if (LMIC.dataLen > 0) {  // Si se recibe algo por el downlink

    uint8\_t downlinkPayload = LMIC.frame[LMIC.dataBeg];

    LMIC.dataLen=0; //sino no se almacena bien el dato de LMIC del while de abajo

    Serial.print("🔻 Downlink recibido: ");

    Serial.println(downlinkPayload, HEX);

    // Si se recibe 0x02, el LED comienza a parpadear

    if (downlinkPayload == 0x02) {

      // Enviar uplink solicitando respuesta

        Serial.println("📤 Enviando uplink...");

        static uint8\_t payload[1] = { 0x04 };  // Payload con el mensaje que quieres enviar

        Serial.println("💡 Parpadeando LED...");

        for (int i = 0; i < 10; i++) {  // Parpadeo 10 veces

            digitalWrite(LED\_PINRx, HIGH);

            delay(500);

            digitalWrite(LED\_PINRx, LOW);

            delay(500);

        }

        return;  // Detiene la ejecución después de parpadear

    }

    // Si se recibe 0x03, se inicia el bucle de reintento

    else if (downlinkPayload == 0x03) {

        Serial.println("🔁 Iniciando bucle de espera hasta recibir 0x02...");

        downlinkPayload=0; //la reinicio para luego poder encender el led con el if(payload)

        while (true) {

            // Enviar uplink solicitando respuesta

            Serial.println("📤 Enviando uplink...");

            static uint8\_t payload[1] = { 0x04 };  // Payload con el mensaje que quieres enviar

            LMIC\_setTxData2(1, payload, sizeof(payload), 0);  // Enviar el uplink a TTN en el puerto 1

            // Esperamos respuesta (delay ajustable)

            delay(10000);  // Espera de 10 segundos entre intentos

            // Verificamos si llegó un nuevo downlink

            if (LMIC.dataLen > 0) {

                downlinkPayload = LMIC.frame[LMIC.dataBeg];

                Serial.print("🔻 Downlink recibido en bucle: ");

                Serial.println(downlinkPayload, HEX);

                    Serial.println("💡 Recibido, parpadeando LED y saliendo del bucle.");

                    for (int i = 0; i < 10; i++) {

                        digitalWrite(LED\_PINRx, HIGH);

                        delay(500);

                        digitalWrite(LED\_PINRx, LOW);

                        delay(500);

                    }

            }

                break;  // Salir del while

        }

        return;

    }