**TECNOLÓGICO NACIONAL DE MEXICO**

**INSTITUTO TECNOLÓGICO DE LA LAGUNA**

****

**REPORTE DE PRACTICA**

**UNIDAD 4: Microcontroladores PRÁCTICA 17: Bluetooth**

**DOCENTE: LAMIA HAMDAN M.**

| **NUM DE CONTROL** | **NOMBRE** |
| --- | --- |
| 19130514 | Isaias Gerardo Cordova Palomares |
| 19130545 | Oscar Martinez Ruiz |
| 17130763 | Raúl Martín Ayala Salais |
| 19130541 | Pedro Lopez Ramirez |
| 19130535 | Ivan Herrera Garcia |
| 18131263 | Gerardo Alberto Orozco Villegas |

**FECHA DE ENTREGA:** 06/11/2022

**TABLA DE CONTENIDO**

[**1.INTRODUCCIÓN**](#_555xbbp1n8y5) **3**

[**2. COMPETENCIA A DESARROLLAR**](#_30j0zll) **3**

[**3. CIRCUITO LÓGICO Y/O PROGRAMA**](#_vyaxf5uqhjrg) **4**

[**4. METODOLOGÍA**](#_uikunnlbl3c) **4**

[**5. RESULTADOS**](#_4j5zl4o09ir7) **7**

[**6.CONCLUSIONES Y RECOMENDACIONES**](#_swudbzgg3llj) **9**

[**7. REFERENCIAS**](#_s8u6prgrx9vv) **9**

# **1.INTRODUCCIÓN**

En las anteriores prácticas se estuvo trabajando en la aplicación descargada de play store llamada “Serial Bluetooth Terminal”, en la cual mediante la cual nos brindaba una terminal en la que una vez nos conectamos via Bluetooth a nuestro dispositivo móvil se nos permite mandarle senales a nuestro esp32.

En esta ocasión se buscará replicar algo similar, pero mediante una app construida por nosotros.

Para ello se utilizara un framework llamado “Flutter” :

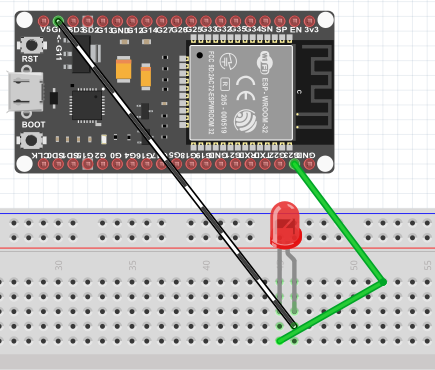
[1] Flutter es un conjunto de herramientas de interfaz de usuario de Google para crear aplicaciones de forma casi nativa para dispositivos móviles y de escritorio a partir de una única base de código.

Flutter funciona con el código existente, lo utilizan los desarrolladores y organizaciones de todo el mundo, ya que es gratuito y de código abierto.[1]

# **2. COMPETENCIA A DESARROLLAR**

* Identifica las características eléctricas de un microcontrolador.
* Conoce la arquitectura interna del microcontrolador.
* Comprende la estructura de registros del microcontrolador.
* Analiza dispositivos de entrada/salida y puertos del microcontrolador.
* Organiza y clasifica información proveniente de fuentes diversas.
* Aplica los conocimientos de programación adquiridos, como base fundamental para la solución de problemas.

# **3. CIRCUITO LÓGICO Y/O PROGRAMA**



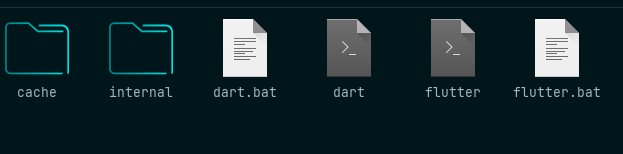
*Montaje del circuito en think cad/fritzing*

# **4. METODOLOGÍA**

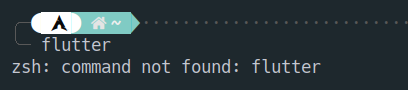
Para ello, primero instalaremos el framework en nuestro equipo, para la instalación iremos a la página oficial de flutter y descargar el archivo descomprimible.



Al descomprimir el comprimido, nos dejara los siguientes archivos.

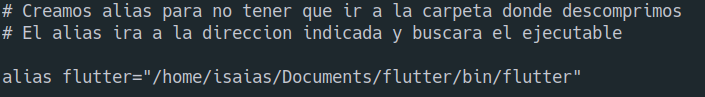


Al no contar con un instalador, deberemos de hacer algunos pasos extra, de lo contrario no podremos trabajar con flutter. Tal como se muestra en la siguiente imagen

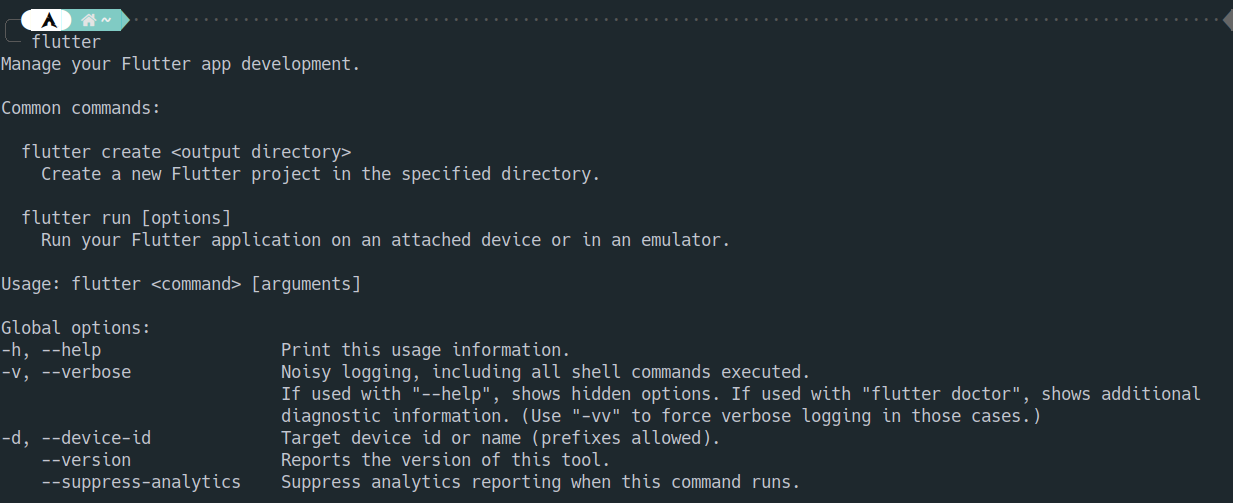


Al estar en un sistema operativo linux, tendremos que dirigirnos al archivo de configuración de nuestro intérprete de comandos, en nuestro caso es zsh. Por ende, debemos ir a ~/.zshrc y editar el archivo para agregar lo siguiente.

**alias flutter=”/home/isaias/Documents/flutter/bin/flutter”**

****

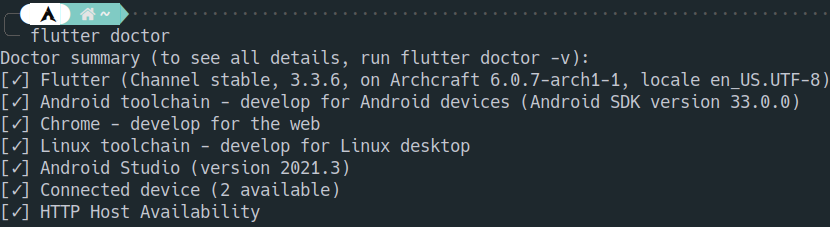
Con ese pequeño comando, de aquí en adelante al introducir el comando “flutter” irá a esa carpeta a buscar el ejecutable. Ahora tocará cerrar la terminal actual y abrir una nueva para ver los cambios reflejados.



Si ya aparecen opciones de ayuda y demás quiere decir que lo hicimos bien y ya nos detecta “flutter” cómo binario/ejecutable.

El siguiente paso es importante, ya que nos indicará si tenemos todas las dependencias satisfechas o si habrá que conseguir/instalar algún elemento.

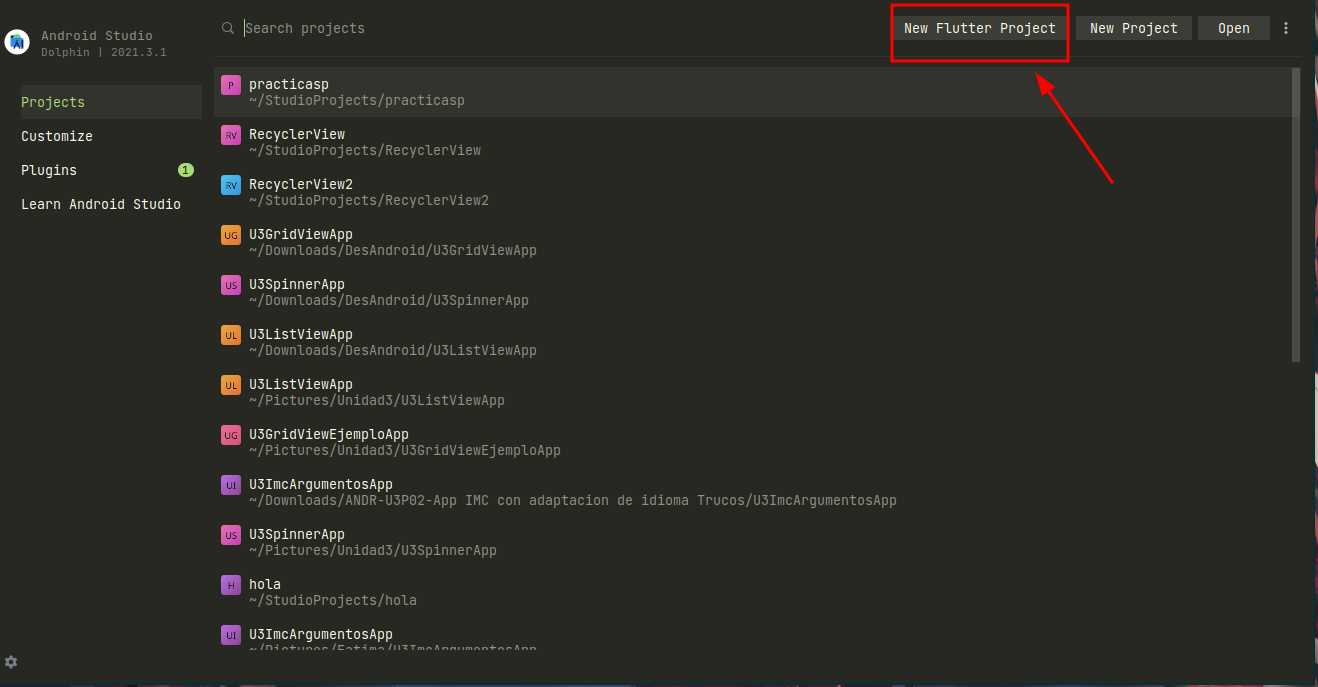
**flutter doctor**

****

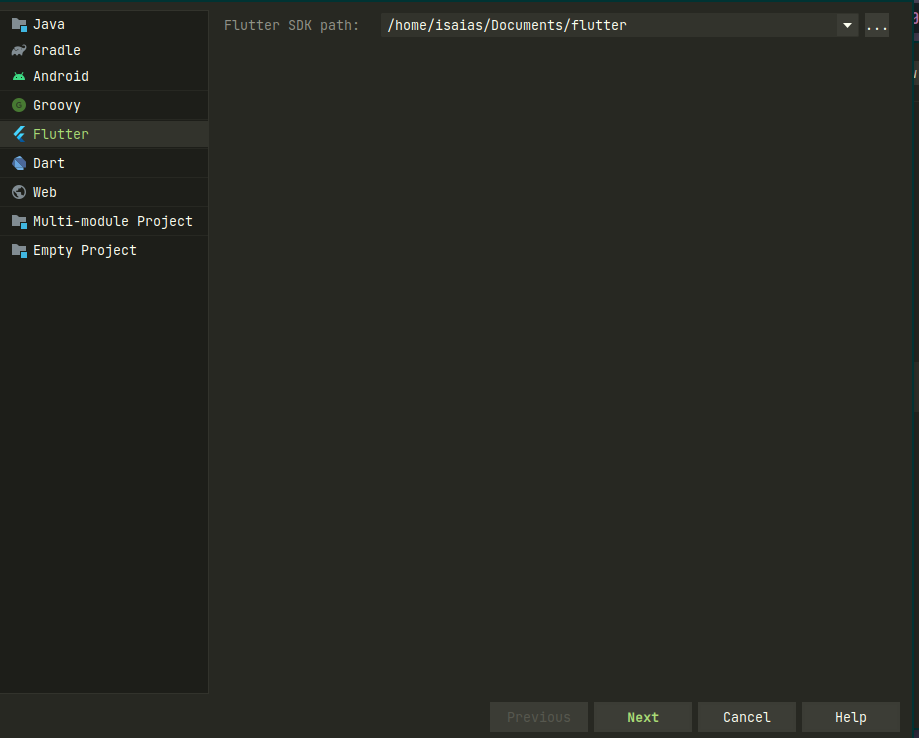
Algunos de los elementos que podemos ver que pide flutter, son elementos como el Android Studio para la compilación de aplicaciones y uso de simuladores si es requerido. Google Chrome, no funciona o no aparenta funcionar en cualquier otro navegador, a menos que este basado en Chromium. En fin, si tenemos todo palomeado eso quiere decir que ya podemos trabajar en dispositivos móviles, aplicación web y aplicaciones para linux.

**Trabajando desde Android Studio**

Antes de iniciar Android Studio podremos verificar en la parte superior si podemos trabajar con flutter, nos tiene que aparecer algo parecido a lo de la siguiente imagen.



Se abrira la siguiente pantalla y tocara pasarle el sdk, que comunmente si encuentra en el comprimido descargado al incio.



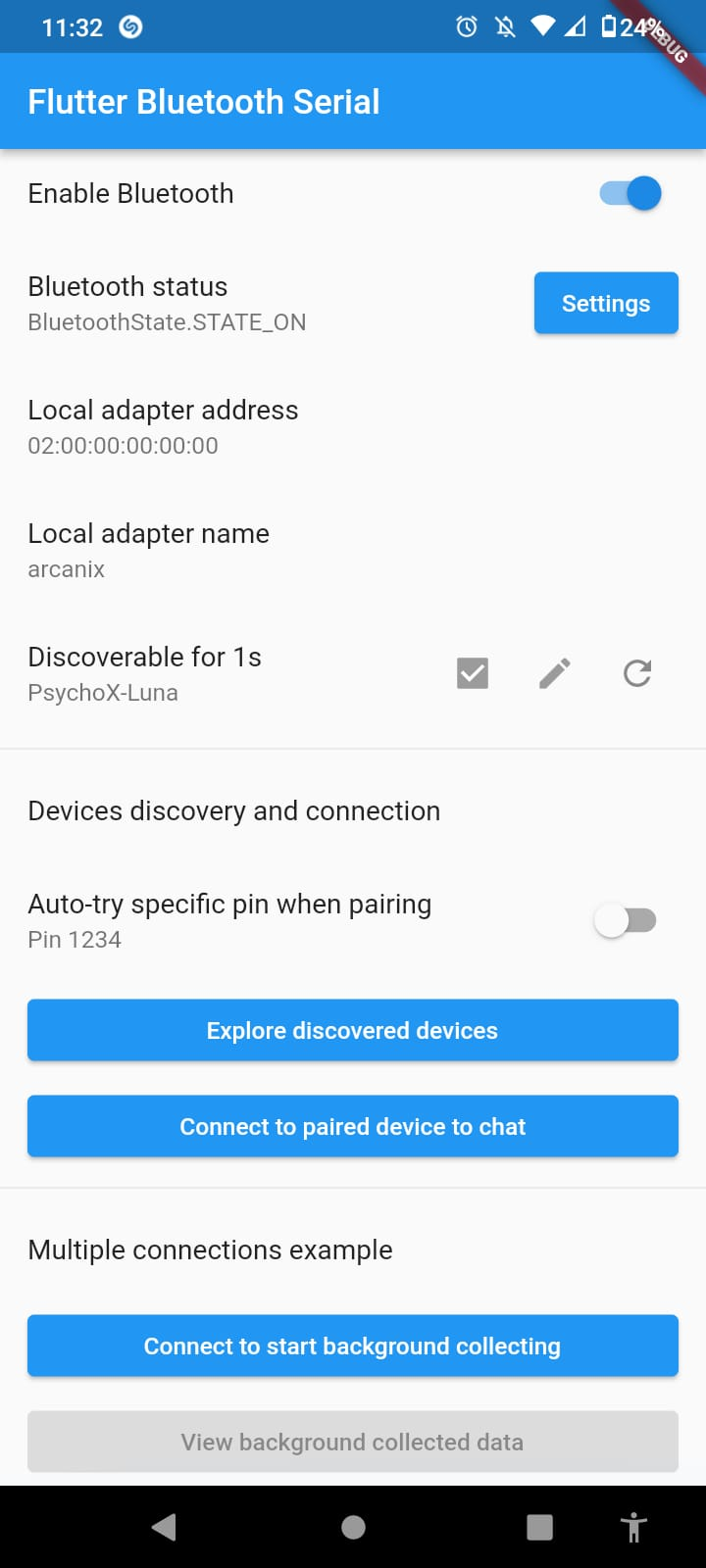
**Material necesario:**

* Arduino IDE
* Cables
* ESP32
* LED Rojo
* Android Studio

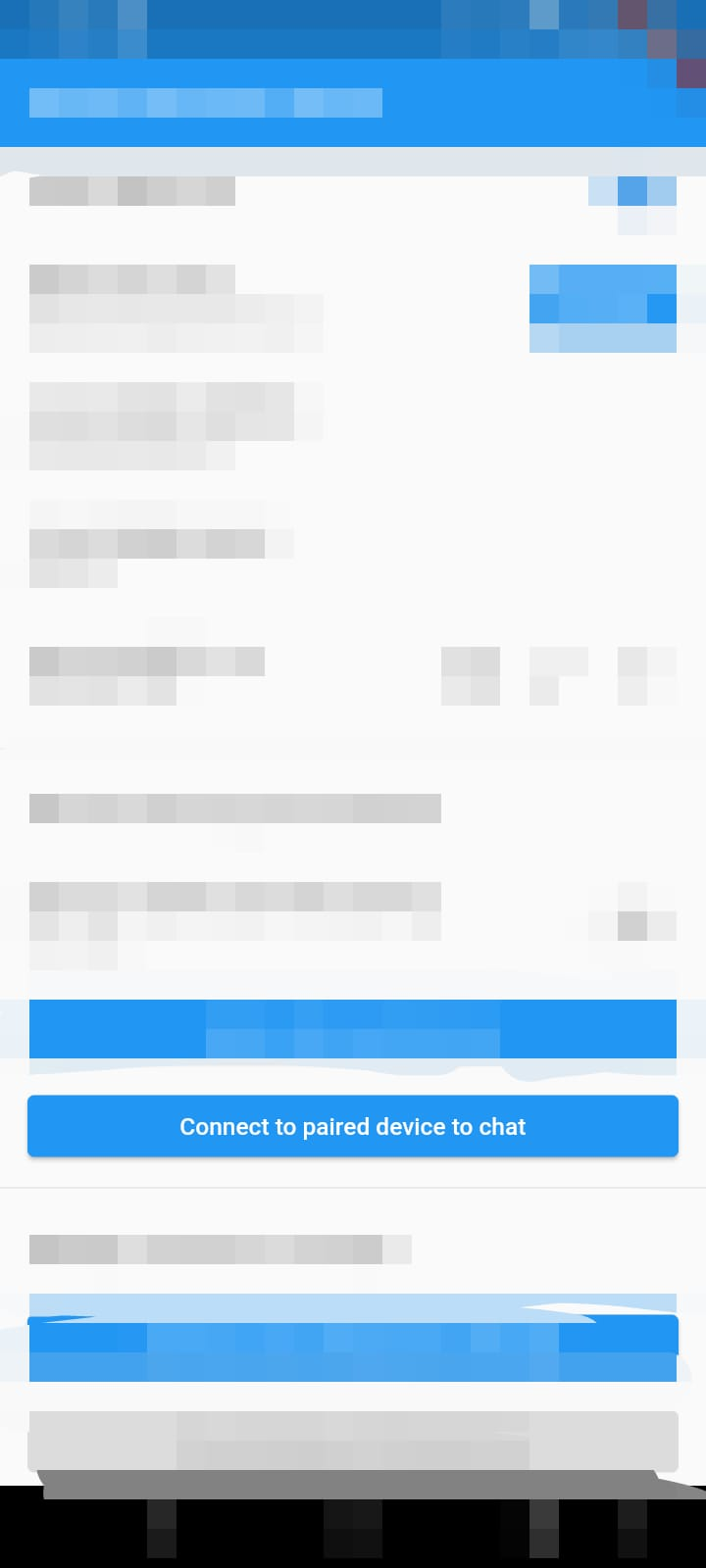
# **5. RESULTADOS**

1. **Pantalla de Inicio**

Al abrir la aplicación nos encontraremos con las siguiente vista :

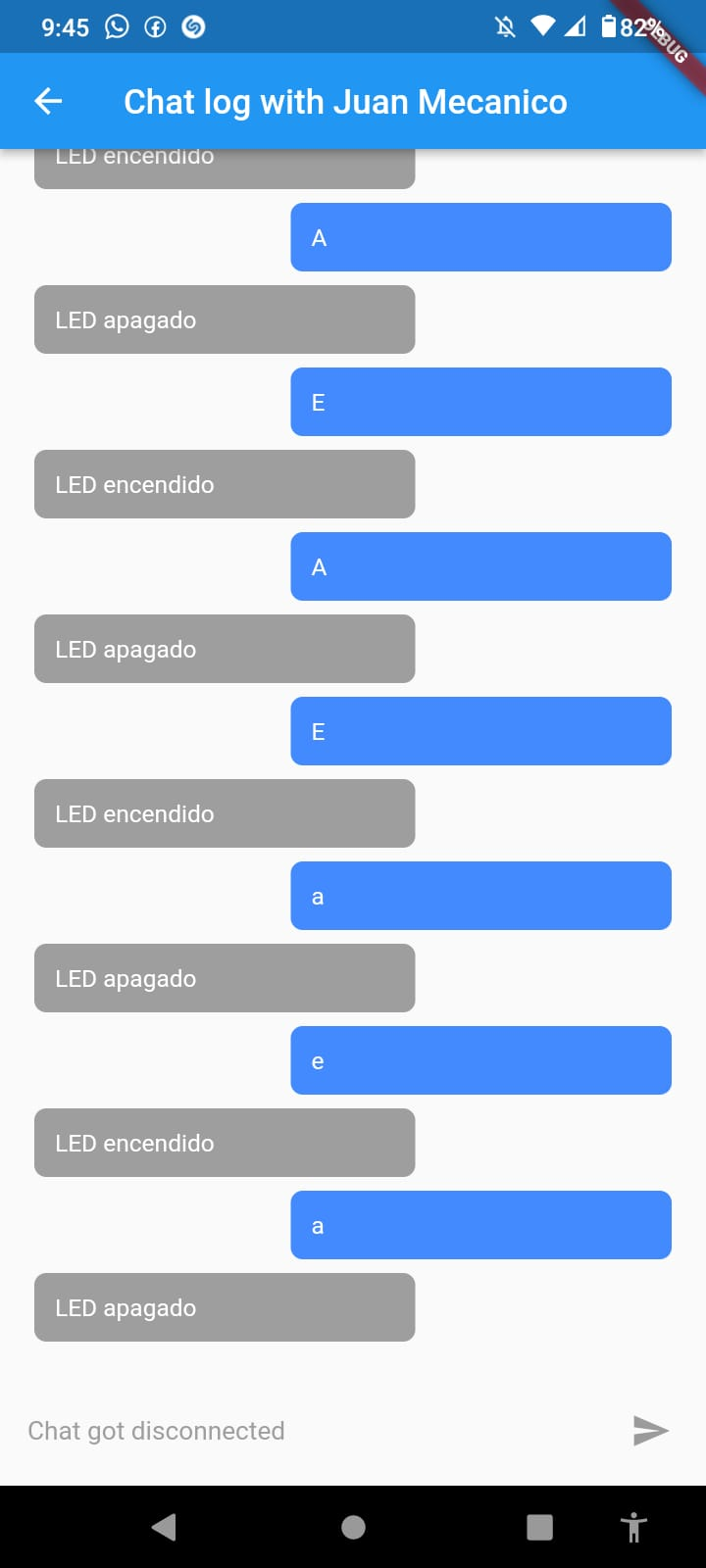


Pero una vez que hayamos hecho la conexión con el esp32 mediante wifi, la opción que usaremos será la siguiente : **Connect to paired device to chat**

****

Una vez ya conectados al chat con nuestro esp32, ya podremos interactuar con él.

Aquí mostramos la conversación que tuvimos con nuestro dispositivo :



Anexamos la siguiente liga, para que pueda ver bien el funcionamiento de nuestra practica : [Practica 17 - Bluetooth\_VideoDemostracion](https://drive.google.com/file/d/1bXRMd0s3cb5Jc3h-RulkQDGNglEL6VB0/view?usp=sharing)

# **6.CONCLUSIONES Y RECOMENDACIONES**

Fue una práctica sencilla de alguna forma, debido a que en internet encontramos mucho material de calidad acerca de este tema, tanto es así, que había alternativas haciéndola nativa desde Android Studio pero con Java, MIT Inventor con programación en bloques. Pero al final nos decidimos por la alternativa de flutter, ya que en sí nos servía para aplicaciones multiplataforma, es decir, de un código salía aplicación para escritorio, web y móvil.

# **7. REFERENCIAS**

[1]. "FAQ". Flutter documentation | Flutter.<https://docs.flutter.dev/resources/faq> (accedido el 6 de noviembre de 2022).

[2] "Install". Flutter documentation | Flutter.<https://docs.flutter.dev/get-started/install?gclid=Cj0KCQjwk5ibBhDqARIsACzmgLR8FhmZEhWERqIYCfa40Knv9vhvNDBgy7pnIq3m5x9BVpJQrIMvl8waAj8KEALw_wcB&amp;gclsrc=aw.ds> (accedido el 6 de noviembre de 2022).

[3] "flutter\_bluetooth\_serial | Flutter Package". Dart packages.<https://pub.dev/packages/flutter_bluetooth_serial> (accedido el 6 de noviembre de 2022).

[4] "scoped\_model | Flutter Package". Dart packages.<https://pub.dev/packages/scoped_model> (accedido el 6 de noviembre de 2022).

[5] "GitHub - edufolly/flutter\_bluetooth\_serial: A basic Flutter Bluetooth Serial". GitHub.<https://github.com/edufolly/flutter_bluetooth_serial> (accedido el 6 de noviembre de 2022).