CITOLAB

PROYECTO DE TECNOLOGÍA · LABORATORIO

Estación de Inclusión IoT

Guía del proyecto: un dispositivo que registra, de forma trazable, quién incluye cada cassette, cuál y cuándo — conectado a la plataforma del laboratorio.

Benjamín & Emilio
Desarrolladores del proyecto

Camilo Espinoza
Mentor técnico

19-06-2026
Versión 1.0

11	$600K	3	0
Etapas con hitos verificables (0 → 10)	Premio total si lo logran ($300K c/u)	Mundos que dominarán: firmware, redes y diseño 3D	Escaneos que se pueden perder (buffer offline)

1. ¿Qué vamos a construir?

En el laboratorio, los **Tecnólogos Médicos (TM)** incluyen cassettes en las **estaciones de inclusión**. Hoy ese paso no queda registrado. Vamos a construir un **dispositivo IoT** que, en cada estación:

- **Identifica al TM** cuando escanea el código 2D de su credencial (inicio de sesión).
- **Registra cada cassette** que escanea (un DataMatrix grabado) antes de incluirlo.
- Envía cada evento a la plataforma del laboratorio: **quién, qué cassette, cuándo** y en qué estación.
- Si se cae el WiFi, **guarda y reintenta** — no se pierde ningún escaneo.

> **El objetivo de verdad es que aprendan.** Más que el aparato, lo valioso es que resuelvan problemas reales: programar el micro, integrar un lector, hablar con una API, diseñar e imprimir una carcasa y depurar lo que falle. El premio es el incentivo; el aprendizaje es el premio mayor.

2. Cómo funciona

Escanea credencial → Inicia sesión (identifica al TM) → Escanea cassettes → Envía a la plataforma → Confirma en pantalla

El cerebro es un **ESP32** con una pantalla OLED. El lector entrega el código escaneado; el ESP32 arma el mensaje y lo manda por WiFi al servidor. La idea clave de diseño es tratar el escaneo como *"llega un*

texto": da igual si ese texto viene del lector real o se escribe a mano para probar. Eso permite avanzar **casi todo el proyecto sin el lector físico**.

3. El hardware

Pieza	Qué es	Detalle clave
Placa	ideaspark ESP32 0.96" OLED	ESP32-WROOM-32 + pantalla integrada. Es el cerebro.
Pantalla	OLED SSD1306 128×64	Va por I²C en GPIO21 (SDA) / GPIO22 (SCL).
Lector	ScanHome SH-400 (2D)	Lee DataMatrix/QR. Tiene beep y LED propios. *(ver sección 4)*
Carcasa	Case impreso en 3D	Lo diseñan y lo imprimen ellos.

Feedback al TM (doble): el *beep + LED del lector* confirman "leí un código"; la *OLED* confirma "lo registré: TM, cassette, OK / guardado offline".

4. El desafío del lector (y la decisión)

El lector que ya tenemos es la versión **USB**: entrega los datos por el conector USB, comportándose como un teclado. El problema: el **ESP32-WROOM no puede leer un dispositivo USB** (su puerto USB es solo para programarlo). Por eso, tal como está, no se conecta directo.

> **Decisión:** usar un lector con salida **UART/TTL** (texto plano por cable). Así el firmware se reduce a "leer una línea".
> • **Opción A (recomendada):** SH-400 versión TTL — misma óptica de calidad para el grabado.
> • **Opción B (económica):** lector GM65 UART — más barato y con muchos tutoriales (validar que lea el grabado).
> • El lector USB actual **no se desperdicia**: sirve para probar en un PC que los DataMatrix de los cassettes se leen bien.

Conexión final: lector por **UART2 (GPIO16/17)** con un pequeño divisor de nivel 5V→3.3V para proteger el ESP32.

5. El registro en la plataforma

El servidor expondrá un punto de entrada `POST /api/v1/inclusion-scans`, protegido con una clave por estación. El dispositivo envía un mensaje como este:

Campo	Significado
`stationId`	Qué estación de inclusión es (ej. `INCL-01`).
`tmCode`	Quién (del escaneo de la credencial).
`cassetteCode`	Qué cassette (el DataMatrix, ej. `1234567-1.1`).
`scannedAt`	Cuándo se escaneó (fecha/hora ISO, del reloj del dispositivo).
`deviceEventId`	Id único que inventa el dispositivo por cada escaneo. Si reenvían desde el buffer offline con el mismo id, el servidor no lo duplica.

Esta parte (el endpoint en el servidor) la implementa Camilo; los chicos consumen algo ya listo y documentado.

6. Qué dice cada cassette (códigos de ejemplo)

El DataMatrix grabado en un cassette codifica **texto plano** con el formato `{informe}-{muestra}.{cassette}`: el número de informe (el caso), un guion, y el subíndice que dice de qué frasco (muestra) salió y qué cassette de ese frasco es. Si una muestra grande se divide, el subíndice gana niveles (ej. `2.3.1`). Las láminas usan el mismo código **más un número al final** — la estación debe distinguirlas y no registrarlas como inclusión. Estos códigos son reales en formato (los números son inventados):

`1234567-1.1`

Informe 1234567, muestra 1, cassette 1. El caso típico.

`1234567-1.2`

Mismo informe y muestra, segundo cassette.

`1234567-2.3.1`

Muestra 2, cassette 3, subdividido: el subíndice puede tener más niveles.

`1234567-1.1-2`

¡Ojo! El `-2` final es número de lámina: esto es una **lámina**, no un cassette. Hay que rechazarlo.

Pruébenlos de verdad:

- **Con el lector USB en un PC:** impriman esta página y escaneen los códigos — el lector debería "tipear" el texto tal cual. Así validan lector y formato antes de escribir una línea de firmware.
- **En la Etapa 4 (escaneo simulado):** escriban estos mismos textos en el monitor serial. Son sus datos de prueba oficiales.
- **En el firmware:** un cassette válido calza con la expresión regular `^\d+-\d+(\.\d+)+$`; una lámina es lo mismo más `-\d+` al final.
- **La credencial del TM:** su formato se define con Camilo (algo como `TM-…`); mientras tanto, simúlenla.

7. La lógica del programa (máquina de estados)

El firmware vive en unos pocos "estados" bien definidos:

- **Arranque:** conecta WiFi y pone la hora (NTP).

- **Espera login:** el primer escaneo se interpreta como credencial → inicia sesión.
- **Sesión activa:** cada escaneo es un cassette → lo envía → confirma en pantalla.
- **Cierre:** al re-escanear una credencial (cambia de TM) o por inactividad (timeout).
- **Sin red:** guarda el escaneo en memoria y reintenta solo.

8. Que nada se pierda

Buffer offline

Si el WiFi se cae, los escaneos se guardan en la memoria del ESP32 y se reenvían con reintentos cuando vuelve la red. Cero pérdidas.

Listo para crecer

Cada estación tiene su `station_id` y su clave, así que mañana se puede instalar en varias estaciones sin rehacer nada.

9. La carcasa 3D

Para diseñar la caja, recomendamos **Tinkercad** (gratis, en el navegador, facilísimo para empezar). Si quieren dar el salto a algo profesional, **Onshape** o **Fusion 360**. *Blender* es genial para modelado artístico, pero incómodo para una caja con tolerancias y encajes — déjenlo para otro día.

10. El plan: 11 etapas

Cada etapa termina con algo que **funciona y se puede mostrar**. Las etapas 0–5 y 7–8 se pueden hacer **sin el lector físico** (escribiendo el código a mano para probar), así que pueden empezar ¡ya!

#	Etapa	Qué aprenden / logran	Lector
0	Setup + Blink	Instalar Arduino IDE y drivers; hacer parpadear un LED. El "hola mundo".	No
1	Pantalla OLED	Escribir texto y estados en la pantalla.	No
2	WiFi + hora	Conectarse a la red y obtener la hora real (NTP).	No
3	Enviar datos	Hacer un POST a un servidor de prueba (HTTP + JSON).	No
4	Escaneo simulado	Máquina de estados: login → cassettes, escribiendo el código por el monitor serial.	No
5	Conexión real con la plataforma	Autenticar con la clave y enviar el evento real; manejar errores.	No
6	Lector físico	Conectar el lector por UART, divisor de nivel, leer escaneos reales.	Sí
7	Buffer offline	Guardar en memoria y reintentar: que nada se pierda.	No
8	Config sin recompilar	Portal para configurar WiFi/estación sin tocar el código.	No
9	Carcasa 3D	Medir, modelar, imprimir y montar.	—
10	Prueba en terreno 🎉	Probarlo en una estación real, ajustar y entregar.	Sí

11. Problemas que pueden aparecer (y cómo enfrentarlos)

Problema posible	Cómo lo enfrentamos
El lector no lee bien el DataMatrix grabado	Probar primero en PC; preferir el SH-400 TTL; ajustar distancia e iluminación.
Los 5V del lector dañan el ESP32	Usar un divisor de nivel en la línea de datos. Medir antes de conectar.
La hora sale mal si arranca sin internet	Sincronizar por NTP al conectar; a futuro, un reloj de hardware (RTC).
"No sé por dónde empezar"	El plan está ordenado de fácil a difícil; cada etapa es un logro. Y está el mentor.

12. Cómo se gana el premio

Se considera logrado cuando:

- Un TM inicia sesión escaneando su credencial y ve su identidad en la pantalla.
- Al escanear un cassette, el evento (quién/qué/cuándo/estación) **queda registrado en la plataforma**.
- Si se cae el WiFi, los escaneos **no se pierden** y se sincronizan al volver la red.
- El dispositivo vive en una **carcasa 3D** impresa por ellos.
- Funciona en una **prueba real** en una estación de inclusión.