

Laboratorio:

Proyecto final

Estudiantes:

Edgar Santiago Quiroz Puentes
Rafael Alejandro Garzon Andosilla
Sebastian Restrepo Sanchez
Guillermo Antonio Hernandez Ortiz
est.edgar.squiroz@unimilitar.edu.co
est.rafael.agarzon@unimilitar.edu.co
est.sebastian.rest@unimilitar.edu.co
est.guillermo.hern@unimilitar.edu.co

Profesor:

Jose de Jesus Rugeles Uribe
jose.rugeles@unimilitar.edu.co

Universidad Militar Nueva Granada
Bogotá, (3/Junio/2025)

I. INTRODUCCIÓN

La necesidad de conocer la ubicación precisa de personas u objetos en espacios cerrados ha impulsado el desarrollo de tecnologías de localización indoor. A diferencia del posicionamiento GPS, que pierde precisión en entornos cerrados, los sistemas basados en mediciones de señal inalámbrica ofrecen una solución efectiva. Este proyecto tiene como objetivo diseñar e implementar un sistema de localización en tiempo real mediante trilateración inalámbrica en los espacios cerrados de la universidad.

II. OBJETIVO GENERAL

Desarrollar un sistema funcional de localización en interiores que utilice mediciones de RSSI desde múltiples nodos, empleando técnicas de trilateración, para determinar en tiempo real la posición de un nodo móvil en un entorno con obstáculos.

III. OBJETIVOS ESPECIFICOS

- Aprender a utilizar la trilateración usando como base el RSSI de 4 nodos fijos.
- Explorar los diferentes tipos de comunicación WiFi para diferentes proyectos de comunicación inalámbrica.
- Comprender los diferentes protocolos y sus diferencias de comunicación WiFi.

IV. ESTADO DEL ARTE

La tecnología de posicionamiento en interiores se ha convertido en un campo de investigación creciente en la moderna tecnología de las comunicaciones debido a la necesidad de localizar equipos o personas en espacios donde los Sistemas de Posicionamiento Global (GPS) no pueden operar. A diferencia del GPS, que funciona con precisión en espacios abiertos, las señales de los satélites se ven considerablemente atenuadas por techos, paredes y obstáculos típicos de los entornos cerrados. Por eso se han desarrollado soluciones alternativas basadas en tecnologías de corto alcance como WiFi, Bluetooth, ZigBee y RF propietaria, cada una con su propio método de estimación de distancia o posición.

Desde entonces, se han propuesto diversas mejoras del uso de RSSI, como el filtrado, el modelado adaptativo y la combinación con otros sensores. Por ejemplo, Liu et al. (2007) realizaron una revisión sistemática de las técnicas de localización en interiores y las clasificaron según su precisión, escalabilidad y facilidad de implementación. El RSSI se caracteriza por su bajo coste y su compatibilidad con las redes inalámbricas existentes; sin embargo, su precisión puede verse afectada por fenómenos como el desvanecimiento, las reflexiones múltiples y las interferencias electromagnéticas, especialmente en entornos obstruidos.

En cuanto a las tecnologías disponibles, se han explorado varias alternativas. Bluetooth Low Energy (BLE) se utiliza ampliamente para el seguimiento de la ubicación en tiendas, museos y centros comerciales, principalmente debido a su bajo consumo de energía y compatibilidad con dispositivos móviles; sin embargo, estudios como el de Faragher y Harle (2015) han

demostrado que la estabilidad RSSI de BLE puede degradarse en entornos dinámicos, lo que resulta en una baja precisión sin filtros adicionales. Por otro lado, tecnologías como la banda ultra ancha (UWB) ofrece una precisión mucho mayor, pero a costa de un elevado coste y de la necesidad de equipos especializados. En este contexto, sistemas modulares como el NRF24L01, que opera en la banda ISM de 2,4 GHz, representan una solución intermedia viable. El sistema es capaz de medir RSSI, tiene un bajo consumo de energía, buenas velocidades de transmisión de datos (hasta 2 Mb/s) y puede integrarse fácilmente con microcontroladores como el Raspberry Pi Pico W.

Por tanto, la selección del protocolo y los módulos para este proyecto se basó en un compromiso entre precisión, coste, disponibilidad y escalabilidad. El módulo NRF24L01 se eligió principalmente por su capacidad para ejecutar varios nodos simultáneamente, crear canalizaciones de recepción independientes, su bajo consumo de energía y su integración inherente con el entorno de programación MicroPython. Además, su uso en combinación con sensores inerciales (como el MPU6050) y cámaras ESP32CAM con reconocimiento facial permite ampliar el sistema no sólo para la detección, sino también para el reconocimiento visual del nodo móvil, ampliando las aplicaciones en sistemas de seguridad, vigilancia y automatización.

Por último, es importante mencionar el papel del software en el tratamiento y la visualización de datos. Herramientas como Python y MATLAB son ampliamente utilizadas para implementar modelos de propagación, filtrar señales y calcular la localización. El proyecto presentado por Rúgeles (2024), cuyo código está disponible en GitHub, muestra una arquitectura completa para NRF24L01, Raspberry Pi Pico y cálculo de RSSI en tiempo real, que puede servir de referencia para el desarrollo de este proyecto.

En general, aunque hay muchas tecnologías y métodos disponibles en el entorno interior, el uso de RSSI en combinación con los módulos NRF24L01 representa una solución práctica y comercial en un entorno real. Particularmente su alta precisión, su facilidad de uso y flexibilidad lo hacen especialmente adecuado para aplicaciones educativas y modelos de trabajo con presupuestos bajos.

V. METODOLOGÍA

A. Preparación

Se comenzó con el planeamiento del proyecto al elegir los materiales que se necesitarían utilizar. Para armar los 5 nodos fue necesario contar con una raspberry pi-pico W, un módulo con antena NRF24L01 y una pantalla OLED de 28x64.

Cuatro de estos nodos fueron programados para conectarse a un Access Point (AP) y en la pantalla mostrar las mediciones en tiempo real del RSSI que reciben de este, además sería energizado con una batería de 9V, con la ayuda de un módulo fuente MB102 o directamente con una batería portátil.

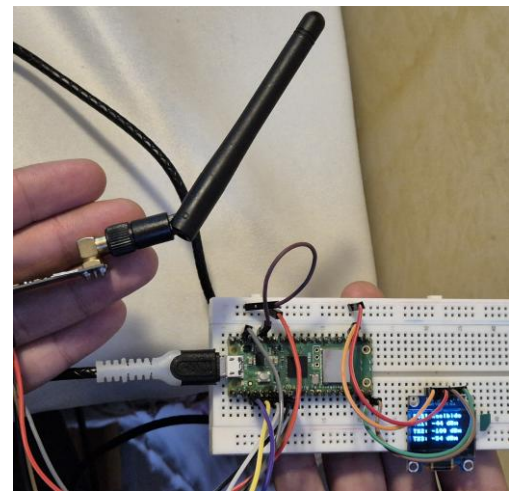


Ilustración 1. Montaje de los nodos

El quinto nodo fue apodado como “Receptor”, el cual está programado para el recibimiento en tiempo real de los datos enviados desde los otros nodos por medio de la programación en MycroPython conocida como “pipes”, de esta forma múltiples comunicaciones pueden ocurrir simultáneamente en eta raspberry.

El AP (Access point) fue armado utilizando un ESP-CAM, un giroscopio MPU5060 y un segundo ESP32. Este montaje, al ser energizado podrá brindar una red WiFi para que diferentes hosts (nodos) se puedan conectar y poder rastrear su movimiento.

Este módulo fue montado sobre un carro a control remoto proporcionado por el profesor para poder demostrar el cambio en tiempo real del RSSI en cada nodo. Fue instalado en el techo del carro, siendo energizado con una batería de 9V, sujetado con velcro para evitar incidentes como daños al módulo.



Ilustración 2. Montaje inicial del AP

B. Programación de los dispositivos

Se desarrollaron varios códigos para lograr el objetivo del laboratorio que fueron probándose y aplicándose al tiempo que se desarrollaban las pruebas.

Los tres primeros códigos implementados en el proyecto corresponden a los nodos transmisores llamados TX2, TX3 y TX4. Su función principal es la de capturar el valor de intensidad de la señal RSSI, promediarlo, visualizarlo localmente en una

pantalla OLED y enviarlo de forma inalámbrica mediante un módulo NRF24L01 al nodo concentrador.

A nivel funcional, los tres códigos comparten la misma estructura y lógica general. Inician estableciendo conexión con la red WiFi configurada, miden la intensidad de la señal utilizando el método “wifi.status('rssi')”. Calculan un promedio móvil basado en las últimas diez mediciones para reducir fluctuaciones, y transmiten el resultado junto con un identificador único mediante el protocolo NRF24L01. Además, todos utilizan el mismo esquema de pines para la conexión SPI del transceptor y para la interfaz I2C del display OLED SSD1306, donde se muestra el identificador del nodo, el nombre de la red, la IP local asignada y el valor del RSSI promedio.

Sin embargo, aunque estos códigos son prácticamente idénticos en su estructura, presentan pequeñas diferencias fundamentales que permiten distinguirlos entre sí. La primera diferencia está en el identificador del nodo transmisor ('TX_ID'), que varía entre 2, 3 y 4 según el nodo. Este valor se incluye en el paquete de datos enviado, permitiendo que el receptor identifique el origen de la medición.

La segunda diferencia se encuentra en la dirección del canal de transmisión ('TX_ADDRESS') asignada a cada nodo; cada uno posee una dirección única, lo cual garantiza que no haya interferencia entre ellos y que el receptor pueda manejar múltiples transmisiones sin ambigüedad. Finalmente, cada nodo muestra en su pantalla OLED su nombre respectivo, de esta manera se permite una rápida identificación visual en el entorno de pruebas. Estas diferencias, aunque mínimas, son cruciales para la correcta operación del sistema, ya que aseguran la independencia y trazabilidad de las señales emitidas por cada transmisor dentro del sistema de localización en interiores.

El código implementado para el funcionamiento del nodo receptor NRF24L01 cumple la función de centralizar y gestionar la recepción de los datos enviados por los transmisores. Su tarea principal es escuchar continuamente el canal de comunicación inalámbrica configurado (canal 10), identificar el transmisor del cual proviene cada mensaje, extraer la información correspondiente al valor de RSSI promedio, y mantener un historial actualizado de estas mediciones. Al igual que los transmisores, este código utiliza comunicación SPI para interactuar con el módulo NRF24L01 y conexión I2C para mostrar los valores en una pantalla OLED SSD1306.

La lógica del receptor se basa en el uso de las “pipes” de recepción con las direcciones únicas asignadas a cada transmisor. A medida que se reciben datos, el receptor los desempaca utilizando “struct.unpack”, actualiza el historial de valores RSSI por nodo y registra el tiempo en que se recibió la última transmisión. Si uno de los transmisores deja de enviar datos durante más de 40 segundos, el sistema detecta un llamado “timeout” y restablece su valor RSSI a -100 dBm como medida de advertencia indicando su desconexión, así se puede identificar rápidamente interrupciones en la comunicación.

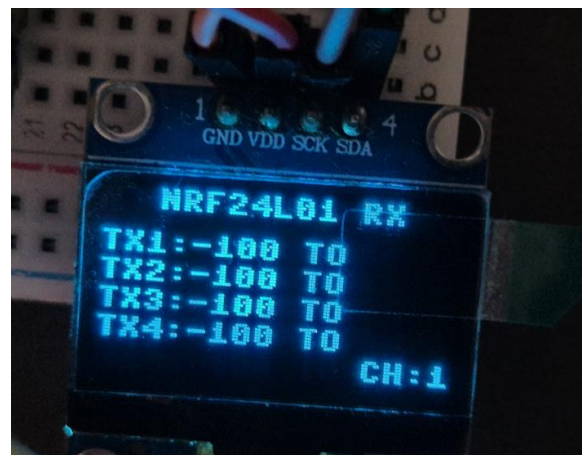


Ilustración 3. Nodo receptor sin conexión



Ilustración 4. Nodo receptor conectado

El receptor también se encarga de enviar los datos procesados al computador central en formato JSON, lo cual facilita su lectura y análisis mediante herramientas externas o scripts de visualización. Adicionalmente, en cada iteración del ciclo principal, el sistema actualiza la pantalla OLED con los valores RSSI actuales de cada transmisor, lo cual permite un monitoreo en tiempo real del comportamiento del sistema.

Se hizo un código con el propósito visualizar en tiempo real las mediciones RSSI enviadas por los nodos transmisores y recolectadas por el nodo receptor. Para ello, el sistema utiliza una conexión serial entre la computadora y el receptor por medio de un puerto USB, a través de la cual se reciben continuamente los datos formateados en JSON que contienen los valores RSSI de los transmisores TX1 a TX4.

La lógica del programa se basa en el uso de la biblioteca matplotlib, junto con su módulo de animación, para crear una gráfica dinámica que se actualiza constantemente conforme llegan nuevas mediciones. Internamente, el código mantiene un historial de los últimos 100 puntos por cada transmisor, lo que permite observar la evolución de la señal de cada nodo sin saturar la memoria ni sobrecargar la visualización.

Al iniciar, el script abre el puerto serial especificado con una velocidad de baudios de 115200. Luego, entra en un bucle

controlado por “FuncAnimation”, el cual ejecuta periódicamente una función encargada de leer nuevas líneas del puerto.

Cada transmisor tiene asignado un color distinto: rojo para TX1, verde para TX2, azul para TX3 y naranja para TX4. Se visualizan simultáneamente en una misma figura, permitiendo comparar en tiempo real la intensidad de señal relativa entre los nodos. La escala del eje Y está configurada para mostrar valores típicos de RSSI entre -110 y -20 dBm, mientras que el eje X representa el número de muestras recientes.

El sistema es fundamental para el análisis y la validación visual del comportamiento del sistema de localización, ya que permite observar en tiempo real cómo varía la señal de cada transmisor con respecto al movimiento del nodo móvil, facilitando la depuración, la interpretación de resultados y la verificación experimental del modelo de propagación.

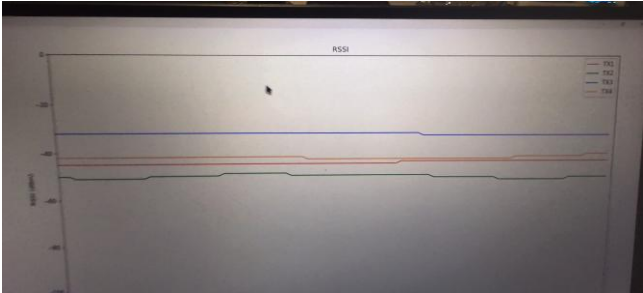


Ilustración 5. Sistema de visualización en tiempo real

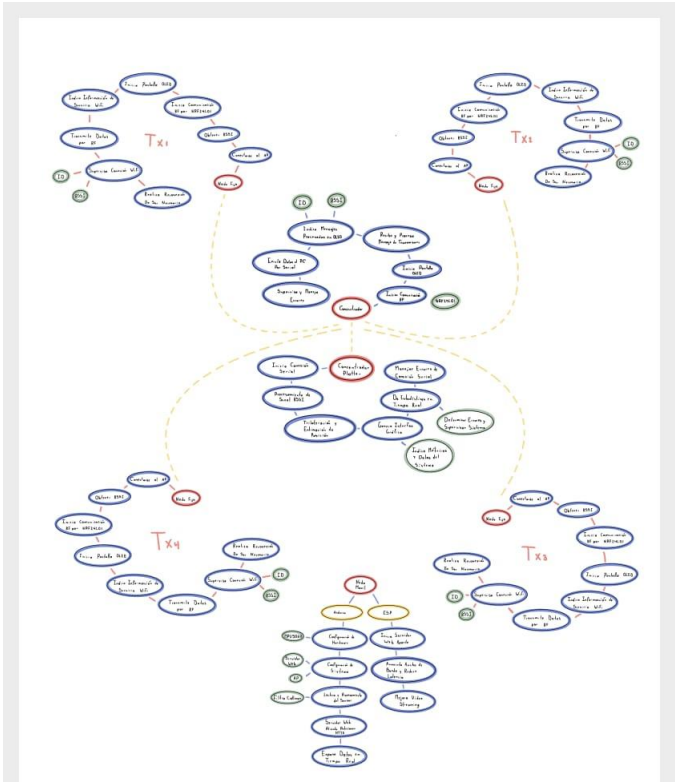


Ilustración 6. Diagrama del funcionamiento del código

C. PCB

Se hizo el proceso para presentar los nodos en una placa de PCB para hacer más cómoda la transportación y disposición de estos a la hora de hacer las pruebas. Se utilizó la herramienta de KiCad para esta tarea, gracias a esta y sus múltiples bibliotecas de dispositivos se pudo hacer un diagrama esquemático del proyecto. Debido al uso de dispositivos específicos como la pantalla y el módulo fuente, estos no contaban con bibliotecas dentro de la aplicación por lo que se tuvo que hacerlas desde cero, haciendo su esquemático y luego su huella.

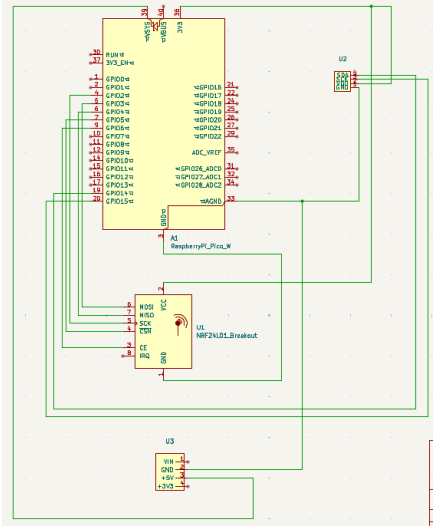


Ilustración 7. Diseño esquemático

La disposición de los pines se puede ejemplificar de mejor forma en un diseño más práctico.

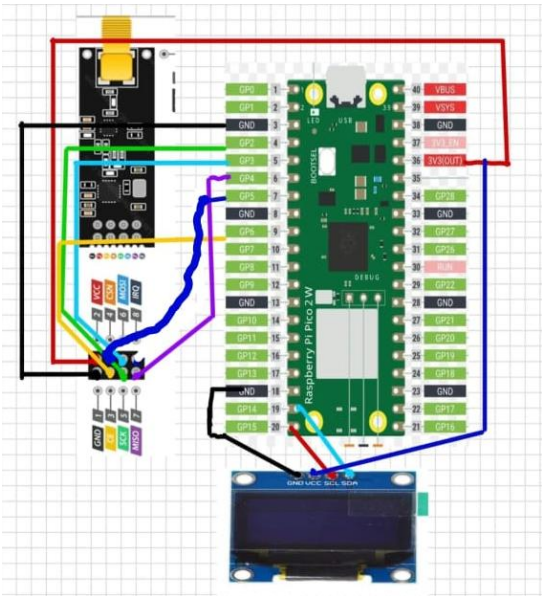


Ilustración 8. Diseño práctico

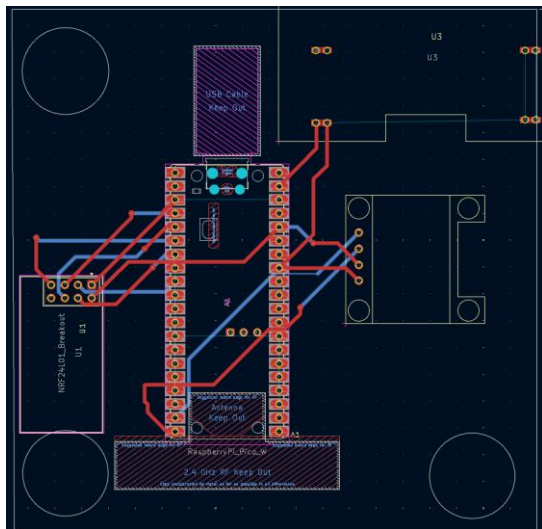


Ilustración 9. Diseño de la PCB

El diseño muestra una disposición de los elementos para que quepa dentro de una placa, hay tracks en color rojo que representan las líneas de conexión que se encuentran en la cara superior de la PCB y las líneas azules serían los tracks en la cara inferior de cobre, de esta forma se puede comprimir más el diseño ya que los tracks rojos se pueden sobreponer y no afectar a los tracks azules.

Se envió el diseño a una empresa dedicada a esto, pero por disposición de esta no fue posible tenerla a tiempo para las pruebas y la presentación de proyecto, por lo que se optó por el uso de baquelitas y soldar el diseño por medio propios.

Se imprimió la cara superior y luego la cara inferior (en forma espejo) en papel termotransferible con la ayuda de una impresora láser y se hizo el proceso de planchado para tener al menos un nodo montado en PCB. Los demás nodos, por comodidad y tiempo, fueron soldados en baquelitas.

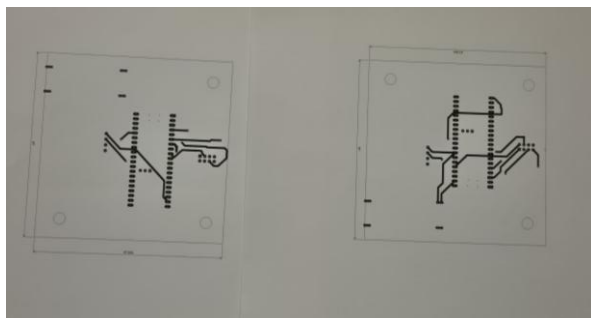


Ilustración 10. Impresiones de PCB



Ilustración 11. Nodo montado en vácueta

En este proceso se pudo evidenciar que la realización de PCB con métodos caseros puede resultar afectada por la imprecisión de la mano humana, además, la aplicación del proceso con materiales rústicos genera grandes problemas cuando se desea imprimir pistas o guías de un tamaño considerable en un entorno con espacio limitado por los módulos, la disposición de los componente también resultan en complicaciones si se tiene en cuenta el tamaño necesario de los pads para realizar la transición entre las dos caras del circuito.

Ya que el proceso anterior no consiguió cumplir con las expectativas para solucionar el inconveniente de practicidad, se optó por realizar la conexión de los componentes de cada nodo por medio de una baquelita universal, sin embargo, los resultados no fueron para nada los esperados.

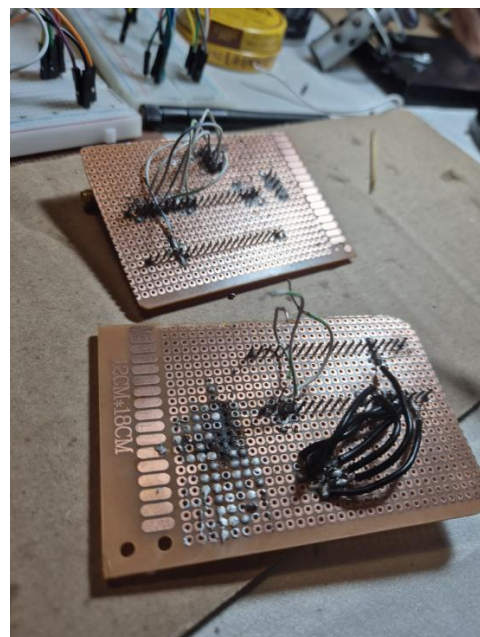


Ilustración 12. Prototipo de vácuetas

En un principio el proceso resultó ser adecuado, pues se optimizaba el espacio y el funcionamiento del dispositivo, sin embargo, poco después al realizar las pruebas correspondientes, los nodos comenzaron a fallar de forma inesperada si opción de arreglo aparente, por lo que se optó por realizar las pruebas mediante la disposición aplicada al principio del desarrollo del laboratorio.

D. Nodo móvil

Se compone de un Arduino r4 (ESP32s3mini), un MPU6050 (giroscopio) y un ESP32-CAM. Se utilizaron hasta 4 tipos diferentes de AP: un ESP simple con una antena integrada en la PCB, luego se modificó para integrarle una antena, después de que el diseño no fuera plausible luego de un daño considerable, se optó por migrar a un microcontrolador esp32s3 mini que de mejores resultados en relación en alcance e intensidad.

El giroscopio ha sido controlado por la misma placa de desarrollo que genera el AP, el ESP32-CAM es capaz de reconocer la cara de algunos de los integrantes del grupo.

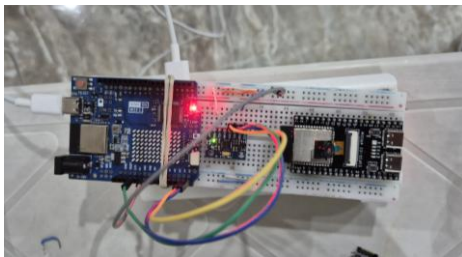


Ilustración 13. Nodo móvil



Ilustración 14. Prueba de reconocimiento facial

E. Protocolos utilizados

En el diseño e implementación de cada nodo del sistema, se emplearon distintos protocolos de comunicación digital que permiten la interacción eficiente entre los componentes. Uno de los principales utilizados es I2C (Inter-Integrated Circuit), empleado para la comunicación con la pantalla OLED. Este protocolo serial síncrono se caracteriza por su simplicidad y eficiencia, al requerir únicamente dos líneas para establecer la comunicación entre múltiples dispositivos:

- La línea SDA (Serial Data Line) se encarga del intercambio de datos bidireccional.
- La línea SCL (Serial Clock Line) proporciona la señal de reloj que sincroniza la transferencia de datos.

I2C resulta particularmente adecuado en este proyecto debido a la baja cantidad de pines disponibles en los microcontroladores utilizados, así como por el volumen relativamente reducido de datos que se intercambian con el display. Durante las pruebas, mediante la herramienta Logic, se monitorearon las líneas del protocolo, evidenciándose una transmisión continua de bits sobre SDA y una señal de reloj estable y regular sobre SCL, lo cual confirma el correcto funcionamiento de la comunicación I2C.



Ilustración 15. Monitoreo en la herramienta Logic

Por otro lado, el módulo NRF24L01 utiliza el protocolo SPI (Serial Peripheral Interface) para comunicarse con el microcontrolador. Este protocolo, también serial y síncrono, se eligió debido a su alta velocidad y fiabilidad, fundamentales para el manejo de paquetes de datos en tiempo real. En la implementación, se utilizaron las siguientes líneas:

- CE (Chip Enable), específica del NRF24L01, para alternar entre modos de transmisión y recepción.
- CSN (Chip Select Not), que habilita la comunicación con el esclavo cuando está en nivel bajo.
- SCK (Serial Clock), que genera los pulsos de reloj para sincronizar la transferencia.
- MOSI (Master Out Slave In), que transmite datos desde el maestro al esclavo.
- MISO (Master In Slave Out), que transmite datos desde el esclavo al maestro.

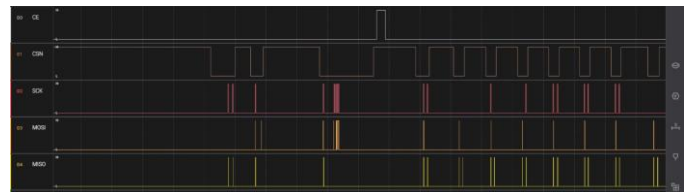


Ilustración 16. Protocolo SPI

El análisis de las señales SPI también se realizó mediante monitoreo digital. Se observó que la línea CE permanece en bajo la mayor parte del tiempo, con breves pulsos en alto, coherentes con las fases de escritura y lectura del módulo. La línea CSN mostró ciclos repetitivos de activación (nivel bajo) y desactivación (nivel alto), correspondientes a las diferentes transacciones SPI. Durante estos períodos, se detectó una señal de reloj limpia en SCK y tráfico activo en las líneas MOSI y MISO, confirmando una comunicación bidireccional exitosa entre el microcontrolador y el transceptor.

F. Pruebas

Se comenzaron las pruebas para determinar el alcance que el AP en campo abierto, para esto se utilizó un metro laser desde un punto fijo y se midió hasta qué punto la conexión entre el nodo y el AP era estable. Para esto se conectaron baterías portátiles para energizar los módulos.



Ilustración 17. Prueba de distancia

Se llegó a una distancia de 48.6 metros con un RSSI de -85dB antes de perder conexión, esto deja ver el alcance que tiene el AP utilizado

Los siguiente fue un ambiente mixto utilizando un decámetro para medir la magnitud del RSSI de cada uno de los nodos utilizando ambos APs para decidir el mejor para el uso de la práctica.

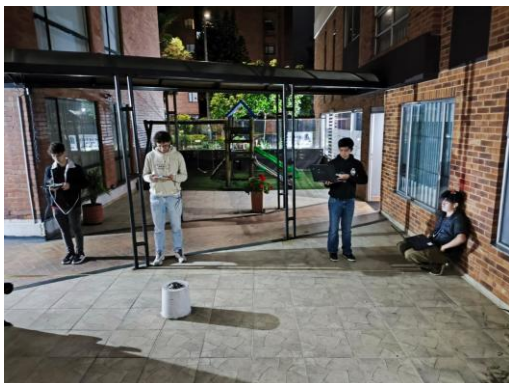


Ilustración 18. Prueba de distancia mixta

Distancia (m)	RSSI TX1 (dB)	RSSI TX2 (dB)	RSSI TX3 (dB)	RSSI TX4 (dB)
0	-40	-52	-52	-42
1	-57	-57	-62	-65
2	-68	-58	-73	-70
3	-70	-67	-68	-72
4	-74	-72	-72	-70
5	-76	-81	-79	-78
6	-70	-80	-83	-75
7	-75	-85	-75	-80
8	-82	-82	-82	-71
9	-76	-83	-89	-72
10	-77	-77	-75	-72
11	-80	-87	-80	-78
12	-74	-71	-70	-76
13	-61	-76	-67	-70
14	-66	-73	-75	-70
15	-69	-71	-70	-80
16	-68	-75	-68	-72
17	-66	-77	-66	-73
18	-75	-80	-74	-70
19	-75	-82	-84	-84
20	-81	-84	-82	-75
21	-76	-79	-77	-74
22	-76	-84	-85	-76
23	-80	-83	-88	-81
24	-80	-74	-77	-79
25	-73	-77	-78	-76
26	-76	-89	-89	-90
27	-84	-87	-83	-82
28	-84	-83	-83	-85
29	-71	-80	-77	-84
30	-90	-93	-95	-87
31	-85	-93	-94	-86
32	-91	-95	-94	-90
33	-95	-94	-100	-96
77	81.24242424	80.78787879	78.81818182	

Ilustración 19. Datos del Arduino r4

Con el Arduino r4 se obtuvieron valores variantes, pero constantes sin desconectarse ninguno de los nodos.

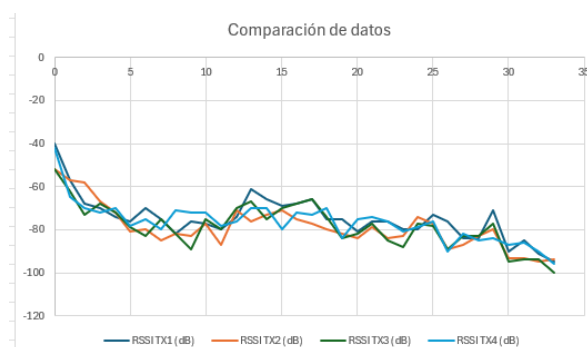


Ilustración 20. Comparación de los nodos

Se puede ver que el TX2, o la línea naranja tiene un movimiento más suave que indica una mejor recepción del AP, posiblemente por contar con modulo NRF mejor o por una mejor alineación frente al AP al hacer la medición.

Los nodos TX1 y TX4 presentan las peores estadísticas al tener alta tasa de fluctuación, es posible debido a una mala orientación de la bandera, un problema de alimentación o directamente un error en el software o hardware. Tx4 presentó problemas para conectarse al AP desde el inicio de la prueba, por lo esto puede ser una razón para su bajo rendimiento.

Ahora utilizando el AP compuesto por el ESP32 MPU6050 se encontraron datos muy diferentes.

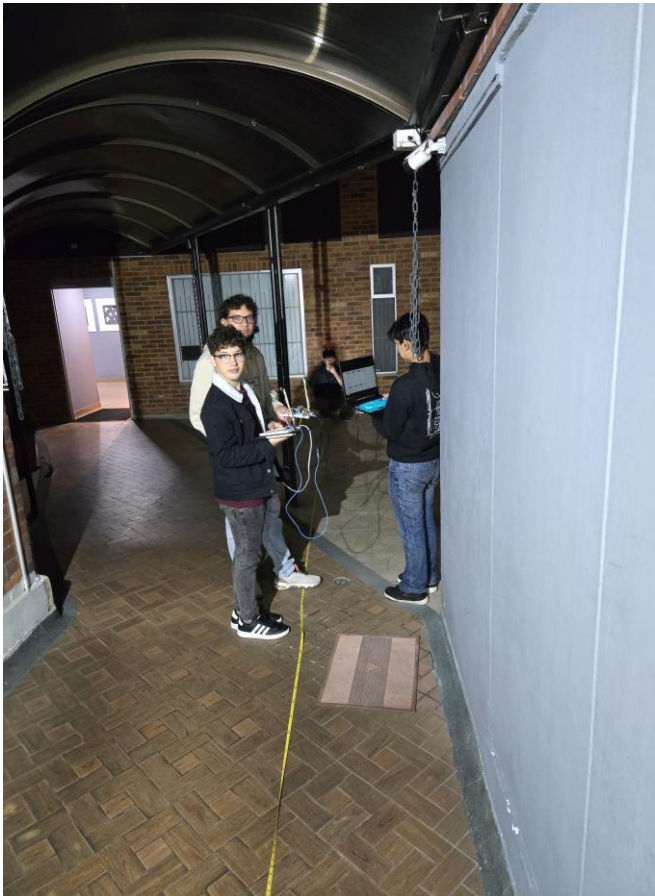


Ilustración 21. Segunda prueba

Distancia (m)	RSSI TX1 (dB)	RSSI TX2 (dB)	RSSI TX3 (dB)	RSSI TX4 (dB)
0	-29	-46	-44	-29
1	-55	-47	-55	-46
2	-57	-67	-57	-56
3	-69	-70	-67	-52
4	-69	-67	-67	-69
5	-72	-76	-76	-68
6	-76	-76	-76	-72
7	-74	-73	-76	-75
8	-76	-71	-100	-79
9	-82	-77	-100	-82
10	-80	-83	-100	-69
11	-74	-81	-100	-72
12	-82	-81	-100	-82
13	-80	-73	-100	-82
14	-70	-76	-100	-71
15	-75	-73	-100	-68
16	-86	-71	-100	-76
17	-80	-80	-100	-73
18	-74	-75	-100	-74
19	-86	-94	-100	-76
20	-87	-85	-100	-92
21	-96	-93	-100	-97
22	-100	-96	-100	-100
23	-100	-100	-100	-100
24	-100	-100	-100	-100
25	-100	-100	-100	-100
26	-100	-100	-100	-100
27	-100	-100	-100	-100
28	-100	-100	-100	-100
29	-100	-100	-100	-100
30	-100	-100	-100	-100
31	-100	-100	-100	-100
32	-100	-100	-100	-100
33	-100	-100	-100	-100
	85,727273	85,787879	94,484848	83,636364

Ilustración 22. Datos de la segunda prueba

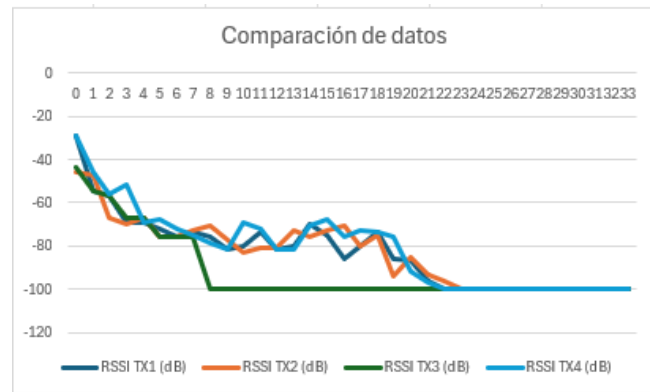


Ilustración 23. Comparación de los datos de la prueba

En esta prueba se vio como TX3, a pesar de tener un rendimiento promedio frente a los demás nodos fue el primero en perder la señal a los 8 metros, seguido del TX4, TX1 y TX1.

La desconexión repentina y a corto plazo de los nodos da a entender que el segundo AP no será la mejor opción para utilizar debido a que los nodos no pueden llegar muy lejos sin desconectarse. Además de que antes del inicio de la prueba TX3 y TX4 presentaron problemas para emparejarse correctamente al AP retrasando la prueba al tener que revisar sus códigos en busca de algún error en a la hora de conectarse.

Se hizo ahora la prueba de trilateración en un espacio de 4.9m X 15.4m distribuyendo los nodos en esquinas de un cuadrado con el nodo receptor en el centro y el AP dando vueltas alrededor.

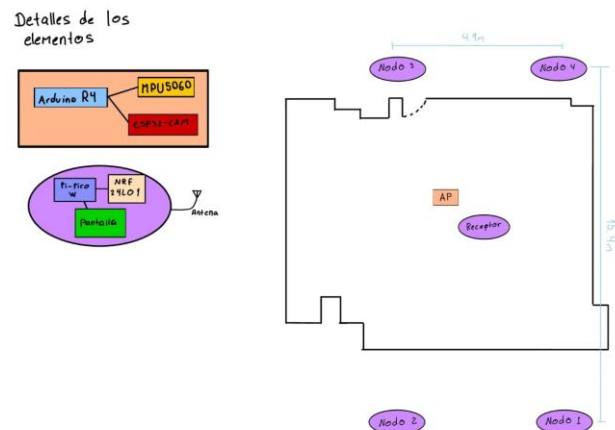


Ilustración 24. Plano de las pruebas

El código que se hizo para registrar la trilateración al mover el AP por el lugar muestra la localización en tiempo real, registro de la tasa de recepción y las distancias que separan los nodos del AP.

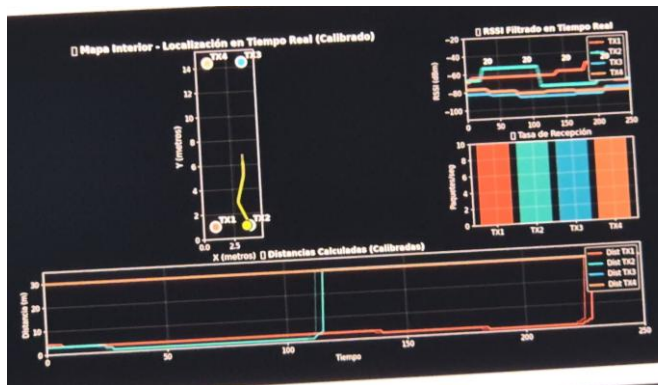


Ilustración 25. Interfaz de trilateración

El punto amarillo sigue el movimiento del AP dentro del perímetro de prueba dejando su trayectoria anterior marcada, muestra la forma en que el AP se acerca o aleja de los nodos, el mayor problema de la interfaz resulta ser que el punto amarillo no se mueve en tiempo real, lo que causa que sea necesario esperar unos instantes a que comience a moverse como lo hace el AP verdadero, el cual debe estar en movimiento continuo para evitar que el punto se mueva en cualquier dirección.

También se hizo un servidor web con el cual se puedan registrar datos del nodo móvil como el giroscopio, la orientación, su aceleración, además de los clientes (nodos) conectados actualmente.

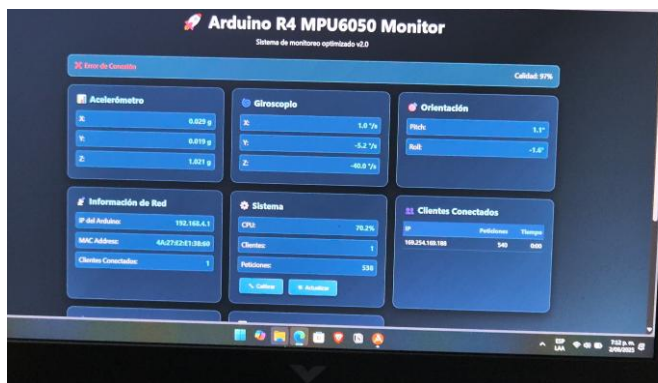


Ilustración 26. Servidor web

G. Trilateración en el proyecto

El espacio decidido fue el interior del edificio de la universidad, se tomó el pasillo y 4 salones. Se tomaron las medidas y se repitieron las pruebas, ahora utilizando un carrito a control remoto para lograr el movimiento del AP a lo largo del escenario.

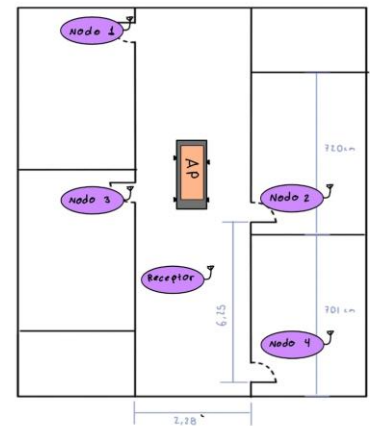
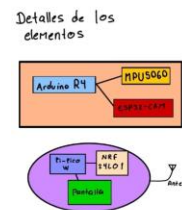


Ilustración 27. Plano del escenario

Al interactuar en entornos cerrados del establecimiento universitario, se pudo evidenciar que los obstáculos como son las paredes de amplio grosor, afectan en gran medida la propagación de las señales, además, las líneas de corriente distribuida por el establecimiento, que generan campos magnéticos, podrían ser una causa adicional de los problemas de conexión.

En resumen, la trilateración es el mecanismo que articula todos los elementos del sistema: los datos RSSI obtenidos por los transmisores fijos, la comunicación inalámbrica establecida por los módulos NRF24L01, y el procesamiento centralizado en Python. Esta integración nos permitió construir un sistema funcional y modular de localización indoor, capaz de seguir el desplazamiento del nodo móvil con una resolución suficiente para su análisis y evaluación en un entorno educativo y práctico.

H. Video

<https://youtu.be/0HrP9MCD90s>

I. Repositorio

<https://github.com/SantiagoQ130/Proyecto-final-comunicaciones-digitales.git>

J. Conclusiones

- El sistema de localización en interiores basado en RSSI y trilateración demostró ser funcional y replicable, permitiendo estimar la posición del nodo móvil con una resolución adecuada para aplicaciones educativas y prácticas.
- La calidad de recepción de cada transmisor depende no solo de la distancia al nodo móvil, sino también de factores físicos como la orientación de las antenas, la interferencia ambiental y la estabilidad del hardware, lo cual se evidenció en el comportamiento desigual de los nodos durante las pruebas.
- La latencia observada en el sistema, tanto en la transmisión de datos desde los transmisores como en su procesamiento en el nodo receptor, afectó la precisión del posicionamiento en tiempo real. Este desfase provocó que la ubicación estimada del nodo

móvil no siempre coincidiera con su posición actual, especialmente durante desplazamientos rápidos, lo que resalta la necesidad de optimizar la frecuencia de actualización y la eficiencia del protocolo de comunicación.

REFERENCIAS

- [1] *ç Bahl, P., & Padmanabhan, V. N. (2000). RADAR: An in-building RF-based user location and tracking system. Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), 2, 775–784. <https://doi.org/10.1109/INFCOM.2000.832252>*
- [2] *Liu, H., Darabi, H., Banerjee, P., & Liu, J. (2007). Survey of wireless indoor positioning techniques and systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 37(6), 1067–1080. <https://doi.org/10.1109/TSMCC.2007.905750>*
- [3] *Faragher, R., & Harle, R. (2015). Location fingerprinting with Bluetooth Low Energy beacons. IEEE Journal on Selected Areas in Communications, 33(11), 2418–2428. <https://doi.org/10.1109/JSAC.2015.2430281>*
- [4] *Zafari, F., Gkelias, A., & Leung, K. K. (2019). A survey of indoor localization systems and technologies. IEEE Communications Surveys & Tutorials, 21(3), 2568–2599. <https://doi.org/10.1109/COMST.2019.2911558>*
- [5] *Rugeles, J. (2024). Simulador de localización en interiores basado en RSSI [Repositorio de código]. GitHub. <https://github.com/jrugeles/RSSI>*
- [6] *Trilateration vs. Triangulation for Indoor Positioning Systems | IoT For All. (n.d.). IoTfor All. <https://www.iotforall.com/trilateration-vs-triangulation-indoor-positioning-s>*
- [7] *Tinker Tech Trove. (2021, April 17). Raspberry Pi Pico: nRF24L01 over SPI [Video]. YouTube. <https://www.youtube.com/watch?v=aP8rSN-1eT0>*