Guía Paso a Paso del Laboratorio 11: Protocolo de Inyección de Fallas (`FI-01` a `FI-05`), Gemelo Digital en Wokwi (`RFC 2217`) y Puesta en Marcha (`FAT/SAT`)

- **Curso:** Internet de las Cosas (`IoT`) con Arduino y Raspberry Pi (`MecatroboticApp`)
- **Módulo 11 (Cierre del Proyecto Integrador y Anexos A, B y C):** Pruebas de Estrés e Inyección de Fallas, Diagnóstico Forense de Campo, Simulación Virtual en `Wokwi` y Certificación de 100 Puntos
- **Duración Estimada de Práctica:** 120 minutos

---

1. Objetivos de Ingeniería del Laboratorio Final

Un sistema ciberfísico industrial jamás se da por terminado cuando funciona únicamente en condiciones nominales de escritorio. En este laboratorio final someteremos nuestra **Estación Mecatrónica Inteligente de Control Térmico, Ambiental y Energético** al protocolo completo de aceptación en fábrica (**FAT**, *Factory Acceptance Test*) y aceptación en sitio (**SAT**, *Site Acceptance Test*):

- **Instrumentar un Banco Físico de Inyección de Fallas:** Incorporar interruptores de corte deliberado (`SW1`, `SW2`, `SW3`) para provocar roturas de sensor, cortes del par diferencial `RS-485` y desconexiones de potencia del ventilador de $12\ V$ sin dañar los componentes.
- **Ejecutar y Medir las 5 Pruebas Oficiales de Estrés (`FI-01` a `FI-05`):** Verificar con cronometraje instrumental que cada subsistema cumpla los tiempos máximos de reacción y criterios de autocuración.
- **Configurar el Gemelo Digital 100 % Simulado (`Anexo B`):** Enlazar el simulador a nivel de ciclo de instrucción `Wokwi` con `pyserial` y `Docker` a través del puente TCP `RFC 2217` (`rfc2217://localhost:4000`).
- **Aplicar la Hoja de Diagnóstico Forense de Campo (`Anexo A`):** Auditar permisos POSIX (`dialout`), temporización `DTR`, subtensión (`vcgencmd get_throttled`), integridad de trama `RS-485` y capacitancia parásita en `I2C`.
- **Certificar los 100 Puntos de la Rúbrica Final:** Ejecutar el validador automatizado para obtener la calificación final del curso y desbloquear los logros de ingeniería.

---

2. Fase 1: Montaje de la Caja de Inyección de Fallas y Ejecución de `FI-01` a `FI-05`

Paso 1.1: Intercalado de los Interruptores de Prueba (`SW1`, `SW2`, `SW3`)

Antes de energizar la estación, intercale en serie los tres interruptores de palanca o deslizantes del banco de pruebas:

- `SW1` **(Corte de Sensor NTC — Prueba `FI-01`):** En serie con el terminal de señal del termistor NTC que va hacia el nodo analógico `A0` (manteniendo la resistencia *pull-down* o *pull-up* de referencia conectada en el lado del microcontrolador para forzar el extremo del ADC al abrir el circuito).
- `SW2` **(Corte de Bus Industrial `RS-485` — Prueba `FI-02`):** Interruptor bipolar (`DPST`) en serie con las líneas diferenciales `A` y `B` entre los transceptores `MAX485` del Arduino y de la Raspberry Pi.
- `SW3` **(Corte de Potencia de Ventilador $12\ V$ — Prueba `FI-04`):** En serie entre la salida de corriente del sensor `INA219` (`VIN-`) y el terminal positivo del ventilador brushless de $12\ V$.

Paso 1.2: Ejecución Cronometrada de las Cinco Pruebas (`FI-01` a `FI-05`)

Código	Procedimiento Experimental Paso a Paso	Verificación Instrumental y Criterio de Aprobación
`FI-01`	Con el calefactor activo y el sistema en `NORMAL_AUTO`, abra bruscamente el interruptor `SW1` (simulando cable de termistor cortado).	En $\Delta t_{reacción} < 100\ ms$, el Arduino detecta lectura fuera de rango físico, pasa a `FAULT_SAFE_SHUT` (`st=3`), apaga el calefactor (`heat=0`), fuerza el ventilador al 100% (`fan=255`) y publica `err=1`.
`FI-02`	Abra el interruptor bipolar `SW2` cortando el enlace `RS-485` durante $60\ s$ continuos y luego vuelva a cerrarlo.	El Arduino **continúa** ejecutando su lazo PID térmico local sin interrupción. El puente Python publica `"status": "serial_fault"` en `MQTT`; al cerrar `SW2`, la telemetría se reanuda sola sin reiniciar la Raspberry Pi.
`FI-03`	En la Raspberry Pi, ejecute `docker stop iot-influxdb`, espere 3 minutos (180 muestras) y ejecute `docker start iot-influxdb`.	Durante los $180\ s$, `store_and_forward_ingestor.py` guarda las 180 muestras en `SQLite` (`WAL`). Al iniciar `InfluxDB`, el lote histórico se vacía y la gráfica de `Grafana` muestra la curva continua sin ningún hueco temporal.
`FI-04`	Eleve la temperatura por encima del *Setpoint* para que el PID comande `PWM >= 120` y abra el interruptor `SW3` del ventilador.	El servicio `edge_anomaly_detector.py` correlaciona $PWM \geq 120$ con $I_{INA219} < 15\ mA$ y publica en $< 3\ s$ la alerta crítica `FAN_OPEN_CIRCUIT` en `MQTT` y `Grafana`.
`FI-05`	Desde una terminal, intente inyectar `SET:SP=38.0` autenticándose con el usuario restringido `esp32_nodo01`.	El broker `Mosquitto` bloquea la publicación aplicando `/etc/mosquitto/acl.conf` y la consigna del Arduino permanece inalterada.

Para automatizar la auditoría de los cinco ensayos, ejecute en la Raspberry Pi:

```
python3 /opt/mecatrobotica/capstone/services/suite_pruebas_inyeccion_fallas_fi01_fi05.py
```

---

3. Fase 2: Gemelo Digital y Simulación Virtual con `Wokwi` (`RFC 2217`)

Si desea validar todo el flujo sin hardware físico o reproducir escenarios extremos antes de ir a planta:

- Abra la carpeta del firmware en **VS Code** con la extensión oficial de **Wokwi** instalada y copie el archivo `wokwi_simulacion_gemelo_digital_rfc2217.toml` como `wokwi.toml`.
- Inicie la simulación (`Wokwi: Start Simulator`). El simulador habilitará el servidor **RFC 2217 (Serial over TCP)** en el puerto local `4000`.
- En el servicio puente de Python, reemplace el dispositivo físico `/dev/ttyACM0` por la URL de red de `pyserial`:

  ```python
  import serial
  ser = serial.serial_for_url("rfc2217://localhost:4000", baudrate=115200, timeout=1.0)
  ```
- Levante la pila de contenedores en su computador con `docker compose up -d`. Verá cómo las variaciones del termistor NTC virtual en Wokwi se reflejan en tiempo real en su dashboard de **Grafana**.

---

4. Fase 3: Diagnóstico Forense de Campo y Certificación Final (`100 Puntos`)

- Ejecute la herramienta de auditoría forense en la Raspberry Pi para descartar errores de permisos, caídas de tensión o buses bloqueados:

  ```bash
  chmod +x diagnostico_forense_campo_troubleshooting.sh
  ```

./diagnostico_forense_campo_troubleshooting.sh
```

- Una vez verificados los 6 puntos de salud y aprobadas las 5 pruebas de inyección de fallas, ejecute el evaluador de puesta en marcha:
  ```bash
  python3 validador_puesta_en_marcha_fat_sat_100pts.py
  ```
- Verifique la obtención de `100.0 / 100.0 puntos` y el veredicto `SOBRESALIENTE — CERTIFICADO INGENIERO IOT FULL-STACK`.
  ```