
PoloPoints

Versión	Fecha	Autor	Descripción de cambios
1.00	23/04/2023	Pedro Parodi	Versión inicial
1.01	6/5/2023	Pedro Parodi	Agregado descanso entre chukkers
1.02	24/1/2024	Pedro Parodi	Actualización según APP y manual de armado
1.03	21/10/2024	Pedro Parodi	Nueva señal con bocinas de aire y panel de configuración en app.

Índice

1	MODO DE USO DEL TABLERO	3
1.1	DINÁMICA DEL JUEGO	3
1.2	CONSIDERACIONES PARA EL USO	4
1.2.1	<i>Configuraciones</i>	4
	Valor actual	4
	Panel de configuración	4
2	ARMADO Y MONTAJE DE LA ELECTRÓNICA.....	5
2.1	MATERIALES PARA EL ARMADO	5
2.2	INSTRUCTIVO.....	6
3	SOFTWARE (<i>FRONT-END</i>).....	9
3.1	WEB SERVER.....	9
3.2	DIAGRAMA DE FLUJO	10
	ANEXO I: PROGRAMAR ESP32.....	12

1 Modo de uso del tablero

La operación comienza con el encendido del cartel, conectándolo a una batería o fuente de 12V. Luego, el dispositivo ESP32 inicia un access point para conectarse a través de una interfaz WiFi y automáticamente se configuran los números del cartel a sus valores de inicio. Por lo general, estos valores son: puntajes en 0, chukker en 1, períodos de 7' 00" (con un agregado de 30" de tiempo extendido) e intervalo de descanso de 3'.

El usuario que administre el cartel deberá loguearse al access point con los siguientes datos:

- **SSID:** PoloPoints
- **PASSWORD:** 12345678

Y luego acceder al sitio web que le permite administrar el cartel:

- **DOMINIO:** <http://polopoints.local> (accesible a través de un navegador web)

La información del cartel físico y lo que muestre la aplicación deben ser idénticas en todo momento.

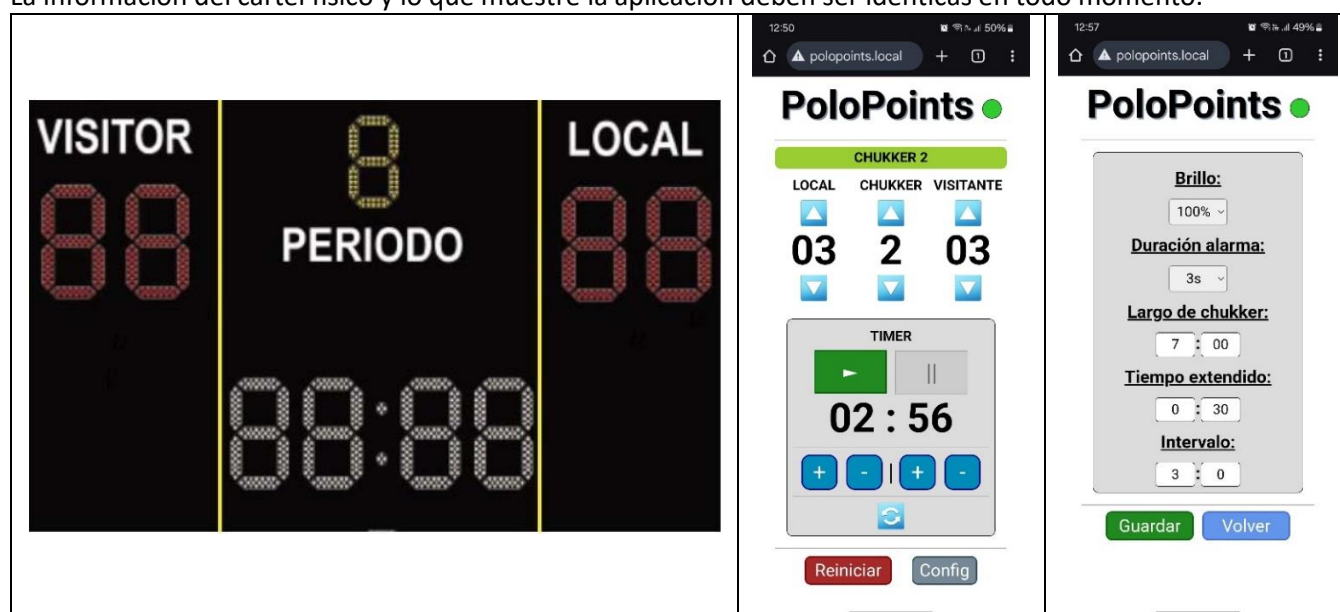


Imagen representativa del tablero y aplicación.

1.1 Dinámica del juego

El modo de juego es el siguiente: el usuario da comienzo al partido iniciando el temporizador desde la aplicación móvil. Durante el transcurso del *chukker*, iniciar/detener el temporizador o incrementar/decrementar puntajes estará a cargo del usuario.



Finalizado los 7' del *chukker*, sonará una bocina y el juego continua con 30" adicionales de tiempo extendido (total 7' 30"). Transcurrido este tiempo, sonará nuevamente la bocina y este evento da por finalizado el *chukker* actual y se pasa a un intervalo de descanso de 3'.


Luego del intervalo, el número de *chukker* se incrementa automáticamente y habilitará que el juego continúe y repetir así el ciclo del juego.

El juego finaliza cuando el usuario lo desee, no es necesario prefijar una cantidad de *chukkers* desde la aplicación.

1.2 Consideraciones para el uso

Si bien la aplicación administra el tablero con poca intervención del usuario, es posible en todo momento iniciar/detener/modificar el reloj, modificar puntajes o el número del *chukker*.

Las flechas  y  sirven para administrar los valores de puntaje local, puntaje visitante y *chukker*. Los signos y incrementan/disminuyen los minutos o segundos del temporizador actual, el cual podría corresponderse al *chukker*, tiempo extendido o intervalo de descanso.

El botón  reinicia el *chukker* actual, es decir, lleva el reloj al valor 07' 00". Independientemente si el juego se encuentra en intervalo de descanso o tiempo extendido. Las modificaciones realizadas a los puntajes o al número de *chukker* no se ven afectadas, conservando las operaciones realizadas por el usuario.

El botón **Reiniciar** lleva el tablero a sus valores de inicio. Salvo modificaciones realizadas por el usuario, esto corresponde a puntajes en cero, *chukker* en 1 y reloj en 07' 00". Con 30" de tiempo extendido y 3' de intervalo de descanso.

La aplicación cuenta con círculo que se colorea en verde  o rojo  en función del estado de la conectividad WiFi con el tablero. El rango de cobertura es de aproximadamente 100mts.

El usuario cuenta con un panel de configuración, accesible a través del botón **Config**. Allí podrá modificar la intensidad lumínica del cartel del 10% al 100%, como así también la duración de la señal sonora o bien modificar los valores *default* de temporizadores.

1.2.1 Configuraciones

Valor actual de temporizador

Es posible modificar el valor actual del reloj y de forma no permanente. El temporizador que se modificará es el que se encuentra activo, que podría corresponderse al *chukker* en proceso, el de tiempo extendido o bien del intervalo de descanso. Esto dependerá justamente del estado en el que se encuentre el juego, fácilmente identificable por la etiqueta en pantalla de la aplicación.



Etiquetas de la aplicación según el estado del juego.

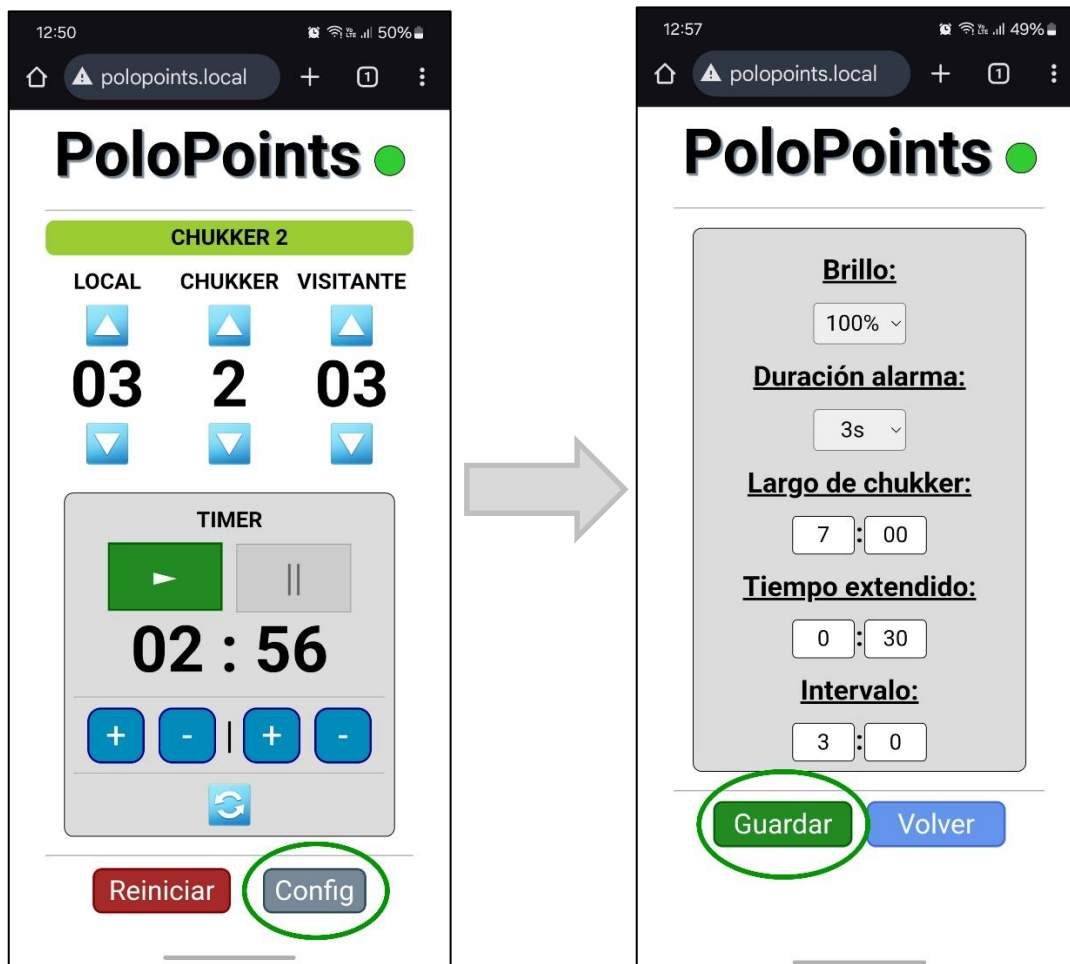
Para ello, simplemente se debe frenar el reloj (en caso de que esté activo) y mediante los botones y ajustarlo al valor deseado tanto en minutos como en segundos. Al iniciar nuevamente el reloj, comenzará desde el valor configurado.

Esto es de utilidad, por ejemplo, si se quisiera agregar 20 segundos al *chukker* actual por demoras en el juego. O bien para agregar un minuto más de descanso entre *chukkers*.

Panel de configuración

La aplicación cuenta con un panel de configuración que permite modificar el brillo de los LEDs entre un 10% y 100%, configurar la duración de la bocina de 1 a 5 segundos o bien modificar el valor *default* de cualquiera de los temporizadores (*chukker*, tiempo extendido e intervalo de descanso). Modificar los temporizadores mediante este método implica que la modificación es permanente.

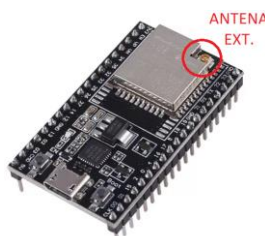
Para acceder al panel de configuración, simplemente presionar el botón **Config**. Luego, realizar los cambios necesarios y guardar los cambios.



2 Armado y montaje de la electrónica

2.1 Materiales para el armado

- ESP32-DevKitC con módulo ESP32-WROOM-32UE (versión con antena externa).



ESP32-based kit con antena externa.

- Antena WiFi 2.4GHz 12dbi.¹

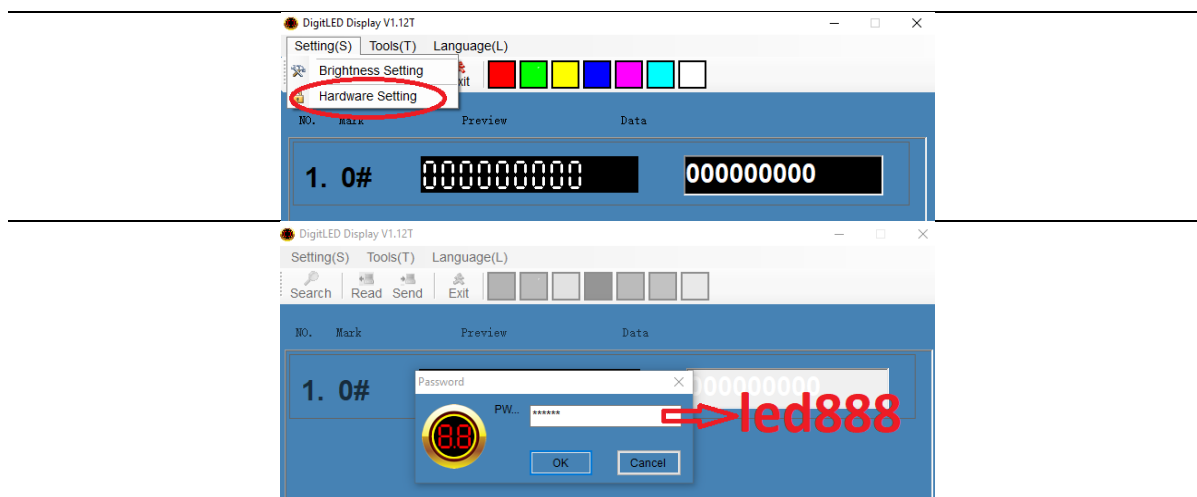
¹ Puede reemplazarse por antena más económica dado que de todas formas no se llega a cubrir el total de la cancha.

- Conector pigtail ufl – SMA female (conexión antena-ESP32).
- Kit del tablero.
 - Segmentos LEDs.
 - Placa controladora general.
 - 1 placa de comando por cada dígito (los puntos no cuentan con placa de comando propia).
- Convertidor DC-DC 12V a 5V.
 - Step Down DC-DC **XL7015**: último módulo probado de ML, pero hay alternativas.
- *Transceiver* RS232-UART.
 - Probados módulos comerciales con IC MAX3232.
- Para señal sonora¹:
 - 2 bocinas de aire plásticas tipo camión, son su respectivo compresor de 12V_{CC}.
 - Relay 5V – 10A.
 - ~~Amp Audio 60W Mono Clase D TPA3118: último módulo probado de ML, pero hay alternativas.~~
 - ~~2 parlantes plásticos de 4". Modelo probado: 4" 15W_{RMS} (máx. 50W) 4Ω.~~
- Gabinete del producto: una caja plástica que permita instalar los componentes electrónicos en el interior.
- Materiales y herramientas para armado y soldado.

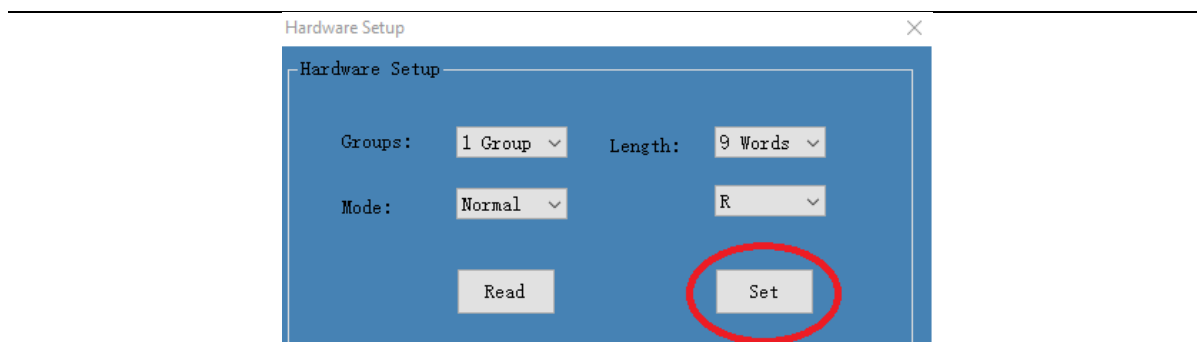
2.2 Instructivo

Los pasos para el armado del tablero son:

1. Programar ESP32 (ver [ANEXO I: PROGRAMAR ESP32](#)).
2. Programar placa controladora (circuito LEDs).
 - a. Utilizar la herramienta provista por fabricante **DigitLED-Display**.
 - b. Utilizar cable RS232-USB provisto por fabricante y conectar la placa controladora de LEDs a una PC con el software del fabricante.
 - c. Programar 1 Grupo, 9 Words, modo normal, LEDs tipo R (no RGB) y brillo al 100% (el brillo luego podrá modificarse desde la app).



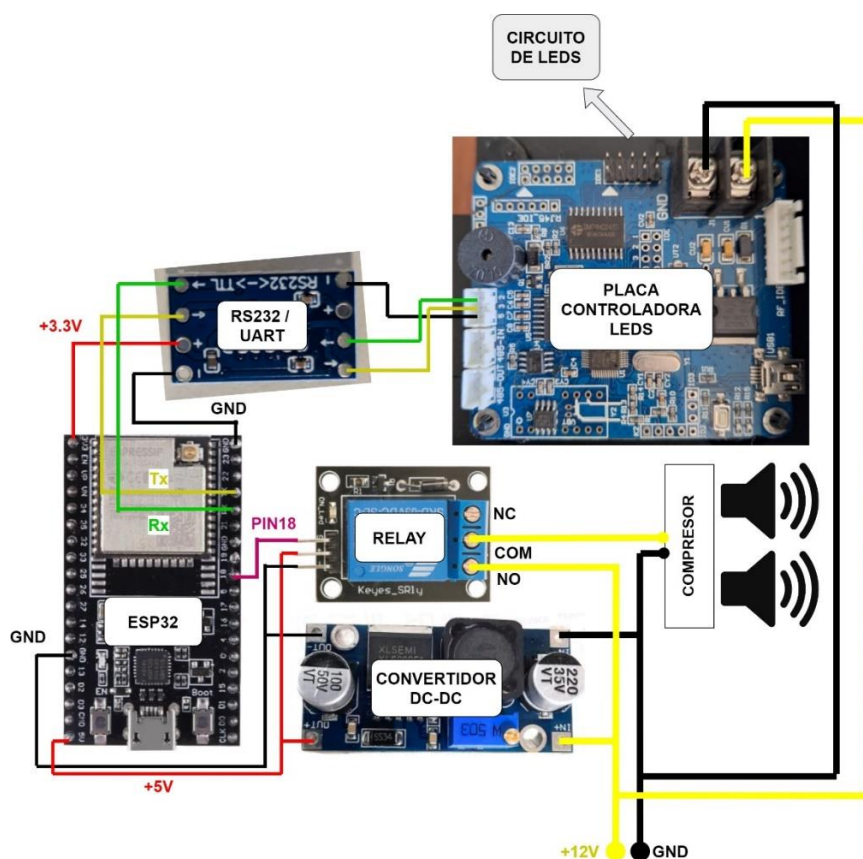
¹ Se reemplaza la señal sonora con amplificador de audio por bocinas de aire con compresor a 12V, pero aún podría utilizarse este método.



3. Setear el nivel de tensión de convertidor DC-DC a 5V.
 - a. Alimentar con 12V la entrada.
 - b. Medir con multímetro a la salida.
 - c. Configurar el preset para salida de 5V_{DC}.
4. Armado de cables.
 - a. Especial cuidado en caso de utilizar el cable serial RS232 que provee el fabricante de tablero de LEDs. Al menos en primeros modelos, el cable rojo se correspondía con línea de masa o GND.
5. Ensamble.

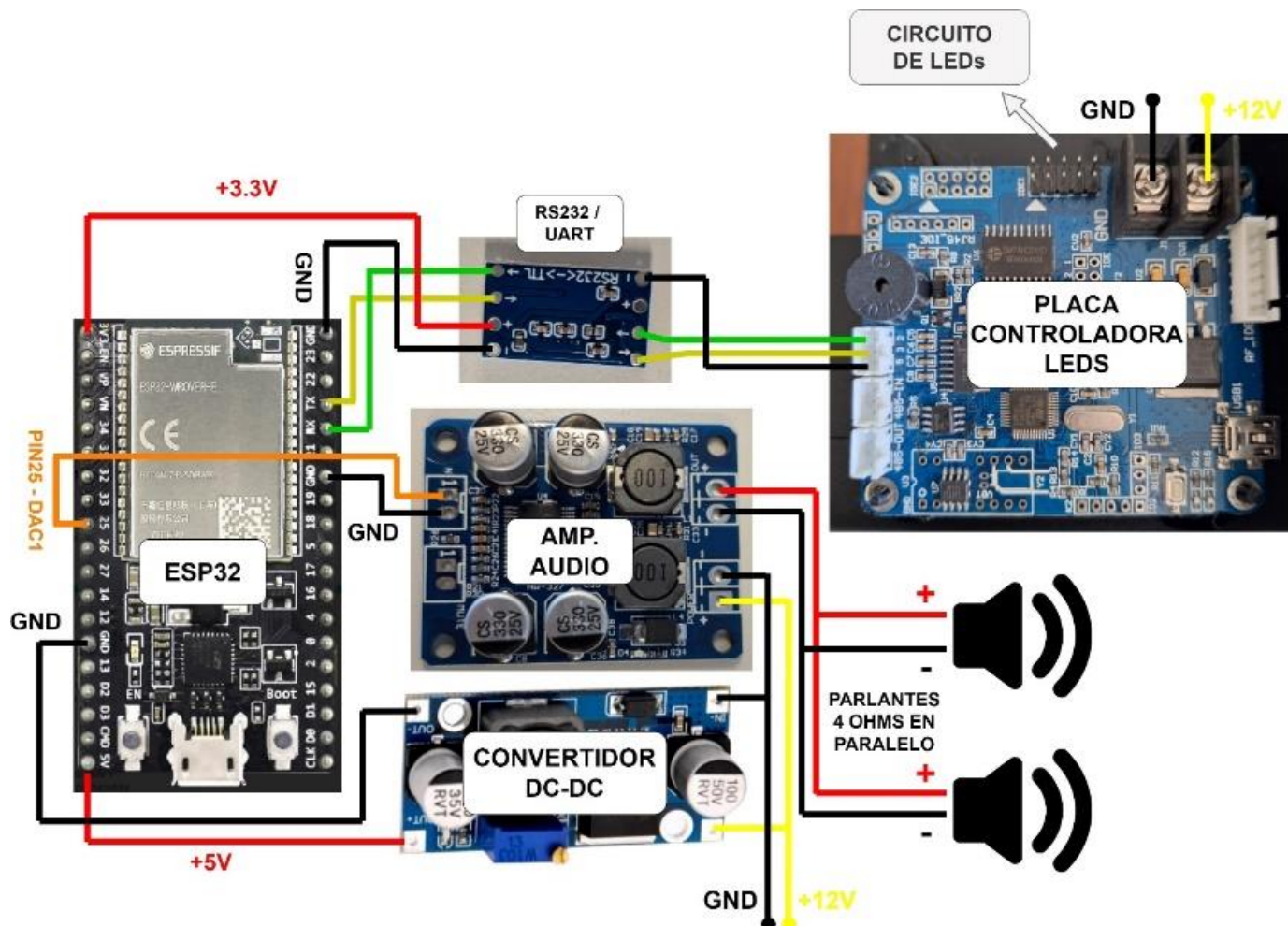
El ensamble y conexionado debe resultar de la siguiente forma, según el tipo de señal sonora:

BOCINAS DE AIRE CON COMPRESOR



Señal sonora: bocinas de aire con compresor. Diagrama de conexión.

PARLANTES CON AMPLIFICADOR DE AUDIO



Señal sonora: parlantes con amplificador de audio. Diagrama de conexión.

Consideraciones:

- El circuito de LEDs debe conectarse en el siguiente orden, partiendo de la placa controladora y respetando la orientación que marcan las flechas del conector de cada placa y conector:
 1. PUNTAJE VISITANTE UNIDAD
 2. PUNTAJE VISITANTE DECENA
 3. *TIMER* SEGUNDOS UNIDAD
 4. *TIMER* SEGUNDOS DECENA (incluir 1er punto del *timer*)
 5. *CHUKKER*
 6. *TIMER* MINUTOS UNIDAD
 7. *TIMER* MINUTOS DECENA (incluir 2do punto del *timer*)
 8. PUNTAJE LOCAL UNIDAD
 9. PUNTAJE LOCAL DECENA

3 Software (front-end)

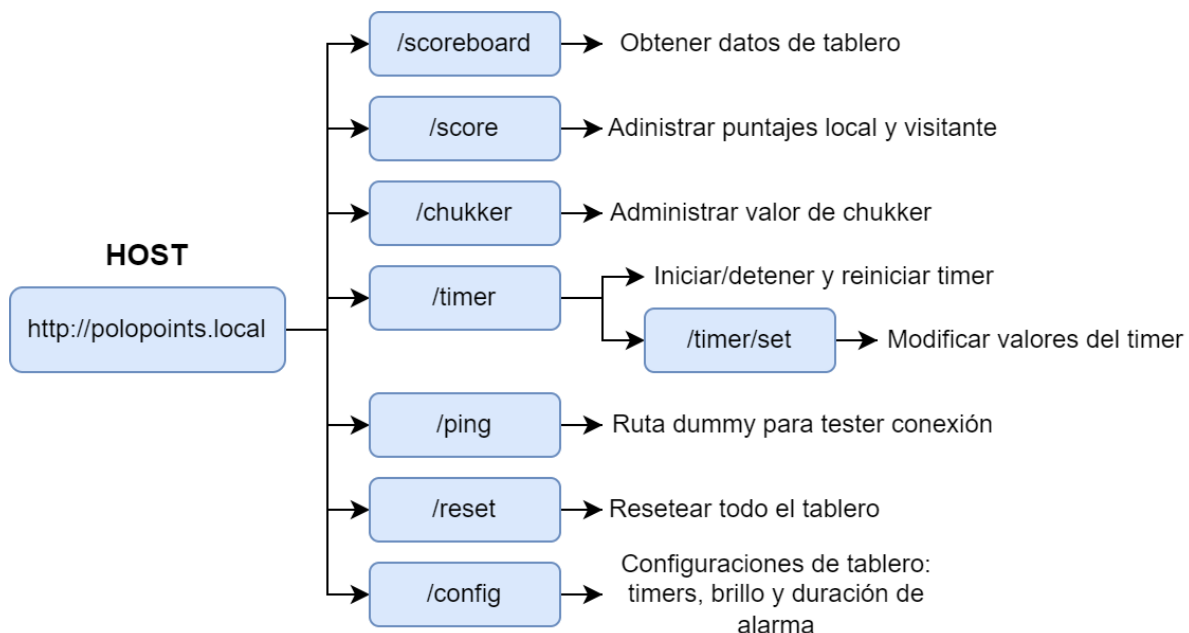
3.1 Web Server

El dispositivo ESP32 genera una red WiFi y funciona como access point. Los datos de la red son los siguientes:

- SSID: "Polo Points"
- Password: 12345678
- DOMINIO: polopoints.local

Se implementa un servidor con una serie de *endpoints* para administrar las variables y funcionalidad del tablero. Para ello, se hace uso de los siguientes comandos disponibles, que serán parámetros parte de la URL en las consultas al servidor:

- INC_SCORE_LOCAL = 1 → incrementar puntaje del local
- INC_SCORE_VISITOR = 2 → incrementar puntaje del visitante
- DEC_SCORE_LOCAL = 3 → decrementar puntaje del local
- DEC_SCORE_VISITOR = 4 → decrementar puntaje del visitante
- INC_CHUKKER = 5 → incrementar chukker (período)
- DEC_CHUKKER = 6 → decrementar chukker (período)
- START_TIMER = 7 → iniciar el temporizador
- STOP_TIMER = 8 → detener el temporizador
- RESET_TIMER = 9 → resetear el temporizador a valor de inicio (*default*)
- SET_CURRENT_TIMER = 10 → modificar el valor actual del temporizador
- SET_DEFAULT_TIMER = 11 → modificar duración de *chukker* del temporizador
- SET_EXTENDED_TIMER = 12 → modificar duración de tiempo extendido
- SET_HALFTIME_TIMER = 13 → modificar duración de intervalo de descanso
- RESET_ALL = 14 → resetear el tablero a valores de inicio (*default*)
- SET_CONFIG = 15 → grabar configuraciones de brillo, señal sonora y *timers*
- GET_CONFIG = 16 → consultar configuraciones de brillo, señal sonora y *timers*



Endpoints implementados en la aplicación.

Todas las solicitudes programadas en el servidor son de tipo GET. Algunas de ellas solo responden con un STATUS_CODE en función de éxito o fracaso de la acción deseada.

Otras, en caso de éxito, responden con los valores actualizados del tablero o la configuración. El dispositivo ESP32 envía los datos como una variable de tipo *string* que concatena todos ellos, con el siguiente formato y orden:

DATOS DE TABLERO:

- Puntaje local : [0;99]
- Puntaje visitante : [0;99]
- Chukker : [0;9]
- Reloj (minutos) : [0;59]
- Reloj (segundos) : [0;99]
- Estado del timer : [0;1]
 - Frenado → 0
 - Corriendo → 1
- Estado de la partida : [0;2]
 - En progreso → 0
 - Tiempo extendido → 1
 - Intervalo → 2

Ejemplo:

```
Content-type: "text/plain"
Body: { "1,3,2,7,0,0,0" }
```

DATOS DE CONFIGURACIÓN:

- Brillo : [1;10] (10% - 100%)
- Largo de señal sonora en segundos : [1;5]
- Minutos de la duración del *chukker* : [0;59]
- Segundos de la duración del *chukker* : [0;59]
- Minutos de la duración del tiempo extendido : [0;59]
- Segundos de la duración del *chukker* : [0;59]
- Minutos de la duración del *chukker* : [0;59]
- Segundos de la duración del *chukker* : [0;59]

Ejemplo:

```
Content-type: "text/plain"
Body: { "10,1,7,30,0,30,3,0" }
```

3.2 Diagrama de flujo

La aplicación en el dispositivo ESP32 puede dividirse en dos bloques sencillos: servidor web y máquina de estados que administra el tablero.

El servidor web es de tipo asincrónico, por lo que corre de fondo como un hilo independiente y administra y ejecuta todas las solicitudes que recibe el *host* (ESP32). Las solicitudes HTTP permiten actuar sobre los valores y estado del tablero. Cada vez que hay un cambio en algún dato (por ejemplo, un puntaje) a través de una solicitud, la misma solicitud informa mediante un *flag* a la máquina de estados principal de la aplicación para denotar que los LEDs del tablero deben actualizarse.

Por otro lado, la máquina de estados alterna entre tres estados:

- INIT: inicialización de tablero con valores iniciales. Se ejecuta por única vez al comienzo del programa.
- IDLE: estado de reposo. Queda a la espera de que pase 1 segundo (con *timer* activo) o que se reciba alguna solicitud HTTP que modifique algún valor del tablero. Luego invoca al siguiente estado para actualizar el tablero.
- REFRESH_SCOREBOARD: actualizar tablero (LEDs). Luego, retorna a estado de reposo IDLE.

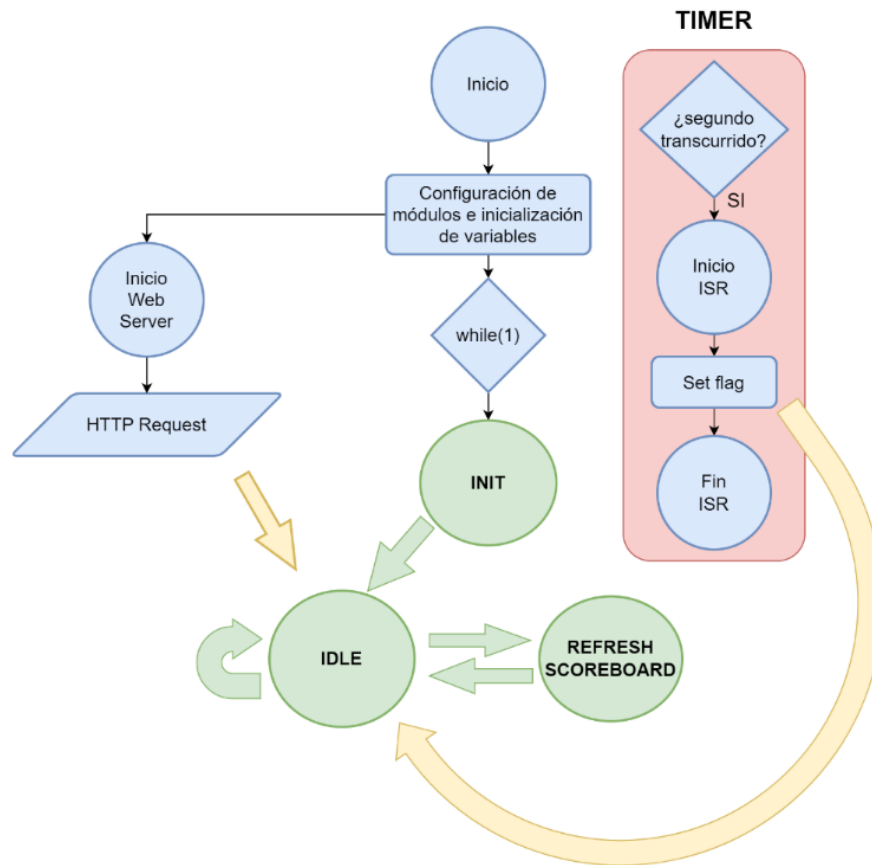


Diagrama de flujo del programa.

ANEXO I: PROGRAMAR ESP32

Para programar el ESP32 sin usar un IDE como Visual Studio Code o Arduino IDE, descargar la herramienta **Flash Download Tools** de Expressif. Para este documento, se utilizó la versión v3.9.5.

Link: <https://www.espressif.com/en/support/download/other-tools>

Flash Download Tools

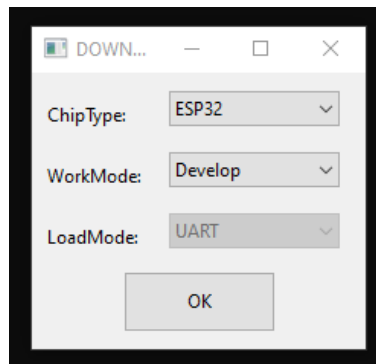
<input type="checkbox"/>	Title	Platform	Version	Release Date ▼	Download
<input type="checkbox"/> +	Flash Download Tools	Windows PC	V3.9.5	2023.06.12	

Download selected

Descomprimir la herramienta **Flash Download Tools** y en la carpeta **bin** copiar los archivos binarios necesarios:

- bootloader.bin
- partitions.bin
- firmware.bin
- spiiffs.bin

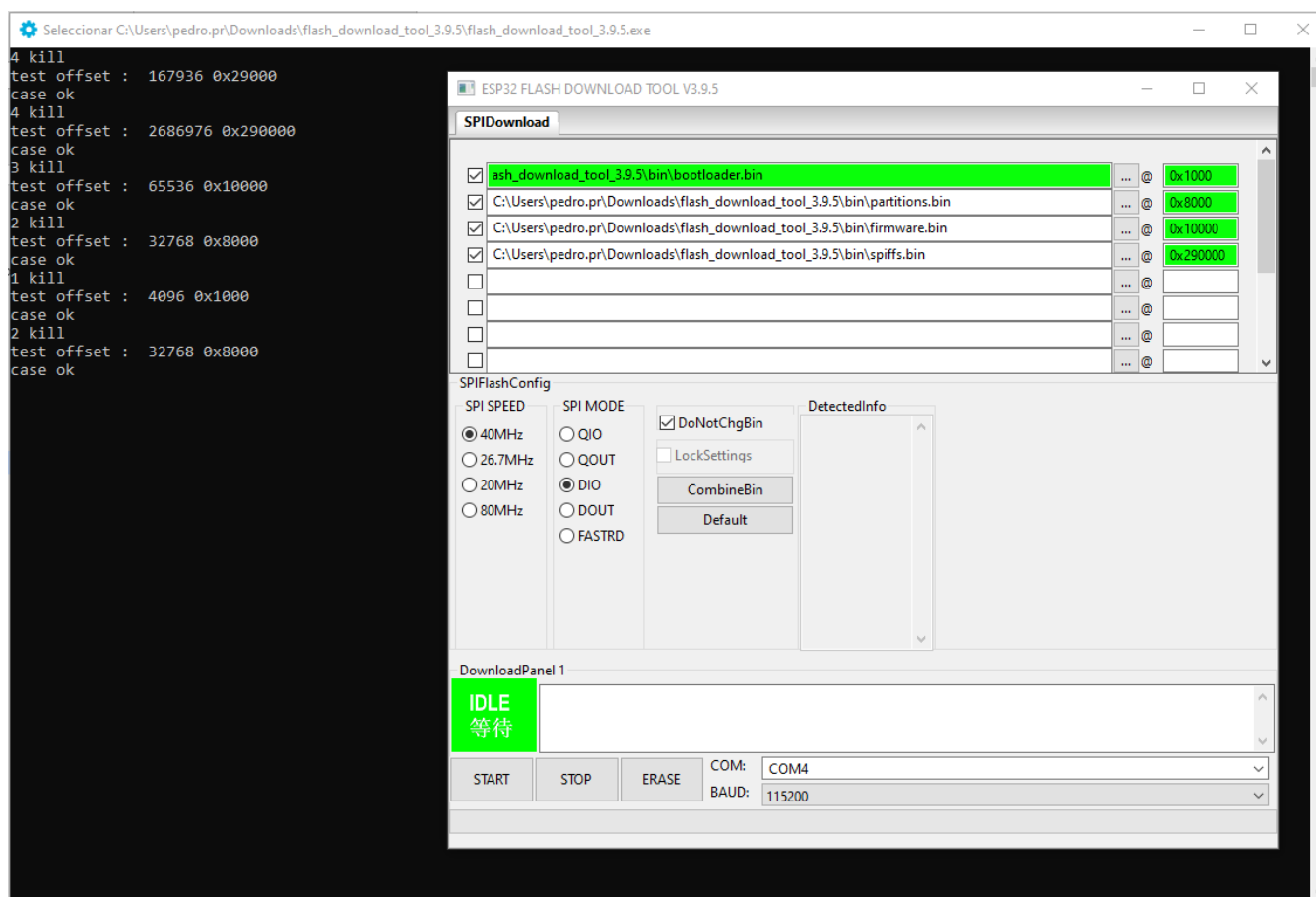
Abrir la herramienta desde su ejecutable **.exe** y especificar **ChipType → ESP32**.



Dar **OK** y en la siguiente ventana se debe cargar cada archivo binario a grabar en el dispositivo ESP32 y asignar la dirección de memoria para el grabado de cada uno de ellos:

- bootloader.bin → 0x1000
- partitions.bin → 0x8000
- firmware.bin → 0x10000
- spiiffs.bin → 0x290000

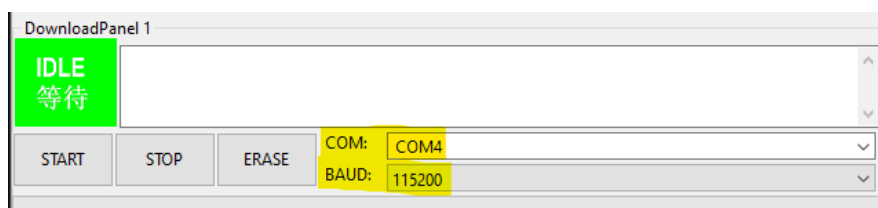
Respetar el orden de los archivos y no olvidar dar *click* al *checkbox* de cada línea.



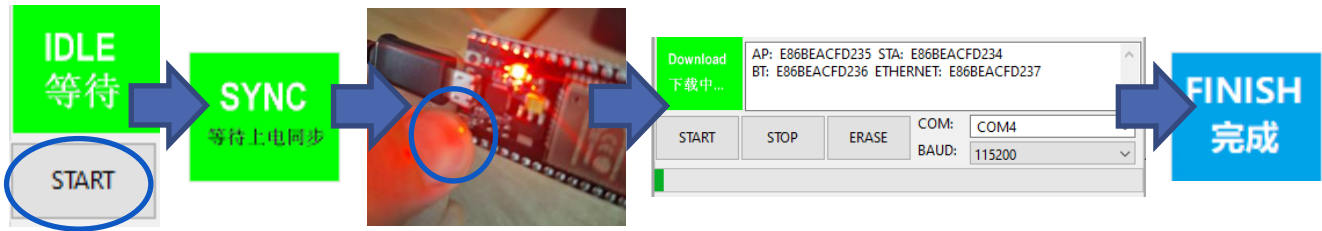
A medida que se cargan los archivos y se establece la dirección de memoria de inicio para el grabado, la terminal de la herramienta **Flash Download Tool** mostrará notificaciones de estado y la línea se coloreará con verde.

En este punto ya se encuentra todo listo para grabar la aplicación en el ESP32.

Conectar el ESP32 a la PC (si no se hizo previamente) y establecer el canal de comunicación **COM**, para este ejemplo se uso **COM4**. Dejar el Baudrate en valor default de 115200 bps.



Dar al botón **START** de la herramienta Flash Download Tool y mantener presionado el botón **BOOT** del dispositivo ESP32 hasta que ver el mensaje **Download** en la herramienta y observar una barra de carga.



Al terminar, la terminal además mostrará un mensaje “is stub and send flash finish”.

El dispositivo ESP32 ya se encuentra programado y listo para su uso. En caso de precisar corroborar el correcto grabado de la aplicación previo a su montaje, reiniciar el módulo mediante el boton de **RESET** y probar las funcionalidad de la aplicación.