



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

**Desarrollo de una aplicación para el monitoreo del crecimiento de tilapias
utilizando machine learning**

**ARMIJOS SIGCHO GALO ALEXANDER
INGENIERO EN TECNOLOGIAS DE LA INFORMACION**

**VASQUEZ RAMON CRISTHIAN ALEJANDRO
INGENIERA EN TECNOLOGIAS DE LA INFORMACION**

**MACHALA
2023**



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

**Desarrollo de una aplicación para el monitoreo del crecimiento de
tilapias utilizando machine learning**

**ARMIJOS SIGCHO GALO ALEXANDER
INGENIERO EN TECNOLOGIAS DE LA INFORMACION**

**VASQUEZ RAMON CRISTHIAN ALEJANDRO
INGENIERA EN TECNOLOGIAS DE LA INFORMACION**

**MACHALA
2023**



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

PROPUESTAS TECNOLÓGICAS

**Desarrollo de una aplicación para el monitoreo del crecimiento de
tilapias utilizando machine learning**

**ARMIJOS SIGCHO GALO ALEXANDER
INGENIERO EN TECNOLOGIAS DE LA INFORMACION**

**VASQUEZ RAMON CRISTHIAN ALEJANDRO
INGENIERA EN TECNOLOGIAS DE LA INFORMACION**

NOVILLO VICUÑA JOHNNY PAUL

**MACHALA
2023**

Coincidencias_TT_Armijos- Vasquez

by Galo Armijos

Submission date: 27-Feb-2024 12:37PM (UTC-0500)

Submission ID: 2208068346

File name: Documento_para_Subir_al_turniting.pdf (2.05M)

Word count: 11822

Character count: 63993

Coincidencias_TT_Armijos-Vasquez

ORIGINALITY REPORT

2%

SIMILARITY INDEX

1%

INTERNET SOURCES

0%

PUBLICATIONS

1%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Hertfordshire Student Paper	1%
2	Submitted to Universidad Anahuac México Sur Student Paper	<1%
3	archive.org Internet Source	<1%
4	qdoc.tips Internet Source	<1%
5	repositorio.uncp.edu.pe Internet Source	<1%
6	outsource-infotech.com Internet Source	<1%
7	catai.net Internet Source	<1%
8	www.sep.gob.mx Internet Source	<1%
9	"Enseñanza de las habilidades del siglo 21: radiografía a marcos teóricos y docentes	<1%

latinoamericanos que revelan un delgado entendimiento común", Pontificia Universidad Católica de Chile, 2023

Publication

10	investigacio.uib.eu Internet Source	<1 %
11	pdfcookie.com Internet Source	<1 %
12	repositorio.unillanos.edu.co Internet Source	<1 %
13	repositorio.utmachala.edu.ec Internet Source	<1 %
14	sedici.unlp.edu.ar Internet Source	<1 %
15	www.cacic2016.unsl.edu.ar Internet Source	<1 %
16	www.ing.unal.edu.co Internet Source	<1 %
17	www.ramblainf.com Internet Source	<1 %
18	www.researchgate.net Internet Source	<1 %
19	aws.amazon.com Internet Source	<1 %

20

www.coursehero.com

Internet Source

<1 %

21

cmap.upb.edu.co

Internet Source

<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

Los que suscriben, ARMIJOS SIGCHO GALO ALEXANDER y VASQUEZ RAMON CRISTHIAN ALEJANDRO, en calidad de autores del siguiente trabajo escrito titulado Desarrollo de una aplicación para el monitoreo del crecimiento de tilapias utilizando machine learning, otorgan a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tienen potestad para otorgar los derechos contenidos en esta licencia.

Los autores declaran que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

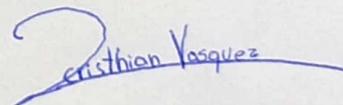
Los autores como garantes de la autoría de la obra y en relación a la misma, declaran que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asumen la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.



ARMIJOS SIGCHO GALO ALEXANDER

0705812147



VASQUEZ RAMON CRISTHIAN ALEJANDRO

0750529596



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

DESARROLLO DE UNA APLICACIÓN PARA EL MONITOREO DEL
CRECIMIENTO DE TILAPIAS UTILIZANDO MACHINE LEARNING

ARMIJOS SIGCHO GALO ALEXANDER

INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

VÁSQUEZ RAMÓN CRISTHIAN ALEJANDRO

INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

NOVILLO VICUÑA JOHNNY PAUL

MACHALA

2023



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

**DESARROLLO DE UNA APLICACIÓN PARA EL MONITOREO DEL
CRECIMIENTO DE TILAPIAS UTILIZANDO MACHINE LEARNING**

ARMIJOS SIGCHO GALO ALEXANDER

INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

VÁSQUEZ RAMÓN CRISTHIAN ALEJANDRO

INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

NOVILLO VICUÑA JOHNNY PAUL

MACHALA

2023

DEDICATORIA

Dedico este trabajo a mi familia, cuyo apoyo ha sido fundamental en cada etapa de mi trayectoria académica y profesional. Especial gratitud a mis padres por su inquebrantable respaldo, por siempre creer en mí y ser el pilar de mi formación. Agradezco también a todas las personas que me han guiado y brindado valiosos consejos a lo largo de este camino. Este logro es tanto de ustedes como mío.

Armijos Sigcho Galo Alexander

Dedico el presente trabajo a mi familia, que me han apoyado desde mis inicios hasta hoy, especialmente a mis padres por su ayuda incondicional en mi formación académica-profesional, por no dejar de creer en mí y a todas las personas que me han guiado y aconsejado.

Vásquez Ramón Cristhian Alejandro

AGRADECIMIENTO

Expreso mi profundo agradecimiento a mi familia, cuyo respaldo ha sido esencial en cada fase de mi desarrollo académico y profesional. Quiero dedicar un reconocimiento especial a mis padres, cuya inquebrantable confianza y apoyo incondicional han sido el cimiento de mi formación. Agradezco sinceramente a todas las personas que han compartido su sabiduría y orientación a lo largo de este viaje. Este logro no solo es mío, sino también de aquellos que han sido mis guías y consejeros.

Armijos Sigcho Galo Alexander

Quiero expresar mi más sincero agradecimiento a mi familia, cuyo amor y apoyo han sido importante durante todo este proceso. Su inquebrantable confianza en mí ha sido mi mayor motivación y fortaleza. A mi tutor de tesis, por su orientación y sabiduría que guiaron este trabajo. A mis amigos, por su apoyo incondicional. Y a todos aquellos que compartieron su tiempo y conocimientos.

Vásquez Ramón Cristhian Alejandro

RESUMEN

La aplicación del aprendizaje automático tiene gran potencial en la acuicultura, ya que puede proporcionar datos importantes sobre el crecimiento de los peces que no requieren medidas de invasión. Además, la introducción de avanzadas técnicas en la acuicultura contribuye al avance tecnológico del sector, lo cual repercute positivamente en los indicadores económicos, la sostenibilidad y el bienestar social. El objetivo del proyecto radica en la necesidad de aplicar estimaciones ágiles y no invasivas sobre el crecimiento de las tilapias que permitan optimizar las mediciones manuales, además de utilizar técnicas estadísticas para la estimación de la longitud y el peso de las tilapias, dejando a un lado los métodos tradicionales. Para lograr este proyecto, se realizaron una serie de pasos. Primero, se llevó a cabo una revisión bibliográfica enfocada en la estimación de peso mediante visión artificial. Segundo, se recolectaron cerca de 1000 imágenes de tilapias para la construcción del dataset. Tercero, se adaptó el modelo pre-entrenado de YOLOv8 para la segmentación de las tilapias, además de realizarse un modelo utilizando una técnica estadística como lo es la regresión lineal específicamente para la estimación del peso y longitud. De este modo, con ambos modelos, se integraron en una página web utilizando la metodología Extreme Programming o XP, a fin de realizar el desarrollo de la aplicación web con sus fases: planificación, diseño, codificación y pruebas. Mediante una matriz de confusión se evaluó el modelo de detección y segmentación con las métricas de precisión y exactitud alcanzando una precisión del 88%, además de evaluar el modelo con la norma ISO 9126 con el objetivo de valorar la funcionabilidad, confiabilidad, usabilidad y eficiencia de esta aplicación web. Al finalizar las pruebas se comprobó con éxito que esta aplicación web puede detectar, segmentar y predecir tanto la longitud como el peso de una tilapia mediante una imagen con cierto grado de precisión a los valores reales, no obstante, las predicciones en algunos casos muestran una discrepancia significativa, posiblemente debido a la limitada variabilidad en los datos recopilados para longitud y peso.

PALABRAS CLAVE:

Aprendizaje automático, detecciones de peces, estimación de peces, red neuronal, tilapias, visión artificial

ABSTRACT

The application of machine learning has great potential in aquaculture as it can provide important data on fish growth that does not require invasive measures. Furthermore, the introduction of advanced techniques in aquaculture contributes to the technological advancement of the sector, which has a positive impact on economic indicators, sustainability and social well-being. The objective of the project lies in the need to apply agile and non-invasive estimates on the growth of tilapia that allow optimizing manual measurements, in addition to using statistical techniques to estimate the length and weight of tilapia, leaving aside traditional methods. To achieve this project, a series of steps were carried out. First, a literature review focused on weight estimation using computer vision was carried out. Second, about 1000 tilapia images were collected for the construction of the dataset. Third, the YOLOv8 pre-trained model was adapted for the segmentation of tilapia, in addition to creating a model using a statistical technique such as linear regression specifically for the estimation of weight and length. In this way, with both models, they were integrated into a web page using the Extreme Programming or XP methodology, in order to develop the web application with its phases: planning, design, coding and testing. Using a confusion matrix, the detection and segmentation model was evaluated with the precision and accuracy metrics, reaching a precision of 88%, in addition to evaluating the model with the ISO 9126 standard with the objective of assessing functionality, reliability, usability and efficiency. of this web application. At the end of the tests, it was successfully verified that this web application can detect, segment and predict both the length and weight of a tilapia using an image with a certain degree of precision to the real values, however, the predictions in some cases show a significant discrepancy, possibly due to limited variability in data collected for length and weight.

KEYWORDS:

Machine learning, fish detections, fish estimate, neural network, tilapia, computer vision

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	10
i. Declaración y formulación del Problema	10
ii. Objeto de estudio y Campo de acción	12
iii. Objetivos	12
iv. Hipótesis y variables o Preguntas de investigación.....	13
v. Justificación.....	14
vi. Organización del documento.....	14
CAPÍTULO I. MARCO TEÓRICO	15
1.1. Antecedentes de la Investigación	15
1.2. Antecedentes históricos.....	19
1.3. Antecedentes Teóricos	20
1.3.1. Tilapias	21
1.3.2. Modelos de Machine Learning.....	22
1.3.3. Técnicas de Procesamiento.....	26
1.3.4. Herramientas de Desarrollo.....	28
1.4. Antecedentes Contextuales.....	29
1.4.1. Ámbito de aplicación.....	30
1.4.2. Establecimiento de requerimientos.....	30
CAPÍTULO II. DESARROLLO DEL PROTOTIPO	30
2.1. Definición del prototipo	30
2.2. Metodología de desarrollo del prototipo	32
2.2.1. Enfoque, alcance y diseño de investigación	32
2.2.2. Unidades de análisis	32
2.2.3. Técnicas e instrumentos de recopilación de datos.....	33
2.2.4. Técnicas de procesamiento de datos para la obtención de resultados	33
2.2.5. Metodología o métodos específicos	33
2.2.6. Herramientas y/o Materiales.....	34
2.3. Desarrollo del prototipo.....	35
2.3.1. Metodología para la construcción de la Red Neuronal Convolutionales.....	35
2.3.1. Metodología XP para el desarrollo de la aplicación web	41
2.4. Ejecución del prototipo	46
CAPÍTULO III. EVALUACIÓN DEL PROTOTIPO	49
3.1. Plan de evaluación.....	49

3.1.1 Planificación.....	49
3.1.2 Herramientas y Técnicas	51
3.2. Resultados de la evaluación	53
3.2.1. Resultados de exactitud y precisión del modelo de segmentación.....	53
CONCLUSIONES	59
RECOMENDACIONES	60
REFERENCIAS	61
ANEXOS.....	64

ÍNDICE DE TABLAS

Tabla 1: Definición de variables y dimensionamiento	13
Tabla 2: Antecedentes Históricos.....	19
Tabla 3: Matriz de Procesos de Automatización.....	29
Tabla 4: Datos de la dataset del entrenamiento y pruebas.....	32
Tabla 5: Técnicas e instrumentos de recopilación de datos.....	33
Tabla 6: Herramientas y/o Materiales	34
Tabla 7: Definición de roles y responsables.....	42
Tabla 8: Plan de publicaciones	43
Tabla 9: Plan de duración de iteraciones.....	44
Tabla 10: Diagrama de Gantt	49
Tabla 11: Matriz de Confusión.....	51
Tabla 12: Comparación de resultados entre aplicación web y mediciones tradicionales.....	52
Tabla 13: Resultados de la Matriz de confusión	53
Tabla 14: Indicadores del entrenamiento	54
Tabla 15: Resultados de métricas de entrenamiento	54
Tabla 16: Configuración de prueba de Carga JMeter.....	55
Tabla 17: Prueba con 50 usuarios.....	55
Tabla 18: Prueba con 200 usuarios.....	55
Tabla 19: Prueba con 500 usuarios.....	55
Tabla 20: Resultados entre aplicación web y mediciones tradicionales primera comparación.....	56
Tabla 21: Resultados entre aplicación web y mediciones tradicionales segunda comparación	57
Tabla 22: Tiempos de Mediciones de tilapias	58
Tabla 23: Historia de usuario Nro. 1.....	12
Tabla 24: Historia de usuario Nro.2.....	12
Tabla 25: Historia de usuario Nro.3.....	13

ÍNDICE DE FIGURAS

Figura 1: Diagrama de causa y efecto	11
Figura 2: Proceso de búsqueda SLR.....	17
Figura 3: Artículos por fuentes de investigación.....	17
Figura 4: Cantidad de artículos Seleccionados por año.....	18
Figura 5: Análisis por palabras claves – VOSviewer.....	18
Figura 6: Mapa conceptual de los antecedentes teóricos.....	20
Figura 7: Entrenamiento de la red neuronal convolucional utilizando yolov8.....	31
Figura 8: Entrenamiento del modelo de estimación de tilapias.....	31
Figura 9: Arquitectura de desarrollo de la aplicación móvil	31
Figura 10: Metodología para la construcción de la Red Neuronal	34
Figura 11: Fases de la metodología XP	34
Figura 12: Diagrama de flujo de la recolección de datos	35
Figura 13: Captura de imagen de tilapias	36
Figura 14: Diagrama de flujo del procesamiento de imágenes	36
Figura 15: Segmentación de la tilapia usando LinkedAI	37
Figura 16: Formato yolov8 - Ultralytics.....	37
Figura 17: Conversión de las máscaras al formato aceptado por yolov8	38
Figura 18: Modelo de segmentación de Tilapias.....	38
Figura 19: Código para la obtención del área, perímetro y relación de aspecto.....	39
Figura 20: Modelo de Regresión Lineal.....	40
Figura 21: Entrenamiento del modelo de segmentación de Tilapias.....	41
Figura 22: Entrenamiento del Modelo de Regresión Lineal	41
Figura 23: Pantalla de Inicio	44
Figura 24: Pantalla de resultados.....	45
Figura 25. Función para actualizar el estado de la interfaz con la máscara	45
Figura 26. Configuración de Flask que utiliza dos modelos. segmentación y regresión.....	46
Figura 27: Predicciones del modelo de Segmentación de Tilapias	46
Figura 28: Predicción del Peso Tilapia del Modelo de Regresión Lineal Múltiple.....	47
Figura 29. Comando para ejecutar la página web	47
Figura 30. Comando para ejecutar el servidor.....	47
Figura 31. Pantalla principal de la página web.....	48
Figura 32. Pantalla de predicción y segmentación de las tilapias	48
Figura 33. Pantalla de resultados de las tilapias segmentadas.....	49
Figura 34: Matriz de confusión	53
Figura 35: Prueba de rendimiento	56
Figura 36: Resultados de la predicción de la aplicación web.....	57

GLOSARIO

A

Algoritmo de Procesamiento de Imágenes: Conjunto de técnicas computacionales para mejorar la calidad o extraer información de una imagen.

B

Base de Datos de Imágenes: Repositorio de imágenes usadas para entrenar o evaluar modelos de machine learning en el contexto de reconocimiento o estimación de parámetros.

C

Clasificación: Tarea de machine learning que consiste en asignar una de varias etiquetas predefinidas a una entrada dada.

E

Entrenamiento de Modelo: Proceso de ajustar un modelo de machine learning utilizando un conjunto de datos específico.

Etiqueta: Valor o categoría que se asigna a una entrada en el aprendizaje supervisado.

M

Machine Learning (Aprendizaje Automático): Campo de la inteligencia artificial que permite a las máquinas aprender de los datos.

Métricas de Validación: Conjunto de medidas utilizadas para evaluar el rendimiento de un modelo de machine learning.

P

Procesamiento de Imágenes: Manipulación y análisis de imágenes para extraer información o realizar ciertas transformaciones.

R

Regresión: Tipo de tarea de machine learning en la que se estima una variable continua.

Rendimiento del Modelo: Evaluación de qué tan bien un modelo de machine learning cumple con la tarea para la que fue entrenado.

INTRODUCCIÓN

El crecimiento de la acuicultura como una medida sostenible que permita abastecer la creciente demanda de productos pesqueros, plantea la necesidad de optimizar la producción, particularmente en especies como la tilapia objeto de estudio. Sin embargo, los métodos tradicionales de medición y monitoreo del crecimiento de estos peces a menudo son invasivos y estresantes, lo cual no solo plantea preocupaciones éticas, sino que también puede influir negativamente en la tasa de crecimiento y bienestar de los peces. A este desafío se añade la complejidad inherente a las mediciones manuales, que son susceptibles a errores humanos y requieren una considerable inversión de tiempo, así como de recursos.

En este contexto, se propone desarrollar una aplicación web basada en machine learning, para estimar el crecimiento de tilapias en lo que respecta al peso y longitud. El desarrollo de esta aplicación está fundamentado mediante el empleo de redes neuronales convolucionales y el uso de una técnica estadística como lo es la regresión lineal múltiple a fin de predecir el peso y longitud de las tilapias, además de utilizar técnicas de procesamiento de datos y un aprendizaje supervisado acabo para entrenar la red neuronal, esta solución propone minimizar el estrés en los peces al eliminar la necesidad de manipulación directa. De esta manera, la aplicación se presenta como una solución innovadora al agilizar los procesos tradicionales de medición invasiva en la producción de tilapias, además de contribuir e a prácticas más sostenibles en la acuicultura.

i. Declaración y formulación del Problema

A nivel mundial, el uso de nuevas tecnologías como lo es la inteligencia artificial es un factor importante dentro del sector de la acuicultura debido a la optimización de procesos como lo es la clasificación, medición de la longitud y pesos de las tilapias que se realizaban de manera manual conllevando inconvenientes como alto consumo de tiempo, recursos y mano de obra El estudio realizado en 1994 nos habla acerca de las primeras interacciones en optimizar estos procesos realizados por Naiberg Little [1], buscaba el reconocimiento de objetos mediante un detector de bordes para localizar a los peces con el objetivo de determinar sus distancias y dimensiones utilizando un método estructural bidimensional con el fin de clasificar y determinar si el objeto en cuestión se trataba de un objeto de interés.

Como se indica en [2], la detección de peces en su medio natural por medio de la visión artificial permitirá evitar la utilización de técnicas que podrían resultar invasivas, como es el caso del uso de sensores.

Como se menciona en [3], la obtención de datos de forma automática mediante cualquier tipo de tecnología es uno de los grandes retos debido a la agilidad y acortamiento de los procesos.

De hecho, como lo hace notar Muñoz [4], la utilización de métodos no convencionales para estimar el peso provoca estrés, lo que disminuye las estimaciones a la hora de estimar el peso del animal.

En [5], las investigaciones sobre los problemas de crecimiento en peces señalan que el estrés es fundamental como un factor clave para su desarrollo y crecimiento.

En la Figura 1, se presenta un esquema de causa y efecto en el que se examinan diversos factores cuya influencia incide en la medición del crecimiento de las tilapias. Este diagrama se centra en varios aspectos clave: métodos, personal, herramientas y estrés ambiental, todos los cuales tienen implicaciones tanto para la precisión de la medición como para el bienestar de la tilapia.

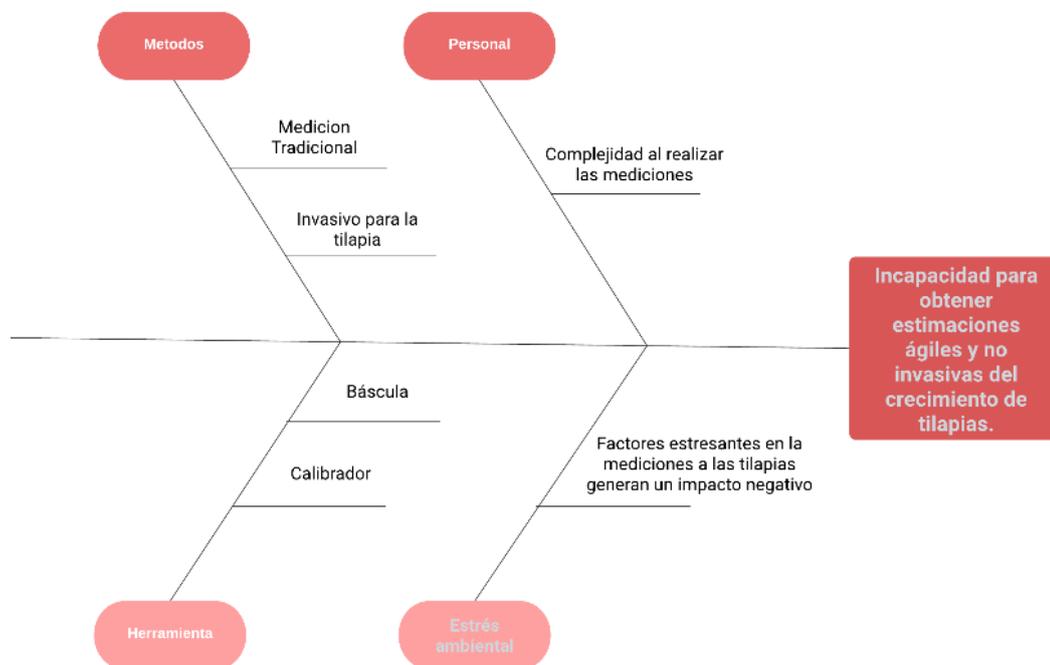


Figura 1: Diagrama de causa y efecto

Formulación del problema

- **Problema principal:**

Necesidad de desarrollar una aplicación para la detección de tilapias que nos permita obtener las estimaciones del crecimiento de una manera ágil debido al alto consumo de tiempo mediante la manera tradicional.

- **Problemas específicos:**

¿Cómo superar las limitaciones de iluminación, ángulos de captura y posibles obstrucciones en las imágenes para asegurar una detección confiable de la tilapia?

¿Cómo recopilar y almacenar los datos de crecimiento de las tilapias, considerando parámetros como tamaño, peso, longitud, mediante el reconocimiento de imágenes para entrenar modelos de machine learning?

¿Qué algoritmos y técnicas de machine learning son los más adecuados para analizar los datos de crecimiento de las tilapias y extraer patrones o tendencias que permitan predecir su crecimiento futuro?

¿Cómo garantizar la facilidad de uso de la aplicación para los usuarios, incluidos los piscicultores y personal encargado del monitoreo, mediante una interfaz intuitiva?

ii. Objeto de estudio y Campo de acción

Objeto de estudio

- Aplicación basada en machine learning para el reconocimiento y estimaciones del crecimiento de tilapias.

Campo de acción

- Desarrollo e implementación de una aplicación basada en machine learning que utilice redes neuronales convolucionales y modelos estadísticos para la detección y estimación del crecimiento de tilapias.

iii. Objetivos

Objetivo General

- Desarrollar una aplicación web utilizando redes neuronales convolucionales, para la estimación del crecimiento de tilapias, mediante el uso de técnicas de modelado estadístico.

Objetivos específicos

- Recopilar información de artículos científicos de fuentes confiables para construir el estado del arte y marco teórico.
- Recopilar datos de tilapias para la creación de un dataset representativo de imágenes de tilapias.
- Implementar la red neuronal convolucional YOLOv8 al servicio de la segmentación y detección de objetos en imágenes.
- Implementar un modelo estadístico, como regresión, para la estimación del tamaño y peso de las tilapias.
- Realizar pruebas de estimación de peso y tamaño de las tilapias, además de validar ambos modelos con sus respectivas métricas.

- Desarrollar una interfaz de usuario intuitiva y amigable que permita a los usuarios subir una imagen y recibir información sobre los parámetros de longitud y peso de las tilapias.

iv. Hipótesis y variables o Preguntas de investigación

Si se desarrolla una aplicación web utilizando machine learning, para detectar, segmentar y estimar la longitud y peso de las tilapias mediante algoritmos de aprendizaje supervisado, agilizando el proceso de medición directa, se evaluará la precisión comparando las estimaciones con las mediciones reales.

Variables y dimensionamiento (o categorización)

La Tabla 1 presenta el marco de referencia para nuestro estudio, especificando las Variables, Categorías, Indicadores y Técnicas, siendo cruciales para entender los componentes metodológicos y tecnológicos que conforman el enfoque del proyecto.

Tabla 1: Definición de variables y dimensionamiento

Variables	Categorías	Indicadores	Técnicas
Independiente Aplicación web utilizando machine learning	Procesamiento De Imágenes	Conjunto de Datos	Dataset de imágenes de tilapias con su respectiva etiqueta
		Aprendizaje supervisado	Entrenamiento de un modelo con el dataset utilizando redes neuronales
	Desarrollo web	Desarrollo BackEnd	Lenguaje de programación Python utilización de flask.
		Desarrollo	Utilización de React.
		Pruebas	Uso de la aplicación
Dependiente Detección, segmentación y Estimaciones del peso y longitud De las tilapias	Precisión de las estimaciones y detección	Matriz de confusión	Comparación de las estimaciones realizadas de la página web con las mediciones manuales.

v. Justificación

El crecimiento de la tilapia constituye un aspecto crítico de la acuicultura, que afecta la productividad y rentabilidad. Es posible analizar una gran cantidad de datos mediante técnicas de aprendizaje supervisado que ayuden a comprender y predecir el crecimiento de la tilapia. Hay varias razones para llevar a cabo este estudio. Entre estas razones, una de ellas es realizar estimaciones manuales de longitud y peso de la tilapia, es un proceso laborioso e invasivo, además de propenso a errores. Es por eso que la justificación de este estudio está basada en la utilización de redes neuronales convolucionales, que nos permiten detectar a la tilapia y el uso de la regresión lineal múltiple para predecir la longitud y peso de las tilapias. La motivación personal de esta investigación radica del interés en el empleo de las nuevas tecnologías, tales como el aprendizaje automático, en la acuicultura. Los beneficiados directos de esta investigación será la finca ubicada en el Sitio “Río Chico” llamada Camping House, que podrán utilizar la aplicación para agilizar los procesos de medición tradicional. El desarrollo de aplicaciones es posible debido a la evolución de la tecnología, de herramientas disponibles y recursos de desarrollo de software. Además, existen bases de datos y conjuntos de datos disponibles, pero decidimos crear nuestra propia base de datos recopilando imágenes para evidenciar el proceso.

vi. Organización del documento

El documento se estructura de la siguiente manera:

Capítulo 1: En el presente capítulo se realizan los fundamentos de la investigación, los fundamentos teóricos, así como los fundamentos contextuales en donde se trata del campo de aplicación y establecimiento de requisitos referentes al tema a trabajar.

Capítulo 2: Se explican el desarrollo del prototipo, tanto como su definición, la metodología aplicada y las herramientas utilizadas en el desarrollo.

Capítulo 3: Se visualizan los resultados del proyecto, también se lo evalúa y se realizan conclusiones, así como recomendaciones según los objetivos específicos planteados.

CAPÍTULO I. MARCO TEÓRICO

1.1. Antecedentes de la Investigación

En los Antecedentes de la Investigación, empezaremos delineando las Preguntas de Investigación que buscan entender el impacto del Machine Learning en el monitoreo y optimización del crecimiento de tilapias en entornos de acuicultura. Seguidamente, nos centraremos en las Palabras Claves y Cadenas de Búsqueda elegidas para guiar nuestra exploración de bases de datos académicas confiables. Después, detallaremos los Criterios sobre Inclusión y Exclusión para asegurar la calidad y la importancia de los estudios seleccionados para nuestra revisión. Finalmente, discutiremos el Proceso y Resultados de la Búsqueda, subrayando cómo estas etapas contribuyen a un entendimiento profundo y actualizado del tema en estudio.

a) Preguntas de investigación

- ¿Cómo puede el uso de técnicas de Machine Learning mejorar el monitoreo y la predicción del crecimiento de tilapias en una aplicación de seguimiento del crecimiento basada en datos de acuicultura?
- ¿Cómo puede la aplicación de Machine Learning mediante el monitoreo del crecimiento de las tilapias mejorar la toma de decisiones en acuicultura y contribuir en la optimización productiva y del bienestar de los peces?
- ¿Cómo podría la aplicación de monitoreo del crecimiento de tilapias basada en Machine Learning contribuir específicamente a mejorar la eficiencia en el seguimiento del crecimiento de tilapias?

b) Palabras claves y Cadena(s) de búsqueda

Se especificó la cadena de búsqueda teniendo en cuenta los principales términos bajo investigación (Aprendizaje automático, estimación del tamaño de los peces, redes neuronales, detecciones de peces)

("Peces") Y ("Aprendizaje automático" O "detecciones de peces" O "estimación del tamaño de los peces" O "redes neuronales")

Cadena de búsqueda en inglés:

("Fish") AND ("Machine learning" OR "fish detections" OR "fish size estimate" OR "neural networks")

Palabras claves

- fish detections
- fish size estimate
- machine learning
- neronal network

c) Criterios de inclusión y exclusión

Criterio de inclusión:

- Los estudios deben abordar temas relacionados con el machine learning para la detección y estimación del crecimiento de peces.
- Los estudios deben ser de fuentes de bases de datos confiables.
- Artículos escritos en cualquier idioma que se relacione con el tema.
- Estudios publicados desde 2019 hasta 2023

Criterio de Exclusión:

- Artículos duplicados
- Estudios claramente irrelevantes para la investigación, que no están relacionados con el tema ni al objetivo del proyecto.
- Artículos anteriores al 2019

d) Proceso y resultados de la búsqueda

En la Figura 2 se muestra el Diagrama de flujo que detalla el procedimiento de búsqueda SRL (Systematic Literature Review). Este diagrama se guía por cuatro pasos clave para la selección y análisis de literatura académica relevante. En el primer paso, se identifican los estudios extraídos de bases de datos reconocidas como SCOPUS, ISI Web of Science y IEEE Digital Library. El segundo paso implica la eliminación de documentos duplicados para garantizar la originalidad del material revisado. En el tercer paso, se lleva a cabo un análisis preliminar de los títulos y resúmenes de los trabajos, excluyendo aquellos que no cumplen con los criterios de inclusión preestablecidos. Finalmente, en el cuarto paso, se recuperan y revisan en detalle los textos completos de los artículos previamente seleccionados, quedando únicamente aquellos que satisfacen los criterios de inclusión para el análisis posterior.

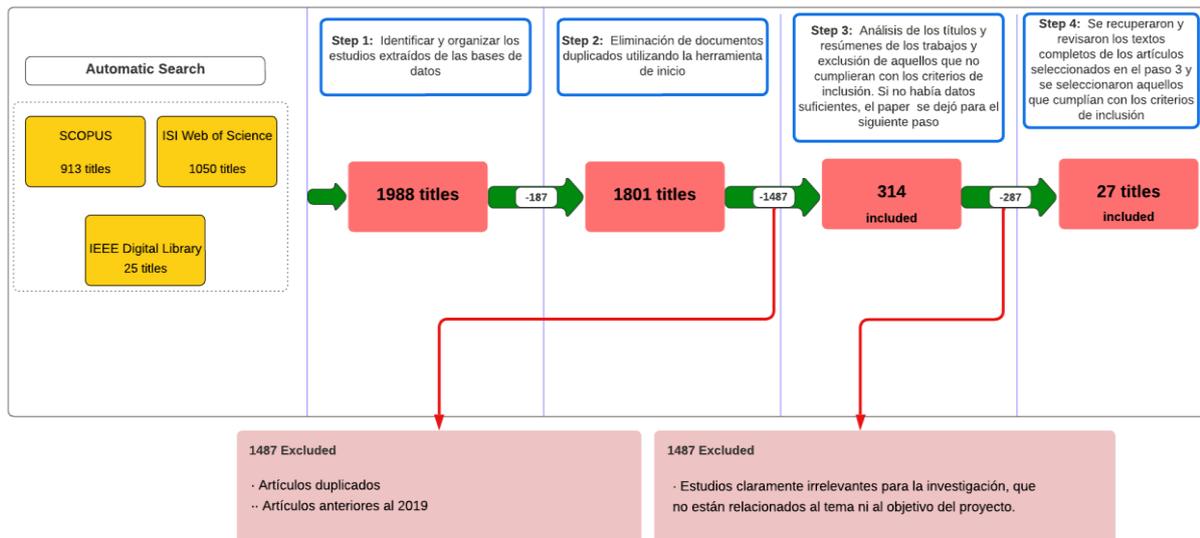


Figura 2: Proceso de búsqueda SLR

Fuente: Parsifal

Resultados de la búsqueda

La Figura 3 muestra un gráfico estadístico donde se comparan las diferentes fuentes bibliográficas utilizadas en la investigación científica. Según los datos presentados, se observa que Scopus representa el 45% del total de publicaciones indexadas, mientras que ISI Web of Science tiene una participación ligeramente mayor, alcanzando el 52.8%. Por otro lado, IEEE Digital se muestra con una presencia más modesta, constituyendo únicamente el 1.3% del total.

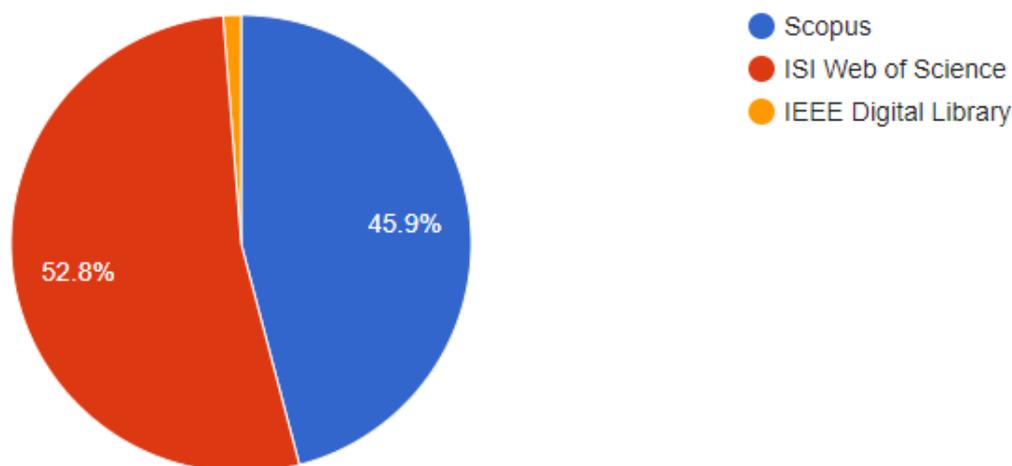


Figura 3: Artículos por fuentes de investigación

fuente: Parsifal

La Figura 4 muestra un gráfico en el que se refleja el número de publicaciones relacionadas con nuestro tema de investigación entre 2019 y 2023. Finalmente, sólo se tuvieron en cuenta aquellos artículos se ajustaban a los criterios de inclusión estipulados.

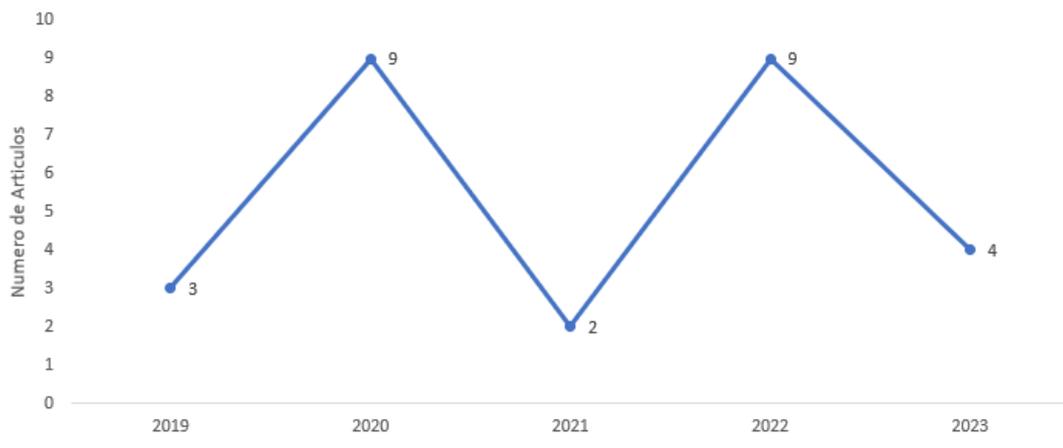


Figura 4: Cantidad de artículos Seleccionados por año
Fuente: Parsifal

Con el uso de la herramienta VOSviewer se pudo realizar el análisis bibliométrico usando palabras claves de la base de datos Scopus. El resultado se lo puede visualizar en la Figura 5, donde se contemplan que el término machine learning, object detection, neural network y entre otros, se encuentran con mayor frecuencia y están relacionadas con el tema de titulación.

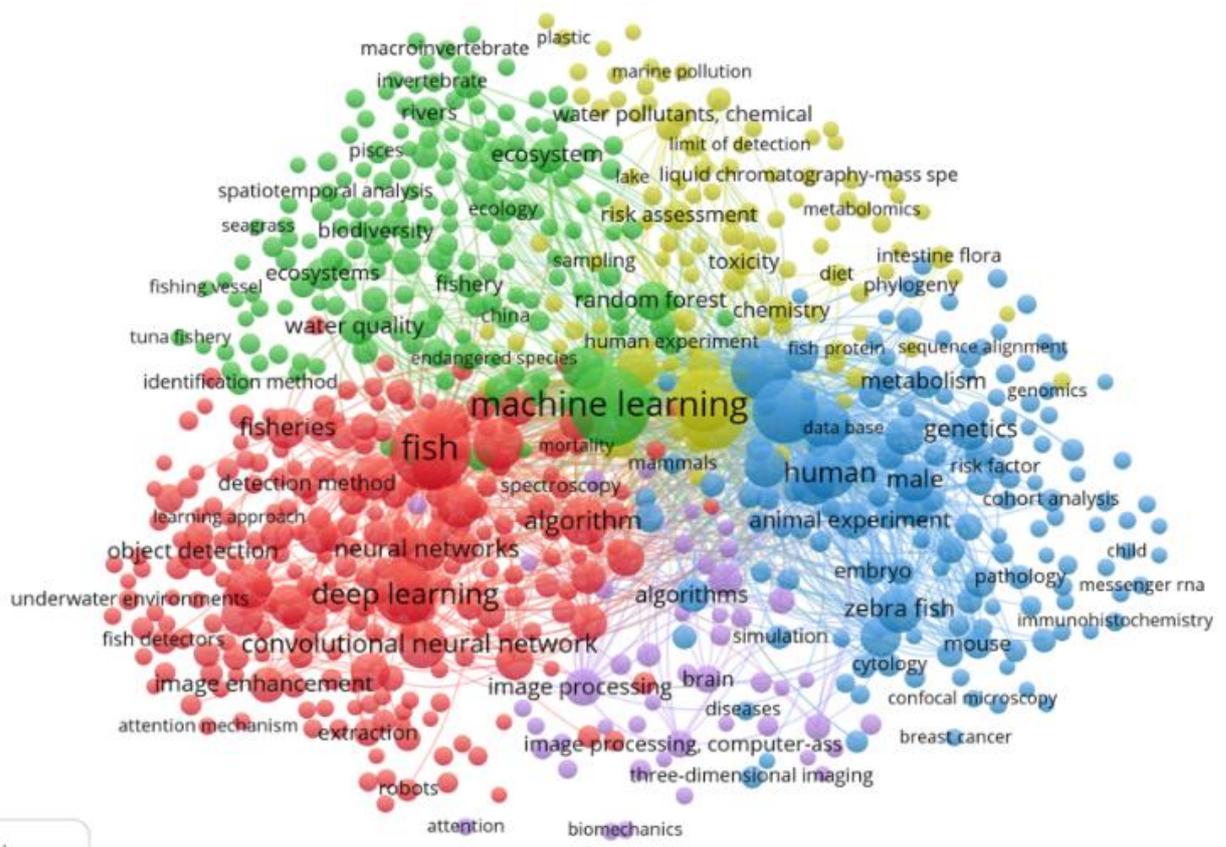


Figura 5: Análisis por palabras claves – VOSviewer
Fuente: VOSviewer

1.2. Antecedentes históricos

En la tabla 2, se ilustran los antecedentes históricos de este estudio, examinando la evolución del campo de la acuicultura y la inteligencia artificial de las herramientas de Machine Learning y su aplicación en diversas áreas, el monitoreo y la optimización del crecimiento de especies acuáticas. Este panorama histórico proporciona un contexto crucial para entender la importancia y la oportunidad de abordar nuestras preguntas de investigación.

Tabla 2: Antecedentes Históricos

Año	Antecedentes Históricos
2001	En [6], realizaron un estudio en el cual clasificaron imágenes de peces en tilapia y en otras especies usando técnicas de procesamiento de imágenes y machine learning de manera simultánea.
2018	Un estudio publicado en [7], se utilizaron tanto enfoques estadísticos basados en suposiciones como no basados en suposiciones en el análisis. Además, se emplearon algoritmos de aprendizaje automático, como redes neuronales y la importancia de las características para la selección de índices morfológicos.
2019	En agosto de 2019, Rungruangsak y Manoonpong [8], presentan un modelo computacional neuronal llamado GrowthEstimate, que utiliza redes neuronales recurrentes de computación de reservorios para estimar el peso y la longitud de los peces a lo largo del tiempo.
2020	Un estudio publicado en [9], en Wiley Online Library exploró la aplicación de redes neuronales convolucionales profundas para predecir la longitud, la circunferencia y el peso de los peces a partir de imágenes en su mayoría deshidratadas.
2021	Un estudio hecho por Yang [10] aplicó técnicas de aprendizaje automático, específicamente redes neuronales convolucionales profundas, para predecir el peso de los peces directamente a partir de imágenes de peces, evitando la necesidad de mediciones de longitud.
2022	En [2], exploraron la utilización de modelos de regresión y aprendizaje profundo en la estimación del peso de peces en aguas.
2023	Un artículo de revisión realizado en [11], analizó la aplicación de las técnicas basadas en el aprendizaje profundo para la detección de objetos en acuicultura de peces, incluida la monitorización automatizada y el análisis del estado de crecimiento y el comportamiento de los peces.

1.3. Antecedentes Teóricos

En La Figura 6, se presentan los temas y subtemas vinculados al proyecto de investigación, que se abordarán para construir el marco teórico que sustentará este trabajo.

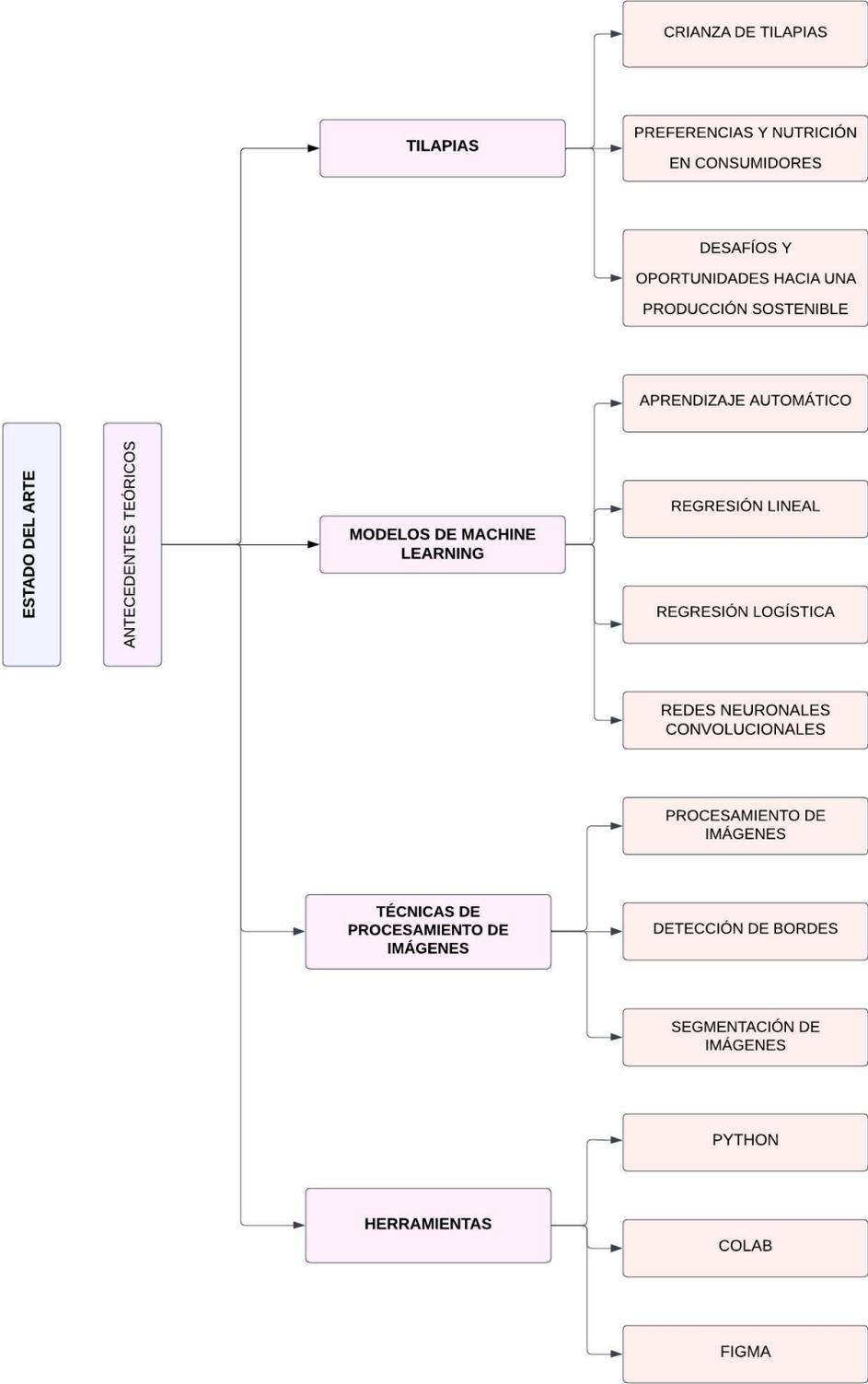


Figura 6: Mapa conceptual de los antecedentes teóricos

1.3.1. Tilapias

1.3.1.1. Crianza de Tilapias

En [12], el proceso de producción de tilapias lo hicieron recopilando datos de un experimento en Yucatán, México, en conformidad con Buenas Prácticas de Gestión. Utilizando juveniles con un peso inicial de $32,71 \pm 3,57$ g, se lleva a cabo la cría en tanques circulares de geomembrana. La región tropical, caracterizada por estaciones climáticas distintas (lluviosa y seca), incidiendo en los costos de producción.

Se construyen cuatro modelos bioeconómicos para evaluar la eficiencia económica de tecnologías en ambas estaciones, considerando parámetros biológicos, de gestión y económicos, y contrastando la rentabilidad a corto y largo plazo. Además, se realiza una proyección financiera para los próximos 10 años, contemplando diversos escenarios y una capacidad de 30 tanques circulares de geomembrana. Este enfoque integral busca optimizar la producción de tilapias considerando las condiciones específicas de la región y las variables clave del proceso [12].

1.3.1.2. Preferencias y Nutrición en Consumidores

En este estudio [13], centrado en la producción de tilapias, adopta una perspectiva de género para analizar las preferencias de cría selectiva de tilapias entre consumidores de bajos ingresos en Egipto. El enfoque se alinea con marcos teóricos de investigación de género y desarrollo, considerando comportamientos de consumo y preferencias alimentarias en entornos con bajos ingresos. Se aborda la necesidad de datos detallados a nivel de hogar para explicar las diferencias en el consumo y preferencias alimentarias entre miembros de la familia de diferentes sexos.

Además, el estudio examina la importancia nutricional de alimentos ricos en micronutrientes, como la tilapia, para madres e infantes. Se destaca el papel potencial del suministro de tilapia en el abordaje de la desnutrición en áreas remotas de Egipto. La investigación también considera las implicaciones de factores socioeconómicos, así como de ubicación de los patrones de consumo, señalando diferencias regionales en empleo, pobreza, seguridad alimentaria y salud de las mujeres. Es importante notar que el estudio reconoce limitaciones al no examinar las relaciones de género dentro del hogar en los procesos de toma de decisiones [13].

1.3.1.3. Desafíos y Oportunidades hacia una Producción Sostenible

Este estudio [14] evalúa la viabilidad de utilizar residuos de tilapia del Nilo en la elaboración de biodiésel, destacando la importancia de encontrar materias primas de bajo costo para competir en el mercado de diésel fósil. Se aborda el proceso de producción de biodiésel mediante

transesterificación alcalina y se destaca la necesidad de utilizar residuos de tilapia para una eliminación ambientalmente segura. Los residuos, particularmente extraídos de las cabezas de tilapia, presentan desafíos debido a su alto valor ácido y viscosidad cinemática. Se discuten enfoques alternativos, como el uso de catalizadores básicos homogéneos y métodos asistidos por ondas ultrasónicas, con el fin de perfeccionar la calidad del biodiésel y minimizar los gastos. Se mencionan también estrategias para mejorar la calidad de los desechos de tilapia, como la neutralización continua en dos pasos antes de la reacción de transesterificación alcalina. Sin embargo, se señala la falta de evaluación económica en algunos estudios previos, y se destaca la necesidad de considerar probables modificaciones en la mejora de la materia prima.

1.3.2. Modelos de Machine Learning

1.3.2.1. Introducción a los modelos de machine learning

El aprendizaje automático es un conjunto de técnicas basadas en la inteligencia artificial para permitir que las máquinas aprendan patrones y tomen decisiones basadas en datos sin necesidad de programarlas explícitamente. Dentro del contexto de la monitorización del crecimiento de la tilapia, el Machine Learning tiene un papel clave a la hora de ofrecer métodos automatizados para la clasificación de especies de peces marinos [15].

Se utilizó una técnica de aprendizaje profundo inteligente para la clasificación de peces con una metodología avanzada que combina el aprendizaje profundo y el procesamiento de imágenes para la identificación y clasificación eficiente de especies de peces marinos a partir de imágenes y vídeos. Se ha comprobado el éxito en la mejora de las técnicas actuales de identificación de peces, consiguiendo una mayor precisión y rendimiento. Esta técnica propone una solución no invasiva, operativa y transportable que mejora la comprensión y protección del medio ambiente oceánico mediante el uso de Machine Learning en la identificación de peces [15].

En [16], destacan tres futuras tendencias en este campo. Primero, la clasificación detallada de los peces, incluyendo la detección de enfermedades y comportamientos inadecuados. En segundo lugar, disponer de conjuntos de datos más completos y variados para entrenar los modelos. Y tercero, el desarrollo de modelos avanzados que combinan el aprendizaje profundo con las características elaboradas a mano, y que son capaces de abordar las secuencias espaciotemporales en la acuicultura. Todas estas tendencias apuntan a revolucionar el modo de supervisar y gestionar la acuicultura, ofreciendo herramientas más precisas y útiles para potenciar el rendimiento y la conservación de los recursos acuáticos.

El aprendizaje automático es muy importante para el seguimiento de las especies, pues permite clasificar los peces de forma adecuada y coherente. De este modo, se evita la necesidad de realizar esfuerzos exhaustivos de muestreo que pueden resultar perjudiciales para el ecosistema marino [15].

Las aplicaciones y beneficios del aprendizaje automático en acuicultura se han explorado ampliamente, en particular en enfoques de aprendizaje profundo, comparado con los sistemas expertos con características diseñadas manualmente, con ventajas significativas, sobre todo la capacidad de extracción automática de características relevantes de las imágenes.

1.3.2.2. Regresión Lineal

Según [17], la regresión lineal como modelo predictivo es ampliamente utilizado en estadística y aprendizaje automático. Establecen la relación matemática existente entre una variable independiente y una dependiente. Dicha relación está representada por una recta en un diagrama disperso, donde los puntos de los datos se encuentran dispersos en torno a la recta.

Como se aborda en el trabajo [17], se utilizó la regresión lineal como una herramienta analítica que permitió evaluar la exactitud y calidad del modelo Mask R-CNN en las predicciones realizadas en la detección y conteo de camarones en el agua. Ello permitió identificar la versión del modelo que ofrecía mejores resultados y la forma en que determinados ajustes de los parámetros podrían mejorar su capacidad de predicción.

Por otra parte, la variable dependiente, también llamada variable de respuesta o variable objetivo, es aquella que se quiere predecir o explicar. Además, tenemos la variable independiente, llamada también variable predictora, que se usa para explicar o predecir la variable dependiente.

El análisis de regresión lineal realizado con el modelo Mask R-CNN para detectar camarones bajo el agua se basa en los conceptos fundamentales de variable dependiente y variable independiente. Primero, La variable dependiente es la cantidad real de camarones que se quiere predecir, por su parte, las variables independientes corresponden a las características y parámetros del modelo Mask R-CNN utilizado para efectuar las predicciones. Dichas variables nos permiten establecer una relación matemática y mejorar mediante aprendizaje profundo la precisión en la estimación del recuento de camarones [17].

El objetivo de la regresión lineal es encontrar una ecuación de línea recta que mejor se ajuste a los datos, de manera que se pueda predecir o estimar el valor de la variable dependiente en función de los valores de las variables independientes [17].

Luego de llevar a cabo el análisis de regresión lineal, los acuicultores pueden emplear el modelo para tomar decisiones informadas sobre el manejo y cuidado de los camarones. Por ejemplo, podrían ajustar la cantidad de alimento suministrado o controlar la temperatura del agua para optimizar el crecimiento y la productividad del camarón en un entorno acuático controlado y sostenible.

1.3.2.3. Regresión Logística

Se menciona en [18], que la regresión logística es un método estadístico muy utilizado en los análisis de datos para la predicción de variables categóricas binarias, es decir, variables que solo pueden tener dos posibles categorías o estados, tales como sí/no, verdadero/falso o positivo/negativo.

La simplicidad y eficiencia computacional de la regresión logística la hacen ideal para tareas de clasificación binaria en problemas donde se requiere interpretabilidad y se dispone de datos etiquetados. Además, puede extenderse a problemas de clasificación multiclase mediante técnicas como la regresión logística multinomial [18].

En los estudios [17] [18], la regresión lineal es un método predictivo para variables continuas, estableciendo una relación matemática entre variables. Por el contrario, la regresión logística constituye un método estadístico para las variables categóricas binarias, que se basa en la probabilidad de los resultados. Ambas resultan de gran utilidad en el estudio de los datos y el aprendizaje automático, pero se diferencian en el tipo de variable y el modo en que se establece la relación.

Se propuso un modelo que emplea la regresión logística dentro de un clasificador ensemble apilable, conjuntamente con SVM (algoritmo de aprendizaje supervisado), KNN (algoritmo de las K vecinas más cercanas) y árbol de decisiones. En el conjunto de datos se generaron cuatro puntuaciones de predicción distintas, y después se aplicó un enfoque de regresión logística destinado para meta aprendices, el cual tomaba las puntuaciones de predicción ya generadas. Con esta implementación se logró una clasificación precisa sobre el crecimiento de la tilapia basada en las características extraídas de las regiones seleccionadas en la fase de preprocesamiento [18].

Se presentó dos enfoques relacionados con la aplicación de aprendizaje automático y análisis de imágenes. El primero es un clasificador ensemble que incluye un modelo de regresión logística para el meta-aprendizaje, mientras que el segundo se enfoca en la aplicación de regresión logística al análisis de imágenes después de utilizar técnicas de detección de objetos destacables en el

preprocesamiento de las imágenes. Estos enfoques pueden ser aplicados en diversas áreas para resolver problemas relacionados con el procesamiento de datos y el gestión de imágenes [18].

Aunque la regresión logística es poderosa en muchos escenarios, su capacidad de modelado puede verse limitada cuando se enfrenta a relaciones no lineales o cuando las clases no son perfectamente separables. En tales casos, es común utilizar técnicas más avanzadas como las redes neuronales o el aprendizaje profundo.

1.3.2.4. Redes Neuronales

Las Redes Neuronales Convolucionales (CNN) constituyen un tipo de red neuronal profunda diseñada especialmente para el análisis y procesamiento de imágenes. Su funcionamiento se inspira en el del sistema visual en el cerebro humano y está compuesto por distintas capas que realizan las operaciones de convolución y pooling con el fin de obtener características relevantes y extraer los patrones a partir de las imágenes [19].

Se utilizó la arquitectura de un clasificador de red neuronal convolucional (CNN) utilizado para clasificar parches de imágenes dentro de cinco clases diferentes. El objetivo principal de la arquitectura era extraer características importantes de las imágenes y usarlas para clasificarlas [19].

Por su parte, la capa de agrupamiento máximo se encarga de reducir la dimensionalidad de las características para facilitar el procesamiento. Para completar la clasificación, se utilizan las capas totalmente conectadas que, utilizando las características extraídas, establecen la clase a la que pertenece cada fragmento de imagen. En definitiva, con esta arquitectura se ha conseguido una clasificación precisa y segura de las imágenes en las cinco clases deseadas [19].

En [20], se entrenaron tres redes neuronales de convolución regionales (R-CNN) con 40 imágenes públicas. Las imágenes de la lubina europea con marcador fiduciario fueron capturadas 38 con 3 cámaras no especializadas. Las 38 imágenes fueron distorsionadas usando propiedades intrínsecas de la lente 38 calculadas por la cámara en OpenCV, luego se estimó el TL utilizando visión artificial (MV) para detectar tanto al marcador como al sujeto.

Para controlar el crecimiento de los peces, se capturaron imágenes de estos en un entorno acuático utilizando cámaras no especializadas. Se aplicaron técnicas de visión artificial y R-CNN para detectar y seguir el crecimiento de los peces a lo largo del tiempo. Mediante la captación de imágenes en diferentes momentos, pueden obtenerse datos sobre el crecimiento individual de los peces y el crecimiento global de la población [20].

Las redes neuronales son una poderosa y versátil herramienta de aprendizaje automático inspirada en el funcionamiento del cerebro humano. Su estructura jerárquica y la capacidad de aprender a partir de datos las convierten en una tecnología fundamental en el campo del aprendizaje profundo. Estas redes han demostrado ser altamente eficientes en la solución de problemáticas de gran complejidad en áreas tales como visión por ordenador, procesamiento de lenguaje natural, juegos y muchas otras aplicaciones.

1.3.3. Técnicas de Procesamiento

1.3.3.1. Introducción al procesamiento imágenes

La importancia de la gestión de imágenes en la vigilancia acuática radica en su capacidad para automatizar y mejorar la recogida de información muy valiosa sobre los ecosistemas acuáticos y sobre las especies que los habitan, ya que, en el caso de los peces submarinos, el procesamiento de imágenes permite la extracción de datos relevantes a partir de imágenes o vídeos captados en el medio acuático, lo cual facilita la obtención de conjuntos de datos precisos y detallados [21].

En [22], el procesamiento de imágenes se compone de diversas etapas esenciales: la adquisición, el preprocesamiento, la segmentación, la extracción de características, la clasificación, la detección y localización. Cada etapa tiene una función específica que mejora la calidad de la imagen, identifica objetos de interés y clasifica los utilizando técnicas de autoaprendizaje como las redes neuronales.

De acuerdo con [23], presentaron un ejemplo de aplicación del procesamiento de imágenes en acuicultura mediante un sistema de visión artificial (MVS), con el que se pueden adquirir imágenes de peces en tiempo real y realizar análisis estadísticos para estimar su masa y tamaño de manera no invasiva. Asimismo, las técnicas de detección y clasificación de especies de peces permiten identificar y diferenciar los distintos tipos de peces presentes en el cultivo.

En el ámbito de la acuicultura, clasificar los peces por especies es un aspecto crucial para el seguimiento y gestión eficaces de los cultivos. Para afrontar este reto, se han utilizado técnicas de procesamiento de imágenes, así como redes neuronales convolucionales. Estos enfoques permiten identificar automáticamente especies de peces partiendo de imágenes de cámaras ópticas, lo que puede mejorar notablemente la eficacia y la precisión del proceso de clasificación [24].

La utilización del aprendizaje profundo y la transferencia de conocimiento a partir de modelos previamente entrenados de conjuntos de datos generales como ImageNet reducen la cantidad de

datos específicos del dominio necesarios para el entrenamiento de los modelos, la cual facilita el desarrollo de sistemas de clasificación más efectivos y menos costosos [24].

Estas tecnologías avanzadas podrán ayudar a los acuicultores a obtener información detallada sobre la composición por especies de sus cultivos, lo que puede influir sobre la toma de decisiones, sobre el control de las poblaciones y sobre la aplicación de prácticas acuícolas sostenibles.

1.3.3.2. Detección de bordes

La detección de bordes es una importante técnica en el análisis de imágenes que es utilizada para identificar y destacar los límites o contornos de objetos y estructuras presentes en una imagen, esto consiste en hallar puntos en la imagen donde existen cambios bruscos de intensidad o color, lo que indica un cambio significativo de la estructura o forma de los objetos. Estos cambios de intensidad suelen corresponder con los bordes del objeto de la imagen [25].

En [26], señalaron algunos de los primeros intentos de emplear técnicas de visión por ordenador para medir las longitudes de peces. Se utilizaron cámaras para captar imágenes individuales de los peces a su paso por una cinta transportadora.

Para estimar la longitud real del pez se aplicaron técnicas de detección de bordes, tales como la identificación de características como la cabeza y la cola. Aunque se indica que se utilizaron métodos de detección de bordes, sin embargo, no se especifican algoritmos concretos al respecto en el artículo.

1.3.3.3. Segmentación de imágenes

En el trabajo [27], describen la segmentación de imágenes como técnica de gestión de imágenes y visión por ordenador utilizada para dividir una imagen en regiones o segmentos significativos y diferenciados. Su objetivo es identificar y etiquetar píxeles o regiones que pertenezcan a objetos, fondos u otras estructuras de interés en la imagen. La segmentación de imágenes resulta fundamental en numerosas aplicaciones, tales como detección de objetos, reconocimiento de patrones, seguimiento de objetos en movimiento, análisis médico, etc.

En [28], se indica que se utilizó YOLOv5 como columna vertebral y objeto de detección combinado con un cabezal de segmentación semántica para detectar y segmentar peces en tiempo real. Se utiliza la técnica de segmentación semántica para identificar y clasificar cada pixel de la imagen de acuerdo a la clase a la que pertenece, en este caso, segmentar las áreas correspondientes a los peces en la imagen.

Al combinar ambas técnicas, se logra una alta precisión tanto en la detección de objetos como en la segmentación de los peces en tiempo real. Estas técnicas son fundamentales para el desarrollo de la piscicultura "inteligente" y el monitoreo preciso de los peces en la industria acuícola.

1.3.4. Herramientas de Desarrollo

1.3.4.1. Python

Su naturaleza de framework a alto nivel posibilita un enfoque de programación enfocado en objetos simple pero efectivo, que simplifica la implementación de algoritmos y modelos de aprendizaje automático necesarios para realizar estimaciones precisas del crecimiento de la tilapia. Tanto el intérprete de Python como su amplia colección de estándares que se encuentran de manera libre en el sitio web de Python, <http://www.python.org/>, y se pueden distribuir gratuitamente [29].

1.3.4.2. Google Colab

La plataforma ofrece a los desarrolladores la posibilidad para escribir y ejecutar código Python a través de un navegador web, sin necesidad de realizar complicadas configuraciones. Una notable ventaja es el libre acceso a las GPU, que permite acelerar significativamente el entrenamiento de diferentes modelos de aprendizaje automático y posibilitar grandes conjuntos de datos. Se puede acceder gratuitamente a Colab desde el sitio web <https://colab.research.google.com/notebooks/intro.ipynb> [30].

1.3.4.3. Figma

Figma se presenta como herramienta esencial en el sector del diseño de maquetas. La posibilidad que ofrece Figma de gestionar distintas versiones de los diseños es fundamental en el proceso de desarrollo, ya que facilita el control y el seguimiento de las maquetas cuando se modifica un componente. De este modo, se garantiza una gestión eficaz del cambio y se garantiza la coherencia del diseño a lo largo de todo el proyecto.

Figma resulta una herramienta esencial por su enfoque en el trabajo colaborativo, la gestión de versiones y la capacidad de implementar y rediseñar componentes de forma modular y escalable, pues su uso garantiza una interfaz de usuario coherente, eficiente y adaptable para monitorizar el crecimiento de la tilapia basándose en el aprendizaje automático, Figma se encuentra disponible gratuitamente en <https://www.figma.com/> [31].

1.4. Antecedentes Contextuales

En la búsqueda de optimizar la producción acuícola en la finca situada en el Sitio " Río Chico", Parroquia Bellamaría, Cantón Santa Rosa, Provincia de El Oro, surge la necesidad de introducir innovaciones tecnológicas que simplifiquen y mejoren el proceso de medición de las tilapias. Mediante el uso de machine learning como lo son las redes neuronales convolucionales y técnicas estadísticas como la regresión lineal para predecir el peso y longitud de las tilapias de una imagen. Este proyecto pondrá énfasis en la programación de una aplicación web diseñada para facilitar y agilizar la obtención de datos cruciales, como la longitud y peso de las tilapias, como se muestra en la tabla 3.

Tabla 3: Matriz de Procesos de Automatización

Etapa	Descripción	Técnica/Algoritmo	Herramienta/Tecnología
1	Captura de Imágenes	Fotografías de tilapias en el estanque	Cámara de seguridad
2	Procesamiento de Imágenes	identificación de tilapias y extracción de características	Redes Neuronales Convolucionales (CNN)
3	Extracción de Datos	Obtención de longitud y peso a partir de características de imagen	Regresión Lineal
4	Subida de Imágenes	Permite a los usuarios subir imágenes desde la aplicación web	Interfaz de Usuario Web
7	Desarrollo de la Aplicación Web	Creación de interfaz para la interacción con los usuarios	React como frontend y backend Python flask
8	Integración de Resultados	Presentación de datos, longitud y peso de las tilapias	Interfaz de usuario en la aplicación web

1.4.1. Ámbito de aplicación

En la actualidad, las técnicas de medición tradicionales invasivas aún se utilizan para medir peces, lo que produce un gran estrés al sacarlas del agua, por eso se buscan nuevas soluciones tecnológicas que optimicen y mejoren la eficiencia al estimar la longitud y peso de las tilapias.

Por eso, se desarrolló una aplicación que permita estimar la longitud y peso de una tilapia mediante imágenes para minimizar el uso de mediciones invasivos y agilizar los procesos en la acuicultura.

1.4.2. Establecimiento de requerimientos

El desarrollo de la aplicación para la estimación de longitud y peso de las tilapias requiere:

- Creación de un dataset de imágenes clasificadas en datos de práctica y validación.
- Desarrollo de un modelo de red neuronal convolucional entrenando con el dataset anteriormente creado.
- Desarrollo de un ejemplar de regresión lineal para la estimación del peso de la tilapia
- Función para capturar tilapias y visualizar las estimaciones esperadas del peso y la longitud de la tilapia.

CAPÍTULO II. DESARROLLO DEL PROTOTIPO

2.1. Definición del prototipo

El prototipo de esta aplicación de estimación del peso y longitud de las tilapias se enfocará en simplificar los procesos de medición, mediante el uso de redes neuronales convolucionales que analizara las características visuales de las imágenes, también se mostrarán las arquitecturas empleadas la misma que consta de dos etapas, la primera etapa abarca todo sobre el entrenamiento del modelo y la segunda el progreso de la aplicación web. A continuación, se detallan cada una de las etapas.

Como se puede observar en la Figura 7, se plantea el entrenamiento del modelo de red neuronal convolucional para el desarrollo del prototipo, en la cual se puede observar como primer punto la recolección de imágenes, las imágenes fueron tomadas en un estanque de tilapias de una finca ubicada en el Sitio “Río Chico”. Estas imágenes tomadas nos ayudarán a crear nuestro DataSet, utilizando LinkedAI que nos facilitó con el proceso de etiquetar las imágenes en una categoría, en este caso tilapias, después de haber finalizado con el proceso de etiquetar, se procedió a clasificar las imágenes en datos de entrenamiento y validación. Posteriormente, se realizó el entrenamiento utilizando Google Colab como entorno para poder ejecutar código Python, siendo este el lenguaje

utilizado para entrenar la red neuronal, como resultado del entrenamiento se obtiene el modelo que permite detectar a las tilapias para luego en la Figura 8 proceder a entrenar el modelo que permitirá estimar la longitud y peso de las tilapias.

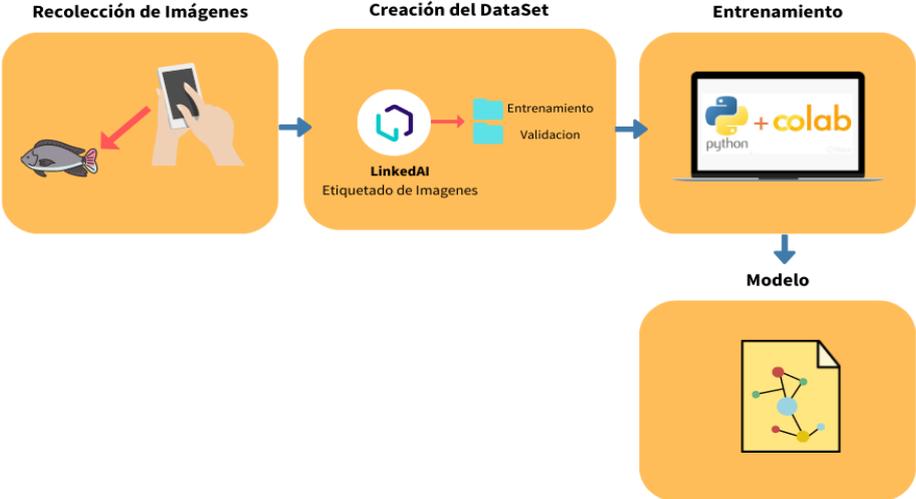


Figura 7: Entrenamiento de la red neuronal convolucional utilizando yolov8



Figura 8: Entrenamiento del modelo de estimación de tilapias

En la Figura 9, podemos percibir el diseño de la arquitectura de la aplicación web en la cual está compuesta por backend y frontend. El frontend es la interfaz con que el usuario podrá interactuar y podrá subir una imagen realizando una consulta a la API, para luego mostrar el resultado de la estimación de peso y longitud de la tilapia. El backend procesa la imagen capturada por la aplicación para procesarla de acuerdo a la configuración de entrada de la red neuronal, el modelo estima el peso y la longitud como resultado devuelve las estimaciones anteriormente mencionadas.

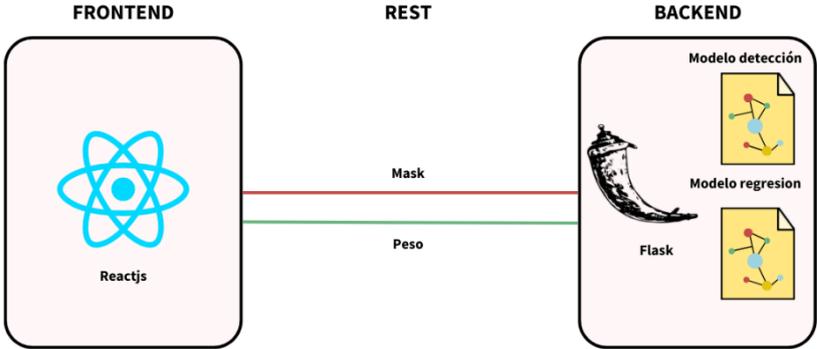


Figura 9: Arquitectura de desarrollo de la aplicación móvil

2.2. Metodología de desarrollo del prototipo

2.2.1. Enfoque, alcance y diseño de investigación

El enfoque será cuantitativo, debido a la aplicación para estimar el crecimiento de tilapias, se podrán recopilar datos de manera ágil y eficiente, además nos permitirá comparar las estimaciones obtenidas con métodos tradicionales de medición invasivos para evaluar la precisión de la aplicación.

El alcance del proyecto tendrá elementos exploratorios y descriptivos, ya que durante el desarrollo se requerirá una investigación exploratoria para comprender las mejores prácticas en el manejo de imágenes y describir y detallar los procesos y algoritmos utilizados para estimar el crecimiento de tilapias.

Finalmente, el diseño de la investigación será cuasi-experimental debido al trabajo está destinado a ser usado en un grupo de tilapias, los cuales determinaran el nivel de confiabilidad en la estimación de longitud y peso de la aplicación.

2.2.2. Unidades de análisis

Población (universo)

La población escogida estará compuesta por todas las tilapias dentro de la piscina.

Muestra

Contando con un total de 1000 imágenes de las cuales se dividió en datos de entrenamiento y validación, teniendo el 70% de las imágenes para entrenar y el 30% restante para validar el modelo. Esta información se ve reflejada en la tabla 4.

Tabla 4: Datos de la dataset del entrenamiento y pruebas

	Cantidad de Imágenes
Entrenamiento (70%)	700
Validación (30%)	300
Total	1000

2.2.3. Técnicas e instrumentos de recopilación de datos

En la Tabla 5 se presentan las técnicas metodológicas que se emplearán en el desarrollo del proyecto, específicamente la observación y el análisis de documentos o datos.

Tabla 5: Técnicas e instrumentos de recopilación de datos

Técnica	Instrumento
Observación	Cámara o dispositivo móvil para la toma de fotos de las tilapias Cuaderno de campo o hojas de registro
Análisis de documentos o datos o procesos	Acceso a documentos científicos y literatura relevante

2.2.4. Técnicas de procesamiento de datos para la obtención de resultados

Las técnicas de tratamiento de datos utilizadas para obtener los resultados de nuestro proyecto fueron el uso de Deep learning debido al uso de redes neuronales convolucionales que son muy eficaces en el reconocimiento de patrones en imágenes. Estas redes se pueden utilizar para extraer características relevantes de imágenes de tilapias y luego utilizar dichas características para predecir su tamaño o crecimiento.

2.2.5. Metodología o métodos específicos

Metodología para la construcción de la Red Neuronal Convolucionales

La metodología para entrenar a la red neuronal consta de las siguientes fases:

- La primera fase consta de crear un dataset con las imágenes tomadas para posteriormente realizar su etiquetado con la ayuda de LinkedAI. Después se aplicó técnicas de aprendizaje profundo para entrenar el modelo, esto con la utilización del entorno de Google Colab.
- Una vez completado el entrenamiento, se procede a realizar las pruebas de ejecución y testeo, obteniendo las métricas para evaluar posteriormente el modelo.
- Por último, se procede a utilizar el modelo en la aplicación. En la figura 10 se puede observar a mayor detalle las fases que se seguirán en la metodología.



Figura 10: Metodología para la construcción de la Red Neuronal

Metodología para el desarrollo de la aplicación

La metodología utilizada para el desarrollo de la aplicación será la Metodología XP porque se centra en la colaboración, la simplicidad y la flexibilidad y es fácilmente adaptable a las circunstancias. Dicha metodología consta de 4 fases que se observan en la Figura 11.



Figura 11: Fases de la metodología XP

2.2.6. Herramientas y/o Materiales

La Tabla 6 detalla las herramientas y materiales esenciales que se utilizarán para ejecutar las tareas de manera eficiente y lograr los objetivos propuestos.

Tabla 6: Herramientas y/o Materiales

HERRAMIENTAS	MATERIALES
Hardware	Cámara de Seguridad Computadora personal Memoria para almacenamiento de Datos
Software	Google Colab Bibliotecas de Procesamiento de Imágenes (opencv 4.8, numpy 1.24) Bibliotecas de Machine Learning (ultralytic 8.19 , joblib 1.3.2) Excel LinkedAI Visual Studio Code (1.86)
Lenguaje de programación	Python (3.11) JavaScript
Datos	Dataset de imágenes de Tilapias

2.3. Desarrollo del prototipo

Para el desarrollo del prototipo, utilizamos la metodología para la red neuronal de la Figura 10, que se utilizará para detectar tilapias, y otra destinada a desarrollar la aplicación web de la Figura 11.

2.3.1. Metodología para la construcción de la Red Neuronal Convolucionales

La metodología para desarrollar la red neuronal tiene varias fases, como se observa en la Figura 10. En este apartado se detallarán mejor las fases seguidas para obtener un modelo que permita estimar la longitud y peso de las tilapias.

FASE 1: ELABORACIÓN DEL DATASET

Para la elaboración del Dataset comprende la captura de imágenes de tilapias donde se recolectaron un total de 1000 imágenes, además de la obtención de sus pesos y longitud utilizando herramientas de medición. La recolección de datos se lo realizo en la finca ubicada en el Sitio “Río Chico” donde se encontraban cerca de 150 tilapias.

La figura 12 muestra el diagrama de flujo del proceso de recogida de datos.



Figura 12: Diagrama de flujo de la recolección de datos

Selección de las tilapias: Para la selección de la tilapia se utilizó una atarraya separándolas del estanque y depositándolas en un tanque.

Captura de Imagen: La captura de imagen se la realizo con la cámara de seguridad con una aproximación de 3.50 metros. Esto puede verse en la figura 13.



Figura 13: Captura de imagen de tilapias

Registro de Peso y Longitud. Luego de la captura de tilapias, se procede a realizar las mediciones de longitud y peso utilizando las herramientas adecuadas de medición, para finalmente devolver a la tilapia al tanque. La recolección de datos de las tilapias tomaba alrededor de unos 2 a 3 minutos.

Después de haber recolectado las muestras necesarias, se procedió al procesamiento de imágenes que comprende las etapas indicadas en el diagrama de flujo que se muestra en la figura 14.

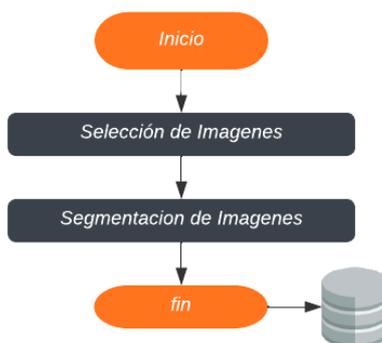


Figura 14: Diagrama de flujo del procesamiento de imágenes

Selección de imágenes: La selección de imágenes es una parte importante en la creación del dataset debido a que debe cumplir ciertos factores para formar parte del dataset, estos factores a tomar en cuenta son:

- Deben aparecer todas las tilapias.
- Se debe visualizar en lo posible a una buena calidad.

Segmentación de Imágenes

Para la segmentación de las imágenes se hizo uso de la plataforma de etiquetado LinkedAI. La herramienta utilizada para poder realizar la segmentación fue la herramienta lápiz y auto segmentación, la figura 15 presenta el resultado obtenido en la segmentación mediante la

plataforma LinkedAI. El tiempo para el etiquetado de la imagen varía en un intervalo de 45 segundos a 5 minutos dependiendo de la imagen de la tilapia.



Figura 15: Segmentación de la tilapia usando LinkedAI

La plataforma LinkedAI nos devuelve como resultado de las imágenes etiquetadas en un archivo formato JSON con la información de la imagen.

Con los datos ya etiquetados se procede a realizar la creación del dataset teniendo esta estructura:

- Total, de imágenes del dataset 1000 en formato PNG.
- Datos de entrenamiento (700 imágenes)
- Datos de validación (300 imágenes)

FASE II: DESARROLLO Y ENTRENAMIENTO DEL MODELO DE RED NEURONAL Y REGRESIÓN LINEAL

La construcción y entrenamiento de ambos modelos se los realizo en el entorno de pycharm y Google colab, para el modelo de segmentación de las tilapias se utilizó un modelo pre-entrenado de yolov8 debido a que estos modelos tienden a tener un buen rendimiento y nos ahorra el tiempo de entrenar de cero toda la red neuronal.

Modelo De Segmentación

Para utilizar un modelo yolov8 pre-entrenado, se debe seguir el formato establecido, tal y como se muestra en la Figura 16.

```
<class-index> <x1> <y1> <x2> <y2> ... <xn> <yn>
```

Figura 16: Formato yolov8 - Ultralytics

Se procedió a convertir las máscaras de nuestro dataset al formato adecuado de yolov8 como se ve en la figura 17.

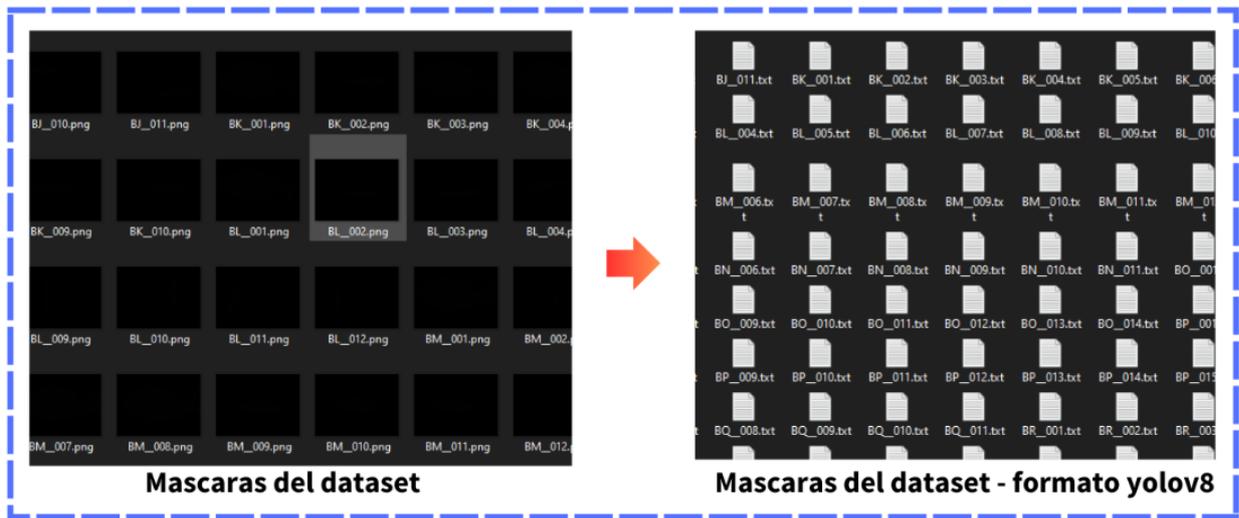


Figura 17: Conversión de las máscaras al formato aceptado por yolov8

Tras realizar la conversión y disponer de las máscaras en el formato adecuado, se procedió al entrenamiento de la red neuronal de segmentación de tilapias mediante un modelo pre-entrenado específicamente para la segmentación yolov8, a continuación, se observa el código en la figura 18 del entrenamiento.

```

✓ [10] from google.colab import drive
1 s drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount(

✓ [1] %pip install ultralytics
16 s import ultralytics
ultralytics.checks()

Ultralytics YOLOv8.0.182 🚀 Python-3.10.12 torch-2.0.1+cu118 CPU (Intel Xeon 2.20GHz)
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 26.3/107.7 GB disk)

✓ [2] from ultralytics import YOLO

✓ [11] model = YOLO('/content/drive/MyDrive/TilapiaProyecto/yolov8n-seg.pt')
0 s
Downloading https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8n-seg.pt
100%|██████████| 6.73M/6.73M [00:00<00:00, 62.8MB/s]

▶ results = model.train(data='/content/drive/MyDrive/TilapiaProyecto/config.yaml',
epochs=10, imgsz=640,
project='/content/drive/MyDrive/TilapiaProyecto/training_results')

```

Figura 18: Modelo de segmentación de Tilapias

Modelo de Regresión Lineal Múltiple

Para estimar el peso de una tilapia, se llevó a cabo un modelo de regresión lineal múltiple, donde se utilizaron las máscaras de segmentación convertidas para calcular el área, el perímetro y la relación de aspecto de la silueta de la tilapia, además del peso, la anchura y la longitud, como se muestra en la Figura 19.

```
coordinates = []
with open(filename, 'r') as file:
    lines = file.readlines()
    for line in lines:
        parts = line.strip().split()
        if len(parts) >= 2:
            # Ignora el primer número que es la categoría y convierte las coordenadas
            coordinates.extend([(float(parts[i]), float(parts[i + 1])) for i in range(1, len(parts), 2)])
return coordinates
def get_area(points):
    n = len(points)
    area = 0.0
    for i in range(n):
        x1, y1 = points[i]
        x2, y2 = points[(i + 1) % n] # El último punto se conecta con el primer punto
        area += (x1 * y2 - x2 * y1)
    area = abs(area) / 2.0
    return area
def get_perimeter(points):
    perimeter = 0.0
    n = len(points)
    for i in range(n):
        x1, y1 = points[i]
        x2, y2 = points[(i + 1) % n] # El último punto se conecta con el primer punto
        distancia = math.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
        perimeter += distancia
    return perimeter
def get_aspect_ratio(points):
    if bool(points):
        # Encontrar los valores mínimos y máximos en las coordenadas x e y
        min_x = min(points, key=lambda punto: punto[0])[0]
        max_x = max(points, key=lambda punto: punto[0])[0]
        min_y = min(points, key=lambda punto: punto[1])[1]
        max_y = max(points, key=lambda punto: punto[1])[1]
        # Calcular la longitud de los lados horizontal y vertical
        horizontal_length = max_x - min_x
        vertical_length = max_y - min_y
        # Calcular la relación de aspecto
        aspect_ratio = vertical_length / horizontal_length
```

Figura 19: Código para la obtención del área, perímetro y relación de aspecto

A continuación, como puede verse en la figura 20, se presenta el modelo de regresión lineal múltiple que aplica los métodos anteriores de área, perímetro y relación de aspecto.

```

data = np.array([])
for f in files:
    formatName = f[:2]
    coordinates = parse_polygon(train_labels + f)

    #print(f, coordinates)
    area = get_area(coordinates)
    perimeter = get_perimeter(coordinates)
    relation = get_aspect_ratio(coordinates)
    filtered_rows = df[df['format'].str.contains(formatName, na=False)]
    row_values = []
    row_values.append(area)
    row_values.append(perimeter)
    row_values.append(relation)
    for index, fila in filtered_rows.iterrows():
        row_values.extend(fila.values[2:]) # Ignoramos la primera columna
    array_format = np.array(row_values)
    if data.size == 0:
        data = array_format
    else:
        data = np.vstack((data, array_format))
#print(data)
X = data[:, :-1] # Las primeras columnas con las características
y = data[:, -1] # La última columna es el peso

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

modelo = LinearRegression()
modelo.fit(X_train, y_train)

y_pred = modelo.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Error cuadrático medio (MSE):", mse)
print("Coeficiente de determinación (R^2):", r2)

```

Figura 20: Modelo de Regresión Lineal

FASE III: EJECUCIÓN DE LA RED NEURONAL Y REGRESIÓN LINEAL

Modelo de Segmentación

El entrenamiento del modelo se lo puede observar en la figura 21, con pesos pre-entrenados algunos de los Hiperparámetros que se utilizó fueron:

- img: indica el tamaño que usa el modelo para la imagen.
- epochs: especifica el número de épocas o iteraciones completas que debe cumplir el entrenamiento.
- data: hace el llamado al archivo con extensión. yaml que contiene las rutas donde se encuentra la data para el entrenamiento.

```
[ ] model = YOLO('/content/drive/MyDrive/TilapiaProyecto/yolov8n-seg.pt')

Downloading https://github.com/ultralytics/assets/releases/download/v8.0.0/yolov8n-seg.pt to '/content/drive/MyDrive/TilapiaProyecto/yolov8n-seg.pt'...
100% ██████████ 6.73M/6.73M [00:00<00:00, 62.8MB/s]

[ ] results = model.train(data='/content/drive/MyDrive/TilapiaProyecto/config.yaml',
epochs=10, imgsz=640,
project='/content/drive/MyDrive/TilapiaProyecto/training_results')

Epoch  GPU_mem  box_loss  seg_loss  cls_loss  dfl_loss  Instances  Size  28/28  [09:01<00:00, 19.34s/it]
2/10   0G      1.329    1.434    2.219    1.722     16         640: 100% ██████████
   Class  Images  Instances  Box(P   R   mAP50  mAP50-95)  Mask(P   R   mAP50  mAP50-95): 100% ██████████ 6/6 [01:22<00:00, 13.69s/it]
   all    192     192     0.551   0.573   0.611   0.318   0.63   0.583   0.641   0.369

Epoch  GPU_mem  box_loss  seg_loss  cls_loss  dfl_loss  Instances  Size  28/28  [08:47<00:00, 18.84s/it]
3/10   0G      1.287    1.363    1.938    1.658     16         640: 100% ██████████
   Class  Images  Instances  Box(P   R   mAP50  mAP50-95)  Mask(P   R   mAP50  mAP50-95): 100% ██████████ 6/6 [01:21<00:00, 13.57s/it]
   all    192     192     0.555   0.396   0.455   0.155   0.667   0.411   0.498   0.223

Epoch  GPU_mem  box_loss  seg_loss  cls_loss  dfl_loss  Instances  Size  28/28  [08:50<00:00, 18.96s/it]
4/10   0G      1.16     1.276    1.698    1.614     16         640: 100% ██████████
   Class  Images  Instances  Box(P   R   mAP50  mAP50-95)  Mask(P   R   mAP50  mAP50-95): 100% ██████████ 6/6 [01:20<00:00, 13.37s/it]
   all    192     192     0.711   0.714   0.785   0.409   0.69   0.745   0.786   0.515

Epoch  GPU_mem  box_loss  seg_loss  cls_loss  dfl_loss  Instances  Size  28/28  [08:51<00:00, 18.99s/it]
5/10   0G      1.224    1.186    1.551    1.636     16         640: 100% ██████████
   Class  Images  Instances  Box(P   R   mAP50  mAP50-95)  Mask(P   R   mAP50  mAP50-95): 100% ██████████ 6/6 [01:17<00:00, 12.88s/it]
   all    192     192     0.817   0.86   0.893   0.488   0.832   0.877   0.89   0.619

Epoch  GPU_mem  box_loss  seg_loss  cls_loss  dfl_loss  Instances  Size  28/28  [08:53<00:00, 19.04s/it]
6/10   0G      1.088    1.122    1.351    1.479     16         640: 100% ██████████
   Class  Images  Instances  Box(P   R   mAP50  mAP50-95)  Mask(P   R   mAP50  mAP50-95): 100% ██████████ 6/6 [01:20<00:00, 13.45s/it]
   all    192     192     0.843   0.758   0.875   0.56   0.809   0.773   0.861   0.647

Epoch  GPU_mem  box_loss  seg_loss  cls_loss  dfl_loss  Instances  Size  28/28  [08:52<00:00, 19.03s/it]
7/10   0G      1.041    1.052    1.187    1.444     20         640: 100% ██████████
   Class  Images  Instances  Box(P   R   mAP50  mAP50-95)  Mask(P   R   mAP50  mAP50-95): 100% ██████████ 6/6 [01:16<00:00, 12.73s/it]
   all    192     192     0.872   0.896   0.943   0.584   0.853   0.905   0.933   0.644
```

Figura 21: Entrenamiento del modelo de segmentación de Tilapias

Modelo de Regresión Lineal

Tras desarrollar el modelo de regresión lineal, se realizó el entrenamiento tal y como se muestra en la Figura 22.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

modelo = LinearRegression()
modelo.fit(X_train, y_train)

y_pred = modelo.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Error cuadrático medio (MSE):", mse)
print("Coeficiente de determinación (R^2):", r2)

Error cuadrático medio (MSE): 4.428372887688374e-05
Coeficiente de determinación (R^2): 0.9362438982771799
```

Figura 22: Entrenamiento del Modelo de Regresión Lineal

2.3.1. Metodología XP para el desarrollo de la aplicación web

La metodología XP se utilizará en el desarrollo de la aplicación web debido a su ciclo corto de entrega funcional, lo que nos permitirá seguir mejorando.

FASE I: PLANIFICACIÓN

Definición de roles

Para el desarrollo de la aplicación se establecen los siguientes roles. Éstos se detallan en la siguiente tabla 7.

Tabla 7: Definición de roles y responsables

Roles	Responsable
Desarrollador	Vásquez Alejandro, Galo Armijos
Encargados del entrenamiento de la red neuronal convolucional y del modelo de estimación de peso y longitud	Vásquez Alejandro, Galo Armijos
Encargados del testeo de la red neuronal convolucional (segmentación) y estimación de características morfológicas de la tilapia	Vásquez Alejandro, Galo Armijos
Cliente	Finca ubicada en el Sitio “Río Chico” Camping House

Historias de Usuario

Las historias de usuario utilizadas en el desarrollo de la aplicación móvil pueden verse en detalle en el Anexo 3.

Análisis del sistema

- **Plan de publicaciones**

En las reuniones de planificación se definió la prioridad de las historias de usuario de forma independiente que incluyen sus actividades en las 3 iteraciones. Estos resultados se detallan en la tabla 8:

Tabla 8: Plan de publicaciones

Historia de Usuario	Nombre de Historia de Usuario	Prioridad	Esfuerzo	Actividades
1	Mostrar información al usuario	Baja	1	<ul style="list-style-type: none"> ✓ Interfaz de Usuario ✓ Información del uso de la aplicación
2	Ejecutar segmentación/predicir	Alta	2	<ul style="list-style-type: none"> ✓ Interfaz de Usuario ✓ Lectura de datos y procesado de la tilapia ✓ Subir imagen de las tilapias para ejecutar la predicción del peso y longitud. ✓ Comprobación de resultados en la interfaz de usuario
3	Visualizar los resultados	Alta	2	<ul style="list-style-type: none"> ✓ Comprobación de resultados en la interfaz de usuario mediante la estimación del procesado de la imagen.

- **Plan de Iteraciones**

La duración de las iteraciones se establece, en función de lo que se necesite previamente al desarrollo de la aplicación, la tabla 9 lo indica con más detalle.

Tabla 9: Plan de duración de iteraciones

	Iteración 1	Iteración 2	Iteración 3
Historia de Usuario	2	2	1
Semanas	2 semanas	3 semanas	1 semanas
Horas semanales	7 h	7 h	7 h
Total, de horas x Semana	14 h	21 h	7 h

FASE II: DISEÑO

Se elaboró un prototipo del diseño de la aplicación utilizando la herramienta Figma para visualizar de una mejor manera el diseño de la aplicación.

- **Pantalla de Inicio**

La Figura 23 presenta una pantalla inicial de la aplicación con su respectivo nombre.



Figura 23: Pantalla de Inicio

- **Pantalla de resultados**

La figura 24 presenta resultados obtenidos de la segmentación de la imagen con su respectiva estimación de pesos y un botón para volver a la pantalla de captura para realizar de nuevo el proceso.



Figura 24: Pantalla de resultados

FASE III: DESARROLLO

En el proceso de elaboración de la página web se utilizó visual studio code como editor de código, además de usar para la parte del frontend un framework llamado react, para la parte del backend se utilizó otro framework llamado flask debido a que agiliza el desarrollo de aplicaciones que no requieren muchas extensiones. Para la parte de la codificación se utilizaron 2 lenguajes: JavaScript y Python. En la figura 25 se nos muestra la función que nos permite enviar un archivo a través de una solicitud POST a una URL específica, procesar la respuesta JSON obtenida y actualizar el estado de componentes en la interfaz de usuario.

```
const handleSubmit = async (event) => {
  event.preventDefault();
  const formData = new FormData();
  formData.append('file', file);
  try {
    const response = await fetch('http://127.0.0.1:5000/predict', {
      method: 'POST',
      body: formData,
    });
    const data = await response.json();
    // Suponiendo que la respuesta contiene la máscara en base64
    setMaskBase64(data.mask_base64);
    setPeso(data.peso)
  } catch (error) {
    console.error('Error al realizar la predicción:', error);
  }
};
```

Figura 25. Función para actualizar el estado de la interfaz con la máscara

En la siguiente figura 26 se configura una aplicación web que utiliza modelo YOLO para procesar imágenes y un modelo de regresión para realizar predicciones. La ruta '/predict' está diseñada para manejar solicitudes POST, esperando recibir archivos.

```
app = Flask(__name__)
CORS(app)
model_path = 'last.pt'
model = YOLO(model_path)
regre_model = joblib.load('predict_model.pkl')

@app.route(rule: '/predict', methods=['POST'])
def predict():
    if 'file' not in request.files:
        return jsonify({'error': 'No se ha enviado ningún archivo'})
```

Figura 26. Configuración de Flask que utiliza dos modelos. segmentación y regresión

2.4. Ejecución del prototipo

Modelo de Segmentación

Una vez entrenado el modelo se observa en la figura 26 las predicciones de las tilapias realizadas con los datos de validación, en donde se observa una buena segmentación de las tilapias.

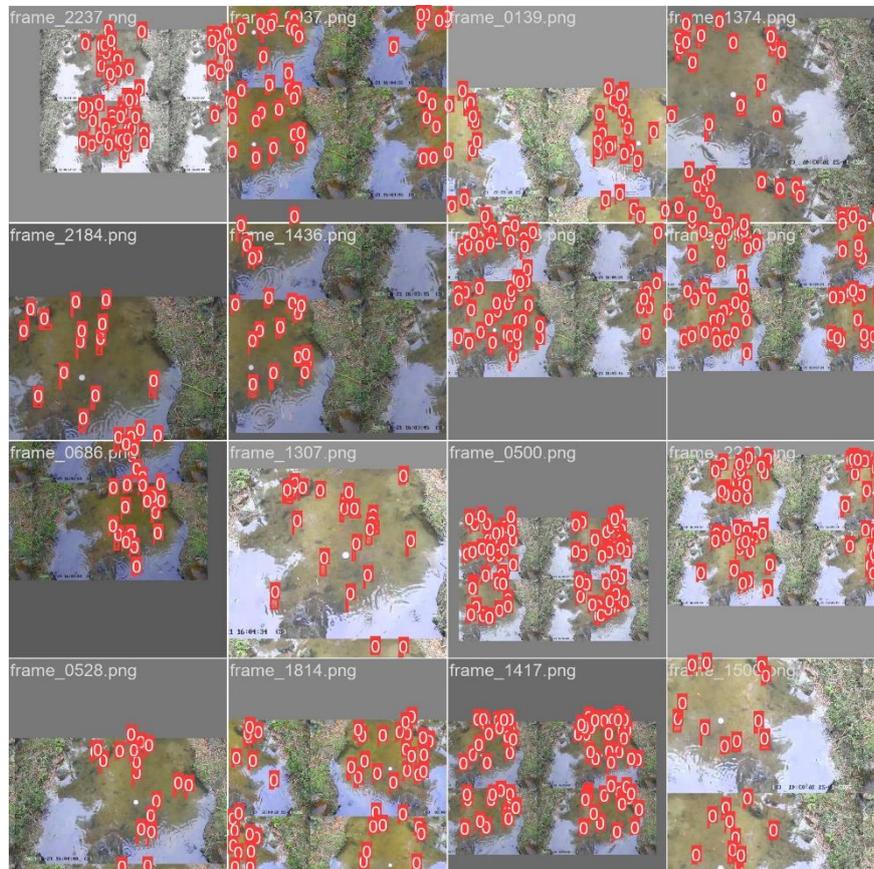


Figura 27: Predicciones del modelo de Segmentación de Tilapias

Modelo de Regresión Lineal Múltiple

Luego se procedió a estimar el peso de la tilapia mediante el modelo de regresión lineal múltiple. Para la predicción del peso de la tilapia, utilizamos la máscara de tilapia segmentada y posteriormente fue convertida al formato yolov8 donde utilizaremos los puntos de la máscara en el modelo a predecir, como se observa en la Figura 28.

```
[8] point = parse_polygon('/content/drive/MyDrive/TilapiaProyecto/pruebasR/CP__007.txt')
    area = get_area(point)
    perimeter = get_perimeter(point)
    relation = get_aspect_ratio(point)
    print("Prueba ")
    valores = np.array([[area, perimeter, relation, 2.2, 4.7, 14.8]])

    peso = modelo.predict(valores)
    print("Peso" , peso)

Prueba
Peso [0.07029361]
```

```
from google.colab.patches import cv2_imshow
import cv2

image_path_results = '/content/drive/MyDrive/TilapiaProyecto/segmentacionSilueta/CP__007.png'
img = cv2.imread(image_path_results, 500)
cv2_imshow(img)
```



Figura 28: Predicción del Peso Tilapia del Modelo de Regresión Lineal Múltiple

Para la ejecución del proyecto se ejecutó tanto el servidor como la página web desde la terminal con el comando **npm run dev** en la parte del frontend como se muestra en la figura 29.

```
PS C:\Users\Alexa\OneDrive\Escritorio\tilapiaproj-master\tilapia-project> npm run dev

> tilapia-project@0.0.0 dev
> vite

VITE v5.0.2 ready in 533 ms
  → Local:   http://localhost:5173/
  → Network: use --host to expose
  → press h + enter to show help
```

Figura 29. Comando para ejecutar la página web

Para la parte del backend el comando utilizado para ejecutar el servidor es el siguiente python.exe C:\Users\Alexa\OneDrive\Escritorio\tilapiaproj-master\server\main.py como se muestra en la figura 30.

```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
C:\Users\Alexa\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:318: UserWarning: Trying to u
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
```

Figura 30. Comando para ejecutar el servidor

Seguidamente, la Figura 31 presenta la página web en la que se puede predecir el peso y el tamaño de las tilapias del sitio denominado "Camping House".

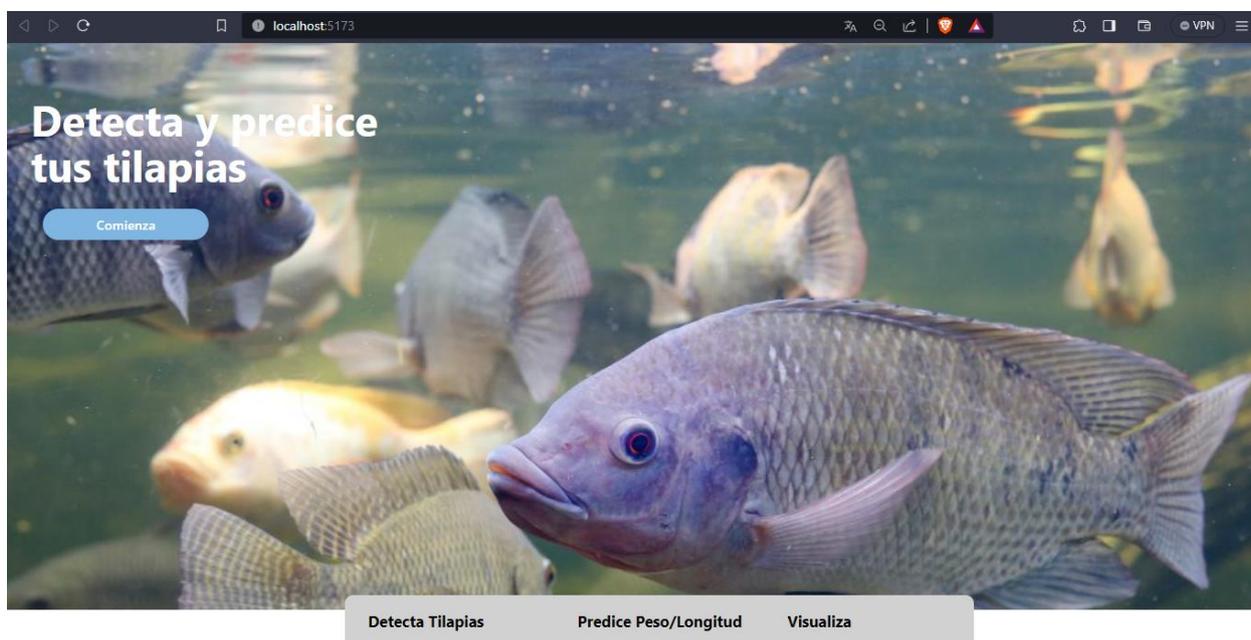


Figura 31. Pantalla principal de la página web

En la figura 32 se observa la segmentación de las tilapias que existen en la imagen para posteriormente predecir su peso de cada tilapia, cada color indica la tilapia a la que pertenece su peso.



Figura 32. Pantalla de predicción y segmentación de las tilapias

La Figura 33 presenta las predicciones del peso de la tilapia segmentada.

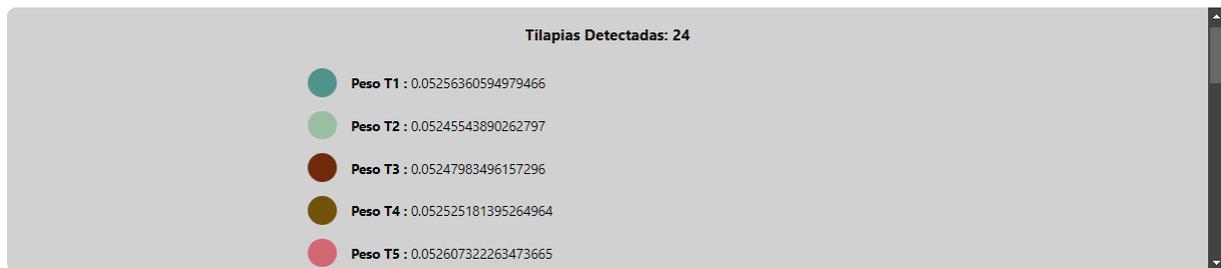


Figura 33. Pantalla de resultados de las tilapias segmentadas

CAPÍTULO III. EVALUACIÓN DEL PROTOTIPO

3.1. Plan de evaluación

Plan de evaluación

Para la evaluación, se llevarán a cabo pruebas tanto para la red neuronal como para la aplicación web. Para la evaluación de la red neuronal se utilizarán diversas métricas de detección y precisión; para la de la aplicación web, la norma ISO 9126.

Objetivos

Evaluar la aplicación web de predicción del crecimiento de tilapias mediante una imagen específicamente ubicado en el sitio “Rio Chico” aplicando métricas y normas de calidad para mejorar en la aplicación y experiencia de usuario.

3.1.1 Planificación

Tabla 10: Diagrama de Gantt

Actividades	Semana 10	Semana 11	Semana 12	Semana 13
Definición de objetivos, técnicas para evaluar el modelo				
Evaluar funcionabilidad y fiabilidad				
Evaluar usabilidad y eficiencia				
Documentar resultados				

Semana 10	
<p>Actividades:</p> <ul style="list-style-type: none"> • Establecer objetivos del plan • Elección de técnicas para la evaluación del modelo • Elaborar matriz de confusión para • calcular métricas de exactitud y precisión 	<p>Resultados:</p> <ul style="list-style-type: none"> • Elaborar un cronograma • Elaborar una conclusión acerca de efectividad del modelo.

Semana 11	
<p>Actividades:</p> <ul style="list-style-type: none"> • Evaluar la funcionalidad de la aplicación web la velocidad de respuesta de la página web • Evaluar la fiabilidad de las predicciones de peso y longitud de las tilapias mediante la medición 	<p>Resultados:</p> <ul style="list-style-type: none"> • Comprobar los tiempos de respuesta de la página web. • Comprobar los resultados de predicciones de la página web con los resultados de la medición tradicional.

Semana 12	
<p>Actividades:</p> <ul style="list-style-type: none"> • Evaluación de la usabilidad de la aplicación web a través de los comentarios de los usuarios • Evaluar la eficacia de la aplicación en términos de tiempo 	<p>Resultados:</p> <ul style="list-style-type: none"> • Comprobación del grado de satisfacción de los usuarios • Análisis de predicción de manera tradicional en comparación a las predicciones de la página web.

Semana 13	
<p>Actividades:</p> <ul style="list-style-type: none"> • Documentar todos los resultados obtenidos del plan de evaluación en el capítulo III. 	<p>Resultados:</p> <ul style="list-style-type: none"> • Comprobar hipótesis y determinar un análisis

3.1.2 Herramientas y Técnicas

Matriz de confusión y métricas para evaluar el modelo de detección y segmentación de tilapias.

El uso de una matriz de confusión nos proporciona una representación visual del rendimiento del modelo, ya que la salida de una red neuronal son resultados binarios entre 0 y 1. La tabla 11 muestra la estructura de una matriz de confusión.

Tabla 11: Matriz de Confusión

		Valor de Predicción	
		Objeto presente	Objeto no presente
Situación Real	Objeto presente	VP	FN
	Objeto no presente	FP	VN

A continuación, se muestran las métricas que se utilizarán para evaluar el modelo de segmentación:

Precisión

Indica la proporción de detecciones positivas que son realmente positivas. Una alta precisión significa menos falsos positivos.

- **Verdaderos Positivos (True Positives):** Estos casos son aquellos en los que el modelo ha detectado correctamente un objeto positivo.
- **Falsos Positivos (False Positives):** Representan casos en los que el modelo ha indicado que ha detectado un objeto, pero en realidad no hay ninguno.

$$\text{Precision} = \frac{\text{Verdaderos Positivos(VP)}}{\text{Verdaderos Positivos(VP)} + \text{Falsos positivos(FP)}}$$

Sensibilidad (Recall)

Es una métrica de evaluación en modelos de machine learning que muestra la proporción de instancias reales positivas que el modelo ha detectado con respecto al total de instancias reales positivas

$$\text{Precision} = \frac{\text{Verdaderos Positivos(VP)}}{\text{Verdaderos Positivos(VP)} + \text{Falsos Negativos(FN)}}$$

Índice de Intersección sobre Unión

Esta métrica es una medida integral que considera múltiples umbrales de IOU, proporcionando así una medida general del rendimiento del modelo desde 0.5 hasta 0.95 con un paso de 0.05. Esta métrica evalúa la capacidad del modelo para realizar detecciones precisas en un rango variado de umbrales de superposición entre las predicciones y las verdaderas instancias.

Métricas según norma ISO 9126 para evaluar aplicación web

Funcionabilidad

JMeter es una herramienta de código abierto que permite probar el rendimiento de aplicaciones web y otros servicios al simular múltiples usuarios, identificando así posibles problemas de rendimiento y vulnerabilidades bajo diferentes condiciones de carga.

Fiabilidad

Para evaluar esta métrica se compara los resultados de las predicciones que nos da la aplicación web con las mediciones tradicionales como se presenta en la tabla 12.

Tabla 12: Comparación de resultados entre aplicación web y mediciones tradicionales

Predicciones del modelo		Mediciones Tradicionales	
Peso	Longitud	Peso	Longitud

3.2. Resultados de la evaluación

3.2.1. Resultados de exactitud y precisión del modelo de segmentación

La Tabla 13 presenta los resultados derivados de la matriz de confusión empleada en la evaluación del funcionamiento de los modelos. La determinación del modelo se basó en la consideración de métricas como la exactitud y la precisión.

Tabla 13: Resultados de la Matriz de confusión

		Valor de Predicción	
		Objeto presente	Objeto no presente
Situación Real	Objeto presente	0.88	0.12
	Objeto no presente	0	1

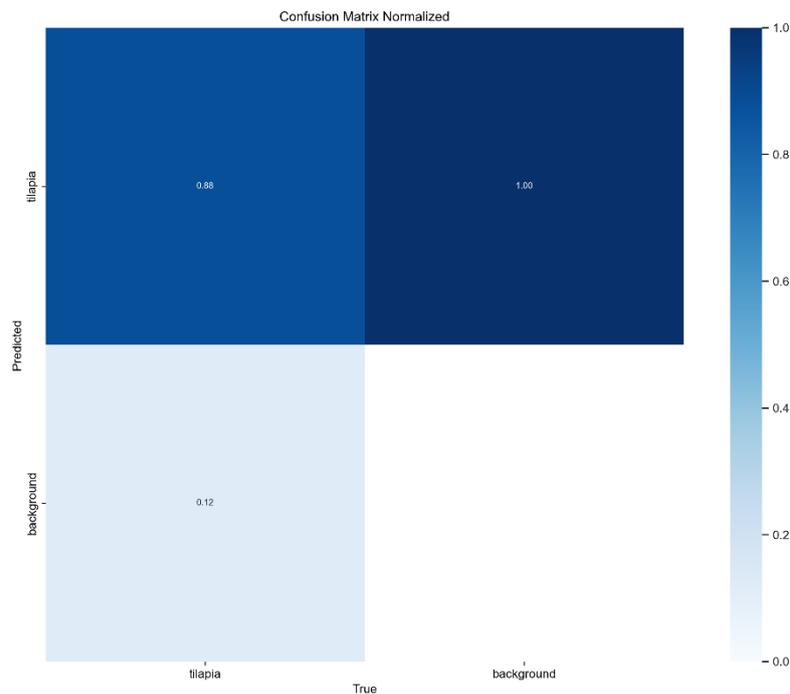


Figura 34: Matriz de confusión

Como se puede ver en la figura matriz de confusión muestra que el modelo tiene un buen rendimiento en la detección de tilapia. El modelo predijo correctamente que la imagen era una tilapia en el 88% de los casos. En el 12% de los casos, el modelo predijo incorrectamente que la imagen era una tilapia.

Además, durante el entrenamiento se obtuvo el mejor modelo con los siguientes parámetros como se observa en la tabla 14.

Tabla 14: Indicadores del entrenamiento

Indicadores	Valor
Conjunto de datos	Propio
Imágenes para entrenamiento	700
Imágenes para validación	300
Pesos pres-entrenados	YOLOv8n-seg
Numero de épocas	13
Dimensión de las imágenes	640 x 640

En la tabla 15 se nos muestra los resultados del entrenamiento como la precisión, recall y el Índice de Intersección sobre Unión y nos indica rendimiento sólido. La alta precisión señala que las predicciones positivas son confiables, mientras que la sensibilidad muestra que el modelo detecta efectivamente la mayoría de las instancias positivas. El Índice de Intersección sobre Unión (IoU) sugiere una superposición razonable entre las áreas predichas y las reales

Tabla 15: Resultados de métricas de entrenamiento

Métricas	Valor
Precisión	0.82108
sensibilidad	0.86381
Índice de Intersección sobre Unión	0.42243

Métricas según norma ISO 9126 para evaluar aplicación web

Funcionabilidad

Para probar la funcionabilidad de la página web se realizó pruebas utilizando la herramienta JMeter para confirmar que la respuesta adecuada a las peticiones generadas por los usuarios.

Se realizo 3 pruebas de carga con alrededor de 50,500 y 1000 usuarios manteniendo un tiempo de transiciones de 60 segundos como se puede ver en la siguiente tabla.

Tabla 16: Configuración de prueba de Carga JMeter.

#Pruebas	Numero de hilos(usuarios)	Bucle	Tiempo de transición(s)
1	50	1	60
2	200	1	60
3	500	1	60

Los resultados obtenidos de las 3 pruebas de cargas fueron los siguientes resultados como se muestra en las siguientes tablas.

Tabla 17: Prueba con 50 usuarios

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	50	3	2	11	2,05	0.000 %	0,85	0,21	0,17	251
TOTAL	50	3	2	11	2,05	0.000 %	0,85	0,21	0,17	251

Tabla 18: Prueba con 200 usuarios

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	200	4	2	10	2,22	0.000 %	3,33	0,82	0,67	251
TOTAL	200	4	2	10	2,22	0.000 %	3,33	0,82	0,67	251

Tabla 19: Prueba con 500 usuarios

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	500	3	2	11	2,21	0.000 %	8,33	2,04	1,68	251
TOTAL	500	3	2	11	2,21	0.000 %	8,33	2,04	1,68	251

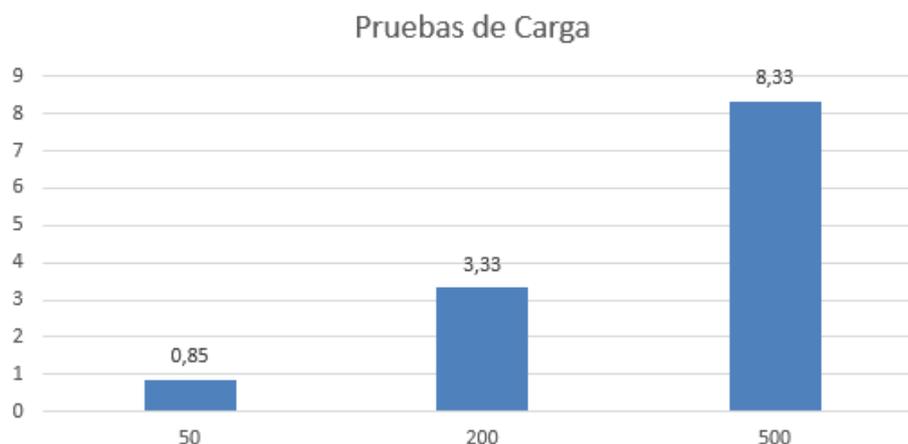


Figura 35: Prueba de rendimiento

Los resultados mostrados indican como la aplicación ha sido capaz de manejar las pruebas de carga sin experimentar fallos, además de que la tasa de peticiones por segundo ha ido aumentando conforme se le añadían más usuarios como se muestra en la Figura 34 lo cual es positivo ya que es capaz de manejar más tráfico.

Fiabilidad

Para la evaluación de fiabilidad se realizó una tabla para comparar las predicciones de la pagina web con las de las mediciones tradicionales realizadas a las tilapias esto con el objetivo de validar las predicciones.

La fiabilidad de los resultados se los realizo comparando las predicciones realizadas con las mediciones reales, desde la primera medición hasta la última medición manual realizada después de 3 meses esto se observa en las tablas 20 y 21.

Tabla 20: Resultados entre aplicación web y mediciones tradicionales primera comparación

Predicciones del modelo		Mediciones Tradicionales	
Peso(g)	Longitud	Peso (g)	Longitud
98	16.48	16	80
97.95	16.47	16	90
97.96	16.46	16	75
98.06	16.48	14	65
98.20	16.48	18	140
98.05	16.46	16,5	95
98.10	16.48	16,4	90
98.12	16.48	17,6	125

Tabla 21: Resultados entre aplicación web y mediciones tradicionales segunda comparación

Predicciones del modelo		Mediciones Tradicionales	
Peso(g)	Longitud	Peso (g)	Longitud
98	16.47	264	24
97.79	16.47	262	22,8
97.78	16.46	281	30
98.15	16.48	646	30
98.12	16.48	516	30

Según los resultados, se puede inferir que las predicciones realizadas para el peso y longitud de las tilapias en la primera comparación se encuentran cercanas a las mediciones reales. Sin embargo, en la segunda comparación, las predicciones del modelo distan considerablemente de las mediciones tradicionales. Aunque las predicciones para la longitud y peso en la primera instancia son consistentes, en la segunda comparación se evidencia una discrepancia notable. Este fenómeno podría atribuirse a la falta de variabilidad en los datos capturados para la longitud y peso en la segunda serie de mediciones, afectando la capacidad del modelo para realizar predicciones precisas.

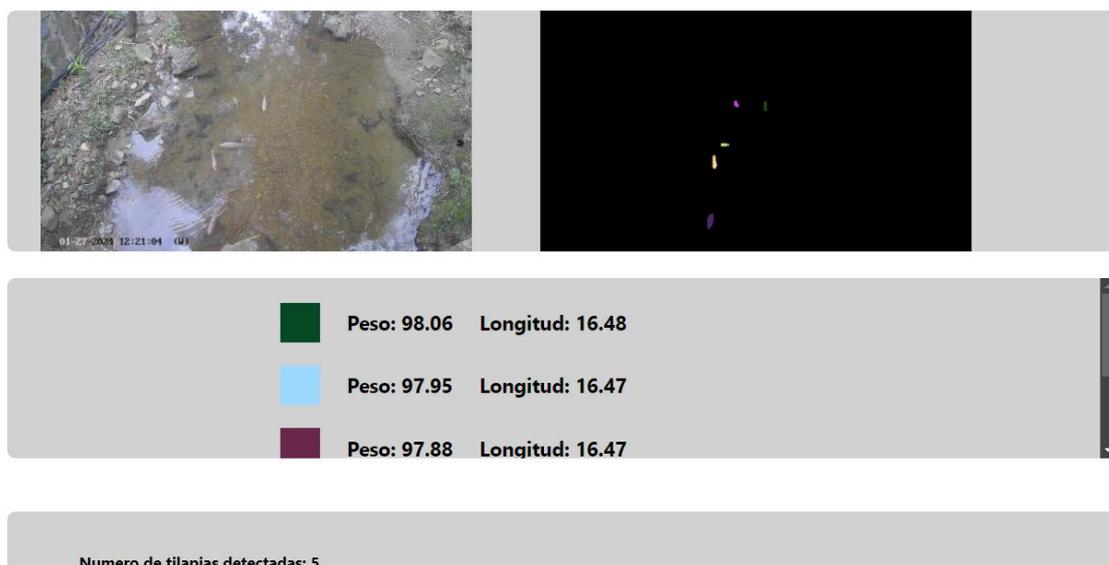


Figura 36: Resultados de la predicción de la aplicación web

Como se puede observar en la figura 36 la aplicación web puede detectar y segmentar las tilapias, de acuerdo a lo evaluado anteriormente con un porcentaje del 88% esto puede variar también debido a la calidad de la imagen e iluminación al momento de predecir y segmentar la imagen.

USABILIDAD

Eficiencia

Para evaluar la eficiencia en términos de tiempo de detectar y predecir el peso y longitud de las tilapias se tomo el tiempo de las predicciones realizadas por la aplicación web y mediciones tradicionales, esto se detalla en la tabla 22.

Tabla 22: Tiempos de Mediciones de tilapias

Acción	Tiempo usado con la aplicación web				Tiempo promedio
Mediciones manuales	2 m	3 m	2 m	4 m	8 m
Predicciones de la aplicación web	1.20 s	1.34 s	1.20 s	1.40 s	1.28 s

Con un promedio de 8 minutos en las mediciones manuales midiendo 4 tilapias y 1.28 segundos prediciendo todas las tilapias en la imagen, lo cual concluye que existe una mejora en lo que respecta a las agilizaciones de mediciones debido a que la diferencia en tiempo entre las mediciones manuales y la predicción del modelo es significativamente grande. Este contraste resalta la eficiencia temporal del modelo, ya que reduce drásticamente el tiempo requerido para realizar mediciones.

CONCLUSIONES

Como resultado del desarrollo de la aplicación de detección y predicción del peso y longitud de las tilapias, aplicando redes neuronales convolucionales se concluye lo siguiente:

- Se logró desarrollar una aplicación web la cual permite detectar, segmentar y predecir las estimaciones de las tilapias como su peso y longitud mediante la implementación de redes neuronales y técnicas estadísticas como la regresión lineal para la estimación de peso y longitud, no obstante, solo converge para ciertos datos de peso y longitud usados en el entrenamiento, ya que se necesita más diversidad en los datos para mejorar su rendimiento.
- Se logró una construcción del marco teórico con importantes conceptos como inteligencia artificial, redes neuronales y visión artificial por medio de una búsqueda bibliográfica en revistas científicas utilizando la metodología SLA.
- Se construyó un dataset de imágenes de tilapias con un total de 1000 tilapias y 60 mediciones tanto de su longitud y peso posteriormente utilizadas en el entrenamiento
- Se logró implementar el modelo pre-entrenado de yolov8 para la detección y segmentación de tilapias permitiendo realizar el entrenamiento y obteniendo métricas importantes además de una matriz de confusión para la posterior evaluación del modelo.
- Se logró implementar un modelo estadístico como lo es una regresión lineal la cual fue entrenada por los datos de mediciones manuales realizadas a 60 tilapias además de utilizar características importantes como lo es el área, relación de aspecto y perímetro del área segmentada de la tilapia para predecir las estimaciones de peso y longitud.
- A partir de la evaluación, se han obtenido los siguientes resultados, el modelo de detección y segmentación tiene una precisión del 88% de los casos y el 12% de los casos predijo incorrectamente que se trataba de una tilapia, mientras que el modelo de estimación de peso y longitud al comparar con las mediciones reales estas fueron bastante alejadas obteniendo resultados no satisfactorios en las predicciones de estimación.
- Se logró realizar una aplicación web con las siguientes tecnologías como parte de la interfaz se usó react como frontend y en el backend se usó flask, además de usar una base de datos de firebase para el guardado de las predicciones realizadas, además la comunicación con los dos modelos se los realizó mediante un api/rest

RECOMENDACIONES

En base a los resultados obtenidos y a la experiencia que se obtuvo en el desarrollo de la aplicación y en el entrenamiento de redes neuronales convolucionales y en la recolección de datos, es necesario recomendar lo siguiente:

- Utilizar una metodología que facilite la identificación de palabras clave y la formulación de cadenas de búsqueda ha posibilitado la selección de artículos científicos pertinentes para respaldar la investigación. Este enfoque asegura la presencia de evidencia sólida proveniente de diversas bases de datos científicas, contribuyendo así a fortalecer la credibilidad y autenticidad del contenido investigativo.
- Aumentar el tamaño del dataset con imágenes que contengan una calidad aceptable, tomando en cuenta la iluminación además de etiquetar debidamente bien los datos para el entrenamiento.
- Mejorar el modelo de detección y segmentación entrenando con más datos y ajustando parámetros importantes como la época y optimizador etc, esto con el propósito de que el modelo aprende a segmentar de mejor manera.
- Mejorar el modelo de regresión lineal para la estimación del peso y longitud mediante el aumento de datos de mediciones manuales, esto con el propósito de tener datos variados, esto se lo podría realizar en intervalos de 3 meses la recolección de datos de mediciones manuales para mejorar la predicción.
- Se recomienda que la evaluación del modelo de segmentación tenga un 95% o más de predicción para segmentar, además también se recomienda que las predicciones de peso y longitud sean aproximadas a las mediciones reales.
- Se recomienda utilizar tecnologías que permitan el fácil desarrollo de la aplicación y una metodología fácil de seguir esto con el propósito de estructurar la aplicación

REFERENCIAS

- [1] A. Naiberg y J. Little, «A unified recognition and stereo vision system for size assessment of fish,» *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, pp. 2-9, 1994.
- [2] N. Tengtrairat , W. Woo, P. Parathai , D. Rinchumphu y C. Chaichana , «Non-Intrusive Fish Weight Estimation in Turbid Water Using Deep Learning and Regression Models,» *Sensors (Basel)*, vol. 22, nº 14, p. 10, 2022.
- [3] S. Chen, W. Qingbin , H. Xinlei , Z. Xiaoshuan y L. Daoliang , «An automatic method of fish length estimation using underwater stereo system based on LabVIEW,» *Computers and Electronics in Agriculture*, vol. 173, 2020.
- [4] P. Muñoz, J. Martínez, G. García, V. Puig, V. Espinosa, I. Pérez, F. De la Gándara y A. Ortega, «Impact evaluation of deep learning on image segmentation for automatic bluefin tuna sizing,» *Aquacultural Engineering*, vol. 99, 2022.
- [5] L. K. Tolentino, C. P. De Pedro, J. D. Icamina, J. B. Navarro, L. J. Salvacion, G. C. Sobrevilla, . A. A. Villanueva, T. M. Amado, M. V. Padilla y G. A. Madrigal, «Weight Prediction System for Nile Tilapia using Image Processing and Predictive Analysis,» *International Journal of Advanced Computer Science and Applications*, vol. 11, nº 8, 2020.
- [6] M. M. Fouad, H. M. Zawbaa, N. El-Bendary y A. E. Hassanien, «Automatic nile tilapia fish classification approach using machine learning techniques,» *in Proc. 13th Int. Conf. Hybrid Intell. Syst. (HIS)*, pp. 173-178, 2013.
- [7] N. V. Barulin, «Using machine learning algorithms to analyse the scute structure and sex identification of sterlet *Acipenser ruthenus* (Acipenseridae).,» *Aquaculture Research*, vol. 50, p. 2810–2825, 2019.
- [8] K. Rungruangsak-Torrissen y P. Manoonpong , «Neural computational model GrowthEstimate: A model for studying living resources through digestive efficiency.,» *PLoS One*, vol. 14, nº 8, p. 28, 2019.
- [9] N. Bravata, D. Kelly, J. Eickholt, J. Bryan, S. Miehls y D. Zielinski, «Applications of deep convolutional neural networks to predict length, circumference, and weight from mostly dewatered images of fish,» *Ecol Evol*, vol. 10, p. 9313– 9325, 2020.
- [10] Y. Yang, B. Xue, J. L. M. Wylie, M. Zhang y M. Wellenreuther, «Deep Convolutional Neural Networks for Fish Weight Prediction from Images,» *2021 36th International Conference on Image and Vision Computing New Zealand (IVCNZ), Tauranga, New Zealand*, pp. 1-6, 2021.
- [11] H. Liu, X. Ma, Y. Yu, L. Wang y L. Hao , «Application of Deep Learning-Based Object Detection Techniques in Fish Aquaculture: A Review,» *Journal of Marine Science and Engineering*, vol. 11, nº 4, p. 867, 2023.

- [12] B. Suarez Puerto, M. Araneda y M. Gullian Klanian, «Bioeconomic analysis of the commercial production of Nile tilapia with biofloc and green water technologies,» *Aquac. Fish*, pp. 1-14, 2023.
- [13] S. Murphy, H. Charo Karisa, S. Rajaratnam, S. Cole, C. McDougall y A. M. Nasr Allah, «Selective breeding trait preferences for farmed tilapia among low-income women and men consumers in Egypt: Implications for pro-poor and gender-responsive fish breeding programmes,» *Aquaculture*, vol. 525, pp. 1-11, 2020.
- [14] P. Silveira Moraes, . J. Ines Engelmann y A. Vallerao Igansi, «Nile tilapia industrialization waste: Evaluation of the yield, quality and cost of the biodiesel production process,» *Journal of cleaner Production*, vol. 287, pp. 1-9, 2021.
- [15] S. C. Mana y T. Sasipraba, «An Intelligent Deep Learning Enabled Marine Fish Species Detection and Classification Model,» *International Journal of Artificial Intelligence Tools: Architectures, Languages, Algorithms*, vol. 31, nº 1, 2022.
- [16] M. Paraschiv, R. Padrino, P. Casari, E. Bigal, A. Scheinin, D. Tchernov y A. Fernández Anta, «Classification of Underwater Fish Images and Videos via Very Small Convolutional Neural Networks,» *Journal of marine Science and Engineering*, vol. 10, nº 6, pp. 2-21, 2022.
- [17] T. Hong Khai, S. N. H. S. Abdullah, M. K. Hasan y A. Tarmizi, «Underwater Fish Detection and Counting Using Mask Regional Convolutional Neural Network,» *WATER*, vol. 14, nº 2, pp. 1-23, 2022.
- [18] J. A. Rafsun , H. Monowar y H. Sabir , «Fish Classification using Saliency Detection Depending on Shape and Texture,» *Computación y sistemas*, vol. 26, nº 1, pp. 303-310, 2022.
- [19] . C.-H. Tseng, C.-L. Hsieh y Y.-F. Kuo, «Automatic measurement of the body length of harvested fish using convolutional neural networks,» *Biosystems Engineering*, vol. 189, pp. 36-47, 2020.
- [20] G. G. Monkman, K. Hyder, M. J. Kaiser y F. P. Vidal, «Using Machine Vision to Estimate Fish Length from Images using Regional Convolutional Neural Networks,» *Methods in ecology and evolution*, vol. 10, 2019.
- [21] . A. Crescitelli, L. Gansel y Z. H, «NorFisk: fish image dataset from Norwegian fish farms for species recognition using deep neural networks,» *Modeling Identification and Control* , vol. 42, nº 1, pp. 1-16, 2021.
- [22] C.-H. Tseng y Y.-F. Kuo, «Detecting and counting harvested fish and identifying fish types in electronic monitoring system videos using deep convolutional neural networks,» *Ices Journal of Marine Science*, vol. 77, nº 4, pp. 1367-1378, 2020.
- [23] S. Lopez Tejeida, G. M. Soto Zarazua, M. T. Ayala, L. M. Contreras Medina, E. A. Rivas Araiza y P. S. Flores Aguilar, «An Improved Method to Obtain Fish Weight Using Machine Learning and NIR Camera with Haar Cascade Classifier,» *Applied Sciences-Basel*, vol. 13, nº 69, pp. 1-14, 2023.
- [24] V. Kandimalla, R. M. F. Smith, J. Quirion, L. Torgo y C. Whidden, «Automated Detection, Classification and Counting of Fish in Fish Passages With Deep Learning,» *Frontiers in Marine Science*, vol. 8, 2022.

- [25] A. J. Rico Diaz, J. R. Rabunal, M. Gestal, O. A. Mures y J. Puertas, «An Application of Fish Detection Based on Eye Search with Artificial Vision and Artificial Neural Networks,» *Water*, vol. 12, nº 11, 2020.
- [26] E. Amaya Alvarez, . P. Miquel , J. L. Lisani y I. Catalan, «Image-based, unsupervised estimation of fish size from commercial landings using deep learning,» *Ices Journal of Marine Science*, vol. 77, nº 4, pp. 1330-1339, 2020.
- [27] S. Cui, Y. Zhou, Y. Wang y L. Zhai, «Fish Detection Using Deep Learning,» *Applied Computational Intelligence and Soft Computing*, vol. 2020, pp. 1-13, 2020.
- [28] Q. Liu, X. Gong, J. Li, R. Liu, D. Liu, R. Zhou, T. Xie, R. Fu y X. Duan, «A multitask model for realtime fish detection and segmentation based on YOLOv5,» *PeerJ Computer Science*, vol. 9, p. 1–31, 2023.
- [29] P. Software Foundation, 2019. [En línea]. Available: <https://docs.python.org/3/tutorial/index.html>. [Último acceso: 19 07 2020].
- [30] Colaboratory, 2014. [En línea]. Available: <https://colab.research.google.com/notebooks/intro.ipynb>. [Último acceso: 19 07 2023].
- [31] Figma, 2016. [En línea]. Available: <https://www.figma.com/>. [Último acceso: 19 07 2023].
- [32] SurveyMonkey, «SurveyMonkey,» [En línea]. Available: <https://es.surveymonkey.com/mp/sample-size-calculator/>.

ANEXOS

Anexo 1: Matriz de consistencia

Problema, objeto y campo	Objetivo	Marco Teórico	Hipótesis	Variables	Metodología
<p>Problema:</p> <ul style="list-style-type: none"> Necesidad de desarrollar una aplicación para la detección de tilapias que nos permita obtener las estimaciones del crecimiento de una manera ágil debido al alto consumo de tiempo mediante la manera tradicional. <p>Objeto de estudio:</p> <ul style="list-style-type: none"> Aplicación basada en machine learning para el reconocimiento y estimaciones del crecimiento de tilapias. <p>Campo de Acción:</p> <ul style="list-style-type: none"> Desarrollo e implementación de una aplicación basada en machine learning que utilice redes neuronales convolucionales y modelos estadísticos para la detección y estimación del crecimiento de tilapias. 	<p>Objetivo General:</p> <ul style="list-style-type: none"> Desarrollar una aplicación web utilizando redes neuronales convolucionales, para la estimación del crecimiento de tilapias, mediante el uso de técnicas de modelado estadístico. <p>Objetivos Específicos:</p> <ul style="list-style-type: none"> Recopilar información de artículos científicos de fuentes confiables para construir el estado del arte y marco teórico. Recopilar datos de tilapias para la creación de un dataset representativo de imágenes de tilapias. Implementar la red neuronal convolucional YOLOv8 para la detección y segmentación de objetos en imágenes. Implementar un modelo estadístico, como regresión, para la estimación del tamaño y peso de las tilapias. Realizar pruebas de estimación de peso y tamaño de las tilapias, además de validar ambos 	<p>Antecedentes históricos a nivel internacional y nacional del objeto, campo:</p> <ul style="list-style-type: none"> Historia del monitoreo de tilapias utilizando machine learning <p>Fundamentos Teóricos de objeto, campo y variables:</p> <p>1.3.1. Tilapias</p> <ul style="list-style-type: none"> 1.3.1.1. Crianza de Tilapias 1.3.1.2. Preferencias y Nutrición en Consumidores 1.3.1.3. Desafíos y Oportunidades hacia una Producción Sostenible <p>1.3.2. Modelos de Machine Learning</p> <ul style="list-style-type: none"> 1.3.2.1. Introducción a los modelos de machine learning 1.3.2.2. Regresión Lineal 1.3.2.3. Regresión Logística 1.3.2.4. Redes Neuronales <p>1.3.3. Técnicas de Procesamiento</p> <ul style="list-style-type: none"> 1.3.3.1. Introducción al procesamiento imágenes 1.3.3.2. Procesamiento de Imágenes 1.3.3.3. Detección de bordes 	<p>Hipótesis General:</p> <ul style="list-style-type: none"> Si se desarrolla una aplicación web utilizando machine learning, para estimar la longitud y peso de las tilapias mediante algoritmos de aprendizaje supervisado, agilizando el proceso de medición directa, se evaluará la precisión comparando las estimaciones con las mediciones reales. 	<p>Variable 1/ Independiente:</p> <ul style="list-style-type: none"> Aplicación web utilizando machine learning <p>Dimensiones o categorías:</p> <ul style="list-style-type: none"> Procesamiento De Imágenes Desarrollo web <p>Variable 2/ Dependiente:</p> <ul style="list-style-type: none"> Estimaciones del peso y longitud de las tilapias <p>Dimensiones o categorías:</p> <ul style="list-style-type: none"> Precisión de las estimaciones 	<p>Enfoque:</p> <ul style="list-style-type: none"> El enfoque será cuantitativo <p>Alcance:</p> <ul style="list-style-type: none"> En cuanto el alcance del proyecto tendrá elementos exploratorios y descriptivos <p>Diseño:</p> <ul style="list-style-type: none"> Finalmente, el diseño de la investigación será cuasiexperimental <p>Unidades de análisis:</p> <p>Población:</p> <ul style="list-style-type: none"> La población escogida estará compuesta por todas las tilapias dentro del área. <p>Técnicas e instrumentos de recolección de datos:</p> <p>Observación;</p> <ul style="list-style-type: none"> Cámara o dispositivo móvil para la toma de fotos de las tilapias Cuaderno de campo o hojas de registro <p>Análisis de documento o datos o procesos</p>

	<p>modelos con sus respectivas métricas.</p> <ul style="list-style-type: none"> • Desarrollar una interfaz de usuario intuitiva y amigable que permita a los usuarios subir una imagen y recibir información sobre los parámetros de longitud y peso de las tilapias. 	<ul style="list-style-type: none"> • 1.3.3.4. Segmentación de imágenes. <p>1.3.4. Herramientas de Desarrollo</p> <ul style="list-style-type: none"> • 1.3.4.1. Python • 1.3.4.2. Google Colab • 1.3.4.3. Figma 			<ul style="list-style-type: none"> • Acceso a documentos científicos y literatura relevante <p>Técnicas de procesamiento de datos:</p> <ul style="list-style-type: none"> • Las técnicas de procesamiento de datos utilizados para la obtención de resultados en nuestro proyecto fue el uso del Deep learning .
--	--	---	--	--	---

Anexo 2: Datos de medición de las tilapias

El Anexo 2 presenta los datos de cada tilapia individual, junto con sus mediciones correspondientes de diversos aspectos físicos.

Datos de Tilapia				
Número De Tilapia	Ancho (cm)	Altura (cm)	Longitud (cm)	Peso (KG)
1	2,1	5,5	15,5	0,09
2	2,5	6,5	17,50	0,115
3	2,1	5,5	15,5	0,09
4	2,3	6	16,5	0,105
5	2,7	6,6	17,3	0,115
6	2,4	6,3	18,5	0,12
7	2,1	5	15	0,075
8	2,5	5,3	17	0,08
9	1,7	4	12,5	0,04
10	2,3	6,5	17,5	0,12
11	2,3	5	15	0,075
12	1,3	3	9,7	0,025
13	2,4	6	16,2	0,1
14	2,5	6	17	0,105
15	2	4,5	13,4	0,045
16	2,1	4,8	14	0,055
17	1,7	4	12,5	0,045
18	2,5	6,2	18	0,125
19	2,1	4,8	13	0,065
20	2,2	5,5	15,5	0,08
21	2,8	7	19,3	0,14
22	2,5	6	17,5	0,115
23	2,3	6	17	0,1
24	2	4,7	15	0,065
25	2,4	6,2	19	0,135
26	2,2	4,7	14,5	0,065
27	2,6	5,7	17,3	0,115
28	2,2	4,9	16	0,08
29	2,5	5,4	16	0,09
30	2,2	5	16	0,075
31	2,2	4,5	14	0,065
32	2,8	6,4	18	0,14
33	2,7	6	16,8	0,1
34	2,7	6	18,2	0,12
35	0,8	5,5	16,5	0,095
36	2,6	6	16,4	0,09
37	2,6	6,2	17,6	0,125
38	2,2	5	14,8	0,065

39	1,9	4,5	13,3	0,05
40	2,2	4,7	15,5	0,075
41	2,3	5	16	0,09
42	2	4,5	14	0,055
43	1,8	4	13	0,05
44	2,6	6	17	0,11
45	2,6	6	17	0,105
46	2	4,5	14,5	0,065
47	2,2	6	18,4	0,11
48	2,3	5,5	16,5	0,1
49	2,7	5,7	17,5	0,11
50	2,7	6,4	18,3	0,125
51	2,9	6,5	17,5	0,135
52	2,4	5,8	17,5	0,105
53	2,1	5	15,8	0,08
54	2,4	6	18	0,105
55	2	4,4	12,7	0,045
56	2,2	5,5	16,5	0,095
57	2,1	5,5	16,5	0,095
58	2,1	4,5	14	0,04
59	1,5	3,6	11,8	0,035
60	2,2	5	16	0,095
61	2,1	4,6	14,5	0,065
62	2,2	4,7	15	0,065
63	2	6	17	0,09
64	1,7	4,3	13	0,045
65	1,9	4	13	0,045
66	2,2	4,5	15,3	0,085
67	2,4	5,5	17,5	0,075
68	2,2	4,7	14,8	0,065

Anexo 3: Historias de Usuario

Tabla 23: Historia de usuario Nro. 1

Historia de usuario	
Número: 1	Usuario: Cliente/Administrador
Nombre de la historia: Mostrar información	
Prioridad en negocio: Baja	
Iteración asignada: 1	
Programador responsable: Armijos Galo, Alejandro Vásquez	
Descripción: Los usuarios podrán ver la información de cómo se maneja la aplicación para que no ocurran problemas a la hora de predecir.	
Observaciones: Ninguna	

Tabla 24: Historia de usuario Nro.2

Historia de usuario	
Número: 1	Usuario: Cliente/Administrador
Nombre de la historia: Predicción del peso y segmentación de la imagen	
Prioridad en negocio: Alta	
Iteración asignada: 2	
Programador responsable: Armijos Galo, Alejandro Vásquez	
Descripción: Los usuarios podrán tomar una toma foto y poder predecir tanto su peso y la tilapia	
Observaciones: Ninguna	

Tabla 25: Historia de usuario Nro.3

Historia de usuario	
Número: 1	Usuario: Cliente/Administrador
Nombre de la historia: Visualizar Información	
Prioridad en negocio: Alta	
Iteración asignada: 3	
Programador responsable: Armijos Galo, Alejandro Vásquez	
Descripción: Los usuarios podrán observar los resultados de las predicciones realizadas.	
Observaciones: Ninguna	