# Receta de Ejecución Directa — Proyecto 04: Central `MQTT` Local (`Mosquitto`) y Puente Bidireccional

> [!IMPORTANT]

> **Objetivo Práctico (Sin Relleno Teórico):** Habilitar el broker `Mosquitto` en la red local en 3 comandos, ejecutar el puente `puente_mqtt.py` de 35 líneas y controlar el ventilador y el relé del Arduino en vivo desde tu PC o celular usando `MQTT Explorer`.

---

## Paso 1: Configurar `Mosquitto` en 3 Comandos (2 minutos)

Ejecuta en la terminal de tu Raspberry Pi:

```
sudo bash -c 'cat > /etc/mosquitto/conf.d/taller.conf <<EOF
listener 1883 0.0.0.0
allow_anonymous true
EOF'
sudo systemctl restart mosquitto
```

---

## Paso 2: Iniciar el Puente Universal `USB <-> MQTT` (3 minutos)

Asegúrate de tener el Arduino UNO conectado por USB a la Raspberry Pi con el circuito del Proyecto 03 y ejecuta:

```
cd ~/iot_practico
source venv/bin/activate
python3 puente_mqtt.py
```

- Todo lo que envía el Arduino se publica en `casa/estacion/datos`.
- Todo lo que envíes a `casa/estacion/orden` llega directo al Arduino.

---

## Paso 3: Prueba de Fuego desde tu PC con `MQTT Explorer` (3 minutos)

- Abre **MQTT Explorer** en tu PC, escribe Host `central-iot.local`, Puerto `1883` y haz clic en **Connect**.
- En el panel derecho (**Publish**):

- **Topic:** `casa/estacion/orden`

- **Payload (raw):** `FAN:255` → Haz clic en **Publish** (el ventilador arranca al máximo).

- **Payload (raw):** `RELE:1` → Haz clic en **Publish** (el relé se enciende al instante).

---

## Paso 4: Verificación en 60 Segundos

- [x] **Check 1:** `sudo systemctl status mosquitto` muestra `active (running)` en verde.
- [x] **Check 2:** En `MQTT Explorer` ves actualizarse `casa/estacion/datos` cada 2 segundos.
- [x] **Check 3:** Al publicar `RELE:1` o `FAN:255` en `casa/estacion/orden`, el hardware físico responde en menos de 1 segundo.