



**UNIVERSIDAD DE MENDOZA
FACULTAD DE INGENIERÍA
INGENIERÍA EN COMPUTACIÓN**

Trabajo Final de Grado

NOVA: Teléfono Móvil

Cercasi Javier

Asesor Especialista: Gabriel Pulido

Profesor a cargo: Ruth Leiton

2023

Dedicatoria:

El presente desarrollo está dedicado a todos los integrantes de mi familia, por haberme brindado la confianza, el ánimo incondicional, y el apoyo económico para poder enriquecerme en el rubro que siempre anhelé. También se lo dedico a todos los profesores de la Universidad de Mendoza, por el trato particular con cada estudiante, el conocimiento y la educación brindada.

Y me lo dedico a mí, por el esfuerzo, perseverancia y por el tiempo invertido para cumplir éste sueño.

Resumen:

Este trabajo tiene como objetivo principal el estudio y desarrollo de una nueva herramienta tecnológica de comunicación móvil, diseñada específicamente para la tercera edad, y para aquellas personas de bajos recursos.

El planteamiento y desarrollo se fundamenta en un interrogante habitual que el gran salto tecnológico del siglo XXI ha dejado, y ésta es la siguiente: ¿De qué manera se puede ayudar a un adulto mayor a fortalecer sus habilidades digitales para disminuir la brecha digital? Una buena forma es facilitando su uso, y haciéndolo agradable para estos usuarios.

Es por ello, que se proveerá de un dispositivo móvil de fácil manejo para el usuario, diseñado especialmente para la tercera edad. Inicialmente será capaz de realizar y recibir llamadas, además de tener disponible un servicio de mensajería móvil con mensajes ya pregrabados, para disminuir así el esfuerzo necesario de comunicación.

Se brindará acceso gratuito a una plataforma web, para un seguimiento diario del ingreso y egreso de llamadas, útil para aquellas personas que realizan la tarea de cuidado. Además, es posible consultar los datos generados por el dispositivo a través de una app móvil de seguimiento.

Palabras clave: Herramienta, Comunicación, Adulto, Brecha Digital, Usuario, Tercera edad, Mensajería móvil, Llamadas, Plataforma Web, Seguimiento, App Móvil.

Abstract:

The main objective of the document is the study and development of a new technological tool for mobile communication, specifically designed for seniors, and people with low resources.

The approach and development is based on a common question that the great technological leap of the 21st century has left, and this is the following: How can we help an older adult to strengthen their digital abilities and bridge the digital divide? A good way is by facilitating its use, and making it pleasant for these users.

For this, we will provide a mobile device easy to use for the user, specially designed for the elderly. Initially, it will be able to make and receive calls, in addition to having a mobile messaging service available with pre-recorded messages, in order to reduce the necessary communication effort.

We will provide free access to a web platform for daily monitoring of incoming and outgoing calls, useful for those who perform the care task. In addition, it is possible to consult the data generated by the device through a mobile tracking app.

Key Words: technological tool, communication, senior, digital divide, user, mobile messaging, calls, elderly, web platform, mobile tracking app.

Índice General:

Introducción:	1
Parte 1: Marco Teórico	2
Capítulo 1: Marco Referencial	2
1.1 Identificación del problema	2
1.2 Justificación del problema	2
1.3 Estado del Arte	3
1.3.1 Historia del Teléfono	3
1.3.1.1 Teléfono mural Gower-Bell	4
1.3.1.2 Centralita manual de batería local	5
1.3.1.3 Teléfono, 1956	6
1.3.1.4 Teléfono Celular	7
1.4 Objetivos	8
1.4.1 Objetivo General	8
1.4.2 Objetivos Específicos	9
Capítulo 2: Marco Conceptual	10
2.1 Marco tecnológico Ingenieril	10
2.1.1 Arduino	10
2.1.1.1 Módulo de comunicaciones SIM900	13
2.1.2 Nube Cloud	14
2.1.3 Plataforma IoT	16
2.1.4 Protocolo de Comunicación HTTP	18
2.2 Marco interdisciplinario	19
Parte 2: Marco Metodológico	21
Capítulo 1: Desarrollo de ingeniería	21
Etapa 1.1: Ambientación del espacio de trabajo	21
Etapa 1.2: Elaboración del Hardware	22
1.2.1 Compra de materiales	22
1.2.2 Diagrama Eléctrico	23
1.2.3 Esquemático del circuito impreso (PCB)	24
1.2.4 Impresión del Circuito	24
1.2.5 Placa pertinax de Cobre	25
1.2.6 Preparación de Placa pertinax	26

1.2.7 Transferencia térmica del circuito	27
1.2.8 Cloruro férrico	28
1.2.8.1 Resultado obtenido	28
1.2.9 Perforado de la Placa	29
1.2.10 Soldadura de componentes electrónicos	30
1.2.10.1 Herramientas utilizadas	30
1.2.10.2 Resultados obtenidos	30
1.2.10.2.1 Vista circuital	30
1.2.10.2.2 Vista Superior	32
1.2.11 Conexionado	33
1.2.11.1 Conexionado de Arduino	34
1.2.12 Ruido Eléctrico	36
Etapa 1.3: Desarrollo del Software	39
1.3.1 Configuración Inicial	41
1.3.2 Funciones Bucle	43
1.3.3 Funciones Principales	44
Etapa 1.4: Nube	46
1.4.1 Registro	46
1.4.2 Inicio de Sesión	48
1.4.3 Crear Canal	49
1.4.4 Código Arduino para la Nube	52
Etapa 1.5: Aplicación Móvil	55
1.5.1 App para Android	55
Capítulo 2: Análisis de Resultados	60
Conclusiones	63
Anexos	64
Fuentes Bibliográficas	65

Índice de figuras:

Figura 1: Telefono mural Gower-Bell	4
Figura 2: Centralita manual de batería	5
Figura 3: Avances en el teléfono	6
Figura 4: Teléfono de Baquelita.	6
Figura 5: Teléfono Celular de Motorola.	7
Figura 6: Smartphone	8
Figura 7: Arduino Mega	10
Figura 8: Parlante	11
Figura 9: Micrófono Electret	11
Figura 10: Fuente Step Down	12
Figura 11: Amplificador de audio	12
Figura 12: Teclado matricial	13
Figura 13: Display LCD	13
Figura 14: Módulo de comunicaciones SIM900L	14
Figura 15: Plataforma IoT	15
Figura 16: Fuente de poder	21
Figura 17: Diagrama Eléctrico	23
Figura 18: Circuito esquemático	24
Figura 19: Circuito impreso	25
Figura 20: Placa pertinax	26
Figura 21: Preparación de placa	26
Figura 22: Transferencia térmica	27
Figura 23: Repaso de tinta	27
Figura 24: Colocado de placa en Cloruro Férrico	28
Figura 25: Circuito en placa	29
Figura 26: Perforado de Placa	29
Figura 27: Herramientas	30
Figura 28: Proceso de soldadura.	31
Figura 29: Proceso de soldadura.	31
Figura 30: Proceso en vista superior	32
Figura 31: Conexionado	34

Figura 32: Vista general de funcionamiento	36
Figura 33: Reparación de Ruido Eléctrico	37
Figura 34: Funcionamiento de la placa con sus componentes	37
Figura 35: Conexionado de Placa con Módulo Arduino Mega	38
Figura 36: Teclado	38
Figura 37: Árbol de ficheros	39
Figura 38: Programa Main	40
Figura 39: Funciones definidas	41
Figura 40: Botones en teclado	43
Figura 41: Registro en plataforma IoT	46
Figura 42: Registro en plataforma IoT	47
Figura 43: Registro en plataforma IoT	47
Figura 44: Inicio de Sesión	48
Figura 45: Crear canal.	49
Figura 46: Configurado de canal	50
Figura 47: Vista principal	51
Figura 48: Configurado de API Key	52
Figura 49: Código de Arduino para la nube	53
Figura 50: Gráfico de duración de llamada por destino	55
Figura 51: Busqueda de Aplicación	56
Figura 52: Busqueda de Aplicación	56
Figura 53: Inicio de Aplicación	57
Figura 54: Configurado de canal	57
Figura 55: Configurado de canal	58
Figura 56: Canal configurado	58
Figura 57: Vista de app de seguimiento	59
Figura 58: Resultado de la fabricación del Hardware	60
Figura 59: Resultado del desarrollo de software	61
Figura 60: Resultados obtenidos en la nube	62
Figura 61: Resultados obtenidos en la App	62

Introducción:

Para realizar el desarrollo y fabricación del dispositivo móvil apropiadamente, se adopta una estructura de tesis con elementos introductorios, cuerpo del trabajo y elementos finales.

En el cuerpo del trabajo estableceremos un marco teórico, subdividido en 2 capítulos: marco referencial y marco conceptual.

En el capítulo 1 se abordará el problema, y la justificación por la cual es importante acercar a las personas de la tercera edad a la tecnología. Se dividen las tareas a realizar en objetivos y se describe el estado actual de conocimiento que hay sobre éste tema.

En el capítulo 2 se describirá el conocimiento sobre las tecnologías a utilizar en la fabricación del dispositivo móvil, y a quien está destinado éste desarrollo

Como parte 2 del cuerpo del trabajo, se definirá la metodología implementada para la fabricación del dispositivo, con todos los resultados parciales que se obtengan.

Para finalizar, se expondrán las conclusiones, anexos y bibliografías consultadas.

Parte 1: Marco Teórico

Capítulo 1: Marco Referencial

1.1 Identificación del Problema:

En la actualidad, muchos habitantes de escasos recursos y personas de la tercera edad encuentran dificultades a la hora de hallar y utilizar un dispositivo móvil que sea cómodo y sencillo de manipular. Una buena forma de disminuir esta brecha digital es facilitando su uso, y haciéndolo agradable para estos usuarios.

Existen 2 tipos de dispositivos móviles: Smartphone o celular básico. El primero refiere a teléfonos móviles avanzados capaces de cumplir con diversas funciones. El segundo se trata de dispositivos con botones grandes, sencillos de usar, y con las funciones principales de realizar y recibir llamadas. Este segundo tipo se afrontará en el proyecto.

1.2 Justificación del Problema:

Acercar la tecnología a la tercera edad es importante e imprescindible para que puedan estar siempre conectados e integrados en la sociedad por medio de ella. Esta integración resulta además de suma importancia para las familias de estos usuarios, por la tranquilidad que brinda el saber que siempre estarán conectados.

Para solucionar este problema, se proveerá de un dispositivo móvil basado en Arduino, que si bien consta de una arquitectura sofisticada, resulta de fácil manejo para el usuario. Especial interés se pondrá en el diseño para usuarios de la tercera edad.

El dispositivo inicialmente será capaz de realizar y recibir llamadas, brindando además acceso a una plataforma web para realizar un seguimiento diario del dispositivo móvil por parte de las familias.

Soportará la tecnología sim card, que almacenará de manera segura un número de teléfono y permitirá registrar hasta 250 contactos.

Además incluirá una pantalla para visualizar contactos telefónicos pre-guardados para un manejo intuitivo y un teclado amplio de botones notables logrando una experiencia cómoda para el usuario. El módulo encargado de realizar la comunicación telefónica, poseerá una antena externa de gran ganancia, pensado para aquellas zonas donde la señal sea débil.

1.3 Estado del Arte:

A continuación, se expondrá un viaje en el tiempo sobre una visión global del nacimiento del teléfono, hasta llegar a nuestros días provocando un cambio radical en la vida diaria de las personas.

1.3.1 Historia del Teléfono:

El primer “teléfono” (procedente del griego “hablar a lo lejos”) se dio a conocer hace 145 años, de la mano de Alexander Graham Bell, quien lo patentó en 1876, considerándose un dispositivo capaz de transmitir sonidos por un cable a través de pulsos eléctricos. Sin embargo, el inventor escocés no fue el creador del aparato telefónico. El verdadero inventor fue Antonio Meucci de origen italiano para el año 1854, que había ya intentado patentarlo pero no tuvo éxito, debido a problemas económicos.

Como afirma Bolognesi Andres, “Sitio web dinámico aplicado a las nuevas tecnologías móviles”, 2013, Universidad del Aconcagua, Argentina: “Durante más de un siglo se mantuvo el error de atribuir a Graham Bell la paternidad del teléfono hasta que finalmente, el 11 de junio de 2002, el Congreso de los Estados Unidos reconoció oficialmente a Meucci como su verdadero inventor.”

A continuación se presenta una síntesis de elaboración propia a partir de las definiciones de la Fundación Telefónica Madrid (2013) y otros artículos encontrados en Google Academic:

1.3.1.1 Teléfono mural Gower-Bell:

Para el año 1876, Bell crea un primer prototipo de teléfono, que comunicó 8km de distancia. Este producto logró ser patentado, ganando el reconocimiento de otros inversores como Western Union Telegraph Company, que sin embargo rechazaron el producto. Es por esto, que Bell decidió crear su propia empresa Bell Telephone Company. Luego de 11 años de haber fundado la compañía, más de 150.000 habitantes norteamericanos poseían un teléfono.



Figura 1: Telefono Gower-Bell.

Fuente:https://espacio.fundaciontelefonica.com/wp-content/uploads/descargas/1420478947-historia_de_las_telecomunicaciones_cuadernoprofesores.pdf

La comunicación se establecía entre 2 teléfonos conectados entre sí. A medida que el número de usuarios comenzó a aumentar, esto se volvió un problema ya que se necesitaba una gran cantidad de hilos

conectando cada aparato con los demás. Con esto surgieron las centrales telefónicas manuales.

1.3.1.2 Centralita manual de batería local:

Las centrales telefónicas manuales se implementaron en 1884, donde cada abonado solo debía encontrarse conectado a una central. Una administradora controlaba las llamadas y establecía la comunicación mediante el número telefónico. La energía para estos aparatos se suministraba a través de baterías con electricidad, provenientes de imanes, poniéndose en funcionamiento girando un dynamo.

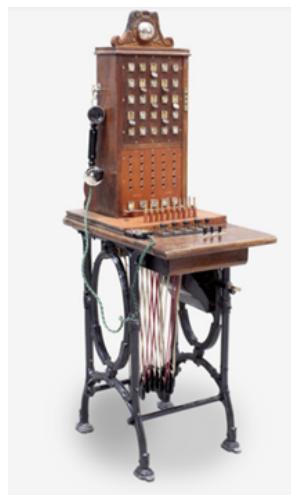


Figura 2: Centralita manual de batería

Fuente:https://espacio.fundaciontelefonica.com/wp-content/uploads/descargas/1420478947-historia_de_las_telecomunicaciones_cuadernoprofesores.pdf

Más tarde, el inventor estadounidense Almon Brown Strowger realizó una serie de experimentos, dando como resultado un sistema telefónico sin la necesidad de requerir una operadora para interconectar 2 puntos de la ciudad.



Figura 3: Avances en el teléfono.

Fuente:https://espacio.fundaciontelefonica.com/wp-content/uploads/descargas/1420478947-historia_de_las_telecomunicaciones_cuadernoprofesores.pdf

1.3.1.3 Teléfono, 1956:

Estos teléfonos fueron realizados con Baquelita, utilizando el uso del plástico. Este era mucho más económico y liviano que los anteriores. Además del material, los teléfonos empiezan a incorporar el disco de marcar, permitiendo a los clientes llamar directamente al número del abonado de destino, sin la necesidad de una telefonista.



Figura 4: Teléfono de Baquelita.

Fuente:https://espacio.fundaciontelefonica.com/wp-content/uploads/descargas/1420478947-historia_de_las_telecomunicaciones_cuadernoprofesores.pdf

1.3.1.4 Teléfono Celular:

Los primeros teléfonos celulares eran capaces de transmitir la voz. Se basaban en la transmisión de pulsos eléctricos a través de ondas de radio, sin la necesidad de estar en un lugar fijo. El aparato se comunicaba con una estación central, y a medida que el usuario cambiaba su posición, los sistemas de red intercambiaban la llamada con otras estaciones transmisoras-receptoras de radio.



Figura 5: Teléfono Celular de Motorola.

Fuente:https://espacio.fundaciontelefonica.com/wp-content/uploads/descargas/1420478947-historia_de_las_telecomunicaciones_cuadernoprofesores.pdf

Aunque los primeros sistemas de telefonía móvil fueron conocidos durante la Segunda Guerra Mundial, no fue hasta los años 80 que se popularizó su uso con la reconocida serie de “Motorola”. En 1973 Martin Cooper que trabajaba en esa empresa, creó la tecnología celular permitiendo desarrollar la telefonía móvil como la conocemos hoy en día. Desde entonces, la telefonía móvil ha evolucionado rápidamente hasta los teléfonos inteligentes 5G, incluyendo cámaras fotográficas, pantallas táctiles, navegación directa a Internet y un mercado de aplicaciones.



Figura 6: Smartphone.

Fuente:https://espacio.fundaciontelefonica.com/wp-content/uploads/descargas/1420478947-historia_de_las_telecomunicaciones_cuadernoprofesores.pdf

Como se observa, el crecimiento del mercado tecnológico ha sido exponencial. Sin embargo, pocas empresas han contemplado la necesidad de sacar al mercado un teléfono móvil más amigable y fácil de utilizar para algunas personas, especialmente como lo son personas de la tercera edad y/o de escasos recursos. Frente a esta necesidad encontrada, se sintetizará la evolución anterior mencionada, en un dispositivo móvil que englobe estos avances, pero de un diseño más amistoso al usuario. Se elegirá la plataforma “Arduino” para el análisis, diseño y desarrollo del dispositivo, debido a su bajo costo y gran versatilidad respecto a las exigencias de funcionalidad requeridas.

1.4. Objetivos:

1.4.1 Objetivo General:

- Diseñar un Dispositivo móvil basado en Arduino de fácil manejo para el usuario, desarrollado especialmente para la tercera edad.

1.4.2 Objetivos Específicos:

- Realizar los diagramas y conexión de todo el hardware necesario para la realización del teléfono móvil.
- Realizar el diseño e implementación del Backend.
- Integrar los ejecutables generados en el lenguaje de programación elegido.
- Enlazar el Módulo de Comunicaciones con la nube, para mostrar y consultar los datos generados por el dispositivo, sobre el ingreso y egreso de llamadas telefónicas.
- Acceder a una aplicación móvil, para realizar el seguimiento del usuario.
- Realizar el testeo de lo implementado.

Capítulo 2: Marco Conceptual

2.1 Marco tecnológico Ingenieril:

2.1.1 Arduino:

En este trabajo se acepta la definición brindada por el sitio oficial de Arduino (2018):

“Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Las placas Arduino pueden leer entradas y convertirlas en salidas [...]. Para hacerlo, utiliza el lenguaje de programación Arduino (basado en C++) y el Software Arduino (IDE), basado en Processing.”

Esta plataforma pone al alcance de cualquier humano, una herramienta para construir circuitos electrónicos y/o robots. El hardware está compuesto por placas con terminales, para poder ensamblar y realizar el conexionado del circuito requerido. Cada placa arduino lleva un microcontrolador en el cual se aloja el programa desarrollado. El software de Arduino es compatible con Windows, Linux y Mac.

A continuación se adjunta la imagen de una placa Arduino Mega, extraída del sitio oficial de Arduino:



Figura 7: Arduino Mega.

Fuente:<https://store.arduino.cc/products/arduino-mega-2560-rev>

Este Arduino Mega consta de un microcontrolador encargado de alojar el programa principal. Además de la versatilidad del dispositivo antes mencionado, existen gran cantidad de componentes y módulos externos con diversas funcionalidades adaptables a dicha placa, que se explican a continuación.

- **Parlante 1W/ 8 OHMS:** Altavoz encargado de reproducir los tonos audibles de una conversación.



Figura 8: Parlante.

Fuente:https://ssdiselect.com/6155-home_default/parl-1w-8ohm-14x2.jpg

- **Micrófono Electret:** Dispositivo de entrada encargado de transformar las ondas sonoras de la voz en señales eléctricas para ser transmitidas al remitente de la llamada



Figura 9: Micrófono Electret.

Fuente:<https://www.amazon.com/-/es/Sparkfun-Micr%C3%B3fon-o-electret-Breakout/dp/B01IC0HVQU>

- **Fuente Step Down DC-DC XL4015:** Fuente utilizada para adaptar los niveles de tensión eléctrica que necesitará el módulo de comunicaciones SIM900L ($V_{in} = 12v$, $V_{out} = 5v$).



Figura 10: Fuente Step Down.

Fuente:<https://www.nubbeo.com.ar/productos/fuente-step-down-ajustable-35v-5a-xl4015-nubbeo/>

- **Fuente Switch 220V / 12V:** Fuente utilizada para alimentar la fuente Step Down DC-DC.
- **Amplificador de audio LM386 5-12V:** Dispositivo encargado de amplificar la señal débil que debe ingresar al altavoz, a niveles que sean perceptibles al oído humano.

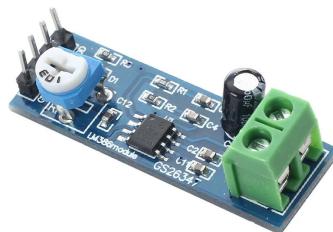


Figura 11: Amplificador de audio.

Fuente: <https://es.aliexpress.com/item/1005003026537919.html>

- **Teclado Membrana Matricial 4X4:** Dispositivo de entrada que permite la introducción de números o símbolos de parte del usuario, con los datos de la persona a contactar.

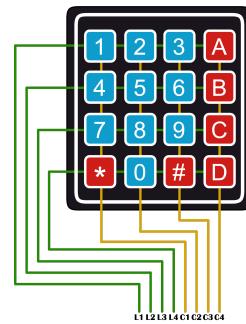


Figura 12: Teclado matricial.

Fuente:<https://naylampmechatronics.com/img/cms/019-teclado%20matricial%204x4.png>

- **Display LCD 2X16:** Dispositivo de Salida que mostrará los datos e información al usuario que utiliza el teléfono.



Figura 13: Display LCD.

Fuente:<https://www.tme.eu/es/details/mikroe-55/accesorios-para-sistemas-de-arranque/mikroe/lcd-2x16/>

Uno de los módulos externos adaptables, se encargará de realizar la comunicación telefónica que ha sido resultado de los avances tecnológicos en la telefonía móvil. Este shield es el llamado Shield SIM900L.

2.1.1.1 Módulo de comunicaciones SIM900L:

El Shield SIM900 es un módulo integrable a la plataforma de Arduino, encargado de conectar distintos sistemas o proyectos a la red de telefonía móvil. De esta forma permite transmitir mensajes de texto, llamadas telefónicas, y principalmente

conectarse a internet vía GPRS, acercando al usuario a la internet de las cosas.

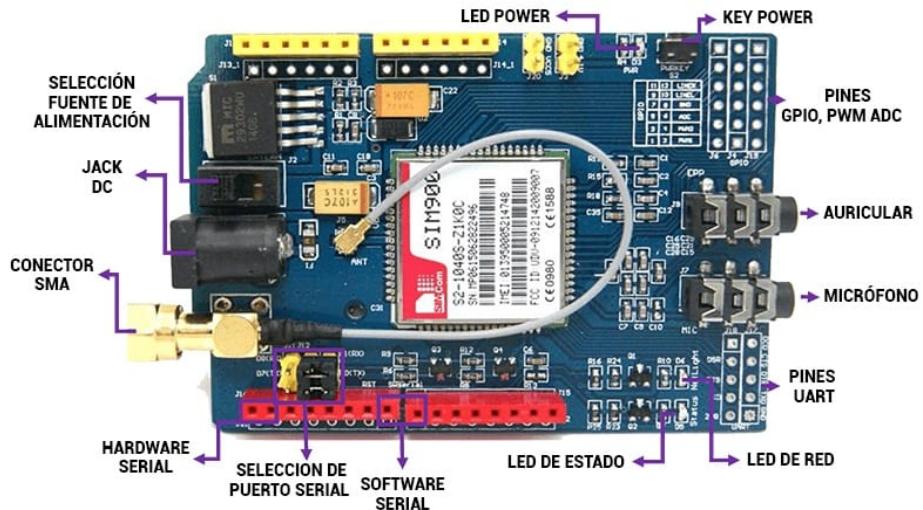


Figura 14: Módulo de comunicaciones SIM900L.

Fuente:<https://uelectronics.com/wp-content/uploads/2018/10/Pinout-SIM900-V1.jpg>

Especificaciones Técnicas principales:

- Voltaje de alimentación: 5-12V DC.
- Voltaje Input/Output: 5V.
- Corriente: 1.5mA en Reposo.
- Tecnología 2G GPRS.
- Apto para recibir comandos AT.
- Conectores para micrófono y altavoces por 3.5mm.
- Conector antena SMA para antena externa.

2.1.2 Nube Cloud:

Para realizar la adquisición de datos del dispositivo móvil, y así mostrar y consultar los datos generados sobre el ingreso y egreso de llamadas telefónicas, se hará uso de la nube cloud, sobre la cual se implementará una plataforma IOT cloud.

Si bien el concepto de Cloud surge en la década de los 90, el concepto fue evolucionando y actualmente da respuesta a múltiples necesidades empresariales y de desarrollo.

El desarrollo de redes de datos empezó a explotar en los años 90 provocando el crecimiento exponencial de usuarios, permitiendo así la aparición de centros de datos eficientes, a costos sumamente reducidos. Esta tecnología creció debido a las nuevas exigencias de velocidad de procesamiento a nivel mundial.

CLOUD o la nube, se refiere a los datos, servicios y aplicaciones que se almacenan y se transfieren mediante Internet.

Así también, sirve para poder operar estos datos sin la necesidad de computadoras físicas, pendrives, discos duros y cualquier otro elemento de almacenamiento local, pudiendo acceder a ellos con un navegador y conexión a internet.

Cloud trata de buscar la mayor eficiencia en la asignación de recursos informáticos, en caso de grandes demandas o horarios picos de trabajo, pagando únicamente por el consumo efectuado.

Dentro de la clasificación de nubes encontramos 4 tipos de ellas:

- Nube pública
- Nube privada
- Nube híbrida
- Nube combinada

Nube pública:

En este trabajo se acepta la definición brindada por Microsoft Azure (2020):

“Nube pública” se define como servicios informáticos que ofrecen proveedores externos a través de la Internet pública y que están disponibles para todo aquel que desee utilizarlos o comprarlos. Pueden

ser gratuitos o venderse a petición, lo que permite a los clientes pagar solo por el uso que hacen de ciclos de CPU, el almacenamiento o el ancho de banda que consumen. [...] Permiten ahorrar a las compañías los enormes gastos que supone tener que comprar, administrar y mantener hardware e infraestructura de aplicaciones locales”

Nube privada:

Las nubes privadas constituyen servicios informáticos que se ofrecen a través de una red interna a un espacio reducido de usuarios y no a un público en general como lo es la nube pública.

Nube híbrida:

Las nubes híbridas combinan las nubes públicas y privadas. Algunos de los recursos de software más solicitados son entre otros: el correo electrónico administrativo, aplicaciones de contabilidad, seguridad y paquetes de programas para oficinas (herramientas ofimáticas).

Nube combinada:

La nube combinada es la unión o combinación de 2 o más nubes públicas o privadas, que se administran por diferentes usuarios o proveedores. En consecuencia, los usuarios también pueden aprovechar los servicios que brinda una nube pública.

2.1.3 Plataforma IoT:

Para el año 1999, el tecnólogo británico experto en transformación digital, inventa el concepto de “Internet de las cosas”. Era director y fundador del centro Auto-ID de Massachusetts.

Se define el IoT como el mundo virtual en el que cada objeto físico posee identidad propia, teniendo la posibilidad de interactuar e integrarse en la red como si fuera una máquina o una persona. Este

desarrollo tecnológico permite interconectar de manera digital objetos cotidianos a través de internet. Su creciente adopción anticipa una tendencia a que la red global vincule más a los dispositivos entre sí.

Por lo tanto, una plataforma de IoT Cloud, es software implementado en una nube privada, que posee la característica de conectar y adquirir datos proporcionados por dispositivos físicos, tales como sensores, actuadores y aparatos electrónicos, con el objetivo de consultar la información generada.



Figura 15: Plataforma IoT.

Fuente:https://www.itmastersmag.com/wp-content/uploads/2018/11/shutterstock_572401690-e1541102000306.jpg

Está compuesta por 4 elementos principales, según ITMasters (2018):

- **Hardware:** Dispositivos y sensores que tienen la función de captar la información del entorno, para ser procesada por el software. Algunos sensores pueden ser de temperatura, proximidad, de temperatura, etc.
- **Software:** Se encarga de analizar la información generada por el Hardware. Esta funcionalidad se encuentra en la nube, generando las plataformas de IoT Cloud.

- **Conectividad:** La red facilita la transmisión y recepción de datos, desde el hardware a la nube, para permitir la interpretación de los datos. Algunas plataformas IoT incorporan nodos intermedios, generando sistemas de Edge Computing.
- **Interfaz de acceso:** Posibilitan el acceso e interacción entre usuarios y sistemas IoTs. Son el punto de acceso del equipo humano a la información generada y recolectada de los distintos elementos.

La nube se implementó en la plataforma de “ThingSpeak Open-source IoT Platform”, utilizada para la recolección de datos en la nube con un análisis de datos avanzado.

La función de la plataforma IoT elegida, será para monitorear los datos generados por el dispositivo, sobre el ingreso y egreso de llamadas telefónicas.

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

2.1.4 Protocolo de Comunicación HTTP:

El Protocolo de transferencia de hipertexto (*Hypertext Transfer Protocol*, abreviado HTTP) es un protocolo de transmisión de datos y recursos, para intercambiar datos a través de la red. Posee la conocida estructura cliente/servidor, en la cual un cliente realiza una petición a un servidor (Generalmente utilizando un navegador Web), y el servidor se encargará de responder en función de la petición hecha por el cliente.

HTTP definirá la sintaxis que usan los distintos puntos para establecer la comunicación. Las últimas versiones son HTTP/1.0, HTTP/1.1, HTTP/2.0 y HTTP/3.0

2.2 Marco interdisciplinario:

Este trabajo de tesis está destinado específicamente en mejorar la adaptación a la tecnología de personas de la tercera edad, y para aquellas personas de bajos recursos.

La tecnología de la información y comunicación ha realizado grandes avances en la sociedad, acortando el tiempo de comunicación, mejorando la conectividad entre personas y facilitando el acceso a la información a través de cualquier dispositivo con conexión a internet. El problema aquí surge cuando no todas las personas tienen capacidades para adaptarse a ellas.

La brecha tecnológica tiene diferentes variaciones, ya sea por género (afectando a más mujeres que hombres), por espacio geográfico (Afecta más a los países no desarrollados) o por grupos etarios (adultos mayores que se encuentran en desventaja, frente a aquellos jóvenes nacidos en la era digital)

Según Encuesta de Consumos Culturales y Entorno Digital (2013), el 8% de los mayores argentinos tienen destrezas digitales con un uso frecuente de computadoras, mientras que el 60% de los menores de 30 años, poseen un uso habitual de la tecnología.

La falta de capacitación digital, provoca que los adultos mayores sean más propensos a sufrir estafas virtuales, relacionadas con datos bancarios o aporte de datos personales.

Cómo superamos esta brecha digital:

1. **Apoyo de familiares:** Es necesario poseer una actitud de escucha y empatía para el manejo de la tecnología. El apoyo intergeneracional es vital, ya que poseerán destrezas adquiridas en edades más tempranas. Se debe dedicar el

tiempo necesario en un espacio tranquilo, haciéndolos partícipes del cambio. De esta manera se puede lograr una mejor comunicación y relación social, disminuyendo el aislamiento.

2. **Formación:** Enseñar a los adultos mayores, donde pueden acceder para capacitación y conocimiento online, ya que existen infinidad de videos tutoriales y explicativos en plataformas audiovisuales de la red.
3. **Apoyo institucional:** Buscar programas de formación en nuevas tecnologías, desde instituciones, centros de mayores o asociaciones
4. **Accesibilidad sencilla:** Existen nuevos dispositivos móviles con sistemas de accesibilidad especiales, que ayudará a los mayores a que el uso de la tecnología sea más sencilla. Estos pueden ser mediante pantallas grandes, con un contraste pronunciado, y teclas notables.

Parte 2: Marco metodológico

Capítulo 1: Desarrollo de ingeniería:

Etapa 1.1: 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.



Figura 16: Fuente de poder.

Fuente:[https://www.forosdelectronica.com/proyectos/imagenes/fuentevariable/fig1.jpg](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 1.2: Elaboración del Hardware:

1.2.1 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 Pertinax 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.

1.2.2 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 licencias necesarias del menú 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 propio software.

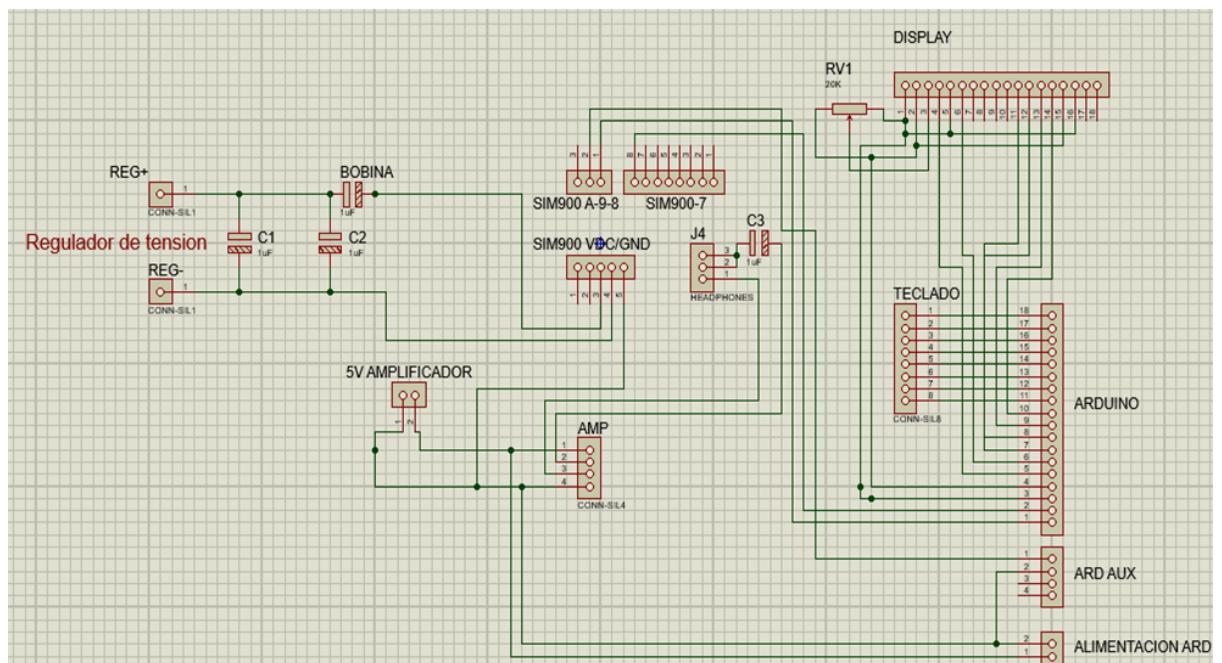


Figura 17: Diagrama Eléctrico. **Fuente:** Elaboración propia

1.2.3 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.

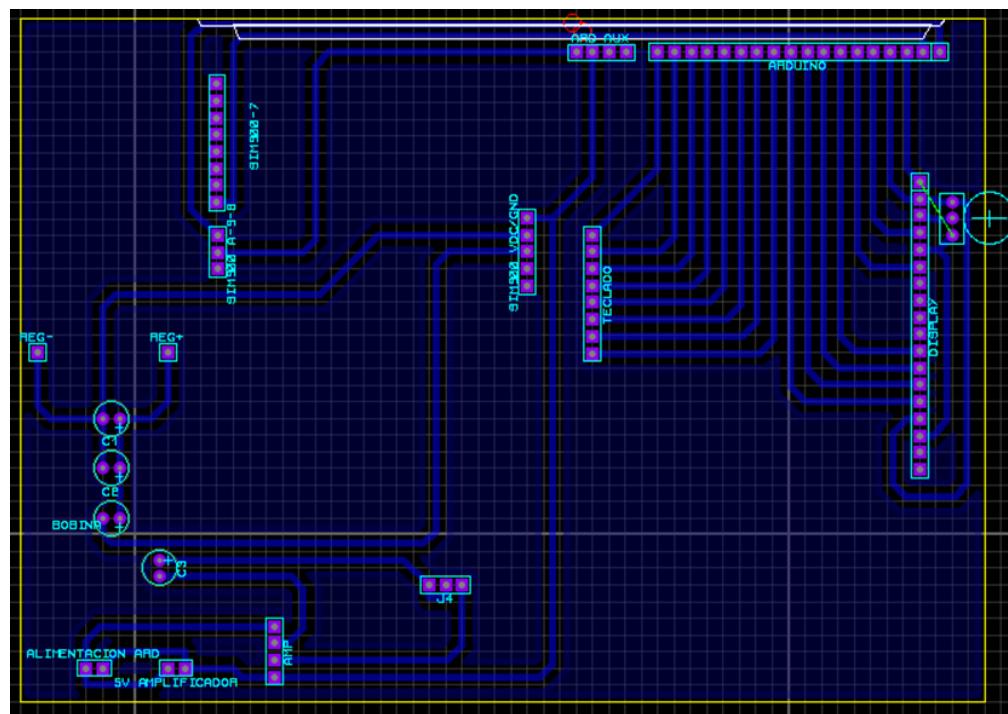


Figura 18: Circuito esquemático. **Fuente:** Elaboración propia

1.2.4 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 y transferir estas pistas

impresas a la placa de cobre utilizando un proceso térmico. El archivo generado está listo para ser impreso en papel fotográfico y se verá como el siguiente:

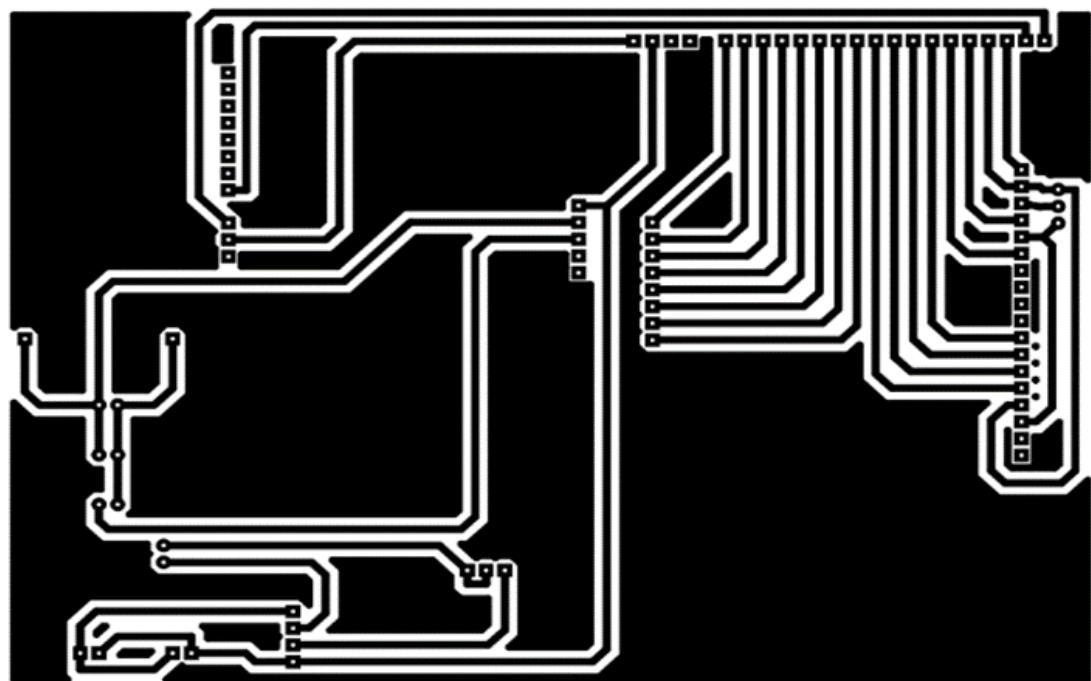


Figura 19: Circuito impreso. **Fuente:** Elaboración propia

1.2.5 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 mediante 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 generados.



Figura 20: Placa pertinax. **Fuente:** Elaboración propia

1.2.6 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.

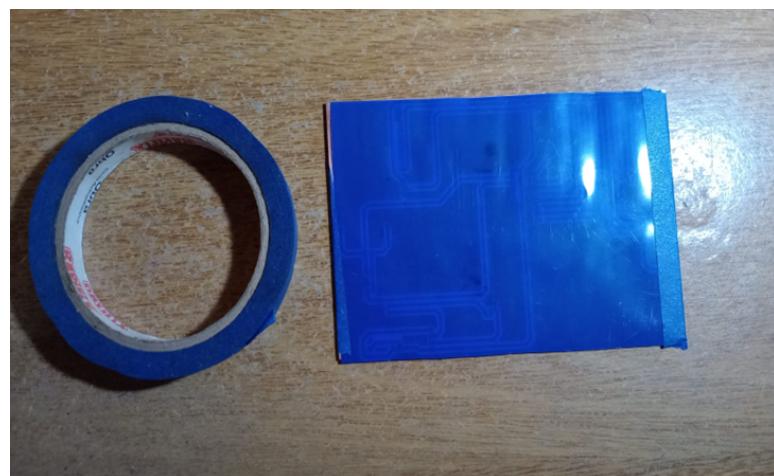


Figura 21: Preparación de placa. **Fuente:** Elaboración propia

1.2.7 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.



Figura 22: Transferencia térmica. **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.

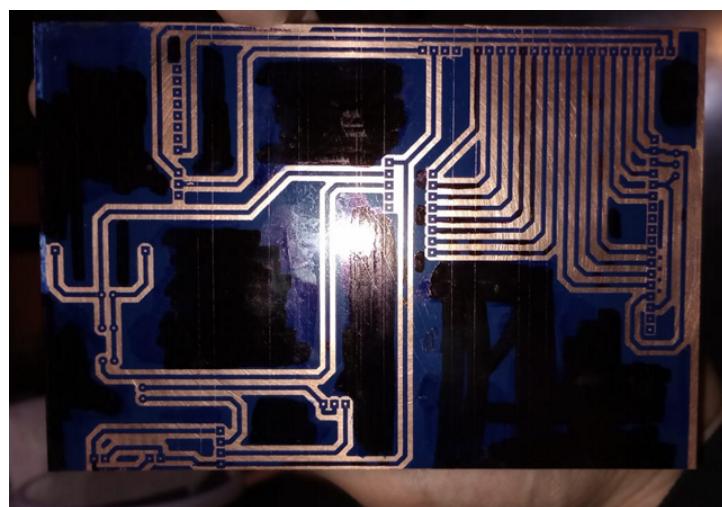


Figura 23: Repaso de tinta. **Fuente:** Elaboración propia

1.2.8 Cloruro férrico:

Una vez retocados los defectos de 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 éste 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 al descubierto serán eliminadas de la superficie de la placa.



Figura 24: Colocado de placa en Cloruro Férrico. **Fuente:** Elaboración propia

1.2.8.1 Resultado obtenido:

Se observan las pistas de cobre separadas por islas no conductoras.

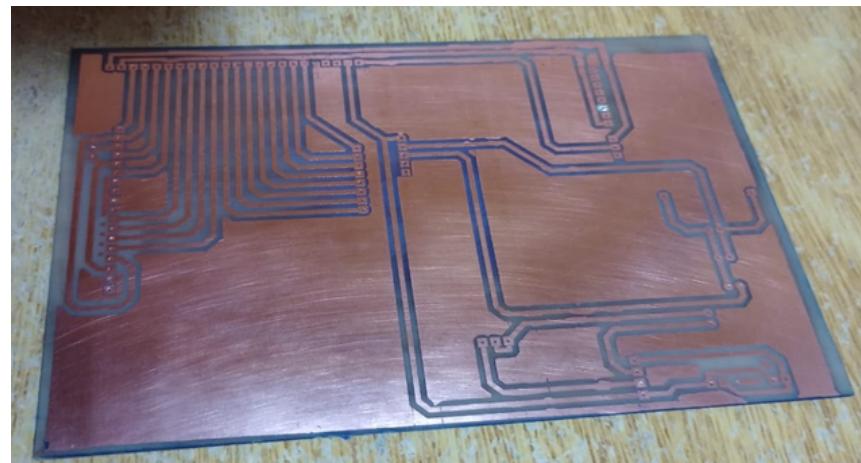


Figura 25: Circuito en placa. **Fuente:** Elaboración propia

1.2.9 Perforado de la Placa:

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



Figura 26: Perforado de Placa. **Fuente:** Elaboración propia

1.2.10 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.

1.2.10.1 Herramientas utilizadas:



Figura 27: Herramientas. **Fuente:** Elaboración propia

1.2.10.2 Resultados obtenidos:

1.2.10.2.1 Vista circuital:

Vemos el proceso de soldadura de los debidos componentes sobre el circuito impreso, utilizando estaño para la sujeción de los terminales.

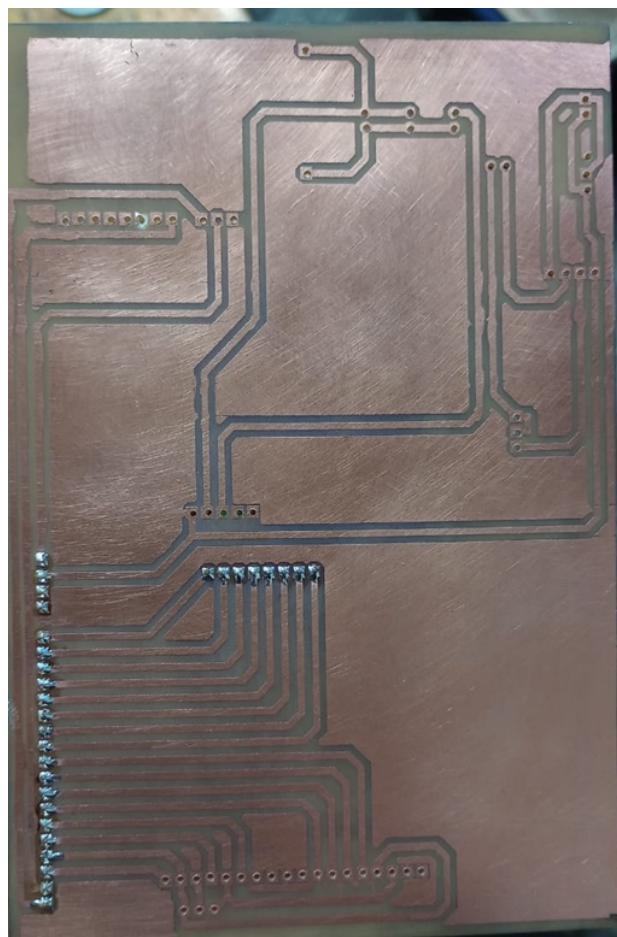


Figura 28: Proceso de soldadura. **Fuente:** Elaboración propia

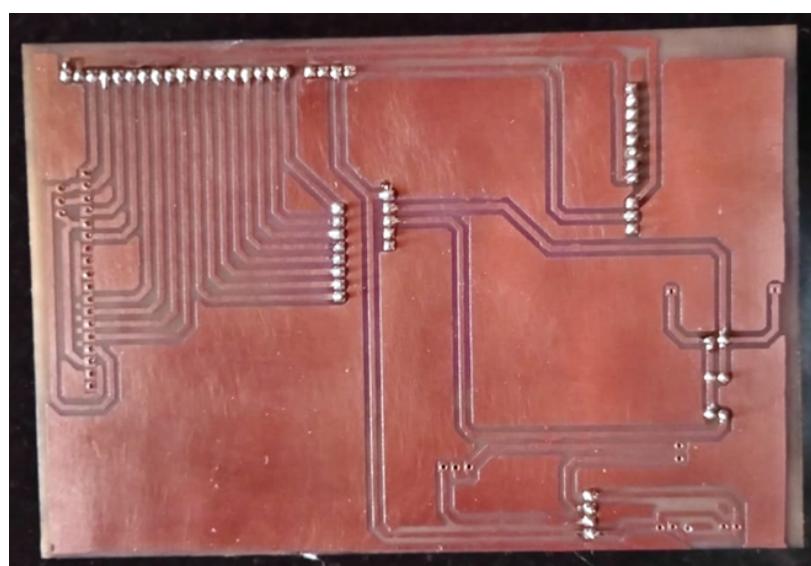


Figura 29: Proceso de soldadura. **Fuente:** Elaboración propia

1.2.10.2.2 Vista Superior:

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

- 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 Ráfaga de transmisión (2A Durante el ingreso de la llamada telefónica).
- Bornera electrónica, para sujetar los cables de la fuente de tensión utilizada para el 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.



Figura 30: Proceso en vista superior. **Fuente:** Elaboración propia

1.2.11 Conexionado:

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 **1.2.10.2.1.** Para ello se hará seguimiento del diagrama eléctrico (1.2.2). 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 lumínica del Display.
- Regulador de Tensión 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.



Figura 31: Conexionado. **Fuente:** Elaboración propia

1.2.11.1 Conexionado de Arduino:

Se debe realizar el conexionado 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)

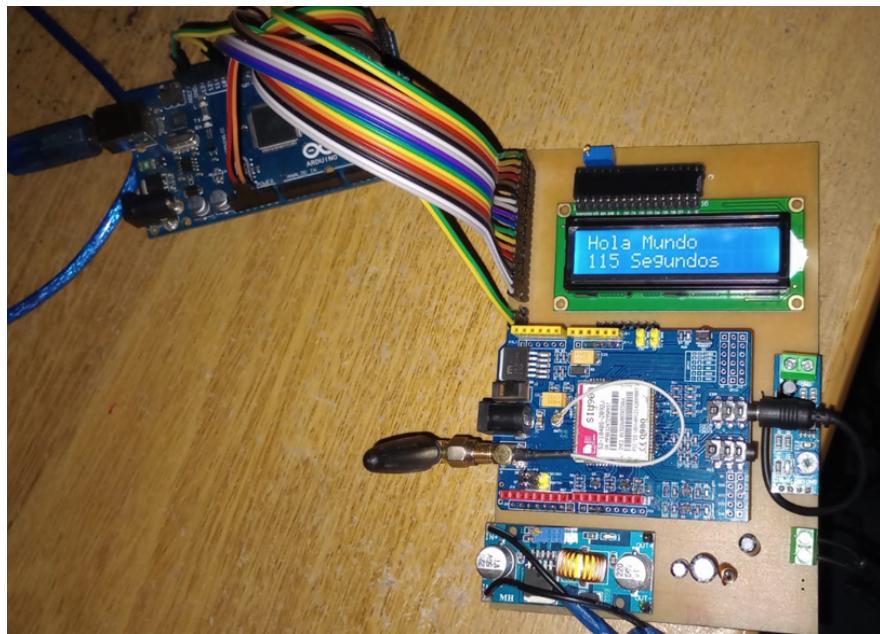


Figura 32: Vista general de funcionamiento. **Fuente:** Elaboración propia

1.2.12 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 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 dispuesto de la siguiente manera:

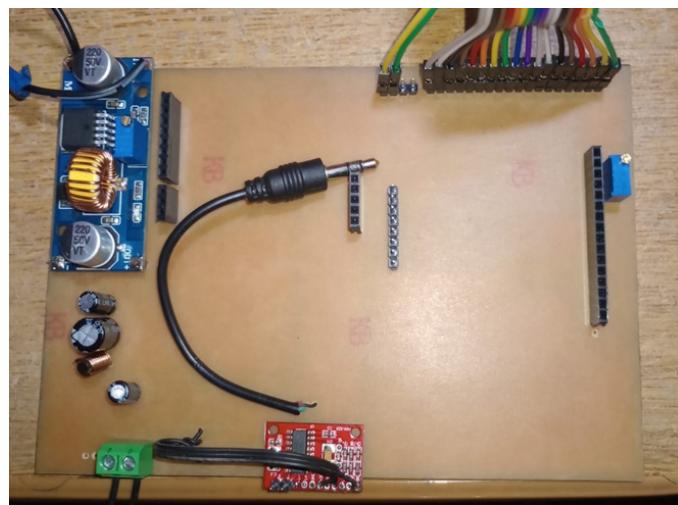


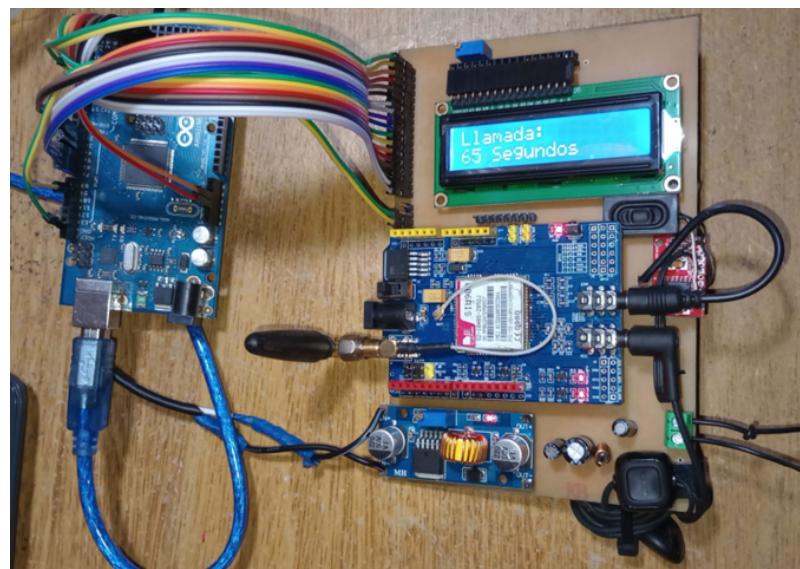
Figura 33: Reparación de Ruido Eléctrico. **Fuente:** Elaboración propia

Funcionamiento de la placa con sus componentes:



Figura 34: Funcionamiento de la placa con sus componentes.

Fuente: Elaboración propia

Conexionado de Placa fabricada con Módulo Arduino Mega:**Figura 35:** Conexionado de Placa con Módulo Arduino Mega.**Fuente:** Elaboración propia**Teclado utilizado para el Dispositivo:****Figura 36:** Teclado. **Fuente:** Elaboración propia

Etapa 1.3: Desarrollo del Software:

El desarrollo del Software está compuesto por todas las funcionalidades e instrucciones requeridas para el funcionamiento del Dispositivo Móvil. El proceso de creación, diseño y despliegue de software fue realizado utilizando la plataforma de Arduino IDE.

El proceso de desarrollo dio como resultado 4 ficheros que albergan todas las funcionalidades:

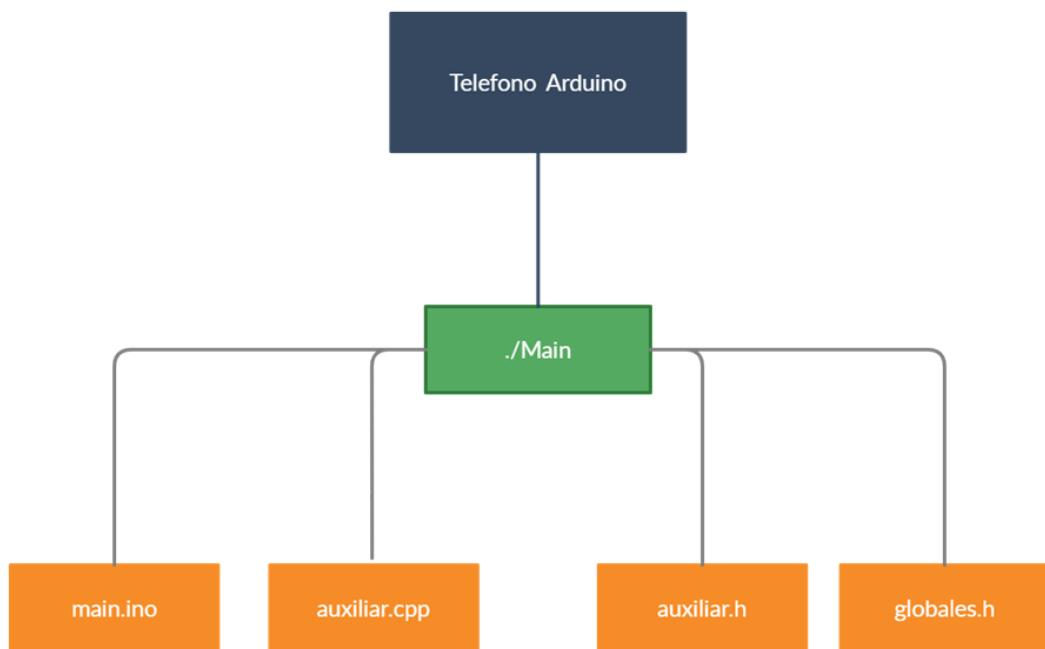


Figura 37: Árbol de ficheros. **Fuente:** Elaboración propia

La carpeta nombrada “**Main**” posee los 4 ficheros:

- **main.ino:** Posee las funciones de setup que realizan la configuración inicial del Dispositivo (**inicio()**). Además se definen las funciones de Loop o bucle, que se ejecutarán repetitivamente.

```
#include "auxiliar.h"
unsigned long tiempo_previo = 0, transcurrido = 0;
const long intervalo = 3000;

void setup() {
    inicio();
}

void loop() {
    teclado();
    transcurrido = millis();

    if (transcurrido - tiempo_previo >= intervalo)
    {
        tiempo_previo = transcurrido;
        oyente();
    }
}
```

Figura 38: Programa Main. **Fuente:** Elaboración propia

- **globales.h:** Posee todas las variables globales, que son accedidas por el resto de los ficheros. Incluye además las librerías utilizadas para teclado, display y módulo de comunicaciones.
- **auxiliar.h:** Nombra todas las funciones definidas en el fichero “auxiliar.cpp”, que serán accedidas por los distintos archivos.

```

//Configuracion inicial del Modulo:

void inicio ();
void encendido();
void standby();
void Configuracion();

// Funciones Loops a ejecutar indefinidamente:

void teclado();
void oyente();
void comando(String at, int unsigned tiempo);
void ShowSerialData();

// Funciones para Llamar y mandar Mensajes:

void llamado(String posicion);
void timmer(String posicion);
void mensaje(String num, String posicion);

// Funcion para conectarse a la red y realizar el envio de datos a la Nube:

void envio_trama(String posicion, String duracion);

```

Figura 39: Funciones definidas. **Fuente:** Elaboración propia

- **auxiliar.cpp:** Este archivo es el núcleo del proceso de desarrollo, y contiene todas las funcionalidades que definen las instrucciones a ejecutar por el dispositivo móvil.

1.3.1 Configuración Inicial:

Para comenzar a utilizar el dispositivo, se deben establecer los parámetros iniciales del dispositivo móvil. Todas estas funcionalidades están definidas en el archivo “**./Main/auxiliar.cpp**”.

```
void inicio();
```

Define la velocidad de transmisión del puerto serial en 9600 Baudios. Inicializa el display 16x02 y ejecuta la función de “Configuracion()”.

```
void encendido();
```

El módulo SIM900 posee encendido por Hardware de la Red (Pulsador onBoard).

Esta función permite el encendido por Software del módulo de comunicaciones SIM900L. Si esta función es ejecutada, dispara un pulso en alto por 1s, y luego un pulso en bajo por 5s para proceder con el encendido.

```
void standby();
```

Define el mensaje de inicialización mostrado en el Display 16x02.

```
void Configuracion();
```

Utiliza los comandos AT para modificar los parámetros del Módulo SIM900.

“AT+CLIP=1” = Activa el identificador de llamada, para las llamadas recibidas.

“AT+IFC=1,1” = Activa el Retorno de datos por Software.

“AT+CMIC=0,7 = Establece la Ganancia del micrófono principal en un rango de (0-15).

“AT+CALS=4” = Tono de llamada deseado entre los 20 preexistentes.

"AT+CLVL=80" = Volumen del parlante en un rango de 0 a 100.

"AT+CRSL=1" = Volumen de Llamada en un rango de 0 a 4.

1.3.2 Funciones Bucle:

Se definen las funciones que se ejecutarán en el loop principal indefinidamente. Todas estas funcionalidades están definidas en el archivo “**./Main/auxiliar.cpp**”.

```
void teclado();
```

Realiza una lectura constante de teclado, para leer cualquier ingreso por parte del usuario y realizar la lógica correspondiente.

Botones de Teclado:

Además de números, los botones que posee el teclado son los siguientes



: Enciende el Módulo de Comunicaciones.



: Cuelga la Llamada.



: Atiende la llamada.



: Permite guardar contactos en la Sim Card (Hasta 250 Contactos).



: Acceder a la agenda de contactos para realizar una llamada.



: Acceder a la mensajería, para enviar textos pre guardados.

Figura 40: Botones en teclado. **Fuente:** Elaboración propia

```
void oyente();
```

Realiza una lectura constante del estado del dispositivo móvil, para detectar una posible llamada o mensaje entrante. Posee toda la lógica de la lectura e identificador de llamada.

"AT+CPAS" = El comando AT utilizado lee el estado del teléfono, y retorna 4 valores posibles:

- 0 = Listo para cualquier acción.
- 2 = Estado desconocido o erróneo.
- 3 = Sonando.
- 4 = Llamada en curso.

```
void comando(String at, int unsigned tiempo);
```

Define una librería propia, para la ejecución de los comandos AT.

```
void ShowSerialData();
```

Escribe y transmite cualquier parámetro detectado en el puerto Serial de Software (Pines 7 y 8 del Módulo SIM900L), al puerto serial de Hardware (Pines de Arduino TX1 y RX1).

1.3.3 Funciones Principales:

Se definen las funciones principales para realizar llamadas, mensajes y envío de datos a la nube. Todas estas funcionalidades están definidas en el archivo “**./Main/auxiliar.cpp**”.

```
void llamado(String posicion);
```

Realiza la ejecución de la llamada, según el contacto elegido por parte del usuario. Una vez establecida la llamada, realiza la ejecución de la función “**timmer()**”.

```
void timmer(String posicion);
```

Se define un cronómetro propio para detectar la duración de la llamada. Luego de finalizar la llamada, realiza el llamado y ejecución de la función “**envio_trama()**” para enviar el dato correspondiente a la nube.

```
void mensaje(String num, String posicion);
```

Realiza el envío de mensajes según una estructura definida, al número o contacto ingresado por parte del usuario. La plantilla de mensajes pregrabados será mostrada al usuario para elegir el deseado.

```
void envio_trama(String posicion, String duracion);
```

Realiza el registro de la red y obtención de la IP brindada por la operadora. Luego realiza el envío del dato requerido, compuesto por la duración de la llamada recibida o realizada por parte del usuario + El identificador del destinatario. Este proceso será explicado detalladamente en el apartado “**4.4**”.

Etapa 1.4: 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”.

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.

1.4.1 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”:

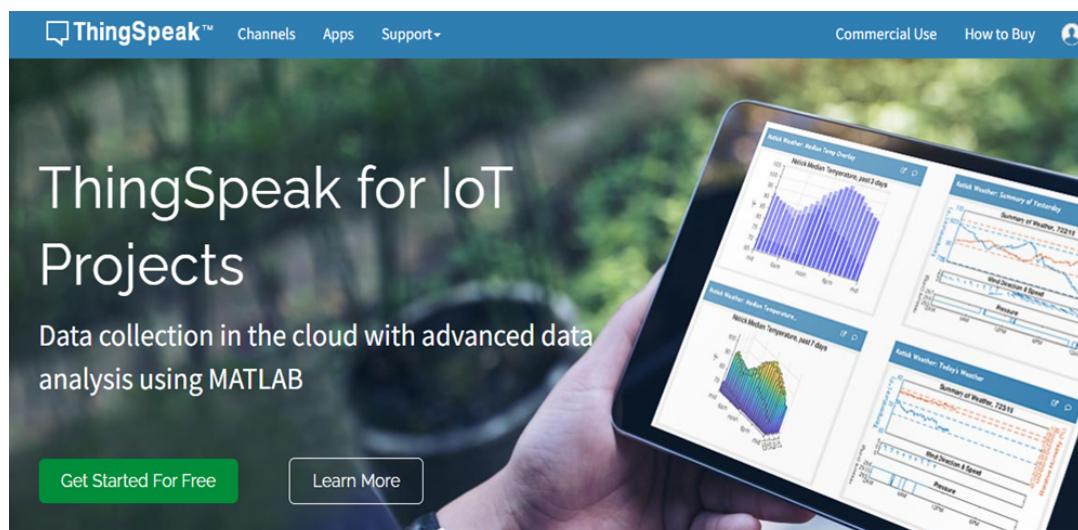


Figura 41: Registro en plataforma IoT. **Fuente:** <https://thingspeak.com/>

2. Elegir la opción “Create one!”:

To use ThingSpeak, you must sign in with your existing MathWorks account or create a new one.

Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evaluation. To get full access to the MATLAB analysis features on ThingSpeak, log in to ThingSpeak using the email address associated with your university or organization.

To send data faster to ThingSpeak or to send more data from more devices, consider the [paid license options](#) for commercial, academic, home and student usage.

Diagram illustrating the IoT architecture:

```

graph LR
    Devices[SMART CONNECTED DEVICES] --> Router[Router]
    Router --> Cloud[DATA AGGREGATION AND ANALYTICS  
ThingSpeak]
    Cloud <--> MATLAB[ALGORITHM DEVELOPMENT  
SENSOR ANALYTICS]
  
```

The diagram shows a network of smart connected devices connected to a central router, which then connects to a cloud-based data aggregation and analytics platform (ThingSpeak). The platform also connects to a MATLAB interface for algorithm development and sensor analytics.

Figura 42: Registro en plataforma IoT.

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

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

To use ThingSpeak, you must sign in with your existing MathWorks account or create a new one.

Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evaluation. To get full access to the MATLAB analysis features on ThingSpeak, log in to ThingSpeak using the email address associated with your university or organization.

To send data faster to ThingSpeak or to send more data from more devices, consider the [paid license options](#) for commercial, academic, home and student usage.

Create MathWorks Account

Email Address:

To access your organization's MATLAB license, use your school or work email.

Location:

First Name:

Last Name:

Continue

Cancel

Diagram illustrating the IoT architecture:

```

graph LR
    Devices[SMART CONNECTED DEVICES] --> Router[Router]
    Router --> Cloud[DATA AGGREGATION AND ANALYTICS  
ThingSpeak]
    Cloud <--> MATLAB[ALGORITHM DEVELOPMENT  
SENSOR ANALYTICS]
  
```

The diagram shows a network of smart connected devices connected to a central router, which then connects to a cloud-based data aggregation and analytics platform (ThingSpeak). The platform also connects to a MATLAB interface for algorithm development and sensor analytics.

Figura 43: Registro en plataforma IoT.

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.

1.4.2 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**).

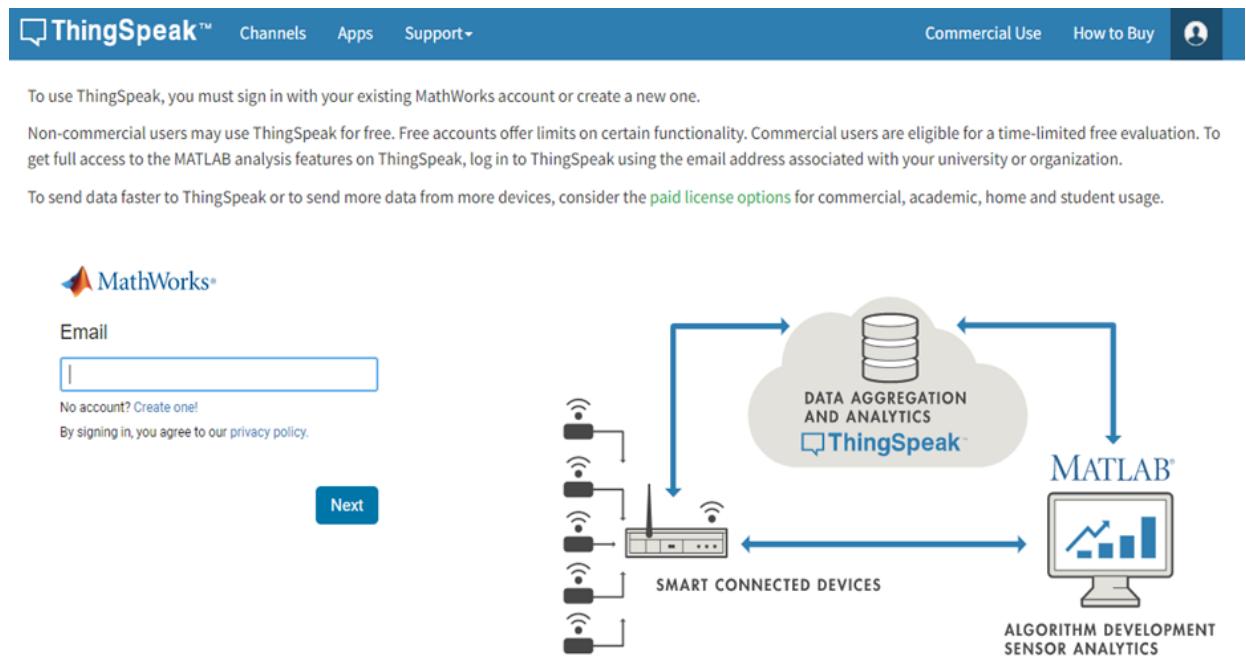


Figura 44: Inicio de Sesión.

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.

1.4.3 Crear Canal:

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”:

Name	Created	Updated
Duración de Llamada	2022-04-15	2022-05-20 15:36

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

Figura 45: Crear canal.

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”.

ThingSpeak™ Channels Apps Devices Support Commercial Use How to Buy JC

Name	Duración de Llamada	
Description	Ingreso y Egreso de Llamadas de Pedro	
Field 1	Javier	<input checked="" type="checkbox"/>
Field 2	Marcos	<input checked="" type="checkbox"/>
Field 3	Juan	<input checked="" type="checkbox"/>
Field 4	Pedro	<input checked="" type="checkbox"/>
Field 5	Lucas	<input checked="" type="checkbox"/>
Field 6	Mateo	<input checked="" type="checkbox"/>
Field 7	Simón	<input checked="" type="checkbox"/>
Field 8	Números Desconocidos	<input checked="" type="checkbox"/>

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

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.

Figura 46: Configurado de canal.

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

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

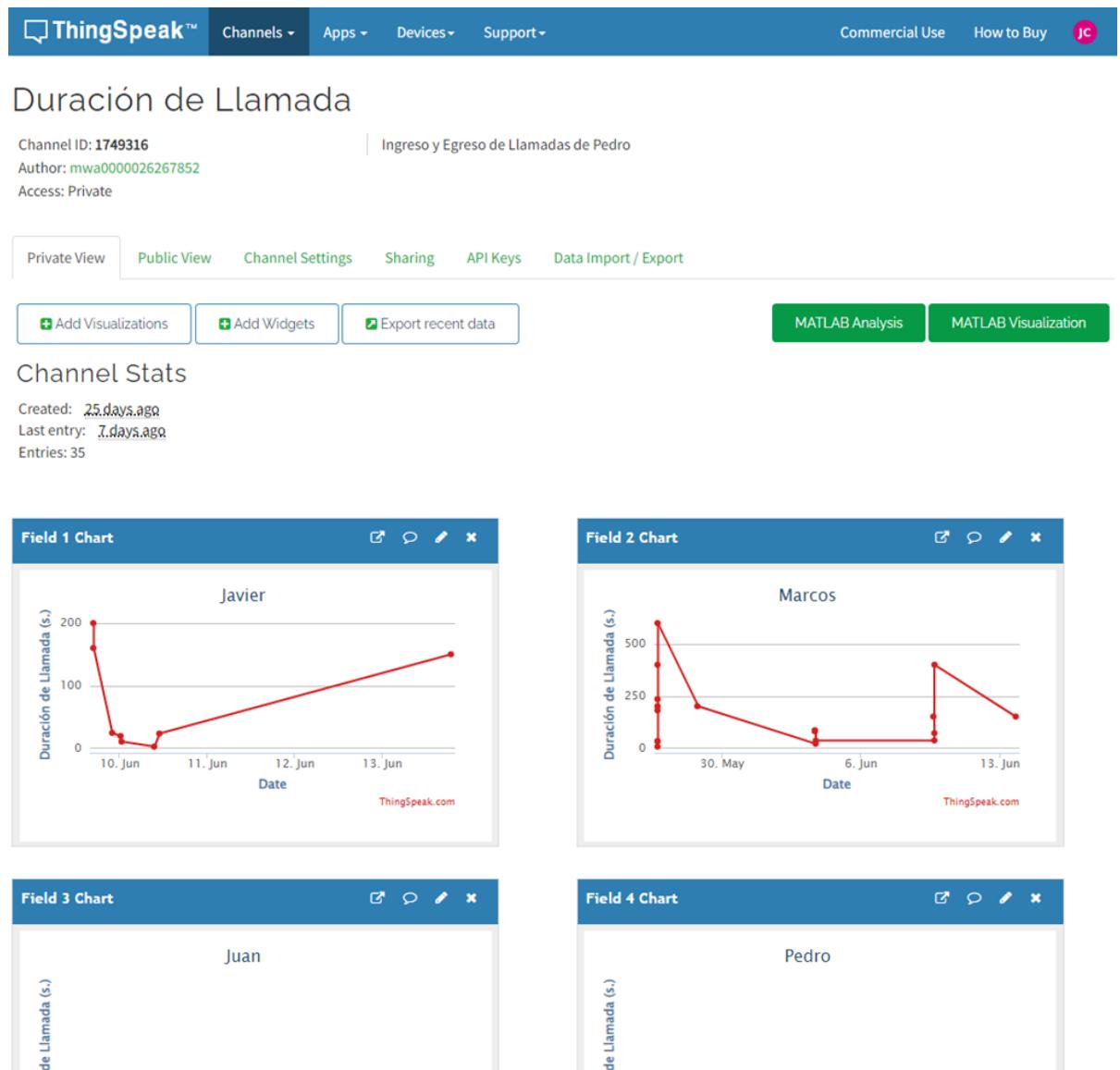


Figura 47: Vista principal.

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

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

Duración de Llamada

Channel ID: **1749316**
 Author: **mwa0000026267852**
 Access: Private

Ingreso y Egreso de Llamadas de Pedro

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Write API Key

Key

96MTTXG33GMWMI2Q

Generate New Write API Key

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- **Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click [Generate New Write API Key](#).
- **Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click [Generate New Read API Key](#) to generate an additional read key for the channel.

Figura 48: Configurado de API Key.

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

1.4.4 Código Arduino para la Nube:

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!...");
    Serial.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);
}

```

Figura 49: Código de Arduino para la nube. **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"](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:



Figura 50: Gráfico de duración de llamada por destino.

Fuente: Elaboración propia.

Etapa 1.5: 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.

1.5.1 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:

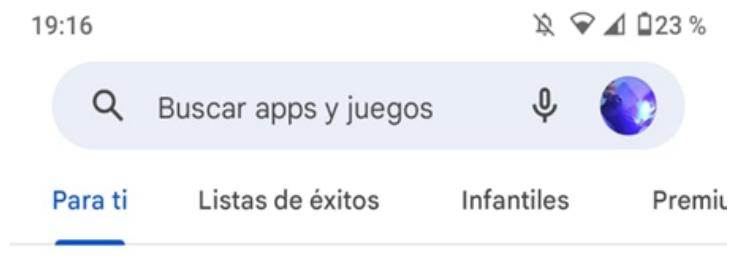


Figura 51: Busqueda de Aplicación. **Fuente:** Elaboración propia.

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



Figura 52: Busqueda de Aplicación. **Fuente:** Elaboración propia.

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

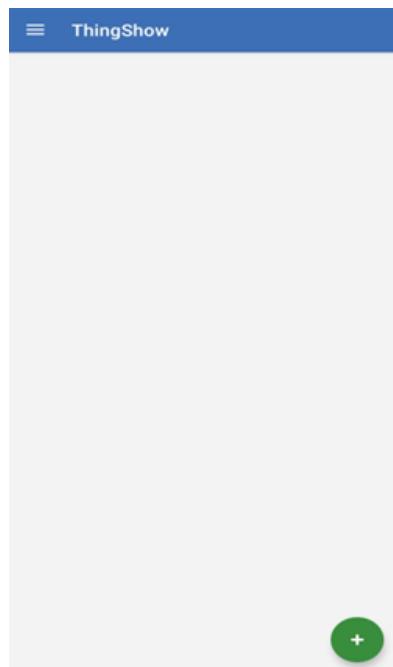


Figura 53: Inicio de Aplicación. **Fuente:** Elaboración propia.

4. Seleccionar el tipo de canal Privado:

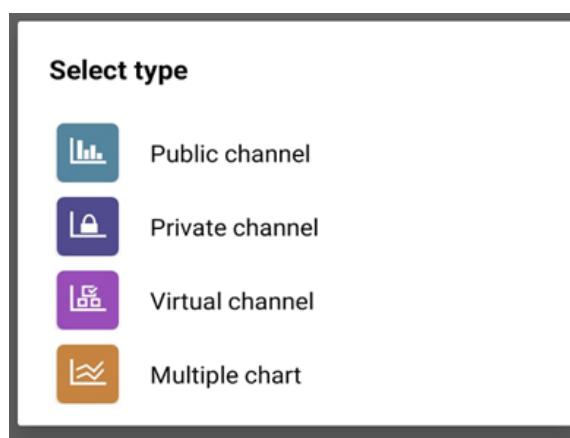


Figura 54: Configurado de canal. **Fuente:** Elaboración propia.

5. Ingresar las credenciales obtenidas en la Etapa 4.3.
 1. Channel ID: 1749316
 2. API Key: 96MTTXG33GMWMI2Q

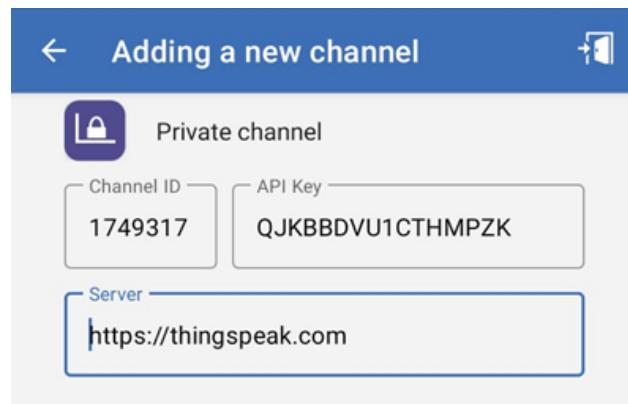


Figura 55: Configurado de canal. **Fuente:** Elaboración propia.

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

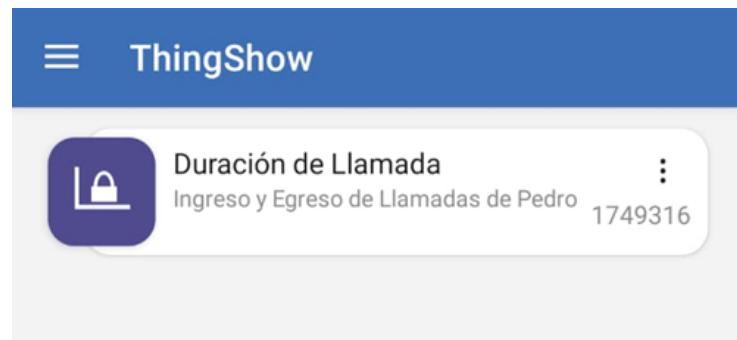


Figura 56: Canal configurado. **Fuente:** Elaboración propia.

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



Figura 57: Vista de app de seguimiento. **Fuente:** Elaboración propia.

Capítulo 2: Análisis de Resultados

Al finalizar los sprints del proyecto desarrollado, en el que se incluyó la fabricación del dispositivo móvil, implementación de plataforma IoT y configuración de la app móvil, se obtuvo el cumplimiento de los objetivos planteados al iniciar dicho proyecto. Para el desarrollo del producto mínimo viable se hizo uso de metodologías ágiles.

Para concluir se desarrolló de manera eficaz, un dispositivo móvil especialmente diseñado para personas de la tercera edad y de bajos recursos. Las funcionalidades logradas fueron:

1. Realizar y recibir llamadas
2. Mensajería móvil.
3. Registrar contactos Telefónicos
4. Uso Agenda de contactos.
5. Conexión a la red para transmitir datos de seguimiento a la nube

Resultado de la fabricación del Hardware:

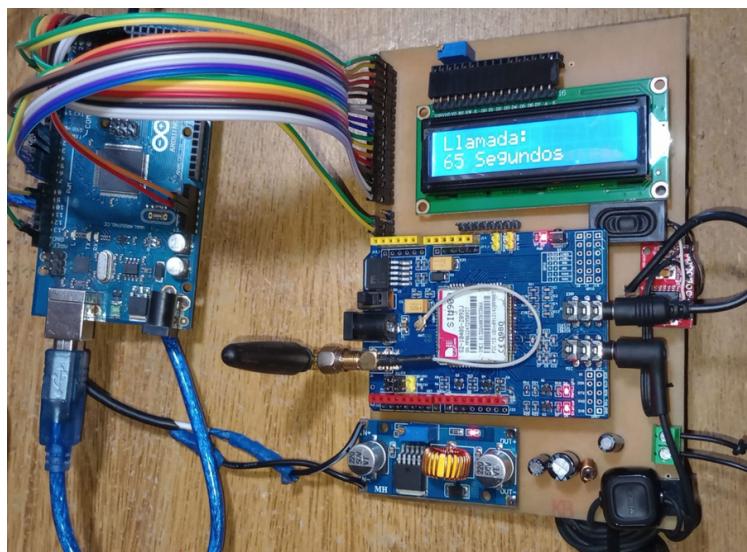


Figura 58: Resultado de la fabricación del Hardware.

Fuente: Elaboración propia.

Esquema de desarrollo de Software:

El desarrollo posee todas las funcionalidades del dispositivo móvil, realizando la codificación en la plataforma de Arduino IDE.

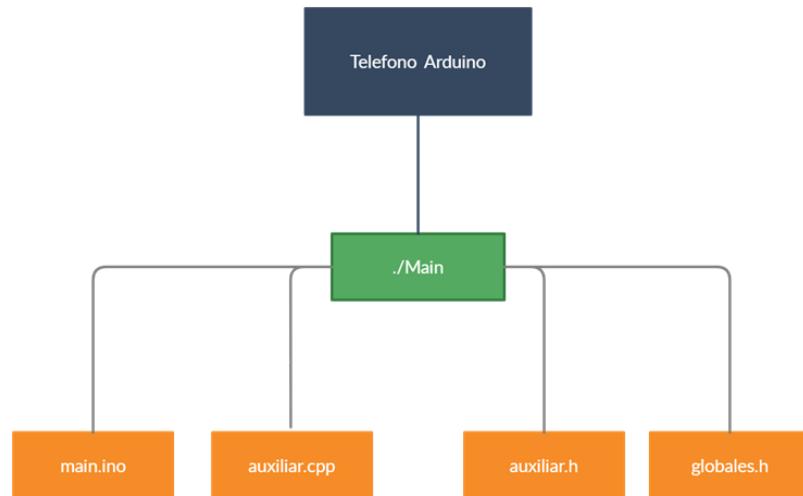
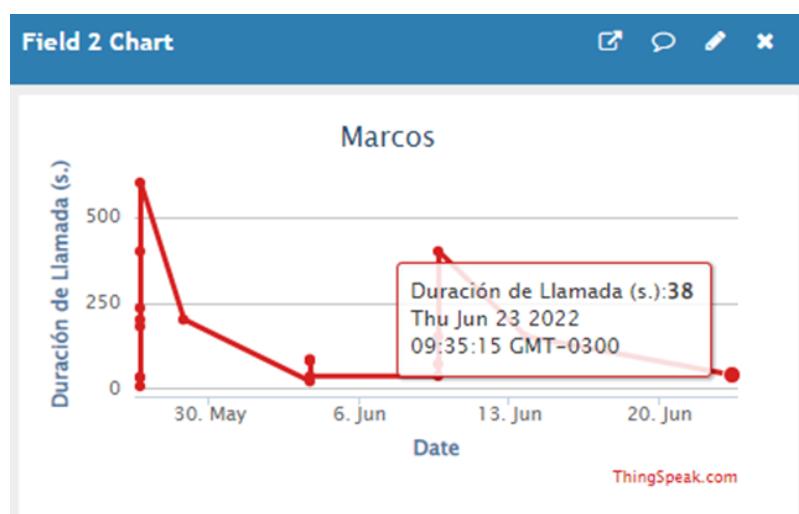


Figura 59: Resultado del desarrollo de software.

Fuente: Elaboración propia.

Datos obtenidos en la nube:

A través de la plataforma IoT, se obtiene el registro de llamadas de cada contacto telefónico con el que el usuario establece una llamada telefónica. El acceso a la plataforma, permitirá un seguimiento diario del ingreso y egreso de llamadas



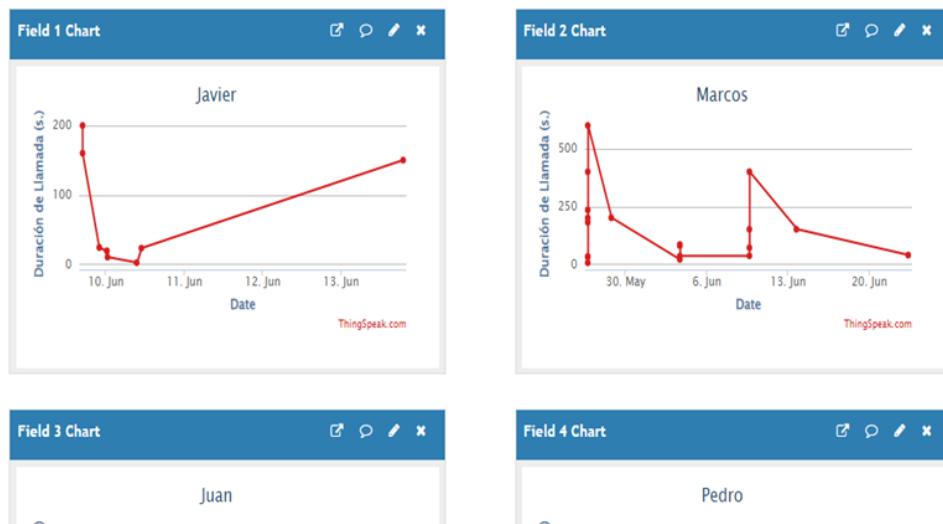


Figura 60: Resultados obtenidos en la nube. **Fuente:** Elaboración propia.

Aplicación Móvil:

A través de la aplicación móvil, se podrá realizar un seguimiento del dispositivo con las mismas funcionalidades que posee la plataforma web, pero agregando agilidad en la consulta, gracias a que la app móvil está disponible para cualquier sistema operativo móvil.



Figura 61: Resultados obtenidos en la App. **Fuente:** Elaboración propia.

Conclusiones:

Luego de analizar los resultados obtenidos en la etapa anterior, se aseguró la consonancia con los objetivos planteados en el marco referencial, obteniendo el 100% de los objetivos establecidos en la fabricación del dispositivo móvil. El producto se encuentra funcionando, habiendo realizado sus pruebas pertinentes.

Se alcanzó el objetivo general:

Diseñar un Dispositivo móvil basado en Arduino de fácil manejo para el usuario, desarrollado especialmente para la tercera edad.

Objetivos Específicos alcanzados:

- Realizar los diagramas y conexión de todo el hardware necesario para la realización del teléfono móvil.
- Realizar el diseño e implementación del Backend.
- Integrar los ejecutables generados en el lenguaje de programación elegido.
- Enlazar el Módulo de Comunicaciones con la nube, para mostrar y consultar los datos generados por el dispositivo, sobre el ingreso y egreso de llamadas telefónicas.
- Acceder a una aplicación móvil, para realizar el seguimiento del usuario.
- Realizar el testeo de lo implementado.

Mejoras a futuro:

- Fabricación de carcasa.
- Desarrollo de manual de usuario.
- Agregado de rastreo móvil.
- Pantalla de mayor tamaño.

Anexos:**1. Repositorio del proyecto:**

Github es una herramienta online y libre, que permite gestionar proyectos y controlar versiones de código. El repositorio de Github del proyecto en el cual se encuentran todos los archivos generados es:

https://github.com/javicercasi/Trabajo_Final_3

2. Tablero Kanban:

Un tablero de kanban es una herramienta ágil de gestión de proyectos diseñada para ayudar a ordenar el trabajo y limitar el trabajo que se encuentra en curso. El link del tablero es:

<https://kb.cloud.um.edu.ar/board/4>

Fuentes Bibliográficas:

Bibliografías y Referencias Bibliográficas:

- Bolognesi, Andres. (2013). “Sitio web dinámico aplicado a las nuevas tecnologías móviles”. Primera Edición, Argentina.
- SIMCom Wireless Solutions. (2013). “SIM900 Hardware Design”. Segunda Edición, China.
- SIMCom Wireless Solutions. (2010). “SIM900 AT Command Manual”. Primera Edición, China.
- Ms. N. Rajeswari. (2019). “CLOUD COMPUTING AND ITS TYPES”. Primera Edición, India.
- Espacio Fundación Telefónica Madrid. (2012). “Historia de las Telecomunicaciones”. Primera Edición, España.
- Lic. Gabriela Hadid. (2017). “Posibilidades y límites en el uso de las tecnologías: las personas mayores”. Primera Edición, Argentina.
- Álvaro Primo Guijarro. (2012). “Protocolo HTTP”. Primera Edición, España.
- Ruben Loureiro Garrido. (2015). “Estudio Plataformas IoT”. Primera Edición, Argentina.
- Pablo Andrés Salinas Rojas. (2008). “Manual Proteus Layout Editor”. Primera Edición, Colombia.
- Texas Instruments. (2017). “LM386 Low Voltage Audio Power Amplifier Datasheet”. Primera Edición, Estados Unidos.

Sitios Consultados:

- Arduino CC. (2018). “Arduino Mega 2560 Rev3”.
<https://store.arduino.cc/products/arduino-mega-2560-rev3>
- Arduino. (2018). “What is Arduino?”.
<https://www.arduino.cc/en/Guide/Introduction>
- Microsoft Azure. (2020). “¿Qué es una nube pública?”.
<https://azure.microsoft.com/es-es/overview/what-is-a-public-cloud/>
- Christopher Holloway. (2018). “Plataformas IoT: Qué son y cómo elegir la mejor para el negocio”.
<https://www.itmastersmag.com/noticias-analisis/plataformas-iot-que-son-y-como-elegir-la-mejor-para-el-negocio/>
- MathWorks. (2022). “ThingSpeak Documentation”.
<https://www.mathworks.com/help/thingspeak/>