PLANO Y ESQUEMÁTICO DEL CIRCUITO — MÓDULO 06: NODOS INALÁMBRICOS Y PLACAS PCB SOLDADAS

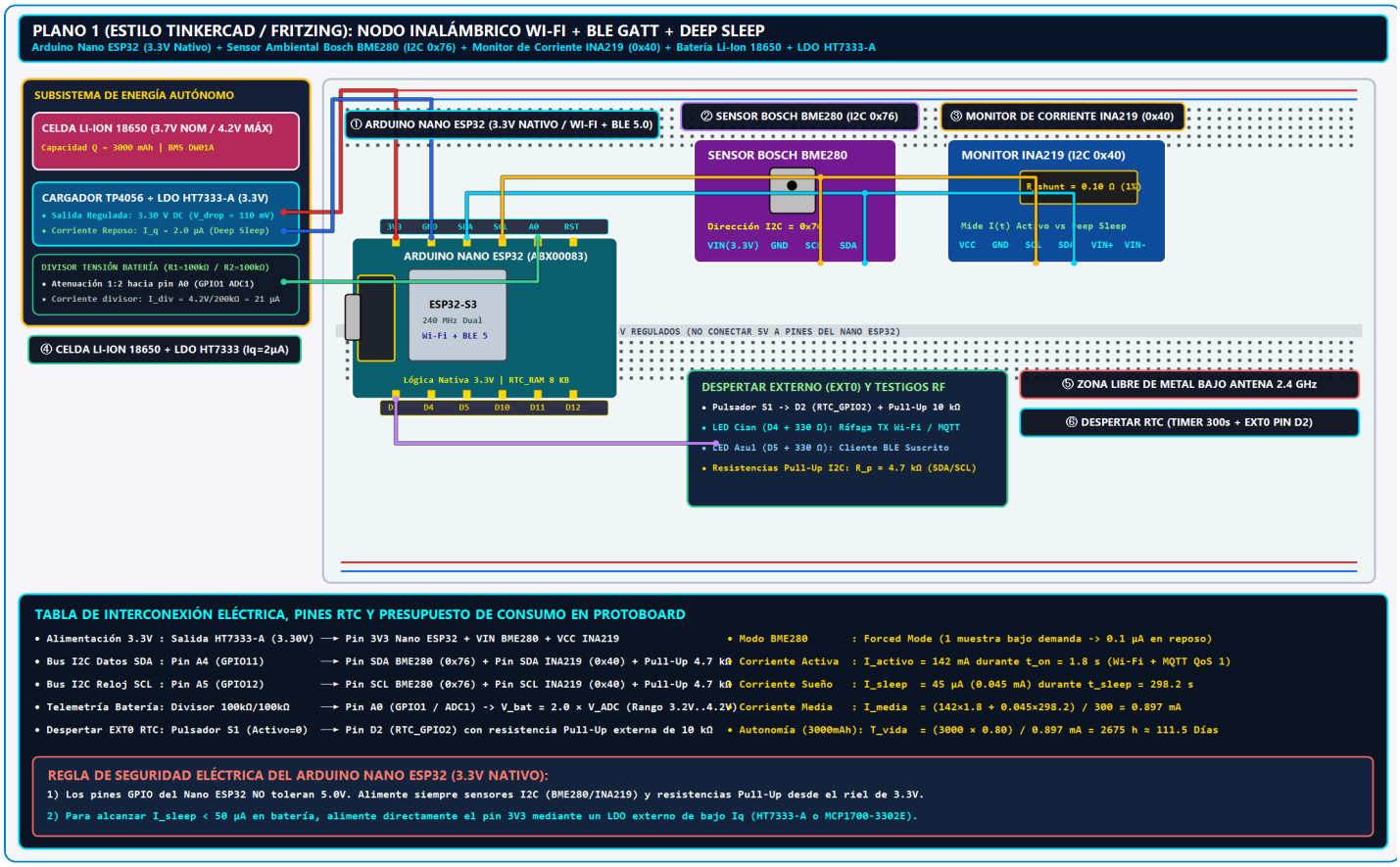
Curso: Internet de las Cosas (IoT) con Arduino y Raspberry Pi — Ingeniería Mecatrobótica

Módulo 06 / Video 06: Planos de Montaje en Protoboard (Estilo Tinkercad / Fritzing) de Alta Legibilidad y Planos de Circuito Impreso (`PCB FR-4`) con Montaje por Soldadura

---

1. Plano 1 (Estilo Tinkercad / Fritzing): Nodo Inalámbrico Wi-Fi + BLE + Deep Sleep en Protoboard

📐 Plano 1: Montaje en Protoboard Estilo Tinkercad del Nodo Wi-Fi y BLE con Sensor BME280, Monitor INA219 y Batería Li-Ion 18650

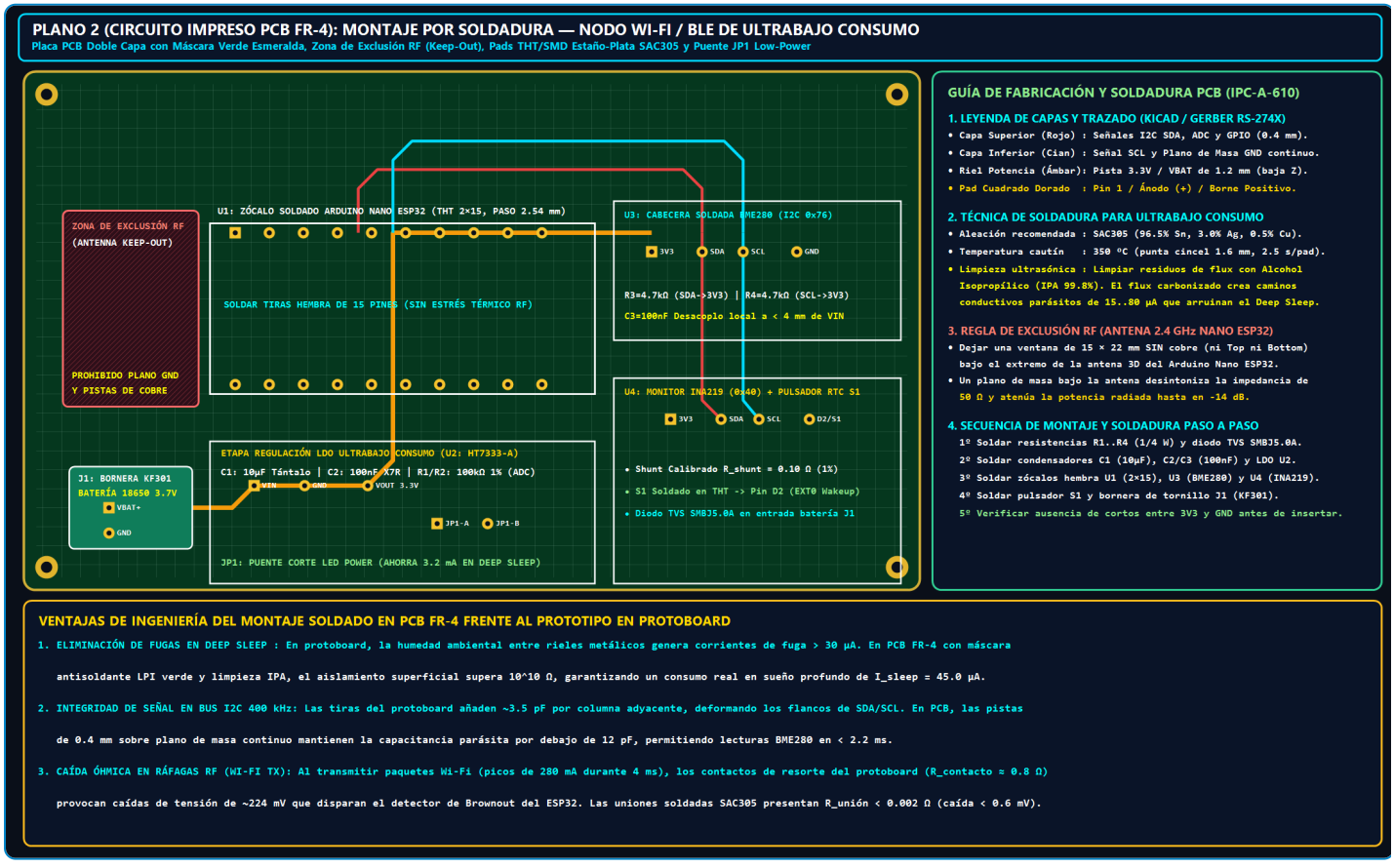


1.1 Resumen de Asignación de Pines del Arduino Nano ESP32 (`3.3 V` Nativo)

Callout en Plano	Subsistema / Módulo	Pines en Protoboard	Parámetros Eléctricos y de Firmware
①	**Arduino Nano ESP32 (** `ABX00083` **)**	Riel `3V3` y `GND`	SoC `ESP32-S3` con $8\ KB$ de memoria RTC (`RTC_DATA_ATTR`)
②	**Sensor Ambiental Bosch** `BME280`	`SDA -> A4` / `SCL -> A5`	Dirección I^2C `0x76` , operado en `MODE_FORCED` ($0.1\ \mu A$ en reposo)
③	**Monitor de Corriente** `INA219`	`SDA -> A4` / `SCL -> A5`	Dirección I^2C `0x40` , resistencia Shunt $R_{shunt} = 0.10\ \Omega\ (1\%)$
④	**Batería** `18650` **+ LDO** `HT7333-A`	Salida `3.30 V` → `Pin 3V3`	Corriente de reposo $I_q = 2.0\ \mu A$ y divisor $100\ k\Omega/100\ k\Omega$ hacia `A0`
⑤	**Antena Cerámica / 3D** `2.4 GHz`	Extremo izquierdo del módulo	Libre de cables puente y planos metálicos a $15\ mm$ a la redonda
⑥	**Despertar RTC (** `Timer` **+** `EXT0` **)**	Pulsador `S1` → `Pin D2`	`esp_sleep_enable_timer_wakeup(300e6)` + `esp_sleep_enable_ext0_wakeup(GPIO_NUM_2, 0)`

2. Plano 2 (Circuito Impreso `PCB FR-4`): Montaje por Soldadura del Nodo de Ultrabajo Consumo

📐 Plano 2: Placa de Circuito Impreso PCB FR-4 Doble Capa con Montaje por Soldadura del Nodo Inalámbrico

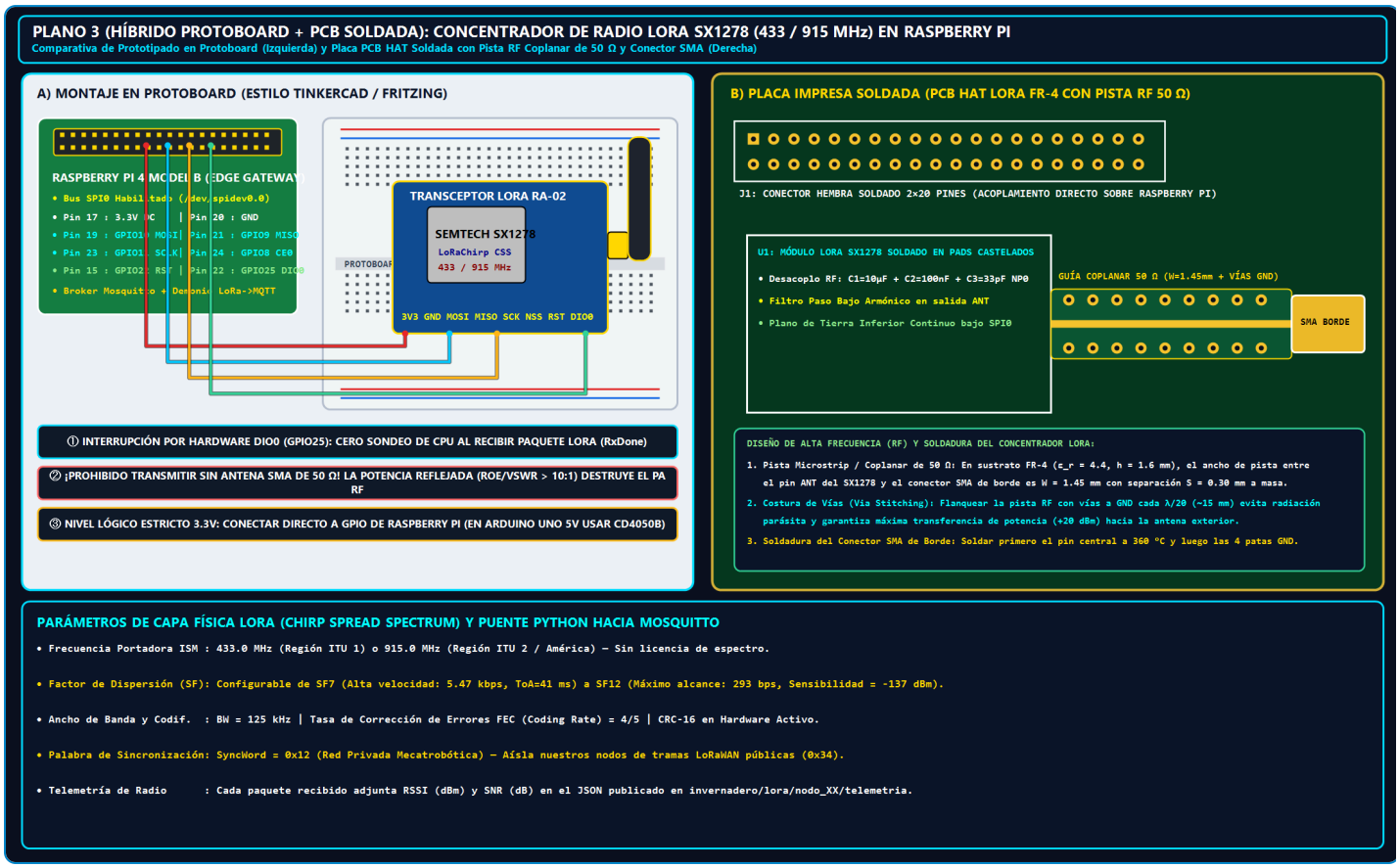


2.1 Especificaciones de Fabricación PCB (`KiCad` / `IPC-2221`) y Soldadura (`IPC-A-610`)

Elemento en la Placa PCB	Especificación Geométrica y Metalúrgica	Razón de Ingeniería (RF y Ultrabajo Consumo)
Zona de Exclusión RF (`Keep-Out`)	Ventana de 15×22 mm **sin cobre** en `Top` ni `Bottom` bajo la antena del Nano ESP32	Evita desintonizar la antena de 2.4 GHz (un plano GND debajo atenúa hasta -14 dB)
Puente Soldable `JP1` (Corte LED)	Dos pads dorados de 2.0×2.0 mm en serie con el LED indicador de encendido	Abrir `JP1` en despliegue final elimina el consumo fijo de 3.2 mA del LED Power
Limpieza de Flux con `IPA 99.8%`	Lavado con cepillo antiestático tras soldar con aleación `SAC305` a $350\ ^\circ$C	Los residuos de flux forman resistencias parásitas superficiales que drenan $15..80\ \mu$A
Zócalos Hembra `THT` (`U1, U3, U4`)	Agujeros metalizados $\varnothing 1.0$ mm con anillo anular dorado $\varnothing 1.8$ mm	Permite sustituir el Nano ESP32 o el `BME280` en campo en 5 segundos sin desoldar

3. Plano 3 (Híbrido Protoboard + `PCB HAT Soldada`): Concentrador `LoRa SX1278` en Raspberry Pi

📐 Plano 3: Montaje en Protoboard y Placa PCB HAT Soldada con Guía de Onda Coplanar de 50 Ohmios del Concentrador LoRa SX1278



4. Plano 4: Arquitectura de Perfil BLE GATT (`0x181A`) y Máquina de Estados con Backoff Exponencial

📐 Plano 4: Jerarquía del Servidor BLE GATT 0x181A y Máquina de Estados de Reconexión Wi-Fi/MQTT

PLANO 4: ARQUITECTURA DE PERFIL BLE GATT (SERVICIO 0x181A) Y MÁQUINA DE ESTADOS WI-FI / MQTT CON BACKOFF EXPONENCIAL

Jerarquía Servidor Periférico GATT (Arduino Nano ESP32) ↔ Cliente Central Asíncrono Bleak (Raspberry Pi) ↔ Broker Eclipse Mosquitto

A) ARQUITECTURA BLUETOOTH LOW ENERGY (BLE 5.0): SERVIDOR GATT 0x181A ↔ PASARELA ASÍNCRONA BLEAK (PYTHON)

1. PERIFÉRICO / SERVIDOR GATT (NANO ESP32)

- Capa GAP (Anuncio): Nombre 'NODO_BLE_INV_01'
  Intervalo Advertising = 100 ms (Canales 37, 38 y 39)

SERVICIO AMBIENTAL ESTÁNDAR SIG: UUID = 0x181A

- ├ Característica Temperatura : UUID 0x2A6E (sint16, 0.01 °C)
  | Propiedades: READ | NOTIFY + Descriptor CCCD 0x2902
- ├ Característica Humedad Rel.: UUID 0x2A6F (uint16, 0.01 %RH)
  | Propiedades: READ | NOTIFY + Descriptor CCCD 0x2902
- ├ Característica Presión Atm.: UUID 0x2A6D (uint32, 0.1 Pa)
  | Propiedades: READ | NOTIFY + Descriptor CCCD 0x2902
- └ Empaquetado Little-Endian: struct.pack('<hHI', T, H, P)
  (Solo 8 bytes por notificación -> Ahorro de 92% de aire RF)

2. CLIENTE CENTRAL ASÍNCRONO EN RASPBERRY PI

- Pila Linux BlueZ (/dev/hci0) + Librería Python 'bleak'
- Bucle de eventos no bloqueante: asyncio.run(main())

SECUENCIA DE SUSCRIPCIÓN A NOTIFICACIONES (PUSH)

1º Escaneo pasivo: BleakScanner.find_device_by_name()

2º Conexión GATT : async with BleakClient(addr) as cli:

3º Activación Push: await cli.start_notify(UUID, callback)
(Escribe 0x0001 en el descriptor CCCD 0x2902 del ESP32)

4º Recepción por Evento: El ESP32 envía paquete ATT_HANDLE VALUE_NOTIFICATION solo cuando cambia la medición.

5º Desempaquetado: T_c = int.from_bytes(d[0:2],'little')/100

3. CONVERGENCIA EN BROKER MOSQUITTO

- Traducción transparente BLE Binario -> MQTT JSON
- Tópico: invernadero/ble/nodo_01/ambiental

PAYLOAD JSON NORMALIZADO EN EL BROKER:

```
{
  "timestamp": "2026-11-01T12:00:05Z",
  "enlace_rf": "BLE_5.0_GATT_0x181A",
  "temperatura_c": 23.45,
  "humedad_rh": 64.20,
  "presion_hpa": 1013.25,
  "bateria_v": 3.98
}
```

- QoS = 1 | Retain = False | LWT Activo

B) MÁQUINA DE ESTADOS FINITOS (FSM) DE RECONEXIÓN WI-FI / MQTT CON BACKOFF EXPONENCIAL Y JITTER ALEATORIO

ESTADO 1: CONECTANDO_WIFI

- WiFi.mode(WIFI_STA)
- WiFi.begin(SSID, PASS)
- Timeout estricto = 8.0 s
- Si falla -> Incrementar k

ESTADO 2: CONECTANDO_MQTT

- Validar IP por DHCP
- mqtt.setWill(LWT, QoS 1)
- mqtt.connect(ID, Usr, Pwd)
- Suscribir tópico comando

ESTADO 3: OPERATIVO (TX/RX)

- Leer BME280 (Forced Mode)
- Leer V_bat + INA219
- Publicar JSON (QoS 1)
- Reiniciar intentos k = 0

ESTADO 4: BACKOFF + DEEP SLEEP

- T_espera = min(Tmax, T0·2^k)+J
- Si k >= 5 caídas seguidas:
- Entrar en Deep Sleep RTC
- Proteger celda Li-Ion 18650

LEY MATEMÁTICA DE RETROCESO EXPONENCIAL CON FLUCTUACIÓN ESTOCÁSTICA (JITTER):

$T_{espera}(k) = \min(T_{max} , T_0 \times 2^k) + U(0, J_{max})$ donde $T_0 = 2.0$ s , $T_{max} = 60.0$ s , $J_{max} = 1.0$ s

- Progresión de espera ante caída del Router/Broker: k=0 -> 2s | k=1 -> 4s | k=2 -> 8s | k=3 -> 16s | k=4 -> 32s | k>=5 -> 60s (Tope).
- ¿Por qué sumar Jitter aleatorio U(0, 1000 ms)? Tras un corte eléctrico en planta, evita que 50 nodos inalámbricos envíen su solicitud TCP SYN / MQTT CONNECT exactamente en el mismo milisegundo (Tormenta de Reconexión o Thundering Herd).
- Protección de Batería: Si tras k = 5 intentos el punto de acceso Wi-Fi sigue apagado, el nodo apaga la radio y duerme 300 s.

---

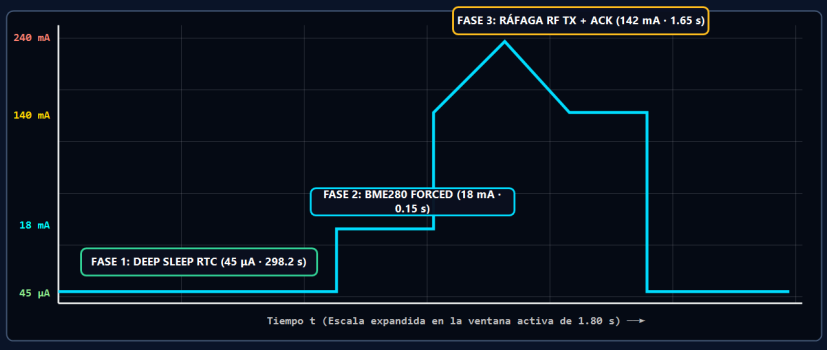
5. Plano 5: Perfil de Corriente $I(t)$ en `Deep Sleep` y Presupuesto de Enlace RF (`SF7..SF12`)

📐 Plano 5: Perfil Temporal de Corriente medido con INA219 y Presupuesto de Enlace RF

PLANO 5: PERFIL DE CORRIENTE I(t) EN SUEÑO PROFUNDO (DEEP SLEEP) Y PRESUPUESTO DE ENLACE RF (LINK BUDGET)

Caracterización Energética medida con INA219 (Ciclo T = 300 s) y Curvas de Sensibilidad LoRa (SF7..SF12) frente a Wi-Fi y BLE

A) PERFIL TEMPORAL DE CORRIENTE I(t) EN UN CICLO DE REPORTE (T_ciclo = 300 s = 5.0 MINUTOS)



BALANCE DE CARGA ELÉCTRICA Y AUTONOMÍA DE BATERÍA:

1. Corriente Media Ponderada en el Ciclo (T = 300 s):
   $I_{media} = (I_{on} \cdot t_{on} + I_{sleep} \cdot t_{sleep}) / T_{ciclo}$

- Caso A: Nodo Wi-Fi + MQTT (t_{on} = 1.80 s, I_{on} = 142 mA):
  $I_{media(Wi-Fi)} = (142 \times 1.80 + 0.045 \times 298.2) / 300 = 0.897$ mA
  $T_{vida(Wi-Fi)} = (3000$ mAh × 0.80) / 0.897 mA = 111.5 Días
- Caso B: Nodo LoRa SX1278 SF7 o BLE GATT (t_{on} = 0.25 s, I_{on} = 95 mA):
  $I_{media(LoRa)} = (95 \times 0.25 + 0.045 \times 299.75) / 300 = 0.124$ mA
  $T_{vida(LoRa)} = (3000$ mAh × 0.80) / 0.124 mA = 19,354 h ≈ 2.2 Años
- Caso C: Sin Deep Sleep (Radio Wi-Fi encendida 24/7 a 125 mA):
  $T_{vida(Continuo)}$ = 2400 mAh / 125 mA = 19.2 Horas (¡Inviable!)

B) PRESUPUESTO DE ENLACE DE RADIOFRECUENCIA (RF LINK BUDGET) Y FACTORES DE DISPERSIÓN LORA (SF7 A SF12)

ECUACIONES DE PROPAGACIÓN Y MARGEN DE DESVANECIMIENTO:

1. Potencia Recibida (Ecuación de Enlace en dBm):
   $P_{rx(dBm)} = P_{tx(dBm)} + G_{tx(dBi)} + G_{rx(dBi)} - FSPL(dB) - L_{misc(dB)}$

2. Pérdida en Espacio Libre de Friis (FSPL):
   $FSPL(dB) = 20 \cdot \log10(d_{km}) + 20 \cdot \log10(f_{MHz}) + 32.44$

3. Ventaja Espectral de Sub-GHz (433 MHz vs 2400 MHz a igual distancia):
   $\Delta FSPL = 20 \cdot \log10(2400 / 433) = +14.87$ dB menos atenuación en 433 MHz

4. Margen de Enlace (Link Margin / Fade Margin):
   $M_{enlace(dB)} = P_{rx(dBm)} - S_{min(dBm)} >= 15.0$ dB (Criterio Industrial)

- Ejemplo LoRa SF10 a d = 3.0 km (f=915 MHz, P_{tx}=+17 dBm, G=2 dBi):
  FSPL = 9.54 + 59.22 + 32.44 = 101.2 dB | P_{rx} = 17+4-101.2-12 = -92.2 dBm
  Como $S_{min(SF10)}$ = -132 dBm -> Margen M = -92.2 - (-132) = +39.8 dB (¡Sólido!)

MATRIZ COMPARATIVA DE TECNOLOGÍAS Y FACTORES LORA (BW = 125 kHz, Payload = 24 B):

TECNOLOGÍA / MODO	BANDA RF	VELOCIDAD	SENSIBILIDAD	TIEMPO AIRE	ALCANCE CAMPO
Wi-Fi 802.11n	2.4 GHz	72 Mbps	-75 dBm	1.5 ms	30 a 90 m
BLE 5.0 (GATT 1M)	2.4 GHz	1 Mbps	-95 dBm	0.4 ms	15 a 60 m
LoRa SF7 (2^7)	433/915M	5469 bps	-123 dBm	56.6 ms	1.5 a 2.5 km
LoRa SF8 (2^8)	433/915M	3125 bps	-126 dBm	102.9 ms	2.5 a 4.0 km
LoRa SF9 (2^9)	433/915M	1758 bps	-129 dBm	185.3 ms	4.0 a 6.0 km
LoRa SF10 (2^10)	433/915M	977 bps	-132 dBm	370.7 ms	6.0 a 8.5 km
LoRa SF11 (2^11)	433/915M	537 bps	-134.5 dBm	741.4 ms	8.5 a 11 km
LoRa SF12 (2^12)	433/915M	293 bps	-137 dBm	1482.8 ms	11.0 a 15 km