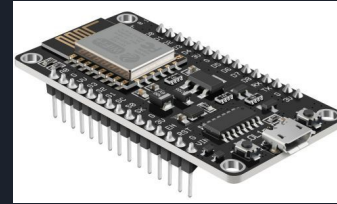
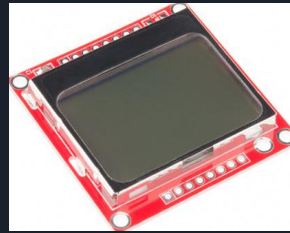


Display de disponibilidad integrado a calendario



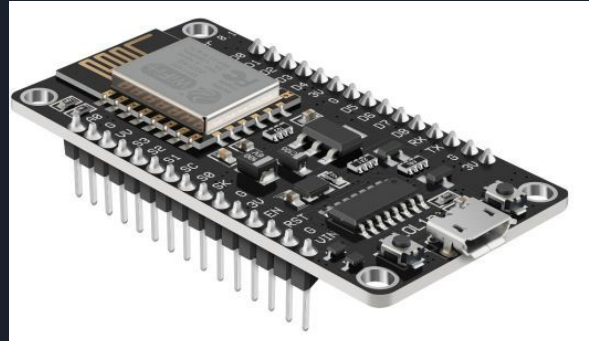
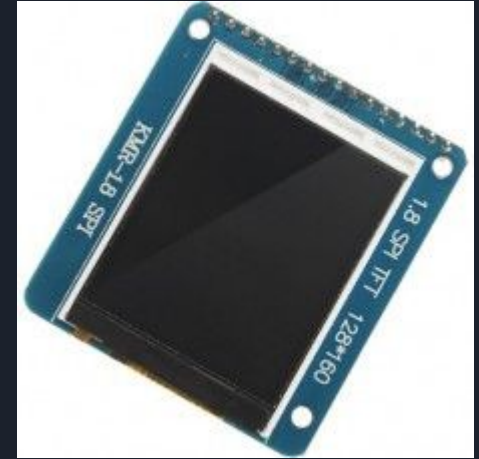
Christabel Alvarado Anchía
Erick Marín Rojas

Caso de uso

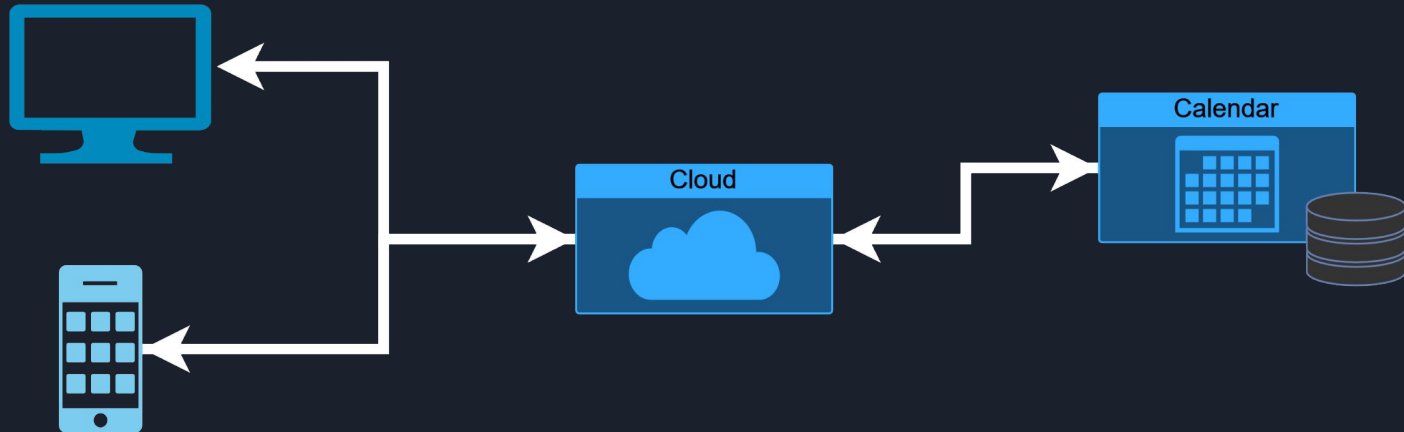


Componentes

- NodeMCU(ESP8266)
- ST7735 TFT Display
- Botones



Infraestructura

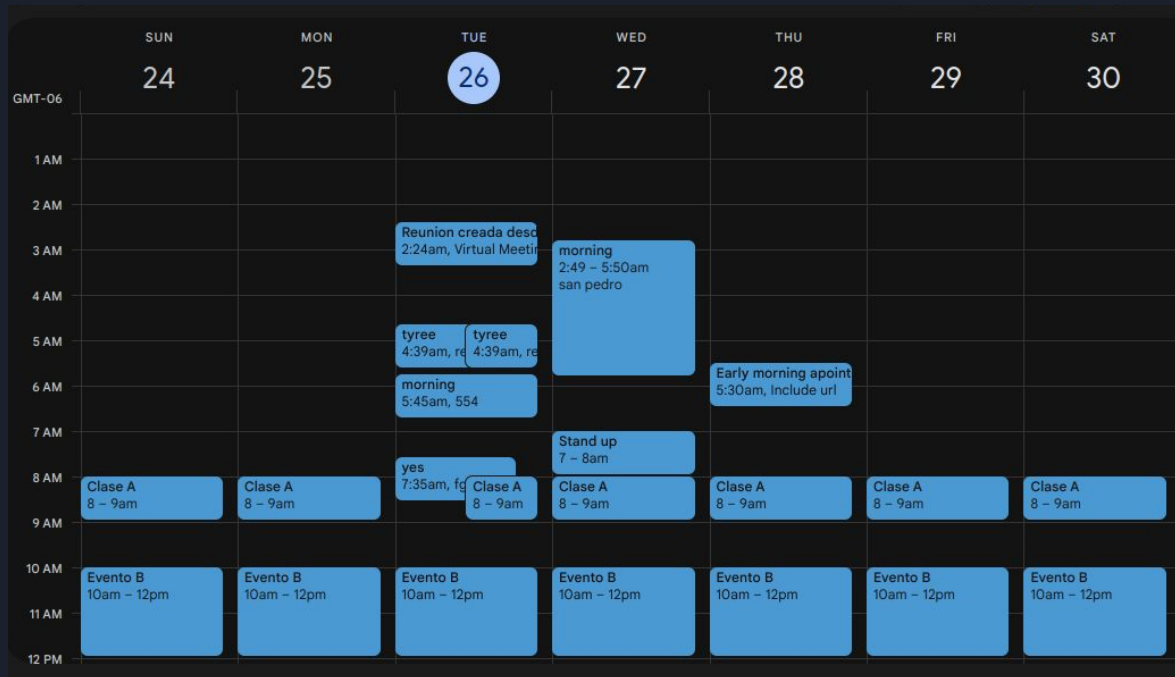


Obtener data con API de Google

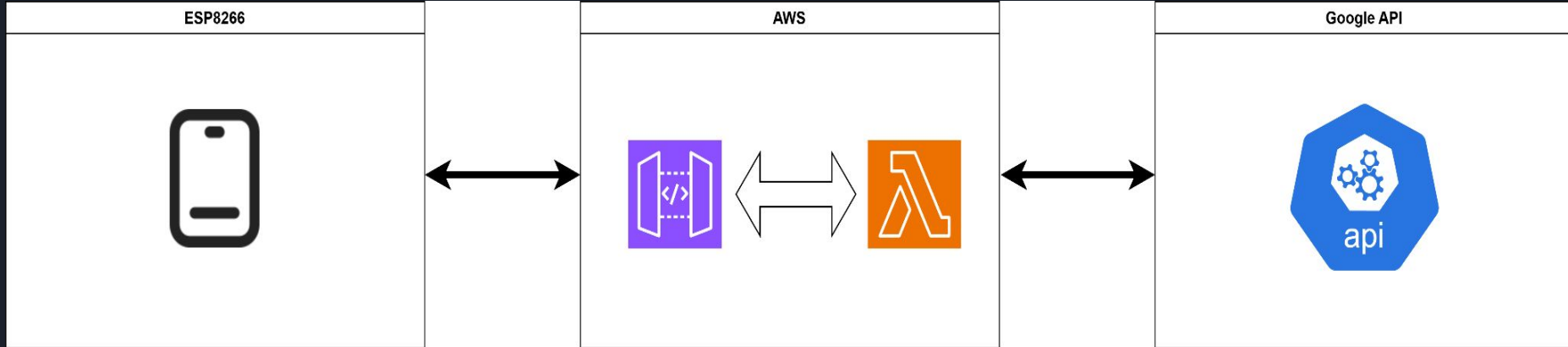
Google Calendar

Google Cloud

OAuth



Servidor remoto





Traer datos del endpoint a ESP8266

```
HTTPClient https;

Serial.println("Solicitando datos...");
tft.println("Solicitando datos...");
if (crearNuevo){
  https.begin(client, serverUrlnew);
}else{
  https.begin(client, serverUrl);
}
int httpResponseCode = https.GET();
```

ST7735 TFT Display

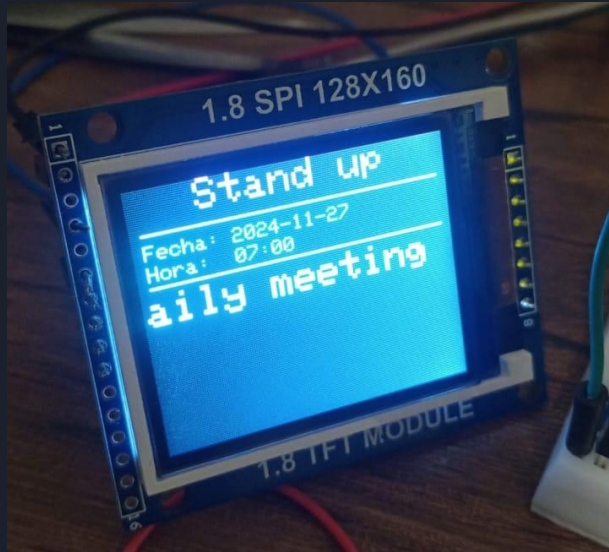
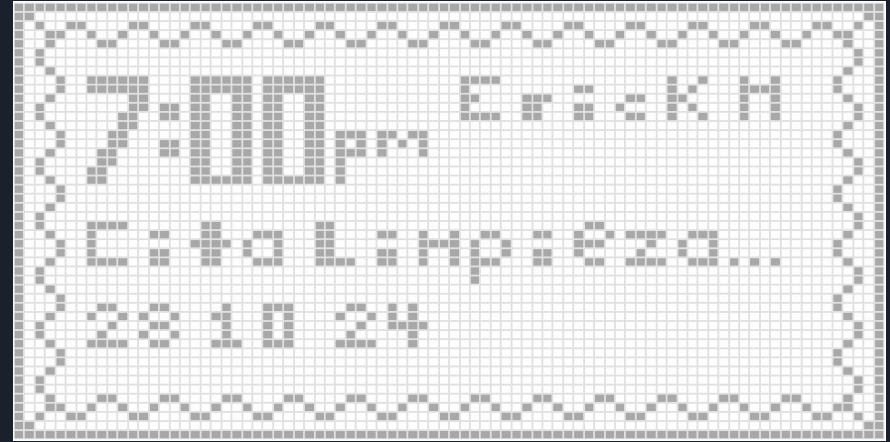
De la propuesta original usando una pantalla de monocrómica de poca resolución...



- 1 - Resolución de 128x160 píxeles
- 2 - Profundidad de color de hasta 18 bit por píxel.
- 3 - Comunicación SPI
- 4 - Tiene un controlador que permite realizar comandos simples para dibujar líneas
- 5 - Bajo framerate

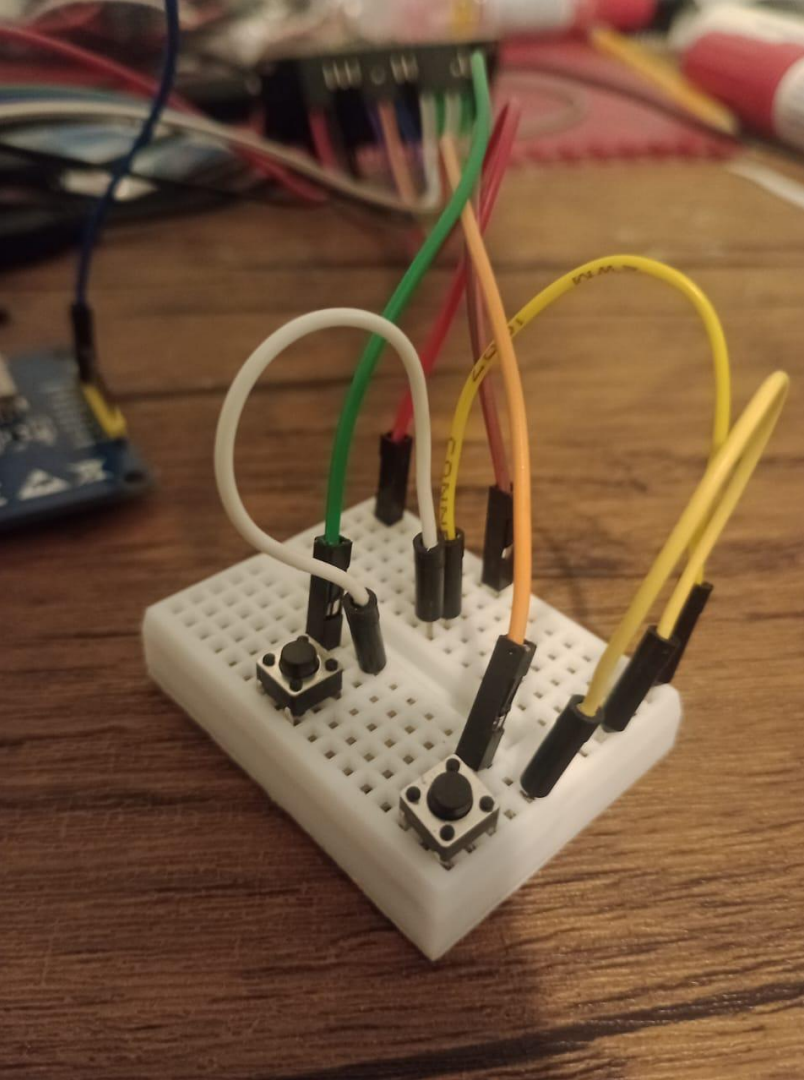
Display en pantalla

Nos distanciamos de nuestro diseño original...



Para producir un diseño que fuera funcional y estético, sesgamos la mayoría de la pantalla para contener la descripción.

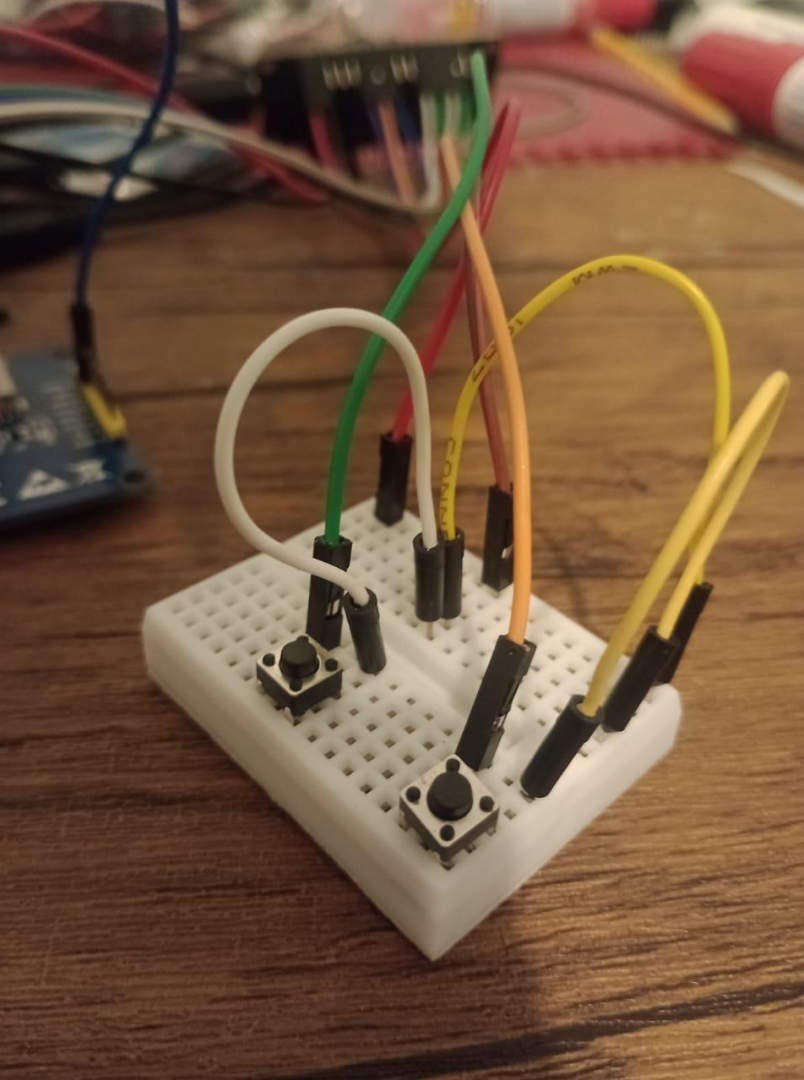
Decidimos decorar con elementos tan simples como pudimos afán de ahorrar espacio, como líneas y colores.



Botones y Gestos

Para presentarle al usuario una interfaz simple usamos la menor cantidad de botones que cómodamente pueden contener los gestos que deseamos tener.

Donde los dos gestos más simples, presionar un botón u el otro sirve para avanzar o devolverse en el calendario.



Botones y Gestos

Ya que el calendario puede contener una gran cantidad de eventos para un día o una semana dada mantener presionado el botón izquierdo y liberarlo nos retorna al evento actual.

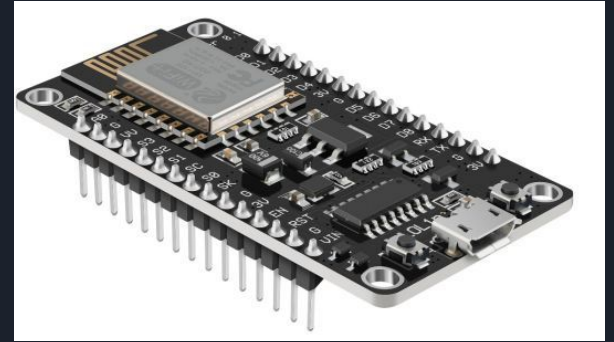
Mantener presionado el botón derecho nos permite crear un evento al llamar un endpoint que usa fastAPI para producir un evento en la hora actual si no encuentra colisiones..

NodeMCU(ESP8266)

Tiene capacidades de WiFi salido de la caja.

Emplea un microcontrolador Tensilica Xtensa L106 de 32 bits con frecuencia de hasta 160 MHz

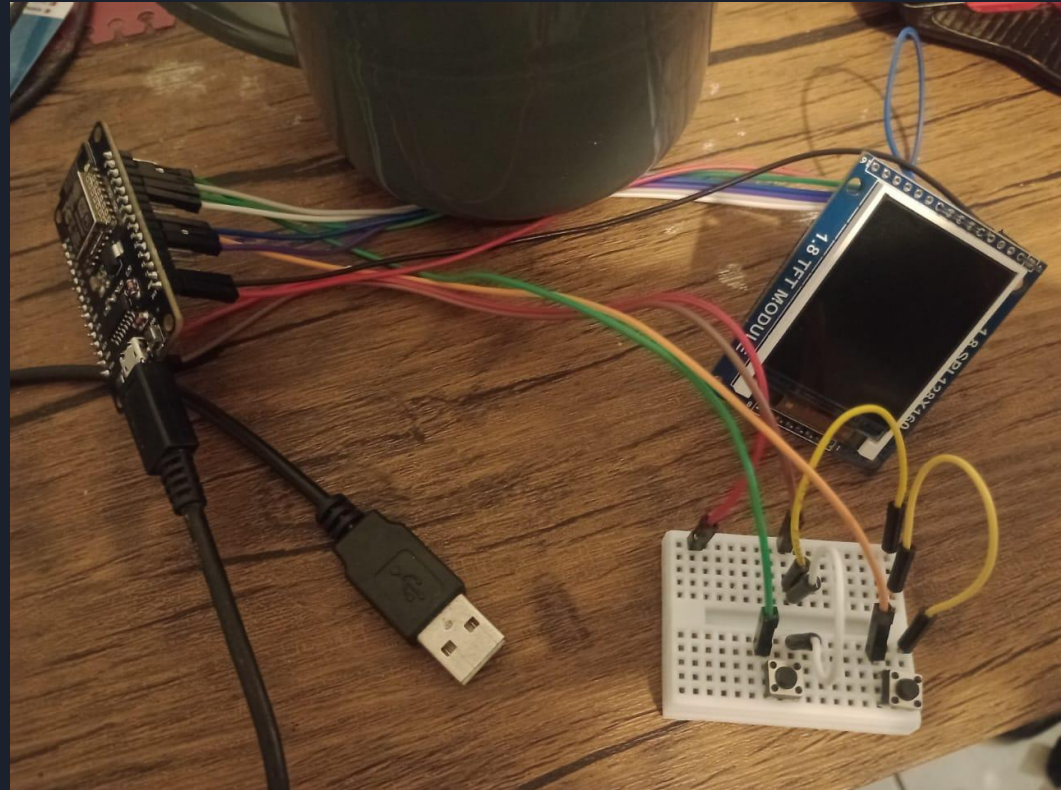
Opera a 3,3V y es compatible con distintos protocolos de comunicación.



Circuito Completo

A nivel de hardware la implementación es relativamente simple.

Para minimizar los errores y ya que cortocircuitamos el jumper de regulación de tensión (pues el board opera a 3,3V), conectamos la pantalla directamente al Node MCU



Crear reunión nueva

Fast API

Sencillo

Campos mandatorios

Nueva reunión

Fecha y hora de inicio:

Fecha y hora de finalización:

Resumen:

Descripción:

Location:

Asistentes (correos separados por comas):

Create Meeting



Retos al desarrollar el proyecto

Google Cloud

ESP-01

Botones

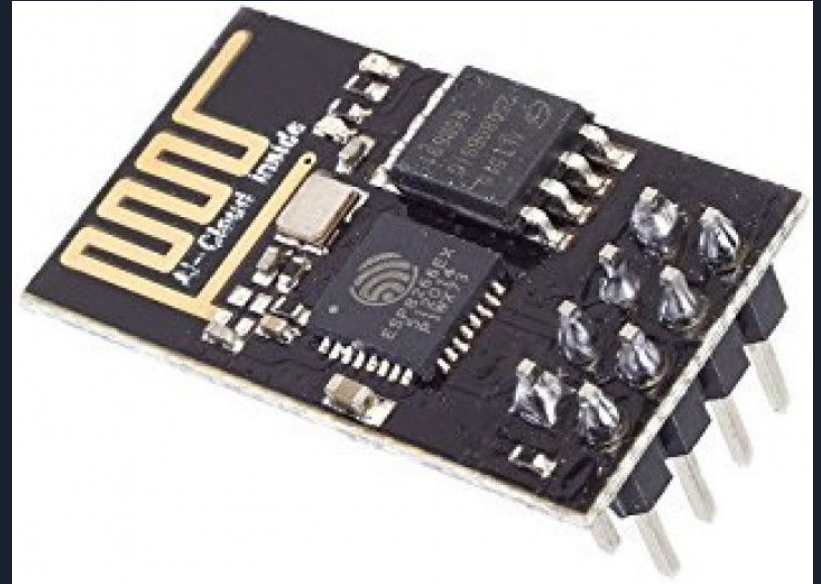


ESP-01

Memoria limitada

Problemas de energía

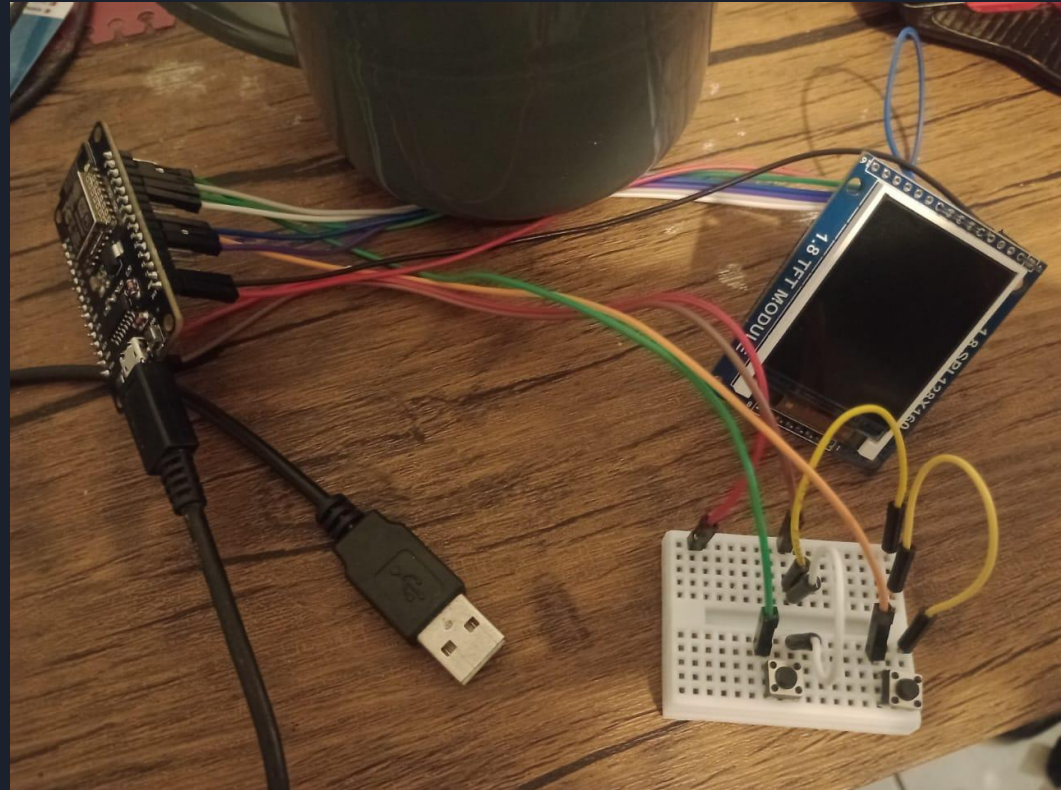
Menor soporte



Limitaciones

Al implementar un botón distinto por cada acción se encontraron limitaciones respecto los valores de los pines durante la inicialización y la operación. Donde pines como D8 generaban problemas al no iniciar en tierra y pines como D0 no permiten pullup interno.

Aunque la velocidad a la que el controlador se comunica con la nube es relativamente rápida, la espera de aproximadamente un segundo para rellenar la nueva ventana de eventos contenida en el dispositivo puede ser molesta.

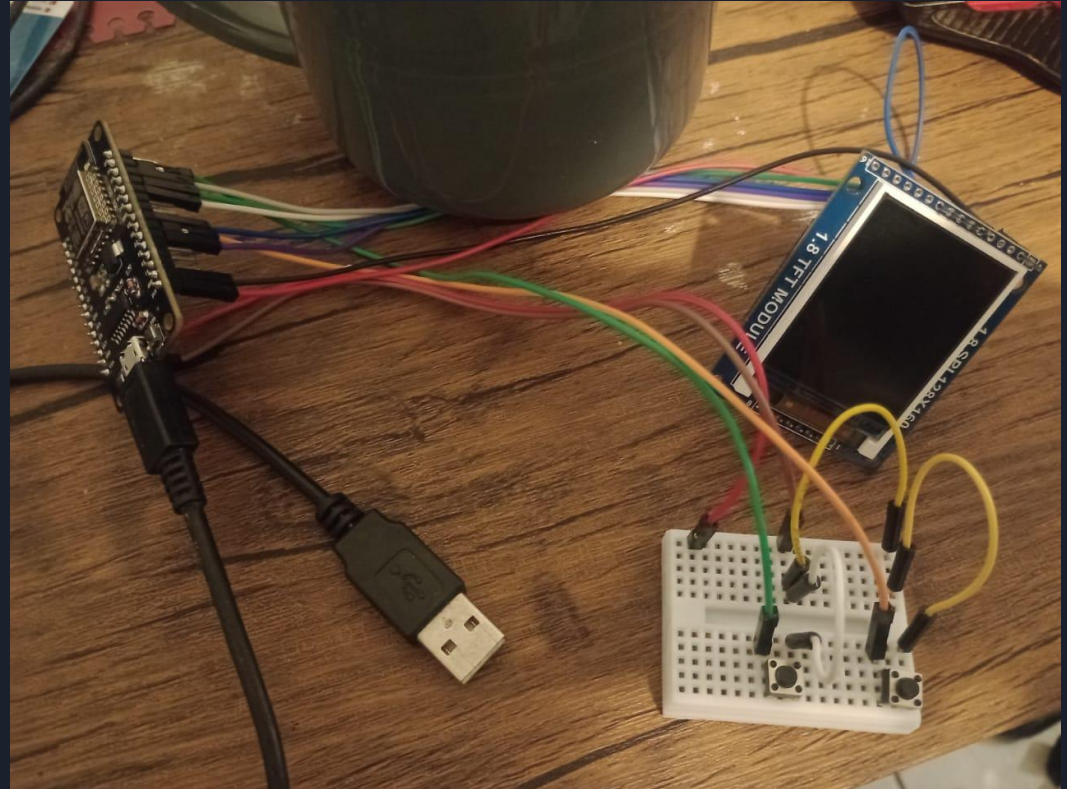


Limitaciones

Sin una carcasa para el proyecto la pantalla es sensible a movimientos en los cables y cae en error (pantalla blanca).

Reiniciar muchas veces el sistema provoca que el servicio del protocolo de tiempo de red (NTP) rechace el dispositivo.

Para diseños enfatizados en el poco consumo energético es necesario implementar las funciones de Deep Sleep del Node MCU para mostrar la información solo cuando sea necesario.





Conclusiones

Es riesgoso implementar un sistema compuesto de subsistemas poco robustamente conectados, como tenemos con un arduino que requiera comunicarse a través de un módulo, por estas razones es mejor optar por subsistemas que integran nativamente los componentes más sensibles del sistema como total.

Los sistemas diseñados para brindar soluciones con IoT aligeran la carga puesta sobre los dispositivos en el borde, levantando los requerimientos en memoria para poder sostener un contenido en apariencia arbitrariamente grande.



Conclusiones

No siempre el precio es un indicador directo de la calidad de un producto o de su rendimiento en una implementación dada, pues el proyecto emplea ahora un controlador que cuesta menos que el Arduino propuesto originalmente y que soluciona el nexo entre el controlador y la conexión Wi-Fi.

La poca disponibilidad de dispositivos como el controlador por el cual se optó hubiese encarecido el proyecto gravemente, al tener que elegir equipo sobredimensionado que vendría de la mano con el rediseño del alcance del proyecto o un desperdicio.