

Nova:

Tu Teléfono Móvil

Alumno: Cercasi Javier

Carrera: Ing. en Computación

Asignatura: Trabajo Final 3

Tutor: Gabriel Pulido - Ruth Leiton

Universidad de Mendoza

Índice:

Etapa 1: Ambientación del espacio de trabajo.....	3
Etapa 2: Elaboración del Hardware.....	3
2.1 Compra de materiales.....	3
2.2 Diagrama Eléctrico.....	4
2.4 Esquemático del circuito impreso (PCB).....	5
2.5 Placa pertinax de Cobre.....	6
2.6 Preparación de Placa pertinax.....	7
2.7 Transferencia térmica del circuito.....	7
2.8 Cloruro ferrico.....	8
2.8.1 Resultado obtenido.....	9
2.9 Agujereado de Placa.....	9
2.10 Soldadura de componentes electrónicos.....	10
2.10.1 Herramientas utilizadas.....	10
2.10.2 Resultados obtenidos.....	11
2.10.2.1 Vista circuital.....	11
2.10.2.2 Vista Superior.....	12
2.11 Conexionado.....	13
2.11.1 Conexionado de Arduino.....	14
2.12 Ruido Eléctrico.....	17
Etapa 3: Desarrollo del Software.....	18
Etapa 4: Nube.....	18
Etapa 5: Aplicación Móvil.....	18

Metodología:

Etapa 1: Ambientación del espacio de trabajo:

Para comenzar con el desarrollo del teléfono arduino se procedió a adecuar el espacio de trabajo, adquiriendo un mueble de trabajo y una protoboard o placa de prueba para realizar el montaje y conexión del proyecto. Luego se adecuó e instaló una fuente de poder propia con salida variable, para realizar las pruebas correspondientes



Fuente: <https://www.forosdeelectronica.com/proyectos/imagenes/fuentevariable/fig1.jpg>

Antes de comenzar con la elaboración del hardware y software, se instaló el software requerido para la codificación del programa. El entorno de desarrollo instalado es el de Arduino; una aplicación multiplataforma que está escrita en lenguaje C++. El software adquirido es de distribución libre y gratuita.

El link de descarga es: <https://www.arduino.cc/en/software>

Etapa 2: Elaboración del Hardware:

2.1 Compra de materiales:

Para comenzar con la fabricación del hardware necesario para la implementación del teléfono móvil, se realizó la compra de todos los componentes. Estos fueron:

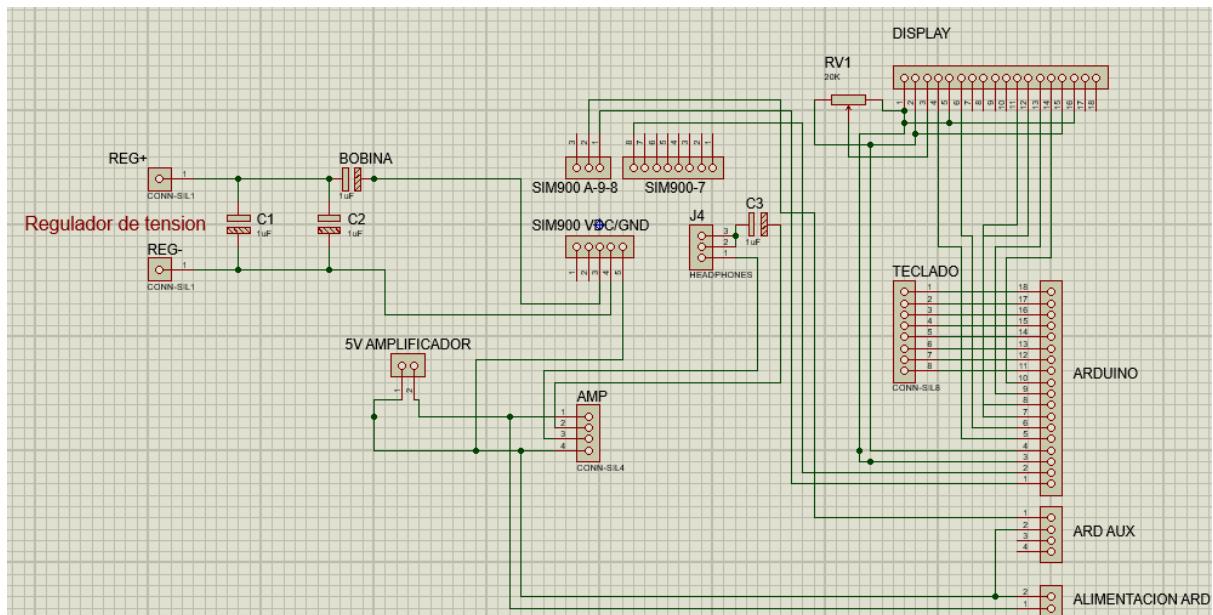
- Arduino Mega 2560
- Módulo de Comunicaciones Sim 900
- Parlante 1w/ 8 Ohms
- Micrófono Electret
- Fuente step down dc-dc XL4015
- Fuente switch 220V / 12V
- Fuente switch 220V / 5V
- Amplificador de Audio lm386 5-12v
- Teclado Membrana matricial 4x4
- Display lcd 2x16
- Buzzer zumbador
- Tarjeta sim card

- Placa Pertinac de cobre, 15x10cm
- Pack 40 Cables Macho Hembra 10cm
- 2 Tiras de pines macho para Electrónica
- 2 Tiras de pines hembra para Electrónica
- 2 Capacitores de 33uF y 1 de 470uF
- Preset de 10k
- Bornera para 2 conductores

Todos los componentes fueron comprados dentro de la provincia, en el fabricante de aparatos electrónicos “It&t:Tu aliado tecnológico”, que se encuentra ubicado en Calle. Pescara 10001, Cruz de Piedra, Mendoza.

2.2 Diagrama eléctrico:

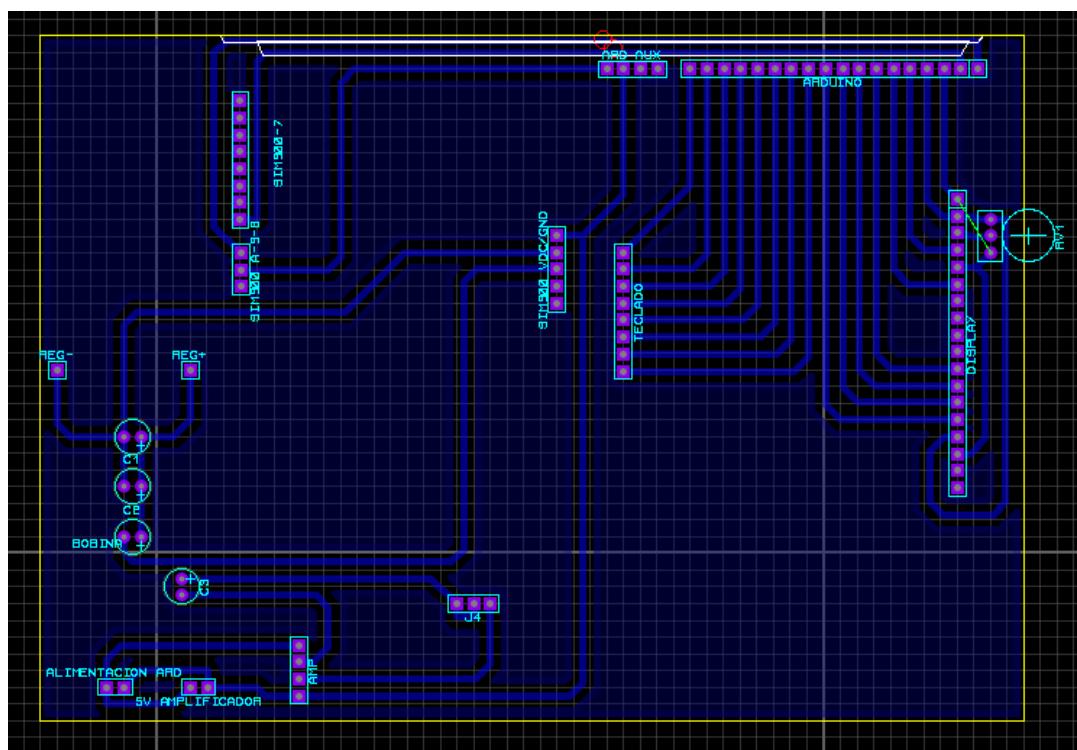
Una vez adquirido todos los componentes, se procedió a instalar el programa Proteus Design Suite 8.0, un software de automatización de diseño electrónico. Luego de completarse con su instalación y aplicación de las licencias necesarias para utilizar el pack de herramientas, se comenzó con el desarrollo del diagrama eléctrico utilizado para el hardware del teléfono. Todos los elementos utilizados fueron adquiridos de la librería de componentes del software.



Fuente: Elaboración propia

2.3 Esquemático del circuito impreso (PCB):

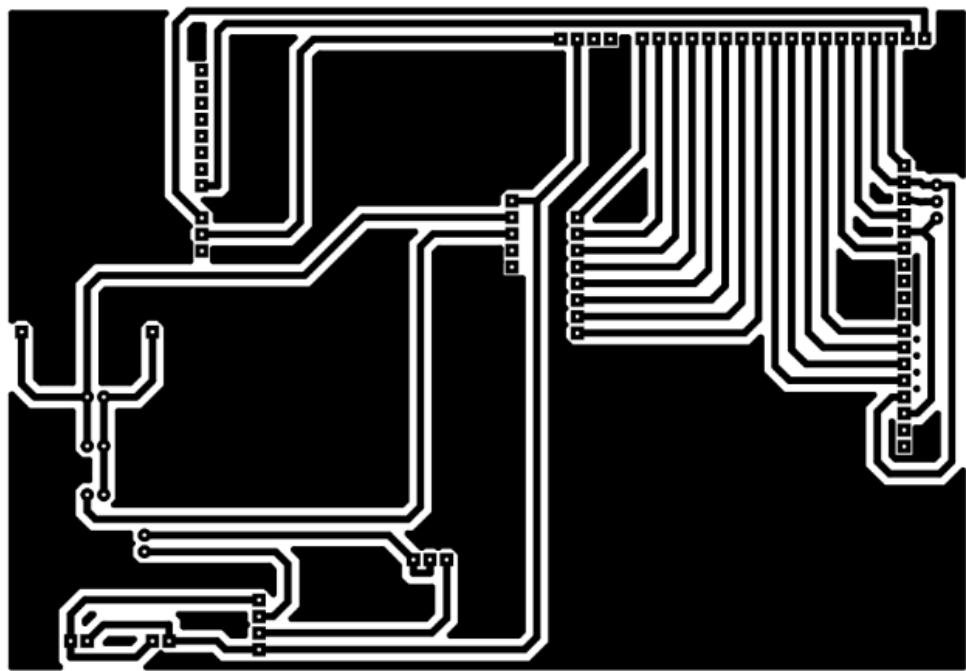
A partir del diagrama eléctrico generado, procederemos con el mismo software a migrar el diagrama eléctrico al circuito esquemático del PCB, a través de la pestaña “PCB Layout”, para posteriormente realizar la placa de circuito impreso. Se deben colocar los pines correctamente según la disposición deseada sobre la placa física, y luego realizar la traza entre los distintos componentes, evitando el entrecruzamiento de las pistas conductoras.



Fuente: Elaboración propia

2.4 Impresión del circuito:

Con el Esquemático del PCB (Printed circuit board) generado, deberá exportarse el circuito en formato PDF, con una escala 102% y con una disposición en espejo. Para realizar la impresión circuital, se debe utilizar impresora láser (a Tonner) y no con tinta, ya que posteriormente necesitaremos despegar estas pistas impresas a través de un proceso térmico. El archivo generado está listo para ser impreso en papel fotográfico y se verá como el siguiente:



Fuente: Elaboración propia

2.5 Placa pertinax de Cobre:

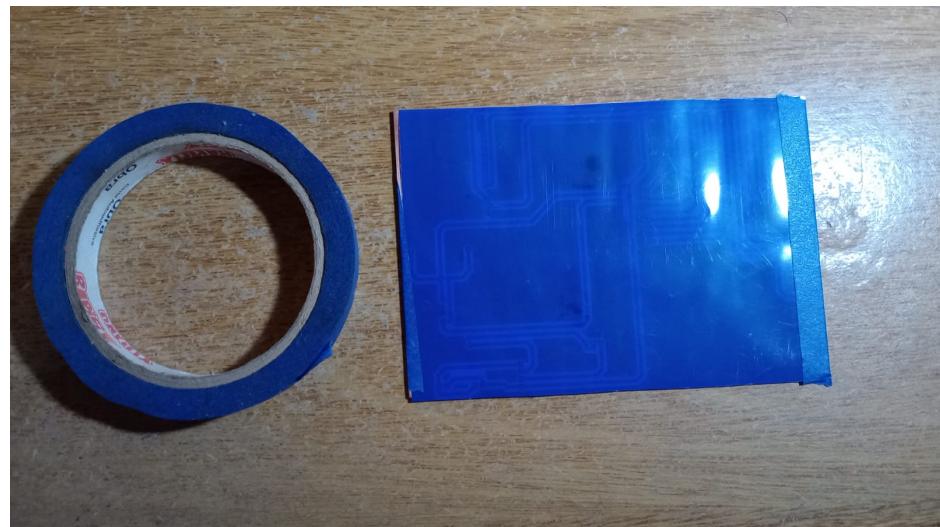
Por otro lado, se debe adecuar la placa de cobre para realizar la transferencia del circuito impreso visto en el apartado anterior. Antes de realizarlo, la placa de cobre debe ser pulida con una virulana realizando movimientos circulares, para remover imperfecciones y grasa ambiental alojada en el cobre. Al finalizar el acabado, se procederá a limpiar con alcohol los restos de polvillo generado.



Fuente: Elaboración propia

2.6 Preparación de Placa pertinax:

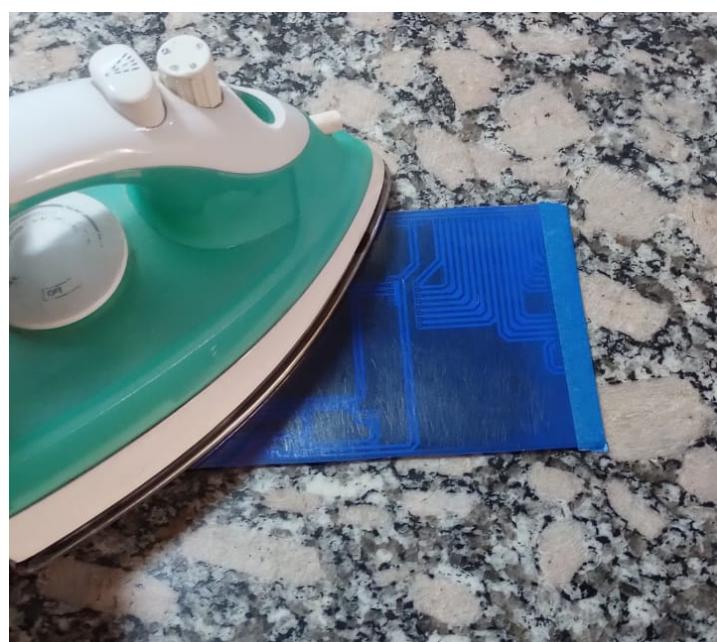
Posteriormente, se debe enfrentar el circuito eléctrico impreso(2.4) con la placa de Cobre, para realizar la transferencia de las pistas conductoras. Para fijar el papel fotográfico a la placa y evitar el movimiento del papel, se coloca cinta adhesiva en los bordes de la placa.



Fuente: Elaboración propia

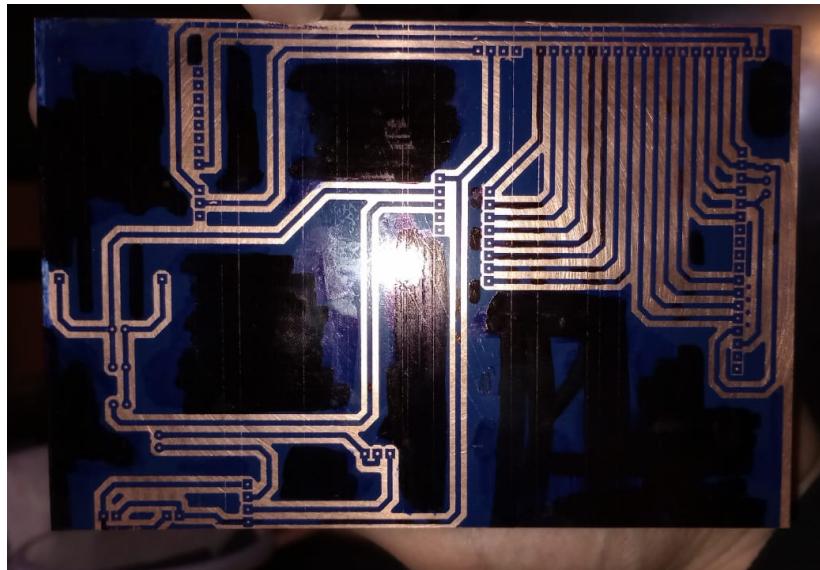
2.7 Transferencia térmica del circuito:

Una vez fijado el material, se debe aplicar calor con la utilización de una plancha durante 10 o 15 minutos, realizando movimientos circulares sobre la placa, hasta que la impresión compuesta por polvo (impresión con tóner), se haya transferido al Cobre.



Fuente: Elaboración propia

Transcurrido este tiempo, se levantará la plancha y se procederá a despegar el papel fotográfico con cuidado, para así proteger el circuito transferido. Se repasará con un marcador negro permanente aquellas pistas que no fueron transferidas correctamente con el calor.



Fuente: Elaboración propia

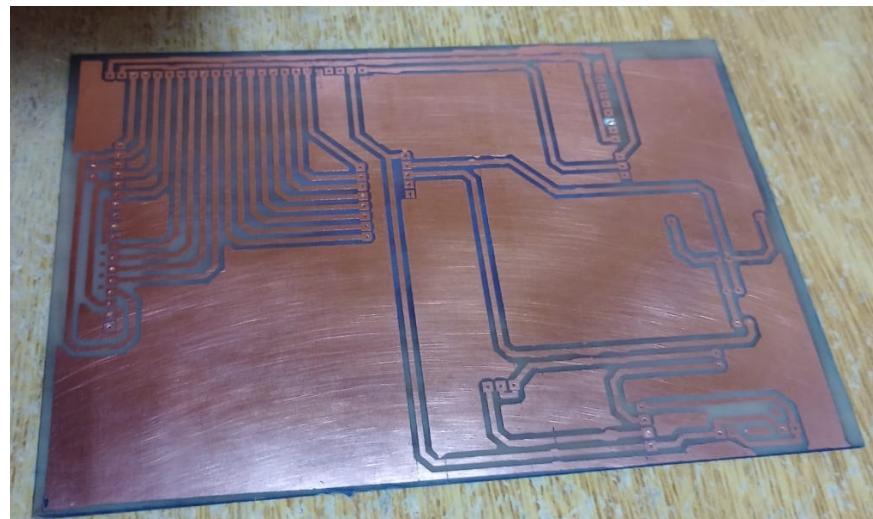
2.8 Cloruro ferrico:

Una vez retocados los defectos en la transferencia térmica del circuito, se procederá a sumergir la placa de cobre en Cloruro Férrico por 20 minutos. El cloruro férrico es un compuesto químico de fórmula FeCl₃ que tiene diversos usos en las industrias. En el campo de la electrónica es utilizado para la fabricación de circuitos impresos, ya que este compuesto produce una rápida oxidación sobre el metal, haciéndolo desaparecer, pero no produce efecto alguno sobre plástico o tintura permanente. Así, las partes de cobre sobrantes serán eliminadas de la superficie de la placa.



Fuente: Elaboración propia

2.8.1 Resultado obtenido:



Fuente: Elaboración propia

2.9 Agujereado de Placa:

Cuando el circuito ha sido retirado del ácido, este se encuentra listo para agujerear. Se procederá a agujerear los terminales de la placa para darle lugar a los componentes electrónicos. El proceso fue realizado con un taladro de precisión propio, utilizando una mecha de 0.5mm de diámetro.



Fuente: Elaboración propia

2.10 Soldadura de componentes electrónicos:

La soldadura de los componentes, fue realizada con un Cautín de 30/130W y estaño en rollo. Esto asegura la conexión entre los componentes, y proporciona continuidad eléctrica, de modo que la señal viaje a través de la conexión sin interrupciones. La soldadura une la pista del circuito impreso con los terminales de cada componente, al aplicarse estaño sobre el terminal y fundirse con el soldador.

2.10.1 Herramientas utilizadas:

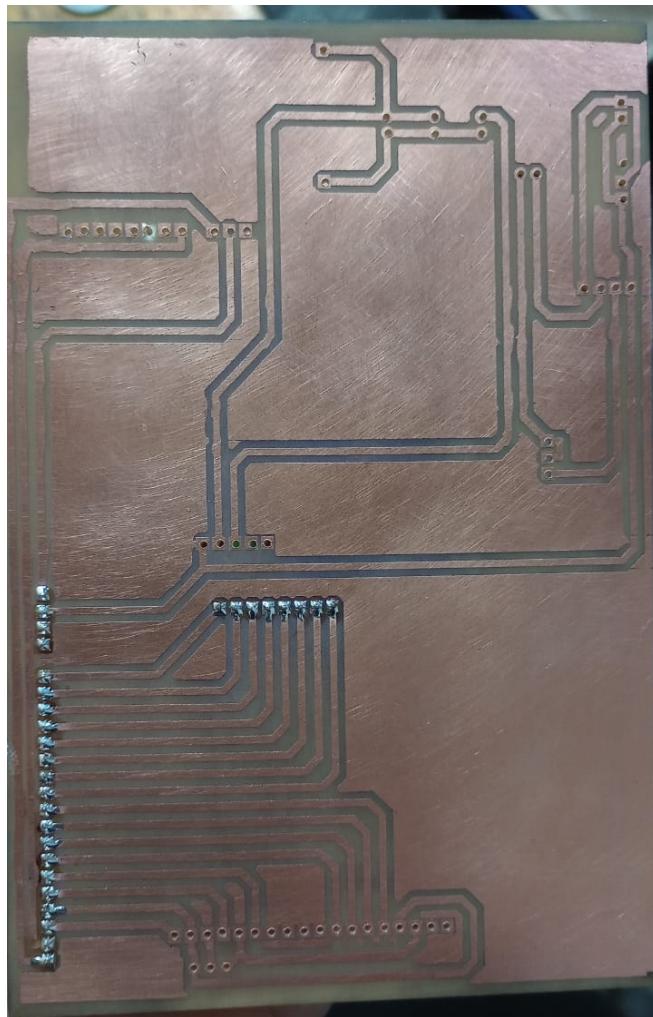


Fuente: Elaboración propia

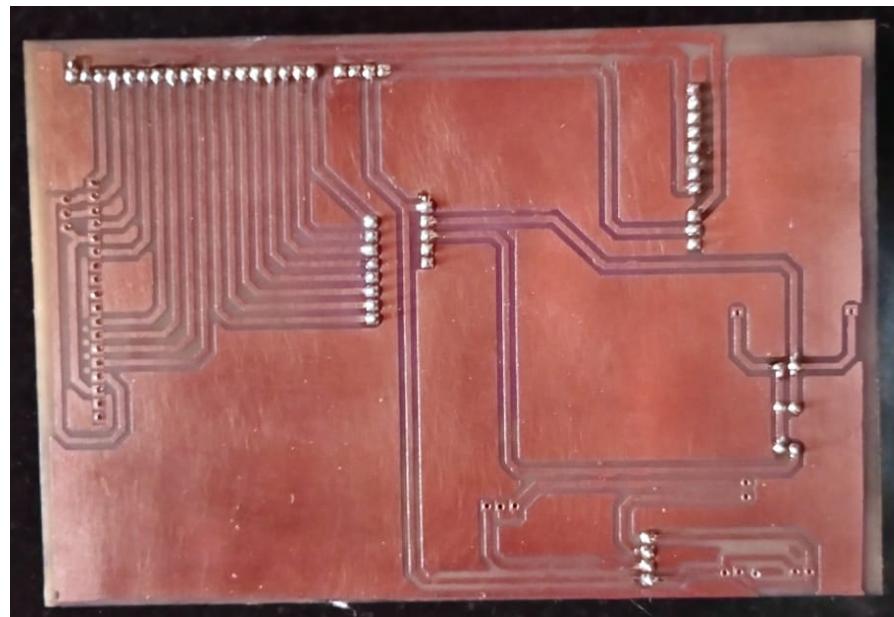
2.10.2 Resultados obtenidos:

2.10.2.1 Vista circuital:

Vemos el proceso de soldadura de los debidos componentes sobre el circuito impreso



Fuente: Elaboración propia



Fuente: Elaboración propia

2.10.2.2 Vista Superior:

Se muestra la vista superior de la placa pertinax, durante el proceso de soldadura de los debidos componentes. Se ubicaran:

- Tiras de pines hembra, para el encastre del módulo de comunicaciones SIM900, el display 16x02, y el amplificador de audio.
- Tiras de pines macho, para el encastre del teclado matricial, y el conexionado con el módulo Arduino Mega.
- 2 Capacitores Electrolíticos en paralelo de 100uf y 47uf. Estos capacitores junto con el regulador de tensión, se encargarán de brindarle una alta corriente instantánea al Módulo SIM900L, que es necesaria durante la Rafaga de transmisión (2A Durante el ingreso de la llamada telefónica).
- Bornera electrónica, para sujetar los cables de la fuente de tensión del Amplificador de Audio.
- Capacitor Electrolítico de 33uF. Utilizado para filtrar la señal proveniente del Altavoz del Módulo SIM900 hacia el Amplificador de Audio.
- Bobina de Choque, para el filtrado de la señal de alimentación.



Fuente: Elaboración propia

2.11 Conexiónado:

Una vez finalizada la soldadura de los componentes, se procederá con el ensamblaje de componentes, encastrando cada uno de los módulos sobre los pines hembra que se aprecian en el apartado 2.10.2.1. Para ello se hará seguimiento del diagrama eléctrico (2.2). El ensamblaje incluirá el agregado de:

- Lcd Display 16x02, como pantalla para la visualización de datos.
- Módulo de comunicaciones SIM900L encargado de conectarse a la red de telefonía celular para enviar llamadas, recibir llamadas y conectarse a Internet.
- Preset de 20K, encargado de modificar la intensidad luminica del Display.
- Regulador de Tension Step-down, encargado de reducir el voltaje de entrada de 12V a 5V, para alimentación del Módulo SIM900.
- Teclado Matricial 4x4, como interfaz para el cliente utilizado para realizar todo tipo de funciones. Algunas son: registro de contactos, ingreso a la agenda telefónica, ingreso y egreso de llamadas, contestar o rechazar llamadas, etc.
- Parlante 1w/ 8 Ohms, para reproducir los sonidos audibles durante la llamada.



Fuente: Elaboración propia

2.11.1 Conexiónado de Arduino:

Se debe realizar el conexiónado de la placa fabricada, con el Módulo Arduino Mega. Para ello se realizará el seguimiento del diagrama eléctrico (2.2).

Tira de pines macho (18 terminales)	Arduino Mega
1 (Verde, Módulo SIM900)	TX1 (18)
2 (Amarillo, Módulo SIM900)	RX1 (19)
3 (Naranja)	GND (Power)
4 (Rojo)	5V (Power)
5 (Marrón, Display)	2 (PWM)
6 (Negro, Display)	3 (PWM)
7 (Blanco, Display D4)	4 (PWM)
8 (Gris, Display D5)	5 (PWM)
9 (Violeta, Display D6)	6 (PWM)

10 (Azul, Display D7)	7 (PWM)
11 (Verde, Teclado)	22 (Digital)
12 (Amarillo, Teclado)	24 (Digital)
13 (Naranja, Teclado)	26 (Digital)
14 (Rojo, Teclado)	28 (Digital)
15 (Marron, Teclado)	30 (Digital)
16 (Negro, Teclado)	32 (Digital)
17 (Blanco, Teclado)	34 (Digital)
18 (Gris, Teclado)	36 (Digital)

Tira de pines macho (4 terminales)	Arduino Mega
-	-
-	-
3 (Verde)	GND (Power)
4 (Amarillo, SIM900)	9 (PWM)

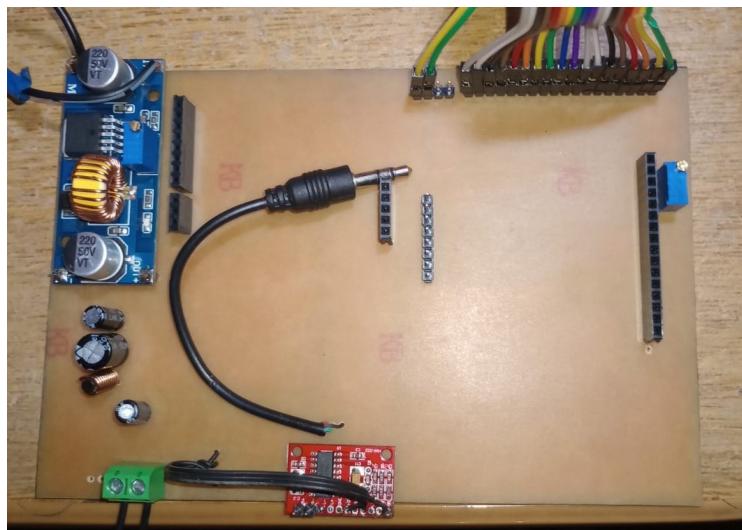


Fuente: Elaboración propia

2.12 Ruido Eléctrico:

A la hora de realizar las pruebas de audio y altavoces, se percibió altas señales de interferencias de origen eléctrico, no deseadas y unidas a la señal principal. Se encontró que el problema era el Amplificador de Audio, que era muy sensible al ruido y de una baja calidad de fabricación. Es por esto, que se cambió a un nuevo amplificador de audio: Pam8403 3W. Se alimentó con un voltaje de 5V, y se conectó con los pines de salida de parlante del SIM900. Además se soldaron los terminales del Altavoz al nuevo amplificador.

El teléfono quedó de la siguiente manera:



Fuente: Elaboración propia



Fuente: Elaboración propia



Fuente: Elaboración propia

Etapa 3: Desarrollo del Software:

(Falta desarrollo de código y últimos retoques, para poder explicar sus partes. Todo el núcleo de la programación y el desarrollo de software será explicado aquí)

Etapa 4: Nube:

La nube será implementada en la plataforma de [ThingSpeak Open-source IoT Platform](#). Según ThingSpeak, se utiliza para la “Recolección de datos en la nube con análisis de datos avanzado usando MATLAB”.

El protocolo de comunicación entre el Dispositivo móvil y la Nube será a través del protocolo HTTP.

La función de la plataforma IoT elegida, será para monitorear los datos generados por el dispositivo, sobre el ingreso y egreso de llamadas telefónicas. También se mostrará la duración de las llamadas realizadas o recibidas desde el Teléfono.

4.1 Registro:

Para poder realizar el seguimiento online del teléfono móvil, debemos empezar creando nuestras credenciales en el sitio Web de [ThingSpeak](#) . Esto permitirá loguearse únicamente en la nube elegida.

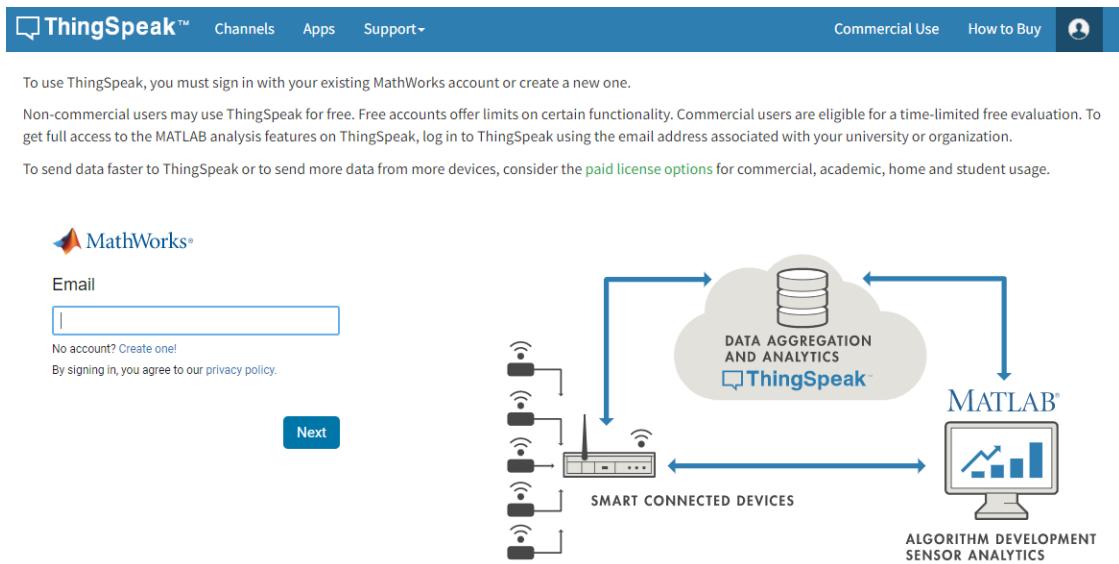
Los pasos a realizar son:

1. Acceder a [ThingSpeak](#) y elegir la opción “Get Started for Free”:



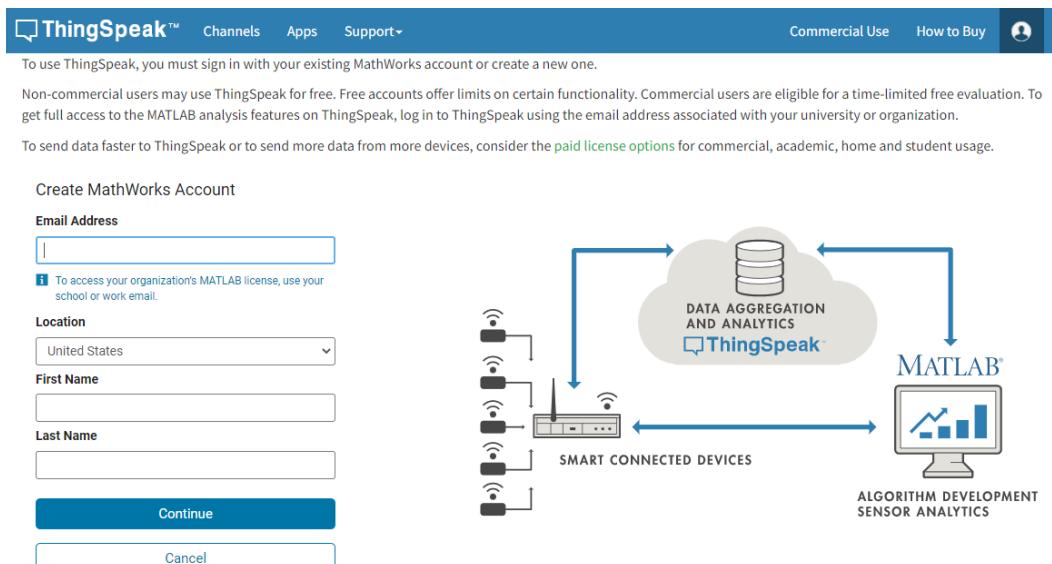
Fuente: <https://thingspeak.com/>

2. Elegir la opción “Create one!”:

The screenshot shows the ThingSpeak login page. At the top, there's a navigation bar with the MathWorks logo, 'ThingSpeak™', and links for 'Channels', 'Apps', 'Support', 'Commercial Use', 'How to Buy', and a user icon. Below the navigation, a message says 'To use ThingSpeak, you must sign in with your existing MathWorks account or create a new one.' It also notes that non-commercial users can use it for free. A link to 'Create one!' is highlighted. To the right of the text is a diagram illustrating the ThingSpeak ecosystem. It shows 'SMART CONNECTED DEVICES' (represented by icons of various sensors and actuators) connected via a router to a central cloud icon labeled 'DATA AGGREGATION AND ANALYTICS' and 'ThingSpeak'. This cloud is also connected to a 'MATLAB' interface, which includes a monitor icon showing a graph and the text 'ALGORITHM DEVELOPMENT SENSOR ANALYTICS'.

Fuente: <https://thingspeak.com/login?skipSSOCHECK=true>

3. Completar los datos de usuario, para obtener una cuenta gratuita:

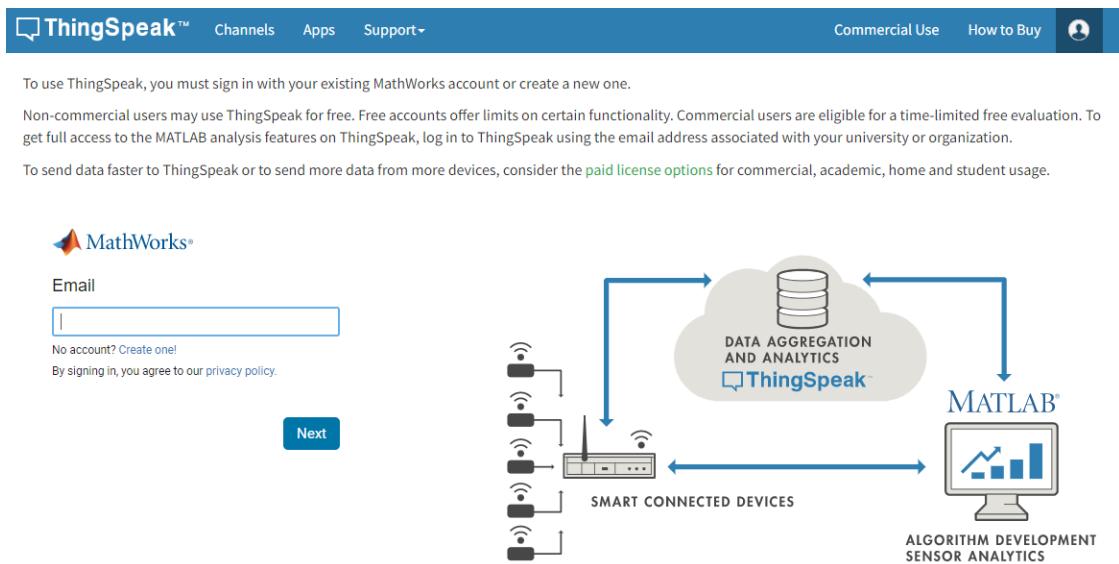
The screenshot shows the 'Create MathWorks Account' page. It has fields for 'Email Address' (with a note about using school or work email), 'Location' (set to 'United States'), 'First Name', and 'Last Name'. Below these are 'Continue' and 'Cancel' buttons. To the right of the form is the same ecosystem diagram as the previous screenshot, showing data flow from smart devices through a router to the ThingSpeak cloud, which then connects to MATLAB for analysis.

Fuente: <https://thingspeak.com/login?skipSSOCHECK=true>

4. Realizar la validación de e-mail, ingresando al link recibido en la bandeja de entrada del correo electrónico.

4.2 Inicio de Sesión:

Para realizar el inicio de sesión en la plataforma elegida, se debe acceder a la página oficial de [Thingspeak](#) y repetir el paso 2 visto anteriormente (4.1).



Fuente: <https://thingspeak.com/login?skipSSOCHECK=true>

Una vez en el sitio, se deberán completar las credenciales validadas durante el registro de la cuenta gratuita.

4.3 Crear Canal:

Al alcanzado un correcto login, se deberá crear un canal para recibir los datos generados desde el celular a través de la red móvil. El registro gratuito otorga una licencia libre que permite la creación de 4 canales en simultáneo, con 8 campos (contactos telefónicos) para mostrar por canal. Si se desea aumentar este número, el sitio web da la posibilidad de comprar distintos planes según la necesidad del cliente.

Los pasos serán:

1. Elegir la opción “New Channel”:

The screenshot shows the ThingSpeak interface with the title 'My Channels'. A table lists one channel: 'Duración de Llamada'. The table has columns for Name, Created, and Updated. Below the table are buttons for Private, Public, Settings, Sharing, API Keys, and Data Import / Export.

Help

Collect data in a ThingSpeak channel from a device, from another channel, or from the web.

Click **New Channel** to create a new ThingSpeak channel.

Click on the column headers of the table to sort by the entries in that column or click on a tag to show channels with that tag.

Learn to [create channels](#), explore and transform data.

Learn more about [ThingSpeak Channels](#).

Examples

Fuente: <https://thingspeak.com/login?skipSSOCHECK=true>

2. Completar los datos requeridos por el sitio y elegir “Save Channel”:

- a. Name: “Duración de Llamada”.
- b. Description: “Ingreso y Egreso de Llamadas de (Nombre propietario)”.
- c. Field 1 - 7: (Ingresar los primeros 7 contactos almacenados en la tarjeta Sim Card del Dispositivo Móvil).
- d. Field 8: “Números Desconocidos”.

The screenshot shows the 'Create New Channel' form. The 'Name' field is filled with 'Duración de Llamada'. The 'Description' field contains 'Ingreso y Egreso de Llamadas de Pedro'. Below these, there are eight fields labeled 'Field 1' through 'Field 8', each containing a name ('Javier', 'Marcos', 'Juan', 'Pedro', 'Lucas', 'Mateo', 'Simón', 'Número Desconocido') and a checked checkbox. To the right of the form is a 'Channel Settings' sidebar with detailed instructions for each field type.

Channel Settings

- **Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- **Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- **Show Channel Location:**
 - **Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.

Fuente: <https://thingspeak.com/channels/new>

3. Si todo ha salido correcto, debemos obtener una vista como la siguiente:

The screenshot shows the ThingSpeak channel interface for Channel ID 1749316. The top navigation bar includes links for Channels, Apps, Devices, Support, Commercial Use, How to Buy, and a profile icon. Below the header, the channel title is "Ingreso y Egreso de Llamadas de Pedro". The channel details show it was created "less than a minute ago" and has 0 entries. The interface includes tabs for Private View, Public View, Channel Settings, Sharing, API Keys, and Data Import / Export. Below these are buttons for Add Visualizations, Add Widgets, and Export recent data. A MATLAB Analysis and MATLAB Visualization button are also present. The main content area is titled "Channel Stats" and displays four separate line charts labeled "Field 1 Chart", "Field 2 Chart", "Field 3 Chart", and "Field 4 Chart". Each chart has "Duración de Llamada" on the Y-axis and "Date" on the X-axis. The charts are titled with names: Javier, Marcos, Juan, and Pedro. The bottom of the page provides a link to the source: https://thingspeak.com/channels/1749316/private_show.

4. Ingresar en la opción “API Keys” y apuntar el identificador único, ya que será ingresado en el teléfono móvil para poder autenticar al usuario y enviar datos desde el teléfono móvil hacia la API:

The screenshot shows the "API Keys" section of the ThingSpeak channel interface. At the top, the channel details are shown again. Below the header, the "API Keys" tab is selected. The main area is titled "Write API Key" and contains a text input field with the value "QJKBBDVU1CTHMPZK" and a "Generate New Write API Key" button. To the right, there is a "Help" section with text about API keys and their auto-generation, and an "API Keys Settings" section with a note about using the key for channel read.

Fuente: https://thingspeak.com/channels/1749316/private_show

4.4 Código implementado en Arduino:

La programación fue realizada en el entorno de Arduino IDE (Basado en lenguaje C++). Se puede acceder a todos los archivos generados del proyecto mediante el link: https://github.com/javicercasi/Trabajo_Final_3.

Para realizar la conexión del Dispositivo a la red móvil y su posterior envío de datos hacia la nube, el código implementado preliminar fue el siguiente:

```
#include <String.h>
unsigned long previo, actual, a;
int campo=2, valor=200;           // (Valores de Prueba)
void setup()

{
    Serial1.begin(9600);
    Serial.begin(9600);
    Serial.println("Start!...");
    Serial1.println("AT");
    delay(1000);
    ShowSerialData();
    envio_trama();
}

void loop()
{
}

void ShowSerialData(){

    while (Serial1.available()) // if serial 1, which is Gsm, is sending information,
        Serial.write(Serial1.read()); // then write those information to serial monitor
    if (Serial.available()) // if something is sent using serial monitor
        Serial1.write(Serial.read()); // then write it to Serial 1 which is GSM
}

void envio_trama(){
//field = filtrado(2616227734);
comando("AT+CIPSTATUS", 2000);
comando("AT+CIPMUX=0", 2000);
comando("AT+CSTT=\\"datos.personal.com\\",\\"datos\\",\\"datos\\\"", 1000);
comando("AT+CIICR", 3000);
comando("AT+CIFSR", 2000);
comando("AT+CIPSPRT=0", 3000);
comando("AT+CIPSTART=\\"TCP\\",\\"184.106.153.149\\",\\"80\\\"", 6000);
comando("AT+CIPSEND", 4000);
String datos="GET https://184.106.153.149/update?api_key=QJKBBDVU1CTHMPZK&field"+String(campo)+"="+String(valor);
comando(datos, 4000);
Serial1.println((char)26);
delay(5000);
comando("",500);
comando("AT+CIPSHUT",4000);
}
```

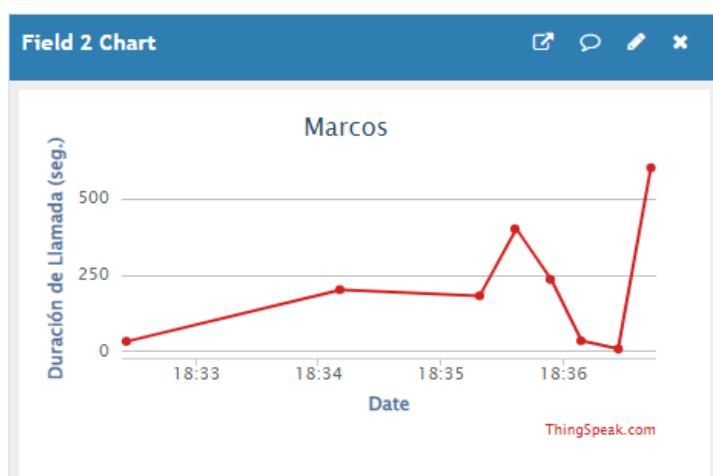
Fuente: Elaboración propia.

Para comenzar, se inicializa el puerto serial y su velocidad, para permitir la comunicación entre el Arduino y el Módulo SIM900. Desde el punto de vista de la Red, la función principal es la de “envio_trama”, encargada de conectarse a la red y enviar los datos que son generados durante las llamadas telefónicas.

Luego de una ardua búsqueda en la documentación del Módulo SIM900, los comandos AT elegidos y utilizados fueron:

1. "AT+CIPSTATUS" = Conocer el estado de la Red.
2. "AT+CIPMUX=0" = Configura el dispositivo para una conexión IP única
3. "AT+CSTT="datos.personal.com\","datos\","datos\"" = Configura el APN (Nombre de punto de acceso para que el dispositivo se pueda conectar a Internet usando las redes de nuestra operadora).
4. "AT+CIICR" = Encender la conexión inalámbrica de datos (Encender los datos móviles).
5. "AT+CIFSR" = Detectar la IP brindada por la operadora.
6. "AT+CIPSPRT=0" = Sigue el envío de datos con ">" y muestra "Send OK".
7. "AT+CIPSTART="TCP\","184.106.153.149\","80\"" = Indicamos el tipo de conexión en capa de transporte, url destino y puerto destino.
8. "AT+CIPSEND" = Envía los datos con el protocolo elegido (TCP o UDP).
9. "GET https://184.106.153.149/update?api_key=QJKBBDVU1CTHMPZK&field" +String(campo)+"="+String(valor) = Datos enviados.
10. "AT+CIPSHUT" = Desactiva los datos móviles luego del envío de datos.

Resultado obtenido:



Fuente: Elaboración propia.

Etapa 5: Aplicación Móvil:

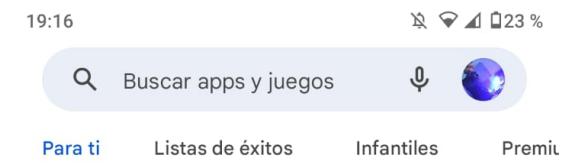
Es posible la utilización de una aplicación móvil de visualización y monitoreo de ThingSpeak, para facilitar la tarea de seguimiento web. De esta manera podremos realizar una implementación más sencilla y veloz que desde el navegador web. Su descarga es opcional, y el origen es seguro, ya que es brindado por el sitio de un [tercero](#) verificado.

5.1 App para Android:

Para descargar la aplicación de seguimiento, se deberá acceder desde la tienda de Play Store y realizar la búsqueda de la aplicación “[ThingShow](#)”.

Pasos para la instalación:

1. Buscar la Aplicación de seguimiento:



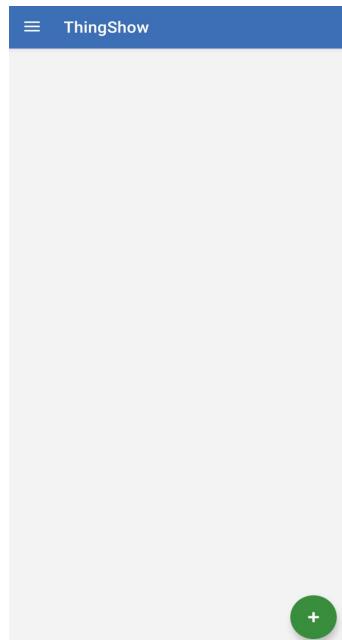
Fuente: Elaboración propia.

2. Dar click en el botón “Instalar” y “Aceptar”:



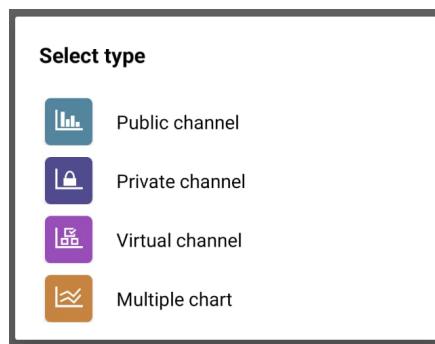
Fuente: Elaboración propia.

3. Buscar la Aplicación en el Menú de nuestro Smartphone, y elegir el botón “+” para acceder a un nuevo canal:



Fuente: Elaboración propia.

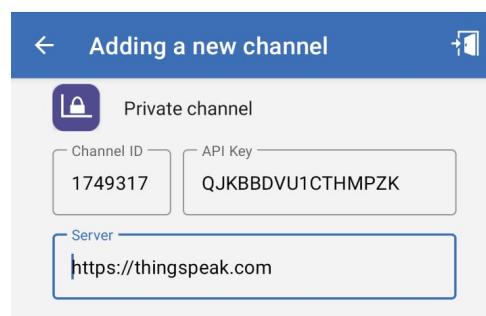
4. Seleccionar el tipo de canal Privado:



Fuente: Elaboración propia.

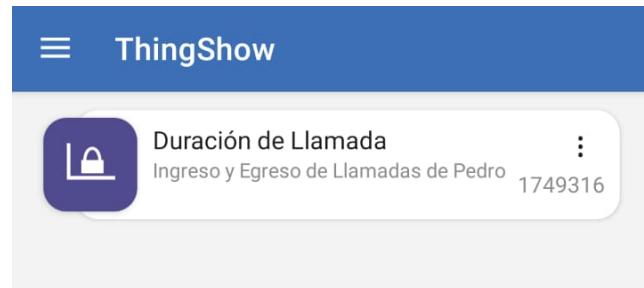
5. Ingresar las credenciales obtenidas en la Etapa 4.3.

- a. Channel ID: 1749316
- b. API Key: QJKBBDVU1CTHMPZK



Fuente: Elaboración propia.

6. Se habrá generado un nuevo canal visible en el inicio de la aplicación. Accedemos al canal “Duración de Llamadas”:



Fuente: Elaboración propia.

7. Finalmente podremos visualizar las mismas herramientas de seguimiento vistas en el navegador Web:



Fuente: Elaboración propia.