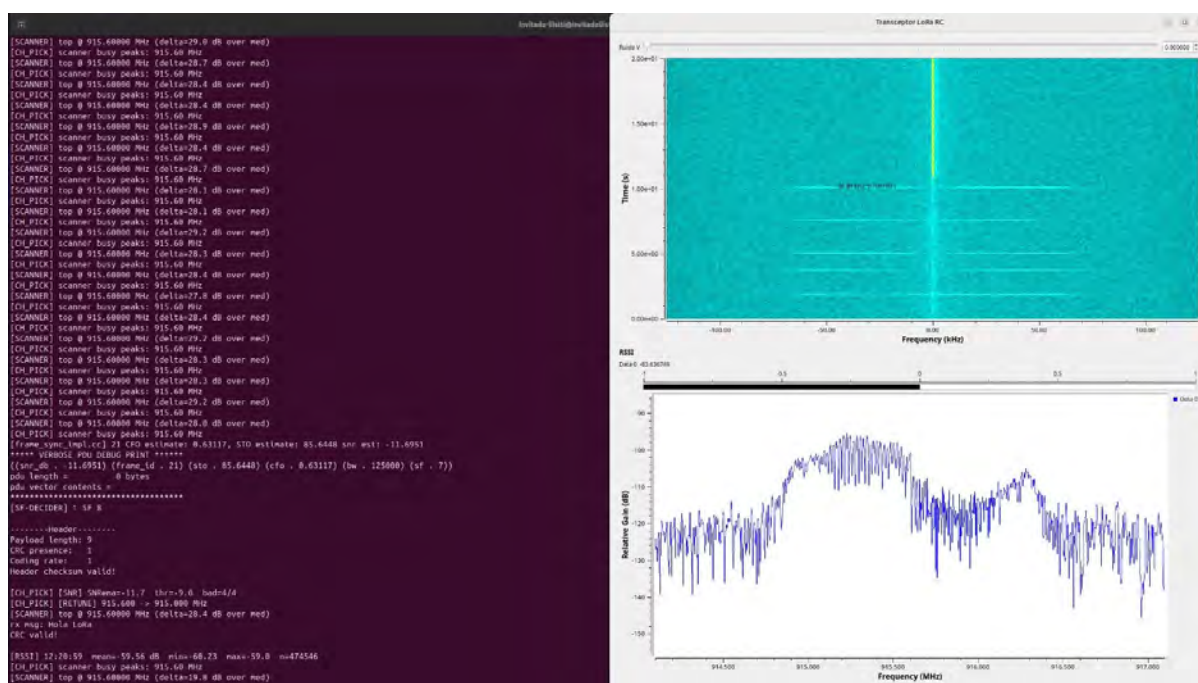


Figura 35
Captura de pantalla del transceptor LoRa cognitivo durante un evento de cambio de canal.

De igual manera, la figura 36 muestra el instante en que el bloque SfDeciderMin decide incrementar el SF de 7 a 8 tras detectar un SNR persistentemente bajo. En la consola se registra el cambio de SF y en las representaciones espectrales muestra la línea vertical amarilla del tono CW persistente a lo largo de toda la ventana observada, confirmando que el receptor permanece sintonizado en la misma frecuencia. El espectro instantáneo (panel inferior derecho) corrobora la presencia del pico interferente con un contraste superior a 30 dB sobre el piso de ruido, mientras que el RSSI medio reportado es de -62.58 dBm. La trama decodificada exitosamente (rx msg:

Hola LoRa, CRC valid) inmediatamente después del cambio de SF verifica la efectividad del ajuste de robustez como mecanismo de protección del enlace cuando la condición de canal aún no justifica una resintonización de frecuencia.

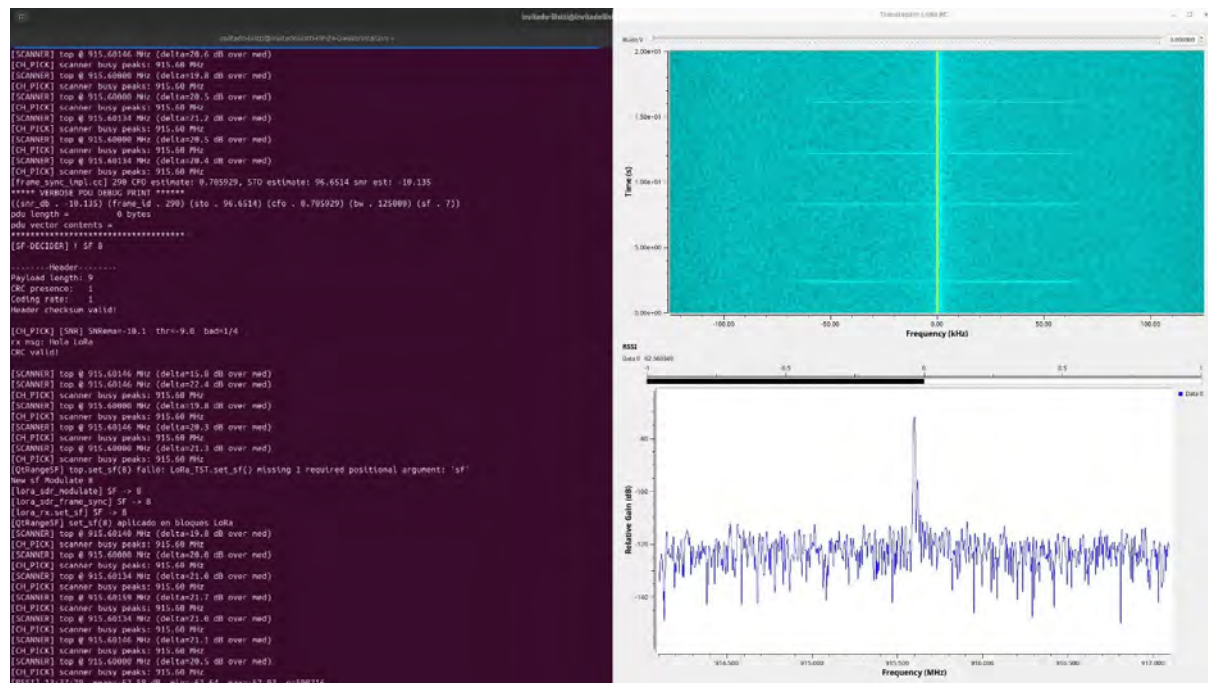


Figura 36
Captura de pantalla del transceptor LoRa cognitivo durante un evento de cambio de factor de ensanchamiento.

Antes de analizar los resultados, conviene aclarar la convención temporal de figuras 37 a 40: el eje horizontal “Índice de Medición” corresponde al número de paquete recibido en un periodo de tiempo. Las líneas verticales punteadas marcan los instantes en que se ejecutó una decisión cognitiva: las líneas rojas señalan cambio de canal de frecuencia (ejecutadas por el bloque ChannelPickerAndTune_SNR), y los marcadores en forma de rombo amarillo identifican los eventos de cambio de SF (ejecutados por el bloque SfDeciderMin).

Las figuras 37 a 40 resumen la evolución temporal de los principales parámetros del enlace cognitivo para la misma secuencia de mediciones. La figura 37 muestra el SNR en función del índice de medición. Se aprecia una degradación inicial del SNR, desde valores positivos hasta valores cercanos a -9 dB. En los instantes marcados por las líneas verticales punteadas se producen los eventos de resintonización de frecuencia: al cambiar a 915.0 MHz y luego a 915.2 MHz, el SNR se recupera rápidamente hasta aproximadamente 9 dB, lo que confirma la efectividad del salto de canal.

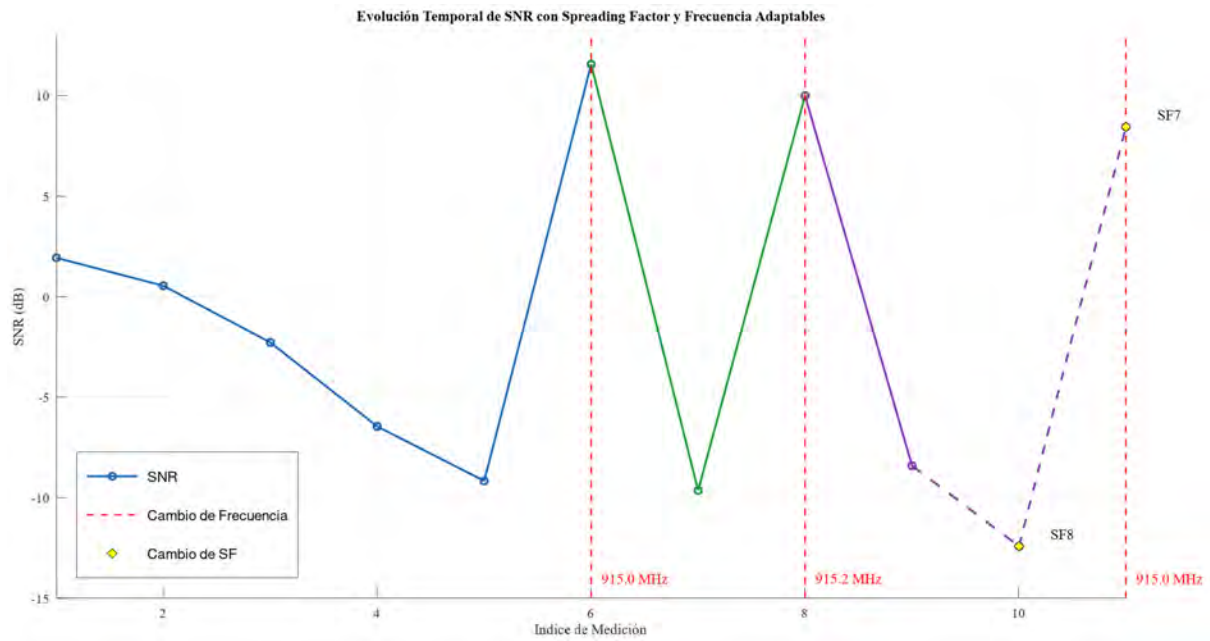


Figura 37
Evolución temporal del SNR del enlace LoRa cognitivo.

La figura 38 presenta la evolución del RSSI. Aunque el RSSI es menos sensible que el SNR a los cambios de canal y de SF, se observa que durante los periodos de mayor interferencia el nivel recibido se aproxima a -60 dBm, mientras que tras la resintonización se sitúa nuevamente por debajo de -80 dBm, lo que refleja que la señal LoRa se desplaza hacia regiones espectrales con menor potencia interferente.

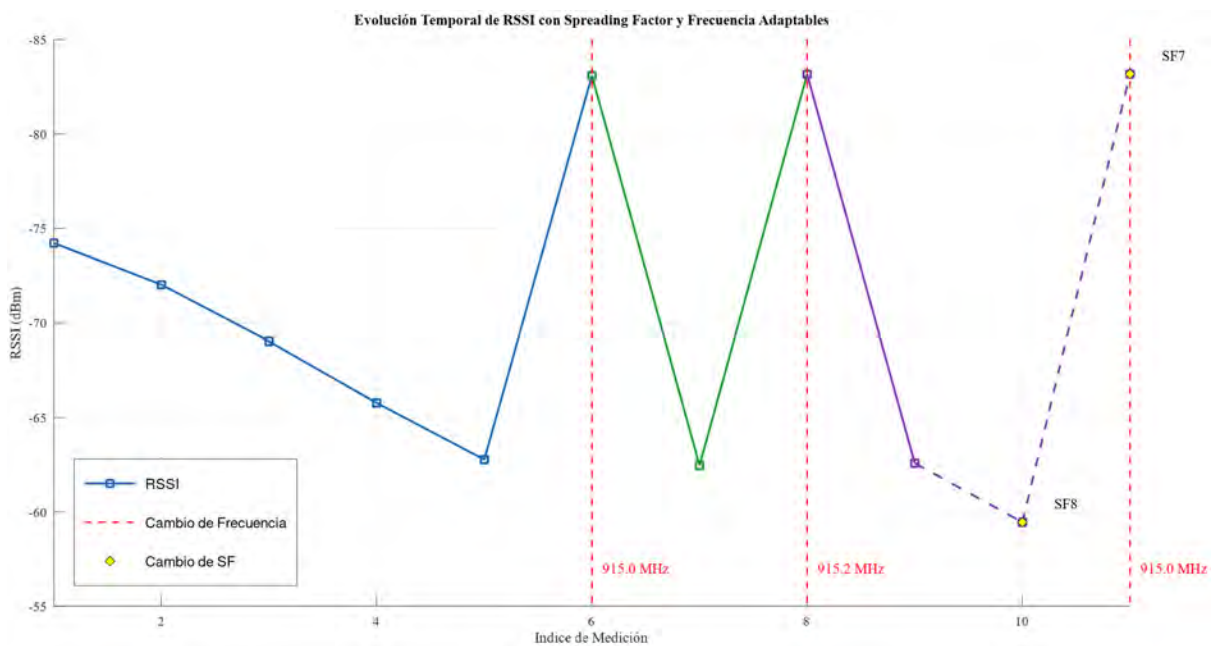


Figura 38
Evolución temporal del RSSI para el enlace LoRa cognitivo.

La figura 39 explicita la secuencia de canales utilizados por el sistema cognitivo. Se aprecia cómo el enlace permanece inicialmente en 915.6 MHz, luego migra a 915.0 MHz y, ante la presencia de nueva interferencia, salta a 915.2 MHz antes de regresar finalmente a 915.0 MHz cuando las condiciones mejoran. Estas transiciones coinciden con los eventos marcados en las Figuras 37 y 38.

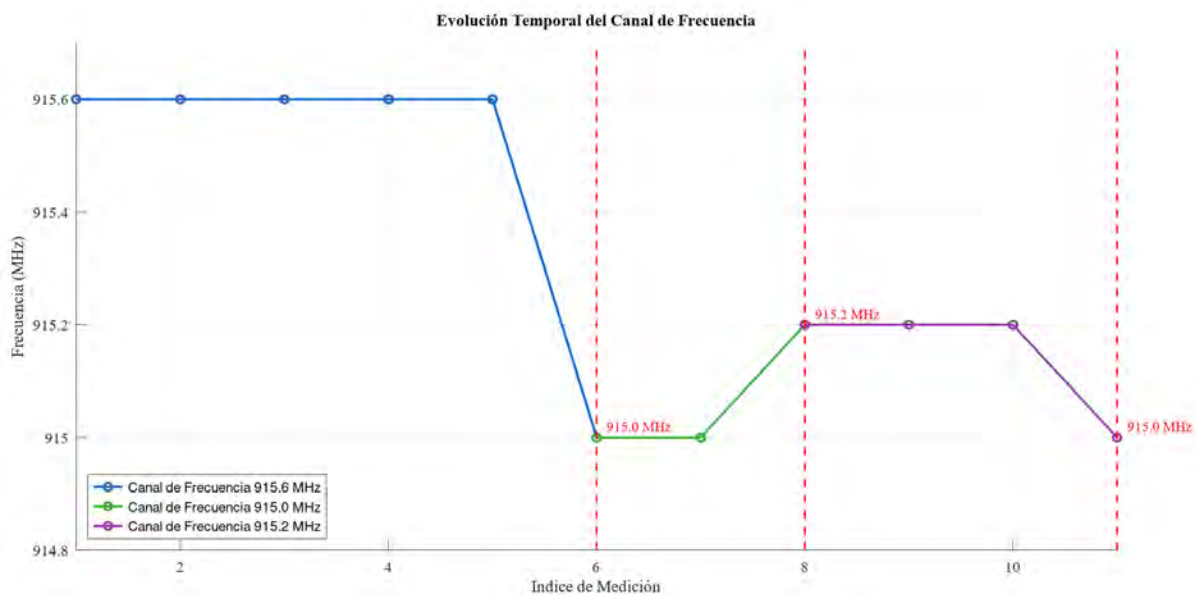


Figura 39
Evolución temporal del canal de frecuencia utilizado por el transceptor LoRa cognitivo.

Por último, la figura 40 muestra la evolución del *factor de ensanchamiento*. El valor del factor de ensanchamiento se mantiene en SF7 (valor inicial) durante la mayor parte del experimento, hasta que el SNR promedio cae por debajo del umbral configurado y el bloque SfDeciderMin decide aumentar temporalmente a SF8. Una vez que el enlace recupera margen de SNR gracias a la resintonización de frecuencia, el sistema reduce de nuevo el SF a 7, recuperando así la mayor tasa de datos posible sin comprometer la robustez del sistema LoRa.

En conjunto, estas evidencias cuantitativas y cualitativas muestran que el prototipo de radio cognitivo es capaz de anticiparse al colapso observado en el sistema no cognitivo: detecta la degradación del SNR, salta a canales menos congestionados y ajusta el SF cuando es necesario, manteniendo un PER bajo incluso bajo niveles de interferencia que antes provocaban la pérdida casi total de la comunicación.

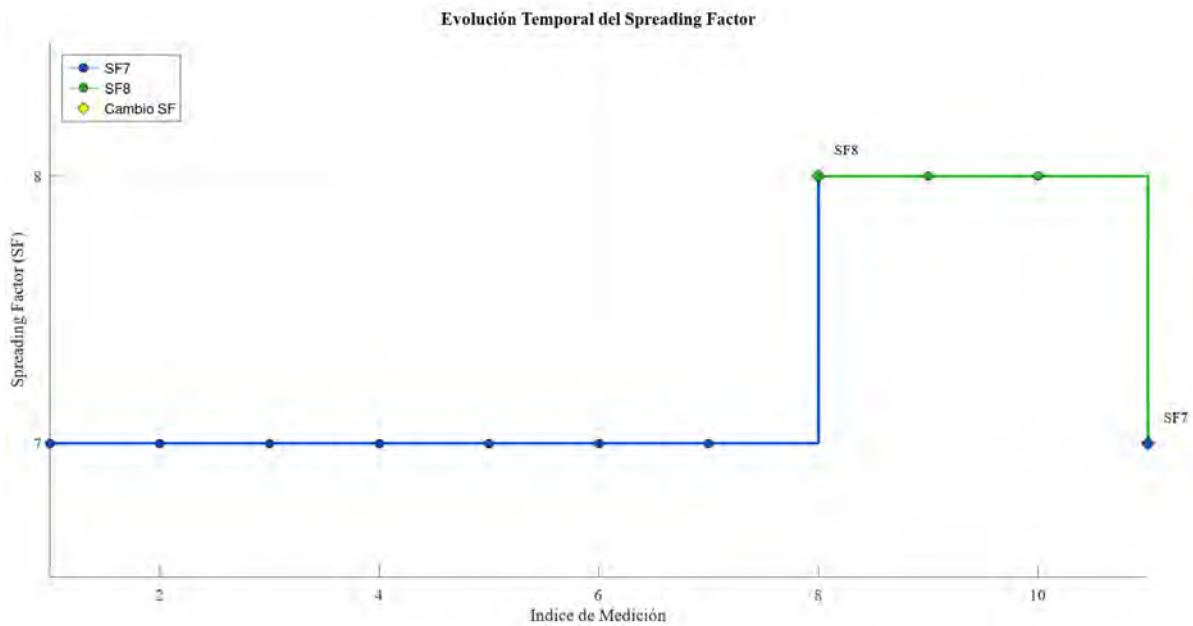


Figura 40
Evolución temporal del Factor de ensanchamiento en el sistema LoRa cognitivo.

4.3 Prueba de adaptación de robustez bajo canal simulado (AWGN)

En la sección 4.2 se evaluó el sistema cognitivo frente a una interferencia de banda estrecha (un tono CW), donde cambiar de canal resultaba una estrategia efectiva para escapar del interferente. Sin embargo, en un escenario de comunicación real el enlace también está sometido a ruido de banda ancha (*Additive White Gaussian Noise*, AWGN), que eleva el piso de ruido en todo el espectro. En estas condiciones la agilidad de frecuencia es poco efectiva, pues el nivel de ruido es similar en todos los canales disponibles; la única forma de mejorar la calidad del enlace es aumentar o reducir la robustez de la capa física mediante cambios del *factor de ensanchamiento* (SF).

La tabla 7 resume una traza experimental obtenida con el bloque `Channel Model` de GNU Radio. Para cada bloque de medición se registran la frecuencia central, el SF utilizado, el SNR y el RSSI medidos en el receptor, el valor de ruido configurado en el modelo de canal (columna *Canal (mV)*) y la PER resultante. A medida que aumenta el parámetro de ruido del canal, el SNR desciende desde valores positivos hasta niveles claramente negativos, mientras que el RSSI se hace menos negativo (mayor potencia recibida, dominada por ruido). La PER se mantiene próxima a cero en los primeros bloques y comienza a incrementarse cuando el SNR se acerca

Tabla 7

Resumen de parámetros medidos del sistema cognitivo LoRa bajo interferencia Channel Model

Bloque	Freq. (MHz)	SF	SNR (dB)	RSSI (dBm)	Canal (mV)	PER
1	915.6	7	9.77	-83.67	0.0000	0.00 %
2	915.6	7	6.10	-78.36	0.0001	0.00 %
3	915.6	7	2.07	-73.43	0.0002	0.00 %
4	915.6	7	-0.58	-70.15	0.0003	0.00 %
5	915.6	7	-3.10	-67.83	0.0004	0.00 %
6	915.6	7	-4.78	-65.89	0.0005	0.00 %
7	915.6	7	-6.04	-64.36	0.0006	0.00 %
8	915.6	7	-7.58	-63.10	0.0006	1.28 %
9	915.6	8	-9.98	-61.90	0.0008	6.60 %
10	915.6	9	-12.09	-58.44	0.0012	9.13 %
11	915.6	10	-14.13	-57.50	0.0013	8.49 %
12	915.6	11	-16.50	-57.02	0.0013	7.33 %
13	915.6	10	-12.74	-60.81	0.0009	3.15 %
14	915.6	9	-9.27	-65.93	0.0004	4.06 %
15	915.6	8	-6.98	-68.50	0.0002	4.49 %
16	915.6	7	-3.07	-70.19	0.0001	4.30 %

a la región crítica alrededor de -9 dB.

La frecuencia central permanece fija en 915.6 MHz durante toda la prueba, sin cambios de canal. Esto es un resultado esperado, con un bajo ruido AWGN aditivo, el detector CA-CFAR del bloque ED_CFAR_Scanner observa un piso de ruido uniforme, por lo que ningún canal alternativo aparece como “disponible” y el bloque ChannelPickerAndTune_SNR no encuentra justificación para el cambio de canal. La única alternativa disponible es la adaptación del SF. Finalmente, el PER no regresa a 0 % en la fase descendente (filas 13–16) aunque el SF se reduce nuevamente. Esto refleja que durante el periodo de transición —mientras el SF se ajusta y el receptor se re-sincroniza con el nuevo conjunto de parámetros— algunos paquetes se pierden temporalmente. El PER residual (1.28 % a 9.13 %) se mantiene debajo del umbral de fallo y muestra el costo de la adaptación dinámica.

En la figura 41 se representa la evolución del SNR en función del índice de medición. La curva continua muestra cómo el SNR disminuye gradualmente a medida que se incrementa el ruido en el canal simulado. A partir de la región donde el SNR cae por debajo del umbral de decisión, el enlace cognitivo comienza a modificar el SF, lo que se indica mediante la parte segmentada de la curva y las etiquetas correspondientes a cada valor de SF. También revela el estímulo externo: la calidad del canal se degrada y luego se recupera.

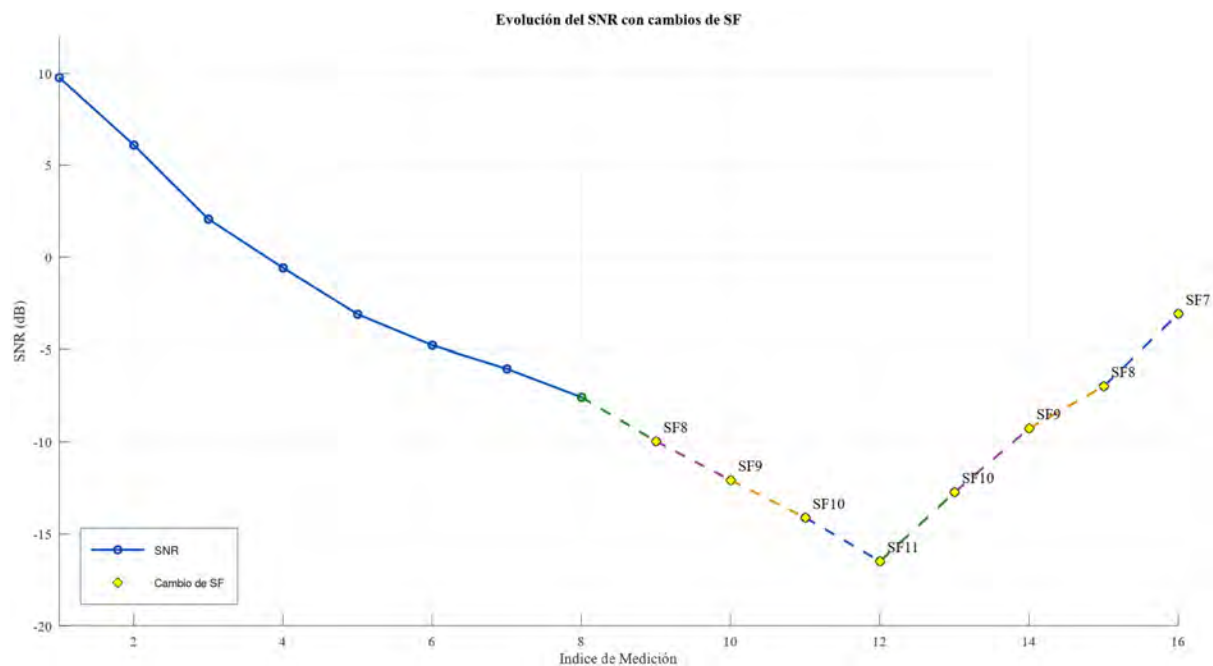


Figura 41
Evolución de SNR con Channel Model

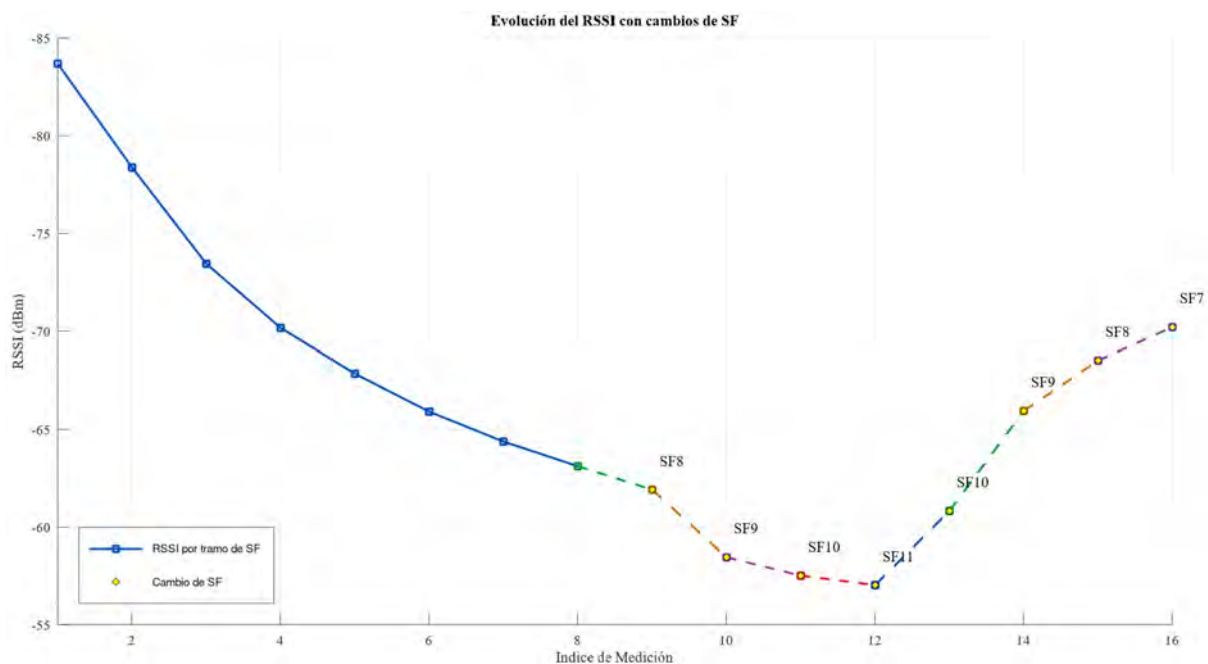


Figura 42
Evolución de RSSI con Channel Model

La figura 42 muestra la evolución del RSSI para la misma traza. Se observa una tendencia creciente (valores menos negativos) conforme se incrementa el ruido del canal, reflejando que la potencia total recibida aumenta aunque la calidad de la señal útil empeore. El cambio de frecuencia en este caso no genera una variación significativa en el RSSI, lo que refuerza la idea de que el ruido de banda ancha afecta de manera similar a todos los canales. También confirma

que la potencia total recibida aumenta uniformemente con el ruido, sin variaciones asociadas a cambios de frecuencia.

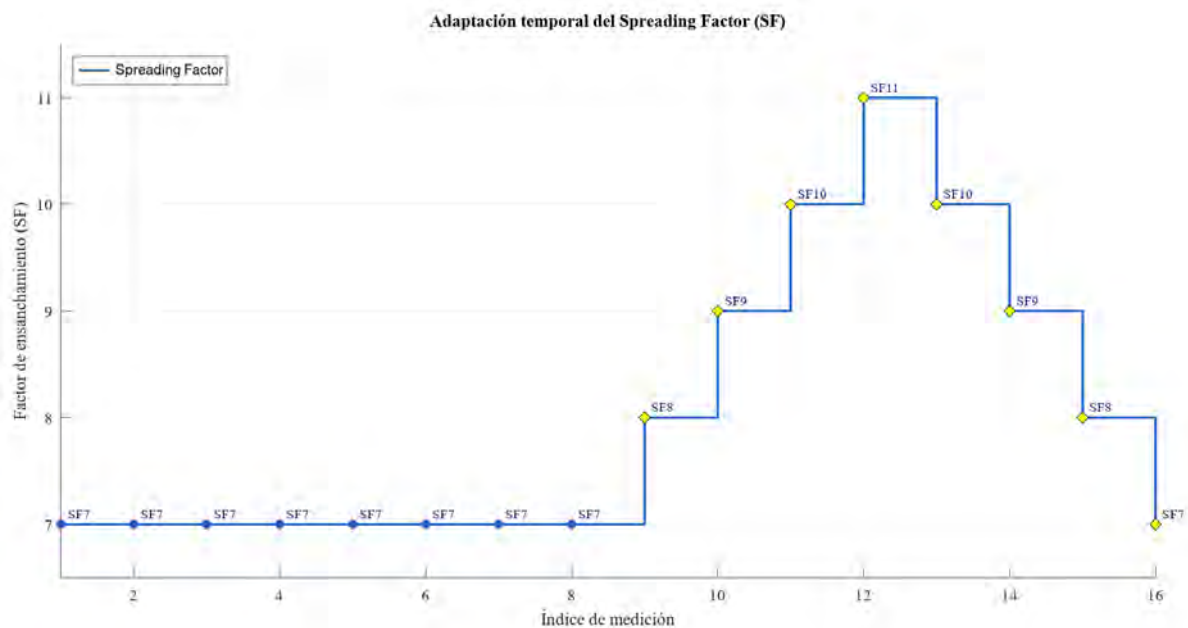


Figura 43
Evolución de SF con Channel Model

Finalmente, la figura 43 ilustra la respuesta del módulo cognitivo a las variaciones del canal. Durante los primeros bloques el SF se mantiene fijo (por ejemplo, en SF7) mientras el SNR se mantiene en una zona segura. Una vez que el SNR cae por debajo del umbral configurado, el controlador incrementa progresivamente el SF (SF8, SF9, SF10, SF11), aumentando así la robustez del enlace a costa de una menor tasa de datos. En las extensiones de esta prueba, al reducir nuevamente el nivel de ruido del canal, el mismo lazo de control permite revertir el SF en sentido contrario, recuperando capacidad de transmisión cuando las condiciones del canal mejoran.

La figura 44 muestra el instante en que el bloque SfDeciderMin decide incrementar el SF de 8 a 9 ante una degradación progresiva del canal por ruido AWGN. En la consola se observa el SNR estimado de -12.62 dB con SF8 activo, emitiendo la orden de incrementar el valor a SF 9. De manera complementaria la figura 45 muestra el instante en que el bloque SfDeciderMin decide reducir el SF de 9 a 8 al detectar una mejora sostenida del canal. En la consola se observa el registro una SNR estimada de -8.66 dB con SF9 activo, emitiendo la orden de bajar el valor a SF 8.

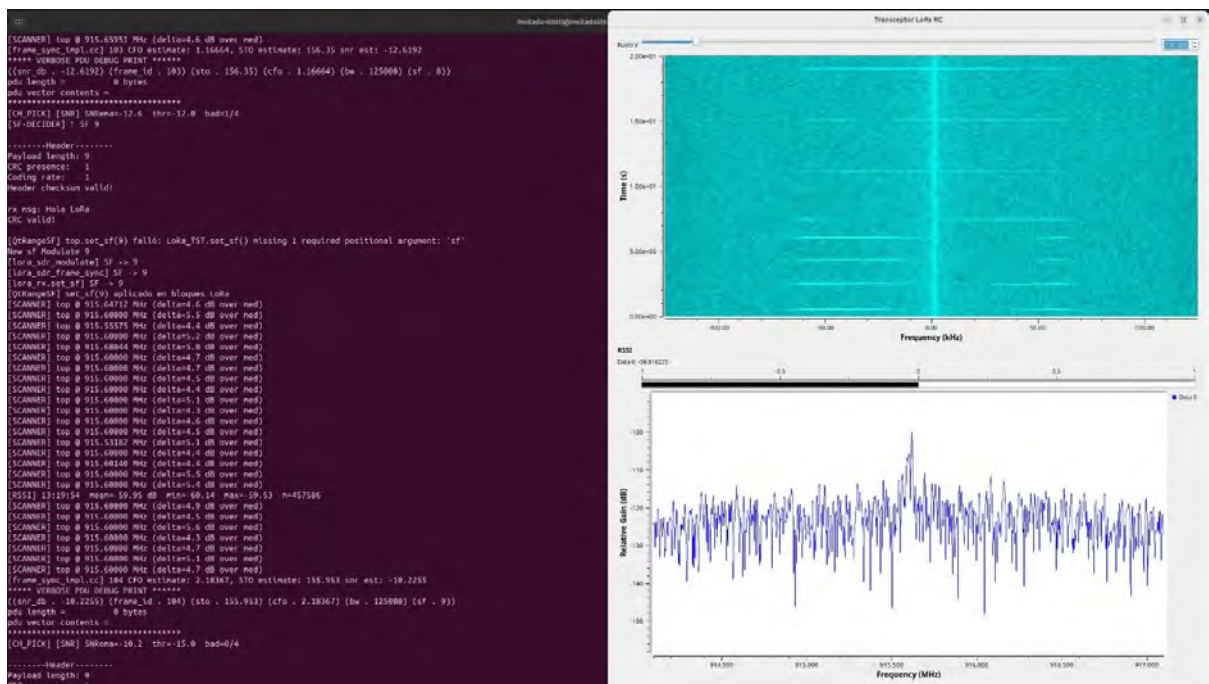


Figura 44
Captura de pantalla del transceptor LoRa cognitivo durante un evento de aumento de SF con Channel Model

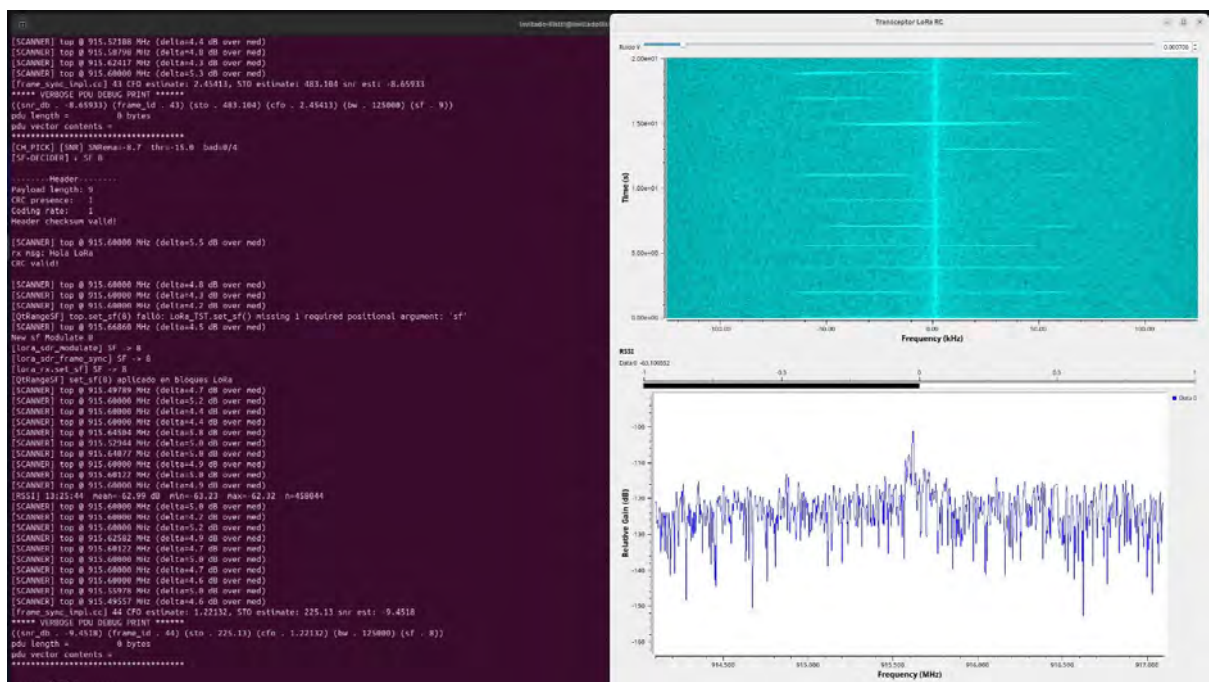


Figura 45
Captura de pantalla del transceptor LoRa cognitivo durante un evento de descenso de SF con Channel Model

En el panel izquierdo de las figuras 44 y 45 se observan los mensajes [SF DECIDER] emitidos por el bloque SfDeciderMin en el instante exacto de la transición, junto con los mensajes [CFAR_FOUND] del escáner espectral que confirman que no se identificaron canales

alternativos con mejor relación señal/ruido — situación esperada bajo ruido AWGN uniforme.

La tabla 7 y las figuras de esta sección muestran que, bajo un canal dominado por AWGN, el cambio de frecuencia deja de ser una herramienta útil, y que la decisión cognitiva basada en SNR se traduce principalmente en ajustes del SF. Este comportamiento confirma que el prototipo no solo es capaz de detectar la degradación del canal simulado, sino también de adaptar su robustez para mantener el enlace operativo en un rango más amplio de condiciones de ruido.

DISCUSIONES

El problema central de esta tesis fue determinar en qué medida la implementación de radio cognitivo afecta al rendimiento de un enlace típico de WSN. Las pruebas realizadas muestran que un enlace LoRa estático responde de manera pasiva a la degradación del enlace. El prototipo cognitivo es capaz de cambiar dinámicamente sus parámetros (SF y frecuencia) para mitigar el punto de fallo y mantener niveles aceptables de rendimiento. En términos de la definición clásica de radio cognitivo como sistema consciente de su entorno y capaz de adaptarse a las variaciones estadísticas del canal (Haykin, 2005; Mitola, 1999), el prototipo desarrollado constituye una implementación funcional del ciclo sensor–decidir–actuar sobre una tecnología LPWAN concreta.

Los patrones observados son coherentes con los modelos analíticos de LoRa bajo ruido e interferencia. Diversos estudios (Afisiadis et al., 2019; Eigner, 2021) muestran que la modulación CSS de LoRa ofrece una elevada robustez frente a ruido blanco, pero que la interferencia co–canal —especialmente de otros usuarios LoRa— puede incrementar severamente la tasa de error más allá de los límites de demodulación asociados a cada SF. Los resultados obtenidos confirman este comportamiento y añaden una solución práctica: en lugar de aceptar un límite fijo, el enlace cognitivo utiliza umbrales de SNR para activar mecanismos cognitivos (cambio de canal, aumento de SF), conciliando así la teoría de radio cognitivo con las características de LoRa.

La tesis de Cruz Yupanqui (2023) en la UNSAAC evaluó una red LoRaWAN real sometida a interferencia co–canal y por canal adyacente, concluyendo que la tecnología mantiene una alta robustez gracias a su esquema de modulación. En dicho trabajo el RSSI se mantiene prácticamente constante mientras el SNR es el parámetro que marca el límite de operación de cada SF y que el uso de distintos factores de ensanchamiento permite mitigar de forma distinta la interferencia co–canal y la de canal adyacente. Esos hallazgos son coherentes con los de este trabajo; la diferencia clave es que aquí se exploran mecanismos cognitivos que permiten

mejorar dinámicamente el enlace en esa región crítica mediante decisiones automáticas sobre SF y frecuencia.

La literatura reciente sobre LoRa cognitiva se ha centrado mayormente en propuestas a nivel de protocolo y gestión de recursos. Por ejemplo, en LoRaCog se plantea un acceso cognitivo al espectro para redes LoRa (Salika et al., 2022), mientras que otros trabajos proponen esquemas de gestión dinámica de recursos para LoRaWAN cognitivo (Balyan, 2024). También se han desarrollado mecanismos adaptativos de selección de SF y potencia de transmisión, ya sea mediante extensiones al ADR de LoRaWAN (Silva et al., 2023; Aliagas et al., 2025) o mediante técnicas ligeras de ajuste de parámetros físicos (Li et al., 2014). Asimismo, se han explorado enfoques de *spectrum sensing* inteligente basados en inteligencia artificial para comunicaciones LoRa (Mutescu et al., 2025). Frente a estas propuestas, el aporte de la presente tesis es la implementación y validación en SDR de un lazo cognitivo directamente en la capa física de LoRa, demostrando que estas ideas no solo son viables en simulación o a nivel de red, sino también en enlaces reales representativos de WSN.

Los resultados permiten responder de forma concreta qué conceptos de radio cognitivo son más relevantes para WSN en bandas ISM. Por un lado, se confirma la utilidad del sensado local del canal —mediante SNR, RSSI y PER— como estimador práctico de ocupación e interferencia, en línea con la idea de *radio-scene analysis* planteada en la literatura de radio cognitivo (Haykin, 2005). Por otro lado, la experiencia con el canal simulado AWGN muestra que no todas las acciones cognitivas son igualmente efectivas: cuando el problema dominante es un interferente de banda estrecha, la selección dinámica de canal resulta claramente beneficiosa; cuando el fenómeno dominante es un aumento del piso de ruido en todo el espectro, el cambio de canal deja de ser útil y la adaptación del SF se convierte en la herramienta principal. Esta distinción aporta una guía de diseño para futuros esquemas cognitivos en LoRa: las decisiones deben considerar el tipo de degradación del canal y no solo el valor instantáneo de SNR.

Las implicaciones para aplicaciones WSN son relevantes. En escenarios donde múltiples tecnologías comparten bandas no licenciadas, es poco realista suponer una planificación centralizada del espectro. La evidencia obtenida sugiere que dotar a los nodos o *gateways* LoRa de capacidades cognitivas sencillas puede extender la región operativa del enlace sin cambios

en las capas superiores del protocolo y con un costo de complejidad moderado. Esta mejora en robustez incrementa el tiempo en aire y, por lo tanto, el consumo energético y la ocupación del canal. Cualquier estrategia cognitiva para WSN deberá, por tanto, balancear resiliencia, eficiencia espectral y consumo energético, en consonancia con las revisiones sobre radio cognitivo en LPWAN para aplicaciones industriales (Nurelmadina et al., 2021).

Finalmente, los resultados deben interpretarse considerando ciertas limitaciones. En primer lugar, las pruebas se realizaron sobre un único enlace nodo-*gateway* en laboratorio, con generadores de interferencia y modelos de canal controlados; no se realizaron pruebas en redes densas con múltiples nodos compitiendo por el canal ni en entornos exteriores con movilidad y multitrayecto complejo. En segundo lugar, el lazo cognitivo se basa en reglas estáticas con umbrales fijados a partir del análisis de datos; esquemas más avanzados podrían beneficiarse de técnicas de aprendizaje por refuerzo o de *spectrum sensing* inteligente, como las propuestas para LoRa (Mutescu et al. 2025), capaces de explotar mejor la información histórica del canal. En tercer lugar, no se abordaron aspectos de equidad entre usuarios ni de cumplimiento estricto de restricciones regulatorias de ciclo de trabajo, aspectos críticos cuando se escalan estas ideas a redes LoRaWAN de gran tamaño.

Los resultados y conclusiones de esta tesis se establecen, en sentido estricto, para el escenario experimental evaluado: un único enlace nodo-*gateway* operando en banda de 915 MHz, en el entorno controlado del laboratorio, sometido a interferencia de banda estrecha (tono CW del ERASynth Micro) y a degradación de banda ancha emulada con el bloque Channel Model. La proyección a redes WSN reales debe entenderse como una extrapolación cualitativa fundamentada: los mecanismos validados (sensado por SNR/RSSI, salto de canal y adaptación de SF) operan en la capa física y son independientes de la topología de red, por lo que se espera que conserven su efecto mitigador en despliegues reales. Por tanto, las conclusiones se afirman plenamente para el escenario experimental y se proponen como hipótesis de trabajo, con respaldo teórico, para el caso de redes WSN reales, cuya verificación se plantea en las recomendaciones.

Bajo las pruebas realizadas se anticipa el siguiente comportamiento. (a) Varios emisores y colisiones: la presencia simultánea de múltiples transmisores eleva la ocupación detectada por el escáner CA-CFAR y reduce la disponibilidad de canales limpios; el salto de frecuencia

seguiría siendo útil mientras existan huecos espectrales, pero perdería eficacia a medida que la banda se satura, situación en la que la adaptación de SF pasaría a ser el mecanismo dominante como ocurre en la prueba AWGN, que actúa como una aproximación al efecto de un piso de interferencia agregada. Las colisiones entre tramas LoRa del mismo SF son el caso más severo, pues son cuasi-ortogonales solo entre SF distintos; aquí el aumento de SF puede incluso ayudar al separar la trama propia de las interferentes por ortogonalidad de SF. (b) Canales adyacentes: la rejilla de 200 kHz con $BW = 125$ kHz deja una banda de guarda que limita el solape, por lo que la interferencia de canal adyacente sería menos crítica que la co-canal, en concordancia con lo reportado por Cruz Yupanqui (2023). (c) Patrones no estacionarios: ante ráfagas breves y frecuentes podría aumentar la latencia de reacción o inducir conmutaciones más frecuentes (flapping), mitigadas por la histéresis de 2 dB ya implementada. En conjunto, se espera que el prototipo conserve su capacidad de mitigación, pero con un margen de mejora menor y una mayor frecuencia de decisiones; esto refuerza la pertinencia de incorporar esquemas de decisión con memoria del canal (filtros bayesianos o aprendizaje por refuerzo) para tráfico no estacionario, según se detalla en las recomendaciones.

En resumen, la tesis muestra que los principios de radio cognitivo formulados por Mitola y Haykin pueden trasladarse de manera efectiva a un enlace LoRa sobre SDR, proporcionando mejoras tangibles de adaptabilidad frente a interferencia y ruido en un contexto WSN. Al mismo tiempo, revela un amplio espacio de investigación futura en torno a la integración de decisiones cognitivas distribuidas, algoritmos de aprendizaje y pruebas a gran escala que permitan llevar estas capacidades desde el laboratorio hacia aplicaciones reales de LoRaWAN cognitivo.

CONCLUSIONES

- Se evaluó la implementación de un prototipo de radio cognitivo sobre SDR para enlaces LoRa en WSN. El enlace no cognitivo colapsó con PER mayor a 10 % mientras el prototipo cognitivo mantuvo PER menor a 10 %, por lo tanto el prototipo de radio cognitivo es más robusto que un enlace no cognitivo.
- Se analizó que los conceptos de radio cognitivo aplicables a WSN son: el sensado del entorno espectral, la toma de decisiones basada en SNR y la reconfiguración adaptativa de parámetros físicos (frecuencia, SF). Estos conceptos permiten compensar la falta de planificación de espectro y aumentar la robustez de los enlaces nodo-*gateway* en bandas ISM.
- Se diseñó e implementó un prototipo funcional sobre SDR (nodo-*gateway*) incorporando mecanismos de sensado y acciones de adaptación (cambio de frecuencia y ajuste de SF), integrados al flujo de transmisión/recepción en GNU Radio.
- Se configuró el prototipo para operar en un escenario WSN, habilitando parámetros de sensado y decisión basados en calidad de enlace. La operación adaptativa permitió sostener el enlace bajo interferencia con PER entre 0.00 % y 9.13 % tal como se muestra en la Tabla 6 y Tabla 7.
- Se realizó prueba en canal limpio y bajo interferencia co-canal mediante un generador CW ERASynth Micro. Bajo interferencia creciente, el sistema no cognitivo presentó PER mayor a 10 %, mientras que el prototipo cognitivo mantuvo la comunicación mediante la resintonización de frecuencia y ajustes de SF con un PER menor al 10 %.
- Se analizó la SNR y el RSSI para caracterizar cuantitativamente el rendimiento del prototipo. En presencia de interferencia, el enlace cognitivo redujo significativamente el PER comparado con el enlace no cognitivo y sostuvo niveles más estables de SNR/RSSI

mediante reconfiguración dinámica, demostrando una mejora en el rendimiento del enlace para escenarios WSN en bandas ISM.

RECOMENDACIONES

Para posteriores trabajos se recomienda tomar en cuenta:

- Extender las pruebas a escenarios multi-nodo implementando futuros ensayos con múltiples nodos sensores y un *gateway* único. Esto permitiría evaluar el comportamiento del radio cognitivo bajo colisiones entre tramas LoRa y no solo frente a interferencia CW, acercándose más a un entorno real de WSN.
- Evaluar el impacto energético de las decisiones cognitivas incorporando mediciones de consumo de energía en el nodo transmisor para medir el costo de los cambios de SF y de la resintonización de frecuencia. Asimismo diseñar políticas de decisión que consideren explícitamente la autonomía de baterías.
- Realizar pruebas en entornos reales y de mayor alcance. Repetir las mediciones en escenarios reales con mayores distancias y condiciones de propagación más complejas, a fin de validar la robustez del prototipo en aplicaciones reales de monitoreo.
- Refinar la lógica de decisión cognitiva. Investigar esquemas más avanzados de decisión (por ejemplo, filtros bayesianos o técnicas de aprendizaje automático) para combinar información histórica de SNR, PER y ocupación espectral, reduciendo aún más la probabilidad de decisiones erróneas.
- Explorar otras bandas y regulaciones. Adaptar y validar el prototipo en otras bandas ISM (p. ej., 868 MHz) y bajo distintos marcos regulatorios, con el propósito de evaluar la portabilidad de la solución a diferentes regiones y aplicaciones de IoT industrial.

BIBLIOGRAFÍA

- Adelantado, Ferran y Vilajosana, X. y. T.-P. P. y. M. B. y. M.-S. J. y. W. T. (2017). Understanding the limits of LoRaWAN. *IEEE Communications Magazine*, 55(9):34–40.
- Afisiadis, Orion y Burg, A. y. B.-S. A. (2019). Coded LoRa frame error rate analysis. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.
- Akram, S. V. y Singh, R. y. A. M. A. y. G. A. y. R. M. y. F. O. S. y. E. S. W. y. P. D. (2021). Performance analysis of IoT and long-range radio-based sensor node and gateway architecture for solid waste management. *Sensors*, 21(8):2774.
- Akyildiz, Ian F. y Lee, W.-Y. y. V. M. C. y. M. S. (2006). NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks*, 50(13):2127–2159.
- Aliagas, Carles y Pueyo Centelles, R. y. M. R. y. M. P. y. M. C. (2025). Dynamic selection and detection of spreading factors and channels for end-node devices of lora networks. *Electronics*, 14(17):3341.
- Augustin, Alpar y Yi, J. y. C. T. y. T. W. M. (2016). A study of LoRa: Long range & low power networks for the Internet of Things. *Sensors*, 16(9):1466.
- Axell, Erik y Leus, G. y. L. E. G. y. P. H. V. (2012). Spectrum sensing for cognitive radio: State-of-the-art and recent advances. *IEEE Signal Processing Magazine*, 29(3):101–116.
- Balyan, V. (2024). Resource management for cognitive radio-based lorawan. *International Journal on Smart Sensing and Intelligent Systems*, 17(1).
- Barkat, M. (1991). *Signal Detection and Estimation*. Artech House, Boston, MA.

- Blossom, E. (2004). GNU radio: Tools for exploring the radio frequency spectrum. *Linux Journal*, 2004(122):4.
- Bor, Martin y Vidler, J. E. y. R. U. (2016). LoRa for the Internet of Things. In *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks (EWSN)*, pages 361–366.
- Buitrago Rodil, C. A. (2018). Evaluación de desempeño de un sistema de radio cognitivo para una red LoRaWAN. Tesis de maestría, Pontificia Universidad Javeriana.
- Busacca, F. y Mangione, S. y. P. S. y. R. F. y. T. I. (2024). SDR LoRa, an open-source, full-fledged implementation of LoRa on Software-Defined Radios: Design and potential exploitation. *Computer Networks*, 241:110194.
- Croce, Danilo y Gucciardo, M. y. T. I. y. G. D. y. M. S. (2018). Impact of LoRa imperfect orthogonality: Analysis of link-level performance. *IEEE Communications Letters*, 22(4):796–799.
- Cruz Yupanqui, N. c. (2023). Evaluación de los parámetros de transmisión de una red lorawan en presencia de interferencia electromagnética sobre la banda de 900 mhz. Tesis de grado, Universidad Nacional de San Antonio Abad del Cusco.
- Cuña, [German] y Durán, R. y. O. M. (2014). Redes cognitivas: Estudio de la movilidad en el espectro.
- Domínguez Ibarra, R. F. (2019). Diseño de una red inalámbrica de sensores para un sistema de monitoreo de temperatura y humedad relativa en un campus universitario. Tesis de grado, Pontificia Universidad Católica del Perú.
- Eigner, H. (2021). Interference analysis of LoRaWAN systems. Diploma thesis, Technische Universität Wien.
- Federal Communications Commission (2002). Spectrum policy task force report. ET Docket No. 02-135.

- Federal Communications Commission (2021). Operation of unlicensed devices. <https://www.sciencedirect.com/topics/computer-science/unlicensed-device>.
- García-Requejo, A. y Pérez-Rubio, M. C. y. V. J. M. y. H. Á. (2023). Activity monitoring and location sensory system for people with mild cognitive impairments. *IEEE Sensors Journal*, 23(5):5448–5458.
- Gardner, William A. y Napolitano, A. y. P. L. (2006). Cyclostationarity: Half a century of research. *Signal Processing*, 86(4):639–697.
- Georgiou, Orestis y Raza, U. (2017). Low power wide area network analysis: Can LoRa scale? *IEEE Wireless Communications Letters*, 6(2):162–165.
- Goldsmith, Andrea J. y Chua, S.-G. (1998). Adaptive coded modulation for fading channels. *IEEE Transactions on Communications*, 46(5):595–602.
- Goswami, P. K. and Goswami, G. (2020). Compact corner truncated fractal slot antenna for cognitive radio sensor network. *IETE Journal of Research*.
- GSMA Intelligence (2024). The mobile economy latin america 2024. Reporte regional, GSM Association.
- Gurugopinath, S. y Kishore, R. y. B. V. N. (2020). A comprehensive survey on recent advancements in cognitive radio-based internet of things. *Advanced Computing & Communications*, 4(1-2):13–22.
- Harris, F. J. (1978). On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, 66(1):51–83.
- Haykin, S. (2005). Cognitive radio: Brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, 23(2):201–220.
- Joachim, T., Afisiadis, O., Mayoraz, P., Balatsoukas-Stimming, A., and Burg, A. (2020). An Open-Source LoRa Physical Layer Prototype on GNU Radio. pages 1–5.

- Li, S. K. y Jayaweera, S. K. y. B. M. y. G. C. (2014). Learning-aided sub-band selection algorithms for spectrum sensing in wide-band cognitive radios. *IEEE Transactions on Wireless Communications*, 13(4):2012–2024.
- Lin, Kaiqiang y Hao, T. (2021). Experimental link quality analysis for LoRa-based wireless underground sensor networks. *IEEE Internet of Things Journal*, 8(8):6565–6577.
- LoRa Alliance Technical Committee (2019). LoRaWAN[®] specification v1.0.3. LoRa Alliance.
- LoRa Alliance Technical Committee (2021). LoRaWAN[®] regional parameters RP002-1.0.2. LoRa Alliance.
- Manna, S. y Misra, S. (2019). Cognitive radio-based IoT gateway for smart healthcare system. *Physical Communication*, 35:100713.
- Ministerio de Transportes y Comunicaciones (MTC) (2022). Políticas y lineamientos para la masificación 5g, smart cities y soluciones iot. Resoluciones y reportes de gestión, Gobierno del Perú, Lima, Perú.
- Mitola, Joseph y Maguire, G. Q. (1999). Cognitive radio: Making software radios more personal. *IEEE Personal Communications*, 6(4):13–18.
- Mitola, J. (2000). *Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio*. PhD thesis, Royal Institute of Technology (KTH), Stockholm, Sweden.
- Mutescu, Partemie-Marian y Lavric, A. y. P. A.-I. y. C. A.-M. y. P. V. (2025). Radio spectrum sensing framework for large-scale, high-density iot sensor networks. *IEEE Access*, 13:64594–64609.
- National Instruments (2014). *NI USRP-29xx Getting Started Guide*. National Instruments Corporation.
- Nurelmadina, N. y Hasan, M. K. y. M. I. y. S. R. A. y. A. K. A. Z. y. A. E. S. y. M. R. A. y. I. S. y. H. E. y. H. M. A. (2021). A systematic review on cognitive radio in low power wide area network for industrial IoT applications. *Sustainability*, 13(1):338.

- Onumanyi, A. J. y Abu-Mahfouz, A. M. y. H. G. P. (2020). Adaptive threshold techniques for cognitive radio-based low power wide area network. *Transactions on Emerging Telecommunications Technologies*, 31:e3908.
- OSIPTEL (2023). Análisis del desempeño financiero del sector telecomunicaciones 2023 e informes de congestión de espectro. Informe regulatorio, Organismo Supervisor de Inversión Privada en Telecomunicaciones, Lima, Perú.
- Pérez, D. y Risco, R. (2020). Implementación de LoRa y LoRaWAN como escenario futuro de la Industria 4.0 en el sector agroindustrial peruano. *Revista Campus*, 25(29):133–147.
- Reynders, Brecht y Meert, W. y. P. S. (2017). Power and spreading factor control in low power wide area networks. In *Proceedings of the 2017 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.
- Richards, M. A. (2014). *Fundamentals of Radar Signal Processing*. McGraw-Hill, New York, NY, 2 edition.
- Salika, F. y Nasser, A. y. M. M. y. P. B. y. M. A. (2022). LoRaCog: A protocol for cognitive radio-based LoRa network. *Sensors*, 22:3885.
- Santiago, J. (2017). Implementación de una metodología para la medición de la interferencia inalámbrica en la banda ISM en zonas exteriores urbanas para garantizar la comunicación de una red inalámbrica de sensores.
- Semtech Corporation (2020). SX1276/77/78/79 datasheet: Long range, low power transceiver. Semtech Corporation, Rev. 7.
- Silva, Pedro E. Gória y Marchetti, N. y. N. P. H. J. y. d. S. R. A. A. (2023). Enabling semantic-functional communications for multiuser event transmissions via wireless power transfer. *Sensors*, 23(5):2707.
- Slabicki, Mariusz y Preamsankar, G. y. D. F. M. (2018). Adaptive configuration of LoRa networks for dense IoT deployments. In *NOMS 2018 – 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9. IEEE.

- Statista (2024). Number of IoT connected devices worldwide 2019–2030. Statista Research Department.
- Statista (2025). Internet of things (IoT): number of connected devices worldwide 2015–2033. Statista Research Department.
- Stoica, Petre y Moses, R. (2005). *Spectral Analysis of Signals*. Prentice Hall, Upper Saddle River, NJ.
- Tapparel, J. and Burg, A. (2024). Design and implementation of lora physical layer in GNU Radio. In *Proceedings of the 14th GNU Radio Conference*.
- Tesfay, Angesom Ataklity y Simon, E. P. y. N. I. y. C. L. (2020). Multiuser detection for downlink communication in lora-like networks. *IEEE Access*, 8:199001–199015.
- Thilina, K. M. y Choi, K. W. y. S. N. y. H. E. (2013). Machine learning techniques for cooperative spectrum sensing in cognitive radio networks. *IEEE Journal on Selected Areas in Communications*, 31(11):2209–2221.
- Tian, Z. y Giannakis, G. B. (2006). A wavelet approach to wideband spectrum sensing for cognitive radios. In *Proc. 1st Int. Conf. on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, pages 1–5. IEEE.
- Wang, J. y. o. (2019). Decoding superposed lora signals. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*. IEEE.
- Wang, Xiaofeng y Han, Y. y. W. C. y. W. Q. (2018). Cognitive Internet of Things: A New Paradigm Beyond Connection. *IEEE Internet of Things Journal*, 5(3):2010–2023.
- Welch, P. D. (1967). The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73.
- Xhonneux, Martin y Tapparel, J. y. A. O. y. B.-S. A. y. B. A. (2020). A maximum-likelihood-based multi-user LoRa receiver implemented in GNU Radio. In *Proceedings of the 54th Asilomar Conference on Signals, Systems and Computers*, pages 1299–1303. IEEE.

- Xia, Xianjin y Zheng, Y. y. G. T. (2020). Ftrack: Parallel decoding for lora transmissions. *IEEE/ACM Transactions on Networking*, 28(6):2573–2586.
- Xia, Xianjin y Zheng, Y. y. G. T. (2022). An open-source gnu radio framework for lora physical layer and collision resolution. *IEEE Internet of Things Journal*.
- Ye, Zhuan y Memik, G. y. G. J. (2008). Energy detection using estimated noise variance for spectrum sensing in cognitive radio networks. In *2008 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 711–716. IEEE.
- Yucek, Tevfik y Arslan, H. (2009). A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE Communications Surveys & Tutorials*, 11(1):116–130.
- Zegarra Chávez, K. R. (2021). Diseño de una red inalámbrica de sensado remoto con aplicación en un escenario de selva baja. Tesis de grado, Pontificia Universidad Católica del Perú.
- Zeng, Yonghong y Liang, Y.-C. (2009). Eigenvalue-based spectrum sensing algorithms for cognitive radio. *IEEE Transactions on Communications*, 57(6):1784–1793.
- Zhang, L. y Liang, Y. C. y. X. M. (2019). Spectrum sharing for Internet of Things: A survey. *IEEE Wireless Communications*, 26(3):132–139.

ANEXOS

Anexo A: Datos de las pruebas experimentales

- Fragmento de datos “Sin interferencia” SNR y RSSI

epoch	snr_db	frame_id	cfo	sto	sf	bw
1762211470	10.2369	1	0.583083	7.059061	7	125000
1762211472	9.6203	2	0.586093	20.040064	7	125000
1762211474	7.4855	3	-0.414353	4.848683	7	125000
1762211476	8.2899	4	-0.415067	4.614792	7	125000
1762211478	9.7089	5	0.586168	120.09446	7	125000
1762211480	6.6402	6	-0.418087	5.822803	7	125000
1762211482	9.3791	7	-0.415149	-0.095097	7	125000
1762211484	8.1777	8	0.58751	0.351382	7	125000
1762211486	4.7599	9	0.583175	90.343124	7	125000
1762211488	10.3775	10	0.587389	51.491886	7	125000
1762211490	7.3956	11	0.588689	7.333232	7	125000
1762211492	9.4272	12	0.62798	-0.579166	7	125000
1762211494	7.5394	13	0.587591	120.347649	7	125000
1762211496	9.4472	14	0.629655	15.081852	7	125000
1762211498	6.5117	15	0.590119	107.315247	7	125000
1762211500	8.8906	16	-0.40838	6.623966	7	125000
1762211502	7.3567	17	-0.404877	11.843303	7	125000
1762211504	6.4188	18	0.592322	14.197657	7	125000
1762211506	10.393	19	0.597676	62.449543	7	125000
1762211508	8.6768	20	0.596146	37.399773	7	125000

1762211680	6.7666	106	1.497816	108.195297	8	125000
1762211682	9.8915	107	0.500078	193.581604	8	125000
1762211684	10.249	108	1.498725	158.059875	8	125000
1762211686	10.5597	109	1.498435	139.453491	8	125000
1762211688	7.9035	110	0.500974	173.353683	8	125000
1762211690	9.4802	111	1.504296	174.599274	8	125000
1762211694	10.6553	112	1.506634	8.057824	8	125000
1762211696	5.3934	113	1.287145	116.770287	8	125000
1762211698	6.8542	114	0.513497	206.824005	8	125000
1762211700	7.4775	115	1.293238	125.16066	8	125000
1762211702	6.947	116	1.292094	194.815674	8	125000
1762211704	10.1792	117	1.295481	136.936737	8	125000
1762211706	9.3769	118	1.293558	181.907013	8	125000
1762211708	7.7795	119	1.291494	149.348816	8	125000
1762211710	6.8111	120	1.295198	121.823975	8	125000
1762211712	10.4084	121	0.293248	78.51432	8	125000
1762211714	6.2538	122	1.29593	30.291378	8	125000
1762211716	8.7401	123	1.29555	160.375031	8	125000
1762211718	8.5363	124	1.290669	149.884949	8	125000
1762211720	9.7173	125	1.291125	179.596344	8	125000
1762211722	8.1514	126	1.291468	155.864288	8	125000

1762211959	8.9858	1	2.698438	415.613708	9	125000
1762211961	5.399	2	2.69833	493.273651	9	125000
1762211963	7.9611	3	2.703362	427.151947	9	125000
1762211965	5.3079	4	1.706496	21.739075	9	125000
1762211967	5.6756	5	2.706287	484.232422	9	125000
1762211969	10.4611	6	2.708079	405.571869	9	125000
1762211971	6.7564	7	2.712748	492.197662	9	125000
1762211973	8.644	8	1.710986	8.633959	9	125000
1762211975	10.472	9	2.705318	473.081268	9	125000
1762211977	8.5172	10	2.703156	478.146149	9	125000
1762211979	6.4752	11	2.701135	452.207214	9	125000
1762211981	8.1638	12	2.698171	503.148041	9	125000
1762211983	10.0955	13	1.69111	476.586426	9	125000
1762211985	10.3152	14	2.689305	471.436035	9	125000
1762211987	11.0081	15	2.690115	487.035187	9	125000
1762211989	9.2602	16	2.688364	484.110596	9	125000
1762211991	9.35	17	2.689416	442.900177	9	125000
1762211993	10.8661	18	1.689018	503.505188	9	125000
1762211995	6.5045	19	2.684644	483.814087	9	125000
1762211997	8.66	20	2.68699	479.37793	9	125000

1762212159	10.7144	101	5.085501	479.491882	10	125000
1762212161	10.1301	102	5.090879	102.96254	10	125000
1762212163	5.843	103	5.089339	99.302963	10	125000
1762212165	7.6198	104	5.079507	96.157593	10	125000
1762212167	9.188	105	5.070189	95.60952	10	125000
1762212169	6.1099	106	5.064232	97.709229	10	125000
1762212171	9.3692	107	5.057399	94.592598	10	125000
1762212173	10.1522	108	5.060169	101.959427	10	125000
1762212175	5.6765	109	5.058553	86.795311	10	125000
1762212177	10.1292	110	5.059219	103.001083	10	125000
1762212179	5.136	111	5.075525	95.278282	10	125000
1762212181	9.4222	112	5.081727	97.932953	10	125000
1762212183	10.118	113	5.085529	100.445854	10	125000
1762212185	9.8515	114	5.085738	91.56292	10	125000
1762212187	9.135	115	5.08277	81.412201	10	125000
1762212189	6.5661	116	5.074301	109.693398	10	125000
1762212191	10.4471	117	5.081591	94.549026	10	125000
1762212193	8.9789	118	5.082249	96.111496	10	125000
1762212195	8.3869	119	5.082316	103.646881	10	125000
1762212197	8.6907	120	5.0817	41.133301	10	125000
1762212199	8.1297	121	5.08515	149.874115	10	125000

1762212469	8.7367	37	10.111566	1115.382812	11	125000
1762212471	7.3433	38	10.12368	363.69281	11	125000
1762212473	6.3121	39	10.124928	350.812683	11	125000
1762212475	9.6438	40	10.118445	353.414032	11	125000
1762212477	11.1662	41	10.133883	350.052429	11	125000
1762212479	5.6885	42	10.128654	355.789124	11	125000
1762212481	5.045	43	10.138028	339.264221	11	125000
1762212483	9.8574	44	10.164577	340.425995	11	125000
1762212485	11.078	45	10.169333	357.56189	11	125000
1762212487	10.1002	46	10.180784	335.095428	11	125000
1762212489	8.7159	47	10.185339	305.882294	11	125000
1762212491	5.658	48	10.211065	409.246521	11	125000
1762212493	9.5855	49	10.222765	365.617432	11	125000
1762212495	8.4381	50	10.236134	340.877655	11	125000
1762212497	8.6907	51	10.235135	357.883667	11	125000
1762212499	11.1325	52	10.232022	338.565125	11	125000
1762212501	5.3254	53	11.934055	368.279602	11	125000
1762212503	5.6311	54	10.233927	345.24408	11	125000
1762212505	6.9407	55	11.949908	361.834473	11	125000
1762212507	8.0999	56	10.235441	337.168671	11	125000
1762212509	6.5379	57	11.942756	367.81955	11	125000

1762214208	9.7911	1	20.923082	4018.592773	12	125000
1762214210	5.0957	2	21.864847	892.300964	12	125000
1762214212	6.8354	3	21.839794	867.865051	12	125000
1762214214	7.7193	4	21.825823	866.897949	12	125000
1762214216	8.7581	5	21.839724	860.637695	12	125000
1762214218	9.7699	6	21.820415	861.485168	12	125000
1762214220	9.2131	7	21.853043	835.96106	12	125000
1762214222	7.4944	8	21.849789	892.175964	12	125000
1762214224	8.9869	9	21.866449	870.623291	12	125000
1762214226	9.3772	10	21.888548	864.960449	12	125000
1762214228	6.776	11	21.905325	855.709534	12	125000
1762214230	10.2457	12	21.894075	850.532227	12	125000
1762214236	7.4867	13	25.240721	3881.389648	12	125000
1762214238	9.9279	14	25.226271	790.993774	12	125000
1762214240	8.4725	15	25.267445	883.15979	12	125000
1762214242	8.903	16	24.254782	861.659485	12	125000
1762214244	8.1074	17	25.300243	870.693115	12	125000
1762214246	3.4893	18	23.979696	837.199341	12	125000
1762214248	10.4748	19	24.027847	874.556396	12	125000
1762214250	6.537	20	24.028278	860.358704	12	125000

epoch	rss_i_mean_db	rss_i_min_db	rss_i_max_db	RSSI_med
1762211470	-82.62	-125.26	-82.14	-80.762839
1762211472	-82.26	-83.03	-70.87	-80.762839
1762211474	-82.37	-83.1	-70.88	-80.762839
1762211476	-82.39	-83.12	-71.13	-80.762839
1762211478	-82.43	-83.06	-71.25	-80.762839
1762211480	-82.43	-83	-71	-80.762839
1762211482	-82.39	-83.01	-70.9	-80.762839
1762211484	-82.44	-83.08	-71.05	-80.762839
1762211486	-82.4	-83.04	-70.63	-80.762839
1762211488	-82.43	-83.16	-71.08	-80.762839
1762211490	-82.46	-83.08	-71.05	-80.762839
1762211492	-82.44	-83.06	-71.06	-80.762839
1762211494	-82.42	-83.09	-71.08	-80.762839
1762211496	-82.43	-83.05	-71.1	-80.762839
1762211498	-82.4	-83.05	-71.02	-80.762839
1762211500	-82.41	-83.06	-71.04	-80.762839
1762211502	-82.42	-83.05	-71.07	-80.762839
1762211504	-82.43	-83.07	-71	-80.762839
1762211506	-82.42	-83.05	-71.02	-80.762839
1762211508	-82.45	-83.1	-71.03	-80.762839

1762211680	-82.38	-83.19	-70.58	-80.762839
1762211682	-82.25	-83.16	-71.12	-80.762839
1762211684	-82.25	-83.05	-71.11	-80.762839
1762211686	-82.24	-83.1	-71.17	-80.762839
1762211688	-82.3	-83.17	-70.9	-80.762839
1762211690	-82.24	-83.07	-71.34	-80.762839
1762211692	-82.24	-83	-71.23	-80.762839
1762211694	-82.2	-83.02	-71.28	-80.762839
1762211696	-82.23	-83.03	-71.19	-80.762839
1762211698	-82.24	-83.02	-71.21	-80.762839
1762211700	-82.27	-83.11	-71.29	-80.762839
1762211702	-82.33	-83.23	-71.16	-80.762839
1762211704	-82.27	-83.06	-71.29	-80.762839
1762211706	-82.27	-83.07	-71.31	-80.762839
1762211708	-82.28	-83.11	-71.27	-80.762839
1762211710	-82.29	-83.14	-71.32	-80.762839
1762211712	-82.29	-83.13	-71.53	-80.762839
1762211714	-82.28	-83.04	-71.35	-80.762839
1762211716	-82.32	-83.08	-71.37	-80.762839
1762211718	-82.31	-83.13	-71.53	-80.762839
1762211720	-82.28	-83.04	-71.26	-80.762839

1762211959	-82.81	-125.26	-82.2	-80.762839
1762211961	-81.82	-83.1	-71.21	-80.762839
1762211963	-81.75	-82.96	-71.08	-80.762839
1762211965	-81.78	-83.13	-70.97	-80.762839
1762211967	-81.89	-83.21	-70.73	-80.762839
1762211969	-81.91	-83.22	-70.76	-80.762839
1762211971	-81.87	-83.21	-70.75	-80.762839
1762211973	-81.87	-83.11	-70.77	-80.762839
1762211975	-81.88	-83.2	-70.76	-80.762839
1762211977	-81.91	-83.17	-70.82	-80.762839
1762211979	-81.92	-83.3	-70.86	-80.762839
1762211981	-81.88	-83.24	-70.75	-80.762839
1762211983	-81.98	-83.25	-71.03	-80.762839
1762211985	-81.95	-83.16	-71.1	-80.762839
1762211987	-81.93	-83.25	-71.09	-80.762839
1762211989	-81.97	-83.16	-71.07	-80.762839
1762211991	-81.97	-83.19	-71.1	-80.762839
1762211993	-82.01	-83.34	-71.15	-80.762839
1762211995	-82.01	-83.28	-71.24	-80.762839
1762211997	-81.96	-83.22	-71.2	-80.762839
1762211999	-81.99	-83.26	-71.15	-80.762839

1762212159	-81.95	-83.28	-71.15	-80.762839
1762212161	-81.31	-83.17	-71.13	-80.762839
1762212163	-81.3	-83.08	-71.4	-80.762839
1762212165	-81.29	-83.13	-71.37	-80.762839
1762212167	-81.29	-83.03	-71.37	-80.762839
1762212169	-81.32	-83.1	-71.35	-80.762839
1762212171	-81.31	-83.15	-71.38	-80.762839
1762212173	-81.35	-83.14	-71.4	-80.762839
1762212175	-81.34	-83.11	-71.39	-80.762839
1762212177	-81.34	-83.13	-71.57	-80.762839
1762212179	-81.32	-83.1	-71.44	-80.762839
1762212181	-81.29	-83.12	-71.43	-80.762839
1762212183	-81.32	-83.17	-71.44	-80.762839
1762212185	-81.3	-83.15	-71.44	-80.762839
1762212187	-81.31	-83.16	-71.43	-80.762839
1762212189	-81.35	-83.2	-71.41	-80.762839
1762212191	-81.33	-83.13	-71.47	-80.762839
1762212193	-81.32	-83.1	-71.49	-80.762839
1762212195	-81.34	-83.21	-71.25	-80.762839
1762212197	-81.31	-83.09	-71.2	-80.762839
1762212199	-81.32	-83.23	-71.12	-80.762839

1762212469	-81.36	-83.38	-70.59	-80.762839
1762212471	-79.77	-83.27	-70.38	-80.762839
1762212473	-79.84	-83.24	-70.86	-80.762839
1762212475	-79.83	-83.22	-70.82	-80.762839
1762212477	-79.85	-83.22	-70.79	-80.762839
1762212479	-79.84	-83.22	-70.78	-80.762839
1762212481	-79.82	-83.27	-70.82	-80.762839
1762212483	-79.82	-83.19	-70.81	-80.762839
1762212485	-79.81	-83.16	-70.83	-80.762839
1762212487	-79.8	-83.13	-70.8	-80.762839
1762212489	-79.81	-83.2	-70.79	-80.762839
1762212491	-79.82	-83.17	-70.8	-80.762839
1762212493	-79.83	-83.13	-70.76	-80.762839
1762212495	-79.81	-83.17	-70.8	-80.762839
1762212497	-79.84	-83.19	-70.76	-80.762839
1762212499	-79.8	-83.2	-70.77	-80.762839
1762212501	-79.82	-83.15	-70.8	-80.762839
1762212503	-79.85	-83.28	-70.77	-80.762839
1762212505	-79.85	-83.14	-70.8	-80.762839
1762212507	-79.82	-83.15	-70.75	-80.762839
1762212509	-79.85	-83.15	-70.82	-80.762839

1762214207	-82.75	-110.14	-82.18	-80.762839
1762214209	-77.25	-83	-71.67	-80.762839
1762214211	-77.2	-82.94	-71.62	-80.762839
1762214213	-77.26	-82.9	-71.81	-80.762839
1762214215	-77.25	-82.93	-71.78	-80.762839
1762214217	-77.27	-82.92	-71.84	-80.762839
1762214219	-77.29	-82.94	-71.84	-80.762839
1762214221	-77.32	-83.03	-71.88	-80.762839
1762214223	-77.31	-83	-71.86	-80.762839
1762214225	-77.28	-82.93	-71.86	-80.762839
1762214227	-77.29	-82.98	-71.79	-80.762839
1762214229	-77.3	-83.04	-71.73	-80.762839
1762214231	-77.24	-83.22	-71.35	-80.762839
1762214233	-77.35	-83.09	-71.73	-80.762839
1762214235	-77.4	-83.09	-71.87	-80.762839
1762214237	-77.4	-83.12	-71.85	-80.762839
1762214239	-77.4	-83.13	-71.88	-80.762839
1762214241	-77.17	-83.19	-71.22	-80.762839
1762214243	-76.93	-83.25	-70.89	-80.762839
1762214245	-76.75	-83.01	-70.26	-80.762839
1762214247	-77.33	-83.02	-71.81	-80.762839

- Fragmento de datos “Con interferencia controlada” SNR y RSSI con Radio Cognitivo con EraSynth micro

epoch	SF	frame_id	SNR(dB)	RSSI_mean(dBm)	PER_total	interferencia	freq
1762889658	7	200	-2.6886	-65.76	0	-41	915.6
1762889660	7	201	-6.4601	-65.35	0	-41	915.6
1762889662	7	202	-8.268	-62.57	0	-38	915.6
1762889664	7	203	-9.0478	-62.27	0	-38	915.6
1762889666	7	204	-9.9343	-62.26	0	-38	915.6
1762889668	7	205	-9.424	-63.79	0	-38	915.6
1762889670	7	206	-9.1739	-62.76	0	-38	915.6
1762889672	7	207	11.5588	-83.07	0	N/A	915
1762889674	7	208	6.3347	-83.05	0	N/A	915
1762889676	7	209	11.4859	-83.06	0	N/A	915
1762889678	7	210	5.5507	-83.24	0	N/A	915
1762889704	7	223	-9.7096	-62.41	0	-38	915
1762889706	7	224	-10.1809	-62.42	0	-38	915
1762889708	7	225	-9.3021	-62.4	0	-38	915
1762889712	7	226	-9.6383	-62.45	0.004405	-38	915
1762889714	7	227	10.0045	-83.14	0.004386	NA	915.2
1762889716	7	228	7.2129	-83.11	0.004367	NA	915.2
1762889718	7	229	8.0472	-83.12	0.004348	NA	915.2
1762889832	7	284	-9.3943	-62.56	0.010453	-38	915.2
1762889836	7	285	-8.7027	-62.63	0.013841	-38	915.2
1762889838	7	286	-8.4525	-62.57	0.013793	-38	915.2
1762889840	7	287	-8.6903	-62.57	0.013746	-38	915.2
1762889844	7	288	-9.278	-62.58	0.017065	-38	915.2
1762889846	7	289	-8.986	-62.58	0.017007	-38	915.2
1762889848	7	290	-10.135	-62.57	0.016949	-38	915.2
1762889850	8	291	-8.4135	-62.57	0.016892	-38	915.2
1762889852	8	292	-9.7879	-62.55	0.016835	-38	915.2
1762889854	8	293	-9.0855	-62.66	0.016779	-38	915.2
1762889856	8	294	-6.947	-64.37	0.016722	-38	915.2
1762889858	8	295	-7.1593	-64.4	0.016667	-38	915.2
1762889984	8	350	-12.2641	-59.44	0.035813	-35	915.2
1762889986	8	351	-12.5824	-59.44	0.035714	-35	915.2
1762889988	8	352	-12.5735	-59.43	0.035616	-35	915.2
1762889990	8	353	-12.4187	-59.44	0.035519	-35	915.2
1762889992	8	354	11.3855	-82.95	0.035422	NA	915
1762889994	8	355	6.0602	-82.95	0.035326	NA	915
1762889996	8	356	12.2746	-83.08	0.03523	NA	915
1762889998	8	357	7.46	-83.02	0.035135	NA	915
1762890000	8	358	9.0508	-83.08	0.03504	NA	915
1762890002	8	359	6.8449	-82.97	0.034946	NA	915
1762890004	8	360	5.7451	-83.02	0.034853	NA	915
1762890006	8	361	7.1316	-82.99	0.034759	NA	915
1762890008	8	362	11.8831	-82.97	0.034667	NA	915

1762890010	8	363	10.2106	-82.99	0.034574	NA	915
1762890012	7	364	8.4499	-83.16	0.034483	NA	915
1762890014	7	365	8.3962	-83.19	0.034392	NA	915
1762890016	7	366	11.7077	-83.18	0.034301	NA	915
1762890018	7	367	10.1589	-83.25	0.034211	NA	915
1762890020	7	368	9.698	-83.13	0.034121	NA	915
1762890022	7	369	9.2236	-83.13	0.034031	NA	915

- Fragmento de datos “Con interferencia controlada” SNR y RSSI con Radio Cognitivo con Channel Model

epoch	snr_db	frame_id	sf	rss_i_db	ruido channel	per_total
1763042400	-10.1754	305	7	-64.36	0.0006	0.012945
1763042402	-10.2706	306	7	-64.35	0.0006	0.012903
1763042404	-9.7369	307	7	-64.34	0.0006	0.012862
1763042406	-10.1571	308	7	-64.35	0.0006	0.012821
1763042408	-8.0725	309	8	-64.35	0.0006	0.01278
1763042410	-9.5207	310	8	-64.35	0.0006	0.012739
1763042412	-11.4782	311	8	-64.35	0.0006	0.012698
1763042414	-9.9833	312	8	-64.36	0.0006	0.012658
1763042716	-10.4876	436	8	-61.92	0.0008	0.066381
1763042718	-10.6193	437	8	-61.89	0.0008	0.066239
1763042720	-13.21	438	8	-61.9	0.0008	0.066098
1763042722	-12.6719	439	8	-61.9	0.0008	0.065957
1763042724	-12.0896	440	9	-61.89	0.0008	0.065817
1763042726	-11.1717	441	9	-61.88	0.0008	0.065678
1763042728	-11.5014	442	9	-61.89	0.0008	0.065539
1763042730	-11.0751	443	9	-61.89	0.0008	0.065401
1763043166	-14.3414	620	9	-58.43	0.0012	0.094046
1763043168	-16.4622	621	9	-58.45	0.0012	0.093896
1763043178	-13.9759	622	9	-58.44	0.0012	0.098883
1763043184	-15.139	623	9	-58.44	0.0012	0.09127
1763043186	-14.1324	624	10	-58.43	0.0012	0.094881
1763043188	-14.845	625	10	-58.44	0.0012	0.095353
1763043190	-15.141	626	10	-58.42	0.0012	0.098795
1763043192	-14.5456	627	10	-58.43	0.0012	0.091638
1763048092	-17.6984	897	10	-57.04	0.0013	0.082439
1763048096	-16.8548	898	10	-57.04	0.0013	0.085714
1763048098	-17.5226	899	10	-57.04	0.0013	0.085294
1763048100	-17.1371	900	10	-57.03	0.0013	0.084884
1763048102	-16.5001	901	11	-57.03	0.0013	0.084483
1763048104	-16.0216	902	11	-57.02	0.0013	0.084091
1763048106	-16.2036	903	11	-57.04	0.0013	0.083708
1763048108	-15.866	904	11	-57.03	0.0013	0.073333
1763048632	-12.848	1088	11	-60.82	0.0009	0.030107
1763048634	-12.9025	1089	11	-60.81	0.0009	0.031227
1763048636	-12.5075	1090	11	-60.8	0.0009	0.031327
1763048638	-12.8437	1091	11	-60.81	0.0009	0.031477
1763048640	-12.0172	1092	10	-60.84	0.0009	0.031827
1763048642	-12.2097	1093	10	-60.84	0.0009	0.032847
1763048644	-11.9657	1094	10	-60.85	0.0009	0.031857
1763048646	-12.74	1095	10	-60.86	0.0009	0.031965
1763048968	-8.1703	1239	10	-65.91	0.0004	0.032896

1763048969	-7.4063	1240	10	-65.91	0.0004	0.032497
1763048971	-8.077	1241	10	-65.91	0.0004	0.034742
1763048973	-7.1761	1242	10	-65.91	0.0004	0.036659
1763048975	-9.2673	1243	9	-65.95	0.0004	0.036332
1763048977	-8.3851	1244	9	-65.94	0.0004	0.038562
1763048979	-7.5682	1245	9	-65.95	0.0004	0.038869
1763048981	-8.0689	1246	9	-65.95	0.0004	0.038995
1763049862	-7.8469	1359	9	-65.94	0.0002	0.039014
1763049864	-7.362	1360	9	-65.94	0.0002	0.039624
1763049866	-6.9205	1361	9	-65.94	0.0002	0.039714
1763049868	-7.4998	1362	9	-65.95	0.0002	0.040562
1763049870	-6.9794	1363	8	-65.97	0.0002	0.040562
1763049872	-6.9037	1364	8	-65.96	0.0002	0.040562
1763049874	-6.4391	1365	8	-65.97	0.0002	0.040562
1763049876	-7.541	1366	8	-65.97	0.0002	0.040662
1763050020	-3.345	1434	8	-70.18	0.0001	0.046512
1763050022	-4.154	1435	8	-70.16	0.0001	0.045977
1763050024	-3.3059	1436	8	-70.17	0.0001	0.045455
1763050026	-3.8969	1437	8	-70.2	0.0001	0.044944
1763050028	-3.0652	1438	7	-70.19	0.0001	0.044556
1763050030	-3.5387	1439	7	-70.2	0.0001	0.043956
1763050032	-3.7553	1440	7	-70.19	0.0001	0.043478
1763050034	-4.2915	1441	7	-70.19	0.0001	0.043011

Anexo B: Códigos para radio cognitivo

B.1 Modificaciones a frame_sync_impl — Receptor LoRa reconfigurable

El archivo frame_sync_impl.cc es el primer eslabón activo de la cadena de recepción LoRa. Su función es detectar el preámbulo de cada trama entrante —una secuencia de upchirps y downchirps— y, a partir de él, estimar y corregir los errores de temporización (STO) y de frecuencia (CFO). En la versión original del módulo gr-lora_sdr, el SF se fijaba en el constructor y no podía modificarse durante la ejecución.

Para habilitar la reconfiguración dinámica, se introdujo la función pública set_sf(int sf) y un puerto de mensajes asíncrono, además de un mecanismo síncrono basado en stream tags.

B.1.1 Puerto de mensajes para cambio asíncrono de SF (constructor)

El siguiente fragmento, ubicado en el constructor, registra el puerto de entrada 'set'. Cuando el bloque SfDeciderMin decide incrementar el SF, envía un diccionario {sf: <int>} que activa la lambda y llama a set_sf.

Segmento B.1.1 — Registro del puerto de mensajes 'set'

```
message_port_register_in(pmt::mp("set"));
set_msg_handler(pmt::mp("set"), [this](pmt::pmt_t msg) {
    if (!pmt::is_dict(msg)) return;
    pmt::pmt_t ksf = pmt::intern("sf");
    if (pmt::dict_has_key(msg, ksf)) {
        int sf_new = (int)pmt::to_long(
            pmt::dict_ref(msg, ksf, pmt::from_long(this->m_sf)));
        this->set_sf(sf_new);
        std::cout << "[frame_sync] set.sf -> " << sf_new << std::endl;
    }
});
```

B.1.2 Función set_sf — reconfiguración completa del receptor

La función recalcula el número de bins, redimensiona todos los buffers internos (upchirp, downchirp, preámbulo, corrección de CFO) y reconstruye el plan de FFT mediante kiss_fft_alloc con el nuevo tamaño.

Segmento B.1.2 — Núcleo de set_sf

```

void gr::lora_sdr::frame_sync_impl::set_sf(int sf) {
    if (sf < 7 || sf > 12) return;
    if ((int)m_sf == sf) return;
    m_sf = (uint8_t)sf;
    m_number_of_bins = (uint32_t)(1u << m_sf);
    m_samples_per_symbol = m_number_of_bins * m_os_factor;
    m_upchirp.resize(m_number_of_bins);
    m_downchirp.resize(m_number_of_bins);
    build_ref_chirps(&m_upchirp[0], &m_downchirp[0], m_sf);
    kiss_fft_free(m_kiss_fft_cfg);
    m_kiss_fft_cfg = kiss_fft_alloc(m_number_of_bins, 0, 0, 0);
    set_output_multiple(m_number_of_bins);
}

```

B.1.3 Reconfiguración síncrona por stream tags

La vía síncrona lee stream tags con la clave 'new_frame' embebidas en el flujo de muestras I/Q. Al detectar una etiqueta, el bloque lee el SF indicado y llama internamente a set_sf antes de procesar la trama, garantizando sincronía paquete a paquete entre transmisor y receptor.

Segmento B.1.3 — Lectura de stream tags en general_work

```

std::vector<tag_t> tags;
get_tags_in_window(tags, 0, 0, ninput_items[0],
    pmt::string_to_symbol("new_frame"));
if (tags.size()) {
    int sf = pmt::to_long(pmt::dict_ref(tags[0].value,
        pmt::string_to_symbol("sf"), err));
    set_sf(sf);
}

```

B.1.4 Cabecera frame_sync_impl.h — declaraciones clave

Se añade la declaración pública de set_sf, el identificador del puerto de salida snr_out, las nuevas variables miembro y los manejadores consolidados de mensajes.

Segmento B.1.4 — Constructor modificado y declaraciones

```

public:
    frame_sync_impl(uint32_t center_freq, uint32_t bandwidth, uint8_t sf,

```

```

        bool impl_head, std::vector<uint16_t> sync_word,
        uint8_t os_factor, uint16_t preamb_len);

    void set_sf(int sf) override;
private:
    pmt::pmt_t d_snr_port;
    uint8_t m_os_factor;
    std::vector<uint16_t> m_sync_words;
    void frame_info_handler(pmt::pmt_t frame_info);
    void noise_est_handler(pmt::pmt_t noise_est);

```

B.2 Modificaciones a modulate_impl — Transmisor LoRa reconfigurable

El bloque modulate es el equivalente en transmisión de frame_sync. En su versión original, la longitud del preámbulo ($n_{up} = 8$) y las sync words (8 y 16) estaban codificadas directamente. Para que el sistema cognitivo pueda cambiar el SF del transmisor de forma coordinada, se introdujo la función `update_var(int new_sf, int new_bw)` y el puerto de mensajes 'set'.

B.2.1 Control asíncrono por mensajes (constructor)

Segmento B.2.1 — Puerto de mensajes 'set' en modulate_impl

```

message_port_register_in(pmt::mp("set"));
set_msg_handler(pmt::mp("set"), [this] (pmt::pmt_t msg) {
    if (!pmt::is_dict(msg)) return;
    pmt::pmt_t ksf = pmt::intern("sf");
    if (pmt::dict_has_key(msg, ksf)) {
        int sf_new = (int)pmt::to_long(
            pmt::dict_ref(msg, ksf, pmt::from_long(this->m_sf)));
        this->set_sf(sf_new);
    }
});

```

B.2.2 Función update_var — reconfiguración del modulador

Encapsula toda la lógica de reconfiguración del modulador. Actualiza las variables derivadas, redimensiona los vectores de chirps y los regenera, garantizando que el modulador produzca señales coherentes con el nuevo SF antes de transmitir la siguiente trama.

Segmento B.2.2 — *update_var en modulate_impl.cc*

```
void modulate_impl::update_var(int new_sf, int new_bw) {  
  
    m_sf = new_sf;  
  
    m_bw = new_bw;  
  
    m_number_of_bins = (uint32_t)(1u << m_sf);  
  
    m_os_factor = m_samp_rate / m_bw;  
  
    m_samples_per_symbol = m_number_of_bins * m_os_factor;  
  
    m_downchirp.resize(m_samples_per_symbol);  
  
    m_upchirp.resize(m_samples_per_symbol);  
  
    build_ref_chirps(&m_upchirp[0], &m_downchirp[0], m_sf, m_os_factor);  
  
    set_output_multiple(m_samples_per_symbol);  
  
}
```

B.2.3 Reconfiguración síncrona por stream tags

Al detectar una etiqueta 'configuration' al final de una trama, el modulador actualiza sus parámetros para la siguiente, asegurando coherencia entre TX y RX.

Segmento B.2.3 — *Lectura de stream tag 'configuration'*

```
get_tags_in_window(tags, 0, 0, 1,  
  
    pmt::string_to_symbol("configuration"));  
  
if (tags.size() > 0) {  
  
    int new_sf = pmt::to_long(pmt::dict_ref(tags[0].value,  
        pmt::string_to_symbol("sf"), err_sf));  
  
    int new_bw = pmt::to_long(pmt::dict_ref(tags[0].value,  
        pmt::string_to_symbol("bw"), err_bw));  
  
    update_var(new_sf, new_bw);  
  
}
```

B.2.4 Cabecera modulate_impl.h — declaraciones clave

Segmento B.2.4 — *Constructor modificado y nuevas declaraciones*

```
public:  
  
    modulate_impl(uint8_t sf, uint32_t samp_rate, uint32_t bw,  
        std::vector<uint16_t> sync_words,  
        uint32_t frame_zero_padd,  
        uint16_t preamb_len);
```

```
    void set_sf(int sf) override;

    void update_var(int new_sf, int new_bw);
private:

    uint32_t m_number_of_bins;

    int m_os_factor;

    std::vector<uint16_t> m_sync_words;

    uint32_t m_inter_frame_padding;

    uint16_t m_preamb_len;
```

B.3 Bloque ED_CFAR_Scanner — Detección espectral adaptativa (Python)

Este bloque implementa la fase sentir del ciclo cognitivo. Analiza el espectro de la banda ISM de 915 MHz mediante el método de Welch y aplica un detector CA-CFAR para determinar qué frecuencias se encuentran ocupadas por señales interferentes.

B.3.1 Parámetros de configuración utilizados en las pruebas

Parámetro	Valor	Descripción
nfft	4096	Puntos de FFT → resolución $\Delta f \approx 61$ Hz/bin
sense_ms	80 ms	Ventana de captura por ciclo de sensado
navg	12 (efectivo: 8)	Máx. de periodogramas; limitado por sense_ms
guard_cells	4	Celdas de guarda a cada lado de la CUT
train_cells	64	Celdas de entrenamiento (estimación del ruido)
pfa	0.05 (5 %)	Probabilidad de falsa alarma objetivo
β (factor umbral)	≈ 1.0237	Calculado como $pfa^{(-1/N)}$ con $N=128$

B.3.2 Lógica principal del detector CA-CFAR (Python)

El umbral adaptativo se calcula como $T_{umbral} = \beta \times \mu_{ruido}$, donde μ_{ruido} es la potencia media de las 128 celdas de entrenamiento. Esto garantiza $P_{fa} = 5 \%$ independientemente del nivel de ruido del canal.

Segmento B.3.2 — Cálculo del umbral CA-CFAR por bin

```
import numpy as np

def ca_cfar_threshold(psd, guard=4, train=64, pfa=0.05):
    N = 2 * train # total reference cells
    beta = pfa ** (-1.0 / N) # = 1.0237 for N=128, pfa=0.05
    thresholds = np.zeros_like(psd)
    for i in range(len(psd)):
        left = psd[max(0, i-guard-train) : max(0, i-guard)]
```

```
        right = psd[min(len(psd), i+guard+1) : min(len(psd),
i+guard+train+1)]

        ref_cells = np.concatenate([left, right])

        if len(ref_cells) == 0:

            thresholds[i] = 0

        else:

            noise_mean = np.mean(ref_cells)

            thresholds[i] = beta * noise_mean

    return thresholds

# Detección: bin i es 'ocupado' si psd[i] > threshold[i]
detections = psd > ca_cfar_threshold(psd)
```

B.4 Bloque SliceStepper — Barrido espectral por rejilla (Python)

El SliceStepper desplaza periódicamente la frecuencia central de escucha a través de cinco posiciones predefinidas de la banda ISM (915.2 / 915.4 / 915.6 / 915.8 / 916.0 MHz), con un paso de 200 kHz que coincide con el espaciado de canal LoRaWAN para la Región 2.

B.4.1 Frecuencias de barrido y temporización

Parámetro	Valor	Descripción
grid_step	200 kHz	Separación entre canales candidatos
Posiciones	5 canales	915.2 / 915.4 / 915.6 / 915.8 / 916.0 MHz
Permanencia/pos.	≥ sense_ms (80 ms) + procesamiento	Tiempo para un ciclo CFAR completo

Segmento B.4 — Lógica de barrido del SliceStepper (Python)

```
FREQ_GRID = [915.2e6, 915.4e6, 915.6e6, 915.8e6, 916.0e6] # Hz
grid_step = 200e3 # 200 kHz
sense_ms = 80 # ms por posición

def step_frequencies(usrp_source, scanner):
    for freq in FREQ_GRID:
        usrp_source.set_center_freq(freq) # resintoniza USRP
        time.sleep(sense_ms / 1000.0) # espera captura
        occupied = scanner.get_occupied_freqs() # resultado CA-CFAR
        yield freq, occupied # mapa de ocupación
```

B.5 Bloque ChannelPickerAndTune_SNR — Decisión y resintonización (Python)

Implementa la fase decidir para el canal de frecuencia. Combina el mapa de ocupación del escáner con la SNR suavizada (EMA) del enlace LoRa. Cuando detecta degradación sostenida, selecciona el canal alternativo más favorable y ordena la resintonización del hardware.

B.5.1 Parámetros de decisión

Parámetro	Valor	Descripción
snr_margin_db	3 dB	Margen de anticipación sobre SNR _{mín}
snr_need_consec	4 pkts	Paquetes consecutivos bajo umbral para confirmar
ema_tau_ms	300 ms	Constante de tiempo del filtro EMA
dwell_ms	800 ms	Tiempo mínimo en canal antes de re-evaluar
post_ms	800 ms	Bloqueo tras cambio (transitorio)
band_min/max	915.0–916.0 MHz	Rango de la rejilla de canales
grid_step	200 kHz	Separación entre canales candidatos
MIN_HOP_HZ	150 kHz	Salto mínimo (evita micro-saltos)
settling_ms	500 ms	Asentamiento HW tras resintonización

B.5.2 Umbral de activación por SF

El umbral que activa la búsqueda de canal alternativo es $\gamma_{\text{canal}}(\text{SF}) = \text{SNR}_{\text{mín}}(\text{SF}) + 3 \text{ dB}$. Para SF7, esto da -9.0 dB , valor coherente con los eventos de resintonización registrados en la Tabla 6 de la tesis.

Segmento B.5.2 — Cálculo del umbral y filtro EMA

```
SNR_MIN = {7: -12.0, 8: -15.0, 9: -18.0, 10: -20.0, 11: -23.0, 12: -27.0}

snr_margin_db    = 3.0
ema_tau_ms       = 300.0
delta_t_ms       = 2000.0 # periodo entre paquetes (Message Strobe)

import math

a = math.exp(-delta_t_ms / ema_tau_ms) # ≈ 0.001 → EMA ≈ muestra actual

def update_ema(snr_ema_prev, snr_new):
    return a * snr_ema_prev + (1 - a) * snr_new

def should_change_channel(snr_ema, current_sf):
```

```
gamma = SNR_MIN[current_sf] + snr_margin_db    # ej. -9.0 dB para SF7
return snr_ema < gamma
```

B.5.3 Selección del canal alternativo

Segmento B.5.3 — Selección del mejor canal libre

```
def pick_best_channel(current_freq, occupied_freqs, band_min=915.0e6,
                      band_max=916.0e6, step=200e3, min_hop=150e3):
    candidates = []
    f = band_min
    while f <= band_max + 1e3:
        if abs(f - current_freq) >= min_hop and f not in occupied_freqs:
            candidates.append(f)
        f += step
    if not candidates:
        return None    # no hay canal libre
    # Seleccionar el candidato más alejado del interferente
    return max(candidates, key=lambda c: abs(c - current_freq))
```

B.6 Bloque SfDeciderMin — Decisión del Factor de Ensanchamiento (Python)

Implementa la fase decidir para el SF. Actúa de forma complementaria al ChannelPickerAndTune_SNR, respondiendo a degradación generalizada (ruido de banda ancha) incrementando la robustez de la modulación. Utiliza histéresis para evitar oscilaciones (flapping) entre SFs.

B.6.1 Umbrales de deterioro y mejora

El bloque define dos umbrales para cada SF: γ_{malo} para subir de SF y γ_{bueno} para bajar. La zona de histéresis de 2 dB entre ambos previene el fenómeno de flapping.

Segmento B.6.1 — Cálculo de umbrales con histéresis

```
SNR_MIN = {7: -12.0, 8: -15.0, 9: -18.0, 10: -20.0, 11: -23.0, 12: -27.0}
snr_margin = 4.0    # dB - margen sobre SNRmín para umbral de deterioro
good_margin = 2.0    # dB - margen adicional para umbral de mejora
(histéresis)

def gamma_malo(sf):
```

```

    # Umbral para subir SF (ej. SF7 → SF8)

    return SNR_MIN[sf] + snr_margin    # = -12.0 + 4.0 = -8.0 dB para SF7

def gamma_bueno(sf):
    # Umbral para bajar SF (ej. SF8 → SF7), evaluado en SF objetivo

    if sf <= 7: return float('inf')    # no se puede bajar de SF7

    return SNR_MIN[sf-1] + snr_margin + good_margin    # = -6.0 dB (SF8→SF7)

# Histéresis: gamma_bueno - gamma_malo = 2.0 dB    (zona muerta)

```

B.6.2 Lógica de decisión con contadores de confirmación

Segmento B.6.2 — Máquina de estados del SfDeciderMin

```

need_bad    = 3    # paquetes consecutivos bajo gamma_malo para subir SF
need_good   = 2    # paquetes consecutivos sobre gamma_bueno para bajar SF
dwell_ms    = 1200 # estabilización tras cambio de SF

bad_counter = 0
good_counter = 0

def update_sf_decision(snr_ema, current_sf):
    global bad_counter, good_counter

    if snr_ema < gamma_malo(current_sf):
        bad_counter += 1
        good_counter = 0
        if bad_counter >= need_bad and current_sf < 12:
            bad_counter = 0
            return current_sf + 1    # subir SF
    elif snr_ema > gamma_bueno(current_sf):
        good_counter += 1
        bad_counter = 0
        if good_counter >= need_good and current_sf > 7:
            good_counter = 0
            return current_sf - 1    # bajar SF
    else:

```

```

        bad_counter = 0

        good_counter = 0

    return current_sf    # sin cambio

```

B.7 Bloque QtRangeSF — Sincronización del SF en el flowgraph (Python)

QtRangeSF actúa como punto único de despacho del valor de SF. Recibe mensajes del tipo {'sf': N} generados por el SfDeciderMin y los propaga a todos los objetos del flowgraph que implementen el método `set_sf()`, garantizando que transmisor (`modulate`) y receptor (`frame_sync`) se reconfiguren de forma simultánea y sin desincronización temporal.

B.7.1 Importancia del punto único de despacho

Si el cambio de SF se enviara directamente al transmisor y al receptor por separado, podría producirse una desincronización temporal en la que uno de los dos extremos utilizara un SF distinto durante una trama, causando pérdida de paquetes. QtRangeSF elimina este riesgo.

Segmento B.7 — Lógica de propagación del QtRangeSF

```

sf_min = 7
sf_max = 12
sf_init = 7

class QtRangeSF:
    def __init__(self, targets):
        # targets: lista de bloques con método set_sf()
        self.targets = targets    # [modulate_block, frame_sync_block]
        self.current_sf = sf_init

    def handle_sf_message(self, msg):
        # msg: pmt dict con clave 'sf'
        new_sf = pmt.to_long(pmt.dict_ref(msg, pmt.intern('sf'),
                                           pmt.from_long(self.current_sf)))

        if sf_min <= new_sf <= sf_max:
            self.current_sf = new_sf
            for target in self.targets:
                target.set_sf(new_sf)    # propagación simultánea

```

Anexo C: Especificaciones USRP 2920

Conditions

Specifications are valid at 25 °C unless otherwise noted.

Transmitter

Frequency range	50 MHz to 2.2 GHz
Frequency step	<1 kHz
Maximum output power (P_{out})	
50 MHz to 1.2 GHz	50 mW to 100 mW (17 dBm to 20 dBm)
1.2 GHz to 2.2 GHz	30 mW to 70 mW (15 dBm to 18 dBm)
Gain range ^[1]	0 dB to 31 dB
Gain step	1.0 dB
Frequency accuracy	2.5 ppm

Maximum instantaneous real-time bandwidth^[2]	
16-bit sample width	20 MHz
8-bit sample width	40 MHz
Maximum I/Q sample rate^[3]	
16-bit sample width	25 MS/s
8-bit sample width	50 MS/s
Digital-to-analog converter (DAC)	2 channels, 400 MS/s, 16 bit
DAC spurious-free dynamic range (SFDR)	80 dB

Receiver

Frequency range	50 MHz to 2.2 GHz
Frequency step	<1 kHz
Gain range ^[4]	0 dB to 31.5 dB
Gain step	0.5 dB
Maximum input power (P_{in})	0 dBm
Noise figure	5 dB to 7 dB
Frequency accuracy	2.5 ppm

Maximum instantaneous real-time bandwidth^[5]	
16-bit sample width	20 MHz
8-bit sample width	40 MHz
Maximum I/Q sample rate^[6]	
16-bit sample width	25 MS/s
8-bit sample width	50 MS/s

Analog-to-digital converter (ADC)	2 channels, 100 MS/s, 14 bit
ADC sFDR	88 dB

Power

Total power, typical operation	
Typical	12 W to 15 W
Maximum	18 W
Power requirement	Accepts a 6 V, 3 A external DC power connector



Note You must use either the power supply provided in the shipping kit, or another UL listed ITE power supply marked **LPS**, with the USRP-2920.

Physical Characteristics

Physical dimensions	
(L × W × H)	15.875 cm × 4.826 cm × 21.209 cm (6.25 in. × 1.9 in. × 8.35 in.)
Weight	1.193 kg (2.63 lb)

Environment

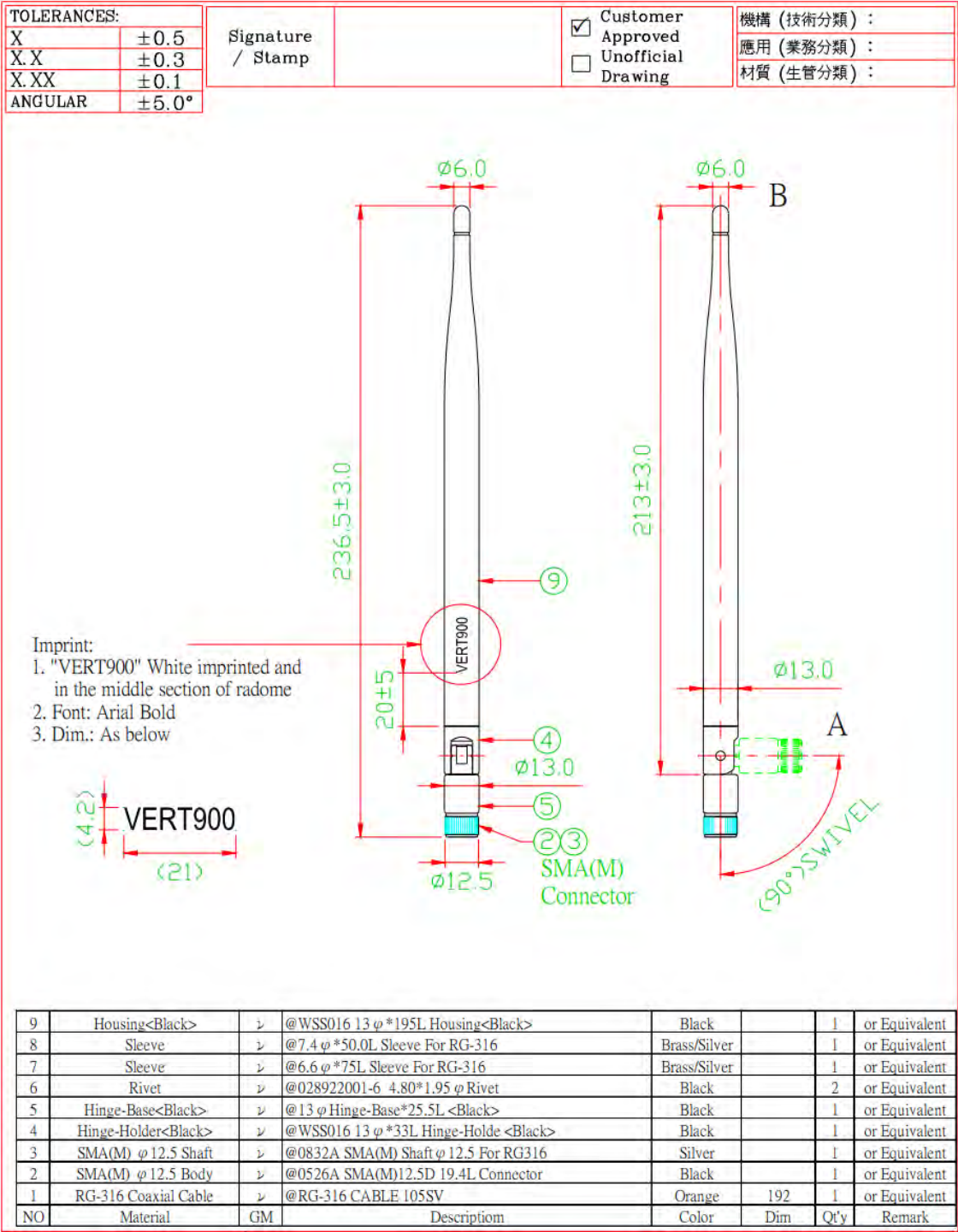
Maximum altitude	2,000 m (800 mbar) (at 25 °C ambient temperature)
Pollution Degree	2

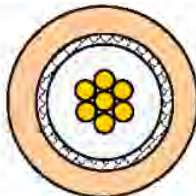
Indoor use only.

Operating Environment

Operating temperature	23 °C ± 5 °C
Relative humidity range	10% to 90%, noncondensing (tested in accordance with IEC 60068-2-56)

Anexo D: Datasheet Antena VERT900



STYLE	105°C 30V UL1979	DOCUMENT NO : A30316U007	
SIZE	RG-316/U	ESTABLISHED DATE: 2004/03/22	
STANDARD : MIL-C-17			
Conductor	Size	AWG	26
	Material	----	Silver-Coated Copper Clad Steel
	Conductors No.	----	7
	Conductors Size	mm	0.17
	O.D.	mm	0.51
Insulation	Average Thickness	mm	0.51
	Diameter	mm	1.53 ± 0.03
	Material	----	FEP
	Color	----	Clear
Braid	Material	----	Silver-Coated Copper
	Construction	mm	16 / 5 / 0.10
	Coverage	%	95
Jacket	Average Thickness	mm	0.30
	Diameter	mm	2.53 ± 0.10
	Material	----	FEP
	Color	----	Brown
Marking	M17/113-RG316 MIL-C-17 WONDERFUL		
Drawing			

Electrical & Physical Properties							
Item				RG-316/U			
Rating Temp Voltage				105°C 30V			
Conductor Resistance				302 OHM/KM/20°C MAX.			
Insulation Resistance				3000 MEGA OHM/KM MIN.			
Dielectric Strength				AC 1 KV/Minute			
Spark Test				5 KV			
Insulation	Unaged	Tensile Strength		2500 PSI MIN.(1.76 Kg / m m ²)			
		Elongation		200% MIN.			
	Aged	Tensile Strength		UNAGED MIN 75%(168HRS×232°C)			
		Elongation		UNAGED MIN 75%(168HRS×232°C)			
Jacket	Unaged	Tensile Strength		2500 PSI MIN.(1.76 Kg / m m ²)			
		Elongation		200% MIN.			
	Aged	Tensile Strength		UNAGED MIN.75%(168HRS×232°C)			
		Elongation		UNAGED MIN.75%(168HRS×232°C)			
Nom. Impedance				50 Ohms			
Nom. Capacitance				95.8 pF/m			
Nom. Vel. of Prop.				69.5%			
VSWR (0 – 6 GHZ)				UNDER 1.3			
Attenuation (dB/100m)	1MHZ	100MHZ	1.8GHZ	2.4GHZ	5.2GHZ	6GHZ	
	10.2	34.1	180	206	315	345	

Anexo E: Datasheet ERASynth Micro

www.erainstruments.com



GENERAL DESCRIPTION

ERASynth Micro is an open source analog RF signal generator featuring an Arduino Micro micro-controller and a touch display. ERASynth Micro uses advanced PLL/VCO technology, coupled with an internal ultra-low phase noise frequency reference to form a programmable analog signal generator capable of generating a low phase noise signals up to 6.4 GHz. ERASynth Micro provides fast frequency switching and fine-tuning resolution using a dual loop PLL architecture. ERASynth Micro also offers frequency, amplitude and pulse modulation capabilities. The frequency tuning and control commands are loaded into the instrument via the serial interface or via Touch Display or Windows GUI.



FEATURES

Architecture: Dual loop Fractional-N PLL driven by a 10 MHz reference.

Frequency Range: 12.5 MHz to 6.4 GHz

Amplitude Range: -50 to +15 dBm

Phase Noise: -115 dBc/Hz @ 1 GHz output and 10 kHz offset

Frequency Switching Time: <1ms

Reference: Ultra-low noise 50 MHz VCXO locked to a ± 500 ppb TCXO

MCU: ATMEGA32U4-MU (same as in the Arduino Micro board)

Interfaces:

- Touch Display Module (3.2" Nextion NX4024T032)
- Micro-USB for power input and serial access
- REF IN (SMA) for external reference input
- REF Out (SMA) for 10 MHz reference output
- RF Out (SMA)
- Expansion connector: External trigger input, external modulation input, microphone input, GPIO (I2C) and SPI

Dimensions: 57 mm x 118 mm x 23 mm

Weight: 120g

Power Input: 5V

Power Consumption: 2W

Enclosure: Injection molded plastic case

Open Source: Schematics, embedded Arduino code, LCD screen source code and RS-232 command set

Modulation: AM, FM, Pulse (Internal and external)

ELECTRICAL CHARACTERISTICS

	Minimum	Typical	Maximum
Supply Voltage	4.5 V	5 V	5.2 V
Supply Current		400 mA	
Supply Current, RF Out Muted		200 mA	
Minimum Output Power		-50 dBm	
Maximum Output Power		15 dBm	
Output Level Accuracy		± 1.5 dB	
Output Power Resolution		1 dB	
Frequency Resolution	1 Hz		
Frequency Accuracy		±500 ppb	
Minimum Dwell Time*	1 ms		
External Reference Input Level	-10 dBm	0 dBm	+10 dBm
External Reference Locking Range		10 MHz ± 30 ppm	
External Trigger Low Level Input Voltage	0 V		0.7 V
External Trigger High Level Input Voltage	2 V		3.3 V
External Modulation Input Voltage Level			± 2.5 V
RF Output Impedance		50 Ohm	
* Dwell time: Duration of each signal point in a sweep sequence set by user.			