



**Ingeniería en Telecomunicaciones**

**“Sistema Prototipo de Monitoreo y registro del Nivel del río  
integrado a IoT mediante tecnología GSM/GPRS”**

Lisandro Raul De La Madrid

Universidad Autónoma de Entre Ríos

Facultad de Ciencia y Tecnología

Profesor a Cargo:

Mario Farina

Concepción del Uruguay, Entre Ríos, Argentina

2023

<b>Introducción:</b> .....	<b>4</b>
<b>Situación Problemática:</b> .....	<b>5</b>
<b>Hipótesis:</b> .....	<b>5</b>
<b>Metodología:</b> .....	<b>5</b>
<b>Objetivos:</b> .....	<b>6</b>
Objetivo General:.....	6
Objetivo Específicos:.....	6
<b>Estado del Arte:</b> .....	<b>6</b>
<b>Marco Teórico:</b> .....	<b>8</b>
Desastres Naturales:.....	8
Inundación:.....	9
Internet de las Cosas (IoT):.....	10
Aplicaciones del IoT:.....	11
Tecnologías de Comunicación Inalámbricas utilizadas en IoT:.....	12
Sistema Global para las Comunicaciones Móviles (GSM):.....	13
Arquitectura GSM:.....	15
Servicio General de Paquetes vía Radio (GPRS):.....	17
Comandos AT:.....	18
<b>Planificación:</b> .....	<b>21</b>
Elección del Hardware:.....	21
Elección del Software:.....	22
Pruebas y Mediciones del Hardware:.....	22
Montaje:.....	22
<b>CRONOGRAMA:</b> .....	<b>23</b>
<b>Desarrollo:</b> .....	<b>24</b>
Descripción de dispositivos que componen el sistema:.....	24
Microcontrolador ATmega328p:.....	28
Sensor HC-SR04:.....	29
Modulo GSM/GPRS Sim800L:.....	33
Display 16x2 con adaptador I2C:.....	35
ThingsPeak:.....	38
<b>Experimentación y diseño:</b> .....	<b>39</b>
Descripción del Dispositivo:.....	39
Arduino UNO R3 y HC-SR04:.....	40
Pruebas con HC-sr04:.....	40
Cálculo de la Distancia:.....	43
Limitaciones del HC-SR04:.....	44
Implementación del Módulo Sim800L:.....	44
Pruebas con Sim800L:.....	44
Comunicación Serial Asíncrona:.....	45
Alimentacion del Proyecto:.....	47
<b>Montaje del Hardware:</b> .....	<b>48</b>
Carcasa del dispositivo:.....	48

<b>Diagramas de Flujo:</b> .....	<b>50</b>
Programa Principal:.....	50
Diagrama de las funciones:.....	51
Diagrama de Alerta():.....	52
Diagrama de EnvioDatos():.....	53
Diagrama de Distancia():.....	54
Diagrama de Signal():.....	56
<b>Diseño del Software:</b> .....	<b>56</b>
Arduino IDE:.....	56
Funciones del Programa:.....	57
<b>Presupuesto:</b> .....	<b>65</b>
<b>Resultados:</b> .....	<b>69</b>
<b>Propuestas de modificaciones futuras:</b> .....	<b>70</b>
<b>Bibliografía:</b> .....	<b>73</b>
<b>Anexo:</b> .....	<b>74</b>
Codigo Completo:.....	74

## **Introducción:**

La variable climática y el proceso de cambio son reconocidos globalmente como las causas principales del incremento en la frecuencia e intensidad de los eventos catastróficos, siendo uno de los factores principales de las inundaciones, preocupando mucho al hombre, ya que son factores incontrolables y que pueden generar desastres de grandes magnitudes. Sin embargo, estos eventos se pueden registrar y monitorear para preparar a los que serían los afectados ante cualquier cambio radical de dicho fenómeno. Por lo tanto, específicamente, ante las amenazas de carácter hidrológico, se requiere especial atención al comportamiento de los ríos, ya que durante la temporada de lluvias, estos suelen crecer a veces inesperadamente, trayendo como consecuencias inundaciones impredecibles que a su vez provocan tanto pérdidas de vidas humanas como pérdidas económicas, entre las comunidades que habitan en las cercanías al río.

Es por eso que el presente informe presenta información sobre el diseño un dispositivo Prototipo de monitoreo del nivel del río, donde por medio de un sensor, se realizará adquisición de datos con respecto a los niveles del río y luego serán procesados para ser enviados mediante comunicación GSM/GPRS a una aplicación de análisis IoT que permite agregar, visualizar, analizar y actuar sobre esos flujos de datos en directo con la nube.

## **Situación Problemática:**

El aumento del nivel del río es una de las principales problemáticas ambientales en ciudades portuarias y costeras, causando una gran cantidad de pérdidas humanas así como de pérdidas materiales generando grandes pérdidas económicas y dejando a familias sin hogares o en climas de extrema precariedad.

En la actualidad existen Organismos que se encargan de mantener un monitoreo y registro constante a través de la utilización de diferentes dispositivos y herramientas, obteniendo información cada cierto tiempo de parámetros que mantendrán alerta a las poblaciones afectadas por estos fenómenos climáticos.

Informar a la población sobre estos, aun teniendo los elementos necesarios para detectarlos, se les dificulta, por eso es importante llevar a cabo un control correcto.

A partir de esto surge la necesidad de investigar y diseñar un Sistema de Monitoreo y Registro del Nivel del Río para zonas donde se producen desbordamientos del río, permitiendo actuar sobre la información recopilada, dando soluciones a la problemática ya nombrado.

Evaluando la situación, surgen los siguientes interrogantes:

- ¿Qué se debe tener en cuenta para desarrollar un sistema de monitoreo?
- En caso de saber como desarrollarlo ¿Se podrá realizar un prototipo de respuesta rápida y económica?
- ¿Es posible llevar un registro de estos fenómenos?
- ¿Qué dispositivos son necesarios?
- ¿Es posible diseñar un dispositivo escalable?

## **Hipótesis:**

El reciente avance de las tecnologías se presenta como una gran ayuda para la detección temprana de desastres naturales, permitiéndole a las poblaciones afectadas actuar frente a esta problemática evitando una mayor cantidad de pérdidas, tanto humanas como materiales.

## **Metodología:**

Para la realización de este proyecto es necesario aclarar que se realizó una investigación de tipo descriptiva y exploratoria con un enfoque cuantitativo, ya que se

realizaron ensayos de prueba del prototipo de manera controlada, realizando un control de los efectos de cada prueba.

En primer lugar se fue investigando y analizando las opciones tecnológicas de dispositivos para poder realizar el prototipo, y luego se fue recopilando información similar a modo de comparación para tener noción de antecedentes de proyectos similares.

A partir de la obtención de información acerca del tema, se comenzó a obtener cada dispositivos y a realizar pruebas de cada uno de manera controlada, analizando cada lectura y funcionamiento de los mismos. Una vez realizado esto se prosiguió con el armado del prototipo, y luego con sus respectivas pruebas para comprobar si se obtenía el resultado esperado.

## **Objetivos:**

### Objetivo General:

- Investigar y diseñar un dispositivo prototipo de medición del nivel del río para monitorear y registrar, utilizando módulos electrónicos y tecnología GSM/GPRS.

### Objetivo Específicos:

- Diseñar un equipo prototipo.
- Construir un prototipo funcional.
- Realizar las pruebas correspondientes para corroborar su correcto funcionamiento.
- Recopilar y Mostrar los datos a través de una aplicación IoT.
- Realizar un equipo escalable para futuras actualizaciones
- Utilizar la Red telefónica GSM/GPRS para informar de cualquier alerta o peligro.

## **Estado del Arte:**

A continuación en esta sección se buscará explicar cuál es la situación en la actualidad del tema planteado en el presente proyecto de investigación final.

Actualmente la utilización de sistemas con sensores provocó un gran avance en desarrollos de diferentes áreas como la médica, industrial, económica, ambientales, entre

otras, trayendo un gran cantidad de beneficios, entre estos se destaca la rapidez con la cual se puede obtener información, proceder sobre ella y suministrar en el tiempo necesario para poder reaccionar frente a adversidades, como por ejemplo el cambio climático, con el objetivo de reducir la magnitud de daño que pueden causarle a la sociedad en general.

Según el documento de investigación realizados por Dilma Dávila, “Estado del arte de los Sistemas de Alerta Temprana (SAT) ante inundaciones en América Latina” (2018) asegura que en la actualidad el 20% de estos sistemas se utilizan de acuerdo a su modalidad con un fin comunitario con participación activa de la población, el 40% son centralizados o instrumentales dependiendo del monitoreo automatizado o convencional, y otro 40% es mixto, dejando en claro que se puede utilizar en múltiples modalidades. La utilización de dispositivos y sensores para un sistema de monitoreo y registro depende de las características particulares de los eventos o amenazas, de su ubicación geográfica, y de los recursos disponibles.

Por otro lado añadiendo información a lo antes dicho, proveniente del Trabajo de investigación de Jose Rigoberto Miranda “Sistema de alerta temprana por riesgo de inundaciones debidas a desbordamiento de los niveles de ríos” (2017) de la Universidad del Salvador, afirma que en el caso de los sistemas automatizados se utilizan instrumentos sofisticados o tecnológicos como satélites, sensores remotos, redes telemétricas y otros que permiten transmitir información directa desde los equipos de medición hasta los centros de análisis y toma de decisión. En cuanto a los comunitarios tienen aplicaciones en cuencas medianas y pequeñas; son fáciles de manejar, ya que sus instrumentos son básicos y no requieren de técnicas especializadas, sus recursos y funciones son limitados. Con estos sistemas las comunidades identifican sus riesgos, aumentan sus capacidades para enfrentar emergencias y reducen la posibilidad de daños y pérdidas.

Estos sistemas a su vez se utilizan de acuerdo a su diseño, con un 31% para gobiernos locales, 25% asociado con cooperación, 18% para institutos científicos tecnológicos, 13% Universidades y el 2% para el sector privado.

Se puede decir que el objetivo fundamental es evitar la posibilidad que se produzcan lesiones personales, pérdida de vidas, daños a los bienes, mediante la emisión de alertas y la pronta aplicación de medidas de protección y reducción de riesgos.

Para que ocurra lo dicho anteriormente debe cumplir un cierto funcionamiento que debe seguir una serie de pasos; lectura y registro de medición de los instrumentos sobre el evento monitoreado; transmisión de los datos registrados; procesamiento y análisis de los datos transmitidos; pronósticos de la situación; establecimiento del nivel y tipo de alerta; difusión del nivel de alerta; activación de un Plan de Emergencias.

Como dato no menor se podría decir que una novedad en la actualidad es la adopción de Smartphones o celulares inteligentes al igual que aplicaciones o APIs de

código abierto a estos Sistemas de monitoreo, proporcionando una plataforma tanto de control como de alerta de bajo costo, como en algunas ocasiones, una solución final. Por ejemplo, las aplicaciones que dan mensajes de alertas o realizan una actualización del estado del dispositivo, también APIs o nubes como Thingspeak que permite registrar y verificar datos de forma continua. Los Smartphone dan una interfaz común para el procesamiento de datos, dando igual la manera y la forma en la que se obtiene las medidas mediante el uso de sensores, siendo entendible tanto para el entorno profesional o entorno centrado en ciudadanos.

Un dato importante a tener en cuenta es su robustez, muchos de los sensores que componen estos Sistemas están expuestos a condiciones extremas. En primer lugar los sensores y por otro lado la electrónica a bordo necesitan protegerse de la lluvia, hielo, polvo y otras fuentes que pueden dañar el dispositivo. Para poder protegerlo de manera correcta se utilizan contenedores especiales, permitiendo trabajar a los sensores con una sensibilidad y fiabilidad adecuada. Diciendo esto se puede decir que las capacidades y sensibilidad de los sensores siguen siendo algo limitados. Se hace necesario el uso de protocolos de calibración cuidadosos y continuos para mejorar el rendimiento de algunos sensores, pero aún no se ha logrado un progreso significativo.

En conclusión se podría decir que al ir aumentando la preocupación del medio ambiente, en específico, el aumento del nivel del río, cada vez se le está dando mayor atención a los proyectos de la sociedad científica y tecnológica. Son muchos los esfuerzos de la comunidad para poder controlar variables del medio ambiente de manera local.

Con la aparición de nuevas tecnologías, menos costosas y más fáciles de entender y utilizar, surgen ideas en las diferentes comunidades que involucran la creación de dispositivos que estén al alcance de la sociedad en general.

### **Marco Teórico:**

En esta sección se elabora una revisión bibliográfica de los conceptos generales a partir de los cuales se sostiene la utilización de Sistema de Monitoreo y registro del Nivel del río con transmisión de datos a través de tecnología GSM/GPRS.

### **Desastres Naturales:**

Un desastre es la combinación de diferentes factores en una comunidad, la vulnerabilidad y la capacidad para reducir el daño, ocasionando diferentes tipos de impactos, pérdidas materiales, económicas, de la vida, así como la generación de enfermedades y efectos negativos en el medio físico, mental y social de la humanidad.

Por lo general, se utiliza indistintamente el término desastre para referirse a eventos adversos de distinta magnitud y grado de impacto. Así, se suelen tomar como sinónimos conceptos como emergencia, desastre y catástrofe, que son diferentes en función de la magnitud e intensidad de sus consecuencias y la capacidad de respuesta social e institucional.(Davis, Ian et al. 2004. Social vulnerability and capacity analysis. Discussion paper and workshop report. Ginebra, Provention Consortium)

En gestión del riesgo se suele utilizar la expresión evento adverso para incluir estos tres conceptos de implicancias distintas.

Si bien existen diferentes conceptualizaciones del riesgo, es importante que nos basemos en aquella que lo define como el resultado de la interacción entre dos factores que se condicionan mutuamente: la amenaza y la vulnerabilidad. Esto significa que un fenómeno natural no se convierte en amenaza si no existe una sociedad vulnerable a su ocurrencia.

Adicionalmente, consideramos interesante incorporar un tercer factor, que compensa a la vulnerabilidad: las capacidades que posee una sociedad para hacer frente a las amenazas. Así, en la medida en que una comunidad fortalece sus capacidades, por ejemplo su organización, se vuelve menos vulnerable a los peligros o amenazas. (Ulloa, F. 2011. Manual de gestión de riesgo de desastre para comunicadores sociales: Una guía práctica para el comunicador social comprometido en informar y formar )

### **Inundación:**

De acuerdo con el Glosario Hidrológico Internacional (WHO-UNESCO, 2012), la inundación se define de la siguiente manera:

- aumento en el nivel de agua de un río o arroyo hasta un máximo desde el cual dicho nivel de agua desciende a menor velocidad;
- caudal alto de un río o arroyo medido por medio de la altura de nivel o por la descarga; • aumento de la marea.

Una inundación puede estar relacionada con precipitaciones intensas o prolongadas, la crecida de un río, la marea de tormenta, el oleaje, o con la falla de alguna estructura hidráulica, todo lo cual provoca un incremento en el nivel de la superficie libre del agua de los ríos o el mar. Otros factores que influyen en la ocurrencia de inundaciones son: la capacidad de absorción de los suelos, insuficiente capacidad de descarga de los cursos de agua y la pendiente del terreno (zonas de estancamiento). El desborde genera la invasión de agua en sitios en los cuales usualmente no la hay y ocasiona, por lo común, daños sobre la población y los bienes que se distribuyen sobre el territorio afectado.(WHO-UNESCO 2012. Glosario Hidrológico Internacional. Ginebra, Junta de Publicaciones de la Organización Meteorológica Mundial)

Es importante acentuar que, en todos los casos, una inundación es un evento que forma parte de la dinámica propia de los recursos y cuerpos de agua. Debido a esto, es previsible que ocurra cada cierto periodo de tiempo.

Este tipo de eventos tiene las características de una inundación repentina, pues se asocian con frecuencia a tormentas severas con importante desarrollo convectivo, de corta duración y concentrada en un área relativamente pequeña.(Tucci Carlos, 2005. Gestão de inundações urbanas. Porto Alegre, UNESCO-PHI).

### **Internet de las Cosas (IoT):**

La Unión Internacional de Telecomunicaciones (ITU, 2012), el Instituto de Ingeniería Eléctrica y Electrónica (IEEE, 2015) y el Grupo de Trabajo de Ingeniería de Internet (IETF, 2019), propone la siguiente definición para el Internet de las Cosas donde afirma que es una “Red global que conecta objetos identificable con capacidad de detección, actuación, procesamiento y comunicación con otros dispositivos, centro de datos o actuación, procesamiento y comunicación con otros dispositivos, centro de datos o nubes computacionales empleando tecnologías de comunicación e información, con el propósito de obtener aplicaciones avanzadas e información productiva”

Dicho de otra manera el internet de las cosas es una red de dispositivos físicos (vehículos, electrodomésticos y otros elementos integrados con componentes electrónicos), software, sensores, actuadores y conectividad que les permite conectarse entre ellos e intercambiar datos. La próxima revolución en la era de la informática estará fuera del ámbito de la visión tradicional.

Según Gabriel Galindo Romera, autor del documento realizado para la obtención de su doctora, llamado Diseño e implementación de sensores y lectores IoT para la

caracterización dieléctrica y la detección de espesores (Septiembre de 2019), afirma que “En el paradigma del IoT, la mayoría de los objetos que hoy en día nos rodean estarán conectados en la red de una forma u otra.” Se podría decir que el establecimiento tanto de dispositivos sensoriales, así como de dispositivos en general, están integrados entre sí a los sistemas de información y comunicaciones. Esto se traduce en la generación de enormes cantidades de datos que deben almacenarse, procesarse y presentarse de forma transparente, eficiente y fácilmente interpretable. Con respecto a esto Nguyen & De Creme (2016), apoya que “El poder de capturar los datos, provenientes de los sensores de los objetos conectados, y analizarlos generan el verdadero valor de IoT.” Estos sistemas tendrán la capacidad de reaccionar de forma autónoma y adecuada en dependencia a sus configuración, reconociendo eventos o cambios, donde además tendrán como finalidad brindar una infraestructura que supere la barrera entre los objetos del mundo físico y su representación en los sistemas de información.

Comprendiendo qué es el Internet de las Cosas, se podría decir que presenta cuatro características importantes, estas son:

- A. Detección y actuación: Los objetos poseen inteligencia embebida y cuentan con sensores para la detección y medición de variables físicas y actuadores para ejecutar acciones sobre otros elementos a su alrededor.
- B. Comunicación: Los objetos establecen vínculos de comunicación empleando protocolos interoperables con otros objetos o aplicaciones y están disponibles de acuerdo a los requerimientos definidos por los usuarios.
- C. Procesamiento: La data obtenida por los objetos es procesada en centros de datos locales o en nubes computacionales, almacenados en bases de datos y analizados para obtener información útil para los usuarios.
- D. Aplicación: La información se presenta a los usuarios finales a través de una interfaz gráfica mediante aplicaciones de software desarrollados para terminales computacionales fijos o móviles.

### **Aplicaciones del IoT:**

Debido al impacto que ha tenido esta tecnología en diversos sectores, el Internet de las Cosas ha demostrado la importancia que tendrá en el futuro dentro de diversos campos. Esto quiere decir que las empresas, organismos y los dispositivos personales serán más

eficientes y competitivos para todos los negocios del mundo actual. Por ende, se relacionan las mejores aplicaciones del IoT, estas son:

- Redes eléctricas inteligentes
- Salud y telemedicina
- Hogares y edificios inteligentes
- Seguridad
- Riego y Agricultura inteligente

### **Tecnologías de Comunicación Inalámbricas utilizadas en IoT:**

Las redes de comunicaciones han ido evolucionando hacia el sector del IoT que, aunque actualmente no compite con el sector de la telefonía móvil a nivel comercial, sí que ha despertado el interés y la inversión de numerosas empresas en este sector.

Existe una diversa cantidad de tecnologías inalámbricas disponibles tanto para IoT, cada una de ellas brinda ventajas y limitaciones de acuerdo al nivel de despliegue que requiera el usuario final. De acuerdo al Proyecto de Investigación, titulado "Sistema Domótico basado en IoT para el Centro Integral del Adulto Mayor de AMBATO" presentado por Jonathan Eduardo Moreta Tigse (Septiembre de 2022), afirma que "Entre las tecnologías que más destacan se tiene: wifi, bluetooth, zigbee, LoRa, Z-Wave y Tecnología Móvil".

A continuación, vemos una primera aproximación de las redes de comunicación más utilizadas para la ejecución de proyectos de IoT.

**Tabla 1**

Comparacion de Tecnologias Inalambricas

<b>Parámetro</b>	<b>WiFi</b>	<b>Bluetooth</b>	<b>Zigbee</b>	<b>LoRa</b>	<b>Z-Wave</b>	<b>Celular</b>
<b>Estándar</b>	IEEE 802.11 a/c/b/d/g/n	IEEE 802.15.1	IEEE 802.15.4	LaRaWAN R1	IEEE 802.15.4	GSM
<b>Frecuencia</b>	2.4GHz 5-60 GHz	2.4 GHz	868/915 MHz, 2.4 GHz	868/900 MHz	900MHz 2.4 GHz	850 y 1900 MHz
<b>Velocidad</b>	1 Mb/s -	1 - 24 Mb/s	40 - 250	0.3 - 50	9,6-	de 56 kbit/s

de datos	6.75 Gb/s		Kb/s	Kb/s	100kb/s	hasta 114 kbit/s
Alcance	20 – 80m	20 – 80m	10 - 20m	< 15 Km	30 m	35 km
Consumo de energía	Alto	Medio	Baja	Muy Baja	Baja	Muy Alto

Nota: Es una comparación de las tecnologías inalámbricas aplicadas en proyectos IoT. Tomado de “Sistema de Domótica Basado en IoT para el centro integral del adulto mayor de la ciudad del Ambato”, por Jonathan Eduardo Moreta Tigse, año 2022.

Principios para una red de comunicaciones de IoT:

- Bajas velocidades de datos
- Baja frecuencia de transmisión
- Movilidad y servicios de localización
- Conexiones bidireccionales seguras
- Bajo consumo de energía
- Largo alcance de comunicación

Las versiones y características mostradas en la Tabla 1, corresponden a las versiones disponibles de manera comercial, siendo las versiones GSM las de mayor relevancia para el proyecto por su compatibilidad y las acciones a realizar con los dispositivos a utilizar.

### **Sistema Global para las Comunicaciones Móviles (GSM):**

#### **Introducción a los Sistemas de Comunicación Móviles:**

Para entender lo qué es GSM, primero se debe explicar qué es y qué propósito tiene un Sistema de comunicación Móvil. El propósito de un sistema de comunicaciones móvil es,

como su nombre indica, prestar servicios de telecomunicaciones entre estaciones móviles y estaciones terrenas fijas, o entre dos estaciones móviles. Existen dos formas de comunicaciones móviles: inalámbrica y celular.

- Comunicación inalámbrica: La comunicación inalámbrica es el tipo de comunicación en la que no se utiliza un medio de propagación físico alguno, esto quiere decir que se utiliza la modulación de ondas electromagnéticas, las cuales se propagan por el espacio sin un medio físico que comunique cada uno de los extremos de la transmisión. En ese sentido, los dispositivos físicos sólo están presentes en los emisores y receptores de la señal, como por ejemplo: antenas, computadoras portátiles, PDA, teléfonos móviles, etc.
- Comunicación celular: Tiene una red totalmente definida que incluye protocolos para establecer y despejar llamadas así como rastrear las unidades móviles dentro de áreas geográficas definidas llamadas células, que dan nombre a la tecnología.

Según Datateca.unad.com (2014) explica qué “Dado que los sistemas celulares operan con una potencia más alta que los demás sistemas inalámbricos, el radio de acción de los primeros es mucho más extenso, siendo el tamaño de las células del orden de kilómetros”.

### **GSM:**

Sabiendo lo qué un sistema de comunicación móvil, podemos entender de qué trata GSM. Se trata de un estándar muy utilizado desde principios de siglo y también se conoce como 2G debido a que supuso un salto de las comunicaciones analógicas a las digitales.

La banda de frecuencia en la que opera el GSM difiere según el territorio. En Europa se utiliza el espectro radioeléctrico de 900 y 1800 MHz, mientras que en Estados Unidos la banda es la de 1900. Esto hace que no todos los móviles GSM puedan funcionar en todo el mundo, a no ser que su tecnología esté preparada para conectarse a todas las bandas.

El estándar GSM permite transmisiones digitales de voz y datos, como mensajes de texto (SMS) o mensajes multimedia (MMS).

Los primeros trabajos con GSM los inició en 1982 un grupo dentro del Instituto Europeo de Normas de Comunicaciones (ETSI, European Telecommunications Standards Institute). Originalmente, este organismo se llamaba Groupe Sociale Mobile, lo que dio pie al acrónimo GSM.

El objetivo de este proyecto era poner fin a la incompatibilidad de sistemas en el área de las comunicaciones móviles y crear una estructura de sistemas de comunicaciones a nivel europeo.

### **Arquitectura GSM:**

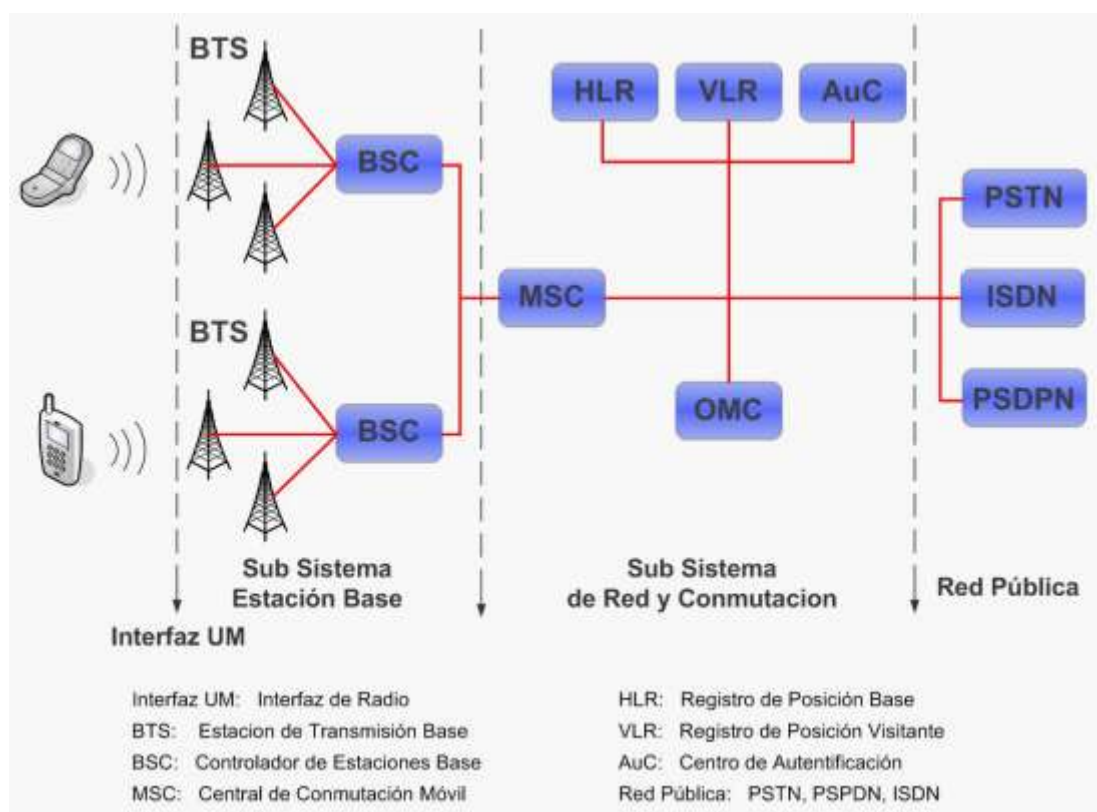
En relación a su arquitectura de red, Según la información obtenida del documento “Comunicación Móviles digitales”, realizado por La Universidad Politécnica de Madrid (Julio de 2013) explica que en “GSM todo terminal móvil debe estar constituido por una tarjeta SIM (Módulo de identificación de abonado) y el propio dispositivo, normalmente un teléfono móvil”.

La SIM es la tarjeta que identifica al usuario. Este a su vez contiene un identificador único, el cual es llamado por la siglas IMSI (Identificador internacional de abonados móviles), el cual es la identidad del usuario, única en el mundo. Este está compuesto por tres partes, el código de país móvil (MCC), el código de red móvil (MNC) y el número de identificación de abonado móvil (MSIN). El MCC forma los tres primeros dígitos de la IMSI e indica en qué país está registrado el usuario de la SIM.

Por otro lado existe un identificador único de 15 dígitos en el equipo móvil, que lo identifica como IMEI (Identificador internacional de equipos móviles). Es decir, es la identidad del móvil, única en el mundo.

### **Figura 1**

Arquitectura GSM



Nota:Arquitectura GSM con los elementos que la componen.Tomado de “Sistema de control de Temperatura a través de arduino y la tecnología GSM/GPRS”, por Alberto Castro Dominguez,año 2000

En la figura 1 podemos ver la arquitectura de red correspondiente al sistema GSM. Está compuesta por múltiples estaciones base (BTS), que a su vez, se conectan a un controlador de estaciones base (BSC), encargado de la administración de la red. A éste sistema compuesto por el BSC y sus correspondientes estaciones base conectadas al mismo, se le conoce como BSS (Subsistema de estaciones base).

En un nivel superior estarían los Centros de conmutación móvil (MSC), al que se conectan físicamente los controladores de estaciones base. Éste es el encargado de establecer la conexión con la red de telefonía pública y con Internet. Su administración corre a cargo del operador de la red telefónica.

El MSC pertenece a un Subsistema de conmutación de red (NSS), el cual se encarga de identificar a los usuarios, determinar su ubicación, y gestionar las comunicaciones con otros usuarios de la red. A su vez, el Centro de Conmutación móvil (MSC) se conecta a una serie de base de datos que le proporcionan funciones adicionales:

- Registro de posición base (HLR): en esta base de datos se almacena la información de los abonados (posición geográfica, información administrativa, etc.).

- Registro de posición visitante (VLR): contiene información de usuarios que no son abonados locales. Los datos se conservan mientras el usuario está dentro de la zona, y se eliminan en cuanto abandona la zona o tras un largo período de inactividad.
- Registro de identificación del equipo (EIR): es una base de datos que contiene la lista con los dispositivos móviles.
- El Centro de autenticación (AUC): su misión es verificar las identidades de los usuarios.

### **Servicio General de Paquetes vía Radio (GPRS):**

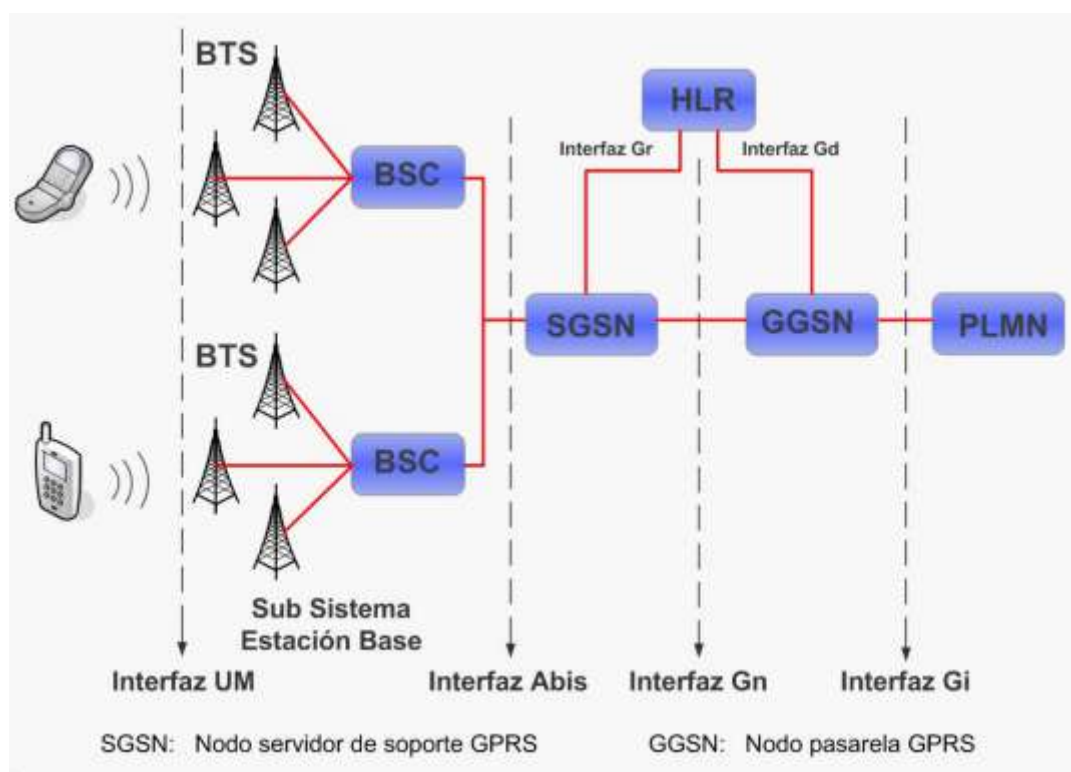
El servicio general de paquetes vía radio o también conocido como GPRS es una red de paquetes de datos que provee de acceso de radio paquetes para GSM y TDMA. Trabaja análogamente a la red de voz GSM y ofrece una plataforma para transferencia de datos y señalización.

A la inversa de lo que ocurre en la conmutación de circuitos, en el estándar GPRS, gracias a su modo de transferencia de paquetes, la transmisión de datos solo utilizan la red cuando es necesario, generando tarificación por la cantidad de información transmitida en vez de por el tiempo de conexión, provocando que el usuario esté conectado sin generar costos adicionales, ya que solo utiliza la red al enviar o recibir datos.

El sistema GPRS con respecto a la infraestructura de GSM, comparte recursos e incluso permite mandar SMS por medio de los componentes GPRS, lo cual significa que ocurren algunos cambios en relación a la red GSM. En la figura 2 podemos observar la adición de nuevos nodos sobre la infraestructura ya existente, conociéndolos como GSN o Nodos de soporte GPRS.

### **Figura 2**

Arquitectura GPRS



Nota:Arquitectura de la Red GPRS, donde se puede observar que comparte recursos con GSM.Tomado de “Sistema de control de Temperatura a través de arduino y la tecnología GSM/GPRS”, por Alberto Castro Dominguez,año 2000

- El SGSN (Nodo de soporte de servicio GPRS) se encarga del re-direccionamiento de paquetes IP entrantes y salientes, dirigido hacia o desde cualquier abonado GPRS situado dentro de la zona geográfica a la que da servicio ese SGSN.
- El GGSN (Nodo de soporte pasarela GPRS) sirve como interfaz hacia las redes externas de paquetes IP.Se encarga de la asignación de IPs a los terminales, estáticas o dinámicas y intercambia información de ruteo con la red externa.Este nodo prepara y gestiona la sesión con redes externas de GPRS y a su vez recopila datos de tarificación y uso de la red de datos externa.

### Comandos AT:

Antes de la introducción del BBS (Sistema de Tabla de Anuncios), los modem normalmente operaban con línea telefónica de marcación directa que comenzaba desde el inicio hasta el fin con modem en cada extremo.Los módems funcionaban en los modos “originar” o “responder”, alternando manualmente entre dos conjuntos de frecuencias para la transferencia de datos.

Cuando se requería automatizar, generalmente solo se necesitaba en el extremo de donde se generaba la respuesta, por ejemplo, un banco podría necesitar recibir llamadas de varias sucursales para el procesamiento al final de día. Para cumplir esta función, algunos módems incluyen capacidad de levantar el teléfono de manera automática cuando estaba en modo “respuesta” y borrar la línea cuando el otro usuario se desconecta manualmente. La necesidad de marcar automáticamente era considerado menos común, y se manejaba a través de un dispositivo periférico, llamado marcador. Este método de operación se utilizó de manera correcta generando satisfacción en la década de 1960 y principios de 1970.

La industria de la informática necesitaba una forma de decirle al modem que número marcar a través del software. Los primeros marcadores separados tenían esta capacidad, pero solo a costa de un puerto separado, que una microcomputadora podría no tener disponible.

Fue así como surgen los Comandos AT en 1977 desarrollado por Denis Hayes como una interfaz de comunicación, con un modem para así poder configurarlo y proporcionarle instrucciones, tales como marcar un número de teléfono (Victor Hugo Hernandez Joffre, et al., Noviembre de 2009).

Aunque el objetivo principal de estos comandos es la comunicación con módems, la telefonía móvil GSM también ha adoptado como estándar este lenguaje para poder comunicarse con sus terminales. Más adelante, con el avance del audio, fueron las compañías Microcom y US Robotics las que siguieron desarrollando y expandiendo el juego de comandos hasta universalizarse. Los comandos AT se denominan así por la abreviatura de attention. Aunque la finalidad principal de los comandos AT es la comunicación con cualquier modem, la telefonía móvil GSM también ha adoptado como estándar este lenguaje para poder comunicarse con sus terminales. De esta forma, todos los teléfonos móviles GSM poseen un juego de comandos AT específico que sirve de interfaz para configurar y proporcionar instrucciones a los terminales.

Estas instrucciones permiten funciones tales como llamada de voz, leer y escribir en la agenda de contactos y enviar mensajes SMS, entre otras funciones.

La implementación de estos comandos AT varían dependiendo del dispositivo GSM y no depende del canal de comunicación a través del cual estos comandos sean enviados, ya sea cable de serie, canal infrarrojos, Bluetooth, etc. De esta forma, es posible distinguir distintos teléfonos móviles del mercado que permiten la ejecución total de comandos o solo parcialmente (Victor Hugo Hernandez Joffre, et al., Noviembre de 2009).

En la tabla 2 se muestran los comandos más utilizados en GSM.

## **Tabla 2**

Lista de Comandos AT básicos

Comandos AT	Descripción
AT+IPR=?	Muestra la velocidad que acepta el módulo
AT+IPR=9600	Define la velocidad en 9600 baudios.
AT+CGSN	Devuelve el IMEI(Identidad de Equipo de Estación Móvil internacional).
AT+CPIN?	Comprobamos que la tarjeta SIM esté funcionando
AT+CGREG?	Comprobar el estado de registro del dispositivo
AT+COPS?	El dispositivo devuelve la lista de redes que se encuentran disponibles para conectarse.
AT+CSQ	Medir el rendimiento de la fuerza de la señal.
AT+CMGF?	Seleccionar formato de mensaje SMS (0=PDU, 1=Texto).
AT+CMGL="ALL"	Mirar todos los mensajes
AT+CMGL="REC UNREAD"	Mirar solo mensajes no leídos
AT+CMGL="REC READ"	Mirar solo mensajes leídos
AT+CMGL="STOP UNSENT"	Mirar solo mensajes no enviados

AT+CMGL="STO SENT"	Mirar solo mensajes enviados
AT+CCALR?	Dispositivo listo para llamada (1 significa listo, 0 significa que no está listo).
AT+CCID	Obtiene el número de la tarjeta SIM
AT+CLIP?	Dispositivo listo para llamada (1 significa listo, 0 significa que no está listo).

Nota:Lista de comandos más utilizados.Tomado del Datasheet del Módulo Sim800L

### Planificación:

Para comenzar, ya teniendo en cuenta el dispositivo a realizar, se inició la búsqueda de los elementos o tecnologías que se podrían utilizar para la implementación del mismo y sus tareas a realizar.

Se debe tener en cuenta que el dispositivo se realizó con el objetivo de registrar datos y analizarlos, y en cuanto a su espacio físico, se encontrará generalmente ubicado en ambientes de extremo al intemperie. Para la realización del proyecto se debe tomar en consideración que la parte física o Hardware estará formado por un dispositivo central, que será el cerebro y se encargará de darle las órdenes a los demás, y por otro lado un dispositivos de sensado y un dispositivo de comunicación encargado de realizar el envío de los datos sensado.

Por otro lado como un dato adicional estará apoyado por una plataforma IoT, que será el encargado de la recepción de los datos y plasmarlos en una interfaz amigable para su monitoreo y registro.

### Elección del Hardware:

Para la selección correcta del hardware a utilizar, se comenzó analizando las tareas que debía realizar el dispositivo a diseñar, por lo tanto se inicio seleccionando la placa con su microcontrolador y el módulo encargado de enviar los datos sensado. Para esto se escogió la Placa Arduino Uno R3 con un microcontrolador Atmega328P, que se será el cerebro encargado de realizar todas las funciones, y el módulo Sim800L EVB, que nos garantiza el envío de datos mediante GSM/GPRS, y debido a que la tarea del dispositivo en concreto generalmente va estar ubicado en zona alejadas de la población, donde por lo general no hay recepción de señal Wifi. En cuanto a sensor de distancia, se seleccionó en cuanto a su precisión y presupuesto el HCSR-04, el cual mediante ondas de ultrasonido obtiene la distancia de un objeto.

Por último para una mejor percepción se decidió utilizar un Display de 16 columnas y 2 filas, con un adaptador I2C.

### **Elección del Software:**

Para la realización del programa se utilizará el entorno de desarrollo integrado (IDE) de Arduino, donde se le asignará todas las tareas a realizar del dispositivo.

### **Pruebas y Mediciones del Hardware:**

Una vez seleccionado y en posesión de los dispositivos que conforman el hardware, se procederá a realizar las pruebas de cada uno, por separado y luego en conjunto, previamente al montaje del dispositivo completo. Esto se lleva a cabo con el objetivo de comprobar su correcto funcionamiento y considerar algún factor para su armado.

### **Montaje:**

Una vez realizadas las pruebas correspondientes, se llevarán a cabo las tareas de montaje sobre el recipiente en el que se resguarda y su correcto conexionado.

**CRONOGRAMA:**

<b>Cronograma de Actividades</b>	<b>Inicio</b>	<b>Fin</b>
<b>Elección de tecnología</b>		
Búsqueda de Dispositivo en general	11/04/2022	18/04/2022
Búsqueda y Comparación de Dispositivos	20/04/2022	29/07/2022
Selección de dispositivos	10/06/2022	15/06/2022
Realización de la compra de los dispositivos principales	12/07/2022	10/09/2022
Recepción de paquetes	22/07/2022	22/07/2022
<b>Realización del Anteproyecto</b>		
Introducción, Objetivos y Marco teórico	29/05/2022	30/05/2022
Planificación	31/05/2022	02/06/2022
Entrega de Anteproyecto	02/06/2022	02/06/2022
<b>Realización y Diseño de Software</b>		
Búsqueda de librerías del Módulo Sim800L EVB	20/06/2022	22/06/2022
Identificación de comandos AT	24/06/2002	29/06/2022
Configuración del Sim800L EVB	03/07/2022	07/07/2022
Configuración del Hc-sr04	11/07/2022	16/07/2022
Configuración de los Step Downs	19/07/2022	19/07/2022
Configuración y prueba con ThingsPeak	21/07/2022	27/07/2022
<b>Armado y diseño de hardware</b>		
Compra de caja estanca	9/10/2022	9/10/2022

Compra de Cables Jumper adicionales para conexionado	20/10/2022	20/10/2022
Acondicionamiento de la caja estanca	25/10/2022	27/10/2022
Agregado de placa de pertinax	22/10/2023	25/10/2023
<b>Pruebas y modificaciones</b>		
Prueba del código del Sensor Hc-sr04	25/07/2022	27/07/2022
Prueba del envío al servidor de ThingsPeak mediante GPRS	20/07/2022	03/04/2023
Realización de código del Display	22/09/2023	03/10/2023
Pruebas finales	10/10/2023	15/10/2023

**Desarrollo:**

**Descripción de dispositivos que componen el sistema:**

**ARDUINO :**

**Figura 4**

Arduino UNO R3



Nota: Imagen del Arduino UNO model R3. Tomado del Datasheet del Dispositivo

Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada tanto en Hardware como en Software, flexible y fácil de utilizar para los creadores y desarrolladores. En esta plataforma permite crear microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso.

Para tener en cuenta un Software Libre, como lo dice el nombre, es el software que respeta la libertad de los Usuarios y la comunidad. A grandes rasgos tiene la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Es decir, es una cuestión de libertad y no de precio, donde cuyo código es accesible por cualquiera, para quien quiera, pueda utilizarlo y modificarlo. Es el caso de Arduino IDE (Integrated Development Environment o Entorno de Desarrollo Integrado), que es un entorno de programación con el que cualquier persona puede crear aplicaciones para las placas Arduino, de manera que se les puede dar todo tipo de utilidades.

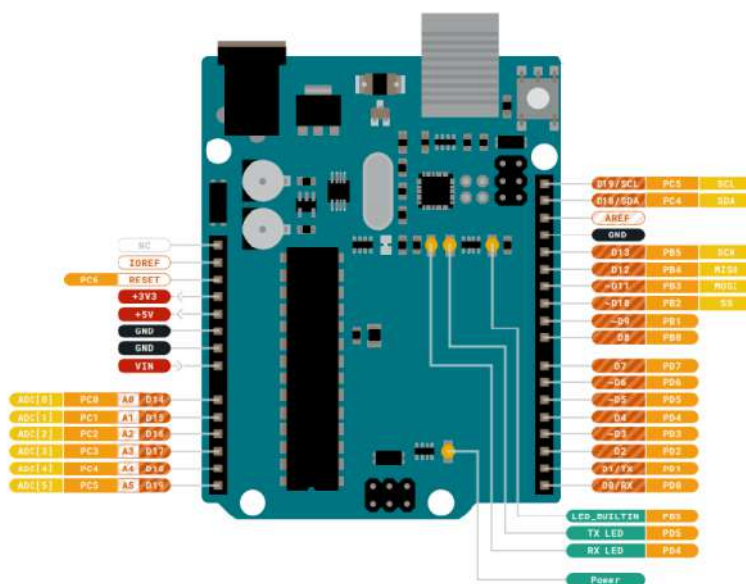
Es igual con el Hardware Libre, ya que son los dispositivos cuya especificación y diagramas son de acceso público de manera que cualquiera puede replicarlo. Esto quiere decir que Arduino ofrece las bases para que cualquier otro individuo o compañía pueda crear sus propios dispositivos, pudiendo ser diferentes entre ellos, pero funcionando a partir de la misma base.

El Arduino es una placa basada en microcontroladores ATMEGA. Estos microcontroladores son circuitos integrados en los que se pueden grabar instrucciones, las cuales las escribes con el lenguaje de programación que puedes utilizar en el entorno Arduino o IDE. Estas instrucciones permiten crear programas que interactúan con los circuitos de la placa.

El Microcontrolador posee lo que se llama una interfaz de entrada, que es una conexión en la que podemos conectar en la placa diferentes tipos de periféricos. La información de estos periféricos que se puede llegar a conectar se traslada al microcontrolador, el cual se encarga de procesar los datos que le llegan a través de ellos. El tipo de periférico que puedes utilizar para enviar datos depende de qué uso se le vaya a asignar.

También cuenta con una interfaz de salida, que es la que se encarga de llevar la información que se ha procesado en el Arduino a otros periféricos. Estos periféricos pueden ser pantallas o altavoces en los que reproducir los datos procesados, pero también pueden ser otras placas o controladores.

**Figura 5**  
Especificaciones de PinOut



Nota: Imagen con indicaciones de los pines que componen al Arduino UNO R3. Tomado de Datasheet del Dispositivo

### **Especificaciones:**

- Microcontrolador: ATmega328P.
- Velocidad de reloj: 16 MHz.
- Voltaje de trabajo: 5V.
- Voltaje de entrada: 7,5 a 12 voltios.
- Pinout: 14 pines digitales (6 PWM) y 6 pines analógicos.
- 1 puerto serie por hardware.
- Memoria: 32KB Flash (0,5 para bootloader), 2KB RAM y 1 KB Eeprom

### **Pines que lo componen:**

Los pines es donde se conectan los cables de un circuito. El Arduino tiene varios tipos diferentes de entradas, cada uno de las cuales está marcado en el tablero y utilizan para diferentes funciones:

- GND: pines conectados a tierra (es decir, a 0[V]). Es una referencia común que todos los componentes de los circuitos han de compartir. Existe otro pin GND cerca

de los pines de entrada y salida digital de la placa. Se puede utilizar cualquiera de los pines GND, según nos resulte más cómodo uno u otro, a la hora de diseñar el cableado del circuito.

- 5 V :Es el pin de voltaje de salida de la placa de 5V que puede venir del adaptador AC/DC (7-12V), desde el pin Vin (7-12V) o a través del USB u otra forma de 5V regulada. Este pin puede suministrar alimentación eléctrica a la placa sin utilizar el jack y ni el USB, sino mediante una batería(pila), conectando su borne positivo a este pin y su borne negativo a tierra (GND), y así el regulador de tensión reducirá el voltaje recibido a 5V. La intensidad de corriente máxima generada es de 40 mA.
- 3.3 V: Pin de salida de 3,3V a través de un circuito específico (el regulador de tensión "LP2965" incorporado en la placa) a partir del voltaje recibido indistintamente a través del conector para USB o el conector para el conector jack del adaptador AC/DC. La intensidad de corriente máxima generada, en este caso, será de 50 mA. También se puede utilizar este pin para alimentar componentes del circuito (los más delicados que requieren dicho voltaje) pero, sin embargo, no se puede conectar una fuente externa a este pin, ya que el voltaje es demasiado limitado como para poder alimentar la placa. Aunque no se recomienda suministrar voltaje a la placa a través de los pines 5V y 3,3V porque no pasa por el regulador de tensión y se puede dañar la placa.
- Analógico: El área de pines en el marco del analógico etiqueta (A0 a A5) son analógicos. Estos pines pueden leer la señal de un sensor analógico y convertirlos en valores digitales
- Digital: Son los pines digitales de 0 al 13. Estos pines se pueden utilizar tanto para la entrada digital y salida digital
- PWM: Se puede apreciar una tilde (~) al lado de algunos pines digitales, estos son 3,5,6,9,10 y 11. Estos pines actúan como pines digitales normales, pero también se pueden usar para Modulación por ancho de pulso.
- AREF: Voltaje de referencia para la entrada analógica. Tiene como función indicar la referencia de la tensión con la que está trabajando internamente la placa (su microcontrolador), con 5V al tratarse de Arduino UNO. Puede ser de utilidad si se acoplan placas complementarias (shields) a Arduino UNO, y con este pin indicarles el voltaje de trabajo de los pines E/S de Arduino UNO, para que se adapten automáticamente a dicho voltaje.

**Botón de reinicio(RESET):**

Con la utilización de este botón se conectará temporalmente el pin de reset a tierra y provocará que se reinicie cualquier código que se carga en el Arduino. Esto puede ser muy útil si el código solo se ejecuta una vez sin repetirse.

### **Microcontrolador ATmega328p:**

Este es el cerebro del Arduino uno R3. Como se menciona al principio del proyecto, la placa utilizada es una plataforma basada en microcontroladores con entradas y salida listas para usar. El microcontrolador es el que se encarga de ejecutar nuestros programas, lee nuestras entradas, envía señales para controlar dispositivos a la velocidad que el cristal le permite.

### **Figura 6**

ATmega328p



Nota: Se puede apreciar un ATmega328p utilizados en los Arduino UNO del modelo R3. Tomado de <https://electronoobs.io/shop/courses/curso-arduino-desde-cero-nivel-1/cursos/clase-1-que-es-arduino/>

Los principales elementos que componen al microcontrolador ATmega328P, son: CPU, memoria y pines E/S (conectores con formas de “patas”). La CPU ejecuta cada instrucción y controla que se realicen correctamente, donde generalmente las instrucciones responden de acuerdo a los datos de entrada, generando datos de salida que serán utilizados, o no, por otra instrucción.

Los 3 tipos distintos de memoria sirven para alojar las instrucciones y los datos que éstas necesitan. De forma general, las memorias se suelen clasificar en **persistentes** (lo que quiere decir que el contenido se almacena de forma permanente, aunque se dejen de alimentar eléctricamente) o **volátiles** (lo que quiere decir que se pierde al momento de no recibir alimentación eléctrica) y según las características de la información a guardar, esta se grabará en un tipo de memoria, habitualmente, de forma automática. El ATmega328P

consta de una memoria Flash (Volátil) y dos memorias que son la SRAM y EEPROM (Persistentes). Por su parte, los 23 pines E/S, sirven para la comunicación del microcontrolador con los periféricos. Se dice que son pines GPIO (General Purpose Input/Output), porque son pines de propósito general, pero suele tener otra función específica. Entre los pines se encuentra un pin VCC que recibe la alimentación eléctrica, un pin AVCC que recibe alimentación suplementaria para ADC (Analogic Digital Converter) interno, un pin AREF que recibe referencias voltaje analógica para el ADC y dos pines GND conectados a tierra.

### Sensor HC-SR04:

**Figura 7**

HC-SR04



Nota: Sensor de Distancia de Ultrasonido HC-SR04. Tomado de <https://adichip.net/tienda/producto/sensor-distancia-ultrasonido-ultrasonico-sr04-hc-arduino/>

El HC-SR04 es un sensor de distancia, que utiliza el ultrasonido para determinar la distancia, con un rango de 2 cm a 450 cm. Destaca por su tamaño, bajo consumo energético, buena precisión y confiabilidad. Es uno de los sensores ultrasónicos más utilizados en proyectos de robótica y sistemas de alerta.

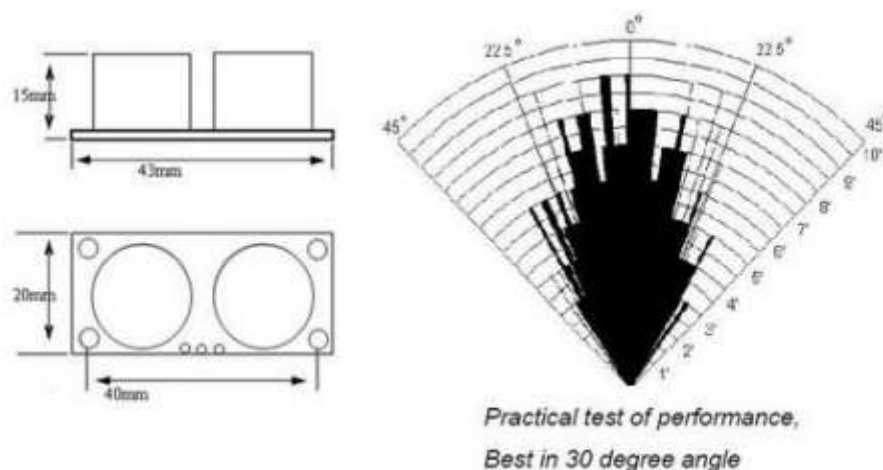
Este consta de dos transductores: uno es un emisor de ultrasonido, mientras que el otro es el receptor piezoeléctrico. El funcionamiento se compone de los siguientes pasos: El receptor envía 8 pulsos de ultrasonidos a una frecuencia de 40KHz, luego de recibir la orden en el pin TRIG, las ondas de sonidos viajan en el aire y rebotan sobre el objeto que chocan, el sonido de rebote es detectado por el receptor piezoeléctrico, luego el pin ECHO cambia a 5 V por un tiempo igual al que demora la onda desde que fue emitida hasta que fue detectada, y el tiempo de pulso ECO es medido por el microcontrolador y así se puede

calcular la distancia al objeto. Es importante aclarar que el sensor no se ve afectado por la luz solar o material de color negro.

### Especificaciones:

**Figura 8**

Dimensiones y ángulo de detección



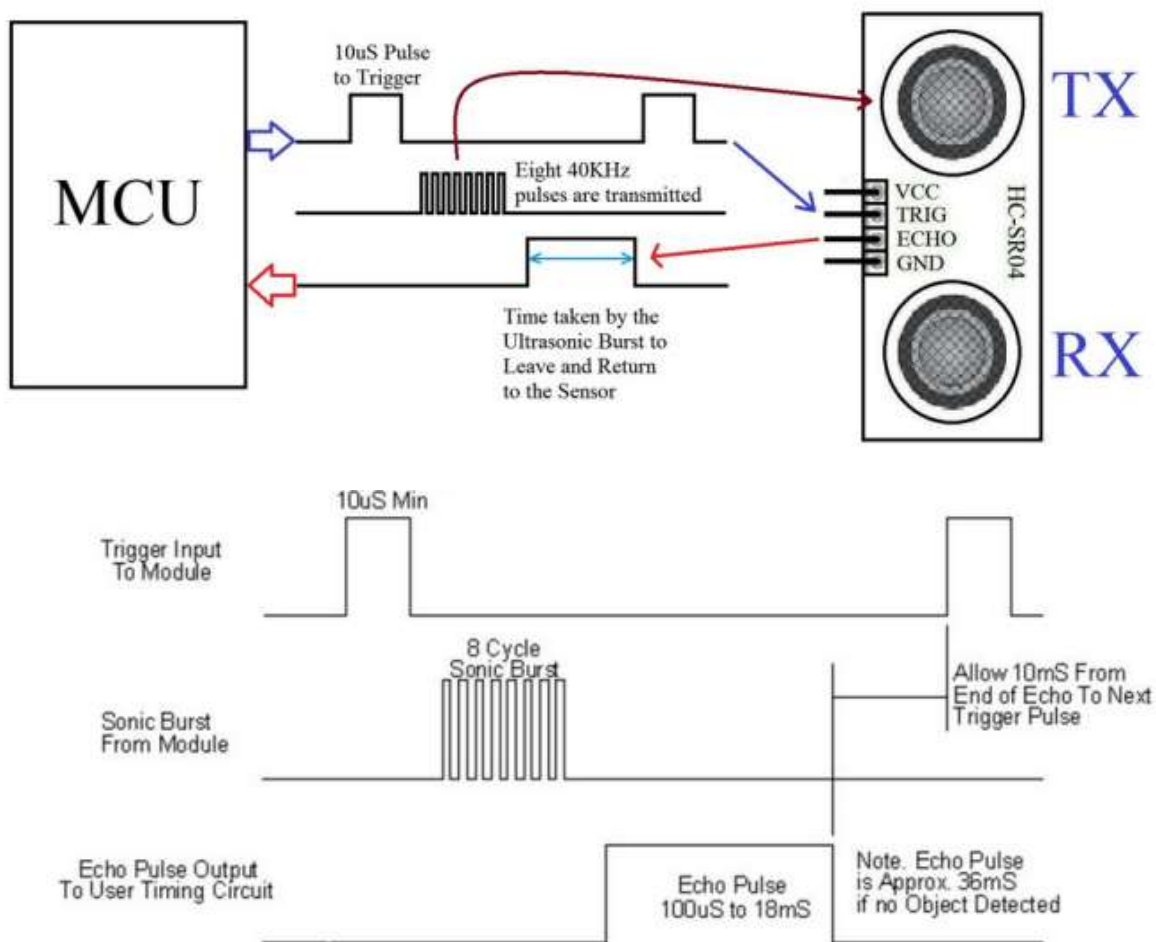
Nota: Diámetro del Sensor y representación del rango de espacio del mismo. Tomado del Datasheet "HC-SR04 HandsOn Technology"

- Alimentación de 5 volts.
- Interfaz de cuatro hilos (VCC, trigger, echo, GND).
- Rango de medición: 2 cm a 450 cm.
- Corriente de alimentación: 1.5mA.
- Frecuencia de pulso: 40Khz.
- Apertura del pulso ultrasónico o ángulo efectivo: 15°.
- Señal de disparo: 10us.
- Dimensiones del módulo: 45x20x15mm.

### Diagrama de Tiempo:

**Figura 9**

Diagrama de Tiempo de Pulsos

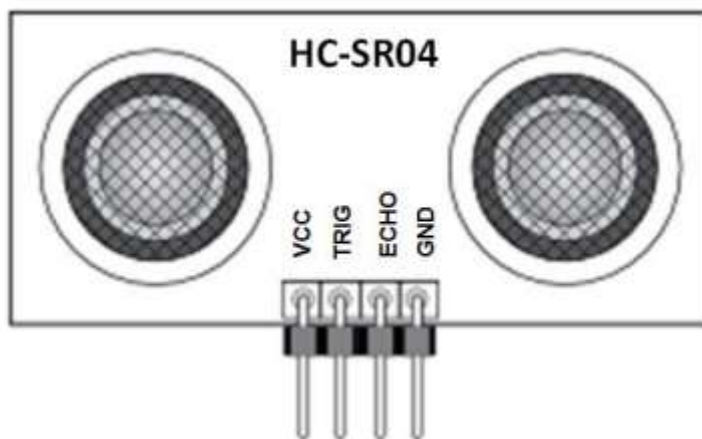


Nota: Diagrama del Tiempo de los pulsos efectuados por el HC-SR04. Tomado de "HC-SR04 HandsOn Technology".

**Pines que lo componen y dimensiones:**

**Figura 10**

Imagen ilustrativa de pines



Nota: Representación ilustrativa del Sensor con sus respectivos pines. Tomado de Datasheet "HC-SR04 HandsOn Technology"

1. VCC, tensión de alimentación
2. Trigger, disparo
3. Echo, retorno
4. Tierra

**Características Eléctricas:**

Parámetro	Valor
Tensión de alimentación	5.0 Vdc
Corriente de trabajo	15 mA
Corriente de reposo	2 mA
Frecuencia de señal	40 KHz

## Modulo GSM/GPRS Sim800L:

Figura 11

### Módulo Sim800L V2



Nota: Imagen del Módulo Sim800L V2 GSM/GPRS. Tomado de <https://www.compelelectronica.com/product/modulo-sim800l-v2-2-gsm-gprs-quad-band-sim800/>

El Sim800L V2 GSM/GPRS es un modem GSM de tamaño de pequeño que puede incorporarse en una gran infinidad de proyectos. Este módulo permite realizar todas las funciones que un celular común y corriente puede realizar en cuanto a comunicación, desde enviar SMS, realizar o recibir llamadas, conectarse a internet a través de GPRS, TCP/IP, entre otros, soportando 4 frecuencias comunes de datos GSM: 850, 900, 1800 y 1900 MHz teniendo como ventaja poder comunicarse casi desde cualquier punto del mundo.

### Especificaciones:

- Dimensiones: 40 x 28 (L x W) mm.
- Soporta la banda cuádruple: GSM850, EGSM900, DCS1800 y PCS1900
- Conéctate a cualquier red global GSM con cualquier SIM 2G
- Hacer y recibir llamadas de voz usando un altavoz externo 8Ω y un micrófono electret
- Enviar y recibir mensajes SMS
- Enviar y recibir datos GPRS (TCP/IP, HTTP, etc.)
- Escanear y recibir emisiones de radio FM
- Transmitir energía:

- Clase 4 (2W) para GSM850
- Clase 1 (1W) para el DCS1800
- Conjunto de comandos AT basados en serie
- Conectores FL para antenas celulares
- Acepta la tarjeta micro SIM

**Tabla 2**

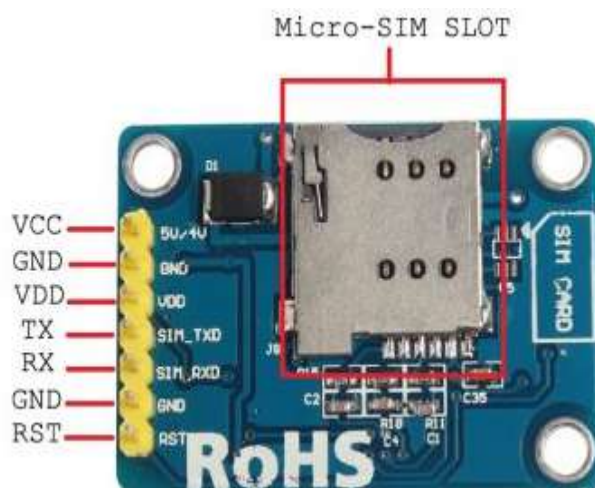
Características eléctricas Sim800L EVB

<b>Modos</b>	<b>Frecuencia</b>	<b>Consumo</b>
Alimentación		5.0 Vdc
Corriente de reposo		1 mA
Corriente de preparado		18 mA
Llama a	GSM850	199 mA
	EGSM900	216 mA
	DCS1800	146 mA
	PCS1900	131 mA
GPRS		453 mA
Rafaga de Transmision		2 A

**Pines qué los componen:**

**Figura 12**

Pines del Sim800L V2



Nota: Imagen con pines que componen el Sim800L V2. Tomado de <https://www.dev.faranux.com/product/sim800l-v2-0-5v-wirelessgsm-gprs-module-quad-band/>

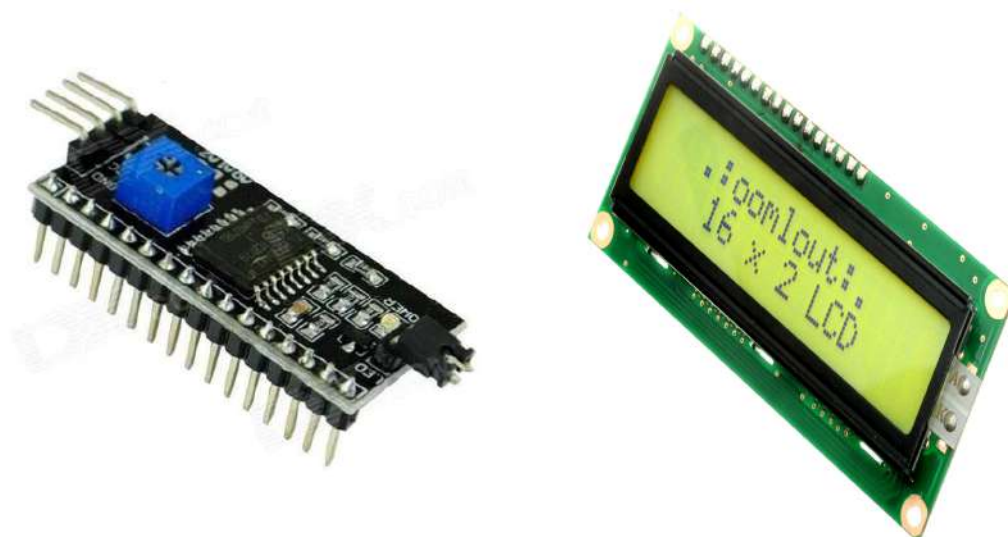
Hay 7 pines en total en el SIM800L, que usaremos para interactuar con Arduino. Estos son:

1. VCC: entrada de voltaje de suministro externo para SIM800L
2. GND: Tierra externa para SIM800L
3. VDD: Entrada de tensión de alimentación del microcontrolador para SIM800L
4. RST: pin de reinicio para SIM800L
5. RXD: Comunicación serie (Pin del receptor)
6. TXD: Comunicación serie (Pin de transferencia)

### Display 16x2 con adaptador I2C:

### Figura 13:

Display 16x2 con adaptador I2C:



Nota: Imagen del adaptador I2C a la izquierda y a la derecha Display LCD. Tomado de <https://www.arcaelectronica.com/blogs/tutoriales/display-lcd-16x2-con-modulo-i2c>

Una pantalla LCD (Liquid Crystal Display) es una forma muy sencilla y económica de leer información desde nuestro Arduino. Un formato muy habitual es el 16x2 (LCD1602), es decir, pantallas que muestran 2 filas de 16 caracteres cada una.

El I2C es un bus de comunicaciones que usa 2 líneas para enviar y recibir información y 2 más para alimentación. La gran cantidad de pines que tiene el display LCD 16x2 (16 pines) hace necesario en algunas ocasiones que tenga incorporado (o incorporar) un módulo para permitir la comunicación por I2C ya que con este necesita sólo 4 pines para funcionar.

El conjunto de ambos, además de las ventajas ya nombradas, cuenta con otras secundarias como un potenciómetro en el adaptador I2C para regular el contraste de la pantalla y la posibilidad de imprimir contrastes especiales.

Características:

- Fácil de usar con cualquier display de caracteres
- Compatible con interface HD44780
- Hasta 8 controladores en el mismo bus
- Ajuste de contraste con preset
- Control de encendido de backlight con jumper
- Máximo reloj: 100 KHz

**Pines qué los componen:**

**Figura 14:**

Pines del adaptador I2C



Nota: Descripción de los pines del adaptador I2C. Tomado de <https://www.zonamaker.com/arduino/modulos-sensores-y-shields/display-lcd-mediante-bus-i2c-twi-spi>

1. VCC = Alimentación de 5 Voltios
2. GND = Referencia negativa
3. SDA = Pin de datos
4. SCL = Pin de señal de reloj

**Tabla 3:**

Características Eléctricas Display LCD 16x2

Modos	Consumo
Alimentación	5V DC
Con luz de fondo al máximo	25 mA max

### ThingsPeak:

ThingsPeak es una aplicación de código abierto de Internet de la Cosas (IoT) que permite recoger y almacenar datos de sensores en la nube y desarrollar aplicaciones IoT. Además ofrece aplicaciones que permiten analizar y visualizar tus datos en MATLAB y actuar sobre ellos. Los datos de los sensores pueden ser enviados desde Arduino, Raspberry Pi, BeagleBone Black, entre otros Hardwares.

Algunas de las capacidades claves de ThingsPeak incluyen:

- Configurar fácilmente los dispositivos para que envíen datos a ThingSpeak utilizando los protocolos de IO más populares.
- Visualizar los datos de los sensores en tiempo real.
- Agregar datos bajo demanda de fuentes de terceros.
- Utilizar la potencia de MATLAB para dar sentido a tus datos de IO.
- Ejecutar análisis de IO automáticamente en base a horarios o eventos.
- Prototipo y construcción de sistemas de IO sin necesidad de crear servidores o desarrollar software web.
- Actúa automáticamente sobre tus datos y se comunica utilizando servicios de terceros como Twilio o Twitter.

Uno de los elementos centrales que componen a ThingsPeak es el “Canal ThingsPeak”, el cual almacena los datos que enviamos a ThingSpeak y se compone de los siguientes elementos:

- 8 campos para almacenar datos de cualquier tipo – Estos pueden ser usados para almacenar los datos de un sensor o de un dispositivo integrado.
- 3 campos de ubicación – Se pueden utilizar para almacenar la latitud, la longitud y la elevación. Estos son muy útiles para rastrear un dispositivo en movimiento.
- 1 campo de estado – Un mensaje corto para describir los datos almacenados en el canal.

## **Experimentación y diseño:**

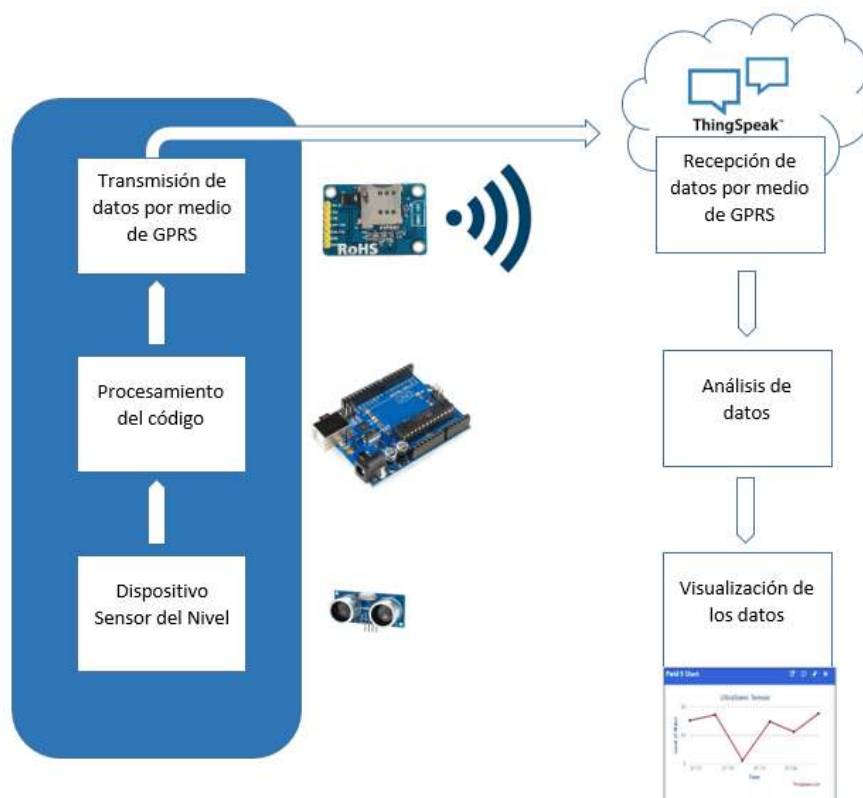
### **Descripción del Dispositivo:**

El desarrollo está proyectado sobre una Placa Microcontroladora Arduino UNO, como el controlador principal de las funciones, a la cual se le van anexar conformando la estructura del dispositivo, un sensor de ultrasonido HC-SR04, el cual por medio de las ondas de sonido que emitirá desde su cabezal, recibirá una onda reflejada que retornará desde el objeto sobre el que impacte, en este caso sobre el agua, y así medirá la distancia a la cual se encuentra el objeto contando el tiempo de emisión y la recepción.

A su vez se le incorporará al dispositivo principal un Módulo de Telefonía Móvil Sim800L EVB que opera en las bandas de frecuencias de los 850, 900, 1800 y 1900 MHz. Este módulo será el encargado de enviar la información obtenida por el Sensor Ultrasónico, hacia un Software de código abierto, donde se podrá visualizar todas las medidas realizadas cada cierto periodo de tiempo, y además como otra tarea, se encargará de dar la alerta por medio de la realización de llamados telefónicos teniendo en cuenta el nivel que sense en ese momento el sensor ultrasónico.

### **Figura 3**

Diseño Conceptual del Sistema



### Arduino UNO R3 y HC-SR04::

Como cerebro principal se utiliza el modulo Arduino UNO R3 debido a que los dispositivos adicionales que conforman la estructura del dispositivo son compatibles y también dispone de parámetros equilibrados y de memoria suficiente para realizar todas las tareas asignadas. Además y lo más importante posee un microcontrolador ATmega328P que cuenta con el nivel de procesamiento necesario.

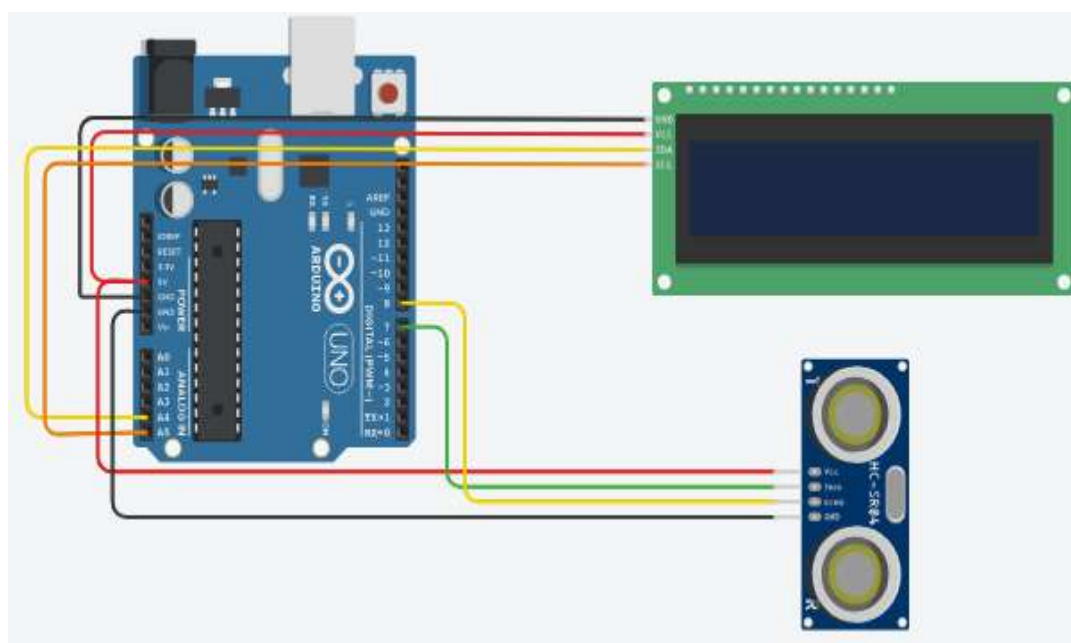
En cuanto al HC-SR04 la elección se debió a su gran precisión a la hora de realizar la medición y por su buena relación costo-calidad que presenta. Si bien existen mayor cantidad de sensores, este fue el que se pudo llegar a utilizar.

### Pruebas con HC-sr04:

Las primeras pruebas se realizaron en el Sensor ultrasónico, iniciando con el conexionado del sensor hacia el Arduino Uno R3 y conectando a su vez un display de 16x2, para verificar las distancias que toma el sensor. Para el conexionado se utilizó cables jumper de 30 Cm macho-hembra en los pines correspondientes, usualmente utilizados en experimentos relacionado con plataformas Arduinos.

**Figura 15:**

Conexión del Sensor y Display



Nota: Diagrama de conexionado realizado en software. Realizado y tomando en Tinkercad

**Tabla 4:**

Conexión de pines

Arduino UNO R3	HCSR-04
Pin Digital 7	Trigger
Pin Digital 8	Echo
5V	VCC
GND	GND
	<b>Display LCD I2C</b>

Display analogico A4	Pin SDA
Display analogico A5	Pin SCA
5V	VCC
GND	GND

Nota. Tabla de Conexionado entre el Arduino , el Sensor HCSR-04 y Display

En cuanto a la alimentación, en estas instancias se le suministró por medio de un transformador de 12V con una intensidad de 2A, lo cual generó en el pin de salida del Arduino los 5V necesarios para alimentar el display LCD y el Sensor Ultrasónico.

Luego de completar el conexionado se procedió a colocar el Sensor sobre una estructura recta, colocando frente a él una caja de tamaño pequeño para realizar pruebas de medición.

Al tener todo preparado de manera correcta, se comenzaron las pruebas del sensor, alejando y acercando en diferentes momentos el objeto a diferentes distancias y verificando si el cambio de la distancia impacta en el display LCD.

### Figura 16:

Prueba de Prototipo



Nota: Pruebas de medidas mediante la colocación de un objeto a diferentes distancias.

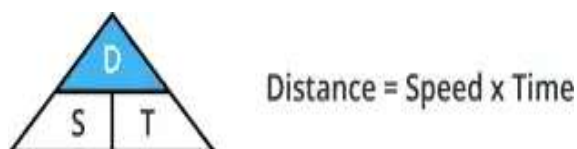
Como se puede observar en la **Figura 16**, el Sensor realiza la medición correspondiente y la misma impacta en el Display de 16x2.

### **Cálculo de la Distancia:**

Como se describió anteriormente, el ancho del pulso recibido se utiliza para calcular la distancia desde el objeto reflejado. Esto se puede resolver usando la sencilla ecuación distancia-velocidad-tiempo.

### **Figura 17**

Cálculo de Distancia



Nota. Gráfico que describe cómo obtener la ecuación de distancia. Tomado de <https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/>.

Para que quede más claro, supongamos que tenemos un objeto frente al sensor a una distancia desconocida y recibimos un pulso de 500  $\mu\text{s}$  de ancho en el pin de eco. Ahora calculemos qué tan lejos está el objeto del sensor. Para esto usaremos la siguiente ecuación.

$$\text{Distancia} = \text{Velocidad} \times \text{Tiempo}$$

Como dijimos tenemos un ancho del eco de 500  $\mu\text{s}$ , lo que quiere decir que tenemos tiempo, y la velocidad también la tenemos, que es la correspondiente a la velocidad del sonido, la cual es 340 m/s. A su vez se debe tener en cuenta que el pulso de eco indica el tiempo que tarda la señal en enviarse y reflejarse. Entonces, para obtener la distancia, debes dividir el resultado por dos.

$$\text{Distancia} = \text{Velocidad} \times \text{Tiempo} / 2$$

Y así de esta manera, concluye la explicación de cómo el dispositivo realiza el cálculo de la distancia de un objeto.

#### **Limitaciones del HC-SR04:**

Dentro de las pruebas que se realizaron con el sensor, si bien el dispositivo es fiable y preciso, a su vez se tuvo en cuenta ciertas limitaciones al momento de utilizarlo. Estas son:

- La distancia entre el Sensor y el objeto/Obstáculo no debe ser mayor a 4 metro (13 Pies)
- Elementos blandos como telas o lanas, no las detecta por lo tanto el eco no vuelve
- La orientación de la superficie a medir puede provocar que la onda se refleje, falseando la medición.
- El ángulo de incidencia sobre el objeto sea de máximo 45°, los objetos muy pequeños no los puede detectar con facilidad, o incluso objetos con superficies irregulares o muy suaves.

#### **Implementación del Módulo Sim800L:**

Como módulo de comunicación se seleccionó el Sim800L EVB o también conocido como V2, ya que este módulo trabaja mediante tecnología de comunicación GPRS, un tipo de red celular que sigue operando hasta en la actualidad y con una gran cobertura en la inalámbrica completa y velocidades de transferencia de entre 56 a 114 kbps mayoría de los países, trabajando en bandas de frecuencia de 850MHz, 900MHz, 1800MHz y 1900MHz. Este nos permite comunicarnos simplemente insertando una Sim perteneciente a algunos de los operadores de red celular, a través de SMS o mediante Internet.

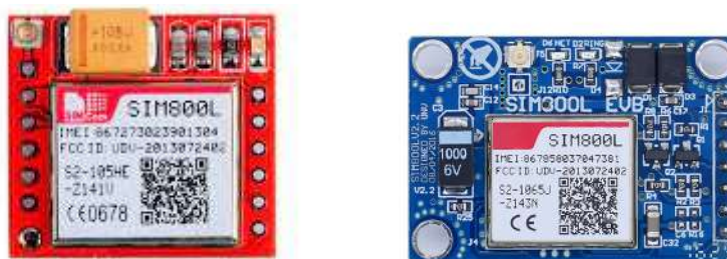
#### **Pruebas con Sim800L:**

Es importante destacar que un principio se decidió utilizar la primera versión del Sim800L, sin embargo por un problema inesperado de alimentación provocó que se dañe,

llevando a la obligación de invertir en otro. Es por eso que se decidió elegir el Sim800L V2, el cual requiere una alimentación de tensión de 5V con rafagas de intensidad máxima de 2 A al momento de transmitir los datos mediante GPRS, a diferencia del primero el cual requería una alimentación de 3,4 V ~ 4,4 V a una intensidad máxima de 2 A. La alimentación durante los periodos de pruebas se le brindó mediante alimentación externa al Arduino Uno R3, debido a que si bien el Arduino otorga los 5 V necesario de tensión no puede otorgar la intensidad necesaria, debido a que por características de diseño que permite entregar solo hasta un máximo de 1 A.

**Figura 18:**

Sim800L V1



Nota. Imagen a modo de comparación de la primera versión de sim800L y EVB. Tomado de <https://descubrearduino.com/sim800l-gsm/>

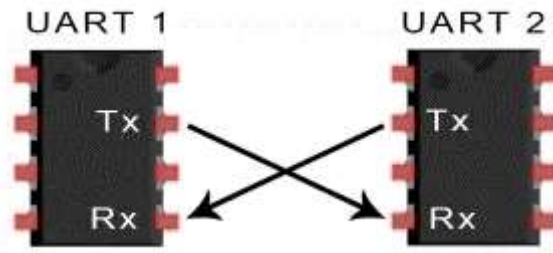
### Comunicación Serial Asíncrona:

Para la comunicación serial se utilizó cables jumper de 30 Cm macho-hembra, conectándolos en los pines 2 y 3 del Arduino Uno R3. Se debe tener en cuenta que el Sim800L EVB tiene la característica de comunicarse por medio del dispositivo UART (Universal Asynchronous Receiver-Transmitter), que controla los puertos serie. Gracias a esto y el uso de la librería correspondiente, se pudo utilizar los pines digitales 2 y 3 para la comunicación serial.

Cuando un dispositivo se comunica, su UART transmite en la línea "TX" y recibe datos por la línea "RX".

**Figura 19:**

Conexión UART

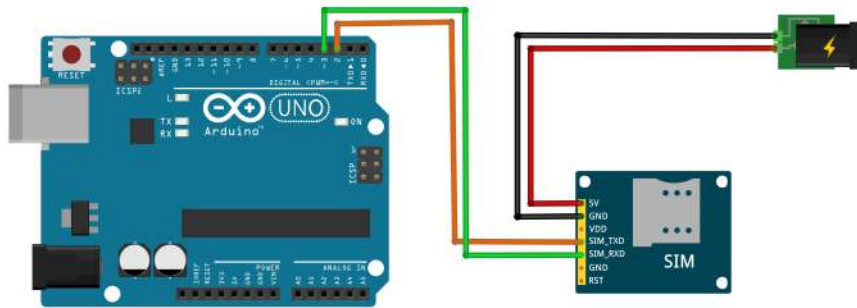


Nota. Conexión de Tx y Rx en protocolo UART. Tomado de <https://arduino.cl/como-configurar-la-comunicacion-uart-en-arduino/>

Cuando se conectan dos dispositivos de esta manera, un dispositivo envía un carácter usando su línea TX y el otro la recibe por su línea RX y viceversa. La conexión correcta de dos dispositivos para su comunicación serie asíncrona, se muestra a continuación.

**Figura 20:**

Conexión de Comunicación y Alimentación del Sim800L



Nota. Conexión de comunicación serial asíncrona y de alimentación externa. Realizado con Fritzing.

**Tabla 5:**

Conexión de pines

Arduino UNO R3	Sim800L EVB
----------------	-------------

Pin Digital 2 (Rx)	TX
Pin Digital 3 (Tx)	Rx

Nota. Tabla de Conexionado entre el Arduino y el Sim800L EVB

Si se observa la **Figura 2** se puede señalar que el pin 2 configurado como RX del Arduino Uno R3 está conectado al pin TX del Sim800L V2 y viceversa. El Arduino envía los datos por el pin TX y el Sim800L los recibe por el pin RX.

Se debe tener en cuenta qué en este tipo de comunicación, solo un dispositivo puede tener el control de la línea UART en un determinado momento. Si más de un dispositivo transmite en la línea de RX de otro dispositivo, esto provocará un error llamado “contención de bus”, lo que quiere decir que habrá varios dispositivos compitiendo por la línea de datos y no se podrá enviar ni recibir.

### Alimentación del Proyecto:

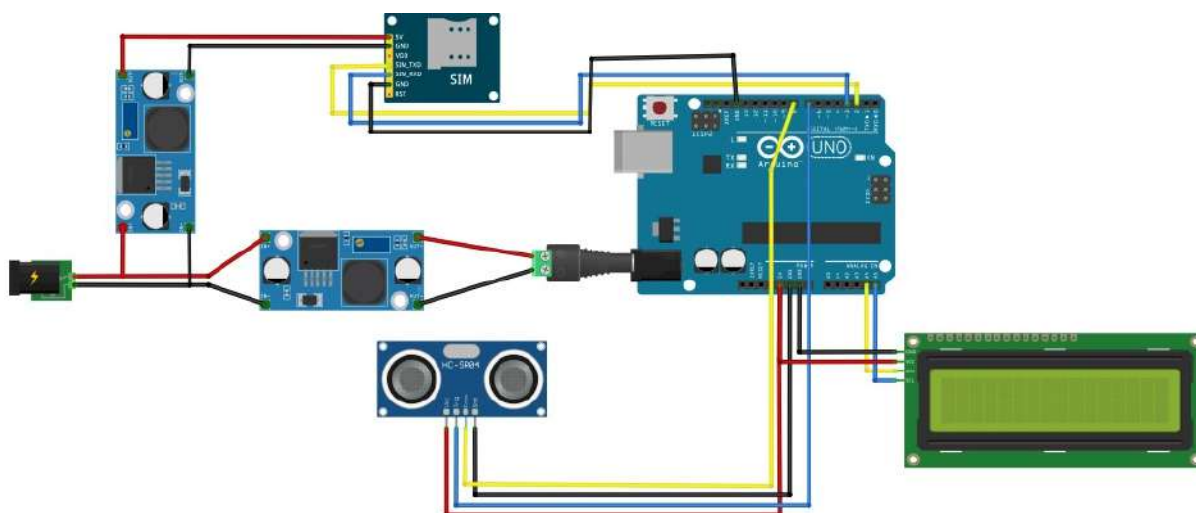
En un principio cada componente del dispositivo se decidió alimentarlo desde el Arduino UNO R3, proporcionándoles con una Fuente de alimentación de 12V con una intensidad de corriente de 2A. En ese momento no se había tenido en cuenta que solo el Sim800L EVB consume 2A de intensidad de corriente por cada rafaga de transmisión, provocando que cada vez que módulo Sim800L intentase conectarse se reiniciara. Por lo tanto en cada prueba realizada de cada elementos que componen el dispositivo en concreto, se fue suministrado alimentación de manera individual a cada dispositivo. En este punto se decidió encontrar una manera de alimentar el proyecto de forma conjunta, debido a que el arduino tiene un pinout de 5 V pero que solo puede llegar a una intensidad máxima de corriente de 1 A, pudiendo sólo alimentar el Sensor de ultrasonido y HCSR-04.

Es por eso que se decidió utilizar una fuente de alimentación de 19V con una Intensidad de corriente de 3,7 A, donde de él saldrán dos líneas de alimentación, una para el Arduino UNO R3 y otro para el Sim800L EVB. Para ambos se le adaptara a la línea

saliente de la fuente de alimentación dos LM2596 Step-Down para qué de los 19V que suministra la Fuente, obtengamos 12V y 5V para cada dispositivo.

**Figura 21:**

Conexión de alimentación



Nota. Diagrama del Conexión de alimentación

### Montaje del Hardware:

#### Carcasa del dispositivo:

Para el resguardo y protección del dispositivo se utilizó una caja estanca PR 1006/81 de 161x161x81 mm. A la misma se le debió realizar ciertas modificaciones para la adaptación de cada módulo o equipo del Dispositivo. Estos se colocaron ordenados de manera que pueda caber el dispositivo armado.

**Figura 22:**

## Caja Estanca



Nota. Caja estanca IPR 1006/81

Para empotrar el Display LCD se procedió a realizar un abertura encima de la cubierta de la caja estanca de 64 mm de largo y 26.8 mm de ancho.

## Figura 23:

Tapa modifica



Nota. tapa con el Display empotrado.

Luego se procedió a montar todos los dispositivos sobre un placa prototipo para circuitos perforada de pertinax. Para montarlo sobre la placa se atornillo sobre ella. Una vez realizado el montaje, se continuó montado dentro de la caja estanca.

## Figura 24:

## Dispositivos Montados



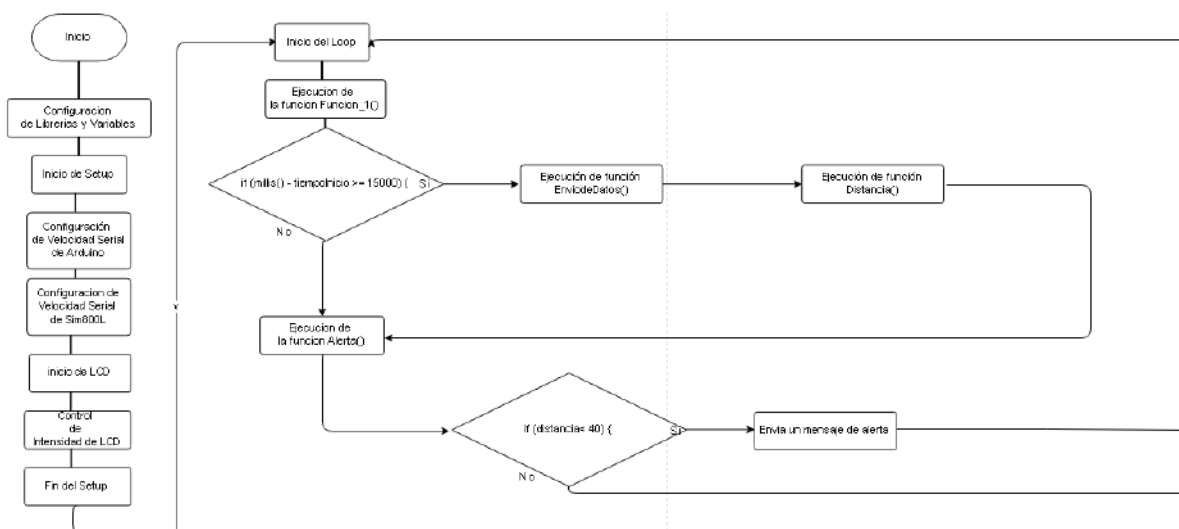
Nota: Equipos montados en Placa prototipo y resguardado en caja estanca.

## Diagramas de Flujo:

### Programa Principal:

### Figura 25

Diagrama de flujo de Programa Principal

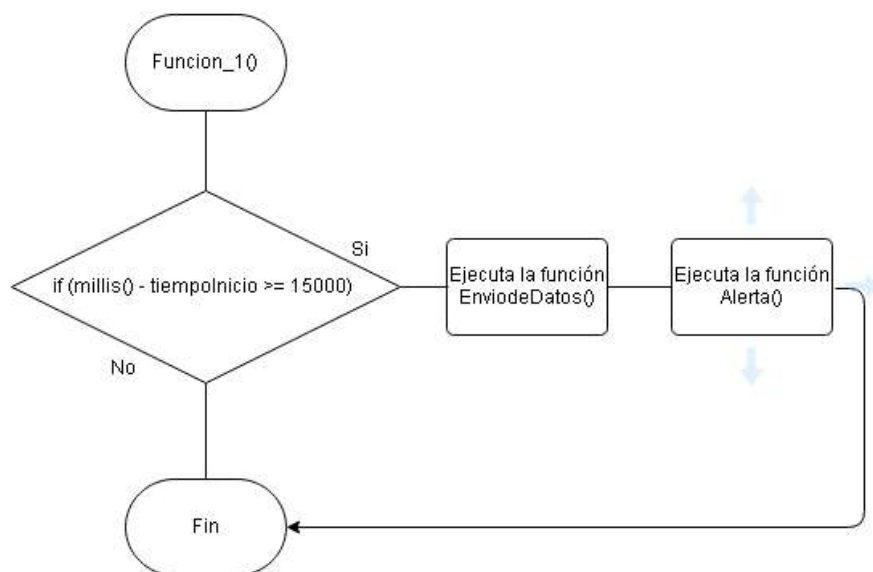


**Diagrama de las funciones:**

En esta función una vez que se ejecute, a su vez cada 15 minutos ejecutar la función `EnvíeDatos()`, la cual establece la conexión GPRS y envía los datos necesario hacia la API de ThingsPeak, y a su vez se ejecuta la función `Alerta()`, la encargada de verificar que si se cumple cierta condición, enviar un mensaje de alerta.

**Figura 26**

Diagrama de flujo de funciones



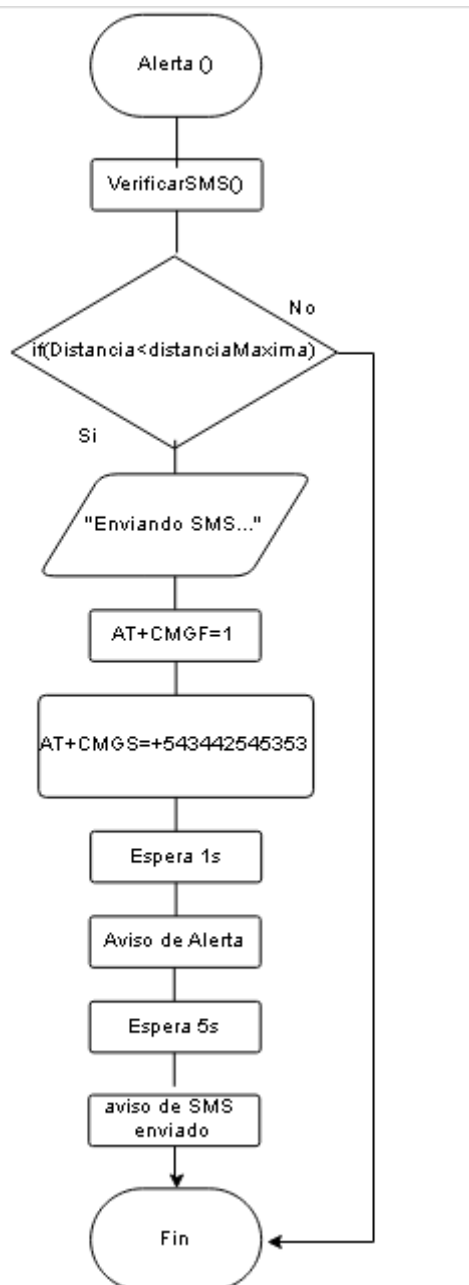
En este diagrama de flujo de puede observar que en la configuración del software se utilizó millis(), el cual devuelve la cantidad de tiempo que transcurre desde que comienza a ejecutar el código el Arduino UNO R3. Gracias a esta función realiza la ejecución de medición y envío de datos cada 15 minutos.

**Diagrama de Alerta():**

Esta función una vez que se ejecuta, tiene como objetivo dar aviso al usuario de que el río se encuentra en un nivel cercano de peligro. En este caso se utilizó delay() entre cada función, el cual es un código bloqueante, ya que no hacía diferencia en el rendimiento del sistema.

**Figura 27:**

Diagrama de Flujo de la función Alerta():

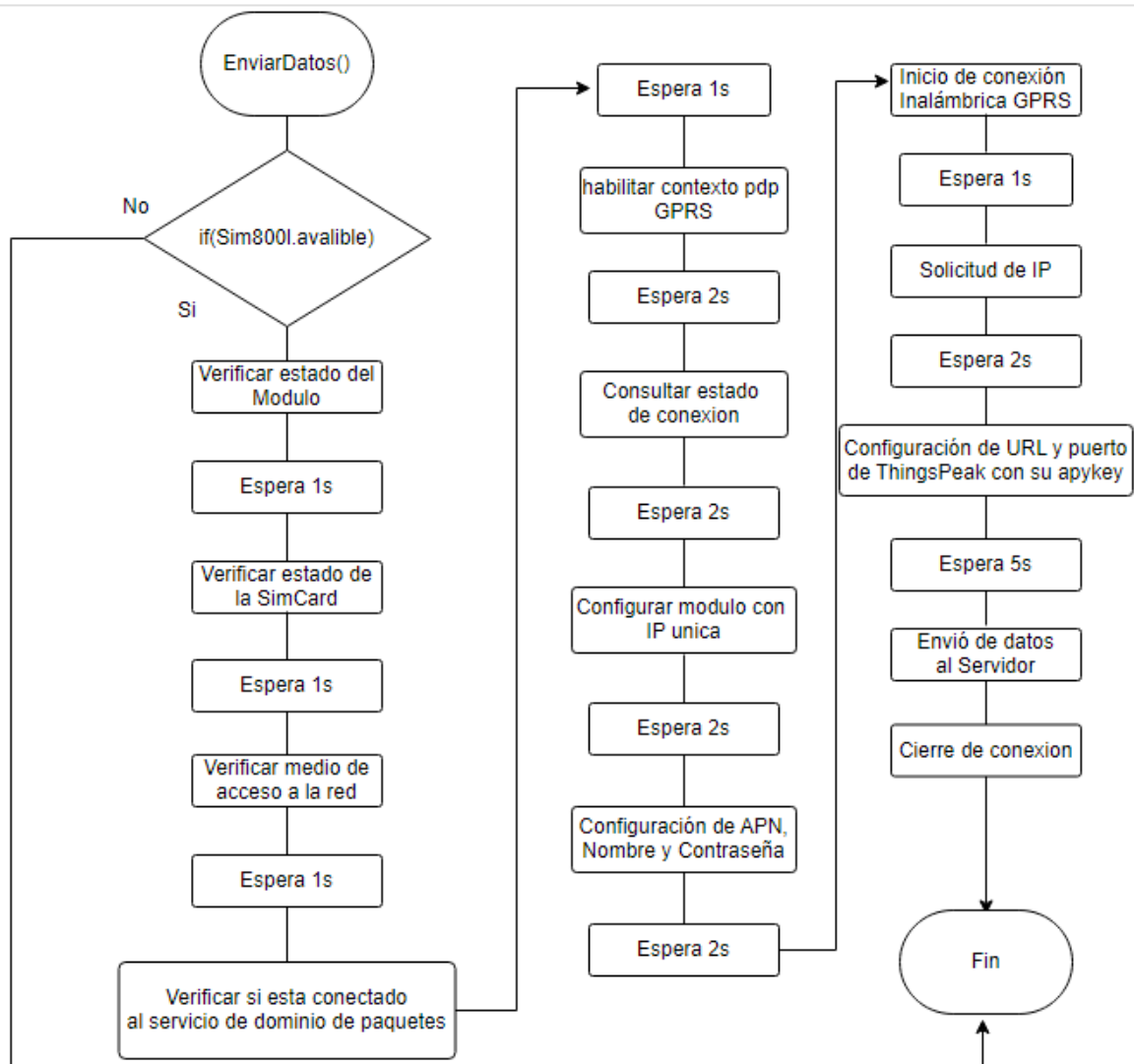


**Diagrama de EnvioDatos()**

En cuanto a la función nombrada, tiene el objetivo de ejecutar todos los comandos AT necesario para saber el estado del módulo Sim800L EVB , así como realizar la conexión con el con la red GPRS y envío de datos hacia la API de ThingsPeak. Al igual que la función alerta() se utilizó código bloqueante, es decir Delay(), para esperar entre cada ejecución a realizar de los comando AT.

**Figura 28:**

Diagrama de flujo de la función de envío de datos



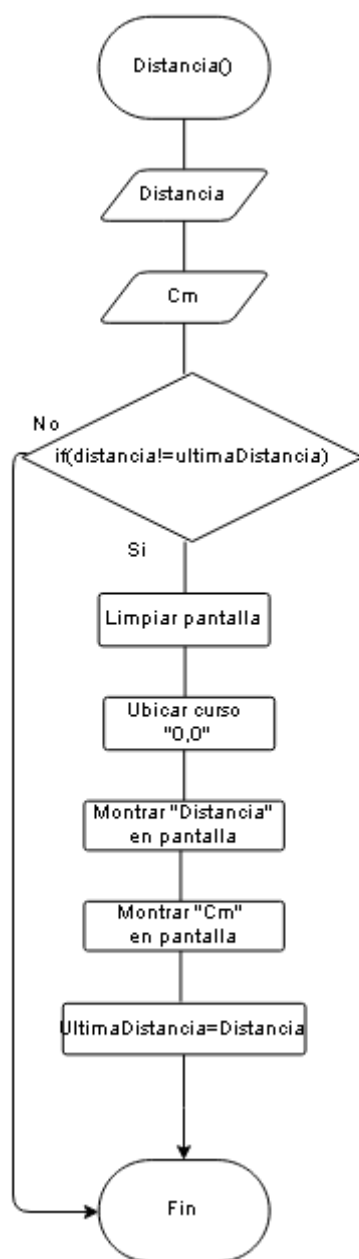
Entre cada función que realiza para la conexión a la red GPRS, en la parte de bucle del código, llamamos a la función personalizada llamada `updateSerial()` que espera continuamente cualquier entrada del monitor serie y la envía al módulo Sim800L mediante el pin Rx del módulo.

**Diagrama de Distancia():**

Esta función consiste en tomar la distancia e impactar la medida tomada en el Display LCD de 16x2. En esta función se hace uso de la librería `LiquidCrystal_I2C.h`, la cual incluye nos permite la utilización del Display LCD 16x2, y a su vez la librería `Wire.h`, que nos permite facilitar la conexión el display mediante un adaptador I2C, resumiendo en 2 pines analógicos ( SDA y SCL o Pin A4 y A5).

**Figura 29**

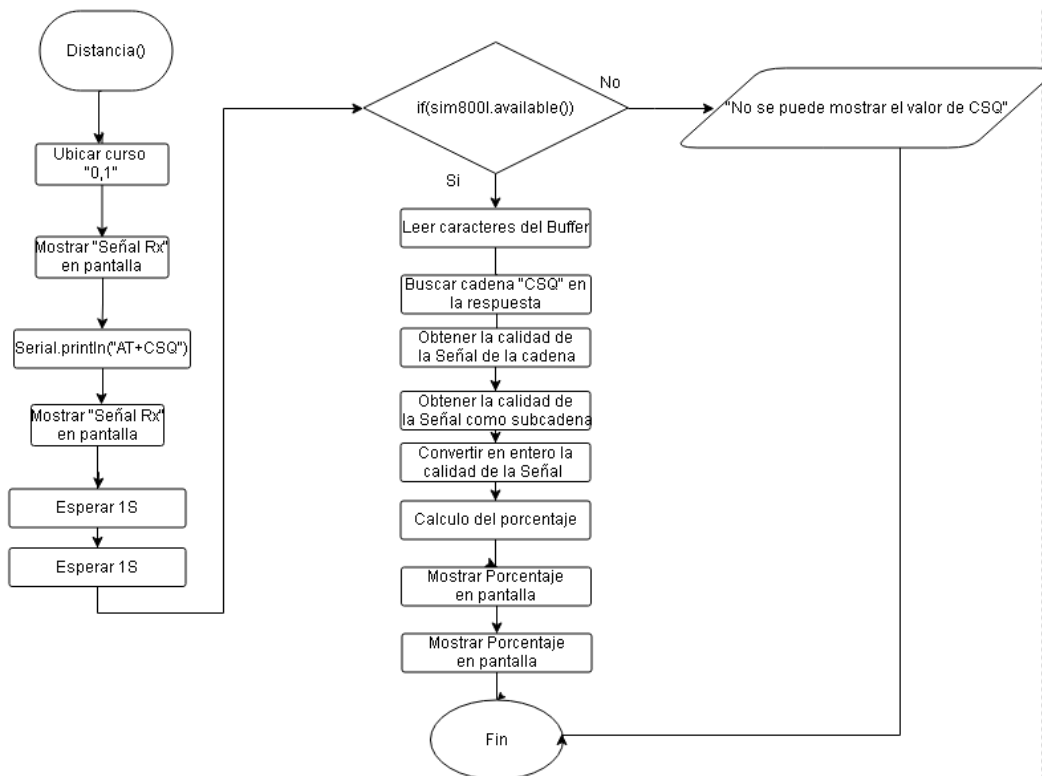
Diagrama de flujo de Distancia()



**Diagrama de Signal():**

**Figura 30:**

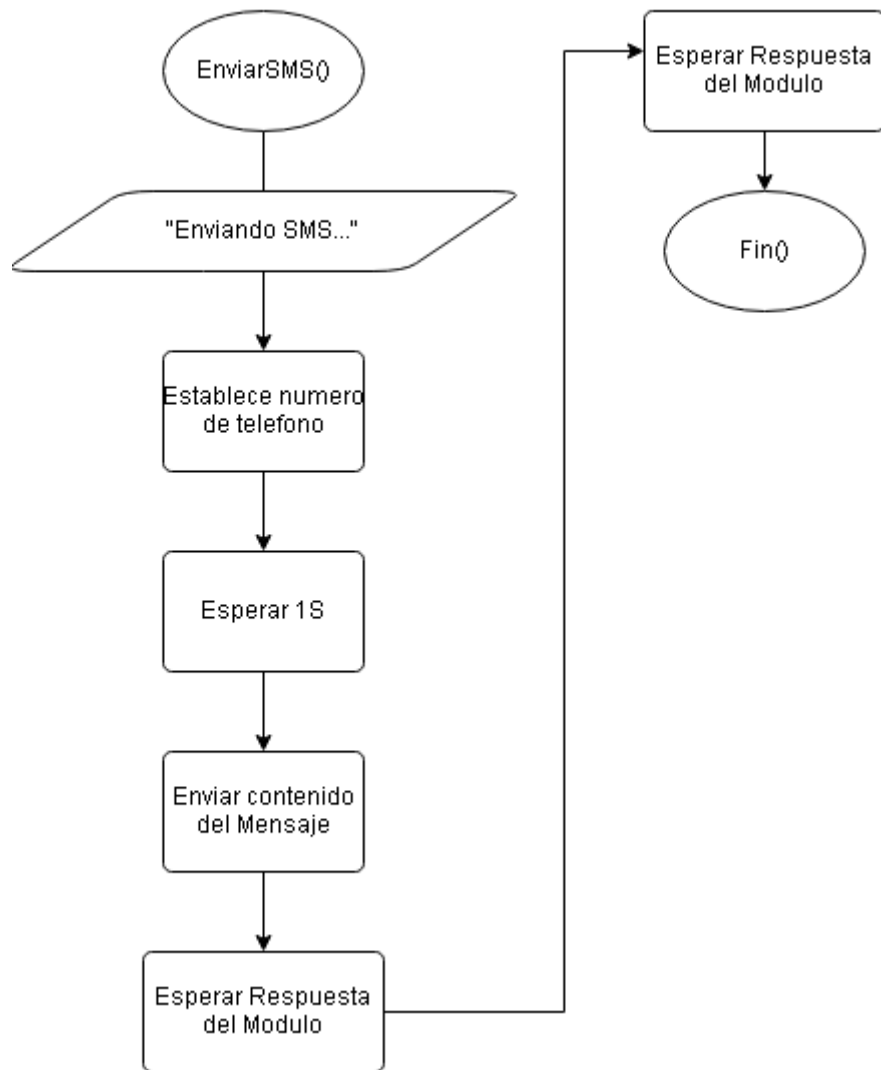
Diagrama de flujo de Signal()



**Diagrama de EnviarSMS():**

**Figura 31:**

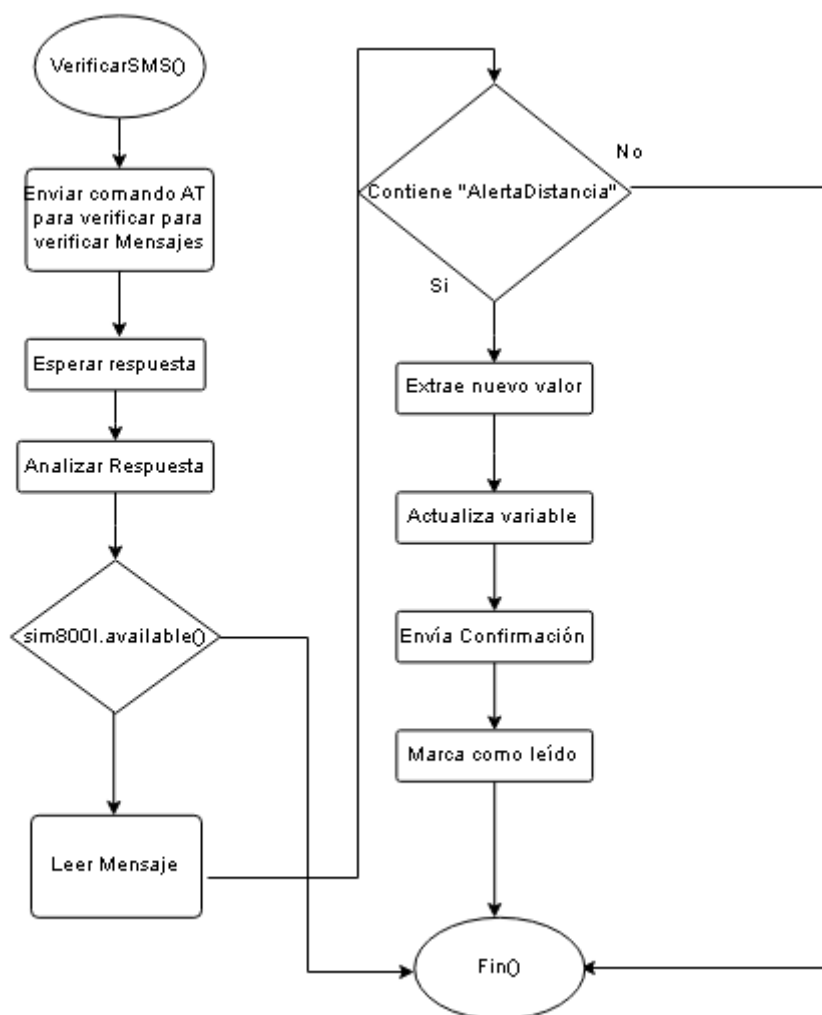
Diagrama de flujo de EnviarSMS()



**Diagrama de VerificarSMS():**

**Figura 32:**

Diagrama de flujo de VerificarSMS()



## Diseño del Software:

### Arduino IDE:

Como se nombró desde un principio se utilizará el Entorno de Desarrollo Integrado de Arduino. Este software nos brinda las herramientas necesarias para implementar el código de nuestro proyecto, con una interfaz amigable y fácil de entender. Este está conformado por un editor de texto para escribir nuestro sketch, una consola de error y un área con los menús y los botones que realizan las funciones más comunes como son abrir sketch, guardar sketch, compilar y cargar programa, y además se puede utilizar cualquier dispositivo arduino. A continuación se explica cada parte de la programación.

## Funciones del Programa:

### Void loop():

La función del método loop que implementa Arduino será recibir una serie de parámetros y órdenes del servidor o placa Arduino, como sera llamar a la función Funcion1() la cual activa a su vez las funciones que se encargan de tomar las distancia y de realizar la comunicación con la API de Thingspeak y tambien la funcion Alerta(), que se encargará de realizar un mensaje de alerta indicando que la altura del río se encuentra en niveles preocupantes.

Código:

```
void loop() {  
  
    Funcion_1();  
    alerta();  
}
```

### Void Alerta():

Una vez que es llamada la función alerta() esta se encargará de activar los mensajes de texto, y luego proceder a mandar un mensaje de alerta, indicando las distancias críticas del nivel del río, siempre y cuando se cumpla la condición.

Código:

```
void alerta() {
```

```

verificarSMS();
if (distancia < distanciaMaxima) {
  Serial.println("Enviando SMS...");
  sim8001.print("AT+CMGF=1\r");
  delay(1000);
  sim8001.println("AT+CMGS=\"+543442545353\"");
  delay(1000);
  sim8001.println("Alerta, El Nivel es critico");
  delay(1000);
  sim8001.println((char)26);
  delay(100);
  sim8001.println();
  delay(5000);
  Serial.println("SMS enviado");
} else {
  sim8001.println("No hay ningun peligro");
}
}

```

Al iniciar esta función comenzará mostrando un mensaje que dirá “Enviando Mensaje...” y a través de comando AT “AT+CMGF=1” activará la función de SMS permitiendo al módulo enviar mensajes instantáneos. A partir de se indica el número al cual se enviar el mensaje mediante el comando “AT+CMGS” y luego procederá a enviarlo.

### Void Funcion():

En esta función se utiliza la función millis() para que después de que pasen 15 minutos ejecute la función EnviodeDatos() y Distancia().

Código:

```

void Funcion_1() {
  if (millis() - tiempoInicio >= 15000) {

```

```

    EnviodeDatos();
    Distancia();
    Signal();
    tiempoInicio = millis();
  }
}

```

El Arduino millis es una función que se usa para medir el tiempo. Está instrucción te da el tiempo en milisegundos desde que se encendió la tarjeta Arduino UNO R3. Una de los principales usos es el determinar el tiempo que ha pasado entre dos o más procesos del programa principal. En nuestro caso se definirá la variable del tipo unsigned long nombrada com tiempoInicio, que se utilizara para almacenar los valores que arroja millis(), y que una vez guardado el primer valor, cuando llegue a 15000 se ejecutarán las funciones dentro de la sentencia **if**.

### **Void Distancia()**

Esta función, en palabras simples, se encarga de devolver la medición que realice el sensor ultrasónico HC-SR04. Se comienza con la variable definida como float distancia, donde dentro se almacenará la función measureDistanceCm(), que se utiliza para medir y mostrar la distancia en centímetros. Si la distancia es mayor que 400 cm, devolverá un valor negativo.

Código:

```

void Distancia() {
  Serial.print("Distancia:");
  Serial.print(distancia);
  Serial.print("cm");
  Serial.println();

  if (distancia != ultimaDistancia) {
    lcd.clear();
    lcd.setCursor(0, 0);

    lcd.print("Distancia:"); // Prints string "Distance" on the LCD
    lcd.print(distancia);
    lcd.print("cm");
    ultimaDistancia = distancia;
  }
}

```

}

Luego continúa con un condicional, el cual si se cumple, es decir si la última medida obtenida es diferente a la anterior, llamara a la funciones `lcd.print()`, las cuales se utilizan para mostrar los valores de las distancia en la pantalla LCD y a través de la función `lcd.setCursor()` se indica la posición del cursor, a partir de cual se mostrará en nuestro caso las distancia medidas.

### Void Signal():

La función `Signal()` al ser ejecutada se lee la respuesta completa del módulo SIM800L y se busca la cadena "+CSQ: " en la respuesta. Luego, se extrae la calidad de la señal como una subcadena y se convierte a un número entero. La función `map()` se utiliza para mapear el valor de la calidad de la señal de la escala original (0-31) a un porcentaje (0-100). Finalmente, se muestra el porcentaje en el LCD.

Código:

```
void Signal() {
  lcd.setCursor(0, 1);
  lcd.print("Señal RX: ");

  sim800l.println("AT+CSQ");
  delay(1000);

  if (sim800l.available()) {
    // Leer la respuesta completa del módulo SIM800L
    String response = sim800l.readStringUntil('\n');

    // Buscar la cadena "+CSQ: " en la respuesta
    int csqIndex = response.indexOf("+CSQ: ");
    if (csqIndex != -1) {
      // Obtener la calidad de la señal como una subcadena después de
"+CSQ: "
      String rssiString = response.substring(csqIndex + 6);

      // Convertir la calidad de la señal a un número entero
      int rssi = rssiString.toInt();

      // Calcular el porcentaje de señal
```

```

    int reception = map(rssi, 0, 31, 0, 100);

    // Mostrar la recepción en el LCD
    lcd.print(reception);
    lcd.print("%");
} else {
    lcd.print("No se pudo obtener la CSQ");
}
}

```

**Void verificarSMS():**

La función verificarSMS() se llama antes de la lógica de la alerta. Si se recibe un mensaje SMS que contiene "AlertaDistancia", se actualiza la variable distanciaMaxima.

Código:

```

void verificarSMS() {
    // Verificar si hay mensajes SMS disponibles
    sim8001.println("AT+CMGL=\"REC UNREAD\",1");
    delay(1000);

    while (sim8001.available()) {
        String response = sim8001.readStringUntil('\n');
        if (response.indexOf("AlertaDistancia") != -1) {
            // Si el mensaje contiene "AlertaDistancia", procesar el nuevo
valor
            int index = response.indexOf(":") + 1;
            if (index > 0) {
                String nuevoValor = response.substring(index);
                distanciaMaxima = nuevoValor.toInt();
                Serial.print("Nueva distancia máxima: ");
                Serial.println(distanciaMaxima);
                enviarSMS("Distancia máxima actualizada a " + nuevoValor);
            }
        }
    }
}

```

**Void EnviarSMS():**

La función enviarSMS() también se ha movido a una función separada para facilitar el envío de mensajes SMS desde diferentes partes del código.

Código:

```

void enviarSMS(String mensaje) {
    Serial.println("Enviando SMS...");

    // Establecer el número de teléfono al que se enviará el SMS
    sim8001.println("AT+CMGS="+543442545353+"");
    delay(1000);

    // Enviar el contenido del mensaje SMS
    sim8001.print(mensaje);
    sim8001.write((char)26); // Ctrl+Z para finalizar el mensaje
    delay(100);

    // Esperar la respuesta del módulo SIM800L
    while (sim8001.available()) {
        Serial.write(sim8001.read());
    }

    Serial.println("SMS enviado");
}

```

### **Void EnviodeDatos():**

La función EnviodeDatos() al ser ejecutada, realiza todas las funciones necesarias para la comunicación y envío de los datos hacia la API de ThingsPeak.

Código:

```

if (sim8001.available())

    Serial.write(sim8001.read());

    sim8001.println("AT");
    delay(1000);

    sim8001.println("AT+CPIN?");
    delay(1000);

    sim8001.println("AT+CREG?");

```

```

delay(1000);

sim8001.println("AT+CGATT?");
delay(1000);

sim8001.println("AT+CIPSHUT");
delay(2000);

sim8001.println("AT+CIPSTATUS");
delay(2000);

sim8001.println("AT+CIPMUX=0");
delay(3000);
MostrarDatosSeriales();

sim8001.println("AT+CSTT=\"datos.personal.com\"");
delay(1000);
MostrarDatosSeriales();

sim8001.println("AT+CIICR");
delay(3000);
MostrarDatosSeriales();

sim8001.println("AT+CIFSR");
delay(2000);
MostrarDatosSeriales();

sim8001.println("AT+CIPSPRT=0");
delay(3000);
MostrarDatosSeriales();

sim8001.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \" 80\"");
delay(6000);
MostrarDatosSeriales();

sim8001.println("AT+CIPSEND");
delay(4000);
MostrarDatosSeriales();

```

```
String          datos          =          "GET
https://api.thingspeak.com/update?api_key=0WZ3T8A8ZPUF0SZ3&field1=0"  +
String(distancia);          Serial.println(datos);
sim8001.println(datos);

delay(4000);
MostrarDatosSeriales();

sim8001.println((char)26);
delay(5000);
sim8001.println("AT+CIPSHUT");//
delay(5000);
MostrarDatosSeriales();
}
```

Se explicará a continuación la tarea de cada comando AT ejecutado:

- AT: Este comando, al ejecutarlo responderá cual es el estado del módulo.
- AT+CPIN: Este comando, devolverá el estado de la Simcard.
- AT+CREG: Este comando será necesario para saber por qué medio se conecta a la celda.
- AT+CGATT: Verifique el estado del servicio de paquetes adjunto. '0' implica que el dispositivo no está conectado y '1' implica que el dispositivo está conectado.
- AT+CIPSHUT: Cierra el contexto PDP del GPRS.

El contexto PDP es un recurso necesario para conectarse a internet según lo establece el estándar 3GPP quien define cómo se debe realizar la comunicación a nivel físico y lógico en los equipos de tecnología celular. A continuación se muestra el procedimiento de activación definido por 3GPP. En nuestro caso realizaremos el transporte de datos mediante Protocolo IP.

- AT+CIPSTATUS: devuelve el estado de conexión actual. Este comando devuelve el estado del servidor aplicable, el estado del cliente, el número de conexión (para IP múltiple) y la información del portador GPRS.
- AT+CIPMUX=0: Se ejecuta para configurar el equipo con IP única o múltiple. En nuestro caso utilizaremos IP única.

- AT+CSTT =\"datos.personal.com\":En este caso se configuró el APN, o también llamado Nombre del Punto de Acceso para ingresar a la red, en nuestro caso de Personal.
- AT+CIICR:Mediante este comando iniciamos la conexión inalámbrica mediante GPRS.
- AT+CIFSR:A través de este obtenemos la IP local.
- AT+CIPSPRT=0:Establece un indicador '>' al enviar datos.
- AT+CIPSPRT=\"TCP\", \"api.thingspeak.com\", \" 80\":Indicamos el tipo de conexión, url o dirección IP y puerto al que realizamos la conexión.En nuestro caso es a la API de ThingsPeak.

```
String          datos          ="GET
https://api.thingspeak.com/update?api_key=0WZ3T8A8ZPUF0SZ3&field1=0"          +
```

String(distancia) : En esta línea de código se realizó la concatenación de datos.Por un lado la url de ThingsPeak donde vamos a ir actualizando los datos a medida que vayamos subiendo con la api key de escritura (Obtenido del canal de ThingsPeak creado en la página) y las distancias que toma el sensor ultrasónico.Una vez realizado se enviar al servidor remoto, donde luego se ejecuta la línea con el código `sim8001.println((char)26);` lo que es igual a CTRL + Z en ASCII.Por último último se ejecuta el comando AT+CIPSHUT para cerrar la conexión.

### **Void MostrarDatosSeriales():**




Esta función ejecuta las líneas de código necesarias para mostrar los datos que se van subiendo a la API de ThingsPeak.





Código:


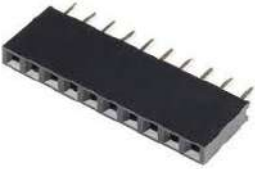


```
void MostrarDatosSeriales ()
{
  while (sim8001.available() != 0)
    Serial.write(sim8001.read());
  delay(5000);
}
```


**Presupuesto:**

A continuación se indica, a modo de referencia, los costos de los gastos aproximados que se llevaron a cabo para hacer posible la realización del proyecto. Para esto se realizó una tabla con sus respectivos precios y la cantidad de materiales que se compró. Cabe destacar que ya se contaba con la posesión de algunos insumos.

Producto	Fecha de compra	Cantidad	Costo por Unidad	Total
Plancha de pertinax perforada 10x20 	9/11/2023	1	\$ 4700	\$4700
Alambre de Estaño 	24/01/2023	2	\$3500	\$7000
Caja estanca PR 1006/81 161x161x81 	20/09/2023	1	\$5400	\$5400

<p>Cables Jumpers Hembra-Macho Macho-Macho</p> 	<p>27/10/2023</p>	<p>1</p>	<p>\$2200</p>	<p>\$2200</p>
<p>Arduino UNO R3</p> 	<p>12/07/2022</p>	<p>1</p>	<p>\$8000</p>	<p>\$8000</p>
<p>Sensor Ultrasonico HC-sr04</p> 	<p>12/07/2022</p>	<p>1</p>	<p>\$395</p>	<p>\$395</p>
<p>Fuente Step-Down Dc Dc Lm2596 1.25v 35v 3a Dc-Dc Lm2596</p> 	<p>16/07/2022</p>	<p>2</p>	<p>\$475</p>	<p>\$950</p>

<p>Modulo Celular Gsm Gprs Sim800l Sim800 Antena</p> 	<p>10/09/2022</p>	<p>1</p>	<p>\$3352</p>	<p>\$3352</p>
<p>Tira de Pines Hembra</p> 	<p>13/10/2022</p>	<p>3</p>	<p>\$673.20</p>	<p>\$2019.6</p>
<p>Módulo LCD 16x2 con Serial I2C</p> 	<p>9/10/2023</p>	<p>1</p>	<p>\$5890.50</p>	<p>\$5890.5</p>
<p>Ficha Jack Macho 5.5x2.1 mm</p> 	<p>19/09/2022</p>	<p>1</p>	<p>\$421</p>	<p>\$421</p>

<p>Ficha Jack Hembra 5.5x2.1 mm</p> 	<p>19/09/2022</p>	<p>1</p>	<p>\$421</p>	<p>\$421</p>
<p><b>Gastos envios</b></p>		<p>1</p>	<p>\$2394,12</p>	<p>\$2394,12</p>
<p><b>Total</b></p>				<p>\$43143.22</p>

**Resultados:**

A continuación se llegara a explicar cuál fue el resultado obtenido de la realización y puesta en marcha del dispositivos.

Para comenzar se conectó la fuente de alimentación a la red de energía verificando que encienda de manera correcta, lo cual se comprobó, y una vez encendido se pudo apreciar en el display montado que se encontraba con una medida realizada en las pruebas anteriores. Además se pudo observar que cada cierto tiempo sensaba una nueva medición, impactando el cambio en el display.

**Figura 33:**

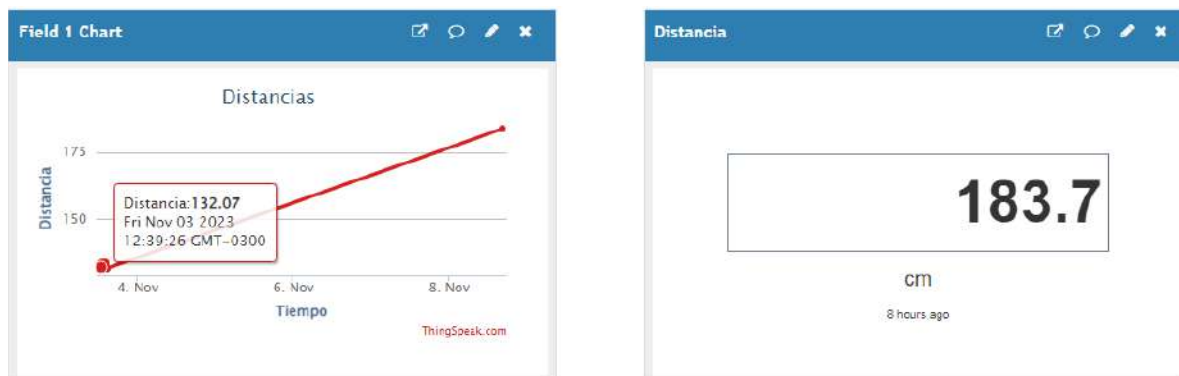
Dispositivo puesto en marcha



Se podría decir, teniendo en cuenta los objetivo a los cual se aspiraba, que se logró obtener un dispositivo, que obtenga de manera simple a que distancia se encuentra el río, y pudiendo registrar esa información de manera correcta en una interfaz de programación de aplicación como lo que es Thingspeak, pudiendo impactar todas las medidas realizadas en un ciclo de tiempo.

**Figura 34:**

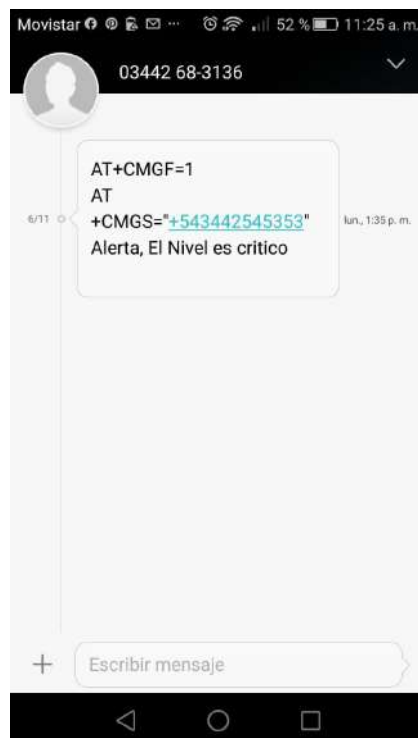
Registro de medidas contempladas en ThingsPeak



Además de lo ya nombrado, también se pudo comprobar que a una cierta distancia de peligro, el módulo Sim800L se activa y envía un mensaje de alerta a el celular ya preconfigurado.

**Figura 35:**

Mensaje de alerta enviado al celular



## Propuestas de modificaciones futuras:

En la siguiente sección se explica las modificaciones que se le puede llegar a realizar al dispositivo con el objetivo de evolucionar, mejorar su funcionamiento y su rendimiento a largo plazo.

- Debido a que este es un prototipo de Monitoreo y registro del nivel del río, se podría llevar a un siguiente nivel en un futuro y transformarlo en un Sistema de Alerta temprana. Para eso se debería agregar nuevos instrumentos de medición, como un sensor de humedad, sensor de temperatura y un caudalímetro.
- Optar por otras formas de alimentación, debido que por lo general, dispositivos como este se encuentran en lugares alejados de las zonas urbanas. Se podría utilizar pantallas solares o alimentación generada por la propia corriente del agua (Energía Hidráulica).
- Se podría agregar un botón de encendido y apagado
- Además del módulo GSM/GPRS se podría añadir un módulo wifi, como lo que es el ESP8266, como sistema de comunicación de apoyo en caso de que le ocurra algo al Sim800L EVB.
- Se podría anexar otra herramienta de alerta, como una sirena o realizar una llamada a los celulares cercanos cuando el nivel es crítico.
- Implementar mejor la codificación: Se debe tener en cuenta que código realizado en el proyecto está hardcodeado, lo que quiere decir que se escribe todos los valores sobre el código, sin poder realizar ningún cambio desde una fuente externa. En el caso que se requiera realizar una modificación, se debe interactuar directamente con el código.
- Para la puesta en marcha real, se podría optar por un mejor sensor ultrasónico, como lo es el Jsn-sr04t, ya que tiene características adicionales, como por ejemplo que posee una protección contra agua para poder tener usos rudos o acuáticos, siendo útil en aplicaciones donde el sensor estará expuesto a la intemperie, además puede soportar temperaturas extremas. En cuanto a las características es similar al HC-SR04

Características:

- Dispositivo base: JSN SR04T
- Modelo: AJ-SR04M
- Voltaje de Operación: 3V a 5V DC
- Corriente de trabajo: 30mA
- Rango de detección: 25 a 450 cm
- Zona muerta: < 20 cm
- Precisión: 3mm
- Frecuencia de emisión acústica: 40KHz
- Duración mínima del pulso de disparo (nivel TTL): 10  $\mu$ S.
- Tiempo mínimo de espera entre una medida y el inicio de otra: 20 ms.

- Ángulo de detección: menor a 50°
- A prueba de agua (parte delantera)
- Dimensiones PCB: 40 mm x 28 mm
- Longitud del cable: 2.5 m
- Temperatura de trabajo: -10°C hasta 70°C

## **Conclusión:**

Después de haber realizado la investigación, incluyendo tareas de armado y configuración, se llegó a conclusión que este tipo de experiencias son más que enriquecedores, generando más conocimiento y además pudiendo decir que las tecnologías, como lo que son las telecomunicaciones, son necesaria para actuar frente a problemáticas que en épocas anteriores no se podía realizar. La implantación de IoT tanto en estas tareas, como las de industria y en la vida en general, son algo positivo ya que nos facilitan a la hora de operar cualquier tipo de tecnología, permitiéndonos en la mayoría del tiempo mantener un orden y control de ciertos parámetros.

Desde un punto de vista del proyecto planteado y llevado a cabo, se puede decir que fue un reto, ya que si bien la tecnología Gsm/Gprs es un tipo de comunicación que hasta el día de hoy sigue vigente y tiene una gran cobertura en el área, es obsoleta, y por lo tanto se podría implementar otro tipo de medio de conexión o comunicación para realizar este tipo de dispositivos.

Por otro lado quiero destacar que los beneficios, como la versatilidad y facilidad de utilizar una Api como Thingspeak permitiendo realizar monitoreo diferente tipos de datos de información, ya sea de temperatura, medición de distancias, humedad, entre otras, registrando y así decidiendo cómo actuar frente a la información sensada.

Por último gracias a la utilización de estos dispositivos o similares, acompañados de un sistema de registro, se puede llegar a lograr grandes beneficios, impactando a la sociedad, evitando que ocurran pérdidas tanto de materiales como humanas.

## Bibliografía:

Celestino Hernandez (9 de diciembre de 2018), Sistema de alerta temprana para la reducción de riesgo de inundación, utilizando tecnología arduino y comunicación con redes de datos para el área de la cuenca media del Río Lempa. Universidad de Oriente.

Esp. Ing. José María Ducloux (16 de diciembre de 2016), SISTEMA DE MONITOREO DE LA CALIDAD DEL AIRE INTEGRADO A IoT. INSTITUTO UNIVERSITARIO AERONÁUTICO DEPARTAMENTO DE ELECTRÓNICA Y TELECOMUNICACIONES

Jose Rigoberto Osegueda Miranda (Marzo de 2016), Sistema de alerta temprana por riesgo de inundaciones debidas a desbordamiento de los niveles de ríos. UNIVERSIDAD DE EL SALVADOR FACULTAD DE INGENIERÍA Y ARQUITECTURA ESCUELA DE INGENIERÍA ELÉCTRICA

Adrianzen, B., Rodolfo Augusto, Puma Carrasco, Giancarlo (Junio de 2019). Diseño y desarrollo de un sistema portable de proximidad para invidentes que permite el uso de ambas manos. Universidad Tecnológica de Perú.

B., Alejandro, C., Rodrigo, L., Leandro (27 de febrero de 2019). TrackPoint, sistema telemático para monitoreo del movimiento de vehículos. Facultad Regional Avellaneda. Universidad Tecnológica Nacional

G., Oscar Alejandro, M., Miguel Angle, R. Acosta, Francisco Javier (17 de Octubre de 2016). Prototipo de un Sensor de nivel láser para tanques atmosféricos. Instituto Politécnico Nacional de México.

M., Mariana, G.G., Jazmin (28 de Mayo de 2014). Medidor de Distancia por ultrasonido. Instituto Politécnico Nacional de México

Arduino (2022). Arduino. <https://www.arduino.cc/>

O. Garcia, Carlos (Febrero de 2003). Desempeño de los Radios canales de una red EGPRS para Servicios de voz y datos. Instituto Nacional de Tijuana

R. Casillas, Reyes (Diciembre de 2008). Sistema de Telemetría utilizando las redes GPS, GPRS e Internet. Instituto Politécnico de Tijuana

C. Yarleque, Jesus Jhonatan (23 de noviembre de 2018). SISTEMA DE TELEMETRÍA CON TECNOLOGÍA GSM/GPRS PARA CONTROL DE POSICIÓN EN ATENAS DE MICROONDAS. Universidad Nacional de Piura.

P., Ariel (Noviembre de 2018). Internet de las Cosas. Universidad de San Andrés.

A.,Zabaleta.(Febrero de 2020).DISEÑO DE UNA SOLUCIÓN BASADA EN EL INTERNET DE LAS COSAS (IoT) EMPLEANDO LARAWAN PARA EL MONITOREO DE CULTIVOS AGRÍCOLAS EN PERÚ.

M.Tigse, Jonathan Eduardo(Septiembre de 2022).SISTEMA DOMÓTICO BASADO EN IOT PARA EL CENTRO INTEGRAL DEL ADULTO MAYOR DE LA CIUDAD DE AMBATO.Universidad Tecnica de Ambato.

G.Romero,Gabriel(Septiembre de 2019).Diseño e implementación de sensores y lectores IoT para la caracterización dieléctrica y la detección de espesores.Universidad Carlos III de Madrid.

H.Guacho,W.Patricio(Septiembre de 2015).DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ADQUISICIÓN DE SEÑALES BIOMÉTRICAS MEDIANTE MENSAJES SMS.Escuela Superior Politécnica de Chimborazo.

J.Hernandez,Victor Hugo,A.,Eduardo Jimenez,S.,Gabriel Medina,G.,Paolo Montero(Noviembre de 2009).Control de Dispositivo Electrónico mediante protocolo Modbus RS-485 y SMS.Instituto Politécnico Nacional unidad Culhuacan.

Siscom (27 de julio de 2015).Datasheet Sim800L.Siscom.

ETC2 (5 de diciembre de 2019).Datasheet de Hc.sr04.Etc2

**Anexo:****Codigo Completo:**

```

#include <LiquidCrystal_I2C.h> //Libreria del Display LCD//
#include <ThingSpeak.h> //Libreria para la Comunicacion con
ThingSpeak//
#include <SoftwareSerial.h> //Libreria que permite la utilizacion
de otros pines digitales para comunicacion serial
#include <String.h> //Conjunto de funciones para manipular
cadenas
#include <HCSR04.h> //Libreria del Sensor
#include <Wire.h> //Esta Libreria le permite comunicarse
con dispositivos I2C/TWI

//Se define el pin 2 para RX y 3 para TX
#define Rx 2
#define Tx 3

SoftwareSerial sim8001(Rx, Tx);
LiquidCrystal_I2C lcd(0x27, 16, 2);

/*const String telefono="+543442540376";Constante que se le asocia el
numero de telefono a*/
unsigned long tiempoInicio = millis();
const int Trigger = 7; // Trigger
const int echo = 8; // Echo
UltrasonicDistanceSensor distanceSensor(Trigger, echo);
float distancia = distanceSensor.measureDistanceCm();
int ultimaDistancia = 0;
int distanciaMaxima= 400;

void setup() {
  //pantalla
  lcd.init(); //iniciamos pantalla
  lcd.backlight();

```

```
    Serial.begin(9600);
    sim800l.begin(9600);
}

void loop() {

    Funcion_1();
    alerta();
}

void alerta() {
verificarSMS();
    if (distancia < distanciaMaxima) {
        Serial.println("Enviando SMS...");
        sim800l.print("AT+CMGF=1\r");
        delay(1000);
        sim800l.println("AT+CMGS=\"+543442545353\"");
        delay(1000);
        sim800l.println("Alerta, El Nivel es critico");
        delay(1000);
        sim800l.println((char)26);
        delay(100);
        sim800l.println();
        delay(5000);
        Serial.println("SMS enviado");
    } else {
        sim800l.println("No hay ningun peligro");
    }
}

void Funcion_1() {
    if (millis() - tiempoInicio >= 15000) {
        EnviodeDatos();
        Distancia();
        Signal();
        tiempo Inicio = millis();
    }
}

void Signal() {
    lcd.setCursor(0, 1);
    lcd.print("Señal RX: ");

    sim800l.println("AT+CSQ");
```

```

delay(1000);

if (sim8001.available()) {
    // Leer la respuesta completa del módulo SIM800L
    String response = sim8001.readStringUntil('\n');

    // Buscar la cadena "+CSQ: " en la respuesta
    int csqIndex = response.indexOf("+CSQ: ");
    if (csqIndex != -1) {
        // Obtener la calidad de la señal como una subcadena después de
        "+CSQ: "
        String rssiString = response.substring(csqIndex + 6);

        // Convertir la calidad de la señal a un número entero
        int rssi = rssiString.toInt();

        // Calcular el porcentaje de señal
        int reception = map(rssi, 0, 31, 0, 100);

        // Mostrar la recepción en el LCD
        lcd.print(reception);
        lcd.print("%");
    } else {
        lcd.print("No se pudo obtener la CSQ");
    }
}

void verificarSMS() {
    // Verificar si hay mensajes SMS disponibles
    sim8001.println("AT+CMGL=\"REC UNREAD\",1");
    delay(1000);

    while (sim8001.available()) {
        String response = sim8001.readStringUntil('\n');
        if (response.indexOf("AlertaDistancia") != -1) {
            // Si el mensaje contiene "AlertaDistancia", procesar el nuevo
            valor
            int index = response.indexOf(":") + 1;
            if (index > 0) {
                String nuevoValor = response.substring(index);
                distanciaMaxima = nuevoValor.toInt();
                Serial.print("Nueva distancia máxima: ");
                Serial.println(distanciaMaxima);
            }
        }
    }
}

```

```

        enviarSMS("Distancia máxima actualizada a " + nuevoValor);
    }
}
}

void enviarSMS(String mensaje) {
    Serial.println("Enviando SMS...");

    // Establecer el número de teléfono al que se enviará el SMS
    sim8001.println("AT+CMGS=\"+543442545353\"");
    delay(1000);

    // Enviar el contenido del mensaje SMS
    sim8001.print(mensaje);
    sim8001.write((char)26); // Ctrl+Z para finalizar el mensaje
    delay(100);

    // Esperar la respuesta del módulo SIM800L
    while (sim8001.available()) {
        Serial.write(sim8001.read());
    }

    Serial.println("SMS enviado");
}

void Distancia() {
    Serial.print("Distancia:");
    Serial.print(distancia);
    Serial.print("cm");
    Serial.println();

    if (distancia != ultimaDistancia) {
        lcd.clear();
        lcd.setCursor(0, 0);

        lcd.print("Distancia:"); // Prints string "Distance" on the LCD
        lcd.print(distancia);
        lcd.print("cm");
        ultimaDistancia = distancia;
    }
}

```

```

}

void EnviodeDatos() {

/*****
*****/
  if (sim8001.available())

    Serial.write(sim8001.read());

  sim8001.println("AT"); //Verificar el estado del modulo
  delay(1000);

  sim8001.println("AT+CPIN?"); //Verificamos el estado de conexion de
la Sim a la red
  delay(1000);

  sim8001.println("AT+CREG?"); //Verificar el estado del servicio y
mediante cual tecnologia de acceso a la celda esta conectado
  delay(1000);

  sim8001.println("AT+CGATT?"); //Verifica el estado del servicio de
dominio esta conectado.
  delay(1000);

  sim8001.println("AT+CIPSHUT"); //Abrir el contesto pdp GPRS
(Protocolo )
  delay(2000);

  sim8001.println("AT+CIPSTATUS"); //consulta el estado de la conexion
  delay(2000);

  sim8001.println("AT+CIPMUX=0"); //Comando para configurar en el
dispositivo una IP unica o multiples, como se seleccion 0, sera una IP
estatica.
  delay(3000);
  MostrarDatosSeriales();

  sim8001.println("AT+CSTT=\"datos.personal.com\""); //Configuracion
del nombre del punto de acceso, con contraseña y nombre de usuaria, en
este caso no tiene contraseña

```

```

delay(1000);
MostrarDatosSeriales();

sim8001.println("AT+CIICR"); //iniciar Conexión inalámbrica por GPRS
delay(3000);
MostrarDatosSeriales();

sim8001.println("AT+CIFSR"); //obtenemos IP local
delay(2000);
MostrarDatosSeriales();

sim8001.println("AT+CIPSPRT=0");
delay(3000);
MostrarDatosSeriales();

sim8001.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"
80\""); //indicaremos el protocolo de transporte TCP y puerto
delay(6000);
MostrarDatosSeriales();

sim8001.println("AT+CIPSEND"); //enviar datos mediante TCP
delay(4000);
MostrarDatosSeriales();

String datos = "GET
https://api.thingspeak.com/update?api_key=0WZ3T8A8ZPUF0SZ3&field1=0" +
String(distancia); // String creado para almacenar y concatenar los
datos relacionados al url y el danto de sensado
Serial.println(datos);
sim8001.println(datos);

delay(4000);
MostrarDatosSeriales();

sim8001.println((char)26); //codigo ASCII de Ctrl + Z
delay(5000);

sim8001.println("AT+CIPSHUT"); //Cerrar contexto GPRS
delay(5000);
MostrarDatosSeriales();
}

```

```
void MostrarDatosSeriales() {  
    while (sim8001.available() != 0)  
        Serial.write(sim8001.read());  
    delay(5000);  
}
```