



Universidad Técnica del Norte  
Facultad de Ingeniería en Ciencias Aplicadas  
Carrera de Ingeniería en Electrónica y Redes de Comunicación

---

**DISEÑO DE UNA APLICACIÓN ACADÉMICA  
PARA EL ESTUDIO DE REDES DE SENSORES  
INALÁMBRICOS SUBACUÁTICOS EN MATLAB**

---

*Trabajo de Grado previo a la obtención del título de*  
**Ingeniero en Electrónica y Redes de Comunicación**

**Autor:** Angel Jampier Velasco Suárez  
**Director:** Ing. Fabián Geovanny Cuzme Rodríguez MSc.  
**Asesor:** Ing. Hernán Mauricio Domínguez Limaico MSc.

**Ibarra - Ecuador**

**2022**

Documento maquetado con T<sub>E</sub>X<sup>I</sup>S v.1.0+.

Este documento está preparado para ser imprimido a doble cara.

# Diseño de una aplicación académica para el estudio de Redes de Sensores Inalámbricos Subacuáticos en MATLAB

*Trabajo de Grado previo a la obtención del título de*  
**Ingeniero en Electrónica y Redes de Comunicación**

*Elaborado por*

**Angel Jampier Velasco Suárez**

*Dirigido por el Magíster*

**Fabián Geovanny Cuzme Rodríguez**

**Carrera de Ingeniería en Electrónica y Redes de Comunicación**  
**Facultad de Ingeniería en Ciencias Aplicadas**  
**Universidad Técnica del Norte**

**Ibarra - Ecuador**

**2022**

Copyright © Angel Jampier Velasco Suárez



# UNIVERSIDAD TÉCNICA DEL NORTE

## BIBLIOTECA UNIVERSITARIA

### AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

#### 1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	0803901628		
APELLIDOS Y NOMBRES:	Velasco Suárez Angel Jampier		
DIRECCIÓN:	Quinindé - Calle Eleucipo Quiñónez y Melecia Quiñónez		
EMAIL:	ajvelascos@utn.edu.ec		
TELÉFONO FIJO:	062746076	TELÉFONO MÓVIL:	0980117116

DATOS DE LA OBRA	
TÍTULO:	Diseño de una aplicación académica para el estudio de Redes de Sensores Inalámbricos Subacuáticos en MatLab
AUTOR (ES):	Velasco Suárez Angel Jampier
FECHA: DD/MM/AAAA	12/12/22
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	Ingeniero en Electrónica y Redes de Comunicación
ASESOR /DIRECTOR:	Ing. Cuzme Rodríguez Fabián Geovanny MSc.

#### 2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 12 días del mes de diciembre de 2022

EL AUTOR:

Sr. Angel Velasco



**Universidad Técnica del Norte**  
**Facultad de Ingeniería en Ciencias Aplicadas**

**Certificación**

Magister Fabián Cuzme, Director del presente Trabajo de Titulación.

Que, el presente Trabajo de Titulación “Diseño de una aplicación académica para el estudio de Redes de Sensores Inalámbricos Subacuáticos en MatLab”. Ha sido desarrollado por el Sr. Velasco Suárez Angel Jampier, bajo mi supervisión.

Es todo en cuanto puedo certificar en honor de la verdad.

---

**Ing. Cuzme Rodríguez Fabián Geovanny MSc.**  
CI: 1311527012

# Dedicatoria

*A mi padre y a mi madre,  
a mi leal enamorada,  
y a aquellos nobles que  
siempre confiaron en mi.*

*Hoy un lector,  
mañana un líder.  
Margaret Fuller*

# Agradecimientos

*Cuando la gratitud es tan absoluta,  
las palabras sobran.*

Álvaro Mutis

Es innegable el inmensurable esfuerzo que he puesto en la investigación y desarrollo de este laborioso proyecto, pero este no habría sido posible sin la colaboración de ciertos personajes.

Quisiera expresar mi agradecimiento hacia mi tutor, Fabián Cuzme, quién a más de haber sembrado en mi la semilla de curiosidad en este maravilloso campo, como es el de redes subacuáticas, ha sido una gran guía en el proceso. Asimismo, no puedo dejar atrás el aporte de los ingenieros Mauricio Domínguez, Luis Suárez, Carlos Vásquez y Jaime Michilena, quienes supieron ofrecerme su ayuda cuando lo requería.

Especiales reconocimientos merecen mis amigos Anthony Andrango, Santiago Estévez y Alejandro Ortega, quienes no han dudado en otorgarme parte de su tiempo voluntariamente en incontables ocasiones a lo largo del progreso de este trabajo.

Me gustaría reconocer también la predisposición de los compañeros que han participado en la recopilación del material que ha servido de análisis y pruebas para mi proyecto.

Una persona a quien debo mucho es a mi pareja, Tatiana Toro, quien ha sido partícipe de este proyecto tanto académica como moralmente, ya

que nunca vaciló en ofrecerse para una tarea, y sus ánimos siempre lograron mantenerme lleno de vigor sin permitirme desmayar en el camino.

Por último pero no menos importante, agradezco profundamente a mis padres, Maritza y Roger, quienes sin duda alguna han tenido un papel insustituible en toda mi trayectoria académica; su paciencia, su apoyo, su confianza y sus consejos han sido la clave de mi desenvolvimiento en esta prolongada etapa, porque siempre tuve la total certeza de que estarían conmigo en las buenas y en las malas.

Para mis queridas abuelas, para mis tíos y mis primos, para mis amigos y mis compañeros, para esta infinita lista de individuos, les expreso mi infinita gratitud, porque cada uno ha forjado lo que soy ahora mismo.

A todos los mencionados, mis más sinceros agradecimientos, y quedan liberados de toda responsabilidad por los errores que presenten estos textos.

# Resumen

Una UWSN (Redes de Sensores Inalámbricos Subacuáticos), es un conjunto de sensores desplegados en un área del entorno subacuático capaces de recoger cierto tipo de información a su alrededor y transferirla hacia un destino a través de uno o varios saltos entre nodos. El aumento de popularidad en distintos sectores de las redes subacuáticas, hace que nazca la necesidad de una alternativa para el estudio de este tipo de redes sin disponer de los sensores.

El principal propósito del presente trabajo es diseñar una aplicación académica que sirva como herramienta de simulación de redes subacuáticas utilizando el lenguaje y entorno de programación MATLAB<sup>®</sup>. Para llevar a cabo esta meta, se pone en práctica la metodología del *Ciclo de Vida Clásico del Software*, que consta de 4 etapas: Análisis, Diseño, Implementación y Pruebas.

En la etapa de análisis se elabora el levantamiento de requerimientos basado en el estándar IEEE 830 y sus recomendaciones. Se utilizan dos técnicas de recopilación de requerimientos: encuesta hacia los potenciales usuarios del software que son los estudiantes de la materia de Comunicaciones Inalámbricas de la Carrera de Ingeniería en Telecomunicaciones y entrevista al docente de la misma materia, hay que destacar que las funcionalidades propuestas en los formularios de encuesta y entrevista fueron previamente discutidos y tomados en consideración por los implicados en el proyecto.

La etapa de diseño es probablemente la más extensa del proceso, siendo esta la parte donde se establecen los diagramas lógicos para efecto de que la etapa de implementación se realice de manera más eficiente. Por último, en la etapa de pruebas, nuevamente se presta de la disposición de los estudiantes de Comunicaciones Inalámbricas con el objetivo de probar y validar los requerimientos del software.

Con las respuestas obtenidas en la encuesta se concluye el proyecto haciendo una comparación con los niveles de interés y conocimiento obtenidos en la primera encuesta, como es de esperar, también se obtienen resultados de la satisfacción de los estudiantes con el desempeño de la aplicación después de haberla utilizado.

# Abstract

A UWSN (Underwater Wireless Sensor Networks) is a set of sensors deployed in an area of the underwater environment capable of collecting certain types of information around them and transferring it to a destination through one or several hops between nodes. The increase in popularity in different sectors of underwater networks, gives rise to the need for an alternative for the study of this type of network without having the sensors.

The main purpose of this work is to design an academic application that serves as a simulation tool for underwater networks using the MATLAB<sup>®</sup> language and programming environment. To carry out this goal, the Classical Software Life Cycle model is put into practice, which consists of 4 stages: Analysis, Design, Implementation and Testing.

In the analysis stage, the gathering of requirements is prepared based on the IEEE 830 standard and its recommendations. Two requirements gathering techniques are used: a survey of potential users of the software who are students of the Wireless Communications subject of the Telecommunications Engineering Career and an interview with the teacher of the same subject. It should be noted that the functionalities proposed in the survey and interview forms were previously discussed and taken into consideration by those involved in the project.

The design stage is probably the most extensive of the process, this being the part where the logical diagrams are established so that the implementa-

tion stage can be carried out more efficiently. Finally, in the testing stage, it is once again available to Wireless Communications students in order to test and validate the software requirements.

With the answers obtained in the survey, the project is concluded by making a comparison with the levels of interest and knowledge obtained in the first survey, as expected, results of student satisfaction with the performance of the application after using it is also obtained.

# Tabla de contenidos

<b>Autorización</b>	<b>V</b>
<b>Certificación</b>	<b>VI</b>
<b>Dedicatoria</b>	<b>VII</b>
<b>Agradecimientos</b>	<b>IX</b>
<b>Resumen</b>	<b>XI</b>
<b>Abstract</b>	<b>XIII</b>
<b>1 Antecedentes</b>	<b>1</b>
1.1 Tema . . . . .	1
1.2 Problema . . . . .	1
1.3 Objetivos . . . . .	3
1.3.1 Objetivo General . . . . .	3
1.3.2 Objetivos Específicos . . . . .	3
1.4 Alcance . . . . .	3
1.5 Justificación . . . . .	6
<b>2 Marco Teórico</b>	<b>9</b>
2.1 Comunicaciones Inalámbricas . . . . .	9
2.1.1 Redes de Sensores Inalámbricos (WSN) . . . . .	10
	<b>XV</b>

2.1.1.1	Características de una WSN . . . . .	12
2.2	Comunicación Acústica Subacuática . . . . .	14
2.2.1	Modelo del Canal . . . . .	15
2.2.1.1	SNR con respecto a la Potencia de Transmisión	15
2.2.1.2	SDR con respecto a SNR . . . . .	19
2.2.2	Método de la Ruta más Corta . . . . .	21
2.2.2.1	Distancia entre puntos . . . . .	21
2.2.2.2	Algoritmo de Dijkstra . . . . .	22
2.3	Redes de Sensores Inalámbricos Subacuáticos (UWSN) . . . . .	24
2.3.1	Características y desafíos que presenta una UWSN . . . . .	24
2.3.2	Componentes de una UWSN . . . . .	27
2.3.2.1	Componentes principales de un nodo ordinario	27
2.3.3	Arquitecturas de UWSNs . . . . .	28
2.3.3.1	UWSN Bidimensional . . . . .	29
2.3.3.2	UWSN Tridimensional . . . . .	30
2.3.3.3	UWSN Tridimensional con AUVs . . . . .	31
2.3.4	Aplicaciones de una UWSN . . . . .	31
2.4	Ingeniería de Software . . . . .	33
2.4.1	Ciclo de Vida Clásico del Software . . . . .	33
2.4.2	Herramientas de Desarrollo de Software . . . . .	34
2.5	Matlab . . . . .	35
2.5.1	Matlab en la Ingeniería . . . . .	35
2.5.2	App Designer . . . . .	37
2.5.3	Matlab Student Suit . . . . .	37
2.6	Trabajos Relacionados . . . . .	38
<b>3</b>	<b>Diseño y Desarrollo</b>	<b>41</b>
3.1	Análisis . . . . .	41

---

3.1.1	Recopilación de Requerimientos . . . . .	42
3.1.2	Descripción General . . . . .	43
3.1.2.1	Funciones del Producto . . . . .	43
3.1.2.2	Características de los Usuarios . . . . .	43
3.1.2.3	Suposiciones y Dependencias . . . . .	44
3.1.2.4	Requerimientos Futuros . . . . .	44
3.1.3	Requerimientos Específicos . . . . .	45
3.1.3.1	Interfaces Externas . . . . .	45
3.1.3.2	Funciones . . . . .	47
3.1.3.3	Atributos del Sistema . . . . .	47
3.2	Diseño . . . . .	48
3.2.1	Diagrama de casos de uso . . . . .	48
3.2.2	Diagramas de flujo . . . . .	49
3.2.3	Algoritmo del sistema . . . . .	55
3.3	Implementación . . . . .	56
3.3.1	Codificación . . . . .	56
3.3.2	Interfaces gráficas de usuario . . . . .	61
3.3.3	Empaquetado del sistema . . . . .	64
<b>4</b>	<b>Pruebas y Resultados</b>	<b>67</b>
4.1	Uso del Software . . . . .	67
4.2	Taller y encuesta . . . . .	70
4.2.1	Guías de Laboratorio . . . . .	70
4.2.2	Encuesta . . . . .	72
4.2.3	Tabulación de la Encuesta . . . . .	73
4.3	Discusión . . . . .	75
4.4	Trabajos futuros . . . . .	76
	Conclusiones . . . . .	77

---

Recomendaciones . . . . .	79
<b>A Formularios de Recopilación de Requerimientos</b>	<b>81</b>
A.1 Encuesta acerca de Redes de Sensores Inalámbricos Subacuá- ticos . . . . .	81
A.2 Entrevista al Docente sobre Redes de Sensores Inalámbricos Subacuáticos . . . . .	84
<b>B Tabulación de Respuestas Recopiladas</b>	<b>87</b>
B.1 Tabulación de Encuesta . . . . .	87
B.2 Respuestas de Entrevista . . . . .	89
<b>C Casos de Uso</b>	<b>93</b>
<b>D Manual de Usuario de SARAS</b>	<b>103</b>
D.1 Introducción . . . . .	103
D.2 Instalación de SARAS . . . . .	104
D.2.1 Requisitos de Hardware . . . . .	104
D.2.2 Requisitos de Software . . . . .	104
D.2.3 Proceso de Instalación . . . . .	105
D.2.4 Previo a la simulación . . . . .	107
D.3 Diagrama general de SARAS . . . . .	108
D.4 Pasos para realizar una simulación en SARAS . . . . .	108
D.4.1 Crear o Importar sensores . . . . .	109
D.4.2 Ejecutar protocolo de enrutamiento . . . . .	110
D.4.3 Transmitir datos . . . . .	110
D.4.4 Generar reporte . . . . .	111
D.4.5 Miscelánea . . . . .	111
D.5 Desinstalación de SARAS . . . . .	112

---

<b>E</b>	<b>Guías de Laboratorio</b>	<b>115</b>
E.1	Guía de Laboratorio 1 . . . . .	115
E.2	Guía de Laboratorio 2 . . . . .	120
<b>F</b>	<b>Formulario de Validación de Requerimientos</b>	<b>127</b>
	<b>Referencias</b>	<b>131</b>
	<b>Glosario</b>	<b>137</b>
	<b>Acrónimos</b>	<b>139</b>



# Índice de figuras

1.1	Arquitectura tentativa para el diseño de la aplicación. . . . .	5
2.1	Red de Sensores Inalámbricos. . . . .	13
2.2	Distancia entre dos puntos en el plano cartesiano. . . . .	22
2.3	Grafo de nodos con aristas con peso. . . . .	23
2.4	Componentes principales de un nodo ordinario. . . . .	27
2.5	Arquitectura bidimensional de una UWSN. . . . .	30
2.6	Arquitectura tridimensional estática de una UWSN. . . . .	31
2.7	Arquitectura tridimensional de una UWSN con AUVs. . . . .	32
2.8	Etapas de la metodología del Ciclo de Vida Clásico del Software. . . . .	34
3.1	Diagrama de casos de uso del sistema SARAS. . . . .	49
3.2	Diagrama de flujo del CU01. . . . .	50
3.3	Diagrama de flujo del CU02. . . . .	52
3.4	Diagrama de flujo del CU03. . . . .	52
3.5	Diagrama de flujo del CU04. . . . .	53
3.6	Diagrama de flujo del CU05. . . . .	54
3.7	Diagrama de flujo del CU06. . . . .	54
3.8	Componentes de la ventana principal de SARAS. . . . .	62
3.9	Componentes de la ventana Crear Sensores de SARAS. . . . .	63
3.10	Empaquetado del software SARAS. . . . .	64
4.1	Diagrama de bloques general del uso de SARAS. . . . .	68

---

4.2	Generación aleatoria de 10 nodos en un área de 100 x 150 [m].	68
4.3	Datos de sensores exportados en un archivo <code>.csv</code> .	69
4.4	Plot de distribución de 50 nodos con una ruta entre el nodo 19 y 3.	69
4.5	Reporte de simulación exitosa de 1500 nodos.	70
4.6	Diagrama de bloques de Laboratorio 1.	71
4.7	Diagrama de bloques de Laboratorio 2.	71

# Índice de Tablas

2.1	Diferencias entre UWSN y TWSN. . . . .	12
2.2	Comparación de las tecnologías de radiofrecuencia, óptica y acústica. . . . .	15
3.1	Listado de Requerimientos del Sistema. . . . .	46
3.2	Requerimientos de Hardware de SARAS. . . . .	65
C.1	Caso de Uso: Crear Sensores. . . . .	93
C.2	Caso de Uso: Ingresar Parámetros de los Nodos. . . . .	96
C.3	Caso de Uso: Ejecutar Protocolo de Enrutamiento. . . . .	97
C.4	Caso de Uso: Transmitir Datos. . . . .	98
C.5	Caso de Uso: Desplegar Resultados. . . . .	99
C.6	Caso de Uso: Editar Parámetros. . . . .	100



# Índice de Ecuaciones

2.1	Relación Señal a Ruido por bit . . . . .	15
2.2	Nivel de Señal . . . . .	16
2.3	Potencia de Transmisión . . . . .	16
2.4	Intensidad de Señal . . . . .	17
2.5	Nivel de Señal con respecto a la Potencia de Transmisión . . . . .	17
2.7	Coefficiente de Absorción (Fórmula de Thorp) . . . . .	17
2.8	Propagación Esférica de la Señal Acústica . . . . .	18
2.9	Pérdida de Transmisión . . . . .	18
2.10	Nivel de Ruido . . . . .	18
2.11	SNR con respecto a la potencia y absorción . . . . .	19
2.12	SNR con respecto a la Potencia, Distancia de transmisión y Frecuencia . . . . .	19
2.13	Tasa de Bits Errados . . . . .	20
2.14	Relación de Entrega Exitosa de un Bit . . . . .	20
2.15	Relación de Entrega Exitosa de $m$ Bits . . . . .	20
2.16	Distancia entre dos puntos en el plano . . . . .	22
2.17	Distancia entre dos puntos un espacio tridimensional . . . . .	22



# Índice de Códigos

3.1	Definición de variables. . . . .	57
3.2	Establecimiento de valor de semilla. . . . .	57
3.3	Creación de tabla de datos de nodos. . . . .	58
3.4	Generación aleatoria de posición de nodos. . . . .	58
3.5	Cálculo de distancia euclidiana entre nodos. . . . .	58
3.6	Construcción de grafo. . . . .	59
3.7	Ploteo de distribución de nodos. . . . .	59
3.8	Enrutamiento entre nodos. . . . .	60
3.9	Transmisión de datos entre nodo origen y destino. . . . .	60



# Capítulo 1

## Antecedentes

**RESUMEN:** Para comenzar el documento, se ofrece al lector una puesta en contexto de qué es lo que incita al desarrollo del proyecto, qué objetivos se han propuesto, de qué manera se plantea realizar el proceso de elaboración y cuál es el motivo por el que se necesita este trabajo.

### 1.1. Tema

Diseño de una aplicación académica para el estudio de Redes de Sensores Inalámbricos Subacuáticos en MatLab

### 1.2. Problema

Tomando en cuenta el incremento considerable del uso de sensores conectados a internet en los últimos años, no se ha quedado atrás el despliegue de dispositivos inalámbricos bajo el agua, este tipo de red se denomina UWSN (*Underwater Wireless Sensor Network*, Red de Sensores Inalámbricos Subacuáticos), y consiste en una red de nodos ubicados de manera estratégica en ciertos puntos del océano (Hattab et al. 2013; Mohamed et al. 2011), siendo encabezados por uno o varios nodos, normalmente superficiales o comprendidos en vehículos autónomos, de mayor robustez (Haque et al. 2020).

Hoy en día es una tecnología que puede desenvolver un sinnúmero de aplicaciones en el ámbito de redes de comunicación, y por supuesto, los nodos necesitan una transmisión constante de datos entre sí, con el objeto de coordinar su posición exacta mientras se encuentran sumergidos (Tan et al. 2011). Evidentemente, al hablar de sensores colocados en aguas profundas, el uso de tecnologías inalámbricas a las que estamos acostumbrados, tales como IEEE 802.11, acaban siendo inviables; de modo que se utilizan señales acústicas (Bahrami et al. 2016) como las portadoras de los datos transmitidos. Sumando a estos factores, que la eficiencia del canal acuático puede variar con respecto a la localización de los nodos, provocan que la idea de experimentar con las UWSN se convierta en una situación caótica (Menna et al. 2016; Tan et al. 2011).

El proyecto nace con el propósito de ofrecer una alternativa de estudio de las redes de sensores inalámbricos subacuáticos. Académicamente hablando, se obtendrá como resultado una herramienta que no solo aporte a la comprensión de las arquitecturas utilizadas en las UWSN, sino que también ofrezca variedad de escenarios en simulación, utilizando como referencia ciertos parámetros constituidos por diferentes variables físicas del agua en localidades de la Zona 1 de la república del Ecuador.

Resumiendo, el hecho de que se hayan implementado tantos sensores y se encuentre a flote el concepto de IoUT (*Internet of Underwater Things*, Internet de las Cosas Subacuáticas) (García et al. 2020; Liou et al. 2018), nos fuerza a los especializados en Redes de Comunicación y Telecomunicaciones, esencialmente en el área de Comunicaciones Inalámbricas, a indagar y estar al corriente en la tecnología UWSN. Bajo este criterio, se toma la iniciativa para formar parte del avance tecnológico en este tipo de redes.

## 1.3. Objetivos

### 1.3.1. Objetivo General

Diseñar una aplicación académica como herramienta de simulación, utilizando la plataforma de programación MATLAB con el propósito de enriquecer la comprensión y el estudio de Redes de Sensores Inalámbricos Subacuáticos (UWSN) en la materia de Comunicaciones Inalámbricas.

### 1.3.2. Objetivos Específicos

- Identificar documentación de interés cuyo contenido aporte al desarrollo del proyecto, haciendo uso de centros de repositorios académicos digitales para su posterior utilización.
- Definir los requerimientos necesarios para el funcionamiento de la aplicación académica, considerando la metodología del ciclo de vida clásico de software.
- Desarrollar el sistema en base a los requerimientos establecidos.
- Validar el funcionamiento de la herramienta a través de una etapa de pruebas.
- Elaborar los recursos oportunos para asegurar el uso adecuado del software por parte del usuario.

## 1.4. Alcance

El proyecto tendrá como propósito facilitar el estudio de las Redes de Sensores Inalámbricos Subacuáticos a través de una aplicación desarrollada en la plataforma de programación y cálculo numérico MATLAB como un entorno gráfico, intuitivo y didáctico, para efectuar satisfactoriamente la simulación de los escenarios UWSN, con el objeto de incentivar a los interesados en el área de Comunicaciones Inalámbricas a experimentar y estudiar tecnologías

emergentes tales como UWSN o IoUT. Partiendo de esta premisa, se propone la ejecución del proyecto en 3 de las fases recomendadas por el PMBOK (Project Management Body of Knowledge): planificación, ejecución y cierre.

Para comenzar, en la fase de planificación se busca identificar tanto documentación teórica en general, como un conjunto de algoritmos previamente desarrollados por la comunidad, haciendo uso de foros y centros de repositorios académicos para su posterior utilización en el programa que se va a diseñar; y al mismo tiempo se definirán el número de guías de laboratorio que se van a elaborar.

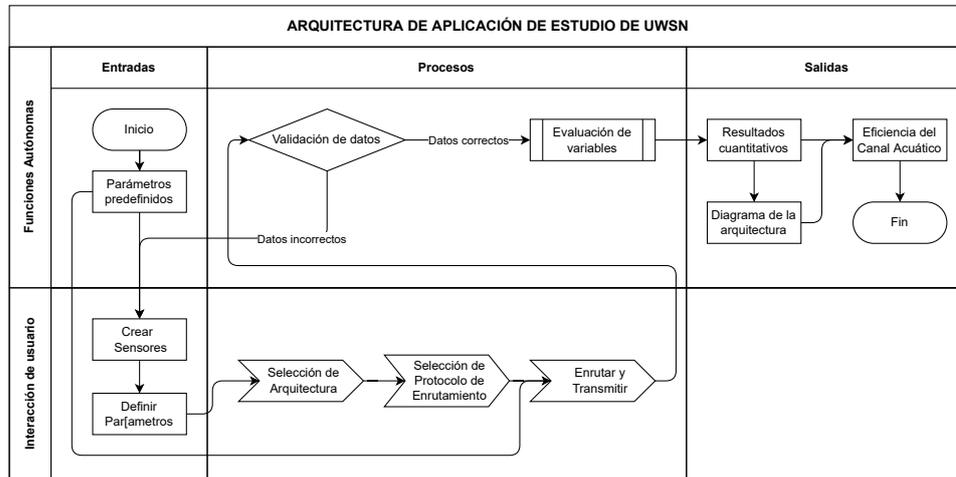
Para la fase de ejecución del programa se utilizará el método del ciclo de vida clásico de software, más conocido como metodología en cascada, que comprende lo siguiente: análisis, diseño, implementación y pruebas del sistema (Prokopets, 2020).

En la etapa de análisis se concretarán los requerimientos y características con las que el software deberá contar, se identificarán las variables físicas del agua que se tomarán en cuenta en el proceso de cálculo matemático que efectuará el algoritmo, así como parámetros de posición de los sensores en un área determinada, y además se decidirá el tipo o tipos de arquitecturas UWSN que el sistema será capaz de simular. Es preciso señalar que no se tomará como referencia un modelo de sensor inalámbrico en específico, sino que todos los resultados presentados en el programa serán independientes a las características de los sensores que se utilicen; dicho de otro modo, la aplicación se encargará de simular las redes en base a sensores inalámbricos subacuáticos genéricos.

En la etapa de diseño se establecerá la lógica con la que el producto trabajará, se creará un modelo que explique gráficamente la estructura de la programación y un diagrama de casos de uso explicando las opciones que la

**Figura 1.1**

*Arquitectura tentativa para el diseño de la aplicación.*



aplicación proporcionará, en las cuales tendrá lugar la interacción del usuario.

En la etapa de implementación se consideran un par de aspectos importantes: el primero es el diseño físico de codificación propiamente, este se realizará de manera que los tiempos de respuesta sean mínimos, es decir, por consola y al mismo tiempo con gráficas ploteadas con pocos detalles. Indudablemente, el diseño físico tendrá que acogerse al diseño lógico planteado en la etapa anterior. El segundo criterio por tomar en cuenta es el diseño físico de la interfaz gráfica, que pese a ser un apartado trivial en el desempeño de las simulaciones, se destaca por ser la imagen estética de la aplicación, y en muchas ocasiones es determinante al momento de llamar la atención del usuario; es preciso insistir en que esta interfaz debe ser intuitiva para el interesado, con el afán de facilitar el entendimiento del programa. Para la completa implementación del proyecto se utilizará MATLAB<sup>®</sup> Student Suite como lenguaje de programación y a su vez como entorno de programación.

En la etapa de pruebas además de confirmar la implementación de todos

los requerimientos propuestos con anterioridad, se tomarán en cuenta ciertas características propuestas por el modelo de calidad del producto definido por la ISO/IEC 25010, cuya taxonomía propone atributos y subatributos de calidad de los cuales se seleccionarán los ideales para medir el desempeño del producto final, de modo que se elabore una encuesta con el propósito de conocer la experiencia de uso de los estudiantes. Evidentemente, no se hará mayor hincapié en los ámbitos de menor importancia. Es importante recalcar que, según los resultados obtenidos en esta etapa, es posible que se modifique el programa e incluso el Análisis y/o Diseño.

Por último, en la fase de cierre se proyecta la elaboración de los insumos que el usuario necesita para utilizar de manera óptima la herramienta proporcionada, en estos se incluyen:

- Guía de instalación de la aplicación.
- Manual de utilización de la aplicación.
- Guías de laboratorio.

Se dará por concluido el proyecto una vez se complete la elaboración de todos los entregables y se reciba una aceptación por parte de los estudiantes encuestados.

## **1.5. Justificación**

En los últimos años la sociedad ha tendido a actualizarse en términos de la vida digital y el acceso universal, esto conlleva a que emerjan nuevas tecnologías de comunicación (Gvirtz & Necuzzi, 2011), y por consiguiente que despierte el interés no solo de especialistas en el área sino también de personas naturales que de cierto modo sienten curiosidad por la manera en que se transmiten datos alrededor del mundo.

El Instituto Oceanográfico y Antártico de la Armada de Ecuador se encarga de garantizar la seguridad a la navegación, el desarrollo, defensa y soberanía del territorio marítimo ecuatoriano, donde comprende también el mar de la provincia de Esmeraldas (Anchaluisa, 2015; INOCAR, 2021), perteneciente a Zona 1 de la República. La Universidad Técnica del Norte como parte de Zona 1, es en parte responsable de generar aportes que favorezcan al desarrollo del territorio mencionado.

Tomando en cuenta el inminente incremento de sensores inalámbricos submarinos en el mundo y los numerosos sistemas que utilizan UWSN como tecnología primordial en una cantidad considerable de aplicaciones en zonas costeras u oceánicas (Chicaiza, 2009); tanto los estudiantes interesados en este tipo de redes como los docentes con ánimos de ofrecer una clara explicación sobre UWSN, se ven en la necesidad de adquirir una herramienta de enseñanza-aprendizaje que facilite el desenvolvimiento de sus ideas.

Cujilema (2015) menciona la importancia que tienen los simuladores y su indispensable participación en el estudio de UWSN, ya que permiten evaluar diferentes arquitecturas o parámetros y al mismo tiempo probar la funcionalidad de ciertos protocolos contando con beneficios como la facilidad de uso, bajo costo y el amplio abanico de posibilidades que ofrece. Este trabajo pretende no solo aportar al estudio de las redes de sensores inalámbricos subacuáticos, sino llevarlo a cabo con una alta precisión de cálculo para obtener resultados confiables y muy cercanos a la exactitud real utilizando algoritmos matemáticos (Sotnik et al. 2020), por este motivo se ha seleccionado el entorno de MATLAB como software de programación (Yu, 2018).



## Capítulo 2

# Marco Teórico

**RESUMEN:** Este capítulo describe los temas principales que directa o indirectamente se relacionan con el proyecto y al mismo tiempo permite partir por una idea clara de los conceptos técnicos que se tocarán en la redacción del documento.

### 2.1. Comunicaciones Inalámbricas

Cuando se habla de comunicación inalámbrica, uno de los primeros pensamientos que se idealizan es la tecnología Wi-Fi (*Wireless Fidelity*, Fidelidad Inalámbrica), el famoso estándar IEEE 802.11 (Stallings, 2004, Ch. 17), pero a decir verdad, Wi-Fi es una invención relativamente reciente, tomando en cuenta que las comunicaciones inalámbricas se vienen estudiando desde finales del siglo XVIII (Molisch, 2012).

Hoy en día se reconocen innumerables tecnologías cuya transmisión de datos se realiza de forma inalámbrica, sin ir muy lejos, el continuo crecimiento de la comunicación celular inalámbrica, troncalización de radio, WLAN (*Wireless Local Area Networks*, Redes de Área Local Inalámbricas), PAN (*Personal Area Networks*, Redes de Área Personal), Ad hoc, etc. (Molisch, 2012).

La comunicación inalámbrica tiene gran popularidad en la actualidad, ya que ofrece numerosas ventajas cotidianamente, como su versatilidad, su facilidad de instalación y configuración, la movilidad del usuario y en muchos casos el costo (Stallings, 2004, p. 17).

### 2.1.1. Redes de Sensores Inalámbricos (WSN)

Goldsmith (2005) define WSN (*Wireless Sensor Network*, Red de Sensores Inalámbricos) de la siguiente manera: “Las redes de sensores inalámbricos consisten en pequeños nodos con capacidades de detección, computación y redes inalámbricas, como tal, estas redes representan la convergencia de tres tecnologías importantes. Las redes de sensores tienen un enorme potencial tanto para el consumidor como para aplicaciones militares” (p. 502). Estas pueden ser usadas para identificar y rastrear objetivos, detectar varios tipos de ataques o el movimiento de vehículos robóticos. En el sector industrial puede ser aprovechado en el control de edificios (controlando el aire acondicionado, las luces, etc.), automatización de fábricas, vigilancia, etc. (Molisch, 2012, p. 15). Por último, un área en el que su despliegue puede tener un gran potencial es en el control de grandes accidentes, por ejemplo, el colapso de puentes, pasarelas y balcones o la localización de supervivientes atrapados (Goldsmith, 2005).

El diseño de WSN depende del entorno, el objetivo de las aplicaciones, el costo, el hardware y las limitaciones del sistema, como por ejemplo: energía limitada, escasez de rango de comunicación, escasez de ancho de banda, y por último pero no menos importante, procesamiento y almacenamiento limitados en cada nodo. El entorno determina los factores de las redes como el tamaño, la topología y los esquemas (El-Rabaie et al. 2015). El-Rabaie et al. describen cinco tipos de WSN que se mencionan a continuación:

**WSNs Terrestres.** Consisten en una serie de nodos de sensores inalámbricos económicos desplegados en un área determinada. En esencia, se trata de la ya mencionada WSN, pero también se puede especificar como TWSN (*Terrestrial Wireless Sensor Network*, Red de Sensores Inalámbricos Terrestres).

**WSNs Subterráneas.** Varios nodos desplegados bajo tierra para sentir las condiciones circundantes. Además de eso, se implementa un nodo sink<sup>1</sup> para recopilar estos datos detectados en la estación base.

**WSNs Subacuáticas.** Los nodos y los vehículos de los sensores forman las redes utilizadas para monitorear las condiciones submarinas.

**WSNs Multimedia.** Consisten en un número de nodos sensores equipados con cámaras y micrófonos. Se utiliza para monitorear y rastrear eventos en forma de video, audio e imágenes.

**WSNs Móviles.** Consisten en nodos que se mueven por sí mismos los cuales tienen la capacidad de reposicionarse y organizarse en la red.

A pesar de que las UWSNs son una variante de WSNs, los mecanismos de comunicación utilizados tanto para *Underwater Wireless Sensor Network*, Red de Sensores Inalámbricos Subacuáticos, como para *Terrestrial Wireless Sensor Network*, Red de Sensores Inalámbricos Terrestres, divergen gracias a las características únicas que cada modelo ofrece (Han et al. 2015). En la Tabla 2.1 se explican ciertas diferencias entre ambos tipos de redes de sensores.

---

<sup>1</sup>También llamado sumidero o colector, obtiene datos de otros nodos y los reenvía con destino al gateway. Ver Figura 2.1.

<sup>2</sup>Los acrónimos BER y PLR hacen referencia a la *Tasa de Bits Errados* y a la *Tasa de Pérdida de Paquetes* en una transmisión, respectivamente.

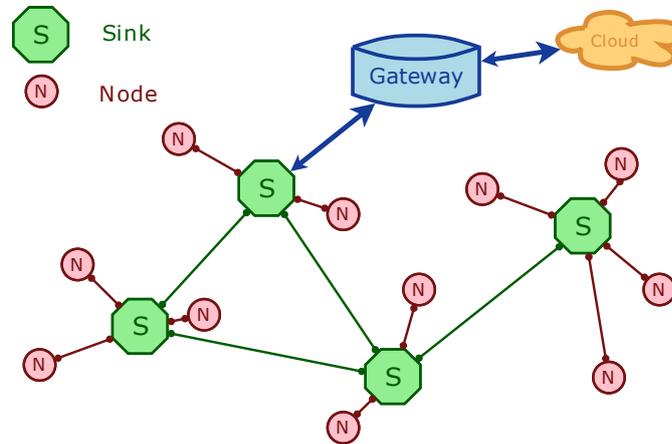
**Tabla 2.1***Diferencias entre UWSN y TWSN.*

Características	UWANs	TWSNs
Ancho de banda	Inversamente proporcional a la distancia de comunicación	No se ve afectado frente a la distancia
Arquitectura	Mayormente 3D	Mayormente 2D
Calidad del enlace	Baja (Alto BER y PLR <sup>2</sup> )	Depende de la aplicación pero relativamente mejor
Despliegue	Disperso	Denso
Dificultad de recarga	Difícil	Depende de la aplicación
Frecuencia	Frecuencias bajas (Hz, KHz), porque la señal con altas frecuencias es rápidamente absorbida en agua	Frecuencias altas (MHz, GHz)
Medio de transmisión	Por señales acústicas u ópticas	Comunicación por radio
Rango de transmisión	Entre 100[m] y 10,000[m]	Entre 10[m] y 100[m]
Retraso de propagación	Alto, debido a la comunicación lenta	Baja
Topología	Altamente dinámica debido a las corrientes de agua	Estático o ligeramente dinámica
Velocidad de propagación	Velocidad del sonido en el agua (aproximadamente 1500[m/s])	Velocidad de radiofrecuencias en el aire (aproximadamente $3 \times 10^8$ [m/s])

*Nota.* Adaptado de Han et al. (2015) y Kao et al. (2017).

### 2.1.1.1. Características de una WSN

Una WSN cuenta por lo general, con una gran cantidad de nodos ordinarios, que vienen siendo los sensores inalámbricos; estos sensores, una vez ubicados dentro del área a inspeccionar, informan acerca de cambios que ocurren en la misma con el objetivo de transmitir estos eventos a un nodo central llamado colector, sumidero o *nodo sink*, el cual también es un sensor, pero tiene comunicación con otros nodos sumideros o directamente con el *gateway*, que a su vez puede estar conectado a otros sistemas de WSN o hacia un servidor/ordenador en el cual están centralizados los datos de la(s)

**Figura 2.1***Red de Sensores Inalámbricos.*

red(es) (Ver Figura 2.1). Es importante mencionar que estos dispositivos son pequeños, cuentan con poca capacidad de cómputo y tienen una fuente de energía que se agota. Sin embargo, esto no es una limitante a la hora de realizar las labores para los cuales fueron diseñados (Blandón Andrade & López Sotelo, 2012).

Las redes de sensores inalámbricos también se caracterizan por la ubicación de cientos o miles de dispositivos en un área geográfica; éstos a su vez tienen un alcance limitado y cuando quieren informar un evento al nodo central, la información debe saltar a varios sensores antes de llegar a su destino. Es decir, se deben trazar unas posibles rutas, para cuando sucede un determinado evento en una zona específica, esto también es llamado el problema de enrutamiento en las redes de sensores inalámbricos (Blandón Andrade & López Sotelo, 2012).

En cuanto al tipo de comunicación inalámbrica que utiliza, según Molisch (2012), las WSNs pueden utilizar las tecnologías descritas por los estándares *IEEE 802.15.4*, *802.15.4a* o el protocolo de redes asociado *Zigbee*.

## 2.2. Comunicación Acústica Subacuática

Las redes de sensores inalámbricos subacuáticos, al igual que las redes terrestres, tienen un largo trayecto recorrido para llegar a ser como se las conoce hoy en día. “Gertrude”, fue uno de los primeros sistemas de comunicación acústica creados; fue desarrollado por Estados Unidos en 1945 y curiosamente, es una tecnología que aún se usa (Sánchez Matías, 2013).

Pero, ¿por qué cuando se habla de redes de sensores inalámbricos subacuáticos se da por hecho que la transmisión de información se ejecutará mediante tecnología acústica? La respuesta corta es que es la más eficiente, Sánchez Matías (2013, Sec. 2.2) hace una detallada comparativa de tecnologías en el ámbito subacuático, donde justifica el potencial que tiene la tecnología acústica para propagar la señal inalámbrica en entornos subacuáticos, frente a ondas de radiofrecuencia y comunicaciones ópticas. Para resumir, se plasman las conclusiones de Sánchez Matías en la Tabla 2.2, demostrando que el mayor problema de la radiofrecuencia y las comunicaciones ópticas es la alta absorción. Añade también que “Cuanto mayor es la conductividad de un material, mayor es la atenuación de las señales de radiofrecuencia que lo atraviesan”, dado que la conductividad del agua es elevada sobretodo en entornos submarinos u oceánicos<sup>3</sup>, el uso de radiofrecuencia en ambientes subacuáticos no ofrecería una alta eficiencia en condiciones acordes a las posibilidades<sup>4</sup>. En cambio, si se habla de comunicaciones ópticas, la mayor desventaja es su gran dependencia de la turbidez del agua, ya que cada partícula que se encuentre en suspensión en el agua provoca la dispersión del haz de luz producido.

---

<sup>3</sup>Como ejemplos, la conductividad media del agua salada se considera de  $4[S/m]$  y del agua dulce unos pocos  $mS/m$ .

<sup>4</sup>Debido a que se requeriría una antena de desmesuradas proporciones para alcanzar las frecuencias ultra-bajas necesarias para aplicaciones subacuáticas.

**Tabla 2.2**

*Comparación de las tecnologías de radiofrecuencia, óptica y acústica.*

Particularidad	Tipo de Comunicación		
	Por Radiofrecuencia	Óptica	Acústica
Absorción	Alta	Media	Baja
Conductividad del Agua	Alta	-	-
Turbidez del Agua	-	Alta	-

*Nota.* Se expresa el nivel de afectación de los factores de absorción, conductividad y turbidez del agua en las tecnologías de radiofrecuencia, comunicaciones ópticas y acústicas.

### 2.2.1. Modelo del Canal

Se establecen modelos con el propósito de calcular la *relación de entrega exitosa* de cada enlace sobre UWSNs, en otras palabras, la relación de entrega exitosa se refiere a la fiabilidad de un enlace entre dos nodos (Kao et al. 2017). Kao et al. dividen los modelos en dos partes principales, en la primera investigan la relación entre la potencia del transmisor y la SNR (*Signal to Noise Ratio*, Relación Señal a Ruido); luego, en la segunda parte investigan la relación entre la SNR y la relación de entrega exitosa.

#### 2.2.1.1. Relación Señal a Ruido con respecto a la Potencia de Transmisión

Kao et al. (2017) plantean que en un enlace entre un nodo origen  $S$  y un nodo destino  $D$ , la SNR por bit  $\gamma$  de la señal acústica transmitida en el nodo receptor puede ser calculada como

$$\gamma = S_{level} - T_{loss} - N_{level} + D_{index} \quad (2.1)$$

donde  $S_{level}$ ,  $T_{loss}$ ,  $N_{level}$  y  $D_{index}$  son el *nivel de señal*, la *pérdida de transmisión*, el *nivel de ruido* y el *índice de directividad*, respectivamente. Cabe destacar que todos los valores están en *decibelios [dB]* y el valor  $D_{index}$  es de 0, ya que en entornos subacuáticos se suelen usar *hidrófonos omnidireccionales*<sup>5</sup>.

El nivel de señal  $S_{level}$ , según Kao et al. se define como el nivel efectivo de sonido. Para calcularlo se necesita partir por la relación entre  $S_{level}$  y la *intensidad de señal transmitida*  $I$ . Se puede expresar de la siguiente manera

$$S_{level} = 10 [\log(I) - \log(0,67 \times 10^{-18})] \quad (2.2)$$

en cuanto al valor  $I$ , este puede ser calculado respecto a la *potencia del transmisor*  $P$ . La relación puede ser expresada como se indica a continuación

$$P = A \times I \quad (2.3)$$

donde  $A = 4\pi r^2$ , es el área de la superficie de una esfera imaginaria con un rango de transmisión  $r$ . Específicamente, se considera que la fuente de sonido es omnidireccional, y su potencia se distribuye de manera uniforme sobre el área de superficie de una esfera imaginaria de radio  $r$ . No está por demás mencionar que las unidades de  $P$  y  $r$  son *vatios [W]* y *metros [m]*, respectivamente. Por lo tanto, teniendo la Ecuación (2.3), se puede determinar la intensidad como

$$I = \frac{P}{4\pi r^2} \quad (2.4)$$

---

<sup>5</sup>Hidrófonos que no transmiten la señal acústica en una sola dirección, sino de manera esférica.

así mismo, utilizando las Ecuaciones (2.2) y (2.4), se logra obtener el nivel de señal  $S_{level}$  con respecto a la potencia de transmisión  $P$  y el rango  $r$  de esta forma

$$\begin{aligned} S_{level} &= 10 \left[ \log \left( \frac{P}{4\pi r^2} \right) - \log(0,67 \times 10^{-18}) \right] \\ &= 10 [\log(P) - \log(4\pi r^2) - \log(0,67 \times 10^{-18})] \end{aligned} \quad (2.5)$$

El segundo término especificado en la Ecuación (2.1) es la pérdida de transmisión  $T_{loss}$ , este es definido por Kao et al. como una medida de la tasa a la que se pierde energía sonora. La pérdida de transmisión subacuática  $T_{loss}$  a una distancia  $d[m]$  para una señal de frecuencia  $f[KHz]$  puede derivarse por

$$T_{loss} = SS + \alpha(f) \times d \times 10^{-3} \quad (2.6)$$

donde  $\alpha(f)$  es el coeficiente de absorción (del sonido en el agua de mar) en  $[dB/Km]$ . Para calcular  $\alpha(f)$ , se puede usar la fórmula de Thorp tal que así

$$\alpha(f) = \frac{0,11f^2}{1 + f^2} + \frac{44f^2}{4100 + f^2} + 2,75 \times 10^{-4}f^2 + 0,003 \quad (2.7)$$

además,  $SS$  es la propagación esférica de la señal acústica, en otras palabras, es una aproximación utilizada para describir cómo disminuye el nivel de sonido cuando la onda sonora se propaga uniformemente desde una fuente de sonido en todas las direcciones. La velocidad a la que decrece esta energía se puede obtener utilizando la intensidad de la señal. Sea  $I_{SS}$  la intensidad

de la señal de la propagación esférica, y tenemos la intensidad de la señal de origen  $I$ . Mediante la propagación esférica, la intensidad disminuye en relación al cuadrado inverso de la distancia de transmisión  $d$ . Por consiguiente, la propagación esférica  $SS$  se expresa generalmente en decibelios  $[dB]$  de la siguiente manera (Kao et al. 2017)

$$\begin{aligned} SS &= -10 \log \left( \frac{I_{SS}}{I} \right) \\ &= -10 \log \left( \frac{1}{d^2} \right) \\ &= 20 \log d \end{aligned} \tag{2.8}$$

de forma que reemplazando 2.8 en la Ecuación (2.6), tenemos

$$T_{loss} = 20 \log d + \alpha(f) \times d \times 10^{-3} \tag{2.9}$$

Para terminar con  $\gamma$ , Kao et al. también definen el nivel de ruido  $N_{level}$  como la suma de un sinnúmero de afecciones provenientes de distintas fuentes, como turbulencias, embarques y olas. En el mundo real, es difícil calcular el valor exacto del nivel de ruido, de tal forma que se utiliza una aproximación práctica del ruido en ambientes subacuáticos. Esta aproximación se establece a continuación

$$N_{level} = 50 - 18 \log f \tag{2.10}$$

Para este punto, se pueden combinar las Ecuaciones (2.5), (2.9) y (2.10) en la Ecuación (2.1), para obtener la SNR  $\gamma$ , de la siguiente manera

$$\begin{aligned} \gamma = 10 [\log(P) - \log(4\pi r^2) - \log(0,67 \times 10^{-18})] \\ - 20 \log d - \alpha(f) \times d \times 10^{-3} - 50 + 18 \log f \quad (2.11) \end{aligned}$$

y reemplazando la Ecuación (2.7) en la 2.11, se obtiene la fórmula destinada a relacionar la potencia de transmisión y el SNR

$$\begin{aligned} \gamma = 10 [\log(P) - \log(4\pi r^2) - \log(0,67 \times 10^{-18})] - 20 \log d \\ - \left( \frac{0,11f^2}{1+f^2} + \frac{44f^2}{4100+f^2} + 2,75 \times 10^{-4}f^2 + 0,003 \right) \times d \times 10^{-3} \quad (2.12) \\ - 50 + 18 \log f \end{aligned}$$

donde  $P$  es la potencia transmitida,  $r$  es el rango de transmisión, que vendría a ser lo mismo que  $d$  distancia de transmisión y  $f$  es la frecuencia (Kao et al. 2017).

### 2.2.1.2. Relación de Entrega Exitosa con respecto a SNR

El objetivo de la segunda parte del modelo del canal, es calcular la relación de entrega exitosa, es decir, la confiabilidad de los enlaces sobre IoUT.

Para lograr la meta, se necesita elegir los modelos adecuados para entornos submarinos. Primero, para el esquema de modulación, Kao et al. eligen la modulación BPSK ya que se usa ampliamente en redes de sensores acústicos submarinos. En segundo lugar, para la propagación de la señal, eligen el canal de desvanecimiento de Rayleigh porque es un modelo apropiado para el efecto *multipath* en aguas poco profundas y profundas. Según los modelos elegidos, la BER (*Bit Error Rate*, Tasa de Error de Bits)<sup>6</sup> de BPSK en un

---

<sup>6</sup>Número de errores de bit respecto al total de bits enviados por unidad de tiempo.

canal de desvanecimiento de Rayleigh, se puede determinar de la siguiente manera:

$$BER(\gamma) = \frac{1}{2} \left( 1 - \sqrt{\frac{10^{\gamma/10}}{1 + 10^{\gamma/10}}} \right) \quad (2.13)$$

donde  $BER(\gamma)$  es el número estimado de bits errados recibidos de un flujo de datos sobre un canal de comunicación que han sido alterados debido a ruido, interferencias, distorsión o errores de sincronización de bit, por cada unidad de tiempo, siendo  $\gamma$  el SNR (Kao et al. 2017).

Basándose en la tasa de error de un solo bit, expuesto en la Ecuación (2.13), se puede obtener la relación de entrega exitosa de un solo bit tal que así

$$\begin{aligned} P_{\text{exito}}^1(\gamma) &= 1 - BER(\gamma) \\ &= 1 - \frac{1}{2} \left( 1 - \sqrt{\frac{10^{\gamma/10}}{1 + 10^{\gamma/10}}} \right) \\ &= \frac{1}{2} + \frac{1}{2} \sqrt{\frac{10^{\gamma/10}}{1 + 10^{\gamma/10}}} \end{aligned} \quad (2.14)$$

donde  $P_{\text{exito}}^1(\gamma)$  se define como SDR (*Successful Delivery Ratio*, Relación de Entrega Exitosa). Por lo tanto, se define  $P_{\text{exito}}^m(\gamma)$  como la probabilidad de que un paquete de  $m$  bits sea entregado exitosamente. Así, utilizando el valor de  $\gamma$ , se puede hallar la relación de entrega exitosa de un paquete con tamaño de  $m$  bits, como se especifica en la Ecuación (2.15) (Kao et al. 2017).

$$P_{\text{exito}}^m = \left( \frac{1}{2} + \frac{1}{2} \sqrt{\frac{10^{\gamma/10}}{1 + 10^{\gamma/10}}} \right)^m \quad (2.15)$$

### 2.2.2. Método de la Ruta más Corta

En teoría de grafos, se realiza el estudio de las propiedades de los *grafos*, que son un conjunto de vértices<sup>7</sup> y aristas<sup>8</sup> que puede representar un esquema de casi cualquier índole (Recuero, 1994). La Figura 2.1 es un ejemplo de grafo que expresa la topología de una WSN.

Una de las maneras más populares de hallar una ruta entre dos vértices de un grafo (uno de origen y uno de destino), es utilizando un *algoritmo de ruta más corta*; Recuero (1994) hace mención de algunos de los matemáticos que han aportado científicamente en la investigación de dichos algoritmos, uno de estos es Dijkstra (Dijkstra, 1959), cuyo trabajo cuenta con gran relevancia en la informática (Torrubia & Lozano Terrazas, 2001).

Como se puede haber intuido, para hallar la ruta más corta es necesario conocer la distancia que existe entre los nodos, de modo que se declaren dos pasos a seguir para este efecto.

1. Calcular la distancia entre los nodos que lleguen a tener comunicación.
2. Aplicar el algoritmo de Dijkstra para encontrar la ruta más corta entre un nodo origen y un nodo destino.

#### 2.2.2.1. Distancia entre puntos

Un sistema de representación usado para calcular la distancia entre dos puntos está estrechamente relacionado con sus pares ordenados (Cáceres et al. 2018). Cáceres et al. plantean el siguiente ejemplo, teniendo los puntos  $A(x_1, y_1)$  y  $B(x_2, y_2)$  que también se han representado en la Figura 2.2, se puede calcular la distancia entre ellos utilizando la Ecuación (2.16)

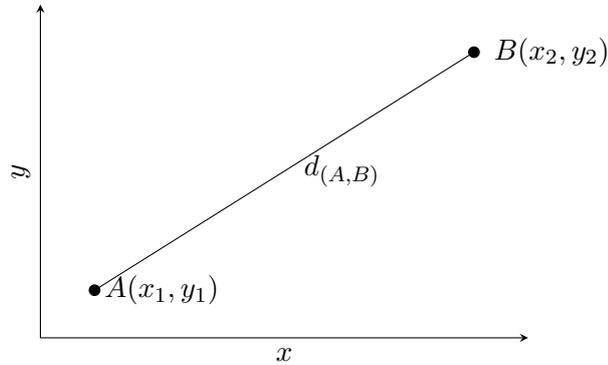
---

<sup>7</sup>Son el punto de intersección de dos aristas, en este caso se pueden describir como nodos.

<sup>8</sup>Hacen referencia a segmentos de recta que limitan un lado del grafo, en este caso las aristas representan la comunicación entre dos nodos.

**Figura 2.2**

*Distancia entre dos puntos en el plano cartesiano.*



$$d_{(A,B)} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.16)$$

Para llegar a esta fórmula se parte del famoso *Teorema de Pitágoras*. Espinosa (2022) explica claramente la manera de obtener la Ecuación (2.16).

En el caso de que los puntos se encuentren ubicados en un espacio tridimensional, la distancia se deduce de manera similar (Espinosa, 2022); osea que teniendo los puntos  $A(x_1, y_1, z_1)$  y  $B(x_2, y_2, z_2)$ , la distancia se puede calcular como se expresa en la Ecuación (2.17)

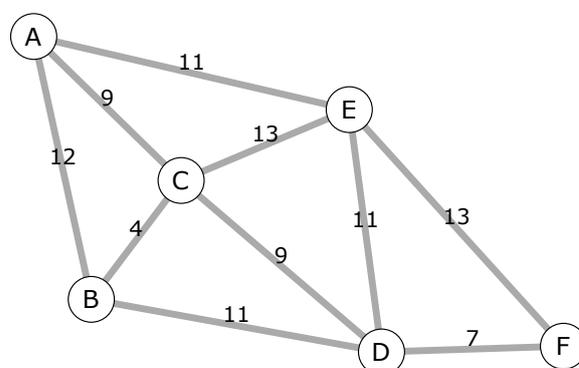
$$d_{(A,B)} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (2.17)$$

**2.2.2.2. Algoritmo de Dijkstra**

Torrubia y Lozano Terrazas (2001) explican el *Algoritmo de Dijkstra* de la siguiente manera: “Dado un grafo a cuyas aristas se han asociado una serie

**Figura 2.3**

*Grafo de nodos con aristas con peso.*



*Nota.* Los vértices, aristas y etiquetas por arista representan los nodos ordinarios, la comunicación acústica y la distancia entre nodos, respectivamente. Graficado en Dijkstra's Algorithm Solver by Mostafa Dahshan.

de pesos<sup>9</sup>, se define la ruta de coste mínimo entre dos vértices, como la ruta donde la suma de los pesos de las aristas que lo forman es la más baja entre las de todos los caminos posibles”.

En la Figura 2.3 se observa un grafo con 6 vértices, y cada arista cuenta con una etiqueta de peso que pretende representar la distancia entre los nodos. El algoritmo se asienta sobre el principio de optimalidad: si la ruta más corta entre los vértices  $A$  y  $F$  pasa por el vértice  $C$ , entonces la parte de la ruta que va de  $A$  a  $C$  debe ser más corta entre todas las rutas que llegan a  $F$ . De esta forma, se va construyendo la ruta hasta el vértice de destino (Torrubia & Lozano Terrazas, 2001).

<sup>9</sup>Con este valor el algoritmo determina una ruta más óptima, para este caso el peso de la comunicación será determinado por la distancia (Ver §2.2.2.1).

## 2.3. Redes de Sensores Inalámbricos Subacuáticos (UWSN)

Las UWSNs (Underwater Wireless Sensor Networks) o Redes de Sensores Inalámbricos Subacuáticos nacen de la necesidad de explorar y monitorear el entorno subacuático en tiempo real, con el fin de proporcionar cierta agilidad en el desenvolvimiento de un sinnúmero de aplicaciones, ya sea con propósitos científicos, comerciales o militares (El-Rabaie et al. 2015; Haque et al. 2020; Tan et al. 2011). Una UWSN está compuesta por una cantidad considerable de nodos sensores que son desplegados a diferentes alturas/profundidades del volumen del mar (Haque et al. 2020). El número de nodos sensores depende del volumen del mar que se cubrirá y otros factores, como el rango de transmisión de los nodos o el rendimiento deseado de la red.

### 2.3.1. Características y desafíos que presenta una UWSN

En la Tabla 2.1, se enumeran a breves rasgos las principales características de las WSNs subacuáticas, para profundizar en cuanto a propiedades se refiere, a continuación se explica más a detalle los aspectos más desafiantes que se presentan en el diseño de UWSNs (Kao et al. 2017).

**Medios de transmisión** En TWSNs, las ondas de radio a menudo se usan para comunicaciones. Sin embargo, los UWSN generalmente dependen de las comunicaciones acústicas en lugar de radiofrecuencia (Ver §2.2). Esto se debe a que las señales de radio serían absorbidas por el agua muy rápidamente. Desafortunadamente, las propiedades de las ondas acústicas son muy diferentes de las de las ondas de radio; motivo por el cual, muchos de los protocolos de comunicaciones aplicados a TWSN no pueden aplicarse directamente a

UWSN.

**Velocidad de propagación** La velocidad de propagación en UWSN es alrededor de 200,000 veces más lenta que la de TWSNs. Específicamente, la velocidad de propagación de los canales de radio TWSN es de  $300,000,000[m/s]$ , mientras que la velocidad de propagación de los canales acústicos UWSN es de solo alrededor de  $1500[m/s]$ . Por lo tanto, garantizar el retraso de extremo a extremo limitado sería un problema desafiante.

**Rango de transmisión** El rango de transmisión de UWSNs podría ser diez veces más largo que el de TWSNs. En ambientes submarinos, para evitar ser absorbidos por el agua, las señales deben transmitirse mediante la baja frecuencia. La frecuencia más baja implica un mayor rango de transmisión; y el rango de transmisión más largo implica más posibilidades de que las interferencias y colisiones ocurran durante la transmisión de datos.

**Tasa de transmisión** En comparación con las comunicaciones de radio en TWSNs, las comunicaciones acústicas en UWSN usan un ancho de banda estrecho. Debido al ancho de banda estrecho, la velocidad de transmisión en UWSN es generalmente muy baja (aproximadamente  $10[Kbps]$ ).

**Dificultad de recarga** La energía de la batería de un nodo subacuático es limitada y las baterías no pueden recargarse tan fácilmente, ya que la energía solar no puede ser aprovechada<sup>10</sup>. Cuando consideramos el costo potencial de recargar las baterías de los sensores submarinos, está claro que la eficiencia energética sería otra preocupación importante.

---

<sup>10</sup>Debido a las propiedades de la propagación de la luz bajo el agua.

**Movimiento de nodos** Las UWSNs son WSNs móviles por naturaleza. Cuando hay corrientes de agua, los sensores pueden moverse y sufrir cambios de topología de red dinámica. Es una tarea difícil lidiar con los cambios dinámicos para IoUT.

**Calidad del enlace** La calidad del enlace en UWSNs es típicamente inestable y baja. La fiabilidad de un enlace significa que la relación de entrega exitosa<sup>11</sup> entre un par de nodos sensores. En una UWSN, la relación de entrega exitosa se vería gravemente afectada por la pérdida de transmisión<sup>12</sup> y ruidos ambientales. Debido a que las señales serían absorbidas por el agua en ambientes submarinos, la pérdida de transmisión es un grave problema en una UWSN. Además, los ruidos ambientales en UWSNs están compuestos por varios factores complicados, incluyendo la turbulencia, embarques, las olas, etc.

A mayores, El-Rabaie et al. (2015) mencionan otros desafíos tales como:

- Los sensores submarinos son caros en términos de equipo, implementación y mantenimiento.
- Componentes de red (Ver §2.3.3).
- La memoria limitada por la capacidad del dispositivo de almacenamiento a bordo.
- Los sensores son propensos a fallas debido a que la suciedad, corrosión y el tamaño de los sensores es grande.
- El canal deteriorado debido al *multipath* y el *fading*.
- La sincronización del tiempo, y por ende, la localización de nodos, es difícil de lograr en el agua debido al retraso de propagación y la velocidad del sonido.

---

<sup>11</sup>La relación de entrega exitosa se define como la relación del número de datos que se han entregado con éxito a un receptor en comparación con el número de datos que el remitente ha enviado.

<sup>12</sup>La pérdida de transmisión es la disminución acumulada en la intensidad de una energía en forma de onda cuando una onda se propaga hacia afuera desde una fuente.

**Figura 2.4**

*Componentes principales de un nodo ordinario.*

Sensor Temper.	Transceiver Acústico	Modulador	Códec	Control de Potencia
Sensor Salin.				
Sensor Profun.	Conversor Análogo Digital	Microcontrolador		Memoria
Sensor Presión				
⋮	Fuente de Alimentación			Recolección de energía (Solar, Corrientes)
Otros Sensores				

*Nota.* Tomado de (El-Banna et al. 2020)

- Variación en la velocidad del sonido debido a las condiciones del agua. Cualquier cambio en uno de estos criterios afecta a la velocidad del sonido. Puede causar inexactitud en la predicción de posición.

### 2.3.2. Componentes de una UWSN

Los componentes que tienen protagonismo en una UWSN depende en parte del tipo de arquitectura que se desee utilizar (Ver §2.3.3). A groso modo, los componentes de red que suelen utilizarse son: nodos ordinarios submarinos, nodos sumidero, AUV (*Autonomous Underwater Vehicle*, Vehículo Autónomo Subacuático) y estación base en tierra, aunque de cierta forma, este último ya no pertenece del todo a la UWSN.

#### 2.3.2.1. Componentes principales de un nodo ordinario

La Figura 2.4 muestra los componentes principales de un nodo o mota de sensor típico según El-Banna et al. (2020). En la capa inferior se encuen-

tra el módulo de fuente de alimentación donde las baterías recargables o no recargables se utilizan para suministrar otros componentes con la energía requerida. También pueden existir algunos tipos de módulos de recolección de energía en esa capa, donde la energía se puede recolectar en el entorno submarino de las fuentes ambientales disponibles, como los rayos solares<sup>13</sup> o corrientes de agua. Una segunda capa incluye el controlador que es el cerebro y el despachador de todas las tareas realizadas por el nodo, la memoria para almacenar las rutinas y algoritmos requeridos, y los convertidores analógicos a digitales (ADC) que se utilizan para probar y digitalizar los datos recopilados provienen de la capa de sensores. Esta capa de sensores incluye varios tipos de sensores para medir y monitorear los eventos requeridos y las características submarinas, como la temperatura, la salinidad, la profundidad del agua, la presión, las ondas internas, etc., y también contiene varios tipos de actuadores como motores. Otra capa involucra los módulos de comunicación como el modulador, el codificador/decodificador de canal, la unidad de control de potencia y el transceptor acústico para transmitir y recibir los mensajes procesados. Además, y dependiendo de la aplicación de destino, el nodo puede equiparse con módulos adicionales como micrófonos, cámaras y sus unidades de procesamiento de audio/imagen adyacentes. Además, los módulos GPS tal vez incluidos en algunos nodos, sin embargo, el costo de tales nodos se vuelve más alto que los típicos.

### 2.3.3. Arquitecturas de UWSNs

A modo general existen dos arquitecturas comunes de comunicación para UWSNs, que son bidimensionales y tridimensionales. Existe una tercer arquitectura también considerada tridimensional en la cual se trabaja jun-

---

<sup>13</sup>Aunque no presenta gran eficiencia en zonas muy profundas.

to a un *Vehículo Autónomo Subacuático*, o en muchos casos, se trabaja con varios.

### 2.3.3.1. UWSN Bidimensional

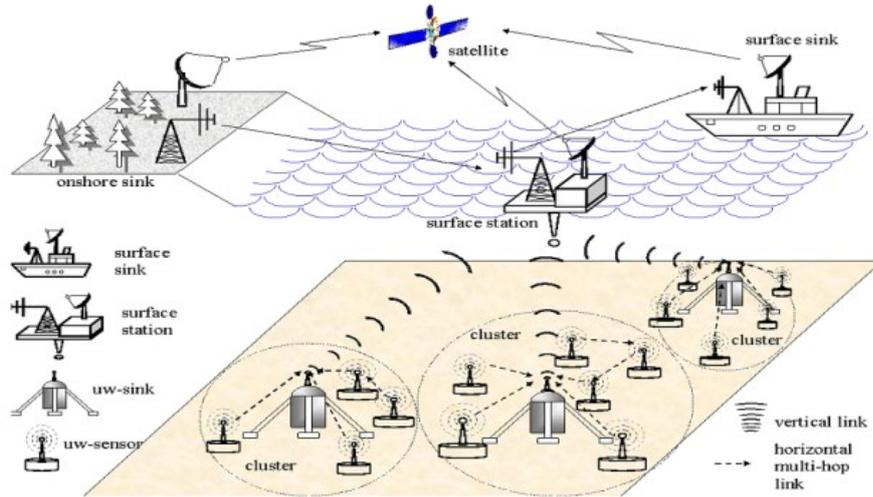
En la Figura 2.5 se muestra la disposición de cada componente dentro de la arquitectura, todos los nodos se encuentran anclados en el fondo del océano. Un *uw-sink* (o sumidero subacuático) recopila los datos de los nodos sensores, para conseguir este objetivo, los sumideros están equipados con dos *transceivers acústicos*, uno llamado vertical y otro horizontal. El transceiver horizontal es utilizado por el sumidero para comunicarse con los nodos sensores, ya sea para enviar comandos y datos de configuración a los sensores (de sumidero a sensores), o para recoger los datos monitoreados (de sensores a sumidero). Luego, transmite la información a una estación de superficie que en este caso se puede considerar el gateway, de esto se encarga el transceiver vertical. La estación de superficie tiene una señal de RF (*Radiofrecuencia*) para comunicarse con los sumideros *onshore* (en tierra) y *surface* (de la superficie). Los sensores se comunican con el sumidero utilizando enlaces directos o rutas de múltiples saltos (El-Rabaie et al. 2015; Ibrahim & Hussein, 2014).

**Enlace directo.** En este, cada sensor envía datos directamente a su nodo sumidero asociado. Este no pretende ser el más eficiente en cuanto a consumo de energía.

**Rutas de múltiples saltos.** El sensor de origen transmite los datos a sensores intermedios hasta llegar al sumidero. Esto ahorra energía y aumenta la capacidad de la red, pero también aumenta la dificultad del enrutamiento.

**Figura 2.5**

*Arquitectura bidimensional de una UWSN.*



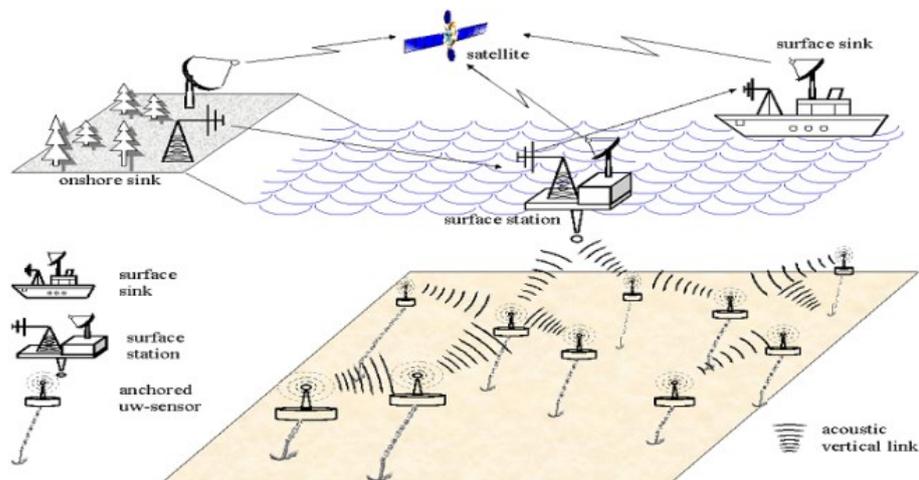
*Nota.* Tomado de (El-Rabaie et al. 2015).

### 2.3.3.2. UWSN Tridimensional

En este caso, al tener los nodos a diferentes alturas/profundidades, cada nodo se encuentra anclado con un cable al fondo del océano y para ofrecerles características de flote se equipan con una boya. Con esto se logra tener los nodos estáticos, dentro de lo que cabe. La profundidad del sensor se puede regular ajustando la longitud del cable que conecta el sensor al ancla, mediante un motor controlado electrónicamente que reside en el sensor. Un desafío a abordar en dicha arquitectura es el efecto de las corrientes oceánicas en el mecanismo descrito para regular la profundidad de los sensores. Como se observa en la Figura 2.6, la comunicación se rige por rutas de múltiples saltos, ya que el/los nodos sumideros se encuentran en la superficie (El-Rabaie et al. 2015; Ibrahim & Hussein, 2014).

**Figura 2.6**

*Arquitectura tridimensional estática de una UWSN.*



*Nota.* Tomado de (El-Rabaie et al. 2015).

### 2.3.3.3. UWSN Tridimensional con Vehículos Autónomos Subacuáticos

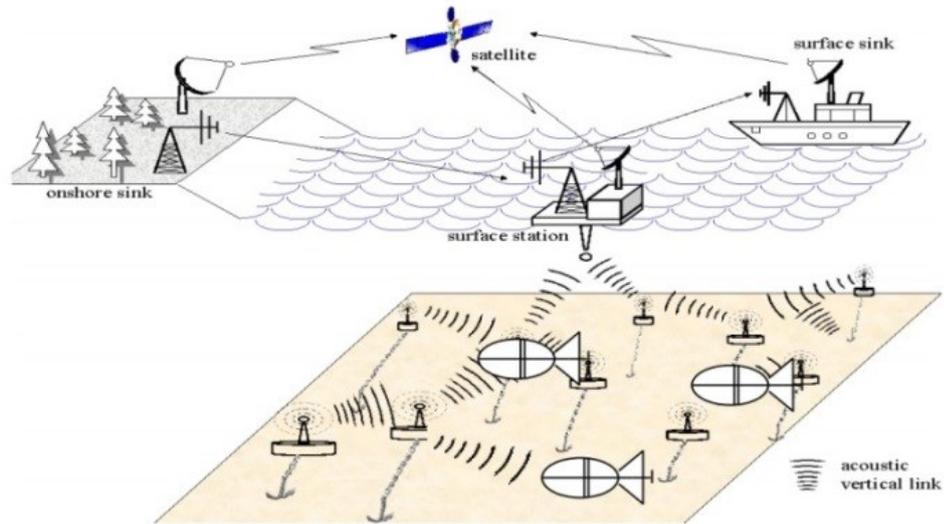
Consiste en muchos sensores estáticos junto con algunos AUVs, como se muestra en la Figura 2.7. Los AUVs juegan un papel clave para el soporte adicional en la recolección de datos. Los AUVs podrían considerarse como súper nodos, que tienen más energía, pueden moverse de forma independiente, y podría ser un enrutador entre sensores fijos, o un gerente para la reconfiguración de la red, o incluso un sensor normal.

### 2.3.4. Aplicaciones de una UWSN

Las aplicaciones de las redes submarinas se dividen en clasificaciones similares que las redes de sensores terrestres, se especifica a continuación (El-Rabaie et al. 2015):

**Figura 2.7**

*Arquitectura tridimensional de una UWSN con AUVs.*



*Nota.* Tomado de (El-Rabaie et al. 2015).

**Aplicaciones científicas.** Que observan el medio ambiente desde los procedimientos geológicos en el fondo del mar, hasta las características del agua (temperatura, salinidad, niveles de oxígeno, contenido bacteriano y de otro contaminante, materia disuelta) a contar u obtener imágenes de la vida animal (microorganismos, peces o mamíferos).

**Aplicaciones industriales.** Que monitorean y controlan el proceso comercial, como las herramientas submarinas relacionadas con la extracción de petróleo.

**Aplicaciones de seguridad militar y nacional.** Que incluyen asegurar y monitorear instalaciones portuarias o barcos en puertos extranjeros, y comunicarse con submarinos.

También se puede hallar UWSNs en aplicaciones más específicas tales

como:

- Red acústica de aguas poco profundas para operaciones de contramedidas de minas.
- Redes de sensores inalámbricos en la Gran Barrera de Coral.
- UASN para la generación de alerta temprana de eventos naturales.
- Arquitecturas de red de sensores para monitorear tuberías submarinas.
- Red de Sensores de Vigilancia Submarina Autónoma (AUSSNet).
- UWSN a base de grupos para granjas de pesca marina.
- Red acústica submarina para la protección de plataformas en alta mar y plantas de energía.
- Plataforma de Monitoreo Submarino SeaSTAR.

## 2.4. Ingeniería de Software

Se define la Ingeniería de Software como una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos (software) (Pérez & Gardey, 2013).

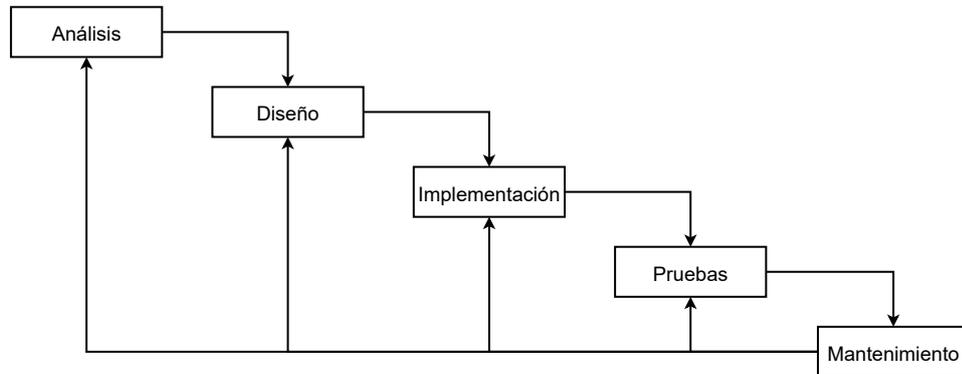
Es necesario recalcar que esta disciplina va más allá de únicamente programar el código para una aplicación, que ciertamente es la actividad más representativa del proceso; el desarrollo del proyecto debe contar con varias etapas, donde toma lugar el análisis de requerimientos, el diseño del proyecto, la implementación, las pruebas y corrección de errores, entre otras actividades que dependen de la metodología SDLC (*Software Development Life Cycle*, Ciclo de Vida de Desarrollo de Software) utilizada (Pérez & Gardey, 2013).

### 2.4.1. Ciclo de Vida Clásico del Software

Como explica Prokopets (2020), SDLC es un método utilizado para organizar el proceso del desarrollo de software, usando diferentes metodologías

**Figura 2.8**

*Etapas de la metodología del Ciclo de Vida Clásico del Software.*



*Nota.* Adaptado de Prokopets (2020)

y herramientas, para mejorar la eficiencia de producción.

Una de estas metodologías es la llamada *Ciclo de Vida Clásico del Software*, también conocida como *metodología en Cascada*. La metodología en Cascada es un enfoque secuencial y lineal, donde cada nueva fase SDLC comienza al finalizar la anterior (Ver Figura 2.8). Debido a su esquema altamente estructural, esta metodología carece de la flexibilidad ágil. Por ejemplo, es mucho más difícil volver a las etapas anteriores en el ciclo de vida del producto para cambiar los requisitos durante la implementación (Prokopets, 2020).

### 2.4.2. Herramientas de Desarrollo de Software

Existen un sinnúmero de lenguajes de programación que permiten al desarrollador de software compilar un algoritmo con el fin que un ordenador lo ejecute. Muchas de las herramientas que en la actualidad se encuentran en el mercado son *open source* (*Perl, Java, C++, Python, etc.*), y otras de software propietario (*Matlab, Mathematica, Visual Basic, etc.*) cuya ventaja

es el soporte que la corporación desarrolladora ofrece.

Blandón Andrade y López Sotélo (2012) menciona algunas de estas herramientas junto con sus ciertas características ventajosas de cada lenguaje, Matlab cuenta con la mayoría de estas (Ver §2.5), convirtiéndose en un software ideal para desarrollo de una aplicación académica.

## 2.5. Matlab

MathWorks (2022f) define a MATLAB (*MATrix LABoratory*, Laboratorio de Matrices) como una plataforma de programación diseñada específicamente para ingenieros y científicos, utilizando un lenguaje de programación del mismo nombre. MATLAB se especializa en analizar y diseñar sistemas y productos que transforman nuestro mundo.

Es evidente la robustez que el software ofrece, su lenguaje de programación está basado en matrices, lo que permite que la matemática computacional se exprese en todo su esplendor.

### 2.5.1. Matlab en la Ingeniería

Tal como se introdujo en §2.5, MATHWORKS<sup>14</sup> lanza el producto MATLAB con la visión principal enfocada en la ingeniería e investigación científica, siendo algunas de sus prestaciones las enlistadas a continuación:

**Análisis de datos:** Explicación y visualización de datos de todo tipo.

**Gráficas:** Despliegue de diagramas y gráficas que aportan un entendimiento superior en el análisis de datos.

**Desarrollo de algoritmos:** Programación de aplicaciones embebidas tanto para uso empresarial como personal.

**Creación de apps:** Desarrollo de aplicaciones web y de escritorio utilizando la herramienta AppDesigner.

---

<sup>14</sup>Corporación encargada del desarrollo de MATLAB.

**Uso de MATLAB con otros lenguajes:** Utilización de la aplicación de escritorio con soporte para varios lenguajes de programación como *Python*, *C/C+*, *Fortran*, *Java*, entre otros.

**Hardware:** Conexión de MATLAB con hardware externo con propósitos administrativos o de desarrollo<sup>15</sup>.

**Cálculo paralelo:** Ejecución de cálculos a gran escala a través de equipos multinúcleo, GPU, clusters y nubes.

**Despliegue en escritorio y web:** Manipulación del programa a través de la aplicación de escritorio o a su vez desde un navegador de manera online.

**Cálculo en la nube:** Ejecución de aplicaciones y programas a partir de un servicio en la nube como por ejemplo: *MathWorks Cloud*, *AWS* o *Azure*<sup>16</sup>.

Sumando todas las funcionalidades explicadas junto con la documentación que provee la empresa, el soporte que ofrece el equipo de MATHWORKS y la amplia comunidad que se ha ganado con el pasar de los años, hacen que MATLAB gane terreno en áreas como:

- Sistemas de control
- Machine Learning
- Procesamiento de señales
- Deep Learning
- Mantenimiento predictivo
- Prueba y medición
- Procesamiento de imágenes y visión artificial
- Robótica
- Comunicaciones inalámbricas

---

<sup>15</sup>Un ejemplo de uso en este ámbito es la administración de un Raspberry Pi y la programación de sus pines.

<sup>16</sup>Servicios de Cloud ofrecidos por MATHWORKS, Amazon y Microsoft respectivamente.

En los últimos años, los softwares matemáticos más populares son *Mathematica*, MATLAB y *Maple* (Yu, 2018), y de entre estos MATLAB destaca por el soporte de todas las funciones mencionadas anteriormente. Se puede hallar información más detallada acerca de las características y aplicaciones de MATLAB en MathWorks (2022a).

### 2.5.2. App Designer

MATLAB App Designer es una herramienta de MATLAB que permite crear apps profesionales de manera sencilla e intuitiva. Con tan solo arrastrar los componentes visuales en un lienzo, se puede crear el diseño de la GUI (*Graphical User Interface*, Interfaz Gráfica de Usuario) sin mucho esfuerzo, al mismo tiempo que se programa cada uno de los elementos de forma que cambie su comportamiento en la aplicación (MathWorks, 2022b).

Cabe destacar que App Designer está basado en OOP (*Object-Oriented Programming*, Programación Orientada a Objetos)<sup>17</sup> (MathWorks, 2022b), lo que quiere decir que cada elemento puede ser programado a voluntad de manera independiente (MathWorks, 2004).

### 2.5.3. Matlab Student Suit

A día de hoy, MATHWORKS distribuye una amplia gama de productos, orientados al desarrollo y la investigación, encabezando MATLAB y SIMULINK como sus dos ramas principales, y desencadenando dos grandes familias con diversas *toolboxes*<sup>18</sup> que en la mayoría de casos se expenden de forma separada, dependiendo del tipo de licencia.

Con respecto a las licencias, MATHWORKS ofrece unas cuantas con dife-

---

<sup>17</sup>OOP Es lo que se conoce como un paradigma de programación, es decir, un estilo de programación que trabaja con el concepto de clases y objetos.

<sup>18</sup>Paquetes proveídos a MATLAB, para la adición de características de cualquier índole.

rente contenido (para industrias, estudiantes, universidades, empresas, entre otros), una de ellas es MATLAB Student Suite, una licencia exclusiva para estudiantes de una institución de educación superior que cuenta con la paquetería básica de MATLAB y SIMULINK. A continuación, se detalla su contenido (MathWorks, 2022c).

- MATLAB
- SIMULINK
- Control System Toolbox
- Curve Fitting Toolbox
- DSP System Toolbox
- Image Processing Toolbox
- Instrument Control Toolbox
- Optimization Toolbox
- Parallel Computing Toolbox
- Signal Processing Toolbox
- Statistics and Machine Learning Toolbox
- Symbolic Math Toolbox

## 2.6. Trabajos Relacionados

Breve repaso de documentos de interés que serán de ayuda para el desarrollo de este proyecto.

### **Underwater Wireless Sensor Networks (UWSN), Architecture, Routing Protocols, Simulation and Modeling Tools, Localization, Security Issues and Some Novel Trends**

En 2015, El-Rabaie et al. realizan una extensa investigación acerca de UWSN, tocando temas como los anunciados en el título, es decir, arquitect-

tura, protocolos de enrutamiento, herramientas de simulación y modelado, localización, problemas de seguridad y algunas tendencias novedosas. A mayores, toca temas de menor manifiesto como lo es la entrega de paquetes, su relación, retrasos, consumo energético, prioridades e inconvenientes de cada protocolo de enrutamiento, entre otros aspectos. Se detalla cada una de las características poniendo como premisa los retos que se tienen que enfrentar al estudiar acerca de las UWSNs. Al tener una amplia recopilación acerca de las características fundamentales en lo que a UWSN se refiere, lo hace un trabajo ideal para tomar como punto de referencia teórico para este proyecto.

### **Opportunistic Cooperative Transmission for Underwater Communication based on the Water's key Physical Variables**

El-Banna et al. (2020) comenta las complicaciones que conlleva el trabajar en un ambiente subacuático ya sea por motivos naturales o simplemente por inconvenientes causados por la mano del hombre; por lo que en estos casos es necesaria una *comunicación oportunista y cooperativa*, como es de esperarse, el autor también ofrece un esquema para cumplir con sus requerimientos de comunicación, creando el concepto de KPV (*Key Physical Variable*, Variable Física Clave). Estas variables físicas clave del agua de mar hacen referencia a los parámetros de temperatura, salinidad, densidad y velocidad de ondas, con el fin de indicar el esquema de comunicación, método de modulación y el nivel de potencia de transmisión adecuados para que consiga adaptarse a las situaciones explicadas anteriormente. Este trabajo aporta al proyecto con dos esquemas de comunicación ideales para la simulación de redes de sensores inalámbricos subacuáticos, que son los esquemas AF (*Amplify-and-Forward*, Amplifica y Reenvía) y DF (*Decode-and-Forward*, Decodifica y Reenvía), cada uno con cierta ventaja dependiendo de la situa-

ción.

### **A Comprehensive Study on the Internet of Underwater Things: Applications, Challenges, and Channel Models**

Kao et al. (2017) es otro de los trabajos que no se conforma con centrarse en un ámbito específico, toca variados aspectos sobre IoUT (*Internet of Underwater Things*, Internet de las Cosas Subacuáticas), que es un tema con estrecha cercanía a Redes de Acústicas Subacuáticas, por no mencionar que son sinónimos. Lo que más destaca del trabajo es la manera en que propone modelos del canal acuático, divide en dos partes la forma de determinar la calidad o eficiencia del medio, primero calcula la relación entre la potencia de transmisión y la relación señal a ruido y luego calcula la relación entre la relación señal a ruido y la relación de entrega exitosa al receptor. La propuesta será aplicada al presente proyecto, utilizando la matemática explicada en el código que se va a programar, de forma que se pueda simular el canal acuático sin mayores dificultades.

## Capítulo 3

# Diseño y Desarrollo

**RESUMEN:** En este capítulo se elabora la aplicación propuesta, de modo que se cumplirá con las primeras 3 etapas del método del ciclo de vida clásico del software, que son: análisis, diseño e implementación.

### 3.1. Análisis

La etapa de análisis corresponde a la definición de los requerimientos con que la aplicación debe cumplir. Este análisis está basado en el estándar *IEEE 830* (IEEE, 1998). Más adelante se enumeran aspectos a partir de los cuales se creará el diseño.

En §1.2, se ha hecho mención de lo que se espera como resultado del desarrollo del proyecto. Para resumir, el propósito de la aplicación es facilitar a los interesados en las UWSN el estudio de dichas redes, de forma que se obtengan resultados bastantes cercanos a la realidad, se salden ciertas dudas acerca de esta tecnología y por qué no mencionar, que crezca la comunidad inclinada hacia este campo. Por este motivo, es necesario hacer énfasis en los futuros usuarios potenciales del software y tomar en cuenta su opinión acerca del tema.

Antes de continuar, es necesario mencionar que se ha definido el nombre del sistema utilizando el acrónimo SARAS (*Simulador Académico de Redes Acústicas Subacuáticas*), en adelante la aplicación será denominada de esta manera.

### 3.1.1. Recopilación de Requerimientos

Para esta actividad, se utilizan dos técnicas de recopilación de requerimientos, la primera consiste en una encuesta dirigida a estudiantes de la materia de Comunicación Inalámbrica y la segunda en realizar una entrevista al docente de la materia y con los datos recopilados hacer una valoración de funciones necesarias para el programa.

Las preguntas realizadas a los estudiantes y al docente, cuyos formularios se evidencian en el Apéndice A, van enfocadas con los siguientes propósitos:

- Hacer un sondeo para evaluar el grado de conocimiento de los estudiantes y el docente en el ámbito de Redes de Sensores Inalámbricos Subacuáticos.
- Conocer el nivel de interés de los estudiantes y el docente por aprender más acerca de las UWSNs, esta información es radical al momento de evaluar cuán útil puede ser el software que se va a desarrollar.
- Hacer un análisis de la importancia de ciertas funcionalidades que aspira tener el software y el nivel de prioridad que los encuestados creen que cada uno debería tener. Las funcionalidades propuestas fueron previamente debatidas entre tesista y director para llegar a un planteamiento moderado.
- Evaluar en este caso los atributos de calidad que define la ISO 25010 y qué es lo mejor para esta aplicación desde la perspectiva de los usuarios

potenciales de la aplicación.

- Recaudar las distintas sugerencias que los encuestados proponen como adiciones al proyecto.

Tanto la tabulación de respuestas obtenidas en la encuesta como las de la entrevista se presentan en el Apéndice B.

### **3.1.2. Descripción General**

IEEE (1998) manifiesta que “en esta sección se describen todos aquellos factores que afectan al producto y a sus requisitos. No se describen los requisitos, sino su contexto.”. En este sentido y teniendo como soporte las respuestas de los estudiantes y el docente de la materia de Comunicación Inalámbrica, se toma en consideración las mejores funcionalidades para SARAS.

#### **3.1.2.1. Funciones del Producto**

Tomando en cuenta las funciones propuestas y valoradas por los futuros potenciales usuarios de la aplicación, SARAS tendrá soporte de las siguientes funciones:

- Crear nodos en una ventana específica para un entorno bidimensional.
- Hallar una ruta entre un nodo origen y uno destino.
- Importar datos guardados con anterioridad.
- Exportar datos ingresados o resultados obtenidos en el sistema.

#### **3.1.2.2. Características de los Usuarios**

El software va dirigido principalmente a estudiantes y docentes de ingeniería en redes con cierto nivel de conocimiento en el campo de redes

subacuáticas.

### 3.1.2.3. Suposiciones y Dependencias

Como se ha explicado en §2.3.2, existe una gran variedad de sensores subacuáticos con características de todo tipo y unos con más beneficios que otros; a la hora de simular una red de sensores en software, el hecho de contar con atributos distintos por cada sensor hace que la tarea que realiza la aplicación tome una mayor cantidad de recursos de procesamiento. Bajo este criterio se hará referencia a un nodo como un sensor inalámbrico subacuático genérico y sin características especiales.

Todas las características del simulador están diseñadas para trabajar de manera offline (sin conexión a internet), SARAS será una aplicación cien por ciento de escritorio, y únicamente trabajará con archivos locales del equipo en cuestión.

SARAS está pensado para ser ejecutado únicamente en sistemas operativos Windows, ya que el archivo ejecutable tendrá la extensión `.exe`<sup>1</sup>.

### 3.1.2.4. Requerimientos Futuros

Para el desarrollo de futuras versiones se proponen las siguientes mejoras para el software:

- Implementar protocolos de localización de nodos.
- Implementar otros protocolos de enrutamiento.
- Determinar la naturaleza de cada nodo (Ver §2.3.3.1).
- Implementar arquitectura tridimensional.
- Implementar consumo energético.

---

<sup>1</sup>Extensión propia de Microsoft Windows para archivos ejecutables (`executable`).

### 3.1.3. Requerimientos Específicos

Según IEEE (1998), en esta sección define a detalle los requisitos que permiten al desarrollador diseñar un sistema que satisfaga estos requerimientos. Es importante que los requerimientos sean legibles y entendibles para personas de distinta formación, que sean identificables mediante un código e idealmente que se puedan calificar como correctos, no ambiguos, completos, consistentes, clasificados, verificables, modificables y trazables. Todas estas características son explicadas en el estándar IEEE 830 (IEEE, 1998).

La Tabla 3.1 muestra un listado de requerimientos con los que SARAS cumplirá, a la vez se asigna un *ID* para cada CU (Caso de Uso); por ejemplo: *CU01* y su respectivo nivel de prioridad para el sistema clasificado en: *Esencial*, *Condicional* u *Opcional*. No está de más recalcar que estos requerimientos funcionales conllevan a una sucesión de funciones que juntas logran un fin (Ver §3.1.3.2).

#### 3.1.3.1. Interfaces Externas

SARAS contará con una GUI (*Graphical User Interface*, Interfaz Gráfica de Usuario) de escritorio, desde donde el usuario operará y manipulará todas las funciones del software.

Para empezar, una *Ventana Principal* con la que iniciará el simulador, en esta se encontrarán la mayoría de botones y elementos de interacción, además de los campos que determinan las características que se atribuirán a los sensores que se simularán. Además, ocupando la mayor cantidad de espacio se tendrá el área de dispersión de los sensores que se van a graficar en forma de puntos en el plano, junto con la ruta que se va a determinar resaltada con un color que permita al usuario distinguir y comparar resultados de manera más visual.

**Tabla 3.1***Listado de Requerimientos del Sistema.*

ID	Requerimiento	Descripción	Prioridad
01	Crear Sensores	Permite que en una ventana desplegable se creen los sensores deseados.	Esencial
02	Ingresar Parámetros de los Nodos	Permite que el usuario ingrese los parámetros necesarios para que el sistema realice los cálculos pertinentes.	Esencial
03	Ejecutar Protocolo de Enrutamiento	Permite trazar una ruta entre un nodo origen y uno destino haciendo uso de un protocolo de enrutamiento.	Esencial
04	Transmitir Datos	Permite simular la eficiencia de una transmisión de bits utilizando los parámetros de entrada y la ruta creada.	Esencial
05	Desplegar Resultados	Permite generar valores legibles para el usuario que indiquen el resultado de la simulación.	Esencial
06	Editar Parámetros	Permite que el sistema se pueda reutilizar sin necesidad de un reinicio.	Opcional

Se va a contar con una ventana flotante que destaca por la utilidad de *Crear o Editar Sensores*, donde se tomarán datos referentes a los atributos del sistema de nodos. En esta ventana se hará énfasis en una tabla con valores de configuración para cada nodo.

Los botones con función de *Importar* o *Exportar* desplegarán ventanas del explorador de archivos para facilitar la búsqueda de una carpeta o de un archivo en cuestión.

No está por demás destacar que se pretende trabajar con una GUI lo menos saturada de información innecesaria posible, por lo que se procura utilizar botones con íconos que representen una función en lugar de usar muchas palabras; como es de esperar, el significado de cada ícono será explicado

tanto en el *Manual de Uso* como en la propia aplicación.

### 3.1.3.2. Funciones

Las funciones del sistema se organizan por jerarquía funcional, por el hecho de que para la ejecución del sistema se requiere un proceso de acciones como los casos descritos en la Tabla 3.1: *Crear sensores, Ingresar Parámetros de los Nodos, Ejecutar Protocolo de Enrutamiento, Transmitir Datos entre Nodos y Desplegar Resultados*. Como casos opcionales, el usuario tiene la oportunidad de *Importar Datos, Exportar Datos, Editar Parámetros y/o Reiniciar Simulación*.

Los casos de uso han sido especificados en el Apéndice C, donde se detalla cada caso de uso del sistema en las Tablas C.1 a C.6.

### 3.1.3.3. Atributos del Sistema

Se espera que el software cumpla con al menos características de Adecuación Funcional, Usabilidad, Mantenibilidad y Portabilidad del modelo de calidad del producto definido por la *ISO/IEC 25010* en (ISO, 2011). A modo de resumen, el modelo de calidad define estos atributos de la siguiente manera:

**Adecuación Funcional** El programa es capaz de suplir las necesidades declaradas, cumplir con las tareas propuestas y proveer resultados correctos y con el nivel de precisión requerido. De igual manera, el programa debe proporcionar un conjunto de funciones para tareas y objetivos del usuario especificados.

**Usabilidad** El programa es apto para ser entendido, aprendido, usado y resulta atractivo al usuario así como permite ser operado y controlado con facilidad. A esto también se suma hacer conocer al usuario cuando comete

un error de operabilidad del programa, explicando la manera correcta de efectuar.

**Mantenibilidad** El programa es apto para ser modificado efectiva y eficientemente, de cara a la evolución o perfeccionamiento del mismo. En otras palabras, el programa debe tener la capacidad para cambiar componentes sin comprometer considerablemente los demás.

**Portabilidad** El programa ofrece la facilidad para ser transferido de forma efectiva y eficiente de un entorno a otro con las limitantes que el mismo dispone; teniendo en consideración actividades como la instalación y desinstalación sencilla del mismo, y a su vez con facilidad de ser reemplazado.

## 3.2. Diseño

En la etapa de diseño se comienza a representar los requerimientos levantados en la etapa anterior de manera que el equipo encargado de la programación del software tenga una idea clara del proyecto. Esta representación puede llevarse a cabo con diferentes técnicas, en este caso se realiza un *diagrama de casos de uso*<sup>2</sup> a modo de resumen de la aplicación, cada caso de uso va acompañado de su respectiva tabla y *diagrama de flujo*<sup>3</sup>; además, se plantea un algoritmo para dar a conocer por pasos una breve explicación de la programación.

### 3.2.1. Diagrama de casos de uso

Con el propósito de realizar una explicación más amena, en la Figura 3.1 se observa el diagrama de casos de uso de SARAS, donde se denotan las funciones principales previamente mencionadas en §3.1.3.2.

---

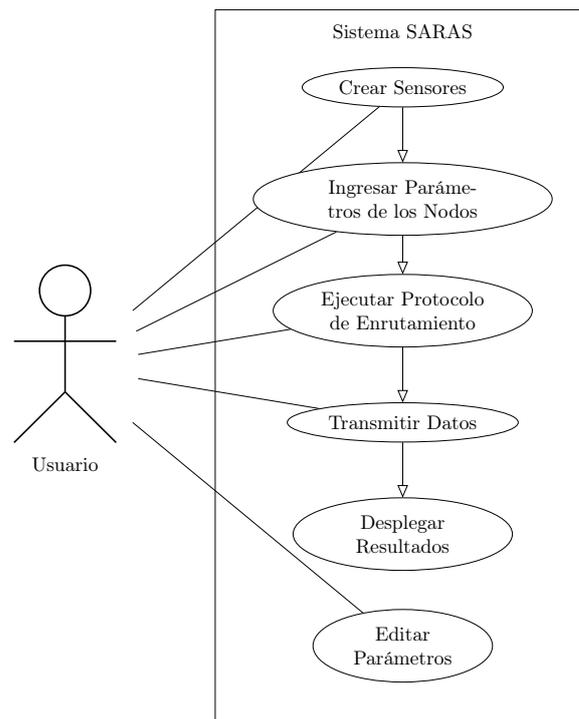
<sup>2</sup>Es un diagrama de comportamiento UML que representa los casos de uso del sistema.

<sup>3</sup>Es un diagrama que representa el proceso, sistema o algoritmo de un programa informático.

Estos 6 casos de uso en su mayoría se encuentran concatenados ya que ciertas funciones tienen dependencia de una anterior, tal como se declara en las tablas descritas en el Apéndice C.

### Figura 3.1

*Diagrama de casos de uso del sistema SARAS.*

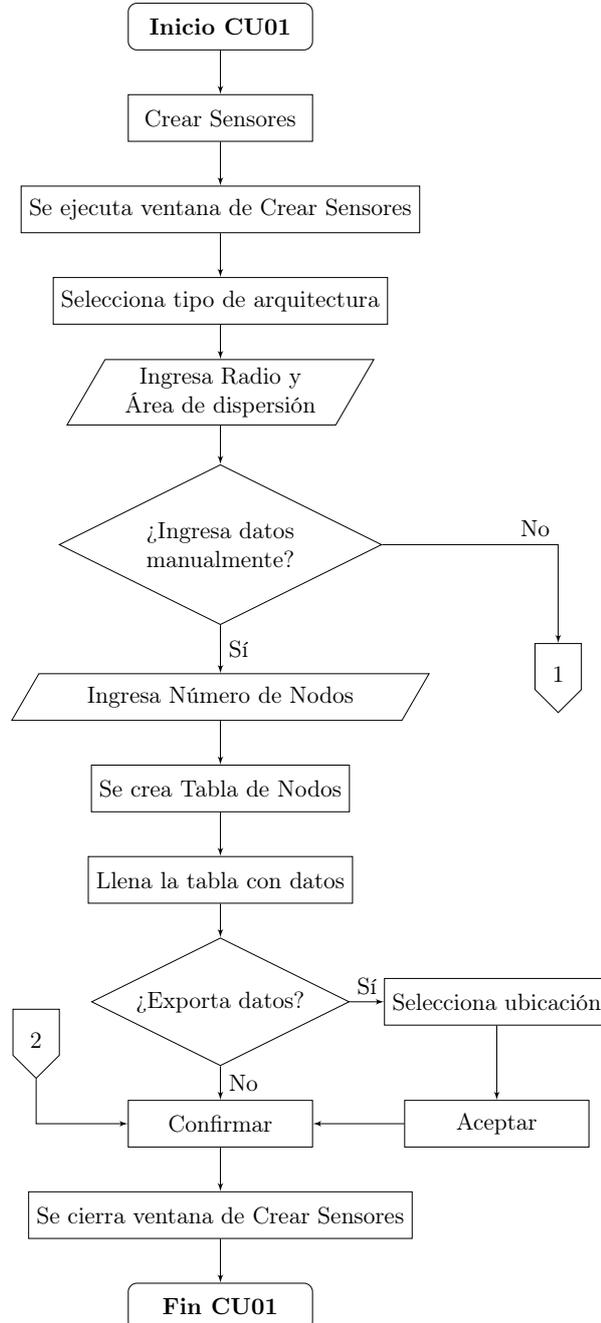


#### 3.2.2. Diagramas de flujo

Otra parte fundamental del diseño, es la elaboración de los diagramas de flujo, los cuales demuestran gráficamente la lógica de cada caso de uso definido.

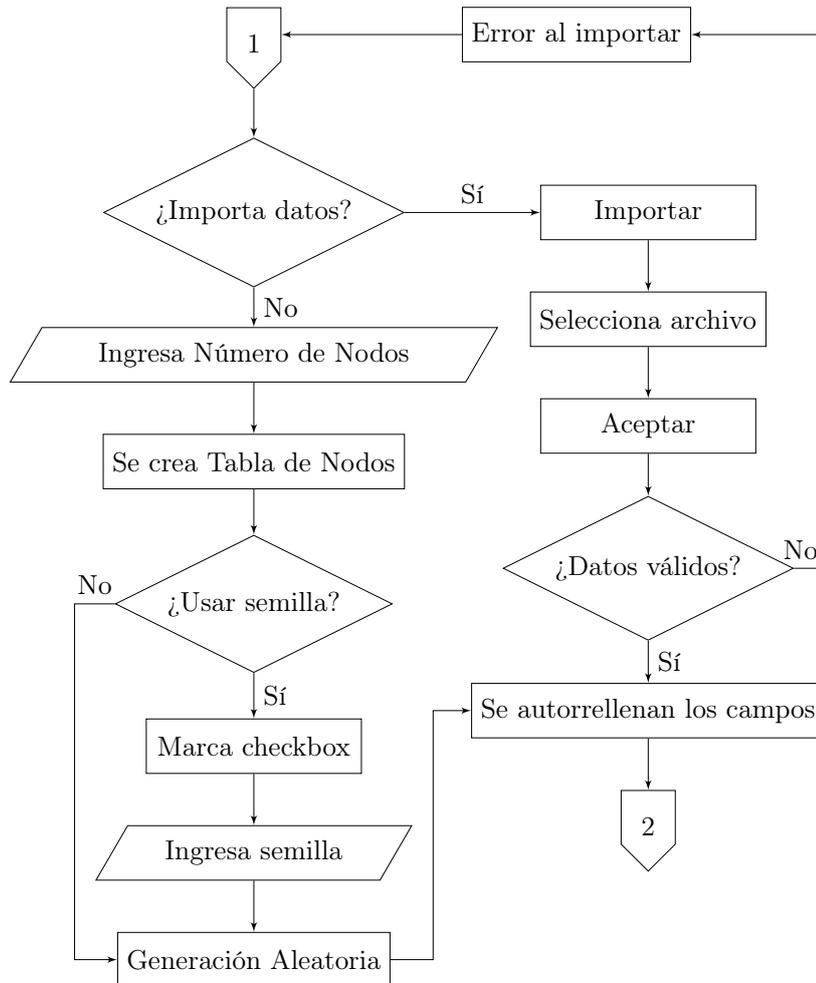
La Figura 3.2 representa el primer caso de uso, que permite que en una ventana desplegable se creen los sensores deseados.

Como se puede apreciar, el primer caso de uso está representado en dos diagramas dependientes el uno del otro, la Figura 3.2a hace referencia al flujo

**Figura 3.2***Diagrama de flujo del CU01.***(a) Flujo Principal.**

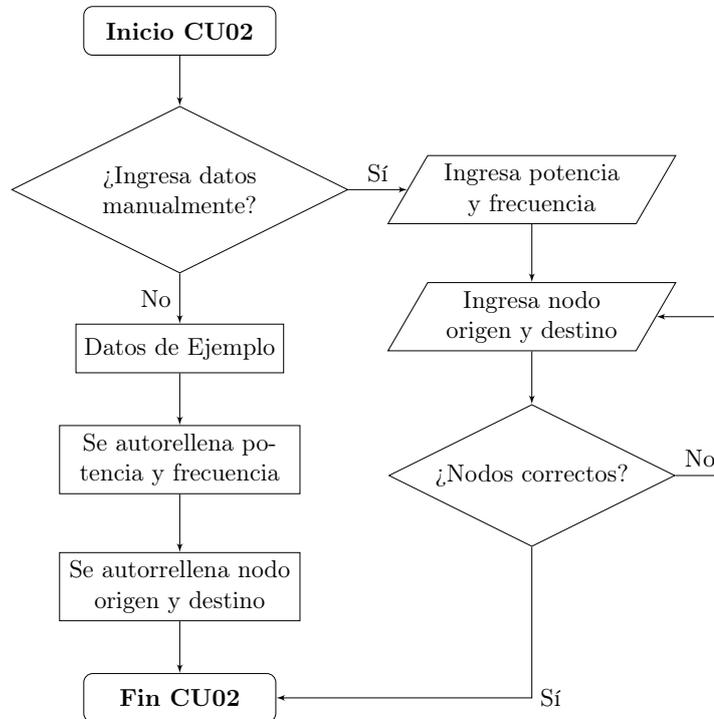
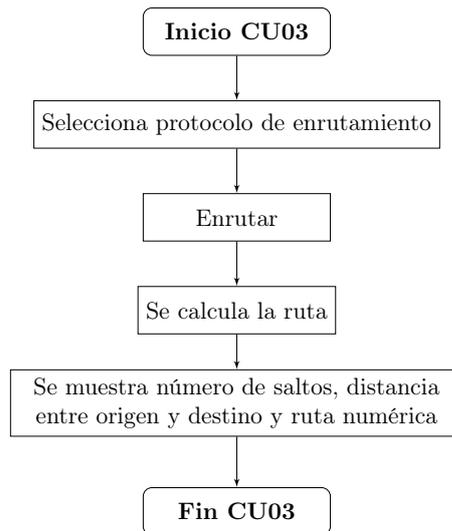
principal del caso, mientras que la Figura 3.2b representa el flujo alterno y de excepción.

(b) *Flujos Alternos y de Excepción.*



La Figura 3.3 expresa la representación del segundo caso de uso, el cual permite que el usuario ingrese los parámetros necesarios para que el sistema realice los cálculos pertinentes.

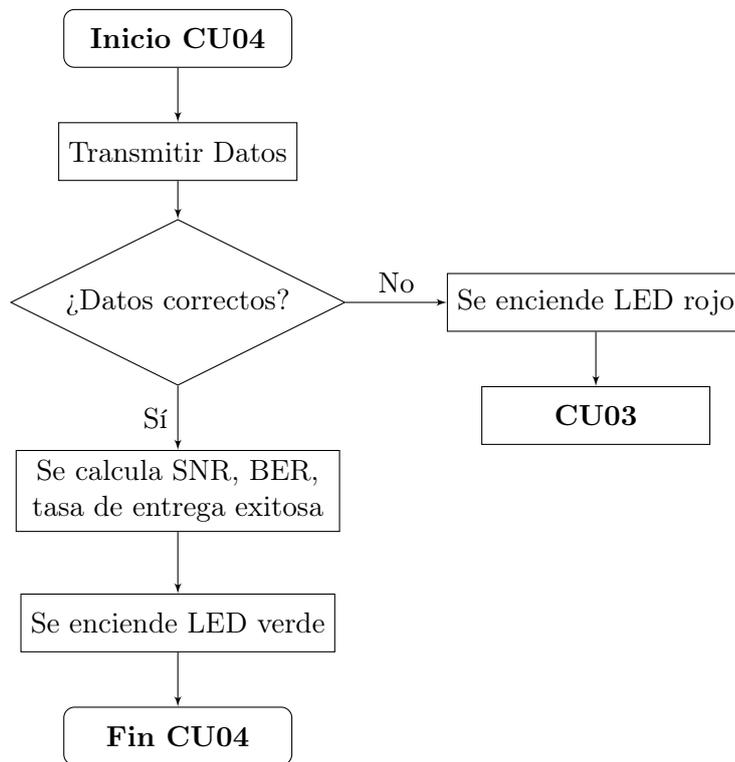
En la Figura 3.4 se encuentra el diagrama de flujo del CU03, que permite trazar una ruta entre un nodo origen y uno destino haciendo uso de un protocolo de enrutamiento.

**Figura 3.3***Diagrama de flujo del CU02.***Figura 3.4***Diagrama de flujo del CU03.*

Seguidamente, la Figura 3.5 explica gráficamente el cuarto caso de uso, este permite simular la eficiencia de una transmisión de bits utilizando los parámetros de entrada y la ruta creada.

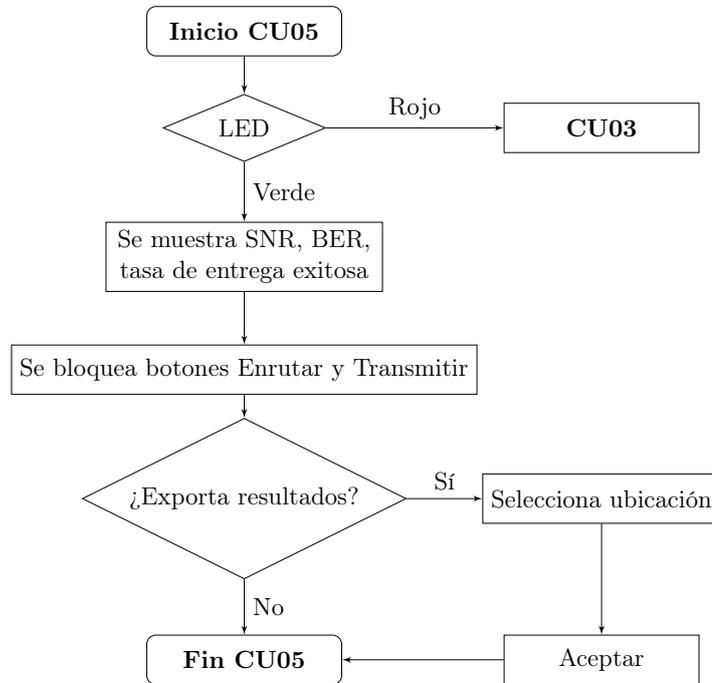
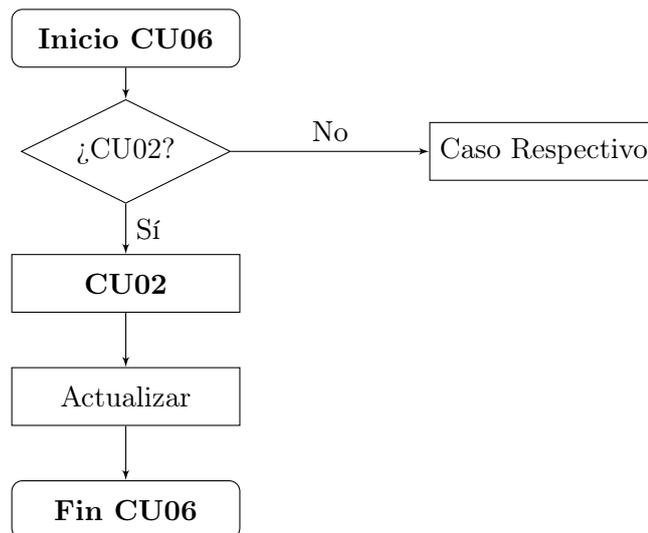
**Figura 3.5**

*Diagrama de flujo del CU04.*



En la Figura 3.6 se observa el diagrama de flujo de su respectivo caso de uso, este permite generar valores legibles para el usuario que indiquen el resultado de la simulación.

Por último, la Figura 3.7 representa de manera gráfica el caso de uso 06, que básicamente permite que el sistema se pueda reutilizar sin necesidad de un reinicio. Vale la pena aclarar que el sistema puede simular nuevamente a partir de cualquier etapa del proceso, es decir, desde cualquier caso de uso.

**Figura 3.6***Diagrama de flujo del CU05.***Figura 3.7***Diagrama de flujo del CU06.*

### 3.2.3. Algoritmo del sistema

Para finalizar con la etapa de diseño, se realiza la explicación por pasos del pseudocódigo del programa en el Algoritmo 1. Es importante destacar que este algoritmo no abarca el 100 % del sistema SARAS, únicamente cuenta con la secuencia general de acciones que se realiza en una simulación, también acompañada de las respectivas ecuaciones previamente documentadas en el Capítulo 2.

---

**Algoritmo 1** Relación de entrega exitosa entre dos nodos.

---

**Entrada:**  $P, f, m, r$

**Salida:**  $P_{\text{exito}}^m$

- 1: Calcular  $d$  usando Ecuación (2.16)
- 2: **para todo**  $DistanciaEntreNodos$
- 3:     **si**  $d < r$  **entonces**
- 4:          $enlace \leftarrow verdadero$                              ▷ Existe comunicación
- 5:     **sino**
- 6:          $enlace \leftarrow falso$                                  ▷ No existe comunicación
- 7:     **fin si**
- 8: **fin para**
- 9:  $ruta \leftarrow ProtocoloEnrutamiento(origen, destino, nodos(enlace))$
- 10:  $aux \leftarrow 0$
- 11: **para todo**  $SaltoEnRuta$
- 12:     Calcular  $S_{level}(P, d)$  usando Ecuación (2.5)
- 13:     Calcular  $T_{loss}(d, f)$  usando Ecuación (2.9)
- 14:     Calcular  $N_{level}(f)$  usando Ecuación (2.10)
- 15:     Calcular  $aux(S_{level}, T_{loss}, N_{level})$  usando Ecuación (2.1)
- 16:      $\gamma \leftarrow \gamma + aux$                                      ▷ Sumatoria de SNRs en cada salto
- 17: **fin para**
- 18:  $\gamma \leftarrow \gamma / numerodesaltosenruta$              ▷ Promedio de SNRs en cada salto
- 19: Calcular  $BER(\gamma)$  usando Ecuación (2.13)
- 20: Calcular  $P_{\text{exito}}^m(BER)$  usando Ecuación (2.15)

---

Otro detalle a mencionar acerca del Algoritmo 1, es que las variables de entrada  $P, f, m, r$  hacen referencia a la Potencia de transmisión, frecuencia, cantidad de información transmitida y rango de comunicación máxima entre nodos respectivamente. Todos a excepción de  $r$  han sido explicados en §2.2.

Con respecto al rango de comunicación máxima, se refiere a la distancia máxima a la que un nodo puede comunicarse con otro (por lo regular es una característica constante del equipo), este también ha sido mencionado como Radio en la Figura 3.2a.

### 3.3. Implementación

La fase de implementación, se ha dividido en tres partes claves para el desarrollo; la primera es la codificación, donde se programa las secciones que no tienen necesidad de una interfaz gráfica para funcionar, como son las partes especificadas en el Algoritmo 1, luego se realiza el diseño de las interfaces gráficas siendo fiel a lo especificado en §3.1.3.1 y por último se elabora un empaquetado del código para que pueda ser ejecutado por el usuario.

#### 3.3.1. Codificación

Para empezar, es importante mencionar que parte del código que ejecuta SARAS fue tomado de (Silva, 2022). En este trabajo, Silva propone un programa con el que simular redes *WSN* con nodos generados aleatoriamente y utilizando un método de desgaste de energía de cada nodo por separado. Tanto la creación aleatoria de nodos como el código de enrutamiento por ruta más corta fue adaptado utilizando este programa.

La versión *raw*<sup>4</sup> de SARAS, comienza con la definición de variables. La mayor parte de la programación está basada en *estructuras*<sup>5</sup>, donde se han almacenado un sinnúmero de campos representando las variables a las que estamos acostumbrados. En el Código 3.1 se definen las variables de semilla,

---

<sup>4</sup>Sin interfaz.

<sup>5</sup>Es un tipo de dato que agrupa datos relacionados mediante contenedores de datos llamados campos.

número de nodos, área de distribución (X Y Z), rango de comunicación, nodo origen, nodo destino, frecuencia, potencia y datos; especificados entre la línea 7 y 15 respectivamente.

### Código 3.1

*Definición de variables.*

```
6 %% Variables
7 dataset.seed = 6;
8 dataset.nodeNo = 25;
9 dataset.area = [500 250 0]; % [m]
10 dataset.range = 100; % [m]
11 dataset.nodeSrc = 1;
12 dataset.nodeDst = 25;
13 dataset.frecuency = 10; % [KHz]
14 dataset.power = 2; % [W]
15 dataset.data = 5000; % [b]
```

*Nota.* Los valores proporcionados son genéricos, estos pueden ser cambiados arbitrariamente.

A continuación, se ejecuta la función RNG (*Random Number Generation*, Generación de Números Aleatorios) para generar valores pseudoaleatorios; la función toma como criterio el valor de semilla siempre y cuando este sea diferente de 0 (Ver Código 3.2).

### Código 3.2

*Establecimiento de valor de semilla.*

```
18 if dataset.seed ~= 0
19     rng(dataset.seed);
20 end
```

La tabla donde se guarda la posición y naturaleza de cada nodo se genera a partir del script especificado en el Código 3.3, cabe recalcar que por el momento SARAS no soporta naturaleza distinta a un nodo ordinario (Ver §2.3.3.1).

**Código 3.3***Creación de tabla de datos de nodos.*

```

23 varNames = {'X','Y','Naturaleza'};
24 varTypes = {'single','single','categorical'};
25 dataset.nodePosition = table('Size',[dataset.nodeNo 3], ...
26     'VariableTypes',varTypes,'VariableNames',varNames);
27 dataset.nodePosition.Properties.VariableUnits = {'m','m',''};
28 dataset.nodePosition.Naturaleza = categorical( ...
29     dataset.nodePosition.Naturaleza,{'Ordinario','Sumidero'});
30 dataset.nodePosition.Naturaleza = fillmissing( ...
31     dataset.nodePosition.Naturaleza, 'constant', "Ordinario");

```

En el Código 3.4, se realiza la generación aleatoria de nodos según las variables especificadas con anterioridad.

**Código 3.4***Generación aleatoria de posición de nodos.*

```

34 for i = 1 : dataset.nodeNo
35     dataset.nodePosition.X(i) = randi([1 dataset.area(1)]);
36     dataset.nodePosition.Y(i) = randi([1 dataset.area(2)]);
37 end
38 disp(dataset.nodePosition)

```

Los ciclos anidados del Código 3.5, se encargan de hacer el cálculo de la distancia euclidiana entre cada nodo y guardar estos valores en una matriz.

**Código 3.5***Cálculo de distancia euclidiana entre nodos.*

```

41 for i = 1 : dataset.nodeNo
42     for j = 1: dataset.nodeNo
43         garbage.x1 = dataset.nodePosition.X(i);
44         garbage.x2 = dataset.nodePosition.X(j);
45         garbage.y1 = dataset.nodePosition.Y(i);
46         garbage.y2 = dataset.nodePosition.Y(j);
47         dataset.euclidiana(i,j) = sqrt((garbage.x1 - ...
48             garbage.x2) ^2 + (garbage.y1 - garbage.y2)^2);
49     end
50 end

```

La generación del grafo es una de las partes que más recursos necesita la aplicación, dependiendo de la cantidad de nodos que se generen. El grafo se genera estableciendo los pesos con relación a las distancias entre nodos y el rango de comunicación que se haya determinado (Ver Código 3.6).

### Código 3.6

*Construcción de grafo.*

```

53 dataset.weights = lt(dataset.euclidiana,dataset.range);
54 G=graph(dataset.weights,'omitselfloops');
55 for a = 1 : height(G.Edges)
56     garbage.s = G.Edges.EndNodes(a,1);
57     garbage.t = G.Edges.EndNodes(a,2);
58     garbage.Z(a,:) = dataset.euclidiana(garbage.s,garbage.t);
59 end
60 G.Edges.Euclidiana = garbage.Z(:,1);

```

A continuación, en el Código 3.7, se realiza el ploteo del grafo en una figura, con la comunicación respectiva entre nodos, tomando en cuenta los valores de distancia entre nodos y el rango de comunicación establecido.

### Código 3.7

*Ploteo de distribución de nodos.*

```

63 close all
64 figure('units','normalized','innerposition',[0 0 1 1], ...
65     'MenuBar','none')
66 p = plot(G,'XData',(dataset.nodePosition.X),'YData',( ...
67     dataset.nodePosition.Y),'MarkerSize',8,'NodeColor', ...
68     '0.80,0.80,0.80','EdgeColor','0.65,0.65,0.65', ...
69     'LineWidth',0.7,'NodeLabelColor','0.90,0.90,0.90');
70 line(dataset.nodePosition.X(dataset.nodeSrc), ...
71     dataset.nodePosition.Y(dataset.nodeSrc),'color','green', ...
72     'marker','o','linestyle','none','markersize',50)
73 line(dataset.nodePosition.X(dataset.nodeDst), ...
74     dataset.nodePosition.Y(dataset.nodeDst),'color','green', ...
75     'marker','o','linestyle','none','markersize',50)
76 grid on
77 hold on
78 title('Dispersion de Nodos')

```

Para este punto, ya es posible realizar el enrutamiento entre el nodo origen y el nodo destino establecidos en el Código 3.1, el identificador de cada nodo viene siendo un número entre 1 y el número de nodos creados; de esta manera, tal como demuestra el Código 3.8, la función de MATLAB `shortestpathtree` permite generar la ruta más corta entre los nodos origen y destino.

### Código 3.8

*Enrutamiento entre nodos.*

```

81 Gpath = shortestpathtree(G,dataset.nodeSrc,dataset.nodeDst);
82 dataset.routepath = shortestpath(G,dataset.nodeSrc, ...
83     dataset.nodeDst);
84 dataset.hopsNo = length(dataset.routepath)-1;
85 highlight(p,dataset.routepath, 'NodeColor', 'green', ...
86     'NodeLabelColor', 'green')
87 disp(dataset.routepath)
88 disp(dataset.hopsNo)

```

Finalmente, la etapa más larga consiste en realizar la transmisión de la cantidad de bits proporcionados entre la ruta encontrada haciendo uso del Código 3.9, para llegar a obtener los resultados de SNR, BER y SDR.

### Código 3.9

*Transmisión de datos entre nodo origen y destino.*

```

91 f = dataset.frecuency;
92 p = dataset.power;
93 SNRsum = 0;
94 for i = 1:dataset.hopsNo
95     d = dataset.euclidiana(dataset.routepath(i), ...
96         dataset.routepath(i+1));
97     absortC = (0.11*f^2)/(1+f^2) + (44*f^2)/(4100+f^2) ...
98         + 2.75e-4*f^2 + 0.003; % Coeficiente de absorcion
99     SS = 20*log10(d); % Propagacion esferica
100     Slevel = 10*(log10(p) - log10(4*pi*d^2) ...
101         - log10(0.67e-18));
102     Tloss = SS + absortC*d*10^(-3);
103     Nlevel = 50 - 18*log10(f);

```

```
104     Dindex = 0; % Debido a la transmision omnidireccional
105     aux = 10^(Slevel/10) - 10^(Tloss/10) - 10^(Nlevel/10) ...
106         + 10^(Dindex/10);
107     SNRsum = SNRsum + aux;
108 end
109 dataset.SNR = 10 * log10(SNRsum/dataset.hopsNo);
110 dataset.BER = 0.5 * (1 - sqrt((10^(dataset.SNR/10))/(1 + ...
111     10^(dataset.SNR/10))));
112 dataset.SDR = (1 - dataset.BER)^(dataset.data);
113 disp(dataset.SNR)
114 disp(dataset.BER)
115 disp(dataset.SDR)
```

El código raw completo se puede encontrar en el Repositorio de GitHub de SARAS.

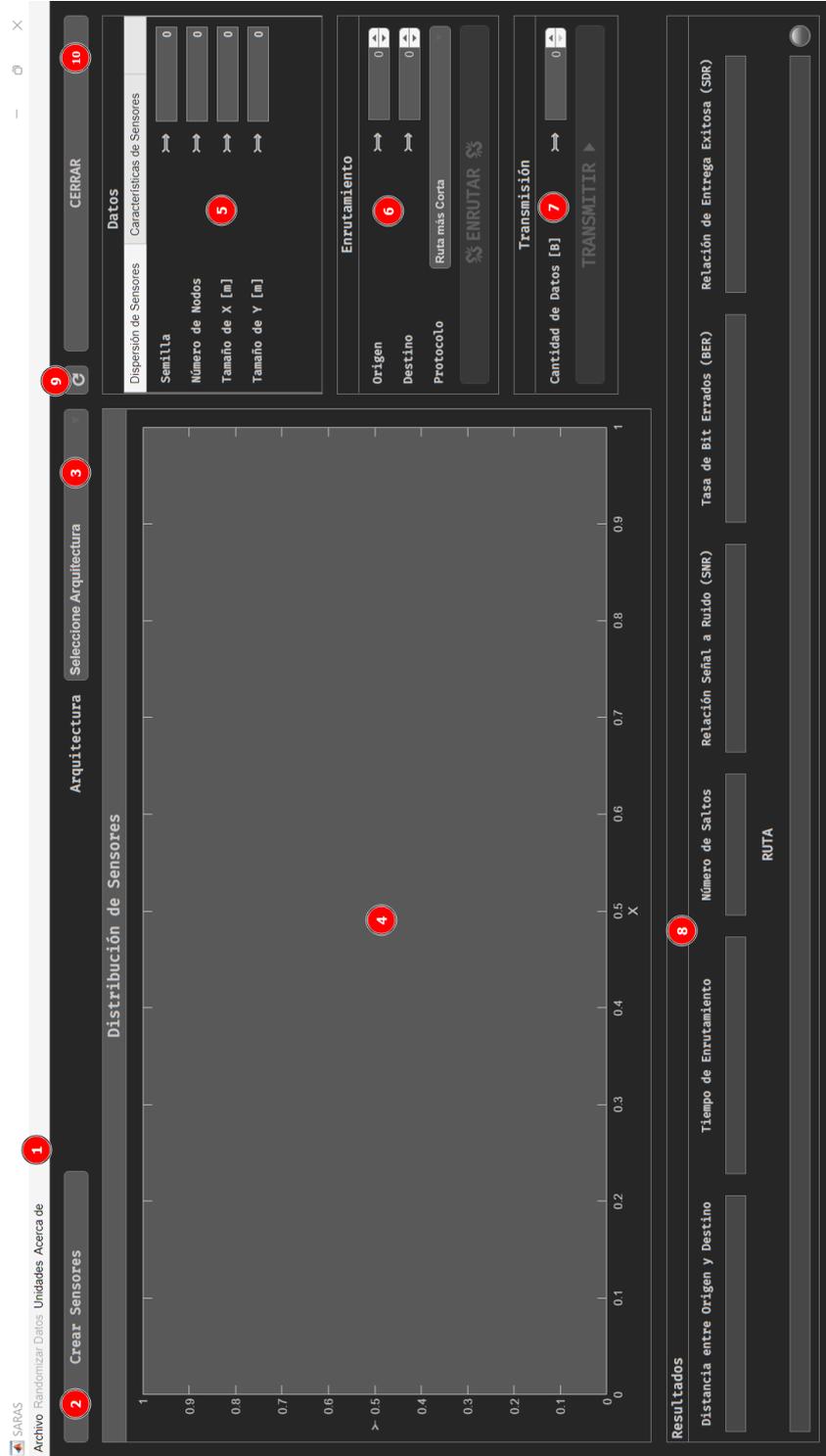
### 3.3.2. Interfaces gráficas de usuario

SARAS consta de dos interfaces con las que interactúa el usuario. La primera ventana, denominada `main.mlapp`, es la interfaz que da inicio a la aplicación. Esta ventana consta de los siguientes elementos principales de simulación (Ver Figura 3.8):

1. Menú de opciones
2. Botón de crear sensores
3. Selección de arquitectura
4. Panel de distribución de sensores
5. Panel de ingreso de datos
6. Panel de enrutamiento de nodos
7. Panel de transmisión de datos
8. Panel de resultados
9. Botón actualizar
10. Botón de cerrar ventana

**Figura 3.8**

*Componentes de la ventana principal de SARAS.*



**Figura 3.9**

*Componentes de la ventana Crear Sensores de SARAS.*



La segunda ventana tiene como propósito la creación de los sensores con los cuales se va a trabajar, de ahí el nombre de `createSensors.mlapp`, en esta se visualiza y se rellenan datos de posición de cada nodo. Esta ventana consta de los siguientes elementos principales de simulación (Ver Figura 3.9):

1. Selección de arquitectura
2. Panel de datos
3. Tabla de disposición de nodos
4. Botón de confirmar
5. Botón de cancelar
6. Botón de importar
7. Botón de exportar
8. Botón de generar aleatoriamente

Como se ha venido desarrollando a lo largo de este capítulo, SARAS está equipado con gran cantidad de funcionalidades que está por demás detallar a profundidad en esta sección, pero es necesario destacar que la aplicación cuenta con un manual de usuario donde se toma en consideración todas y cada una de las características del software. El manual se puede encontrar en el Apéndice D.

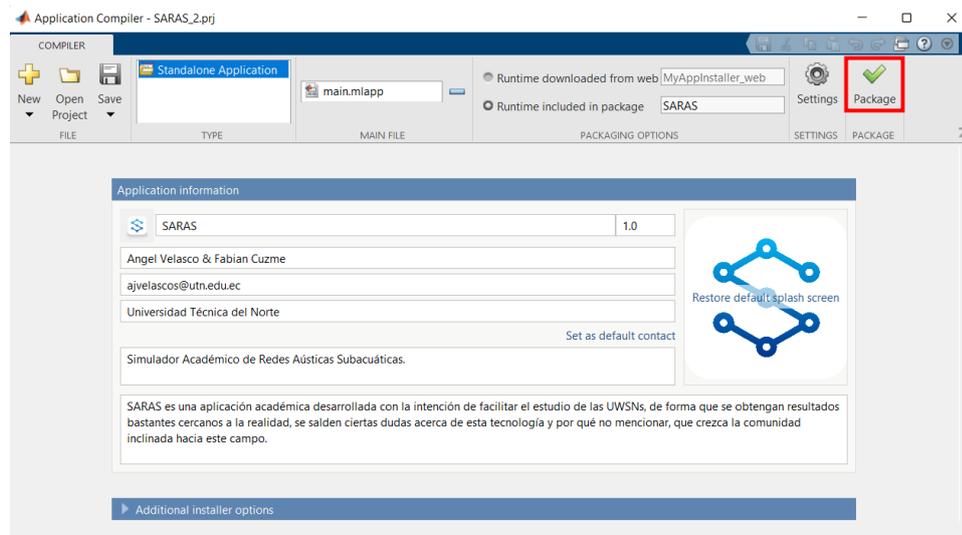
### 3.3.3. Empaquetado del sistema

Por último, en la implementación se toma en cuenta la preparación del empaquetado del código, que servirá como instalador para los usuarios naturales de SARAS, bajo este criterio se procede a utilizar un toolbox llamado MATLAB<sup>®</sup> Compiler. Con esta herramienta se puede ejecutar el programa sin necesidad de contar con una licencia de MATLAB.

Para crear el instalador es necesario haber completado el programa y una vez que se han resuelto todos los requerimientos que se han propuesto se dirige al menú *Share/Standalone Desktop App*; esta acción ejecuta una ventana donde se colocan los datos del software como viene siendo el nombre, los autores, el logo y una pequeña descripción. La Figura 3.10 muestra el proceso de creación del archivo empaquetado que contiene el código de SARAS, dando como resultado el fichero *SARAS.exe*.

**Figura 3.10**

*Empaquetado del software SARAS.*



Teniendo en cuenta que SARAS trabaja con MATLAB Runtime como

compilador (MathWorks, 2022e), se debe hacer un reconocimiento de los requisitos mínimos y recomendados que este admite para su instalación. Como es de esperar, los requisitos de sistema para esta función de MATLAB son dependientes de los requisitos de la versión desde la cual se realizó la compilación, en este caso MATLAB R2022b. En la Tabla 3.2 se presentan los requerimientos de hardware mínimos y recomendados de SARAS.

**Tabla 3.2**

*Requerimientos de Hardware de SARAS.*

<b>Hardware</b>	<b>Mínimo</b>	<b>Recomendado</b>
Procesador	Cualquier Intel o AMD x86-64	Cualquier Intel o AMD x86-64 con cuatro núcleos y soporte de AVX2
Memoria RAM	4 [GB]	8 [GB]
Espacio de almacenamiento	2 [GB]	4 [GB] SSD Altamente Recomendado
Gráficos	No requiere una tarjeta gráfica específica, pero se recomienda una tarjeta gráfica acelerada por hardware compatible con OpenGL 3.3 con memoria GPU de 1 GB	

*Nota.* Tomado de (MathWorks, 2022d).



## Capítulo 4

# Pruebas y Resultados

**RESUMEN:** El último capítulo explica las pruebas que se realizaron para evidenciar y validar los requerimientos establecidos en el capítulo anterior, de manera que se expresen los resultados del proyecto completo.

### 4.1. Uso del Software

En este apartado se establecen determinados casos donde se puede evidenciar el funcionamiento del software.

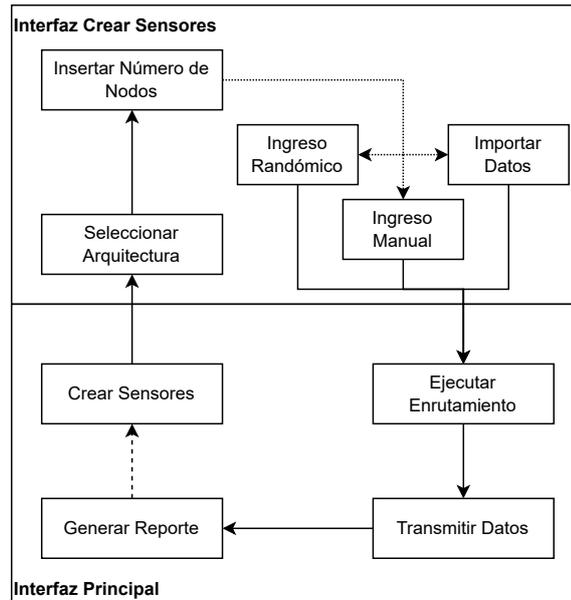
Para dar una mejor idea del orden que debe llevar el usuario al momento de realizar una simulación, la Figura 4.1 representa de manera general la secuencia de utilización del software. Los pasos se enumeran a continuación:

1. Presionar botón Crear Sensores.
2. Seleccionar Arquitectura.
3. Insertar Número de nodos.
  - Ingreso Randómico.
  - Ingreso Manual.
  - Importar Datos.
4. Ejecutar Protocolo de Enrutamiento.
5. Transmitir Datos.

6. Generar Reporte.

**Figura 4.1**

*Diagrama de bloques general del uso de SARAS.*



Se añade información más detallada en el Apéndice D.

La Figura 4.2 muestra cómo el usuario interactúa con la ventana de creación de sensores para registrar la disposición de nodos.

**Figura 4.2**

*Generación aleatoria de 10 nodos en un área de 100 x 150 [m].*

	X [m]	Y [m]	Naturaleza
1	92	95	Ordinario
2	10	42	Ordinario
3	55	144	Ordinario
4	97	24	Ordinario
5	98	144	Ordinario
6	49	121	Ordinario
7	15	64	Ordinario
8	92	119	Ordinario
9	96	99	Ordinario
10	4	128	Ordinario

Arquitectura		Arquitectura 2D
Datos		
Semilla	→	0
Número de Nodos	→	10
Rango de Comunicación [m]	→	25
Tamaño de X [m]	→	100
Tamaño de Y [m]	→	150
Tamaño de Z [m]	→	0

La Figura 4.3 muestra cómo luce el archivo exportado con datos de la

disposición de sensores creados.

**Figura 4.3**

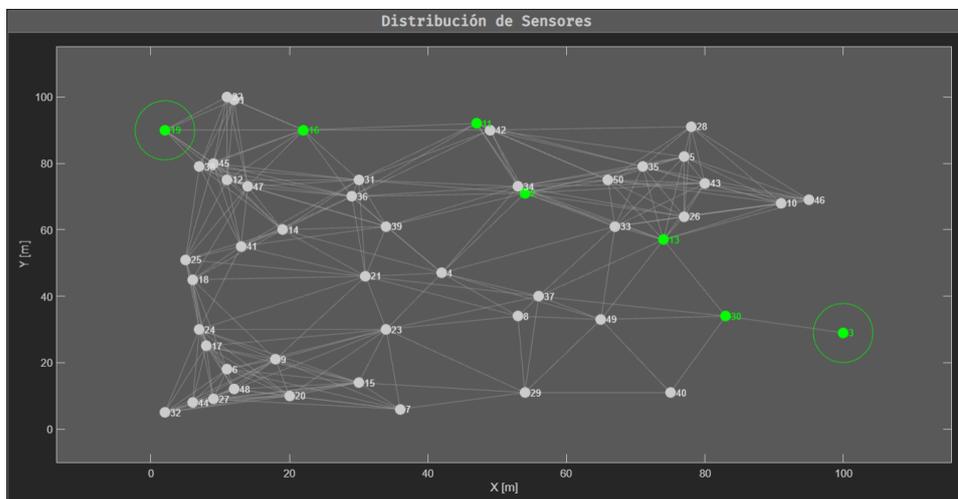
*Datos de sensores exportados en un archivo .csv.*

	A	B	C	D
1	X	Y	Naturaleza	
2	1921	1400	Ordinario	
3	2000	441	Ordinario	
4	723	1480	Ordinario	
5	1993	633	Ordinario	
6	274	768	Ordinario	
7	642	733	Ordinario	
8	1420	1801	Ordinario	
9	1069	495	Ordinario	

En la Figura 4.4 se observa una malla de sensores intercomunicados entre sí, con una ruta resaltada entre los nodos 19 y 3.

**Figura 4.4**

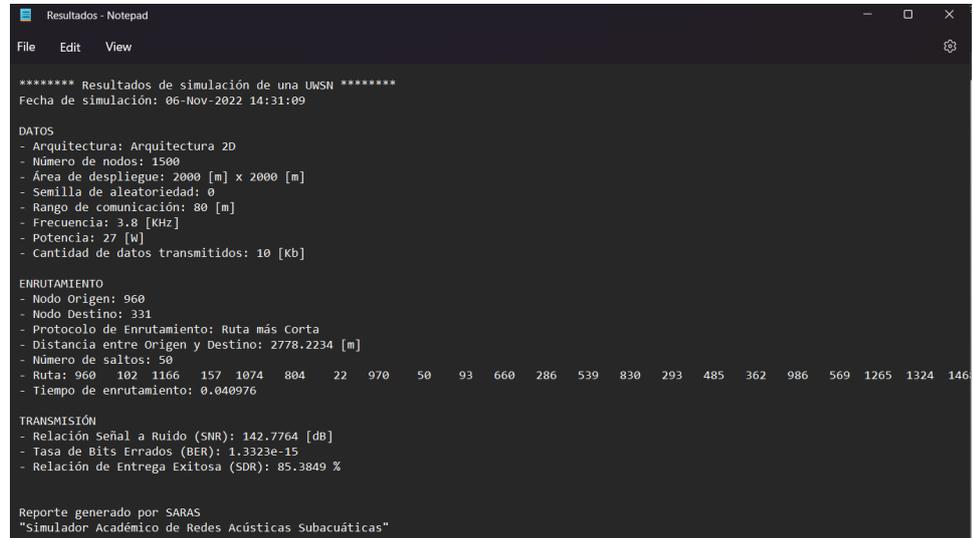
*Plot de distribución de 50 nodos con una ruta entre el nodo 19 y 3.*



La manera en que SARAS genera un reporte de resultados se ve reflejada en la Figura 4.5, mismo archivo que también contiene todos los datos que se usaron en la simulación para llegar a esa respuesta.

### Figura 4.5

*Reporte de simulación exitosa de 1500 nodos.*



```

***** Resultados de simulación de una UWSN *****
Fecha de simulación: 06-Nov-2022 14:31:09

DATOS
- Arquitectura: Arquitectura 2D
- Número de nodos: 1500
- Área de despliegue: 2000 [m] x 2000 [m]
- Semilla de aleatoriedad: 0
- Rango de comunicación: 80 [m]
- Frecuencia: 3.8 [KHz]
- Potencia: 27 [W]
- Cantidad de datos transmitidos: 10 [Kb]

ENRUTAMIENTO
- Nodo Origen: 960
- Nodo Destino: 331
- Protocolo de Enrutamiento: Ruta más Corta
- Distancia entre Origen y Destino: 2778.2234 [m]
- Número de saltos: 50
- Ruta: 960 102 1166 157 1074 804 22 970 50 93 660 286 539 830 293 485 362 986 569 1265 1324 146
- Tiempo de enrutamiento: 0.040976

TRANSMISIÓN
- Relación Señal a Ruido (SNR): 142.7764 [dB]
- Tasa de Bits Errados (BER): 1.3323e-15
- Relación de Entrega Exitosa (SDR): 85.3849 %

Reporte generado por SARAS
"Simulador Académico de Redes Acústicas Subacuáticas"

```

## 4.2. Taller y encuesta

Las pruebas más cruciales en este proyecto dependen de la respuesta otorgada por los estudiantes de Comunicaciones Inalámbricas de la Carrera de Ingeniería en Telecomunicaciones en la encuesta realizada una vez han probado el software. Bajo ese criterio, se ha realizado un taller a 42 asistentes para que generen experticia en el uso de SARAS, teniendo como apoyo dos guías de laboratorio propuestas a continuación.

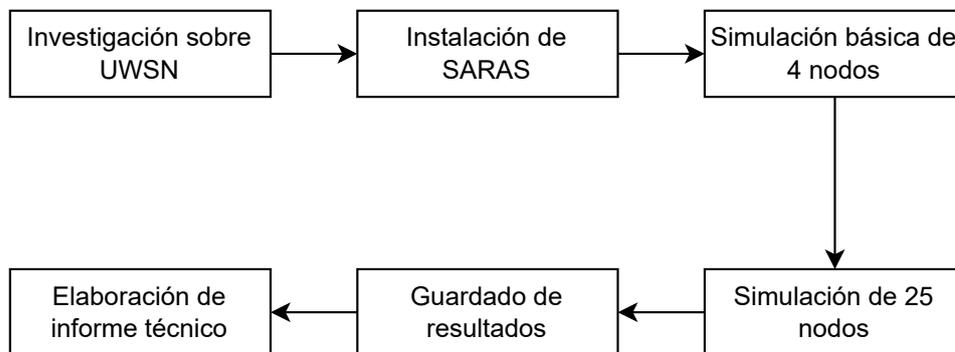
### 4.2.1. Guías de Laboratorio

Para efecto de orientar a los usuarios en la utilización del software se han elaborado dos guías de laboratorio, detalladas en el Apéndice E, que implican desde la instalación de SARAS hasta la simulación y cálculo de SNR, BER y SDR a partir de las fórmulas explicadas.

La primer guía de laboratorio (Apéndice E.1) lleva como título “Instalación y Reconocimiento de SARAS” y en esta se establece la instalación del *Simulador Académico de Redes Acústicas Subacuáticas* así como la exploración de las funciones que se encuentran en el entorno. Su procedimiento se manifiesta en la Figura 4.6.

**Figura 4.6**

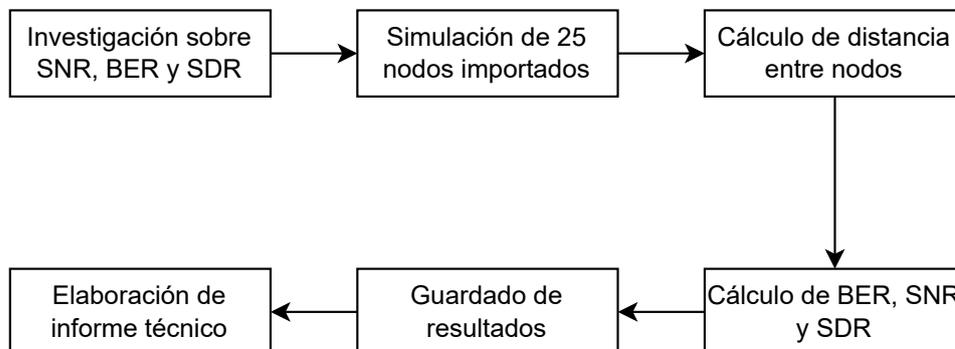
*Diagrama de bloques de Laboratorio 1.*



La segunda guía de laboratorio (Apéndice E.2) lleva como título “Cálculo y simulación de UWSNs usando SARAS” y esta propone realizar una simulación utilizando SARAS, seguido de los cálculos respectivos para comprobar la precisión de la aplicación. Su procedimiento se manifiesta en la Figura 4.7.

**Figura 4.7**

*Diagrama de bloques de Laboratorio 2.*



### 4.2.2. Encuesta

Posterior al cumplimiento del taller, se realizó una encuesta a los asistentes para conocer su opinión acerca de su posición actual en cuanto a conocimiento sobre UWSNs y la experiencia utilizando SARAS. Este procedimiento se valora con el afán de demostrar que los requerimientos establecidos en la fase de análisis se han respetado en el desarrollo de la aplicación; además, la encuesta sirve como comparación del nivel de comprensión de los estudiantes en el tema de redes acuáticas. El formulario de la encuesta se halla en el Apéndice F.

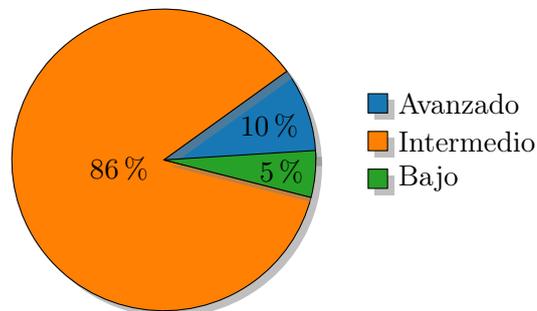
Las preguntas planteadas en la encuesta tienen como enfoque los siguientes propósitos.

- Evaluar el grado de conocimiento de los estudiantes acerca de Redes de Sensores Inalámbricos Subacuáticos una vez recibido el taller.
- Conocer el nivel de interés de los estudiantes por aprender más acerca de las UWSNs luego de haber demostrado el potencial de SARAS como simulador.
- Evaluar el desempeño de SARAS ante las funciones implementadas en la aplicación.
- Analizar la opinión de los usuarios acerca de la fidelidad de SARAS hacia los atributos de calidad manifestados.
- Recaudar las distintas sugerencias que los encuestados proponen como adiciones al proyecto.

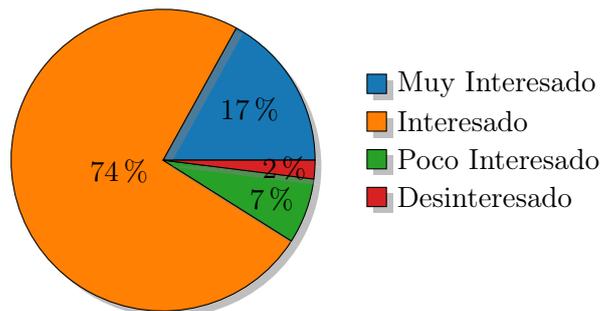
### 4.2.3. Tabulación de la Encuesta

Para terminar, se realiza un resumen de las respuestas obtenidas por cada pregunta de la encuesta, de manera que se muestran resultados visuales y cuantitativos.

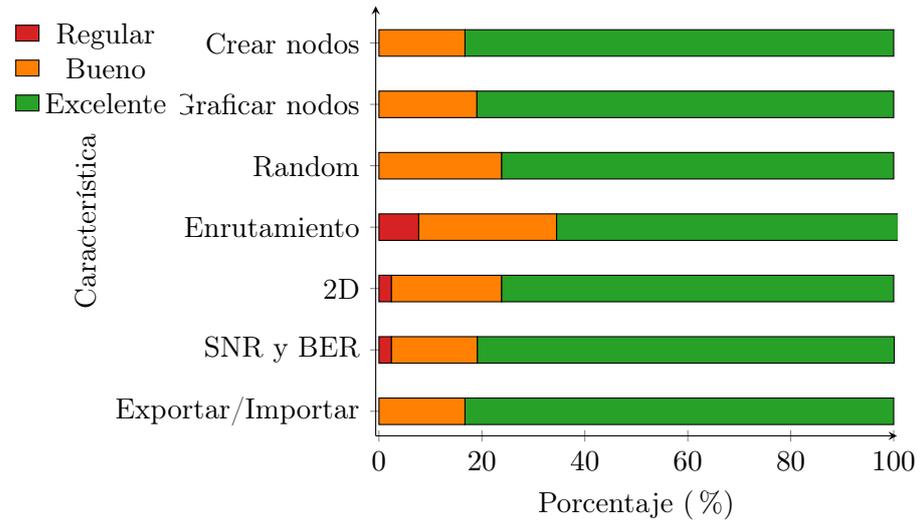
1. ¿Qué nivel de conocimiento considera tener acerca de las UWSNs?



2. ¿Cuán interesado está usted en las Redes de Sensores Inalámbricos Subacuáticos?



3. ¿Cuál cree usted que es el desempeño de SARAS en las funciones enumeradas a continuación?



4. En una escala del 0 al 10 ¿Con cuánto calificaría a SARAS respecto al atributo de calidad “Adecuación Funcional”?

Promedio: 8.93/10

5. En una escala del 0 al 10 ¿Con cuánto calificaría a SARAS respecto al atributo de calidad “Usabilidad”?

Promedio: 9.07/10

6. En una escala del 0 al 10 ¿Con cuánto calificaría a SARAS respecto al atributo de calidad “Mantenibilidad”?

Promedio: 8.93/10

7. En una escala del 0 al 10 ¿Con cuánto calificaría a SARAS respecto al atributo de calidad “Portabilidad”?

Promedio: 9.16/10

**8. ¿Cómo calificaría a SARAS tomando en cuenta la experiencia generada en el taller?**

Promedio: 4.6/5 estrellas

**9. ¿Tiene usted alguna sugerencia adicional para la aplicación?**

- Implementación de aplicación online.
- Implementación de aplicación móvil.
- Implementación para otros sistemas operativos.
- Implementación de arquitectura 3D.
- Más protocolos de enrutamiento.
- Cambiar paleta de colores de la interfaz.
- Mejorar rendimiento.
- Características de sensores por separado.

### **4.3. Discusión**

SARAS permite al usuario crear con facilidad un conjunto de nodos ubicados en un plano bidimensional, con capacidad de modificar sus características de frecuencia, potencia y rango de transmisión, crear una ruta entre dos nodos y comprobar la eficiencia de la ruta en forma de probabilidad de entrega exitosa de un paquete compuesto por cierta cantidad de información.

La importancia de los softwares de simulación en las comunicaciones inalámbricas radica en la disminución de costos y tiempo de implementación de una red. La característica que marca la diferencia entre el software propuesto y otros programas desarrollados como OMNET++, NS2 o NS3, que ofrecen mayor robustez en la simulación de redes (Ronit, 2018), recae en la curva de aprendizaje que estos implican. Bajo este criterio se busca tener

un respaldo para docentes y estudiantes que buscan una herramienta que sirva de apoyo para la educación en esta área y que cuente con una interfaz sencilla e intuitiva.

El software propuesto establece características de adecuación funcional, usabilidad, mantenibilidad y portabilidad del modelo de calidad del producto definido por la *ISO/IEC 25010* en (ISO, 2011).

#### 4.4. Trabajos futuros

SARAS es un sistema que ofrece un alto nivel de escalabilidad, es decir que cuenta con posibilidades para mejorar la aplicación haciendo de esta una alternativa de mayor fiabilidad para los usuarios y con elevada robustez. Bajo este criterio, en esta sección se proponen ciertas metas para futuras versiones del software.

**Posibles funcionalidades** Durante el desarrollo del proyecto, así como en las encuestas realizadas han surgido algunas ideas que pueden enriquecer el estudio de UWSNs con SARAS, dentro de las cuales se encuentran: implementación de arquitectura 3D, nodos de diferente naturaleza, consumo energético, protocolos de localización, configuración individual de sensores y más protocolos de enrutamiento.

**Accesibilidad** Se puede trabajar para adaptar la compatibilidad del software con otros sistemas operativos, implementación de aplicación online y aplicación móvil.

**Optimización** Para terminar, SARAS, como cualquier software en desarrollo, puede ser optimizado con el objeto de mejorar el funcionamiento adecuado y agregar nuevas características.

## Conclusiones

En base a la investigación de un sinnúmero de bibliografías, repositorios, foros y demás sitios de recolección de información, ha sido posible la comprensión y dominio de todo lo relacionado con Redes de Sensores Inalámbricos Subacuáticos, desde sus principales ventajas y desafíos, hasta su compleja mecánica de comunicación en entornos submarinos.

Haciendo uso de la metodología del ciclo de vida clásico del software se ha llevado un orden estipulado en el proceso de elaboración del proyecto. Una de las etapas con mayor relevancia es el análisis, la cual fue planteada tomando en cuenta las recomendaciones aportadas por el estándar IEEE 830 y su definición para cada parte del análisis. Por lo tanto, se establecieron seis requerimientos que a modo general expresan el sucesión de actividades que necesita una simulación.

Partiendo de los requerimientos planteados se realizó el diseño lógico conformado por el diagrama de casos de uso, tablas de cada caso de uso, diagramas de flujo y pseudocódigo; siendo este último un primer acercamiento a lo que sería el código del software, basándose en la información obtenida en la investigación. El diseño posteriormente serviría como estructura para el desarrollo de la aplicación de escritorio, de forma que se codificó el programa con mayor facilidad y teniendo un objetivo claro para cada parte del proceso. Como es evidente, el sistema atravesó por constantes correcciones de errores e implementación de características, para finalmente ofrecer a los usuarios una experiencia agradable. Se validó el funcionamiento y utilidad de la aplicación a través de una encuesta a 42 estudiantes de Comunicaciones Inalámbricas de la Carrera de Ingeniería en Telecomunicaciones, luego de recibir un taller teniendo como protagonista el Simulador Académico de Redes

Acústicas Subacuáticas (SARAS); donde se recibe una gran aceptación del software con valoraciones muy positivas para cada una de sus funciones y atributos de calidad establecidos.

Se crearon algunos recursos que otorgan al usuario una orientación para el uso de SARAS, dentro de estos recursos se encuentra el *Manual de Usuario de SARAS*, donde se describe una guía de instalación y desinstalación de la aplicación, y a su vez el detalle de cada función encontrada. También se elaboraron dos guías de laboratorio que ponen en práctica la aplicación, realizando ejercicios de simulación para efecto de generar experticia en el usuario.

## Recomendaciones

Dentro de los atributos de calidad tomados en cuenta para el desarrollo de SARAS, no se encuentra el atributo de *Eficiencia de Desempeño* (que el software se encuentra optimizado de forma que funcione bajo limitados recursos), por lo tanto, se sugiere verificar los requisitos mínimos y recomendados para no tener problemas posteriores con el uso del simulador.

SARAS está diseñado para operar únicamente con los archivos exportados desde sí mismo. No se recomienda crear archivos desde aplicaciones externas con el objetivo de ser importado en el sistema SARAS, esto podría causar un resultado no esperado.

Para los interesados en la codificación de SARAS y posibles actualizaciones, así como usuarios que deseen realizar un aporte a la aplicación, visitar el Repositorio de GitHub de SARAS.



## Apéndice A

# Formularios de Recopilación de Requerimientos

**RESUMEN:** El presente apéndice, en colaboración con el análisis de requerimientos del software, evidencia el formulario de encuesta para los estudiantes de la materia de Comunicación Inalámbrica en la Carrera de Ingeniería en Telecomunicaciones, así como la entrevista realizada al docente de la materia, acerca de sus conocimientos en redes subacuáticas y su interés en un software de simulación de estas redes.

### A.1. Encuesta acerca de Redes de Sensores Inalámbricos Subacuáticos

Universidad Técnica del Norte  
Facultad de Ingeniería en Ciencias Aplicadas  
Carrera de Ingeniería Electrónica y Redes de Comunicación

Encuesta vinculada al proyecto de titulación “*Diseño de una aplicación académica para el estudio de Redes de Sensores Inalámbricos Subacuáticos en MATLAB*”, por *Angel Jampier Velasco Suárez*.

**Objetivo de la encuesta:** El objetivo de la presente encuesta es saber si los estudiantes de la materia de Comunicación Inalámbrica tienen conocimientos acerca de Redes de Sensores Inalámbricos Subacuáticos (UWSNs) y su necesidad por utilizar un software de simulación para su estudio.

**Indicaciones:** Lea detenidamente las preguntas y seleccione la opción u opciones que crea conveniente. Esta encuesta tomará aproximadamente 5 minutos en contestar.

1. **¿Tiene usted conocimiento acerca de Redes de Sensores Inalámbricos Subacuáticos?**
  - Sí
  - No
  
2. **¿Que nivel de conocimiento considera tener acerca de las UWSNs?**
  - Avanzado
  - Intermedio
  - Bajo
  
3. **¿Cree usted que un software de simulación puede contribuir en el estudio de las redes inalámbricas para entornos subacuáticos?**
  - Sí
  - No
  
4. **¿Cuán interesado estaría usted en una aplicación académica de simulación de UWSNs?**
  - Muy Interesado
  - Interesado
  - Poco Interesado
  - Desinteresado
  
5. **¿Con qué nivel de prioridad calificaría las siguientes características funcionales para el software de simulación propuesto?**

	Baja	Media	Alta
Crear y editar nodos UWSN.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Graficar nodos UWSN.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Randomizar funciones a gran escala.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aplicar protocolos de enrutamiento.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aplicar protocolos de localización de nodos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Simular arquitectura bidimensional.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Simular arquitectura tridimensional.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Simular vehículos autónomos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Calcular atributos de SNR, BER y Tasa de Entrega de Paquetes.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Exportar o importar entradas y salidas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6. ¿Con qué nivel de prioridad calificaría los siguientes atributos de calidad para el software de simulación propuesto?

**Adecuación Funcional:** Que el software proporcione las funciones que satisfagan las necesidades.

**Eficiencia de desempeño:** Que el desempeño del software a nivel de recursos utilizados sea eficiente.

**Compatibilidad:** Que el software soporte dos o más sistemas o componentes para realizar sus funciones.

**Usabilidad:** Que el software sea entendido, aprendido, usado y resulte atractivo para el usuario.

**Fiabilidad:** Que el software desempeñe correctamente sus funciones al usarse bajo ciertas condiciones.

**Seguridad:** Que personas o sistemas no autorizados no puedan leer o modificar la información y datos.

**Mantenibilidad:** Que el software pueda ser modificado efectiva y eficientemente por necesidades futuras.

**Portabilidad:** Que el software pueda ser instalado o reemplazado de forma efectiva y eficiente.

	Baja	Media	Alta
Adecuación Funcional	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eficiencia de Desempeño	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Compatibilidad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Usabilidad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fiabilidad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Seguridad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mantenibilidad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Portabilidad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

7. ¿Qué otra función o funciones cree usted que sería indispensable para el software propuesto?

  
 Ing. Fabián Cuzme MSc.  
 Director de Tesis

  
 Angel Velasco  
 Tesista

## A.2. Entrevista al Docente sobre Redes de Sensores Inalámbricos Subacuáticos

Universidad Técnica del Norte  
Facultad de Ingeniería en Ciencias Aplicadas  
Carrera de Ingeniería Electrónica y Redes de Comunicación

Entrevista vinculada al proyecto de titulación “*Diseño de una aplicación académica para el estudio de Redes de Sensores Inalámbricos Subacuáticos en MATLAB*”, por *Angel Jampier Velasco Suárez*.

**Objetivo de la entrevista:** El objetivo de la presente entrevista es conocer acerca del interés del docente de la materia de Comunicaciones Inalámbricas por utilizar un software de simulación para el estudio de Redes de Sensores Inalámbricos Subacuáticos (UWSNs).

**Indicaciones:** Lea detenidamente las preguntas y seleccione la opción u opciones que crea conveniente. Esta entrevista tomará aproximadamente 5 minutos en contestar.

1. **¿Tiene usted conocimiento acerca de Redes de Sensores Inalámbricos Subacuáticos?**

- Sí
- No

2. **¿Que nivel de conocimiento considera tener acerca de las UWSNs?**

- Avanzado
- Intermedio
- Bajo

3. **¿Cree usted que un software de simulación puede contribuir en el estudio de UWSNs?**

- Sí
- No

4. **¿Cree usted que la utilización de esta aplicación es viable en la materia de Comunicación Inalámbrica?**

- Sí
- No

5. **¿Cuán interesado estaría usted en una aplicación académica de simulación de UWSNs?**

- Muy Interesado
- Interesado
- Poco Interesado
- Desinteresado

6. **¿Con qué nivel de prioridad calificaría las siguientes funcionalidades para el software de simulación propuesto?**

	Baja	Media	Alta
Crear y editar nodos UWSN.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Graficar nodos UWSN.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Randomizar funciones a gran escala.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aplicar protocolos de enrutamiento.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aplicar protocolos de localización de nodos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Simular arquitectura bidimensional.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Simular arquitectura tridimensional.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Simular vehículos autónomos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Calcular atributos de SNR, BER y Tasa de Entrega de Paquetes.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Exportar o importar entradas y salidas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

7. **¿Con qué nivel de prioridad calificaría los siguientes atributos de calidad para el software de simulación propuesto?**

**Adecuación Funcional:** Que el software proporcione las funciones que satisfagan las necesidades.

**Eficiencia de desempeño:** Que el desempeño del software a nivel de recursos utilizados sea eficiente.

**Compatibilidad:** Que el software soporte dos o más sistemas o componentes para realizar sus funciones.

**Usabilidad:** Que el software sea entendido, aprendido, usado y resulte atractivo para el usuario.

**Fiabilidad:** Que el software desempeñe correctamente sus funciones al usarse bajo ciertas condiciones.

**Seguridad:** Que personas o sistemas no autorizados no puedan leer o modificar la información y datos.

**Mantenibilidad:** Que el software pueda ser modificado efectiva y eficientemente por necesidades futuras.

**Portabilidad:** Que el software pueda ser instalado o reemplazado de forma efectiva y eficiente.

	Baja	Media	Alta
Adecuación Funcional	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eficiencia de Desempeño	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Compatibilidad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Usabilidad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fiabilidad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Seguridad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mantenibilidad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Portabilidad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

8. ¿Qué otra función o funciones considera indispensable para el software propuesto?

  
 \_\_\_\_\_  
**Ing. Fabián Cuzme MSc.**  
 Director de Tesis

  
 \_\_\_\_\_  
**Angel Velasco**  
 Tesista

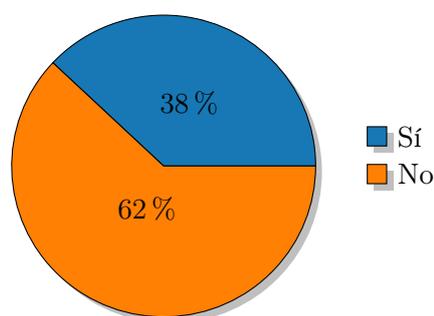
## Apéndice B

# Tabulación de Respuestas Recopiladas

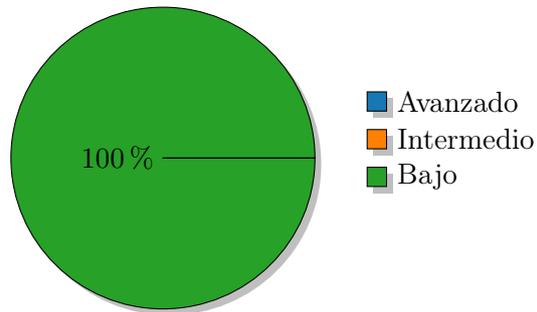
**RESUMEN:** El presente apéndice, muestra las respuestas de los encuestados a través de los formularios evidenciados en el Apéndice A, utilizando diagramas que aportarán a la comprensión de los resultados.

### B.1. Tabulación de Encuesta

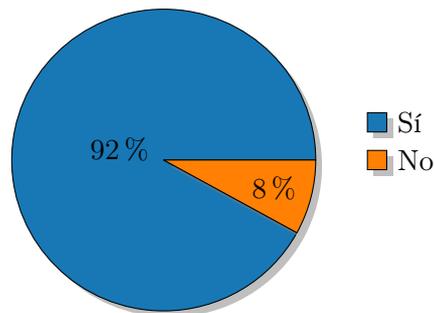
1. ¿Tiene usted conocimiento acerca de Redes de Sensores Inalámbricos Subacuáticos?



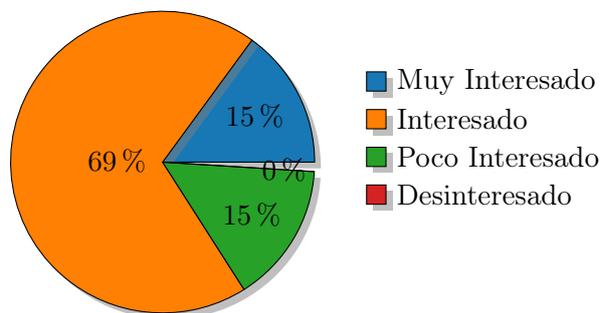
2. ¿Qué nivel de conocimiento considera tener acerca de las UWSNs?



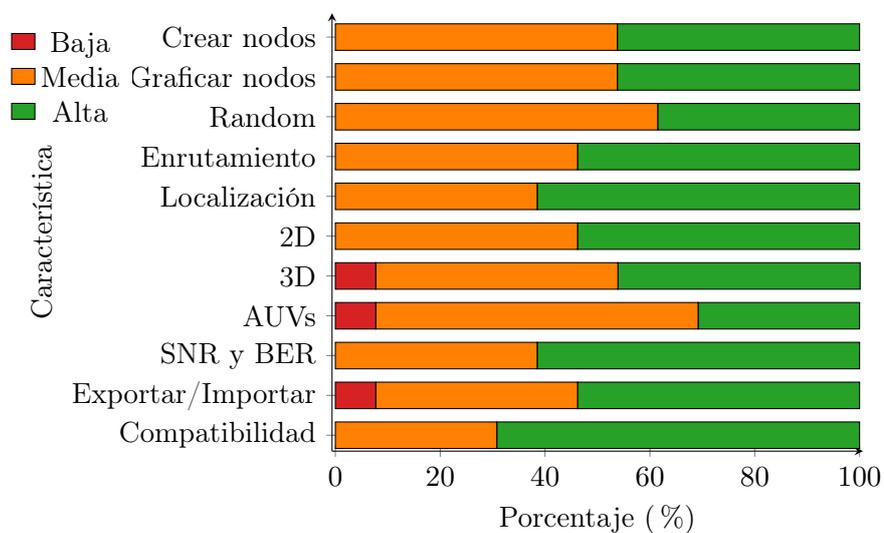
3. ¿Cree usted que un software de simulación puede contribuir en el estudio de las redes inalámbricas para entornos subacuáticos?



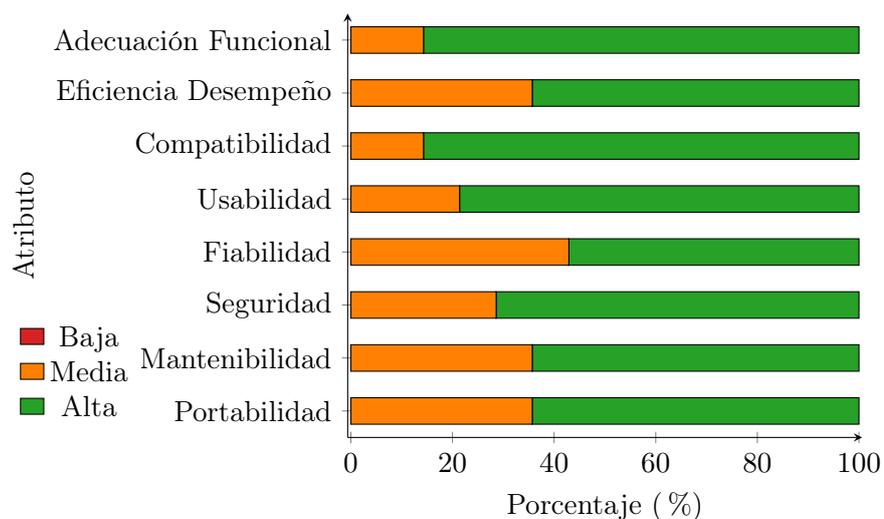
4. ¿Cuán interesado estaría usted en una aplicación académica de simulación de UWSNs?



5. ¿Con qué nivel de prioridad calificaría las siguientes características funcionales para el software de simulación propuesto?



6. ¿Con qué nivel de prioridad calificaría los siguientes atributos de calidad para el software de simulación propuesto?



7. ¿Qué otra función o funciones cree usted que sería indispensable para el software propuesto?

- Implementación de Sniffer
- Simulación de consumo energético

## B.2. Respuestas de Entrevista

Universidad Técnica del Norte  
Facultad de Ingeniería en Ciencias Aplicadas  
Carrera de Ingeniería Electrónica y Redes de Comunicación

Entrevista vinculada al proyecto de titulación “*Diseño de una aplicación académica para el estudio de Redes de Sensores Inalámbricos Subacuáticos en MATLAB*”, por *Angel Jampier Velasco Suárez*.

**Objetivo de la entrevista** El objetivo de la presente entrevista es conocer acerca del interés del docente de la materia de Comunicaciones Inalámbricas por utilizar un software de simulación para el estudio de Redes de Sensores Inalámbricos Subacuáticos (UWSNs).

1. **¿Tiene usted conocimiento acerca de Redes de Sensores Inalámbricos Subacuáticos?**
  - Sí
  - No
  
2. **¿Que nivel de conocimiento considera tener acerca de las UWSNs?**
  - Avanzado
  - Intermedio
  - Bajo
  
3. **¿Cree usted que un software de simulación puede contribuir en el estudio de UWSNs?**
  - Sí
  - No

4. **¿Cree usted que la utilización de esta aplicación es viable en la materia de Comunicación Inalámbrica?**

- Sí
- No

5. **¿Cuán interesado estaría usted en una aplicación académica de simulación de UWSNs?**

- Muy Interesado
- Interesado
- Poco Interesado
- Desinteresado

6. **¿Con qué nivel de prioridad calificaría las siguientes funcionalidades para el software de simulación propuesto?**

	Baja	Media	Alta
Crear y editar nodos UWSN.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Graficar nodos UWSN.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Randomizar funciones a gran escala.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Aplicar protocolos de enrutamiento.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Aplicar protocolos de localización de nodos.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Simular arquitectura bidimensional.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Simular arquitectura tridimensional.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Simular vehículos autónomos.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Calcular atributos de SNR, BER y Tasa de Entrega de Paquetes.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Exportar o importar entradas y salidas.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

7. **¿Con qué nivel de prioridad calificaría los siguientes atributos de calidad para el software de simulación propuesto?**

**Adecuación Funcional:** Que el software proporcione las funciones que satisfagan las necesidades.

**Eficiencia de desempeño:** Que el desempeño del software a nivel de recursos utilizados sea eficiente.

**Compatibilidad:** Que el software soporte dos o más sistemas o componentes para realizar sus funciones.

**Usabilidad:** Que el software sea entendido, aprendido, usado y resulte atractivo para el usuario.

**Fiabilidad:** Que el software desempeñe correctamente sus funciones al usarse bajo ciertas condiciones.

**Seguridad:** Que personas o sistemas no autorizados no puedan leer o modificar la información y datos.

**Mantenibilidad:** Que el software pueda ser modificado efectiva y eficientemente por necesidades futuras.

**Portabilidad:** Que el software pueda ser instalado o reemplazado de forma efectiva y eficiente.

	Baja	Media	Alta
Adecuación Funcional	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Eficiencia de Desempeño	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Compatibilidad	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Usabilidad	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Fiabilidad	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Seguridad	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Mantenibilidad	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Portabilidad	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

8. **¿Qué otra función o funciones considera indispensable para el software propuesto?**

Ninguna.

---

**Angel Velasco**  
Tesista



---

**Ing. Jaime Michilena, MSc.**  
Docente de la Materia

## Apéndice C

# Casos de Uso

**RESUMEN:** En este apéndice se establece una parte importante de la especificación de requerimientos, se trata de la definición de casos de uso; donde se presentan todos los requisitos funcionales que previamente se han planteado para el desarrollo del sistema SARAS.

Antes de comenzar, es importante mencionar que la abreviatura CU hace referencia a “Caso de Uso”. Cada Tabla presentada manifiesta un caso que el usuario está tendente a realizar.

**Tabla C.1**

*Caso de Uso: Crear Sensores.*

<b>Caso de Uso</b>	<b>Crear Sensores</b>
<b>ID. del CU</b>	<b>01 Estado En elaboración/Propuesto/<u>Validado</u></b>
<b>Actor</b>	Usuario
<b>Descripción</b>	Permite que en una ventana desplegable se creen los sensores deseados.
<b>Pre-condiciones</b>	Si el usuario planea realizar la importación de un archivo con datos de sensores, lo debe haber exportado con anterioridad.

**Flujo****Principal**

1. El usuario clica sobre el botón “*Crear/Editar Sensores*”.
2. El sistema despliega una ventana flotante para crear los nodos.
3. El usuario en todo momento cuenta con la opción de anular el proceso usando el botón “*Cancelar*”, haciendo que el sistema cierre esta ventana.
4. El usuario selecciona el tipo de *Arquitectura UWSN*: *Arquitectura 2D* o *3D*.
5. El usuario ingresa valores de *Radio de Comunicación*, y tamaño del lienzo en cada coordenada.
6. Si se desea ingresar los datos manualmente, el usuario debe llenar el campo de *Número de Nodos*.
7. El sistema crea una tabla con número de filas igual al número de nodos.
8. El usuario ingresa en la tabla valores de posición para cada nodo.
9. El usuario puede exportar los datos generados usando el botón “*Exportar*”
10. El usuario confirma los datos ingresados clicando el botón “*Confirmar*”.
11. El sistema cierra la ventana flotante.
12. El sistema plotea los nodos creados en el lienzo de la ventana principal.

**Flujos  
Alternos**

Punto 6. en Flujo Principal

A. Si el usuario desea importar datos de sensores a partir de un archivo de datos previamente exportado.

1. El usuario debe clicar en el botón “*Importar*”
2. El sistema despliega una ventana flotante de explorador de archivos.
3. El usuario tiene que buscar y seleccionar el archivo de datos de nodos.
4. El sistema autorellena los campos de la ventana actual con los datos del archivo importado.
5. Retorna al Punto 10.

B. El usuario desea generar valores aleatorios para la posición de los nodos.

1. El usuario debe ingresar el número de nodos a simular.
2. El sistema crea una tabla con número de filas igual al número de nodos.
3. El usuario decide si utilizar o no un valor de *semilla*, marcando un checkbox en caso afirmativo y asignando el valor de semilla.
4. El usuario debe clicar el botón “*Generación Aleatoria*”.
5. El sistema autorellena la tabla con valores aleatorios.
6. Retorna al Punto 10.

**Flujos de  
Excepción**

Punto 3. del Flujo Alterno A.

A. Si el usuario selecciona un archivo que no es compatible con SARAS o se encuentra corrupto.

1. El sistema cancela la lectura del archivo.
2. El sistema manifiesta el error al usuario y realiza la petición de un nuevo archivo.

**Post-  
condiciones**

Nodos creados.

---

Tabla C.2

*Caso de Uso: Ingresar Parámetros de los Nodos.*

<b>Caso de Uso</b>	Ingresar Parámetros de los Nodos
<b>ID. del CU</b>	<b>02 Estado En elaboración/Propuesto/Validado</b>
<b>Actor</b>	Usuario
<b>Descripción</b>	Permite que el usuario ingrese los parámetros necesarios para que el sistema realice los cálculos pertinentes.
<b>Pre-condiciones</b>	CU01 El usuario tiene que haber creado los nodos UWSN.
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. Si se desea ingresar los datos manualmente, el usuario debe llenar los campos de <i>Potencia de Transmisión</i> y <i>Frecuencia de la Señal</i>.</li> <li>2. El usuario llena los campos de <i>Nodo Origen</i> y <i>Nodo Destino</i>.</li> </ol>
<b>Flujos Alternos</b>	<p>Punto 1. del Flujo Principal.</p> <p>A. Si el usuario no tiene una noción de los datos que el programa requiere, este puede ofrecer ayuda con valores cercanos a la realidad.</p> <ol style="list-style-type: none"> <li>1. El usuario clicca el botón "<i>Datos de Ejemplo</i>".</li> <li>2. El sistema rellena los campos de <i>Potencia de Transmisión</i> y <i>Frecuencia de la Señal</i> con valores de un rango predefinido.</li> <li>3. El sistema rellena los campos de <i>Nodo Origen</i> y <i>Nodo Destino</i> con 2 nodos aleatorios de los creados previamente.</li> <li>4. Fin del caso.</li> </ol>

---

<b>Flujos de Excepción</b>	Punto 2. del Flujo Principal.
	A. Si el usuario define como nodo origen o destino, un número de nodo mayor al número de nodos predefinidos. <ol style="list-style-type: none"> <li>1. El sistema arroja un error y realiza la petición de un valor dentro del rango permitido.</li> </ol>
<b>Post-condiciones</b>	Parámetros Ingresados.

---

**Tabla C.3**

*Caso de Uso: Ejecutar Protocolo de Enrutamiento.*

---

<b>Caso de Uso</b>	Ejecutar Protocolo de Enrutamiento
<b>ID. del CU</b>	<b>03 Estado En elaboración/Propuesto/Validado</b>
<b>Actor</b>	Usuario
<b>Descripción</b>	Permite trazar una ruta entre un nodo origen y uno destino haciendo uso de un protocolo de enrutamiento.
<b>Pre-condiciones</b>	CU02 El usuario tiene que haber ingresado los números del nodo origen y nodo destino.
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el protocolo de enrutamiento a utilizar.</li> <li>2. El usuario clicla en el botón “<i>Enrutar</i>”.</li> <li>3. El sistema aplica el protocolo programado para hallar la ruta entre el origen y el destino.</li> <li>4. El sistema resalta en la gráfica los nodos que forman parte de la ruta encontrada.</li> <li>5. El sistema muestra el número de saltos que contiene la ruta, la distancia relativa entre el origen y el destino, y la ruta en formato numérico.</li> </ol>
<b>Flujos Alternos</b>	Ninguno.

<b>Flujos de Excepción</b>	Punto 1. del Flujo Principal.  A. Si el usuario no ha seleccionado un protocolo de enrutamiento.  1. El sistema arroja un error y realiza la petición de un protocolo de enrutamiento.
<b>Post-condiciones</b>	Ruta generada.

Tabla C.4

*Caso de Uso: Transmitir Datos.*

Caso de Uso	Transmitir Datos
<b>ID. del CU</b>	<b>04 Estado En elaboración/Propuesto/Validado</b>
<b>Actor</b>	Usuario
<b>Descripción</b>	Permite simular la eficiencia de una transmisión de bits utilizando los parámetros de entrada y la ruta creada.
<b>Pre-condiciones</b>	CU02 El usuario tiene que haber ingresado los parámetros de los nodos. CU03 El sistema tiene que haber creado la ruta de comunicación entre el origen y el destino.
<b>Flujo Principal</b>	1. El usuario clicla en el botón “ <i>Transmitir Datos</i> ”.  2. El sistema realiza cálculos de SNR, BER y Tasa de Entrega Exitosa.  3. El sistema hace saber al usuario cuando la simulación se ha realizado exitosamente mediante un LED.
<b>Flujos Alternos</b>	Ninguno.

<b>Flujos de Excepción</b>	<p>Punto 3. del Flujo Principal.</p> <p>A. Si el usuario ha ingresado valores en los parámetros fuera de los rangos normales.</p> <ol style="list-style-type: none"> <li>1. El sistema bloquea el botón “<i>Transmitir Datos</i>” hasta que se se corrijan los errores.</li> <li>2. El sistema arroja un error y realiza la petición de corregir los parámetros.</li> <li>3. El sistema indica que ha habido un error en la simulación poniendo el LED de color rojo.</li> </ol>
<b>Post-condiciones</b>	LED de color verde. Simulación exitosa.

Tabla C.5

*Caso de Uso: Desplegar Resultados.*

<b>Caso de Uso</b>	Desplegar Resultados
<b>ID. del CU</b>	<b>05 Estado En elaboración/Propuesto/Validado</b>
<b>Actor</b>	Usuario
<b>Descripción</b>	Permite generar valores legibles para el usuario que indiquen el resultado de la simulación.
<b>Pre-condiciones</b>	CU04 El usuario tiene que haber clicado el botón “ <i>Transmitir Datos</i> ”.
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra los resultados de SNR, BER y Tasa de Entrega Exitosa.</li> <li>2. El sistema bloquea el botón “<i>Enrutar</i>” y el botón “<i>Transmitir Datos</i>” hasta entrar en el CU06.</li> </ol>

<b>Flujos Alternos</b>	<p>Punto 1. del Flujo Principal.</p> <p>A. Si el usuario desea tener un reporte de los resultados obtenidos en la simulación.</p> <ol style="list-style-type: none"> <li>1. El usuario clicca el botón “<i>Exportar</i>”.</li> <li>2. El sistema despliega una ventana flotante de explorador de archivos.</li> <li>3. El usuario busca y selecciona la ubicación donde guardar el archivo de resultados.</li> <li>4. El sistema exporta un archivo de resultados.</li> </ol>
<b>Flujos de Excepción</b>	<p>Punto 1. del Flujo Principal.</p> <p>A. Si el sistema indica una simulación no exitosa en el CU04.</p> <ol style="list-style-type: none"> <li>1. El sistema no muestra resultados.</li> </ol>
<b>Post-condiciones</b>	Resultados desplegados.

Tabla C.6

*Caso de Uso: Editar Parámetros.*

<b>Caso de Uso</b>	Editar Parámetros
<b>ID. del CU</b>	<b>06 Estado En elaboración/Propuesto/Validado</b>
<b>Actor</b>	Usuario
<b>Descripción</b>	Permite que el sistema se pueda reutilizar sin necesidad de un reinicio.
<b>Pre-condiciones</b>	El sistema debe haber sido manipulado en cualquiera de los casos especificados.
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El usuario realiza las funciones del CU01.</li> <li>2. El usuario continúa el proceso especificado en los casos de uso.</li> </ol>

---

<b>Flujos Alternos</b>	<p>Punto 1. del Flujo Principal.</p> <p>A. Si el usuario desea puede obviar el CU01.</p> <ol style="list-style-type: none"><li>1. El usuario realiza las funciones del CU02, CU03 o CU04.</li></ol> <p>B. Si el usuario cambia el valor de arquitectura UWSN.</p> <ol style="list-style-type: none"><li>1. El usuario realiza las funciones del CU01 en adelante.</li></ol> <p>C. Si el usuario cambia el valor de Protocolo de Enrutamiento.</p> <ol style="list-style-type: none"><li>1. El usuario realiza las funciones del CU03 en adelante.</li></ol>
<b>Flujos de Excepción</b>	Ninguna.
<b>Post-condiciones</b>	Edición exitosa.

---



## Apéndice D

# Manual de Usuario de SARAS

*La información sólo es útil cuando puede ser entendida.*

Muriel Cooper

**RESUMEN:** En este apéndice se recopilan todas las funciones del *Simulador Académico de Redes Acústicas Subacuáticas* sin excepción, partiendo desde su instalación hasta su desinstalación. Este manual proporciona una explicación detallada de cada panel, campo o botón encontrado en la *Graphical User Interface*, Interfaz Gráfica de Usuario.

### D.1. Introducción

SARAS es una aplicación académica desarrollada con la intención de facilitar a los interesados en las UWSN el estudio de dichas redes, de forma que se obtengan resultados bastante cercanos a la realidad, se salden ciertas dudas acerca de esta tecnología y por qué no mencionar, que crezca la comunidad inclinada hacia este campo.

Como simulador, SARAS ofrece la posibilidad de crear de manera individual o masiva cierta cantidad de nodos UWSN, representarlos gráficamente en un área determinada en un plano cartesiano (al momento bidimensional), y bajo un conjunto de parámetros definidos por el usuario obtener los cálculos de *Bit Error Rate*, Tasa de Error de Bits, *Signal to Noise Ratio*, Relación Señal a Ruido y *Successful Delivery Ratio*, Relación de Entrega Exitosa en

una ruta hallada por el software entre un nodo origen y un nodo destino. A mayores, se plantea la visión de llegar a simular arquitecturas UWSN tridimensionales, nodos con distintas naturalezas, ya sean ordinarios o sumideros, e incluso protocolos de enrutamiento más complejos.

A continuación, el presente manual explica el procedimiento para realizar la instalación y puesta a punto del software, seguido de los pasos para utilizar el simulador de manera apropiada y culminando con una guía para desinstalar la aplicación.

## D.2. Instalación de SARAS

### D.2.1. Requisitos de Hardware

Antes de realizar la instalación debe tomarse en cuenta los requisitos necesarios para la ejecución de SARAS. La tabla presentada a continuación detalla los requisitos de hardware mínimos y recomendados de la aplicación.

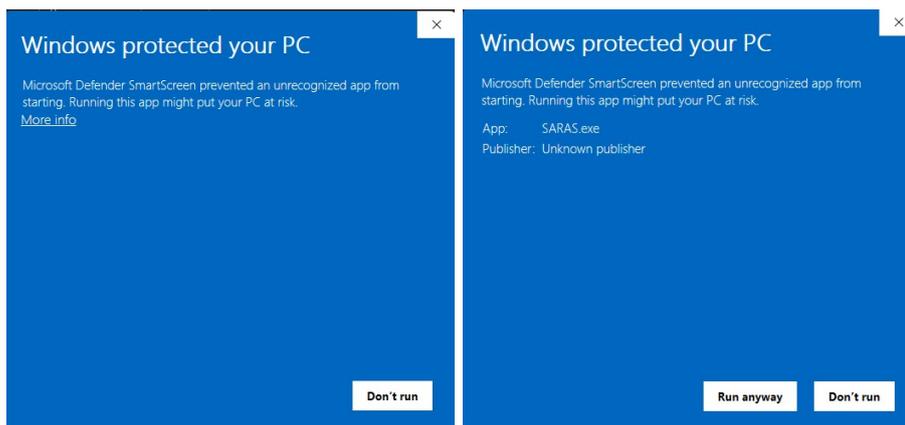
Hardware	Mínimo	Recomendado
Procesador	Cualquier Intel o AMD x86-64	Cualquier Intel o AMD x86-64 con cuatro núcleos y soporte de AVX2
Memoria RAM	4 [GB]	8 [GB]
Espacio de almacenamiento	2 [GB]	4 [GB] SSD Altamente Recomendado
Gráficos	No requiere una tarjeta gráfica específica, pero se recomienda una tarjeta gráfica acelerada por hardware compatible con OpenGL 3.3 con memoria GPU de 1 GB	

### D.2.2. Requisitos de Software

También es importante conocer el sistema operativo en el que se va a ejecutar. SARAS requiere estrictamente el uso de *Windows* en sus versiones 11 o 10 (versión 20H2 o superior), y es también compatible con *Windows Server 2019 o 2022*. Al momento de redacción del presente manual, SARAS NO cuenta con soporte para sistemas operativos Linux o MacOS.

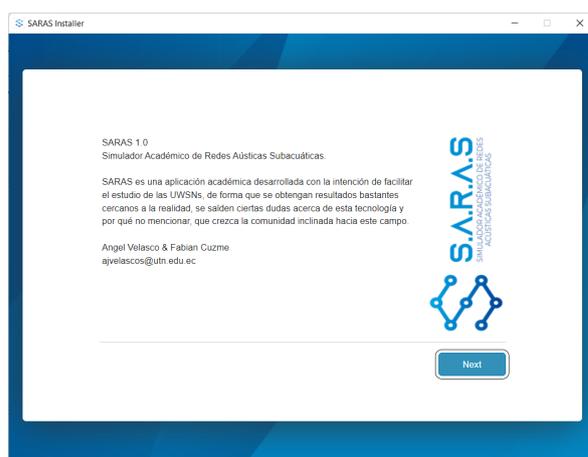
### D.2.3. Proceso de Instalación

**Paso 1** Para comenzar se debe ejecutar con doble clic el instalador `SARAS.exe`. Normalmente *Microsoft Defender* hace una advertencia de aplicación desconocida, a lo que se debe clicar Más información y continuar con el botón *Ejecutar de todas formas* como se muestra en las siguiente figura.



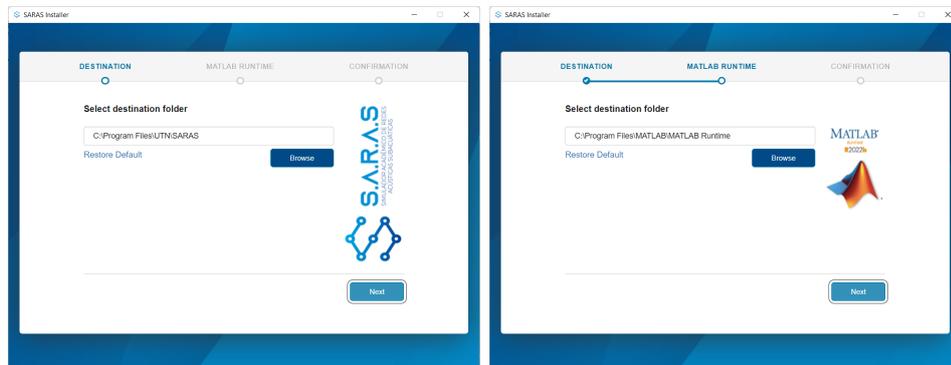
**Importante:** Para ejecutar el instalador es necesario otorgar permisos de administración.

**Paso 2** El instalador detalla información acerca de la aplicación, clicar *Siguiente*.

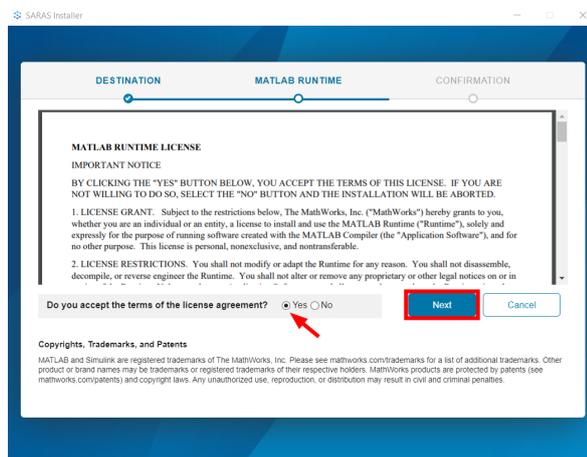


**Paso 3** Se elige la ubicación de instalación de SARAS y al clicar *Siguiente* se elige la ubicación de instalación de *Matlab Runtime*<sup>1</sup>, clicar *Siguiente*.

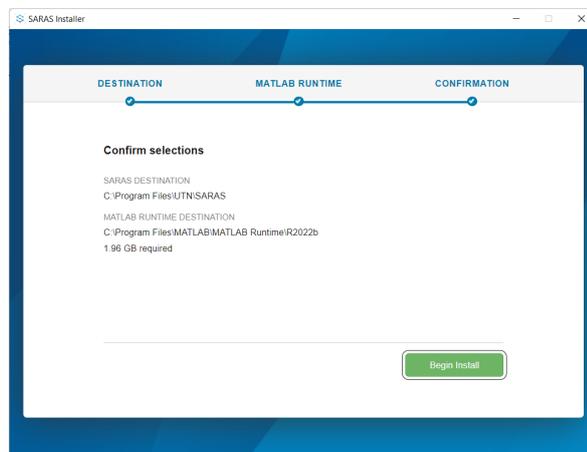
<sup>1</sup>Se recomienda que la ubicación de instalación sea la denominada por defecto.



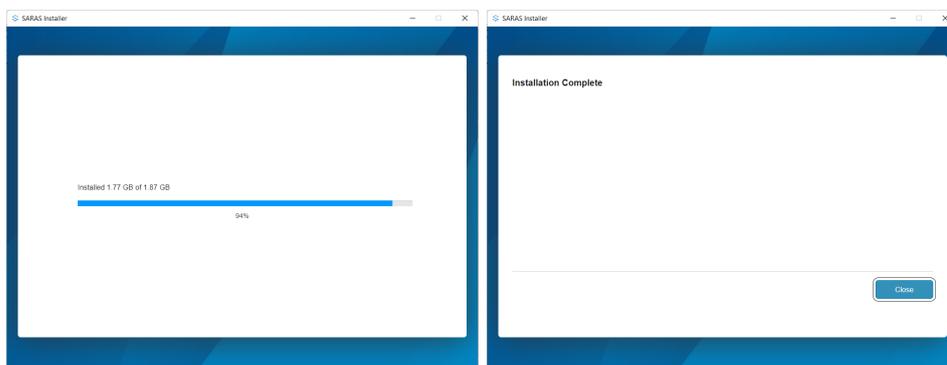
**Paso 4** Se lee y se acepta los términos de la licencia de Matlab Runtime seleccionando la opción Sí y clicando Siguiente.



**Paso 5** El instalador hace un resumen de lo especificado previamente y pide confirmación para comenzar. Clicar Comenzar Instalación.



**Paso 6** Por último se espera a que finalice la instalación, el tiempo de instalación depende estrictamente del hardware del ordenador. Cuando muestre el mensaje de Instalación Completada clicar Cerrar.



#### D.2.4. Previo a la simulación

SARAS cuenta con la opción de definir las unidades de medida que se quieran utilizar en el transcurso de la simulación. Esta opción se halla en el menú superior *Unidades*. Al clicar en el botón, se despliegan los tipos de unidades que se utilizan en SARAS, las cuales se especifican en la siguiente tabla.

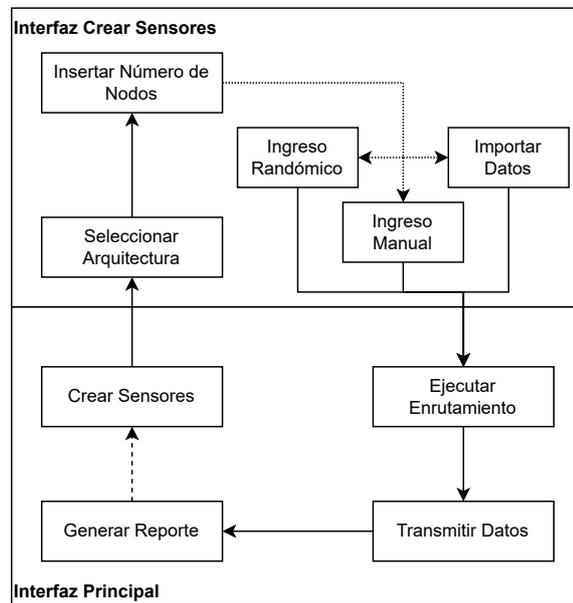
Tipo	Unidad	¿Por defecto?
Frecuencia	Herzio [ $Hz$ ]	No
	Kiloherzio [ $KHz$ ]	Sí
Potencia	Milivatio [ $mW$ ]	No
	Vatio [ $W$ ]	Sí
Información	Bit [ $b$ ]	No
	Kilobit [ $Kb$ ]	No
	Byte [ $B$ ]	Sí
	Kilobyte [ $KB$ ]	No

Al hacer clic en una unidad de medida esta se marcará con un visto .

**Importante:** Todas las unidades de medida con las que trabaja SARAS están expresadas [entre corchetes] con la abreviatura de su respectiva unidad; de esta manera se conoce en todo momento la unidad de medida que se encuentra habilitada.

### D.3. Diagrama general de SARAS

A pesar de la gran cantidad de funciones independientes con las que cuenta la aplicación, SARAS establece una secuencia de acciones concatenadas con el fin de obtener los resultados. Este proceso se detalla de manera general en la figura mostrada a continuación; donde se establecen las etapas primordiales por las que atraviesa una simulación en ambas interfaces del programa.



### D.4. Pasos para realizar una simulación en SARAS

Esta sección explica el procedimiento para realizar una simulación apropiadamente utilizando SARAS. Es importante conocer con anterioridad los valores de frecuencia, potencia y rango de comunicación con los que trabajan los sensores que se van a simular. Esta información puede ser hallada en el *datasheet*<sup>2</sup> del modelo que se va a usar<sup>3</sup>.

<sup>2</sup>La hoja de datos es un documento que resume el rendimiento y otras características de un producto.

<sup>3</sup>Un ejemplo de *datasheet* es el siguiente: Evo Logics Underwater Acoustic Modems.

### D.4.1. Crear o Importar sensores

Luego de iniciar la aplicación, es posible importar datos previamente exportados siguiendo los siguientes pasos:

1. Dirigirse al menú *Archivo/Importar Datos* o usar la combinación de teclas **Ctrl + I**.
2. Seleccionar el fichero con extensión **.csv** que contenga los datos.
3. Presionar el botón **Abrir**.
4. Dirigirse a la pestaña **Características de Sensores** que se encuentra en el panel de *Datos*.
5. Insertar el valor de Rango de Comunicación.
6. Presionar el botón .

Comúnmente, para crear por primera vez los nodos es necesario presionar el botón **Crear Sensores** para ejecutar la ventana *Crear Sensores*, continuando el proceso de 3 modos distintos.

#### Insertar nodos manualmente

1. Seleccionar Arquitectura 2D.
2. Ingresar el Número de nodos.
3. Ingresar el Rango de Comunicación.
4. Ingresar el área de dispersión en los campos *X* y *Y*.
5. Insertar los valores de *X* y *Y* de cada nodo.
6. Presionar el botón **Confirmar**.

#### Insertar nodos aleatoriamente

1. Seleccionar Arquitectura 2D.
2. Ingresar el Número de nodos.
3. Ingresar el Rango de Comunicación.

4. Ingresar el área de dispersión en los campos  $X$  y  $Y$ .
5. Presionar el botón **Generar Aleatoriamente**.
6. Presionar el botón **Confirmar**.

### Insertar nodos importados

1. Presionar el botón **Importar**.
2. Seleccionar el fichero con extensión `.csv` que contenga los datos.
3. Presionar el botón **Abrir**.
4. Presionar el botón **Confirmar**.

**Importante:** El sistema está programado para cumplir con la interacción del usuario, es decir, los datos son completamente modificables en cualquier momento siempre y cuando se encuentren dentro de los parámetros. Es posible combinar una inserción manual de datos con una aleatoria.

### D.4.2. Ejecutar protocolo de enrutamiento

Para esto es necesario conocer los extremos entre los cuales se realizará el enrutamiento. Los pasos son los siguientes:

1. Insertar el ID del nodo Origen.
2. Insertar el ID del nodo Destino.
3. Seleccionar protocolo de enrutamiento.
4. Presionar el botón **Enrutar**.

Para este punto ya se obtienen los valores de Distancia entre Origen y Destino, Tiempo de enrutamiento, Número de saltos y la ruta creada por el algoritmo.

### D.4.3. Transmitir datos

Los requisitos en esta etapa son los encontrados en la pestaña **Características de Sensores**. Como ayuda para usuarios que no tengan a mano las características con las que va a simular, SARAS cuenta con un menú

llamado *Randomizar Datos* que como puede esperarse, provee los valores necesarios de manera randómica<sup>4</sup>. Para ingresar los datos manualmente los pasos son los siguientes:

1. Insertar valor de Frecuencia.
2. Insertar valor de Potencia.
3. Insertar valor de Cantidad de Datos.
4. Presionar el botón .

Si el proceso ha sido cumplido bajo las condiciones pertinentes la simulación habrá terminado con éxito, obteniendo los valores de *Signal to Noise Ratio*, Relación Señal a Ruido, *Bit Error Rate*, Tasa de Error de Bits y *Successful Delivery Ratio*, Relación de Entrega Exitosa, así como también se habrá encendido el LED ubicado en la parte inferior derecha en color verde.

#### D.4.4. Generar reporte

Habiendo obtenido los últimos resultados, llega la hora de guardar un reporte para futuras referencias. Para esto únicamente requerimos de haber terminado la simulación asegurándonos de que el LED se encuentre de color verde. Los pasos son los siguientes:

1. Dirigirse al menú *Archivo/Guardar Resultados* o usar la combinación de teclas **Ctrl + S**.
2. Seleccionar la carpeta donde se planea guardar el archivo `.txt`.
3. Escribir el nombre del archivo (por defecto es `Resultados.txt`).
4. Presionar el botón .

#### D.4.5. Miscelánea

En esta parte se explican algunas de las funcionalidades de SARAS que no tienen cabida en el proceso de simulación y no han sido explicadas anteriormente.

---

<sup>4</sup>Este método es poco fiable para hacer cálculos certeros.

**Exportar Nodos** En la ventana de *Crear Sensores* es posible guardar los nodos que ya se han generado de la siguiente manera:

1. Presionar el botón **Exportar**.
2. Seleccionar la carpeta donde se planea guardar el archivo `.csv`.
3. Escribir el nombre del archivo (por defecto es `Sensores.csv`).
4. Presionar el botón **Guardar**.

**Semilla Aleatoria** En la ventana de *Crear Sensores* existe un campo llamado *Semilla*; este valor provoca que el generador de números aleatorios se convierta en uno pseudoaleatorio, es decir que si el botón **Generar Aleatoriamente** es presionado de nuevo con la misma semilla, producirá la misma secuencia de números, y por ende, los mismos nodos. Si el valor de *Semilla* se encuentra establecido en 0, el generador seguirá siendo aleatorio.

**Comunicación entre Interfaces** Como se puede apreciar en el programa, ciertos campos que se hallan en la ventana principal, también se los puede encontrar en la ventana de *Crear Sensores*. Cualquier valor modificado se toma en cuenta internamente sin importar la interfaz donde se haya cambiado. Esto da facilidad al usuario para editar los valores y realizar una nueva simulación.

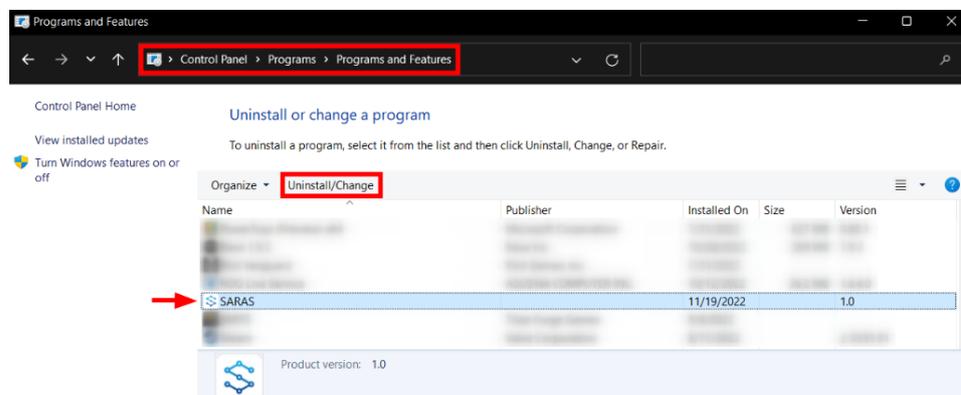
**Cuadros de Diálogo** La aplicación cuenta con un sinnúmero de mensajes de advertencia o de error que indican una mala práctica en la simulación. Procurar leer los textos para mitigar los errores.

**Información de herramientas** Otra ayuda que ofrece el programa al usuario es la información de herramientas, que son textos cortos que aparecen al mantener el cursor sobre ciertas zonas interactivas. Prestar atención sobre todo en los botones icónicos.

## D.5. Desinstalación de SARAS

Al igual que la mayoría de programas instalados en el equipo, SARAS puede ser desinstalado desde **Panel de Control/Programas/Programas** y

**Características.** A continuación, se selecciona SARAS y se clicla el botón Desinstalar/Cambiar como se muestra en la figura.



**Importante:** Para ejecutar la desinstalación es necesario otorgar permisos de administración.



## Apéndice E

# Guías de Laboratorio

*La práctica es un maestro excepcional.*

Plinio el Joven

**RESUMEN:** El presente apéndice es otro material de apoyo que aporta al aprendizaje del uso de SARAS, en este se manifiestan dos guías de laboratorio que actuarán a modo de práctica para aquellos dispuestos a conocer la aplicación SARAS y su funcionamiento.

### **E.1. Guía de Laboratorio 1: Instalación y Reconocimiento de SARAS**

Como primera práctica se establece la instalación del *Simulador Académico de Redes Acústicas Subacuáticas* así como la exploración de las funciones que se encuentran en el entorno.

Universidad Técnica del Norte  
Facultad de Ingeniería en Ciencias Aplicadas  
Carrera de Ingeniería Electrónica y Redes de Comunicación

**Elaborado por:** Angel Velasco Suárez

**Revisado por:** Ing. Fabián Cuzme

**Tema:**

Instalación y Reconocimiento de SARAS

### **Objetivos**

#### **Objetivo General**

Estudiar la forma en la que operan las Redes de Sensores Inalámbricos Subacuáticos (UWSN) haciendo uso del programa SARAS (Simulador Académico de Redes Acústicas Subacuáticas).

#### **Objetivos Específicos**

- Investigar acerca de las Redes de Sensores Inalámbricos Subacuáticos.
- Instalar el Simulador Académico de Redes Acústicas Subacuáticas (SARAS).
- Familiarizarse con el entorno de SARAS y probar las funciones que otorga.

### **Marco Teórico**

#### **Redes de Sensores Inalámbricos Subacuáticos**

Las UWSNs (Underwater Wireless Sensor Networks) nacen de la necesidad de explorar y monitorear el medio submarino en tiempo real, con el fin de brindar cierta agilidad en el desarrollo de innumerables aplicaciones, ya sea con fines científicos, comerciales o militares.

## Características de UWSNs

Dentro de las principales características de las UWSNs tenemos:

- Medio de transmisión: comunicación acústica.
- Velocidad de propagación: alrededor de  $1500[m/s]$ .
- Rango de transmisión: altas distancias debido a la baja frecuencia.
- Tasa de transmisión: aproximadamente  $10[Kbps]$ .
- Dificultad de recarga: energía de baterías limitada.
- Movimiento de nodos: topología dinámica gracias a corrientes submarinas.
- Calidad del enlace: baja e inestable.

## Arquitecturas de UWSNs

A modo general existen dos arquitecturas comunes de comunicación para UWSNs, que son bidimensionales y tridimensionales. Existe una tercer arquitectura también considerada tridimensional en la cual se trabaja junto a un vehículo autónomo subacuático, o en muchos casos, se trabaja con varios.

## Simulador Académico de Redes Acústicas Subacuáticas

SARAS es una aplicación académica desarrollada con la intención de facilitar a los interesados en las UWSN el estudio de dichas redes, de forma que se obtengan resultados bastantes cercanos a la realidad, se salden ciertas dudas acerca de esta tecnología y por qué no mencionar, que crezca la comunidad inclinada hacia este campo.

## Materiales y Equipos

Para la práctica se necesita de lo enumerado a continuación:

- Computador.
- Instalador de SARAS [↗](#).

### Diagrama del bloques



### Desarrollo

1. Realizar un marco teórico acerca de UWSN que contenga:
  - Definición.
  - Características comparadas con WSN terrestre.
  - Arquitecturas y sus componentes.
  - Aplicaciones.
2. Descargar e instalar SARAS.
3. Primera Simulación.
  - Crear 4 sensores.
  - Ingresar la posición de los nodos manualmente formando un cuadrado.
  - Los valores quedan a criterio del estudiante.
  - Cada nodo debe tener comunicación con los nodos, adyacentes mas no con el opuesto.
  - Generar las características de los sensores aleatoriamente.
  - Realizar enrutamiento entre dos nodos opuestos.
  - Realizar una transmisión de  $1000\text{bits}$ .
  - Guardar resultados obtenidos en un archivo llamado `Resultados1.txt`.
4. Segunda Simulación.

- Crear 25 sensores randómicos con un valor de semilla de acuerdo a su número de lista.
- El área será de  $500[m] \times 250[m]$ .
- El rango de comunicación será de  $150[m]$ .
- Exportar los datos de nodos en un archivo llamado `Sensores2.csv`.
- Los valores de frecuencia y potencia serán de  $10[KHz]$  y  $2[W]$  respectivamente.
- Realizar enrutamiento entre dos de los nodos más alejados.
- Realizar una transmisión de  $2[Kb]$ .
- Guardar resultados obtenidos en un archivo llamado `Resultados2.txt`.

## Resultados

1. Primera Simulación.
  - Tabla de datos de sensores.
  - Plot de dispersión de nodos.
  - Resultados1.txt
2. Segunda Simulación.
  - Tabla de datos de sensores.
  - Sensores2.csv
  - Plot de dispersión de nodos.
  - Resultados2.txt

## Entregables

1. Informe técnico
2. Resultados1.txt
3. Sensores2.csv
4. Resultados2.txt

## **E.2. Guía de Laboratorio 2: Cálculo y simulación de UWSNs usando SARAS**

La segunda práctica propone realizar una simulación utilizando SARAS, seguido de los cálculos respectivos para comprobar la precisión de la aplicación.

Universidad Técnica del Norte  
Facultad de Ingeniería en Ciencias Aplicadas  
Carrera de Ingeniería Electrónica y Redes de Comunicación

**Elaborado por:** Angel Velasco Suárez

**Revisado por:** Ing. Fabián Cuzme

**Tema:**

Cálculo y simulación de UWSNs usando SARAS

### Objetivos

#### Objetivo General

Estudiar la forma en la que operan las Redes de Sensores Inalámbricos Subacuáticos (UWSN) haciendo uso de cálculos matemáticos y el programa SARAS (Simulador Académico de Redes Acústicas Subacuáticas).

#### Objetivos Específicos

- Investigar acerca de comunicación acústica subacuática.
- Realizar la simulación de una UWSN utilizando SARAS.
- Calcular el BER, SNR y SDR de una ruta creada por SARAS.

### Marco Teórico

#### Distancia entre dos puntos en el plano

Teniendo los puntos  $A(x_1, y_1)$  y  $B(x_2, y_2)$ , se puede calcular la distancia entre ellos utilizando la Ecuación (E.1)

$$d_{(A,B)} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (\text{E.1})$$

### Relación Señal a Ruido

La SNR por bit  $\gamma$  de la señal acústica transmitida en el nodo receptor puede ser calculada como

$$\gamma = S_{level} - T_{loss} - N_{level} + D_{index} \quad (E.2)$$

donde  $S_{level}$ ,  $T_{loss}$ ,  $N_{level}$  y  $D_{index}$  son el *nivel de señal*, la *pérdida de transmisión*, el *nivel de ruido* y el *índice de directividad*, respectivamente. Cabe destacar que todos los valores están en *decibelios [dB]* y el valor  $D_{index}$  es de 0, ya que en entornos subacuáticos se suelen usar hidrófonos omnidireccionales (que no transmiten la señal acústica en una sola dirección, sino de manera esférica).

El nivel de señal  $S_{level}$  con respecto a la potencia de transmisión  $P$  y la distancia entre el nodo de origen y el de destino  $d$  se determina de esta forma

$$S_{level} = 10 [\log(P) - \log(4\pi d^2) - \log(0,67 \times 10^{-18})] \quad (E.3)$$

La pérdida de transmisión subacuática  $T_{loss}$  a una distancia  $d[m]$  para una señal de frecuencia  $f[KHz]$  puede derivarse por

$$T_{loss} = 20 \log d + \alpha(f) \times d \times 10^{-3} \quad (E.4)$$

donde  $\alpha(f)$  es el coeficiente de absorción (del sonido en el agua de mar) en  $[dB/Km]$ . Para calcular  $\alpha(f)$ , se puede usar la fórmula de Thorp tal que así

$$\alpha(f) = \frac{0,11f^2}{1 + f^2} + \frac{44f^2}{4100 + f^2} + 2,75 \times 10^{-4} f^2 + 0,003 \quad (E.5)$$

En el mundo real, es difícil calcular el valor exacto del nivel de ruido, de tal forma que se utiliza una aproximación práctica del ruido en ambientes subacuáticos. Esta aproximación se establece a continuación

$$N_{level} = 50 - 18 \log f \quad (E.6)$$

Tomando en cuenta la Ecuación (E.3), E.4, E.5 y E.6, la Relación Señal a Ruido toma la siguiente forma

$$\begin{aligned} \gamma = & 10 [\log(P) - \log(4\pi r^2) - \log(0,67 \times 10^{-18})] - 20 \log d \\ & - \left( \frac{0,11f^2}{1+f^2} + \frac{44f^2}{4100+f^2} + 2,75 \times 10^{-4}f^2 + 0,003 \right) \times d \times 10^{-3} \quad (\text{E.7}) \\ & - 50 + 18 \log f \end{aligned}$$

### Tasa de Bits Errados

El valor de BER se puede determinar de la siguiente manera

$$BER(\gamma) = \frac{1}{2} \left( 1 - \sqrt{\frac{10\gamma/10}{1+10\gamma/10}} \right) \quad (\text{E.8})$$

donde  $BER(\gamma)$  es el número estimado de bits errados recibidos de un flujo de datos sobre un canal de comunicación que han sido alterados debido a ruido, interferencias, distorsión o errores de sincronización de bit, por cada unidad de tiempo, siendo  $\gamma$  el SNR calculado en la Ecuación (E.7).

### Relación de Entrega Exitosa

Basándose en la tasa de error de un solo bit, expuesto en la Ecuación (E.8), se puede obtener la relación de entrega exitosa de un solo bit tal que así

$$\begin{aligned} P_{\text{exito}}^1(\gamma) &= 1 - BER(\gamma) \\ &= 1 - \frac{1}{2} \left( 1 - \sqrt{\frac{10\gamma/10}{1+10\gamma/10}} \right) \quad (\text{E.9}) \\ &= \frac{1}{2} + \frac{1}{2} \sqrt{\frac{10\gamma/10}{1+10\gamma/10}} \end{aligned}$$

donde  $P_{\text{exito}}^1(\gamma)$  se define como la probabilidad de que un único bit se entregue con éxito cuando el SNR es igual a  $\gamma$ . Bajo ese amparo, se define  $P_{\text{exito}}^m(\gamma)$  como la probabilidad de que un paquete de  $m$  bits sea entregado exitosamente. Así, utilizando el valor de  $\gamma$ , se puede hallar la relación de entrega exitosa de un paquete con tamaño de  $m$  bits, como se especifica en la Ecuación (E.10).

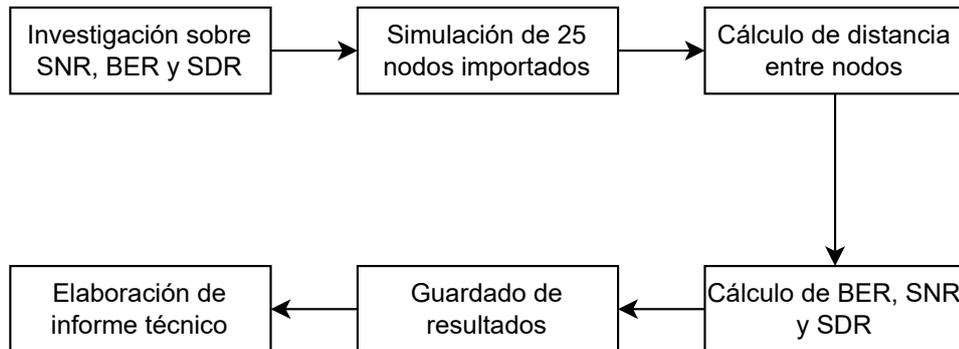
$$P_{\text{exito}}^m = \left( \frac{1}{2} + \frac{1}{2} \sqrt{\frac{10\gamma/10}{1 + 10\gamma/10}} \right)^m \quad (\text{E.10})$$

## Materiales y Equipos

Para la práctica se necesita de lo enumerado a continuación:

- Computador.
- SARAS instalado.

## Diagrama del bloques



## Desarrollo

1. Realizar un marco teórico acerca de UWSN que contenga:

- Algoritmo de Dijkstra
- Relación Señal a Ruido.
- Tasa de Bits Errados.
- Relación de Entrega Exitosa.

2. Simulación.

- Importar los datos de nodos exportados en Laboratorio 1.
- Realizar una búsqueda de un datasheet de cualquier módem acústico.
- Utilizar las características del datasheet para la simulación.

- Realizar enrutamiento entre dos de los nodos más alejados.
  - Realizar una transmisión de  $5[Kb]$
  - Guardar resultados obtenidos en un archivo llamado `Resultados1.txt`.
3. Con los mismos datos de la simulación realizar los siguientes cálculos matemáticos:
- De la ruta obtenida, calcular la distancia entre nodos en cada salto
  - Calcular el promedio de los valores obtenidos de SNR en cada salto.
  - Calcular el BER del SNR total.
  - Calcular el SDR.

## Resultados

1. Datasheet utilizado.
2. Simulación.
  - Tabla de datos de sensores.
  - `Sensores.csv`.
  - Plot de dispersión de nodos.
  - `Resultados.txt`.
3. Comparación de resultados de simulación y cálculos matemáticos.

## Entregables

1. Informe técnico
2. `Sensores.csv`
3. `Resultados.txt`



## Apéndice F

# Formulario de Validación de Requerimientos

**RESUMEN:** El presente apéndice, evidencia el formulario de encuesta para los estudiantes de la materia de Comunicación Inalámbrica en la Carrera de Ingeniería en Telecomunicaciones, acerca del desempeño y funcionamiento de SARAS en el taller, como soporte para el capítulo de pruebas del proyecto.

Universidad Técnica del Norte  
Facultad de Ingeniería en Ciencias Aplicadas  
Carrera de Ingeniería Electrónica y Redes de Comunicación

Encuesta vinculada al proyecto de titulación “*Diseño de una aplicación académica para el estudio de Redes de Sensores Inalámbricos Subacuáticos en MATLAB*”, por *Angel Jampier Velasco Suárez*.

**Objetivo de la encuesta:** El objetivo de la presente encuesta es conocer la opinión y feedback de los estudiantes de la materia de Comunicación Inalámbrica acerca del software de simulación SARAS (Simulador Académico de Redes Acústicas Subacuáticas), luego de haber recibido la capacitación práctica referente a su utilización.

**Indicaciones:** Lea detenidamente las preguntas y seleccione la opción u opciones que crea conveniente. Esta encuesta tomará aproximadamente 5 minutos en contestar.

1. **¿Que nivel de conocimiento considera tener acerca de las UWSNs?**

- Avanzado  
 Intermedio  
 Bajo

2. **¿Cuán interesado está usted en las Redes de Sensores Inalámbricos Subacuáticos?**

- Muy Interesado  
 Interesado  
 Poco Interesado  
 Desinteresado

3. **¿Cuál cree usted que es el desempeño de SARAS en las funciones enumeradas a continuación?**

	Regular	Bueno	Excelente
Crear y editar nodos UWSN.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Graficar nodos UWSN.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Randomizar funciones a gran escala.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aplicar protocolos de enrutamiento.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Simular arquitectura bidimensional.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Calcular atributos de SNR, BER y SDR.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Importar y exportar datos y resultados.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

4. **En una escala del 0 al 10 ¿Con cuánto calificaría a SARAS respecto al atributo de calidad “Adecuación Funcional”?**

El programa es capaz de suplir las necesidades declaradas, cumplir con las tareas propuestas y proveer resultados correctos con el nivel de precisión requerido.

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

Pésimo

Regular

5. **En una escala del 0 al 10 ¿Con cuánto calificaría a SARAS respecto al atributo de calidad “Usabilidad”?**

El programa es apto para ser entendido, aprendido, usado y resulta atractivo al usuario así como permite ser operado y controlado con facilidad.

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

Pésimo

Regular

6. **En una escala del 0 al 10 ¿Con cuánto calificaría a SARAS respecto al atributo de calidad “Mantenibilidad”?**

El programa es apto para ser modificado efectiva y eficientemente, de cara a la evolución o perfeccionamiento del mismo.

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

Pésimo

Regular

7. **En una escala del 0 al 10 ¿Con cuánto calificaría a SARAS respecto al atributo de calidad “Portabilidad”?**

El programa ofrece la facilidad para ser transferido de forma efectiva y eficiente de un entorno a otro con las limitantes que el mismo dispone.

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

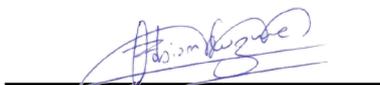
Pésimo

Regular

8. **¿Cómo calificaría a SARAS tomando en cuenta la experiencia generada en el taller?**

☆ ☆ ☆ ☆ ☆

9. **¿Tiene usted alguna sugerencia adicional para la aplicación?**



**Ing. Fabián Cuzme MSc.**

Director de Tesis



**Angel Velasco**

Tesista



# Referencias

- Anchaluisa, O. (2015). *Atlas Marino Costero del Ecuador* (inf. téc.). Instituto Oceanográfico y Antártico de la Armada del Ecuador.
- Bahrami, N., Khamis, N. H. H., Baharom, A. & Yahya, A. (2016). Underwater Channel Characterization to Design Wireless Sensor Network by Bellhop. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 14(1), 110. <https://doi.org/10.12928/telkomnika.v14i1.2454>
- Blandón Andrade, J. C. & López Sotélo, J. A. (2012). BIO-ROUTE: UN SIMULADOR PARA REDES DE SENSORES INALÁMBRICOS BIO-ROUTE: A SIMULATOR FOR WIRELESS SENSOR NETWORKS. *Educación en Ingeniería*, 7, 23-31. <http://www.educacioneningenieria.org>
- Cáceres, M., Moreno, Y., Tello, J. & Vargas, I. (2018). Cálculo de la distancia entre dos puntos. *Diseño, implementación y evaluación de unidades didácticas de matemáticas en MAD 3* (pp. 67-129).
- Chicaiza, D. F. (2009). *ESTUDIO DE LAS REDES DE SENSORES BAJO EL AGUA Y SUS PRINCIPALES APLICACIONES* (Tesis doctoral). ESCUELA POLITÉCNICA DEL EJÉRCITO.
- Cujilema, G. E. (2015). *PLATAFORMA DE EVALUACIÓN DE MECANISMOS DE LOCALIZACIÓN DE NODOS EN REDES DE SENSO-*

- RES INALÁMBRICOS BAJO EL AGUA (UWSN)* (Tesis doctoral).  
Escuela Superior Politécnica De Chimborazo.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269-271. <https://doi.org/10.1007/BF01386390>
- El-Banna, A. A., Wu, K. & Elhalawany, B. M. (2020). Opportunistic Cooperative Transmission for Underwater Communication Based on the Water's Key Physical Variables. *IEEE Sensors Journal*, 20(5), 2792-2802. <https://doi.org/10.1109/JSEN.2019.2953277>
- El-Rabaie, E.-S., Nabil, R. & Alsharqawy, M. (2015). Underwater Wireless Sensor Networks (UWSN), Architecture, Routing Protocols, Simulation and Modeling Tools, Localization, Security Issues and Some Novel Trends. *CiiT International Journal of Networking and Communication Engineering*, vol. 7, no. 8, , pp. 335-354, August 2015., 7, 335-354.
- Espinosa, J. (2022). Distancia entre dos puntos. [http://campusvirtual.cua.uam.mx/material/tallerm/10\\_Distancia\\_entre\\_dos\\_puntos\\_html/index.html#](http://campusvirtual.cua.uam.mx/material/tallerm/10_Distancia_entre_dos_puntos_html/index.html#)
- Garcia, L., Jimenez, J. M., Parra, L., Lloret, J. & Lorenz, P. (2020). An overview on IoUT and the performance of WiFi low-cost nodes for IoUT Applications. *2020 IEEE Global Communications Conference, GLOBECOM 2020 - Proceedings, 2020-Janua*, 8-13. <https://doi.org/10.1109/GLOBECOM42002.2020.9348057>
- Goldsmith, A. (2005). *Wireless Communications*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511841224>
- Gvirtz, S. & Necuzzi, C. (2011). *Educación y tecnologías: Las voces de los expertos* (V. Tatti, Ed.; 1st).

- Han, G., Jiang, J., Sun, N. & Shu, L. (2015). Secure Communication for Underwater Acoustic Sensor Networks. *IEEE Communications Magazine*, 53(8), 54-60. <https://doi.org/10.1109/MCOM.2015.7180508>
- Haque, K. F., Kabir, K. H. & Abdelgawad, A. (2020). Advancement of routing protocols and applications of Underwater Wireless Sensor Network (UWSN)-A survey. *Journal of Sensor and Actuator Networks*, 9(2), 1-31. <https://doi.org/10.3390/jsan9020019>
- Hattab, G., El-Tarhuni, M., Al-Ali, M., Joudeh, T. & Qaddoumi, N. (2013). An underwater wireless sensor network with realistic radio frequency path loss model. *International Journal of Distributed Sensor Networks*, 2013. <https://doi.org/10.1155/2013/508708>
- Ibrahim, D. M. & Hussein, D. (2014). *Modelling and Performance Enhancement of Underwater Wireless Sensor Networks by Petri Nets* (Tesis doctoral). Tanta University. <https://www.researchgate.net/publication/280609377>
- IEEE. (1998). IEEE Recommended Practice for Software Requirements Specifications. *IEEE Std 830-1998*, 1-40. <https://doi.org/10.1109/IEEESTD.1998.88286>
- INOCAR. (2021). Misión y visión del Instituto Oceanográfico y Antártico de la Armada. <https://www.inocar.mil.ec/web/index.php/institucion/mision-y-vision>
- ISO. (2011). Systems and software Quality Requirements and Evaluation (SQuaRE) (ISO/IEC 25010). <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>
- Kao, C. C., Lin, Y. S., Wu, G. D. & Huang, C. J. (2017). A comprehensive study on the internet of underwater things: Applications, challenges,

- and channel models. *Sensors (Switzerland)*, 17(7). <https://doi.org/10.3390/s17071477>
- Liou, E. C., Kao, C. C., Chang, C. H., Lin, Y. S. & Huang, C. J. (2018). Internet of underwater things: Challenges and routing protocols. *Proceedings of 4th IEEE International Conference on Applied System Innovation 2018, ICASI 2018*, 1171-1174. <https://doi.org/10.1109/ICASI.2018.8394494>
- MathWorks. (2004). *MATLAB® Release Notes* (inf. téc.). The MathWorks, Inc. 1 Apple Hill Drive Natick, MA 01760-2098.
- MathWorks. (2022a). MATLAB. <https://la.mathworks.com/products/matlab.html>
- MathWorks. (2022b). MATLAB App Designer. [https://la.mathworks.com/products/matlab/app-designer.html?s\\_tid=srchtitle\\_app%20designer\\_1](https://la.mathworks.com/products/matlab/app-designer.html?s_tid=srchtitle_app%20designer_1)
- MathWorks. (2022c). MATLAB para estudiantes. [https://la.mathworks.com/products/matlab/student.html?s\\_tid=hp\\_ff\\_p\\_student](https://la.mathworks.com/products/matlab/student.html?s_tid=hp_ff_p_student)
- MathWorks. (2022d). MATLAB R2022b System Requirements for Windows. <https://la.mathworks.com/support/requirements/matlab-system-requirements.html>
- MathWorks. (2022e). MATLAB Runtime. [https://la.mathworks.com/help/compiler/matlab-runtime.html?searchHighlight=runtime&s\\_tid=srchtitle\\_runtime\\_3](https://la.mathworks.com/help/compiler/matlab-runtime.html?searchHighlight=runtime&s_tid=srchtitle_runtime_3)
- MathWorks. (2022f). What Is MATLAB? <https://la.mathworks.com/discovery/what-is-matlab.html>
- Menna, B. V., Acosta, G. G., Member, S. & Villar, S. A. (2016). Modelo de canal acústico submarino para aguas poco profundas.

- Mohamed, N., Jawhar, I., Al-Jaroodi, J. & Zhang, L. (2011). Sensor network architectures for monitoring underwater pipelines. *Sensors*, 11(11), 10738-10764. <https://doi.org/10.3390/s111110738>
- Molisch, A. F. (2012). *Wireless communications*. John Wiley & Sons.
- Pérez, J. & Gardey, A. (2013). Definición de Ingeniería de Software. Consultado el 10 de junio de 2022, desde <https://definicion.de/ingenieria-de-software/>
- Prokopets, E. (2020). The SDLC Process Explained: Key Phases and Methodologies. <https://www.edvantis.com/blog/software-development-process/>
- Recuero, A. (1994). Aplicaciones de la teoría de grafos: búsqueda de caminos en una red y análisis de su conectividad. *Informes de la Construcción*, 46(433), 33-45. <https://doi.org/10.3989/ic.1994.v46.i433.1115>
- Ronit, P. (2018). *Survey on Network Simulators* (inf. téc.). <https://www.researchgate.net/publication/333262623>
- Sánchez Matías, A. M. (2013). *Sistema de comunicación acústica para redes de sensores inalámbricas subacuáticas en aguas someras* (Tesis doctoral). Universitat Politècnica de València. <https://doi.org/10.4995/Thesis/10251/35326>
- Silva, C. (2022). matlab wsn code. <https://github.com/cesarfgs/matlab-wsn-code>
- Sotnik, D., Goetz, M. & Nissen, I. (2020). *Cognitive Underwater Acoustic Networking Techniques*.
- Stallings, W. (2004). *COMUNICACIONES Y REDES DE COMPUTADORES* (D. Fayerman & A. I. García, Eds.; Séptima). PEARSON EDUCACIÓN, S.A. <http://www.williamstallings.com/DCC/DCC7e.html>

- Tan, H.-p., Seah, W. K. G., Diamant, R. & Waldmeyer, M. (2011). *A Survey of Techniques and Challenges in Underwater Localization* (Tesis doctoral). Singapore Management University.
- Torrubia, G. S. & Lozano Terrazas, V. M. (2001). Algoritmo de Dijkstra. Un Tutorial Interactivo. *VII Jornadas de Enseanza Universitaria de la Informática (JENUI 2001)*, 254-258. <http://www.dma.fi.upm.es/dijkstra/>
- Yu, L. (2018). Matlab programming environment based on web. *Proceedings of 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference, ITOEC 2018*, (Itoec), 509-512. <https://doi.org/10.1109/ITOEC.2018.8740716>

# Glosario

## *Vehículo Autónomo Subacuático*

Es un robot que viaja bajo el agua sin requerir la intervención de un operador humano (Ver AUV).

## *fading*

También llamado desvanecimiento, es la variación en la atenuación que experimenta una señal y depende de múltiples variables. Un canal con fading es un canal de comunicación que experimenta desvanecimiento.

## *gateway*

Es un dispositivo que recibe toda la información obtenida por el sistema completo y enruta estos paquetes hacia un servidor u ordenador.

## *hidrófono*

Es un transductor de sonido a electricidad para ser usado en agua o en otro líquido.

## *multipath*

Es un fenómeno de propagación que resulta en dos o más trayectorias de una señal que llega a un receptor al mismo tiempo o dentro de nanosegundos entre sí.

## *nodo ordinario*

Se conoce como nodo ordinario a un sensor inalámbrico cuya función es recoger información de cierta naturaleza en su entorno.

## *nodo sink*

También llamado controlador o sumidero, es un nodo encargado de recolectar los datos obtenidos de los nodos ordinarios y transferirlos

hacia otros nodos sink o directamente al *gateway*.

***open source***

Se conoce como código abierto a un código fuente que se encuentra disponible de forma gratuita para posible modificación o redistribución.

***pseudoaleatorio***

Número generado en un proceso que parece producir números al azar, pero las mismas condiciones iniciales producen siempre el mismo resultado..

***semilla***

Número utilizado para inicializar un generador de números pseudoaleatorios. Si el generador es inicializado de nuevo con la misma semilla, producirá la misma secuencia de números.

# Acrónimos

## **MATLAB**

*MATrix LABoratory*, Laboratorio de Matrices.

## **AF**

*Amplify-and-Forward*, Amplifica y Reenvía.

## **AUV**

*Autonomous Underwater Vehicle*, Vehículo Autónomo Subacuático.

## **BER**

*Bit Error Rate*, Tasa de Error de Bits.

## **CU**

Caso de Uso.

## **DF**

*Decode-and-Forward*, Decodifica y Reenvía.

## **GUI**

*Graphical User Interface*, Interfaz Gráfica de Usuario.

## **IoUT**

*Internet of Underwater Things*, Internet de las Cosas Subacuáticas.

## **KPV**

*Key Physical Variable*, Variable Física Clave.

**OOP**

*Object-Oriented Programming*, Programación Orientada a Objetos.

**PAN**

*Personal Area Networks*, Redes de Área Personal.

**PLR**

*Packet Loss Rate*, Tasa de Pérdida de Paquetes.

**RF**

*Radiofrecuencia*.

**RNG**

*Random Number Generation*, Generación de Números Aleatorios.

**SARAS**

*Simulador Académico de Redes Acústicas Subacuáticas*.

**SDLC**

*Software Development Life Cycle*, Ciclo de Vida de Desarrollo de Software.

**SDR**

*Successful Delivery Ratio*, Relación de Entrega Exitosa.

**SNR**

*Signal to Noise Ratio*, Relación Señal a Ruido.

**TWSN**

*Terrestrial Wireless Sensor Network*, Red de Sensores Inalámbricos Terrestres.

**UWSN**

*Underwater Wireless Sensor Network*, Red de Sensores Inalámbricos Subacuáticos.

**Wi-Fi**

*Wireless Fidelity*, Fidelidad Inalámbrica.

**WLAN**

*Wireless Local Area Networks*, Redes de Área Local Inalámbricas.

**WSN**

*Wireless Sensor Network*, Red de Sensores Inalámbricos.

*Me llena de orgullo bajar el telón con este objetivo cumplido,  
en adelante espero estar a la altura del porvenir.*

*Angel Velasco Suárez*