GUÍA PASO A PASO DE LABORATORIO — MÓDULO 08

Control en Lazo Cerrado (`PID` Discreto), Llamadas a Procedimiento Remoto (`RPC` sobre `MQTT`) e Inteligencia Artificial en el Borde (`Edge AI`)

Curso: Ingeniería en Internet de las Cosas (`IoT`) con Arduino y Raspberry Pi

Módulo: 08 — Regulación `PID` en Tiempo Discreto, Protocolo `RPC` Correlacionado, `Isolation Forest` y Visión por Eventos

Duración estimada de laboratorio: 4 horas prácticas

---

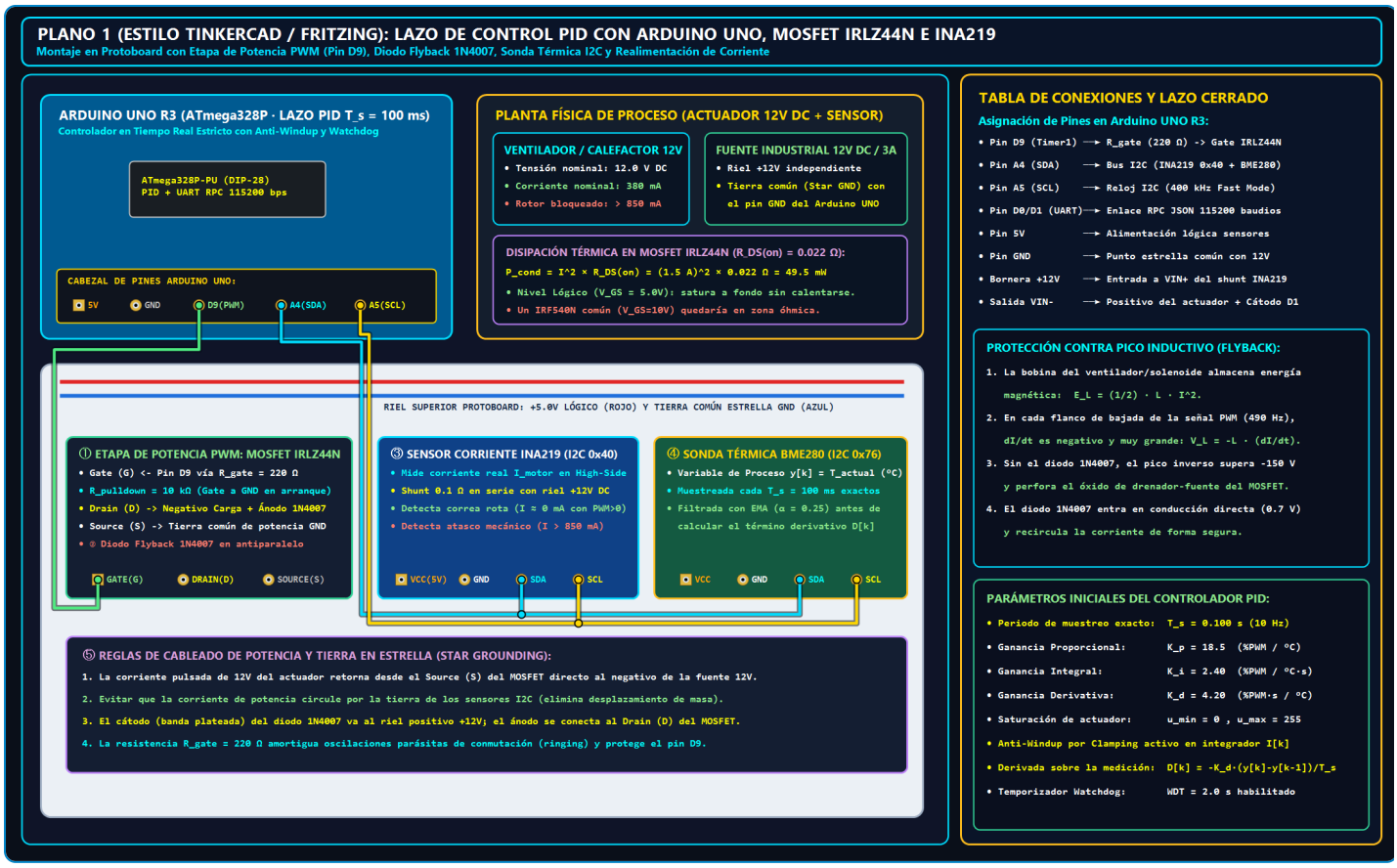
1. Objetivos de Ingeniería del Laboratorio

- **Separar la jerarquía de control en tiempo real estricto (`Hard Real-Time`) de la supervisión analítica en el borde (`Soft Real-Time`):** ejecutar el lazo cerrado proporcional-integral-derivativo (`PID`) cada $T_s = 100\ ms$ en el microcontrolador **Arduino UNO R3 (`ATmega328P`)**, delegando el ajuste de consigna (`Setpoint`), la inferencia de anomalías y la visualización a la **Raspberry Pi 4 / 5**.
- **Construir la etapa de potencia en modulación por ancho de pulso (`PWM`) con transistor `MOSFET` de nivel lógico (`IRLZ44N`):** calcular la disipación térmica por conducción, instalar la resistencia limitadora de compuerta ($220\ \Omega$), la resistencia de anclaje a tierra ($10\ k\Omega$) y el diodo volante de rueda libre (`1N4007`) en antiparalelo con la carga inductiva de $12\ V\ DC$.
- **Implementar ambos planos de construcción electrónica:** el prototipo experimental en **protoboard (estilo Tinkercad / Fritzing)** con tipografía ampliada de alta legibilidad y la **placa de circuito impreso (`PCB FR-4` de doble capa) con montaje por soldadura** y pistas de potencia de $2.50\ mm$.
- **Programar un protocolo de Llamada a Procedimiento Remoto (`RPC`) correlacionado sobre `MQTT` (`QoS 1`):** emparejar solicitudes y respuestas mediante un identificador único universal (`req_id`), sincronizar hilos con `threading.Event()` , establecer un temporizador de expiración de $1500\ ms$ y aplicar reversión segura (`Rollback`) en la interfaz gráfica si el hardware no confirma la orden.
- **Desplegar Inteligencia Artificial en el Borde (`Edge AI`) y captura visual dirigida por eventos:** ejecutar detección estadística por puntaje Z móvil (`Rolling Z-Score` 3σ) e inferencia multivariada con **Bosque de Aislamiento (`Isolation Forest`)** sobre el vector $\mathbf{x} = [T_{actual}, PWM, I_{motor}]^T$, disparando la cámara `CSI` (`Picamera2` + `OpenCV`) en el milisegundo exacto de una anomalía.

---

2. Arquitectura Física del Lazo Cerrado y Etapa de Potencia (`Plano 1`)

📐 Plano 1: Montaje en Protoboard Estilo Tinkercad del Lazo de Control PID con MOSFET IRLZ44N e INA219



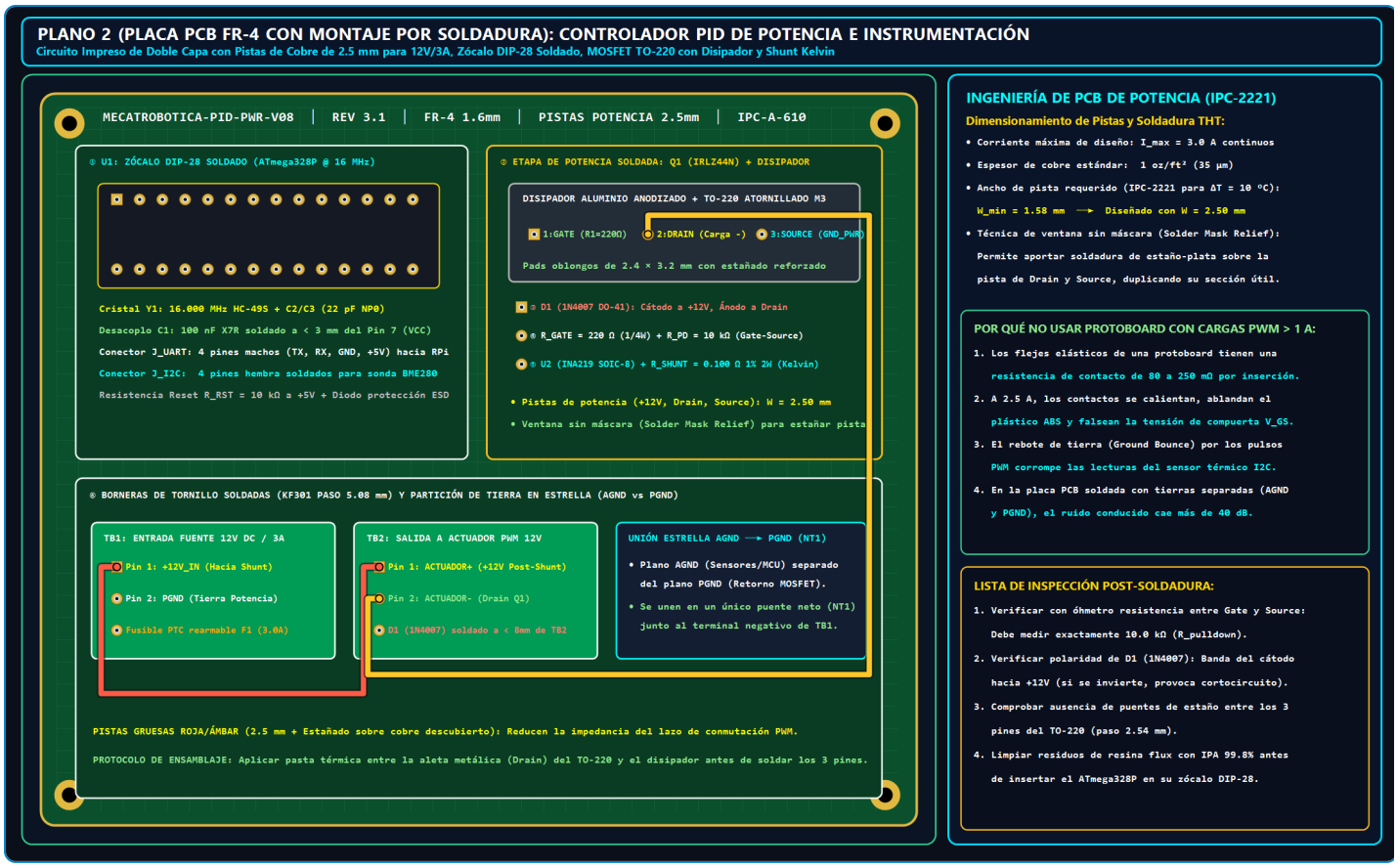
Paso 2.1 — Reglas Críticas de Cableado de Potencia y Tierra en Estrella (`Star Grounding`)

- **Por qué es obligatorio un `MOSFET` de nivel lógico (`IRLZ44N`):** el pin `D9` del Arduino UNO entrega $5.0\ \text{V}$ (y las placas de $3.3\ \text{V}$ entregan aún menos). Un transistor `MOSFET` estándar como el `IRF540N` requiere $V_{GS} = 10\ \text{V}$ para saturar completamente; si se excita con $5\ \text{V}$, permanece en su región lineal/óhmica y se destruye por sobrecalentamiento. El `IRLZ44N` posee tensión umbral $V_{GS(th)} \in [1.0, 2.0]\ \text{V}$ y alcanza $R_{DS(on)} = 0.022\ \Omega$ con $V_{GS} = 5.0\ \text{V}$.
- **Diodo de rueda libre (`Flyback 1N4007`):** conéctelo directamente entre los terminales del actuador inductivo (ventilador o válvula solenoide de $12\ \text{V}$), con el **cátodo (banda plateada)** hacia el riel positivo de $+12\ \text{V}$ y el **ánodo** hacia el drenador (`Drain`) del `IRLZ44N`.
- **Tierra en estrella (`Star Ground`):** el surtidor (`Source`) del `IRLZ44N` debe retornar directamente por un cable grueso al borne negativo de la fuente de $12\ \text{V DC}$, evitando que los pulsos de corriente de conmutación `PWM` circulen por la pista de tierra de los sensores `I2C` (`BME280` e `INA219`).

---

3. Montaje Definitivo en Placa de Circuito Impreso Soldada (`Plano 2`)

📐 Plano 2: Placa PCB FR-4 de Doble Capa con Montaje por Soldadura para Controlador PID de Potencia



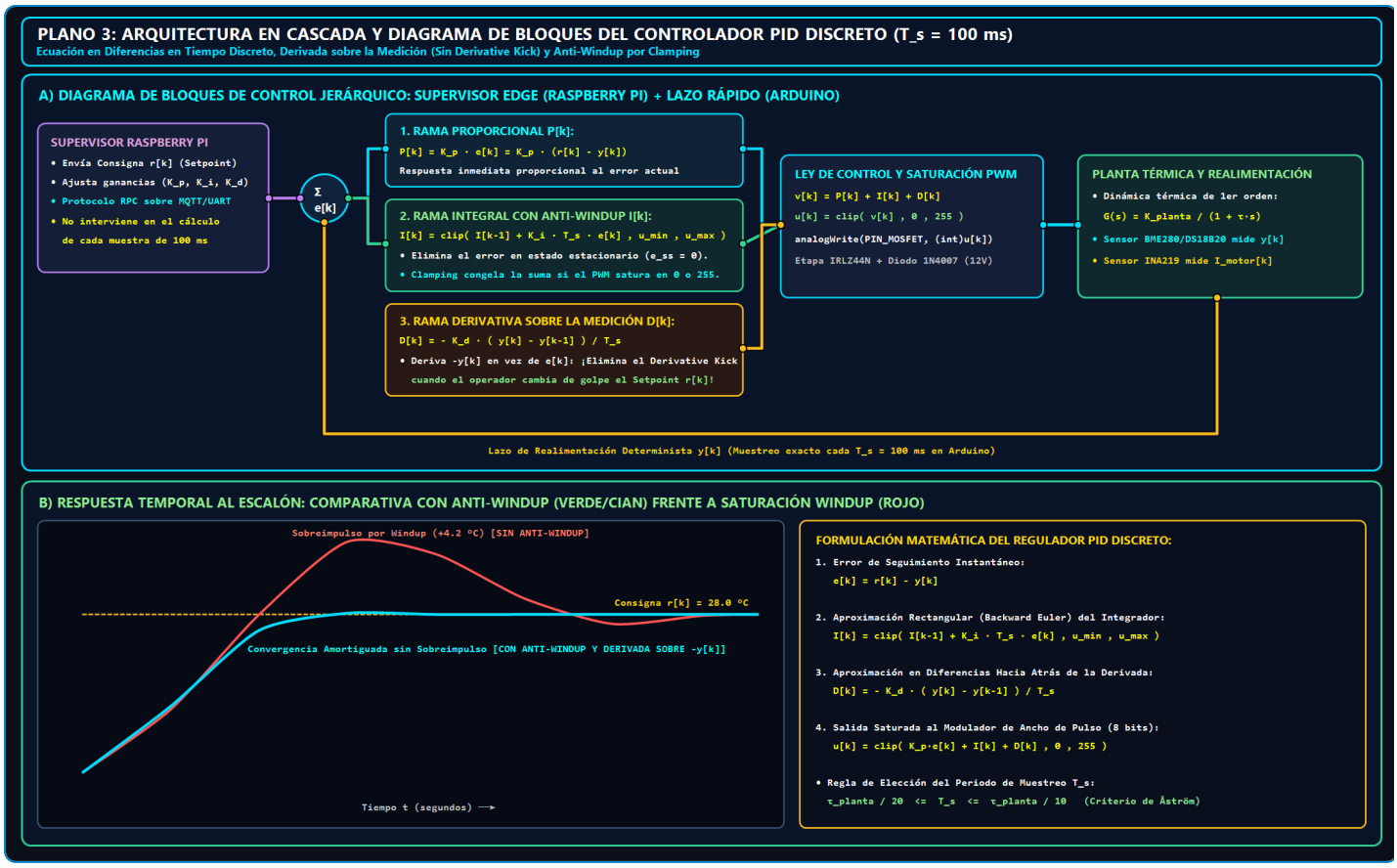
Paso 3.1 — Procedimiento de Soldadura de Potencia según `IPC-2221` e `IPC-A-610`

- Suelde primero los componentes de bajo perfil: resistencias de película metálica ($220\ \Omega$, $10\ k\Omega$), condensadores cerámicos `X7R` de $100\ nF$, el cristal de $16.000\ MHz$ y el sensor de corriente `INA219` junto a su resistencia `Shunt` Kelvin de $0.100\ \Omega$ (1%, $2\ W$).
- Suelde el **zócalo torneado** `DIP-28` para el microcontrolador `ATmega328P-PU` sin el chip insertado.
- Atornille la aleta metálica del `MOSFET IRLZ44N` (`TO-220`) al disipador de aluminio anodizado aplicando pasta térmica y suelde sus tres terminales en los pads oblongos reforzados.
- Suelde las borneras de tornillo `TB1` (entrada $12\ V\ DC$) y `TB2` (salida al actuador) y aporte una capa uniforme de aleación `SAC305` sobre las ventanas descubiertas de máscara (*Solder Mask Relief*) en las pistas de $2.50\ mm$.

---

4. Discretización del Controlador `PID` y Anti-Windup (`Plano 3`)

📐 Plano 3: Diagrama de Bloques del Controlador PID Discreto y Respuesta con Anti-Windup



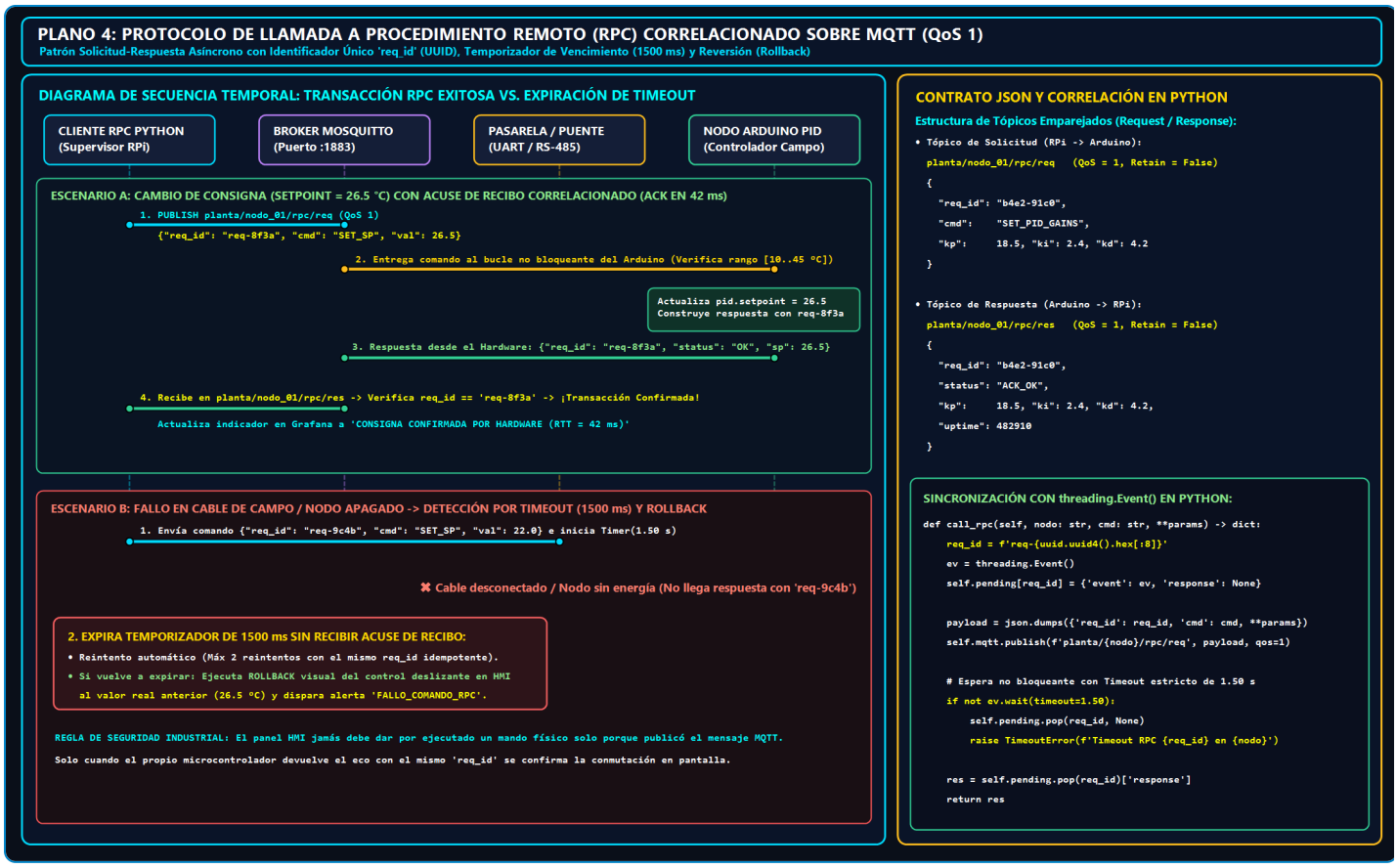
Paso 4.1 — Compilación y Carga del Firmware Determinista en Arduino UNO

- Abra el archivo `codigo_fuente/controlador_pid_discreto_rpc_arduino.ino` en Arduino IDE o PlatformIO.
- Verifique las tres mejoras industriales implementadas en la función `computeDiscretePID()`:
- **Muestreo exacto por temporizador no bloqueante:** $T_s = 100\ ms$ $(10\ Hz)$.
- **Derivada sobre la medición (**$-y[k]$**):** $D[k] = -K_d \cdot (y[k] - y[k-1]) / T_s$, eliminando el golpe derivativo (*Derivative Kick*) cuando el operador modifica de golpe el `Setpoint` desde Grafana.
- **Anti-Windup por recorte condicional (*Clamping*):** congela la acumulación del integrador $I[k]$ cuando la salida `PWM` alcanza 0 o 255.

---

5. Protocolo `RPC` Correlacionado sobre `MQTT` (`Plano 4`)

📐 Plano 4: Diagrama de Secuencia del Protocolo RPC Correlacionado con Timeout y Rollback



Paso 5.1 — Prueba de Transacción Confirmada y Prueba de Fallo con `Rollback`

- Ejecute el cliente supervisor en la Raspberry Pi:

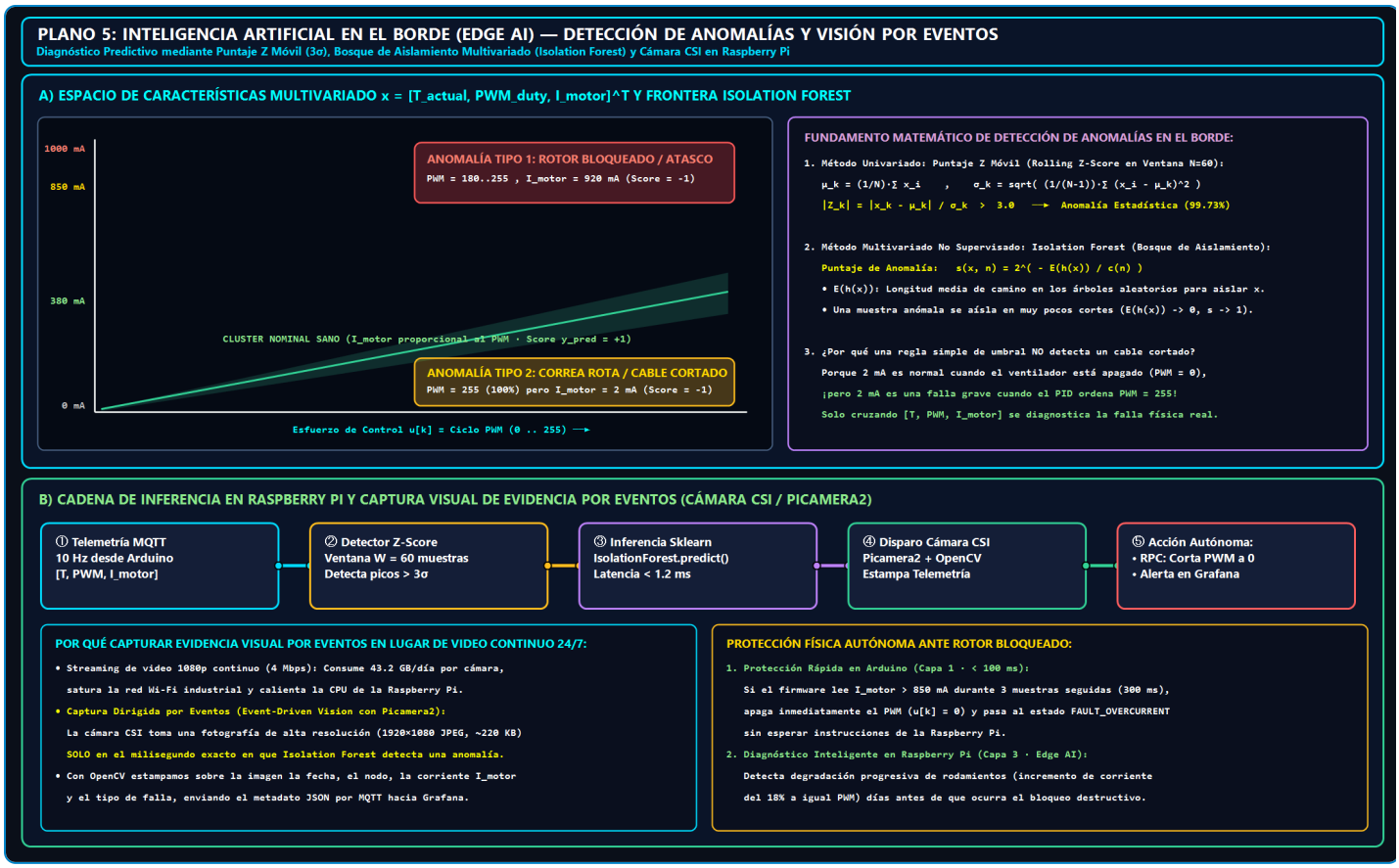
  ```bash
  python3 codigo_fuente/cliente_rpc_correlacionado_mqtt.py
  ```

- Verifique en consola que cada comando enviado en `planta/nodo_01/rpc/req` incluye un `req_id` único (por ejemplo, `req-8f3a19c2`) y que el hilo espera en `threading.Event.wait(timeout=1.50)` hasta recibir exactamente ese mismo `req_id` en `planta/nodo_01/rpc/res`.
- Desconecte momentáneamente el cable serial del Arduino y envíe una nueva consigna: compruebe cómo tras 1.50 s y 2 reintentos idempotentes, el servicio publica el evento de `Rollback` restaurando el indicador de la interfaz al valor real anterior.

---

6. Detección de Anomalías con `Edge AI` y Cámara `CSI` (`Plano 5`)

📐 Plano 5: Arquitectura Edge AI con Puntaje Z Móvil, Isolation Forest y Cámara CSI por Eventos



Paso 6.1 — Ejecución del Detector `Isolation Forest` y Captura Forense por Eventos

- Inicie en terminales separadas el motor de inferencia y el servicio de cámara `CSI`:

  ```bash
  python3 codigo_fuente/detector_anomalias_edge_ai_isolation_forest.py
  python3 codigo_fuente/captura_visual_eventos_camara_csi.py
  ```

- **Prueba de Anomalía Tipo 1 (Rotor Bloqueado):** frene suavemente el eje del ventilador durante $400\ ms$. Observe cómo la corriente I_{motor} supera los $850\ mA$, el Arduino corta autónomamente el `PWM` a 0 en menos de $300\ ms$ y el motor `Edge AI` clasifica `ROTOR_BLOQUEADO_SOBRECORRIENTE`, disparando una fotografía `1080p` con marca de agua en `/var/lib/mecatrobotica/evidencia_visual/`.
- **Prueba de Anomalía Tipo 2 (Correa Rota / Cable Abierto):** desconecte el hilo positivo del ventilador mientras el controlador `PID` ordena $\text{PWM} = 255$. Aunque $I_{motor} \approx 2\ mA$ no viola ningún límite de sobrecorriente, el modelo multivariado **`Isolation Forest`** detecta inmediatamente que la combinación $[\text{PWM}=255,\ I_{motor}=2\ mA]$ se ubica fuera de la variedad nominal ($\text{Score} = -1$) y reporta `CORREA_ROTA_O_CABLE_ACTUADOR_ABIERTO`.