



UNIVERSIDAD NACIONAL
DE LA
PATAGONIA SAN JUAN BOSCO
FACULTAD DE INGENIERÍA

TESINA FINAL DE LA CARRERA:
LICENCIATURA EN INFORMÁTICA

*“Simulador de técnicas de modulación y
codificación aplicados a entornos interiores”*

David Omar Olivera

Lic. Pablo Rosales
Dr. Ing Santiago Murano

COMODORO RIVADAVIA
ARGENTINA
Octubre 2024

PÁGINA PARA LOS EVALUADORES

Calificación:

Comentarios:

.....

.....

.....

.....

Lugar para la Fecha de la Evaluación

DEDICATORIAS

A mis padres y hermano, hermosas personas a las cuales admiro y agradezco que me hayan tocado como familia, quienes siempre me contuvieron, acompañaron e incentivaron en todos los aspectos de la vida. Gracias por el ejemplo de ser padres.

A Camila, mi amor, y a Amelia y Valentino, la hermosa familia que formamos, por su compañía, paciencia y contención en el día a día y durante todo el proceso de elaboración de este trabajo.

PREFACIO

La presente tesina forma parte de los requisitos finales para optar al grado académico *Licenciatura en Informática*, de la Universidad Nacional de la Patagonia San Juan Bosco y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otras. La misma es el resultado de un proceso de desarrollo llevado a cabo en la Facultad de Ingeniería, en el período comprendido entre Mayo de 2023 y Octubre de 2024, bajo la dirección de *Lic. Pablo Rosales* y *Dr. Ing. Santiago Murano*.

David Omar Olivera
FACULTAD DE INGENIERÍA
UNIVERSIDAD NACIONAL DE LA PATAGONIA SAN JUAN BOSCO
Comodoro Rivadavia, 2024.

AGRADECIMIENTOS

A mis padres por haberme acompañado siempre, respaldándome y brindándome la fuerza y el amor necesarios para superar adversidades y desarrollarme en todos los aspectos de la vida.

A mi nueva familia, Camila, Amelia y Valentino por bancarme y apoyarme, por la paciencia, por el tiempo que les robe para poder terminar este trabajo.

A mis tutores, Pablo Rosales y Santiago Murano, por todo el apoyo que me han brindado, por su guía valiosa y comprometida, y por su paciencia en los momentos difíciles.

RESUMEN

El incremento en el uso de sistemas de servicios basados en localización que necesitan de información precisa sobre los requerimientos de localización del objetivo, han impulsado la investigación en Sistemas de Posicionamiento en Interiores (IPSs), en diferentes aplicaciones.

La necesidad de mejorar el desempeño y reducir los costos de implementación de estos sistemas motivan el uso de herramientas de simulación que permitan evaluar el comportamiento de los sistemas frente a diversos efectos, el comportamiento de las señales utilizadas y cómo afectan las interferencias y efectos propios del entorno al resultado final de la posición.

En trabajos previos, llevados a cabo por el Grupo de Ingeniería Electrónica Aplicada a Espacios Inteligentes y Transporte (GEINTRA) de la Universidad de Alcalá (UAH) y Grupo de Investigación en Procesamiento de la Información y Sensores (GIPIS) de la Universidad Nacional de la Patagonia San Juan Bosco (UNPSJB), se han desarrollado herramientas de simulación que permiten estudiar estos fenómenos. Esta herramienta está desarrollada en el entorno App-Designer de MATLAB, sin aplicar buenas prácticas de desarrollo, lo cual motiva un proceso de reingeniería y refactorización de código para lograr un sistema escalable, mantenible y accesible a través de la web.

La tesina es presentada para cumplir con los requisitos finales para la obtención del título de Licenciado en Informática de la Universidad Nacional de la Patagonia San Juan Bosco. El objetivo es desarrollar una aplicación web moderna de calidad, basada en desarrollos previos, que pueda ser escalable, mantenible, accesible a través de la web.

La aplicación es una herramienta de simulación que permite diseñar las señales emitidas por los transductores, configurar la emisión, y ajustar los diversos parámetros que caracterizan el entorno, así como seleccionar y posicionar el hidrófono receptor. Mostrando los resultados en varias etapas del procesamiento: desde la señal en banda base, la señal modulada, la señal recibida, hasta la estimación de la posición final para el receptor.

La nueva aplicación web intenta mejorar la performance y usabilidad de las herramientas generadas en otras investigaciones, y agrega nuevas funcionalidades como la gestión de usuario y proyectos. Permite a los investigadores y profesionales del área la colaboración y el intercambio de conocimientos entre distintos usuarios y organizaciones.



CAPÍTULO 1:

MOTIVACIÓN DEL PROYECTO

Introducción

En la actualidad, la localización y los sistemas de posicionamiento precisos representan un desafío significativo en múltiples campos como la logística, la robótica, la salud y la seguridad. A diferencia de las aplicaciones en entornos exteriores, donde tecnologías como el GNSS (Global Navigation Satellite Systems) proporcionan una localización efectiva, en interiores la falta de una señal satelital confiable impone la necesidad de sistemas alternativos. Entre las soluciones disponibles, los sistemas de posicionamiento acústico han ganado relevancia debido a su capacidad para proporcionar alta precisión en la localización de dispositivos y personas dentro de espacios cerrados.

El uso de señales ultrasónicas para el posicionamiento en interiores ofrece varias ventajas, incluyendo una menor interferencia electromagnética y una mayor precisión en la determinación de distancias y ubicaciones. Sin embargo, el diseño e implementación de estos sistemas requiere un análisis detallado de múltiples factores como la propagación de las ondas acústicas, la interferencia de obstáculos, la gestión de múltiples emisores y receptores, y la eficiencia en el procesamiento de datos.

Para desarrollar sistemas de posicionamiento acústico eficientes, es de gran ayuda contar con herramientas de simulación que permitan modelar y analizar diferentes escenarios y configuraciones antes de la implementación práctica. Las simulaciones facilitan la identificación de posibles problemas, la optimización de parámetros y la validación de algoritmos de localización sin incurrir en los elevados costos y dificultades asociados con las pruebas en entornos reales.

Tradicionalmente, muchas de estas simulaciones se han realizado utilizando plataformas como MATLAB, que, aunque muy potentes, presentan limitaciones en términos de accesibilidad, escalabilidad y facilidad de uso para usuarios no especializados en programación. Además, las aplicaciones basadas en MATLAB suelen carecer de una interfaz de usuario amigable y requieren conocimientos técnicos avanzados para su configuración y ejecución. Otro factor importante es que MATLAB requiere una licencia de pago, lo que puede ser una barrera para su adopción en entornos educativos o de investigación con presupuestos limitados. En contraste, Python es un lenguaje de código abierto, lo que permite a los desarrolladores utilizarlo sin costo alguno, fomentando una mayor accesibilidad y colaboración dentro de la comunidad científica y académica.

En los grupos de investigación GEINTRA de la UAH y GIPIS de la UNPSJB se ha desarrollado una herramienta con estas características, destinada al análisis y simulación de procesamiento digital de señales [1].

Motivan la realización de esta tesina, mejorar el sistema actual desarrollado en MATLAB como aplicación de escritorio, migrando a un sistema web que permita la escalabilidad de funcionalidades y su disponibilidad a usuarios finales de manera simultánea. Esto permitirá la protección intelectual del código y de los modelos matemáticos desarrollados por los grupos de investigación.

Por lo anterior, con el desarrollo de este proyecto se espera mejorar la accesibilidad a simulaciones para distintos usuarios habilitados para utilizar la herramienta, poder



realizar una herramienta que facilite la extensión de su funcionalidad y que sea intuitiva, aprovechando los beneficios de las aplicaciones web ampliamente utilizadas en el día a día. Además, se busca permitir a los grupos de investigación de GEINTRA y GIPIS administrar los usuarios que podrán utilizar la herramienta, así como la posibilidad de guardar distintos proyectos y compartirlos con otros usuarios dentro de la misma plataforma.

Objetivo

Objetivo General

El objetivo principal de esta tesina es comprender técnicas de codificación y modulación de señales acústicas codificadas, aplicados a sistemas de posicionamiento ultrasónico en interiores y desarrollar una herramienta de simulación web. Dicha herramienta, permitirá a los usuarios configurar y ejecutar dichas simulaciones de manera intuitiva y eficiente, proporcionando visualizaciones interactivas de los resultados y facilitando la comparación entre diferentes configuraciones y algoritmos.

Objetivos Específicos

Para el cumplimiento del objetivo general propuesto, se hará uso de los sistemas desarrollados previamente, y se plantean los siguientes objetivos específicos para este trabajo:

- Comprender las propiedades de técnicas de codificación, tales como:
 - Secuencias Kasami
 - Secuencias Loosely Synchronous (LS)
 - Secuencias Zadoff-Chu (ZC)

- Conocer las propiedades de los efectos típicos que afectan la propagación de señales ultrasónicas, entre ellos el multitrayecto (multipath), el ruido, la temperatura, la absorción del aire y el efecto Doppler relacionado con el movimiento relativo del emisor y el receptor.

- Analizar y comprender el funcionamiento de herramientas desarrolladas en los grupos de investigación GEINTRA de la UAH y GIPIS de la UNPSJB.

- Adquirir competencias vinculadas a la metodología de investigación en proyectos científicos en temáticas de procesamiento digital de señales.

- Desarrollar un software para simular el desempeño de las técnicas analizadas previamente en un sistema de posicionamiento en entornos interiores.
 - Desarrollar una interfaz de usuario interactiva utilizando Angular que permita la configuración personalizada de emisores y receptores ultrasónicos, la selección de esquemas de codificación y modulación, y la definición de parámetros ambientales.



- Implementar el backend de la aplicación con Flask, proporcionando APIs RESTful que gestionen las solicitudes del frontend, manejen la lógica de negocio y coordinen las tareas asíncronas mediante Celery.
- Integrar una base de datos PostgreSQL para almacenar configuraciones de simulación, resultados y datos de usuarios, utilizando SQLAlchemy como ORM para facilitar las operaciones de base de datos.
- Proporcionar visualizaciones avanzadas de los resultados de las simulaciones utilizando Plotly, una librería que permite la generación de gráficos interactivos para facilitar el análisis de los datos obtenidos.

Organización del Trabajo

Los capítulos siguientes, organizan el desarrollo de esta tesina de la siguiente manera:

Capítulo 2: Estado del arte

Se revisan las tecnologías y métodos previos utilizados en la simulación de sistemas acústicos para posicionamiento en interiores, destacando las limitaciones de las soluciones existentes y justificando la necesidad de una herramienta moderna y accesible.

Capítulo 3: Estudio y Análisis

Aquí se realizará una investigación de las técnicas de codificación, y todos los procesos que involucra cada etapa de la simulación de estos sistemas de posicionamiento. Organizando entre las siguientes etapas principales, emisión, recepción, respuesta del transductor, acceso al medio, efectos en la señal y resultados; especificando en cada caso los parámetros comprometidos.

Capítulo 4: Dominio de la Aplicación y Enfoque de Desarrollo

El capítulo aborda el dominio y enfoque de desarrollo de la aplicación web para cumplir con los requerimientos. Se menciona cómo los usuarios pueden configurar las simulaciones y cómo éstos interactúan con el sistema. También se describen las funcionalidades clave que debe tener la aplicación.

Capítulo 5: Diseño, Desarrollo e Implementación

En este capítulo se describe la aplicación desarrollada de manera abstracta presentando un diagrama de clases conceptual y un diagrama de arquitectura de la plataforma para transmitir una visión general de cómo está implementada. Luego se adjuntan imágenes concretas de la interfaz de usuario y se describe cómo es el flujo de la simulación, para un caso base. Se comenta además sobre la implementación en el entorno productivo.

Capítulo 6: Conclusión y Trabajo Futuro

Presenta la conclusión de la tesina y del desarrollo, mencionando posibles trabajos futuros.



CAPÍTULO 2:

ESTADO DEL ARTE

Se pueden encontrar varias investigaciones y desarrollos de plataformas de simulación enfocadas en aplicaciones de posicionamiento por medios acústicos.

Por ejemplo, en [2] se desarrolla una plataforma que permite analizar la propagación de señales acústicas bajo distintas condiciones, en este caso en entornos subacuáticos mostrando resultados relacionados con las señales moduladas recibidas. De manera similar, en [3] se presenta una herramienta desarrollada en LabVIEW que, a través de una interfaz web, ofrece visualizaciones sobre la propagación del sonido en entornos subacuáticos, incluyendo niveles de ruido ambiental y la atenuación que sufren las señales acústicas en este medio.

Además de estas plataformas, hay estudios que, aunque no ofrecen una simulación completa, aportan elementos clave. El trabajo presentado en [4] se centra en la simulación de diferentes esquemas de codificación y su impacto en la precisión de las mediciones. Asimismo, los estudios en [5] y [9] exploran diversos algoritmos de localización, esenciales para la estimación de posiciones en entornos submarinos.

En cuanto al modelado de la propagación de señales acústicas, destacan los trabajos de [7] y [8], donde se investigan distintos enfoques basados en el modelo de trazado de rayos BELLHOP. Adicionalmente, en [9] se propone un modelo de propagación implementado en MATLAB que considera fenómenos como la divergencia geométrica, las pérdidas por absorción y reflexiones, así como el ensanchamiento Doppler causado por las olas generadas por el viento en la superficie marina.

La Tabla 1, extraída de [1] sintetiza las principales características de los trabajos mencionados, especificando el software utilizado, el año de publicación, la disponibilidad de una interfaz de usuario (IU), las características clave abordadas, los parámetros de entrada y los resultados obtenidos.

Ref.	Año	Software	IU	Características	Parámetros	Resultados
[5]	2006	MATLAB	No	Algoritmos de localización para redes de sensores subacuáticos	Rango , distribuido o centralizado, número y posición de emisores	Precisión
[6]	2009	MATLAB	No	Localización 3D en redes de sensores subacuáticos	Posición de los nodos y rango	Eficiencia en la localización y robustez a errores
[4]	2010	MATLAB	No	Eficiencia en la localización de los esquemas de codificación	Ancho de banda, retardo de la propagación y atenuación	Error, desviación media, traza individual
[10]	2012	-	No	Modelo de trazado de rayos BELLHOP	Velocidad del sonido	Error en la distancia emisor-receptor, ToA
[2]	2014	MS Visual Basic	Si	Comunicación en red de sensores y patrón de radiación	Profundidad fondo marino, posición emisores, pérdidas de transmisión	Propiedades señales acústicas en el agua
[9]	2015	MATLAB	No	Modelo de propagación	Divergencia geométrica, perfil de velocidad, batimetría y velocidad del viento	Llegadas al receptor, pérdidas de transmisión, y ToF de las llegadas



[8]	2016	Python, MATLAB y FORTRAN	No	Modelo de trazado de rayos BELLHOP	Perfil de velocidad del sonido	ToA
[3]	2020	LabView	Si	Simulación comunicación subacuática.	Temperatura, salinidad, velocidad del sonido	Atenuación de las señales recibidas, ruido ambiente

Tabla 1 - Resumen del estado del Arte, extraído de [1]

Finalmente, en [1] se presenta una plataforma completa de simulación que, a diferencia de las anteriores, no se centra sólo en aspectos específicos, sino que propone una herramienta de simulación integral para sistemas de posicionamiento ultrasónicos, la cual es configurable, cuenta con una interfaz gráfica y aborda todos los pasos involucrados en estas aplicaciones. Además, se ha orientado principalmente a entornos acústicos subacuáticos abordando los principales desafíos que enfrentan estos sistemas, como la propagación de las señales acústicas, las técnicas de codificación y modulación para la transmisión de señales o los efectos del medio.

El trabajo [1] posee gran relevancia para el desarrollo de esta tesina, ya que proporciona una base sólida para el diseño y la evaluación de sistemas de posicionamiento acústico en entornos complejos. Además, presenta una implementación de una plataforma que abarca desde la configuración de emisores y receptores hasta la simulación de la propagación de las señales y la estimación de la posición. Este enfoque integral permite obtener una comprensión profunda de los desafíos y limitaciones que enfrentan estos sistemas.

Como se puede observar las soluciones previas han utilizado principalmente plataformas como MATLAB y Simulink para modelar y analizar sus soluciones.

MATLAB es una herramienta ampliamente utilizada en el ámbito académico y profesional para la simulación y análisis de sistemas acústicos. Su robusta biblioteca de funciones matemáticas y de procesamiento de señales la convierte en una opción atractiva para investigadores y desarrolladores. Sin embargo, presenta varias limitaciones:

- **Accesibilidad y Usabilidad:** Requiere licencias costosas y un conocimiento avanzado de programación, lo que puede ser una barrera para usuarios no especializados.
- **Escalabilidad:** La ejecución de simulaciones complejas puede ser lenta y consumir grandes cantidades de recursos, especialmente en entornos con múltiples emisores y receptores.
- **Interfaz de Usuario:** Las interfaces gráficas desarrolladas en MATLAB son menos intuitivas y requieren una personalización significativa para ser utilizadas por usuarios finales sin experiencia técnica.
- **Administración de Usuarios:** No poseen una gestión de usuarios para poder autorizar o restringir su uso.
- **Propiedad intelectual:** Muchas veces compartir la utilización de la aplicación significa compartir su código fuente.

El presente trabajo aborda el desarrollo de una plataforma de simulación de sistemas de posicionamiento acústico utilizando un enfoque distinto, que busca superar las limitaciones mencionadas anteriormente al emplear tecnologías abiertas y accesibles, con una interfaz amigable, ecosistema web y manteniendo la confiabilidad y precisión necesarias para investigaciones científicas y aplicaciones prácticas. En particular, se destaca el uso de Python como el lenguaje base para el desarrollo de la herramienta en el backend, siendo en la actualidad uno de los lenguajes más utilizados en la comunidad científica. Bibliotecas como NumPy y SciPy ofrecen funciones equivalentes y, en algunos



casos, superiores a las de MATLAB, permitiendo el desarrollo de algoritmos de simulación robustos y eficientes.

Además, se divide la lógica de la aplicación de la interfaz gráfica, facilitando la separación de responsabilidades y permitiendo una mayor modularidad en el código. Para la interfaz, se optó por tecnologías web como HTML, CSS y JavaScript, en combinación con Angular [17], un framework moderno que no solo mejora la experiencia del usuario, sino que también facilita la portabilidad y accesibilidad desde distintos dispositivos. De este modo, los usuarios pueden acceder a la plataforma desde cualquier navegador sin necesidad de instalaciones adicionales, aumentando su usabilidad.

El uso de un enfoque de arquitectura modular permite también la integración de nuevos módulos o funciones con relativa facilidad, lo que abre la puerta a futuras expansiones o personalizaciones sin comprometer la estabilidad del sistema central. Asimismo, la implementación de un ecosistema web basado en tecnologías abiertas reduce los costos de licencias y las barreras de entrada para la adopción del sistema por parte de instituciones educativas y de investigación, democratizando el acceso a herramientas de simulación avanzadas.

A continuación, se presenta un cuadro comparativo entre Matlab y Python extendiendo este último con las librerías mencionadas tan importantes para el cálculo científico como son Numpy [19], SciPy [20], y Plotly [16] que como se verá más adelante, se utilizó para la generación de diferentes tipos de gráficos.

Aspecto	MATLAB	Python(Numpy, SciPy, Plotly)
Costos y Accesibilidad	- Licencias Costosas: Requiere adquisición de licencias, lo que puede ser una barrera para usuarios con presupuestos limitados, especialmente estudiantes o pequeñas empresas.	- Software Libre y Abierto: Gratuito y de código abierto, eliminando barreras económicas y favoreciendo la adopción en diversos entornos.
	- Entorno Estable y Probado: Herramienta ampliamente utilizada y reconocida en el ámbito académico y profesional, con décadas de desarrollo y optimización.	- Comunidad Abierta: Al ser de código abierto, la calidad del software depende en gran medida del apoyo de la comunidad y puede variar entre versiones o bibliotecas.
Flexibilidad y Escalabilidad	- Entorno Integrado: Ofrece un entorno completo con herramientas integradas para el desarrollo rápido de prototipos, simulación y visualización, facilitando el flujo de trabajo.	- Ecosistema Extensible: Python ofrece un ecosistema creciente con una amplia variedad de bibliotecas y frameworks, como Celery y Flask, permitiendo la integración con tecnologías modernas.
	- Limitaciones en Escalabilidad: Aunque eficiente para muchas aplicaciones, las simulaciones muy complejas pueden requerir recursos significativos, lo cual puede limitar su escalabilidad en algunos casos.	- Alta Escalabilidad: Herramientas como Celery permiten gestionar tareas asíncronas, y frameworks como Flask y Angular facilitan el desarrollo de aplicaciones web escalables.
	- Optimización Integrada: Las funciones matemáticas y de procesamiento en MATLAB están altamente optimizadas, ofreciendo un rendimiento excelente para operaciones numéricas.	- Optimización con Librerías: Bibliotecas como NumPy y SciPy también están optimizadas para operaciones numéricas, aprovechando implementaciones en C y Fortran para mejorar el rendimiento. Comparables a MATLAB.
Comunidad y Soporte	- Comunidad Establecida: MATLAB cuenta con una comunidad académica y profesional consolidada, así como con recursos extensos y un soporte técnico estructurado.	- Comunidad Activa: Python posee una de las comunidades más grandes y activas, que contribuye a su crecimiento y a la mejora continua de sus librerías científicas.
	- Soporte Profesional: Soporte oficial y profesional disponible a través de MathWorks, ideal para empresas y	- Soporte Comunitario: Principalmente basado en la comunidad y foros, aunque



	proyectos críticos que requieren asistencia técnica garantizada.	existen empresas que brindan servicios de soporte comercial para Python.
Integración y Compatibilidad	- Integración Limitada con Tecnologías Web: Aunque MATLAB permite integrarse con otros lenguajes, su flexibilidad para desarrollar aplicaciones web es limitada comparada con Python.	- Integración Flexible: Python se integra fácilmente con frameworks modernos como Flask y Angular, permitiendo el desarrollo de aplicaciones web de manera sencilla.
	- Compatibilidad con Otros Lenguajes: MATLAB puede interactuar con lenguajes como C y Python, pero con una curva de integración más pronunciada.	- Compatibilidad Multiplataforma: Python funciona en diversas plataformas y sistemas operativos, lo cual facilita la implementación en diferentes entornos.
Visualización y Análisis de Datos	- Visualización Potente e Integrada: MATLAB incluye herramientas avanzadas de visualización que son muy efectivas para crear gráficos y visualizaciones científicas con facilidad.	- Visualizaciones Interactivas y Dinámicas: Plotly ofrece capacidades para generar gráficos interactivos, mejorando la experiencia de usuario para la visualización de datos en tiempo real.
	- Facilidad de Uso: MATLAB permite la creación de gráficos de alta calidad de manera rápida y sencilla, sin necesidad de agregar librerías externas.	- Mayor Flexibilidad: Python permite mayor personalización y la creación de gráficos dinámicos gracias a su integración con frameworks web.
Propiedad Intelectual y Distribución	- Código Cerrado: MATLAB tiene un entorno cerrado, lo cual puede limitar la distribución del software y dificultar la compartición del código de forma abierta.	- Código Abierto: Python es de código abierto, lo cual facilita la compartición y la colaboración.

Tabla 2 - Tabla comparativa MATLAB vs Python bajo ciertos aspectos



CAPÍTULO 3:

ESTUDIO Y ANÁLISIS:

Métodos de codificación

La transferencia de información en sistemas de comunicación se basa en la emisión de señales ultrasónicas que representan la información a transmitir. Sin embargo, al enviar estas secuencias de manera directa en entornos reales, se deben tener en cuenta adversidades que pueden generarse por factores como el ruido, las interferencias y las distorsiones causadas por el medio de propagación. Estas adversidades pueden degradar la señal y provocar errores en la recepción, afectando la fiabilidad y precisión de la comunicación.

Para superar estas limitaciones, es esencial aplicar **esquemas de codificación y técnicas de modulación** que permitan una transmisión más robusta y eficiente. La codificación de canal introduce redundancia y estructura en la secuencia de datos mediante códigos específicos, lo que mejora la detección y corrección de errores en la recepción. Por otro lado, la modulación adapta la señal a las características del medio de transmisión, transformando los datos en formas de onda que pueden propagarse eficazmente a través del canal.

Una técnica ampliamente utilizada que combina codificación y modulación es el ensanchamiento de espectro. Este método consiste en aumentar el ancho de banda de la señal transmitida al máximo soportado por el emisor. El ensanchamiento se logra mediante la modulación de la señal con secuencias de código especiales proporcionadas por los esquemas de codificación, como las secuencias de Kasami [10], los códigos con zonas de correlación cero como los Loosely Synchronous [11] y los códigos polifásicos Zadoff-Chu [12].

El ensanchamiento de espectro ofrece múltiples beneficios:

- **Resistencia al ruido y a las interferencias:** Al dispersar la energía de la señal en un espectro más amplio, se reduce la susceptibilidad a interferencias de banda estrecha y al ruido ambiental.
- **Seguridad y privacidad:** Las señales ensanchadas son más difíciles de interceptar o detectar por receptores no autorizados, ya que se asemejan al ruido si no se conoce la secuencia de código utilizada.
- **Acceso múltiple por división de código (CDMA):** Permite que varios usuarios compartan el mismo canal de frecuencia simultáneamente, utilizando diferentes secuencias de código que minimizan la interferencia entre ellos.
- **Mitigación del efecto multicamino:** En medios como el agua, donde las señales pueden reflejarse y llegar al receptor por múltiples rutas, el ensanchamiento ayuda a distinguir la señal directa de las reflejadas, mejorando la precisión en la recepción.

Secuencias de Kasami

El primer esquema de codificación que se presenta son las **secuencias de Kasami** [10], las cuales son secuencias binarias que, aunque se generan de manera determinista (pseudoaleatoria), exhiben comportamientos similares a los de una secuencia



completamente aleatoria. Estas secuencias son especialmente útiles en comunicaciones y procesamiento de señales debido a sus propiedades de correlación favorables, lo que las hace ideales para minimizar interferencias y mejorar la detección en presencia de ruido.

Para generar las secuencias de Kasami, se sigue un proceso estructurado que involucra los siguientes pasos:

1- Creación de una secuencia primaria (m_1):

- Para generar m_1 , se utiliza un **Registro de Desplazamiento de Retroalimentación Lineal (LFSR)**, por sus siglas en inglés). Un LFSR es un dispositivo o algoritmo que produce una serie de bits en función de sus estados anteriores, siguiendo una lógica definida. Es ampliamente utilizado en criptografía y comunicaciones para generar secuencias pseudoaleatorias.
- m_1 es una **m-secuencia**, es decir, una secuencia binaria de máxima longitud. Esto significa que recorre todos los estados posibles de una secuencia binaria de cierta longitud, excepto el estado de todos ceros.

2- Diezmado de la secuencia primaria:

- La secuencia m_1 se **diezma**, lo que significa que se selecciona un subconjunto de sus elementos tomando cada cierto número de bits. Por ejemplo, si el factor de diezmado es 5, se tomaría cada quinto bit de m_1 .
- Este proceso produce una nueva secuencia más corta llamada m_2 , que conserva ciertas propiedades de m_1 pero tiene un periodo diferente.

3- Construcción de la secuencia diezmada repetida (m_2):

- La secuencia diezmada m_2 se repite o concatena varias veces hasta alcanzar la misma longitud que la secuencia m_1 . Esto permite que ambas secuencias tengan la misma longitud y puedan combinarse de manera efectiva.

4- Formación del conjunto de secuencias de Kasami:

- El conjunto completo de secuencias de Kasami se forma combinando la secuencia primaria m_1 con versiones desplazadas de la secuencia m_2 . Esta combinación se realiza mediante la **operación XOR** (suma módulo-2) entre los bits correspondientes de m_1 y m_2 o sus desplazamientos.
- Al sumar m_1 con m_2 y con sus versiones desplazadas, se obtiene un conjunto de secuencias distintas que comparten propiedades específicas de correlación. Este conjunto incluye a m_1 y varias combinaciones resultantes de aplicar desplazamientos a m_2 antes de realizar la operación XOR con m_1 .

Las secuencias de Kasami son valoradas por sus excelentes propiedades de **autocorrelación** y **correlación cruzada**, lo que las hace ideales para aplicaciones donde se requiere minimizar la interferencia entre señales y mejorar la detección en presencia de ruido. En particular:

- **Autocorrelación (AC):** La autocorrelación de una secuencia de Kasami presenta un pico pronunciado en el desfase cero y valores bajos en otros desfases, facilitando la sincronización y detección precisa de la señal.
- **Correlación Cruzada (CC):** La correlación cruzada entre diferentes secuencias del conjunto es baja, lo que reduce la interferencia entre múltiples transmisiones en sistemas de acceso múltiple.

Las secuencias de Kasami, como las **secuencias Zadoff-Chu**, muestran mejores propiedades de correlación en emisiones periódicas en comparación con emisiones aperiódicas. Sin embargo, en sistemas de localización y posicionamiento, es común



espaciar las emisiones introduciendo intervalos de guarda. Esto se hace para reducir los efectos negativos del **multicamino**, donde las señales reflejadas pueden interferir con la señal directa, degradando la precisión del sistema.

La herramienta que se desarrollará permite configurar y comparar ambas modalidades de emisión:

- **Emisión Periódica:** Las señales se transmiten de manera continua y repetitiva, aprovechando las propiedades óptimas de correlación en este modo.
- **Emisión Aperiódica:** Se introducen intervalos de guarda entre emisiones para mitigar el efecto del multicamino, lo cual es más representativo de las condiciones reales en sistemas de localización.

Secuencias Loosely Synchronous (LS)

El siguiente esquema de codificación a considerar son los códigos **Loosely Synchronous (LS)** [13]. Estos códigos son una forma de códigos ortogonales generalizados, lo que significa que en sus funciones de correlación poseen una **Zona de Correlación Cero (Zero Correlation Zone, ZCZ)** alrededor del pico principal. Esta característica permite que las interferencias se reduzcan significativamente dentro de esa zona, lo que es especialmente útil en sistemas de comunicación donde las señales pueden sufrir retardos variables o efectos de multicamino.

Estos ofrecen un equilibrio entre complejidad y rendimiento, siendo una opción eficaz para sistemas que requieren alta precisión y resistencia a interferencias. Su capacidad para mantener bajas correlaciones cruzadas dentro de una zona específica los hace ideales en escenarios donde los retardos y el efecto multicamino pueden afectar significativamente el rendimiento del sistema.

La generación de los códigos LS se basa en el uso de **Conjuntos de Secuencias Complementarias (Complementary Set of Sequences, CSS)** y otras técnicas que permiten obtener secuencias con propiedades de correlación específicas. Aunque el proceso matemático es complejo, a grandes rasgos se sigue el siguiente procedimiento:

1- Selección de Secuencias Base:

- Se utilizan parejas de secuencias de Golay, que son secuencias binarias con propiedades especiales de correlación, son correlacionadas entre sí y sirven como base para construir los códigos LS.

2- Combinación y Concatenación:

- Las secuencias de Golay se combinan y concatenan siguiendo ciertas reglas y patrones establecidos, se aplican coeficientes de polaridad, que determinan cómo se suman o restan las secuencias entre sí.
- Se utiliza una **matriz de Hadamard** para definir las polaridades y combinaciones necesarias.

3- Creación de la Zona de Correlación Cero (ZCZ):

- Al combinar las secuencias de manera específica, se logra que la función de correlación presente una zona alrededor del pico principal donde la correlación es nula o muy baja. Esta ZCZ es fundamental para minimizar las interferencias entre señales y mejorar la detección en presencia de retardos.

4- Generación de Múltiples Códigos LS:

- Se pueden generar varios códigos LS, lo que permite que múltiples usuarios o dispositivos utilicen el sistema simultáneamente sin interferir entre sí. Los códigos adicionales se pueden obtener modificando ciertos parámetros, como las polaridades o el orden de las secuencias. La posibilidad de generar un gran número de códigos los hace flexibles y escalables para diferentes aplicaciones.



Entre los beneficios y aplicaciones que este método ofrece se encuentran:

- **Reducción de Interferencias:**

La presencia de la ZCZ minimiza las interferencias tanto por acceso múltiple como entre símbolos, mejorando la integridad de los datos transmitidos.

- **Mejora en la Detección y Sincronización:**

Las propiedades de correlación de los códigos LS facilitan la detección precisa de las señales y la sincronización entre el transmisor y el receptor.

- **Adaptabilidad:**

Los códigos LS pueden ajustarse para adaptarse a diferentes tamaños de ventana de correlación y requerimientos de rendimiento, ofreciendo flexibilidad en el diseño del sistema.

Códigos Zadoff-Chu

Los **códigos Zadoff-Chu**[12] son secuencias especiales utilizadas en sistemas de comunicación por sus propiedades únicas y beneficiosas. Son secuencias complejas de amplitud constante y fase variable que presentan una autocorrelación periódica perfecta, es decir, tienen un pico pronunciado en el desfase cero y valores prácticamente nulos en otros desfases dentro de un período. Esto las hace ideales para aplicaciones que requieren una sincronización precisa y resistencia a interferencias.

Características principales:

- **Amplitud constante:** Todos los elementos de la secuencia tienen la misma magnitud, evitando variaciones en la potencia transmitida.
- **Autocorrelación ideal:** Permite detectar con exactitud el momento de llegada de una señal, esencial para la sincronización entre transmisor y receptor.
- **Baja correlación cruzada:** Mínima interferencia entre diferentes secuencias, lo que es útil en sistemas con múltiples usuarios o señales simultáneas.
- **Resistencia a desplazamientos en frecuencia:** Mantienen sus propiedades incluso si la señal experimenta cambios en frecuencia, como en situaciones con efecto Doppler.

Aplicaciones comunes:

- **Sincronización en comunicaciones LTE:** Utilizadas para sincronizar transmisiones en redes móviles de alta velocidad.
- **Sistemas de localización y posicionamiento:** Mejoran la precisión en la detección del tiempo de llegada de señales, esencial para calcular distancias.
- **Comunicaciones subacuáticas:** Su robustez frente a reflexiones y desplazamientos en frecuencia las hace adecuadas para entornos marinos.

Ventajas de usar secuencias Zadoff-Chu:

- **Precisión en detección y sincronización:** La autocorrelación perfecta facilita la identificación de señales específicas.



- **Reducción de interferencias:** La baja correlación cruzada minimiza la interferencia entre usuarios o dispositivos.
- **Robustez frente a condiciones adversas:** Mantienen su rendimiento en entornos con multicamino o movimientos que causan desplazamientos en frecuencia.

Si bien las secuencias de Kasami son binarias y ofrecen buenas propiedades de correlación, las Zadoff-Chu, al ser multifase y complejas, proporcionan una autocorrelación perfecta y son más robustas frente a desplazamientos en frecuencia. Por otro lado, los códigos LS tienen una Zona de Correlación Cero que es útil en entornos con multicamino. Pero las secuencias Zadoff-Chu ofrecen autocorrelación ideal sin necesidad de una zona específica, lo que puede ser una ventaja en ciertos sistemas. Sin embargo, aunque este método destaca por su eficiencia, requiere muchos más recursos computacionales que los demás métodos mencionados, es importante asegurarse de que el sistema de localización esté diseñado para aprovechar las propiedades de las secuencias Zadoff-Chu, incluyendo la capacidad de procesar señales complejas.

En la herramienta que se desarrollará, se podrá ingresar el código ya generado con cualquiera de los métodos, y a partir de allí se podrá avanzar con la emisión.

Emisión

La emisión de señales en un sistema de localización acústica es un proceso complejo que abarca la generación, modulación y transmisión de señales ultrasónicas. Cada uno de estos pasos es fundamental para asegurar que las señales puedan ser recibidas y procesadas correctamente en el receptor. En la aplicación desarrollada, la pantalla de configuración de emisión debe permitir al usuario definir todos los parámetros necesarios para simular una emisión efectiva, brindando flexibilidad y control sobre las particularidades del caso.

En esta sección, se describen los parámetros que se configuran para la emisión, abordando aspectos como la modulación de la señal, la elección del transductor, y otros parámetros complementarios que influyen como la calidad de información y la relación entre emisores.

Configuración de la Modulación

La **modulación** es un proceso esencial para adaptar los códigos que queremos transmitir a una señal **portadora** que ocupe el ancho de banda que puede manejar el **transductor**. Cada transductor tiene ciertas limitaciones sobre la cantidad de información que puede transmitir de manera eficiente, y la modulación permite ajustar la señal para que se adapte a estas características.

La elección de la técnica de modulación tiene un impacto directo en la calidad de la correlación de las señales, es decir, en la capacidad del sistema para comparar y reconocer patrones en la señal recibida. En particular, las modulaciones por desplazamiento de fase, como **BPSK (Binary Phase-Shift Keying)** y **QPSK (Quadrature Phase-Shift Keying)**, son ampliamente utilizadas en aplicaciones de localización y posicionamiento acústico **ALPS (Acoustic Local Positioning System)**.

En la plataforma de simulación desarrollada, se ha integrado BPSK como la modulación principal para trabajar con secuencias binarias.

Esta técnica es relativamente sencilla de implementar y permite un uso eficiente del ancho de banda disponible en el transductor. Sin embargo, una de las desventajas de estas



modulaciones es que generan lóbulos laterales alrededor del pico principal de correlación. Estos picos secundarios, aunque más pequeños, pueden causar errores en la estimación del tiempo de vuelo de la señal, afectando la precisión de la localización si no se manejan adecuadamente.

Los parámetros que se pueden configurar relacionados con la modulación se muestran En la siguiente tabla, se muestran los parámetros configurables para la modulación:

Parámetro	Notación	Descripción
Tipo de modulación	BPSK	BPSK se utiliza para modular la señal antes de su transmisión. Cambia la fase de la portadora entre dos valores (0° y 180°) para representar los bits 0 y 1.
Frecuencia de la portadora	F_c	Define la frecuencia de la señal portadora, es importante para la transmisión efectiva, asegurando una buena propagación de la señal al adaptarse a las características del transductor y el medio.
Tipo de Portadora	Carrier Type	Tipo de onda que se usará como portadora. Las ondas sinusoidales son comunes en aplicaciones ultrasónicas por su estabilidad y facilidad de sincronización con el receptor, lo cual es esencial para una correcta demodulación.
Ciclos de la Portadora	N_c	Define cuántos ciclos de la portadora se usan para transmitir cada bit.
Factor de Sobremuestreo	O_f	Relación entre la frecuencia de muestreo (calidad de la representación digital de la frecuencia analógica) y la frecuencia de la portadora. Mientras más grande sea, mejora la resolución temporal de la señal, reduciendo los lóbulos laterales en la correlación y mejorando la calidad de la señal transmitida.

Tabla 3 - Parámetros para configurar la modulación

Configuración del Transductor:

Los transductores son los dispositivos que convierten la señal eléctrica en ondas ultrasónicas y viceversa, por lo cual tienen un papel fundamental tanto en la etapa de emisión como en la de recepción. Las propiedades del transductor afectan significativamente la calidad y las características de la señal que se transmite, por lo cual es importante, en la medida que se pueda, modelar estos efectos en la simulación.

En la aplicación desarrollada, se permite seleccionar entre un modelo de transductor precargado, introducir un modelo personalizado, o trabajar sin simular el efecto del transductor. Cada transductor tiene una respuesta en frecuencia característica que define cómo amplifica o atenúa las distintas frecuencias de la señal transmitida.

Esta simulación se logra utilizando un archivo CSV con los coeficientes del filtro FIR (Finite Impulse Response) para el transductor que se desee configurar.

- El modelo de transductor incluido es: Avi328ST160 [15]

Configuración de la técnica de acceso al medio:

En un sistema con múltiples emisores, es esencial gestionar cómo y cuándo cada emisor transmite sus señales para evitar interferencias entre las mismas. La herramienta de simulación facilitará tres opciones distintas de técnicas de acceso al medio para coordinar las transmisiones: CDMA, T-CDMA y la modalidad híbrida H-CDMA. Estas

técnicas permitirían que múltiples emisores compartan el mismo medio sin causar conflictos de señal.

- CDMA (Code-Division Multiple Access):

En la técnica de acceso CDMA, todos los emisores transmiten de forma simultánea utilizando un código modulado único para cada uno. El receptor es capaz de identificar de quién proviene cada señal. Existe un tiempo de espera definido entre el final de una emisión y el comienzo de la siguiente repetición del mismo código, denominado **Tgap**. Esto permite que la emisión tenga un ciclo de pausa, ayudando en la identificación de las señales en la etapa de recepción.

- T-CDMA (Time-Code Division Multiple Access)

A diferencia de CDMA, en T-CDMA los emisores no transmiten de manera simultánea, sino que lo hacen uno después de otro de manera sucesiva en una secuencia temporal. Cada emisión está separada por un tiempo **Tshift = Te** que es equivalente al tiempo de emisión. En esta técnica, cada emisor transmite de manera sucesiva para evitar solapamientos y minimizar la interferencia directa. Además, comparte con el método anterior el tiempo **Tgap**.

- H-CDMA

En esta modalidad híbrida. Cada emisión está separada por un tiempo **Tshift** configurable. Además, también comparte con el método anterior el tiempo **Tgap**.

En la Figura 1, se observan estas características de los tipos de acceso al medio, en a) se representa la técnica CDMA, en b) se representa la técnica T-CDMA y el c) la técnica H-CDMA. Además, en la figura aparece T_{rep} , que es el tiempo que transcurre entre el inicio de una emisión y el inicio de la siguiente para el mismo emisor.

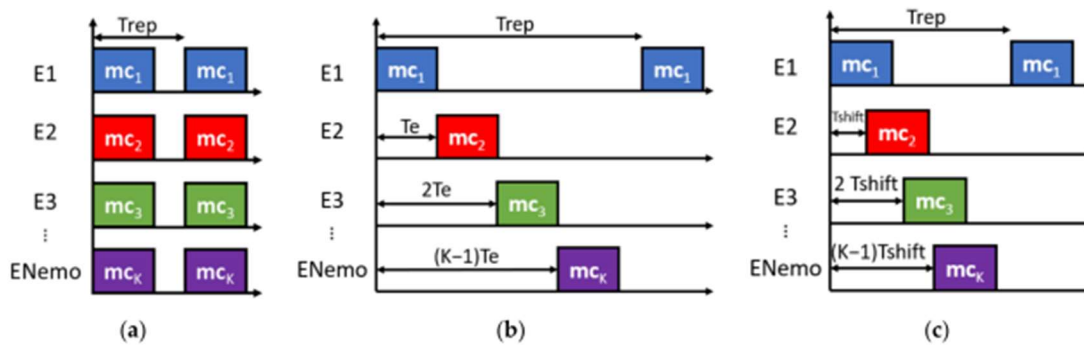


Figura 1 - Técnica de acceso al medio, extraída de [1]

En la siguiente tabla, se muestran los parámetros configurables para la técnica de acceso al medio:

Parámetro	Notación	Descripción
Técnica de acceso al medio	CDMA T-CDMA H-CDMA	Tipo de acceso al medio
Tiempo entre repeticiones del mismo emisor	Tgap	Es el tiempo de espera definido entre el final de una emisión y el comienzo de la siguiente repetición del mismo código. Este parámetro ayuda a mantener un ciclo regular de transmisión y permite gestionar la carga de transmisión en la red.
Tiempo entre emisores	Tshift	Define la cantidad de retardo temporal entre los emisores que transmiten de manera sucesiva. Este retardo se utiliza para gestionar



		cuándo cada emisor puede comenzar su transmisión, asegurando que las emisiones estén bien organizadas en el tiempo.
--	--	---

Tabla 4 - Parámetros para configurar técnica de acceso al medio

Configuración de otros aspectos generales de la emisión:

Además de la modulación, el transductor y la técnica de acceso al medio, la emisión también depende de otros aspectos importantes, que son necesarios agregar para la simulación, y se listan en la siguiente tabla:

Parámetro	Notación	Descripción
Frecuencia de Muestreo (Sampling Frequency)	fse	La frecuencia de muestreo afecta la calidad de la representación digital de la señal. Una frecuencia alta garantiza una mejor representación de la señal analógica, mejorando la precisión de la transmisión.
Número de Emisores Simultáneos	Nemo	Es el número de emisores que se utilizan en la simulación.
Posiciones de cada emisor	En	Cada n-emisor deberá tener sus coordenadas [x, y, z]
Tiempo de emisión	Te	Determina la duración total de la transmisión de la señal modulada
Número de Repeticiones	Nrep	Indica cuántas veces se repetirá la misma secuencia de emisión en la simulación
Ventana de Exclusión alrededor del Pico Principal	Vent	Define una ventana en el receptor para excluir los lóbulos laterales en el proceso de correlación. Se aplica para intentar reducir errores introducidos por picos secundarios y mejorar la precisión en la estimación de la posición.

Tabla 5 - Parámetros para configurar aspectos generales de la emisión

Recepción:

Configuración general del receptor:

La recepción en la herramienta es conformada por la configuración de la posición del receptor, y de su frecuencia de muestreo. También se podrá configurar un factor de submuestreo o downsampling, para cuando sea necesario una reducción de la tasa de datos para una menor carga computacional o mejorar la optimización.

Además, la aplicación para la simulación deberá proveer el medio para mostrar las posiciones en el espacio de los emisores con el receptor. En la siguiente tabla se muestran y se clasifican los parámetros para la configuración del receptor.

Parámetro	Notación	Descripción
Posición del receptor	Rec	El receptor deberá tener las coordenadas [x, y, z].
Frecuencia de muestreo del receptor	fser	Es el número de muestras por segundo que se utilizan para digitalizar una señal recibida. Se expresa en hercios (Hz), que indica la cantidad de muestras por segundo.
Downsampling (o submuestreo)	down	Según el factor ingresado se disminuye la cantidad de muestras que representan la señal a lo largo del tiempo, lo que permite reducir la cantidad de datos y ancho de banda necesario para almacenar o procesar la señal, pero perdiendo algo de la información original de la señal, lo cual puede afectar la calidad o precisión de la representación.

Tabla 6 - Parámetros para configurar el receptor



Efectos:

Esta sección se despliega en la herramienta como una tercera etapa. Hasta aquí se ha configurado la emisión del código modulado que cada emisor envía a través del transductor, cuya respuesta afecta la señal enviada y se ha configurado el receptor y su frecuencia de muestreo. Pero antes de llegar al receptor, en la práctica, existen muchos factores que pueden intervenir en la señal enviada, para eso la herramienta dispone la sección de efectos con la posibilidad de añadir algunos de estos efectos correspondientes al medio que pueden incidir en la propagación de las señales.

Efecto Ruido Blanco Gaussiano:

El ruido blanco gaussiano se añade a la señal recibida para simular el ruido que se encuentra naturalmente en muchos entornos. La adición de ruido blanco gaussiano se realiza teniendo en cuenta la energía de las señales moduladas, para lograr un valor específico de relación señal/ruido. En la siguiente tabla se clasifican los parámetros para la configuración del efecto ruido blanco.

Parámetro	Notación	Descripción
Signal to Noise Ratio	SNR	Relación señal/ruido, es un valor específico que determina la proporción de la potencia de la señal con respecto al ruido.

Tabla 7 - Parámetros para configurar efecto Ruido blanco Gaussiano

Efecto Ruido Impulsivo:

El ruido impulsivo es un tipo de ruido que afecta a la señal durante un breve intervalo de tiempo y con gran amplitud, simulando interferencias repentinas que pueden ocurrir en ciertos entornos. Se aplica sólo a una parte de la señal. En la siguiente tabla se clasifican los parámetros para la configuración del efecto ruido impulsivo.

Parámetro	Notación	Descripción
Signal to Noise Ratio	SNR	Relación señal/ruido, es un valor específico que determina la proporción de la potencia de la señal con respecto al ruido.
Duración	duration	Duración del ruido impulsivo especificado como un porcentaje de la duración total de la señal.
Localización	location	Localización en el tiempo en la señal.

Tabla 8 - Parámetros para configurar efecto Ruido impulsivo

Efecto Multicamino:

El efecto multicamino se produce cuando la misma señal llega al receptor a través de diferentes trayectorias, lo que genera interferencias debido a que las señales reflejadas llegan con retardos y atenuaciones distintas. Este fenómeno se puede dar por presencia de paredes o de objetos en el entorno que producen la reflexión. Para simular el efecto multicamino, se agregan réplicas retardadas y atenuadas de la señal transmitida. En la siguiente tabla se clasifican los parámetros para la configuración del efecto multicamino.



Parámetro	Notación	Descripción
Tipo efecto	Manual Random	Si es Manual: Se especifica el retardo y la ganancia para cada señal emitida. En cambio si es Aleatoria: Se introducen dos retardos aleatorios con una ganancia proporcionalmente atenuada.
Delay	Delay	Si es manual se debe ingresar un tiempo específico de retardo para cada réplica de la señal.
Gain	Gain	Si es manual se debe ingresar la atenuación correspondiente para cada señal replicada.

Tabla 9 - Parámetros para configurar efecto Multicamino

Efecto Doppler:

El efecto Doppler modela el cambio aparente en la frecuencia de las señales emitidas debido al movimiento relativo entre los emisores y el receptor. Este efecto ocurre, por ejemplo, cuando el emisor y el receptor se están acercando o alejando entre sí, provocando un cambio en la frecuencia percibida de la señal. Para simular este efecto, se establece la velocidad del sonido, y una velocidad relativa entre emisores y receptor. En la siguiente tabla se clasifican los parámetros para la configuración del efecto doppler.

Parámetro	Notación	Descripción
Velocidad del sonido	vel	Se establece el medio ya que puede ser en el aire o en el agua, dependiendo de esa respuesta el sonido tiene diferente velocidad.
Velocidad relativa entre emisor-receptor	Relative speed between	Velocidad relativa entre emisor-receptor, teniendo en cuenta las posiciones (x, y, z) de cada emisor y del receptor.

Tabla 10 - Parámetros para configurar efecto Doppler

Efecto de Temperatura y Absorción Atmosférica:

Este efecto tiene en cuenta dos factores importantes para la propagación del sonido en un medio gaseoso (como el aire): la temperatura del ambiente y la absorción atmosférica. Ambos afectan la propagación de la señal acústica, influyendo en la velocidad del sonido y en la atenuación de la señal. En la siguiente tabla se clasifican los parámetros para la configuración del efecto de temperatura y absorción atmosférica.

Parámetro	Notación	Descripción
Temperatura	temp	Valor en grados Celsius para ajustar la velocidad del sonido según las condiciones ambientales.
Absorción Atmosférica:	Atmospheric absorption	Valor en decibelios por metro para definir la cantidad de atenuación que sufre la señal en el medio debido a los efectos atmosféricos.

Tabla 11 - Parámetros para configurar efecto de temperatura y absorción atmosférica

Resultados:

Resultados de Correlación y métricas

La plataforma ofrece varias métricas que permiten analizar y comparar las diferentes señales emitidas a lo largo de la simulación. Estos factores se aplican tanto en etapas iniciales, cuando se emiten los códigos en banda base, como en la fase de recepción. Entre los factores más importantes se encuentra la cota de autocorrelación (AC), que mide cuán



bien se puede distinguir el pico principal de los picos secundarios (lóbulos laterales), y la cota de correlación cruzada (CC), que mide la diferencia entre códigos diferentes, minimizando la interferencia entre emisores. Además, la plataforma ofrece herramientas para visualizar tanto las autocorrelaciones de los códigos que identifican a cada emisor como las correlaciones cruzadas, así como la transformada rápida de Fourier (FFT) de las señales, permitiendo analizar el espectro de frecuencia de las mismas.

Resultados en la Etapa de Recepción

En la etapa de recepción, la plataforma muestra cómo la señal ha sido afectada por los efectos configurados, como el ruido o los reflejos debidos al multicamino. Aquí se presentan los resultados de la correlación cruzada entre la señal recibida y los códigos originales, además de la FFT de la señal recibida, lo cual da una visión detallada del contenido frecuencial. Un parámetro clave que se calcula es el Sidelobe-to-Mainlobe Ratio (SMR), que compara los valores de los lóbulos secundarios con el pico principal. Este valor nos indica qué tan fácil es identificar el pico principal de la correlación, y es esencial para evaluar la calidad de la señal y su resistencia frente a interferencias. Una relación baja implica una señal clara, mientras que una relación alta sugiere dificultades en la detección precisa de la señal.

Resultados de Posicionamiento

Finalmente, la plataforma presenta los resultados de la estimación de la posición del receptor, basados en la correlación, utilizando dos métodos de posicionamiento: trilateración esférica o trilateración hiperbólica, y permite al usuario seleccionar entre dos algoritmos de posicionamiento, el GN (Gauss-Newton) o el CM (Método Combinado), en el caso esférico.

El usuario puede ajustar parámetros como el número de repeticiones o el método utilizado para volver a lanzar la simulación de esta sección. Los resultados incluyen estadísticas como la cantidad de medidas no válidas, la media y la desviación estándar de las posiciones. Adicionalmente, la plataforma permite visualizar gráficamente la disposición de los emisores y la ubicación estimada del receptor, proporcionando un panorama claro del rendimiento del sistema y su precisión para localizar el receptor.

Gestión de usuarios

La plataforma dispondrá los métodos para tener diferentes perfiles en la aplicación. Un perfil de usuario podrá utilizar la herramienta de simulación, y un perfil administrador que será capaz de realizar altas, bajas, y modificación de los usuarios. Además, el usuario que cuente con permisos de administrador podrá dar de alta nuevos recursos en la plataforma como nuevos transductores, universidades, y demás.

Gestión de proyectos

La plataforma tendrá la capacidad de almacenar proyectos, con los valores de los parámetros que hayan sido ingresados para volver a revisarlos en otro momento.

También se agrega la funcionalidad de que un usuario puede compartir un proyecto con algún otro usuario.



CAPÍTULO 4:

DOMINIO DE LA APLICACIÓN Y ENFOQUE DE DESARROLLO

Visión

El objetivo principal de la plataforma desarrollada es simular sistemas de localización acústica en entornos interiores, donde se busca determinar la posición de un receptor utilizando múltiples emisores. Esta simulación se basa en la transmisión y procesamiento de señales moduladas, con la intención de evaluar la precisión de la localización en condiciones específicas.

La plataforma permite a los usuarios configurar la emisión y recepción de señales, definiendo parámetros como el esquema de codificación y modulación, la posición de los emisores y receptores, así como el modelo de transductor a utilizar. Los usuarios pueden elegir entre un conjunto de modelos predefinidos de transductores o, alternativamente, introducir un modelo personalizado que se ajuste mejor a sus necesidades. Esta flexibilidad permite personalizar la simulación y adaptar los parámetros para analizar distintos escenarios de propagación y localización.

La herramienta ofrecerá una opción personalizada donde el usuario podrá cargar cualquier código modulado por cualquier método para la transmisión de las señales, lo cual permite analizar cómo estas modulaciones afectan la precisión de la localización. Adicionalmente, se pueden aplicar técnicas para el acceso al medio, como T-CDMA (Acceso Múltiple por División de Código y Tiempo) o CDMA (Acceso Múltiple por División de Código), que garantizan que múltiples señales puedan ser emitidas y recibidas simultáneamente sin interferencias significativas.

La configuración también incluye la especificación de la posición del receptor, así como la frecuencia de muestreo a la que operará el sistema. Esta frecuencia de muestreo determina la precisión temporal de las señales recibidas, un factor clave en la estimación precisa de la posición. La herramienta permite a los usuarios configurar parámetros adicionales para el procesamiento de las señales recibidas, incluyendo el uso de un detector de picos de correlación y la selección de algoritmos de posicionamiento.

Una vez configurada la emisión y el receptor, la plataforma permite a los usuarios definir parámetros para simular efectos propios del medio de propagación de las señales acústicas. Los efectos que se tienen en cuenta para esta sección son los siguientes:

- Ruido blanco gaussiano
- Multipath
- Efecto Doppler
- Ruido impulsivo
- Absorción atmosférica

Posteriormente, se simula la recepción de la señal, procesando los datos recibidos para calcular la correlación con cada uno de los códigos emitidos originalmente. Para ello, se emplean detectores de picos para identificar las correlaciones significativas, que luego



son convertidas en distancias en función de la velocidad del sonido y de la frecuencia de muestreo establecida.

La plataforma proporciona herramientas para la visualización interactiva de los resultados obtenidos durante todas las fases de la simulación. Los usuarios pueden visualizar desde la configuración de la señal en banda base hasta las señales moduladas emitidas por los transductores y las señales recibidas tras ser procesadas. Además, se proporcionan estadísticas útiles como el número de picos detectados, la cota de autocorrelación y la estimación de la posición del receptor.

La visualización se realiza a través de Plotly, permitiendo gráficos interactivos donde se muestran los trayectos de las señales, las posiciones estimadas y las llegadas al receptor. Esto facilita la interpretación de los resultados y permite ajustar los parámetros para obtener simulaciones más precisas.

La plataforma debe garantizar la administración de usuarios permitiendo la autorización y autenticación de cada usuario que desee utilizar la herramienta. Además, debe proveer las herramientas para que un usuario pueda guardar los datos de entrada de una simulación si lo desea, y para que ciertas simulaciones puedan ser compartidas entre usuarios.

Especificación de los requerimientos:

Con el objetivo de mejorar el orden del desarrollo, a continuación, se presentan los principales requerimientos que la plataforma debe permitir realizar a los usuarios:

1. Acceder a la aplicación (Loguearse)

- El usuario accede al sistema mediante un formulario de inicio de sesión. La autenticación se realiza utilizando **JWT**, que permite al usuario recibir un token único para acceder a las funcionalidades de la plataforma.

2. Realizar una nueva simulación

- El usuario tiene la posibilidad de iniciar una nueva simulación, y para ello se le presentan varias opciones:
 - **Cargar Variables de Emisión (Código sin Modular):** El usuario puede cargar un código sin modular, que luego será procesado y modulado dentro de la aplicación. En este punto, se ofrece un **caso general personalizado**, donde se permite al usuario definir distintos parámetros antes de proceder con la modulación de las señales.
 - **Cargar un Código Modulado:** Alternativamente, el usuario puede ingresar directamente un código que ya ha sido modulado, permitiendo saltar el análisis de banda base.

3. Ingresar datos de emisión

- El usuario puede introducir parámetros específicos para las señales emitidas, como la frecuencia de transmisión, la potencia de la señal y otros atributos relevantes para la simulación.

4. Ingresar coordenadas de los emisores



- El usuario puede definir la posición de cada emisor dentro del entorno de simulación. Estos datos se utilizan para calcular la propagación de las señales en el espacio definido.
5. **Ingresar coordenadas de los receptores, y sus características**
- Similar al caso anterior, el usuario introduce la ubicación de los receptores. Esto es crucial para determinar la calidad y precisión de las señales recibidas y para evaluar los algoritmos de localización.
6. **Agregar efectos a la emisión**
- Los efectos disponibles que el usuario puede agregar a la emisión incluyen:
 - Ruido Gaussiano:** Simula el ruido blanco que puede afectar la señal.
 - Multipath:** Modela el fenómeno de múltiples trayectorias que las señales pueden seguir al reflejarse en superficies.
 - Efecto Doppler:** Introduce la variación de frecuencia debido al movimiento relativo entre emisor y receptor.
 - Ruido Impulsivo:** Modela eventos de ruido esporádico y de gran intensidad.
 - Absorción Atmosférica y Temperatura:** Permiten simular la atenuación de la señal debido a las condiciones ambientales y los efectos del medio.
7. **Visualizar los resultados**
- Después de configurar todas las variables y ejecutar la simulación, el usuario puede visualizar los resultados de forma interactiva mediante gráficos generados con **Plotly**. Los resultados incluyen la estimación de la posición del receptor, la propagación de las señales, y el análisis de los efectos incorporados.
8. **Guardar los datos de la simulación**
- Los datos de la simulación, incluyendo las variables de emisión, posicionamiento de emisores y receptores, y el código del transductor, pueden ser guardados en la base de datos. Esto permite al usuario acceder a la simulación en el futuro o compartirla con otros usuarios.
9. **Acceder a una simulación guardada**
- La aplicación permite cargar simulaciones previamente guardadas, brindando al usuario la posibilidad de revisar, analizar y modificar configuraciones pasadas.
10. **Compartir con otro usuario, un proyecto guardado**
- El sistema permite que los usuarios compartan sus simulaciones guardadas con otros usuarios registrados, facilitando la colaboración y el trabajo en equipo. Esto es posible gracias a la gestión de usuarios basada en **JWT**, que permite definir permisos específicos para cada usuario con el que se comparte un proyecto.



Metodología Ágil

Para garantizar un desarrollo eficiente y flexible, se adoptó una metodología ágil. Esta metodología permitió trabajar en ciclos iterativos, haciendo uso de herramientas de CI/CD (Integración Continua y Despliegue Continuo) donde cada iteración se orienta a la entrega de pequeñas funcionalidades que son integradas al desarrollo actual mejorando cada versión y acercándose al producto final. La adopción de este enfoque proporciona los siguientes beneficios:

Flexibilidad para Cambiar Requisitos y Soluciones: Dado que el proyecto fue evolucionando a medida que se identificaban nuevas necesidades, el enfoque ágil permite incorporar cambios de manera rápida y efectiva. Se valora más la adaptación del software que funciona contra la documentación exhaustiva.

Entrega Continua: Los desarrollos iterativos y la integración continua facilitan la entrega regular de incrementos del sistema, lo cual permite a los interesados evaluar las funcionalidades desarrolladas y ofrecer una retroalimentación en etapas tempranas y de manera recurrente.

Mejora Continua: Este tipo de enfoque tiene la particularidad de que el desarrollo se va gestando, corrigiendo y alineándose con las expectativas y necesidades que van surgiendo iterativamente.

CAPÍTULO 5:

DISEÑO, DESARROLLO E IMPLEMENTACIÓN

En esta sección se da un acercamiento al diseño de software y se muestra la aplicación web para simulación a desarrollar.

Diseño

Diagrama conceptual:

En el desarrollo de un sistema, es importante establecer una base sólida de entendimiento sobre los elementos que forman parte del dominio del problema y las relaciones entre ellos, el siguiente diagrama conceptual, tiene como propósito ofrecer una visión general y simplificada del sistema, permitiendo a los lectores comprender la estructura fundamental sin necesidad de detalles técnicos o específicos de implementación.

El diagrama conceptual facilita la comunicación entre las partes interesadas, ya sean miembros del equipo de desarrollo o usuarios finales, al representar de manera clara los principales conceptos que forman parte de la aplicación y cómo estos interactúan entre sí. Esta representación bastante abstracta sirve como una guía visual para identificar las entidades involucradas y cómo se relacionan, proporcionando una base de conocimiento compartido para el diseño, la implementación y el análisis del sistema.

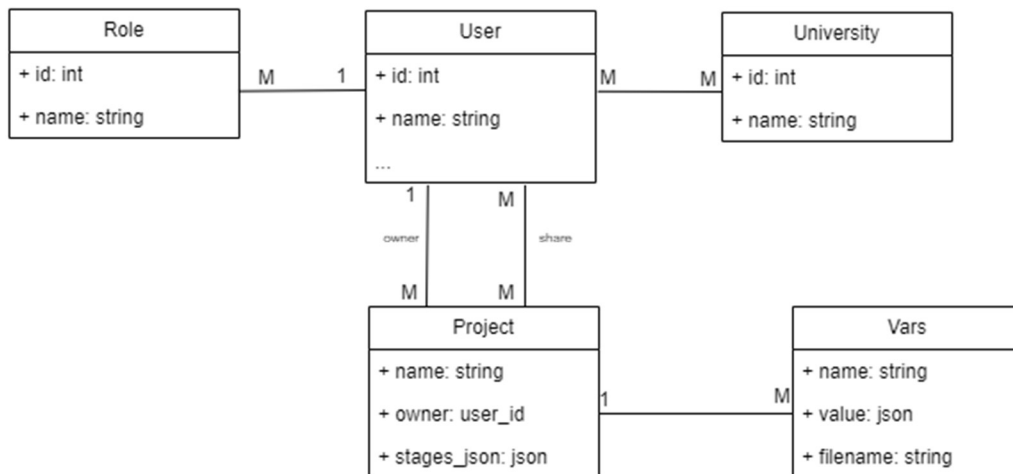


Figura 2 - Diagrama conceptual (elaboración propia)

Como se puede observar en el diagrama conceptual de la aplicación, en la Figura 2, el sistema tiene cuatro entidades fundamentales: Usuarios (User), Roles (Role), Proyectos (Project) y Variables (Variable). A continuación, se describen.



- **Usuario (User):** Representa a las personas que interactúan con el sistema. Cada usuario tiene atributos como nombre, apellido, correo electrónico, entre otros, y puede tener múltiples roles y participar en varios proyectos.
- **Rol (Role):** Define los permisos y niveles de acceso que puede tener un usuario dentro del sistema. Los roles están relacionados con los usuarios a través de una relación muchos a muchos.
- **Proyecto (Project):** Cada proyecto tiene un propietario (usuario), así como otros usuarios que pueden colaborar en él. Un proyecto también contiene varias variables asociadas a él, representadas en **JSON** (JavaScript Object Notation) para definir las etapas del proyecto.
- **Variable (Variable):** Las variables representan los parámetros de cada proyecto, organizados en diferentes etapas como emisión, recepción y efectos. Cada variable está asociada a un proyecto, y, por lo general, existe una variable de tipo JSON por cada etapa.

Se decidió almacenar las variables de esta forma para mejorar la flexibilidad y escalabilidad del sistema. Al guardar las variables en una tabla general, utilizando un campo de nombre y un campo de valor en formato JSON, se logra una mayor versatilidad para futuros cambios o ampliaciones de la aplicación. Esta estructura permite agregar nuevos parámetros o requisitos sin necesidad de modificar la estructura de la base de datos subyacente, lo que facilita la adaptación a nuevas funcionalidades.

Por ejemplo, en la etapa de emisión, existe una sola variable denominada objetoEmision, que contiene un objeto JSON. Este JSON puede guardar los pares clave-valor que hayan sido ingresados sin necesidad de que sean todos.

En el caso de almacenar un código modulado como parte del proyecto, seguramente el contenido del JSON será muy grande, en ese caso el sistema crea un archivo en el servidor y en la tabla de la base de datos, guarda solo el nombre del archivo recién creado, lo que permite buscarlo fácilmente cuando sea necesario cargar la información. Esto ayuda a gestionar de manera eficiente grandes volúmenes de datos y asegura que la aplicación siga siendo eficiente y escalable a medida que crecen los requisitos.

Relaciones:

- **Usuario - Rol:** Relación muchos a muchos, que permite asignar múltiples roles a un usuario y múltiples usuarios a un rol.
- **Usuario - Proyecto:** Relación muchos a muchos, permitiendo que múltiples usuarios colaboren en un proyecto.
- **Proyecto - Variable:** Relación uno a muchos, ya que un proyecto puede tener múltiples variables.

Diagrama de arquitectura:

A continuación, en la figura 3 se presenta el diagrama de arquitectura de la aplicación:

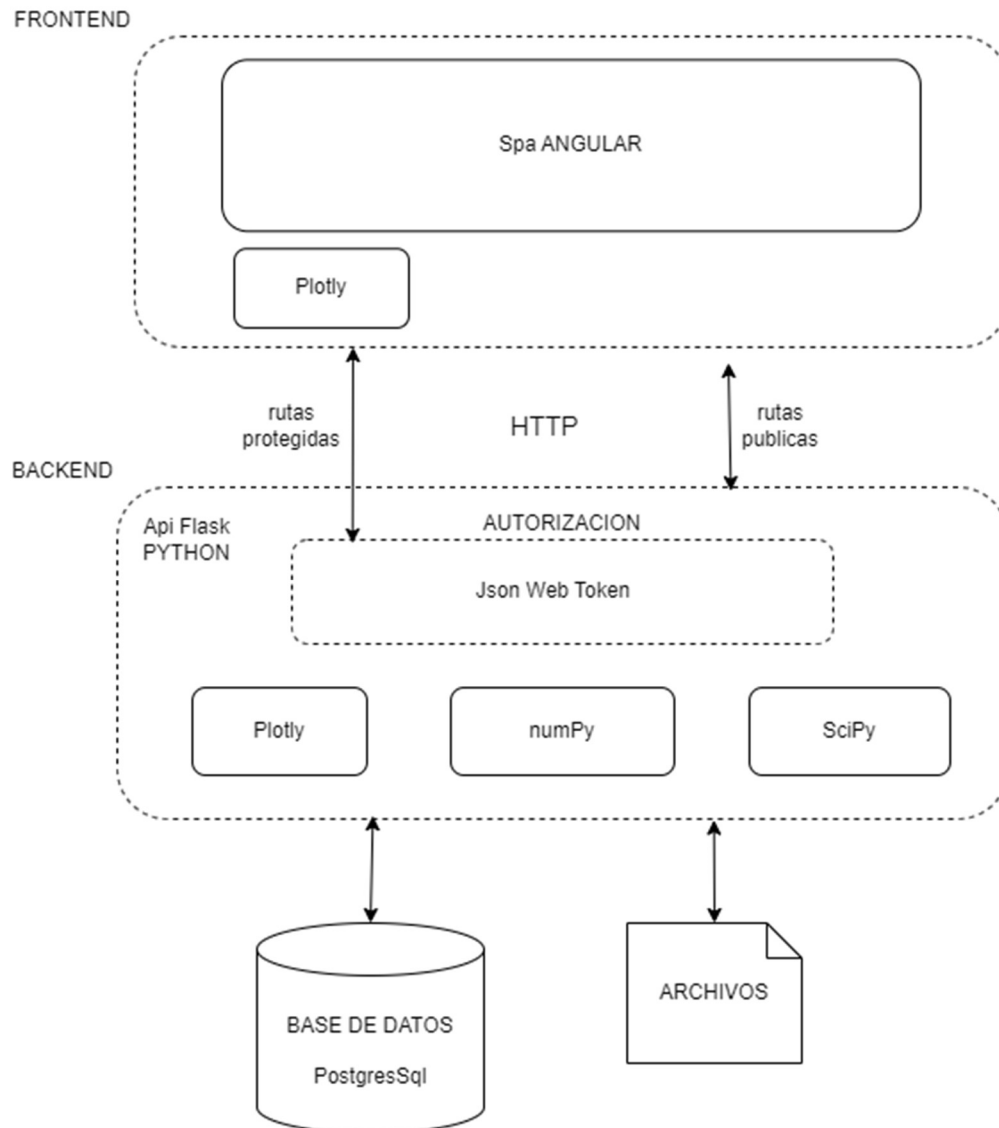


Figura 3 - Diagrama de arquitectura (elaboración propia)

El objetivo de este diagrama es proporcionar una visión clara de cómo está estructurada la aplicación desde un punto de vista global. Este diagrama muestra la arquitectura web que soporta la aplicación, mostrando cómo cada uno de los componentes se interconectan para cumplir con los objetivos del sistema.

Esta arquitectura de la aplicación se basa en un modelo cliente-servidor, donde el frontend (interfaz de usuario), desarrollado en Angular, se comunica con el backend mediante peticiones a una API REST. En el diagrama se representa cómo el frontend envía solicitudes HTTP al backend, que está desarrollado en Flask, para interactuar con los datos y la lógica del negocio. El backend procesa estas solicitudes y responde al frontend, proporcionando los datos necesarios para ser visualizados por el usuario.

En el backend, se utilizan diversas librerías para garantizar el rendimiento y la eficiencia del sistema, como NumPy y SciPy para el manejo de cálculos matemáticos y



procesamiento de señales. Además, se hace uso de Plotly para la generación de gráficos tanto en el backend como en el frontend, proporcionando visualizaciones interactivas y de alta calidad que permiten a los usuarios analizar los resultados de la simulación de manera detallada. Estas visualizaciones se generan en el backend como parte de los cálculos y se envían al frontend para ser mostradas de una forma clara y comprensible para el usuario.

El sistema de autenticación y autorización se gestiona mediante JSON Web Tokens (JWT), asegurando que solo los usuarios autenticados puedan acceder a ciertos recursos o funciones de la aplicación. En el diagrama se muestra cómo se utiliza JWT para autenticar y autorizar las solicitudes entre el cliente y el servidor, protegiendo así los datos y servicios sensibles.

También se incluye la conexión del backend con la base de datos, donde se almacenan y gestionan los datos relacionados con usuarios, proyectos, variables, y más. Esto permite que toda la información relevante se mantenga organizada y esté disponible para ser utilizada por la aplicación.

Este diagrama es importante para entender cómo se organizan y comunican los distintos componentes de la aplicación, asegurando la separación de responsabilidades y la escalabilidad del sistema. Al visualizar la arquitectura, se facilita la comprensión del flujo de datos y la interacción entre los componentes, lo cual es esencial para el análisis, diseño, desarrollo y mantenimiento de la aplicación. Además, proporciona una referencia clara para futuras mejoras y expansiones del sistema, asegurando que todos los desarrolladores tengan una visión común sobre la estructura y funcionamiento del sistema.

Desarrollo:

A continuación, se presenta la aplicación desarrollada junto con las pantallas correspondientes a cada etapa analizada en el capítulo anterior. Además, se realizará una prueba de la implementación. La aplicación es bastante grande e intuitiva, en este análisis sólo se verá un flujo general.

Login

Al querer ingresar a la aplicación por primera vez, el sistema solicita al usuario que introduzca su usuario y password con los cuales fue dado de alta. El administrador se encarga de dar de alta a los usuarios y comunicar estos datos al interesado. En la figura 4 se ve con claridad:

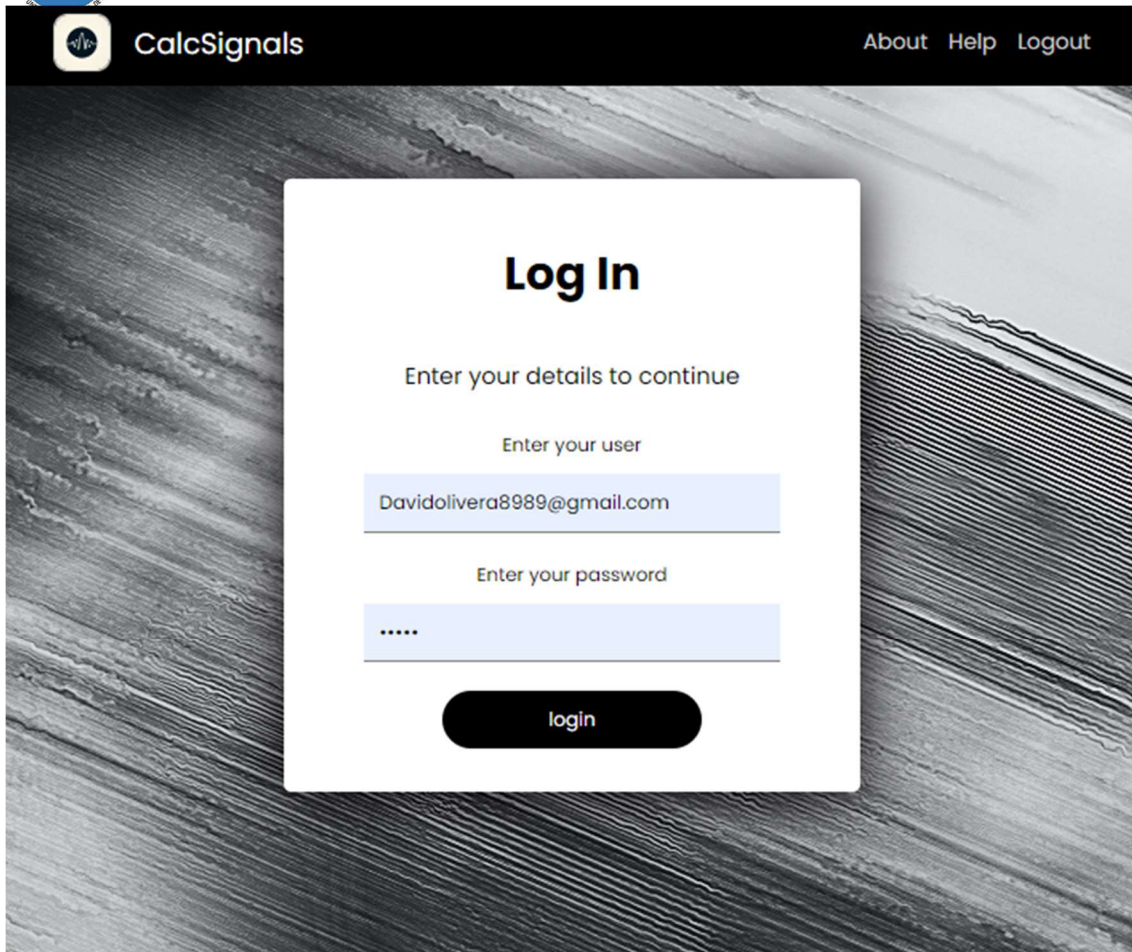


Figura 4 - Pantalla de login

Una vez logueado en el sistema, el usuario accede a una pantalla principal donde se muestra una tabla con sus proyectos guardados (si los tuviera), como se observa en la figura 5.

Los usuarios con permisos para realizar simulaciones verán el botón **New Simulation**, que permite ingresar al simulador. Por otro lado, aquellos con permisos de administrador también tendrán acceso al botón **Admin Users**.

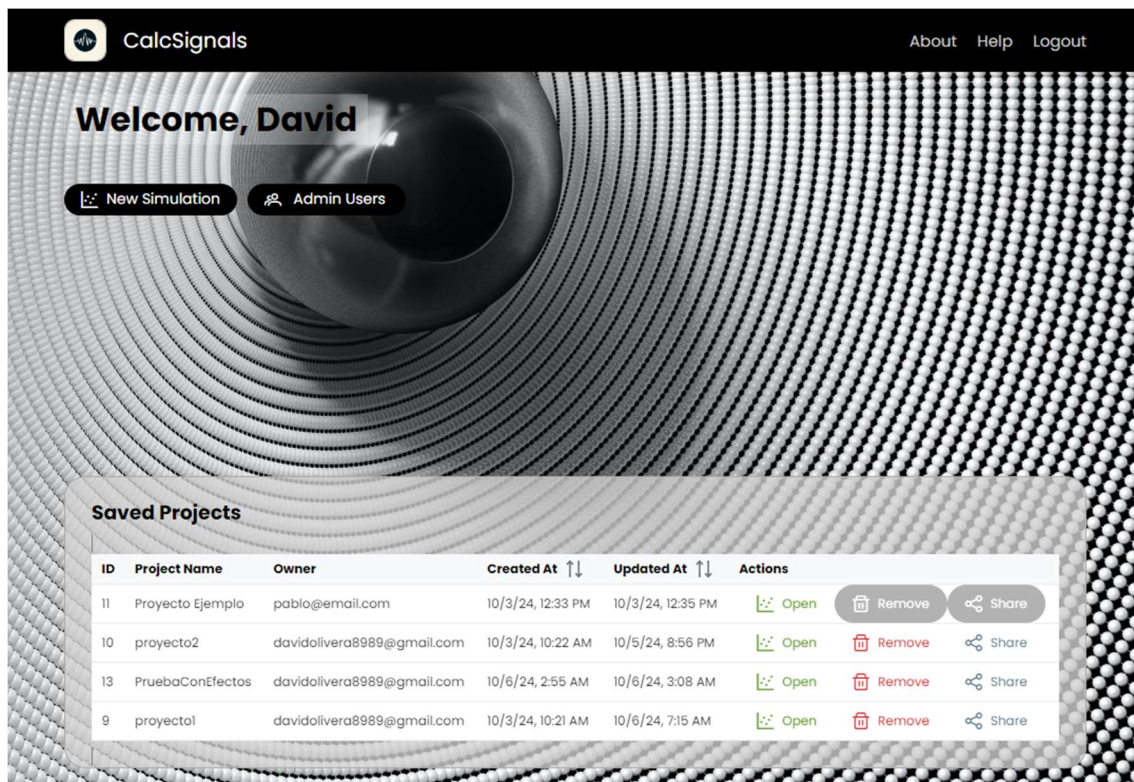


Figura 5 - Pantalla principal personal

Si se clickea en algún proyecto de la lista en el botón Open se abrirá el simulador y continuará el proyecto desde donde se guardó. En el caso que el usuario sea el creador del proyecto, tiene disponibles también los botones de Remove y Share para eliminar o compartir un proyecto. En ese caso puede realizar esas acciones si lo desea.

Emisión:

Si se desea crear una nueva simulación, al hacer click en el botón New Simulation se abrirá una ventana modal que le pedirá ingresar el nombre del nuevo proyecto. Igualmente, si lo desea luego podrá cambiar este nombre.

Al ingresar el nombre del proyecto y presionar en continuar se despliega la primera pantalla del simulador y pertenece a la etapa de emisión:

proyecto1 / Emission / About Help Logout

Emission Configuration: Customized

Inputs

Load based binary code **Cargar CSV**

Load modulated code **Cargar CSV**

Sampling frequency (fse): _____ kHzkHz
Exclusion window around the main peak: _____ samples

Length (L): _____
Number of simultaneous emitters (Nemo): _____

BPSK modulation

Carrier type:
Carrier frequency (fc) kHz: _____
Sampling frequency (fse) kHz: _____
Carrier cycles (Nc): _____
Oversampling factor (Of): _____
Emission time (Te) ms: _____

Emission

Emitters:
Number of repetitions (Nrep): _____

Guidance image

CDMA TDMA Hybrid

Delay between emitters (Tshift) ms: _____
Transductor model:

Open file

Default values **Save** **Cancel** **Next**

Figura 6 - Ventana Emisión

En esta ventana se ve claramente lo que se estudió en el capítulo anterior y es posible ingresar todos los parámetros de la emisión, del transductor, para la modulación, de acceso al medio, etc. Ver parámetros en tablas 3, 4 y 5 del capítulo 3.

Además, al principio se encuentran los botones para agregar un archivo CSV con el código ya modulado para simular la emisión o ingresar un código que ya haya sido codificado con alguno de los métodos vistos para realizar la modulación, completando los demás parámetros.



Cuando se hayan completado los parámetros necesarios para poder continuar con la emisión el botón Next en el lado inferior de la pantalla podrá ser clickeado, mientras tanto estará bloqueado. Con el botón Cancel se cancela la simulación y se vuelve a la pantalla anterior, con el botón Save claramente se avanza hacia el guardado del proyecto con los parámetros que se hayan cargado. Esto se almacenará en la variable objetoEmisión.

Abajo a la izquierda se agregó un botón Default values, éste sirve para completar en cada etapa valores por defecto, para poder avanzar, si en este momento avanzamos con este botón el formulario se completará de la siguiente manera, como muestra la figura 7:



Emission Configuration: Customized

Inputs

Load based binary code

Cargar CSV

Load modulated code

Cargar CSV

[CodigoKasamiModulado1630ft2Nc2.csv.X](#)

Sampling frequency (fse): khz/ki-hz

Exclusion window around the main peak: samples

Length (L):

Number of simultaneous emitters (Nemo):

BPSK modulation

Carrier type:

Carrier frequency (fc) khz:

Sampling frequency (fse) khz:

Carrier cycles (Nc):

Oversampling factor (Of):

Emission time (Te) ms:

Emission

Emitters:

e1:

x (cm):

y (cm):

z (cm):

e2:

x (cm):

y (cm):

z (cm):

e3:

x (cm):

y (cm):

z (cm):

e4:

x (cm):

y (cm):

z (cm):

e5:

x (cm):

y (cm):

z (cm):

Number of repetitions (Nrep):

Guidance image

CDMA TDMA Hybrid

Delay between emitters (Tshift) ms:

Transductor model:

Open file

Figura 7 - Valores por Defecto en emisión

Aquí vemos que se han cargado los valores por defecto para la emisión. Este caso disponible por defecto, carga a la emisión un código modulado generado con secuencias de Kasami el cual es posible descargar en el archivo csv correspondiente, desde la misma plataforma al hacer click en el nombre.

En este momento la variable emisión es un json con los siguientes valores:

```
{
  "opciones_load": "load_modulated_code",
  "exclusion_window_around_main_peak": 24,
  "length_L": 1512,
  "num_sim_emmitters_nemo": 5,
  "carrier_type": ["Sine"],
  "carrierFreq": 41.6667,
  "sampling_frequency_modulation": 500,
  "oversampling_factor": 12,
  "carrier_cycles": 2,
  "emission_time": 3.024,
  "tgap": 150,
  "num_Repetitions": 20,
  "modulation": "CDMA",
  "Tshift": 0,
  "transductor_model": [
    "328ST160"
  ],
  "cod_mod": [
    [...],
    [...],
    [...],
    [...],
  ]
}
```

Si se guardara este proyecto en este punto, como ya posee cargado un código modulado, la variable es bastante grande y se guardaría en un archivo en el servidor y en la base de datos se guardaría una referencia al archivo para esta variable y para este proyecto.

Al momento en que se llenan todos los parámetros de la emisión también se observa que se habilitan botones de la emisión que están en la parte inferior. Estos enlaces permiten visualizar algunos detalles de la emisión como el código modulado, la señal portadora, el modelo del transductor, entre otros. La siguiente imagen, figura 8 muestra la disponibilidad de visualizar los gráficos que se pueden generar con los datos de emisión:



Figura 8 - Salidas y gráficos de la emisión

En la figura 9, como ejemplo, se observa cómo responde el sistema al hacer click en Display Modulated Signal.

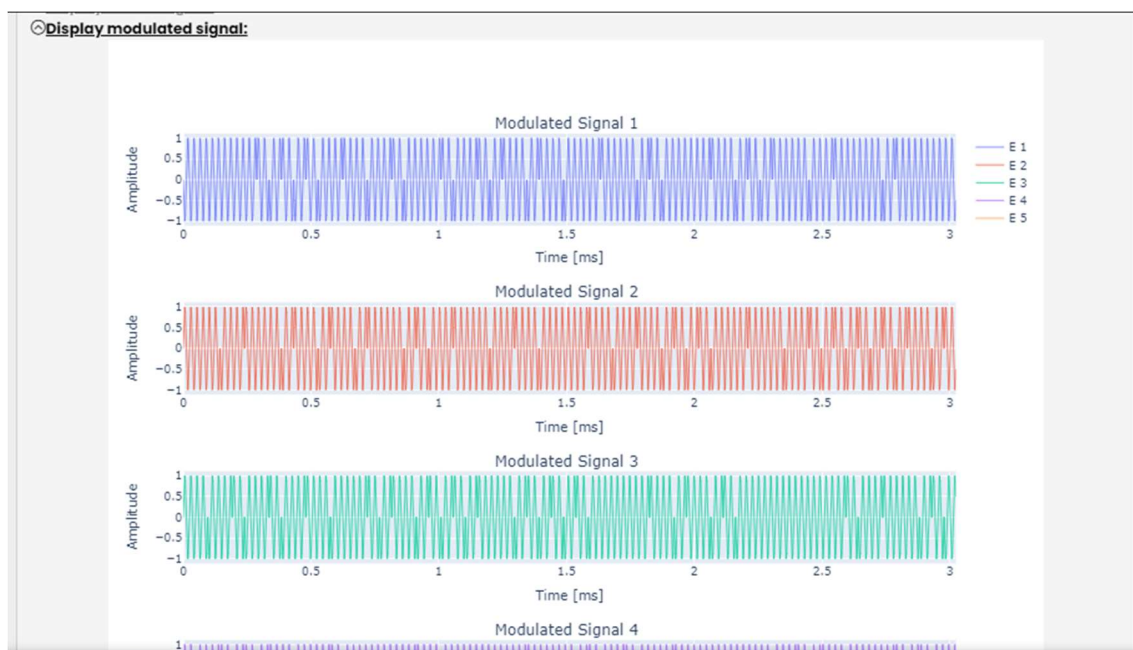


Figura 9 - Vista de gráficos emisión: Display modulated signa

De manera análoga si hiciera selección de Display signal with transducer Effects el sistema se mostraría como la figura 10:



Figura 10 - Vista de gráficos emisión: Display signal with transducer Effects

Se agrega también el ejemplo de mostrar el modelo del transductor, en la figura 11:



Figura 11 - Vista de gráficos emisión: See transducer model

Reception:

Avanzando, al clicar en el botón Next, se bloquea la sección de emisión y la pantalla se traslada a la siguiente sección Scene configuration: Reception.

En esta pantalla se ve la posición de cada emisor que fue cargado por defecto en la etapa de emisión y se observa el receptor que al llegar a esta pantalla, también por defecto es posicionado en $x:0$, $y:0$ y $z:0$ pero se puede modificar intuitivamente. A continuación, se renderiza un gráfico de tres dimensiones que permite visualizar claramente las posiciones de los emisores y el receptor. Debajo se pueden ingresar los parámetros para la frecuencia de muestreo del receptor y para downsampling, estos parámetros se describen en la tabla 6 del capítulo 3. Se puede observar lo descrito en la figura 12:

Scene configuration

Reception

Emitters:

e1: X (cm): <input type="text" value="0"/>	Y (cm): <input type="text" value="0"/>	Z (cm): <input type="text" value="328"/>
e2: X (cm): <input type="text" value="0"/>	Y (cm): <input type="text" value="51,2"/>	Z (cm): <input type="text" value="342,5"/>
e3: X (cm): <input type="text" value="51,5"/>	Y (cm): <input type="text" value="0"/>	Z (cm): <input type="text" value="336"/>
e4: X (cm): <input type="text" value="0"/>	Y (cm): <input type="text" value="-52,2"/>	Z (cm): <input type="text" value="342,5"/>
e5: X (cm): <input type="text" value="-51,4"/>	Y (cm): <input type="text" value="0"/>	Z (cm): <input type="text" value="336"/>

Receiver position:

rec1: X (cm): <input type="text" value="0"/>	Y (cm): <input type="text" value="0"/>	Z (cm): <input type="text" value="0"/>
--	--	--

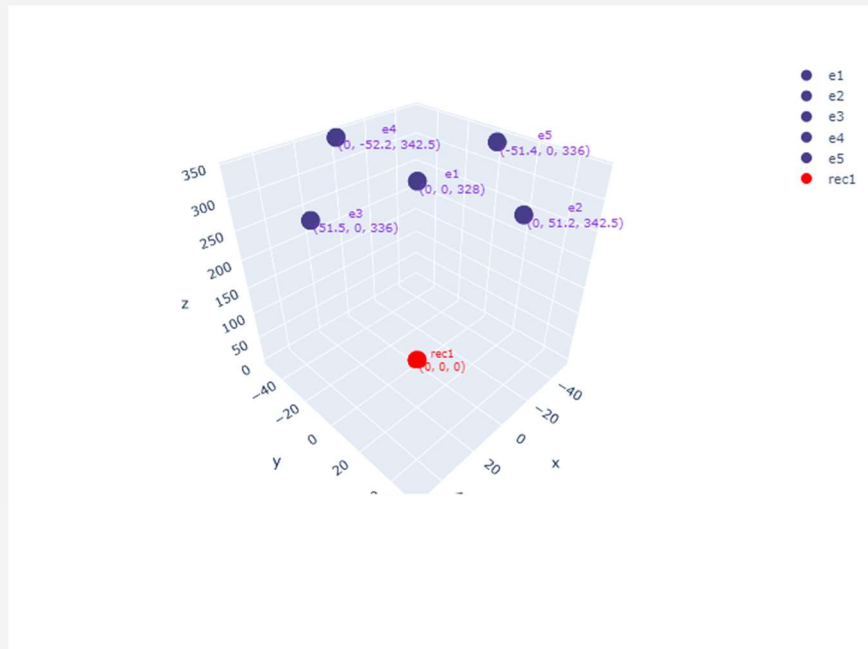
[Actualizar Gráfico](#)Receiver sampling frequency: kHzDownsampling: [Default values](#)[Save](#)[Back](#)[Next](#)

Figura 12 - Vista de Reception

Volviendo a clicar de default values aquí también completa los parámetros necesarios como muestra la figura 13 y luego permite avanzar a la sección siguiente:

Scene configuration

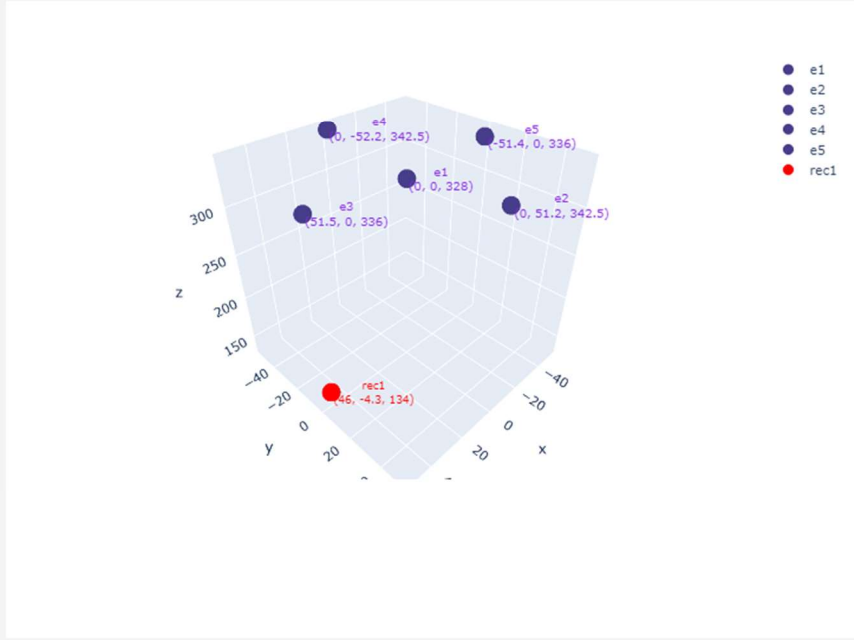
Reception

Emitters:

e1:	X (cm): <input type="text" value="0"/>	Y (cm): <input type="text" value="0"/>	Z (cm): <input type="text" value="328"/>
e2:	X (cm): <input type="text" value="0"/>	Y (cm): <input type="text" value="51,2"/>	Z (cm): <input type="text" value="342,5"/>
e3:	X (cm): <input type="text" value="51,5"/>	Y (cm): <input type="text" value="0"/>	Z (cm): <input type="text" value="336"/>
e4:	X (cm): <input type="text" value="0"/>	Y (cm): <input type="text" value="-52,2"/>	Z (cm): <input type="text" value="342,5"/>
e5:	X (cm): <input type="text" value="-51,4"/>	Y (cm): <input type="text" value="0"/>	Z (cm): <input type="text" value="336"/>

Receiver position:

rec1: X (cm): Y (cm): Z (cm):



Receiver sampling frequency: kHz
Downsampling:

Figura 13 - Vista de Reception default values

Effects:

Al avanzar y clickear en next la pantalla se desliza hasta la 3era etapa, la pantalla de efectos, los cuales no son requeridos y se puede avanzar sin seleccionar ninguno. Pero en el caso que se desee elegir alguno de los 5 tipos de efectos se deberán completar todos los parámetros que estos requieren para avanzar. Recordar las tablas 7, 8, 9, 10 y 11 del capítulo 3.



Para este ejemplo seleccionaremos solamente el efecto Doppler con la siguiente configuración, como se puede ver en la figura 14:



Figura 14 - Vista de Effects seleccionado Doppler

Results:

Al avanzar se llega finalmente a la pantalla resultados, se observa en la figura 15:

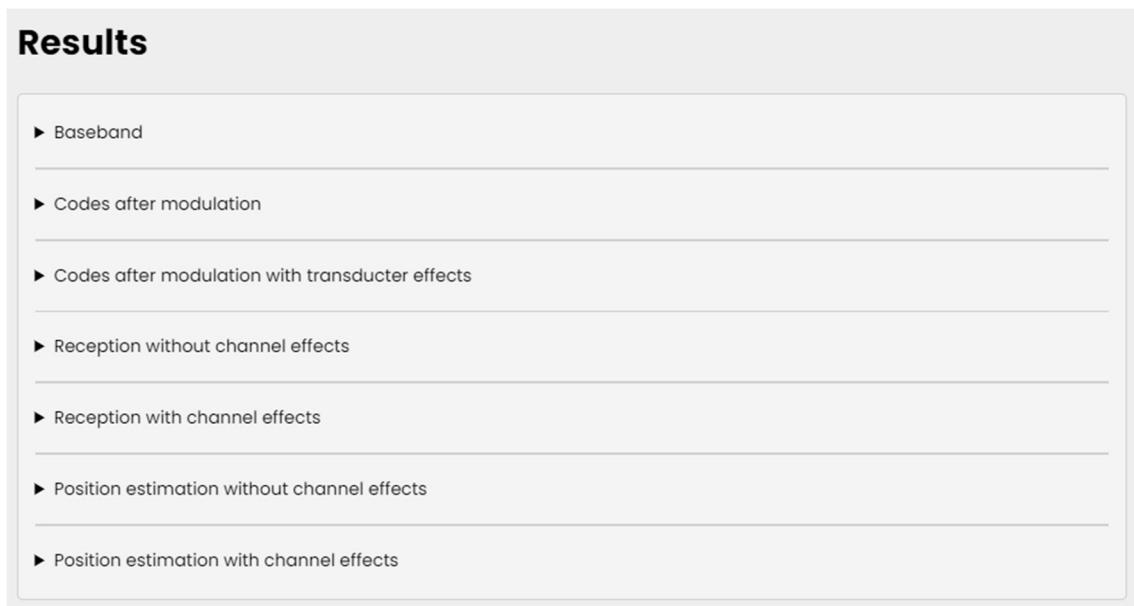


Figura 15 - Vista de Resultados

En la figura 16 se muestra la primera subsección **Baseband**. En este caso del ejemplo, presenta un mensaje indicando que no hay información para mostrar, debido a que desde el inicio se configuró la carga del código de una señal modulada.



Figura 16 - Vista de Resultados Baseband

La segunda subsección Codes after modulation se visualiza de la siguiente manera, como muestran las figuras 17 y 18:

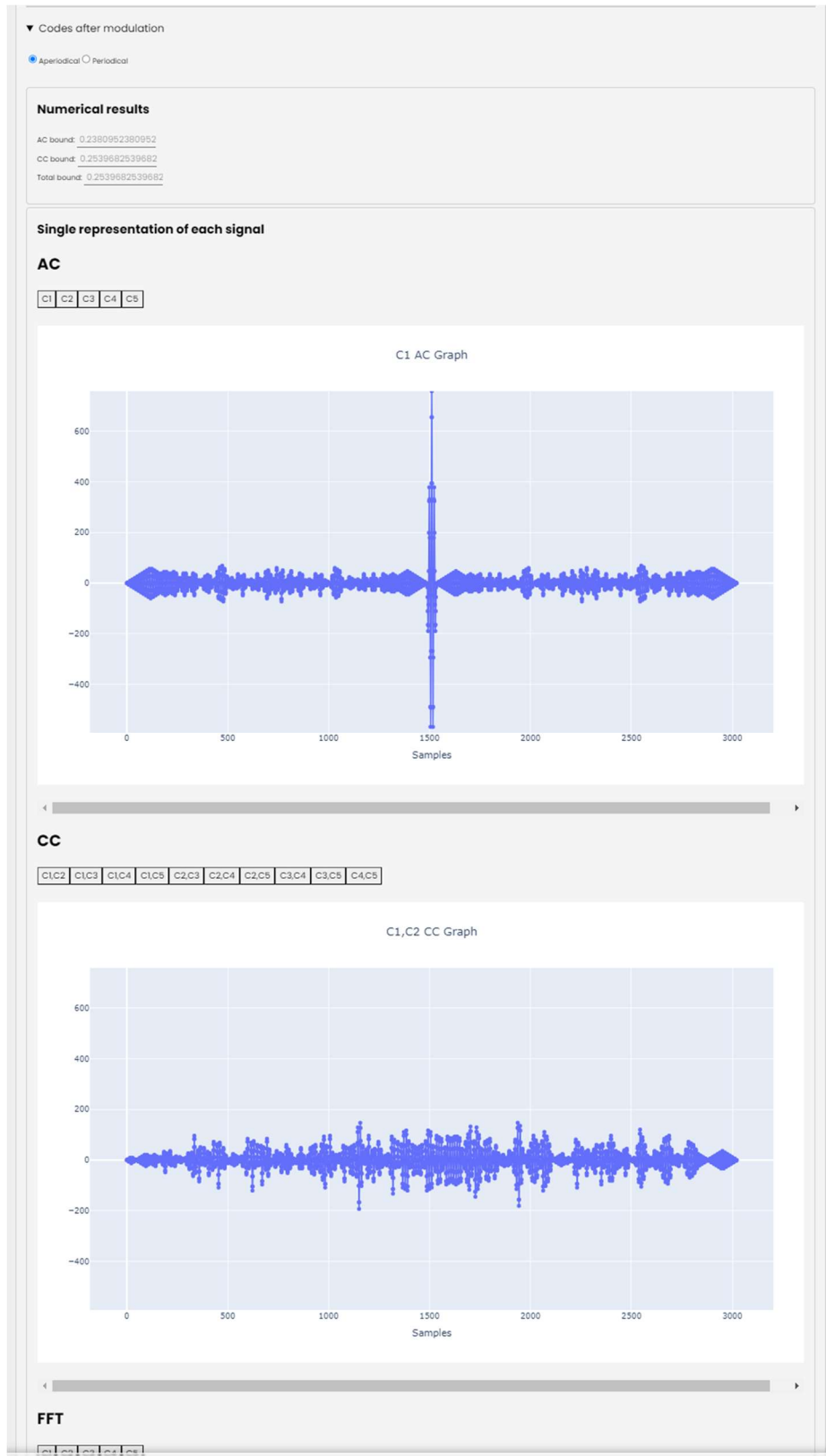


Figura 17 - Vista de Resultados Codes after modulation 1era parte

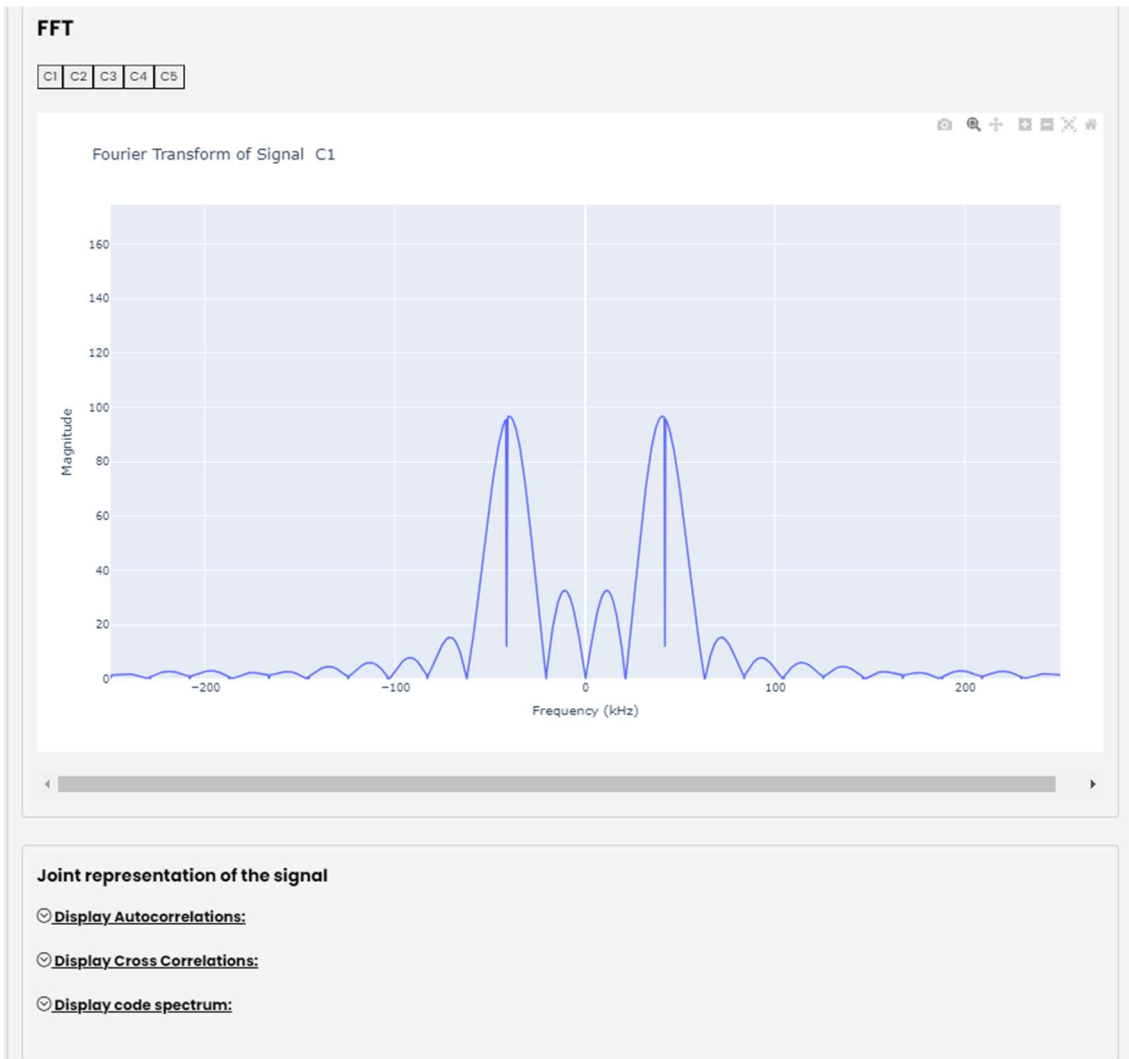


Figura 18 - Vista de Resultados Codes after modulation 2da

Luego en los links que se encuentran debajo, se pueden abrir a la vez todos los gráficos de cada cota para todas las señales de esta simulación, para tener el detalle de todos los emisores juntos. Como se observa en la figura 19:

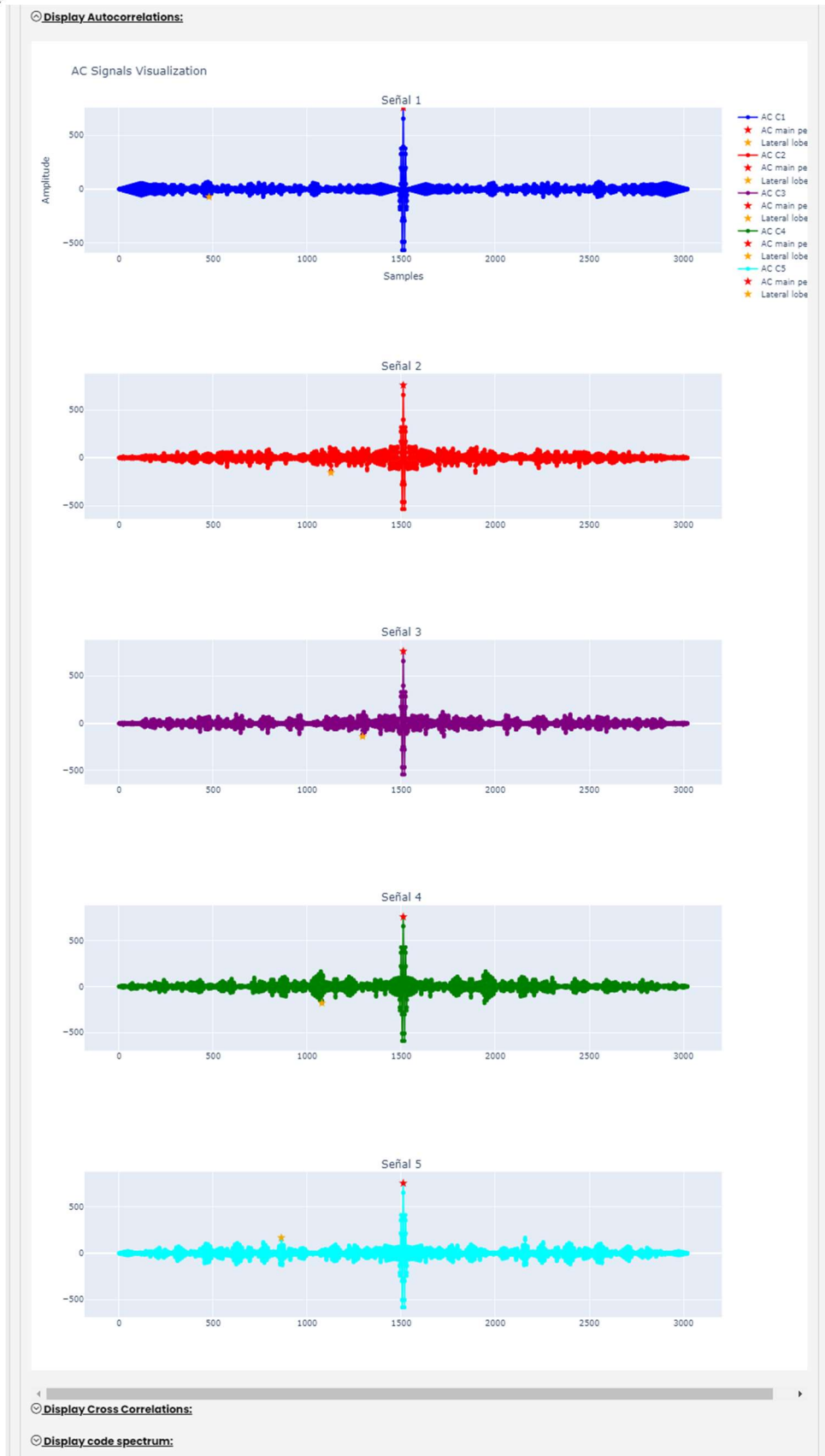


Figura 19 - Vista de Resultados Codes after modulation cotas de autocorrelación

La siguiente imagen, figura 20, es la tercera subsección y representa Codes after modulation with transducer effects

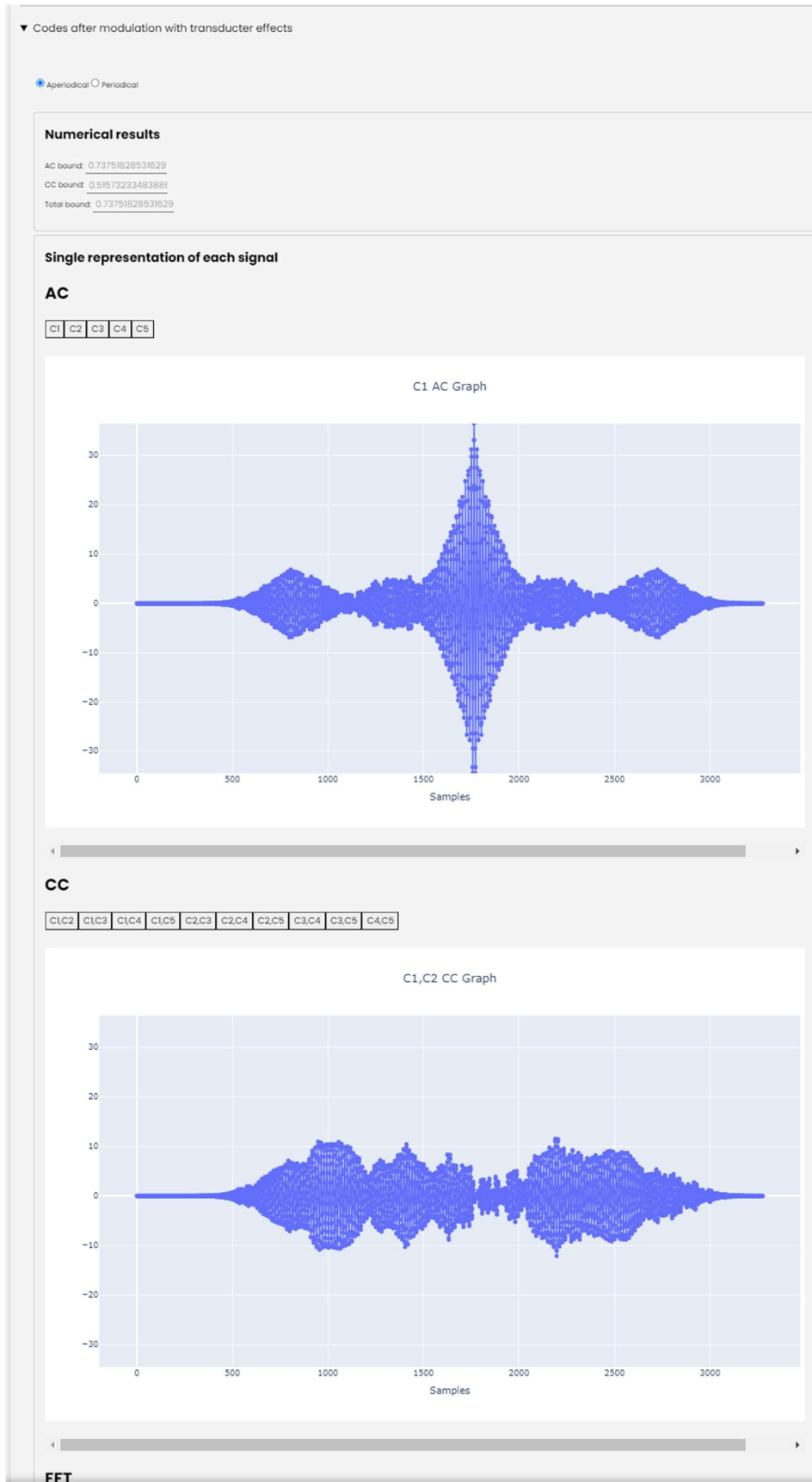


Figura 20 - Vista de Resultados Codes after modulation with transducer effects

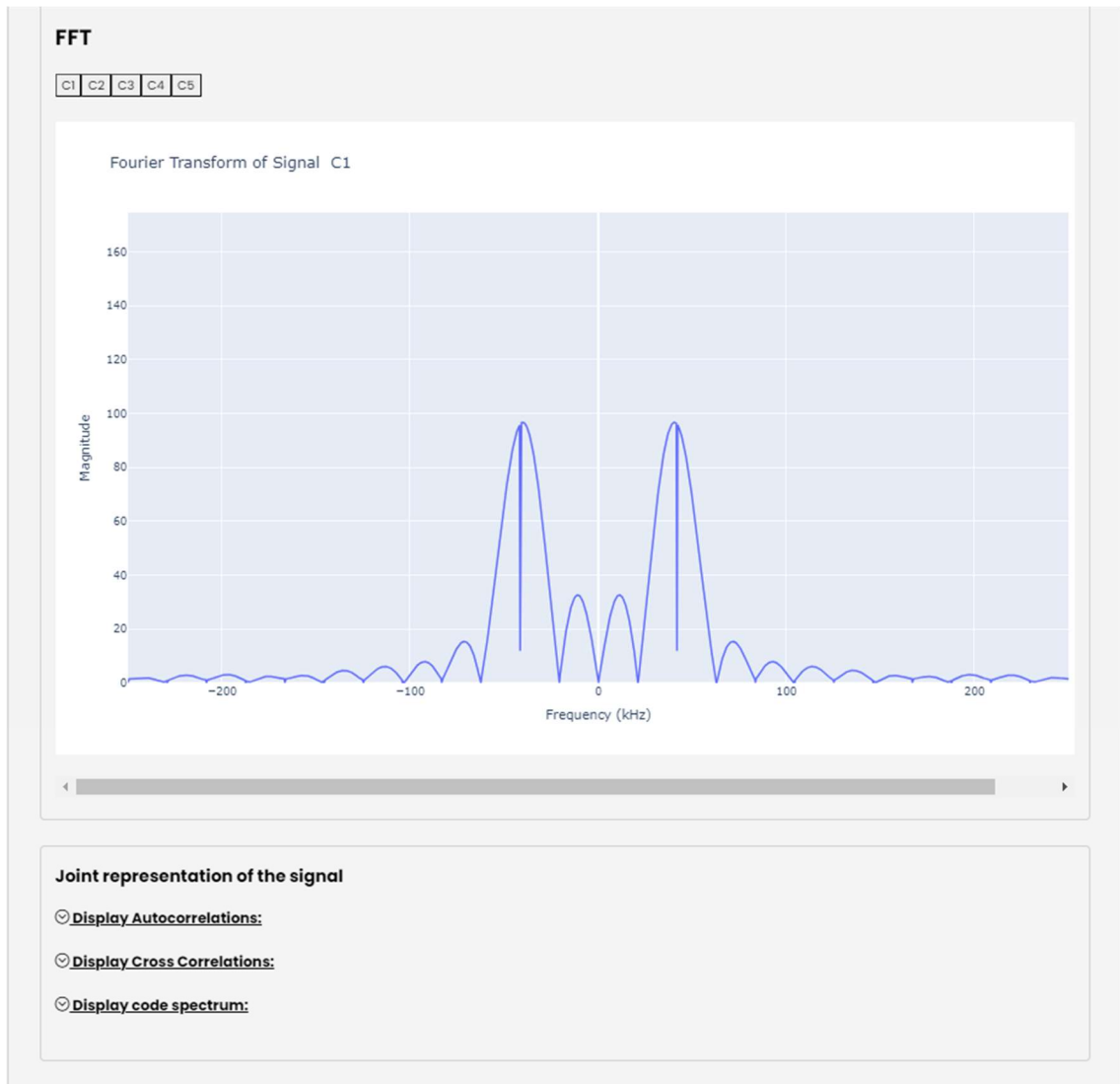


Figura 21 - Vista de Resultados Codes after modulation with transducer Effects 2

Luego se encuentra la sección de resultados de Reception without channel Effects, como se muestra en la figura 22:

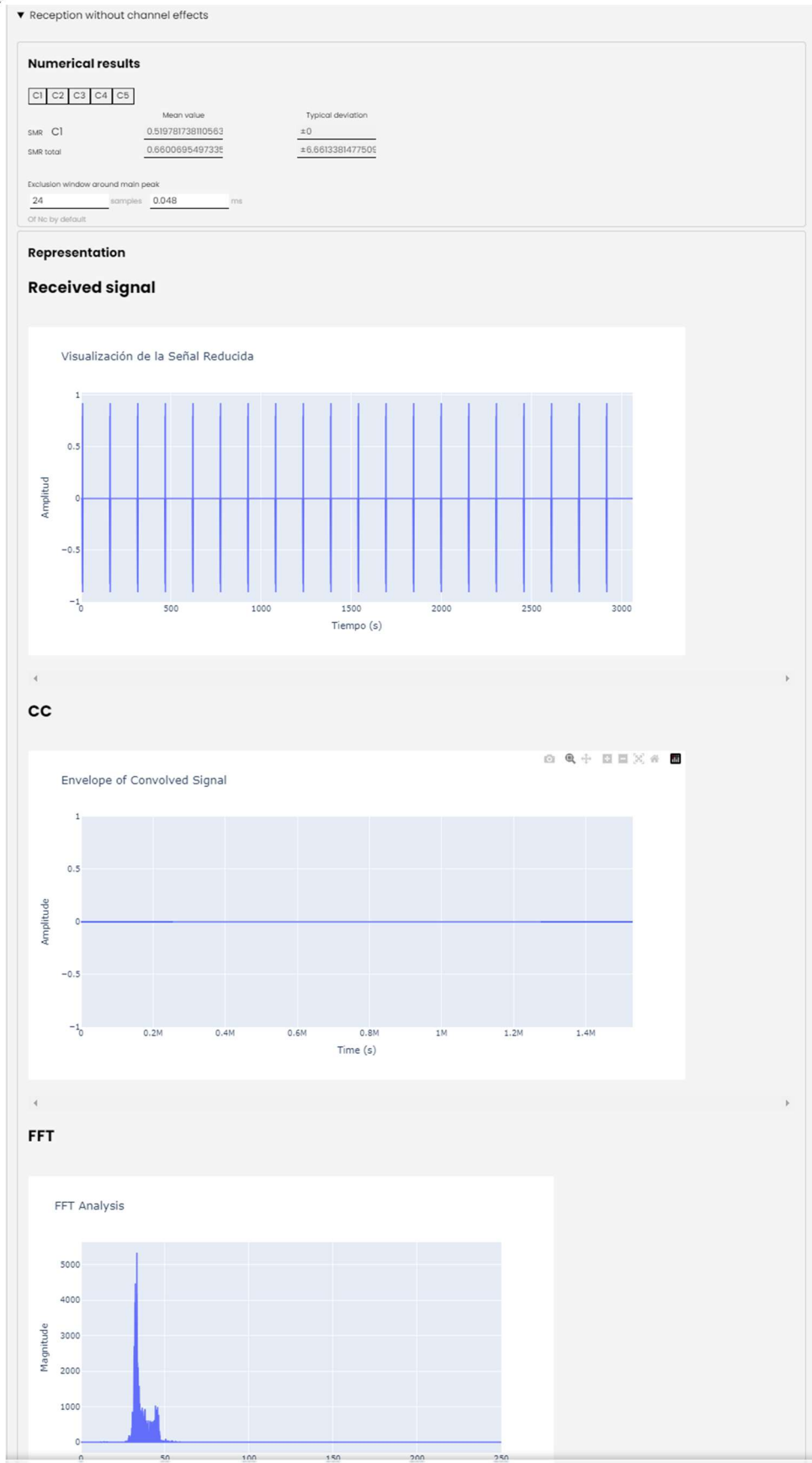


Figura 22 - Vista de Resultados Reception without channel effects

Además en la figura 23 se muestra la salida de Correlation Envelope para esta sección:

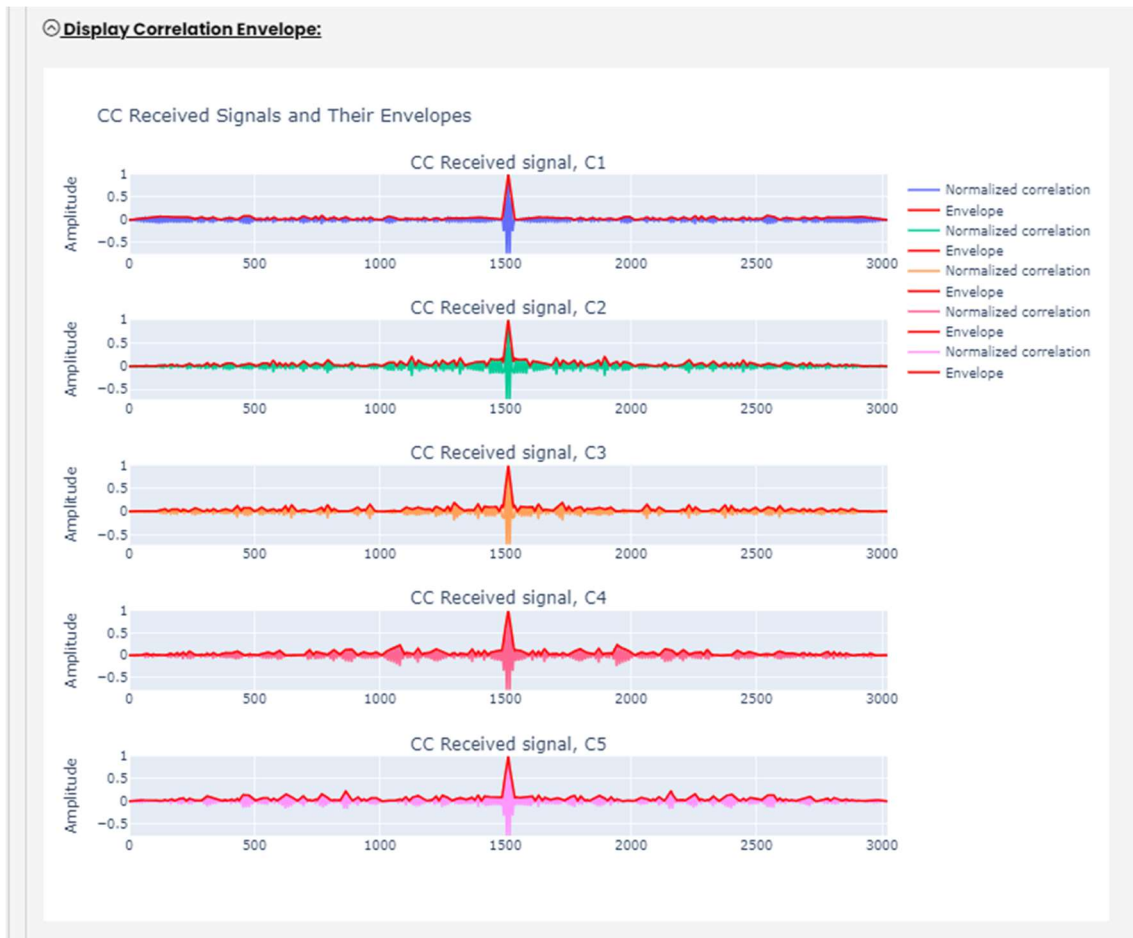


Figura 23 - Vista de Result. Recept. without channel effects (Correlation Envelope)

Avanzando a la resultados de Reception with channel Effects, se puede observar la figura 24:

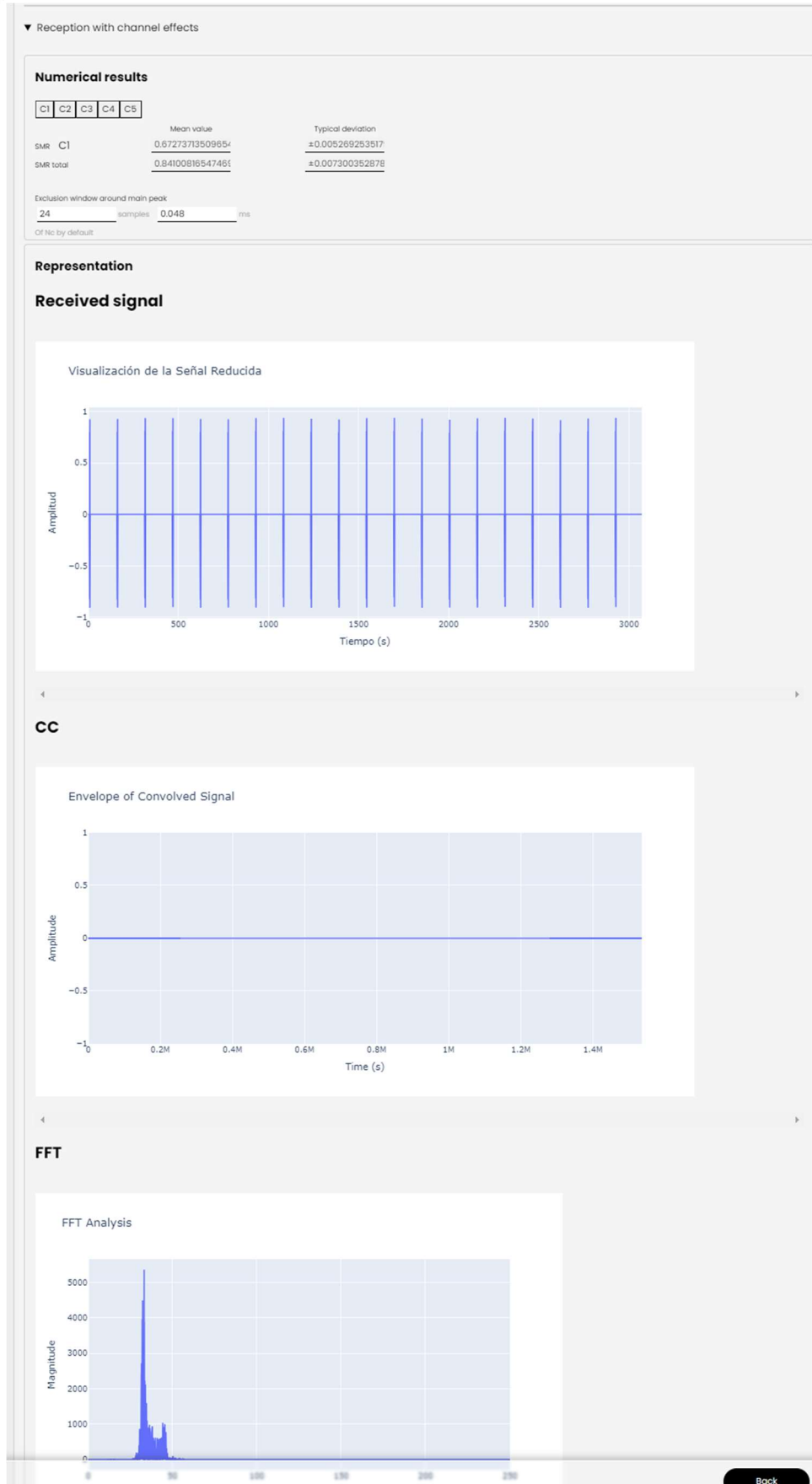


Figura 24 - Vista de Resultados Reception with channel effects

Finalmente las últimas secciones de los resultados son la estimación de la posición del receptor, primeramente sin canal de efectos como se muestra en la figura 25:

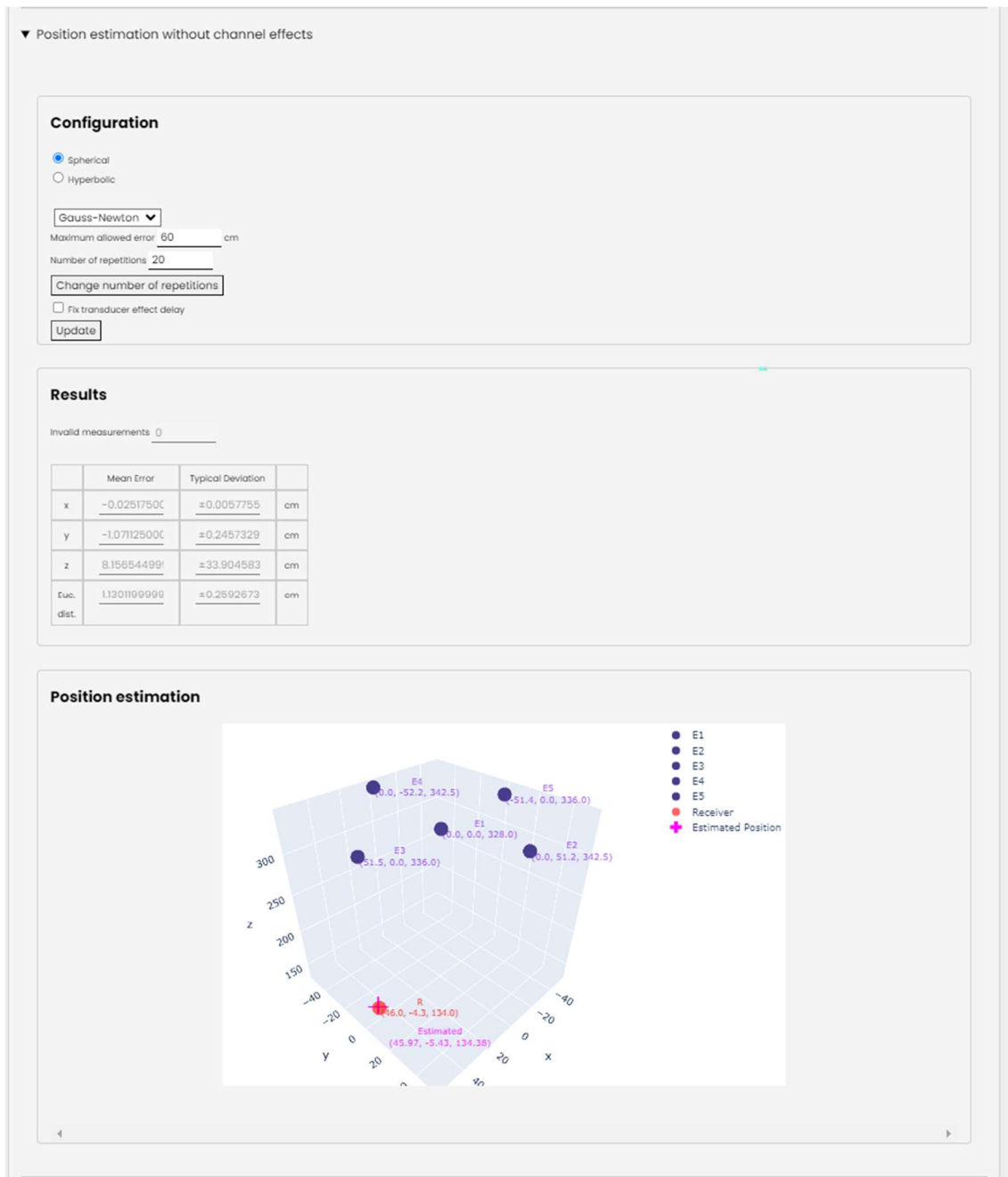


Figura 25 - Vista de Resultados Position estimation without channel Effects

Y en la figura 26 se observa la última sección del simulador, el resultado para la estimación de la posición del receptor teniendo en cuenta el canal de efectos.



Implementación

La aplicación web desarrollada ha sido nombrada CalcSignals, fue puesta en producción y puede ser accedida para su prueba mediante el siguiente enlace: <https://calcsignals.netlify.app/>

Usuario de prueba para acceder a la aplicación:

Tipo de usuario	Email	Contraseña
básico	testuser1@calcsignals.com	CalcSignals2024

Tabla 12 - Usuario para probar la aplicación

CAPÍTULO 6:

RESULTADOS, CONCLUSIONES Y TRABAJOS FUTUROS

En el capítulo anterior se ilustró un ejemplo completo del flujo de una simulación, desde su inicio hasta la obtención de los resultados finales, incluyendo la aplicación de efectos y el análisis de las distintas métricas. En este capítulo se presentan las conclusiones generales de la investigación, de la herramienta desarrollada, destacando sus características y se exploran las posibles líneas de trabajo futuro.

A lo largo de la tesina, se ha podido investigar y llegar a comprender conocimientos sobre el análisis de las señales ultrasónicas, así como métodos de codificación y modulación, sus frecuencias, implicaciones de transductores, efectos del medio en la propagación de señales, estrategias de reducción de interferencia, y métodos de localización basados en correlación, entre otros. Todo esto ha permitido un enfoque integral en el diseño del sistema resultante, considerando tanto los aspectos teóricos como los desafíos prácticos involucrados en su implementación.

Finalmente se pudo implementar una herramienta web interactiva que permite simular y analizar distintos escenarios en sistemas de localización por señales ultrasónicas. Gracias a su interfaz sencilla, es posible visualizar conceptos teóricos y comprender el impacto de factores diversos, como la configuración del transductor, la modulación utilizada y los efectos del medio.

La herramienta fue desarrollada utilizando tecnologías web modernas como Flask para construir una API en el *backend* y Angular para el *frontend*, lo que garantiza su flexibilidad y capacidad de expansión. Además, se ha empleado una variedad de bibliotecas de código abierto como NumPy y SciPy, que se especializan en cálculos matemáticos y aplicaciones científicas, facilitan el procesamiento de datos y cuentan con el respaldo de una comunidad extensa que contribuye al aprendizaje y mejora del desarrollo. También se integró la biblioteca Plotly, lo cual permitió la generación de gráficos interactivos en el frontend, ofreciendo visualizaciones claras y detalladas de los resultados de la simulación.

Entre las funcionalidades extras más destacadas se incluyen la gestión de usuarios mediante autenticación con tokens JWT, la posibilidad de almacenar proyectos y compartirlos con otros usuarios, lo que convierte a la plataforma en una herramienta poderosa para la colaboración. Estas características, junto a la gestión de simulaciones, hacen que la herramienta sea útil no solo en el ámbito educativo sino también para el desarrollo continuo de investigaciones en sistemas de localización tanto en espacios interiores como en subacuáticos.

Este desarrollo ha sido posible gracias al aprovechamiento de la base fundamental construida en el trabajo previo citado en [1], el cual sirvió como punto de partida para llevar esta herramienta a un nivel más accesible, moderno e intuitivo.

De cara al futuro, se vislumbran distintas líneas de trabajo que podrían mejorar las capacidades y el alcance de esta herramienta, que no pudieron ser desarrollados en este trabajo.

En primer lugar, se plantea la validación de la herramienta en entornos más desafiantes, como aplicaciones en entornos reales. También se propone ampliar la base



de datos de transductores, incorporando modelos adicionales y opciones personalizables, lo cual permitiría una mayor variedad de simulaciones y escenarios.

Otro posible avance consiste en la incorporación de técnicas de codificación dentro de la herramienta, como las secuencias Kasami, Zadoff-Chu y Loosely Synchronous. Además, se contempla la optimización del procesamiento de señales mediante algoritmos más eficientes, lo cual podría reducir tiempos de respuesta y haría la simulación más fluida.

También se contempla continuar mejorando la interfaz gráfica mediante la implementación de herramientas de visualización más poderosas, lo cual permitiría representar de manera más clara cada gráfico.

Otro posible futuro desarrollo, más abstracto quizás, es que en la plataforma construida se podrían desarrollar otros simuladores o herramientas de cálculo con otros propósitos que se pueden sumar a esta herramienta de simulación, gestionando permisos para el acceso a cada herramienta de estudio.

BIBLIOGRAFÍA

- [1] Vicente Ranera, J. de. (2022). Diseño de un simulador para sistemas de localización acústicos subacuáticos basados en secuencias de espectro ensanchado [Trabajo de Fin de Máster, Universidad de Alcalá].
- [2] Gilhotra, K., & Mahalik, N. P. (2014). Functional simulation platform for studying underwater radiation pattern. *American Journal of Engineering and Applied Sciences*, 7(3), 327-337. <https://doi.org/10.3844/ajeassp.2014.327.337>
- [3] Zia, M. Y. I., Otero, P., Siddiqui, A., & Poncela, J. (2021). Design of a web based underwater acoustic communication testbed and simulation platform. *Wireless Personal Communications*, 116(2), 1171-1193. <https://doi.org/10.1007/s11277-020-07825-3>
- [4] Dong, B., & Mahdy, A. M. (2010). Underwater wireless sensor networks: Efficient localization schemes using semidefinite programming. *International Journal on Advances in Networks and Services*, 3(1), 186-195.
- [5] Chandrasekhar, V., Seah, W. K. G., Choo, Y. S., & Ee, H. V. (2006). Localization in underwater sensor networks - Survey and challenges. In *Proceedings of the First Workshop on Underwater Networks* (pp. 33-40). ACM. <https://doi.org/10.1145/1161039.1161047>
- [6] Teymorian, A. Y., Cheng, W., Ma, L., Cheng, X., Lu, X., & Lu, Z. (2009). 3D underwater sensor network localization. *IEEE Transactions on Mobile Computing*, 8(12), 1610-1621. <https://doi.org/10.1109/TMC.2009.48>
- [7] Zeiger, V., Badri-Hoehner, S., & Hoehner, P. A. (2012). Impact of speed of sound uncertainties on model-based positioning. *Oceanology International*, 1-5.
- [8] Yang, J., He, L., & Shuai, C. (2016). The simulation of underwater acoustic propagation with the horizontal changes of sound speed profiles. In *2016 IEEE/OES China Ocean Acoustics (COA)* (pp. 1-4). IEEE. <https://doi.org/10.1109/COA.2016.7535887>
- [9] Aparicio, J., Jiménez, A., Álvarez, F. J., Ureña, J., De Marziani, C., De Diego, D., Cruz, N., & Campos, H. (2017). Accurate detection of spread-spectrum modulated signals. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation* (pp. 206-212).
- [10] Kasami, T. (1968). Weight distribution formula for some class of cyclic codes (Technical report R-285). Coordinated Science Lab., University of Illinois.
- [11] Stanczak, S., Boche, H., & Haardt, M. (2001). Are LAS-codes a miracle? In *IEEE Global Telecommunications Conference* (Vol. 1, pp. 589-593). <https://doi.org/10.1109/GLOCOM.2001.965588>
- [12] Murano, S., Pérez, M. C., Ureña, J., Bleakley, C. J., & De Marziani, C. (2018). Comparison of Zadoff-Chu encoded modulation schemes in an ultrasonic local positioning system. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation* (pp. 206-212). IEEE. <https://doi.org/10.1109/IPIN.2018.8533783>
- [13] Zhang, C., Yamada, S., & Hatori, M. (2005). General method to construct LS codes by complete complementary sequences. *IEICE Transactions on Wireless Communications Technologies*, E88-B(1), 52-58.
- [15] Prowave. (s.f.). Open type ultrasound transducer (328S160). *Prowave*. <http://www.prowave.com.tw/english/products/ut/open-type/328s160.htm> (Accedido el 15 de octubre de 2024).



- [16] Plotly. (s.f.). *Dash User Guide & Documentation*. <https://dash.plotly.com/> (Accedido el 15 de octubre de 2024).
- [17] Angular. (s.f.). *The modern web developer's platform*. <https://angular.dev/> (Accedido el 15 de octubre de 2024).
- [18] Python Software Foundation. (s.f.). *Python*. <https://www.python.org/> (Accedido el 15 de octubre de 2024).
- [19] NumPy Developers. (s.f.). *NumPy*. <https://numpy.org/> (Accedido el 15 de octubre de 2024).
- [20] SciPy Developers. (s.f.). *SciPy Documentation*. <https://docs.scipy.org/doc/scipy/> (Accedido el 15 de octubre de 2024).