

Desarrollado por: Jacobo Rave Londoño  
Juan Sebastián Guerra H.

Codigo:202010081010  
202010097010

## **Parcial 2 - Teoría de la Conmutación - Grupo 004**

### **Código: <https://bit.ly/2ZSQrvB>**

### **Descripción de la idea a desarrollar**

El objetivo principal del proyecto es mandar un mensaje de texto por teclado 4x4 usando el sistema t9 a un servidor web montado en la plataforma esp32, siendo en este mismo servidor la visualización. Por otro lado, tenemos el mensaje de respuesta que es introducido como un input y mandado al esp32 para mostrarlo en la pantalla lcd.

### **Descripción de la solución**

#### **Módulo de hardware**

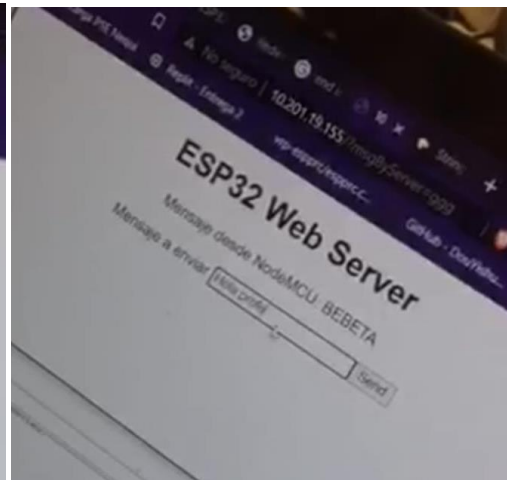
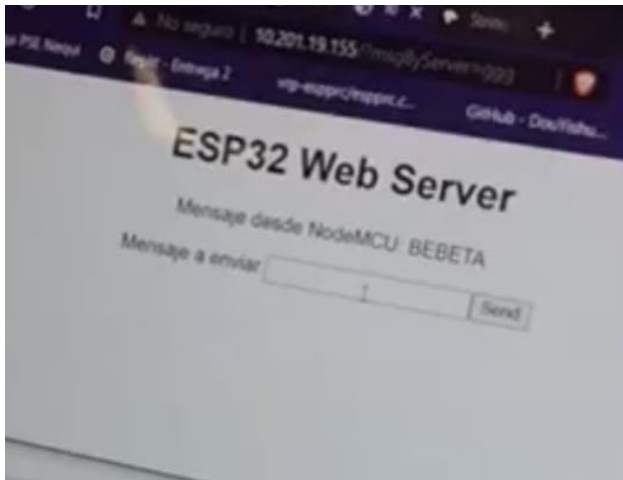
El módulo lógico para hardware elegido fue el NodeMCU ESP32, las instrucciones se codificaron en el IDE y lenguaje de Arduino. El NodeMCU está conectado con un keypad 4x4 que sirve de entrada para el mensaje, este está configurado en el código con una configuración t9, esta misma es la que usaban los celulares antiguos, consta de un abecedario en el teclado numérico y con este se pueden mandar mensajes vía sms. Se usó la librería *<Keypad.h>* que se encarga de leer las filas y columnas para saber cuál botón fue presionado y manda un uno en la coordenada pulsada, ya con este dato el algoritmo determina por el numero de pulsaciones a cual letra corresponde. Así se van concatenando letras para formar frases que se van enviando al servidor.

También el NodeMcu está conectado a una pantalla lcd que visualiza el mensaje mandado por el usuario en el servidor web, el servidor guarda los mensajes en variables que después son mostradas en la pantalla. La librería usada *<LiquidCrystal.h>* permite la comunicación con pantallas alfanuméricas de cristal líquido (LCD) con la placa NodeMC.

Los mensajes son mandados a un servidor web cada medio segundo así que va actualizando constantemente el contenido. De igual forma el servidor permite input de texto que se va actualizando en la pantalla lcd, estas dos formas crean una comunicación reciproca y fluida.

#### **Módulo de visualización (Web)**

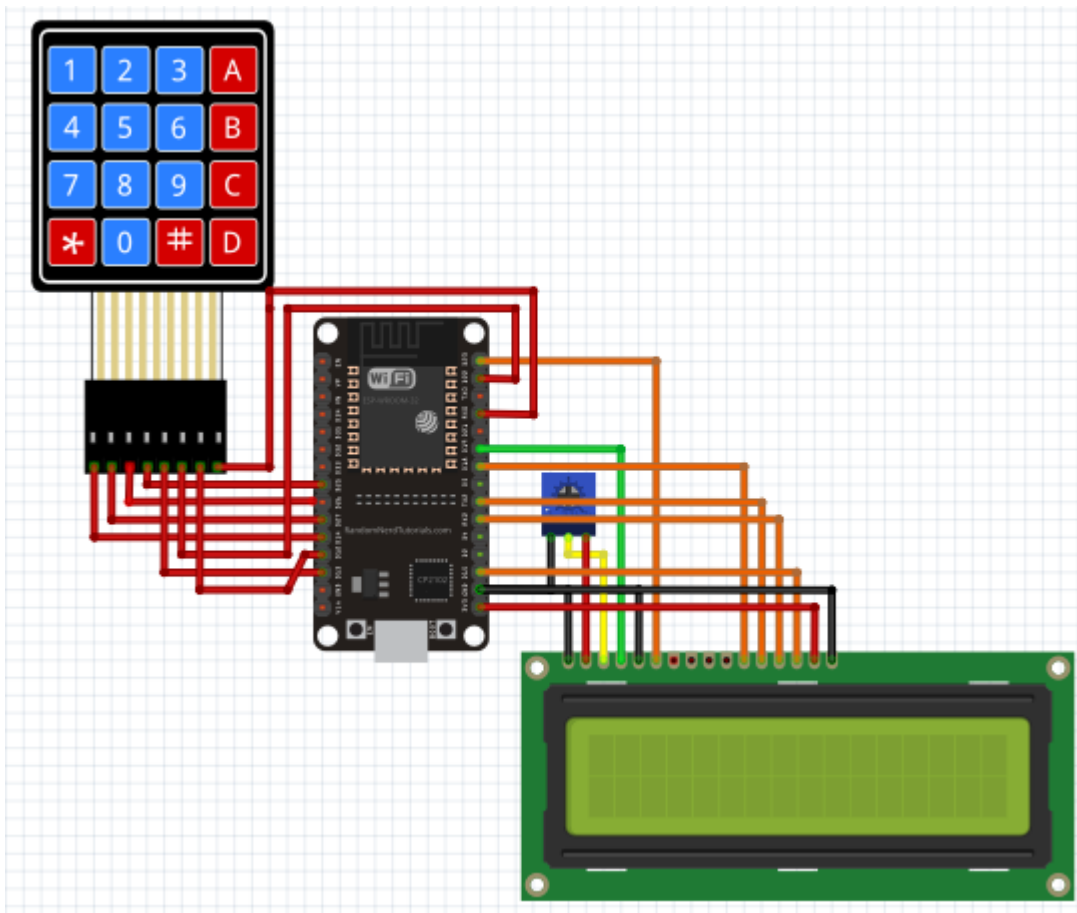
La placa esp32 permite la creación de un servidor web construido en html, este a su vez permite estilos para la mejor visualización del código. El html cuenta con un cuerpo donde va el título de la página, después va el mensaje que se va cargando desde teclado (hay una línea de código que recarga el servidor cada vez que una letra es mandada al servidor) y un label con un input de texto y un botón de enviar para cargar el mensaje al nodeMCU.



### Lenguajes

- Arduino para la lógica y comportamiento del proyecto
- HTML y CSS para la visualización del servidor

### Montaje de hardware





### Explicación más detallada de la lógica

Para lograr la interacción entre los diversos componentes utilizados se hizo uso de 3 librerías:

**Keypad:** Contiene métodos para mapear en teclados matriciales y registrar teclas presionadas.

**WiFi:** Soporta la creación de un servidor wifi tomando provecho de la construcción de la placa con módulo de internet.

**LiquidCrystal:** Maneja la lógica de envío de datos hasta una pantalla LCD.

El *setup* está organizado para validar la conexión a Internet de la placa. Dado el éxito de este proceso, se inicializa tanto el servidor como la pantalla LCD. Dentro del *loop*, constantemente estamos validando la conexión con un posible cliente. Ante esto, se procede a hacer un análisis de la petición HTTP que mantenga actualizadas las variables relacionadas con visualización. Luego, indiferente del éxito de conexión con un cliente, utilizamos el método *typing* que revisa las posibles entradas de teclado y soporta las condiciones necesarias para simular el funcionamiento de un teclado T9. Para finalizar, según la persistencia de lo registrado del servidor web, actualizamos la visualización de la pantalla LCD.

La simulación de teclado T9 está acompañada de funciones que se encargan de manejar las entradas y de interpretar el mensaje; con el registro de un número *k*, damos espera para una posible nueva entrada de ese mismo número *k*. Luego con esto, desde la función *interpret* - que recibe el número *k* y la cantidad de veces presionada - hacemos uso de código *ascii* y un diccionario que nos permita entender desde qué letra debemos empezar a iterar para interpretar la entrada. Finalmente, concatenamos el carácter resultante.

Dentro del algoritmo está implementado un límite de presionado para evitar entradas ambiguas con una secuencia larga del mismo número *k*, permitiendo así que letras asociadas a una misma tecla se puedan concatenar sin riesgo a ambigüedad.

A continuación, un esquema para usuarios del teclado:



## Guía de uso

**1:** ABC    **2:** DEF    **3:** GHI  
**4:** JKL    **5:** MNÑ    **6:** OPQ  
**7:** RST    **8:** UVW    **9:** XYZ  
**C:** Limpiar pantalla (CLEAR)  
**D:** Borrar caracter (DELETE)

**Nota:** Las teclas que no están registradas no son funcionales ni afectan el funcionamiento.