


TRABAJO FIN DE MÁSTER

INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

# Sincronización entre UE y gNB en redes 5G

**Captura real del SIB9**

**Autor**
Ángel Gómez Hurtado

**Directores**
Jorge Navarro Ortiz
Félix Delgado Ferro




Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

—

Granada, julio de 2025

# Sincronización entre UE y gNB en redes 5G

**Captura real del SIB9.**

**Autor**
Ángel Gómez Hurtado

**Director**
Jorge Navarro Ortiz
Félix Delgado Ferro

# Sincronización entre UE y gNB en redes 5G: Captura real del SIB9

Ángel Gómez Hurtado

**Palabras clave**: 5G, sincronización, SIB9, reloj UTC, FN990

**Resumen**

La evolución de las redes móviles hacia Fifth Generation (5G) ha introducido requisitos más estrictos en cuanto a latencia, fiabilidad y, especialmente, sincronización temporal. En este contexto, el mensaje System Information Block (SIB)9 adquiere un papel fundamental al permitir que el User Equipment (UE) obtenga una referencia horaria absoluta proporcionada por la red, basada en el tiempo Coordinated Universal Time (UTC)/Global Positioning System (GPS). Este Trabajo de Fin de Máster (TFM) se centra en la obtención, análisis y validación del mensaje SIB9 en entornos reales, utilizando el módem comercial Telit FN990 y redes 5G tanto privadas (Amarisoft) como comerciales.

Se detalla el diseño e implementación de un entorno experimental que permite capturar y procesar el SIB9 por parte de un UE, combinando herramientas como comandos *Attention Command* (AT) y programación en C. Se presentan los resultados obtenidos en distintos escenarios, analizando la precisión y estabilidad temporal de los datos extraídos. El trabajo demuestra que es posible sincronizar el reloj interno del dispositivo sin necesidad de módulos Global Navigation Satellite System (GNSS), lo cual abre la puerta a aplicaciones sensibles al tiempo como redes Time-Sensitive Networking (TSN), IoT industrial o comunicaciones vehiculares.

# Synchronization between UE and gNB in 5G networks: Captura real del SIB9

Ángel Gómez Hurtado

**Keywords**: 5G, synchronization, SIB9, UTC clock, FN990

**Abstract**

The evolution of mobile networks towards 5G has introduced stricter requirements in terms of latency, reliability, and especially time synchronization. In this context, the message SIB9 plays a key role by allowing the UE to obtain an absolute time reference provided by the network, based on UTC/GPS time. This Master's Thesis focuses on the acquisition, analysis, and validation of the SIB9 message in real environments, using the commercial Telit FN990 modem and both private (*Amarisoft*) and commercial 5G networks.

This work describes the design and implementation of an experimental setup that allows capturing and processing the SIB9 in the UE, combining tools such as AT and C programming. The results obtained in different scenarios are presented, analyzing the temporal precision and stability of the extracted data. The study demonstrates that it is possible to synchronize the internal clock of the device without requiring GNSS modules, which opens the door to time-sensitive applications such as TSN networks, industrial Internet of Things (IoT), or vehicular communications.

Yo, **Ángel Gómez Hurtado**, alumno de la titulación Master Universitario en Ingeniería de Telecomunicación de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77770549M, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Máster en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Ángel Gómez Hurtado

Granada a julio de 2025 .

D. **Jorge Navarro Ortiz**, Profesor del Área de Ingeniería Telemática del Departamento Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada.

D. **Félix Delgado Ferro**, estudiante de doctorado del Área de Ingeniería Telemática del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada.

**Informan:**

Que el presente trabajo, titulado ***Sincronización entre UE y gNB en redes 5G, Captura real del SIB9*** , ha sido realizado bajo su supervisión por **Ángel Gómez Hurtado**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a julio de 2025.

**Los directores:**

**Jorge Navarro Ortiz** **Félix Delgado Ferro**

# Agradecimientos

Hace dos años tomé la decisión de realizar este máster que hoy termino. Una decisión adecuada ahora que lo puedo ver con la perspectiva de saber que me ha aportado. No solo se enseña conocimiento técnico, también se aprenden otro tipo de habilidades: la capacidad de análisis y resolución de problemas, el pensamiento crítico, la comunicación técnica y la capacidad de adaptarse y de aprendizaje continuo. Todas ellas son competencias que he tenido que poner en práctica, y que considero inherentes a quien aspire a ser llamado ingeniero. Tras la consecucion de este trabajo, me gustaría agradecer a:

Mi tutor Jorge Navarro y mi co-tutor Felix Delgado, por todas las horas dedicadas en reuniones, laboratorio y correcciones. Por querer hacer de este trabajo un mejor trabajo siempre.

Mi familia, piedra angular, por haber confiado en mí y en mi trabajo, dejándome trabajar como mejor sé. Cuidando de mí y haciendo que el camino haya sido lo más liviano posible.

# Índice general

**Glosario** **1**

**1. Introducción** **5**
1.1. Motivación . . . . . 6
1.2. Objetivos . . . . . 7
1.3. Estructura del documento . . . . . 8

**2. Estado del arte** **11**

**3. Fundamento Teórico** **15**
3.1. Canal y señal . . . . . 15
3.1.1. Enlace ascendente . . . . . 15
3.1.2. Enlace descendente . . . . . 16
3.2. Estructura de una trama en 5G . . . . . 19
3.2.1. Casos de configuración de SS Block . . . . . 20
3.2.2. Ejemplo: Caso A para $f_c < 3$ GHz . . . . . 21
3.3. Canales implicados en la obtención del mensaje SIB9 . . . . . 22
3.3.1. PBCH . . . . . 22
3.3.2. PDCCH . . . . . 23
3.3.3. PDSCH . . . . . 24
3.4. Mensajes implicados en la obtención del SIB9. . . . . 24
3.4.1. MIB . . . . . 24
3.4.2. Mensajes SIBs . . . . . 26
3.5. Sincronización en 5G . . . . . 38
3.5.1. SS/PBCH BLOCKS AND BURSTS . . . . . 39

**4. Planificación y estimación de costes** **41**
4.1. Planificación de tareas . . . . . 41
4.2. Planificación de recursos . . . . . 45
4.2.1. Recursos hardware. . . . . 45
4.2.2. Recursos software. . . . . 45
4.2.3. Recursos humanos. . . . . 45
4.3. Presupuesto . . . . . 45

**5. Herramientas usadas. 49**
5.1. Telit EVB 2.0 . . . . . 49
5.2. Telit FN990 . . . . . 49
5.3. Ubuntu/Linux . . . . . 51
5.4. Amarisoft . . . . . 51

**6. Preparación del entorno de pruebas 53**
6.1. Diseño del entorno de pruebas . . . . . 53
6.1.1. Configuración y puesta en marcha . . . . . 54
6.2. Interacción con el módem y comandos AT . . . . . 54
6.2.1. Conexión a la red 5G . . . . . 58

**7. Resultados y análisis 63**
7.1. Red privada con estación base Amarisoft . . . . . 63
7.1.1. Configuración de Amarisoft . . . . . 63
7.1.2. Resultados . . . . . 64
7.2. Red móvil de un operador comercial . . . . . 68

**8. Conclusiones 71**
8.1. Conclusiones . . . . . 71
8.2. Trabajos futuros . . . . . 72

# Índice de figuras


3.1. Mapa de canales y señales para Up-link (UL). . . . . 16
3.2. Mapa de canales y señales para Down-link (DL). . . . . 17
3.3. Canales y señales contenidos en el canal PBCH. . . . . 18
3.4. Canales y señales contenidos en el canal PDSCH. . . . . 18
3.5. Estructura de trama en 5G New Radio (NR) según el valor de $\mu$. . . . . 20
3.6. Posicionamiento de los bloques SS en los símbolos OFDM para el caso A. . . . . 21
3.7. Opciones de mapeado de la señal DMRS en el PBCH para pci mód 4. . . . . 23
3.8. Compensación Bloques de Recursos. . . . . 28
3.9. Compensación Bloques de Recursos en espectrograma. . . . . 28
3.10. Duración del canal tanto en símbolos como en frecuencia. . . 29
3.11. Estructura general del Synchronization Signal (SS)/Physical Broadcast Channel (PBCH) Block. . . . . 38
3.12. Estructura general del SS/PBCH Block. . . . . 39
3.13. Ejemplo de asignación de Physical Layer Cell Identity (PCI) mod 3. . . . . 40

4.1. Diagrama de Gantt. . . . . 44

5.1. Módulo Telit FN990 utilizado. . . . . 50
5.2. Estación base Amarisoft. . . . . 52

6.1. Versiones del kernel y de Ubuntu usados en la realización de este proyecto. . . . . 55
6.2. Comprobaciones iniciales mediante comandos AT tras acceder por `minicom`. . . . . 56
6.3. Estado inicial del módem conectado a una red Long Term Evolution (LTE). . . . . 56
6.4. Ejemplo de salida del comando `AT#SERVINFO` según la documentación oficial del fabricante. . . . . 57
6.5. Estado de la conexión a la red 5G. . . . . 58

6.6. Estado del módem registrado en red NR, sin sesión de datos activa. . . . . 59
6.7. Estado del módem conectado en red NR con sesión de datos activa. . . . . 60
6.8. Captura 1 logs de Amarisoft. . . . . 61
6.9. Captura 2 logs de Amarisoft. . . . . 61

7.1. Capturas SIB9 desde UE. . . . . 64
7.2. Función de python para la conversión del tiempo UTC a horario legible. . . . . 65
7.3. Programa de python para el análisis del registro de SIB9 obtenido por el UTC. . . . . 65
7.4. Código en C para actualizar el reloj interno en cuanto se recibe el SIB9. . . . . 68

# Índice de cuadros


3.1. Valor $\mu$ en función de la separación entre subportadoras . . . 19
3.2. Duración del símbolo Orthogonal Frequency-Division Multiplexing (OFDM) en función de la separación entre subportadoras . . . 19
3.3. Resumen de estructura de trama según $\mu$ . . . 19
3.4. Características de los casos según separación entre subportadoras . . . 21
3.5. Estructura del Master Information Block (MIB) . . . 25
3.6. Tabla configuración COntrol REsource SET (CORESET) para separación entre subportadoras 15 KHz en SS/PBCH y del Physical Down-Link Control Channel (PDCCH). . . . 27
3.7. Tabla configuración CORESET para separación entre subportadoras 30 KHz en SS/PBCH y del PDCCH. . . . 29
3.8. Tabla valor *searchSpaceZero* para FR1 y patrón de multiplexación 1. . . . 30
3.9. Contenido SIB1. . . . 31
3.10. Parámetro si-Schedulingltifo. . . . 32
3.11. Parámetro servingCellConfigCommon Parte 1. . . . 33
3.12. Parámetro servingCellConfigCommon Parte 2. . . . 34
3.13. Parámetro servingCellConfigCommon Parte 3. . . . 35
3.14. Parametros uac-BarringInfo. . . . 36

4.1. Tabla resumen de las horas dedicadas al TFM. . . . 43
4.2. Características del portátil. . . . 45
4.3. Recursos humanos necesarios para la elaboración del TFM. . . 45
4.4. Estimación del coste proporcional de uso de cada componente. 46
4.5. Presupuesto final. . . . 46

7.1. Resultados parciales de sincronización temporal obtenidos a partir del SIB9 . . . 66

# Glosario

**1G** First Generation.

**3G** Third Generation.

**3GPP** Third Generation Partnership Project.

**4G** Fourth Generation.

**5G** Fifth Generation.

**5GNR** 5G New Radio.

**AT** *Attention Command*.

**BCCH** Broadcast Control Channel.

**BCH** Broadcast Channel.

**BS** Base Station.

**CCCH** Common Control Channel.

**CORESET** COntrol REsource SET.

**CRC** Cyclic Redundancy Check.

**CSIRS** Channel State Information Reference Signal.

**DCCH** Dedicated Control Channel.

**DCI** Down-link Control Information.

**DL** Down-link.

**DMRS** Demodulation Reference Signal.

**DTCH** Dedicated Traffic Channel.

**eMBB** enhanced Mobile Broadband.

**EVB** Evaluation Board.

**FPGA** Field-Programmable Gate Array.

**FR1** Frequency Range 1.

**FR2** Frequency Range 2.

**GNSS** Global Navigation Satellite System.

**GPIO** General Purpose Input/Output.

**GPS** Global Positioning System.

**IEEE** Institute of Electrical and Electronics Engineers.

**IoT** Internet of Things.

**LTE** Long Term Evolution.

**MAC** Medium Access Control.

**MIB** Master Information Block.

**mMTC** massive Machine Type Communication.

**NDA** Non-Disclosure Agreement.

**NR** New Radio.

**OFDM** Orthogonal Frequency-Division Multiplexing.

**OTA** Over The Air.

**PBCH** Physical Broadcast Channel.

**PCCH** Paging Control Channel.

**PCH** Paging Channel.

**PCI** Physical Layer Cell Identity.

**PDCCH** Physical Down-Link Control Channel.

**PDSCH** Physical Down-Link Shared Channel.

**PLMN** Public Land Mobile Network.

**PPS** Pulse Per Second.

**PRACH** Physical Random Access Channel.

**PSS** Primary Synchronisation Signal.

**PTP** Precision Time Protocol.

**PTRS** Phase Tracking Reference Signal.

**PUCCH** Physical Up-link Control Channel.

**PUSCH** Physical Up-link Shared Channel.

**QAM** Quadrature Amplitude Modulation.

**QCL** Quasi Co-Location.

**QPSK** Quadrature Phase-Shift Keying.

**RACH** Random Access Channel.

**RAN** Radio Access Network.

**Rel.** Release.

**RLC** Radio Link Control.

**RNTI** Radio Network Temporary Identifier.

**RRC** Radio Resource Control.

**SA** Standalone.

**SCH** Shared Channel.

**SFN** Sytem Frame Number.

**SIB** System Information Block.

**SIM** Subscriber Identity Module.

**SRB** Signalling Radio Bearer.

**SRS** Sounding Reference Signal.

**SS** Synchronization Signal.

**SSS** Secondary Synchronization Signal.

**TFG** Trabajo de Fin de Grado.

**TFM** Trabajo de Fin de Máster.

**TRS** Tracking Reference Signal.

**TSN** Time-Sensitive Networking.

**UAC** Unified Access Control.

**UART** Universal Asynchronous Receiver-Transmitter.

**UCI** Up-link Control Information.

**UE** User Equipment.

**UL** Up-link.

**URLLC** Ultra Reliable Low Latency Communication.

**USB** Universal Serial Bus.

**UTC** Coordinated Universal Time.

# Capítulo 1

# Introducción

Desde la aparición de las primeras generaciones de redes móviles (First Generation (1G)), las comunicaciones han evolucionado no solo en capacidad, velocidad y cobertura, sino también en complejidad y exigencias funcionales. A medida que el número de dispositivos conectados se ha multiplicado y las aplicaciones se han vuelto más sofisticadas, ha surgido una necesidad creciente de que estos dispositivos no solo se comuniquen, sino que lo hagan de forma coordinada y precisa en el tiempo. En este contexto, la sincronización temporal ha adquirido una relevancia clave.

En generaciones anteriores como Third Generation (3G) o Fourth Generation (4G), los mecanismos de sincronización eran suficientes para los servicios tradicionales como voz, mensajería o acceso a Internet, pero se quedaban cortos ante requisitos más estrictos. En muchos casos, la sincronización se limitaba a nivel de capa física para permitir la correcta transmisión y recepción de datos. Sin embargo, con la llegada de Fifth Generation (5G) y la aparición de nuevos casos de uso como la Industria4.0, la automatización en tiempo real o las redes deterministas, esta sincronización ya no puede depender exclusivamente de las señales físicas de la red.

La tecnología 5G se ha diseñado para responder a un abanico muy amplio de necesidades. Su arquitectura flexible y su uso de tecnologías como el acceso por ondas milimétricas, el beamforming o el *network slicing* permiten soportar distintos tipos de servicios simultáneamente. Entre sus principales características destacan la capacidad para conectar un número masivo de dispositivos (massive Machine Type Communication (mMTC)), velocidades de transmisión muy superiores a las de 4G (enhanced Mobile Broadband (eMBB)) y comunicaciones con latencia ultra baja y alta fiabilidad (Ultra Reliable Low Latency Communication (URLLC)). Esto hace posible aplicaciones como la realidad aumentada, el control remoto de maquinaria pesada, la cirugía a distancia o la gestión inteligente de ciudades.

En este ecosistema, la sincronización temporal precisa no es solo una mejora, sino un requisito. Para poder garantizar que múltiples dispositivos

trabajen en conjunto dentro de una red con mínima latencia y sin interferencias, es necesario que todos estén alineados en el tiempo con márgenes de error extremadamente pequeños. Esta necesidad afecta especialmente a escenarios industriales, donde un fallo de sincronización puede traducirse en errores de producción, colisiones de señales o incluso fallos de seguridad.

5G introduce mejoras clave en este aspecto, permitiendo una sincronización mucho más precisa gracias a la combinación de técnicas físicas y mecanismos de capa superior. Entre estos mecanismos, destaca el System Information Block (SIB)9, un mensaje que la red transmite con información de sincronización basada en Coordinated Universal Time (UTC)/Global Positioning System (GPS). Este mensaje contiene campos como `timeInfoUTC`, que proporciona una referencia temporal absoluta, y otros campos auxiliares como `leapSeconds` o `localTimeOffset`, que ayudan al dispositivo a interpretar y adaptar esa hora a su contexto local.

Lo fundamental del SIB9 no es que informe sobre el horario local o el cambio de hora, sino que ofrece al dispositivo la capacidad de alinearse con la hora de red con una precisión de microsegundos o incluso nanosegundos, dependiendo de la versión (*release*) del estándar Third Generation Partnership Project (3GPP) y la implementación. Esta capacidad permite construir redes mucho más deterministas, algo esencial para servicios sensibles al tiempo como la conducción autónoma, los sistemas industriales sincronizados o las redes Time-Sensitive Networking (TSN).

En resumen, el SIB9 representa un avance significativo en el objetivo de lograr sincronización precisa en redes móviles. Es el puente entre la sincronización física inicial y una alineación temporal profunda que permite a los dispositivos operar como nodos coordinados dentro de una infraestructura crítica. Su papel en la arquitectura de 5G lo convierte en una pieza clave para el futuro de las comunicaciones móviles ultra fiables y deterministas.

Este Trabajo de Fin de Máster (TFM) da continuidad a los estudios que realicé previamente en el Trabajo de Fin de Grado (TFG) [1], en el cual se abordó el proceso de captura y decodificación de las señales de sincronización en redes 5G. Aquel trabajo supuso una base sólida para la comprensión de los mensajes MIB y SIB1, necesarios para el acceso al SIB9, que ahora se profundiza y amplía en este documento.

## 1.1. Motivación

En los últimos años, el número de dispositivos conectados ha crecido exponencialmente en múltiples sectores: desde el transporte inteligente hasta las infraestructuras industriales, pasando por servicios críticos como la sanidad o la energía. Esta transformación ha venido acompañada de nuevas exigencias en cuanto a latencia, fiabilidad, movilidad y sincronización temporal. En particular, la Industria4.0 impone requisitos muy estrictos

de coordinación temporal entre dispositivos distribuidos geográficamente, donde márgenes de error inferiores a un microsegundo pueden marcar la diferencia entre el funcionamiento correcto de un sistema y un fallo crítico.

Para dar respuesta a estas necesidades, 5G introduce capacidades avanzadas que permiten comunicaciones ultra fiables y con baja latencia (URLLC), además de un control temporal más preciso gracias a mecanismos como el SIB9. Este mensaje permite a los dispositivos sincronizarse con la hora de red basada en UTC/GPS sin necesidad de sistemas Global Navigation Satellite System (GNSS) externos, lo que lo convierte en una herramienta fundamental para escenarios industriales, redes TSN o entornos sin cobertura satelital.

Sin embargo, acceder a esta información desde dispositivos reales no es inmediato. Los fabricantes no suelen ofrecer interfaces completamente abiertas, y el acceso a funciones como la obtención del SIB9 puede requerir un conocimiento profundo del hardware y del software de control. Hasta donde se tiene conocimiento al inicio de este TFM, únicamente existen dos módems comerciales, ambos del fabricante Telit, que permiten la obtención del SIB9. Por tanto, resulta de gran interés explorar de forma práctica cómo acceder a esta información utilizando un equipo comercial como el módem Telit FN990, lo que facilitaría su integración en aplicaciones sensibles al tiempo.

Este trabajo se enmarca dentro de la temática del proyecto nacional *6G-CHRONOS* [2], liderado por el grupo de investigación *WiMuNet*, cuyo objetivo es avanzar hacia redes móviles deterministas, confiables y plenamente integradas con redes TSN. La obtención fiable y precisa del SIB9 constituye un paso esencial hacia este objetivo.

## 1.2. Objetivos

El objetivo principal de este trabajo es lograr la sincronización precisa de un móvil (User Equipment (UE)) con la red 5G. Para ello, es necesario acceder al mensaje SIB9, que contiene la información horaria proporcionada por la red. En este estudio, se explora cómo obtener dicho mensaje utilizando un módem comercial 5G, concretamente el Telit FN990, conectado a redes reales. Si bien este módem no es la única opción posible, se emplea por ser uno de los pocos dispositivos comerciales que permite acceder al SIB9 de forma directa. Esta información puede ser utilizada posteriormente para sincronizar el reloj interno del dispositivo sin necesidad de módulos GNSS.

Para alcanzar este objetivo general, se plantean los siguientes objetivos específicos:

- Estudiar el contenido, formato y relevancia del mensaje SIB9 en redes 5G, según las especificaciones del 3GPP.

- Analizar la documentación técnica del módem Telit FN990 para identificar los mecanismos disponibles que permiten acceder a la información del sistema.
- Configurar un entorno experimental basado en dos redes diferentes: una red 5G comercial y una red privada implementada con Amarisoft.
- Utilizar comandos *Attention Command* (AT) y funciones avanzadas a nivel de software (en lenguaje `C`) para interactuar con el módem y obtener la información de tiempo obtenida con el mensaje SIB9.
- Validar la precisión y estabilidad de la información temporal obtenida desde ambos entornos.

## 1.3. Estructura del documento

En esta sección se explican brevemente en qué consisten cada una de las partes que componen este Trabajo de Fin de Máster.

1. **Capítulo 1: Introducción**. En este capítulo se contextualiza el trabajo, se expone la importancia de la sincronización temporal en redes 5G y se presenta el papel del mensaje SIB9.
   *a*) Motivación. Se justifica el interés del estudio en relación con las aplicaciones sensibles al tiempo.
   *b*) Objetivos. Se definen los objetivos generales y específicos que se persiguen durante el proyecto.
   *c*) Estructura del documento. Se muestra la organización del mismo.
2. **Capítulo 2: Estado del arte**. En este capítulo se recopilan estudios, herramientas y proyectos relacionados con la sincronización en redes móviles, con especial énfasis en las capacidades del SIB9 y su uso práctico.
3. **Capítulo 3: Fundamento teórico**. En este capítulo se describe la base teórica necesaria para comprender los elementos involucrados en la sincronización 5G y en la transmisión del SIB9.
   *a*) Canal y señal. Se explica la estructura del canal de transmisión en 5G, incluyendo los enlaces ascendente y descendente.
   *b*) Estructura de una trama en 5G. Se presentan las configuraciones posibles del SS Block y cómo se organiza la señal a nivel temporal.
   *c*) Canales implicados en la obtención del mensaje System Information Block (SIB9). Se describen los canales físicos y lógicos necesarios para su transmisión: Physical Broadcast Channel (PBCH),

Physical Down-Link Control Channel (PDCCH) y Physical Down-Link Shared Channel (PDSCH).

*d*) Mensajes implicados en la obtención del SIB9. Se detallan los bloques previos como el Master Information Block (MIB) y otros SIBs que son necesarios para poder acceder al SIB9.

*e*) Sincronización en 5G. Se analiza cómo se realiza la sincronización entre el UE y la Base Station (BS) y el papel de cada señal en dicho proceso.

4. **Capítulo 4: Planificación y estimación de costes**. Se detalla la planificación temporal del proyecto, los recursos necesarios y la estimación económica asociada.

   *a*) Planificación de tareas. Se desglosan las tareas principales y su distribución en el tiempo.

   *b*) Planificación de recursos. Se enumeran los recursos hardware, software y humanos utilizados.

   *c*) Presupuesto. Se estima el coste total del proyecto.

5. **Capítulo 5: Herramientas usadas**. Se describen los elementos hardware y software utilizados para llevar a cabo la implementación y las pruebas. Se incluyen detalles sobre la tarjeta de desarrollo Telit EVB 2.0, el módem FN990, el entorno Ubuntu/Linux y la estacion base de Amarisoft.

6. **Capítulo 6: Preparación del entorno de pruebas**. En este capítulo se expone cómo se ha construido el entorno experimental necesario para realizar la extracción del SIB9.

   *a*) Diseño del entorno de pruebas. Se describen las redes utilizadas y la arquitectura general del sistema.

   *b*) Configuración y puesta en marcha. Se explica cómo se ha inicializado el sistema y configurado cada elemento.

   *c*) Interacción con el módem y comandos AT. Se detalla cómo se ha accedido al SIB9 mediante comandos AT y código en `C`.

   *d*) Conexión a la red 5G. Se describen los procedimientos realizados para lograr la conexión y obtener datos útiles.

7. **Capítulo 7: Resultados y análisis**. Se presentan los resultados obtenidos en los diferentes entornos y se analiza la validez de la información temporal recuperada, así como su utilidad práctica.

8. **Capítulo 8: Conclusiones y trabajos futuros**. En este último capítulo se resumen las principales conclusiones del trabajo y se proponen posibles líneas de continuación y mejora.

# Capítulo 2

# Estado del arte

En este capítulo, se examinará el estado del arte sobre la sincronización temporal en redes 5G New Radio (5GNR) mediante la distribución de tiempo de día a través de la interfaz aérea. Este aspecto adquiere especial relevancia en aplicaciones industriales, redes privadas y escenarios de comunicaciones críticas, donde la precisión y fiabilidad de la sincronización son fundamentales. Se analizarán algunas de las investigaciones más recientes y relevantes en torno al uso del SIB9, un mecanismo introducido en 3GPP Release 16 para la provisión de información temporal precisa (UTC o GPS) al UE desde la red de acceso.

El objetivo de este análisis es ofrecer una visión global sobre los avances tecnológicos alcanzados en este ámbito, incluyendo estudios teóricos, pruebas de laboratorio y validaciones en entornos reales. Asimismo, se explorarán los distintos enfoques empleados para distribuir tiempo Over The Air (OTA), sus resultados experimentales, y su comparación con métodos tradicionales como GNSS o Precision Time Protocol (PTP). De este modo, se establece una base sólida de conocimiento que permitirá contextualizar y sustentar el desarrollo del presente trabajo. A continuación, se describen los principales trabajos encontrados en la literatura reciente que abordan el uso de SIB9 y otras soluciones de sincronización en 5G.

El artículo [3] examina cómo la integración de 5G con TSN puede satisfacer las exigencias de sincronización temporal en la automatización industrial. El documento enfatiza la importancia de una sincronización precisa para aplicaciones industriales y detalla cómo las especificaciones de 3GPP TS 23.501 Release 16 respaldan esta integración.

En particular, el artículo aborda el papel del SIB9 en la provisión de información de tiempo absoluto a los UE. Explica que el SIB9 permite que el 5G actúe como un sistema consciente del tiempo, facilitando la distribución de información de tiempo UTC/GPS a los dispositivos. Esto es esencial para sincronizar dispositivos en redes industriales, permitiendo que el traductor

TSN del dispositivo (DS-TT) funcione como un reloj maestro Institute of Electrical and Electronics Engineers (IEEE) 802.1AS en la red Ethernet industrial.

Un análisis detallado sobre la precisión de las soluciones de sincronización en redes 5G industriales fue realizado por Patel *et al.* en [4]. Este estudio evaluó el error de sincronización extremo a extremo en redes 5G que integran TSN, considerando aplicaciones industriales con requerimientos estrictos de temporización, como márgenes de error inferiores a 1 µs. El trabajo identificó que uno de los principales desafíos para lograr este nivel de precisión es la incertidumbre en el retardo de propagación sobre el enlace radio, debido a las variaciones introducidas en el medio inalámbrico.

Para abordar este problema, los autores compararon dos métodos de compensación de retardo: el uso de *Timing Advance* (TA) unidireccional, y mediciones bidireccionales de ida y vuelta (*round-trip*). Mediante un análisis teórico y pruebas de simulación, demostraron que el método bidireccional permite reducir significativamente el error de sincronización respecto al TA estándar. Además, el estudio señala que la provisión de información temporal mediante SIB9 o mensajes Radio Resource Control (RRC) dedicados constituye una pieza esencial del sistema para alcanzar sincronización submicrosegundo en escenarios industriales, siempre que se combine con mecanismos de estimación de retardo más robustos. Este trabajo refuerza la viabilidad del uso de 5G para aplicaciones sensibles al tiempo, subrayando la importancia de compensar adecuadamente los efectos del canal radioeléctrico.

Un avance relevante en la validación práctica de la sincronización OTA en 5GNR fue presentado por Ansari *et al.* en [5], mediante el desarrollo de un prototipo de red 5G integrada con TSN aplicada al control de robótica móvil en entornos industriales. En dicho trabajo, se implementó un escenario experimental en laboratorio utilizando una red 5G Standalone (SA) con capacidad URLLC, en la cual la BS se encontraba sincronizada con un reloj maestro mediante pulsos *PPS*. El sistema transmitía periódicamente el mensaje SIB9 cada 5 segundos, proporcionando al UE una referencia de tiempo UTC que posteriormente era utilizada para sincronizar su traductor TSN (Device-Side TSN Translator), el cual se encargaba de generar marcas de tiempo precisas para la red Ethernet local.

Durante las pruebas, se midió la desviación entre el reloj del UE y el reloj maestro utilizando un analizador Field-Programmable Gate Array (FPGA) de alta resolución (4ns) y pulsos Pulse Per Second (PPS) para comparar ambas fuentes. Los resultados experimentales mostraron que la sincronización obtenida a través de SIB9 se mantenía con un desfase medio inferior a 3µs y un error máximo por debajo de 8 µs durante un periodo de observación de seis horas. Esta precisión fue suficiente para satisfacer los requisitos de temporización exigidos por la planificación determinista en IEEE 802.1Qbv.

Además, los autores señalaron que una mayor frecuencia en la transmisión del SIB9 podría mejorar aún más la estabilidad de la sincronización. Este trabajo constituye una de las primeras demostraciones empíricas de la viabilidad del uso de SIB9 como fuente de tiempo OTA en redes 5G para aplicaciones industriales exigentes, sin necesidad de recurrir a cableado adicional de sincronización ni a receptores GNSS en los dispositivos.

Una línea de investigación complementaria al ámbito industrial es la validación del uso de sincronización temporal en 5G para aplicaciones audiovisuales profesionales. En [6], Kernen *et al.* presentan los resultados de una serie de pruebas llevadas a cabo en el marco del proyecto europeo 5G-RECORDS, centrado en la producción de contenidos multimedia en vivo. En este estudio, se utilizó una red 5G experimental que integraba capacidades de sincronización dedicadas, incluyendo el uso de SIB9 para proporcionar tiempo de día a los dispositivos de usuario. El objetivo principal era evaluar si dicha red podía transportar PTP —concretamente el perfil de sincronización audiovisual SMPTE ST 2059-2— con la precisión necesaria para alinear cámaras, codificadores y otros equipos de producción en tiempo real.

Los resultados demostraron que, cuando la red 5G opera como sistema *time-aware*, utilizando SIB9 o mensajes RRC dedicados como fuente de tiempo UTC, es posible transportar PTP con una calidad equiparable a redes Ethernet dedicadas. Las pruebas incluyeron una cámara real generando vídeo comprimido con el estándar ST2110 (JPEG-XS), logrando sincronización con márgenes de error dentro del rango de $\pm 1\mu s$, cumpliendo los requisitos de la industria broadcast. El estudio subraya que estas funcionalidades —especificadas en los Release (Rel.) 16 y 17 del 3GPP— son esenciales para garantizar la interoperabilidad de redes 5G con sistemas audiovisuales profesionales, y que sin soporte dedicado de sincronización, los requisitos de precisión no se alcanzarían. Este trabajo valida así el uso de SIB9 en un dominio diferente al industrial, destacando su versatilidad y aplicabilidad práctica en escenarios críticos.

De forma complementaria a los estudios anteriores, Chandramouli *et al.* en [7] presentan una visión más general sobre la evolución de los servicios de sincronización temporal desde las redes 5G actuales hacia 5G-Advanced y 6G. El artículo destaca que en las redes 5G basadas en Release 16 y 17 ya se proporciona información de tiempo de referencia a los UE utilizando dos mecanismos principales: la transmisión en difusión mediante SIB9 y la entrega mediante mensajes RRC unicast dedicados. Estos mecanismos permiten alcanzar precisiones de sincronización en el rango sub-microsegundo, fundamentales para aplicaciones de comunicaciones críticas y redes industriales.

El trabajo también analiza desafíos prácticos asociados a la provisión de sincronización OTA, como la robustez frente a pérdidas de la referencia GNSS o la necesidad de mecanismos de seguridad mejorados para la dis-

tribución de tiempo. En particular, se señala que la transmisión de tiempo mediante SIB9 en modo broadcast, al no estar cifrada ni autenticada, puede ser susceptible a ataques de suplantación, por lo que en escenarios críticos se recomienda el uso de canales dedicados unicast protegidos por la capa de seguridad del 5G. De cara al futuro, el artículo anticipa que los desarrollos hacia 5G-Advanced y 6G fortalecerán los servicios de sincronización, integrándolos de forma más explícita como parte de las ofertas de red, permitiendo la provisión de *Time Synchronization as a Service* para múltiples aplicaciones verticales.

Un caso práctico especialmente relevante es el presentado por Artur Wachtel en [8], donde se documenta la obtención de información horaria mediante SIB9 utilizando un dispositivo comercial Telit FN990 en una red privada 5G. En esta publicación técnica se muestra cómo el UE es capaz de recibir la hora UTC desde la red sin necesidad de recurrir a GNSS, validando así una de las capacidades clave definidas en Release 16 para escenarios industriales o interiores donde la señal satelital no está disponible. Aunque se trata de una fuente no académica, su valor reside en el hecho de que utiliza equipamiento real y reproduce un caso de uso directamente alineado con los objetivos de este trabajo.

A partir del análisis de los trabajos revisados, se constata que el uso del SIB9 en redes 5G representa un avance significativo en la provisión de sincronización temporal precisa a través de la interfaz aérea. Tanto en entornos industriales como en aplicaciones de medios profesionales, la capacidad de distribuir tiempo UTC/GPS mediante mecanismos OTA ha sido validada experimentalmente, alcanzando precisiones del orden de pocos microsegundos. Estos resultados evidencian que la integración de 5G con TSN y la adopción de métodos avanzados de compensación de retardo son factores clave para cumplir los requisitos de sincronización en aplicaciones críticas.

Asimismo, la literatura señala la importancia de considerar aspectos de robustez y seguridad en la entrega de información temporal, especialmente en el contexto de futuras evoluciones hacia 5G-Advanced y 6G. De este modo, se establece una base tecnológica sólida que habilita nuevos escenarios donde la sincronización precisa es un requisito fundamental. El presente trabajo de fin de máster se enmarca en esta línea de investigación, profundizando en la utilización práctica de SIB9 para la sincronización OTA en redes privadas 5G, con especial atención a las limitaciones, desafíos y oportunidades detectados en los estudios analizados.

# Capítulo 3

# Fundamento Teórico

En este capítulo se describen los canales y señales empleados en las transmisiones en 5GNR. Además, se detallan las posibles estructuras de la trama en 5G New Radio (NR), y se explica en profundidad el procedimiento mediante el cual el UE obtiene la hora UTC de la BS y sincroniza su reloj interno [9]. La obtención de la hora UTC se basa en la decodificación de los mensajes SIB, con especial atención al mensaje SIB9, ya que contiene la información temporal crítica para la sincronización precisa requerida en este proyecto.

## 3.1. Canal y señal

Antes de describir los diferentes canales y señales en 5G, es necesario distinguir entre los dos tipos de enlace existentes: el enlace ascendente o Up-link (UL), utilizado por el UE para transmitir hacia la BS; y el enlace descendente o Down-link (DL), usado por la BS para comunicarse con los UE [10].

### 3.1.1. Enlace ascendente

En la figura 3.1 se representa de forma esquemática la organización de los canales y señales en el UL [11].

Entre la capa Radio Link Control (RLC) y la capa Medium Access Control (MAC) encontramos los canales lógicos Common Control Channel (CCCH), Dedicated Control Channel (DCCH) y Dedicated Traffic Channel (DTCH). Los canales CCCH y DCCH transportan mensajes de señalización RRC, mientras que el DTCH se emplea para transmitir datos de usuario.

Entre la capa MAC y la capa física se sitúan los canales de transporte Random Access Channel (RACH) y UL-Shared Channel (SCH). El RACH se utiliza para la transmisión de preámbulos de acceso aleatorio, facilitando el acceso inicial y la resolución de colisiones entre UEs, mientras que el

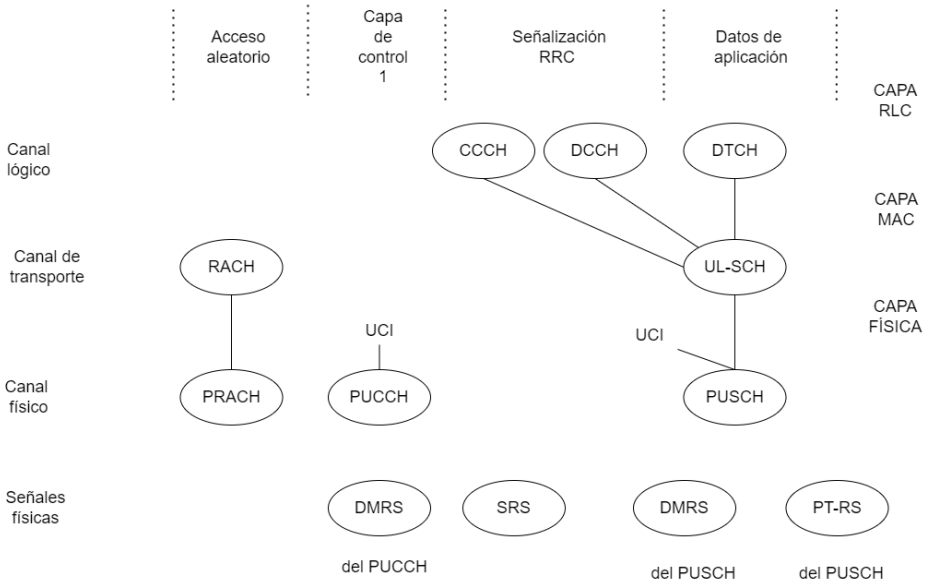



Figura 3.1: Mapa de canales y señales para UL.

UL-SCH soporta la transmisión de datos de usuario en el enlace ascendente.

Los canales físicos del UL son Physical Random Access Channel (PRACH), Physical Up-link Control Channel (PUCCH) y Physical Up-link Shared Channel (PUSCH). El PRACH habilita el acceso aleatorio, el PUCCH transporta información de control del UE (Up-link Control Information (UCI)), como informes de calidad de canal, y el PUSCH se emplea para el transporte de datos de usuario.

Respecto a las señales físicas, encontramos las Demodulation Reference Signal (DMRS) asociadas a PUCCH y PUSCH, utilizadas para estimar el canal, la señal Sounding Reference Signal (SRS) para optimizar la selección de recursos, y la Phase Tracking Reference Signal (PTRS), empleada para compensar errores de fase en transmisiones de alta frecuencia.

Dado que el UL no es el foco de este trabajo, no se profundizará en sus detalles.

### 3.1.2. Enlace descendente

La organización de canales y señales para el enlace descendente se muestra en la figura 3.2 [12].

Entre la capa RLC y la MAC encontramos los canales lógicos Broadcast Control Channel (BCCH), CCCH, DCCH, DTCH y Paging Control Channel (PCCH).

El canal BCCH transmite el MIB y los SIBs. El MIB viaja a través del Broadcast Channel (BCH) sobre el canal físico PBCH, mientras que

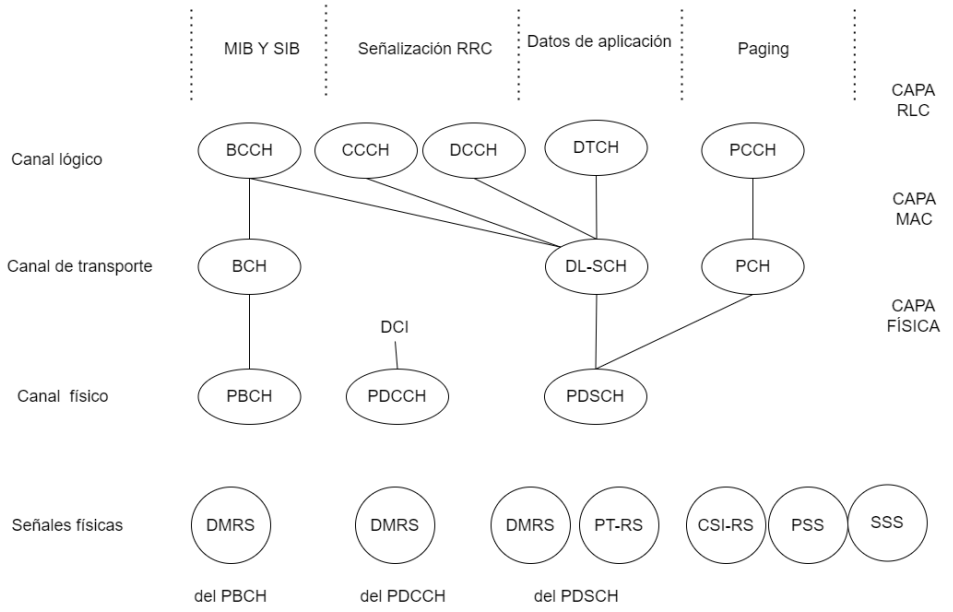



Figura 3.2: Mapa de canales y señales para DL.

los SIBs son transportados sobre PDSCH. De estos mensajes se hablará en detalle en las secciones 3.4.1 y 3.4.2. Los canales CCCH y DCCH sirven para el transporte de mensajes RRC, esenciales para la gestión de recursos radio, mientras que el canal DTCH transporta datos de usuario. Finalmente, el canal PCCH difunde mensajes de radio-búsqueda, informando al UE de eventos de red relevantes.

Entre los canales de transporte se encuentran BCH, DL-SCH y Paging Channel (PCH). El BCH vehicula el MIB a través del PBCH, el DL-SCH transporta SIBs y datos de usuario, y el PCH permite la emisión de mensajes de radio-búsqueda de forma energéticamente eficiente.

Respecto a los canales físicos:

- El PBCH transporta información necesaria para la sincronización inicial, como se detallará en la sección 3.3.1.
- El PDCCH contiene el Down-link Control Information (DCI), que asigna recursos radio a UE tanto para UL como DL.
- El PDSCH es el principal canal de transporte de datos y SIBs.

La estructura de estos canales se ilustra en las figuras 3.3 y 3.4.
Finalmente, entre las señales físicas del DL destacan:

- DMRS y PTRS, destinadas a estimación de canal y compensación de errores de fase respectivamente.

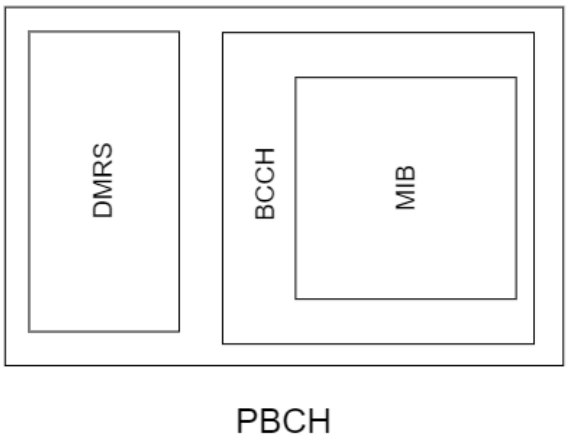



Figura 3.3: Canales y señales contenidos en el canal PBCH.

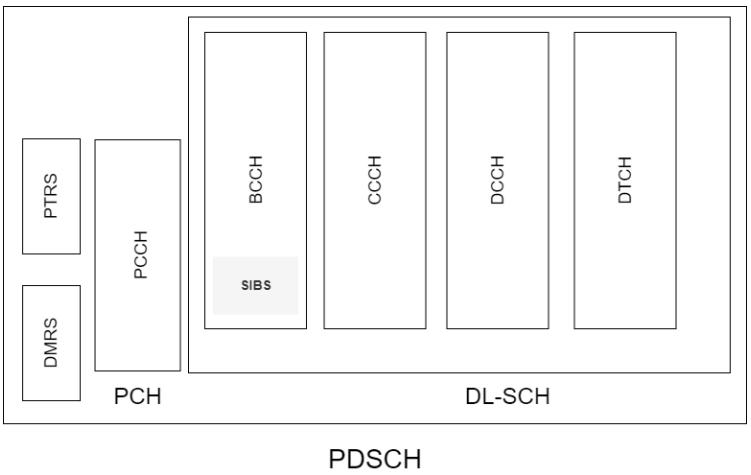



Figura 3.4: Canales y señales contenidos en el canal PDSCH.

- Channel State Information Reference Signal (CSIRS), utilizada para optimizar la selección de recursos radio.
- Primary Synchronisation Signal (PSS) y Secondary Synchronization Signal (SSS), fundamentales para la sincronización temporal y de identificación de celda del UE, que se describirán en la sección 3.5.1.

Las señales y canales directamente relacionados con el objetivo de este proyecto se tratarán en profundidad en las secciones 3.5, 3.3 y 3.4.

## 3.2. Estructura de una trama en 5G

En esta sección se describe la estructura de trama en 5G NR, conforme a las especificaciones definidas por el 3GPP [12].

La estructura de la trama está determinada por la *numerología*, concepto que engloba parámetros como la separación entre subportadoras, la duración de símbolos OFDM y el número de slots por subtrama o trama. Esta numerología se define mediante el parámetro $\mu$, cuyo valor afecta directamente a dichas características.

La separación entre subportadoras en función de $\mu$ se muestra en la tabla 3.1.

Tabla 3.1: Valor $\mu$ en función de la separación entre subportadoras

| $\mu$ | Separación Subportadoras (kHz) |
|---|---|
| 0 | 15 |
| 1 | 30 |
| 2 | 60 |
| 3 | 120 |
| 4 | 240 |

La configuración temporal asociada a cada valor de $\mu$ se ilustra en la figura 3.5.

La duración de cada símbolo Orthogonal Frequency-Division Multiplexing (OFDM) también depende de $\mu$, como recoge la tabla 3.2.

Tabla 3.2: Duración del símbolo OFDM en función de la separación entre subportadoras

| Separación Subportadoras (kHz) | 15 | 30 | 60 | 120 | 240 |
|---|---|---|---|---|---|
| Duración Símbolo OFDM ($\mu s$) | 66,67 | 33,33 | 16,67 | 8,33 | 4,17 |
| Duración Prefijo Cíclico ($\mu s$) | 4,69 | 2,34 | 1,17 | 0,57 | 0,29 |
| Duración Total ($\mu s$) | 71,35 | 35,68 | 17,84 | 8,92 | 4,46 |

A modo de resumen, la tabla 3.3 recoge el número de símbolos por slot, slots por trama y slots por subtrama en función de $\mu$.

Tabla 3.3: Resumen de estructura de trama según $\mu$

| $\mu$ | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| Nº Símbolos por Slot | 14 | 14 | 14 | 14 | 14 |
| Nº Slots por Trama | 10 | 20 | 40 | 80 | 160 |
| Nº Slots por Subtrama | 1 | 2 | 4 | 8 | 16 |

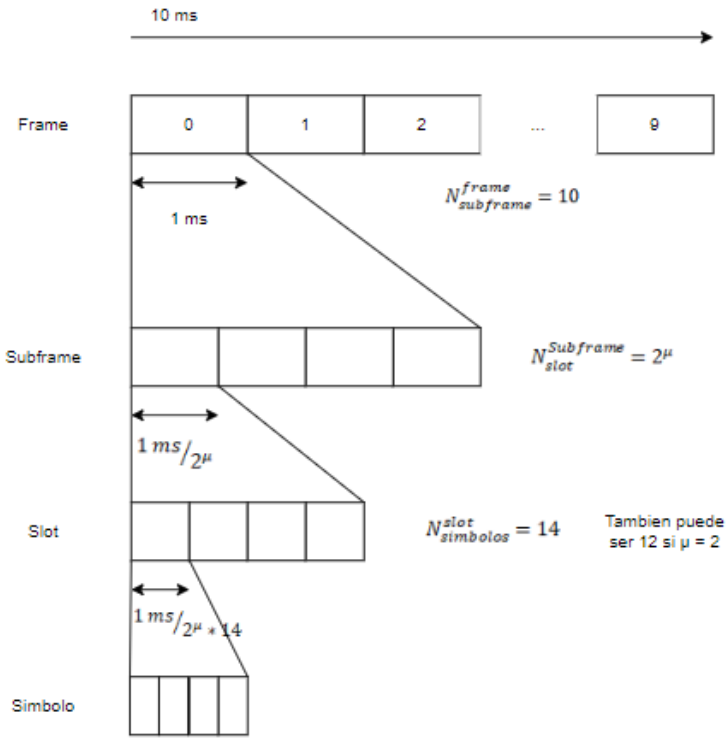



Figura 3.5: Estructura de trama en 5G NR según el valor de $\mu$.

### 3.2.1. Casos de configuración de SS Block

Además de la separación entre subportadoras, otros parámetros varían según la configuración, como el número máximo de haces transmitidos y la posición de los bloques Synchronization Signal (SS).

La asignación de casos (A a G) y sus características se muestra en la tabla 3.4.

Cabe destacar que el *caso C* ha sufrido modificaciones en diferentes revisiones de la especificación 3GPP [**TS-casoC**], por lo que pueden existir ligeras discrepancias entre fuentes.

En cuanto a los rangos de frecuencia, Frequency Range 1 (FR1) comprende de 450 MHz a 7125 MHz y Frequency Range 2 (FR2) de 24250 MHz a 71000 MHz [13]. De este modo:

- Para FR1 se utilizan los casos A, B y C.
- Para FR2 se aplican los casos D, E, F y G.

Tabla 3.4: Características de los casos según separación entre subportadoras

| **Separación entre subportadoras** | **Símbolo OFDM inicio** | **fc <3 GHz** | **3 GHz <fc <7,125 GHz** | **7,125 GHz <fc** |
|---|---|---|---|---|
| Caso A<br>15 KHz | {2,8} + 14n | n = 0,1<br>Lmax = 4 | n = 0,1,2,3<br>Lmax = 8 | |
| Caso B<br>30 KHz | {4,6,16,20} + 28n | n = 0<br>Lmax = 4 | n = 0,1<br>Lmax = 8 | |
| Caso C<br>30 KHz | {2,8} + 14n | n = 0,1<br>Lmax = 4 | n = 0,1,2,3<br>Lmax = 8 | |
| Caso D<br>120 KHz | {4,8,16,20} + 28n | | | n = 0,1,2,3,5,6,7,<br>8,10,11,12,13,15,16,<br>17,18<br>Lmax = 64 |
| Caso E<br>240 KHz | {8,12,16,20,32,36,<br>40,44} + 56n | | | n = 0,1,2,3,5,6,7,8<br>Lmax = 64 |
| Caso F<br>480 KHz | {2,9} + 14n | | | n = 0,1,2,3,4,5,6,<br>7,8,9,10,11,12,13,<br>14,15,16,17,18,19,<br>20,21,22,23,24,25,26,<br>27,28,29,30,31<br>Lmax = 64 |
| Caso G<br>960 KHz | {2,9} + 14n | | | n = 0,1,2,3,4,5,6,<br>7,8,9,10,11,12,13,<br>14,15,16,17,18,19,<br>20,21,22,23,24,25,26,<br>27,28,29,30,31<br>Lmax = 64 |

### 3.2.2. Ejemplo: Caso A para $f_c < 3$ GHz

Como ejemplo, se analiza el *caso A* para una frecuencia portadora inferior a 3 GHz. Aquí, los símbolos de inicio de los bloques SS son 2, 8, 16 y 22, como se ilustra en la figura 3.6.

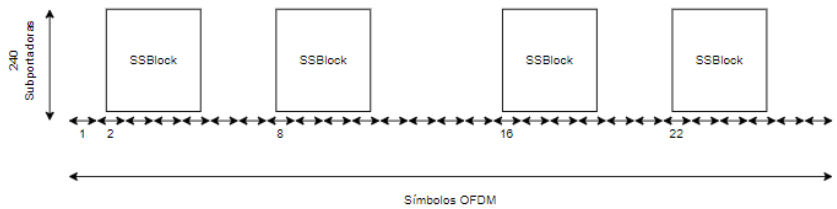



Figura 3.6: Posicionamiento de los bloques SS en los símbolos OFDM para el caso A.

Cada bloque SS ocupa 4 símbolos consecutivos, tal como se profundizará en la sección 3.5.1.

## 3.3. Canales implicados en la obtención del mensaje SIB9

En esta sección se describen los canales físicos implicados en la obtención del mensaje SIB9. Para alcanzar dicho objetivo, es necesario que el UE decodifique previamente el MIB y el SIB1. Por tanto, también se explicarán los canales utilizados en la obtención de estos bloques.

De manera general:

- El canal PBCH transporta el MIB.
- El canal PDSCH transporta los SIBs.
- La decodificación del PDSCH requiere previamente la decodificación del canal PDCCH, que contiene la información DCI necesaria [12].

### 3.3.1. PBCH

El canal PBCH utiliza modulación Quadrature Phase-Shift Keying (QPSK) [**TS-casoC**] y ocupa los símbolos 2, 3 y 4 del SSBlock. Está presente en todas las subportadoras durante los símbolos 2 y 4, y en el símbolo 3 en aquellas subportadoras no ocupadas por la señal SSS (ver figura 3.12 en la sección 3.5).

El canal PBCH transmite información esencial para la sincronización, así como una señal DMRS pseudoaleatoria, cuya ubicación depende del Physical Layer Cell Identity (PCI) de la celda. La figura 3.7 muestra un ejemplo de mapeado para pci  mód 4.

La detección del PCI se realiza a partir de la recepción de las señales PSS y SSS (ver sección 3.5.1). Esta información permite realizar una correcta estimación del canal.

El PBCH transporta una carga útil de 56 bits, que se distribuyen de la siguiente forma:

- 24 bits para el Cyclic Redundancy Check (CRC).
- 23 bits para el mensaje MIB.
- 1 bit para la identificación BCCH.
- 8 bits para información adicional:
  - 4 bits del Sytem Frame Number (SFN).
  - En FR1, 1 bit para la compensación de subportadora del SSBlock y 2 bits reservados.
  - En FR2, estos 3 bits son usados para el SSBlock Index.
  - 1 bit para indicar la mitad temporal dentro del periodo de 10 ms.

Los detalles sobre el MIB se abordaran en la sección 3.4.1.

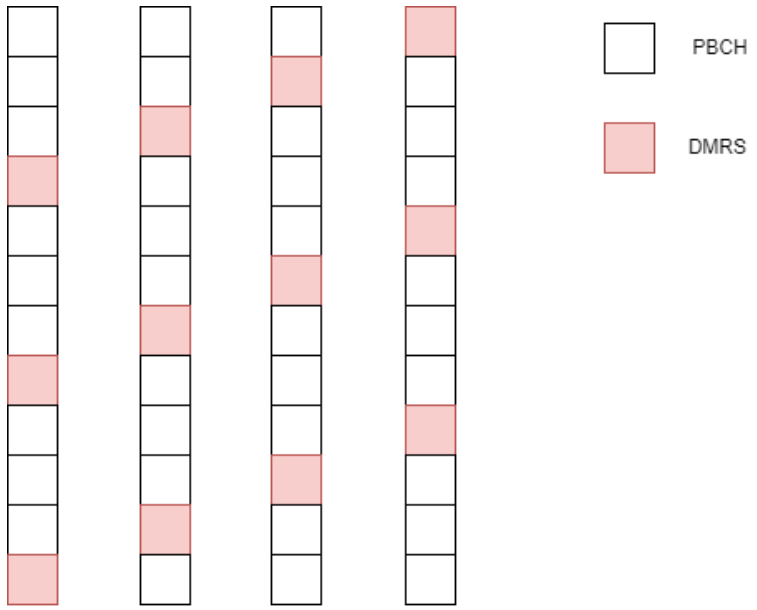



Figura 3.7: Opciones de mapeado de la señal DMRS en el PBCH para pci mód 4.

### 3.3.2. PDCCH

El canal PDCCH [12] transporta mensajes de DCI [14], que contienen la información necesaria para la decodificación del canal PDSCH donde residen los SIBs.

El PDCCH se transmite dentro de un COntrol REsource SET (CORESET) y emplea modulación QPSK, con su correspondiente señal DMRS asociada.

Existen varios tipos de mensajes DCI (un total de 14 formatos), pero nos centramos en los formatos 1.0 y 1.1:

- **Formato 1.0**: contiene información para la programación de transmisiones unicast y broadcast, incluyendo los siguientes Radio Network Temporary Identifier (RNTI) relevantes:
  - **SI-RNTI** (65535): identifica mensajes de información de sistema.
  - **P-RNTI** (65534): utilizado para mensajes de paginación.
  - **Otros RNTI**: relacionados con acceso aleatorio (RA-RNTI, TC-RNTI) y programación específica para UEs (C-RNTI, MCS-C-RNTI, CS-RNTI).
- **Formato 1.1**: especifica parámetros sobre los recursos temporales y frecuenciales usados para la transmisión en el PDSCH.

Una vez decodificada la información del PDCCH, el UE está capacitado para proceder a la recepción del PDSCH.

### 3.3.3. PDSCH

El canal PDSCH, como se mostró en la figura 3.2, transporta los canales de transporte DL-SCH y PCH [12]. Su uso principal incluye:

- Transmisión de datos de usuario.
- Mensajes de señalización de capa RRC (Signalling Radio Bearer (SRB)).
- Difusión de mensajes SIBs [15].

Puede operar con modulación QPSK, 16Quadrature Amplitude Modulation (QAM) o 256QAM, y cuenta con señal DMRS asociada, así como opcionalmente PTRS o Tracking Reference Signal (TRS). Además, puede aplicar el concepto de Quasi Co-Location (QCL) para facilitar la recepción coherente entre el SSBlock y el PDSCH.

Entre los parámetros relevantes para la decodificación del PDSCH se encuentran:

- **dataScramblingIdentityPDSCH**: identificador de la secuencia pseudoaleatoria de codificación.
- **pdsch-TimeDomainAllocationList**: tabla que define desplazamientos de ranura ($k_0$), mapeo temporal (A o B) y duración.
- **rbg-Size**: tamaño del grupo de bloques de recursos.
- **mcs-Table**: especifica la modulación y esquema de codificación aplicados.

Estos parámetros son esenciales para la correcta recepción y decodificación de los mensajes SIB9.

## 3.4. Mensajes implicados en la obtención del SIB9.

En esta sección se describirán los bloques de información útiles para la sincronización de los dispositivos con una estación base. Se hablará del mensaje MIB y de los mensajes SIB 1 y 9 [16].

### 3.4.1. MIB

El mensaje MIB [15] se encuentra dentro del canal lógico BCCH, que se transmite dentro del canal de transporte BCH y este a su vez se transmite dentro del canal físico PBCH(figura 3.2). Dentro del canal PBCH, como se

dijo anteriormente, el mensaje MIB es una secuencia de 23 bits. En esta sección se pretende desglosar la información que nos aporta este bloque. En la tabla 3.5 se observan para que están destinados cada uno de los 23 bits que componen el mensaje MIB.

Tabla 3.5: Estructura del MIB

| **MIB** | **Nº de bits** |
|---|---|
| $systemFrameNumber$ | 6 |
| $subCarrierSpacingCommon$ | 1 |
| $ssb-SubcarrierOffset$ | 4 |
| $dmrs-TypeA-position$ | 1 |
| $pdcch-ConfigSIB1$ | 8 |
| $cellBarred$ | 1 |
| $intraFreqReselection$ | 1 |
| $spare$ | 1 |

- $systemFrameNumber$. El $systemFrameNumber$ son los 6 bits más significativos del SFN, que junto con los 4 del canal PBCH, forma una cadena de 10 bits que nos indica el número de la trama, cuando este llega a 1023 se reinicia a 0.

- $subCarrierSpacingCommon$. Indica la separación entre subportadoras usada para el SIB1. Si estamos en FR1, puede ser 15 KHz o 30 KHz (bit a 0 o 1 respectivamente) y si estamos en FR2, puede ser 60 KHz o 120 KHz (bit a 0 o 1 respectivamente).

- $ssb-SubcarrierOffset$. Da el valor de la $k_{ssb}$, que indica el desplazamiento en el dominio de la frecuencia en número de subportadoras entre el SSB y la cuadrilla de bloque de recursos general.

- $dmrs-TypeA-position$. Indica cuál es la posición del primer símbolo de la señal DMRS del canal PDSCH. Puede tomar los valores 2 o 3 (bit a 0 o 1 respectivamente).

- $pdcch-ConfigSIB1$. Da una importante información sobre la configuración del CORESET0, del $searchspace0$ y del canal PDCCH necesaria para la posterior obtención del mensaje SIB1. Contiene dos campos, el $controlResourceSetZero$ y el $searchSpaceZero$, de 4 bits cada uno.

- $cellBarred$. Indica si los UE en la celda pueden acceder a ella o no. Toma valores barred o notBarred (bit a 0 o 1 respectivamente).

- *intraFreqReselection*. Indica si el UE tiene permitido conectarse a otra celda que trabaje a la misma frecuencia si ha sido rechazado de la celda de mayor rango. Toma valores allowed o notAllowed (bit a 0 o 1 respectivamente).
- *spare*. Un bit libre, sin uso determinado dentro del mensaje MIB.

[14] [**TS-casoC**]

### 3.4.2. Mensajes SIBs

Por último, se hablará sobre los mensajes SIBs. En concreto, se hablará sobre el mensaje SIB1, que contiene información, entre otras cosas, para saber dónde y si están los otros mensajes SIBs, y del mensaje SIB9, que contiene la información de la hora en formato UTC de la BS [15].

El SIB1 es transmitido usando el canal lógico BCCH, el canal de transporte DL-SCH y el canal físico PDSCH con un periodo de 160ms. Contiene información que permite al UE saber si puede conectarse o no a la celda. Además, contiene la información sobre los demás mensajes SIBs e información sobre la configuración de los recursos radio. Usa el formato DCI 1.0 y con un conjunto de espacio de búsqueda que se explicarán a continuación.
Para ello tenemos que recordar dos campos del mensaje MIB vistos en la tabla 3.5. En concreto, el *subCarrierSpacingCommon*, que nos indica la separación entre subportadoras que utiliza el mensaje SIB1 y el $pdcch - configSIB1$, que contiene 8 bits, 4 para el *controlResourceSetZero* y los otros 4 para el *searchSpaceZero*. Estos valores se usan como punteros para tablas dentro de la configuración del CORESET. Para el valor del *controlResourceSetZero* 3GPP tiene especificadas muchas tablas, cada una específica para una separación entre subportadoras del SS/PBCH Block y del canal PDCCH. Están en la especificación [10]. Aquí se expondrán dos de ellas en la tabla 3.6 y 3.7. La tabla 3.6 corresponde a 15 KHz de separación entre subportadoras para ambos bloques. El índice es el valor que toma el parámetro *controlResourceSetZero*, el nº de bloques de recursos son el ancho de banda que ocupa el CORESET, y el nº de símbolos son los símbolos que ocupa (dominio temporal). Por último, el valor de la compensación de bloques de recursos da la compensación entre el bloque SS/PBCH y el CORESET. Para esclarecer esto último está la figura 3.8. Los valores RB son los valores que puede tomar la compensación de bloques de recursos para los casos de índice 0 a 5.
En las figuras 3.9 3.10 y se muestra un ejemplo de una señal que cumple con las especificaciones de la tabla 3.6. En este ejemplo, el valor que toma el parámetro *controlResourceSetZero* es 0. Por lo tanto, tiene 24 bloques de recursos, ocupa 2 símbolos y la compensación es de 0. En la figura 3.9 podemos ver cómo la compensación es 0. Y en la figura 3.10 podemos ver

Tabla 3.6: Tabla configuración CORESET para separación entre subportadoras 15 KHz en SS/PBCH y del PDCCH.

| **Indice** | **Nº Bloques de recursos** | **Nº de símbolos** | **Compensación de Bloques de recursos.** |
|---|---|---|---|
| 0 | 24 (4.32 MHz) | 2 | 0 |
| 1 | | | 2 |
| 2 | | | 4 |
| 3 | | 3 | 0 |
| 4 | | | 2 |
| 5 | | | 4 |
| 6 | 48 (8.64 MHz) | 1 | 12 |
| 7 | | | 16 |
| 8 | | 2 | 12 |
| 9 | | | 16 |
| 10 | | 3 | 12 |
| 11 | | | 16 |
| 12 | 96 (17.28 MHz) | 1 | 38 |
| 13 | | 2 | |
| 14 | | 3 | |
| 15 | Reservado | | |

cómo ocupa dos símbolos y como ocupa 24 bloques de recursos, que son 288 subportadoras. Ahora se muestra la tabla del valor *controlResourceSetZero* cuando la separación entre subportadoras del bloque SS/PBCH y del canal PDCCH es de 30 KHz. La cual es la configuración que se usará en las señales con las que se trabajarán en este proyecto. La tabla de la que se está hablando es la tabla 3.7

Para el valor del *searchSpaceZero* 3GPP propone diferentes tablas en función del rango de frecuencia de la señal, FR1 o FR2, y del patrón de multiplexación. En la tabla 3.8 vemos la tabla que se usa para los casos FR1 y patrón de multiplexación 1. El índice es el valor que toma la variable *searchSpaceZero*. El valor de la $O$ indica el retardo en número de slot relativo al inicio de la trama, el número de espacios de búsqueda establecido por ranura y el valor de la $M$ esta relacionados, y ayudan a calcular el número de ranuras de un bloque SS/PBCH concreto. Por último, el índice del primer símbolo, como su nombre indica, marca el primer símbolo, dentro del slot, que contiene información útil.

Ahora se muestra, en la tabla 3.9, todo el contenido que tiene el mensaje SIB1.

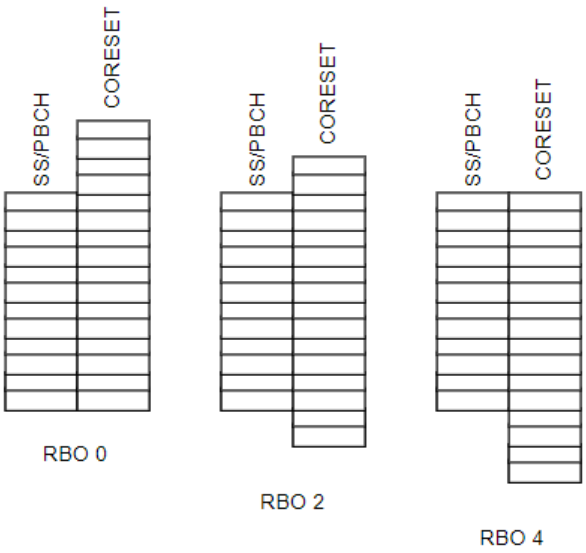



Figura 3.8: Compensación Bloques de Recursos.

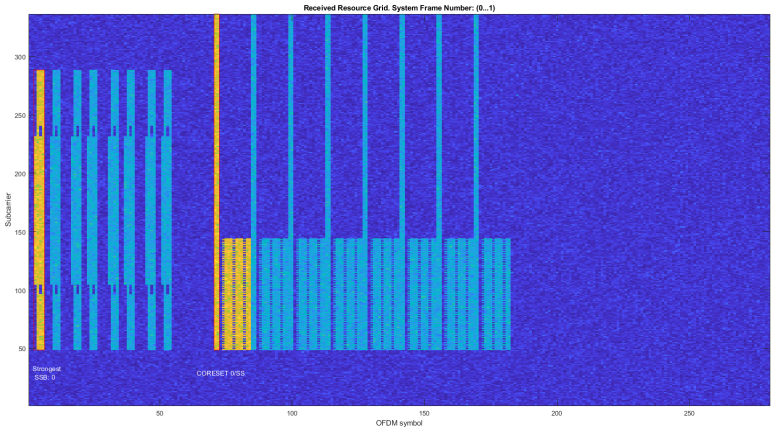



Figura 3.9: Compensación Bloques de Recursos en espectrograma.

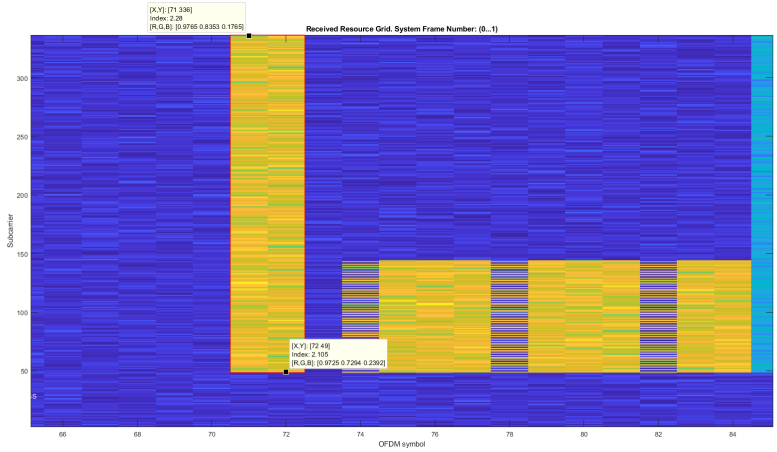



Figura 3.10: Duración del canal tanto en símbolos como en frecuencia.

Tabla 3.7: Tabla configuración CORESET para separación entre subportadoras 30 KHz en SS/PBCH y del PDCCH.

| **Indice** | **Nº Bloques de recursos** | **Nº de símbolos** | **Compensación de Bloques de recursos** |
|---|---|---|---|
| 0 | 24 (4.32 MHz) | 2 | 0 |
| 1 | | | 1 |
| 2 | | | 2 |
| 3 | | | 3 |
| 4 | | | 4 |
| 5 | | 3 | 0 |
| 6 | | | 1 |
| 7 | | | 2 |
| 8 | | | 3 |
| 9 | | | 4 |
| 10 | 48 (8.64 MHz) | 1 | 12 |
| 11 | | | 14 |
| 12 | | | 16 |
| 13 | | 2 | 12 |
| 14 | | | 14 |
| 15 | | | 16 |

Tabla 3.8: Tabla valor *searchSpaceZero* para FR1 y patrón de multiplexación 1.

| **Indice** | **O** | **Nº de espacio de búsqueda establecido por ranura** | **M** | **Indice Primer Símbolo** |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 |
| 1 | | 2 | 1/2 | 0 si 'i' es par<br>1 si 'i' es impar |
| 2 | 2 | 1 | 1 | 0 |
| 3 | | 2 | 1/2 | 0 si 'i' es par<br>1 si 'i' es impar |
| 4 | 5 | 1 | 1 | 0 |
| 5 | | 2 | 1/2 | 0 si 'i' es par<br>1 si 'i' es impar |
| 6 | 7 | 1 | 1 | 0 |
| 7 | | 2 | 1/2 | 0 si 'i' es par<br>1 si 'i' es impar |
| 8 | 0 | 1 | 2 | 0 |
| 9 | 5 | | | 0 |
| 10 | 0 | | 1 | 1 |
| 11 | | | | 2 |
| 12 | 2 | | | 1 |
| 13 | | | | 2 |
| 14 | 5 | | | 1 |
| 15 | | | | 2 |

Tabla 3.9: Contenido SIB1.

| | | | |
|---|---|---|---|
| **cellSelectionInfo** | q-RxLevMin | -70 a -22 dBm (valor real = valor señalado * 2) | |
| | q-RxLevMinOffset | 1 a 8 dB (valor real = valor señalado * 2) | |
| | q-RxLevMinSUL | -70 -22 dBm (valor real = valor señalado * 2) | |
| | q-QualMin | -43 a -12 dBm | |
| | q-QualMinOffset | 1 a 8 | |
| **cellAccessRelatedInfo** | 1 to 12 instances of PLMN-Identity Info | 1 to 12 instances of PLMN-Identity | mcc (secuencia de 3 dígitos) |
| | | | mnc (secuencia de 2 o 3 dígitos) |
| | | trackingAreaCode | 24 bits |
| | | ranac | De 0 a 255 |
| | | cellIdentity | 36 bits |
| | | ccllReservedForOperatorUse | Reserved, notReserved |
| | cellReservedForOtherUse | true | |
| **connEstFailureControl** | connEstFailCount | 1,2,3,4 | |
| | connEstFailOffsetValidity | 30, 60, 120, 240, 300, 420, 600, 900 seconds | |
| | connEstFailOffset | 0 a 15 | |
| **si-SchedulingInfo** | mirar tabla 3.10. | | |
| **servingCellConfigCommon** | mirar tabla 3.11 3.12 3.13 | | |
| **ims-EmergencySupport** | true | | |
| **eCallOverIMS-Support** | true | | |
| **ue-TimerAndConstants** | t300 | 100,200,300,400,600, 1000, 1500, 2000ms | |
| | t301 | 100,200,300,400,600, 1000, 1500,2000ms | |
| | t310 | 0, 50, l00 , 200,500, 1000, 2000 ms | |
| | n310 | 1, 2, 3, 4, 6, 8, 10, 20 | |
| | t311 | 1000, 3000, 5000, 10 000, 15 000, 20 000, 30 000 ms | |
| | n311 | 1, 2, 3, 4, 5, 6, 8, 10 | |
| | t319 | 100,200,300,400,600, 1000, 1500,2000ms | |
| **uac-BarringInfo** | mirar tabla 3.14 | | |
| **useFullResumeID** | true | | |

Tabla 3.10: Parámetro si-Schedulingltifo.

<table>
<tr><td rowspan="6">schedulinglnfoList</td><td rowspan="6">1 to 32 instances of Scbedulinglnfo</td><td>si-BroadcastStatus</td><td colspan="3">broadcasting, notBroadcasting</td></tr>
<tr><td>si-Periodicity</td><td colspan="3">8, 16, 32, 64, 128, 256, 512 radio frames</td></tr>
<tr><td rowspan="4">sib-Mappinglnfo</td><td rowspan="4">1 to 32 type instances of SIB-Typelnfo</td><td>type</td><td>SIB 2, 3, 4,<br>5, 6, 7, 8, 9</td></tr>
<tr><td>valueTag</td><td>0 a 31</td></tr>
<tr><td>areaScope</td><td>true</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td>si-WindowLength</td><td colspan="5">5, 10, 20, 40, 80, 160,320,640, 1280 slots</td></tr>
<tr><td rowspan="10">si-RequestConfig</td><td rowspan="8">rach-OccasionsSI</td><td rowspan="7">rach-ConfigSI</td><td colspan="2">prach-Configuration Index</td><td>0 a 255</td></tr>
<tr><td colspan="2">msg l-FDM</td><td>1, 2, 4, 8</td></tr>
<tr><td colspan="2">msg1-FrequencyStart</td><td>0 a 274</td></tr>
<tr><td colspan="2">zeroCorrelationZoneConfig</td><td>0 a 15</td></tr>
<tr><td colspan="2">preambleReceivedTargetPower</td><td>-202 a -60</td></tr>
<tr><td colspan="2">preambleTransMax</td><td>3, 4, 5, 6, 7, 8,<br>10, 20,<br>50, 100, 200</td></tr>
<tr><td colspan="2">powerRampingStep</td><td>0, 2, 4, 6 dB</td></tr>
<tr><td colspan="2">ra-ResponseWindow</td><td>1,2,4,8, 10,20,<br>40, 80 slots</td></tr>
<tr><td>ssb-perRACH-Occasion</td><td colspan="3">oneEighth, oneFourth, oneHalf, one,<br>two, four, eight, sixteen</td></tr>
<tr><td>si-RequestPeriod</td><td colspan="4">1,2,4,6,8, 10, 12, 16</td></tr>
<tr><td rowspan="3">si-RequestResources</td><td rowspan="3">I to 32 instances<br>of SI-RequestResources</td><td colspan="2">ra-PreambleStartIndex</td><td>0 a 63</td></tr>
<tr><td colspan="2">ra-Association PeriodIndex</td><td>0 a 15</td></tr>
<tr><td colspan="2">ra-ssb-OccasionMaskIndex</td><td>0 a 15</td></tr>
<tr><td>si-RequestConfigSUL</td><td colspan="5">mismos parámetros que la estructura usada para el si-RequestConfig</td></tr>
<tr><td>systemInformationArealD</td><td colspan="5">24 bits.</td></tr>
</table>

Tabla 3.11: Parámetro servingCellConfigCommon Parte 1.

<table>
<tr><td rowspan="15">dowlinkConfig Common</td><td rowspan="6">frequencyInfoDL</td><td rowspan="4">frequencyBandList</td><td colspan="4">Secuencia de 1 a 8 valores</td></tr>
<tr><td>freqBandIndicalorNR</td><td colspan="3">1 a 1024</td></tr>
<tr><td rowspan="3">nr-NS-PmaxList</td><td colspan="3">Secuencia de 1 a 8 valores</td></tr>
<tr><td>additionalPmax</td><td colspan="2">-30 a 33 dBm</td></tr>
<tr><td>additionalSpectrumEmission</td><td colspan="2">0 a 7</td></tr>
<tr><td>offsetToPointA</td><td colspan="4">0 a 2199</td></tr>
<tr><td rowspan="5">scs-SpecificCarrierrList</td><td colspan="4">Secuencia de 1 a 5 valores</td></tr>
<tr><td>offsetToCarrier</td><td colspan="3">0 a 2199</td></tr>
<tr><td>subcarrierrSpacing</td><td colspan="3">15, 30, 60, 120 kHz</td></tr>
<tr><td>carrierBandwidth</td><td colspan="3">1 a 275</td></tr>
<tr><td>txDirectCurrentLocation</td><td colspan="3">0 a 4095</td></tr>
<tr><td rowspan="3">initialDowlinkBWP</td><td rowspan="3">genericParameters</td><td>locationAndBandwidth</td><td colspan="3">0 a 37949</td></tr>
<tr><td>sucarrierspacing</td><td colspan="3">15, 30, 60, 120 kHz</td></tr>
<tr><td>cyclicPrefix</td><td colspan="3">extended</td></tr>
<tr><td>pdcch-ConfigCommon</td><td colspan="4">SetupRelease { PDCCH-ConfigCommon }</td></tr>
<tr><td>pdsch-ConfigCommon</td><td colspan="4">SetupRelease { PDSCH-ConfigCommon )</td></tr>
<tr><td>bcch-Config</td><td>modificationPeriodCoeff</td><td colspan="4">2, 4, 8, 16</td></tr>
<tr><td rowspan="4">pcch-Config</td><td>defaultPagingCycle</td><td colspan="4">32, 64, 128, 256 radio frames</td></tr>
<tr><td rowspan="2">nAndPagingFrameOffset</td><td colspan="4">Elegir</td></tr>
<tr><td>oneT</td><td>halfT</td><td>oneEighT</td><td>oneSixteenthT</td></tr>
<tr><td>0, 1</td><td>0 a 3</td><td>0 a 7</td><td>0 a 15</td></tr>
<tr><td>ns</td><td colspan="4">four, two, one</td></tr>
<tr><td colspan="5">firstPDCCH-MonitoringOccasionOlPO</td></tr>
</table>

Tabla 3.12: Parámetro servingCellConfigCommon Parte 2.

<table>
<tr><td rowspan="14">uplinkConfig Common</td><td rowspan="9">frequencyInfoUL</td><td rowspan="4">frequencyBandList</td><td colspan="3">Secuencia de 1 a 8 ejemplos</td></tr>
<tr><td>freqBandIndicatorNR</td><td colspan="2"></td></tr>
<tr><td rowspan="3">nr-NS-PmaxList</td><td colspan="2">Secuencia de 1 a 8 ejemplos</td></tr>
<tr><td>additionalPmax</td><td>-30 a 33 dBm</td></tr>
<tr><td>additionalSpectrumEmission</td><td>0 a 7</td></tr>
<tr><td colspan="2">absoluteFrequencyPointA</td><td colspan="2">0 a 3279165</td></tr>
<tr><td colspan="2" rowspan="4">scs-SpecificCarrierList</td><td colspan="2">Secuencia de 1 a 15 ejemplos</td></tr>
<tr><td>offsetToCarrier</td><td>0 a 2199</td></tr>
<tr><td>subcarrierSpacing</td><td>15,30,60, 120,240</td></tr>
<tr><td>carrierBandwidth</td><td>1 to 275</td></tr>
<tr><td colspan="2">p-Max</td><td colspan="2">-30 to 33 dBm</td></tr>
<tr><td colspan="2">frequencyShift7p5kHz</td><td colspan="2">true</td></tr>
<tr><td rowspan="5">initialUplinkBWP</td><td rowspan="5">genericParameters</td><td rowspan="3">BWP</td><td>locationAndBandwidth</td><td>0 a 37949</td></tr>
<tr><td>subcarrierSpacing</td><td>15,30,60, 120,240</td></tr>
<tr><td>cyclic Prefix</td><td>extended</td></tr>
<tr><td colspan="3">SetupRelease { RACH-ConfigCommon }</td></tr>
<tr><td colspan="3">SetupRelease { PUSCH-ConfigCommon }</td></tr>
<tr><td colspan="2">timeAlignmentTimerCommon</td><td colspan="3">SetupRelease { PUCCH-ConfigCommon }</td></tr>
<tr><td colspan="2">500, 150, 1280, 1920. 2560, 5120, 10240 ms, infinity</td><td colspan="3"></td></tr>
<tr><td colspan="2">supplementaryUplink</td><td colspan="4">mismos parámetros que el parámetro uplinkConfigCommon</td></tr>
</table>

Tabla 3.13: Parámetro servingCellConfigCommon Parte 3.

| | | |
|---|---|---|
| n-TimingAdvanceOffset | 0, 25600, 39936 | |
| ssb-PositionsInBurst | inOneGroup | 8 bits |
| | groupPresence | 8 bits |
| ssb_PeriodicityServingCell | 5, 10, 20, 40, 80, 160 ms | |
| tdd-UL-DL-ConfigurationCommon | Tabla configuración UL-DL | |
| ss-PBCH-BlockPower | -60 a 50dBm | |

Tabla 3.14: Parametros uac-BarringInfo.

<table>
<tr><td rowspan="15">uac-BarringInfo</td><td rowspan="3">uac-BarringForCommon</td><td colspan="4">Secuencia de 1 a 63 ejemplos</td></tr>
<tr><td colspan="3">accessCategory</td><td>1 a 63</td></tr>
<tr><td colspan="3">uac-barringInfoSetIndex</td><td>1 a 8</td></tr>
<tr><td rowspan="7">uac-BarringerPLMN-List</td><td colspan="4">Secuencia de 1 a 12 ejemplos</td></tr>
<tr><td colspan="3">plmn-IdentityIndex</td><td>1 a 12</td></tr>
<tr><td rowspan="5">uac-ACBarringListType</td><td colspan="3">Elegir</td></tr>
<tr><td>uac-ImplicitACBarringList</td><td colspan="2">uac-ExplicitACBarringList</td></tr>
<tr><td>Secuencia de 63 ejemplos</td><td colspan="2">Secuencia de 1 a 63</td></tr>
<tr><td rowspan="2">1 a 8</td><td>accessCategory</td><td>1 a 63</td></tr>
<tr><td>uac-barringInfoSet Index</td><td>1 a 8</td></tr>
<tr><td rowspan="4">uac-BarringInfoSetList</td><td colspan="4">Secuencia de 1 a 8 ejemplos</td></tr>
<tr><td colspan="3">uac-BarringFactor</td><td>[0-0.95] de 0.05 en 0.05</td></tr>
<tr><td colspan="3">uac-BarringTime</td><td>4, 8, 16, 32, 64, 128, 256, 512 s</td></tr>
<tr><td colspan="3">uac-BarringForAccessIdentity</td><td>7 bits</td></tr>
<tr><td rowspan="4">uac-AccessCategory1-SelectionAssistanceInfo</td><td colspan="4">Elegir</td></tr>
<tr><td colspan="3">plmnCommon</td><td>individualPLMNList</td></tr>
<tr><td colspan="3" rowspan="2">a, b, c</td><td>Secuencia de 2 a 12 ejemplos</td></tr>
<tr><td>a, b, c</td></tr>
</table>

De todas estas tablas que conforman los parámetros del mensaje SIB1 se va a hablar solo de los parámetros más importantes para la decodificación del mensaje SIB9 y la sincronización del UE con la BS.
En la tabla 3.9, el parámetro $cellAccesRelatedInfo$ permite enumerar las diferentes Public Land Mobile Network (PLMN).
En la tabla 3.10, el parámetro $schedulingInfoList$ nos da información sobre los diferentes mensajes SIBs. El parámetro $si - BroadcastStatus$ indica si el mensaje SIB del que se está informando se manda en periódicamente o tiene que ser solicitado para ser mandado. El parámetro $si - Periodicity$ indica la periodicidad de este mensaje SIB en tramas radio y el parámetro $si - MappingInfo$ nos da la información sobre qué mensaje SIB es del que se está refiriendo en los anteriores datos y si ha cambiado con respecto a la anterior transmisión del mismo, valor $valueTag$, este valor incrementa en 1 si ha cambiado y el UE debe actualizarlo. El valor $si - WindowLenght$ da la duración de la ventana de tiempo durante la cual se transmiten los mensajes SIBs.
La tabla 3.11 trata los parámetros del mensaje SIB1 que tratan sobre la parte de DL, la tabla 3.12 trata los parámetros sobre la transmisión UL y la tabla 3.13 trata más opciones sobre la transmisión de las señales. En concreto hay unos parámetros ($ssb - PositionsInBurst$ y $ss - PDCI - BlockPower$) que dan información sobre el SS/PBCH Block
La tabla 3.14 da información aplicable a la restricción de celdas basadas en el Unified Access Control (UAC).

El mensaje SIB9 da información sobre la hora UTC, la hora GPS y la hora local. Sus campos son los siguientes:

- $timeInfoUTC$. Especifica, en saltos de 10 ms, la hora UTC de la trama en la que se envía el propio SIB9.
- $dayLightSavingTime$. Es necesario debido al cambio de hora que se produce en los cambios de estación de algunos países, ya que la hora UTC no tiene estos cambios. Indica propiamente si se han de tener en cuenta esos cambios o no a la hora de decodificar la hora del sistema.
- $leapSeconds$. Permite calcular la hora GPS a partir de la hora UTC, ya que, la hora GPS es la suma del UTC y el $leapSeconds$.
- $localTimeOffset$. Permite calcular la hora local. Da la diferencia, en saltos de 15 minutos, entre la hora UTC y la hora local, esto es necesario debido a que no es la misma hora en todos los países a la vez.
- $referenceTimeInfo - r16$. Este campo fue añadido en la Release 16 del [17]. Este valor añade una granularidad de 10 ns a la hora UTC, por lo que permite mayor precisión.

El SIB9 se transmite en el canal lógico BCCH y dentro del canal físico PDSCH y bajo demanda del UE, como todos los mensajes SIBs que no sean el 1.

Un ejemplo de mensaje SIB9 real se muestra en la figura 3.11

```
{
  message c1: systemInformation: {
    criticalExtensions systemInformation: {
      sib-TypeAndInfo {
        sib9: {
          timeInfo {
            timeInfoUTC 395914372191,
            dayLightSavingTime '00'B,
            leapSeconds 18,
            localTimeOffset 0
          },
          referenceTimeInfo-r16 {
            time-r16 {
              refDays-r16 16599,
              refSeconds-r16 36539,
              refMilliSeconds-r16 910,
              refTenNanoSeconds-r16 0
            }
          }
        }
      }
    }
  }
}
```

Figura 3.11: Estructura general del SS/PBCH Block.

## 3.5. Sincronización en 5G

En esta sección se describen los bloques y señales utilizados para la sincronización temporal, en frecuencia y en fase entre la BS y el UE.

### 3.5.1. SS/PBCH BLOCKS AND BURSTS

El SS Burst está compuesto por tantos SS Blocks como haces se generen en el transmisor. Cada SS Block contiene las señales necesarias para establecer la sincronización entre el terminal y la estación base. Concretamente, incluye la señal PSS, la señal SSS y el canal PBCH, como ya se explicó en la sección 3.3.1 [14, 12, 18].

Cada SS Block se transmite a lo largo de 4 símbolos en el dominio temporal y ocupa 20 bloques de recursos en frecuencia, es decir, 240 subportadoras (12 subportadoras por bloque).

La estructura general del SS/PBCH Block se muestra en la Figura 3.12:

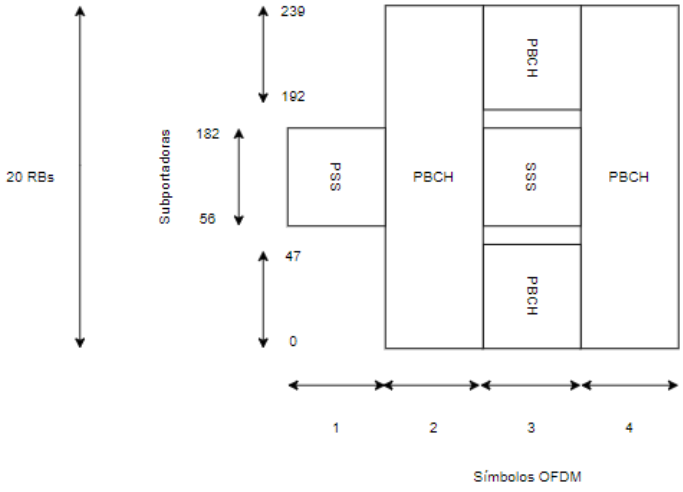



Figura 3.12: Estructura general del SS/PBCH Block.

La periodicidad de transmisión de estos bloques puede variar dependiendo de la configuración establecida en las especificaciones del 3GPP.

#### PSS

La señal PSS es la primera que recibe el terminal dentro del SS/PBCH Block. Está ubicada en el primer símbolo y ocupa las 127 subportadoras centrales, tal como se observa en la Figura 3.12.

Esta señal consiste en una secuencia predeterminada que facilita su detección por parte del receptor. Su función principal es:

- Corregir errores en frecuencia.
- Determinar el instante temporal en el que se transmitió el SS Block.
- Contribuir a la identificación parcial del sector de la BS, combinándose con la información proporcionada por la SSS.

El número de secuencias posibles de PSS depende del modo de operación del PCI. Un ejemplo para el modo 3 se presenta en la Figura 3.13:




Figura 3.13: Ejemplo de asignación de PCI mod 3.

### SSS

La señal SSS es la segunda componente clave para la sincronización entre el UE y la BS. Se ubica en el tercer símbolo del SS/PBCH Block y ocupa también las 127 subportadoras centrales, como se aprecia en la Figura 3.12.

La SSS es una secuencia predeterminada de 127 símbolos que permite:

- Determinar de manera precisa el identificador de celda y de sector al que se conecta el UE.
- Contribuir, junto con la señal DMRS del PBCH, a la estimación del canal radioeléctrico.

# Capítulo 4

# Planificación y estimación de costes

En este capítulo, se va a presentar tanto la planificación temporal como la estimación de costes del Trabajo Fin de Máster.
La planificación temporal ofrece una visión inicial de cómo se ha organizado el desarrollo del proyecto, qué tareas se han llevado a cabo y cuánto tiempo se ha dedicado aproximadamente a cada una de ellas.
Por otro lado, en la sección de estimación de costes se realiza un desglose de los recursos utilizados, tanto materiales como humanos junto con el presupuesto asociado a su uso, calculado en base al tiempo de dedicación o a la amortización correspondiente.

El objetivo de este capítulo es el de realizar una planificación temporal a priori y de los medios necesarios para la realización de este trabajo, lo cual resulta especialmente relevante en entornos reales donde este tipo de proyectos pueden ser replicados, presupuestados o escalados.

## 4.1. Planificación de tareas

En esta sección se van a explicar las tareas que se han llevado a cabo durante la realización de este TFM. Todas estas tareas están enumeradas a continuación.

1. Estudio sobre 5G.
   Para poder trabajar con señales 5G, primero hay que conocer el escenario en el que se desarrolla esta tecnología. En esta tarea se ha realizado una búsqueda de artículos, documentación técnica y proyectos relacionados, con el objetivo de familiarizarme con el entorno 5G y la sincronización temporal.

2. Estudio de la señal 5G.

Dado que el trabajo se centra en señales 5G, ha sido necesario entender cómo se transmiten, cuál es su estructura y qué información contienen. Esta tarea ha consistido en el estudio detallado de estos aspectos.

3. Familiarización con el módem Telit FN990.
   El dispositivo FN990 es el principal módem utilizado durante el proyecto, siendo uno de los pocos módem comerciales que permite la obtención del SIB9. Por tanto, se ha dedicado tiempo a comprender su funcionamiento, los modos de operación y las interfaces que expone. Para ello, se ha trabajado con la guía de comandos AT, se ha configurado la comunicación mediante USB en un entorno Linux y se han realizado pruebas básicas de conectividad y consulta de parámetros. Esta tarea ha sido clave para poder interactuar con el módulo y extraer los mensajes necesarios para el análisis del SIB9.

4. Familiarización con la estación base de Amarisoft.
   El hardware/software empleado para generar y transmitir señales 5G ha sido la estación base de Amarisoft. Por ello, ha sido necesario comprender su funcionamiento, su configuración básica y los parámetros relevantes para este proyecto.

5. Estudio de diferentes software para el control del módem Telit FN990.
   Se han probado distintas formas de interactuar con el módulo FN990, tanto en sistemas Windows como en Linux. Finalmente, se ha comprobado que el entorno Linux ofrece mayor compatibilidad y facilidad para el control del módem mediante comandos AT.

6. Captura del SIB9 y tratamiento del mismo.
   Una vez entendido cómo capturar el SIB9, tanto en redes comerciales como con la estación base de Amarisoft, se ha procedido a su obtención y posterior análisis. Se ha trabajado con su contenido para extraer la información necesaria de cara a su uso en el proyecto.

7. Escritura de la memoria.
   En esta última tarea se ha redactado la memoria del TFM. También se ha preparado la presentación para la defensa y se ha trabajado en la propia preparación de la exposición final.

En la tabla 4.1 podemos ver la planificación en horas para este proyecto.

Y en la figura 4.1 se muestra el diagrama de Gantt que se pretende seguir para la realización del proyecto.

Tabla 4.1: Tabla resumen de las horas dedicadas al TFM.

| **ID** | **Descripción de la tarea** | **Tiempo (h)** |
|---|---|---|
| 1 | Estudio sobre 5G | 36 |
| 2 | Estudio señal 5G. | 154 |
| 3 | Familiarización con el módem FN990. | 100 |
| 4 | Familiarización con la estación base de Amarisoft. | 60 |
| 6 | Estudio de diferentes software para el control del FN990. | 55 |
| 7 | Captura del SIB9 y tratamiento del mismo. | 200 |
| 8 | Escritura de memoria. | 150 |
| Tiempo Total | | 755 |

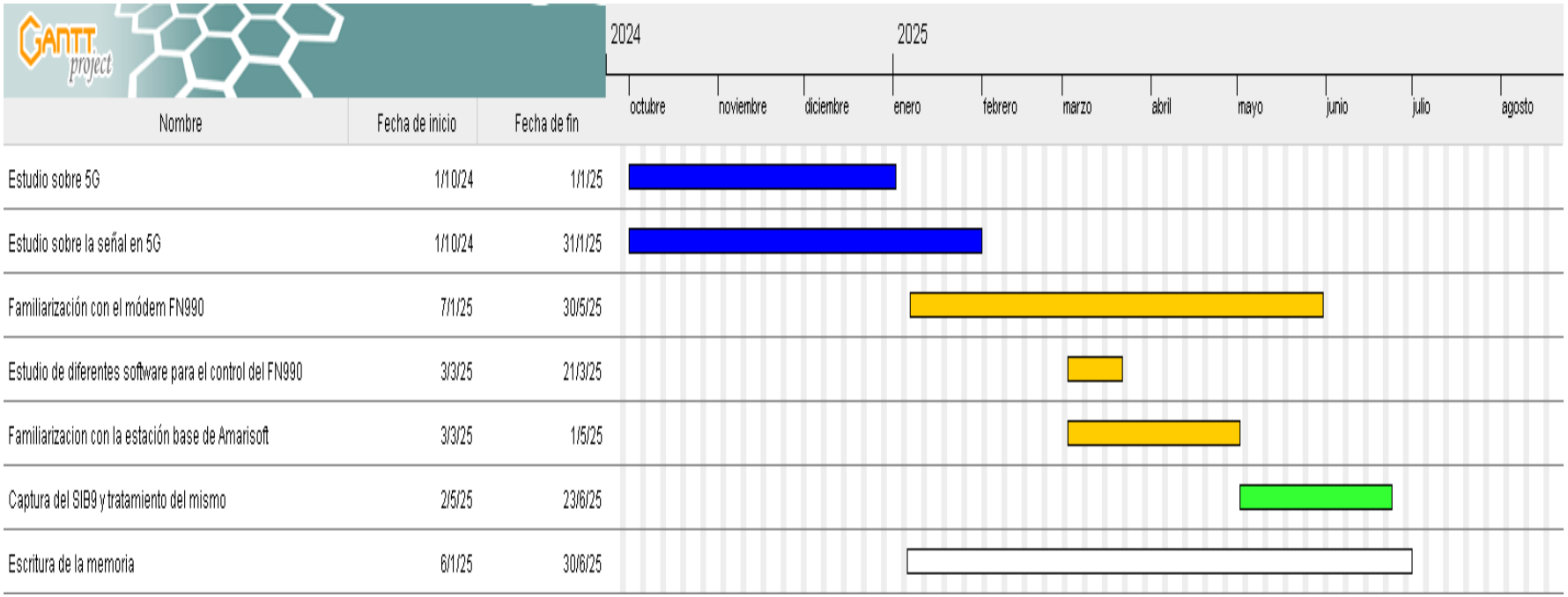



Figura 4.1: Diagrama de Gantt.

## 4.2. Planificación de recursos

Se analizarán y desglosarán los tres tipos de recursos usados, el hardware, el software y el humano.

### 4.2.1. Recursos hardware.

Como recursos hardware tenemos el portátil usado para escribir la memoria y para poder realizar todo el proyecto y los dispositivos usados para la transmisión y recepción de las señales, tales como el módem FN990, las antenas y la estación base de Amarisoft.
Las características del portátil se pueden ver en la tabla 4.2.1.

Tabla 4.2: Características del portátil.

| Procesador 11th Gen Intel(R) Core(TM) i5-1135G7 |
|---|
| Memoria RAM de 8 GB |
| Tarjeta gráfica Gráficos Intel UHD |
| Capacidad de 512 GB |

### 4.2.2. Recursos software.

Los recursos software usados son los comandos AT y los ejecutables usados para poder recibir y procesar el SIB9 5.2. Todos estos comandos han sido usados en Linux 5.3. Además de Wireshark y la GUI de Amarisoft 5.4.

### 4.2.3. Recursos humanos.

Los recursos humanos son las personas que han ayudado a la realización de este proyecto, que son Jorge Navarro, Félix Delgado y el alumno, Ángel Gómez. Estos recursos están descritos en la tabla 4.3.

Tabla 4.3: Recursos humanos necesarios para la elaboración del TFM.

| Persona | Horas | Coste/Hora |
|---|---|---|
| Ángel | 755 | 25 |
| Felix | 20 | 40 |
| Jorge | 20 | 50 |

## 4.3. Presupuesto

En esta sección se mostrará el coste total de los tres tipos de recursos utilizados durante el desarrollo del proyecto: hardware, software y recursos

humanos.

El portátil usado para la realización del proyecto ha sido el equipo principal para escribir la memoria y ejecutar herramientas auxiliares. Su coste es de 850 euros, con una vida útil estimada de 4 años.

La estación base de Amarisoft ha sido el equipo utilizado para generar señales 5G controladas. Su precio de adquisición es de 15000 euros, con un periodo de amortización de 3 años.

El módem Telit FN990 ha sido el principal dispositivo para la recepción y análisis de las señales 5G y para la ejecución de comandos AT. Su coste es de 683,5 euros y se estima una vida útil de 3 años.

En la tabla 4.4 se presenta el coste proporcional estimado que ha supuesto el uso de cada uno de estos recursos.

Tabla 4.4: Estimación del coste proporcional de uso de cada componente.

| **Herramienta** | **Precio unidad (€)** | **Vida útil (años)** | **Tiempo usado (meses)** | **Coste estimado (€)** |
|---|---|---|---|---|
| Portátil | 850 | 4 | 9 | 159,38 |
| Amarisoft | 15000 | 3 | 1,5 | 625 |
| FN990 | 683,5 | 3 | 4 | 75,90 |

A continuación, se muestra el presupuesto total agrupando todos los recursos utilizados durante el proyecto.

Tabla 4.5: Presupuesto final.

| **Herramienta** | **Precio total (€)** | **Unidades** | **Subtotal (€)** |
|---|---|---|---|
| Portátil | 159,38 | 1 | 159,38 |
| Amarisoft | 625 | 1 | 625 |
| FN990 | 75,90 | 1 | 75,90 |
| Ángel Gómez | 25 | 755 | 18.875 |
| Jorge Navarro | 20 | 50 | 1.000 |
| Félix Delgado | 20 | 40 | 800 |
| **TOTAL (€)** | | | **21.535,28** |

Por lo tanto, como se puede observar en la tabla 4.5, el presupuesto total estimado del proyecto es de **21.535,28 euros**.

# Capítulo 5

# Herramientas usadas.

En este capítulo se detallan las principales herramientas y dispositivos que se han utilizado para llevar a cabo el análisis y estudio de las señales 5G, concretamente orientado al mensaje SIB9. Se destacan principalmente el hardware utilizado para la captura de señales y los entornos de desarrollo y análisis para procesar dichas señales.

## 5.1. Telit EVB 2.0

La placa de evaluación Telit EVB 2.0 [19] es una herramienta fundamental en este trabajo, pues permite la interacción y pruebas con módulos de comunicación celular, específicamente de la serie FN990 de Telit. La placa facilita la integración y prueba rápida de funciones principales del módulo, desde la gestión de alimentación hasta la configuración avanzada mediante comandos AT. Este equipo cuenta con múltiples interfaces como Universal Serial Bus (USB), Universal Asynchronous Receiver-Transmitter (UART) y soporte Subscriber Identity Module (SIM) dual, que han sido esenciales para las pruebas con redes 5G.

Entre sus características más destacadas están:

- Soporte SIM dual.
- Interfaces USB para comunicación con PC.
- Múltiples General Purpose Input/Output (GPIO) para pruebas y depuración.

## 5.2. Telit FN990

El módulo FN990 es uno de los elementos clave en este estudio debido a sus capacidades de comunicación en redes 5GNR, incluyendo soporte para funcionalidades avanzadas como sincronización temporal a través del

mensaje SIB9. Este módulo ha sido gestionado mediante comandos AT, permitiendo configurar aspectos como la selección de bandas, la gestión de red, y particularmente, la captura del mensaje SIB9.

Para la configuración y control del módulo FN990, se ha empleado extensamente el conjunto de comandos AT definidos en su manual. Estos comandos permiten gestionar aspectos críticos del módulo, desde la configuración inicial hasta funciones específicas como la sincronización temporal, la selección de operadores y bandas, resultando esenciales en el análisis y prueba del mensaje SIB9.

En la Figura 5.1 se muestra el módulo Telit FN990 empleado durante todo el proceso de experimentación y análisis.




Figura 5.1: Módulo Telit FN990 utilizado.

Por motivos de confidencialidad, no se puede incluir en este documento el código fuente realizado con la API de Qualcomm para el módem, ya que su uso está sujeto a un acuerdo de no divulgación (Non-Disclosure Agreement (NDA)) firmado durante el desarrollo del trabajo. Este código ha sido fundamental para la implementación y prueba del sistema, pero, la API usada es propiedad del fabricante y su distribución está restringida. En su lugar, se

describen en detalle los procedimientos realizados y los resultados obtenidos a partir de su ejecución.

## 5.3. Ubuntu/Linux

El sistema operativo Ubuntu (Linux) [20] ha sido utilizado para ejecutar comandos AT y programas específicos desarrollados en este trabajo. Su estabilidad y versatilidad en entornos de desarrollo ha facilitado enormemente la interacción con las herramientas hardware, especialmente para la comunicación y control mediante comandos AT de los módulos Telit. Además, la capacidad de Ubuntu para gestionar eficientemente procesos y comunicaciones seriales ha resultado fundamental para el análisis detallado del mensaje SIB9.

En conjunto, todas estas herramientas han proporcionado una base robusta y completa para llevar a cabo el análisis detallado y riguroso que se presenta en este trabajo. Cada una de ellas aporta un valor específico que, combinado, permite abordar el estudio de la sincronización en redes 5G con precisión y eficiencia.

## 5.4. Amarisoft

Amarisoft [21] es un software que permite implementar tanto una estación base 4G y 5G así como la red troncal. En este proyecto se configurará para implementar una estación base 5G y su red troncal. Tiene un fichero de configuración para el gNB que permite cambiar parámetros tales como la frecuencia, el ancho de banda y el número de antenas en la transmisión, entre otros. Permite también añadir los mensajes SIBs 2, 3, 4 y 9 a la transmisión. La estación base de Amarisoft Callbox Classic[22], se puede ver en la figura 5.2.

Para este trabajo, usaremos la release 16. La cual aporta mayor granularidad temporal, tal y como se contó en la sección 3.4.



Figura 5.2: Estación base Amarisoft.

# Capítulo 6

# Preparación del entorno de pruebas

Antes de poder obtener resultados reales sobre la sincronización temporal y la decodificación del mensaje SIB9, es necesario realizar una preparación exhaustiva del entorno de trabajo. Este proceso no solo implica la conexión física del hardware, sino también una comprensión profunda del funcionamiento interno del módem, el entorno software necesario y el flujo de trabajo adecuado para interactuar con el dispositivo de forma controlada y fiable.

A diferencia de los entornos simulados o de herramientas de propósito específico, trabajar con hardware real, como el módulo *Telit FN990* junto a su Evaluation Board (EVB), requiere una comprensión previa detallada de su arquitectura interna, de los mecanismos de comunicación y de los procedimientos adecuados de configuración. Interactuar con este tipo de dispositivos sin el conocimiento necesario puede derivar en fallos operativos, resultados erróneos o incluso en situaciones de bloqueo del sistema. Por tanto, este capítulo recoge de forma estructurada todos los pasos seguidos para habilitar un entorno de pruebas funcional, estable y fiable, que permita obtener resultados válidos en el análisis posterior.

## 6.1. Diseño del entorno de pruebas

El diseño del entorno de pruebas constituye la base sobre la que se apoya toda la experimentación realizada en este trabajo. Para poder interactuar con un módem 5G real y acceder a los mensajes de sincronización transmitidos por la red, ha sido necesario definir una arquitectura que combine tanto elementos hardware como software de forma coordinada y coherente.

Desde el punto de vista hardware, el sistema se apoya en un módulo FN990 montado sobre una EVB, lo que permite acceder a múltiples interfaces del dispositivo y realizar tareas de monitorización, configuración y control. Este conjunto se conecta a un ordenador a través de interfaces USB,

que se utilizan tanto para la comunicación de comandos AT como para el transporte de datos de usuario y señales auxiliares.

En el plano software, se han empleado herramientas específicas proporcionadas por el fabricante, entre las que se incluyen controladores *Windows* para la gestión de puertos, paquetes de utilidades para la comunicación por consola y documentación técnica asociada. Todo ello ha sido integrado sobre un entorno de trabajo basado en *Ubuntu*, donde se ha preparado un flujo de ejecución que permite inicializar el módem, configurar el perfil de red y recoger los mensajes transmitidos.

### 6.1.1. Configuración y puesta en marcha

El módulo FN990 utilizado se encuentra montado sobre su correspondiente EVB, lo que facilita el acceso a sus distintas interfaces funcionales sin necesidad de modificaciones adicionales. La alimentación del conjunto se realiza mediante una fuente externa conectada a la red eléctrica. Para la comunicación con el portátil se emplean dos enlaces USB, que permiten interactuar con el módem desde el entorno de desarrollo mediante el envío de comandos y la monitorización de su comportamiento interno.

El dispositivo cuenta con cuatro antenas de tipo wide-band conectadas, garantizando cobertura para los diferentes rangos de frecuencia empleados en redes 5G. Adicionalmente, se han utilizado dos tarjetas SIM diferentes, con el objetivo de evaluar el funcionamiento del sistema en entornos de prueba diversos, lo cual será analizado en capítulos posteriores.

Desde el punto de vista software, se ha habilitado una partición específica con sistema operativo *Ubuntu*, desde la cual se ha ejecutado todo el proceso de interacción con el módulo. La comunicación por consola para los comandos AT se ha realizado mediante la herramienta `minicom`, sin necesidad de configuraciones adicionales ni instalación de controladores específicos, dado que el sistema detecta correctamente las interfaces del dispositivo al conectarlo.

Se documenta en la figura 6.1 el entorno software utilizado durante las pruebas, con el fin de garantizar la trazabilidad y reproducibilidad del proceso. La versión del kernel de *Linux* empleada se obtuvo mediante el comando `uname -r`, mientras que la distribución específica del sistema se verificó con `lsb_release -a`, como se aprecia en la figura 6.1.

## 6.2. Interacción con el módem y comandos AT

Una vez configurado el entorno físico y establecida la conexión entre el módem y el sistema anfitrión, el siguiente paso consiste en acceder a la interfaz de control del dispositivo para verificar que se encuentra operativo y en condiciones de iniciar las pruebas. Para ello se emplea la herramienta

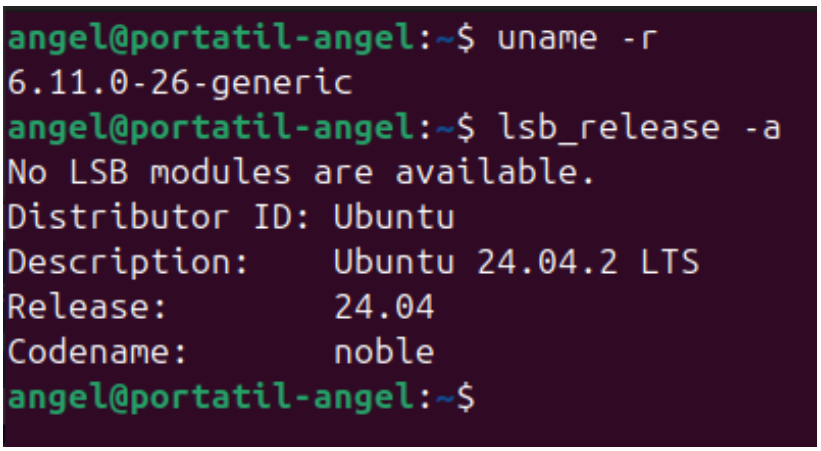



Figura 6.1: Versiones del kernel y de Ubuntu usados en la realización de este proyecto.

`minicom`, accediendo a uno de los puertos `/dev/ttyUSBx` que expone el dispositivo al ser conectado por USB. Para saber exactamente que puertos USB están habilitados para el intercambio de comandos AT, se recomienda el uso del comando `mmcli -L` en la consola, que da como resultado lo mostrado en la figura 6.6, como se verá posteriormente.

Una vez dentro del terminal, se ejecuta una secuencia de comandos AT básicos con el objetivo de comprobar el estado del módem, la presencia de la tarjeta SIM y la correcta inserción del PIN. Esta rutina permite confirmar que el módulo está inicializado correctamente, que tiene acceso a la red móvil y que responde a las órdenes desde el entorno de usuario. En la figura 6.2 se muestra un ejemplo de esta interacción inicial.

Se puede observar que el módem responde a los comandos de forma esperada, mostrando información de identificación del modelo y confirmando que la tarjeta SIM está disponible y desbloqueada.

- `AT`: Comando de prueba para verificar que el módem responde correctamente. Si el módem está operativo, la respuesta será `OK`.
- `AT+CPIN?`: Consulta el estado de la tarjeta SIM. Si está protegida por PIN y aún no se ha desbloqueado, la respuesta será `+CPIN: SIM PIN`. Si ya se ha desbloqueado correctamente, la respuesta será `+CPIN: READY`.
- `AT+CPIN="XXXX"`: Introduce el PIN de la tarjeta SIM. En este ejemplo, se proporciona el valor `XXXX` como código. Si el PIN es correcto, la SIM quedará desbloqueada y se podrá acceder a la red móvil, se oculta el valor del PIN por temas de privacidad. Habria que sustituir XXXX por el valor real.

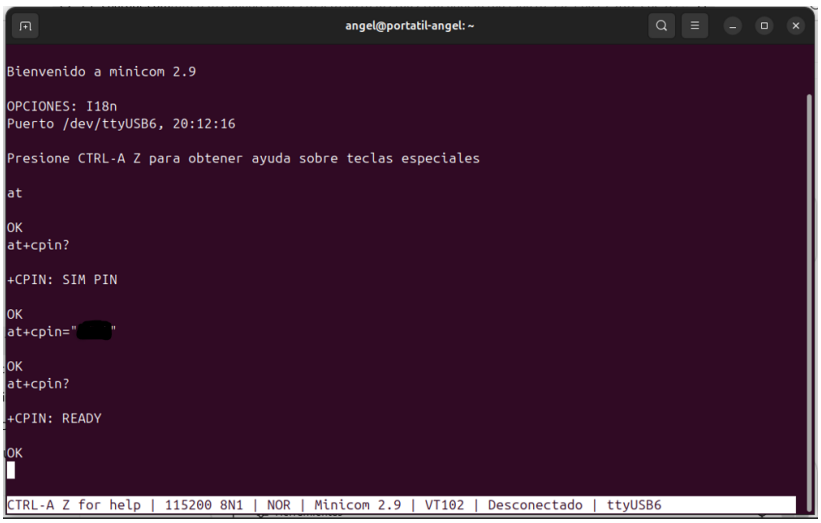



Figura 6.2: Comprobaciones iniciales mediante comandos AT tras acceder por `minicom`.

- `AT+CPIN?` (segunda vez): Se repite el comando para confirmar que la tarjeta SIM ha sido desbloqueada correctamente. La respuesta `+CPIN: READY` indica que la SIM está lista para su uso.

Una vez verificado el estado inicial, se lanza el comando `AT#SERVINFO` para obtener información más detallada sobre la celda a la que se encuentra conectado el módem. En este punto, la salida refleja una conexión activa en tecnología Long Term Evolution (LTE), tal y como se aprecia en la figura 6.3.

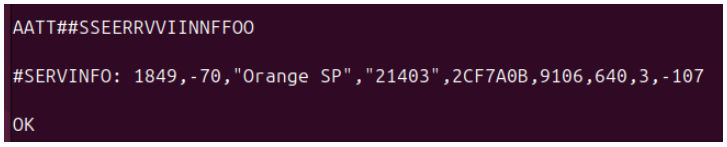



Figura 6.3: Estado inicial del módem conectado a una red LTE.

Para facilitar la interpretación de los campos presentes en la salida del comando `AT#SERVINFO`, se incluye en la figura 6.4 un ejemplo tomado del manual de referencia de comandos AT del fabricante, en el que se detallan las distintas métricas asociadas a la celda servidora.

La salida obtenida es la siguiente:

```
#SERVINFO: 1849,-70,"Orange SP","21403",2CF7A0B,9106,640,3,-107
```

Cada campo puede interpretarse de la siguiente manera, según la documentación oficial del fabricante:




Figura 6.4: Ejemplo de salida del comando `AT#SERVINFO` según la documentación oficial del fabricante.

- **EARFCN** (*E-UTRA Absolute Radio Frequency Channel Number*): `1849`. Indica el canal de frecuencia LTE asignado.
- **dBm**: `-70`. Nivel de señal recibido, en decibelios.
- **NetNameAsc**: `.⁰range SP"`. Nombre de la red a la que está conectado el módem.
- **NetCode**: `"21403"`. Código de red, compuesto por el MCC (Mobile Country Code) y el MNC (Mobile Network Code).
- **CID** (*Cell Identifier*): `2CF7A0B`. Identificador único de la celda servidora.
- **TAC** (*Tracking Area Code*): `9106`. Código del área de seguimiento asignado a la celda.
- **DRX** (*Discontinuous Reception cycle length*): `640`. Parámetro que define el ciclo de escucha del módem en reposo.
- **SD** (*Service Domain*): `3`. Indica los servicios disponibles en la celda; en este caso, tanto CS (Circuit Switched) como PS (Packet Switched).
- **RSRP** (*Reference Signal Received Power*): `-107`. Potencia recibida de la señal de referencia, usada para medir la calidad de la conexión LTE.

### 6.2.1. Conexión a la red 5G

Tal y como se ha observado anteriormente, el módem establece por defecto una conexión a la red LTE, incluso en entornos con cobertura NR disponible. Esto no se puede cambiar con comandos AT, al menos en este modelo. Por ello, ha sido necesario emplear funciones internas que permiten modificar directamente las preferencias de conexión a la red del modem.

En concreto, se ha utilizado un archivo fuente llamado "nas_set_system_selection_preference.c", incluido entre los ejemplos disponibles en el paquete de utilidades proporcionado por el fabricante. Este archivo contiene una estructura de configuración en la que se define la red preferida mediante una variable clave: ïnput.mode_pref". Por defecto, esta variable está configurada como "QMI_NAS_RAT_MODE_PREF_LTE", lo que da prioridad a la conexión LTE. Para forzar la selección de red NR, se ha modificado esta línea por "QMI_NAS_RAT_MODE_PREF_NR5G", según lo indicado en la documentación técnica del fabricante.

Una vez modificado el archivo, se procede a compilarlo utilizando un sistema de construcción basado en *Makefile*, también incluido en el paquete de ejemplos. Este proceso genera un ejecutable que puede lanzarse desde terminal sin necesidad de permisos especiales.

Tras ejecutar este ejecutable, se reinició el módem con el fin de aplicar correctamente la nueva configuración de prioridades de red. Al consultar de nuevo el estado de conexión mediante el comando `AT#SERVINFO`, se observó que el módem había establecido conexión con una red NR, tal y como se muestra en la figura 6.5.

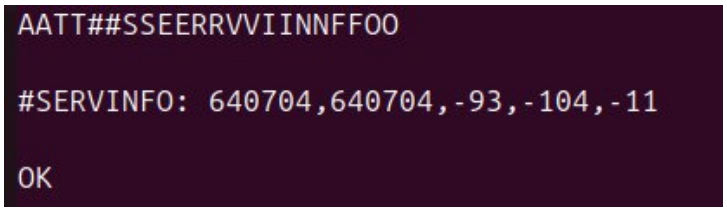



Figura 6.5: Estado de la conexión a la red 5G.

La estructura de la salida coincide con el formato definido para conexiones NR en modo SA de la figura 6.4, compuesto por cinco parámetros. A continuación se detalla el significado de cada uno de ellos:

```
#SERVINFO: 640704, 640704, -93, -104, -11
```

- **NR_CH** (*NR Downlink Active Channel*): `640704`. Indica el canal de bajada activo en la conexión 5G.
- **NR_ULCH** (*NR Uplink Active Channel*): `640704`. Canal de subida activo, que en este caso coincide con el de bajada.

- **NR_RSSI** (*NR Received Signal Strength Indicator*): `-93`. Potencia total de la señal recibida en dBm.
- **NR_RSRP** (*Reference Signal Received Power*): `-104`. Nivel de potencia del canal de referencia 5G.
- **NR_RSRQ** (*Reference Signal Received Quality*): `-11`. Métrica que relaciona la potencia del canal de referencia con el nivel general de interferencia y ruido.

Hasta aquí, se han explicado los pasos a seguir para la conexión a una red 5G cualquiera. Para la realización de este TFM, se va a trabajar, como ya se ha mencionado anteriormente, con la estación base de Amarisoft. Esto nos permite conocer qué ocurre también en el operador de la red 5G. A continuación, se muestra una configuración extra que se puede hacer una vez conectado el módem a la red de Amarisoft.

Una vez configuradas las preferencias del módem, se emplea la herramienta `mmcli` para verificar el estado de registro en la red móvil. En una primera consulta, el módem aparece como `registered` y conectado a tecnología NR, pero sin haber establecido aún una sesión de datos, como se observa en la figura 6.6.

```
angel@portatil-angel:~$ mmcli -m 18
  ---------------------------------
  General  |                 path: /org/freedesktop/ModemManager1/Modem/18
           |           device id: f696f6c527369119c13078e1b96477a796b2c479
  ---------------------------------
  Hardware |         manufacturer: Telit
           |                model: FN990A28
           |    firmware revision: M0R.100005
           |       carrier config: default
           |         h/w revision: 1.00
           |            supported: gsm-umts, lte, 5gnr
           |              current: gsm-umts, lte, 5gnr
           |         equipment id: 359918920062833
  ---------------------------------
  System   |               device: /sys/devices/pci0000:00/0000:00:14.0/usb2/2-1
           |              physdev: /sys/devices/pci0000:00/0000:00:14.0/usb2/2-1
           |              drivers: qmi_wwan, option
           |               plugin: telit
           |         primary port: cdc-wdm0
           |                ports: cdc-wdm0 (qmi), ttyUSB0 (ignored), ttyUSB1 (gps),
           |                       ttyUSB2 (at), ttyUSB3 (at), ttyUSB4 (ignored), wwan0 (net)
  ---------------------------------
  Status   |                 lock: sim-pin2
           |       unlock retries: sim-pin (3), sim-puk (10), sim-pin2 (3), sim-puk2 (10)
           |                state: registered
           |         power state: on
           |         access tech: 5gnr
           |       signal quality: 50% (recent)
```

Figura 6.6: Estado del módem registrado en red NR, sin sesión de datos activa.

Para completar la conexión, se ejecuta el comando:

```
mmcli -m 18 --simple-connect="apn=ims"
```

Tras la ejecución satisfactoria del comando, se verifica nuevamente el estado mediante `mmcli -m 18`. La salida, mostrada en la figura 6.7, indica

el estado `connected`, con tecnología de acceso `5gnr`, confirmando así que el módem se encuentra completamente conectado y preparado para recibir tráfico de red y, por tanto, el SIB9.

```
angel@portatil-angel:~$ mmcli -m 18
  ---------------------------------
  General  |                 path: /org/freedesktop/ModemManager1/Modem/18
           |           device id: f696f6c527369119c13078e1b96477a796b2c479
  ---------------------------------
  Hardware |         manufacturer: Telit
           |                model: FN990A28
           |     firmware revision: M0R.100005
           |        carrier config: default
           |          h/w revision: 1.00
           |            supported: gsm-umts, lte, 5gnr
           |              current: gsm-umts, lte, 5gnr
           |          equipment id: 359918920062833
  ---------------------------------
  System   |               device: /sys/devices/pci0000:00/0000:00:14.0/usb2/2-1
           |              physdev: /sys/devices/pci0000:00/0000:00:14.0/usb2/2-1
           |              drivers: qmi_wwan, option
           |               plugin: telit
           |         primary port: cdc-wdm0
           |                ports: cdc-wdm0 (qmi), ttyUSB0 (ignored), ttyUSB1 (gps),
           |                       ttyUSB2 (at), ttyUSB3 (at), ttyUSB4 (ignored), wwan0 (net)
  ---------------------------------
  Status   |                 lock: sim-pin2
           |       unlock retries: sim-pin (3), sim-puk (10), sim-pin2 (3), sim-puk2 (10)
           |                state: connected
           |          power state: on
           |          access tech: 5gnr
           |       signal quality: 50% (recent)
```

Figura 6.7: Estado del módem conectado en red NR con sesión de datos activa.

Todo el proceso de registro del UE a la BS es registrado en los logs de Amarisoft. En las figuras 6.8 y 6.9, podemos ver este proceso de registro. Se pueden ver mensajes como el *Registration request*, los mensajes de autenticación y los mensajes del establecimiento de la conexión.

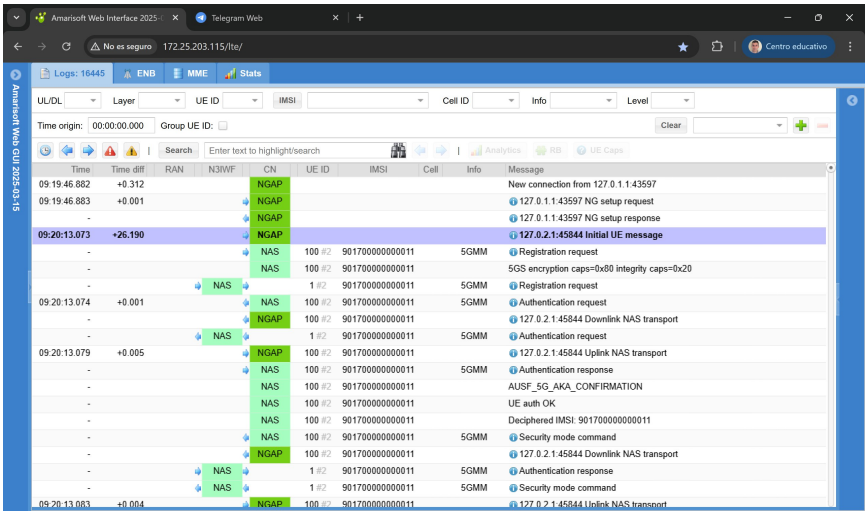



Figura 6.8: Captura 1 logs de Amarisoft.

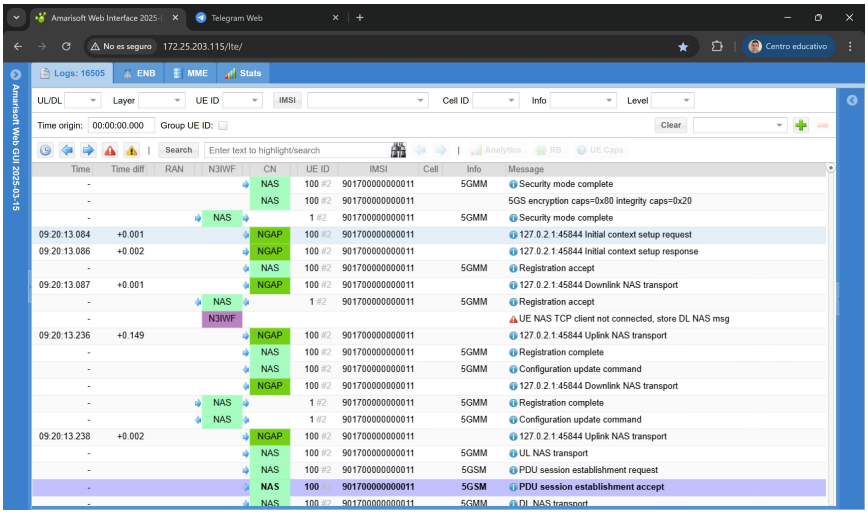



Figura 6.9: Captura 2 logs de Amarisoft.

# Capítulo 7

# Resultados y análisis

En este capítulo se presentan los resultados obtenidos tras la ejecución del programa desarrollado en lenguaje C, cuyo propósito es alcanzar y decodificar el mensaje SIB9. Como se explicó en capítulos anteriores, dicho programa encapsula todo el procedimiento necesario para acceder a este bloque, desde la sincronización física con la celda hasta la decodificación de los canales de control y compartidos.

A continuación, se muestran los resultados extraídos directamente de la ejecución del programa en dos entornos diferenciados: una red 5G privada basada en Amarisoft y una red comercial proporcionada por un operador. En cada caso, se documentan los valores obtenidos del SIB9 y se analiza su validez para la sincronización temporal del dispositivo.

## 7.1. Red privada con estación base Amarisoft

### 7.1.1. Configuración de Amarisoft

Antes de mostrar los resultados obtenidos, es necesario detallar cómo se configuró la estación base Amarisoft para asegurar la transmisión del mensaje SIB9. Como se ha comentado anteriormente, al tratarse de una red privada, se dispone de control completo sobre los parámetros de emisión, lo que permite incluir bloques opcionales del sistema como el SIB9.

La configuración se basa en los esquemas definidos en la *Release 16* del estándar 3GPP. Concretamente, se habilitó el parámetro `sibType9` dentro del campo `systemInformationBlockTypeList` en el archivo de configuración de la celda. Esta inclusión es imprescindible, ya que el SIB9 es un bloque opcional cuyo envío no está habilitado por defecto.

Gracias a esta configuración explícita, la estación base Amarisoft transmite periódicamente el mensaje SIB9, permitiendo que el UE lo detecte y procese. Esta capacidad de personalizar la configuración de red es una de las principales ventajas del uso de redes privadas para experimentación.

### 7.1.2. Resultados

En esta sección se mostrarán los resultados obtenidos tras la ejecución del programa cuando el módem estaba conectado a la red 5G configurada según lo descrito en la sección 7.1.1.

Tal y como se ha comentado anteriormente, por cláusulas de privacidad, no se puede mostrar el código en sí. Por tanto, en la figura 7.1 se muestra únicamente una parte de la salida generada por el programa. En ella, se

```
[1750154901.489728252] INFO: sfn = 600
[1750154901.489789281] INFO: nta = 0
[1750154901.489792821] INFO: nta_offset = 0
[1750154901.489796888] INFO: leapseconds = 18
[1750154901.489799699] INFO: utc_time = 3959143721509999999
[1750154902.736716880] INFO: sfn = 728
[1750154902.736743915] INFO: nta = 0
[1750154902.736747424] INFO: nta_offset = 0
[1750154902.736750326] INFO: leapseconds = 18
[1750154902.736753019] INFO: utc_time = 3959143722789999999
[1750154904.017806108] INFO: sfn = 856
[1750154904.017833661] INFO: nta = 0
[1750154904.017837121] INFO: nta_offset = 0
[1750154904.017839993] INFO: leapseconds = 18
[1750154904.017842668] INFO: utc_time = 3959143724069999999
[1750154905.297851545] INFO: sfn = 984
[1750154905.297877656] INFO: nta = 0
[1750154905.297881089] INFO: nta_offset = 0
[1750154905.297884035] INFO: leapseconds = 18
[1750154905.297886848] INFO: utc_time = 3959143725349999999
[1750154906.577479221] INFO: sfn = 88
[1750154906.577506143] INFO: nta = 0
[1750154906.577509612] INFO: nta_offset = 0
[1750154906.577512513] INFO: leapseconds = 18
[1750154906.577515217] INFO: utc_time = 3959143726629999999
```

Figura 7.1: Capturas SIB9 desde UE.

pueden ver los campos explicados en la sección 3.4. Existen dos campos de interés para nuestro análisis, el *sfn* y el *utc_time*. A la izquierda de cada campo, se puede observar el *timestamp*, que es la hora del reloj interno del portátil cuando muestra la salida.

El campo *sfn* nos está diciendo el número de la trama que se recibe, número que analizaremos más tarde, y el campo *utc_time* es el que nos da el tiempo del reloj maestro en UTC.

Para el análisis de todos los datos extraídos se ha utilizado la programación de Python. En nuestro caso, son muy interesantes dos análisis. El primero, saber que hora nos está dando el SIB9 obtenido. Y, el segundo, saber cuanto tiempo tarda el código en mostrar este mensaje.

Para ello, se ha creado una función de python (figura 7.2), que transforma el valor UTC a horario legible. Como se comentó, la estación base de Amarisoft trabaja usando la realease 16 del 3GPP, por lo tanto, el UTC se envía con una granularidad de 10 nanosegundos.

```python
def utc_time_to_datetime(utc_ns):
    """
    Convierte el campo utc_time del SIB9 (en nanosegundos desde 1900-01-01) a tiempo legible.
    """
    base_date = datetime(1900, 1, 1)
    utc_sec = utc_ns / 1e9  # de nanosegundos a segundos
    return base_date + timedelta(seconds=utc_sec)
```

Figura 7.2: Función de python para la conversión del tiempo UTC a horario legible.

Tras realizar un análisis de los datos obtenidos de la figura 7.1 con la función de pyhton 7.2 en el código mostrado en la figura 7.3, se obtiene la salida parcial de la tabla 7.1.

```python
from datetime import datetime, timedelta
import pandas as pd
import re
# Ruta al archivo
ruta = 'registro_SIB9.txt'

# Lista donde guardaremos los datos
datos = []

# Expresiones regulares para extraer info
pattern = re.compile(r"\[(?P<timestamp>[\d.]+)\].*?utc_time = (?P<utc_time>\d+)")

with open(ruta, 'r') as f:
    for linea in f:
        match = pattern.search(linea)
        if match:
            datos.append({
                'timestamp': float(match.group('timestamp')),
                'utc_time': int(match.group('utc_time'))
            })

# Crear DataFrame
df = pd.DataFrame(datos)
def utc_time_to_datetime(utc_ns):
    """
    Convierte el campo utc_time del SIB9 (en nanosegundos desde 1900-01-01) a tiempo legible.
    """
    base_date = datetime(1900, 1, 1)
    utc_sec = utc_ns / 1e9  # de nanosegundos a segundos
    return base_date + timedelta(seconds=utc_sec)
# Aplicar la conversión
df['log_datetime'] = pd.to_datetime(df['timestamp'], unit='s')
df['utc_datetime'] = df['utc_time'].apply(utc_time_to_datetime)
df['delta (s)'] = (df['utc_datetime'] - df['log_datetime']).dt.total_seconds()
df['delta_log (s)'] = df['log_datetime'].diff().dt.total_seconds()
df['delta_utc (s)'] = df['utc_datetime'].diff().dt.total_seconds()
df.head()
```

Figura 7.3: Programa de python para el análisis del registro de SIB9 obtenido por el UTC.

Tabla 7.1: Resultados parciales de sincronización temporal obtenidos a partir del SIB9

| timestamp | utc_time | log_datetime | utc_datetime | delta (s) | delta_log (s) | delta_utc (s) |
|---|---|---|---|---|---|---|
| 1.750155e09 | 3959144243749999999 | 20250617 10:17:03.718743563 | 20250617 10:17:23.750 | 20.031256 | | |
| 1.750155e09 | 3959144245029999999 | 20250617 10:17:04.998820066 | 20250617 10:17:25.030 | 20.031180 | 1.280077 | 1.28 |
| 1.750155e09 | 3959144246309999999 | 20250617 10:17:06.278744459 | 20250617 10:17:26.310 | 20.031256 | 1.279924 | 1.28 |
| 1.750155e09 | 3959144247589999999 | 20250617 10:17:07.558815479 | 20250617 10:17:27.590 | 20.031185 | 1.280071 | 1.28 |
| 1.750155e09 | 3959144248869999999 | 20250617 10:17:08.838863850 | 20250617 10:17:28.870 | 20.031136 | 1.280048 | 1.28 |
| 1.750155e09 | 3959144250149999999 | 20250617 10:17:10.119357586 | 20250617 10:17:30.150 | 20.030642 | 1.280494 | 1.28 |
| 1.750155e09 | 3959144251429999999 | 20250617 10:17:11.399347067 | 20250617 10:17:31.430 | 20.030653 | 1.279989 | 1.28 |
| 1.750155e09 | 3959144252709999999 | 20250617 10:17:12.679456949 | 20250617 10:17:32.710 | 20.030543 | 1.28011 | 1.28 |
| 1.750155e09 | 3959144253989999999 | 20250617 10:17:13.959394455 | 20250617 10:17:33.990 | 20.030606 | 1.279938 | 1.28 |
| 1.750155e09 | 3959144255269999999 | 20250617 10:17:15.239108086 | 20250617 10:17:35.270 | 20.030892 | 1.279714 | 1.28 |
| 1.750155e09 | 3959144256549999999 | 20250617 10:17:16.519243479 | 20250617 10:17:36.550 | 20.030757 | 1.280135 | 1.28 |
| 1.750155e09 | 3959144257829999999 | 20250617 10:17:17.799742699 | 20250617 10:17:37.830 | 20.030257 | 1.280499 | 1.28 |
| 1.750155e09 | 3959144259109999999 | 20250617 10:17:19.079805374 | 20250617 10:17:39.110 | 20.030195 | 1.280063 | 1.28 |
| 1.750155e09 | 3959144260389999999 | 20250617 10:17:20.359844685 | 20250617 10:17:40.390 | 20.030155 | 1.280039 | 1.28 |
| 1.750155e09 | 3959144261669999999 | 20250617 10:17:21.639503717 | 20250617 10:17:41.670 | 20.030496 | 1.279659 | 1.28 |
| 1.750155e09 | 3959144262949999999 | 20250617 10:17:22.919568539 | 20250617 10:17:42.950 | 20.030431 | 1.280065 | 1.28 |

En la tabla 7.1 se recogen, para cada iteración del proceso, tanto las marcas temporales internas del sistema (`timestamp` y `log_datetime`) como las referencias horarias absolutas transmitidas por la red a través del campo `timeInfoUTC` del SIB9, representadas como `utc_time` (en formato entero) y `utc_datetime` (en formato legible).

Además, se incluyen tres columnas adicionales que permiten cuantificar la diferencia temporal entre las referencias internas del sistema y la información horaria proporcionada por la red: `delta (s)` representa el desfase total, mientras que `delta_log (s)` y `delta_utc (s)` reflejan las variaciones entre registros consecutivos del sistema y del SIB9, respectivamente.

La primera observación relevante a partir de la tabla es la diferencia temporal entre el tiempo UTC recibido en el mensaje SIB9 y el instante en que se registra la traza en el log del sistema. Esta diferencia, reflejada en la columna `delta (s)`, se mantiene prácticamente constante en torno a los 20 segundos. Este valor representa el desfase entre la hora transmitida por la red y la hora del sistema Linux que registra el mensaje.

Se han analizado distintas posibles causas de este desfase. Candidatos como la latencia introducida por la pila USB, el retardo del sistema operativo o el comportamiento del software de logging han sido descartados, ya que estos factores suelen generar retardos de magnitud mucho menor y con mayor variabilidad. El hecho de que la diferencia sea estable sugiere que el reloj del sistema Linux no estaba sincronizado con una fuente horaria precisa, como un servidor NTP, y presenta una desviación fija respecto a la hora que transmite la red.

También se ha considerado que esta diferencia pueda estar relacionada con la separación entre el tiempo GPS y UTC, actualmente de 18 segundos [23], aunque no se ha podido confirmar si esta compensación se aplica explícitamente en el mensaje recibido o si influye en la forma en la que el módem reporta la información.

En cualquier caso, al tratarse de una diferencia constante y conocida, se ha podido tener en cuenta durante el análisis sin afectar a la validez de los resultados.

En cuanto a las columnas `delta_log (s)` y `delta_utc (s)`, se puede extraer un segundo análisis. Mientras que el valor de `delta_utc` es constante —indicando que el tiempo absoluto transmitido por la red varía de forma regular y predecible—, los valores de `delta_log` presentan pequeñas fluctuaciones inferiores a 1ms, probablemente debido a variaciones menores en el procesamiento interno o en el mecanismo de registro de eventos.

La razón por la cual el valor de `delta_utc` se mantiene constante en 1,28 segundos está directamente relacionada con la estructura temporal del sistema 5GNR. En concreto, el campo `timeInfoUTC` incluido en el SIB9 se actualiza siguiendo el valor del SFN, el cual avanza en pasos de 128 unidades. Dado que cada unidad del SFN representa una trama de 10 ms, un incremento de 128 tramas equivale exactamente a 1,28 segundos. Por

tanto, el SIB9 proporciona una referencia temporal que avanza en saltos discretos de 1,28 segundos, lo que explica la regularidad observada en la columna `delta_utc`.

Una vez que ya tenemos la certeza de obtener correctamente el SIB9, el siguiente paso es el de actualizar el reloj interno del UE con el valor UTC que nos proporciona. Para ello, se ha realizado una modificación al código inicial para que, en cuanto reciba ese valor, actualize el reloj interno. Este código si se puede mostrar al ser propio, y se muestra en la figura 7.4

```c
if (t->utc_time_available) {
    RLOGI("utc_time = %lu", t->utc_time);

    // Convertir y aplicar la hora del sistema desde utc_time
    struct timeval tv;
    struct tm base_tm = { .tm_year = 0, .tm_mon = 0, .tm_mday = 1 };
    time_t base_sec = timegm(&base_tm);  // segundos desde epoch hasta 1900-01-01

    uint64_t total_sec = t->utc_time / 1000000000ULL;
    uint64_t micros = (t->utc_time % 1000000000ULL) / 1000;

    tv.tv_sec = base_sec + total_sec;
    tv.tv_usec = micros;

    if (settimeofday(&tv, NULL) != 0) {
        perror("settimeofday");
    } else {
        RLOGI("Hora del sistema actualizada correctamente con utc_time");
    }
}
```

Figura 7.4: Código en C para actualizar el reloj interno en cuanto se recibe el SIB9.

Antes de poder actualizar el reloj, se ha de pasar el valor UTC a formato hora legible. Tras esto, se hace la actualización del reloj con el comando *settimeofday(&tv, NULL);*

## 7.2. Red móvil de un operador comercial

En esta sección se mostrarán los resultados obtenidos tras el uso del ejecutable cuando el módem está conectado a una red de 5G de un operador comercial.

Se replicó el procedimiento descrito en el capítulo de Preparación del entorno, esta vez utilizando una red 5G proporcionada por un operador comercial. El dispositivo se conectó correctamente a la red en modo *standalone* y el programa desarrollado fue ejecutado bajo las mismas condiciones que en la red privada. Sin embargo, durante toda la duración de la prueba —aproximadamente 30 minutos de ejecución continua— no se recibió en ningún momento el mensaje SIB9.

Este comportamiento está en línea con lo establecido por el estándar 3GPP TS 38.331 [24], el cual define el contenido y la transmisión de los bloques de información del sistema. Según este documento, la emisión del SIB9 es opcional, lo que implica que cada operador puede decidir si incluirlo o no en su red. Todo indica que, en este caso concreto, el operador no está transmitiendo dicho bloque, impidiendo así que el UE pueda obtener referencias temporales absolutas desde la red. Esta limitación resalta la importancia del despliegue específico de cada red en la disponibilidad de esta funcionalidad. Como posible trabajo futuro, se plantea repetir este experimento con redes de otros operadores para comprobar si alguno de ellos ofrece el SIB9 de forma activa.

# Capítulo 8

# Conclusiones

En este capítulo se recogen las ideas más relevantes extraídas a lo largo del desarrollo del proyecto, tanto desde el punto de vista teórico como práctico. Se presentan las principales conclusiones alcanzadas tras el estudio del mensaje SIB9 y su aplicación en la sincronización horaria de un dispositivo, así como las implicaciones que esto puede tener en futuros escenarios donde el tiempo juega un papel clave. Además, se plantean posibles líneas de trabajo que permitirían continuar o ampliar esta investigación, bien sea optimizando el sistema actual o explorando nuevas capacidades ofrecidas por la evolución del estándar 5G.

## 8.1. Conclusiones

A lo largo del presente trabajo se ha profundizado en el estudio teórico y práctico del mensaje SIB9 en redes 5G NR, con el objetivo de analizar su potencial como fuente de sincronización horaria precisa para dispositivos finales sin necesidad de GNSS. Este mensaje, a diferencia de su homónimo en LTE, aporta una referencia absoluta de tiempo UTC/GPS directamente desde la Radio Access Network (RAN), lo cual lo posiciona como una pieza clave en aplicaciones sensibles al tiempo como TSN, Internet of Things (IoT) industrial o automatización vehicular.

Se comenzó con una revisión detallada de la arquitectura de las redes 5G, las señales y canales implicados en la transmisión de información del sistema, y la estructura interna del mensaje SIB9, prestando especial atención a los campos `timeInfoUTC`, `leapSeconds`, `dayLightSavingTime` y `localTimeOffset`. Esta base teórica ha sido fundamental para entender el flujo completo de sincronización y cómo se accede a dicha información desde el punto de vista del UE.

Posteriormente, se abordó el estudio del módem *Telit FN990*, tanto desde el análisis de sus comandos AT como del entorno de hardware y software necesario para interactuar con él. Se ha trabajado con el entorno de desarrollo

Telit EVB 2.0, se configuraron adecuadamente los puertos de comunicación y se identificaron los comandos clave para habilitar la recepción de mensajes de sistema.

En la parte práctica del trabajo se logró con éxito la obtención del mensaje SIB9 en condiciones reales. A través del uso de comandos AT y herramientas de análisis proporcionadas por el fabricante, se extrajo el contenido completo del mensaje y se validó su correspondencia con la hora UTC enviada por la estación base.

Uno de los principales logros de este TFM ha sido utilizar esa información horaria contenida en el SIB9 para sincronizar el reloj interno del sistema *host*, alcanzando una precisión muy superior a la obtenida únicamente con los mecanismos tradicionales de sincronización por capas físicas. Esta sincronización se realizó transformando la información del SIB9 a formato legible, corrigiendo desfases horarios y comparando la estabilidad del reloj local respecto a esta nueva referencia.

En definitiva, se ha demostrado que es posible prescindir de fuentes GNSS para obtener una sincronización de alta precisión usando únicamente la infraestructura 5G y el contenido del SIB9, lo cual abre la puerta a nuevos escenarios de aplicación en redes privadas, entornos industriales o dispositivos con restricciones energéticas y de coste.

## 8.2. Trabajos futuros

A partir del desarrollo realizado y de los resultados obtenidos, surgen varias líneas de trabajo que podrían abordarse en el futuro para continuar explorando y perfeccionando el uso del mensaje SIB9 como fuente de sincronización temporal precisa:

- Evaluación comparativa de la precisión temporal obtenida mediante SIB9 frente a soluciones basadas en GNSS, en diferentes condiciones de red y entornos.
- Captura y análisis del mensaje SIB9 en redes 5G comerciales de operadores y con diferentes proveedores de estaciones base, verificando la consistencia y estructura de los campos temporales transmitidos.
- Integración del sistema en el banco de pruebas del proyecto 6G-CHRONOS, para su uso en aplicaciones sensibles al tiempo dentro de redes industriales o escenarios de sincronización distribuida.

# Bibliografía


[1] Ángel Gómez Hurtado. *Sincronización en redes 5G: dispositivo final. Captura y decodificación de señales de sincronización en equipos reales*. Trabajo Fin de Grado, Universidad de Granada. 2023. URL: https://wpd.ugr.es/~jorgenavarro/thesis/2023_TFG_AngelGomezHurtado.pdf (visitado 01-07-2025).

[2] WiMuNet. *WiMuNet*. URL: https://wimunet.ugr.es/.

[3] István Gódor et al. "A Look Inside 5G Standards to Support Time Synchronization for Smart Manufacturing". En: *IEEE Communications Standards Magazine* 4.3 (sep. de 2020), págs. 64-72. DOI: 10.1109/MCOMSTD.001.2000010. URL: https://ieeexplore.ieee.org/document/9171323 (visitado 01-07-2025).

[4] D. Patel et al. "Time Error Analysis of 5G Time Synchronization Solutions for Time Aware Industrial Networks". En: *2021 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*. 2021. DOI: 10.1109/ISPCS49990.2021.9615318.

[5] Junaid Ansari et al. "Prototype of 5G Integrated with TSN for Edge-Controlled Mobile Robotics". En: *Electronics* 11.11 (2022), pág. 1666. DOI: 10.3390/electronics11111666.

[6] Thomas Kernen et al. "Supporting Live Production Using Precision Time Protocol Over 5G Systems With Dedicated Time Synchronization Support". En: *Proceedings of the NAB Broadcast Engineering and IT Conference*. 5G-RECORDS project, IEEE BTS/NAB Proceedings. 2023.

[7] D. Chandramouli, P. Andrés-Maldonado y T. Kolding. "Evolution of Timing Services from 5G-A towards 6G". En: *IEEE Access* 11 (2023), págs. 40492-40508. DOI: 10.1109/ACCESS.2023.3265213.

[8] Nokia. *GNSS-free time sync in private 5G with Telit FN990*. Consultado en junio de 2025. 2023. URL: https://www.linkedin.com/posts/nokia_enterprise-private5g-5g-activity-7078477229202679808-QE6Y.

[9] Aymen Omri et al. "Synchronization Procedure in 5G NR Systems". En: *IEEE Access* 7 (2019), págs. 41286-41295. DOI: 10.1109/ACCESS.2019.2907970.

[10] 3GPP. *NR Physical layer procedures for control (Release17)*. Inf. téc. 3GPP TS 38.213 version 17.6.0. Especificación técnica. 3GPP, jun. de 2024. URL: https://www.3gpp.org/ftp/Specs/archive/38_series/38.213/38213-g60.zip (visitado 01-07-2025).

[11] electronicsnotes. *5G Data Channels: Physical; Transport; Logical*. URL: https://www.electronics-notes.com/articles/connectivity/5g-mobile-wireless-cellular/data-channels-physical-transport-logical.php.

[12] 3GPP. *NR Physical channels and modulation (Release17)*. Inf. téc. 3GPP TS 38.211 version 17.4.0. Especificación técnica. 3GPP, ene. de 2023. URL: https://www.etsi.org/deliver/etsi_ts/138200_138299/138211/17.04.00_60/ts_138211v17400p.pdf (visitado 01-07-2025).

[13] 3GPP. *NR User Equipment (UE) radio transmission and reception; Part 1: Range 1 Standalone (Release17)*. Inf. téc. 3GPP TS 38.101-1 version 17.13.0. Especificación técnica. 3GPP, jun. de 2024. URL: https://www.3gpp.org/ftp/Specs/archive/38_series/38.101-1/38101-1-m30.zip (visitado 01-07-2025).

[14] 3GPP. *NR Multiplexing and channel coding (Release17)*. Inf. téc. 3GPP TS 38.212 version 17.6.0. Especificación técnica. 3GPP, jun. de 2024. URL: https://www.3gpp.org/ftp/Specs/archive/38_series/38.212/38212-g60.zip (visitado 01-07-2025).

[15] 3GPP. *NR Radio Resource Control (RRC) protocol specification (Release17)*. Inf. téc. 3GPP TS 38.331 version 17.10.0. Especificación técnica. 3GPP, jun. de 2024. URL: https://www.3gpp.org/ftp/Specs/archive/38_series/38.331/38331-g00.zip (visitado 01-07-2025).

[16] Seokjoon Hong et al. "5G NR based initial access procedure simulation environment implementation including system information using ns-3 simulator". En: *2023 25th International Conference on Advanced Communication Technology (ICACT)*. 2023, págs. 171-173. DOI: 10.23919/ICACT56868.2023.10079674.

[17] 3GPP. *NR Radio Resource Control (RRC) protocol specification (Release16)*. Inf. téc. 3GPP, mar. de 2023.

[18] Mridula Korde. "Synchronization Aspects in 5G". En: *2020 International Conference on Communication and Signal Processing (ICCSP)*. 2020, págs. 0474-0478. DOI: 10.1109/ICCSP48568.2020.9182120.

[19] Telit Cinterion. *Telit EVB 2.0 User Guide*. Rev. 3, 2022. URL: https://www.telit.com.

[20] Canonical Ltd. *Ubuntu Documentation*. Consultado en 2025. URL: https://ubuntu.com.

[21] Amarisoft. *Amarisoft*. URL: https://www.amarisoft.com/.

[22] Amarisoft. *Amarisoft*. URL: https://www.amarisoft.com/app/uploads/2022/03/userguide_callbox_classic.pdf.

[23] US Naval Observatory. *Leap Seconds and GPS–UTC Offset*. https://www.cnmoc.usff.navy.mil/Our-Commands/United-States-Naval-Observatory/Precise-Time-Department/Global-Positioning-System/USNO-GPS-Time-Transfer/Leap-Seconds/. Último acceso: 1 de julio de 2025.

[24] 3GPP. *TS 38.331 - NR; Radio Resource Control (RRC); Protocol specification*. Inf. téc. TS 38.331. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/38_series/38.331/. 3rd Generation Partnership Project (3GPP), 2024.