

SEP

TNM

INSTITUTO TECNOLÓGICO DE CULIACÁN



Aplicación web para el monitoreo remoto de los parámetros de
calidad del agua en estanques acuícolas usando tecnologías de
Aprendizaje maquina e Internet de las cosas

TESIS

PRESENTADA ANTE EL DEPARTAMENTO ACADÉMICO DE ESTUDIOS DE POSGRADO
DEL INSTITUTO TECNOLÓGICO DE CULIACÁN EN CUMPLIMIENTO PARCIAL DE LOS
REQUISITOS PARA OBTENER EL GRADO DE

MAESTRA EN CIENCIAS DE LA COMPUTACIÓN

POR:

Jaqueline Parra Valenzuela
INGENIERA AMBIENTAL

DIRECTOR DE TESIS:

Dr. Ricardo Rafael Quintero Meza

CULIACÁN, SINALOA

marzo de 2023

Instituto Tecnológico de Cullacán

Cullacán, Tlax., **03 de Noviembre del 2022**

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
OFICIO: OEPI-2022/10222

ASUNTO: **Autorización Impresión**

JAQUELINE PARRA VALENZUELA
ESTUDIANTE DE LA MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN
PRESENTE.

Por medio de la presente y en virtud de que ha completado los requisitos para el examen de grado de **Maestro en Ciencias de la Computación**, se concede autorización para la impresión de la tesis titulada: **"APLICACIÓN WEB PARA EL MONITOREO REMOTO DE LOS PARÁMETROS DE CALIDAD DEL AGUA EN ESTANQUES ACUÍCOLAS USANDO TECNOLOGÍAS DE MACHINE LEARNING E IOT"** bajo la dirección de la **M.C. RICARDO RAFAEL QUINTERO MEZA**.

Sin otro particular reciba un cordial saludo.

ATENTAMENTE
Culiacán, Tlax., 03 de Noviembre del 2022



M.C. MARÍA ARACELY MARTÍNEZ AMAYA
JEFE(A) DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN

 **EDUCACIÓN** 
INSTITUTO TECNOLÓGICO DE CULLACÁN
DEPARTAMENTO DE CALIDAD DE
SERVICIO DE POSGRADO E INVESTIGACIÓN

CC: OEPI

MARACELY*



Av. de Obis 8821s 210 Pto. Cal, Cuexleque C.P. 90320 Culiacán, Sinaloa. Tel. 807-454-0700
y 807-713-3884 | <http://www.cullacan.tecnm.mx>



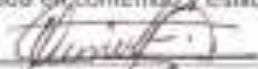
2022 **10 años**
de **la** **UNAM**

"APLICACIÓN WEB PARA EL MONITOREO REMOTO DE LOS
PARÁMETROS DE CALIDAD DEL AGUA EN ESTANQUES
ACUÍCOLAS USANDO TECNOLOGÍAS DE MACHINE LEARNING
E IOT"

Tesis presentada por el(a):

JAQUELINE PARRA VALENZUELA

Aprobada en contenido y estilo por:



M.C. RICARDO RASMEL QUINTERO MEZA
Director de Tesis



DR. HÉCTOR RODRÍGUEZ RANGEL
Secretario



M.C. GLORIA ENATERNE PERALTA PERUNURI
Vocal -1



ROSALIO ZATARAIN CABADA
Vocal -2



M.C. María Aracely Martínez Amaya
Jefe(a) de la División de Estudios de
Posgrado e Investigación



Dedicatoria

Dedico este trabajo a mis padres, por ser un ejemplo para seguir, por apoyarme y confiar en mí, por darme ánimos y fortaleza para cumplir cualquier meta que me proponga.

¡Infinitas gracias!

Agradecimientos

Agradezco a mis padres **Alma Irasema** y **Rafael** por la educación que me han brindado y por su apoyo incondicional no solo durante este proceso sino de toda la vida.

A mi hermana **Quetzally** y a su esposo **Jorge** por su apoyo, consejos y por ser un ejemplo que seguir para mí.

A mis hermanos **Alma Livier** y **Miguel Ángel** por motivarme y animarme en los momentos difíciles.

A mis **compañeros** y **amigos** por su apoyo y motivación.

A mi tía **Alicia** y a su esposo **Jesús**, por brindarme información y haberme abierto las puertas en el laboratorio acuícola para que pudiera realizar pruebas.

Al **Dr. Ricardo Rafael Quintero Meza** por brindarme su apoyo, asesoría y conocimiento a lo largo de este proyecto.

Al **Dr. Iván Esteban Villalón Turrubiates** por haberme aceptado en la estancia de investigación en el ITESO, por su asesoría y por su apoyo.

Al **Tecnológico Nacional de México Campus Culiacán** por aceptarme en el programa y bríndame un espacio, herramientas y profesores para que pudiera aprender y mejorar profesionalmente.

Al **Consejo Nacional de Ciencia y Tecnología (CONACYT)** por haberme brindado el apoyo financiero para que pudiera dedicarme a mis estudios de posgrado.

Declaración de autenticidad

Por la presente declaro que, salvo cuando se haga referencia específica al trabajo de otras personas, el contenido de esta tesis es original y no se ha presentado total o parcialmente para su consideración para cualquier otro título o grado en esta o cualquier otra Universidad. Esta tesis es resultado de mi propio trabajo y no incluye nada que sea resultado de algún trabajo realizado en colaboración, salvo que se indique específicamente en el texto.

Jaqueline Parra Valenzuela. Culiacán, Sinaloa, México, 2023

Resumen

En la industria acuícola es fundamental mantener un monitoreo continuo de los parámetros de calidad del agua en los estanques; ya que los organismos acuáticos deben sobrevivir en un entorno simulado, de lo contrario el rendimiento y la productividad de la industria se verían afectados. Para ello, este trabajo propone una solución basada en ingeniería de software para el monitoreo utilizando tecnologías de Internet de las cosas y pronóstico utilizando tecnologías de Aprendizaje máquina de los parámetros de calidad del agua en los estanques acuícolas, utilizando servicios externos para recibir la información de los sensores.

En el sector acuícola, al menos hablando de la cría de camarón existen diferentes etapas en las que se pueden aplicar estos sistemas de monitoreo: En laboratorios de postlarva (donde se lleva a cabo la siembra de camarón), en el transporte hacia una granja acuícola y en las granjas acuícolas (una vez que el camarón alcanza cierta etapa se envían a una granja acuícola para que continúen su desarrollo hasta alcanzar la etapa comercial).

El sistema se implementó para la etapa del transporte, pero su arquitectura fue diseñada para que se pudiera adaptar al resto de las etapas fácilmente. Este sistema se basa en una aplicación móvil que recibe por Bluetooth los datos que son enviados de los sensores de oxígeno disuelto y temperatura, una vez recibidos los procesa, es decir les agrega la fecha y la hora a la fueron recibidos y los envía a una aplicación web para que los usuarios los puedan visualizar en cualquier dispositivo. La aplicación web, envía alertas por correo electrónico a los usuarios, realiza un pronóstico a múltiples pasos al futuro del estado del oxígeno disuelto y temperatura y muestra los datos mediante tableros de control y gráficos.

Desarrollar un sistema Internet de las cosas es de gran ayuda para recolectar datos y así mismo aplicar otros sistemas que permitan estudiar esos datos para resolver distintas problemáticas, en el caso de los laboratorios y granjas acuícolas, este sistema puede brindar información anticipada a los usuarios sobre el futuro estado de los parámetros de calidad del agua, para que se puedan prevenir riesgos en la producción. Sin embargo, como trabajo a futuro es necesario que se tome un tiempo mínimo de un año para recolectar datos por medio de la aplicación y que los modelos de entrenamiento mejoren y puedan brindar información más acertada.

Palabras clave

- Acuicultura
- Aprendizaje máquina
- Aprendizaje profundo
- Bluetooth low energy
- Calidad del agua
- Desarrollo web
- Inteligencia artificial
- Internet de las cosas
- Ingeniería de software
- Monitoreo remoto

Índice general

1. Introducción.....	1
1.1. Planteamiento del problema.....	2
1.2. Objetivos	3
1.2.1. Objetivo general	3
1.2.2. Objetivos específicos	3
1.3. Hipótesis.....	3
1.4. Justificación	4
1.5. Estructura de la tesis.....	5
2. Marco teórico.....	6
2.1. Acuicultura	6
2.1.1. Calidad del agua en la acuicultura	7
2.2. Internet de las cosas	8
2.2.1. Medios de comunicación en Internet de las cosas.....	8
2.3. Diseño y desarrollo web	9
2.4. Arquitectura de software.....	10
2.5. Inteligencia artificial	12
2.5.1. Aprendizaje maquina.....	13
2.5.2. Aprendizaje profundo.....	14
2.5.2. Aprendizaje supervisado	15
2.5.3. Series de tiempo.....	15
2.5.4. Pronóstico de series de tiempo	16
2.3.5. Modelos de Aprendizaje profundo para el pronóstico.....	16
2.5. Configuración y evaluación de modelos de Aprendizaje maquina	18
3. Estado del arte	22
3.1. Sistemas de monitoreo	22
3.2. Sistemas de pronóstico	26
4. Desarrollo del proyecto.....	34
4.1. Planeación e investigación	35
4.2. Análisis	35
4.2.1. Diagrama de contexto	35
4.2.2. Requisitos funcionales.....	36
4.2.3. Requisitos de calidad.....	37

4.2.4.	Casos de uso	38
4.2.5.	Mapeo de los requisitos funcionales y casos de uso	40
4.3.	Diseño	40
4.3.1.	Modelo de dominio.....	41
4.3.2.	Arquitectura lógica	41
4.3.3.	Modelo arquitectónico cliente-servidor	45
4.3.4.	Arquitectura física.....	45
4.3.5.	Diagrama de secuencia.....	46
4.4.	Codificación	47
4.4.1.	Aplicación móvil	47
4.4.2.	Aplicación web.....	49
5.	Pruebas.....	58
5.1.	Sistema de monitoreo	58
5.2.	Modelo de pronóstico	60
6.	Conclusiones y trabajo futuro	72
	Bibliografía	74
	A. Interfaces graficas del sistema.....	77

Índice de figuras

Figura 2-4 Relación entre Aprendizaje profundo e IA (Josh Patterson, 2017).	13
Figura 2-5. Arquitectura típica de una red neuronal (Negnevitsky, 2005).....	14
Figura 2-6. Red neuronal biológica (Negnevitsky, 2005).	14
Figura 2-4. Arquitectura de las Redes Neuronales Recurrentes (Francesca Lazzeri, 2021).	17
Figura 2-4. Función lineal (Negnevitsky, 2005)..	19
Figura 2-4. Función sigmoidea (Negnevitsky, 2005).....	19
Figura 2-4. Función tanh (Josh Patterson, 2017).....	19
Figura 3-2. Sistema propuesto para el monitoreo de autobuses escolares (Zambada Juan, 2015)....	23
Figura 3-3. Visualización del monitoreo de los autobuses (Zambada Juan, 2015).	24
Figura 3-1. Sistema para el monitoreo y control de la calidad del agua en la acuicultura (Abinaya T, 2019).	25
Figura 3-4. Arquitectura del sistema de distribuido de Internet de las cosas (Encinas Cesar, 2017).	26
Figura 3-5. Configuración de la red neuronal LSTM (Rodriguez Hector, 2020).....	27
Figura 3-6. Proceso de optimización de hiperparámetros (Rodriguez Hector, 2020).	27
Figura 3-7. Arquitectura de CNN para del nivel del agua (Baek Sang-Soo, 2020).	28
Figura 3-8. Arquitectura LSTM para los parámetros de calidad del agua (Baek Sang-Soo, 2020)..	29
Figura 3-9. RMSE de radiación solar de los diferentes modelos (Aslam Muhammad, 2019).....	30
Figura 3-10. Comparación de modelos en periodos mensuales (Aslam Muhammad, 2019).....	30
Figura 4-1. Metodología.....	34
Figura 4-2. Diagrama de contexto.....	36
Figura 4-3. Diagrama de casos de uso.....	40
Figura 4-4. Mapeo de casos de uso con requisitos funcionales.....	40
Figura 4-5. Diagrama de modelo de dominio.....	41
Figura 4-6. Arquitectura lógica	44
Figura 4-7 Modelo arquitectónico cliente – servidor.	45
Figura 4-8. Arquitectura física	46
Figura 4-9. Diagrama de secuencia.	47
Figura 4-8 Visualización de los datos en la aplicación móvil.	48
Figura 4-9 Archivo de texto que almacena los datos en caso de un fallo	49
Figura 4-10 Modelo lógico de base de datos.....	50
Figura 4-11 Visualización de los datos en la aplicación web.....	51

Figura 4-11 Notificación.....	51
Figura 4-12 Vectores de entrada y salida.....	56
Figura 5-1 Laboratorio de postlarva Oleajes.....	59
Figura 5-2 Sensores.....	60
Figura 5-3 Resultados oxígeno disuelto LSTM Arquitectura 1.....	62
Figura 5-4 Resultados temperatura LSTM Arquitectura 1.....	62
Figura 5-5 Resultados oxígeno disuelto GRU Arquitectura 1.....	63
Figura 5-6 Resultados oxígeno disuelto GRU Arquitectura 1.....	63
Figura 5-7 Precisión Arquitectura 1, modelo LSTM.....	64
Figura 5-8 Precisión Arquitectura 1, modelo GRU.....	64
Figura 5-9 Pérdida Arquitectura 1, modelo LSTM.....	65
Figura 5-10 pérdida Arquitectura 1, modelo GRU.....	65
Figura 5-11 Resultados oxígeno disuelto LSTM arquitectura 2.....	67
Figura 5-12 Resultados oxígeno disuelto LSTM arquitectura 2.....	67
Figura 5-13 Resultados temperatura LSTM arquitectura 2.....	68
Figura 5-14 Resultados temperatura GRU arquitectura 2.....	68
Figura 5-15 Precisión Arquitectura 2, modelo LSTM.....	69
Figura 5-16 Precisión Arquitectura 2, modelo GRU.....	69
Figura 5-17 Precisión Arquitectura 2, modelo LSTM.....	70
Figura 5-18 Precisión Arquitectura 2, modelo GRU.....	70

Índice de tablas

Tabla 3-1. Comparación de las características principales de los proyectos del estado del arte.....	32
Tabla 4-1. Requisitos funcionales.....	37
Tabla 4-2. Requisitos de calidad.....	38
Tabla 4-3. Casos de uso.....	39
Tabla 4-4. Resultados LSTM Arquitectura 1.....	61
Tabla 4-5. Resultados GRU Arquitectura 1.....	61
Tabla 4-6. Resultados LSTM Arquitectura 2.....	66
Tabla 4-7. Resultados GRU Arquitectura 2.....	66

Capítulo 1

1. Introducción

La Organización de las Naciones Unidas para la Alimentación y la Agricultura, mejor conocida como FAO (por sus siglas en inglés), define a la acuicultura como una serie de actividades para el cultivo y la cría de especies marinas como peces, moluscos, crustáceos y plantas acuáticas (FAO, 2017). La economía de las personas que se dedican a actividades acuícolas depende de la calidad del agua de las granjas y laboratorios acuícolas, ya que con malas condiciones en la calidad del agua, los organismos acuáticos pueden morir afectando el rendimiento y productividad. Debido a esto surge la necesidad de mantener un monitoreo continuo de los parámetros de calidad del agua como: temperatura, pH, oxígeno disuelto entre otros más.

En la actualidad se han implementado sistemas para poder monitorear en tiempo real las variables fisicoquímicas más importantes del agua y con esto tener una respuesta más rápida con respecto a qué acciones tomar cuando surgen condiciones anormales en la calidad del agua de los estanques (Encinas Cesar, 2017). También se han empleado modelos para la predicción del oxígeno disuelto enfocado a la acuicultura mediante Aprendizaje máquina (Baek Sang-Soo, 2020) . Sin embargo, aún no existen tecnologías que se apliquen a las diferentes escalas del sector acuícola

En el sector acuícola existen diferentes funciones que se le pueden dar a este tipo de sistemas, tales como el monitoreo en los estanques de postlarva de laboratorios acuícolas, al momento de transportar el producto de un laboratorio a una granja acuícola y en granjas acuícolas. Una de las principales problemáticas que se tiene en esta industria es que durante el transporte de un laboratorio a una granja acuícola se vuelve más difícil mantener el monitoreo de los parámetros de calidad del agua, lo cual puede implicar pérdidas económicas para las organizaciones que se dedican a este sector.

Este proyecto pretende desarrollar un sistema de monitoreo remoto, así como el pronóstico del oxígeno disuelto y la temperatura en estanques acuícolas. La implementación de este sistema se enfocará en el transporte de productos acuícolas pero la aplicación está diseñada

para que también pueda ser utilizada en laboratorios y granjas acuícolas. En este documento de tesis se presentará la investigación, el desarrollo de la aplicación web, las pruebas, los resultados y conclusiones que se obtuvieron.

1.1. Planteamiento del problema

Dado que se está creando un entorno controlado para la supervivencia de los organismos acuáticos, se requiere realizar un monitoreo continuo de los parámetros de calidad del agua (como el oxígeno disuelto, temperatura, turbidez, etc.) con la finalidad de mantener a los biólogos informados sobre el comportamiento de los estanques y que se puedan tomar medidas, en caso de que la calidad del agua de los estanques se vea afectada.

Las técnicas que son utilizadas para el monitoreo de los parámetros de calidad del agua en la industria acuícola no son tan eficientes ya que consume mucho tiempo y normalmente estos parámetros son monitoreados cuando ya se detectó alguna anomalía. En la actualidad, se han implementado sistemas de monitoreo y pronóstico de los parámetros de calidad del agua en los estanques acuícolas. Estos sistemas contribuyen a que las organizaciones reduzcan el riesgo de pérdida de producto. Sin embargo, en el sector acuícola, existen diferentes etapas en las que se pueden utilizar estos sistemas. Si hablamos al menos de la producción de camarones, el sistema puede implementarse en diferentes etapas:

- Para los laboratorios de postlarvas, que es donde se siembra el camarón y se mantienen hasta que alcanzan una determinada fase.
- Para el transporte del producto, una vez que alcanzan una determinada fase, deben enviarse a una granja de acuícola para que continúen su desarrollo.
- En granjas de acuícolas, que es donde se mantiene el producto hasta que alcanza el desarrollo comercial.

Para iniciar con la implementación del sistema, se realizó una visita a un laboratorio de acuicultura llamado “Oleajes”, ubicado en Sinaloa, México, encargado de la cría de camarones. Se realizaron entrevistas con los biólogos responsables y en base a su experiencia se decidió tomar el oxígeno disuelto y la temperatura como los principales parámetros de calidad del agua; La etapa más crítica y peligrosa es el transporte del producto del camarón,

esto se debe a que los conductores a menudo no hacen el monitoreo correctamente debido a la insuficiencia de tiempo para detenerse y comprobar con sus equipos portátiles. Esta actividad humana representa un riesgo importante que puede mitigarse mediante un enfoque automatizado que elimine la intervención manual, aquí es donde el enfoque presentado en este proyecto podría ofrecer una solución eficaz al problema, ya que mediante el monitoreo continuo y el pronóstico del estado de los parámetros de calidad del agua mediante Aprendizaje máquina e Internet de las cosas, es posible obtener la información más reciente e incluso poder visualizar con anticipación si pudiera existir una anomalía en el futuro.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar un sistema inteligente para el monitoreo, pronóstico de los principales parámetros de calidad del agua (temperatura y oxígeno disuelto) durante el transporte de laboratorios a granjas acuícolas mediante información obtenida a través de Aprendizaje máquina e Internet de las cosas.

1.2.2. Objetivos específicos

- Ofrecer información actualizada del estado a través de tableros de control disponibles en computadoras y dispositivos móviles.
- Ofrecer notificaciones sobre cuando se presentó o se podría presentar una anomalía en los parámetros de calidad del agua.
- Desarrollar un modelo predictivo mediante Aprendizaje máquina para obtener información sobre el futuro comportamiento de los parámetros de calidad del agua.

1.3. Hipótesis

El monitoreo y la predicción de los parámetros de calidad del agua de forma remota mediante Internet de las cosas y Aprendizaje máquina permitirá detectar oportunamente las malas condiciones en el agua, generará alertas y posibilitará la actuación oportuna de acciones para asegurar la calidad de la producción de los estanques acuícolas.

1.4. Justificación

En el sector acuícola es de suma importancia mantener un monitoreo continuo de los parámetros de calidad del agua de los estanques acuícolas, ya que, con malas condiciones en la calidad del agua, la producción de alimentos de las organizaciones acuícolas puede verse afectada. En la actualidad, aún no existen tecnologías que se apliquen a las diferentes escalas del sector acuícola (laboratorios, transporte, granjas), así como tampoco existen sistemas que integren un pronóstico de los parámetros de calidad del agua junto con el monitoreo continuo. La investigación propuesta busca contar con un mecanismo inteligente que facilite la obtención de información y el pronóstico los parámetros de calidad del agua de forma continua a los usuarios.

Para cumplir los objetivos del proyecto se realizó una visita a un laboratorio acuícola para obtener información sobre sus necesidades y se realizaron investigaciones previas sobre el estado del arte. Con base en esos análisis se diseñó y desarrolló el sistema. De tal manera que la arquitectura de la aplicación web esté desacoplada del resto y nos pueda servir para poder implementarlo en las diferentes etapas de la industria acuícola (estanques, transporte y granjas acuícolas).

Se espera que el proyecto sirva como apoyo a las organizaciones del sector acuícola para poder tomar decisiones de manera oportuna y el aseguramiento de la calidad del agua de los estanques acuícolas.

1.5. Estructura de la tesis

La presente tesis está compuesta por 6 capítulos y por se organiza de la siguiente manera:

Capítulo 1.- Introducción: En este capítulo se presenta la introducción al proyecto, el objetivo general, los objetivos específicos, el planteamiento de problema y la justificación.

Capítulo 2.- Marco teórico: En este capítulo se presenta el marco teórico, que consta de todos los conceptos teóricos que conforman el proyecto, como una introducción a la acuicultura, la calidad del agua y las tecnologías utilizadas como Internet de las cosas y Aprendizaje máquina.

Capítulo 3.- Estado del arte: En este capítulo se presenta el estado del arte, que consta de los trabajos relacionados que sirvieron de referencia para realizar el presente trabajo de investigación.

Capítulo 4.- Desarrollo del proyecto: En este capítulo se presenta la información relacionada con la metodología que se siguió y las tecnologías que se utilizaron para el desarrollo del proyecto.

Capítulo 5.- Pruebas: En este capítulo se presentan los experimentos que se llevaron a cabo para comprobar que el sistema funciona de forma adecuada y si será posible cumplir con la hipótesis planteada.

Capítulo 6.- Conclusiones y trabajo a futuro: En este capítulo se presenta las conclusiones del trabajo, es decir si se cumplió con los resultados esperados, en que otras áreas es posible aplicarlo y de qué manera es posible mejorarlo.

Capítulo 2

2. Marco teórico

En este capítulo se presentan los fundamentos teóricos que se siguieron para el desarrollo del proyecto. El cual se divide en la sección 2.1 que contiene los conceptos básicos sobre acuicultura, calidad del agua y los parámetros de calidad del agua más importantes en la industria acuícola. En la sección 2.2 se presentan los conceptos y tecnologías referentes al Internet de las Cosas. En la sección 2.3 los conceptos y tecnologías que engloban el desarrollo web. En la sección 2.4 se muestra una breve introducción a la arquitectura de software. Y finalmente en la sección 2.5 todos aquellos conceptos, tecnologías y modelos basados en inteligencia artificial para realizar pronósticos.

2.1. Acuicultura

La acuicultura se conforma por un conjunto de actividades de intervención humana para la producción y crianza de especies marinas. Tales como la reproducción el cultivo y comercialización en un ambiente controlado (Encinas Cesar, 2017). Según (Trejo-Igueravide, 2016) en México existen 3 tipos de sistemas de acuícolas:

- a) Los extensivos, que son los sistemas de cultivo que se encuentran directamente en el medio en este caso en aguas naturales continentales o áreas semicerradas, como lagunas
- b) Los semi-intensivos que son sistemas que se encuentran en instalaciones cerradas y bajo condiciones y factores productivos controlados.
- c) Y los intensivos que son sistemas que se encuentran en condiciones similares a los intensivos, pero utilizan altas densidades de siembra y recambio de agua.

Según (FAO, 2017) la acuicultura es muy importante económicamente, ya que no solo sirve para abastecer de alimentos a la población, sino que además genera fuentes de empleo y riqueza para quienes se dedican a esta actividad.

2.1.1. Calidad del agua en la acuicultura

La calidad del agua es una medición del estado del agua en relación con una o más especies bióticas o con necesidades o propósitos humanos. Existen tres tipos de parámetros de calidad del agua que permiten medir el estado del agua: los físicos (como la temperatura, color, turbidez, conductividad eléctrica etc.), los químicos (como pH, oxígeno disuelto, alcalinidad, etc.) y los biológicos (como bacterias, algas y virus) (Summers, 2020).

En el caso de los laboratorios acuícolas, su objetivo es simular un ecosistema acuático, por lo cual, el agua debe estar en condiciones adecuadas para que sobrevivan organismos vivos. Una forma de asegurar la calidad del agua en los estanques, es mediante el monitoreo en tiempo real de los parámetros fisicoquímicos y biológicos como la temperatura, el oxígeno disuelto, el pH, clorofila, entre otros. Lo cual sirve para poder controlar los sistemas acuáticos, predecir futuras condiciones negativas y poder tomar las medidas adecuadas para evitar el colapso en la producción acuícola (N.C. Ferreira, 2011).

Según la experiencia de algunos biólogos de camarón, los parámetros de calidad del agua que más influyen al momento de controlar los estanques son el oxígeno disuelto y la temperatura.

- **Oxígeno disuelto (OD)**

Es un indicador del estado de la calidad del agua, es necesario mantener un control de este en los estanques acuícolas debido a que cuando baja la concentración por debajo de 5 ppm los camarones presentan bajas tasas de alimentación y crecimiento, sin embargo, la fluctuación diaria de oxígeno disuelto en un estanque aparentemente tiene poco efecto sobre la alimentación y el crecimiento para algunas especies, siempre y cuando la concentración no caiga por debajo de 2 ppm en la mañana. La concentración mínima de oxígeno para mantener una tasa aceptable de sobrevivencia varía con el tiempo de exposición y debe mantenerse en un rango entre 4 y 5 ppm. (Carbajal Hernández & Sánchez Fernández, 2013)

- **Temperatura**

La temperatura es otro de los indicadores que son de suma importancia controlarse en los estanques acuícolas, ya que afecta los procesos químicos y biológicos en el cultivo en los estanques, donde un aumento de la temperatura de 10 °C conduce al doble o al triple de

la velocidad de las reacciones químicas y biológicas. La temperatura también controla la solubilidad de los gases en el agua y la toxicidad del amonio. El rango ideal para la temperatura se considera de entre 25 a 32 °. (Carbajal Hernández & Sánchez Fernández, 2013)

2.2. Internet de las cosas

El Internet de las cosas, es un paradigma emergente que permite la comunicación entre una gran cantidad de dispositivos y sensores que se comunican entre si a través de internet. El Internet de las cosas tiene una visión multidisciplinaria, ya que puede ser aplicado para varios dominios como en la automatización en industria, la agricultura, la medicina, el transporte, entre otras aplicaciones más (Sachin Kumar, 2019).

El Internet de las cosas tiene como propósito conectar objetos físicos comunes a través de una red de comunicación a dispositivos informáticos con la finalidad de brindar algún servicio inteligente. El Internet de las cosas se compone de seis elementos: cosas u objetos, contexto, canal de comunicación, servicio, relación y tiempo; de esta forma busca conectar con un lugar, comunicarse con una red, recopilar datos, servir un objeto, procesar información e integrar objetos (Pérez, Mendoza, & Suarez, 2019).

2.2.1. Medios de comunicación en Internet de las cosas

Los medios de comunicación son un canal o enlace físico entre dos nodos de una red, a través del cual se transmite la información. Los principales medios de comunicación son:

- a) Inalámbricos son un espacio libre por donde se transmite la información a través de ondas electromagnéticas de radiofrecuencia que son portadoras de señales de datos.
- b) Los medios de comunicación alámbricos se definen como un cable y posiblemente otros dispositivos electrónicos que conectan físicamente los adaptadores de comunicación entre sí.
- c) La fibra óptica transmite haces de luz a altas velocidades y tráfico muy altos.

Como se menciona anteriormente el Internet de las cosas necesita de un canal de comunicación para poder conectar a los objetos comunes con los dispositivos informáticos.

A continuación, se mencionan los más usados en el Internet de las cosas:

- **Ethernet**

Ethernet fue creada en 1973 por un equipo del Centro de Investigación de *Palo Alto de Xerox Corporation* (Xerox PARC) en California. Se trata de un medio de comunicación alámbrico, utilizado en redes de área local (LAN) (Britannica, Britannica, 2022).

- **Bluetooth clásico:** *Bluetooth Classic* es una radio de baja potencia para la transmisión de datos. *Bluetooth Classic*, se utiliza principalmente para la transmisión inalámbrica de audio y se ha convertido en el protocolo de radio estándar detrás de los altavoces, auriculares y sistemas de entretenimiento inalámbricos para automóviles. Admite aplicaciones de transferencia de datos, incluida la impresión móvil (ABI, 2022).
- **Bluetooth Low Energy:** Es un medio de comunicación por radio que está diseñado para un funcionamiento de muy baja potencia. Se conocía originalmente por su capacidad para comunicarse con dispositivos, ahora se usa ampliamente como una tecnología de ubicación de dispositivos para satisfacer la creciente demanda de servicios de ubicación en interiores de alta precisión. *Bluetooth Low Energy (BLE)* ahora incluye funciones que permiten que un dispositivo determine la presencia, la distancia y la dirección de otro dispositivo (ABI, 2022).
- **ZigBee:** Zigbee es un estándar de red de sensores inalámbricos ampliamente utilizado con las características de bajo consumo, baja y baja tasa de datos y características de corto retardo de tiempo. Es muy fácil de desarrollar e implementar y tiene una gran seguridad y confiabilidad de datos (Muthu Ramya.C, 2011).
- **Wifi:** Es una tecnología de red que utiliza ondas de radio para transmitir datos de alta velocidad en distancias cortas. Se usa comúnmente para proporcionar acceso inalámbrico a Internet de banda ancha a una variedad de dispositivos (Britannica, 2022).

2.3. Diseño y desarrollo web

El diseño y desarrollo web son disciplinas diferentes que comúnmente suelen confundirse. El diseño web comprende las etapas del proyecto que se ocupan de la navegación, la usabilidad, la interacción, la arquitectura de la información y de la parte gráfica de la web. Mientras que el desarrollo web es la programación necesaria para la implementación del sitio web. El desarrollo web se divide en dos áreas:

- a) El lado del cliente, también conocido como *front-end*. Es el código básico para la creación de páginas web que interactúa con el usuario, es decir donde se crea la representación gráfica del sistema, para generar esta parte del código se usan herramientas como: HTML, CSS, JavaScript y DOM.
- b) El lado del servidor, también conocido como *back-end*. Es la parte del código que no tiene interacción directa con el usuario. Su objetivo es el diseño de bases de datos y asegurar la seguridad de la web.

Cuando estas dos partes se comunican, se dice que se está trabajando con un modelo cliente y servidor. Esta conexión permite al usuario interactuar con el contenido almacenado en la base de datos, registrarse a nuevos contenidos y generar cuentas de usuario (Soler, 2014).

2.4. Arquitectura de software

A medida que aumenta la complejidad de un sistema, surge la necesidad de diseñar y especificar la estructura global del sistema, la organización bruta y la estructura de control global; protocolos de comunicación, sincronización y acceso a los datos; asignación de funcionalidad a los elementos de diseño; la distribución física; la composición de los elementos de diseño, el escalado y el rendimiento, y la selección entre las alternativas de diseño (Garlan & Shaw, 1993)

La arquitectura de software es la estructura organizacional de un sistema o componente. El diseño de la arquitectura de software se basa en definir una colección de componentes de hardware y software y sus interfaces para establecer el marco para el desarrollo de un sistema informático (Engineers, 1990). Esto con la finalidad de seguir un orden y un plan para su desarrollo basado en el cumplimiento de sus objetivos o requerimientos.

Algunos de los modelos arquitectónicos más utilizados por los desarrolladores son:

- **Pipas y filtros:** Es el modelo arquitectónico más básico y se basa en componentes llamados “filtros” que leen un conjunto de entradas y generan una salida, para que esto se lleve a cabo de forma adecuada, se suele aplicar una transformación local a los datos de entrada para que se realice el cálculo computacional de la información de forma incremental, por lo que la salida se obtiene antes de que se consuma la

entrada. Los conectores, llamados “pipas” son los encargados de transmitir la información de un filtro a otro (Garlan & Shaw, 1993).

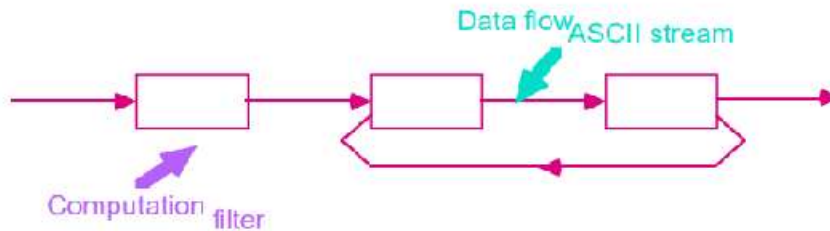


Figura 2-1. Modelo arquitectónico pipas y filtros (Garlan & Shaw, 1993).

- Modelo arquitectónico basado en capas.** Los modelos basados en capas se caracterizan por estar organizados jerárquicamente, de tal forma que cada capa provee un servicio a la capa superior y así mismo sirve como cliente de la capa inferior. Los conectores de estos sistemas generalmente son los protocolos utilizados para transmitir la información a través de los diferentes paquetes, módulos o capas por las que este compuesto.

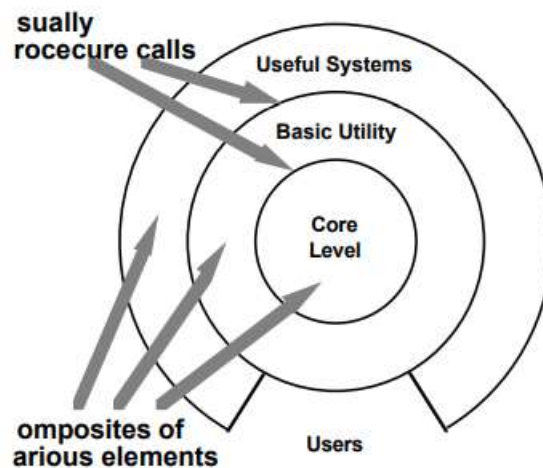


Figura 2-2. Modelo arquitectónico basado en capas (Garlan & Shaw, 1993).

- Modelo arquitectónico cliente-servidor.** Los modelos arquitectónicos cliente-servidor son uno de los más utilizados hoy en día, su principal característica es que

mantienen la comunicación entre procesos mediante el intercambio de información entre un cliente y un servidor. En donde el cliente es quien envía peticiones, mientras que el servidor es quien responde a las solicitudes enviadas.

Algunos de los protocolos que los modelos cliente-servidor utilizan para comunicarse son: File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP) y Hypertext Transfer Protocol (HTTP) (Oluwatosin, 2014).

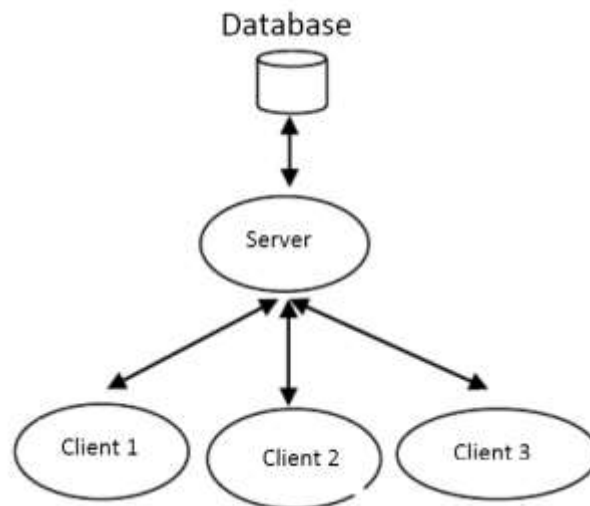


Figura 2-3. Proceso de comunicación entre el cliente y el servidor (Oluwatosin, 2014).

2.5. Inteligencia artificial

Para poder entender que es la inteligencia artificial, primero hay que definir el término “Inteligencia”. La inteligencia se define como capacidad que tiene alguien para entender y aprender cosas y así poder desarrollar la habilidad de resolver problemas y tomar decisiones (Negnevitsky, 2005).

La inteligencia artificial se define como el desarrollo de programas y mecanismos computacionales que pueden mostrar comportamientos considerados inteligentes. El objetivo de la inteligencia artificial es hacer que las máquinas o computadoras puedan realizar tareas o funciones que se consideren inteligentes, como si estas fueran realizadas por humanos (Negnevitsky, 2005).

Dentro de la Inteligencia Artificial existen dos campos que son muy utilizados hoy en día para el desarrollo de sistemas inteligentes, el Aprendizaje máquina y el Aprendizaje profundo. En la Figura 1, se muestra la relación que existe entre el Aprendizaje máquina y el

Aprendizaje profundo con la Inteligencia Artificial. Siendo *el* Aprendizaje profundo un subconjunto del Aprendizaje máquina que es un subcampo de la Inteligencia Artificial (IA) (Josh Patterson, 2017) .

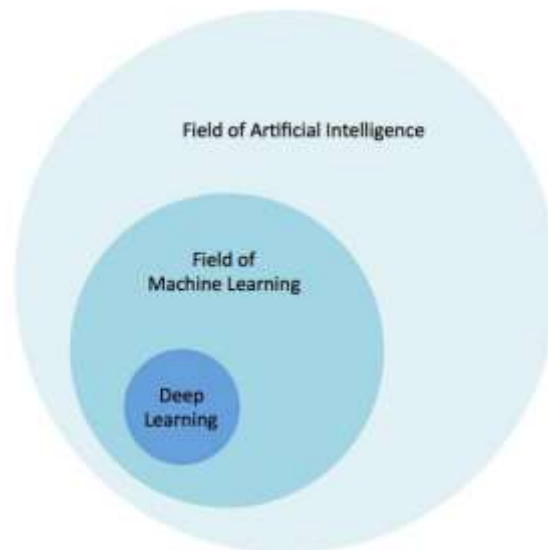


Figura 2-4 Relación entre Aprendizaje profundo e IA (Josh Patterson, 2017).

2.5.1. Aprendizaje máquina

el Aprendizaje máquina se define como un mecanismo adaptativo que permite a las computadoras aprender a través de la experiencia, aprender con el ejemplo y por analogía (Negnevitsky, 2005).

En el lenguaje natural, nos referimos a “aprender” como adquirir conocimiento sobre algún tema mediante el estudio, la experiencia o la enseñanza, las computadoras pueden aplicar este mismo concepto a través de algoritmos que extraen las descripciones estructurales de un conjunto de datos. En otras palabras, las computadoras aprenden mediante el análisis de los datos para después procesar esa información y generar modelos para poder predecir datos desconocidos y así cumplir con diferentes tareas.

Entre los modelos de Aprendizaje máquina más populares se encuentran: los árboles de decisión, la regresión lineal y las redes neuronales (Josh Patterson, 2017).

2.5.2. Aprendizaje profundo

Las redes neuronales son un modelo que trata de simular el comportamiento del cerebro humano, cuentan con dos características que simulan algunas de las funciones del cerebro humano (Negnevitsky, 2005) las cuales se mencionan a continuación:

- a) Al igual que las neuronas biológicas, las neuronas artificiales transmiten información que reciben a otras neuronas a menudo con algunas transformaciones.
- b) Las redes neuronales se pueden entrenar para que solo transmitan información útil para cumplir con sus objetivos, al igual que las neuronas de nuestro cerebro, que pueden ser entradas para transmitir señales para lograr objetivos más amplios del cerebro.

En la Figura 2-5 se muestra la arquitectura básica de una red neuronal en comparación con la Figura 2-3 que muestra la estructura de las redes neuronales biológicas.

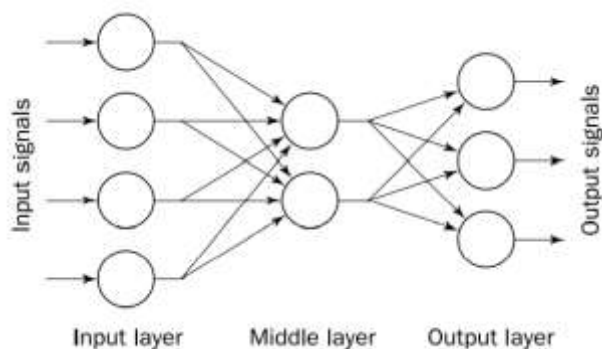


Figura 2-5. Arquitectura típica de una red neuronal (Negnevitsky, 2005).

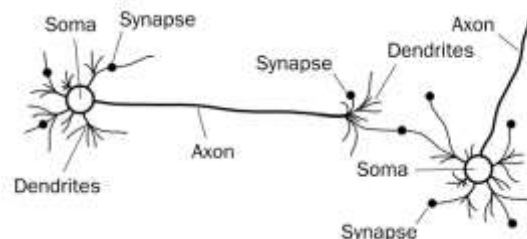


Figura 2-6. Red neuronal biológica (Negnevitsky, 2005).

Una red neuronal aprende mediante sus neuronas que están conectadas por enlaces, cada enlace tiene asociado un peso numérico. Los pesos son el medio básico de la memoria a largo

plazo de las redes neuronales. Las redes neuronales aprenden mediante ajustes repetitivos a estos pesos (Negnevitsky, 2005).

Las redes neuronales de propagación hacia adelante o *Feed-forward* producen una salida y, mediante una entrada x . En las redes neuronales multicapa esta información es enviada las neuronas de las siguientes capas hacia adelante.

Para entrenar las redes neuronales de aprendizaje supervisado es necesario aplicar otro algoritmo llamado propagación hacia atrás o *Backpropagation*. Este algoritmo consiste en comparar la salida de la red neuronal con el resultado esperado, de no ser así, será necesario regresar a ajustar los pesos para volver a generar una nueva salida y así continuar hasta obtener el resultado esperado (Josh Patterson, 2017).

2.5.2. Aprendizaje supervisado

El aprendizaje supervisado es aquel que requiere de un conjunto de datos de ejemplo para aprender a identificar patrones, estos ejemplos contienen el patrón de entrada y la salida esperada, mediante algoritmos como el de *Backpropagation*, donde red es capaz de determinar si la salida actual es igual o muy similar a la salida esperada, de lo contrario se medicaran los pesos (Negnevitsky, 2005).

2.5.3. Series de tiempo

Una serie de tiempo es un tipo de dato que muestra cómo puede ir cambiando una variable con respecto al tiempo. En una serie de tiempo, el tiempo puede ser utilizado como una estructura para ordenar los datos, pero no representa una variable en sí. Las series de tiempo son muy utilizadas por los científicos de datos para generar modelos para futuras estrategias y operaciones de planificación en varias industrias como: de finanzas, manufactura y el cuidado de la salud. El pronóstico de series temporales siempre ha desempeñado un papel importante a la hora de revelar información empresarial sobre el tiempo (Francesca Lazzeri, 2021). Existen dos tipos de series de tiempo:

- a) **Univariabes:** aquellas en donde solo se observa una sola variable en el tiempo.
- b) **Multivariabes:** donde se tiene más de una variable observada con respecto al tiempo que servirán para realizar la predicción de una más variables al futuro.

2.5.4. Pronóstico de series de tiempo

El pronóstico de series de tiempo es un mecanismo para la predicción de eventos mediante el análisis del comportamiento y el desempeño de los fenómenos pasados en una secuencia de tiempo y asumiendo que los sucesos futuros se mantendrán similares a las tendencias y comportamientos del pasado (Francesca Lazzeri, 2021). Existen dos tipos de pronósticos:

- a) **Pronóstico a un solo paso ($t + 1$):** cuando se realiza el pronóstico únicamente para el siguiente tiempo.
- b) **Pronóstico a múltiples pasos ($t + n$):** cuando se requiere realizar el pronóstico a más de dos pasos al futuro.

2.3.5. Modelos de aprendizaje profundo para el pronóstico

2.3.5.1. Recurrent Neural Networks (RNN)

Son redes neuronales que cuentan con un estado oculto (una memoria interna) que permite que información pasada persista en el tiempo. La memoria se mantiene dentro de un vector en la neurona y se combina con la información de entrada permitiendo generar una predicción y actualizar su propia memoria como se muestra en la figura 2-4.

El estado oculto, denotado como H , puede ser retroalimentado a la red. La RNN procesa los valores de entrada X y produce los valores de salida Y . El estado oculto denotado como H , permite reenviar la información de un nodo de la red al siguiente nodo, organizado como secuencias o un vector, dicho vector tendrá información de la entrada actual y las entradas anteriores. Luego pasará por una función de activación tangente hiperbólica (\tanh) para regular los valores entre -1 y 1 .

W indica que cada unidad tiene tres conjuntos de pesos, uno para las entradas (X) otro para las salidas del paso de tiempo anterior (H), y el otro para la salida del paso de tiempo actual (Y) (Francesca Lazzeri, 2021).

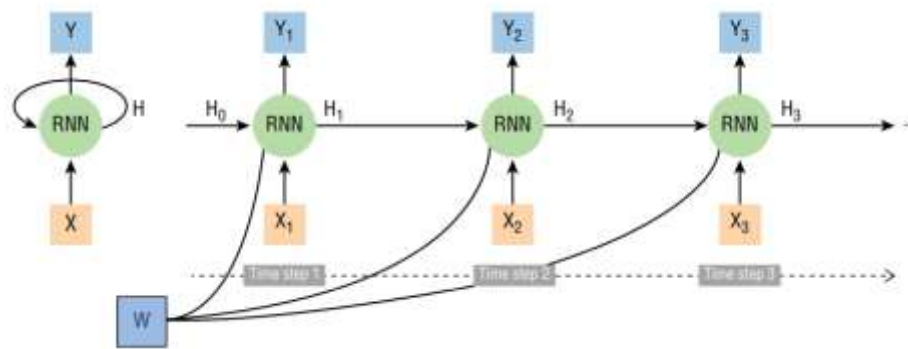


Figura 2-4. Arquitectura de las Redes Neuronales Recurrentes (Francesca Lazzeri, 2021).

Sin embargo, las RNN tienen algunas limitaciones, dado que solo tienen memoria a corto plazo, esto quiere decir que, si se tiene un vector muy grande de datos, los datos más antiguos no se tomarán en cuenta para realizar las siguientes predicciones. A este problema se le conoce como *vanishing gradient* (Francesca Lazzeri, 2021), debido a esto se han diseñado algunas variaciones de RNN que pueden resolver este problema como las LSTM y las GRU, las cuales se definen a continuación.

2.3.5.2. Redes neuronales Long-Short Term Memory (LSTM)

Son capaces de almacenar memoria a corto y largo plazo, permitiendo mantener información más antigua, lo cual posibilita eliminar el problema de *vanishing gradient*. Las neuronas de una red neuronal LSTM tienen funciones internas llamadas puertas, que son las que se encargan de controlar la información que se almacena o se desecha:

- **Forget gate:** Es la puerta que controla que partes de la memoria a largo plazo deben ser eliminadas.
- **Input gate:** Es la puerta que partes de los datos deben ser añadidos a la memoria de largo plazo
- **Output gate:** Controla que partes de la memoria de largo plazo deben ser considerados en la salida (Géron, 2017).

2.3.5.3. Redes neuronales Gated Recurrent Unit (GRU).

La neurona de una GRU es una versión simplificada de una LSTM (Francesca Lazzeri, 2021).

Las puertas de una neurona GRU son:

- **Update gate:** Determina la cantidad de información de los pasos a futuro anteriores que debe ser transmitida al futuro.

- **Resete gate:** Decide cuanta información pasada se debe eliminar.
- **Current memory gate:** Esta puerta se integra en la puerta de reinicio y se emplea para iniciar alguna no linealidad en la entrada y desarrollar media cero de la entrada. Otra de las motivaciones para incluirla en la puerta de reinicio es la de reducir el impacto que la información anterior tiene sobre la información actual que se pasa al futuro.

2.5. Configuración y evaluación de modelos de Aprendizaje máquina

En esta sección se presentan algunos conceptos de Aprendizaje máquina que usualmente son útiles para la configuración y evaluación de dichos modelos.

- **Hiperparámetros:** son los parámetros que se aplican para la configuración del modelo de Aprendizaje máquina. Estos datos pueden ser elegidos y modificados por los programadores o científicos de datos con la finalidad de mejorar los resultados de los modelos. Algunos ejemplos serían la fuerza de regularización de una penalización L2 en la función de pérdida de la regresión logística, un valor para establecer la profundidad máxima de un clasificador de árbol de decisión o el número de capas ocultas y unidades (neuronas) de una red neuronal.
- **Función de activación:** Es la función que se encarga de generar la salida de una neurona mediante sus datos de entrada y son las que se encargan de propagar la información de una de salida de una capa de neuronas a otra (Negnevitsky, 2005). Existen diferentes tipos de funciones de activación las más comunes son las siguientes:
 - a) **Lineal:** La función de activación lineal genera un valor de salida equivalente a la entrada ponderada de la neurona. Esta función es comúnmente usada para aproximaciones lineales (Negnevitsky, 2005).

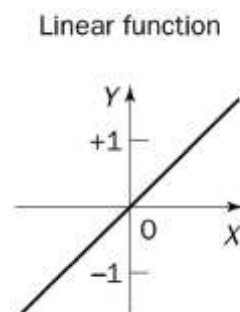


Figura 2-4. Función lineal (Negnevitsky, 2005)..

- b) **Sigmoidea:** La función sigmoidea transforma la entrada, que puede tener cualquier valor en el rango entre 0 y 1 (Negnevitsky, 2005).

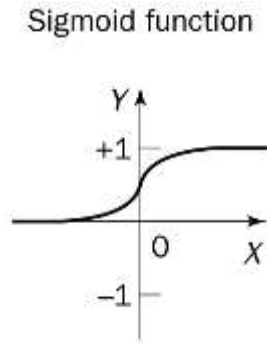


Figura 2-4. Función sigmoidea (Negnevitsky, 2005).

- c) **Tangente hiperbólica(tanh):** Los rangos normalizados de la función tanh van entre valores de -1 y 1. Su ventaja es que puede tratar más fácilmente con los números negativos (Josh Patterson, 2017).

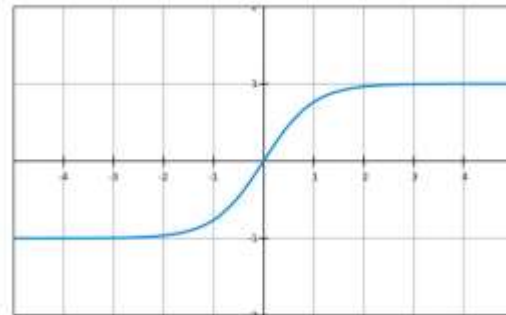


Figura 2-4. Función tanh (Josh Patterson, 2017).

- **Parámetros:** son los datos que el algoritmo ajusta a los datos de entrenamiento, a diferencia de los hiperparámetros, los parámetros no pueden ser configurados por los programadores o científicos de datos. Por ejemplo, los coeficientes de peso (o la pendiente) de una línea de regresión lineal y su término de sesgo.
- **Precisión:** Se define como el número de todas las predicciones correctas dividido entre el número de muestras en el conjunto de datos (Raschka, 2018).

- **Función de pérdida:** La función de pérdida se encarga de medir que tan cerca se encuentra la salida real de una red neuronal del resultado esperado. Se calcula una métrica basada en el error de las predicciones, esos errores son agregados a todo el conjunto de datos y se promedian para obtener un solo valor representativo de lo cerca que esta una red neuronal de su resultado esperado. En base a ese error se pueden determinar los nuevos pesos para minimizar esa pérdida como un problema de optimización (Josh Patterson, 2017). A continuación, se presentan unas de las funciones de pérdida más utilizadas para la regresión:
- **MSE:** El Error Cuadrático Medio es una función comúnmente utilizada cuando se trabaja en un modelo de regresión que requiere una salida de valor real. Consiste en tomar el error de predicción, elevarlo al cuadrado y promediarlo, como se muestra en la ecuación 2.1, consideremos el caso en el que tenemos que predecir sólo una característica de salida de salida ($M = 1$) (Josh Patterson, 2017).

$$MSE = L(W, b) \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2 \quad (2.1)$$

Donde:

N Es el número de muestras o conjunto de entradas con sus correspondientes resultados.

$L(W, b)$: Dados los datos disponibles, la notación de la función de pérdida indica que su valor

depende únicamente de W y b , los pesos y los sesgos de la red neuronal.

Y y \hat{Y} : Son la salida deseada y la predicha respectivamente

- **RMSE:** La Raíz del Error Cuadrático Medio es una medida de desempeño típica para los problemas de regresión. Nos da una idea de la cantidad de errores que el sistema presenta en sus predicciones dando un mayor peso a los errores grandes, como se muestra en la función 2.2.

$$RSME(X, h) = \sqrt{\frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})^2} \quad (2.2)$$

Donde:

m es el número de instancias del conjunto de datos en el que se está midiendo el RMSE.

$x^{(i)}$ es un vector de todos los valores de características (excluyendo la etiqueta) de la instancia i en el conjunto de datos, y $y^{(i)}$ es su etiqueta (el valor de salida deseado para esa instancia).

X es una matriz que contiene todos los valores de las características (excluyendo las etiquetas) de todas las instancias en el conjunto de datos.

h es la función de predicción del sistema, también llamada hipótesis.

$RSME(X, h)$ es la función de coste medida en el conjunto de ejemplos utilizando su hipótesis h .

- **MAE:** El Error Absoluto Medio es una medida de desempeño comúnmente usada para problemas de regresión donde hay muchos casos atípicos, como se muestra en la función 2.3.

$$\text{MAE}(X, h) = \frac{1}{m} \sum_{i=1}^m |h(x^{(i)}) - y^{(i)}| \quad (2.3)$$

Tanto el RMSE como el MAE son formas de medir la distancia entre dos vectores: el vector de predicciones y el vector de valores objetivo (Géron, 2017).

Los conceptos mencionados en esta sección fueron de mucha utilidad para el desarrollo del sistema, ya que permitieron conocer conceptos, herramientas y algoritmos para así poder elegir los mejores para el proyecto, así como adquirir conocimiento sobre el propósito y el desarrollo del proyecto.

Capítulo 3

3. Estado del arte

En este capítulo se presentan algunos trabajos que han utilizado tecnologías de Aprendizaje máquina e Internet de las cosas y han apoyado el sector acuícola, desarrollando tecnologías para el monitoreo, y el pronóstico de los parámetros de calidad del agua, así como otras problemáticas de diferentes sectores. En la sección 3.1 se presentan algunos sistemas de monitoreo desarrollados mediante Internet de las cosas y en la sección 3.2 sistemas de pronóstico desarrollados mediante Aprendizaje máquina.

3.1. Sistemas de monitoreo

A continuación, se presentan algunos sistemas de monitoreo que se han implementado mediante tecnologías de Internet de las cosas, los cuales han sido de mucha utilidad para las partes interesadas, ya que mediante dichos sistemas es posible obtener información en tiempo real de muchos procesos en diferentes tipos de industrias, información que puede servir para mejorar el desarrollo de muchos productos y servicios, ya sea para automatizar procesos, utilizar la información obtenida para el análisis de datos, así como también mejorar la seguridad de las personas.

3.1.1. Sistema de monitoreo de autobuses escolares basado en Internet de las cosas

Zambada y otros (Zambada Juan, 2015), desarrollaron un sistema de monitoreo y localización de autobuses escolares basado en Internet de las cosas para que las partes interesadas como padres de familia, gobiernos escuelas y otras autoridades puedan mantener un seguimiento en tiempo de real del comportamiento de los autobuses.

Este sistema se desarrolló utilizando sensores de velocidad y localización, los datos que son obtenidos por los sensores son enviados por medio del protocolo de comunicación MQTT que funciona por medio de un patrón publicador/subscriptor. En donde el subscriptor es el cliente, que puede ser cualquier parte interesada que necesita llevar un seguimiento de la información que se encuentra en un bróker en la nube y el publicador es el cliente que se

encarga de enviar los datos obtenidos de los sensores hacia un bróker. Además de almacenar los datos en una base de datos, la aplicación los envía como parámetros a la API "Google Maps JavaScript v3". La aplicación web muestra la información recibida de la API de Google Maps, donde se puede ver el mapa con la ubicación y la velocidad del transporte escolar como se muestra en la Figura 3-3. La arquitectura del este sistema se muestra en la Figura 3-2.

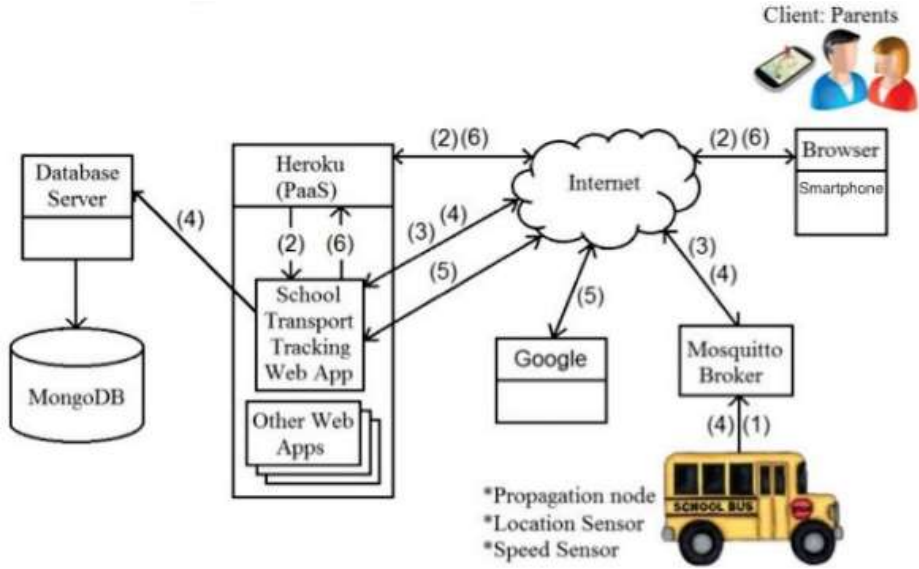


Figura 3-2. Sistema propuesto para el monitoreo de autobuses escolares (Zambada Juan, 2015).

Los datos son visualizados en una aplicación móvil desarrollada para Android y en una aplicación web desplegada en Heroku. (Center, 2022)

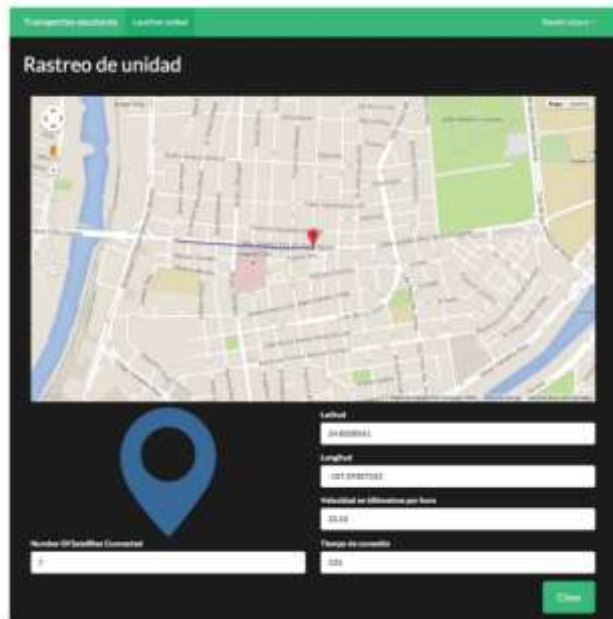


Figura 3-3. Visualización del monitoreo de los autobuses (Zambada Juan, 2015).

3.1.2. Una metodología novedosa para el monitoreo y control de la calidad del agua en la acuicultura usando Internet de las cosas

En este proyecto desarrollado por Abinaya y otros (Abinaya T, 2019), se diseñó un sistema que permite monitorear y controlar en tiempo real los parámetros de calidad del agua como: pH, temperatura, oxígeno disuelto, nivel de agua, olor y amoníaco en la. Los valores de los parámetros son detectados por diferentes sensores y son enviados a un dispositivo Arduino donde se detecta si los parámetros exceden los rangos establecidos, de ser así, se llevan a cabo una serie de acciones para controlar algunos de estos parámetros:

- a) Si el nivel de pH supera los rangos establecidos, significa que las algas han crecido en el agua, por lo cual, afectaría el crecimiento de los peces. Por lo tanto, se envía una alerta a las personas autorizadas para remover las algas y además se activa un controlador para reducir el valor de pH.
- b) Si el nivel de agua está por debajo de los rangos establecidos, se activa el motor de agua para volver a llenar el agua hasta el rango establecido, pero si supera este rango el motor se apaga.

Los datos que son obtenidos son enviados a un servidor en la nube a través de un módem inalámbrico conectado a él y son mostrados en la sala de control. La arquitectura de este sistema se muestra en la Figura 3-1.

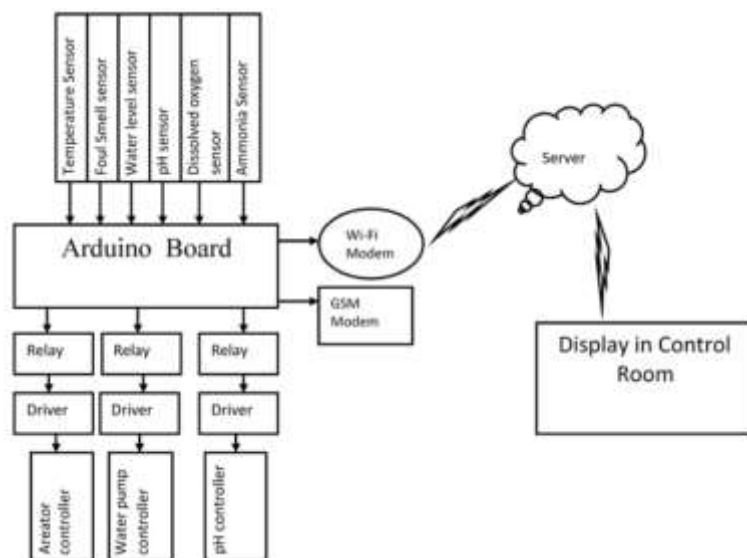


Figura 3-1. Sistema para el monitoreo y control de la calidad del agua en la acuicultura (Abinaya T, 2019).

3.1.3. Diseño e implementación de distribuido de Internet de las cosas para el monitoreo de la calidad del agua en la acuicultura

En este proyecto desarrollado por Cesar Encinas (Encinas Cesar, 2017), se diseñó y desarrolló una red de sensores inalámbricos utilizando el protocolo Zigbee para enviar los datos que se obtienen de los sensores a un servidor local y así poder visualizar los datos y recibir alertas para controlar los parámetros de calidad del agua en la acuicultura a través de internet. El sistema fue propuesto para funcionar en granjas acuícolas.

El sistema consiste en 3 sensores para medir 3 diferentes parámetros, en este caso oxígeno disuelto, temperatura y pH, los cuales envían la información de los estanques vía UART (*Universal Asynchronous Transmitter Receiver*), los datos recibidos de los sensores en el Arduino son enviados a una aplicación de escritorio a través del protocolo Zigbee. Los datos recibidos son desplegados en dicha aplicación, almacenados en una base de datos local y son enviados a un servicio web para que también puedan ser mostrados en una aplicación móvil.

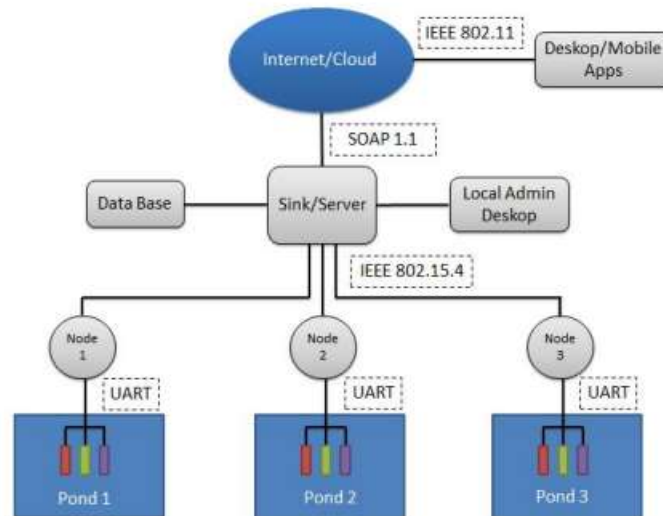


Figura 3-4. Arquitectura del sistema de distribuido de Internet de las cosas (Encinas Cesar, 2017).

En este proyecto también se propone un módulo de IA que tomará la información de la base de datos para inferir cuándo enviar alarmas sobre las anormales en los estanques acuícolas, el cual se propone que funcione con lógica difusa.

En todos estos trabajos que se han implementado para el sector acuícola se han enfocado en las granjas o en otras áreas de otros sectores. A diferencia del presente proyecto, que da la oportunidad al usuario de poder implementarlo en diferentes áreas y contar con el monitoreo y el pronóstico integrado. Además de contar con un control de usuarios que permite el acceso y envío de notificaciones a los usuarios registrados con su correo electrónico (esto lo hace más seguro, en caso de que la organización quiera mantener su información privada) y contar una base de datos para futuras funciones que quisieren implementar en el futuro.

3.2. Sistemas de pronóstico

En esta sección se presentarán algunos proyectos sobre pronósticos de series de tiempo utilizando Aprendizaje máquina. Los modelos de pronóstico permiten obtener información anticipadamente, lo cual puede ser de mucha utilidad al momento de la prevención de riesgos en diferentes áreas, a continuación, se mencionan algunos ejemplos que han contribuido con dichos datos.

3.2.1. Estrategias de pronóstico en múltiples pasos para series de tiempo de la velocidad del viento

En el proyecto de Rodríguez y Medrano (Rodríguez Hector, 2020), especificaron un conjunto de estrategias para el pronóstico a futuro utilizando cinco conjuntos de datos de series de tiempo de la velocidad del viento. Tomando en cuenta que una buena estimación de las ráfagas de viento implica una correcta planificación en la generación de energía eólica limpia, propusieron la estimación de ráfagas de viento mediante técnicas de Inteligencia Artificial. Utilizaron una red neuronal LSTM, cinco conjuntos de datos de velocidad del viento diferentes, y diferentes estrategias de pronóstico de múltiples pasos. Las estrategias utilizadas son la recursiva, la directa, la MIMO (entrada múltiple a salida múltiple), la DIRM (combinación de estrategia directa y MIMO) y la DirREC (combinación de estrategia directa y recursiva).

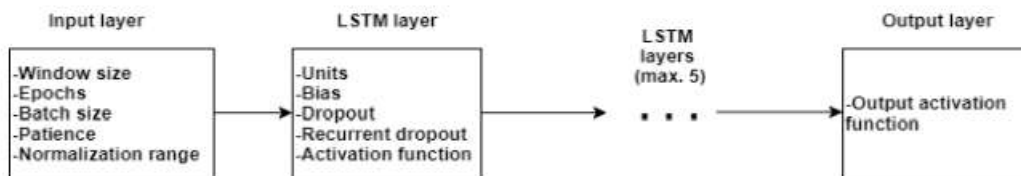


Figura 3-5. Configuración de la red neuronal LSTM (Rodríguez Hector, 2020).

La optimización de hiperparámetros se automatizó mediante Algoritmos Genéticos (AG) y búsqueda aleatoria (RS).

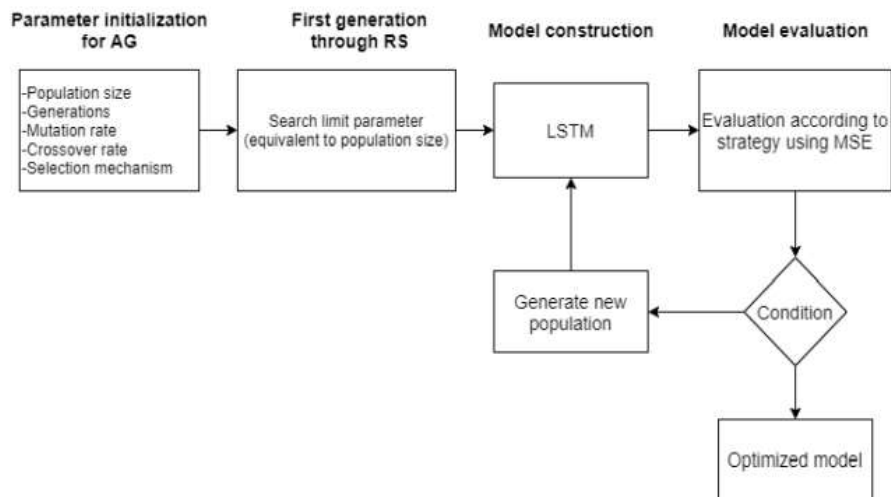


Figura 3-6. Proceso de optimización de hiperparámetros (Rodríguez Hector, 2020).

Se concluyó que, con el uso de estas estrategias, se puede reducir la cantidad de error e incertidumbre acumulados para cada paso en el tiempo, evitando la reducción de la precisión a medida que se avanza.

3.2.2. Predicción del nivel y de la calidad del agua utilizando un enfoque de aprendizaje profundo combinado CNN-LSTM

Para este proyecto se combinó una red neuronal convolucional con una red neuronal *Long Short-Term Memory (LSTM)*, que sirvieron para simular la calidad del agua en la cuenca del río Nakdong, en Corea del Sur. Los parámetros de calidad del agua utilizados para la simulación fueron: nitrógeno total, fosforo total, carbono orgánico total y el nivel del agua. Los datos sobre el nivel del agua fueron recopilados mediante el Sistema de Información de Gestión de Recursos Hídricos (WAMIS) y la información sobre la calidad del agua fue obtenida mediante la Información sobre la calidad del agua en tiempo real (RTWQI). Las precipitaciones y la información sobre el funcionamiento de la presa del estuario se obtuvieron de la Administración Meteorológica de Corea.

Se utilizó el modelo CNN (representado en la Figura 3-7) para simular el nivel del agua, mientras que se seleccionó el modelo LSTM (representado en la figura 3-8) para simular la concentración de los contaminantes mencionados anteriormente.

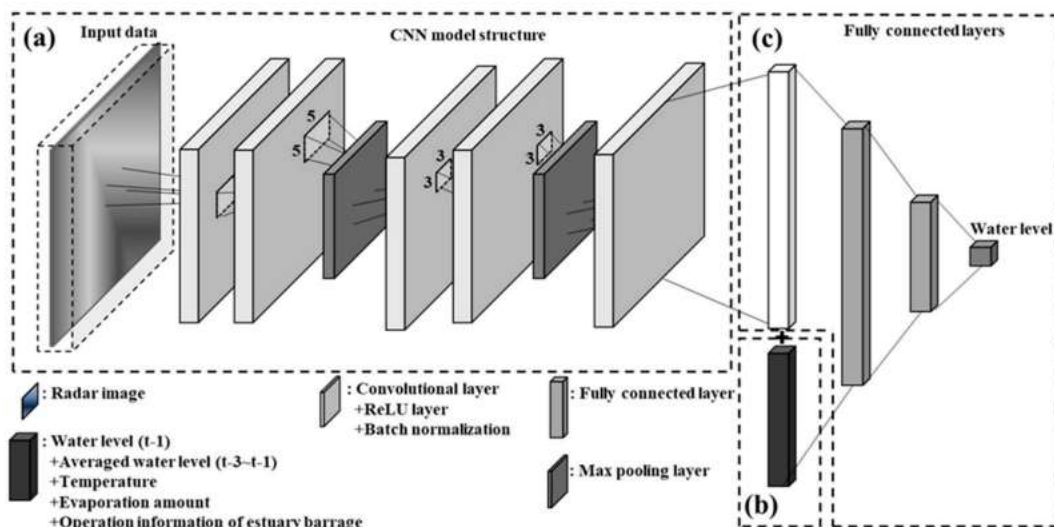


Figura 3-7. Arquitectura de CNN para del nivel del agua (Baek Sang-Soo, 2020).

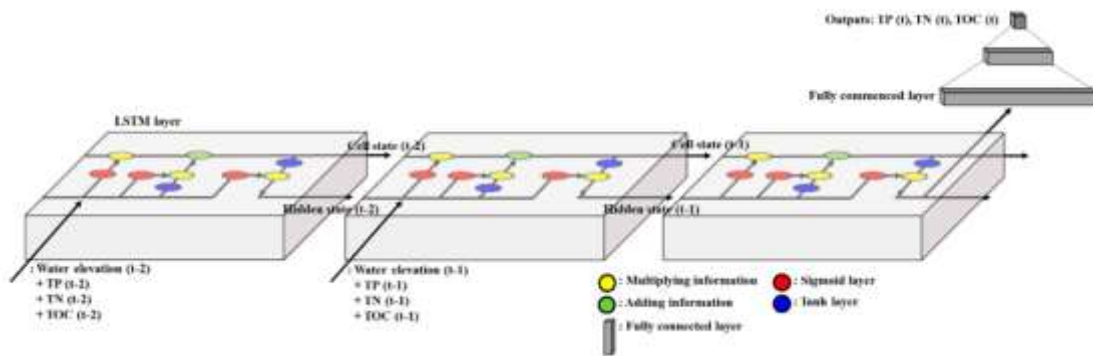


Figura 3-8. Arquitectura LSTM para los parámetros de calidad del agua (Baek Sang-Soo, 2020).

En este estudio se concluyó que el nivel de agua del modelo CNN produjo un valor de Eficiencia de *Nash-Sutcliffe* (NSE) de 0,933 que puede considerarse como rendimiento aceptable del modelo. Los niveles de agua aumentaron en la estación lluviosa, mientras que fueron bajos en la estación seca. Y para todos los parámetros de calidad, los valores NSE del modelo LSTM para los periodos de entrenamiento y validación fueron superiores a 0.75, lo que está dentro del rango de rendimiento "muy bueno". El modelo LSTM en este estudio representó bien las diferentes variaciones temporales de cada tipo de parámetro.

Los resultados de este estudio arrojaron que estas herramientas pueden ser muy útiles para la simulación de la calidad del agua. Aunque para este estudio solo se utilizaron los parámetros de nitrógeno total, fosforo total, carbono orgánico total y el nivel del agua, puede ser recomendado utilizar estos modelos para simular otro tipo de parámetros (Baek Sang-Soo, 2020).

3.2.3. Pronóstico de radiación solar a largo plazo utilizando un enfoque de aprendizaje profundo-GRU

En el proyecto de Aslam y otros (Aslam Muhammad, 2019), se realizó un pronóstico de la radiación solar, ya que mediante este indicador es posible estimar la generación solar, que es un tipo de energía renovable. Se realizó el pronóstico por hora y por día, con un año de anticipación, utilizando un modelo basado en una red neuronal *Gated Recurrent Unit* (GRU) el cual fue comparado con los otros modelos más conocidos como las *Long-short Term*

Memory (lstm), las *Recurrent Neural Networks*, los *Support Vectore Regression (SVR)*, y *Feed Forward Neural Network (FFNN)*.

Year	SVR	FFNN	RNN	LSTM	GRU
2016	4.8204	4.9423	5.3857	4.8175	4.7936
2015	5.3084	5.2552	5.5243	5.2000	5.0880

Figura 3-9. RMSE de radiación solar de los diferentes modelos (Aslam Muhammad, 2019).

Los resultados de este experimento arrojaron que modelo GRU fue el que obtuvo valores de RMSE más bajo en comparación con los otros modelos.

Se pudo concluir que además GRU se desempeñó entre un 14 % y un 23 % más rápido que LSTM y fue el modelo que obtuvo mejores resultados en comparación con el resto. En la Figura 3-9 se presenta la comparación de los modelos frente a los datos reales en periodos de tiempo mensuales.

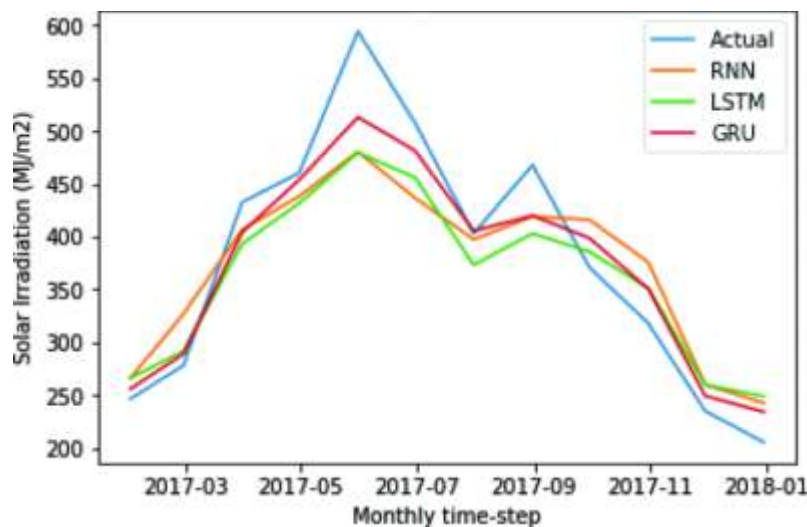


Figura 3-10. Comparación de modelos en periodos mensuales (Aslam Muhammad, 2019).

3.2.4. Aprendizaje profundo para el pronóstico de series de tiempo multivariantes no estacionarias.

En el proyecto de Almuammar y Fasli (Almuammar Manal, 2019), se utilizaron diferentes tipos de tecnologías para realizar un pronóstico de los espacios de estacionamientos. Utilizaron Internet de las cosas para monitorear y obtener datos suficientes para entrenar un

modelo GRU para realizar el pronóstico. Los datos utilizados para estimar disponibilidad de estacionamientos se basan un contador que aumenta cada que un usuario paga la tarifa de estacionamiento y se reduce en función de la llegada de nuevos elementos en el flujo de sensores que indican que los coches, vinculados con la información de pago de pago, han abandonado el estacionamiento, los datos fueron almacenados en MS-SQL ordenados por la hora de pago y la hora de salida del estacionamiento. Los estacionamientos se controlan sólo durante una parte del día, desde las 7:00 de la mañana hasta las 16:00. Por lo cual se generó una serie temporal de 184.323 registros en total, para las 11 semanas.

Para conocer el rendimiento del modelo GRU se realizaron comparaciones con un modelo MLP utilizando el mismo conjunto de datos para el entrenamiento. Se concluyó que los modelos GRU supera a los métodos estadísticos en cuestión de realizar predicciones en ambientes dinámicos pero un modelo MLP puede tener mejores resultados y hasta mejor rendimiento que una GRU.

Model	Accuracy	Loss	MAE	MSE
1	0.88870	0.00797	0.062989	0.00869
2	0.89882	0.00700	0.05520	0.00725
3	0.84830	0.00904	0.06707	0.00995
4	0.86003	0.00746	0.05961	0.00789
5	0.90371	0.00720	0.06063	0.00816
6	0.89358	0.00933	0.06951	0.00990
7	0.90203	0.00820	0.06486	0.00898
8	0.90541	0.00750	0.05894	0.00792
9	0.85448	0.00738	0.05890	0.00798
10	0.87817	0.00782	0.05791	0.00825
11	0.90863	0.00822	0.06317	0.00914
12	0.84264	0.00893	0.06114	0.00862
13	0.87671	0.00654	0.05176	0.00654
14	0.82191	0.00698	0.06029	0.00849
15	0.87407	0.00442	0.04960	0.00566
16	0.81667	0.00934	0.07370	0.01182

Figura 3-11. Funcionamiento de los modelos GRU (Almuammar Manal, 2019).

MLP Model	Accuracy	Loss	MAE	MSE
1	0.86531	0.00402	0.05242	0.00402
2	0.76178	0.01525	0.09314	0.01523
3	0.83922	0.00340	0.04481	0.00340

Figura 3-12. Funcionamiento de los modelos MLP (Almuammar Manal, 2019).

Como se presentó anteriormente, estos trabajos son de gran ayuda en diferentes áreas o industrias. Los sistemas de monitoreo de Internet de las cosas permiten obtener información en tiempo real que sirve para mantener informado al usuario de los diferentes procesos, esta disciplina al ser combinada con modelos de Aprendizaje máquina para el pronóstico, ayudan a las partes interesadas a anticiparse a posibles sucesos, lo cual hace que se puedan desarrollar proyectos más completos que puedan abordar más problemáticas.

El enfoque presentado en este trabajo, da la oportunidad al usuario de poder implementarlo en diferentes áreas del sector acuícola y contar con el monitoreo y el pronóstico integrado. Además de contar con un control de usuarios que permite el acceso y envío de notificaciones a los usuarios registrados con su correo electrónico (esto lo hace más seguro, en caso de que la organización quiera mantener su información privada) y contar una base de datos para futuras funciones que quisieren implementar en el futuro.

En la Tabla 3-1 se presentan las 4 características principales que se tomaron en cuenta al momento de realizar el proyecto, en este caso Monitoreo, Pronóstico, Control de usuarios y Alertas. Se realiza una comparación del presente proyecto con los proyectos presentados en el estado del arte, los cuadros marcados con naranja representan las características que contiene cada proyecto, dichos proyectos fueron seleccionados y estudiados para obtener información y servir como apoyo al presente proyecto.

En esta comparativa se puede observar que ninguno de los proyectos del estado del arte cuenta con las 4 características y que solo los proyectos 3.1.2, 3.1.3 y 3.2.2 están enfocados a la acuicultura.

Tabla 3-1. Comparación de las características principales de los proyectos del estado del arte.

Proyectos Estado del arte	Monitoreo	Pronóstico	Control de usuarios	Alertas
Aplicación web para el monitoreo remoto de los parámetros de calidad del agua en estanques acuícolas usando tecnologías de Aprendizaje maquina e Internet de las cosas				
3.1.1. Sistema de monitoreo de autobuses escolares basado en Internet de las cosas.				
3.1.2. Una metodología novedosa para el monitoreo y control de la calidad del agua				

en la acuicultura usando Internet de las cosas.				
3.1.3. Diseño e implementación de distribuido de Internet de las cosas para el monitoreo de la calidad del agua en la acuicultura.				
3.2.1. Estrategias de pronóstico en múltiples pasos para series de tiempo de la velocidad del viento.				
3.2.2. Predicción del nivel y de la calidad del agua utilizando un enfoque de aprendizaje profundo combinado CNN-LSTM.				
3.2.3. Pronóstico de radiación solar a largo plazo utilizando un enfoque de aprendizaje profundo-GRU.				
3.2.4. Aprendizaje profundo para el pronóstico de series de tiempo multivariantes no estacionarias.				

Capítulo 4

4. Desarrollo del proyecto

La metodología que se llevó a cabo para el desarrollo del proyecto presentada en la Figura 4-1, se divide en 3 etapas. La primera es la etapa de análisis e investigación, en donde se llevaron a cabo actividades como la revisión del estado del arte (presentada en el Capítulo 5) y el análisis de requisitos. La segunda, es la etapa del desarrollo del sistema, en donde se llevaron a cabo las actividades de diseño de las arquitecturas, codificación, realización de pruebas y obtención de resultados. Finalmente, la etapa de análisis de resultados, en donde se presentan las pruebas realizadas, resultados obtenidos (presentados en el capítulo 5) y las conclusiones (presentadas en el capítulo 6). En este capítulo se presenta como se llevó a cabo el análisis de requisitos y las actividades relacionadas con el desarrollo del sistema.

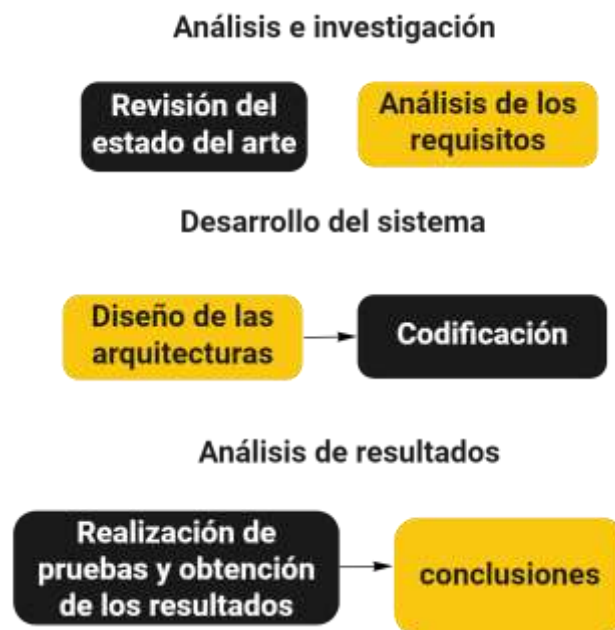


Figura 4-1. Metodología

4.1.Planeación e investigación

Durante esta etapa se llevó a cabo la investigación del estado del arte, el análisis de requisitos, y se realizó la selección de las herramientas. Dichas actividades se describen a continuación.

4.2. Análisis

En la etapa de análisis, se llevó a cabo el modelo de requisitos, este modelo se desarrolló en base a investigaciones previas y a una visita que se realizó a un laboratorio acuícola en dónde se realizaron algunas preguntas para saber cuáles requisitos debe cumplir el sistema en base a sus necesidades. Durante esta etapa se realizó el diagrama de contexto, se describieron los requisitos funcionales y de calidad, los casos de uso, así como el mapeo de los casos de uso con los requisitos funcionales y finalmente el modelo de interacción de usuarios.

4.2.1. Diagrama de contexto

Para conocer los límites del sistema y su interacción con otras entidades se realizó el diagrama de contexto presentado en la Figura 4-2. Dichas entidades se describen a continuación:

- **Base de datos:** La aplicación web depende de la base de datos para almacenar y obtener el estado de los parámetros de calidad del agua provenientes de los sensores y la información del registro de los usuarios.
- **Dispositivo Internet de las cosas:** La aplicación web usa al dispositivo de Internet de las cosas, para obtener los datos de los sensores de oxígeno disuelto y temperatura.
- **Modelo de pronóstico:** La aplicación web usa al modelo de pronóstico para obtener el pronóstico de los datos.
- **Usuario:** La aplicación web es usada por el usuario por medio de la información del registro y acceso de usuarios y la Interfaz gráfica.

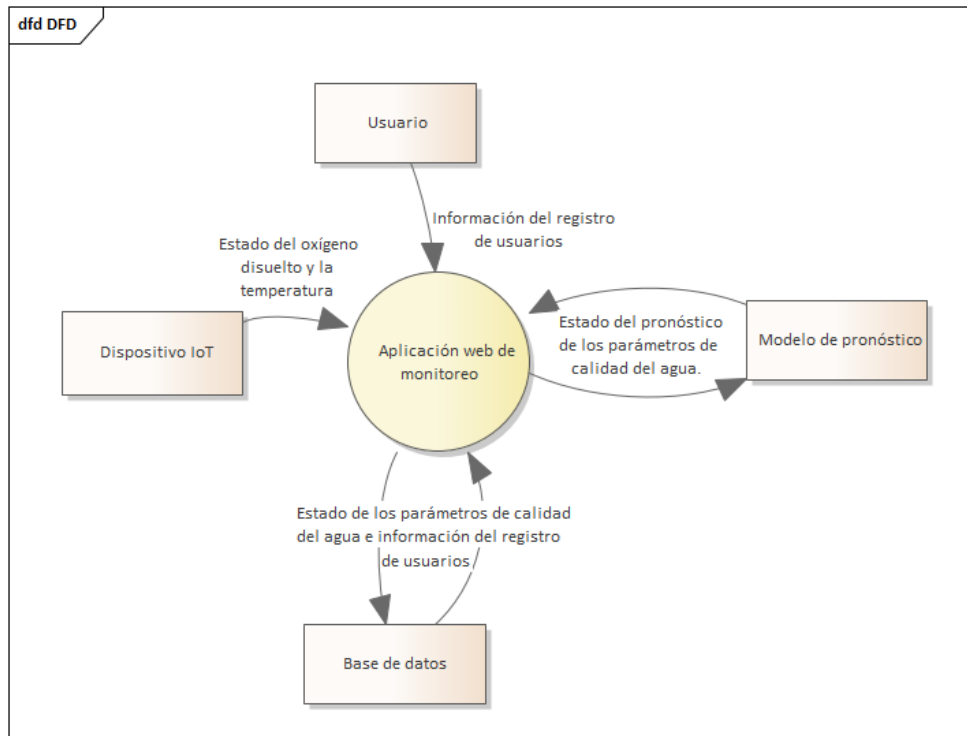


Figura 4-2. Diagrama de contexto

4.2.2. Requisitos funcionales

Los requisitos funcionales descritos en la Tabla 4-1, se definieron para determinar cuáles son los servicios que deberá ofrecer el sistema a los usuarios. Se definieron en base a una visita que se realizó a un laboratorio acuícola, en donde se llevó a cabo una entrevista con las partes interesadas para conocer cuáles eran sus necesidades. Se describieron mediante un identificador, el nombre y la descripción, así como su nivel de importancia, dichos indicadores se describen a continuación:

- **Alto:** Son aquellos requisitos que son indispensables para funcionalidad del proyecto, es decir que deben ser implementados para cumplir con el objetivo principal del sistema.
- **Medio:** Son aquellos requisitos que construyen a mejorar la funcionalidad del proyecto, pero no son indispensables, es decir, el no ser implementados, no afecta al desarrollo del objetivo principal del proyecto.
- **Bajo:** Son aquellos requisitos que no afectan la funcionalidad principal del proyecto y el implementarlos solo agrega funcionalidad extra al usuario.

Tabla 4-1. Requisitos funcionales.

No.	Nombre	Descripción	Nivel de Importancia
RF01	Recibir y actualizar datos	El sistema recibirá los datos provenientes de los sensores de oxígeno disuelto y temperatura cada 10 min.	Alto
RF02	Notificar anomalías	El sistema será capaz de detectar cuando un parámetro este fuera del rango permitido y enviará notificaciones por correo o por mensaje de texto al usuario. También se notificará al usuario cuando la predicción detecte anomalías.	Alto
RF03	Almacenar datos	Los datos se almacenarán en un archivo de texto y posteriormente se subirán a la nube para asegurar la disponibilidad de estos tanto en la aplicación móvil como en la aplicación web.	Alto
RF04	Mostrar el estado de los parámetros	Los datos más actuales del monitoreo y pronóstico serán mostrados en la interfaz gráfica de usuario.	Alto
RF05	Predecir el comportamiento de los parámetros.	El sistema generara una predicción de los parámetros de calidad del agua. La predicción se realizará cada hora y se graficará en intervalos de 5 minutos.	Alto
RF06	Registrar usuarios	El sistema deberá registrar al usuario solicitando su nombre, puesto de trabajo, correo electrónico, usuario y contraseña.	Medio
RF07	Autenticar usuarios	El usuario podrá acceder al sistema mediante su usuario y contraseña. El sistema validara y si ambos datos son correctos.	Medio
RF08	Cambiar contraseña	El sistema permitirá al usuario cambiar su contraseña en caso de no ser recordada.	Bajo

4.2.3. Requisitos de calidad

Los requisitos de calidad son la base de las decisiones arquitectónicas. Se definieron con la finalidad de evaluar que el sistema funcione de forma correcta, estos requisitos nos dan un margen sobre que modelos arquitectónicos o patrones de diseño utilizar para su cumplimiento. Al igual que los requisitos funcionales estos se definieron mediante una entrevista en el laboratorio y se les asigno un identificador, el atributo de calidad, su descripción y su nivel de importancia, alto, medio y bajo, dichos indicadores se describen a continuación:

- **Alto:** Son aquellos requisitos que son necesarios para cumplir con el objetivo principal del proyecto, prevenir riesgos y manejar de forma segura la información.
- **Medio:** Son aquellos requisitos cuya implementación no afecta el objetivo principal del proyecto, pero los usuarios lo consideran necesario para su interacción con el sistema.
- **Bajo:** Son aquellos requisitos cuya implementación no es necesaria para cumplir con el objetivo principal del proyecto, sin embargo, contribuyen a mejorar su desempeño.

Tabla 4-2. Requisitos de calidad.

No.	Atributo	Descripción	Nivel de importancia
RC01	Fiabilidad	El sistema será capaz de mantener disponibles los datos más recientes en caso de alguna interrupción o fallo.	Alto
RC02	Seguridad	El sistema almacenara y maneja los datos de forma segura.	Alto
RC03	Mantenibilidad	El sistema será capaz de aceptar cambios.	Alto
RC04	Usabilidad	El sistema permitirá que nuevos usuarios aprendan a usarlo en un máximo de 15 min.	Medio
RC05	Portabilidad	El sistema podrá ser utilizado en cualquier navegador web.	Bajo

4.2.4. Casos de uso

En base a los requisitos funcionales se definieron los casos de uso, para representar las interacciones que tendrás los actores con el sistema. Entre los actores se encuentran:

- **El prototipo Internet de las cosas:** Es el sistema que se encarga del monitoreo de los parámetros de calidad del agua en el estanque. Recibe el estado del oxígeno disuelto y temperatura, mediante sensores envía los datos a una aplicación móvil a través de bluetooth, para posteriormente enviar dichos datos a la aplicación web a través de internet.
- **El Usuario:** Es la persona que se registrará y dará uso al sistema para mantenerse informado sobre el estado del estanque
- **El modelo de pronóstico:** Es un servicio externo que recibe la información actual de los parámetros mediante la aplicación web, procesa los datos para realizar el pronóstico y lo envía a la aplicación para que los datos sean visibles para el usuario.

En la Tabla 4-3 se muestra la descripción de los casos de uso, se les asignó un identificador, el nombre, la descripción y el actor que va a realizar dicha interacción. Y en la Figura 4-3 se muestra el diagrama de casos de uso, el cual representa de forma la interacción entre los casos de uso y los actores.

Tabla 4-3. Casos de uso.

No.	Nombre	Descripción	Actor
CU01	Enviar información del estado	El sistema recibe la información del estado de los parámetros de calidad del agua mediante el sistema de sensores.	Prototipo Internet de las cosas
CU02	Enviar el pronóstico de los parámetros	El modelo de pronóstico recibirá información del estado de los sensores y enviará el pronóstico realizado a la aplicación mediante dicha información.	Modelo de pronóstico
CU03	Consultar el estado de los parámetros	El usuario podrá visualizar la información actualizada del estado actual de los parámetros y la predicción mediante la interfaz gráfica de usuario.	Usuario
CU04	Notificar anomalías	El usuario recibirá notificaciones de cuando el sistema detecte que los parámetros están fuera de los rangos permitidos o cuando esto se detecte en la predicción.	Usuario
CU05	Registrar en la aplicación	El usuario se registra en el sistema mediante su nombre, correo electrónico, usuario y contraseña.	Usuario
CU06	Autenticar al usuario	El usuario ingresa al sistema mediante su nombre de usuario y contraseña. En caso de no recordar su usuario o contraseña, estos datos podrán ser modificados.	Usuario

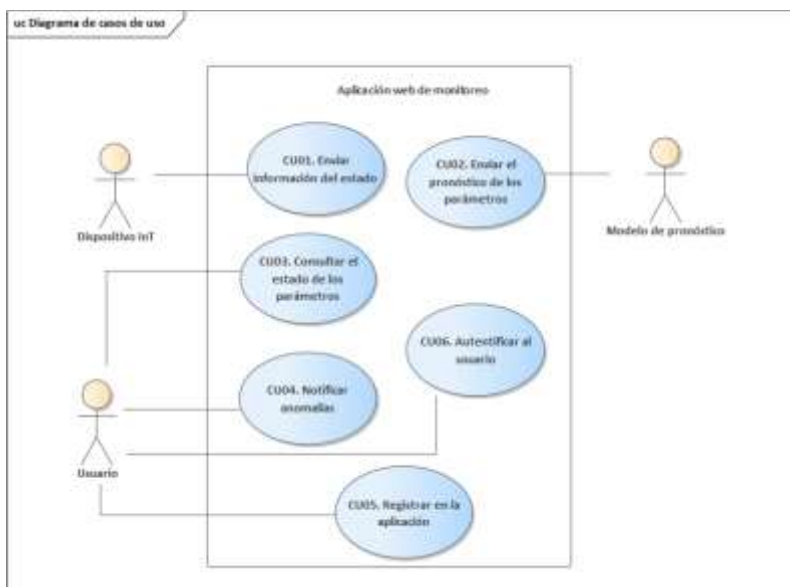


Figura 4-3. Diagrama de casos de uso

4.2.5. Mapeo de los requisitos funcionales y casos de uso

Finalmente se realizó un mapeo de los requisitos funcionales con los casos de uso para identificar las relaciones que hay entre ellos, es decir cuales casos de uso están bajo la cobertura de los diferentes requisitos funcionales, dicha interacción se muestra en la Figura 4-4.

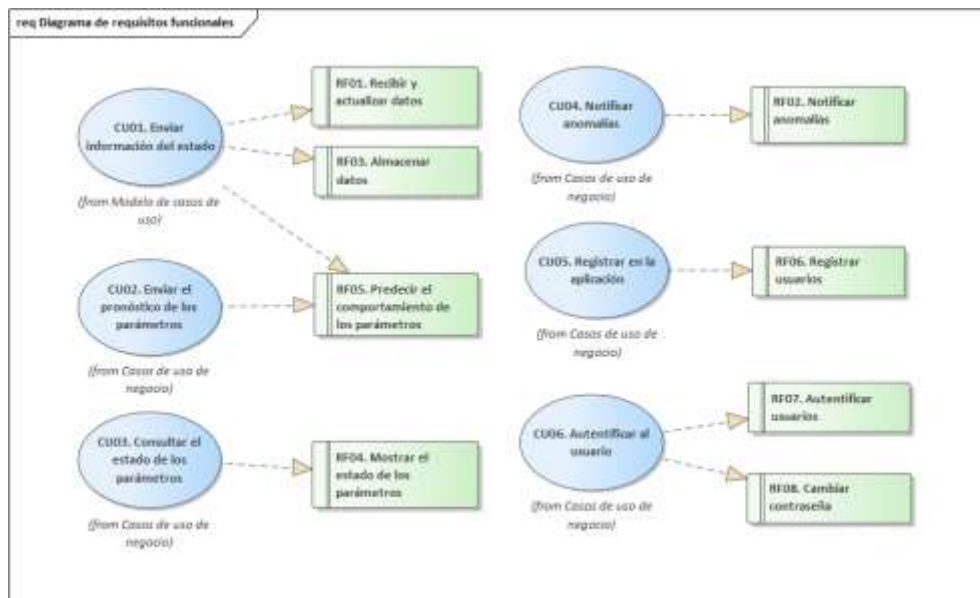


Figura 4-4. Mapeo de casos de uso con requisitos funcionales

4.3. Diseño

Una vez que se obtuvo el modelo de requisitos se obtuvieron los arquetipos y se llevó a cabo el diseño de la arquitectura física y lógica, así como el patrón de diseño que se utilizaría. La implementación del sistema se realizó para apoyar al personal con el monitoreo y la predicción de los parámetros al momento de transportar el producto acuícola. Debido a esto se decidió utilizar *Bluetooth Low Energy* y una aplicación móvil como medio de comunicación entre el Prototipo Internet de las cosas y la aplicación web. La idea de este diseño es que la arquitectura de la aplicación web esté desacoplada del resto y se pueda implementar en diferentes escalas, es decir tanto para el transporte del producto, como en laboratorios o granjas acuícolas, tomando en cuenta que solo se tendría que cambiar el medio de comunicación entre en prototipo Internet de las cosas y la aplicación web.

4.3.1. Modelo de dominio.

La Figura 4-5 representa las abstracciones al más alto nivel del sistema, los cuales se describen a continuación:

- **Usuario:** Representa al usuario que realiza diversas actividades en el sistema como registrarse y acceder a la aplicación para visualizar los datos de los parámetros y recibir alertas cuando se detecte alguna anomalía.
- **Estado del estanque:** Representa los datos del estado del estanque que se obtienen a través de los sensores en formato Json.
- **Gestor:** Es la entidad que administra la información, recibe, almacena y muestra la información de los sensores, usuarios y el pronóstico.
- **Pronóstico:** Es la entidad que recibe el estado de los estanques, procesa los datos para obtener el pronóstico y lo envía al gestor para que los usuarios lo puedan visualizar.



Figura 4-5. Diagrama de modelo de dominio

4.3.2. Arquitectura lógica

Los requisitos de calidad y el modelo de dominio fueron una parte fundamental para el diseño arquitectónico del sistema. El modelo de dominio permitió conocer las entidades y de qué manera abordan los objetivos que el sistema debe cumplir, lo cual fue de gran utilidad al

momento de diseñar los módulos del sistema. Se optó por un modelo arquitectónico en capas porque permite cumplir los requisitos de calidad de la siguiente manera:

- **Fiabilidad:** El sistema cuenta con una capa de datos en donde se localiza la base de datos que almacena y asegura su disponibilidad por medio de peticiones HTTP, además de que la aplicación móvil cuenta con un módulo que permite almacenar temporalmente los datos obtenidos de los sensores en caso de un fallo.
- **Seguridad:** Al estar dividido en diferentes paquetes o capas, el sistema se vuelve más seguro, ya que la capa de datos se encuentra desacoplada del resto, lo cual permite que no se tenga acceso directo a los datos, utilizando la capa de dominio como intermediario entre la capa de presentación y la capa de datos.
- **Mantenibilidad:** El sistema es mantenible ya que está dividido en diferentes capas y módulos que representan diferentes funcionalidades, de esta manera cuando sea necesario implementar un cambio, este solo afectaría al módulo en donde se encuentra dicha funcionalidad y no al resto del sistema.
- **Usabilidad:** El sistema contiene una capa que representa la interfaz gráfica de usuario, la cual se diseñó para que pueda ser manejada por el usuario de forma simple.
- **Portabilidad:** El usuario puede acceder a la información mediante una página web desde cualquier dispositivo mediante un navegador web.

El sistema se compone por la capa de presentación, que es la que realiza la interacción del sistema con el usuario. La capa de dominio es en donde se encuentra la lógica del sistema. La capa de datos que donde se encuentra la base de datos. La capa física que donde se encuentra la interfaz de conexión con el dispositivo Internet de las cosas y finalmente la capa de servicios externos, donde se encuentra desplegado el modelo de Aprendizaje máquina. A continuación, se describen los componentes y módulos que conforman cada una de las capas:

Capa de presentación

- **Interfaz GUI:** Este es el módulo donde se encuentra la interfaz gráfica de usuario, que se encarga de crear una representación visual del sistema (como los formularios de registro, botones de acceso, graficas etc.) para que el usuario pueda hacer uso de el de una forma mucho más amigable.

Capa de dominio

- **Modulo usuario:** En este módulo se encuentran los componentes que generan el registro y autenticación de usuarios en el sistema. Este módulo se encarga de recibir, almacenar y solicitar en la base de datos, la información que los usuarios proporcionan o solicitan a través de la capa de presentación para realizar dichas funciones.
- **Modulo monitoreo:** En este módulo se encuentran los componentes para almacenar y actualizar datos. El primero es el que se encarga de recibir los datos que provienen de la capa física, que es la que hace interacción con el prototipo Internet de las cosas y los almacena en la base de datos, mientras que el componente de actualización es el que se encarga de hacer peticiones a la base de datos para mostrarlos en la GUI y que estos puedan ser visualizados por el usuario.
- **Modulo notificación:** Este módulo se encarga de detectar si los datos de los parámetros se encuentran dentro de los rangos establecidos, de no ser así envía un correo electrónico a los usuarios registrados en el sistema con la finalidad de alertar sobre una condición anormal en el estanque.
- **Módulo predicción:** Este módulo es el que se encarga de predecir mediante un algoritmo de Aprendizaje máquina los datos en un tiempo en el futuro, con la finalidad de alertar al usuario que podría ocurrir una anomalía en el estanque y que de esta manera le sea posible tomar medidas a tiempo.

Capa de datos

- **Base de datos:** En esta capa se encuentra la base de datos del sistema, la cual contiene una tabla para almacenar la información del monitoreo que envía el prototipo Internet de las cosas, otra para la información del usuario y otra para la predicción de los parámetros.

Capa física

- **Interfaz parámetros:** Este es el módulo que comunica al prototipo Internet de las cosas con la aplicación web. Se trata de una aplicación móvil que recibe los datos que provienen de los sensores a través de *Bluetooth Low Energy* y los procesa para

enviarlos en el formato adecuado a la aplicación web a través de internet. Con la finalidad de asegurar la disponibilidad de los datos, esta aplicación también contiene un componente que se encarga de almacenar los datos en un archivo de texto en caso de que se pierda la conexión a internet para que una vez que se reestablezca, estos datos puedan ser visualizados por el usuario en la aplicación web.

- **Software:** Es el módulo que recibe los datos que provienen de la aplicación web a través de peticiones HTTP.

Capa de servicios externos

- **Modelo de pronóstico:** Es un modelo de Aprendizaje máquina que recibe los datos más actuales del monitoreo, los procesa para obtener el pronóstico y los envía de nuevo a la aplicación web a través de peticiones HTTP para que estos sean visibles para el usuario.

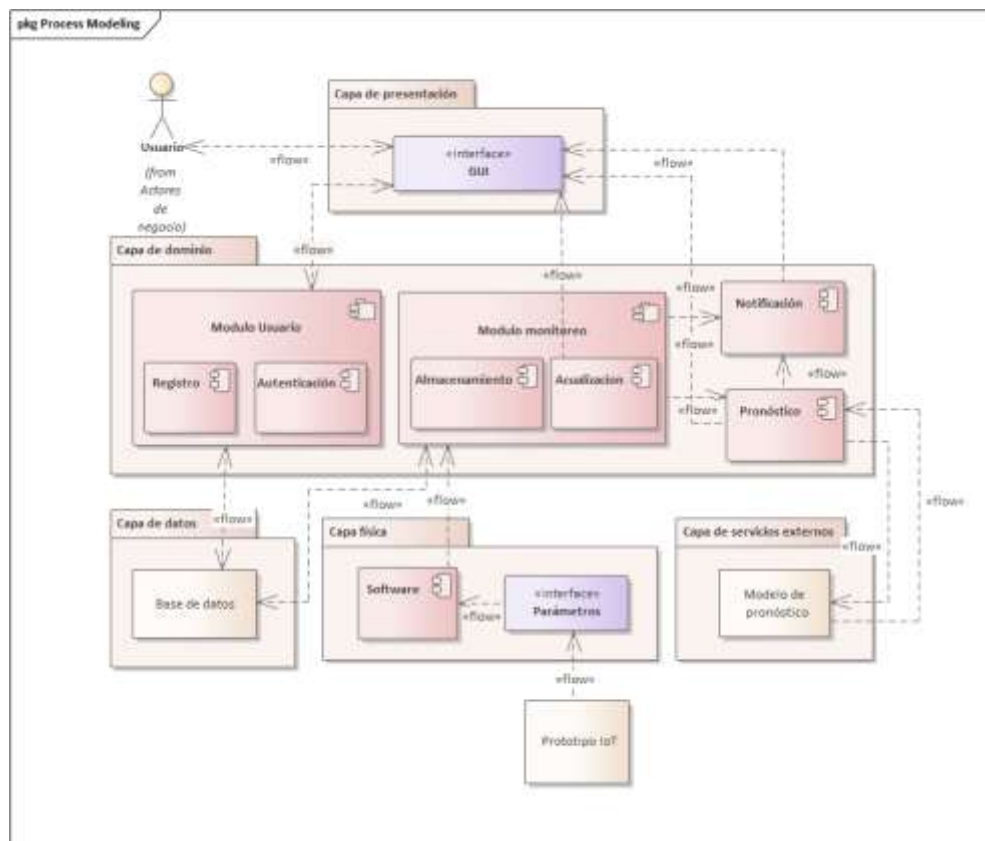


Figura 4-6. Arquitectura lógica

4.3.3. Modelo arquitectónico cliente-servidor

La Figura 4-7 representa los aspectos dinámicos de la arquitectura propuesta, están basados en el protocolo de comunicación cliente-servidor. La aplicación móvil juega el rol de cliente que envía los datos a través de peticiones HTTP al servidor. Al igual que la aplicación móvil, la aplicación web del lado del usuario también juega el rol de cliente que envía y recibe información de los sensores y del control de usuarios.

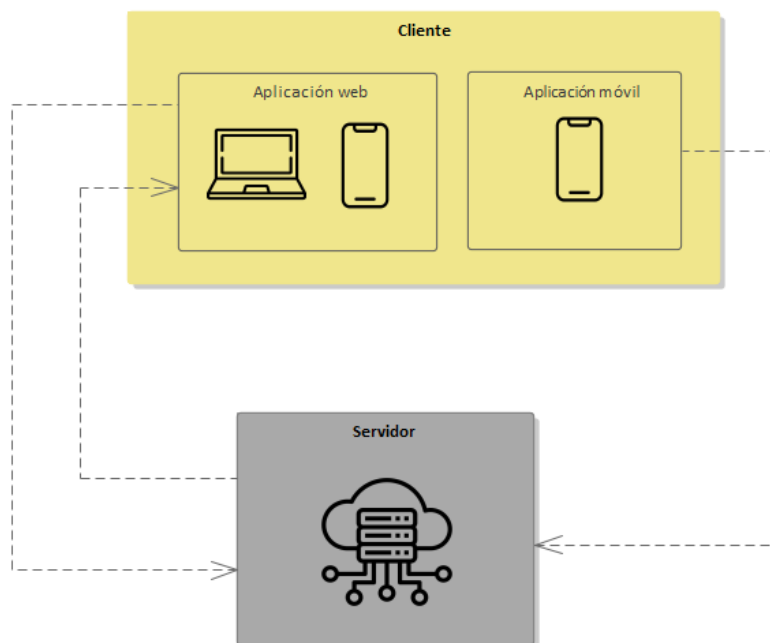


Figura 4-7 Modelo arquitectónico cliente – servidor.

4.3.4. Arquitectura física

La arquitectura física del sistema consiste en un sistema de sensores de oxígeno disuelto y temperatura (prototipo Internet de las cosas) que envía los datos de los estanques a través de bluetooth a un dispositivo móvil, en este dispositivo se recibirán los datos en una aplicación móvil en donde se procesarán y se enviarán a una aplicación web a través de internet para que dichos datos puedan ser visualizados por el usuario en cualquier dispositivo y navegador web.

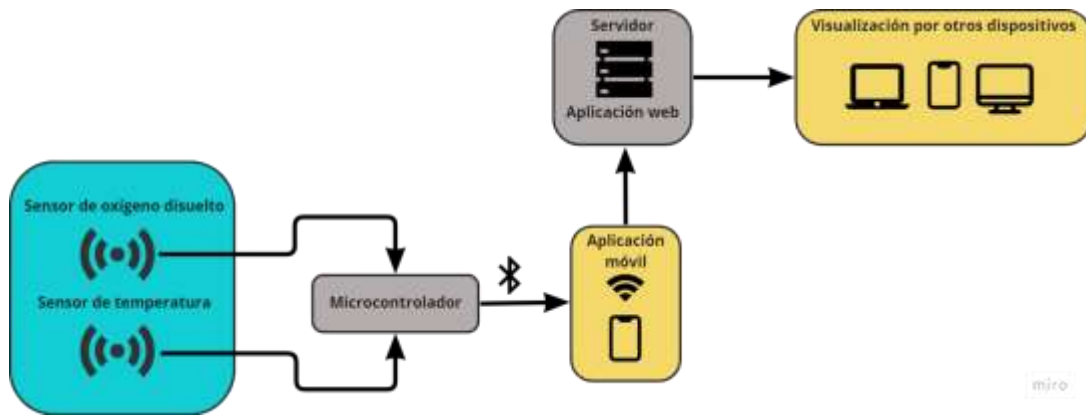


Figura 4-8. Arquitectura física

4.3.5. Diagrama de secuencia

En la Figura 4-9 se muestra el diagrama de secuencia que representa el comportamiento dinámico entre los módulos y actores del sistema, estos módulos se comunican mediante una secuencia de funciones que se irán ejecutando hasta cumplir con toda la funcionalidad del sistema. El diagrama de secuencia fue de gran utilidad para organizar el proceso y comportamiento del sistema, pero sobre todo para elegir las funciones que el sistema debe tener para cumplir con su propósito.

En el diagrama, la secuencia inicia cuando el dispositivo de Internet de las cosas envía el estado de los parámetros, en este caso oxígeno disuelto y temperatura a el módulo software que representa a la aplicación web, después mediante los módulos posteriores se irán ejecutando una serie de funciones hasta que la información es recibida o visualizada por usuario final.

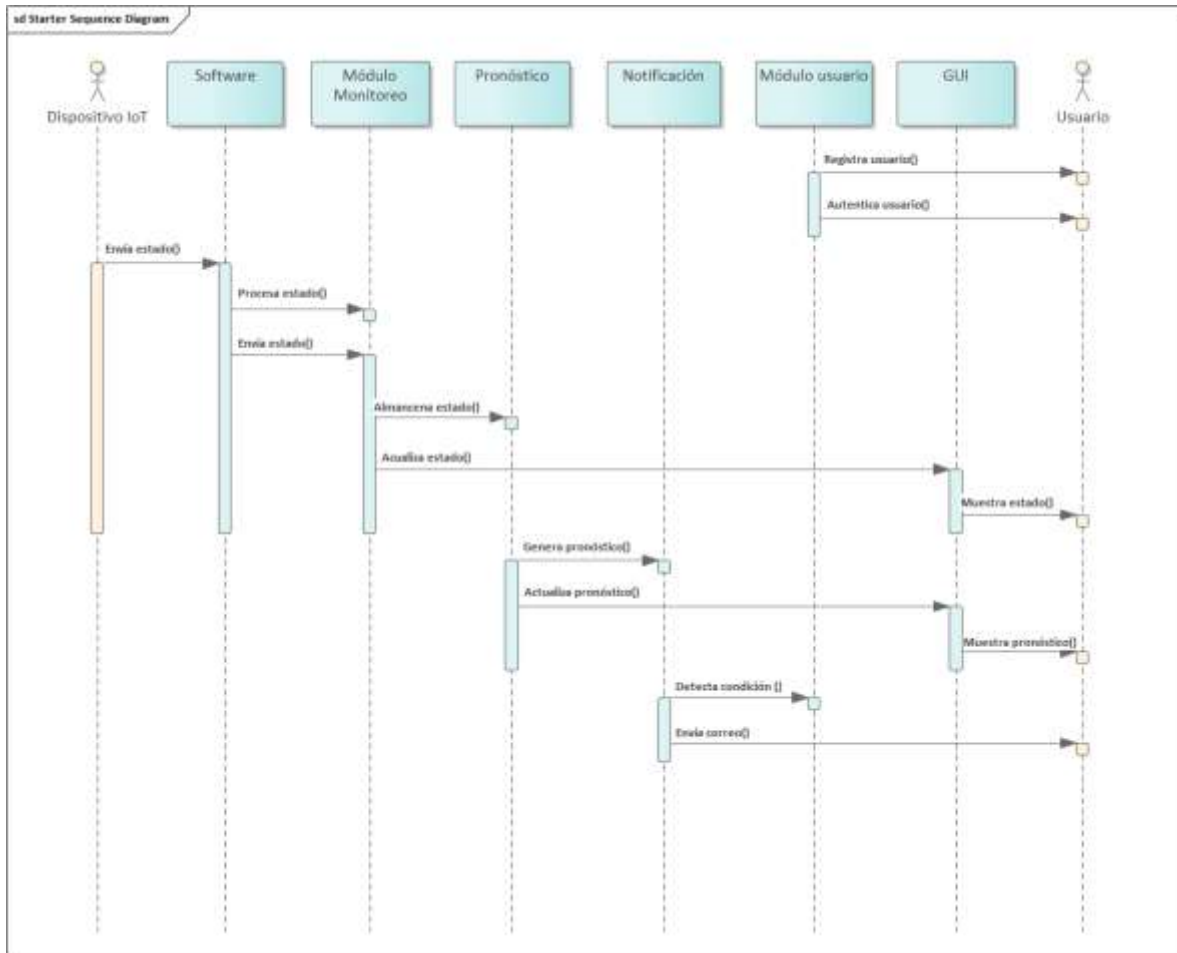


Figura 4-9. Diagrama de secuencia.

4.4. Codificación

En esta sección se describe la implementación y las herramientas que se utilizaron para el desarrollo de la aplicación móvil, el *back-end* y el *front-end* de la aplicación web y su despliegue.

4.4.1. Aplicación móvil

La aplicación móvil forma parte de la capa física del sistema, tal como se muestra en la Figura 4-6 Arquitectura lógica, esta aplicación representa el módulo software, y es el intermediario entre sistema de internet de las cosas y la aplicación web. Se desarrolló en Android Studio, en el lenguaje de programación Java (Oracle, s.f.) y con el apoyo de la librería *bluetooth-le-android-hm10* (Stann, 2018) para realizar la conexión con los dispositivos de BLE. En esta aplicación es donde se reciben los datos a través de BLE en el dispositivo móvil, se les agrega

la fecha y la hora a la que fueron recolectados y se envían los datos en formato *Json* a la aplicación web a través de peticiones HTTP. Esta aplicación también cuenta con la funcionalidad de almacenar los datos en un archivo de texto en caso de que se pierda la conexión a Internet, para poder recuperarlos y enviarlos a la aplicación web una vez que se recupere la conexión.

En la figura 4-8 se muestra cómo se reciben los datos de la temperatura y el oxígeno del dispositivo Internet de las cosas en formato *Json* en la aplicación móvil y la Figura 4-9 representa una muestra del formato *Json* que se envía a la aplicación web. La imagen muestra los datos que fueron almacenados en un archivo de texto cuando se perdió la conexión a Internet

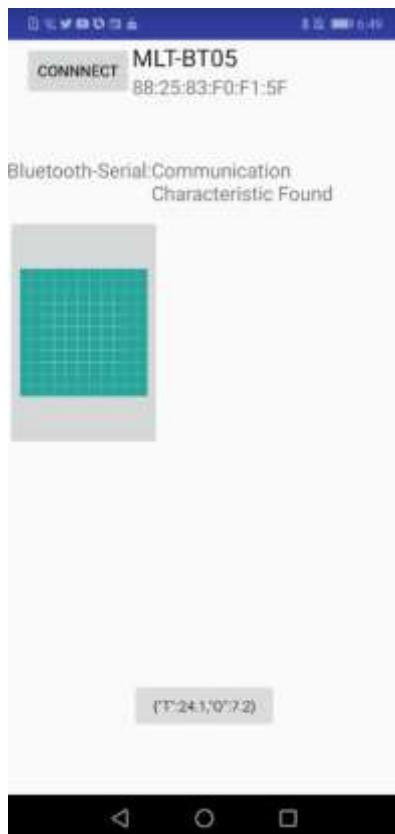


Figura 4-8 Visualización de los datos en la aplicación móvil.



```
roidManifest.xml × BleConnectionService.java × DataMonitoring.java × monitoreo.txt ×
{"T":24.3,"O":8.4,"F":"2022-01-23","H":"09:53 p. m."}
{"T":24.3,"O":8.4,"F":"2022-01-23","H":"09:54 p. m."}
{"T":24.4,"O":8.4,"F":"2022-01-23","H":"09:55 p. m."}
```

Figura 4-9 Archivo de texto que almacena los datos en caso de un fallo

4.4.2. Aplicación web

El *back-end* de la aplicación, es decir, el lado del servidor de la aplicación que representa la capa de dominio del sistema es dónde se administra la información a través de las otras capas para que no se pueda acceder directamente a los datos. Fue desarrollado en el lenguaje de programación Python y el *framework* Flask (Flask documentation, 2010).

La base de datos que representa la capa de datos se desarrolló en PostgreSQL (PostgreSQL 14.2 Documentation, 2022). En este paquete es donde se almacena la información tanto de los usuarios como de los sensores, mediante tablas diferentes para dividir dicha información, la base de datos fue basada en el modelo de dominio utilizando sus entidades como referencia. En la Figura 4-10 se muestra la composición de las tablas (el tipo de dato, las columnas y sus restricciones) y sus relaciones.

En este caso se tiene que la tabla predicción depende de los datos de la tabla parámetros y envía su información a los usuarios cuyos datos se encuentran registrados en la tabla usuarios, así mismo, la tabla parámetros envía su información a dichos usuarios y su información es utilizada para obtener los datos de la tabla predicción y finalmente la tabla usuarios que representa la información de los usuarios que se registran en el sistema para tener acceso a dicha información.

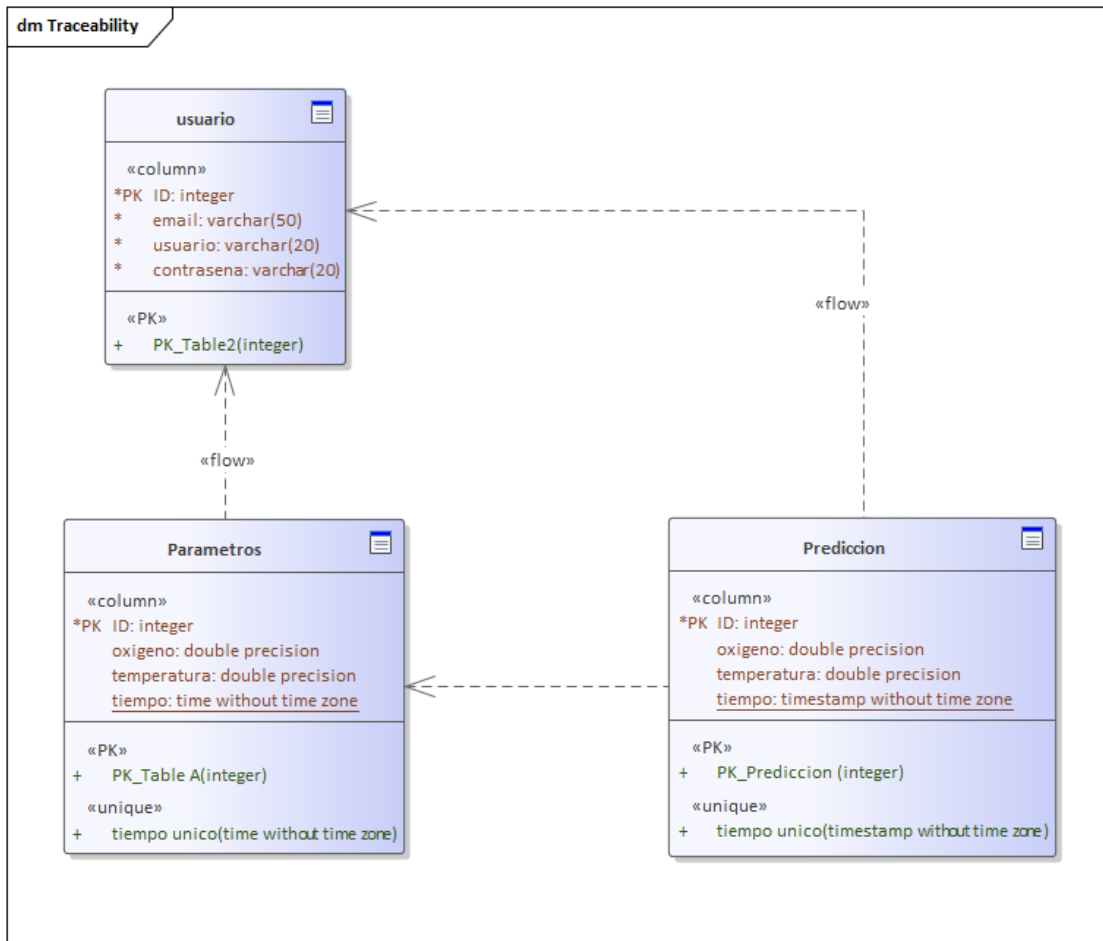


Figura 4-10 Modelo lógico de base de datos.

El *front-end* de la aplicación representa la capa de presentación del sistema, en este paquete se muestra la información a los usuarios de forma dinámica. Para su desarrollo se utilizó el lenguaje de programación JavaScript (MND, 2022), las librerías Bootstrap y Chart.js, HTML y CSS.

El despliegue tanto de aplicación web como de la base de datos fue desarrollado en un servidor virtual de *Heroku* (Center, 2022).

En la figura 4-11 se muestra la visualización de los datos en la aplicación web mediante una gráfica. La información mostrada representa los últimos 10 registros almacenados en la base de datos

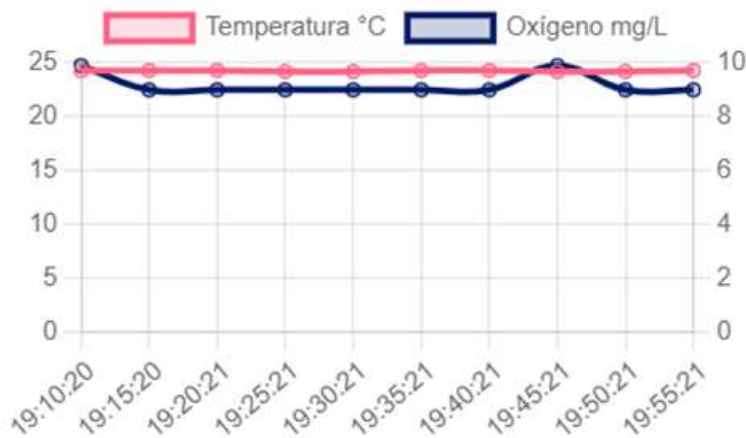


Figura 4-11 Visualización de los datos en la aplicación web.

La figura 4-10 muestra las alertas que son enviadas al correo electrónico de los usuarios registrados en el sistema en caso de que los parámetros se salgan de los rangos establecidos.



Figura 4-11 Notificación.

4.4.3. Modelo de pronóstico

Para la implementación del modelo de pronósticos se seleccionaron dos de los modelos de Aprendizaje maquina más utilizados para el pronóstico de series de tiempo para ponerlos a prueba. Los modelos seleccionados fueron las redes neuronales *Long-Short Term Memory* (LSTM) y las *Gated Recurrent Unit (GRU)*. Su implementación fue basada en el proyecto de Rasifaghihi (Rasifaghihi, 2020) en dónde se realizó una comparación de estos mismos modelos para el consumo diario de agua de la ciudad de Brossard, Quebec, Canadá. Se utilizó el modelo de AutoML (Medrano, 2021) para su configuración y debido los resultados de entrenamiento de ambos modelos se decidió tomar el modelo GRU para la implementación del pronóstico en la aplicación. La implementación se llevó a cabo en diferentes pasos, los cuales ser presentan en la Figura 4-12 y se explicaran a continuación:



Figura 4-12. Etapas de desarrollo del modelo de pronóstico.

4.4.3.1. Descripción del dataset

Inicialmente se planeó utilizar un *dataset* con datos reales para que fuera de apoyo en la implementación del modelo, pero no fue posible encontrar un dataset que se ajustara a los datos que se necesitaban. Por lo tanto, se optó por hacer una simulación de los datos para realizar el entrenamiento del modelo.

Para la simulación se tomaron en cuenta los valores máximos y mínimos del oxígeno disuelto y la temperatura, ya que durante la noche los valores de ambos parámetros tienden a bajar y durante la tarde tienden a subir. Mediante esta información se pudo concluir que el sistema de los estanques funciona de forma cíclica y se optó por utilizar una función seno para realizar la simulación de los datos.

La simulación se realizó para obtener los valores de oxígeno disuelto y temperatura en intervalos de 5 minutos durante 15 días, obteniendo un total de 4705 datos. El *dataset* fue simulado mediante una función seno representada en la ecuación 4.1.

$$y = (d)Sen\left(\frac{2\pi}{T}(t)\right) + \mu \quad (4-1)$$

Donde:

y : Representa el valor simulado para la temperatura o el oxígeno disuelto.

d : representa la diferencia entre los valores máximos y mínimos de la temperatura y el oxígeno disuelto.

T : representa el periodo de 24 horas representadas en segundos.

t : Representa el tiempo actual en segundos, el cual ira aumentando en intervalos de 5 minutos.

μ : Representa la media de los valores máximos y mínimos de la temperatura o el oxígeno disuelto.

En las Figuras 4-13 y 4-14 se muestra el resultado de los datos para la temperatura en °C y el oxígeno disuelto en ppm una vez simulados.

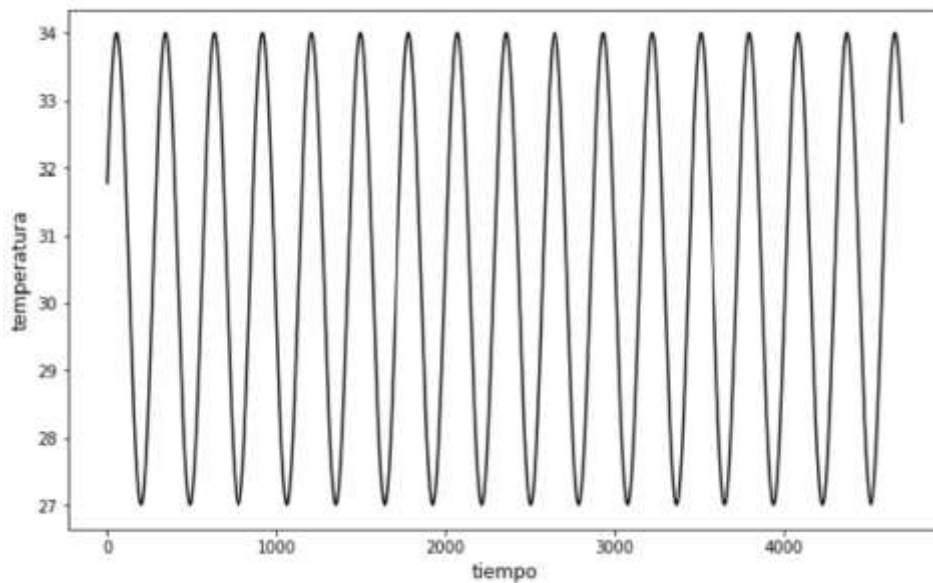


Figura 4-13. Datos simulados para la temperatura en °C.

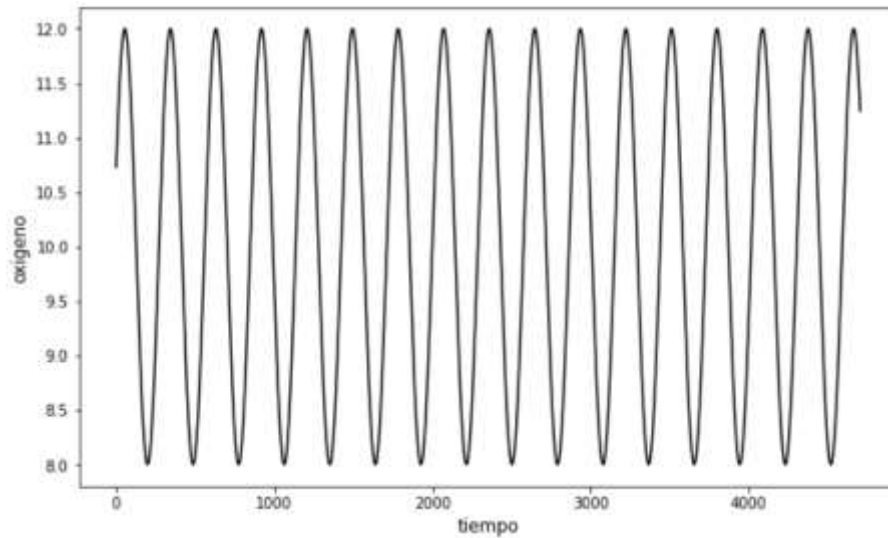


Figura 4-14. Datos simulados para el oxígeno disuelto en mg/L.

4.4.3.2. Preprocesamiento de datos

Durante esta etapa se llevó a cabo el análisis, limpieza y división de los datos. Los datos se dividieron en 80% para el entrenamiento y 20% para pruebas. A continuación, se presentan los pasos que se llevaron a cabo para desarrollar la etapa de preprocesamiento.

a) Reemplazar valores nulos o fuera de rangos

Una vez que se cargan los datos del dataset, los datos atípicos (es decir los valores en una región de baja probabilidad) se detectan bajo el concepto de máxima verosimilitud, representado en la ecuación 4.2. Mediante la cual se calculan los valores máximos y mínimos de la media y se determina cuales valores están fuera de los rangos. Los valores que son detectados como atípicos son reemplazados por valores nulos, para posteriormente ser rellenados por la función de extrapolación lineal que detecta los valores nulos y los reemplaza.

$$m = \mu \pm r\sigma \tag{4-2}$$

Donde:

m : Valor atípico para detectar.

$\mu \pm$: Media.

σ : Desviación estándar.

r : Rango para la generación de valores máximos y mínimos.

b) Normalización de los datos

Una vez que se llevó a cabo la limpieza de los datos se pasaron por un proceso de normalización representado en la ecuación 4-3. La normalización consiste en cambiar la escala original de los datos a un rango entre 0 y 1, esto debido a que los datos de entrada (oxígeno disuelto y temperatura) se encuentran en diferentes escalas y puede causar confusión al modelo de pronóstico, ya que estos modelos generalmente se guían por los valores más grandes. (Francesca Lazzeri, 2021)

$$\frac{x - \min}{\max - \min}$$

(4-3)

Donde:

x : Es cada uno de los valores del conjunto de datos.

\min : Es el valor mínimo de los datos.

\max : Es el valor máximo de los datos.

c) Dividir las variables objetivo y las variables dependientes.

En este caso las variables de entrada (dependientes) serán el oxígeno y temperatura en el pasado, mientras que las variables de salida (objetivo) serán las mismas, pero en el siguiente paso al futuro. Como se muestra en la figura 4-12, las entradas (x) son los valores de oxígeno y temperatura que van desde el primer valor del dataset hasta el largo del valor de la ventana, mientras que la salida (y) será el valor posterior a donde termina el valor de la ventana, formando así una matriz de tres dimensiones para el vector de entrada y una matriz de dos dimensiones para el vector de salida.

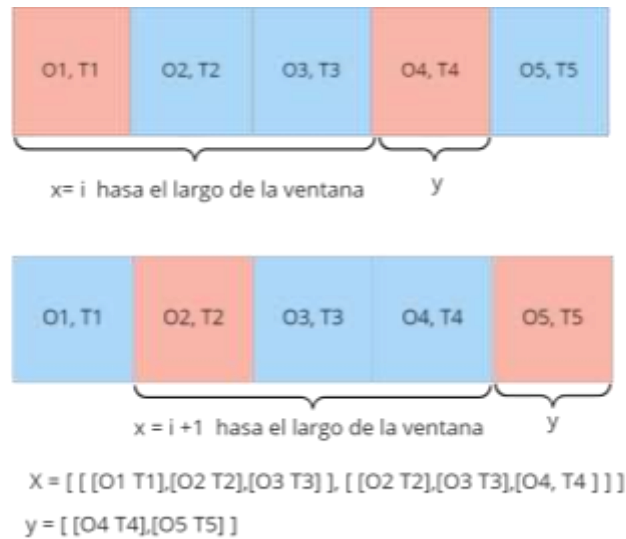


Figura 4-12 Vectores de entrada y salida.

4.4.3.3. Configuración del modelo

Para la configuración del modelo se utilizó el proyecto *Auto Machine Learning Optimizer Model Generation*, desarrollado por investigadores del Instituto Tecnológico de Culiacán. Este proyecto tiene como objetivo la automatización de arquitecturas e hiperparámetros de modelos de Aprendizaje profundo para la regresión y la clasificación mediante un algoritmo bayesiano (Medrano, 2021). Mediante este sistema se eligieron dos arquitecturas para los modelos GRU Y LSTM, de las cuales se eligió la que tuvo mejores resultados durante las pruebas, en este caso el modelo GRU fue el modelo con mejores resultados y consiste en dos capas ocultas y una de salida con una función de activación tangente hiperbólica. La primera capa oculta contiene 144 neuronas, la segunda capa oculta contiene 64 y la capa de salida contiene 2 neuronas.

4.4.3.4. Entrenamiento

El modelo fue entrenado con datos de entrenamiento para 30 épocas y con un tamaño de lote de 32. Se usa el 20 % de los datos de entrenamiento como datos de validación. Se configuro una parada temprana (early stop) que detiene el entrenamiento cuando el valor de perdida no mejora después de 10 épocas, esto con la finalidad de evitar el sobreajuste.

4.4.3.5. Implementación del pronóstico a múltiples pasos

Una vez que el modelo fue entrenando y se obtuvieron los resultados de la predicción, se eligió el modelo con los mejores resultados y se implementó en la aplicación web para múltiples pasos al futuro. Para esto se decidió usar el método recursivo, el cual consiste en realizar el pronóstico a un solo paso y agregar ese pronóstico al vector de entrada para obtener la siguiente salida y así hasta completar todas las iteraciones de los pasos a futuro. Una vez que se obtienen todos los pronósticos, los datos son graficados en la interfaz gráfica de usuario.

En este capítulo se presentó el desarrollo del sistema, el cual fue la parte más importante del proyecto ya que se describe como se realizó el proyecto en cada una de las etapas desde que se inició con la investigación del estado del arte, hasta concluir la implementación. El sistema consiste en varios procesos que van desde la implementación física que es la parte en donde la aplicación móvil recibe los datos de los sensores por BLE, luego, dichos datos son enviados a una aplicación móvil con la finalidad de informar al usuario desde cualquier dispositivo sobre el estado actual de los estanques y finalmente una vez que son obtenidos los datos actuales se realiza un pronóstico por medio de un modelo de Aprendizaje máquina, que permite informar al usuario sobre el futuro comportamiento de los estanques. Además de contar con un registro de usuarios que permite almacenar la información de los trabajadores de los laboratorios o granjas, que además de hacer que los usuarios registrados reciban alertas a su correo electrónico, puede ser utilizado para futuras tareas de las empresas.

Capítulo 5

5. Pruebas

En este capítulo se muestran los experimentos y los resultados obtenidos, con la finalidad de comprobar que la aplicación web funciona de forma correcta, así como la comparación de diferentes modelos y arquitecturas de Aprendizaje máquina con la finalidad de escoger el más adecuado para realizar un pronóstico con los datos obtenidos de la simulación.

En la sección 5.1 se mencionan los experimentos y resultados obtenidos correspondientes a la funcionalidad de la aplicación web, mientras que en el apartado 5.2 se mencionan los experimentos y resultados obtenidos con respecto al modelo de pronósticos.

5.1. Sistema de monitoreo

Para probar la funcionalidad de la aplicación web se realizó una visita al laboratorio post-larva "Oleajes" ubicado en Sinaloa, México mostrado en la Figura 5-1. Para este experimento se utilizó el prototipo de Internet de las cosas realizado por un estudiante de la maestría en Ciencias de la Computación del Instituto Tecnológico de Culiacán. El prototipo consiste en un sensor de oxígeno disuelto Atlas (Dissolved Oxygen Probe) y otro de temperatura (DS18B20), ambos sumergibles en agua salada, un microcontrolador Arduino UNO y un módulo BLE hm-10. Como se muestra en la Figura 5-2, los sensores fueron sumergidos en el recipiente y agua utilizados en el transporte de camarón y se comprobó lo siguiente:

- La aplicación móvil recibe y envía la información de los sensores en el formato correcto a la aplicación web. Además, es capaz de almacenar los datos cuando se pierde la conexión a internet, con esta función la aplicación asegura la tolerancia a la partición de datos y hace que la aplicación sea confiable.
- La aplicación web muestra los datos en la GUI, envía notificaciones para alertar a los usuarios que están registrados con su correo electrónico y almacena los datos en la base de datos.
- Los usuarios pueden registrarse e iniciar sesión en la aplicación y su información se almacena en la base de datos.

- El diseño de la arquitectura lógica hace que la aplicación sea más segura y mantenible, ya que está dividida en paquetes y módulos, de esta manera no es posible acceder directamente a los datos y se pueden hacer modificaciones, sin afectar a las otras funciones.
- La aplicación web pudo ser accedida mediante dispositivos móviles como en computadoras personales.



Figura 5-1 Laboratorio de postlarva Oleajes.



Figura 5-2 Sensores.

5.2. Modelo de pronóstico

Para determinar cuál sería el mejor modelo de pronóstico, se eligieron dos de las arquitecturas de Aprendizaje máquina más utilizadas en el estado del arte, en este caso una red neuronal LSTM y una GRU. Al ser arquitecturas muy similares, se realizaron algunos experimentos, donde se configuraron algunos hiperparámetros y se evaluaron sus resultados para elegir el modelo que mejor se ajustara a los datos. El entrenamiento y pronóstico de los modelos se ejecutaron 5 veces y se calculo un promedio, esto con la finalidad de obtener datos más exactos sobre los resultados de los modelos, a continuación, se presenta la descripción de cada arquitectura, así como los resultados de las evaluaciones de los pronósticos utilizando MAE y RSME.

- **Arquitectura 1:** Esta arquitectura se implementó para los modelos tanto de LSTM como GRU y consiste en 2 capas ocultas con 192 neuronas en la primera capa oculta, 64 en la segunda y 2 en la capa de salida y se utilizó una función de activación tangente hiperbólica. Los modelos fueron evaluados mediante el promedio del Error Cuadrático Medio (MAE) y la Raíz del Error Cuadrático Medio, tal como se muestra en las Tablas 4-4 y 4-5.

Tabla 4-4. Resultados LSTM Arquitectura 1.

LSTM (Arquitectura 1)				
	Temperatura		Oxígeno	
	MAE	RMSE	MAE	RMSE
	0.031783366	0.039293592	0.033094046	0.035512718
	0.188407311	0.24532478	0.134771739	0.1496189
	0.029526033	0.033593124	0.014823566	0.017723201
	0.013048951	0.114589811	0.049886413	0.062975091
	0.009250526	0.011345408	0.003826159	0.005502717
PROMEDIO	0.054403237	0.088829343	0.047280384	0.054266525

Tabla 4-5. Resultados GRU Arquitectura 1.

GRU (Arquitectura 1)				
	Temperatura		Oxígeno	
	MAE	RMSE	MAE	RMSE
	0.105585771	0.112444039	0.049218504	0.054039518
	0.055955576	0.06173047	0.031969405	0.03502139
	0.080795059	0.095419538	0.063331921	0.080929572
	0.029919019	0.033592899	0.014640011	0.016336188
	0.018562104	0.023996491	0.014237993	0.017080482
PROMEDIO	0.058163506	0.065436688	0.034679567	0.04068143

De la Figura 5-3 a la 5-6 se muestra la comparación de resultados de la predicción vs el valor real para los modelos LSTM y GRU utilizando la arquitectura 1.

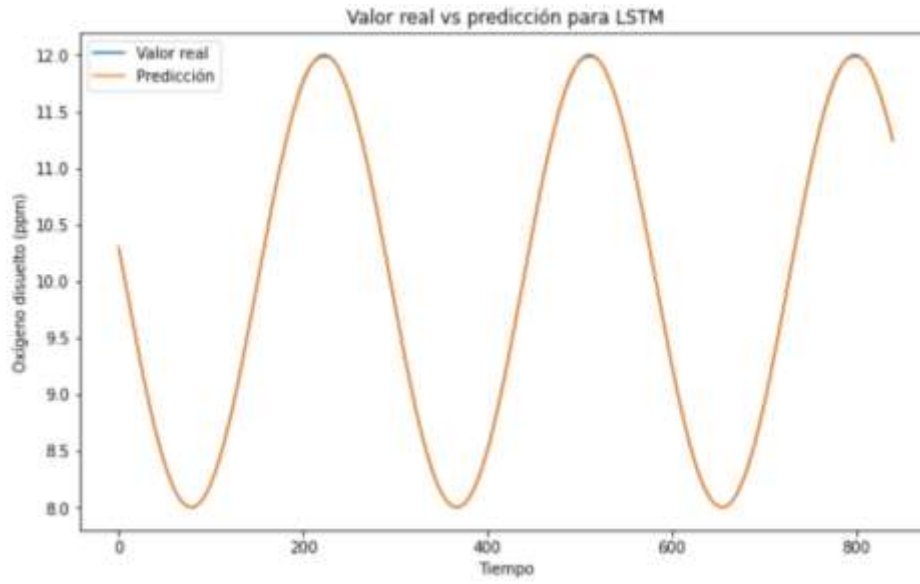


Figura 5-3 Resultados oxígeno disuelto LSTM Arquitectura 1.

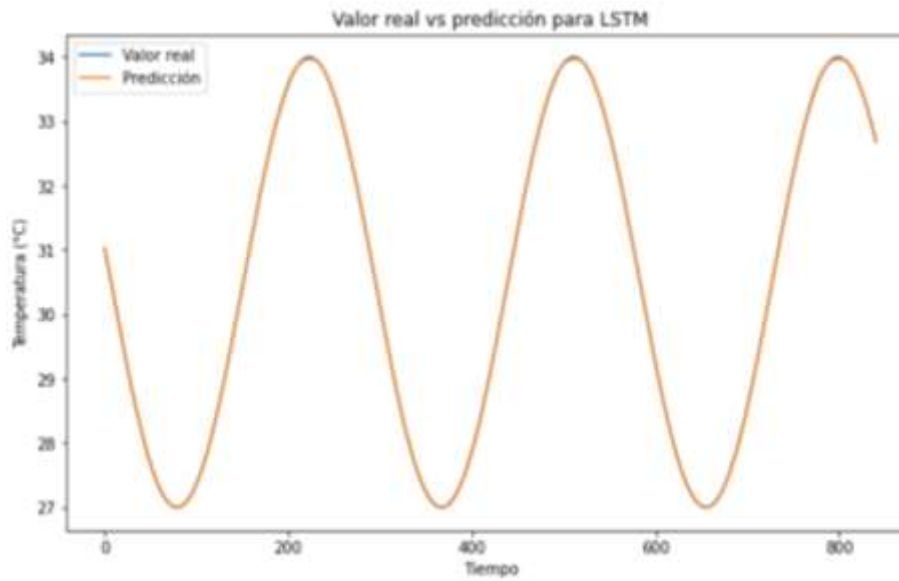


Figura 5-4 Resultados temperatura LSTM Arquitectura 1.

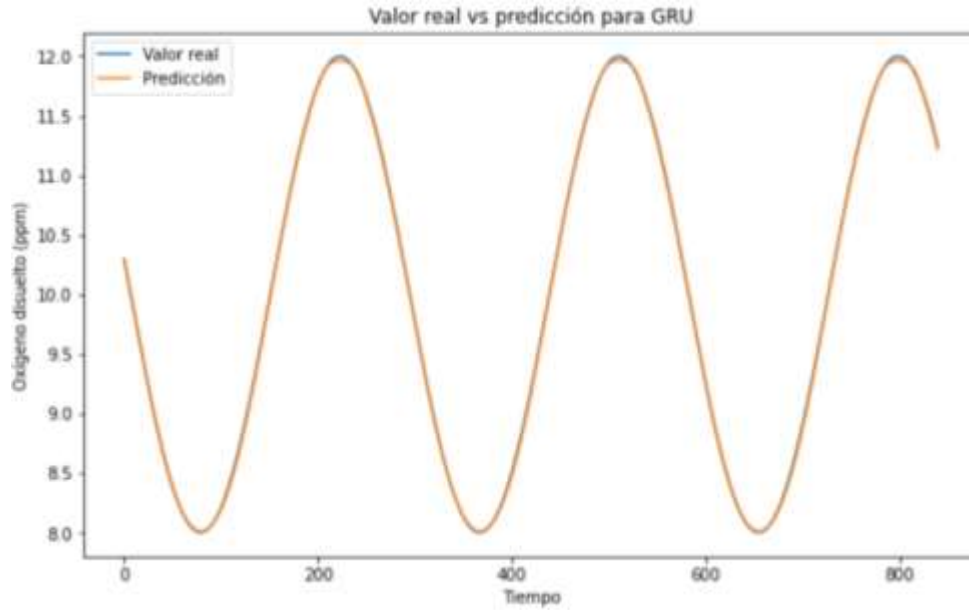


Figura 5-5 Resultados oxígeno disuelto GRU Arquitectura 1.

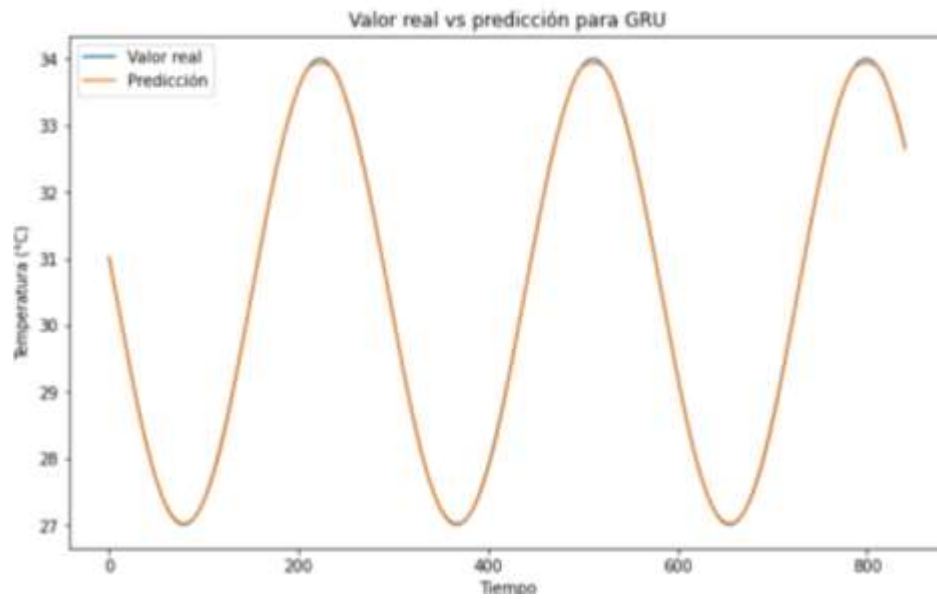


Figura 5-6 Resultados oxígeno disuelto GRU Arquitectura 1.

Las Figuras 5-7 y 5-8, muestran el valor de la precisión de los modelos GRU y LSTM con la arquitectura 1 para los datos de entrenamiento y validación, se puede observar

que en el entrenamiento obtuvieron valores muy similares, a diferencia de la etapa de validación en donde el modelo GRU fue el que mantuvo valores de precisión más altos, por lo tanto, fue el que en promedio tuvo mejores predicciones.

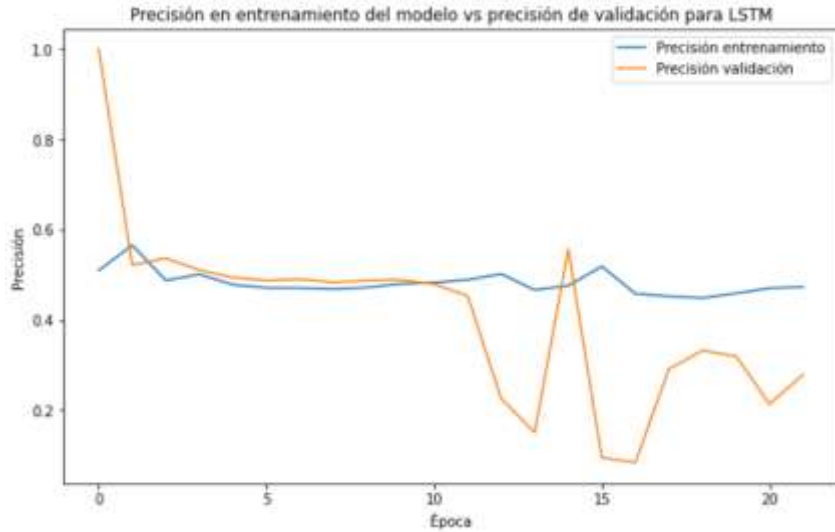


Figura 5-7 Precisión Arquitectura 1, modelo LSTM.

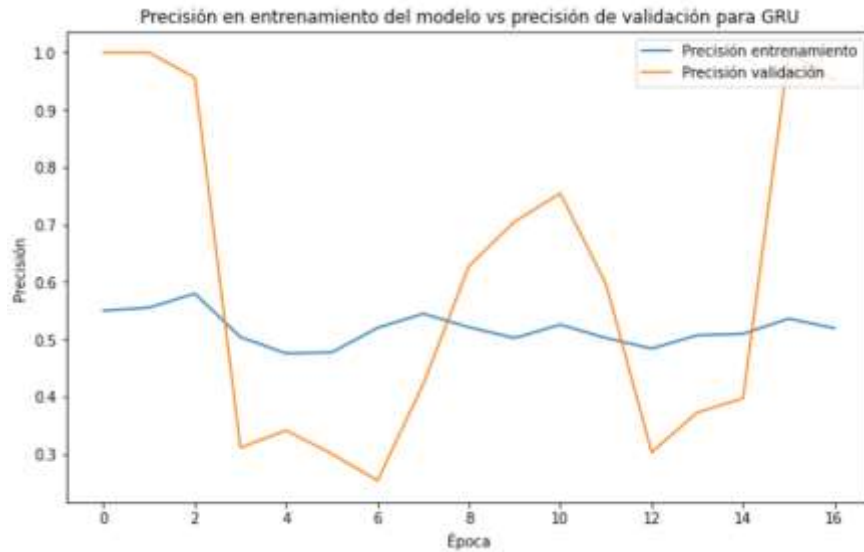


Figura 5-8 Precisión Arquitectura 1, modelo GRU.

Las Figuras 5-9 y 5-10, muestran el valor de pérdida (MSE) de los modelos GRU y LSTM con la arquitectura 1, para los datos de entrenamiento y validación, se puede

observar que el modelo GRU fue el que se mantuvo los resultados de pérdida más bajos en ambas etapas, tanto de validación como de entrenamiento.

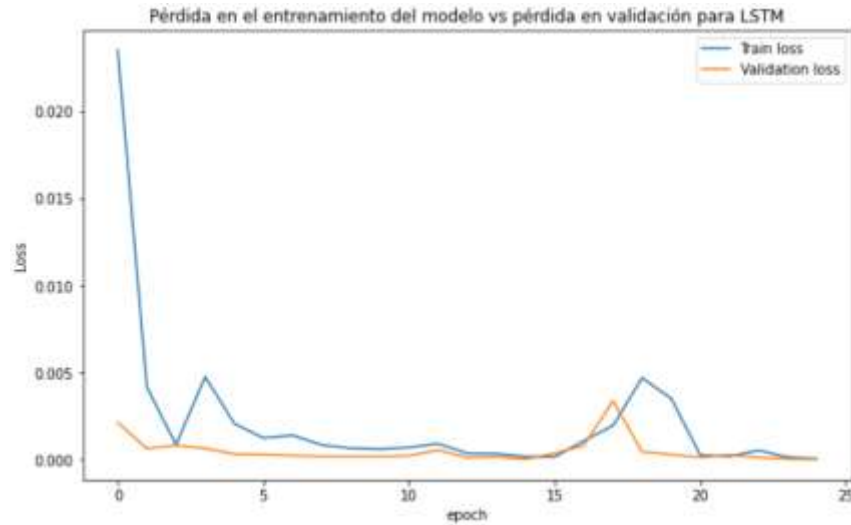


Figura 5-9 Pérdida Arquitectura 1, modelo LSTM.

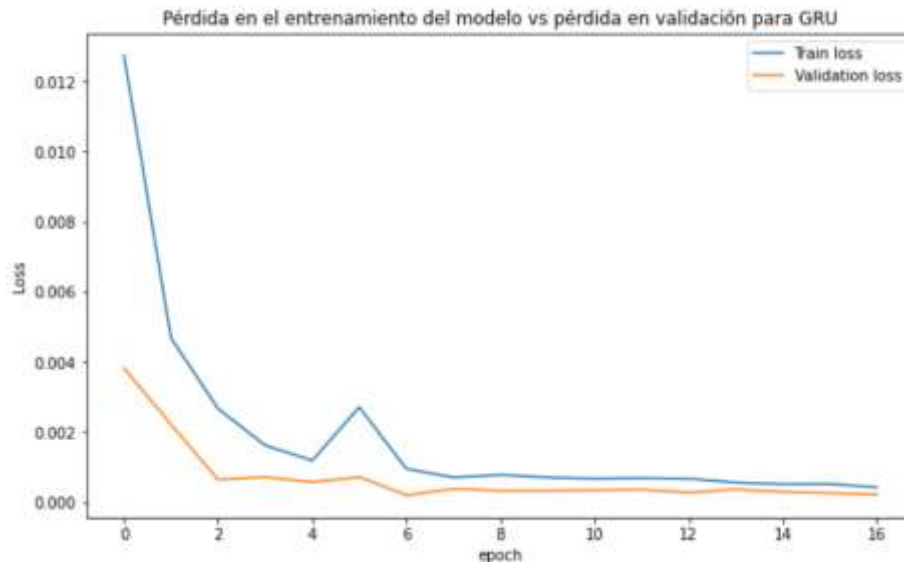


Figura 5-10 pérdida Arquitectura 1, modelo GRU.

- **Arquitectura 2:** al igual que en la arquitectura 1, se implementó para los modelos tanto de LSTM como GRU y consiste en 2 capas ocultas con 144 neuronas en la

primera capa oculta, 32 en la segunda y 2 en la capa de salida y se utilizó una función de activación tangente hiperbólica. Los modelos fueron evaluados mediante el promedio del Error Cuadrático Medio (MAE) y la Raíz del Error Cuadrático Medio tal como se muestra en las Tablas 4-6 y 4-7.

Tabla 4-6. Resultados LSTM Arquitectura 2.

LSTM (Arquitectura 2)				
	Temperatura		oxígeno	
	MAE	RMSE	MAE	RMSE
	0.024948464	0.031808297	0.015199578	0.018835474
	0.034842033	0.045747289	0.017799295	0.022978902
	0.144417697	0.15341665	0.090915921	0.094798463
	0.06249995	0.073523287	0.032276737	0.039631198
	0.029482446	0.037084106	0.022223343	0.023585932
PROMEDIO	0.059238118	0.068315926	0.035682975	0.039965994

Tabla 4-7. Resultados GRU Arquitectura 2.

GRU (Arquitectura 2)				
	Temperatura		oxígeno	
	MAE	RMSE	MAE	RMSE
	0.0145933	0.019579348	0.007855241	0.010354399
	0.014174103	0.019053042	0.008972741	0.011240122
	0.019308248	0.022427285	0.011331692	0.013048951
	0.195148997	0.239510881	0.121162155	0.149647426
	0.051568374	0.060495973	0.029037052	0.034242207
PROMEDIO	0.058958604	0.072213306	0.035671776	0.043706621

De la Figura 5-11 a la 5-14 se muestra la comparación de resultados de la predicción vs el valor real para los modelos LSTM y GRU utilizando la arquitectura 2

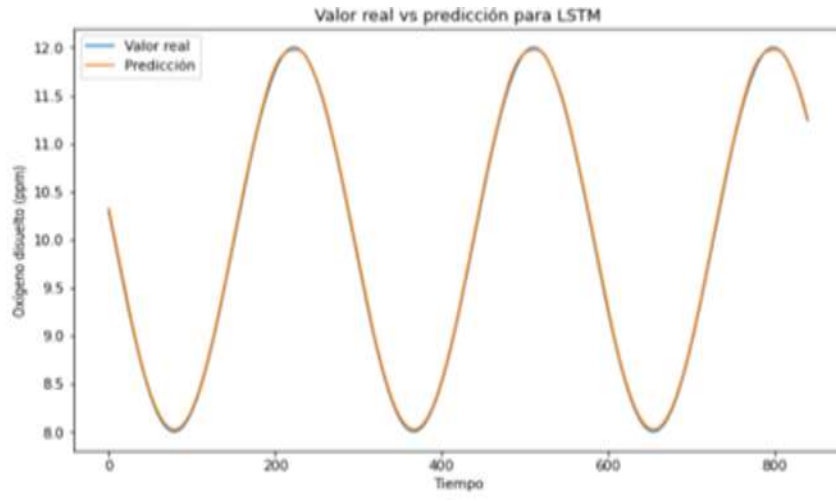


Figura 5-11 Resultados oxígeno disuelto LSTM arquitectura 2.

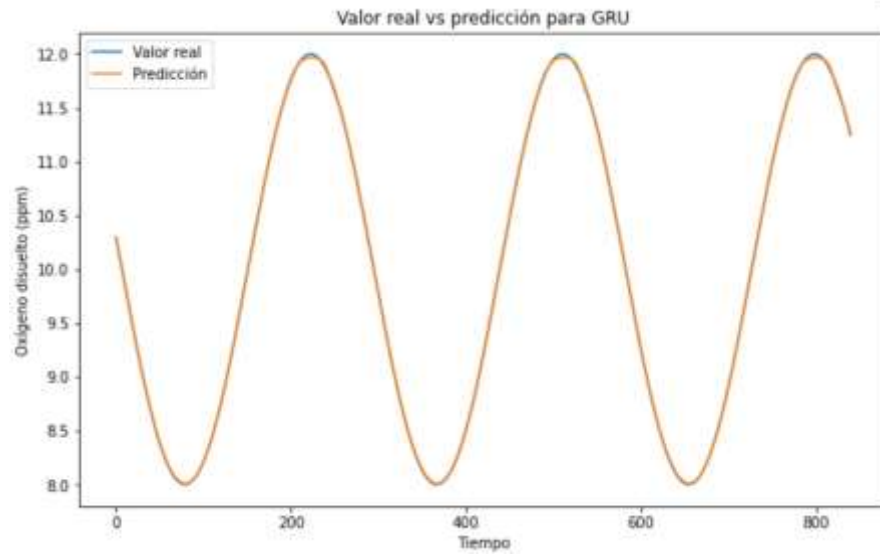


Figura 5-12 Resultados oxígeno disuelto LSTM arquitectura 2.

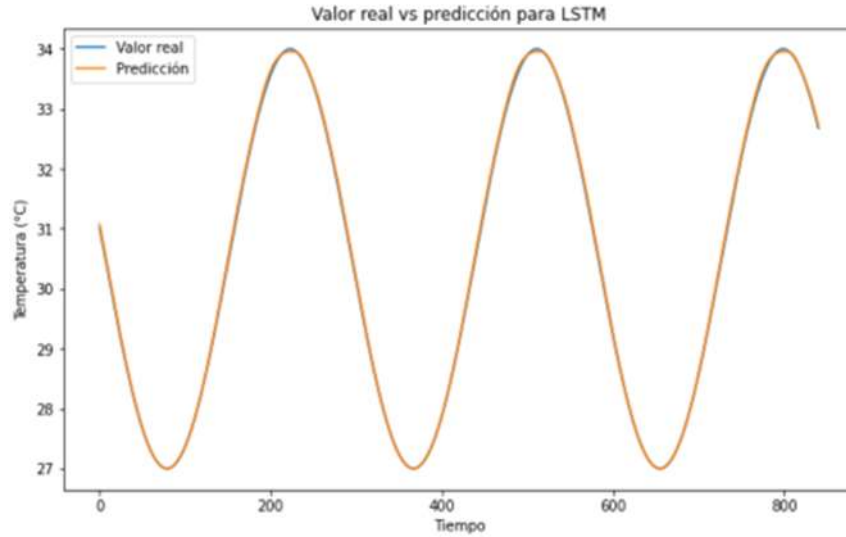


Figura 5-13 Resultados temperatura LSTM arquitectura 2.

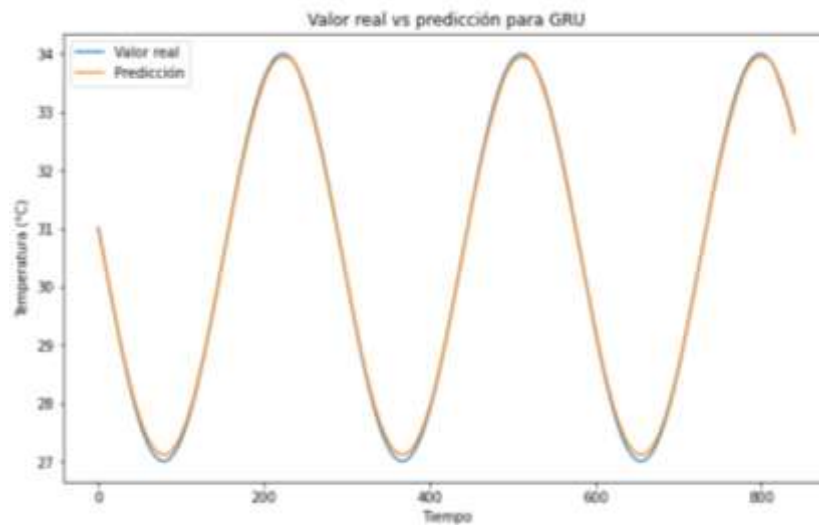


Figura 5-14 Resultados temperatura GRU arquitectura 2.

Las Figuras 5-15 y 5-16, muestran el valor de la precisión de los modelos GRU y LSTM con la arquitectura 1 para los datos de entrenamiento y validación, se puede observar que el modelo LSTM fue el que mantuvo los resultados más altos de precisión desde la época 4 hasta la 12, y el modelo GRU arroja valores más altos en la época 12 y 18.

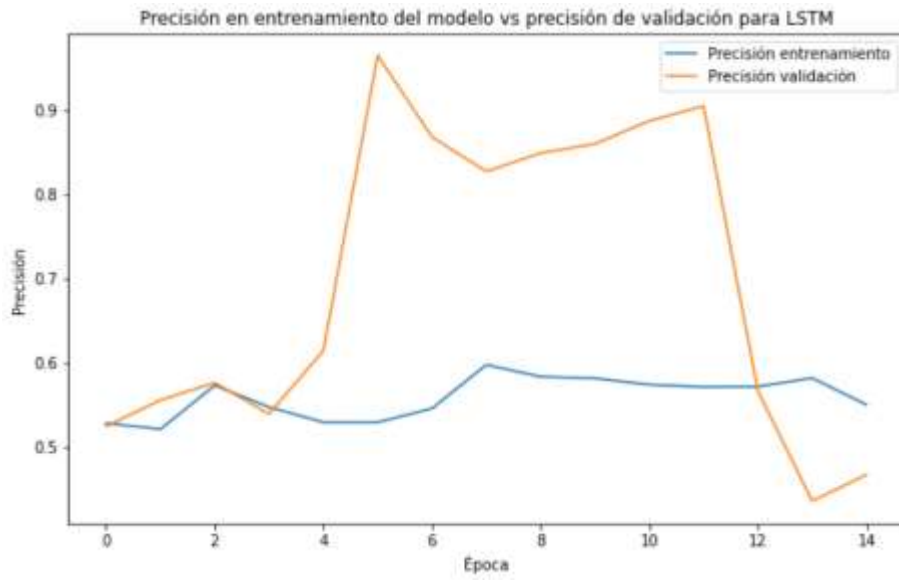


Figura 5-15 Precisión Arquitectura 2, modelo LSTM.

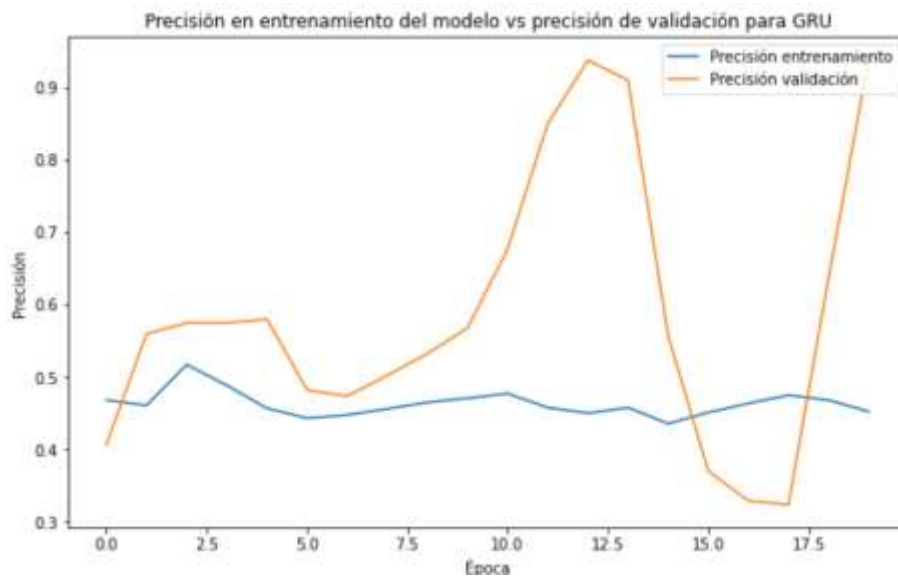


Figura 5-16 Precisión Arquitectura 2, modelo GRU.

Las Figuras 5-17 y 5-18, muestran el valor de pérdida (MSE) de los modelos GRU y LSTM con la arquitectura 2 para los datos de entrenamiento y validación, se puede observar que el modelo LSTM fue el que mantuvo los resultados de pérdida más bajos en la etapa de validación, mientras que el modelo GRU obtuvo los valores más bajos en la etapa de entrenamiento.

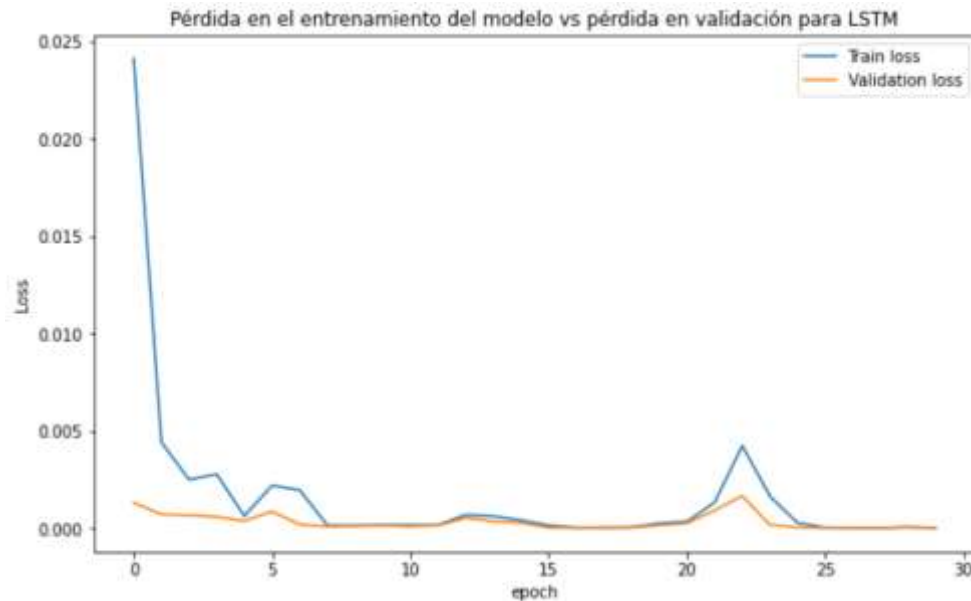


Figura 5-17 Precisión Arquitectura 2, modelo LSTM

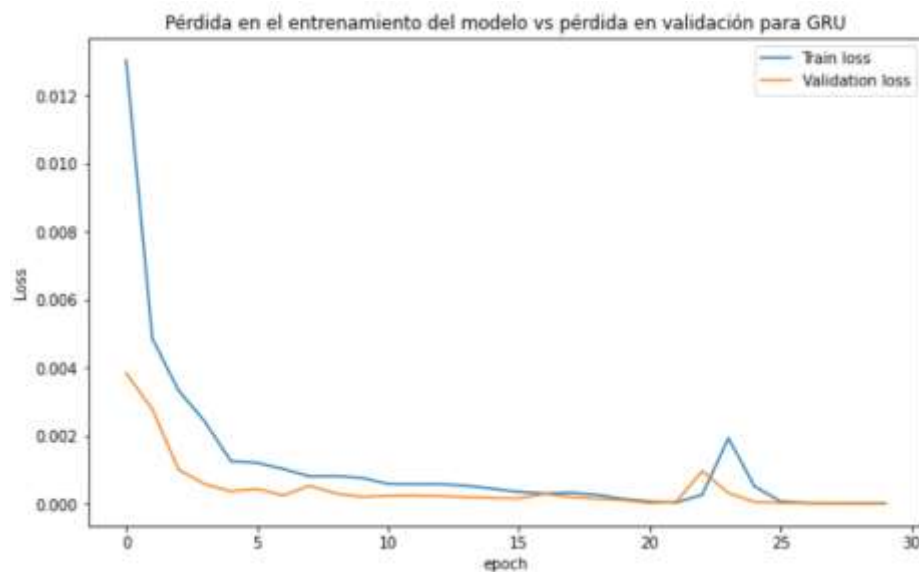


Figura 5-18 Precisión Arquitectura 2, modelo GRU.

En este capítulo se llevó a cabo una serie de experimentos que fueron de mucha utilidad para analizar si la aplicación cumple con los requerimientos funciones y de calidad definidos en la sección 4.2.2 y 4.2.3, así como para elegir la arquitectura del modelo pronósticos que más se adecua a los datos. Las pruebas del modelo de Aprendizaje máquina se realizaron con datos simulados y como trabajo a futuro se recomienda continuar la recopilación de datos reales en campo para mejorar dicho modelo.

Capítulo 6

6. Conclusiones y trabajo futuro

De acuerdo con los resultados de las pruebas presentadas sobre el modelo de pronóstico en el capítulo 5, la arquitectura 1 para el modelo GRU fue la que obtuvo resultados más bajos de error. Obteniendo un promedio de 0.065436688 en el RSME para la temperatura y un promedio de 0.034679567 de MSE para el oxígeno disuelto, siendo el modelo que obtuvo promedios más bajos de RSME y MSE en alguno de los dos parámetros. A pesar de que en valores de precisión las arquitecturas de ambos modelos se comportaron muy similares, en los valores de pérdida el modelo GRU de la arquitectura 1 fue el que tuvo mejores resultados.

También se puede concluir que, comparando ambos modelos de ambas arquitecturas, la arquitectura 1 fue la que obtuvo mejores resultados de temperatura y oxígeno, tanto para el modelo GRU como para el modelo LSTM. De acuerdo con los resultados obtenidos se decidió implementar dicha arquitectura con el modelo GRU en la aplicación web.

Cabe mencionar que estos modelos fueron entrenados con datos simulados, como trabajo a futuro se plantea utilizar las aplicaciones para recolectar datos en campo y así poder reentrenar dichos modelos para poder obtener resultados basados en los datos reales. Sin embargo, realizar estos experimentos por medio de una simulación permite tener implementados posibles modelos que tienen altas posibilidades de funcionar con datos reales.

El combinar estos sistemas de Internet de las cosas para el monitoreo y el pronóstico de datos con modelos de Aprendizaje máquina, se pueden crear herramientas muy poderosas, ya que permite generar nueva información con los datos obtenidos y así anticiparnos a posibles eventos, predecir riesgos y generar acciones que mitiguen esos riesgos, no solo en la industria acuícola si no también en otras industrias de diferentes sectores.

El sistema fue implementado para transportar el producto de un laboratorio a una granja acuícola, pero fue diseñado de tal manera que puede funcionar para resolver diferentes problemas en el sector de la acuicultura. En este caso podría ser para monitorear parámetros

de calidad del agua en laboratorios de post larva o en granjas acuícolas, sólo cambiando la interfaz de comunicación a una con mayor alcance, como podría ser de radio o las físicas.

También puede ser adaptado para añadir otros indicadores de calidad como el pH, la turbidez, entre otros, dependiendo de las necesidades de los usuarios. Por ello, el sistema también podría ayudar a resolver este problema en la industria de tratamiento de aguas, como plantas de tratamiento de aguas residuales, plantas de tratamiento de agua, plantas desalinizadoras, etc. Dónde también es muy importante conocer los indicadores de calidad del agua.

Bibliografía

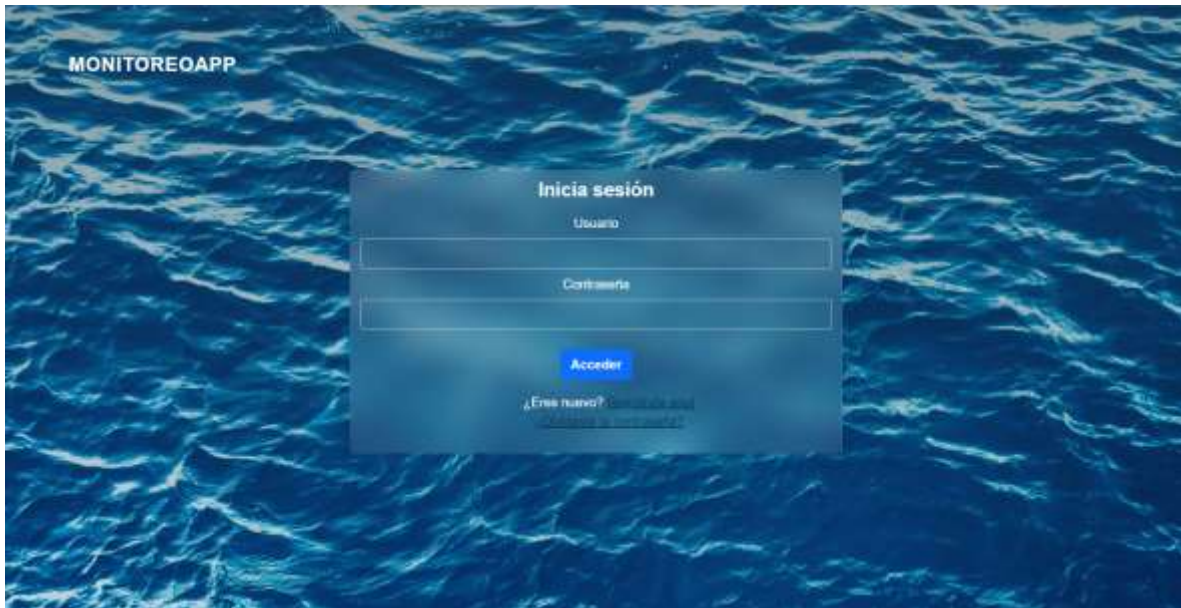
- ABI, R. (2022). *Bluetooth*. Obtenido de <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>
- Abinaya T, I. J. (2019). A Novel Methodology for Monitoring and Controlling of Water Quality in Aquaculture using Internet of Things (IoT). *International Conference on Computer Communication and Informatics*.
- Almuammar Manal, F. M. (2019). Deep Learning for Non-stationary Multivariate Time Series Forecasting. *IEEE International Conference on Big Data (Big Data)*.
- Aslam Muhammad, S. K.-M. (2019). Long-term Solar Radiation Forecasting using a Deep Learning Approach-GRUs. *IEEE 8th International Conference on Advanced Power System Automation and Protection (APAP)*.
- Baek Sang-Soo, P. J. (2020). Prediction of Water Level and Water Quality Using a CNN-LSTM Combined Deep Learning Approach. *MDPI*.
- Britannica, E. o. (6 de enero de 2022). *Britannica*. Obtenido de Enciclopedia Britannica: <https://www.britannica.com/technology/Ethernet>
- Britannica, E. o. (3 de febrero de 2022). *Britannica*. Obtenido de <https://www.britannica.com/technology/Wi-Fi>
- Carbajal Hernández, J. J., & Sánchez Fernández, L. P. (2013). Diagnóstico y predicción del hábitat en la camaronicultura. *Red de Revistas Científicas de América Latina, el Caribe, España y Portugal*.
- Center, H. D. (2022). *What is Heroku?* Obtenido de <https://www.heroku.com/what>
- Encinas Cesar, R. E. (2017). Design and implementation of a distributed IoT system for the monitoring of water quality in aquaculture. *Wireless Telecommunications Symposium*, 1-7.
- Engineers, T. I. (Septiembre de 1990). IEEE Standard Glossary of Software Engineering Terminology.
- FAO. (2017). www.fao.org/. Obtenido de <https://www.fao.org/aquaculture/es/>
- Flask documentation*. (2010). Obtenido de <https://flask.palletsprojects.com/en/2.1.x/>
- Foundation., P. S. (26 de abril de 2022). *Python 3.10.4 documentation*. Obtenido de <https://docs.python.org/3/>
- Francesca Lazzeri, P. (2021). *Machine Learning for Time Series Forecasting with Python*. Indianapolis, Indiana: John Wiley & Sons, Inc.

- Garlan, D., & Shaw, M. (1993). An Introduction to Software Architecture. *Advances in Software Engineering and Knowledge Engineering*.
- Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. United States of America: O'Reilly Media, Inc.
- Josh Patterson, A. G. (2017). *Deep Learning A Practitioner's Approach*. Sebastopol, CA: O'Reilly Media, Inc.
- Medrano, M. (2021). *Construcción de un modelo automático de regresión y clasificación en un ambiente distribuido*. Culiacán Sinaloa: Instituto Tecnológico de Culiacán.
- MND. (abril de 27 de 2022). *JavaScript*. Obtenido de <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- Muthu Ramya.C, M. R. (2011). STUDY ON ZIGBEE TECHNOLOGY. *International Conference on Electronics Computer Technology*.
- N.C. Ferreira, C. B. (2011). Hydrological and Water Quality Indices as management tools in marine shrimp culture. *Elsevier*.
- Negnevitsky, M. (2005). *Artificial Intelligence A Guide to Intelligent Systems*. England: Addison Wesley.
- Oluwatosin, H. S. (2014). Client-Server Model . *IOSR Journal of Computer Engineering*.
- Pérez, M., Mendoza, M. A., & Suarez, M. J. (2019). Paradigma IoT: desde su conceptualización hacia su aplicación en la agricultura. *Espacios*.
- Pesquera, S. d. (diciembre de 2013). *www.gob.mx*. Obtenido de <https://www.gob.mx/siap/prensa/sinaloa-lider-en-la-produccion-de-camaron-y-atun?idiom=es>
- PostgreSQL 14.2 Documentation*. (10 de febrero de 2022). Obtenido de <https://www.postgresql.org/docs/current/>
- Raschka, S. (2018). Model Evaluation, Model Selection, and Algorithm. *arXiv preprint arXiv:1811.12808*.
- Rasifaghihi, N. (30 de Agosto de 2020). *Towards Data Science*. Obtenido de Predictive Analytics: Time-Series Forecasting with GRU and BiLSTM in TensorFlow: <https://towardsdatascience.com/predictive-analytics-time-series-forecasting-with-gru-and-bilstm-in-tensorflow-87588c852915>
- Rodriguez Hector, M. M. (2020). Multi-step forecasting strategies for wind speed time series. *2020 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*. Ixtapa, Mexico.
- Sachin Kumar, P. T. (2019). Internet of Things is a revolutionary approach for future technology enhancement: a review. *Jurnal of Big Data*.
- Soler, J. P. (2014). *DISEÑO Y DESARROLLO WEB*. Universidad Politecnica de Valencia.

- Stann, H. (12 de noviembre de 2018). *GitHub*. Obtenido de <https://github.com/henry-stann/bluetooth-le-android-hm10>
- Summers, K. (2020). *Water Quality: Science, Assessments and Policy*. . IntechOpen.
- Trejo-Igueravide, R. (2016). Evaluación de los cultivos acuícolas en Tamaulipas, México. *CienciaUAT*.
- Xue Li, J. S.-I. (2016). A comparative study of multiple linear regression, artificial neural network and support vector machine for the prediction of dissolved oxygen. *Hydrology Research*.
- Zambada Juan, Q. R. (2015). An IoT based scholar bus monitoring system. *2015 IEEE First International Smart Cities Conference (ISC2)*.

Anexos

A. Interfaces graficas del sistema



Inicio de sesión de usuario.



Resumen del estado y el pronóstico de los parámetros de calidad

Estado del estanque

Estado del oxígeno disuelto y temperatura del estanque.



Grafica del estado de los parámetros de calidad obtenido de los sensores

Pronóstico

Pronóstico del oxígeno disuelto y temperatura del estanque.



Grafica del pronóstico de los parámetros de calidad

MONITOREOAPP

Crear cuenta

Nombre

Correo electrónico

Usuario

Contraseña

Registrarse

Formulario de registro de usuarios

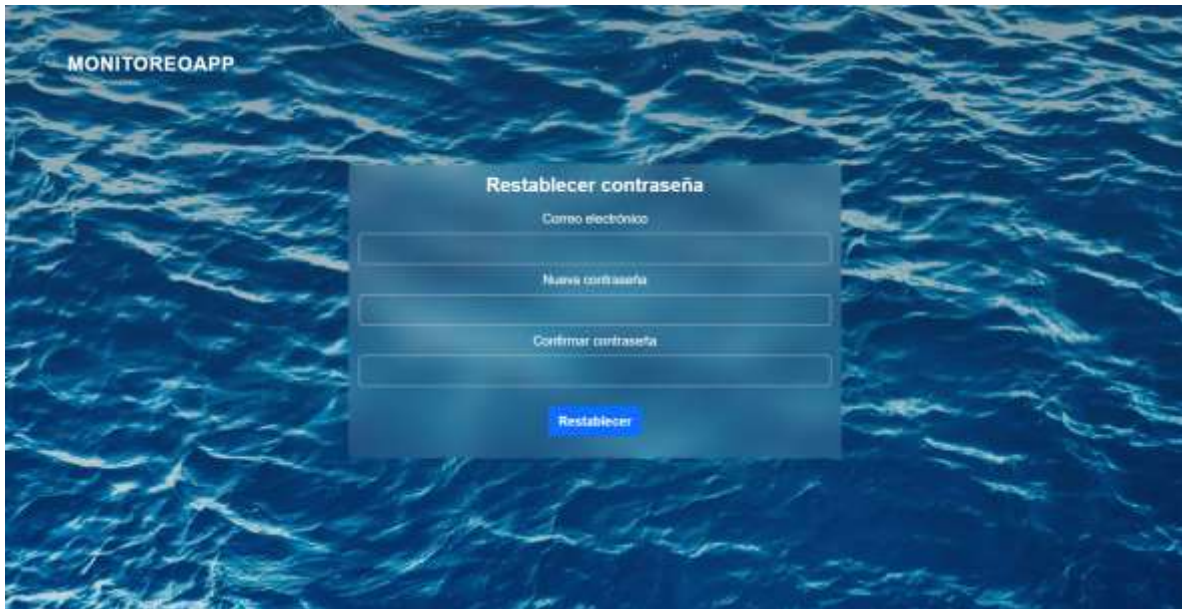
MONITOREOAPP

Restablecer contraseña

Correo electrónico

Enviar

Formulario para el envío del restablecimiento de contraseña a través del correo electrónico



Formulario para el cambio de restablecimiento de contraseña

