

Entradas y salidas digitales

Video de *kick-off*

binefa.com/iot-v

- * [IO básico](#). IO usando gemelos sin overlay: [doutbit](#) y [dout](#)

- * [IO con overlay](#)

- * Pregunta al Gemini con [archivo de contexto](#) adjuntando [IO con overlay](#)

A partir del ejemplo buttonRelay haz buttonRelay4. Tiene dos gemelos conectables, uno es de 4 interruptores de neón (conectados de DI0 a DI3) y 4 relés (el programa en Python conecta DO4 con DO7). DI3 controla el relé de DO4 y DI0 el relé de DO7 (DI1-DO6, DI2-DO3).

- * Uso de Traductor IA para que el código sea transpilable

- * El [proyecto](#) funciona sin enseñar los componentes

- * Añadidos los componentes

En twin-switches:

```
"components": [  
  { "id": "sw0", "type": "switch", "props": { "label": "DI0" }, "binding": { "toPLC": "sw0" } },  
  { "id": "sw1", "type": "switch", "props": { "label": "DI1" }, "binding": { "toPLC": "sw1" } },  
  { "id": "sw2", "type": "switch", "props": { "label": "DI2" }, "binding": { "toPLC": "sw2" } },  
  { "id": "sw3", "type": "switch", "props": { "label": "DI3" }, "binding": { "toPLC": "sw3" } }  
]
```

En twin-relays:

```
"components": [  
  { "id": "led4", "type": "led", "props": { "label": "DO4" }, "binding": { "fromPLC": "rel4" } },  
  { "id": "led5", "type": "led", "props": { "label": "DO5" }, "binding": { "fromPLC": "rel5" } },  
  { "id": "led6", "type": "led", "props": { "label": "DO6" }, "binding": { "fromPLC": "rel6" } },  
  { "id": "led7", "type": "led", "props": { "label": "DO7" }, "binding": { "fromPLC": "rel7" } }  
]
```

- * Proyecto [buttonRelay4](#) finalizado

- * **Reto:** Desarrollar buttonRelay8 controlando DI0-DI7 y DO0-DO7