

# **ESCUELA POLITÉCNICA NACIONAL**

**ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

**IMPLEMENTACIÓN DE PROTOTIPOS PARA LA TRANSMISIÓN  
INALÁMBRICA BASADOS EN TECNOLOGÍA LORA**

**IMPLEMENTACIÓN DE UN PROTOTIPO PARA LA TRANSMISIÓN  
INALÁMBRICA DE ALTITUD Y PRESIÓN DEL AIRE BASADO EN  
TECNOLOGÍA LORA**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR  
EN REDES Y TELECOMUNICACIONES**

**ALEX STEVEN ORBE ENCARNACIÓN**

**DIRECTOR: ANDRÉS FERNANDO REYES CASTRO**

**DMQ, marzo 2023**

## CERTIFICACIONES

Yo, ALEX STEVEN ORBE ENCARNACIÓN declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



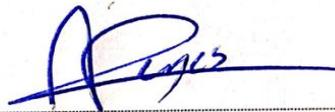
---

**ALEX STEVEN ORBE ENCARNACIÓN**

**alex.orbe@epn.edu.ec**

**lخورbe@gmail.com**

Certifico que el presente trabajo de integración curricular fue desarrollado por ALEX STEVEN ORBE ENCARNACIÓN, bajo mi supervisión.



**ANDRES FERNANDO REYES CASTRO**

**andres.reyes@epn.edu.ec**

## DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

A handwritten signature in blue ink, appearing to read 'Alex Steven Orbe Encarnación', written in a cursive style.

ALEX STEVEN ORBE ENCARNACIÓN

## DEDICATORIA

Dedico este Trabajo a la Escuela Politécnica Nacional por abrirme sus puertas para poder formarme como el profesional que soy ahora.

Quiero dedicar este Trabajo de integración Curricular, mi vida, mis logros a Dios ya que me ha dado fuerzas, valentía, sabiduría y sobre todo valor para no rendirme y quedarme en el camino, ya que sin él no estuviese donde estoy ahora. El ser que me acompaño en todos los años de mi vida, me cuida, y me guía por el camino del bien, infinitamente GRACIAS DIOS.

Dedico este Trabajo a mis Padres por tener la meta en su vida de darnos la educación a mi hermana y a mí. Al no darse por vencidos hasta que consigamos ser unos profesionales de bien. LOS AMO PADRES.

A mis Abuelitos que ya no se encuentran presencialmente, pero si espiritualmente y siempre quisieron verme como un profesional, después de tanto tiempo si en algún lugar me escuchan les quiero decir, ¡Lo logre Abuelitos! He cumplido mi promesa, los extraño un abrazo al cielo Abuelita María Orbe, Clemente Encarnación, Martha Cuenca.

A mi Abuelito Daniel que lo quiero con todo mi corazón, y seré muy feliz el día que venga a mi graduación. Lo quiero Abuelito Daniel.

ALEX STEVEN ORBE ENCARNACIÓN

## AGRADECIMIENTO

Agradezco a mi Papi Luis Alberto Orbe por trabajar tan duro, arriesgando su vida para que no nos falte nada y hacer lo posible para que tengamos la educación y hoy en día un título profesional, le agradecer toda mi vida por ser un ejemplo a seguir tener constancia y lucha para no rendirse. Lo Amo mucho Papi.

Agradezco a mi Mami Elsa Encarnación por su esfuerzo y buena voluntad que hace día a día para preparar los alimentos y no dejar que vaya con el estómago vacío. Por apoyarme incontable número de veces, por quererme y apoyarme, ¡Te amo mucho Mami!

A mi hermana Jael Orbe, aunque no seamos los mejores hermanos gracias por apoyarme cuando lo he necesitado. A mis perritos por siempre sacarme una sonrisa y alegrarme la vida los quiero mucho Connie, Juñor, Chicki, Suko pero en especial a mi perrito Cocki que ha sido mi mejor amigo, ya que ha compartido conmigo más de 12 años. ¡Te quiero mucho Cocki!

Agradezco a mi amiga Belén Calderón por acompañarme en las noches de desvelo y siempre darme ánimos en todo momento, por creer en mí y con sus palabras que nunca me rinda en mi vida y en este gran paso llamado universidad.

Agradezco a David Guachamin por compartir varias aventuras conmigo, por brindarme su conocimiento y ayudarme en cualquier situación presentada, eres como un hermano, te quiero mucho brosin, Gracias.

A Josselyn Zambrano que ha sido la persona que me alegra con solo saber de ella, que me a apoyado en todo momento, en donde quiera que estes siempre te agradeceré por todo lo que compartimos.

Agradezco a mi Tutor de Tesis Andrés Reyes por su paciencia y colaboración para que el presente Trabajo de Integración Curricular sea un éxito.

Agradezco a Paola Tierra por llegar a mi vida y alegrarla con su actitud positiva y amable. Aprecio sus palabras de ánimo y apoyo, que me han ayudado a seguir adelante en momentos difíciles. Además, valoro el tiempo que ha compartido conmigo para despejar mi mente y disfrutar de su compañía. Paola es una persona especial y estoy agradecido por tenerla en mi vida.

ALEX STEVEN ORBE ENCARNACIÓN

# ÍNDICE DE CONTENIDOS

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA .....	II
DEDICATORIA .....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDOS .....	V
RESUMEN.....	VII
ABSTRACT.....	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general .....	2
1.2 Objetivos específicos .....	2
1.3 Alcance.....	2
1.4 Marco Teórico.....	3
IoT ( <i>Internet of Things</i> ).....	3
Tecnología LoRa .....	3
Ambientes Barométricos .....	4
Sensor BMP 180 .....	4
WiFi LoRa 32 (V2) .....	5
2 METODOLOGÍA.....	5
3 RESULTADOS .....	7
3.1 Establecer los requerimientos del prototipo .....	7
3.2 Definición de los componentes de software y hardware.....	8
3.3 Diseño del prototipo .....	11
Circuito Electrónico.....	11
Diseño 3D .....	12
Diagrama de flujo .....	13
3.4 Implementación del prototipo.....	14
Fabricación de placa electrónica.....	14

Programación del módulo transmisor .....	19
Programación del módulo receptor .....	21
3.5 Pruebas de funcionamiento .....	23
Primera prueba de funcionamiento .....	23
Segunda Prueba de funcionamiento – sin línea de vista .....	25
Prueba de funcionamiento en el punto inicial 0 (m) .....	25
Prueba de funcionamiento a 100 (m) - sin línea de vista .....	26
Prueba de funcionamiento a 200 (m) - sin línea de vista .....	27
Prueba de funcionamiento a 300 (m) - sin línea de vista .....	29
Prueba de funcionamiento a 400 (m) - sin línea de vista .....	30
Tercera prueba de funcionamiento .....	31
Cuarta prueba de funcionamiento - con línea de vista .....	33
Prueba de funcionamiento en el punto inicial 0 (m) .....	33
Prueba de funcionamiento a 100 (m) con línea de vista .....	34
Prueba de funcionamiento a 200 (m) - con línea de vista .....	36
Prueba de funcionamiento a 300 (m) - con línea de vista .....	37
Prueba de funcionamiento a 400 (m) - con línea de vista .....	38
4 CONCLUSIONES.....	41
5 recomendaciones.....	42
6 REFERENCIAS BIBLIOGRÁFICAS .....	43
7 ANEXOS.....	45
ANEXO I: Certificado de Originalidad .....	i
ANEXO II: Enlaces .....	ii
ANEXO III: Diagrama de flujo del módulo transmisor y receptor .....	iii
Diagrama del Módulo Transmisor .....	iii
Diagrama del Módulo Receptor .....	iv
ANEXO IV: Códigos Fuente .....	v
Código Fuente del Módulo Transmisor .....	v
Código Fuente del Módulo Receptor .....	viii

## RESUMEN

El propósito de este proyecto es desarrollar un prototipo que emplee tecnología LoRa para enviar información inalámbrica de la altitud y la presión atmosférica. El prototipo recopila los valores barométricos y los muestra en una pantalla OLED tanto en el transmisor como en el receptor, con el objetivo de comparar la información recibida con la transmitida y detectar posibles errores.

Durante el desarrollo de este proyecto se llevaron a cabo diversas etapas, tales como el diseño, la fabricación y la implementación.

Se comenzó el proyecto con un análisis detallado de los requerimientos necesarios para garantizar la funcionalidad del prototipo.

La metodología que se ha establecido asegurará que este proyecto se lleve a cabo de forma organizada y fácil de entender.

Durante el desarrollo del proyecto se utilizó un sensor barométrico junto con un módulo LoRa que fueran compatibles con el software requerido. Además, se llevó a cabo la fabricación de una placa electrónica y un módulo protector para los dispositivos electrónicos.

Se llevarán a cabo pruebas de funcionamiento del sistema en dos escenarios diferentes: el primero sin línea de vista, a una distancia máxima de 400 metros, y el segundo con línea de vista, también a 400 metros del punto de inicio. Durante estas pruebas, se compararán las lecturas enviadas por el módulo transmisor y recibidas por el receptor para determinar el margen de error entre la información transmitida y recibida.

**PALABRAS CLAVE:** LoRa, sensor barométrico, PCB, OLED, IoT

## ABSTRACT

*The purpose of this project is to develop a prototype that uses LoRa technology to wirelessly transmit information about altitude and atmospheric pressure. The prototype collects barometric values and displays them on an OLED screen on both the transmitter and the receiver, with the aim of comparing the received information with the transmitted information and detecting any possible errors.*

*During the development of this project, various stages were carried out, such as design, manufacturing, and implementation. The project began with a detailed analysis of the requirements necessary to ensure the functionality of the prototype. The established methodology will ensure that this project is carried out in an organized and easy-to-understand manner.*

*During the project development, a barometric sensor was used along with a LoRa module that was compatible with the required software. In addition, the manufacturing of an electronic board and a protective module for the electronic devices was carried out.*

*Tests of the system's performance will be carried out in two different scenarios: the first one without line of sight at a maximum distance of 400 meters, and the second one with line of sight, also at 400 meters from the starting point. During these tests, the readings sent by the transmitter module and received by the receiver will be compared to determine the margin of error between the transmitted and received information.*

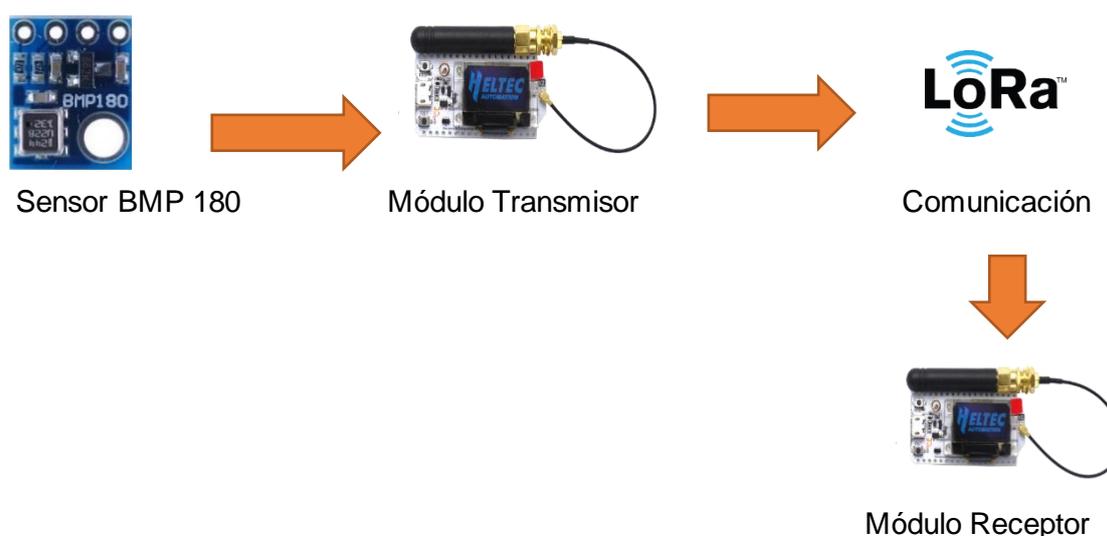
**KEYWORDS:** LoRa, barometric sensor, PCB, OLED, IoT.

# 1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Se implementó un prototipo para la medición de altitud y presión del aire, y transmisión inalámbrica mediante tecnología LoRa (*Long Range*). A través del uso del sensor adecuado es posible detectar los cambios de altura, presión y temperatura según la posición geográfica donde se encuentre dicho sensor. El prototipo cuenta con dos módulos: el primer dispositivo funciona como transmisor, el cual recibe la información que es enviada desde sensor. En el segundo dispositivo funciona como receptor que mediante comunicación LoRa recibirá y presentará los datos enviados por el transmisor.

El prototipo cuenta con la facilidad de ser portátil, por lo tanto, es posible realizar mediciones de altitud y presión del aire en cualquier lugar. Esta implementación puede ser de mucha utilidad para las personas que practican el andinismo, ya que con un módulo transmisor que se encuentre en un cierto punto elevado del volcán y con otro módulo receptor en las faldas, los guías van a saber la presión, la altitud e incluso la temperatura en ese punto, donde podrán tomar las medidas pertinentes para los andinistas.

El funcionamiento del prototipo se muestra en la Figura 1.1; el sensor va a captar los valores de la presión, altitud y temperatura, enviando al módulo transmisor en el cual se va a reflejar la información, mediante la comunicación inalámbrica de LoRa se enviarán los datos al receptor los cuales deben ser reflejados y deben contrastar con la información transmitida.



**Figura 1.1** Diagrama de funcionamiento del prototipo LoRa

## 1.1 Objetivo general

Implementar un prototipo para transmisión inalámbrica de altitud y presión del aire basado en tecnología LoRa.

## 1.2 Objetivos específicos

- Establecer los requerimientos del prototipo
- Definir los componentes de hardware y software
- Diseñar el prototipo
- Implementar el prototipo
- Realizar pruebas de funcionamiento

## 1.3 Alcance

Se llevará a cabo un análisis para determinar todos los requisitos necesarios para garantizar el correcto funcionamiento del prototipo.

Se investigará a fondo los dispositivos electrónicos que se utilizarán asegurando que sean compatibles en términos de *software* y *hardware*. El prototipo constará de los siguientes componentes principales: un sensor barométrico, un módulo LoRa maestro y un módulo LoRa esclavo. La investigación de los dispositivos también incluirá la consideración de los precios y comparación de las características principales para la selección de los componentes.

El prototipo se desarrollará con una representación en 3D, con el objetivo de tener una primera visualización del prototipo final. Se incluirán los cálculos necesarios para justificar el funcionamiento del circuito, y se contará con un diagrama de flujo que ayudará a entender mejor el lenguaje de programación de alto nivel utilizado en los dispositivos, de manera que tanto el *software* como el *hardware* trabajen juntos como un sistema.

Se tiene que considerar la elaboración de dos placas electrónicas, una para el transmisor y otra para el receptor. Además, el prototipo tiene que contar con módulo protector para garantizar portabilidad y la seguridad de los elementos. Durante las pruebas se tomarán mediciones en varios lugares de la ciudad, y se buscará tener un error mínimo entre la transmisión y recepción de los datos mediante comunicación LoRa.

## 1.4 Marco Teórico

### IoT (*Internet of Things*)

IoT es un término que hace referencia al proceso de conectar elementos físicos cotidianos a la red de Internet. Esto abarca una amplia gama de objetos, desde electrodomésticos como bombillas, hasta dispositivos médicos para la atención de la salud, ropa y accesorios personales inteligentes, y sistemas de ciudades inteligentes. Dentro del IoT, los dispositivos físicos se pueden dividir en dos categorías: interruptores, que envían comandos a otros dispositivos, y sensores, que recopilan datos del entorno y los envían a otra ubicación [1].

### Tecnología LoRa

La tecnología LoRa utiliza frecuencias que permiten la comunicación a largas distancias, es decir en el orden de kilómetros, utiliza bandas de frecuencia menor a 1 (GHz). La tecnología LoRa utilizan distintas bandas de frecuencias no licenciadas con el fin de evitar interferencias con otras señales inalámbricas, como se muestran en la Tabla 1.1.

**Tabla 1.1** Bandas de frecuencia en las regiones del mundo [2].

Región	Banda
Estados Unidos y América	902 MHz – 928 MHz
Europa	863 MHz – 870 MHz
China	779 MHz – 787 MHz

Con respecto a los modos de comunicación de datos, LoRa puede utilizar los siguientes:

*Simplex*: es una comunicación unidireccional donde un dispositivo únicamente va a transmitir y los N dispositivos solo van a recibir la información.

*Half-dúplex*: todos los dispositivos van a transmitir y recibir información, pero no podrán hacerlo simultáneamente.

*Full dúplex*: los dispositivos van a transmitir y recibir información de manera simultánea.

Las grandes ventajas de utilizar la tecnología LoRa incluye su alcance, bajo consumo de energía, inmunidad a interferencias y bajo costo [2].

Alcance y consumo de energía: la tecnología LoRa tiene un gran alcance usando un consumo de energía menor a comparación de otras tecnologías de tipo radio. El uso de

antenas adecuadamente dimensionadas, se puede lograr distancias de varios kilómetros.

Inmunidad a interferencia: la tecnología LoRa trabaja en una banda de frecuencia menor a 1 [GHz], por lo tanto, opera banda de frecuencia distinta a la que comúnmente los dispositivos electrónicos trabajan que es a 2.4 y 5 (GHz). Por esta razón las interferencias son muy bajas.

LoRa no puede transmitir un gran flujo de datos como voz o imágenes, ya que cuenta con una velocidad de datos baja; para los chips que funcionan a 915 (MHz) su velocidad llega a ser de 37.5 [Kbps]. LoRa no puede garantizar la seguridad al transmitir la información por lo que el usuario debe garantizar la encriptación, caso contrario puede sufrir ataques [2].

### **Ambientes Atmosféricos**

La presión atmosférica se define como el peso que va a ejercer el aire en la atmosfera terrestre, es decir se ejerce una presión en un punto dado por la columna de aire desde el suelo en ese punto hasta la parte superior de la atmósfera. Toma las unidades de medición en pascales, hectopascales para que sea exacta [3].

La altura sobre el nivel del mar va a influir en la presión atmosférica, esto quiere decir que mientras se aumente la altitud en la atmósfera, menor será la presión que se ejerza en todas las direcciones. Se puede explicar con un ejemplo sencillo, si se compara dos distintos escenarios: Ciudad de Barranquilla tiene una altura de 18 msnm (metros sobre el nivel del mar) y ejerce una presión de 1009.14 (hPa). Segundo escenario: Cima del Monte Everest se encuentra a una altura de 8.848 msnm y ejerce una presión alrededor de 333.30 (hPa). [4].

### **Sensor BMP 180**

Es un sensor barométrico que juega un papel importante en el cálculo de la presión atmosférica, es fácil de usar y no requiere componentes externos adicionales para poder operar. El transductor se observa en la Figura 1.2; el sensor BMP 180 se va a comunicar mediante su interfaz I2C, es capaz de medir temperatura, presión y altura. Usa una entrada de voltaje de 3.3 (V) hasta 5 (V). Si se quiere obtener el resultado de la presión en las unidades de Pascales a Hectopascales o milibares al resultado se tiene que dividir entre 100 [5].

El sensor BMP 180 cuenta con 4 pines [6].

- VIN: Voltaje de entrada 3.3 (V) – 5 (V)

- GND: Pin destinado a conexión a tierra
- SCL: Produce la señal de reloj mediante la interfaz de comunicación I2C
- SDA: Pin para la señal de datos de la interfaz de comunicación I2C



**Figura 1.2** Sensor BMP 180 [5].

### **WiFi LoRa 32 (V2)**

Es una placa de desarrollo de aplicaciones IoT diseñada por Heltec Automatic y consiste en un microprocesador ESP32 y un chip LoRa SX1276/SX1278. Cuenta con funciones de WiFi, BLE (*Bluetooth Low Energy*), LoRa y un sistema que admite baterías de tipo LI-PO (Litio y Polímero). Para visualizar la información la placa cuenta adicionalmente con una pantalla oled de 2.44 (cm) y permite el acceso al entorno de Arduino para su programación, la placa se muestra en la Figura 1.3 [7].



**Figura 1.3** WiFi LoRa 32 V2 [7].

## **2 METODOLOGÍA**

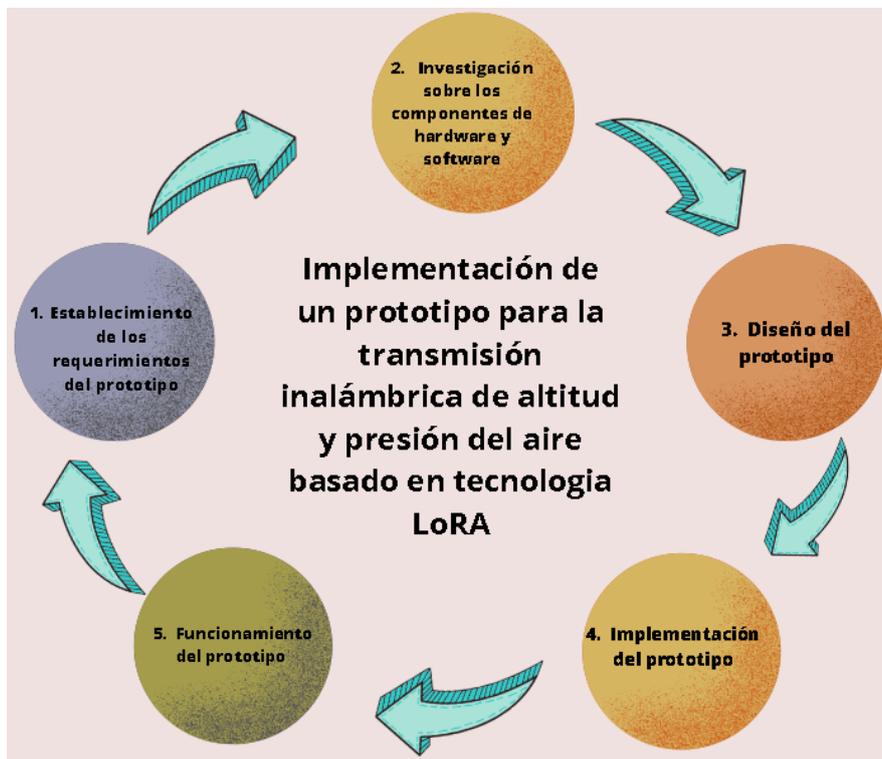
En primer lugar, se estableció los requerimientos del prototipo para el desarrollo e implementación del mismo. A través de la investigación bibliográfica se determinó la frecuencia de funcionamiento y se seleccionaron los componentes de *hardware* necesarios para que el prototipo tenga una alimentación adecuada, sea portátil y las consideraciones necesarias para tomar las lecturas del sensor.

Después de identificar los componentes del *hardware* para el prototipo, se realizó un estudio sobre el módulo LoRa comparando diferentes modelos, versiones y así determinar el más adecuado para el prototipo. De manera similar se seleccionaron los otros componentes considerando los criterios de compatibilidad, facilidad de uso y costo.

Para la parte del *software*, se compararon diferentes programas que fueran compatibles con el módulo LoRa seleccionados y que cumplan con los requerimientos previamente analizados.

Se realizó una simulación y un modelo 3D del prototipo, utilizando la misma herramienta se creó la PCB mediante un esquemático. Se buscaron los elementos dentro del programa para la creación de las pistas y se ubicaron de manera que sea estéticamente agradable. La implementación del prototipo se realizó en una baquelita de cobre, para la instalación de los componentes en la placa se taladro y se soldaron los elementos. El código de programación de alto nivel tiene que estar presente tanto en el transmisor como en el receptor y a través de un diagrama de flujo se puede comprender de manera más clara como está estructurado el código. Además, para proteger el prototipo se usó un acrílico para así darle estabilidad y seguridad.

En la fase final, se llevó a cabo las pruebas del prototipo en diferentes lugares de la ciudad para verificar su correcto funcionamiento. La altura será verificada mediante la georreferenciación de los puntos de prueba en el *software Google Earth* y la presión atmosférica se va a comprobar mediante la aplicación Estación Meteorológica, asegurándose que las mediciones sean precisas y con un error mínimo, teniendo en cuenta la distancia y los obstáculos que puedan afectar a los datos enviados del transmisor al receptor. La metodología usada se muestra en la Figura 2.1.



**Figura 2.1** Metodología empleada

### 3 RESULTADOS

En esta sección se proporciona los detalles para identificar los requerimientos, elegir el *hardware* y *software* adecuados, diseño e implementación del prototipo y las pruebas necesarias para validar las lecturas de altitud y presión.

#### 3.1 Establecer los requerimientos del prototipo

A partir de lo establecido en el alcance del presente Trabajo de Integración Curricular se establecieron los siguientes requerimientos:

- Comunicación Inalámbrica LoRa: La tecnología LoRa es una comunicación inalámbrica útil para las aplicaciones de IoT que necesitan una transmisión de datos a larga distancia, cuenta con distintas bandas de frecuencia en todo el mundo. A diferencia de otras regiones, en la región 2 se emplea la banda de frecuencia de 915 (MHz).
- Módulos portátiles: Para mejorar la movilidad y lograr una recolección de datos eficiente, es necesario contar con un dispositivo portátil. Además, una batería es esencial para garantizar que el prototipo pueda ser llevado a cualquier lugar sin depender de una fuente externa de energía.
- Visualización de resultados: Se necesita una pantalla LCD (*liquid-crystal display*) para facilitar la visualización de los valores obtenidos. Al elegir la pantalla, es importante considerar que se ajuste a la estética del prototipo y que la información se muestre de una manera clara y legible.
- Implementación de placas: La fabricación de placas electrónicas es un proceso importante para lograr una disposición estética de los componentes. La creación de pistas conductoras en la placa permite evitar la necesidad de cables entre los elementos y así lograr una apariencia más ordenada en el circuito del prototipo.
- Pruebas de funcionamiento: Realizar pruebas en diferentes entornos es importante para verificar el buen funcionamiento del prototipo. Al hacer pruebas en distintas condiciones, se pueden analizar los resultados y evaluar el rendimiento del prototipo. Es importante tener un margen de error mínimo entre lo que se transmite y recibe, tanto en ambientes sin obstáculos como en ambientes con obstáculos.
- Frecuencia de operación: La tecnología LoRa opera en la banda de frecuencia de 902 a 928 (MHz) con una potencia de transmisión de  $19\pm 1$  (dBm) [8]. Por lo tanto, la tecnología LoRa cumple con lo establecido el Plan Nacional de Frecuencias emitido por ARCOTEL (Agencia de Regulación y Control de las

Telecomunicaciones), donde se indica que utiliza la frecuencia central de 915 (MHz) para las tecnologías de banda libre [9].

- Mecanismo de alimentación: El módulo LoRa cuenta con dos mecanismos de alimentación. Una es a través de un puerto USB, que permite alimentar al módulo con un voltaje de 5 (V) con un máximo de 6 (V). Además, se puede cargar programas previamente programados en el *software* de Arduino IDE a través del puerto USB. Estos módulos son portátiles debido a su conector de batería de litio (SH.25 X 2 *Socket*), que puede soportar un rango de voltaje de 3.7 (V) hasta 4.2 (V) [8]. Para alimentar tanto el módulo LoRa como el sensor, se utilizó una batería de litio de 3.7 (V), lo que resulta suficiente para su correcto funcionamiento.

### 3.2 Definición de los componentes de software y hardware

Con respecto a la comunicación inalámbrica, existen varios tipos de módulos LoRa disponibles para su uso en comunicaciones inalámbricas e IoT. Estos módulos comparten características similares como la frecuencia de operación, el voltaje de alimentación, potencias de transmisión, etc. Para elegir el *hardware* adecuado, se llevó a cabo un análisis comparativo de los requisitos del prototipo, y se analizaron dos módulos LoRa: Heltec LoRa 32 v2 y SX1276. En la Tabla 3.1 se detalla las características y especificaciones de los dispositivos LoRa.

**Tabla 3.1** Comparación entre módulos LoRa [10], [11]

Especificaciones	Heltec LoRa 32 v2	SX1276
Voltaje de operación	3.7 (V)	3.7 (V)
Frecuencia	863 – 928 (MHz)	915 (MHz)
Puerto USB	Micro USB	-
Dimensiones	51 x 25.5 x 10.6 (mm)	36 x 14.3 x 5.3 (mm)
Potencia máxima de transmisión	19±1 (dBm)	+20 (dBm)
Rango de temperatura	-20 – 70 (°C)	-40 – 85 (°C)
Sensibilidad de recepción	-135 (dBm)	-123 (dBm)
Distancia máx. transmisión	15 (Km)	3 (Km)

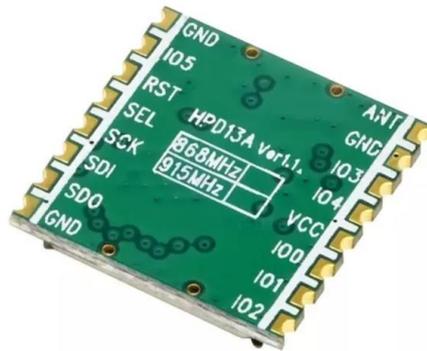
Heltec LoRa 32 v2 es un microcontrolador diseñado para aplicaciones IoT. Este dispositivo incluye una pantalla OLED para visualizar la información. Adicionalmente, tiene incorporado el chip LoRa SX1276 para trabajar a frecuencias de 868/915 (MHz). La placa cuenta con muchos pines que se pueden utilizar para diferentes propósitos en

distintas aplicaciones, es prácticamente una solución todo en uno por lo que va a reducir la cantidad de elementos para el prototipo. El módulo se muestra en la Figura 3.1 [12].



**Figura 3.1** Módulo Heltec LoRa 32 [10].

El módulo SX1276 es un dispositivo pequeño pero potente utilizado en proyectos de IoT. El chip LoRa le permite comunicarse a largas distancias y transmitir datos, puede ser conectado a un microcontrolador para facilitar la programación en aplicaciones que requieren una distancia considerable para poder transmitir información. Debido a su uso en una frecuencia baja y en banda libre, tiene una alta resistencia a las interferencias. [13]. El módulo SX1276 se muestra en la Figura 3.2.



**Figura 3.2** Módulo SX1276 [11].

Los dos módulos permiten la transmisión de datos utilizando la tecnología LoRa. Sin embargo, se seleccionó el módulo Heltec LoRa 32 v2 debido a que es una solución todo en uno, lo que la hace ideal para el prototipo, ya que éste busca ser portable, de bajas dimensiones, tener un rendimiento energético adecuado y ser compatible con el *software* de programación.

Por otra parte, para la selección del sensor, se tomó en cuenta dos sensores barométricos para su selección, considerando criterios como el consumo de energía, el tamaño y la precisión al tomar los datos atmosféricos. La información sobre las características de cada sensor es se presenta en la Tabla 3.2.

**Tabla 3.2** Comparación entre módulos LoRa [14] - [15]

Especificaciones	BMP 180	BMP 280
Voltaje	3.3 (V) – 5 (V)	3.3 (V)
Interfaz	I2C	I2C, SPI
Rango de presión	300 – 1100 (hPa)	300 – 1100 (hPa)
Precisión	±1	±1
Rango de altura	-500; 8000 (m)	-500; 8000 (m)
Precisión	±1	±1
Rango de temperatura	-40 – 85 (°C)	-40 – 85 (°C)

La Figura 3.3 muestra el sensor BMP 180, que es ideal para proyectos de IoT debido a su capacidad de medir datos atmosféricos como la presión, altitud y temperatura. Este sensor se comunica a través de la interfaz I2C y cuenta con un regulador de voltaje incorporado en su placa, lo que permite una alimentación máxima de 5 (V). Los resultados se obtienen a nivel del mar, por lo que los valores atmosféricos pueden variar en función de la presión actual en ese momento. [16].



**Figura 3.3** Sensor barométrico BMP180 [14]

La Figura 3.4 muestra el sensor BMP 280, que es muy similar al BMP 180. En cuanto a las características atmosféricas que mide, este sensor incluye una interfaz de conectividad SPI, lo que requiere dos pines adicionales para la conexión al bus. Sin embargo, a diferencia del BMP 180, no cuenta con un regulador de voltaje incorporado, por lo que solo se puede alimentar con 3.3 (V). El BMP 280 realiza conversiones A/D y proporciona los datos obtenidos a través de una interfaz digital. Este sensor tiene una gran precisión al medir datos atmosféricos en cortos períodos tiempo y con un bajo consumo de energía [17].



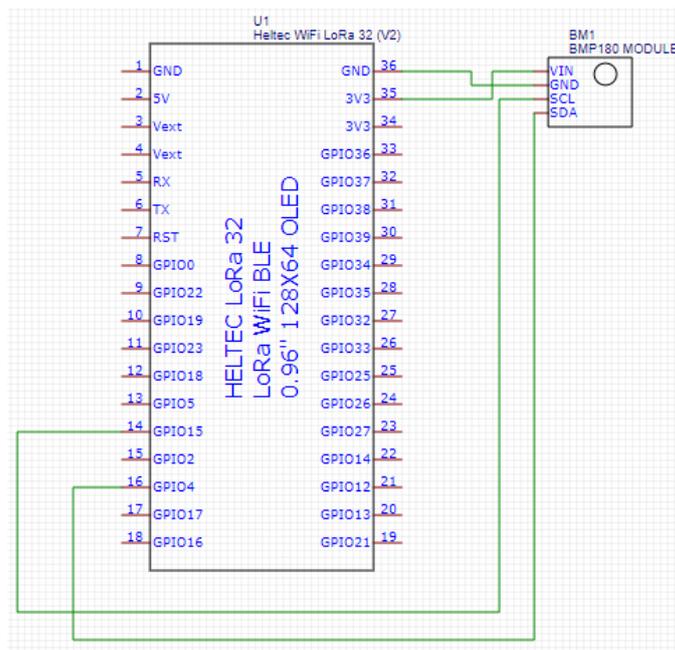
**Figura 3.4** Sensor BMP 280 [15]

Para el prototipo se optó por el sensor BMP 180 debido a sus bajas dimensiones, número de pines, comunicación del bus I2C, eficiencia energética adecuada, compatibilidad con el *software* de programación.

### 3.3 Diseño del prototipo

#### Circuito Electrónico

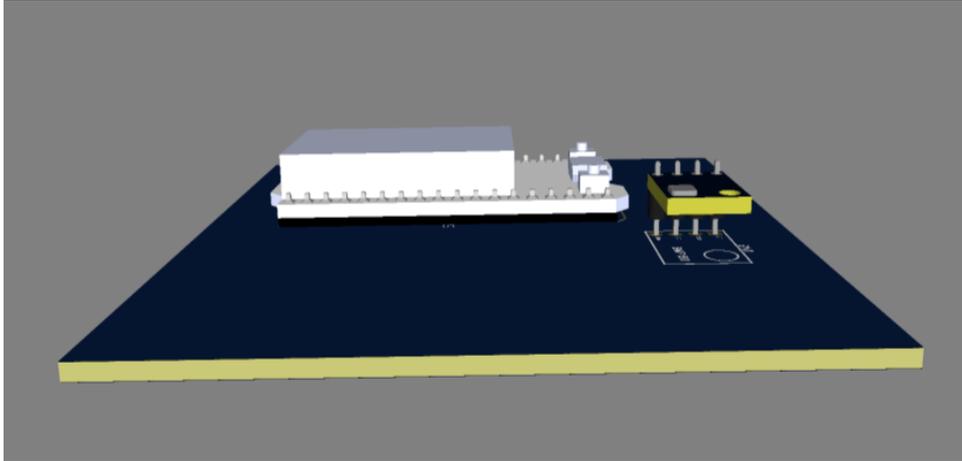
Se simuló el circuito electrónico en el *software EasyEda*, ya que dispone de un esquemático y diseño en 3D del prototipo a implementar. Esta es una herramienta que ayuda a la elaboración de circuitos, simulaciones, diseños de PCB online. Tiene una gran biblioteca de componentes electrónicos y varios ejemplos de esquemas en 3D ideales para tener una idea del circuito a crear [18]. En la Figura 3.5 se muestra el circuito del módulo transmisor.



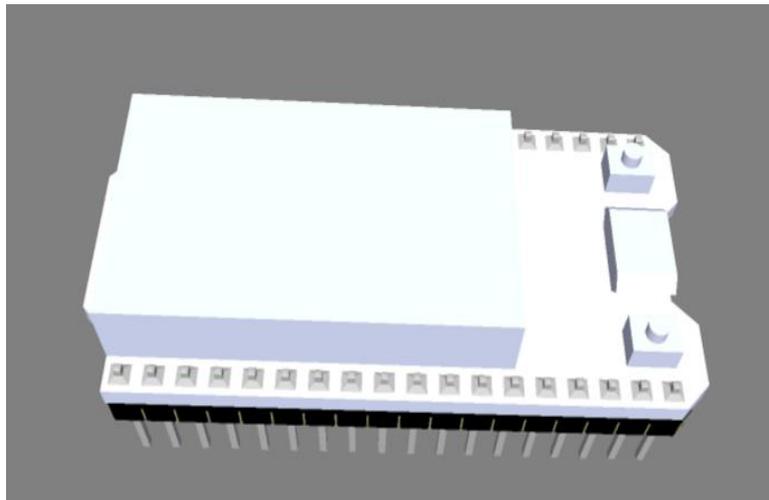
**Figura 3.5** Circuito del módulo transmisor

### Diseño 3D

*EasyEda* ofrece la opción de visualización en 3D, en sus librerías se pueden encontrar los componentes necesarios con medidas precisas que resultarán útiles para la creación de la PCB. En la Figura 3.6 se muestra el diseño en 3D del módulo transmisor y su placa correspondiente. En la Figura 3.7 se muestra el módulo receptor en 3D.

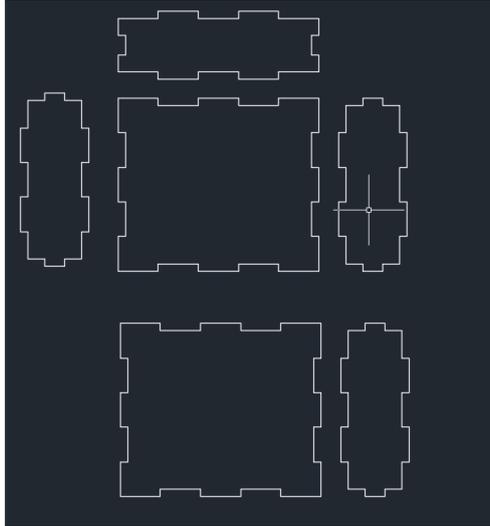


**Figura 3.6** Módulo transmisor en 3D



**Figura 3.7** Módulo receptor en 3D

Con el objetivo de proteger el prototipo y darle un aspecto estético, se utilizó el *software* AutoCAD para diseñar el módulo protector. AutoCAD es una herramienta ampliamente utilizada por Arquitectos, Ingenieros Civiles, debido a su funcionalidad de diseñar en 2D y 3D. Este *software* permitió la creación del plano del módulo protector que se muestra en la Figura 3.8.



**Figura 3.8** Plano del módulo protector para el transmisor.

### **Diagrama de flujo**

En el ANEXO III se muestra el diagrama de flujo del programa de altitud y presión mediante tecnología LoRa del módulo transmisor. En la lectura de datos se define los parámetros internos del programa como el nivel de RSSI, el tamaño del paquete, y el paquete.

Se pasa a la instancia del proceso donde se va a definir la banda de frecuencia a utilizar, se llama a la instancia de Adafruit para que pueda comunicarse con el sensor, después se configura la inicialización del módulo LoRa y la pantalla OLED.

Luego se muestra el logotipo de Heltec y si todo funciona correctamente manda un mensaje de “Inicio exitoso”. Se inicializa el sensor y si no detecta ningún problema pasa a obtener los valores de temperatura, presión y altitud.

Se procedió a configurar la potencia de transmisión, si está correcto entonces el módulo transmisor enviara los datos de temperatura, presión, altitud al receptor mediante comunicación LoRa, en caso de que no esté bien configurado regresa a la configuración de la potencia hasta que este correctamente.

Luego se va a crear una cadena de caracteres que va a contener los valores de temperatura, presión y altitud, los cuales se van a mostrar en el OLED.

En el mismo ANEXO III también se muestra el diagrama de flujo de módulo receptor. donde en la primera parte va a tomar la lectura de datos de los parámetros internos del programa como el RSSI, tamaño del paquete, y el paquete.

Se define la banda de frecuencia a la que va a trabajar, debe ser la misma que se usó en el módulo transmisor.

Luego recibe el paquete de datos mediante la función *drawStringMaxWidth* donde va a mostrar la cadena de texto de la variable *packet*.

Posteriormente pasa a un ciclo *for* donde va a leer cada *byte* del paquete, para que pueda seguir compilando el “i” tiene que ser mayor a 1 ya que si es 0 no está recibiendo paquetes.

Después se llama a la función *LoRa Data* la cual muestra la información recibida y muestra en el OLED. Se configura el módulo, la pantalla para que luego muestre el logotipo de Heltec y si todo funciona manda un mensaje de inicio exitoso.

Luego entra a un bucle de *if*, en que si va por el camino del “no” la recepción de datos no llega al módulo receptor y regresa al logotipo e inicio exitoso. En cambio, si va por el camino del sí va a recibir los valores de temperatura, presión, altitud. Posteriormente imprime los datos atmosféricos en el OLED.

### 3.4 Implementación del prototipo

#### Fabricación de placa electrónica

Se empleó el *software EasyEda* para llevar a cabo el diseño de la placa de circuito impreso. El simulador cuenta con una librería bastante amplia de elementos donde se pudo incorporar tanto el sensor BMP 180 como el módulo LoRa en la PCB, logrando que las dimensiones sean adecuadas para los dispositivos correspondientes. La Figura 3.9 muestra el diseño de la PCB que se implementara sobre una baquelita de cobre.

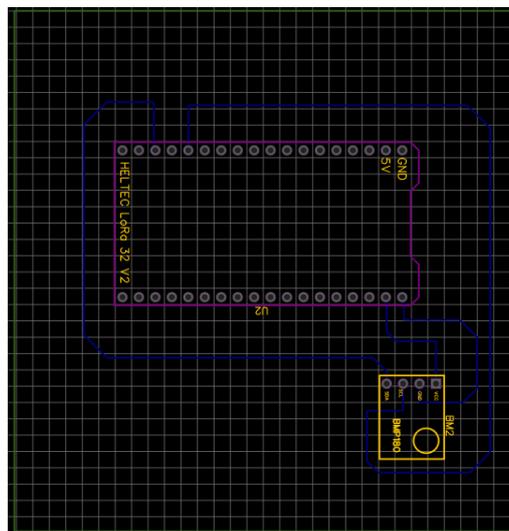
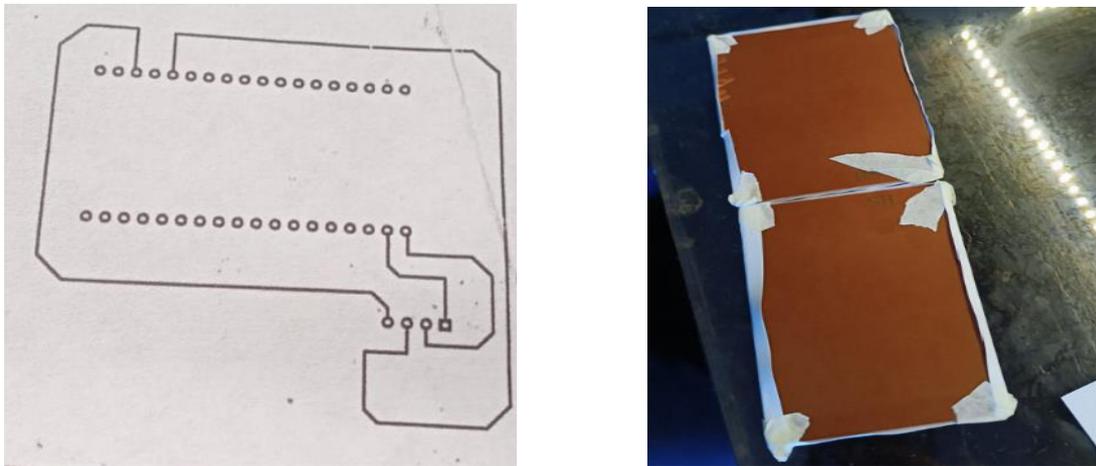


Figura 3.9 Diseño de la PCB en *EasyEda*

Una vez que se completó el diseño del circuito en la PCB, se utilizó una impresora láser para imprimir el diseño en una hoja de transferencia de calor. Posteriormente, se procedió a realizar la transferencia térmica, además, se realizó una limpieza y lijado de la baquelita para retirar el material deteriorado de la superficie. La Figura 3.10 ilustra tanto el circuito impreso como la baquelita junto con la hoja de transferencia de calor utilizada.

Para que el sensor funcione de manera adecuada, es necesario soldar sus pines correctamente, ya que, de lo contrario no tiene un buen contacto entre los dispositivos del prototipo. Esto aplica tanto para el sensor como para el módulo transmisor, ya que ambos son responsables de recopilar los datos atmosféricos.



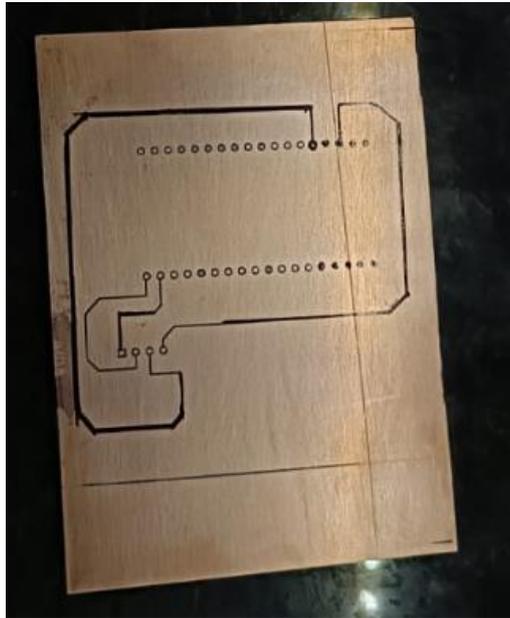
**Figura 3.10** Circuito impreso y baquelita con la hoja de transferencia de calor

En la Figura 3.11 se muestra el proceso de transferencia del diseño del circuito a la baquelita mediante la técnica del planchado. Se ajustó la plancha a la temperatura más alta y se aplicó una presión uniforme para lograr una buena adhesión del circuito a la baquelita. El tiempo estimado para llevar a cabo la técnica del planchado fue de alrededor de 60 segundos.



**Figura 3.11** Transferencia de calor mediante la técnica del planchado.

En la Figura 3.12 se muestra la baquelita con el circuito ya adherido a la misma mediante la técnica de transferencia de calor.



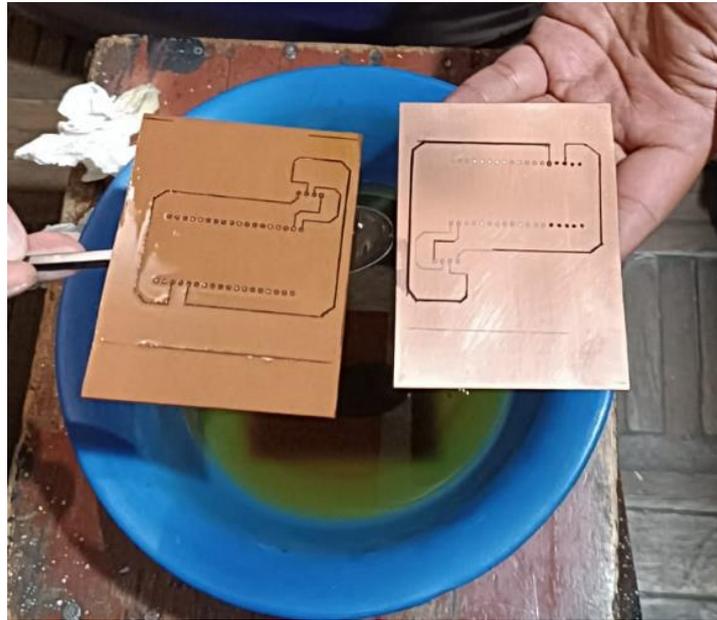
**Figura 3.12** Baquelita con el circuito adherido.

Una vez finalizada la transferencia de calor, se procedió a aplicar cloruro férrico con el fin de eliminar el cobre de la baquelita y dejar únicamente el circuito impreso. La placa se sumergió en un recipiente con agua para que el cloruro férrico pudiera reaccionar y corroer el cobre no deseado, de manera que queda un patrón conductor de cobre en la placa, que corresponde al circuito impreso. La Figura 3.13 se muestra el recipiente que contiene el cloruro férrico y la placa sumergida.



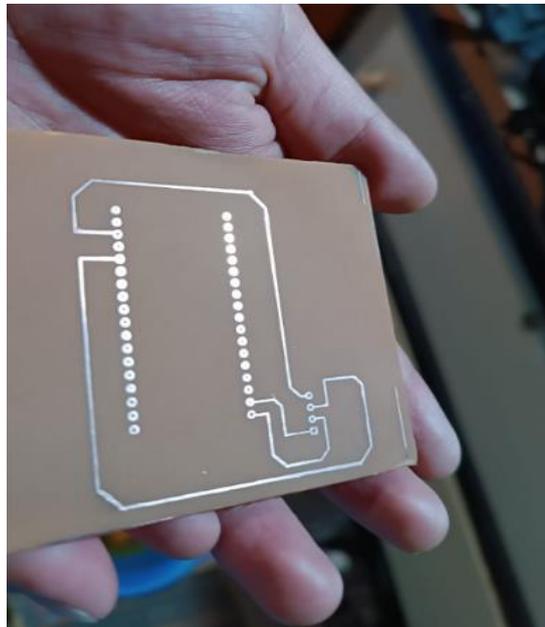
**Figura 3.13** Recipiente con la placa sumergida en el cloruro férrico

En la Figura 3.14 muestra dos placas: una que ha sido sometida al ácido y otra que todavía conserva el cobre.



**Figura 3.14** Comparación de placas.

Después de finalizar el proceso de grabado en el ácido férrico, se llevó a cabo una limpieza con acetona para eliminar cualquier residuo de tinta y otros materiales. La Figura 3.15 muestra la placa finalizada junto con las pistas de cobre.



**Figura 3.15** Placa finalizada con las pistas de cobre.

Una vez que la placa ha sido terminada, se procede a perforar las conexiones para que los dispositivos electrónicos puedan ser instalados. Para hacer esto, la placa se coloca sobre una superficie que sirva de soporte, se utiliza un taladro con una broca de 1 mm para realizar las perforaciones necesarias.

Después de haber quemado y perforado la placa, se lleva a cabo el proceso de soldadura, tal como se muestra en la Figura 3.16. Se usó estaño, pasta para soldar y un caudín para unir los pines de los componentes con las pistas de la placa, además, se revisó la calidad de las soldaduras y la continuidad eléctrica de las pistas.



**Figura 3.16** Proceso de soldadura.

Una vez completado el proceso de quemado, perforación y soldadura de la placa, se procede a insertar los componentes electrónicos en el espaldín tipo hembra y se realiza una comprobación para asegurar que la placa funcione correctamente, sin necesidad de utilizar cables adicionales. En la Figura 3.17 se muestra la placa terminada con los componentes ya funcionando.



**Figura 3.17** Placa terminada con los elementos electrónicos

## Programación del módulo transmisor

Se usó el *software* Arduino IDE donde en la parte inicial se establecieron los parámetros básicos como librerías, banda de frecuencia y parámetros internos del paquete que se pueden observar en la Figura 3.18.

```
#include "heltec.h"
#include "images.h"
#include <Adafruit_BMP085.h>

Adafruit_BMP085 bmp;
#define BAND    915E6

unsigned int counter = 0;
String rssi = "RSSI --";
String packSize = "--";
String packet ;
```

**Figura 3.18** Muestra la inclusión de librerías, la definición de la banda y los parámetros establecidos.

En la función *void setup* se inicializa el módulo Heltec. Se llama a la función *init* para la inicialización de la placa, se ajusta la orientación del texto y se elige el tipo de letra. Si todo funciona correctamente, se muestra un mensaje exitoso. Después, el programa entra en un bucle *if* en el cual se verifica la conexión del sensor. Si la conexión está mal, se muestra un mensaje de error. En cambio, si la conexión está bien, el programa continúa ejecutándose, como se muestra en la Figura 3.19.

```
void setup()
{
  Heltec.begin(true, true, true, true, BAND); // se inicializa el módulo

  Heltec.display->init(); // Instancia init para inicializar la placa
  Heltec.display->flipScreenVertically(); // Se ajusta la orientación de la pantalla
  Heltec.display->setFont(ArialMT_Plain_10); // Se configura el tipo de letra
  logo(); // Llama a la función logo
  delay(1500); // Tiempo de espera 1.5 (s)
  Heltec.display->clear(); // Se limpia el display

  Heltec.display->drawString(0, 0, "Inicio exitoso!"); // Mensaje exitoso
  Heltec.display->display(); // Se actualiza el display
  delay(1000); // tiempo de espera 1(s)
  if (!bmp.begin()) { // Bucle if y se inicializa el sensor
    Serial.println("No se pudo encontrar un sensor BMP085 válido, comprueba el cableado!");
    while (1) {}
  }
}
```

**Figura 3.19** Configuración *Void Setup*

En la función *void loop* se ejecuta constantemente el programa que lee los datos atmosféricos. Además, se inicializa el paquete LoRa y se obtienen valores de temperatura, presión y altitud. Para mayor claridad, se puede ver la parte del código que ejecuta esta función en la Figura 3.20.

```

void loop()
{
    // send packet
    LoRa.beginPacket();

    counter++;

    String packet ;

    Serial.print("Temperature = ");
    Serial.print(bmp.readTemperature());
    Serial.println(" *C");

    Serial.print("Pressure = ");
    Serial.print(bmp.readPressure());
    Serial.println(" Pa");

    Serial.print("Altitude = ");
    Serial.print(bmp.readAltitude());
    Serial.println(" meters");

    Serial.print("Pressure at sealevel (calculated) = ");
    Serial.print(bmp.readSealevelPressure());
    Serial.println(" Pa");
}

```

**Figura 3.20** Muestra el inicio de paquete de LoRa, aumento de contador, y toma de valores atmosféricos.

En la Figura 3.21 muestra la configuración de la potencia de transmisión a 14 (dBm), y el paquete que se va a enviar mediante la comunicación LoRa, cadena de caracteres que se va a mostrar en el OLED del transmisor.

```

LoRa.setTxPower(14,RF_PACONFIG_PASELECT_PABOOST); // Configuración de la potencia a 14 (dBm)
LoRa.print("Temp:"); // Paquete de LoRa que imprime Temp
LoRa.print(bmp.readTemperature()); // Paquete de LoRa que obtiene la temperatura
LoRa.println(" oC "); // Paquete de LoRa que imprime símbolo de grados centígrados
LoRa.print("Presion: "); // Imprime la palabra Presión
LoRa.print(bmp.readPressure()/100.F); // Paquete de LoRa que obtiene la presión dividido para 100 para obtener en Hpa
LoRa.println(" hPa "); // Paquete de LoRa que imprime símbolo de Hectopascuales
LoRa.print("Altitud: "); // Imprime la palabra Presión
LoRa.print(bmp.readAltitude(101500)); // Paquete de LoRa que obtiene la altitud multiplicado para 101500 para obtener la altura a lvl del mar
LoRa.println(" metros "); // Imprime la palabra metros
LoRa.endPacket(); // finaliza el paquete LoRa
packet="Temp:"+String(bmp.readTemperature())+" oC "+String(bmp.readPressure()/100.F)+" hPa "+"Altitud = "+String(bmp.readAltitude(101500))+ " metros "

Heltec.display->clear(); // Se limpia el display
Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT); // Se alinea el texto
Heltec.display->setFont(ArialMT_Plain_10); // Se ajusta el tamaño de letra

```

**Figura 3.21** Configuración de potencia de transmisión, paquete de datos a enviar, cadena de datos.

En la Figura 3.22 muestra el tamaño de que va a ocupar la variable “*packet*” y el encendido y apagado del led para verificar él envío de paquetes.

```
Heltec.display->clear();
Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
Heltec.display->setFont(ArialMT_Plain_10);

Heltec.display->drawStringMaxWidth(0 , 0 , 128, packet);

Heltec.displaySS->display();

digitalWrite(LED, HIGH); // turn the LED on (HIGH is the voltage level)
delay(100); // wait for a second
digitalWrite(LED, LOW); // turn the LED off by making the voltage LOW
delay(100);
}
```

**Figura 3.22** Tamaño de la variable “*packet*”, verificación de envío de datos mediante el led.

### Programación del módulo receptor

En la parte inicial se estableció los parámetros básicos como librerías, banda de frecuencia y parámetros internos del paquete que se pueden observar en la Figura 3.23.

```
#include "heltec.h"
#include "images.h"

#define BAND 915E6
String rssi = "RSSI --";
String packSize = "--";
String packet ;
```

**Figura 3.23** se muestra la inclusión de librerías, la definición de la banda y los parámetros establecidos.

La función *void LoRaData* es llamada para recibir paquetes de datos a través del módulo LoRa. La función *drawStringMaxWidth* ajusta el ancho específico de la cadena de la variable “*packet*” en la Figura 3.24 se muestra la función *void LoRaData*.

```
void LoRaData(){
  Heltec.display->clear();
  Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
  Heltec.display->setFont(ArialMT_Plain_10);
  //Heltec.display->drawString(0 , 0 , "Received "+ packSize + " bytes");
  Heltec.display->drawStringMaxWidth(0 , 0 , 128, packet);
  //Heltec.display->drawString(0, 0, rssi);
  Heltec.display->display();
}
```

**Figura 3.24** Configuración de la función *void LoRaData*.

La función *void cbk* funciona como *callback* que se va a ejecutar cuando recibe un paquete LoRa. Va a utilizar un ciclo *for* para el conteo de *bytes* del paquete, finalmente llama a la función *LoRa Data*. En la Figura 3.25 se muestra la función *void cbk*.

```
void cbk(int packetSize) {
    packet = "";
    packetSize = String(packetSize,DEC);
    for (int i = 0; i < packetSize; i++) { packet += (char) LoRa.read(); }
    rssi = "RSSI " + String(LoRa.packetRssi(), DEC) ;
    LoRaData();
}
```

**Figura 3.25** Función *cbk*.

En la función *void setup* se inicializa el módulo Heltec. Se llama a la función *init* para la inicialización de la placa, se ajusta la orientación del texto y se elige el tipo de letra. Si todo funciona correctamente, se muestra un mensaje exitoso, luego procede a llamar a la función *LoRa receive* en la Figura 3.26 se muestra la configuración de *void setup*.

```
void setup()
{
    Heltec.begin(true, true, true, true, BAND); // se inicializa el módulo

    Heltec.display->init(); // Instancia init para inicializar la placa
    Heltec.display->flipScreenVertically(); // Se ajusta la orientación de la pantalla
    Heltec.display->setFont(ArialMT_Plain_10); // Se configura el tipo de letra
    logo(); // Llama a la función logo
    delay(1500); // Tiempo de espera 1.5 (s)
    Heltec.display->clear(); // Se limpia el display

    Heltec.display->drawString(0, 0, "Inicio exitoso!"); // Mensaje exitoso
    Heltec.display->display(); // Se actualiza el display
    delay(1000); // tiempo de espera 1(s)
    LoRa.receive(); // llama a la función LoRa receive
}
```

**Figura 3.26** Función *void setup*.

Por último, se entra a la función *void loop* donde se va a manejar la función *parsepacket* esta se encarga de leer la variable *packsize*, luego se verifica si es el paquete es mayor a 0 llama a la función *cbk* el cual muestra los resultados en el OLED llamando a la función *LoRa Data*. En la Figura 3.27 muestra la función *void loop*.

```
void loop() {
    int packetSize = LoRa.parsePacket();
    if (packetSize) { cbk(packetSize); }
    delay(10);
}
```

**Figura 3.27** Función *void loop*.

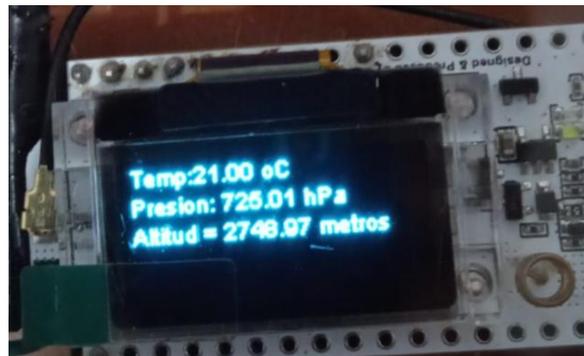
En el ANEXO IV se muestra el código completo del módulo transmisor y receptor.

### 3.5 Pruebas de funcionamiento

Para las pruebas de funcionamiento se va a comparar los datos del sensor con fuentes adicionales para verificar que las lecturas son correctas. Además, se tiene que considerar la distancia, presencia de obstáculos para poder obtener el error entre lo transmitido y recibido.

#### Primera prueba de funcionamiento

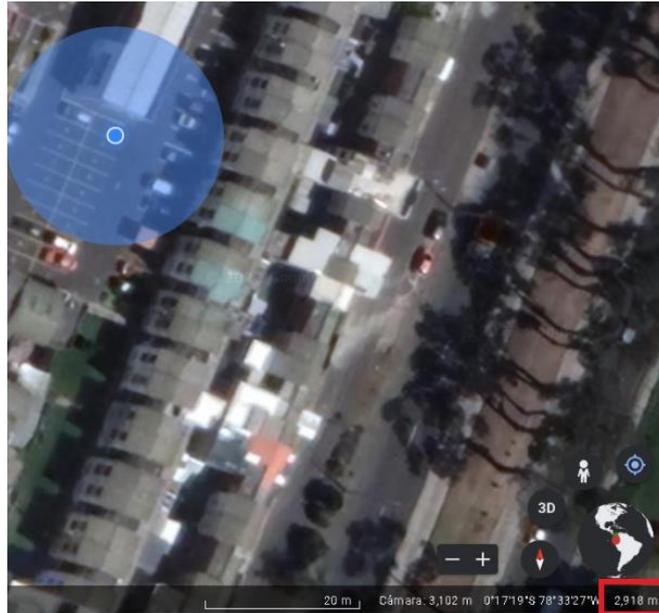
Se llevará a cabo una comparación entre los valores de presión registrados por el módulo transmisor y los valores proporcionados por la aplicación Estación Meteorológica. En cuanto a la altitud, se utilizará el *software Google Earth* para su respectiva comparación, los valores de la altura tomados por la aplicación se basan en los datos obtenidos por satélite, aviones y drones. En la Figura 3.28 se presentan los valores de presión y altitud medidos por el sensor barométrico en un punto de partida, mientras que en las Figuras 3.29 y 3.30 se muestran los valores de presión y altitud proporcionados por la aplicación Estación Meteorológica y el *software Google Earth*, respectivamente.



**Figura 3.28** Valores de presión y altitud medidos desde el módulo transmisor.



**Figura 3.29** Valor de la presión en HPa emitido por la aplicación Estación Meteorológica.



**Figura 3.30** Altura en metros obtenida en el software *Google Earth*.

Las coordenadas geográficas del punto inicial son: -0288704 de longitud y -78.557564 de latitud.

En la Tabla 3.3 se presentan los valores de las lecturas proporcionados por el módulo transmisor, así como por las fuentes adicionales. Por otro lado, la Ecuación 3.1 muestra el cálculo del error de la presión, mientras que la Ecuación 3.2 muestra el cálculo del error relacionado con la altura.

**Tabla 3.3** Valores de las lecturas del prototipo y de las fuentes.

Ambientes Barométricos	Fuentes utilizadas		
	Prototipo	Estación Meteorológica	Google Earth
Presión (hPa)	725.01	720	x
Altitud (m)	2748.97	x	2918

$$\frac{|725.01 - 720|}{720} \times 100 = 0.69 \%$$

**Ecuación 3.1** Cálculo de error de la presión.

$$\frac{|2748.97 - 2918|}{2918} \times 100 = 5.79 \%$$

**Ecuación 3.2** Cálculo de error de la altitud.

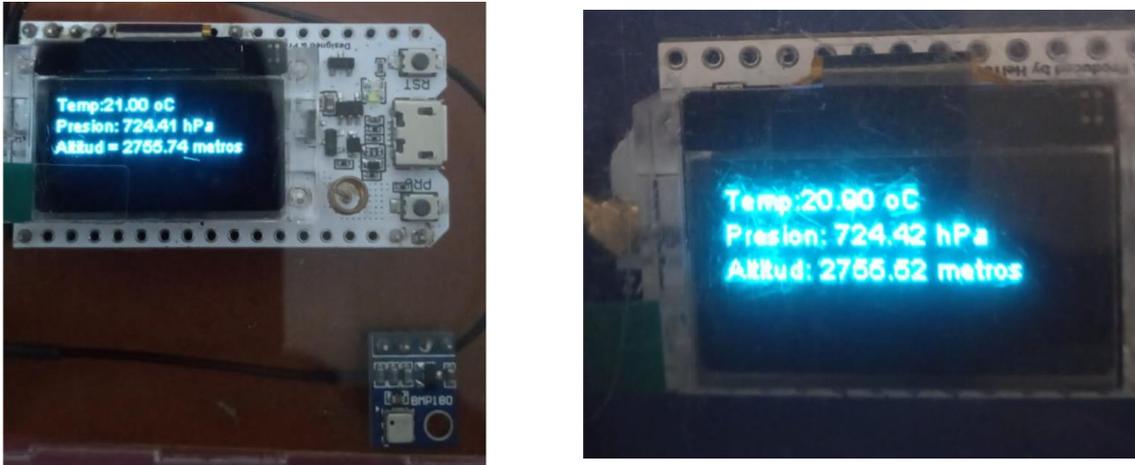
Es importante considerar el error en la presión de un 0.69% y el error en la altura de un 5.79% al interpretar los resultados de las mediciones. Estos errores pueden ser ocasionados por factores como la precisión del sensor, o la fuente, calibración y otros factores ambientales.

### Segunda Prueba de funcionamiento – sin línea de vista

Para la segunda prueba de funcionamiento se compararán las lecturas que envíe el transmisor con los datos recibidos por el receptor, tomando en cuenta la distancia de 100, 200, 300 y 400 metros, así como la presencia de obstáculos.

### Prueba de funcionamiento en el punto inicial 0 (m)

En la Figura 3.31 se muestra el módulo transmisor y el módulo receptor con los valores de presión y altitud en el punto inicial 0 (m).



**Figura 3.31** Módulo transmisor y receptor con valores de presión y altitud.

En la Tabla 3.4 se muestra los valores de latitud, longitud, presión, altitud del módulo transmisor y receptor en el punto inicial.

**Tabla 3.4** Medidas barométricas, de latitud y longitud.

	Latitud	Longitud	Presión (hPa)	Altitud (m)
Transmisor	-0.288704	-78.557564	724.41	2755.74
Receptor	-0.288704	-78.557564	724.42	2755.52

En la Ecuación 3.3 y 3.4 se muestra el cálculo de error de la presión y altitud respectivamente.

$$\frac{|724.42 - 724.41|}{724.41} \times 100 = 0.001\%$$

**Ecuación 3.3** Cálculo de error de la presión.

$$\frac{|2755.52 - 2755.74|}{2755.74} \times 100 = 0.007 \%$$

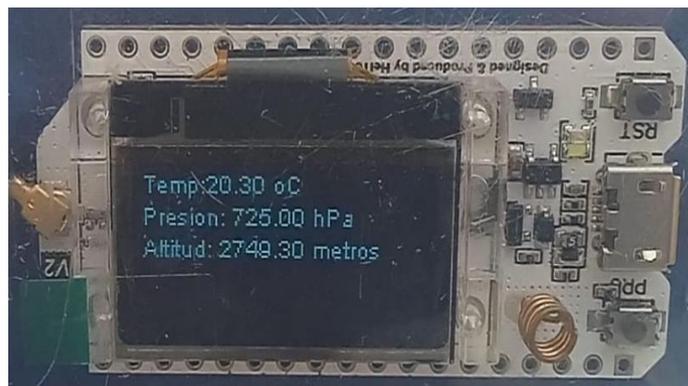
**Ecuación 3.4** Cálculo de error de la altitud.

### Prueba de funcionamiento a 100 (m) - sin línea de vista

En la Figura 3.32 se muestra el módulo transmisor con los valores de presión y altitud a 0 (m), mientras en la Figura 3.33 se muestra el módulo receptor con los valores de presión y altitud a una distancia de 100 (m).



**Figura 3.32** Módulo transmisor con los valores atmosféricos.



**Figura 3.33** Módulo receptor con los valores de presión y altitud.

En la Tabla 3.5 se muestra los valores de latitud, longitud, presión, altitud del módulo transmisor en el punto inicial y el módulo receptor a 100 (m).

**Tabla 3.5** Medidas barométricas a 100 (m).

	Distancia (m)	Latitud	Longitud	Presión (hPa)	Altitud (m)
Transmisor	0	-0.288704	-78.557564	724.92	2749.73
Receptor	100	-0.288559	-78.558250	725.00	2749.30

En la Ecuación 3.5 y 3.6 se muestra el cálculo de error de la presión y altitud respectivamente.

$$\frac{|724.92 - 725|}{725} \times 100 = 0.01\%$$

**Ecuación 3.5** Cálculo de error de la presión.

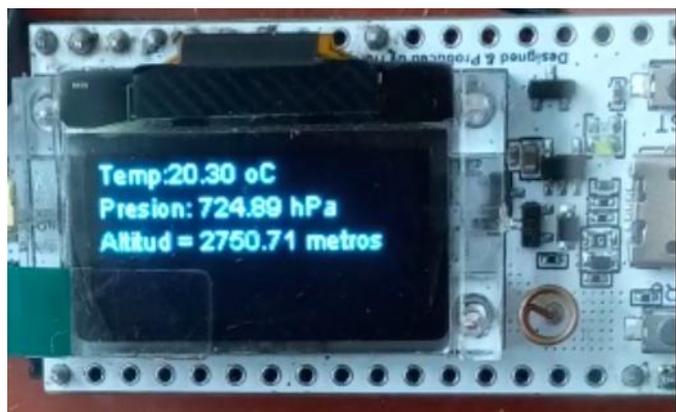
$$\frac{|2749.30 - 2749.73|}{2749.73} \times 100 = 0.01 \%$$

**Ecuación 3.6** Cálculo de error de la altitud.

El cálculo del error fue mínimo debido a la corta distancia y a la falta de obstáculos que pudieran interponerse en la señal transmitida.

### **Prueba de funcionamiento a 200 (m) - sin línea de vista**

En la Figura 3.34 se muestra el módulo transmisor con los valores de presión y altitud a 0 (m), mientras en la Figura 3.35 se muestra el módulo receptor con los valores de presión y altitud a una distancia de 200 (m).



**Figura 3.34** Módulo transmisor con los valores atmosféricos.



**Figura 3.35** Módulo receptor con los valores de presión y altitud.

En la Tabla 3.6 se muestra los valores de latitud, longitud, presión, altitud del módulo transmisor en el punto inicial y el módulo receptor a 200 (m).

**Tabla 3.6** Medidas barométricas a 200 (m).

	Distancia (m)	Latitud	Longitud	Presión (hPa)	Altitud (m)
Transmisor	0	-0.288704	-78.557564	724.89	2750.71
Receptor	200	-0.289224	-78.558713	724.94	2750.06

En la Ecuación 3.7 y 3.8 se muestra el cálculo de error de la presión y altitud respectivamente.

$$\frac{|724.94 - 724.89|}{724.89} \times 100 = 0.01\%$$

**Ecuación 3.7** Cálculo de error de la presión.

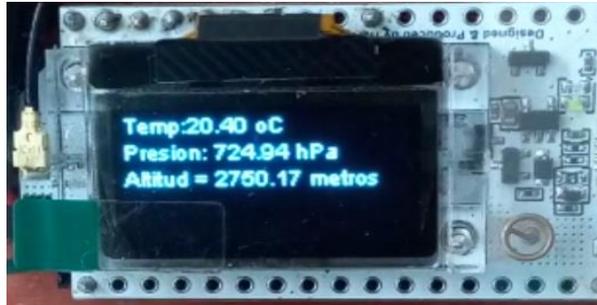
$$\frac{|2750.06 - 2750.71|}{2750.71} \times 100 = 0.01 \%$$

**Ecuación 3.8** Cálculo de error de la altitud.

El error del 0.01% tanto en la presión como en la altura indica una alta precisión en las mediciones realizadas. Sin embargo, es importante tener en cuenta que incluso los pequeños errores pueden afectar significativamente las mediciones a largo plazo. Por lo tanto, es importante corregir cualquier error, por pequeño que sea, para garantizar la precisión de las mediciones futuras.

### Prueba de funcionamiento a 300 (m) - sin línea de vista

En la Figura 3.36 se muestra el módulo transmisor con los valores de presión y altitud a 0 (m), mientras en la Figura 3.37 se muestra el módulo receptor con los valores de presión y altitud a una distancia de 300 (m).



**Figura 3.36** Módulo transmisor con los valores atmosféricos.



**Figura 3.37** Módulo receptor con los valores de presión y altitud.

En la Tabla 3.7 se muestra los valores de latitud, longitud, presión, altitud del módulo transmisor en el punto inicial y el módulo receptor a 300 (m).

**Tabla 3.7** Medidas barométricas a 300 (m).

	Distancia (m)	Latitud	Longitud	Presión (hPa)	Altitud (m)
Transmisor	0	-0.288704	-78.557564	724.94	2750.17
Receptor	300	-0.289191	-78.559654	725.04	2749.30

En la Ecuación 3.9 y 3.10 se muestra el cálculo de error de la presión y altitud respectivamente.

$$\frac{|725.04 - 724.94|}{724.94} \times 100 = 0.01\%$$

**Ecuación 3.9** Cálculo de error de la presión.

$$\frac{|2749.30 - 2750.17|}{2750.17} \times 100 = 0.03 \%$$

**Ecuación 3.10** Cálculo de error de la altitud.

Un error del 0.01% en la presión y un error del 0.03% en la altitud pueden ser aceptables, es importante garantizar que la precisión de las mediciones se mantenga en un nivel alto y constante para obtener resultados confiables.

**Prueba de funcionamiento a 400 (m) - sin línea de vista**

En la Figura 3.38 se muestra el módulo transmisor con los valores de presión y altitud a 0 (m), mientras en la Figura 3.39 se muestra el módulo receptor con los valores de presión y altitud a una distancia de 400 (m).



**Figura 3.38** Módulo transmisor con los valores atmosféricos.



**Figura 3.39** Módulo receptor con los valores de presión y altitud.

En la Tabla 3.8 se muestra los valores de latitud, longitud, presión, altitud del módulo transmisor en el punto inicial y el módulo receptor a 400 (m).

**Tabla 3.8** Medidas barométricas a 400 (m)

	Distancia (m)	Latitud	Longitud	Presión (hPa)	Altitud (m)
Transmisor	0	-0.288704	-78.557564	725	2749.84
Receptor	400	-0.288900	-78.560346	724.97	2749.84

En la Ecuación 3.11 y 3.12 se muestra el cálculo de error de la presión y altitud respectivamente.

$$\frac{|724.97 - 725|}{725} \times 100 = 0.01\%$$

**Ecuación 3.11** Cálculo de error de la presión

$$\frac{|2749.84 - 2749.84|}{2749.84} \times 100 = 0\%$$

**Ecuación 3.12** Cálculo de error de la altitud

Un error de la presión de 0.01% y un error de altura de 0% son muy buenos resultados y garantizan mediciones precisas y confiables.

La Figura 3.40 muestra la ubicación georreferenciada de los puntos donde se colocó el módulo transmisor (Tx) en el punto inicial y el módulo receptor (Rx) a distancias de 100, 200, 300 y 400 metros, respectivamente, sin línea de vista.



**Figura 3.40** Ubicación de los puntos georreferenciados.

### Tercera prueba de funcionamiento

Se llevará a cabo una comparación entre los valores de presión registrados por el sensor y los valores proporcionados por la aplicación Estación Meteorológica. En cuanto a la altitud, se utilizará el *software Google Earth* para su respectiva comparación. En la Figura 3.41 se presentan los valores de presión y altitud medidos por el módulo transmisor en un punto de partida, mientras que en las Figuras 3.42 y 3.43 se muestran

los valores de presión y altitud proporcionados por la aplicación Estación Meteorológica y el software Google Earth, respectivamente.



**Figura 3.41** Valores de presión y altitud medidos desde el sensor barométrico.



**Figura 3.42** Valor de la presión emitido por la aplicación Estación Meteorológica.



**Figura 3.43** Altura obtenida por el software Google Earth.

Las coordenadas geográficas del punto inicial son: -0.209497 de longitud y -78.489279 de latitud.

En la Tabla 3.9 se presentan los valores de las lecturas proporcionados por el sensor, así como por las fuentes adicionales. Por otro lado, la Ecuación 3.13 muestra el cálculo del error de la presión, mientras que la Ecuación 3.14 muestra el cálculo del error relacionado con la altura.

**Tabla 3.9** Valores de las lecturas del prototipo y de las fuentes.

Ambientes Barométricos	Fuentes utilizadas		
	Prototipo	Estación Meteorológica	Google Earth
Presión (hPa)	730.45	719.1	x
Altitud (m)	2690.54	x	2804

$$\frac{|730 - 719.1|}{719.1} \times 100 = 1.57 \%$$

**Ecuación 3.13** Cálculo de error de la presión.

$$\frac{|2690.54 - 2804|}{2804} \times 100 = 4.04 \%$$

**Ecuación 3.14** Cálculo de error de la altitud.

Es importante considerar el error en la presión de un 1.57% y el error en la altura de un 4.04% al interpretar los resultados de las mediciones. Estos errores pueden ser ocasionados por factores como la precisión del sensor, o la fuente, calibración y otros factores ambientales

#### **Cuarta prueba de funcionamiento - con línea de vista**

Para la cuarta prueba de funcionamiento se va a comparar las lecturas que envíe el módulo transmisor con los datos que reciba el módulo receptor, teniendo en cuenta la distancia y la no presencia de obstáculos.

#### **Prueba de funcionamiento en el punto inicial 0 (m)**

En la Figura 3.44 se muestra el módulo transmisor y el módulo receptor con los valores de presión y altitud en el punto inicial 0 (m).



**Figura 3.44** Módulo transmisor y receptor con valores de presión y altitud.

En la Tabla 3.10 se muestra los valores de latitud, longitud, presión, altitud del módulo transmisor y receptor en el punto inicial.

**Tabla 3.10** Medidas barométricas en el punto inicial 0 (m).

	Latitud	Longitud	Presión (hPa)	Altitud (m)
Transmisor	-0.209497	-78.489279	730.50	2689.45
Receptor	-0.209497	-78.489279	730.53	2689.67

En la Ecuación 3.15 y 3.16 se muestra el cálculo de error de la presión y altitud respectivamente.

$$\frac{|730.53 - 730.50|}{730.50} \times 100 = 0.004\%$$

**Ecuación 3.15** Cálculo de error de la presión.

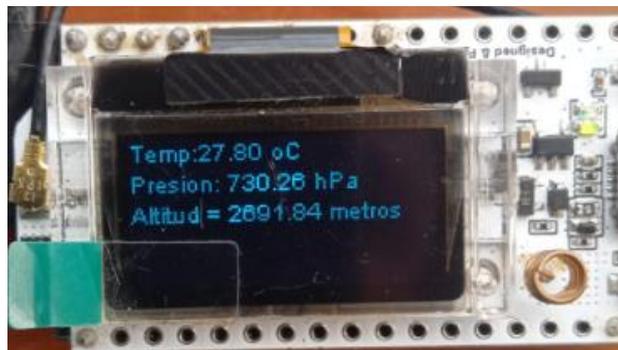
$$\frac{|2689.67 - 2689.45|}{2689.45} \times 100 = 0.008\%$$

**Ecuación 3.16** Cálculo de error de la altitud.

Un error de presión del 0.004 y un error de altitud del 0.008% son relativamente bajos y sugieren que las mediciones son bastante precisas

### Prueba de funcionamiento a 100 (m) con línea de vista

En la Figura 3.45 se muestra el módulo transmisor con los valores de presión y altitud a 0 (m), mientras en la Figura 3.46 se muestra el módulo receptor con los valores de presión y altitud a una distancia de 100 (m).



**Figura 3.45** Módulo transmisor con los valores atmosféricos.



**Figura 3.46** Módulo receptor con los valores de presión y altitud.

En la Tabla 3.11 se muestra los valores de latitud, longitud, presión, altitud del módulo transmisor en el punto inicial y el módulo receptor a 100 (m).

**Tabla 3.11** Medidas barométricas a 100 (m).

	Distancia (m)	Latitud	Longitud	Presión (hPa)	Altitud (m)
Transmisor	0	-0.209497	-78.489279	730.26	2691.84
Receptor	100	-0.210342	-78.489601	730.28	2692.16

En la Ecuación 3.17 y 3.18 se muestra el cálculo de error de la presión y altitud respectivamente.

$$\frac{|730.28 - 730.26|}{730.26} \times 100 = 0.002\%$$

**Ecuación 3.17** Cálculo de error de la presión.

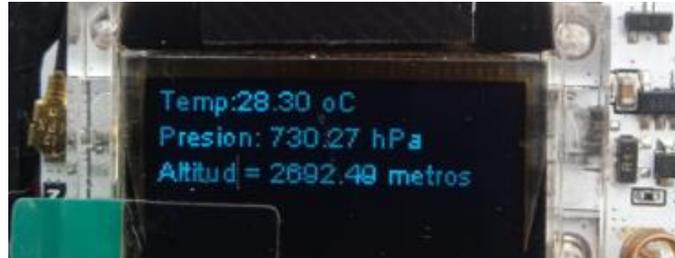
$$\frac{|2692.16 - 2691.84|}{2691.84} \times 100 = 0.01 \%$$

**Ecuación 3.18** Cálculo de error de la altitud.

Un error de presión del 0.002% y un error de altura del 0.01% son valores muy bajos en comparación con muchos otros tipos de mediciones, por lo tanto, se puede decir que las mediciones son altamente precisas y confiables. Estos errores indican que la variación en las mediciones de presión y altura es muy pequeña en comparación con una fuente de referencia, lo que sugiere una alta precisión en las mediciones

### Prueba de funcionamiento a 200 (m) - con línea de vista

En la Figura 3.47 se muestra el módulo transmisor con los valores de presión y altitud a 0 (m), mientras en la Figura 3.48 se muestra el módulo receptor con los valores de presión y altitud a una distancia de 200 (m).



**Figura 3.47** Módulo transmisor con los valores atmosféricos.



**Figura 3.48** Módulo receptor con los valores de presión y altitud.

En la Tabla 3.12 se muestra los valores de latitud, longitud, presión, altitud del módulo transmisor en el punto inicial y el módulo receptor a 200 (m).

**Tabla 3.12** Medidas barométricas a 200 (m)

	Distancia (m)	Latitud	Longitud	Presión (hPa)	Altitud (m)
Transmisor	0	-0.209497	-78.489279	730.27	2692.49
Receptor	200	-0.211408	-78.489552	729.84	2696.83

En la Ecuación 3.19 y 3.20 se muestra el cálculo de error de la presión y altitud respectivamente.

$$\frac{|729.84 - 730.27|}{730.27} \times 100 = 0.05\%$$

**Ecuación 3.19** Cálculo de error de la presión.

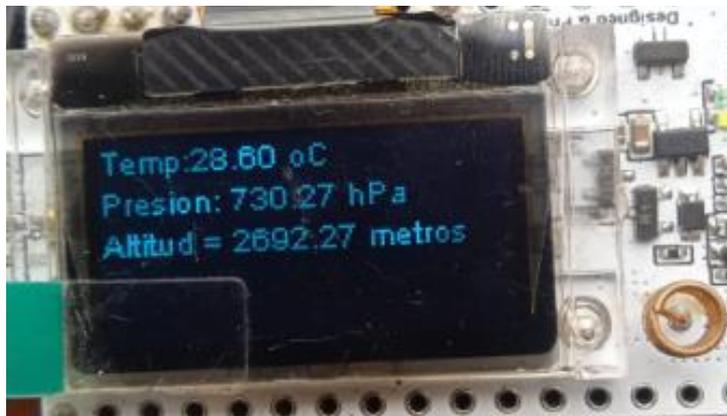
$$\frac{|2696.83 - 2692.49|}{2692.49} \times 100 = 0.16 \%$$

**Ecuación 3.20** Cálculo de error de la altitud.

La existencia de errores de presión y altitud relativamente bajos puede ser una indicación de que las mediciones son confiables, ya que tener una línea de vista despejada puede reducir las posibles interferencias que puedan afectar la precisión de las mediciones.

**Prueba de funcionamiento a 300 (m) - con línea de vista**

En la Figura 3.49 se muestra el módulo transmisor con los valores de presión y altitud a 0 (m), mientras en la Figura 3.50 se muestra el módulo receptor con los valores de presión y altitud a una distancia de 300 (m).



**Figura 3.49** Módulo transmisor con los valores atmosféricos.



**Figura 3.50** Módulo receptor con los valores de presión y altitud.

En la Tabla 3.13 se muestra los valores de latitud, longitud, presión, altitud del módulo transmisor en el punto inicial y el módulo receptor a 300 (m).

**Tabla 3.13** Medidas barométricas a 300 (m).

	Distancia (m)	Latitud	Longitud	Presión (hPa)	Altitud (m)
Transmisor	0	-0.209497	-78.489279	730.27	2692.27
Receptor	300	-0.212266	-78.489412	730.22	2692.38

En la Ecuación 3.21 y 3.22 se muestra el cálculo de error de la presión y altitud respectivamente.

$$\frac{|730.22 - 730.27|}{730.27} \times 100 = 0.006\%$$

**Ecuación 3.21** Cálculo de error de la presión.

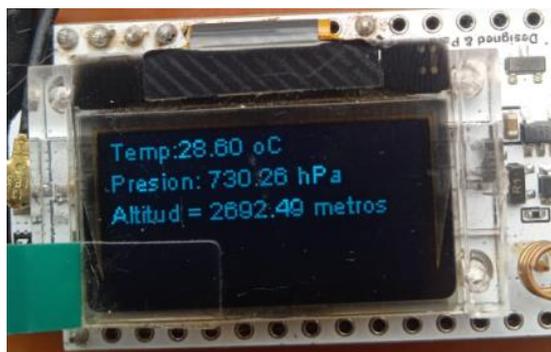
$$\frac{|2692.38 - 2692.27|}{2692.27} \times 100 = 0.004 \%$$

**Ecuación 3.22** Cálculo de error de la altitud.

Los errores de la presión y la altitud son tan bajos que las mediciones realizadas son confiables incluso a línea de vista.

### Prueba de funcionamiento a 400 (m) - con línea de vista

En la Figura 3.51 se muestra el módulo transmisor con los valores de presión y altitud a 0 (m), mientras en la Figura 3.52 se muestra el módulo receptor con los valores de presión y altitud a una distancia de 400 (m).



**Figura 3.51** Módulo transmisor con los valores atmosféricos.



**Figura 3.52** Módulo receptor con los valores de presión y altitud.

En la Tabla 3.14 se muestra los valores de latitud, longitud, presión, altitud del módulo transmisor en el punto inicial y el módulo receptor a 400 (m).

**Tabla 3.14** Medidas barométricas a 400 (m)

	Distancia (m)	Latitud	Longitud	Presión (hPa)	Altitud (m)
Transmisor	0	-0.209497	-78.489279	730.26	2692.49
Receptor	400	-0.213230	-78.489348	729.69	2698.68

En la Ecuación 3.23 y 3.24 se muestra el cálculo de error de la presión y altitud respectivamente.

$$\frac{|729.69 - 730.26|}{730.26} \times 100 = 0.07\%$$

**Ecuación 3.43** Cálculo de error de la presión.

$$\frac{|2698.68 - 2692.49|}{2698.49} \times 100 = 0.22 \%$$

**Ecuación 3.44** Cálculo de error de la altitud.

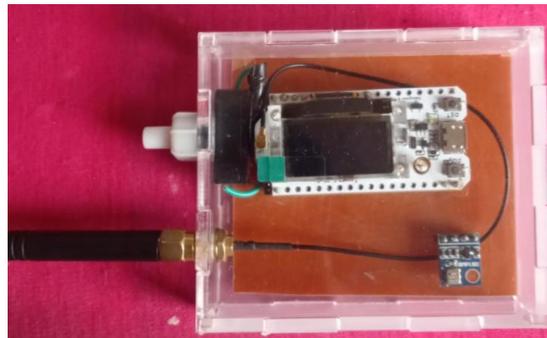
Las mediciones de presión tienen un margen de error del 0.07% y 0.22%, los cuales se consideran aceptables debido a la distancia desde el punto inicial y el tiempo de comunicación. Por lo tanto, se puede confiar en la precisión y confiabilidad de las lecturas.

La Figura 3.53 muestra la ubicación georreferenciada de los puntos donde se colocó el módulo transmisor (Tx) en el punto inicial y el módulo receptor (Rx) a distancias de 100, 200, 300 y 400 metros, respectivamente, con línea de vista.



**Figura 3.53** Ubicación de los puntos georreferenciados.

En la Figura 3.54 muestra el resultado final de los componentes electrónicos en la caja portable y en la Figura 3.55 muestra el módulo receptor en la caja portable.



**Figura 3.54** Caja portable con el módulo transmisor.



**Figura 3.55** Módulo receptor en la caja portable.

## 4 CONCLUSIONES

- La implementación del prototipo para la transmisión inalámbrica de altitud y presión del aire mediante tecnología LoRa se basó en un módulo transmisor que funciona como maestro. Este módulo recolecta datos atmosféricos mediante un sensor barométrico y se comunican los datos al módulo receptor a través de la tecnología LoRa. Esto permite monitorear la presión y altitud del aire en tiempo real y de manera portátil.
- La tecnología LoRa es un punto fundamental para la comunicación inalámbrica, ya que trabaja a frecuencias inferiores a 1 GHz, lo que le permite tener una gran cobertura de hasta los 5 km. Con la velocidad de transmisión que ofrece, es suficiente para enviar información importante, como mediciones atmosféricas.
- El módulo Heltec LoRa 32 es adecuado para implementar el prototipo debido a que cumple con todos los requisitos necesarios de *software* y *hardware*. Es un dispositivo "todo en uno" porque contiene un microcontrolador, una pantalla OLED, un puerto USB, lo que reduce tanto los costos como los componentes electrónicos y evita la necesidad de aumentar el tamaño del prototipo, manteniendo su estética.
- Para el diseño del PCB y el esquema del circuito, se utilizó el software EasyEDA, que cuenta con las bibliotecas necesarias de los componentes electrónicos requeridos. Además, este software proporciona imágenes descargables para obtener una vista previa del prototipo en 3D, lo que permite optimizar el espacio disponible y mejorar la estética de manera más eficiente.
- Se llevaron a cabo pruebas de funcionamiento en dos situaciones distintas: la primera se realizó sin línea de vista y la segunda con línea de vista. En ambos casos se trabajó con distancias de 100, 200, 300 y 400 metros, con el fin de evaluar el error entre los datos transmitidos y recibidos, considerando las posibles interferencias y obstáculos que pudieran afectar la comunicación.
- La medición de la presión atmosférica presentó un error del 0.69% al comparar la lectura del prototipo con una fuente adicional. Por otro lado, al comparar la lectura de altitud del prototipo con el programa Google Earth, se encontró un error del 5.79%. A pesar de estos errores, se considera que las mediciones obtenidas por el módulo transmisor y receptor son aceptables y válidas ya que los niveles de error en las lecturas del receptor con respecto al transmisor son menores al 2%

## 5 RECOMENDACIONES

- Se recomienda soldar los pines de cualquier tipo de sensor ya que, si no se hace, el contacto no será óptimo y puede fallar al momento de realizar la toma de datos.
- Es importante realizar una verificación de continuidad en las pistas después de finalizar su creación, ya que al taladrar o soldar los componentes puede suceder que las pistas se levanten, lo que podría interrumpir el flujo de corriente y provocar el mal funcionamiento del prototipo
- Se recomienda realizar pruebas con al menos dos tipos de sensores y elegir aquel que ofrezca mayor precisión y medidas más cercanas a la realidad, pero también que se integre de manera estética al prototipo. Es importante encontrar un equilibrio entre la funcionalidad y la apariencia del dispositivo.
- Se recomienda conectar correctamente los pines del sensor a los de la placa, ya que si se conectan de forma incorrecta, existe el riesgo de que la pantalla OLED se quemé.

## 6 REFERENCIAS BIBLIOGRÁFICAS

- [1] J. P. Peláez, Internet de las cosas (IoT) con ESP, Madrid, 2020.
- [2] P. Bertoleti, Proyectos con ESP32 y LoRa, Sao Paulo, 2019.
- [3] F. Bendix, Alrededor del trabajo de los metales, Reverte, 1973.
- [4] D. C. Jack Wilmore, Fisiología del deporte, Barcelona, 2004.
- [5] J. P. Peláez, Internet de las cosas con ESP, Madrid : Paraninfo, 2020.
- [6] programarfácil. [En línea]. Available: [https://programarfácil.com/blog/arduino-blog/bmp180-arduino/#Caracteristicas\\_del\\_sensor\\_BMP180](https://programarfácil.com/blog/arduino-blog/bmp180-arduino/#Caracteristicas_del_sensor_BMP180). [Último acceso: 28 Enero 2023].
- [7] H. Automation, «Heltec,» Mayo 2020. [En línea]. Available: <https://heltec.org/project/wifi-lora-32/>. [Último acceso: 30 Enero 2023].
- [8] Heltec Automation , Mayo 2020. [En línea]. Available: <https://heltec.org/project/wifi-lora-32/>. [Último acceso: 5 Febrero 2023].
- [9] Arcotel , «Agencia de Regulacion y Control de las Telecomunicaciones,» 2021. [En línea]. Available: [https://www.arcotel.gob.ec/wp-content/uploads/2021/10/PNF-V.6.0\\_14-07-21\\_v.1.pdf](https://www.arcotel.gob.ec/wp-content/uploads/2021/10/PNF-V.6.0_14-07-21_v.1.pdf). [Último acceso: 5 Febrero 2023].
- [10] NETTIGO. [En línea]. Available: <https://nettigo.eu/products/heltec-wifi-lora-32-v2-esp32-lora-oled-0-96>. [Último acceso: 08 Febrero 2023].
- [11] apogeeweb, 21 Febrero 2022. [En línea]. Available: <https://www.apogeeweb.net/circuitry/sx1276-transceiver-datasheet-circuit-sx1276-vs-sx1272.html>. [Último acceso: 09 Febrero 2023].
- [12] docrIoT, [En línea]. Available: [https://doc.riot-os.org/group\\_boards\\_esp32\\_heltec-lora32-v2.html](https://doc.riot-os.org/group_boards_esp32_heltec-lora32-v2.html). [Último acceso: 07 febrero 2023].
- [13] Visitronica. [En línea]. Available: <https://www.vistronica.com/comunicaciones/wifi/modulo-transceptor-lora-sx1276-915mhz-detail.html>. [Último acceso: 09 Febrero 2023].

- [14] codigoelectronica , 18 Julio 2018. [En línea]. Available: <http://codigoelectronica.com/blog/bmp180-sensor-de-temperatura-con-arduino>. [Último acceso: 09 Febrero 2023].
- [15] Rantec, [En línea]. Available: <https://rantec.mx/tienda/sensor-de-presion-bmp280-3-3v-temperatura-humedad-bme-economico/>. [Último acceso: 09 Febrero 2023].
- [16] Bosch, «Bosch Datasheet BMP180,» 5 Abril 2013. [En línea]. Available: <https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf>. [Último acceso: 09 Febrero 2023].
- [17] Bosch Datasheet, «Bosch sensor BMP280,» 8 Enero 2018. [En línea]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1132069/BOSCH/BMP280.html>. [Último acceso: 09 Febrero 2023].
- [18] Easyeda, [En línea]. Available: <https://easyeda.com/es>. [Último acceso: 10 Febrero 2023].

## 7 ANEXOS

La lista de los **Anexos** se muestra a continuación:

ANEXO I. Certificado de originalidad

ANEXO II. Enlaces

ANEXO III. Diagrama de flujo del módulo transmisor y receptor.

ANEXO IV. Código de programación del módulo transmisor y receptor

# ANEXO I: Certificado de Originalidad

## CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 2 de marzo de 2023

De mi consideración:

Yo, ANDRÉS FERNANDO REYES CASTRO, en calidad de Director del Trabajo de Integración Curricular titulado IMPLEMENTACIÓN DE UN PROTOTIPO PARA LA TRANSMISIÓN INÁLMBRICA DE ALTITUD Y PRESIÓN DEL AIRE BASADO EN TECNOLOGÍA LORA elaborado por el estudiante ALEX STEVEN ORBE ENCARNACIÓN de la carrera en TECNOLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 14%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el link del informe generado por la herramienta Turnitin.

[ReporteSimilitud.pdf](#)

Atentamente,

ANDRÉS FERNANDO REYES CASTRO

Docente

Escuela de Formación de Tecnólogos

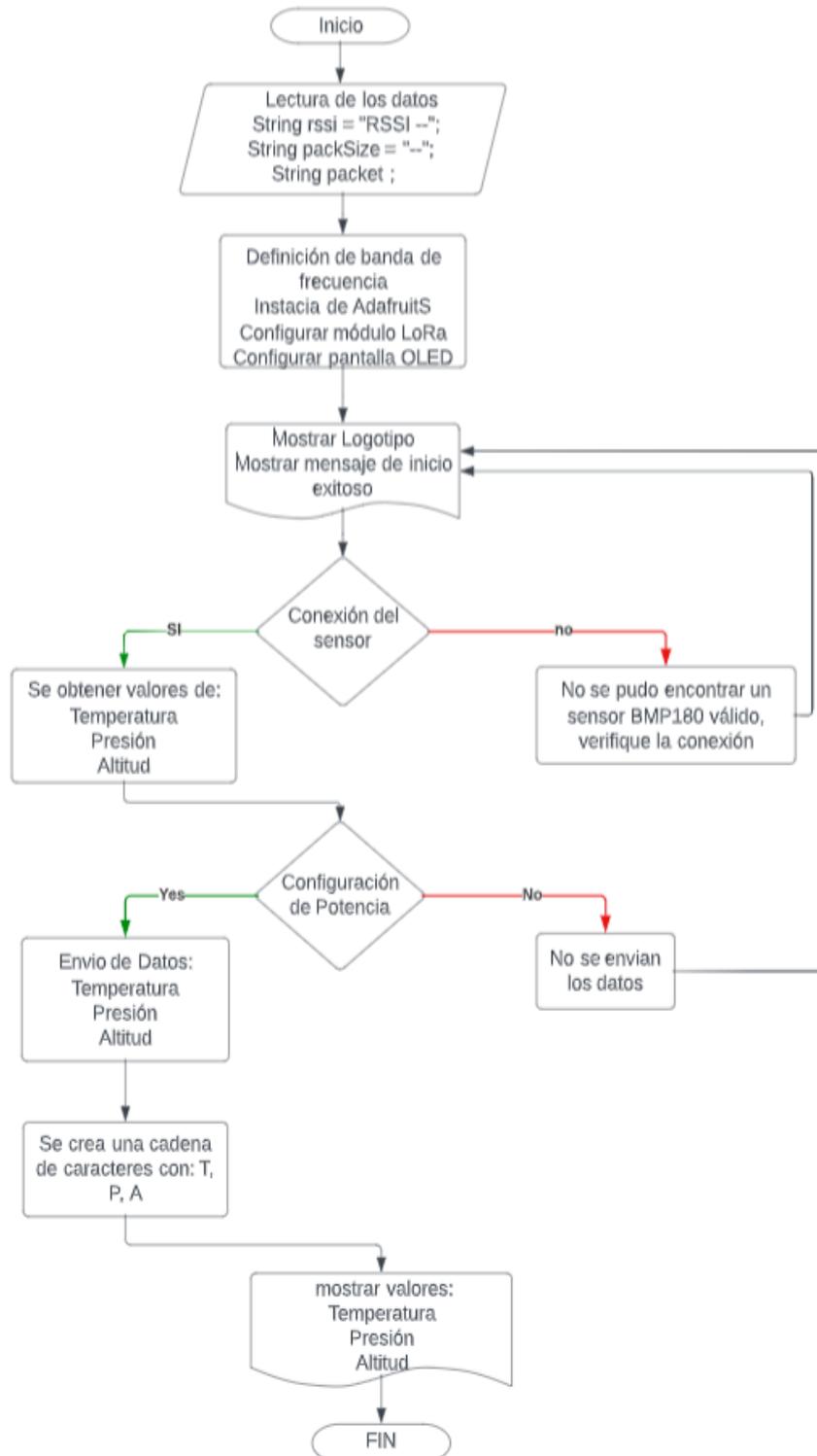
## ANEXO II: Enlaces



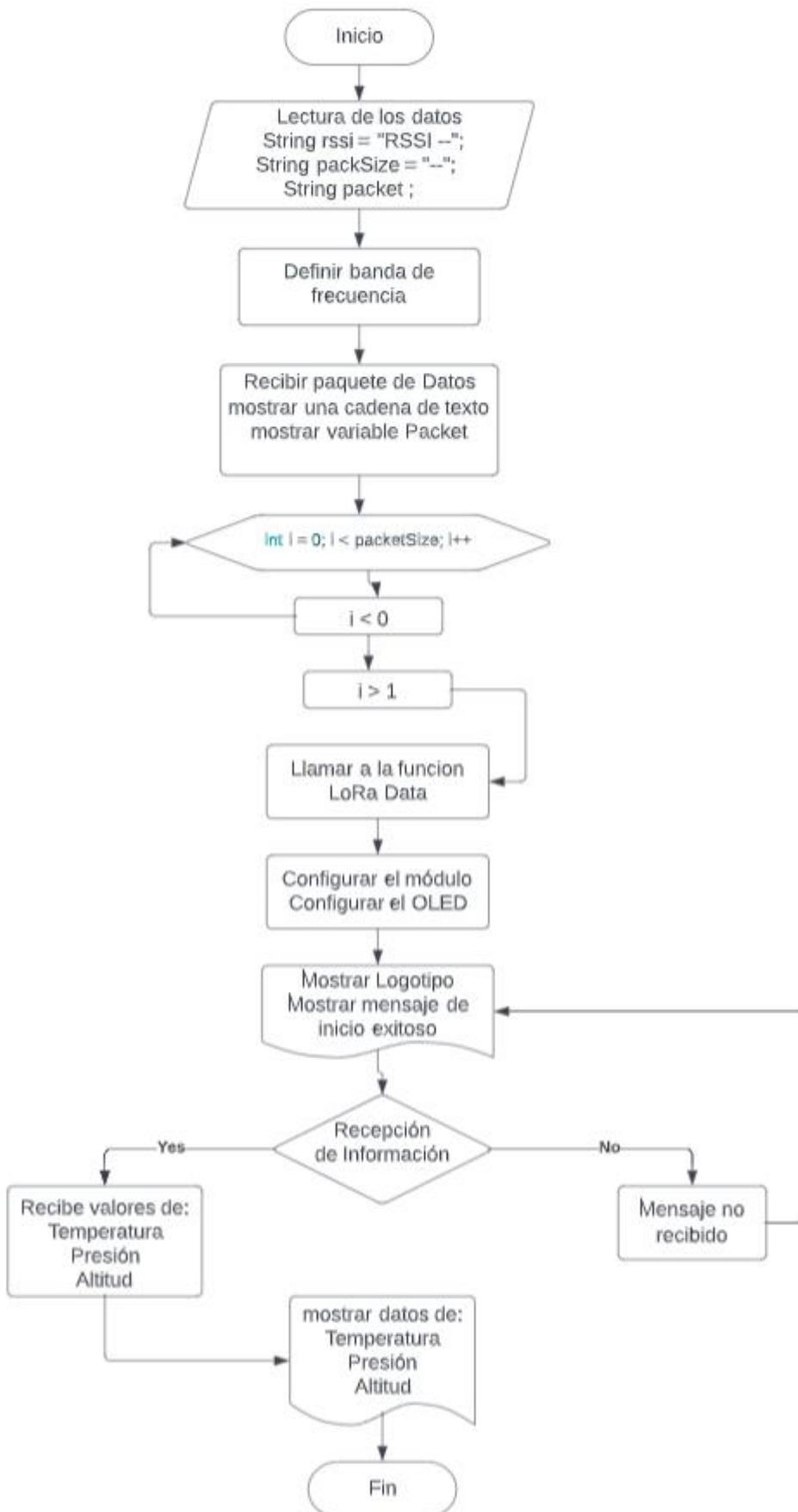
**Anexo II.I** Código QR de la implementación y pruebas de funcionamiento

# ANEXO III: DIAGRAMA DE FLUJO DEL MÓDULO TRANSMISOR Y RECEPTOR

## Diagrama del Módulo Transmisor



## Diagrama del Módulo Receptor



## ANEXO IV: Códigos Fuente

### Código Fuente del Módulo Transmisor

```
#include "heltec.h" //librería que utiliza HELTEC

#include "images.h"

#include <Adafruit_BMP085.h> //librería para el sensor BMP 180

Adafruit_BMP085 bmp; // Instancia de la clase adafruit y bmp

#define BAND 915E6 // Definir banda de frecuencia

unsigned int counter = 0; // Declaración de variable e inicialización

String rssi = "RSSI --"; // Variable para el indicador de fuerza de señal

String packSize = "--"; // Variable para el tamaño del paquete

String packet ; // Variable utilizada para el paquete

void logo() // función para mostrar el logo de Heltec

{

    Heltec.display->clear(); // se limpia el display

    Heltec.display->drawXbm(0,5,logo_width,logo_height,logo_bits); // Se configura la
posición ancho, largo

    Heltec.display->display(); // Se actualiza el display

}

void setup()

{

    Heltec.begin(true /*DisplayEnable Enable*/, true /*Heltec.Heltec.Heltec.LoRa
Disable*/, true /*Serial Enable*/, true /*PABOOST Enable*/, BAND /*long BAND*/); // se
inicializa el módulo
```

```

Heltec.display->init(); // Instancia init para inicializar la placa

Heltec.display->flipScreenVertically(); // Se ajusta la orientación de la pantalla

Heltec.display->setFont(ArialMT_Plain_10); // Se configura el tipo de letra

logo(); // Llama a la función logo

delay(1500); // Tiempo de espera 1.5 (s)

Heltec.display->clear(); // Se limpia el display

Heltec.display->drawString(0, 0, "Inicio exitoso!"); // Mensaje exitoso

Heltec.display->display(); // Se actualiza el display

delay(1000); // tiempo de espera 1(s)

if (!bmp.begin()) { // Bucle if y se inicializa el sensor

    Serial.println("No se pudo encontrar un sensor BMP085 válido, comprueba el
cableado!"); // Mensaje de error en caso que este mal conectado

    while (1) {}

}

}

void loop() // Función loop

{

    LoRa.beginPacket(); // Se inicializa el paquete de LoRa

    counter++; // Aumenta counter en 1

    String packet ; // almacena valores de T, P, A y los concatena

    Serial.print("Temperatura = "); // Imprime Temperatura en el monitor serial

        Serial.print(bmp.readTemperature()); // Obtiene valores de temperatura

        Serial.println(" *C"); // Imprime simbolo de grados centigrados

    Serial.print("Presion = "); // Imprime Presion en el monitor serial

    Serial.print(bmp.readPressure()); // Obtiene valores de presión

```

```

Serial.println(" Pa"); // Imprime simbolo de Pascales

Serial.print("Altitud = "); // Imprime Altitud en el monitor serial

Serial.print(bmp.readAltitude()); // Obtiene valores de presión

Serial.println(" metros"); // Imprime simbolo de Pascales

Serial.print("Presion a lvl del mar (calculated) = "); // Imprime presión a lvl del mar

Serial.print(bmp.readSealevelPressure()); // Obtine valores a lvl del mar

Serial.println(" Pa"); // Imprime simbolo de Pascales

Serial.print("Altitud real = "); // Imprime Real altitude

    Serial.print(bmp.readAltitude(101500)); // Obtiene valores de temperatura y los
multiplica por 101500

    Serial.println(" metros"); // Imprime metros

    Serial.println();

    LoRa.setTxPower(14,RF_PACONFIG_PASELECT_PABOOST); // Configuración de la
potencia a 14 (dBm)

    LoRa.print("Temp:"); // Paquete de LoRa que imprime Temp

    LoRa.print(bmp.readTemperature()); // Paquete de LoRa que obtiene la temperatura

    LoRa.println(" oC          "); // Paquete de LoRa que imprime simbolo de grados
centigrados

    LoRa.print("Presion: "); // Imprime la palabra Presión

    LoRa.print(bmp.readPressure()/100.F); // Paquete de LoRa que obtiene la presión
divido para 100 para obtener en Hpa

    LoRa.println(" hPa          "); // Paquete de LoRa que imprime simbolo de Hectopascales

    LoRa.print("Altitud: "); // Imprime la palabra Presión

    LoRa.print(bmp.readAltitude(101500)); // Paquete de LoRa que obtiene la altitud
multiplicado para 101500 para obtener la altura a lvl del mar

    LoRa.println(" metros "); // Imprime la palabra metros

    LoRa.endPacket(); // finaliza el paquete LoRa

```

```

    packet="Temp:"+String(bmp.readTemperature())+" oC          "+"Presion:
"+String(bmp.readPressure()/100.F)+" hPa          "+"Altitud   =
"+String(bmp.readAltitude(101500))+" metros "; // Crea una variable de caracteres que
contiene T, P, A

    Heltec.display->clear(); // Se limpia el display

    Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT); // Se alinea el texto

    Heltec.display->setFont(ArialMT_Plain_10); // Se ajusta el tamaño de letra

    Heltec.display->drawStringMaxWidth(0, 0, 128, packet); // Imprime la variable packet
alineando en el display

    Heltec.displaySS->display(); // se actualiza

    digitalWrite(LED, HIGH); // encender led

    delay(100);          // esperar tiempo de 100 (ms)

    digitalWrite(LED, LOW); // led apagado

    delay(100); // esperar tiempo de 100 (ms)

}

```

### **Código Fuente del Módulo Receptor**

```

#include "heltec.h" //libreria que utiliza HELTEC

#include "images.h"

#define BAND 915E6 // Definir banda de frecuencia

String rssi = "RSSI --"; // Variable para el indicador de fuerza de señal

String packSize = "--"; // Variable para el tamaño del paquete

String packet ; // Variable utilizada para el paquete

void logo() // función para mostrar el logo de Heltec

{

    Heltec.display->clear(); // se limpia el display

    Heltec.display->drawXbm(0,5,logo_width,logo_height,logo_bits); // Se configura la
posición ancho, largo

    Heltec.display->display(); // Se actualiza el display

```

```

}

void LoRaData(){ // Función LoRa Data

    Heltec.display->clear(); // se limpia el display

    Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT); // se alinea el texto

    Heltec.display->setFont(ArialMT_Plain_10); // Se ajusta la letra

    Heltec.display->drawStringMaxWidth(0 , 0 , 128, packet); // Imprime la variable packet
    alineando en el display

    Heltec.display->display(); // Se actualiza el display

}

void cbk(int packetSize) { // Inicia la funcion Ccbk

    packet = ""; // variable packet

    packSize = String(packetSize,DEC); // tamaño del paquete

    for (int i = 0; i < packetSize; i++) { packet += (char) LoRa.read(); } // Bucle for

    rssi = "RSSI " + String(LoRa.packetRssi(), DEC) ; // Nivel de RSSI

    LoRaData(); // Llama a la función LoRa Data

}

void setup()

{

    Heltec.begin(true /*DisplayEnable Enable*/, true /*Heltec.Heltec.Heltec.LoRa
Disable*/, true /*Serial Enable*/, true /*PABOOST Enable*/, BAND /*long BAND*/); // se
inicializa el módulo

    Heltec.display->init(); // Instancia init para inicializar la placa

    Heltec.display->flipScreenVertically(); // Se ajusta la orientación de la pantalla

    Heltec.display->setFont(ArialMT_Plain_10); // Se configura el tipo de letra

    logo(); // Llama a la funcion logo

    delay(1500); // Tiempo de espera 1.5 (s)

    Heltec.display->clear(); // Se limpia el display

```

```
Heltec.display->drawString(0, 0, "Inicio exitoso!"); // Mensaje exitoso

Heltec.display->display(); // Se actualiza el display

delay(1000); // tiempo de espera 1(s)

LoRa.receive(); // Llama a la función LoRa receive
}

void loop() { // Función loop

    int packetSize = LoRa.parsePacket(); // parsepacket se encarga de leer el tamaño de
    la variable packsize

    if (packetSize) { cbk(packetSize); } // Comparación del tamaño del paquete

    delay(10);

}
```