5Tech Edge Bridge (ESP32) — MVP de la Fase 1 — Manual de usuario y operación

Configuración, flasheo, ajuste de Wi-Fi / MQTT, comandos Bluetooth LE y el flujo de demostración del ESP32 Edge Bridge.

`esp32-edge-bridge` · Revisión Borrador (Preliminar) · 2026-07-08 · Estado: **PRELIMINAR**

> **PRELIMINAR.** Este manual describe un MVP de la Fase 1 sobre una placa de desarrollo ESP32 lista para usar. Es un **prototipo de monitoreo y automatización**, no un producto terminado ni certificado. Los procedimientos y valores marcados con `[TBD]` están pendientes de confirmación. Los documentos de ingeniería autorizados están en `modules/5tech-edge-iot-bridge/`; siga siempre la documentación entregada con el firmware que flasheó.

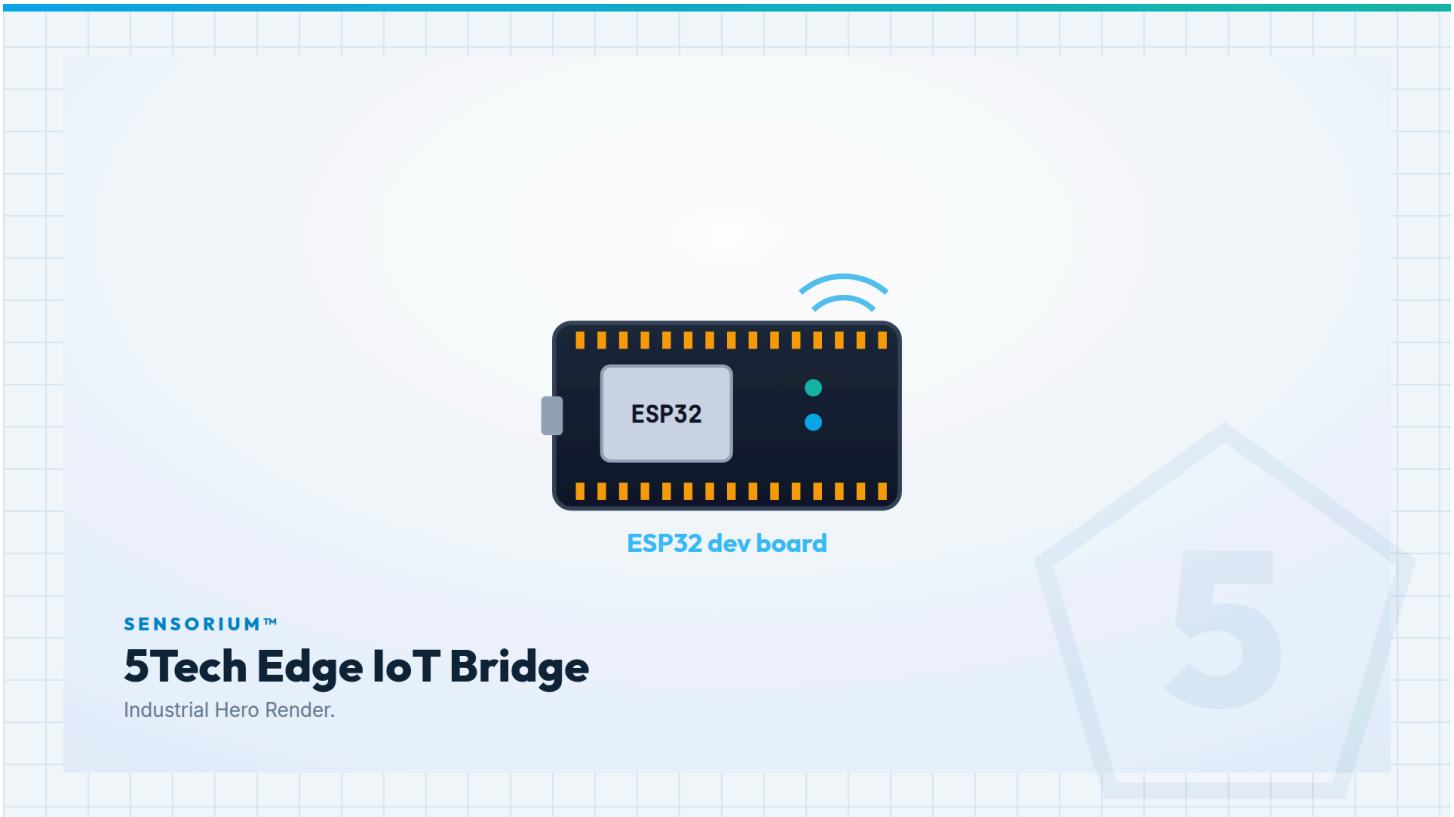

Figura FIG-001 — 5Tech Edge Bridge, el MVP de la Fase 1 que conecta una entrada con un flujo de automatización por Wi-Fi, Bluetooth LE y MQTT.

1. Seguridad

Lea este capítulo antes de configurar, operar o modificar el Edge Bridge.

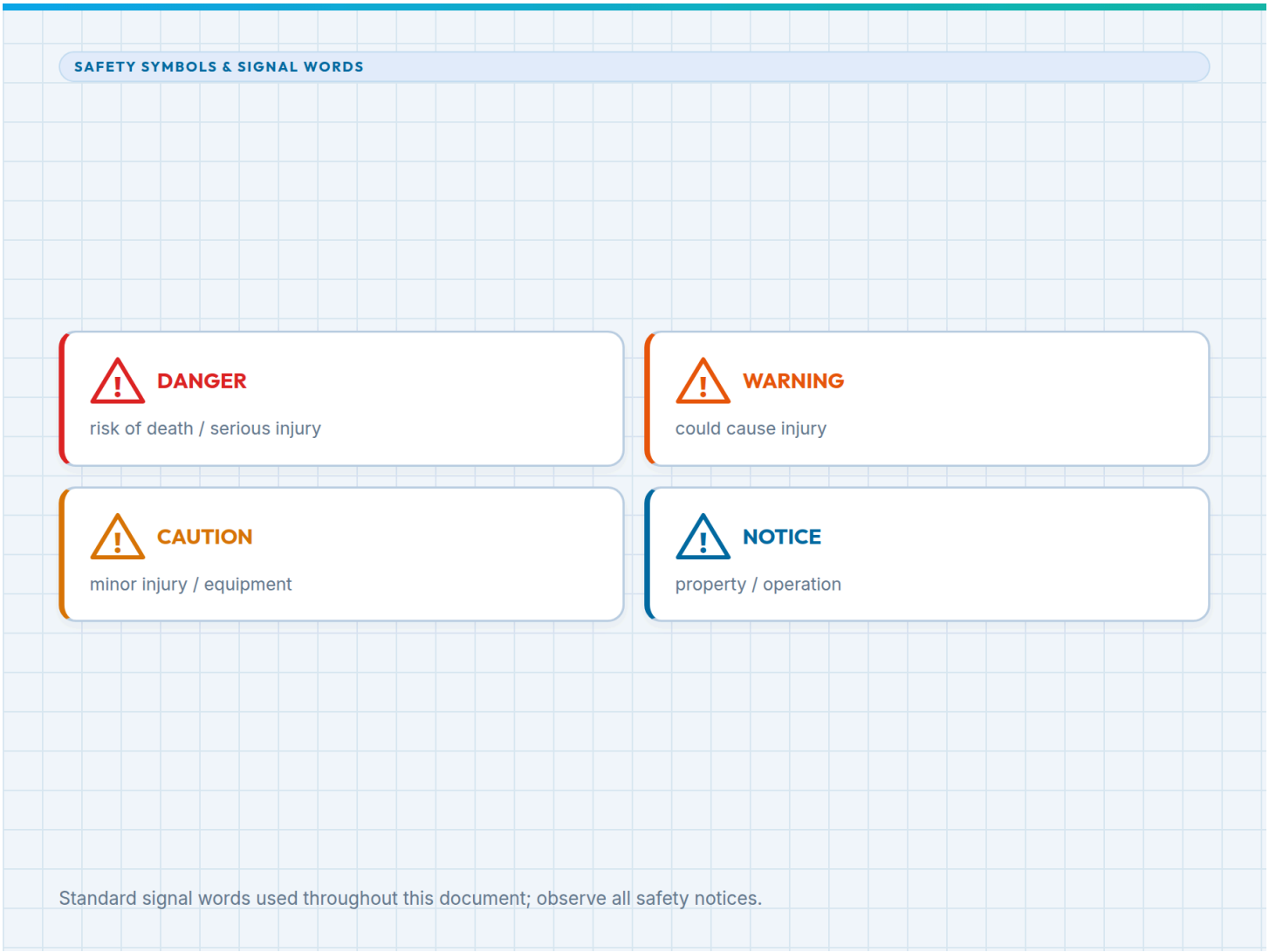

Figura FIG-002 — Palabras de advertencia (PELIGRO · ADVERTENCIA · PRECAUCIÓN · AVISO) y los pictogramas utilizados en este manual.

1.1 Palabras de advertencia

PALABRA DE ADVERTENCIA	SIGNIFICADO
PELIGRO	Peligro que **causará** la muerte o lesiones graves si no se evita.
ADVERTENCIA	Peligro que **podría** causar la muerte o lesiones graves si no se evita.
PRECAUCIÓN	Peligro que podría causar lesiones leves o moderadas si no se evita.
AVISO	Práctica **no relacionada con lesiones personales** — daños a la propiedad o al equipo, o un aviso importante (alcance / limitación) que el lector debe atender.

1.2 Esto no es un dispositivo de seguridad

> **AVISO — No es un dispositivo de seguridad.** El 5Tech Edge Bridge es un **prototipo de monitoreo y automatización** sobre una placa de desarrollo. **No** tiene certificación de seguridad de ningún tipo y **no** debe utilizarse para proteger a personas ni equipos, para detener maquinaria, para crear una zona de protección, ni como parte de ninguna función de seguridad funcional. Sus estados de peligro (`WARNING` / `DANGER`) son señales de demostración en una canalización de datos, no salidas de seguridad. Nunca confíe en él donde un fallo pudiera causar lesiones o pérdidas. *(Este es un aviso de alcance sobre lo que el dispositivo no es — un `AVISO` según §1.1, no una advertencia de peligro sobre el propio dispositivo.)*

1.3 Uso previsto y público

El Edge Bridge está diseñado para uso **por desarrolladores, técnicos e integradores en un banco de pruebas de confianza** para validar la canalización de sensor a automatización y para demostrarla a clientes y partes interesadas. Cualquier otro uso — en particular el despliegue en campo, el uso de seguridad o la conexión a una red no confiable — se considera uso indebido para este MVP de la Fase 1.

1.4 Personal cualificado y competencia requerida

La puesta en marcha, la operación, el cableado y la modificación del Edge Bridge son únicamente para **personal cualificado**. Para este MVP de la Fase 1, eso significa una persona que:

- sea un **desarrollador, técnico o integrador** competente en electrónica de baja tensión y en la manipulación con seguridad ESD de una PCB desnuda;
- pueda flashear firmware y leer una consola serie, y entienda la lógica de 3.3 V y que los pines GPIO **no** toleran 5 V (§1.5);
- entienda MQTT / Bluetooth LE en una LAN de confianza y la naturaleza en texto plano y sin autenticación de este prototipo (§1.7);
- y, antes de cablear cualquier cosa a la salida `Con1` o a las entradas `Det1` / `Det2`, pueda confirmar la tensión, la corriente y el aislamiento del circuito externo contra la ficha técnica.

Una persona sin esta competencia no debe poner en marcha, cablear ni modificar el dispositivo sin supervisión.

1.5 Seguridad eléctrica

La placa se alimenta desde una fuente **USB 5 V** y opera a baja tensión (lógica de 3.3 V). El riesgo de descarga eléctrica de la propia placa es mínimo, pero:

> **AVISO — Daño al equipo.** Los pines GPIO son de **lógica de 3.3 V y no son tolerantes a 5 V**. Aplicar más de 3.3 V a una entrada GPIO, cortocircuitar pines o invertir una conexión puede destruir la placa. Alimente la placa desde una sola fuente a la vez (USB *o* un raíl externo de 3.3 V/5 V, no ambos). Observe precauciones ESD al manipular la placa desnuda.

- Use un cable USB apto para datos y un puerto o fuente USB en buen estado conocido.
- Si conecta un sensor externo o una carga a la salida `Con1`, confirme que su tensión y corriente están dentro del valor nominal del pin (`[TBD]`) antes de cablear. No conmute la red eléctrica ni una carga inductiva directamente desde un pin GPIO.

1.6 Nota sobre radiofrecuencia (RF)

La placa transmite por **Wi-Fi (2.4 GHz)** y **Bluetooth LE** a través de la antena integrada del módulo. La potencia de transmisión es baja y típica de un módulo ESP32 de consumo. Las homologaciones de radio regionales para el *prototipo ensamblado* son `[TBD]` — este es un dispositivo de desarrollo, así que opérelo únicamente donde tal uso esté permitido y manténgalo en una red de banco de confianza.

1.7 Riesgos residuales

Incluso cuando se utiliza según lo previsto, permanecen los siguientes: el dispositivo puede **enclavar un estado de peligro** y seguir reportando `DANGER` después de que un sensor se recupere (esto es deliberado — véase §3.3); en una red en texto plano, cualquiera que pueda alcanzar el broker puede comandar el dispositivo; y Bluetooth LE y Wi-Fi comparten una antena, por lo que la capacidad de respuesta de BLE varía con la carga de Wi-Fi. Ninguno de estos es una mitigación de seguridad.

---

2. Acerca de este manual

Alcance. Este manual cubre la configuración, el flasheo, el ajuste, la operación y la resolución de problemas del Edge Bridge MVP de la Fase 1 en una placa de desarrollo ESP32. **No** reemplaza la documentación de ingeniería detallada del repositorio del firmware.

Público. Desarrolladores, técnicos e integradores según se describe en §1.3.

Convenciones.

- Los avisos de seguridad utilizan las palabras de advertencia de §1.1 y aparecen **antes** del paso al que se aplican.
- Los procedimientos están numerados, con una acción por paso.
- `[TBD]` marca un valor pendiente de confirmación. Nunca sustituya un `[TBD]` por un valor supuesto.
- La `fuente monoespaciada` denota comandos, nombres de archivo, nombres de pines, temas (topics) y texto de interfaz/consola.

Documentos relacionados.

DOCUMENTO	UBICACIÓN	USO
Ficha técnica	`datasheets/esp32-edge-bridge-datasheet/`	Especificaciones de la placa de desarrollo, interfaces, pinout, valores nominales
Descripción general del producto	`products/esp32-edge-bridge/`	Posicionamiento y aspectos destacados
Firmware — configuración	`modules/5tech-edge-iot-bridge/docs/setup.md`	Cadenas de herramientas, secrets, compilación, flasheo
Firmware — API MQTT	`modules/5tech-edge-iot-bridge/docs/mqtt-api.md`	Temas y esquemas de carga útil
Firmware — API Bluetooth	`modules/5tech-edge-iot-bridge/docs/bluetooth-api.md`	Tabla GATT, comandos BLE
Firmware — guion de demostración	`modules/5tech-edge-iot-bridge/docs/demo-script.md`	El recorrido para cliente / inversor

3. Descripción general del producto y teoría de funcionamiento

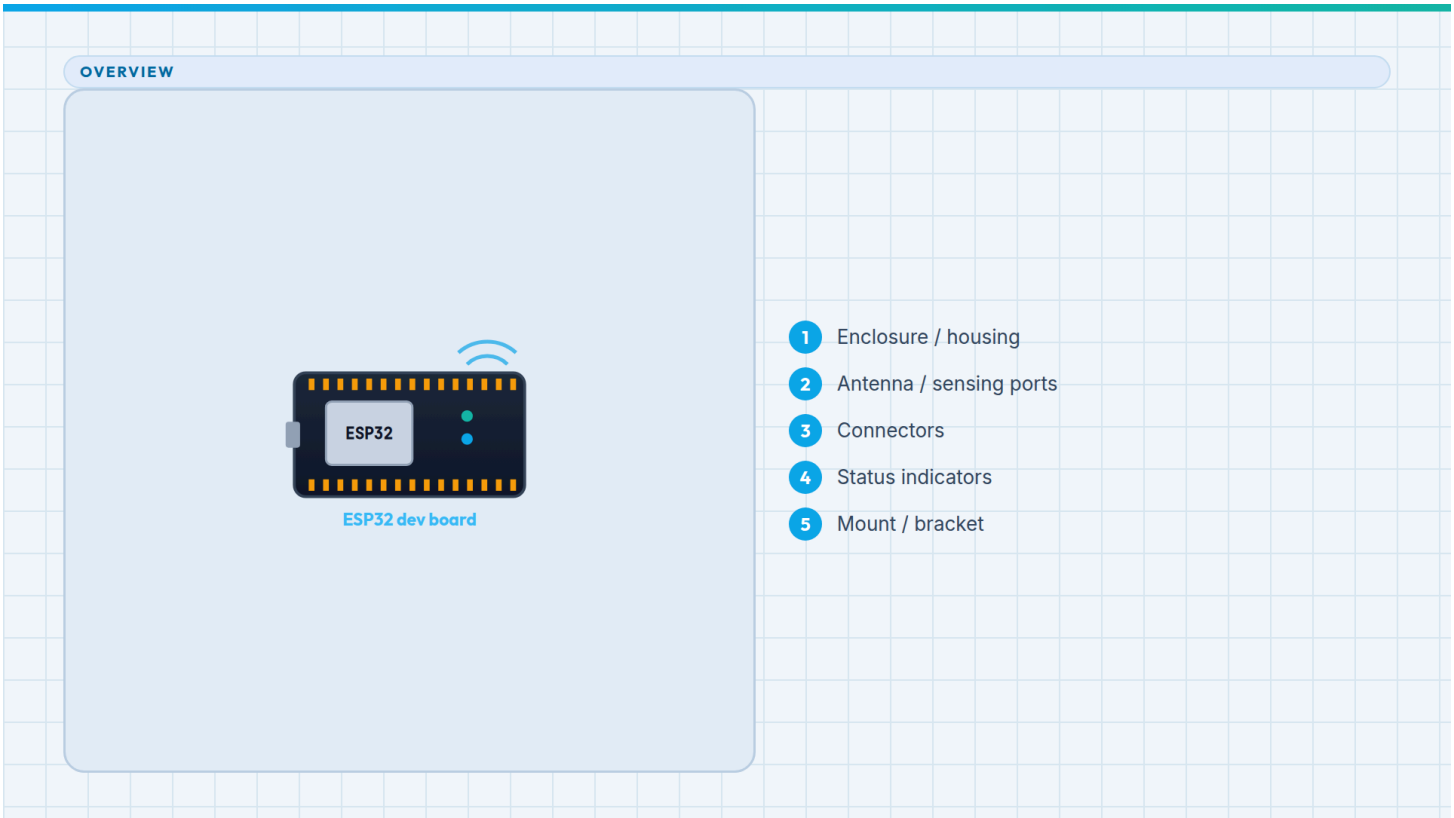

Figura FIG-003 — El Edge Bridge y sus conexiones: USB, entradas opcionales y las dos radios de 2.4 GHz.

3.1 Qué hace

El Edge Bridge lee una entrada, decide si esa entrada representa una condición `WARNING` o `DANGER`, y vuelve a publicar la decisión como JSON estructurado por Bluetooth LE y MQTT para que un flujo de automatización pueda actuar sobre ella. Funciona en una placa de desarrollo ESP32 estándar y no necesita hardware de sensor para demostrarse.

3.2 Dónde se sitúa — Sense → Connect → Decide

- **Sense** — un sensor GPIO (Det1 / Det2), el botón integrado, un sensor simulado o un comando remoto indican una condición.
- **Connect** — el ESP32 resuelve su estado y publica por Wi-Fi a un broker MQTT; un canal Bluetooth LE da control local incluso sin red.
- **Decide** — un flujo de automatización (n8n / Node-RED / personalizado) se suscribe y alimenta paneles, alertas y registros.

3.3 Teoría de funcionamiento

1. **Identidad del dispositivo.** Al arrancar, el dispositivo deriva su identidad de la MAC del módulo: el id MQTT `5tech-bridge-a1b2c3` y el nombre BLE `5Tech-IoT-Bridge-a1b2c3` (los

últimos seis dígitos hex de la MAC).

2. **Máquina de estados.** Nueve estados — `BOOTING`, `WIFI_CONNECTING`, `MQTT_CONNECTING`, `READY`, `WARNING`, `DANGER`, `OFFLINE`, `CONFIG_MODE`, `ERROR`. Una vez operativo, el estado se resuelve en prioridad estricta: `error > danger > warning > config_mode > !wifi > READY`. Cada estado tiene su propio patrón de parpadeo del LED integrado.
3. **Enclavamiento de peligros.** Los niveles de peligro se ordenan `NORMAL < WARNING < DANGER`. Elevar a un nivel superior gana y publica un evento; elevar a un nivel igual o inferior es una operación nula. Un peligro se **enclava** — un sensor que se recupera **no** lo borra automáticamente. Solo un `clear` explícito (o un tiempo de espera configurado, desactivado por defecto) devuelve a `NORMAL`.
4. **Publicación.** La telemetría se publica cada 5 s, un evento se publica de inmediato ante cualquier cambio, y el estado se retiene en el broker con un Last Will, de modo que un suscriptor tardío siempre ve el estado verdadero. Un peligro tiene prioridad sobre la conectividad: un dispositivo en `DANGER` sigue reportando `DANGER` incluso si Wi-Fi se cae.
5. **Comandos.** La misma gramática de comandos funciona en serie, Bluetooth LE y MQTT. Los comandos se ponen en cola y se ejecutan en el bucle principal, nunca en una tarea de callback de radio.

---

4. Controles e indicadores

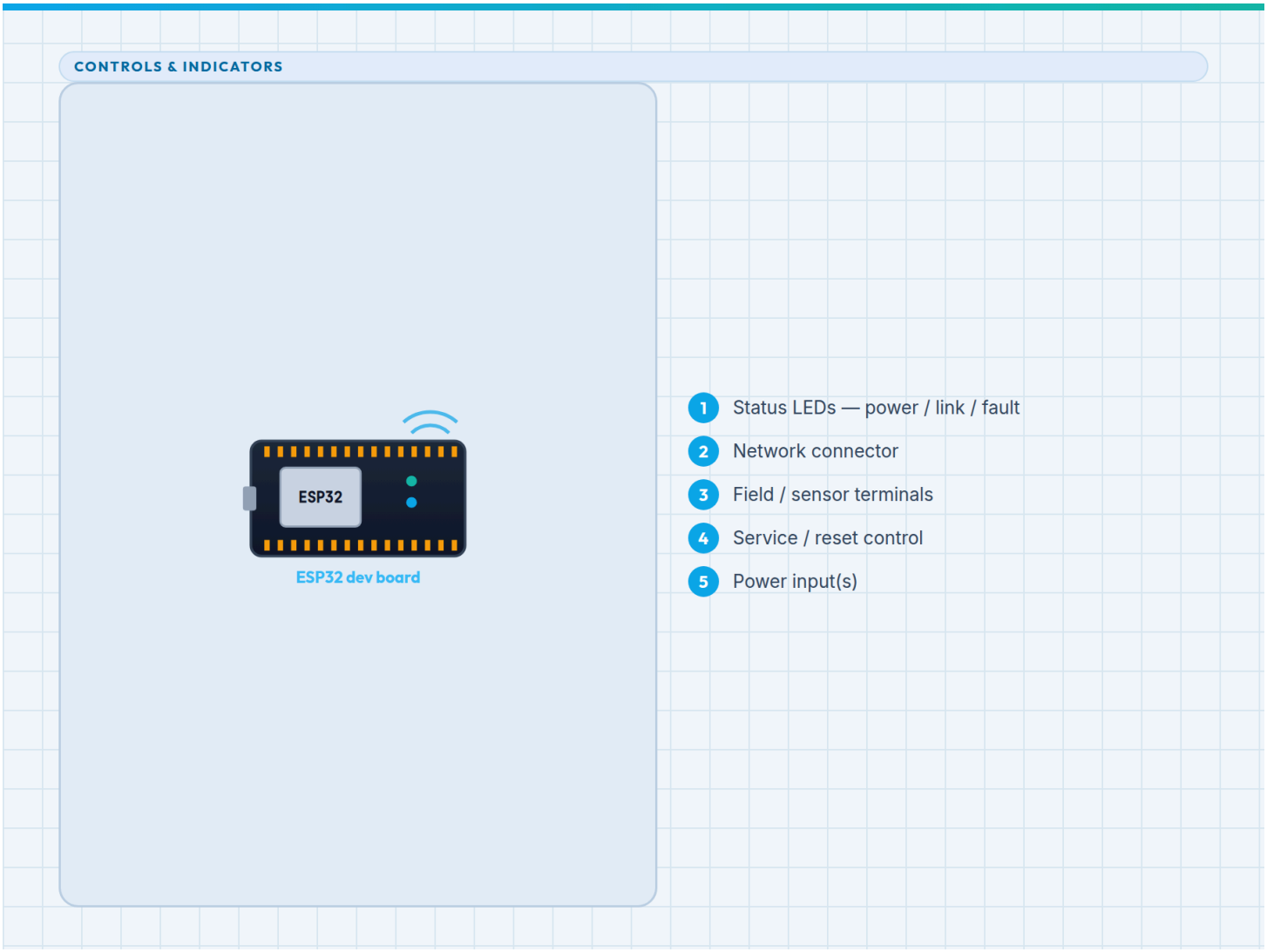

Figura FIG-004 — Los pines, el botón integrado, el LED de estado y el conector USB, con una leyenda numerada.

Los números de pin exactos son los predeterminados de la Fase 1; confírmelos con la ficha técnica y la serigrafía de su placa.

CONTROL / INDICADOR	FUNCIÓN	ESTADOS
Conector USB	Alimentación, flasheo, consola serie (115 200 baudios)	Conectado / desconectado
LED de estado (`GPIO2`)	Estado del dispositivo	Un patrón de parpadeo distinto por estado (READY, WARNING, DANGER, OFFLINE, CONFIG_MODE, …)
Botón (`GPIO0`, BOOT integrado)	Disparador manual	**Pulsación corta** = genera advertencia / borra un peligro activo · **Pulsación larga (≥1.5 s)** = genera peligro
Entrada Det1 (`GPIO18`)	Entrada digital, activa en bajo	Afirmada → WARNING; valor reflejado por BLE
Entrada Det2 (`GPIO19`)	Entrada digital, activa en bajo	Afirmada → DANGER; valor reflejado por BLE
Salida Con1 (`GPIO23`)	Contacto configurable remotamente	Establecido por BLE / MQTT / serie (`set_con1 0|1`)
Zumbador (`GPIO25`, opcional)	Anunciación local	Desactivado por defecto (`ENABLE_BUZZER = 0`)

> **AVISO.** En la mayoría de los devkits ESP32, `GPIO2` es el LED integrado. La Fase 1 lo usa para el estado del dispositivo, así que `Con1` se movió a `GPIO23`. Para restaurar el cableado original, establezca `CON1_GPIO = 2` y `ENABLE_STATUS_LED = 0` (una guarda en tiempo de compilación impone que nunca compartan un pin).

---

5. Operación

5.1 Antes de empezar

Necesita: una placa de desarrollo ESP32-WROOM-32, un cable USB apto para datos, un computador anfitrión con la cadena de herramientas y (para la ruta Wi-Fi/MQTT) un broker MQTT como `mosquitto` en la misma LAN de confianza. No se requiere hardware de sensor.

Hay dos variantes de firmware disponibles y son intercambiables en el cable:

	VARIANTE ESP-IDF	VARIANTE PLATFORMIO / ARDUINO
Cadena de herramientas	ESP-IDF (compila con v5.4.4)	PlatformIO (compila con 6.1.19)
Compilación	`idf.py build`	`pio run`

La configuración completa de la cadena de herramientas está en `modules/5tech-edge-iot-bridge/docs/setup.md`.

5.2 Configurar secrets y flashear

> **AVISO.** `secrets.h` está en gitignore y nunca debe confirmarse (commit). Solo se rastrea `secrets.example.h`.

Variante ESP-IDF:

1. Copie la plantilla: `cp firmware/esp-idf/main/config/secrets.example.h firmware/esp-idf/main/config/secrets.h`
2. Edite `secrets.h` — establezca `WIFI_SSID`, `WIFI_PASSWORD` y `MQTT_HOST`.
3. Compile y flashee: `cd firmware/esp-idf && idf.py set-target esp32 && idf.py build`
4. `idf.py -p /dev/ttyUSB0 flash monitor`

Variante Arduino / PlatformIO:

1. `cp firmware/platformio/src/config/secrets.example.h firmware/platformio/src/config/secrets.h`
2. Edite `secrets.h` como arriba.
3. `cd firmware/platformio && pio run && pio run -t upload`
4. `pio device monitor`

5.3 Primer encendido y comprobación de estado

1. Conecte la placa por USB y abra el monitor serie a **115 200 baudios**.
2. Confirme que el dispositivo imprime su identidad (`5tech-bridge-…`) e inicia su secuencia de arranque.
3. Obsérvelo asociarse con Wi-Fi, conectarse al broker e iniciar la publicidad Bluetooth LE.
4. Confirme que el LED de estado se asienta en el patrón `READY`.
5. Si Wi-Fi no se conecta en ~15 s, el dispositivo continúa **fuera de línea**: BLE y serie siguen funcionando y el estado se lee como `OFFLINE`. Esto es esperado, no un fallo.

5.4 Modo de banco (sin Wi-Fi)

Dejar `WIFI_SSID` en su marcador de posición es un **modo de banco compatible**: el dispositivo omite Wi-Fi, arranca en `CONFIG_MODE` y permanece totalmente utilizable por Bluetooth LE y la consola serie. Todos los comandos funcionan; solo la publicación MQTT no está disponible.

5.5 El flujo de demostración

1. En un portátil en la misma LAN, suscríbase al broker: `mosquitto_sub -h <broker> -t "5tech/iotbridge/+/#" -v`
2. Dispare una advertencia — desde la consola serie, desde un teléfono por Bluetooth LE (escriba `test_warning` en la característica **Cmd**), o publicando en el tema de comandos. Las tres son equivalentes.
3. Observe `[STATE] READY → WARNING`, el cambio de cadencia del LED, y que el evento llega al broker.
4. Dispare `test_danger`. El LED va más rápido y el peligro se **enclava**.
5. Envíe `clear`. El dispositivo vuelve a `READY`.
6. Explique el punto: solo el disparador está simulado — sustitúyalo por un sensor real y nada aguas abajo cambia.

5.6 Apagado ordenado

Simplemente desconecte el USB. En una desconexión abrupta, el broker publica el **Last Will** retenido, por lo que el estado del dispositivo cambia a `OFFLINE` para cualquier suscriptor. En el siguiente arranque, el dispositivo lo sobrescribe con un estado en vivo.

---

6. Configuración

6.1 Métodos de configuración

MÉTODO	USO	NOTAS
`secrets.h` (en tiempo de compilación)	Credenciales de Wi-Fi y del broker	En gitignore; requiere reflasheo
Banderas de características (banderas de compilación)	Activar/desactivar subsistemas	Protegidas con `#ifndef`; anule con `-D`
Comandos en tiempo de ejecución (serie / BLE / MQTT)	Disparar eventos, fijar umbrales, accionar `Con1`, reiniciar	Gramática idéntica en los tres transportes

6.2 Comandos en tiempo de ejecución

Aceptados como texto simple **o** JSON, en cualquier transporte.

COMANDO	EFECTO
`help`	Lista los comandos
`status`	Estado actual, como JSON
`config`	Configuración en ejecución, como JSON
`test_warning`	Genera un evento `WARNING`
`test_danger`	Genera un evento `DANGER`
`clear`	Borra el evento activo
`set_threshold <n>`	Fija el umbral de peligro (la forma JSON puede mover ambos límites)
`set_con1 <0|1>`	Acciona el contacto Con1
`reboot`	Reinicia (la respuesta se vacía primero)

Ejemplos:

`test_danger`
`{"command":"set_threshold","danger_threshold":40,"warning_threshold":90}`

6.3 Umbrales

El valor del sensor se modela como una **distancia en cm** — menor significa más cerca significa peor. Una lectura por debajo de `warning_threshold` (predeterminado 100) genera WARNING; por debajo de `danger_threshold` (predeterminado 50) genera DANGER. `set_threshold` rechaza cualquier par donde warning no sea estrictamente superior a danger o que deje el

rango válido `[0, 200]`. **Los umbrales no se persisten** — un reinicio restaura los valores predeterminados compilados.

6.4 Comandos Bluetooth LE

1. Escanee en busca de `5Tech-IoT-Bridge-…`, o filtre por el UUID de servicio `4fafc201-1fb5-459e-8fcc-c5c9c331914b` (más fiable — el nombre está en la respuesta de escaneo).
2. Conéctese y expanda el servicio personalizado.
3. **Active las notificaciones en la característica `Resp` antes de escribir en `Cmd`.** Las respuestas a un `Resp` no suscrito se descartan — esta es la causa más común de "sin respuesta".
4. Escriba un comando como texto UTF-8 (p. ej. `test_danger`) en `Cmd`. Las respuestas llegan como notificaciones en `Resp`, con la forma `OK <cmd>: <message>` o `ERR <cmd>: <message>`.
5. Escriba `0x01` (un byte en bruto, no el texto "1") en `Con1` para poner `GPIO23` en alto; `0x00` lo pone en bajo.

6.5 Actualización de firmware

Flashee por USB (§5.2). **No** hay OTA ni actualización por BLE en la Fase 1.

---

7. Mantenimiento

El Edge Bridge es firmware sobre una placa de desarrollo; no hay tareas de mantenimiento físico programadas. "Mantenimiento" aquí significa mantener el firmware y su configuración en buen estado.

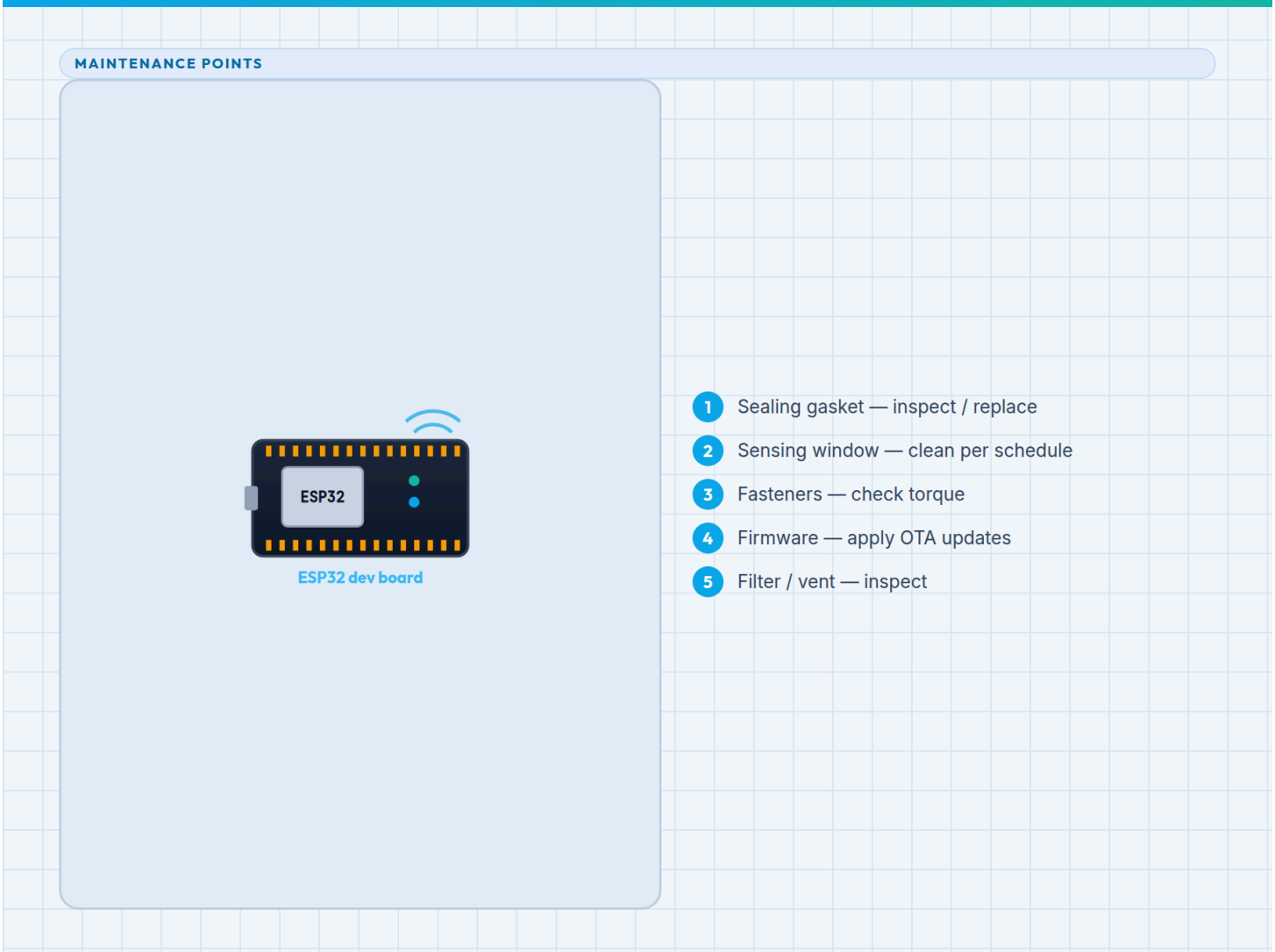

Figura FIG-005 — Puntos de atención rutinaria: la conexión USB, el archivo de secrets y los conjuntos de pruebas automatizadas.

INTERVALO	TAREA	NOTAS
Según necesidad	Reflashear tras un cambio de firmware	§5.2
Según necesidad	Mantener `secrets.h` actualizado y sin confirmar	Verifique con `git check-ignore`
Antes de etiquetar una compilación	Ejecutar el conjunto de pruebas de host (`make -C tests/host`)	77 aserciones, sin hardware
Antes de etiquetar una compilación	Ejecutar la verificación de paridad en el cable	Confirma que ambas variantes siguen coincidiendo
Rutina	Inspeccionar el cable y el conector USB	Un cable solo de carga es un fallo común
Rutina	Mantener la placa seca, protegida de estática y libre de cortos	PCB desnuda, sin envolvente

8. Resolución de problemas

SÍNTOMA	CAUSA POSIBLE	ACCIÓN
Sin salida serie	Baudios incorrectos o cable solo de carga	Use 115 200 baudios y un cable de datos; confirme el puerto serie
Sin respuesta BLE a un comando	Notificaciones de `Resp` no activadas	Active las notificaciones en `Resp` **antes** de escribir en `Cmd` (§6.4)
Dispositivo no está en el broker	`MQTT_HOST` incorrecto, broker caído o puerto equivocado	Compruebe `secrets.h`, confirme el broker en el puerto `1883`, compruebe la LAN
Atascado en `CONFIG_MODE`	Credenciales Wi-Fi de marcador de posición o ausentes	Fije `WIFI_SSID` / `WIFI_PASSWORD` reales en `secrets.h` y reflashee
El estado se lee `OFFLINE`	Wi-Fi no conectado	Esperado sin Wi-Fi; BLE y serie siguen funcionando; compruebe credenciales/cobertura
El broker muestra el dispositivo `OFFLINE`	El Last Will se disparó en una desconexión abrupta	Normal tras una pérdida de energía; se autorrepara al reconectar
El peligro no se borra	Modelo de peligro con enclavamiento	Envíe `clear` explícitamente — un sensor que se recupera no lo borra automáticamente (§3.3)
`Con1` parece conmutar el LED	`Con1` en `GPIO2` (cableado original)	La Fase 1 usa `GPIO23`; véase el AVISO de §4
BLE tartamudea bajo carga	Wi-Fi y BLE comparten una antena	Esperado en ESP32; reduzca la carga de Wi-Fi durante el trabajo con BLE
La compilación ESP-IDF falla por el tamaño de partición	La app desborda la partición predeterminada de 1 MB	Use `CONFIG_PARTITION_TABLE_SINGLE_APP_LARGE` (Arduino: `huge_app.csv`)
Un comando se descarta silenciosamente	Cola de comandos (profundidad 8) llena	Reduzca la tasa de comandos; el descarte se registra en la serie

Si un fallo persiste, capture el registro serie, la salida de `status` / `config` y los mensajes del broker, y luego consulte los documentos de ingeniería en `modules/5tech-edge-iot-bridge/`.

---

9. Especificaciones

Este manual **no** duplica los valores de las especificaciones. Todas las especificaciones de la placa de desarrollo, de interfaz, eléctricas y ambientales se mantienen en la ficha técnica:

- **Ficha técnica:** `datasheets/esp32-edge-bridge-datasheet/document.md`

ÁREA	DÓNDE ENCONTRARLA
Plataforma de hardware (placa de desarrollo ESP32)	Ficha técnica §1, §4
Mecánica (factor de forma)	Ficha técnica §5
Eléctrica (alimentación USB, nivel lógico)	Ficha técnica §6
Interfaces y protocolos (Wi-Fi, BLE, MQTT, GPIO, serie)	Ficha técnica §7
Ambiental y valores nominales	Ficha técnica §8
Conformidad (n/a — prototipo de desarrollo)	Ficha técnica §9

---

10. Garantía y soporte

Garantía. Este es un **MVP de la Fase 1 / prototipo de desarrollo**, proporcionado tal cual para evaluación y demostración. No conlleva garantía de producto comercial ni certificación. Los términos de garantía de producción son `[TBD]` y pertenecen a la Fase 2.

Soporte.

- 5Tech — Vancouver, BC, Canadá
- Correo electrónico: `info@5tech.ca`
- Web: `https://5tech.ca`

Al contactar con el soporte, tenga a mano: la variante y versión del firmware, el id del dispositivo (`5tech-bridge-…`), el registro serie y los mensajes del broker en torno al problema.

AVISO — No es un dispositivo de seguridad. No despliegue este prototipo en ningún rol donde un fallo pudiera dañar a personas o equipos. Es un MVP de monitoreo y automatización, nada más. *(Un aviso de alcance — un `AVISO` según §1.1, no una advertencia de peligro sobre el propio dispositivo.)*

Figuras

ID	TÍTULO	ESTADO	NOMBRE DE ARCHIVO	PROPÓSITO
FIG-001	Edge Bridge — Render principal del prototipo	Ausente	`renders/hero_edge-bridge.png`	Portada / imagen destacada del manual
FIG-002	Símbolos de seguridad y palabras de advertencia	Ausente	`figures/safety_symbols.png`	Palabras de advertencia y pictogramas para el capítulo de seguridad
FIG-003	Descripción general del producto	Ausente	`figures/overview.png`	Orientar al usuario sobre la placa y sus conexiones
FIG-004	Controles e indicadores	Ausente	`figures/controls_indicators.png`	Identificar los pines, el botón, el LED y el conector USB
FIG-005	Puntos de mantenimiento	Ausente	`figures/maintenance_points.png`	Localizar los puntos de atención rutinaria

---

Este manual es preliminar y está sujeto a cambios. Véase `revision.md` para el control de cambios, `references.md` para las fuentes e `image_prompts.md` para el índice de figuras/prompts.