

GUÍA DE COMUNICACIÓN ENTRE MÁQUINAS

1. EL CONCEPTO SERVIDOR Y CLIENTE

Para que dos máquinas se comuniquen mediante OPC-UA, una de ellas debe actuar como Servidor (la que tiene los datos, como un PLC o una máquina) y la otra como Cliente (la que lee o escribe esos datos, como un SCADA o una aplicación de escritorio).

2. PASO 1: CONFIGURACIÓN DEL ESPACIO DE DIRECCIONES (ADDRESS SPACE)

El Servidor organiza sus datos en una estructura de carpetas y nodos. Cada variable (temperatura, estado de marcha, velocidad) tiene un identificador único.

- **NodeID:** Identificador único de la variable.
- **BrowseName:** Nombre legible para humanos.
- **Value:** El dato real.

3. PASO 2: PROTOCOLOS Y PUERTOS

La comunicación suele realizarse sobre `opc.tcp`. El puerto estándar por defecto es el 4840.

- Es vital asegurar que el Firewall permita el tráfico en este puerto para que las máquinas puedan "verse" entre sí.

4. PASO 3: CONECTIVIDAD Y SEGURIDAD

Antes de intercambiar datos, el Cliente y el Servidor deben realizar un "apretón de manos" (Handshake).

- **Endpoint URL:** Dirección donde escucha el servidor (ej: ``opc.tcp://192.168.1.10:4840``).
- **Políticas de Seguridad:** Se puede elegir entre comunicación abierta (None), firmada (Sign) o firmada y cifrada (SignAndEncrypt).

5. PASO 4: LECTURA Y ESCRITURA (DATA ACCESS)

Una vez conectados, el Cliente puede:

- **Leer:** Consultar el valor actual de una variable.
- **Escribir:** Cambiar el valor de una variable (ej. enviar una consigna de velocidad).
- **Suscripción:** El cliente se "apunta" a una variable y el servidor le avisa automáticamente solo cuando el valor cambia, ahorrando mucho tráfico de red.

6. PASO 5: IMPLEMENTACIÓN EN PYTHON

Para este proyecto usaremos la librería ``asyncua``, que es la implementación más moderna y potente para manejar el intercambio de información de forma asíncrona.

- El Servidor simulará datos virtuales de una planta industrial.
- El Cliente visualizará estos datos en una interfaz gráfica premium.