Manual de Ingeniería: Pruebas de Inyección de Fallas (`FI-01` a `FI-05`), Diagnóstico Forense de Campo (`Anexo A`), Gemelo Digital `Wokwi` (`Anexo B`) y Rúbrica de Puesta en Marcha (`100 Puntos`)

- **Curso:** Internet de las Cosas (`IoT`) con Arduino y Raspberry Pi (`MecatroboticApp`)
- **Módulo 11 (Cierre General del Curso):** Validación Experimental, Resiliencia Industrial y Certificación Full-Stack

---

1. Fundamentos de Ingeniería de Confiabilidad e Inyección de Fallas (*Fault Injection Testing*)

En la ingeniería de sistemas embebidos e industriales, demostrar que un prototipo opera cuando todos los cables están intactos y la base de datos responde en $1\ ms$ constituye apenas el 20% del trabajo. El 80% restante consiste en garantizar que el sistema **transicione a un estado físicamente seguro** ante fallas de hardware y **se recupere sin intervención humana** ante cortes de comunicación o mantenimiento de servidores.

1.1. Análisis Matemático y Temporal de las Cinco Pruebas Oficiales (`FI-01` a `FI-05`)

Prueba `FI-01`: Rotura de Sensor Térmico con Calefactor Encendido

Cuando el interruptor `SW1` abre el circuito del termistor `NTC` en plena fase de calentamiento, la lectura del conversor analógico-digital cae al extremo del rango ($ADC \leq 5$ o $ADC \geq 1018$). Dado que el lazo determinista de muestreo del Arduino se ejecuta con periodo $T_s = 50\ ms$ ($20\ Hz$), el tiempo total de reacción hasta el apagado seguro cumple la cota estricta:

$$

\Delta t_{reacción} = T_{muestreo} + T_{cómputo_FSM} + T_{conmutación_SSR} \leq 50\ ms + 1\ ms + 10\ ms = 61\ ms < 100\ ms

$$

En ese instante, la máquina de estados finitos (`FSM`) fuerza:

- Salida del calefactor a cero lógico (`PIN_HEATER_RELAY = LOW`).
- Ciclo de trabajo del ventilador al máximo (`PWM = 255`, equivalente al 100%).
- Máscara de error activa (`err = 1`) y retención en el estado `FAULT_SAFE_SHUT` hasta que un operador subsane el cable y confirme el rearme.

Prueba `FI-02`: Desacoplo entre Control Local y Transporte (`RS-485` desconectado por $60\ s$)

Al abrir `SW2` durante $\Delta t_{corte} = 60\ s$, comprobamos la propiedad fundamental del control jerárquico distribuido: el microcontrolador **jamás bloquea su ejecución esperando al puerto serie**. Mientras tanto, el servicio puente en la Raspberry Pi captura la excepción de *timeout* Modbus, publica `"status": "serial_fault"` en `MQTT` y ejecuta un bucle de reconexión no destructiva cada $2\ s$.

Prueba `FI-03`: Continuidad de Serie Temporal ante Caída de `InfluxDB` ($180\ s$)

Si el contenedor `iot-influxdb` se detiene durante $\Delta t_{caída} = 180\ s$ mientras los nodos publican a una tasa $f_{pub} = 1\ Hz$, el volumen de muestras retenidas en el buffer persistente `SQLite` en modo `WAL` (*Write-Ahead Logging*) es exactamente:

$$

N_{retenidas} = f_{pub} \cdot \Delta t_{caída} = 1\ Hz \cdot 180\ s = 180\ muestras

$$

Al restablecerse `InfluxDB`, el ingestor vacía la cola FIFO respetando la estampa temporal original (`timestamp` de origen), logrando **cero pérdida de muestras** ($N_{perdidas} = 0$).

Prueba `FI-04`: Diagnóstico Predictivo Electromecánico (`PWM` vs. Corriente `INA219`)

Si la temperatura supera la consigna y el controlador PID aplica un mando significativo ($\text{PWM} \geq 120$), un ventilador sano de $12\ \text{V}$ debe absorber una corriente nominal entre $120\ \text{mA}$ y $350\ \text{mA}$. Si el cable de $+12\ \text{V}$ se corta (`SW3` abierto), el sensor `INA219` mide $I_{\text{fan}} < 15\ \text{mA}$. La regla de coherencia física en el borde detecta la falla en menos de $3\ \text{segundos}$, mucho antes de que la inercia térmica de la cámara dispare una alarma tardía de sobretemperatura.

Prueba `FI-05`: Defensa Criptográfica y Autorización por Tópico (`ACL`)

Aunque un atacante o un nodo periférico comprometido (`esp32_nodo01`) posea un certificado `CA` válido y credenciales de conexión al puerto `8883`, la matriz de control de acceso (`ACL`) del broker `Mosquitto` restringe su permiso exclusivamente a `write planta/exterior/env01/#`. Cualquier intento de publicar en `planta/zona1/arduino01/rpc/request` es descartado en memoria por el broker.

---

2. Diagnóstico Forense de Campo (`Anexo A`): Causa Raíz y Solución de las 5 Fallas Clásicas

2.1. Falla 1: `PermissionError: [Errno 13] Permission denied: '/dev/ttyACM0'`

- **Causa física/SO:** En distribuciones Linux basadas en Debian (`Raspberry Pi OS`), los nodos de dispositivo `/dev/ttyACM` *y* `/dev/ttyUSB` pertenecen al grupo `dialout` con permisos `crw-rw----`.
- **Solución definitiva:** Agregar el usuario de ejecución al grupo mediante `sudo usermod -aG dialout $USER` y declarar `SupplementaryGroups=dialout` en la unidad `systemd`.

2.2. Falla 2: Pérdida de los Primeros Comandos tras Abrir el Puerto Serie en Python

- **Causa física:** Al instanciar `serial.Serial()`, el driver USB-CDC del kernel hace caer la línea de control `DTR` (*Data Terminal Ready*). En el Arduino UNO/Nano, `DTR` está acoplada mediante un condensador de $100\ \text{nF}$ al pin `RESET`, provocando el reinicio del microcontrolador y su permanencia en el *bootloader* durante $\sim 1.5\ \text{s}$.
- **Solución definitiva:** Esperar `time.sleep(2.0)` tras abrir el puerto, ejecutar `ser.reset_input_buffer()` y aguardar el mensaje de saludo del firmware antes de transmitir tramas `RPC`.

2.3. Falla 3: Reinicio Espontáneo (*Brown-out*) al Conmutar Relés o Motores

- **Causa física:** El pico de corriente de inserción (*inrush current*) de una bobina o motor DC compartido sobre el riel de $5\ \text{V}$ produce una caída resistiva e inductiva $\Delta V = I_{\text{pico}} \cdot R_{\text{cable}} + L\ (dI)\ /\ (dt)$ que hunde la tensión por debajo del umbral de *Brown-out Detector* ($\approx 4.3\ \text{V}$ en ATmega328P o $4.63\ \text{V}$ en Raspberry Pi).
- **Diagnóstico en Raspberry Pi:** Ejecutar `vcgencmd get_throttled`. Si devuelve un valor distinto de `0x0` (por ejemplo `0x50000` o `0x50005`), el bit `0` o `16` confirma subtensión histórica o actual.
- **Solución definitiva:** Separar la fuente de potencia de $12\ \text{V}\ /\ 5\ \text{V}$ de las cargas respecto de la fuente lógica, uniendo solo la referencia `GND` o aislando mediante optoacopladores.

2.4. Falla 4: Tramas `RS-485` Truncadas en los Últimos 1 o 2 Bytes

- **Causa física:** En el UART del microcontrolador, escribir en el buffer de transmisión retorna antes de que el registro de desplazamiento físico (*Hardware Shift Register*) termine de enviar el último bit de parada por el cable. Si el pin `DE/RE` del `MAX485` pasa a `LOW` prematuramente, el transceptor se apaga a mitad del último carácter.
- **Solución definitiva:** Invocar obligatoriamente `Serial.flush()` seguido de una guarda temporal de $100\ \mu\text{s}$ (`delayMicroseconds(100)`) antes de conmutar `digitalWrite(PIN_DERE, LOW)`.

2.5. Falla 5: Bloqueo o Lecturas Corruptas en el Bus `I2C` con Cables Largos

- **Causa física:** El bus `I2C` emplea colector abierto (*open-drain*). La constante de tiempo del flanco de subida depende de la resistencia *pull-up* R_p y la capacitancia parásita total del cableado C_{bus}:

$$
$\tau_{subida} = R_p \cdot C_{bus}$
$$

Con las resistencias internas débiles del microcontrolador ($R_p \approx 35\ k\Omega$) y un cable de más de $30\ cm$ ($C_{bus} \approx 120\ pF$), el tiempo de subida supera el límite de la especificación `I2C` a $100\ kHz$ ($\approx 1000\ ns$).

- **Solución definitiva:** Instalar resistencias *pull-up* externas de $2.2\ k\Omega$ a $3.3\ k\Omega$ hacia $3.3\ V$ y reducir la frecuencia de reloj a $50\ kHz$ mediante `Wire.setClock(50000)`.

---

3. Gemelo Digital con `Wokwi` (`RFC 2217`) y Rúbrica de Certificación de 100 Puntos

El estándar **RFC 2217** permite transportar no solo los bytes de datos (`TX` / `RX`) sobre un socket TCP, sino también las señales de control de módem (`DTR`, `RTS`, velocidad en baudios y paridad). De este modo, el mismo servicio `serial_mqtt_bridge.py` se conecta sin cambios de lógica a `rfc2217://localhost:4000` (simulador Wokwi) o a `/dev/ttyUSB0` (planta real).

Ponderación Oficial de la Rúbrica de Puesta en Marcha (`100 Puntos`)

$$
$P_{total} = P_{Hardware\,(20)} + P_{Firmware\,(25)} + P_{Edge\,(25)} + P_{Seguridad+FI\,(30)} = 100$ puntos
$$

Al superar los $90\ puntos$ con las cinco pruebas `FI-01` a `FI-05` aprobadas, el estudiante completa el **100 % del programa académico** (10 Módulos, 40 Lecciones, Proyecto Integrador y Anexos Técnicos).