



**UNIVERSIDAD NACIONAL DEL ALTIPLANO**  
**FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA,**  
**ELECTRÓNICA Y SISTEMAS**  
**ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA**



**DISEÑO DE UN ALGORITMO PARA EL CÁLCULO DEL VALOR**  
**ENERGÉTICO EN PERSONAS DE 45 A 75 AÑOS EN EL**  
**DISTRITO DE SAN BORJA – LIMA MEDIANTE INTELIGENCIA**  
**ARTIFICIAL**

**TESIS**

**PRESENTADA POR:**

**LUIS ANGEL CURO ROQUE**

**PARA OPTAR EL TÍTULO PROFESIONAL DE:**

**INGENIERO ELECTRÓNICO**

**PUNO – PERÚ**

**2024**



NOMBRE DEL TRABAJO

DISEÑO DE UN ALGORITMO PARA EL CÁLCULO DEL VALOR ENERGÉTICO EN PERSONAS DE 45 A 75 AÑOS EN EL DISTRITO

AUTOR

LUIS ANGEL CURO ROQUE

RECuento DE PALABRAS

16273 Words

RECuento DE CARACTERES

95720 Characters

RECuento DE PÁGINAS

103 Pages

TAMAÑO DEL ARCHIVO

2.7MB

FECHA DE ENTREGA

Aug 1, 2024 12:10 PM GMT-5

FECHA DEL INFORME

Aug 1, 2024 12:11 PM GMT-5

● **13% de similitud general**

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 10% Base de datos de Internet
- Base de datos de Crossref
- 9% Base de datos de trabajos entregados
- 3% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● **Excluir del Reporte de Similitud**

- Material bibliográfico
- Coincidencia baja (menos de 10 palabras)

  
Mg. Ferdinand F. Pineda Ancco  
INGENIERO ELECTRÓNICO  
CIP. 80687

  
Karlo A. C. Chirpa  
Subdirector de Investigación EPIE  
NO BO el Reporte Pluralis, mas  
no el Contenido de la Tesis



## DEDICATORIA

*A mis queridos padres y hermanos por el apoyo incondicional y constante los cuales fueron pilares importantes en mi formación académica. Ustedes me inculcaron el significado del trabajo duro, la perseverancia y confianza en uno mismo. Cada palabra de aliento me dio las fuerzas para superar cualquier obstáculo y poder lograr mis sueños. No existe palabras suficientes para expresar mi gratitud. Por eso este logro se lo dedico a ustedes.*

*A mis amigos, por su amistad sincera y lealtad inquebrantable. Ustedes han sido mi compañía en momentos de incertidumbre y felicidad.*

**Luis Angel Curo Roque**



## AGRADECIMIENTOS

*La culminación de esta etapa, es necesario mirar hacia atrás para observar el sendero por el cual pasaste y agradecer a las personas que te han apoyado durante este viaje. Como también a aquellos sucesos que para bien o mal te enseñan como enfrentar los nuevos retos que están por venir.*

*Agradecer a mis padres por el apoyo incondicional que han sido mi mayor motivación. A mi asesor Mg. Ferdinand Edgardo Pineda Ancco, el cual me proporciono los conceptos y herramientas indispensables para la ejecución del proyecto.*

*Muchas veces el temor al fracaso puede dañar la manera de pensar de las personas, personalmente eh logrado comprender que la vulnerabilidad es una parte esencial para seguir adelante. En esos momentos de incertidumbre, entendí que fallar no es el fin, mas bien una oportunidad para aprender de tus errores y poder mejorar. La frase “vulnerables no quisimos ser, por miedo a que no salga bien” describe muy bien este concepto. Sin mas agradezco por esos momentos intangibles, pero inolvidables por el paso del tiempo. Esos instantes etéreos, llenos de significados y emociones, han sido las bases de mi alma, y su esencia vive conmigo en cada paso de mi vida.*

**Luis Angel Curo Roque**



# ÍNDICE GENERAL

	<b>Pág.</b>
<b>DEDICATORIA</b>	
<b>AGRADECIMIENTOS</b>	
<b>ÍNDICE GENERAL</b>	
<b>ÍNDICE DE TABLAS</b>	
<b>ÍNDICE DE FIGURAS</b>	
<b>INDICE DE ANEXOS</b>	
<b>ACRÓNIMOS</b>	
<b>RESUMEN .....</b>	<b>17</b>
<b>ABSTRACT.....</b>	<b>18</b>
<b>CAPÍTULO I</b>	
<b>INTRODUCCIÓN</b>	
<b>1.1. PLANTEAMIENTO DEL PROBLEMA.....</b>	<b>20</b>
<b>1.2. HIPÓTESIS .....</b>	<b>20</b>
<b>1.3. OBJETIVO GENERAL .....</b>	<b>21</b>
<b>1.4. OBJETIVOS ESPECIFICOS .....</b>	<b>21</b>
<b>CAPÍTULO II</b>	
<b>REVISIÓN DE LITERATURA</b>	
<b>2.1. ANTECEDENTES DE LA INVESTIGACIÓN.....</b>	<b>22</b>
<b>2.2. OBESIDAD.....</b>	<b>24</b>
<b>2.3. MALNUTRICIÓN .....</b>	<b>25</b>
<b>2.4. REQUERIMIENTO ENERGÉTICO .....</b>	<b>26</b>
<b>2.4.1. Índice de Masa Corporal (IMC).....</b>	<b>26</b>



2.4.2. Índice de Broca .....	26
<b>2.5. PATRON DE CONSUMO ALIMENTARIO .....</b>	<b>27</b>
2.5.1. Transición Nutricional .....	27
2.5.2. Nivel de actividad física .....	28
2.5.3. Problemática Nutricional en Fumadores.....	28
<b>2.6. REDES NEURONALES ARTIFICIALES.....</b>	<b>28</b>
2.6.1 Redes Neuronales no Recurrentes .....	28
2.6.2. Modelo de Regresión .....	29
2.6.3. Funciones de Activación.....	30
2.6.4. Entrenamiento de la red neuronal .....	31
<b>2.7. PYTHON.....</b>	<b>32</b>
2.7.1. Flask.....	33
2.7.2. Jinja .....	34
<b>2.8. LIBRERÍAS.....</b>	<b>34</b>
2.8.1. Tensorflow .....	34
2.8.2. Keras .....	35
2.8.3. Numpy .....	36
2.8.4. Matplotlib.....	36
2.8.5. Train_test_split .....	36
2.8.6. Sequential.....	37
2.8.7. Sklearn.linear .....	37
2.8.8. Xgboost.....	38
2.8.9. Dense .....	38
2.8.10. Normalización.....	38



<b>2.9. DATASET</b> .....	38
<b>2.10. ANACONDA NAVIGATOR</b> .....	39
<b>2.11. VISUAL STUDIO CODE</b> .....	39
<b>2.12. LENGUAJE CSS</b> .....	40

### **CAPÍTULO III**

#### **MATERIALES Y MÉTODOS**

<b>3.1. MATERIALES</b> .....	41
3.1.1. Hardware.....	41
3.1.2. Software .....	41
<b>3.2. DISEÑO Y NIVEL DE LA INVESTIGACIÓN</b> .....	41
3.2.1. Diseño de la investigación .....	41
3.2.2. Nivel de la investigación.....	43
<b>3.3. POBLACIÓN Y MUESTRA DE LA INVESTIGACIÓN</b> .....	43
<b>3.4. UBICACIÓN DE LA INVESTIGACIÓN</b> .....	44
<b>3.5. TÉCNICAS E INSTRUMENTOS PARA LA RECOLECCIÓN DE DATOS</b> .....	45
3.5.1. Técnicas .....	45
3.5.2. Instrumentos.....	46

### **CAPÍTULO IV**

#### **RESULTADOS Y DISCUSIÓN**

<b>4.1. ADQUISICION DE DATOS</b> .....	47
<b>4.2. PROCESO DE CÁLCULO DE PESO IDEAL</b> .....	49
4.3.1. Evaluación del Índice de Masa Corporal .....	49
4.3.2. Cálculo de Peso Ideal.....	51



<b>4.3.</b>	<b>PREPROCESAMIENTO DE DATOS.....</b>	<b>53</b>
<b>4.4.</b>	<b>DATASET.....</b>	<b>58</b>
<b>4.5.</b>	<b>IMPORTACION DE LIBRERIAS .....</b>	<b>59</b>
	4.3.1. Importación de Librerías Básicas .....	60
	4.3.2. Importación de Librerías de Modelos de Inteligencia Artificial.....	61
<b>4.6.</b>	<b>EXPLORACION DE DATOS .....</b>	<b>62</b>
	4.4.1. Visualización de Datos .....	63
<b>4.7.</b>	<b>SEPARACIÓN DE DATOS.....</b>	<b>64</b>
<b>4.8.</b>	<b>DISEÑO DE ALGORITMOS CON MACHINE LEARNING .....</b>	<b>65</b>
	4.8.1. Experimento 1:.....	65
<b>4.9.</b>	<b>DISEÑO DE ALGORITMOS CON REDES NEURONALES.....</b>	<b>67</b>
	4.9.1. Normalización de Datos.....	67
	4.8.2. Experimento 2: Algoritmos mediante Redes Neuronales.....	68
	4.8.3. Épocas Usadas .....	70
<b>4.10.</b>	<b>ASEGURAR LOS MODELOS ENTRENADOS.....</b>	<b>70</b>
<b>4.11.</b>	<b>DISEÑO DE LA INTERFAZ.....</b>	<b>71</b>
<b>4.12.</b>	<b>DISEÑO DEL FORMULARIO.....</b>	<b>75</b>
<b>4.13.</b>	<b>DISEÑO DE LA INTERFAZ DE RESPUESTA .....</b>	<b>79</b>
<b>4.14.</b>	<b>PREDICCIÓN DE NUEVOS DATOS.....</b>	<b>81</b>
<b>V.</b>	<b>CONCLUSIONES.....</b>	<b>85</b>
<b>VI.</b>	<b>RECOMENDACIONES.....</b>	<b>87</b>
<b>VII.</b>	<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>88</b>
<b>ANEXOS</b> .....		<b>96</b>



**ÁREA:** Telecomunicaciones y Redes de Datos

**TEMA:** Inteligencia Artificial

**FECHA DE SUSTENTACIÓN:** 06 de setiembre de 2024



## ÍNDICE DE TABLAS

	<b>Pág.</b>
<b>Tabla 1</b> Índice de Masa Corporal en Adultos .....	50
<b>Tabla 2</b> Muestra de los resultados del cálculo de Peso Ideal .....	52
<b>Tabla 3</b> Método general de cálculo de requerimientos nutricionales .....	53
<b>Tabla 4</b> Muestra de los resultados obtenidos del requerimiento nutricional .....	53
<b>Tabla 5</b> Resultados de las métricas $R^2$ y MAE con machine learning .....	66
<b>Tabla 6</b> Resultados de las métricas $R^2$ y MAE con redes neuronales (parte 1) .....	69
<b>Tabla 7</b> Resultados de las métricas $R^2$ y MAE con redes neuronales (parte 2) .....	69
<b>Tabla 8</b> Épocas y validación split usadas con redes neuronales .....	70



## ÍNDICE DE FIGURAS

	<b>Pág.</b>
<b>Figura 1</b> Esquema de Red Neuronal .....	29
<b>Figura 2</b> Esquema representativo de X y Y .....	30
<b>Figura 3</b> Función de activación ReLu .....	31
<b>Figura 4</b> Gráficos de underfitting y overfitting.....	32
<b>Figura 5</b> Método para trabajar con la variable "for" .....	34
<b>Figura 6</b> Código de train_test_split.....	37
<b>Figura 7</b> Ubicación por mapa satelital del Distrito de San Borja del Departamento de Lima.....	45
<b>Figura 8</b> Muestra de los datos obtenidos de la base de datos en formato delimitado por comas .....	48
<b>Figura 9</b> Muestra de los datos en formato estándar .....	49
<b>Figura 10</b> Formula de cálculo del índice de masa corporal .....	49
<b>Figura 11</b> Resultados del IMC a la base de datos .....	51
<b>Figura 12</b> Imagen gráfica de los resultados del IMC a la base de datos.....	51
<b>Figura 13</b> Fórmula de Broca .....	52
<b>Figura 14</b> Base de datos resultante según las limitantes del proyecto .....	54
<b>Figura 15</b> Codificación de los datos dentro de la variable "Genero" .....	54
<b>Figura 16</b> Codificación de los datos dentro de la variable "Estado_Civil".....	54
<b>Figura 17</b> Codificación de los datos dentro de la variable "Grado_ Estudios" .....	55
<b>Figura 18</b> Codificación de los datos dentro de la variable "Exp_tabaco".....	55
<b>Figura 19</b> Codificación de los datos dentro de la variable "Enfermedad" .....	55
<b>Figura 20</b> Codificación de los datos dentro de la variable "Act_fis" .....	56
<b>Figura 21</b> Codificación de los datos dentro de la variable "Consm_fruta".....	56



<b>Figura 22</b>	Codificación de los datos dentro de la variable "Consm_verdura" .....	57
<b>Figura 23</b>	Muestra de la base de datos después del proceso de codificación .....	58
<b>Figura 24</b>	Muestra de la base de datos .....	58
<b>Figura 25</b>	Muestra de la base de datos de requerimiento energético obtenida .....	58
<b>Figura 26</b>	Entorno de Jupyter Notebook .....	60
<b>Figura 27</b>	Versión de tensorflow .....	60
<b>Figura 28</b>	Librerías básicas importadas .....	61
<b>Figura 29</b>	Librerías importadas de modelos de inteligencia artificial .....	62
<b>Figura 30</b>	Código usado para la visualización de los datos .....	63
<b>Figura 31</b>	Código usado para la selección de los datos de entrada .....	63
<b>Figura 32</b>	Código usado para la selección de los datos de salida .....	64
<b>Figura 33</b>	Muestra de los Datos de entrada .....	64
<b>Figura 34</b>	Muestra de los Datos de salida .....	64
<b>Figura 35</b>	Código usado para la separación de train y test .....	65
<b>Figura 36</b>	Gráfica de los resultados de las métricas $R^2$ y MAE con machine learning .....	66
<b>Figura 37</b>	Código usado para normalizar los datos de entrada .....	68
<b>Figura 38</b>	Código para aplicar el normalizador a los datos de entrenamiento .....	68
<b>Figura 39</b>	Gráfica de los resultados de las métricas $R^2$ y MAE con redes neuronales .....	69
<b>Figura 40</b>	Código usado para guardar el modelo de red neuronal .....	71
<b>Figura 41</b>	Código usado para importar librerías .....	71
<b>Figura 42</b>	Código usado para cargar el modelo de red neuronal .....	72
<b>Figura 43</b>	Código usado para introducir los datos de entrada a la red neuronal .....	72
<b>Figura 44</b>	Código usado para diseñar la estructura de la aplicación web .....	73



<b>Figura 45</b>	Código de decorador para acceder a la aplicación web.....	73
<b>Figura 46</b>	Código usado para hacer la solicitud.....	74
<b>Figura 47</b>	Código usado para mostrar la predicción .....	75
<b>Figura 48</b>	Código para iniciar el servidor Flask.....	75
<b>Figura 49</b>	Código de uso para declarar variables, metadatos y enlaces de recursos externos.....	76
<b>Figura 50</b>	Código para personalizar los elementos del formulario .....	77
<b>Figura 51</b>	Código usado para introducir los campos de entrada de datos al formulario .....	78
<b>Figura 52</b>	Código usado para crear botón para el envío del formulario .....	78
<b>Figura 53</b>	Código para importar "materialize Java Script" .....	79
<b>Figura 54</b>	Código de validación de datos.....	79
<b>Figura 55</b>	Código usado para el diseño de la interfaz para mostrar la predicción .....	80
<b>Figura 56</b>	Código de generación URL de "Materialize JavaScript" .....	80
<b>Figura 57</b>	Código usado para la visualización de la predicción.....	81
<b>Figura 58</b>	Ingreso de datos de paciente con sobrepeso .....	82
<b>Figura 59</b>	Predicción Kcal requeridas para paciente con sobrepeso.....	82
<b>Figura 60</b>	Ingreso de datos de paciente con peso ideal.....	82
<b>Figura 61</b>	Predicción Kcal requeridas para paciente con peso ideal.....	83
<b>Figura 62</b>	Ingreso de datos de paciente con malnutrición.....	83
<b>Figura 63</b>	Predicción Kcal requeridas para paciente con malnutrición .....	84
<b>Figura 64</b>	Resultados de $R^2$ y MAE de los entrenamientos de la red neuronal (2capas) .....	96
<b>Figura 65</b>	Grafica de los resultados de $R^2$ y MAE de los entrenamientos de la red neuronal (2capas).....	97



<b>Figura 66</b>	Resultados de $R^2$ y MAE de los entrenamientos de la red neuronal (3capas)	97
<b>Figura 67</b>	Grafica de los resultados de $R^2$ y MAE de los entrenamientos de la red neuronal (3capas)	97
<b>Figura 68</b>	Resultados de $R^2$ y MAE de los entrenamientos de la red neuronal (4capas)	98
<b>Figura 69</b>	Grafica de los resultados de $R^2$ y MAE de los entrenamientos de la red neuronal (4capas)	98
<b>Figura 70</b>	Resultados de $R^2$ y MAE de los entrenamientos de la red neuronal (5capas_1)	99
<b>Figura 71</b>	Grafica de los resultados de $R^2$ y MAE de los entrenamientos de la red neuronal (5capas_1)	99
<b>Figura 72</b>	Resultados de $R^2$ y MAE de los entrenamientos de la red neuronal (5capas_2)	100
<b>Figura 73</b>	Grafica de los resultados de $R^2$ y MAE de los entrenamientos de la red neuronal (5capas_2)	100



## ÍNDICE DE ANEXOS

	<b>Pág.</b>
<b>ANEXO 1.</b> Resultados de las métricas $R^2$ y MAE de los entrenamientos por cada de las redes neuronales creadas .....	96
<b>ANEXO 2.</b> Códigos en lenguaje Python de los modelos entrenados, base de datos y aplicativo web.....	101
<b>ANEXO 3.</b> Declaración jurada de autenticidad de tesis.....	102
<b>ANEXO 4.</b> Autorización para el depósito de tesis en el Repositorio Institucional....	103



## ACRÓNIMOS

<b>CSS:</b>	Cascading Style Sheets (Hojas de estilo en cascada)
<b>INEI:</b>	Instituto Nacional de Estadística e Informática
<b>OMS:</b>	Organización Mundial de la Salud
<b>DP:</b>	Deep Learning (Aprendizaje Profundo)
<b>Kcal:</b>	Kilocalorías
<b>IA:</b>	Inteligencia Artificial
<b>Relu:</b>	Rectified Lineal Unit (Unidad Lineal Rectificada)
<b>WSL:</b>	Windows subsystem for Linux
<b>HTML:</b>	Hypertext Markup Language (Lenguaje de Marcado de Hipertexto)
<b>MAE:</b>	Mean Absolute Error (Error Absoluto Medio)



## RESUMEN

En la actualidad, la obesidad es una de las enfermedades más comunes a nivel mundial. Asimismo, en muchos lugares han aceptado esta condición como algo normal. Sin embargo, viendo esta problemática desde otra perspectiva; esta enfermedad aumenta exponencialmente el riesgo de contraer otras enfermedades que ponen en tela de juicio la calidad de vida de las personas. Por ende, es necesario promover una alimentación adecuada. En este contexto, es fundamental la necesidad de una evaluación del paciente para llegar a una condición ideal. En la presente investigación, se realizó el diseño de un algoritmo para el cálculo del valor energético en personas de 45 a 75 años en el distrito de San Borja – Lima mediante inteligencia artificial. El problema planteado fue: ¿En qué medida el diseño de un algoritmo para el cálculo de valor energético (Kcal) en personas de 45 a 75 años en el distrito de San Borja - Lima de puede mejorar su tipo de alimentación para una vida más saludable? El objetivo fue diseñar un algoritmo para calcular el valor energético (Kcal) en personas de 45 a 75 años del distrito de San Borja – Lima mediante inteligencia artificial. La metodología usó el enfoque cuantitativo, el nivel de investigación fue exploratorio con diseño no experimental, los métodos se basaron en el aprendizaje supervisado, las técnicas consideraron el aprendizaje automático (ML) y los instrumentos fueron bibliotecas dedicadas a modelos de IA, plataformas enfocadas en el desarrollo de modelos de IA y el uso de datos públicos.

**Palabras Clave:** Aprendizaje automático, Aprendizaje supervisado, Inteligencia artificial, Requerimiento energético.



## ABSTRACT

Currently, obesity is one of the most common diseases worldwide. Moreover, in many places, this condition has been accepted as normal. However, viewing this issue from another perspective, this disease exponentially increases the risk of contracting other illnesses that call into question the quality of life of individuals. Therefore, it is necessary to promote proper nutrition. In this context, it is essential to evaluate the patient to achieve an ideal condition. In the present research, an algorithm was designed to calculate the energy value in individuals aged 45 to 75 in the district of San Borja – Lima using artificial intelligence. The problem posed was: To what extent can the design of an algorithm for calculating the energy value (Kcal) in individuals aged 45 to 75 in the district of San Borja – Lima improve their dietary habits for a healthier life? The objective was to design an algorithm to calculate the energy value (Kcal) in individuals aged 45 to 75 in the district of San Borja – Lima using artificial intelligence. The methodology used a quantitative approach, the research level was exploratory with a non-experimental design, the methods were based on supervised learning, the techniques considered machine learning (ML), and the instruments were libraries dedicated to AI models, platforms focused on the development of AI models, and the use of public data.

**Keywords:** Artificial intelligence, Energy requirement, Machine learning, Supervised learning.



# CAPÍTULO I

## INTRODUCCIÓN

En la actualidad el uso de inteligencia artificial ya es un hecho. Según CIOSPAIN, la inversión en inteligencia artificial en el año 2023 alcanzó los 154.000 millones de dólares, lo que representa un 26.9% más en comparación al año 2022. De la misma manera, en una encuesta realizada por McKinsey & Company en el año 2023 sobre el uso de inteligencia artificial generativa, más del 60% de empresas líderes en Europa ya usan esta tecnología para la optimización de ciclos de desarrollo, automatización de procesos robóticos, chatbots, aprendizaje automático, desarrollo de productos y servicios, etc.

Por otro lado, según el informe “Perú: Enfermedades No transmisibles y transmisibles” en el año 2020, el 65% de las personas mayores de 15 años presentan un problema de sobrepeso, por lo cual se podría considerar que existe un problema de mal nutrición.

Asimismo, el país se encuentra en un proceso de envejecimiento y el porcentaje de adultos crece con mayor frecuencia a diferencia de la población en general. Este proceso va de la mano con el deterioro de la salud y está asociado a diferentes factores como: herencia genérica, género, estilo de vida, enfermedades preexistentes, estado nutricional, entre otros según (Berniu & Delgado 2021).

Según la Organización Mundial de la Salud (OMS), nos explica que el uso de la inteligencia artificial brinda grandes expectativas para la mejora de prestaciones dirigidas a la atención de salud y la medicina en todo el mundo (OMS, 2021).

La realización del cálculo del valor energético es de vital importancia porque mantiene al organismo en un estado de equilibrio nutricional y salud adecuados. Esto se



debe a que una indicación precisa de nutrimentos puede ayudar al paciente a metabolizar mejor la comida, permitiendo así un catabolismo persistente evitando la insuficiencia y/o exceso de nutrientes para su vida diaria.

Asimismo, la aplicación de nuevas tecnologías emergentes basadas en inteligencia artificial dirigidas al área de la salud, abocándonos específicamente al campo de la nutrición alimentaria son herramientas dirigidas para promover una salud estable en las personas. El uso de redes neuronales artificiales para el cálculo de valor energético puede brindar una respuesta más rápida y precisa para saber cuál es el requerimiento energético necesario que cada persona necesita de acuerdo al tipo de estilo de vida que lleva.

Por lo tanto, siguiendo los principios de la Organización Mundial de la Salud para el uso de inteligencia artificial es que, la presente investigación busca el diseño de un modelo mediante inteligencia artificial para el requerimiento energético en kcal para un estado de equilibrio y salud adecuados, adicionalmente se trabajará con una base de datos de personas entre 45 y 75 años. A partir de los resultados del trabajo de investigación nos permitirá proponer una interfaz para la evaluación de requerimiento energético de acuerdo al estado de cada paciente mejorando así la calidad de vida de la población peruana.

## **1.1. PLANTEAMIENTO DEL PROBLEMA**

¿En qué medida el diseño de un algoritmo para el cálculo de valor energético (Kcal) en personas de 45 a 75 años en el distrito de San Borja - Lima mediante inteligencia artificial puede mejorar su tipo de alimentación para una vida más saludable?

## **1.2. HIPÓTESIS**

El diseño de un algoritmo para el cálculo del valor energético (Kcal) en entre 45 a 75 años del distrito de San Borja – Lima mediante inteligencia artificial permitirá a la



población ver la condición en la que se encuentra y poder llegar a mantener un peso ideal con el fin de tener una vida más saludable.

### **1.3. OBJETIVO GENERAL**

Diseñar un algoritmo para calcular el valor energético (Kcal) en personas de 45 a 75 años del distrito de San Borja – Lima mediante inteligencia artificial.

### **1.4. OBJETIVOS ESPECIFICOS**

- Obtener y preprocesar una base de datos de personas de 45 a 75 años del distrito de San Borja – Lima.
- Diseñar y entrenar un algoritmo basado en la implementación de inteligencia artificial.
- Evaluar el modelo de inteligencia artificial mediante el ingreso de nuevo datos para corroborar su eficiencia.
- Diseñar un aplicativo web para cargar el algoritmo de inteligencia artificial para la visualización de las predicciones.



## CAPÍTULO II

### REVISIÓN DE LITERATURA

#### 2.1. ANTECEDENTES DE LA INVESTIGACIÓN

Bhatt et al. (2022) demuestra que el diagnóstico de la retinopatía diabética se basa en la morfología anormal de la vasculatura retiniana, sangrado, exudados, hemorragias, etc. En los cuales la inteligencia artificial puede respaldar la detección automática a través de técnicas de procesamiento de imágenes ayudando así a la detección de condiciones oculares y sistémicas. Por lo cual el uso de la inteligencia artificial sirve de herramienta para el personal de salud para evitar diagnósticos erróneos, reducción de tiempos de análisis e innovación en equipos médicos.

De la misma manera, Papathanail et al. (2021) menciona que un monitoreo continuo de la ingesta de nutrientes puede tener un rol fundamental en la reducción de riesgo nutricional del paciente, por lo cual la aplicación de un sistema basado en inteligencia artificial puede brindar la precisión de la ingesta de nutrientes automáticamente para cada tipo de alimento con un error del 15% mejorando así la precisión de los datos y la reducción de carga de los pacientes como los costos de salud.

Asimismo, Torres et al. (2020). en su conclusión al identificar y describir las aplicaciones prácticas que puede tener la inteligencia artificial en el área de la salud, afirman que la inteligencia artificial podría brindar apoyo en todo lo relacionado con la medicina de precisión, diagnóstico clínico, salud pública, eficiencia de procesos y apps en Salud. En otros casos el uso de la inteligencia artificial podría influir en la toma de decisiones del equipo médico debido a la evaluación rápida de ventajas y desventajas de la acción a tomar.



Por lo otro lado, Reaño. (2020) nos da mención de las ventajas que brinda la inteligencia artificial al campo de la medicina, tales como cambio de flujo de trabajo, mejora de eficiencia, refinamiento del manejo de datos, mayor precisión, efectividad en los servicios, mejor análisis y comprensión de datos. Con lo cual se define que el uso de las tecnologías modernas aporta positivamente en la salud. Asimismo, nos muestra que Colombia y Brasil son los países que investigan más acerca de la aplicación de inteligencia artificial en diagnósticos médicos en comparación a Ecuador, Suiza, Portugal, México y España (Reaño, 2020).

De la misma manera, Gonzales et al. (2017) nos muestran un panorama en donde la eficiencia de la inteligencia artificial en la creación de modelos para la detección y diagnóstico genera un fácil acceso a una atención médica de calidad en la automatización de procesos clínicos y como herramienta de apoyo al personal de salud en la consulta de conocimientos para la determinación de diagnósticos.

De igual modo, Navarro. (2017) muestra un panorama en donde la inteligencia artificial puede integrarse al uso de la medicina personalizada que en años posteriores no pudo ser accesible por la falta de tecnología, pero que hoy en día si se puede mediante el uso de características propias de cada individuo para mejorar la precisión de su diagnóstico y tratamiento como sus características de ADN, estilo de vida e historia clínica; como también la aplicación de inteligencia artificial en el uso de biomarcadores para el diagnóstico afecciones o enfermedades los cuales sean accesibles y útiles para su aplicación de la medicina personalizada.

Asimismo, Medinaceli & Silva. (2021) mencionan que el uso de la inteligencia artificial en la medicina es una herramienta muy práctica para el personal médico en los aspectos de detección de indicios de enfermedades, optimización del intercambio de



información de diferentes áreas y mejora del llenado de las historias clínicas. De esta manera, beneficiar al paciente con una información precisa mejorando así la eficiencia, eficacia y efectividad de los servicios médicos y garantizar la atención inmediata a las necesidades del paciente.

De igual manera, Bernaldo & Raymundo. (2020) explican que su proyecto está basado en la necesidad de brindar atención de calidad escalable y sostenible para la mejora del flujo de trabajo en el Hospital Cayetano Heredia pudiendo reducir los costos de tratamiento en un 50%. Para lo cual, el uso Chatbots implementados con inteligencia artificial a través de un aplicativo en el celular pueda realizar un diagnóstico preciso según la información que se ingrese. Asimismo, el paciente pueda consultar su diagnóstico en cualquier momento y poder agendar una cita.

Por otro lado, López. (2017) nos brinda un panorama donde explica que el 63% de la población entre 30 y 59 años de edad sufre de obesidad y sobrepeso. Asimismo, nos comenta el riesgo de la obesidad en la evolución de ciertas enfermedades y su relación con la tasa de mortalidad. Cabe resaltar que, en este trabajo de investigación se tomaron variables relacionadas con el paciente como edad, consumo de alcohol, consumo de frutas, nivel de educación y género.

Del mismo modo, Campos et al. (2023) nos explica en su análisis que el 44.4% de la población tienen prevalencia de sobrepeso y el 46.5% padece obesidad. Asimismo, nos hace entre ver que las personas con enfermedades crónicas más frecuentes tienden a ser más propensas a padecer de obesidad (Campos et al., 2023).

## **2.2. OBESIDAD**

La obesidad es un trastorno donde su principal rasgo característico es el aumento de grasa corporal, usualmente suele ser a consecuencia de ingerir una cantidad de calorías



por encima de lo que el cuerpo puede quemar por medio de ejercicio y actividades cotidianas. La obesidad es considerada la enfermedad del siglo 21, debido a la cantidad alarmante de personas que padecen esta condición. Asimismo, padecer obesidad aumenta exponencialmente el riesgo de tener otras enfermedades más agresivas que pongan en riesgo la vida (OMS, 2024).

Según la organización mundial de la salud, la obesidad aumenta el riesgo de contraer enfermedades como diabetes de tipo 2 y diferentes cardiopatías, estas enfermedades afectan en gran medida a la salud ósea y aumenta las posibilidades de la aparición de algunos determinados tipos de cáncer. Asimismo, la obesidad afecta en aspectos fundamentales de la calidad vida cotidiana como el sueño y el movimiento (OMS, 2024).

### **2.3. MALNUTRICIÓN**

La malnutrición a grandes rasgos es la consecuencia de la falta de nutrientes en el cuerpo. Es decir, ocurre cuando el cuerpo no obtiene los macronutrientes (carbohidratos, proteínas y grasas) necesarios que aporten energía al cuerpo ni los micronutrientes (vitaminas y minerales) que son necesarios para el desarrollo de físico y cognitivo. Según informes de la organización mundial de la salud, en el año 2022 aproximadamente 190 millones de personas a nivel mundial tenían un peso por debajo del índice normal. Asimismo, casi la mitad de las defunciones infantiles están relacionadas con la desnutrición dando como mayor concentración de estas muertes en los países bajos por el tema de la pobreza que multiplica en gran medida el riesgo de padecer de desnutrición y sus consecuencias (OMS, 2024).

## 2.4. REQUERIMIENTO ENERGÉTICO

El requerimiento energético se le llama a la cantidad de energía necesaria para lograr un equilibrio ideal en el balance energético para poder lograr una masa corporal ideal. Es decir, la cantidad necesaria que un organismo requiere para poder funcionar con normalidad. Según el Instituto Nacional de Estadística e Informática en el documento titulado “*Requerimiento calórico para la población peruana*” nos dice que, las necesidades energéticas de una persona están relacionadas con el gasto energético que la persona realiza, dando como recomendación calcularlos como múltiplo del metabolismo basal. Donde el metabolismo basal es la cantidad de energía necesaria bajo condiciones normales que el cuerpo requiere para sobrevivir (INEI, 2012).

### 2.4.1. Índice de Masa Corporal (IMC)

Huaroto & Policarpo. (2019) definen al IMC como un indicador que relaciona la talla y el peso para identificar el estado actual de una persona con respecto a su masa corporal.

Bauce. (2022) define al índice de masa corporal como un método eficiente porque muestra la correlación que presenta este índice con respecto a la grasa corporal y brinda un panorama del posible riesgo de la salud a nivel mundial. Asimismo, muestra una relevancia al hecho de adicionar medidas más selectivas como el hecho de cuantificar el grosor de la piel lo cual podría proporcionar información adicional.

### 2.4.2. Índice de Broca

En la búsqueda de métodos para definir el ideal de una persona es que, en el año 1871, Paul Broca ideó una fórmula para determinar el peso ideal con



referencia a la talla de una persona. Esta fórmula consiste en cuantificar la medida de la talla en centímetros y descontarle 100. Con lo que, el resultado de esta ecuación sería el peso ideal de una persona (Bauce, G., 2022).

## **2.5. PATRON DE CONSUMO ALIMENTARIO**

El patrón de consumo alimentario se basa en la frecuencia de consumo de alimentos consumidos y preparados por la persona o por el entorno que lo rodea. Asimismo, va en correlación a ciertos aspectos como hábitos, costumbres, tiempo, lugar que la persona tiene como rutina para poder satisfacer las necesidades alimentarias como sociales. Según los resultados obtenidos del estudio *“Patrón de consumo de alimentos a nivel familiar en zonas urbanas de Anzoátegui, Venezuela”*, podemos concluir que existen patrones de consumo alimentario entre diferentes familias según su estrato socioeconómico donde el 90% de la población estudiada tuvo una inclinación al consumo de alimentos procesados y/o industrializados, dando como los alimentos más frecuentes la harina y el café (Ekmeiro et al., 2015).

### **2.5.1. Transición Nutricional**

La transición nutricional es la definición de características como cambios del estado alimentario y nutricional de una determinada población estudiada. Es decir, son los cambios en el perfil nutricional de la persona, analizados en función a la modificación en el patrón alimenticio, actividad física y el estilo de vida que lleva la persona. En síntesis, la transición nutricional analiza los patrones de consumo de frutas, verduras, menestras, grasas y carbohidratos (Bernui, I. & Delgado, D., 2021).



### **2.5.2. Nivel de actividad física**

La actividad física es el movimiento corporal producido por el movimiento de los músculos que genera un gasto de energía por encima del nivel de reposo. De la misma manera reduce el riesgo de enfermedades cardiovasculares y diabetes muy aparte de la reducción de peso (Huaroto, C. & Policarpo, W., 2019).

### **2.5.3. Problemática Nutricional en Fumadores**

El consumo de tabaco afecta en el consumo alimenticio y su estado nutricional, esta situación genera la aparición de diversas enfermedades cardiovasculares, cáncer, osteoporosis, entre otros. Según el trabajo de investigación titulado “*Problemática nutricional en fumadores y fumadores pasivos*”, nos mencionan que existe un patrón en los hábitos alimenticios siendo la carne, la cafeína y el alcohol los alimentos mayormente consumidos por este grupo de personas. Asimismo, hacen énfasis en que si la situación actual continua, el riesgo de padecer enfermedades cardiovasculares, cáncer, cataratas, osteoporosis, entre otros son muy altas (Ortega et al., 2021).

## **2.6. REDES NEURONALES ARTIFICIALES**

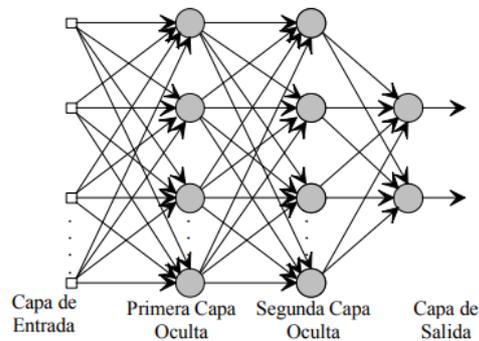
Las redes neuronales artificiales son modelos computacionales interconectadas entre sí que hacen uso de la estructura y procesamiento de datos de manera similar al cerebro humano (Serrano et al., 2009).

### **2.6.1 Redes Neuronales no Recurrentes**

En este tipo de conexión de red neuronal, la propagación de las señales avanza en un solo sentido no permitiendo la posibilidad de retroalimentación.

**Figura 1**

*Esquema de Red Neuronal*



Nota: (Serrano et al., 2009)

### 2.6.2. Modelo de Regresión

El modelo de regresión es una técnica que se enfoca en el análisis de datos para predecir un valor con nuevos datos incognitos basándose en una ecuación lineal, es decir, es usado para encontrar la relación de una variable dependiente “y” a partir de variables independientes “x”. Al proceso de ajustar una función a partir de variables independientes de entrada se le conoce como fase de entrenamiento correspondiente a la creación de una red neuronal donde el modelo aprenderá a partir de los datos de entrada “x” y “y” (Menacho, C., 2014).

## Figura 2

*Esquema representativo de X y Y*

X	Y
3	9
6	10
8	7
2	7
5	15
6	5
1	4

Nota: Elaboración Propia

### 2.6.3. Funciones de Activación

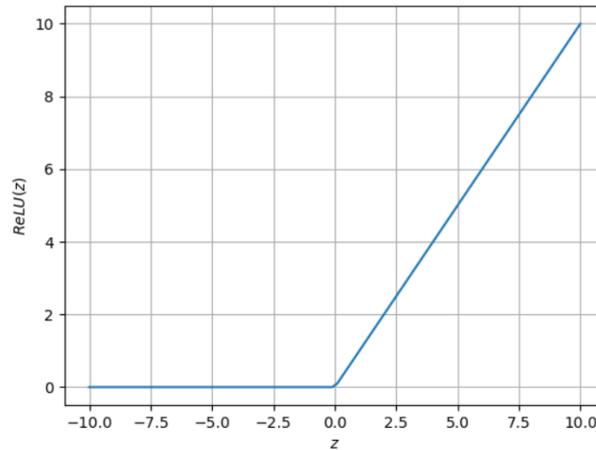
Las funciones de activación son modelos matemáticos aplicados en la salida de cada red neuronal con la finalidad de ajustar el modelo a la función deseada. Esto permite a la red neuronal aprender y modelar los datos (Serrano et al., 2009-2010).

- **Función ReLu (Rectified Lineal Unit)**

Esta función introduce una salida igual a cero cuando el valor de entrada ( $z$ ) es negativa, y una salida de igual valor a la entrada cuando el valor de entrada ( $z$ ) es positiva. Al no haber presencia de saturación el modelo del gradiente decreciente concurre más velozmente haciendo más fácil el proceso de aprendizaje (Sotaquirá, 2018).

### Figura 3

#### *Función de activación ReLu*



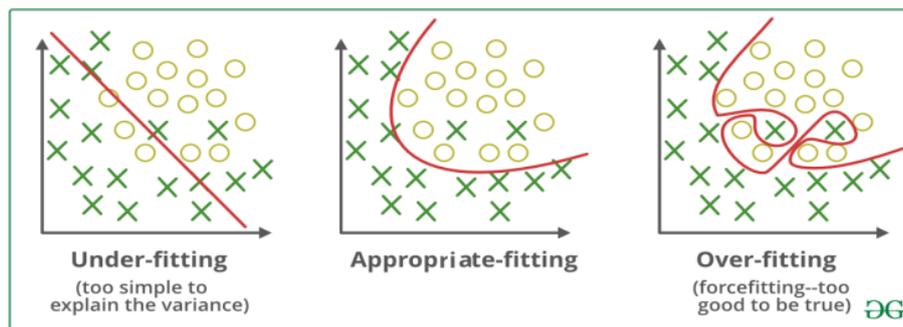
Nota: (Sotaquirá, 2018)

#### **2.6.4. Entrenamiento de la red neuronal**

Para iniciar con el proceso de entrenamiento de la red neuronal es necesario contar con una base de datos preprocesada con anterioridad, seguidamente se carga la base de datos y finalmente se aplica la función "train\_test\_split" para separar los datos de entrenamiento con los de test. Asimismo, es necesario abordar ciertos aspectos que son de importancia al momento de iniciar el proceso de entrenamiento, este aspecto es conocido como overfitting (sobreajuste o memorización), en donde la red neuronal no aprende sino por el contrario memoriza los datos de entrenamiento siendo incapaz de mostrar resultados para datos diferentes; otro problema también es el underfitting (poco ajuste) esto sucede cuando el modelo es incapaz de brindar una correlación entre los datos de entrada, es decir, la red neuronal es muy simple para definir una correlación entre los datos de entrada con respecto a su salida (Raschka et al., 2022).

**Figura 4**

*Gráficos de underfitting y overfitting*



Fuente: (Raschka et al., 2022)

## 2.7. PYTHON

Python es un lenguaje de programación de muy alto nivel y de código abierto. Su creador fue Guido Van Rossum y su primera aparición fue en el año 1991. En la actualidad, Python es uno de los lenguajes más usados en el mundo debido a sus instrucciones simples y su amplia gama de librerías además de su amplia variedad de áreas de aplicación. Asimismo, Python optimiza el campo visual de datos grandes, procesamiento de matrices de datos y la realización del cálculo de valores numéricos rápidamente (Raschka et al., 2022).

El uso de Python es muy versátil y es usado para una gran gama de aplicaciones:

- Diseño de Aplicaciones Web
- Ciencia de Datos
- Inteligencia Artificial
- Automatización de Tareas
- Desarrollo de Videojuegos
- Entre otros



Por lo anteriormente mencionado podemos afirmar que el lenguaje Python es multiplataforma. Por lo tanto, se puede ejecutar en sistemas operativos como Linux, Windows y MacOS. Cabe resaltar que la comunidad de Python es una de las comunidades más grandes y activas a nivel mundial. Esta comunidad está compuesta por personas de diferentes áreas como desarrolladores, programadores, ingenieros de datos, docentes, profesionales y aficionados de distintas áreas que un sentimiento de interés común en el lenguaje y sus distintas aplicaciones.

### **2.7.1. Flask**

Flask es un micro framework que nos permite crear de forma eficaz aplicaciones web mediante el lenguaje Python. Asimismo, Flask cuenta con un servidor web de desarrollo donde es posible correr el aplicativo web que se está desarrollando con la finalidad de visualizar los resultados. Por otro lado, dispone de un depurador y soporte integrado para corregir de manera rápida posibles errores dentro del código que se está desarrollando (Grinberg, M., 2018).

Por último, Flask tiene diferentes extensiones las cuales se pueden instalar dependiendo a los requerimientos del proyecto, las usadas son:

- Flask-script
- Flask-WTF
- Flask-bootstrap
- Flask-login
- Flask-Sqlalchemy

### 2.7.2. Jinja

Jinja es un framework de Python usado en el diseño de páginas y aplicaciones web. También definida como una dependencia, su uso se basa en la generación y control de plantillas “templates” con la finalidad de desarrollar programas con el lenguaje Python. Asimismo, la función de Jinja se encarga de llevar data por la plantilla para poder renderizarla al final del documento. La estructura de control de Jinja está conformada por variables como “for”, “if”, “macros”, “call”, entre otros. Las cuales tienen como función controlar el flujo del programa y las condiciones establecidas dentro del código (Grinberg, M., 2018).

#### Figura 5

*Método para trabajar con la variable "for"*

```
{% for i in range(1, 12) %}  
  
{% endfor %}
```

Nota: Elaboración Propia

## 2.8. LIBRERÍAS

### 2.8.1. Tensorflow

Tensorflow es una librería de código abierto desarrollada por la empresa Google en el año 2015. Una de las bondades de esta librería es la facilidad de la creación de modelos de aprendizaje automático, es decir, esta librería es usada en la aplicación de inteligencia artificial.

Asimismo, Tensorflow tiene la capacidad de realizar diagramas de una gran cantidad de flujos de datos que describen el desplazamiento de los datos de



puntos de procesamiento. Cabe resaltar que, cada punto expresa un cálculo matemático y la conexión entre cada punto son los llamados tensores (Chollet, F., 2018).

Actualmente la versión más estable de Tensorflow es la versión 2.0 que se caracteriza por la facilidad y simplicidad de su uso. De la misma manera, la incorporación de APIs que facilita la puesta en marcha de modelos de Machine Learning entrenados mediante servidores, web o dispositivos móviles.

El desarrollo de Tensorflow implica la incorporación de nodos y capas interconectadas entre sí, esto es llevado a cabo gracias a la librería de keras. Cabe resaltar que, la API de keras es una herramienta simple y accesible con la que es muy fácil crear y entrenar modelos de aprendizaje profundo de manera óptima (Chollet, F., 2018).

### **2.8.2. Keras**

En términos prácticos, Keras es una librería de código abierto desarrollada en Python para facilitar y optimizar el desarrollo de modelos de aprendizaje profundo. El objetivo está enfocado en tres 3 puntos importantes: usabilidad, modularidad y extensibilidad; lo que lo hace una herramienta muy versátil en el aprendizaje profundo e inteligencia artificial. Asimismo, Keras proporciona al desarrollador una amplia selección de capas y módulos preexistentes para la creación de arquitecturas de redes neuronales. Asimismo, permite a los desarrolladores modificar o ampliar la funcionalidad según sea conveniente (Chollet, F., 2018).



### 2.8.3. Numpy

Numpy es una librería de Python usada normalmente en la realización de operaciones numéricas y computacionales en matrices y arreglos multidimensionales. Además, Numpy brinda una estructura de datos denominada “ndarray”, que es un array multidimensional donde su principal función es almacenar elementos de un solo tipo. Asimismo, permite el acceso para modificar un elemento en específico haciendo uso de técnicas indexación y segmentación. Numpy hace uso del término llamado “broadcasting” con la finalidad de realizar arreglos de formas y tamaños de manera óptima y eficiente, lo que facilita la estructura del código para que este sea conciso y eficiente (Chollet, F., 2018).

### 2.8.4. Matplotlib

Así como Numpy o Keras, matplotlib es una librería en lenguaje Python usada para la creación de gráficos personalizados, de esta manera poder adaptarse a las necesidades requeridas. Matplotlib puede integrarse sin problemas con Numpy, lo que permite la visualización de los datos almacenados dentro de los arrays, de esta manera poder realizar operaciones numéricas con dichos datos (Raschka, S. & Mirjalili, V., 2019).

### 2.8.5. Train\_test\_split

“Train\_test\_split” es una función dentro de la librería scikit.learn diseñado para realizar una división de un conjunto de datos en subconjuntos aleatorios de train y test. Esta función estratifica las proporciones por defecto en caso no se le asigne una, el estándar ideal es 75 – 70% para “train” y 25 – 30% para “test” (Raschka, S. & Mirjalili, V., 2019).

## Figura 6

### *Código de train\_test\_split*

```
# Separar en datos de entrenamiento y datos de prueba  
X_train, X_test, y_train, y_test = train_test_split(X, y)
```

Nota: Elaboración Propia

### 2.8.6. Sequential

“Sequential” es un modelo secuencial dentro de la librería Tensorflow, es usado con la finalidad de crear modelos de redes neuronales las cuales estarán conectadas solo con la capa adyacente y poder entrenarlos de manera eficiente y optima (Raschka, S. & Mirjalili, V., 2019).

### 2.8.7. Sklearn.linear

Dentro de la biblioteca de “sklearn.linear” existe los módulos “model”, “ensemble” y “tree”. El módulo “model” ofrece herramientas con las se puede realizar modelos de regresión lineal, caracterizados por emplear aprendizaje supervisado para el modelamiento de una variable dependiente con respecto a variables dependientes. Algunos de los modelos dentro de este módulo son “LinearRegression”, “Rigde”, “Lasso” y “LogisticRegression”. El módulo “tree” dota de herramientas para trabajar con los denominados “arboles de decisión”, como también problemas de “clasificación”. Algunos modelos son “DecisionTreeClassifier” y “DecisionTreeRegressor”. (Géron, A., 2019).

El módulo “ensemble” hace uso de técnicas que combinan diferentes modelos de aprendizaje automático de manera que aumente el rendimiento y reduzca el sobreajuste del modelo, son usados tanto para problemas de regresión



y clasificación. Dentro de este modelo están “RandomForestRegressor” y “RandomForestClassifier” (Arif et al., 2023).

#### **2.8.8. Xgboost**

La biblioteca “xgboost” al igual que la biblioteca “sklearn.linear” es usada para problemas de regresión y clasificación. Esta biblioteca realiza la combinación de diferentes arboles de decisión con la finalidad de generar un modelo más optimo (Arif et al., 2023).

#### **2.8.9. Dense**

En términos prácticos “Dense” es una capa que está totalmente conectada a la red neuronal. Es decir, cada neurona está conectada a todas las neuronas de una capa anterior; estas conexiones son ponderadas en pesos que cambian en el proceso de entrenamiento de la red neuronal. Cabe resaltar que, esta capa es usada dentro de las capas ocultas o en la capa de salida de la red neuronal (Chollet, F., 2018).

#### **2.8.10. Normalization**

“Normalization” proporciona una mejora en la estabilidad y rendimiento del modelo de red neuronal durante la fase de entrenamiento. Es usado en la construcción de modelos de redes neuronales usando las librerías de Tensorflow y Keras (Chollet, F., 2018).

### **2.9. DATASET**

En términos de ciencia de datos, “dataset” se le denomina al grupo de datos obtenidos para entrenar modelos de aprendizaje automático o redes neuronales, evaluar



su precisión y rendimiento. Asimismo, es importante mencionar que existen base de datos liberadas donde la información puede estar incompleta o tenga datos adicionales. Para poder hacer frente a estos problemas es necesario hacer una limpieza de los datos antes de empezar a crear la red neuronal, a este proceso se le denomina como ingeniería de datos (Vanderplas, J., 2016).

## **2.10. ANACONDA NAVIGATOR**

Anaconda Navigator es una interfaz de código abierto y de fácil uso que contiene una gran variedad de aplicaciones, librerías y demás comandos enfocados en el desarrollo de ciencia de datos a través del lenguaje Python. Asimismo, Anaconda Navigator anula la necesidad de realzar comandos a través de la terminal y permite la búsqueda, instalación, ejecución y actualización de librerías de forma sutil. Cabe resaltar que, una de las aplicaciones dentro de Anaconda Navigator para el procesamiento y análisis de datos, así como para la creación de inteligencia artificial es Jupyter Notebook. Esta aplicación brinda un entorno donde el desarrollador puede crear un código y ejecutarlo de forma individual o correlativa. Esto permite una visualización interactiva de los datos y la capacidad de revisar el código paso a paso para poder corregir errores en caso existiera (Vanderplas, J., 2016).

## **2.11. VISUAL STUDIO CODE**

Visual Studio Code es un editor de código abierto desarrollado por la empresa Microsoft. Es el software más preferido por los desarrolladores debido a las herramientas que nos brinda como la edición del código y autocompletado que nos permite agilizar la escritura del código. Asimismo, al momento de digitar el código la interfaz te brinda sugerencias de acuerdo a los comandos que estas usando con el fin de dar asistencia al desarrollador. Asimismo, Visual Studio Code está equipada con una terminal que puede



hacer uso de cualquier Shell instalado en el equipo y gestionar y ejecutar distintos comandos que sean requeridos por el desarrollador. Además, el uso de Visual Studio Code no está limitado al desarrollo de un solo tipo de aplicación, puede ser usado para la creación de interfaces destinados a la aplicación de inteligencia artificial, conexiones remotas hacia máquinas virtuales, contenedores y Windows Subsystem for Linux (WSL) (Johnson, B., 2020).

## **2.12. LENGUAJE CSS**

El lenguaje CSS (Cascading Style Sheets o Hojas de estilos en cascada) es una tecnología enfocada en el diseño visual y estético de una página web, su programación puede ser escrita en inglés o alemán. Su principal característica es la aplicación de formas, colores, márgenes, entre otros a uno o varios documentos HTML y páginas web de manera automática. Es decir, el documento de CSS contiene toda la presentación del contenido de un formulario HTML (Duckett, J., 2011).



## CAPÍTULO III

### MATERIALES Y MÉTODOS

#### 3.1. MATERIALES

##### 3.1.1. Hardware

Laptop

- Ejemplar: LAPTOP-MJIBNJS8
- Procesador: AMD Ryzen 7 6800H with Radeon Graphics 3.20 GHz
- Memoria instalada (RAM) 8.00GB de RAM
- Sistema: Sistema Operativo de 64 bits con Windows 11 Home

##### 3.1.2. Software

- Sistema Operativo de 64 bits Windows 11 home
- Microsoft Office LTSC Profesional Plus 2021
- Anaconda Navigator – Jupyter Notebook 6.5.4

#### 3.2. DISEÑO Y NIVEL DE LA INVESTIGACIÓN

##### 3.2.1. Diseño de la investigación

El presente proyecto de investigación tiene un enfoque cuantitativo y diseño no experimental debido al hecho de la recolección de datos en un tiempo y momento únicos. Con la finalidad de realizar un análisis de su incidencia e interrelación (Sampieri et al., 2018).

En primer lugar, para la recopilación y selección de datos para el modelo; es necesario considerar el uso de una base de datos para la obtención de una



muestra grande y representativa de información de personas de 45 a 75 años del distrito de San Borja - Lima.

En segundo lugar, para tratar con problemas como datos incompletos, datos con valores atípicos, datos inconsistentes y/o datos con tamaños excesivos existentes dentro de la base de datos. Es importante el uso de técnicas de preprocesamiento datos; con la aplicación de limpieza de datos es posible corregir datos erróneos, filtrar datos incorrectos, detectar y revolver discrepancias. De esta manera obtener datos de calidad que tengan validez, consistencia, uniformidad, densidad y unicidad.

En tercer lugar, al ser una base de datos con valores numéricos y texto. Es imprescindible la transformación de los datos en forma de texto a numéricas. Mediante el proceso de discretización de variables es posible convertir variables continuas en variables categóricas, de esta manera hacer más factible y optimo el análisis de los datos.

En cuarto lugar, seleccionar y entrenar el modelo de aprendizaje automático apropiado para el problema genera la disponibilidad de diferentes recursos que brindan orientación sobre como seleccionar y entrenar modelos de aprendizaje automático practico con librerías como Scikit-Learn, Keras y TensorFlow de Aurélien Géron, las cuales son herramientas muy prácticas. Aurélien Géron nos dice que el uso de librerías como Sckit-learn, Keras y TensorFlow, permiten el diseño y entrenamiento de las redes neuronales artificiales. Estas librerías usan algoritmos matemáticos para la detección de patrones en los datos de entrada que se pueden usar para llegar a los datos de salida.



En quinto lugar, en la fase de entrenamiento, las primeras predicciones tienen la posibilidad de tener un gran porcentaje de ser erróneas, pero al repetirse el proceso de entrenamiento  $n$  veces con la cantidad de  $n$  datos. El porcentaje de error de la predicción tiende a bajar según los patrones analizados.

En sexto lugar, la validación y evaluación del modelo; para garantizar el rendimiento óptimo del modelo de aprendizaje es importante el uso de librerías como “scikit.learn” para poder separar el conjunto de datos en dos partes una para entrenar y otra para validar el modelo.

Finalmente, el uso de librerías como flask, keras y numpy así como el uso de “listas”, “array” y “formularios”, los cuales son fundamentales para el diseño del aplicativo web y poder interactuar con el modelo de inteligencia artificial para realizar nuevas predicciones.

### **3.2.2. Nivel de la investigación**

La esencia de un proyecto de investigación es la búsqueda del desarrollo de nuevas tecnologías mediante el enfoque de un tema en concreto. Por ende, el presente proyecto de investigación es de nivel exploratorio, debido a la característica de buscar nuevos enfoques dedicados al desarrollo de nuevas tecnologías aplicables a distintas áreas (Sampieri et al., 2018).

### **3.3. POBLACIÓN Y MUESTRA DE LA INVESTIGACIÓN**

Para el desarrollo del presente proyecto de investigación es necesario el uso una base de datos liberada por el gobierno peruano a través de la Plataforma Nacional de Datos Abiertos titulada “Censo de Salud y Determinantes del Distrito de San Borja [Municipalidad Distrital de San Borja - MSB]”, donde se puede apreciar la información



básica de 50495 personas diferentes edades. Por lo tanto, la población es finita y está conformada por los datos contenidos dentro de la base de datos abierta descrita anteriormente.

Siendo un proyecto enfocado en el uso de inteligencia artificial para la creación de nuevas tecnologías; tener una base de datos lo suficientemente grande puede ayudar al rendimiento óptimo, la generalización del algoritmo, la facilidad de abarcar mayores escenarios posibles, la reducción del riesgo de sesgo y mayor estabilidad frente a las variaciones de los datos.

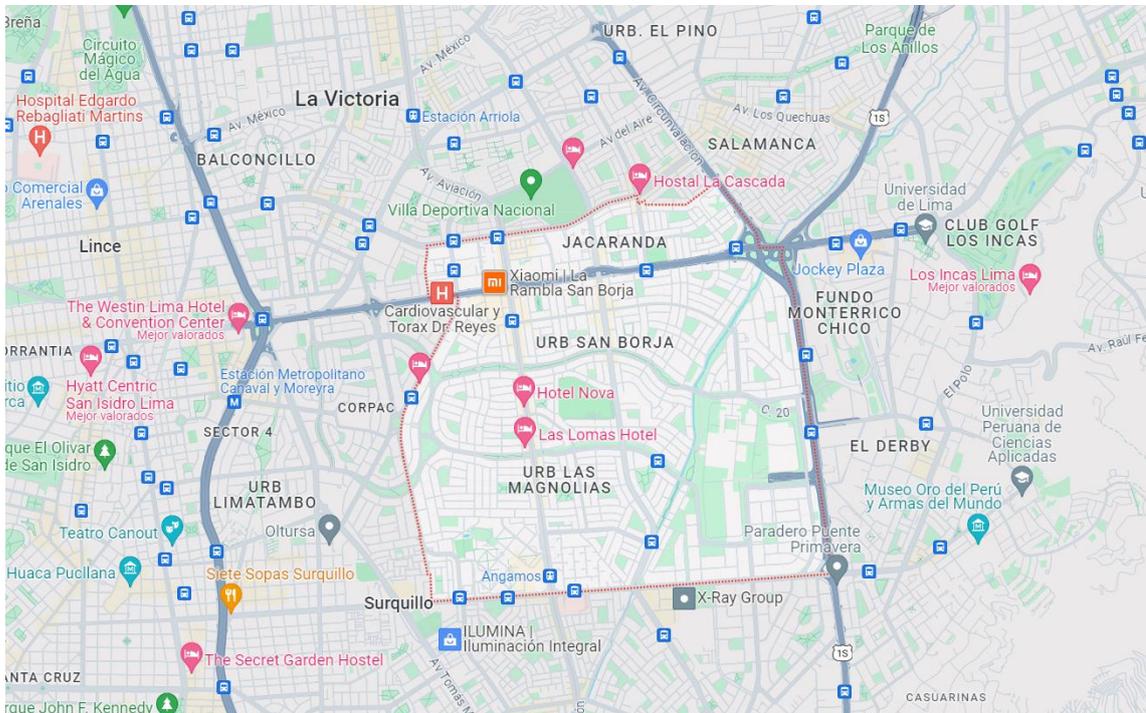
Por lo tanto, realizando un proceso de filtrado y preprocesamiento de los datos según las limitantes del proyecto y la necesidad de datos primordiales como la edad, talla y peso, nos quedaría una población total de 13339 personas lo que vendría a ser los datos de muestra.

### **3.4. UBICACIÓN DE LA INVESTIGACIÓN**

La presente investigación tendrá como lugar de ejecución la ciudad de Puno, Provincia y Departamento de igual nombre. La investigación tendrá como referencia la base de datos del censo de la población de habitantes de 45 a 75 años de edad del Distrito de San Borja Departamento de Lima llevada a cabo en el periodo 2020 hasta 2022.

## Figura 7

*Ubicación por mapa satelital del Distrito de San Borja del Departamento de Lima*



Nota: Elaboración Propia

### 3.5. TÉCNICAS E INSTRUMENTOS PARA LA RECOLECCIÓN DE DATOS

#### 3.5.1. Técnicas

Para la recolección de datos existen diferentes estrategias como observación, encuestas, entrevistas, entre otros. Sin embargo, al usar una base de datos ya existente el uso de técnicas para la recolección de datos es innecesaria debido a que los datos ya están a nuestra disponibilidad.

Con lo mencionado anteriormente, la base de datos utilizada fue obtenida de la Plataforma Nacional de Datos Abiertos del Estado Peruano, el título de esta base de datos es “Censo de Salud y Determinantes del Distrito de San Borja



(Municipalidad Distrital de San Borja – MSB)” con extensión “csv”  
Municipalidad de San Borja (2022).

De la misma manera, siguiendo los principios de ética relacionada con el uso de datos de una base de datos. El presente proyecto de investigación no usa información sensible o confidencial que pueda vulnerar los derechos de ninguna persona.

### **3.5.2. Instrumentos**

Para la base de datos se aplicó ingeniería de datos para limpiar, recopilar y preprocesar los datos según las limitantes del presente proyecto de investigación. De esta manera, se asegura que los datos filtrados tengan validez, objetividad y confiabilidad.



## CAPÍTULO IV

### RESULTADOS Y DISCUSIÓN

#### 4.1. ADQUISICION DE DATOS

Municipalidad de San Borja. (2022) contiene la información de 50494 personas, el formato en el que se encuentra es de tipo “csv” delimitado por comas.

La información contenida muestra los siguientes datos:

- Departamento
- Provincia
- Distrito
- Tipo de vivienda
- Tenencia de vivienda
- Sexo
- Estado civil
- Nivel de educación
- Uso/exposición al tabaco
- Enfermedad
- Lugar de trabajo
- Actividad física
- Consumo de frutas
- Consumo de verduras
- Peso actual
- Talla actual
- Tenencia de seguro

- Edad

## Figura 8

*Muestra de los datos obtenidos de la base de datos en formato delimitado por comas*

	A	B	C	D	E	F	G
50330	20221020,20220712,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Departamento en block o torre de					
50331	20221020,20220712,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Departamento en block o torre de					
50332	20221020,20220712,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Habitación o departamento indep					
50333	20221020,20220712,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Habitación o departamento indep					
50334	20221020,20220713,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Casa independiente,Propia totalm					
50335	20221020,20220713,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Casa independiente,Propia totalm					
50336	20221020,20220713,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Casa independiente,Propia totalm					
50337	20221020,20220713,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Departamento en block o torre de					
50338	20221020,20220713,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Departamento en block o torre de					
50339	20221020,20220713,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Departamento en block o torre de					
50340	20221020,20220713,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Habitación o departamento indep					
50341	20221020,20220714,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Casa independiente,Propia totalm					
50342	20221020,20220714,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Departamento en block o torre de					
50343	20221020,20220714,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Departamento en block o torre de					
50344	20221020,20220714,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Habitación o departamento indep					
50345	20221020,20220714,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Habitación o departamento indep					
50346	20221020,20220715,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Casa independiente,Propia totalm					
50347	20221020,20220715,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Departamento en block o torre de					
50348	20221020,20220715,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Departamento en block o torre de					
50349	20221020,20220715,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Habitación o departamento indep					
50350	20221020,20220715,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Habitación o departamento indep					
50351	20221020,20220718,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Casa independiente,Propia totalm					
50352	20221020,20220718,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Casa independiente,Propia totalm					
50353	20221020,20220718,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Casa independiente,Propia totalm					
50354	20221020,20220718,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Casa independiente,Propia totalm					
50355	20221020,20220718,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Departamento en block o torre de					
50356	20221020,20220718,LIMA,LIMA,SAN	BORJA,150130,Vivienda ocupada con personas presentes,Departamento en block o torre de					

Nota: Plataforma Nacional de Datos Abiertos

Realizando un proceso de importación de los datos del archivo en formato delimitado por comas a un formato Excel estándar para mejorar la visualización de los datos se puede apreciar que:

- Algunos de los datos son innecesarios según las limitantes del proyecto
- Existe la ausencia de datos en ciertos espacios

## Figura 9

*Muestra de los datos en formato estándar*

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
DEPARTAM	PROVINCIA	DISTRITO	TIPO_VIVIE	TENENCIA	SEXO	EDAD	ESTADOC	NIUML	EDU	USO_EXPC	ENFERMED	DONDE_TR	DIAS_ACTH	DIAS_CON	DIAS_COM	PESO_ACT	TALLA_AC	TENENCIA_SEGURO
1	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Femenino	68	Vuolvo / Vuc	Superior un	Ninguno				3 días	7 días			Si
2	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Femenino	0	Inicial	Ninguno					2 días	6 días			No
3	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Femenino	79	Separado /	Superior un	Ninguno				2 días	6 días			Si
4	LIMA	LIMA	SAN BORJA	Habitación	Propia pagu	Masculino	40	Casado / C	Superior un	Ninguno				1 día	7 días			Si
5	LIMA	LIMA	SAN BORJA	Habitación	Propia pagu	Femenino	43	Casado / C	Superior un	Ninguno			Su casa o desde casa	5 días	0 días			Si
6	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Masculino	30	Soltero / So	Superior un	Ninguno				3 días	7 días			Si
7	LIMA	LIMA	SAN BORJA	Habitación	Propia pagu	Femenino	9	Primaria	Ninguno					1 día	7 días			Si
8	LIMA	LIMA	SAN BORJA	Habitación	Propia total	Masculino	67	Casado / C	Superior un	Ninguno	Diabetes o /	Fuera de casa		4 días	0 días			Si
9	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Femenino	32	Divorciado /	Superior un	Ninguno				2 días	0 días			Si
10	LIMA	LIMA	SAN BORJA	Habitación	Propia pagu	Femenino	46	Casado / C	Superior no	Ninguno				2 días	1 día			No
11	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Femenino	11	Inicial	Ninguno					2 días	6 días			No
12	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Masculino	44	Casado / C	Superior un	Ninguno				2 días	6 días			Si
13	LIMA	LIMA	SAN BORJA	Habitación	Propia total	Masculino	30	Soltero / So	Postgrado	Ninguno			Fuera de casa	4 días	0 días			Si
14	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Femenino	30	Soltero / So	Superior un	Ninguno				2 días	0 días			Si
15	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Femenino	60	Casado / C	Superior no	Ninguno	Colesterol o triglicéidos altos			2 días	0 días			Si
16	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Masculino	0	Ninguno	Ninguno					2 días	2 días			Si
17	LIMA	LIMA	SAN BORJA	Habitación	Propia pagu	Masculino	45	Casado / C	Postgrado	Ninguno			Su casa o desde casa	5 días	0 días			Si
18	LIMA	LIMA	SAN BORJA	Habitación	Propia total	Femenino	65	Casado / C	Superior un	Ninguno	Diabetes o /	Fuera de casa		4 días	0 días			Si
19	LIMA	LIMA	SAN BORJA	Habitación	Propia total	Femenino	86	Vuolvo / Vuc	Secundaria	Ninguno	Presión alta o hipertensión			2 días	0 días			No
20	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Masculino	47	Casado / C	Superior no	Ninguno	Presión alta	Fuera de casa		5 días	2 días			No
21	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Femenino	26	Soltero / So	Primaria	Expuerto secundario			Su casa o desde casa	5 días	1 día			Si
22	LIMA	LIMA	SAN BORJA	Departam	Propia total	Femenino	33	Soltero / So	Superior un	Ninguno				3 días	7 días			Si
23	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Femenino	19	Soltero / So	Superior un	Ninguno				2 días	6 días			No
24	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Masculino	45	Casado / C	Secundaria	Ninguno				2 días	6 días			Si
25	LIMA	LIMA	SAN BORJA	Habitación	Propia pagu	Femenino	39	Casado / C	Superior un	Ninguno	Colesterol o triglicéidos altos			1 día	7 días			Si
26	LIMA	LIMA	SAN BORJA	Habitación	Propia pagu	Masculino	0	Primaria	Ninguno					1 día	7 días			Si
27	LIMA	LIMA	SAN BORJA	Habitación	Propia total	Masculino	32	Soltero / So	Postgrado	Ninguno			Fuera de casa	4 días	0 días			Si
28	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Femenino	75	Casado / C	Superior un	Ninguno	Colesterol o triglicéidos altos, Presión			1 día	2 días			Si
29	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Masculino	72	Casado / C	Superior un	Ninguno	Presión alta o hipertensión			1 día	2 días			Si
30	LIMA	LIMA	SAN BORJA	Habitación	Propia pagu	Masculino	13	Soltero / So	Secundaria	Ninguno				2 días	1 día			Si
31	LIMA	LIMA	SAN BORJA	Habitación	Propia pagu	Masculino	20	Soltero / So	Superior un	Cigaro, puro, pipa				2 días	1 día			Si
32	LIMA	LIMA	SAN BORJA	Habitación	Propia total	Masculino	78	Vuolvo / Vuc	Secundaria	Ninguno	Colesterol o triglicéidos altos			5 días	4 días			Si
33	LIMA	LIMA	SAN BORJA	Habitación	Propia total	Femenino	61	Casado / C	Postgrado	Ninguno			Su casa o desde casa	2 días	0 días			Si
34	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Masculino	50	Casado / C	Secundaria	Cigaro, puro, pipa				2 días	4 días			No
35	LIMA	LIMA	SAN BORJA	Habitación	Propia pagu	Femenino	0	Inicial	Ninguno				Fuera de ca	1 día	5 días			Si
36	LIMA	LIMA	SAN BORJA	Habitación	Propia total	Masculino	55	Soltero / So	Superior un	Ninguno				2 días	2 días			Si
37	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Femenino	39	Casado / C	Postgrado	Ninguno				2 días	2 días			Si
38	LIMA	LIMA	SAN BORJA	Casa indepe	Propia total	Masculino	75	Casado / C	Superior un	Ninguno				2 días	2 días			Si
39	LIMA	LIMA	SAN BORJA	Casa indepe	Propia pagu	Femenino	9	Primaria	Ninguno				Su casa o desde casa	2 días				Si
40	LIMA	LIMA	SAN BORJA	Casa indepe	Propia pagu	Masculino	11	Primaria	Ninguno				Su casa o desde casa	2 días				Si
41	LIMA	LIMA	SAN BORJA	Casa indepe	Propia pagu	Masculino	19	Soltero / So	Superior no	Ninguno				2 días				No
42	LIMA	LIMA	SAN BORJA	Casa indepe	Propia pagu	Masculino	40	Casado / C	Superior un	Ninguno				2 días				Si

Nota: Plataforma Nacional de Datos Abiertos

## 4.2. PROCESO DE CÁLCULO DE PESO IDEAL

Teniendo en cuenta la base de datos resultante antes del proceso de codificación (figura 9). Se procede a realizar la evaluación a cada individuo según sus datos, para lo cual, es necesario identificar su estado actual según su índice de masa corporal.

### 4.3.1. Evaluación del Índice de Masa Corporal

Para la evaluación del índice de masa corporal es necesario hacer uso de la formula del IMC (Índice de Masa Corporal):

## Figura 10

*Fórmula de cálculo del índice de masa corporal*

$$IMC = PESO/ALTURA^2$$

Nota: (Huaroto, C. & Policarpo, W.,2019)



Donde:

- Peso tiene como unidad de medida kilogramo
- Altura tiene como unidad de medida metro

Según resultado obtenido de la formula y comparándolo según su clasificación del índice de masa corporal podemos definir es estado actual de la persona.

**Tabla 1**

*Índice de Masa Corporal en Adultos*

IMC (kg/m <sup>2</sup> )	Categoría
< 16.5	Desnutrición Severa
>=16.5 - <18.5	Desnutrición Moderada
>=18.5 - <25	Peso Normal
>=25 - <30	Sobrepeso T1
>=30 - <35	Obesidad T1 (moderada)
>=35 - <40	Obesidad T2 (severa)
>=40	Obesidad T3 (mórbida)

Nota: OMS

Realizando este proceso de cálculo es posible obtener el estado actual de cada persona dentro de la base de datos, dando evidencia la existencia de una cantidad alarmante de casos de personas con estado de sobrepeso.

### Figura 11

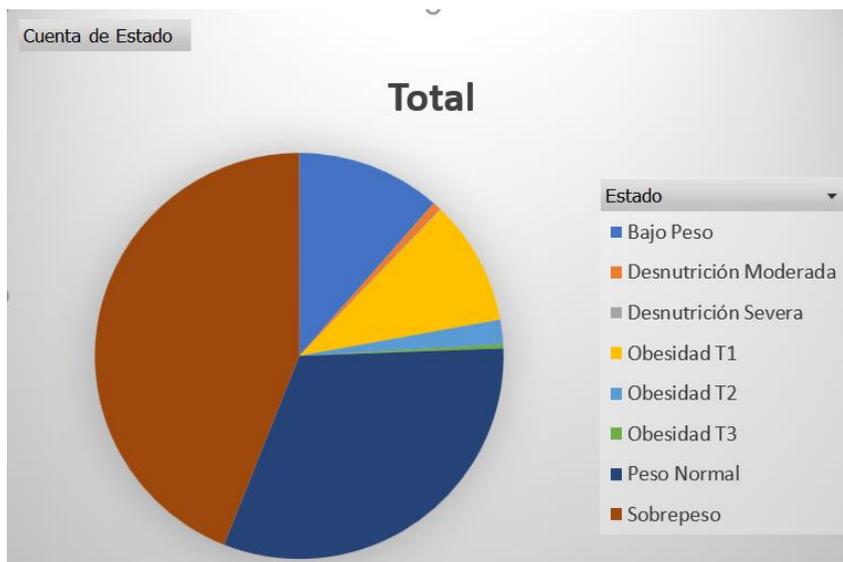
*Resultados del IMC a la base de datos*

Bajo Peso	1528
Desnutrición Moderada	92
Desnutrición Severa	6
Obesidad T1	1326
Obesidad T2	259
Obesidad T3	45
Peso Normal	4215
Sobrepeso	5867

Nota: Elaboración Propia

### Figura 12

*Imagen gráfica de los resultados del IMC a la base de datos*



Nota: Elaboración Propia

#### 4.3.2. Cálculo de Peso Ideal

Para el cálculo del peso ideal es necesario el uso de la fórmula de Broca.

### Figura 13

Fórmula de Broca

$$PI = Talla(cm) - 100$$

Nota: OMS

Haciendo uso de esta fórmula, el cálculo del peso ideal se obtiene de manera rápida para todos los casos presentes en la base de datos.

### Tabla 2

*Muestra de los resultados del cálculo de Peso Ideal*

Talla	Peso Ideal
160	60
150	50
148	48
148	48
150	50
180	80
160	60
170	70
170	70
174	74

Nota: Elaboración Propia

Seguidamente, comparamos el peso de la persona con respecto al peso ideal para determinar el peso excedente o faltante. Asimismo, multiplicamos este resultado por el factor de peso actual para hallar el porcentaje de peso actual. Para definir el factor de multiplicación se usó la media de los valores indicados según el método general de cálculo de requerimientos nutricionales.

**Tabla 3**

*Método general de cálculo de requerimientos nutricionales*

Resultados Deseados	Factor de Multiplicación
Perdida de Peso	20 - 25 cal/Kg de Peso Corporal
Mantener Peso	25 - 30 cal/Kg de Peso Corporal
Ganancia de Peso	30 - 35 cal/Kg de Peso Corporal

Nota: OMS

**Tabla 4**

*Muestra de los resultados obtenidos del requerimiento nutricional*

Diferencia de Pesos	Kcal Requeridas
16	520
2	55
-18	-405
-18	-405
3	82.5
-2	-55
-17	-382.5
11	357.5
-10	-225
-6	-165

Nota: Elaboración Propia

#### **4.3. PREPROCESAMIENTO DE DATOS**

Para el preprocesamiento de los datos, se define la selección de los datos necesarios basándonos en las limitantes de presente proyecto de investigación, así como la aplicación de técnicas para el manejo de datos faltantes. Cabe resaltar que, la base de datos inicial contiene la información total de 50494 personas, después del proceso de selección de datos. La nueva base de datos contiene la información de 13338 personas.

## Figura 14

*Base de datos resultante según las limitantes del proyecto*

DEPARTAME	PROVINCIA	DISTRITO	TIPO_VIVIEN	TENENCIA_V	VSEXO	EDAD	ESTADO_CIVIL	NIVEL EDUCATIVO	USO_EXPOS	ENFERMEDA	DONDE_TRA	DIAS_ACTIVI	DIAS_CONSU	DIAS_CONSU	PESO_ACTUA	TALLA_ACTUAL	
9	LIMA	LIMA	SAN BORJA	Casa indeper	Propia total	Femenino	60	Casado / Casada	Superior universitaria	Ninguno	Su casa o de	4 dias	7 dias	7 dias	71	162	
0	LIMA	LIMA	SAN BORJA	Habitación o	Propia pagán	Femenino	50	Casado / Casada	Superior universitaria	Ninguno	Fuera de cas	3 dias	7 dias	7 dias	96	180	
1	LIMA	LIMA	SAN BORJA	Habitación o	Propia total	Femenino	80	Casado / Casada	Superior universitaria	Ninguno	Diabetes o azúcar alta en	17 dias	7 dias	7 dias	60	145	
2	LIMA	LIMA	SAN BORJA	Habitación o	Propia total	Femenino	45	Casado / Casada	Superior universitaria	Ninguno		0 dias	7 dias	7 dias	60	165	
3	LIMA	LIMA	SAN BORJA	Habitación o	Propia total	Femenino	29	Casado / Casada	Superior no universitaria	Ninguno		0 dias	5 dias	5 dias	70	168	
4	LIMA	LIMA	SAN BORJA	Habitación o	Propia total	Masculino	28	Casado / Casada	Superior universitaria	Ninguno		0 dias	5 dias	5 dias	70	168	
5	LIMA	LIMA	SAN BORJA	Casa indeper	Propia total	Femenino	70	Viudo / Viuda	Postgrado	Ninguno	Diabetes o a;	Fuera de cas:	1 día	3 dias	1 día	66	158
5	LIMA	LIMA	SAN BORJA	Casa indeper	Propia total	Femenino	89	Viudo / Viuda	Superior universitaria	Ninguno	Fuera de cas:	1 día	3 dias	1 día	66	158	
7	LIMA	LIMA	SAN BORJA	Casa indeper	Propia total	Masculino	22	Soltero / Soltera	Superior universitaria	Ninguno		0 dias	2 dias	2 dias	47	149	
8	LIMA	LIMA	SAN BORJA	Casa indeper	Propia total	Masculino	60	Casado / Casada	Superior universitaria	Ninguno		0 dias	7 dias	7 dias	82	166	
9	LIMA	LIMA	SAN BORJA	Casa indeper	Propia total	Masculino	45	Soltero / Soltera	Superior universitaria	Ninguno	Su casa o de	0 dias	7 dias	7 dias	49	156	
2	LIMA	LIMA	SAN BORJA	Departament	Propia pagán	Masculino	59	Casado / Casada	Superior universitaria	Ninguno	Fuera de cas:	0 dias	4 dias	3 dias	75	168	
1	LIMA	LIMA	SAN BORJA	Departament	Propia total	Femenino	69	Soltero / Soltera	Superior no universitaria	Ninguno	Colesterol o triglicéridos a	0 dias	7 dias	4 dias	62	158	

Nota: Elaboración Propia

En la figura 14, se puede apreciar que casi la mitad de los datos necesarios son datos de tipo texto. Esto resulta en un inconveniente debido al hecho que para realizar el modelado de la red neuronal se está aplicando modelos de regresión lineal. Por ende, es necesario el uso de la técnica de codificación de datos donde se asignará un valor numérico a cada palabra con la finalidad de que cada dato en forma de texto tenga una representación numérica.

## Figura 15

*Codificación de los datos dentro de la variable "Genero"*

Femenino	1
Masculino	2

Nota: Elaboración Propia

### Figura 16

*Codificación de los datos dentro de la variable "Estado\_Civil"*

Casado / Casada	1
Conviviente	2
Divorciado / Divorciada	3
Separado / Separada	4
Soltero / Soltera	5
Viudo / Viuda	6
(en blanco)	0

Nota: Elaboración Propia

### Figura 17

*Codificación de los datos dentro de la variable "Grado\_Estudios"*

Inicial	1
No estudió	2
Postgrado	3
Primaria	4
Secundaria	5
Superior no universitaria	6
Superior universitaria	7
(en blanco)	0

Nota: Elaboración Propia

### Figura 18

*Codificación de los datos dentro de la variable "Exp\_tabaco"*

Cigarrillo electrónico	1
Cigarrillo electrónico, Otro tabaco o nicotina	1
Cigarro, puro, pipa	1
Cigarro, puro, pipa, Cigarrillo electrónico	1
Cigarro, puro, pipa, Expuesto secundario	1
Expuesto secundario	1
Ninguno	2
Otro tabaco o nicotina	1
(en blanco)	0

Nota: Elaboración Propia

## Figura 19

### *Codificación de los datos dentro de la variable "Enfermedad"*

Alguna otra enfermedad crónica	1
Cáncer o tumor maligno	2
Cáncer o tumor maligno, Alguna otra enfermedad crónica	3
Colesterol o triglicéridos altos	4
Colesterol o triglicéridos altos, Alguna otra enfermedad crónica	5
Colesterol o triglicéridos altos, Cáncer o tumor maligno	6
Colesterol o triglicéridos altos, Cáncer o tumor maligno, Alguna otra enfermedad crónica	7
Colesterol o triglicéridos altos, Presión alta o hipertensión	8
Colesterol o triglicéridos altos, Presión alta o hipertensión, Alguna otra enfermedad crónica	9
Colesterol o triglicéridos altos, Presión alta o hipertensión, Cáncer o tumor maligno	10
Colesterol o triglicéridos altos, Presión alta o hipertensión, Cáncer o tumor maligno, Alguna otra enfermedad crónica	11
Diabetes o azúcar alta en la sangre	12
Diabetes o azúcar alta en la sangre, Alguna otra enfermedad crónica	13
Diabetes o azúcar alta en la sangre, Cáncer o tumor maligno	14
Diabetes o azúcar alta en la sangre, Colesterol o triglicéridos altos	15
Diabetes o azúcar alta en la sangre, Colesterol o triglicéridos altos, Alguna otra enfermedad crónica	16
Diabetes o azúcar alta en la sangre, Colesterol o triglicéridos altos, Cáncer o tumor maligno	17
Diabetes o azúcar alta en la sangre, Colesterol o triglicéridos altos, Presión alta o hipertensión	18
Diabetes o azúcar alta en la sangre, Colesterol o triglicéridos altos, Presión alta o hipertensión, Alguna otra enfermedad crónica	19
Diabetes o azúcar alta en la sangre, Colesterol o triglicéridos altos, Presión alta o hipertensión, Cáncer o tumor maligno	20
Diabetes o azúcar alta en la sangre, Colesterol o triglicéridos altos, Presión alta o hipertensión, Cáncer o tumor maligno, Alguna otra enfermedad crónica	21
Diabetes o azúcar alta en la sangre, Presión alta o hipertensión	22
Diabetes o azúcar alta en la sangre, Presión alta o hipertensión, Alguna otra enfermedad crónica	23
Diabetes o azúcar alta en la sangre, Presión alta o hipertensión, Cáncer o tumor maligno	24
Diabetes o azúcar alta en la sangre, Presión alta o hipertensión, Cáncer o tumor maligno, Alguna otra enfermedad crónica	25
Ninguno	26
Presión alta o hipertensión	27
Presión alta o hipertensión, Alguna otra enfermedad crónica	28
Presión alta o hipertensión, Cáncer o tumor maligno	29
Presión alta o hipertensión, Cáncer o tumor maligno, Alguna otra enfermedad crónica	30
(en blanco)	0

Nota: Elaboración Propia

## Figura 20

### *Codificación de los datos dentro de la variable "Act\_fis"*

0 días	0
1 día	1
2 días	2
3 días	3
4 días	4
5 días	5
6 días	6
7 días	7

Nota: Elaboración Propia

### Figura 21

*Codificación de los datos dentro de la variable "Consm\_fruta"*

0 días	0
1 día	1
2 días	2
3 días	3
4 días	4
5 días	5
6 días	6
7 días	7

Nota: Elaboración Propia

### Figura 22

*Codificación de los datos dentro de la variable "Consm\_verdura"*

0 días	0
1 día	1
2 días	2
3 días	3
4 días	4
5 días	5
6 días	6
7 días	7

Nota: Elaboración Propia

Una vez concluida la codificación de datos, la base de datos quedaría actualizada de la siguiente manera.

**Figura 23**

*Muestra de la base de datos después del proceso de codificación*

Genero	Estado_Civil	Grado_Estudi	Exp_tabaco	Enfermedad	Act_fis	consm_fruta	consm_verdu	Peso	Talla	Edad
1	6	7	2	30	7	7	7	44	160	72
1	4	6	2	30	0	7	7	48	150	54
2	1	7	2	30	0	2	3	66	148	70
1	1	7	2	5	0	2	3	66	148	64
1	5	7	2	26	7	4	4	47	150	71
1	1	5	2	30	6	4	2	82	180	56
2	1	3	2	27	4	7	7	77	160	66
1	3	3	2	5	3	7	7	59	170	67
1	1	7	2	26	5	2	3	80	170	59
1	1	7	2	30	3	7	2	80	174	54

Nota: Elaboración Propia

#### 4.4. DATASET

La base de datos total contiene la información de 13338 personas entre los 45 a 75 años, esta información contiene el género, estado civil, grado de estudios, exposición al tabaco, enfermedades, actividad física, consumo de frutas, consumo de verduras, peso, talla y edad. Adicionalmente se obtuvo el requerimiento energético en kilocalorías (salida) de los datos con la ayuda del método de broca, cálculos de índice de masa corporal, la ayuda de tablas de gasto energético y requerimientos nutricionales y la consultoría de un personal nutricionista para el uso y desarrollo del proceso.

**Figura 24**

*Muestra de la base de datos*

Genero	Estado_Civil	Grado_Estudi	Exp_tabaco	Enfermedad	Act_fis	consm_fruta	consm_verdu	Peso	Talla	Edad
1	6	7	2	30	7	7	7	44	160	72
1	4	6	2	30	0	7	7	48	150	54
2	1	7	2	30	0	2	3	66	148	70
1	1	7	2	5	0	2	3	66	148	64
1	5	7	2	26	7	4	4	47	150	71
1	1	5	2	30	6	4	2	82	180	56
2	1	3	2	27	4	7	7	77	160	66
1	3	3	2	5	3	7	7	59	170	67
1	1	7	2	26	5	2	3	80	170	59
1	1	7	2	30	3	7	2	80	174	54
1	5	3	2	30	1	3	0	55	165	61
2	5	7	2	30	0	3	7	73	180	55

Nota: Elaboración Propia

## Figura 25

*Muestra de la base de datos de requerimiento energético obtenida*

Kcal requeridas_3
520
55
-405
-405
83
-55
-383
358
-225
-165
325
193
28
423
-138

Nota: Elaboración Propia

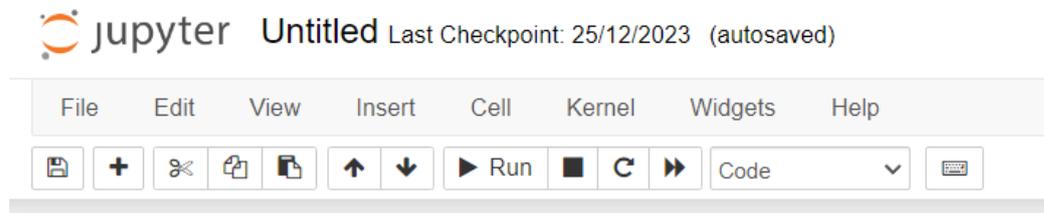
Teniendo terminada la base de datos se procede a guardar el archivo en formato “CSV utf-8 (delimitado por comas)”, de esta manera poder ser ejecutado sin ningún problema al momento de ser llamado por el programa donde se realizará la red neuronal.

### 4.5. IMPORTACION DE LIBRERIAS

Empezamos abriendo “ANACONDA NAVIGATOR”, dentro del programa ejecutamos la aplicación “Jupyter Notebook”. Esta nos enviara a un entorno web donde crearemos un documento nuevo en el cual iniciaremos con la digitación del código en lenguaje Python.

## Figura 26

### *Entorno de Jupyter Notebook*



Nota: Elaboración Propia

Instalamos Tensorflow versión 2.15.0, la cual nos proporciona herramientas para crear y entrenar modelos enfocados en inteligencia artificial y en la optimización del hardware tales como CPU, GPU y TPU.

## Figura 27

### *Versión de tensorflow*

```
print(tf.__version__)  
2.15.0
```

Nota: Elaboración Propia

### **4.3.1. Importación de Librerías Básicas**

Para el diseño de modelos de inteligencia artificial es necesario la importación de librerías tales como “pandas” para la manipulación de datos, “numpy” para la realización de operaciones numéricas, “matplotlib.pyplot” para la creación de gráficos, de “sklearn.model\_selection” importamos “train\_test\_split” para la separación de los datos en entrenamiento y prueba y por

último la importación a través de “sklearn.metrics” las métricas “r2\_score” y “mean\_absolute\_error” usadas para la validación de las predicciones.

### Figura 28

#### *Librerías básicas importadas*

```
# Manipulación de datos
import pandas as pd
# Operaciones numéricas
import numpy as np
# Creación de gráficos
import matplotlib.pyplot as plt
# Para separar datos de entrenamiento y prueba
from sklearn.model_selection import train_test_split
# métricas de validación
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
```

Nota: Elaboración Propia

#### 4.3.2. Importación de Librerías de Modelos de Inteligencia Artificial

Para la presente investigación se optó por probar distintos modelos de machine learning y de regresión por redes neuronales con la finalidad de evidenciar los mejores resultados. De machine learning se usaron los modelos de “Árbol de regresión”, “Regresión lineal”, “Random forest” y “XG\_bosst. De los modelos por redes neuronales, a través de “tensorflow.keras.models” se hizo uso del modelo “Sequential” para la creación de modelos con capas secuenciales de 2, 3, 4 y 5 capas. Asimismo, se hizo importación por medio “tensorflow.keras.layers” las librerías de “Dense” y “Normalización” con la finalidad de generar capas densas y normalizar los datos, de “tensorflow.keras.optimizers” la importación la librería “Adam” para la optimización del modelo de red neuronal y por último la importación “tensorflow” para asegurar la correcta reproducción de los resultados.

## Figura 29

### *Librerías importadas de modelos de inteligencia artificial*

```
# Modelo Red neuronal secuencial
from tensorflow.keras.models import Sequential
# Capa completamente conectada
from tensorflow.keras.layers import Dense, Normalization
# Optimizador
from tensorflow.keras.optimizers import Adam
# Modelos de Machine Learning
from sklearn.tree import DecisionTreeRegressor as dtr
from sklearn.linear_model import LinearRegression
from xgboost import XGBRegressor as xgbr
from sklearn.ensemble import RandomForestRegressor as rfr
# asegurar que los resultados sean "reproducibles"
import tensorflow as tf
```

Nota: Elaboración Propia

## 4.6. EXPLORACION DE DATOS

Llamamos al archivo Excel titulado “Libro5\_base\_de\_datos” con la extensión “csv” donde se encuentra la base de datos que se usara para el presente proyecto de investigación. Asimismo, con el comando “.info” podremos leer y consultar la información contenida dentro del archivo, donde se visualiza el tipo de dato con los que fueron catalogados. En este caso, son columnas de tipo int de 64 bits y otras de tipo float de 64 bits.

## Figura 30

*Código usado para la visualización de los datos*

```
# Leer el set de datos y consultar información básica
req_df = pd.read_csv('Libro5_base_de_datos.csv')
req_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13338 entries, 0 to 13337
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Genero                 13338 non-null  int64
1   Estado_Civil          13338 non-null  int64
2   Grado_Estudios        13338 non-null  int64
3   Exp_tabaco            13338 non-null  int64
4   Enfermedad            13338 non-null  int64
5   Act_fis               13338 non-null  int64
6   consm_fruta          13338 non-null  int64
7   consm_verdura         13338 non-null  int64
8   Peso                 13338 non-null  float64
9   Talla                13338 non-null  int64
10  Edad                 13338 non-null  int64
11  Kcal_requeridas_1    13338 non-null  float64
12  Kcal_requeridas_2    13338 non-null  float64
13  Kcal_requeridas_3    13338 non-null  float64
dtypes: float64(4), int64(10)
memory usage: 1.4 MB
```

Nota: Elaboración Propia

### 4.4.1. Visualización de Datos

Mediante el empleo de “Data frame” y por medio del uso de listas seleccionamos las columnas dentro de la base de datos las cuales serán destinadas a formar parte de los datos de entrada “X” y otro para los datos de salida “y”.

## Figura 31

*Código usado para la selección de los datos de entrada*

```
lista_caract = [
    'Genero',
    'Estado_Civil',
    'Grado_Estudios',
    'Exp_tabaco',
    'Enfermedad',
    'Act_fis',
    'consm_fruta',
    'consm_verdura',
    'Peso',
    'Talla',
    'Edad'
]

X = req_df[lista_caract]
X.head()
```

Nota: Elaboración Propia

### Figura 32

*Código usado para la selección de los datos de salida*

```
lista_etiq = ['Kcal_requeridas_3']  
y = req_df[lista_etiq]  
y.head()
```

Nota: Elaboración Propia

### Figura 33

*Muestra de los Datos de entrada*

	Genero	Estado_Civil	Grado_Estudios	Exp_tabaco	Enfermedad	Act_fis	consm_fruta	consm_verdura	Peso	Talla	Edad
0	1	6	7	2	30	7	7	7	44.0	160	72
1	1	4	6	2	30	0	7	7	48.0	150	54
2	2	1	7	2	30	0	2	3	66.0	148	70
3	1	1	7	2	5	0	2	3	66.0	148	64
4	1	5	7	2	26	7	4	4	47.0	150	71

Nota: Elaboración Propia

### Figura 34

*Muestra de los Datos de salida*

	Kcal_requeridas_3
0	520.0
1	55.0
2	-405.0
3	-405.0
4	82.5

Nota: Elaboración Propia

## 4.7. SEPARACIÓN DE DATOS

Para la separación de datos en prueba y entrenamiento usamos la función “train\_test\_split” con la finalidad de dividir la base de datos en 2 partes llamadas “train” y “test”, este código nos ayuda brindar una proporción adecuada a la base de datos donde

un 75% será destinada a la fase de entrenamiento y un 25% destinadas en prueba. Cabe resaltar que, esta proporción se aplicara a ambos data frame creados.

### Figura 35

*Código usado para la separación de train y test*

```
# Separar en datos de entrenamiento y datos de prueba  
X_train, X_test, y_train, y_test = train_test_split(X, y)
```

Nota: Elaboración Propia

## 4.8. DISEÑO DE ALGORITMOS CON MACHINE LEARNING

### 4.8.1. Experimento 1:

Se realizo el diseño de algoritmos mediante inteligencia artificial a partir de 4 modelos:

- Forest regresor
- Árbol de regresiones
- Regresión simple
- XGboost

Donde obtuvo los siguientes resultados analizados según las métricas de “R<sup>2</sup>” y “main absolute error”:

**Tabla 5**

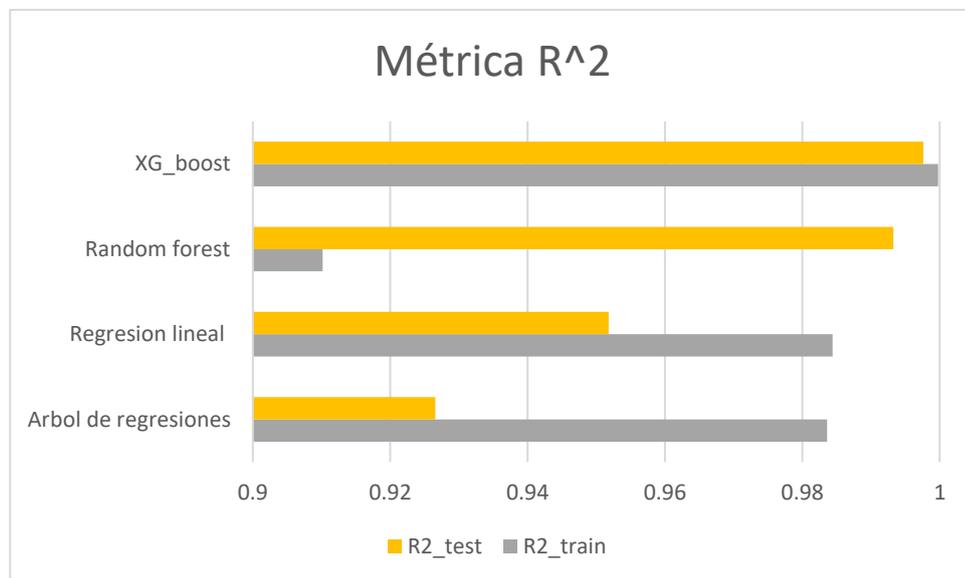
*Resultados de las métricas R<sup>2</sup> y MAE con machine learning*

Machine Learning	Arbol de regresiones	Regresión lineal	Random forest	XGboost
R2_train	0.983596285	0.984406349	0.910164428	0.999772509
R2_test	0.926536732	0.951782541	0.993257291	0.997612678
mae_train	51.78907807	40.3847451	2.68677097	5.427248728
mae_test	52.92834227	39.96507091	3.199967672	6.088505042

Nota: Elaboración Propia

**Figura 36**

*Gráfica de los resultados de las métricas R<sup>2</sup> y MAE con machine learning*



Nota: Elaboración Propia

En la figura 36 se puede apreciar los resultados correspondientes al valor del entrenamiento de los 4 modelos de machine learning usados con la métrica de R<sup>2</sup>. Donde el modelo “XG\_boost” tiene los resultados más cercanos a 1 tanto en los resultados de “test” como “train”. Según la métrica R<sup>2</sup> cuando el valor resultante tiende a 1, el resultado de la predicción es muy cercana al valor real. Sin embargo, cuando el valor tiende a 0 se puede inducir que el resultado de la predicción es incorrecto. Por ende, el modelo es deficiente.

## 4.9. DISEÑO DE ALGORITMOS CON REDES NEURONALES

### 4.9.1. Normalización De Datos

Para el diseño de algoritmos mediante inteligencia artificial con redes neuronales. Es necesario el uso de un normalizador de datos. Por ello, definimos la función “crear\_normalizador(datos)” que será utilizada para crear y adaptar un normalizador al argumento denominado “datos”.

Una vez definido, usamos “normalizer = Normalization(axis=1)” que calculará la media y la desviación estándar para luego normalizar cada columna por separado donde se restará el valor de la media y luego será dividida por la desviación estándar.

- “Normalization”: Crea una capa de normalización para normalizar los datos de entrada.
- “axis=1”: indica que la normalización se realizará dentro de los parámetros del eje 1 de los datos de entrada.

También, hacemos uso de “normalizer.adapt(np.array(datos))” que prácticamente adapta los datos suministrados por “np.array(datos)”, la cual transforma los datos en una matriz “numpy” para luego ser utilizados por “tensorflow”. Por último, la función “return normalizer” devuelve el normalizador adaptado.

### Figura 37

*Código usado para normalizar los datos de entrada*

```
def crear_normalizador(datos):  
    normalizer = Normalization(axis=1)  
    normalizer.adapt(np.array(datos))  
    return normalizer
```

Nota: Elaboración Propia

Una vez terminado el código del normalizador, adaptamos el normalizador a los de entramiento “X\_train”

### Figura 38

*Código para aplicar el normalizador a los datos de entrenamiento*

```
normalizador = crear_normalizador(X_train)
```

Nota: Elaboración Propia

#### 4.8.2. Experimento 2: Algoritmos mediante Redes Neuronales

En esta etapa se realizó el diseño de algoritmos mediante inteligencia artificial a partir de redes neuronales de 2, 3, 4 y 5 capas. Cabe resaltar que, se efectuó el contraste de los resultados obtenidos mediante 12 pruebas de cada modelo. Con lo cual, se obtuvo el promedio de los resultados de las métricas de cada prueba.

**Tabla 6**

*Resultados de las métricas R<sup>2</sup> y MAE con redes neuronales (parte 1)*

regresión red neuronal	2 capas (32,64)	3 capas (32,64,64)	4 capas (32,64,64,128)
R2_train	0.975561982142949	0.997305942260171	0.995849175282370
R2_test	0.998253109611993	0.999422990926274	0.999428367968064
mae_train	3.281222399943570	3.651584214238410	4.327727733532460
mae_test	2.479272952921750	3.414197259508760	3.986043884678840

Nota: Elaboración Propia

**Tabla 7**

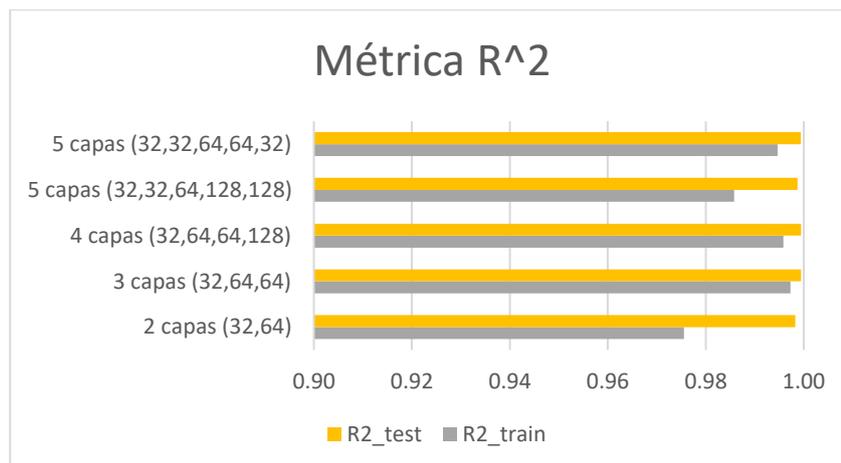
*Resultados de las métricas R<sup>2</sup> y MAE con redes neuronales (parte 2)*

regresión red neuronal	5 capas (32,32,64,128,128)	5 capas (32,32,64,64,32)
R2_train	0.985782949009536	0.9946716
R2_test	0.998745731644433	0.9994153
mae_train	6.726635018144000	4.4043740
mae_test	6.144112469661280	4.0170475

Nota: Elaboración Propia

**Figura 39**

*Gráfica de los resultados de las métricas R<sup>2</sup> con redes neuronales*



Nota: Elaboración Propia

En la figura 39 se puede apreciar los resultados del valor promedio de los 12 entrenamientos realizados a cada red neuronal. Donde, la barra amarilla

corresponde a los datos de “test”, mientras la barra gris corresponde a los datos de “train”. Asimismo, se puede observar que el molde diseñado con 3 capas neuronales presenta la mayor aproximación a 1.

### 4.8.3. Épocas Usadas

Para los distintos entrenamientos de utilizo diferentes épocas aplicando el factor de prueba y error y evitando que la red neuronal entre en un estado de “overfitting” o “underfitting”. Asimismo, se usó el 20% de los datos de entrenamiento para realizar una validación durante el entrenamiento del modelo con la finalidad de supervisar el rendimiento del modelo.

**Tabla 8**

*Épocas y validación split usadas con redes neuronales*

	2 capas (32,64)	3 capas (32,64,64)	4 capas (32,64,64,128)	5 capas (32,32,64,128,128)	5 capas (32,32,64,64,32)
Épocas Usadas	100	50	50	40	50
validación Split	0.2	0.2	0.2	0.2	0.2

Nota: Elaboración Propia

## 4.10. ASEGURAR LOS MODELOS ENTRENADOS

Una vez determinado que la red neuronal es eficiente y capaz de realizar predicciones con valores próximos a los reales con un porcentaje de error mínimo. Es necesario guardar el modelo entrenado para poder usarlo sin la necesidad de volver a entrenarlo.

Para esto usados la función “save(' )” donde dentro del apostrofe designaremos el nombre con el que se quiere desea guardar el modelo. Esto creara un archivo en la

dirección donde se guardó el cuaderno usado para el entrenamiento del modelo de red neuronal.

#### **Figura 40**

*Código usado para guardar el modelo de red neuronal*

```
nn.save('regresion_6_def')
```

Nota: Elaboración Propia

### **4.11. DISEÑO DE LA INTERFAZ**

Empezamos por importar las librerías necesarias para poder crear una aplicación web mediante “Flask” e incorporar el modelo de red neuronal creado anteriormente a través de “Keras”. De la librería de “flask” es necesario importar “Flask” con la que se procederá a crear la aplicación web, “render\_template” para poder presentar plantillas en el texto desarrollado a través de HTML, “request” librería usada para ejecutar peticiones web de manera que se pueda extraer o enviar información, “redirect” para poder redirigir al usuario a un nuevo URL específico y “url\_for” para regresar a una ruta anterior. Asimismo, para poder cargar el modelo de red neuronal es necesario importar “load\_models” el cual es una función dentro del módulo “models” que es parte de la librería de “Keras”. Por último, para poder realizar los cálculos numéricos en Python es necesario la librería de “numpy”.

#### **Figura 41**

*Código usado para importar librerías*

```
from flask import Flask, render_template, request, redirect, url_for  
from keras.models import load_model  
import numpy as np
```

Nota: Elaboración Propia

Posteriormente, definimos la función “get\_model” con la finalidad de poder cargar nuestra red neuronal a nuestro aplicativo web. Para asegurar que el modelo se haya cargado completamente y sin ningún problema ejecutamos el comando “print(\* Modelo cargado!”).

#### Figura 42

*Código usado para cargar el modelo de red neuronal*

```
def get_model():  
    global model  
    model = load_model('regresion_6_def')  
    print(* Modelo cargado!")
```

Nota: Elaboración Propia

De la misma manera, definimos la función “predecir()” para poder introducir los datos de entrada necesarios para realizar la predicción, para lo cual es imprescindible el uso de “np.array()” para transformar las entradas en un array bidimensional, con “model.predict()” ejecutamos el modelo de red neuronal para realizar la predicción de los datos introducidos para luego devolver la predicción con el comando “return prediccion”.

#### Figura 43

*Código usado para introducir los datos de entrada a la red neuronal*

```
def predecir(Genero, Estado_civil, Grado_Estudios, Exp_tabaco, Enfermedad, Act_fis, consm_fruta, consm_verd, Peso, Talla, Edad):  
    X = np.array([[Genero, Estado_civil, Grado_Estudios, Exp_tabaco, Enfermedad, Act_fis, consm_fruta, consm_verd, Peso, Talla, Edad]])  
    prediccion = model.predict(X)  
    return prediccion
```

Nota: Elaboración Propia

Una vez terminado el proceso de predicción, ejecutamos “Flask()” para definir una URL que pueda ejecutar archivos estáticos dentro de la carpeta “/static”. También, definimos una lista llamada “datos” que muestra los valores predeterminados de los

valores de entrada que se puedan modificar al momento de introducir los datos de entrada y otra lista “resul” donde se mostrara la predicción del modelo una vez concluida. Por último, se define la lista “nombre” que contendrá las características de cada dato de entrada para ser visualizado como etiquetas dentro de la aplicación web.

#### Figura 44

*Código usado para diseñar la estructura de la aplicación web*

```
app = Flask(__name__, static_url_path='/static')
datos = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
resul = [[0,0,0]]
nombre = ["Genero", 'Estado_civil', 'Grado_Estudios', 'Exp_tabaco', 'Enfermedad', 'Act_fis', 'consm_fruta', 'consm_verd',
```

Nota: Elaboración Propia

Con “@app.route('/')” enlazamos la función “index()” donde la variable global “nombre” es llamada. Es decir, esta primera etapa vinculamos la función “index()” mediante un decorador a la URL principal de la aplicación, en este caso con “/” para que la plantilla HTML pueda acceder a las características de la variable “nombre”. Una vez terminado este proceso, con “return render\_template('index.html', nombre=nombre)” se devuelve una plantilla llamada “index.html” el cual contiene la formulación para los datos de entrada introducidos por el usuario.

#### Figura 45

*Código de decorador para acceder a la aplicación web*

```
@app.route('/')
def index():
    global nombre
    return render_template('index.html', nombre=nombre)
```

Nota: Elaboración Propia

Hacemos uso de un segundo decorador que se activara solamente cuando se ejecute una solicitud HTTP mediante el método “POST”. Donde “data” tendrá almacenados los datos enviados en el formulario y asignará los valores correspondientes de los datos de entrada realizados por el usuario. Donde una vez finalizado el proceso de solicitud, la aplicación web redirigirá al usuario hacia una ruta nueva llamada “resultado”.

## Figura 46

*Código usado para hacer la solicitud*

```
@app.route('/', methods=['POST'])
def submit():
    data = {key: request.form[key] for key in request.form}
    global datos
    datos = data
    return redirect(url_for('resultado'))
```

Nota: Elaboración Propia

La nueva ruta llamada “resultado” procesara los datos almacenados dentro del formulario para poder realizar la predicción mediante la función “predecir()” definida anteriormente, los datos de la predicción se guardaran en la variable “resul”. Pero, al ser un arreglo multidimensional, es necesario transformar los datos a un arreglo unidimensional, para este proceso hacemos uso del método “flatten()”. Posteriormente, creamos un diccionario llamado “tabla\_tres\_columnas” donde se almacenara el resultado de la predicción almacenado dentro de “arr\_resul”, el cual es el resultado de aplicar el método “flatten()” a la función “resul”. Cabe resaltar que, los datos de entrada introducidos por el usuario serán mostradas a través de una tabla llamada “tabla\_variables” donde los datos serán mostrados según el orden de las variables dentro de la variable global “datos”, para poder visualizar la correspondencia entre el nombre asignado para cada dato de entrada y del valor introducido por el usuario, asignamos los nombres dentro de la variable global “nombre”.

## Figura 47

*Código usado para mostrar la predicción*

```
@app.route('/resultado')
def resultado():
    global nombre
    tabla_variables = {
        'encabezados': nombre,
        'valores': [
            {nombre[0]: datos['input1'], nombre[1]: datos['input2'], nombre[2]: datos['input3'], nombre[3]: datos['input4'], nombre[4]: d
        ]
    }
    global resul
    resul = predecir(int(datos['input1']), int(datos['input2']), int(datos['input3']), int(datos['input4']), int(datos['input5']), int(da
    arr_resul = resul.flatten()
    tabla_tres_columnas = {
        'encabezados': ['Columna 1', 'Columna 2', 'Columna 3'],
        'valores': [
            {'Columna 1': arr_resul[0], 'Columna 2': arr_resul[1], 'Columna 3': arr_resul[2]},
        ]
    }

    return render_template('resultado.html', tabla_variables=tabla_variables, tabla_tres_columnas=tabla_tres_columnas)
```

Nota: Elaboración Propia

Finalmente, comprobamos que nuestro archivo “main” se esté ejecutando a través de la variable “name”, es decir, que se esté ejecutando directamente para luego cargar el modelo de red neuronal y posteriormente iniciar el servidor flask.

## Figura 48

*Código para iniciar el servidor Flask*

```
if __name__ == '__main__':
    get_model()
    app.run(debug=True)
```

Nota: Elaboración Propia

### 4.12. DISEÑO DEL FORMULARIO

Comenzamos por declarar el tipo del formulario y la versión del mismo, en este caso es HTML, precisamos el idioma imperante dentro del documento con el fin de optimizar el procesamiento del contenido dentro de nuestro aplicativo web, con el uso de

“<head>” declaramos todos los metadatos y enlaces a recursos externos que nos permita proporcionar una mejor visualización de los datos.

Dentro de “<head>” hacemos uso de “<meta charset="UTF-8">” para asegurar que los caracteres únicos y de distintos idiomas puedan ser visibles de manera óptima cuando se cargue el formulario, con “<meta name="viewport" content="width=device-width, initial-scale=1.0">” establecemos la escala de visualización según el dispositivo y definimos el grado de zoom inicial al momento de ejecutarse el aplicativo web, con “<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">” importamos el archivo “materialize” para poder proporcionar un estilo al formulario HTML, con “<link href="{{ url\_for('static', filename='css/material-icons.css') }}" rel="stylesheet">” importamos los archivos CSS donde están almacenados los estilos para los iconos a usar y con “<style>” definimos una etiqueta donde se definirá los estilos que se usaran dentro del aplicativo web.

## Figura 49

*Código de uso para declarar variables, metadatos y enlaces de recursos externos*

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Predecir</title>
<!-- Materialize CSS -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
<link href="{{ url_for('static', filename='css/material-icons.css') }}" rel="stylesheet">
<style>
```

Nota: Elaboración Propia

Dentro de “<style>” definimos los aspectos de los elementos que se aplicaran dentro del formulario HTML. Como el tamaño de los elementos con clase “caja”, clase

“error” y clase “icon”. Asimismo, establecemos la prioridad de las clases “caja” e “icon” para aplicarse por encima de otros estilos que pudiesen aplicarse.

## Figura 50

*Código para personalizar los elementos del formulario*

```
<style>
.caja {
  width: 600px !important;
}

.error {
  border-color: red !important;
}

.icon {
  width: 24px;
  height: 24px;
  padding: 2px;
}
</style>
```

Nota: Elaboración Propia

Haciendo uso de “<body>” definimos los elementos que serán introducidos dentro del formulario para realizar la predicción. empezamos por definir las filas y haciendo uso de la clase “caja” se define el estilo y en ancho de cada bloque. Posteriormente, se realizará una definición de los atributos del formulario tales como el identificador, la cantidad de columnas, el método HTTP usado para él envío de datos y una función de validación de los datos a enviarse. Seguido de una fila adicional donde se precisa el uso de un bucle de plantilla con el objetivo de generar elementos de entrada donde antes de cada elemento de entrada hacemos uso de la clase “icon” para la aplicación de los estilos de imagen. Asimismo, cada elemento de entrada debe de tener un identificador único, la definición del tipo de campo, la función de validación y un nombre único al momento de enviar el formulario. Cabe resaltar que, es necesario que la validación se ejecute cada vez que se ingrese un valor dentro del formulario para que el usuario pueda visualizar si en

dato ingresado es válido. Finalmente se define una etiqueta asociada con el campo de entrada para usar el valor de “i” con lo que cada campo tenga un nombre único.

### Figura 51

*Código usado para introducir los campos de entrada de datos al formulario*

```
<h1 class="center">Ingrese datos para predecir</h1>
<div class="row caja">
  <form id="custom-form" class="col s12" method="post" onsubmit="return validateForm()">
    <div class="row">
      {% for i in range(1, 12) %}
        <div class="input-field col s6 custom-input-field">
          
          <input id="input{{ i }}" type="text" class="validate" name="input{{ i }}" oninput="validateInput(this)">
          <label for="input{{ i }}">{{ nombre[i-1] }}</label>
        </div>
      {% endfor %}
    </div>
  </form>
</div>
```

Nota: Elaboración Propia

Siguiendo con este razonamiento, es necesario la creación de un botón para poder enviar el formulario una vez que el usuario haya introducido los datos. Para lo cual hacemos uso de “<button class="btn waves-effect waves-light" type="submit" name="action">PREDECIR</button>” para aplicar un efecto cuando se accione el botón, también establecemos el nombre del botón a “Precedir”.

### Figura 52

*Código usado para crear botón para el envío del formulario*

```
    <div class="center">
      <button class="btn waves-effect waves-light" type="submit" name="action">PREDECIR</button>
    </div>
  </form>
</div>
```

Nota: Elaboración Propia

Es necesario priorizar que el aplicativo web tenga una interfaz donde se pueda interactuar con el usuario de modo que la experiencia sea única. Por esta razón la importación del archivo JavaScript de “materialize” es necesario.

### Figura 53

*Código para importar "materialize Java Script"*

```
<!-- Materialize JavaScript -->  
<script src="https://cdn.jsdelivr.net/npm/materialize@1.0.0/js/materialize.min.js"></script>
```

Nota: Elaboración Propia

Siguiendo con el diseño del aplicativo web, es necesario definir las funciones de validación usadas anteriormente para verificar en tiempo real y antes de enviar el formulario, la existencia de un valor no permitido dentro de los campos de entrada. Con valor no permitido se hace referencia a un valor con decimales o si el campo está vacío. En el primer caso el formulario no se enviará hasta corregir el error el cual mediante la excepción “Error” informara cual es problema a solucionar. En el último caso, al verificar que alguno de los campos este vacío el aplicativo web tendrá que enviar una alerta con la expresión “Por favor, llene todos los campos con números reales”.

### Figura 54

*Código de validación de datos*

```
function validateForm() {  
  var inputs = document.querySelectorAll('input[type="text"]');  
  for (var i = 0; i < inputs.length; i++) {  
    if (inputs[i].value === '' || !isRealNumber(inputs[i].value)) {  
      alert('Por favor, llene todos los campos con números reales');  
      return false;  
    }  
  }  
  return true;  
}
```

Nota: Elaboración Propia

## 4.13. DISEÑO DE LA INTERFAZ DE RESPUESTA

Una vez enviado el formulario, el aplicativo web deberá de mostrar la predicción según los datos enviados para lo cual, el diseño de una interfaz para visualizar los datos

es necesario. Al igual que en el diseño del formulario, comenzamos por definir la estructura básica de la página HTML donde haremos uso de metadatos, uso de enlaces CSS, uso de enlaces JavaScript, el ajuste de escala de la pantalla de visualización y el uso de “Jinja” para mostrar los datos generados por la predicción.

### Figura 55

*Código usado para el diseño de la interfaz para mostrar la predicción*

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Resultado</title>
<!-- Materialize CSS -->
<link rel="stylesheet" href="{{ url_for('static', filename='css/materialize.min.css') }}">
```

Nota: Elaboración Propia

### Figura 56

*Código de generación URL de “Materialize JavaScript”*

```
<!-- Materialize JavaScript -->
<script src="{{ url_for('static', filename='js/materialize.min.js') }}"></script>
</body>
</html>
```

Nota: Elaboración Propia

Prosiguiendo con la estructura del código, se procede a la definición de la estructura donde se mostrará los datos de la predicción generada por la red neuronal. En primer lugar, se mostrará el texto “Resultado de la predicción” como título principal seguido del texto “Valores de las variables:” como subtítulo para luego mostrar una tabla donde se puedan visualizar las variables con su respectivo encabezado. En segundo lugar, se mostrará el texto “Resultados: requerimiento Kcal” seguido de otra tabla donde se visualizarán los datos de la predicción con sus respectivos encabezados. Por último,

designar el botón “Nueva Predicción” que al darle click se vuelva a visualizar el formulario para poder ingresar nuevos datos para una nueva predicción.

### Figura 57

*Código usado para la visualización de la predicción*

```
<h2 class="center-align">Resultados: requerimiento kcal</h2>
<table class="striped centered">

  <tbody>
    {% for fila in tabla_tres_columnas['valores'] %}
      <tr>
        {% for encabezado in tabla_tres_columnas['encabezados'] %}
          <td>{{ fila[encabezado] }}</td>
        {% endfor %}
      </tr>
    {% endfor %}
  </tbody>
</table>

<form action="/" method="get">
  <div class="center-align" style="padding: 20px;">
    <button class="btn waves-effect waves-light" type="submit">Nueva Predicción</button>
  </div>
</form>
```

Nota: Elaboración Propia

## 4.14. PREDICCIÓN DE NUEVOS DATOS

Teniendo en cuenta, el código usado para la creación del modelo de red neuronal y el código usado para diseñar el aplicativo web donde se mostrarán el formulario para la entrada de datos por el usuario y la interfaz de respuesta con la visualización de los datos introducidos y los resultados de la predicción. Se pone en operatividad en aplicativo web con el fin de corroborar los resultados esperados.

## Figura 58

*Ingreso de datos de paciente con sobrepeso*

# Ingreso datos para predecir

Genero	Estado_civil
# 1	# 1
Grado_Estudios	Exp_tabaco
# 7	# 2
Enfermedad	Act_fis
# 4	# 1
consm_fruta	consm_verd
# 7	# 4
Peso	Talla
# 74	# 169
Edad	
# 51	

PREDECIR

Nota: Elaboración Propia

## Figura 59

*Predicción Kcal requeridas para paciente con sobrepeso*

## Resultado de la predicción

Valores de las variables:

Genero	Estado_civil	Grado_Estudios	Exp_tabaco	Enfermedad	Act_fis	consm_fruta	consm_verd	Peso	Talla	Edad
1	1	7	2	4	1	7	4	74	169	51

Resultados: requerimiento kcal

-45.81436	-91.51896	-137.23712
-----------	-----------	------------

NUEVA PREDICCIÓN

Nota: Elaboración Propia

## Figura 60

*Ingreso de datos de paciente con peso ideal*

# Ingreso datos para predecir

Genero	Estado_civil
2	1
Grado_Estudios	Exp_tabaco
7	2
Enfermedad	Act_fis
0	4
consm_fruta	consm_verd
5	5
Peso	Talla
72	172
Edad	
45	

PREDECIR

Nota: Elaboración Propia

## Figura 61

*Predicción Kcal requeridas para paciente con peso ideal*

# Resultado de la predicción

Valores de las variables:

Genero	Estado_civil	Grado_Estudios	Exp_tabaco	Enfermedad	Act_fis	consm_fruta	consm_verd	Peso	Talla	Edad
2	1	7	2	0	4	5	5	72	172	45

Resultados: requerimiento kcal

0.25464920202891034

0.5092984040578207

0.763947606086731

NUEVA PREDICCIÓN

Nota: Elaboración Propia

## Figura 62

*Ingreso de datos de paciente con malnutrición*

# Ingrese datos para predecir

Genero	Estado_civil
# 2	# 5
Grado_Estudios	Exp_tabaco
# 6	# 2
Enfermedad	Act_fis
# 0	# 0
consm_fruta	consm_verd
# 7	# 7
Peso	Talla
# 49	# 164
Edad	
# 49	

PREDECIR

Nota: Elaboración Propia

## Figura 63

*Predicción Kcal requeridas para paciente con malnutrición*

# Resultado de la predicción

Valores de las variables:

Genero	Estado_civil	Grado_Estudios	Exp_tabaco	Enfermedad	Act_fis	consm_fruta	consm_verd	Peso	Talla	Edad
2	5	6	2	0	0	7	7	49	164	49

Resultados: requerimiento kcal

163.19951	326.7935	489.90054
-----------	----------	-----------

NUEVA PREDICCIÓN

Nota: Elaboración Propia



## V. CONCLUSIONES

- PRIMERA:** Se logro obtener y preprocesar la base de datos de personas de entre 45 a 75 años del distrito de San Borja – Lima. Cabe destacar que, en este proceso se hizo uso de la codificación de datos, una herramienta dentro de la ingeniería de datos para poder transformar los datos en forma de texto a número. De esta manera, el preprocesamiento de los datos es fue optimo.
- SEGUNDA:** Se obtuvo el diseño y entrenamiento de los modelos de inteligencia artificial, mediante los programas “Anaconda”, “Jupyter Notebook” y “Python” logrando definir la mejor estructura que mejor se adapte a nuestra problemática de estudio. Asimismo, se recurrió a un proceso de prueba y error para la definición de la cantidad de neuronas por capa de los modelos creados mediante redes neuronales para corroborar que la configuración final tenga un porcentaje de error mínimo.
- TERCERA:** Se separo una cantidad de 3335 muestras equivalentes al 20% de datos de la base de datos con el fin de evaluar el comportamiento de los modelos. De esta manera, elegir cuál de los modelos tiene un menor porcentaje de error según las métricas utilizadas. Asimismo, según los resultados obtenidos del modelo basado en redes neuronales de 3 capas el cual obtuvo un resultado de 0.997305942260171 con respecto a la evaluación con la métrica de R2. Por ende, es el modelo con la predicción más cercana al resultado real.
- CUARTA:** Se diseño un aplicativo web donde se pueda cargar el modelo de red neuronal con la finalidad de que el usuario a través del formulario, pueda



introducir los datos de entrada necesarios para realizar la predicción y estos sean mostrados mediante en una interfaz amigable con el usuario.



## VI. RECOMENDACIONES

- PRIMERA:** En investigaciones futuras considerar la posibilidad de incluir la información como la medición de los pliegues cutáneos del paciente u otros factores a los datos de entrada de entrenamiento del modelo de red neuronal, esto podría potenciar la correlación entre variables con la finalidad de aumentar la precisión del modelo.
- SEGUNDA:** Es recomendable establecer convenios con hospitales y/o centros médicos con la finalidad de optimizar la recopilación de datos para posteriores estudios enfocados al área de la salud. Este convenio haría viable no solo la recopilación de datos de entrada sino también la obtención de información sobre la evolución del paciente según los métodos aplicados en cada paciente.
- TERCERA:** Con los resultados de la presente investigación se pudo evidenciar que la aplicación de inteligencia artificial en el área de la salud podría ser viable. Sin embargo, es necesario que los modelos creados a partir de bases de datos liberadas puedan ser entrenados con bases de datos de hospitales con la finalidad de poder afirmar que la aplicación de inteligencia artificial sea viable.
- CUARTA:** La integración de inteligencia artificial en distintos campos de investigación hace posible que esta tecnología pueda revolucionar el entorno al que estamos acostumbrados. Por tal motivo, se recomienda su investigación continua para mejorar su eficiencia y la calidad de respuesta del mismo.



## VII. REFERENCIAS BIBLIOGRÁFICAS

- Aguilar, J., & Campoverde, M. (2020). *Clasificación de frutas basadas en redes neuronales convolucionales*. *Revista Científico-Académica Multidisciplinaria*, vol. 5(01), pág. 3–22. <https://doi.org/10.23857/pc.v5i01.1210>
- Akbar, S., Akram, M. Sharif, M. Tariq, A., & Khan, S. (2018). *Decision support system for detection of hypertensive retinopathy using arteriovenous ratio*. *Artificial intelligence in medicine*, vol. 90, pág. 15–24, Elsevier. <https://dl.acm.org/doi/10.1016/j.artmed.2018.06.004>
- Arif, Z., Abduljabbar, Z., Tahir, H., Bibo, A., & Almufti, S. (2023). Exploring the Power of eXtreme Gradient Boosting Algorithm in Machine Learning: a Review, vol (12). <https://doi.org/10.25007/ajnu.v12n2a1612>
- Asiri, N., Hussain, M., Al, F., & Alzaidi, N. (2019) *Deep learning based computer-aided diagnosis systems for diabetic retinopathy: Artificial intelligence in medicine*, vol. 99. Elsevier. <https://doi.org/10.1016/j.artmed.2019.07.009>
- Bauce, G. (2022). *Índice de Masa corporal, Peso ideal y porcentaje de grasa corporal en personas de diferentes grupos etarios*, *Revista Digital de Postgrado* vol. 11(1). <https://doi.org/10.37910/RDP.2022.11.1.e331>
- Bernaldo, R., & Raymundo, L. (2020). *Propuesta de implementación de la tecnología emergente basada en inteligencia artificial (AI) en diagnóstico médico en Hospital Cayetano Heredia*. [Tesis de bachiller, Universidad Tecnológica del Perú]. Repositorio de la Universidad Tecnológica del Perú. [https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/3793/Ronel%20Bernaldo\\_Lisset%20Raymundo\\_Trabajo%20de%20Investigacion\\_Bachiller\\_2020\\_2.pdf?sequence=1](https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/3793/Ronel%20Bernaldo_Lisset%20Raymundo_Trabajo%20de%20Investigacion_Bachiller_2020_2.pdf?sequence=1)
- Bernui, I., & Delgado-Pérez, D. (2021). *Factores asociados al estado y al riesgo nutricional en adultos mayores de establecimientos de atención primaria*. *Anales de la Facultad de Medicina*, vol. 82. <https://dx.doi.org/10.15381/anales.v82i4.20799>



- Bhatt, A., Vaghashiya, R., Kulkarni, M., & Kamaraj, P. (2022). *Explainable Artificial Intelligence in Retinal Imaging for the detection of Systemic Diseases*.  
<https://doi.org/10.48550/arXiv.2212.07058>
- Bisong, E., & Google colaboratory. (2019). *Building Machine Learning and Deep Learning Models on Google Cloud Platform, A Comprehensive Guide for Beginners*. <https://link.springer.com/book/10.1007/978-1-4842-4470-8>
- Caballero, L. (2017). *Patrones de Consumo Alimentario, Estado nutricional y Características metabólicas en muestras Poblacionales Urbanas del Nivel del Mar y Altura del Perú*. [Tesis doctoral, Universidad Peruana Cayetano Heredia]. Repositorio Universidad Peruana Cayetano Heredia Postgrado.  
[https://repositorio.upch.edu.pe/bitstream/handle/20.500.12866/1012/Patrones\\_CaballeroGutierrez\\_Lidia.pdf?sequence=3&isAllowed=y](https://repositorio.upch.edu.pe/bitstream/handle/20.500.12866/1012/Patrones_CaballeroGutierrez_Lidia.pdf?sequence=3&isAllowed=y)
- Cámara Costarricense de la Industria Alimentaria. (2017). *¿Cómo afecta nuestro estado civil en la alimentación?* <https://estilosdevida.cacia.org/wp-content/uploads/2017/12/Suplemento-27.pdf>
- Campos, I., Galván, Ó., Hernández, L., Barquera, S., & Oviedo, C. (2023). *Prevalencia de obesidad y factores de riesgo asociados en adultos mexicanos: resultados de la Ensanut 2022*, vol. 65, pág. 238-247. <https://doi.org/10.21149/14809>
- Centeno, J. (2022). *Diseño De Un Sistema De Clasificación De Limones Usando Una Red Neuronal Convolutiva En El Mercado Unión Y Dignidad De La Ciudad De Puno – 2022*. [Tesis para optar el título de ingeniero Electrónico, Universidad Nacional del Altiplano]. Repositorio de la Universidad Nacional del Altiplano.  
[http://repositorio.unap.edu.pe/bitstream/handle/20.500.14082/19923/Centeno\\_Palomino\\_Jharol\\_Iv%c3%a1n.pdf?sequence=1&isAllowed=y](http://repositorio.unap.edu.pe/bitstream/handle/20.500.14082/19923/Centeno_Palomino_Jharol_Iv%c3%a1n.pdf?sequence=1&isAllowed=y)
- Chollet, F. (2018). *Deep Learning with Python*. (2nd ed.). Manning Publications  
<https://sourestdeeds.github.io/pdf/Deep%20Learning%20with%20Python.pdf>
- Cizel, N. (2019). *El Consumo Alimentario Responsable: Una Perspectiva de Género*. Repositorio Universidad Pontificia Comillas ICADE.  
<https://repositorio.comillas.edu/rest/bitstreams/271833/retrieve>



- Corella, D., & Ordovás, J. (2015). *Relación entre el estado socioeconómico, la educación y la alimentación saludable*, *Nutrición y Salud*, 283-306  
<https://dialnet.unirioja.es/servlet/articulo?codigo=5207080>
- Cuya, B., & Ramos, M. (2020). *Sistema de control de calidad utilizando redes neuronales para la clasificación del estado de la granadilla*. [Tesis de título de ingeniero de sistemas, Universidad de Lima]. Repositorio de la Universidad de Lima.  
<https://repositorio.ulima.edu.pe/bitstream/handle/20.500.12724/13854/Trabajo.pdf?sequence=1&isAllowed=y>
- Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. Wiley.  
<https://wtf.tw/ref/duckett.pdf>
- Ekman, M. (2021). *Learning Deep Learning: Theory and Practice of Neural Networks, Computer Vision, Natural Language Processing, and Transformers Using TensorFlow (1er edición)*. Addison-Wesley Professional.  
<https://www.oreilly.com/library/view/learning-deep-learning/9780137470198/>
- Ekmeiro, J., Moreno, R., García, M., & Cámara, F. (2015). *Patrón de consumo de alimentos a nivel familiar en zonas urbanas de Anzoátegui, Venezuela*, vol. 32, pag. 1758-1765. <https://dx.doi.org/10.3305/nh.2015.32.4.9404>
- Expósito, M., & Ávila, R. (2008). *Aplicaciones de la inteligencia artificial en la Medicina: perspectivas y problemas*. Acimed  
<http://scielo.sld.cu/pdf/aci/v17n5/aci05508.pdf>
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (2nd ed.). O'Reilly Media. [https://powerunit-ju.com/wp-content/uploads/2021/04/Aurelien-Geron-Hands-On-Machine-Learning-with-Scikit-Learn-Keras-and-Tensorflow\\_-Concepts-Tools-and-Techniques-to-Build-Intelligent-Systems-OReilly-Media-2019.pdf](https://powerunit-ju.com/wp-content/uploads/2021/04/Aurelien-Geron-Hands-On-Machine-Learning-with-Scikit-Learn-Keras-and-Tensorflow_-Concepts-Tools-and-Techniques-to-Build-Intelligent-Systems-OReilly-Media-2019.pdf)
- Gonzales, A., Valcárcel, M., Cháfer, J., Cuesta, M., Runkle, I., Seara, G., Armengol, M., & Mayol, J. (2017). *Diagnóstico Y Tratamiento De Hiponatremia Usando Modelos Computacionales De Guías De Práctica Clínica*.



[https://www.researchgate.net/publication/298353071\\_Diagnostico\\_y\\_tratamiento\\_de\\_hiponatremia\\_usando\\_modelos\\_computacionales\\_de\\_guias\\_de\\_practica\\_clinica](https://www.researchgate.net/publication/298353071_Diagnostico_y_tratamiento_de_hiponatremia_usando_modelos_computacionales_de_guias_de_practica_clinica)

Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python* (2nd ed.). O'Reilly Media. <https://www.amazon.com/Flask-Web-Development-Developing-Applications/dp/1491991739>

Huaroto, C., & Policarpo, W. (2019). *Índice De Masa Corporal (Imc) Y Actividad Física En Estudiantes Del V Ciclo De La Facultad De Ciencias Contables De La Universidad Nacional Del Callao 2019*. [Título de licenciatura, Universidad Nacional del Callao]. Repositorio de la Universidad Nacional del Callao. [https://repositorio.unac.edu.pe/bitstream/handle/20.500.12952/5366/TESIS\\_PREGRADO\\_HUAROTO\\_POLICARPIO\\_FCS\\_FISICA\\_2019.pdf?sequence=4&isAllowed=y](https://repositorio.unac.edu.pe/bitstream/handle/20.500.12952/5366/TESIS_PREGRADO_HUAROTO_POLICARPIO_FCS_FISICA_2019.pdf?sequence=4&isAllowed=y)

Instituto Nacional de Estadística e Informática, Comisión Consultiva Para La Estimación De La Pobreza Y Otros Indicadores Relacionados En El País. (2012). *Requerimiento Calórico para la Población Peruana*. [https://www.inei.gob.pe/media/cifras\\_de\\_pobreza/nota03.pdf](https://www.inei.gob.pe/media/cifras_de_pobreza/nota03.pdf)

Instituto Nacional de Estadística e Informática. (2022). *Perú: Enfermedades No Transmisibles y Transmisibles, 2022*. <https://www.gob.pe/institucion/inei/informes-publicaciones/4233635-peru-enfermedades-no-transmisibles-y-transmisibles-2022>

Johnson, B. (2020). *Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers*. Apress. <https://es.scribd.com/document/731665080/Download-pdf-Visual-Studio-Code-End-To-End-Editing-And-Debugging-Tools-For-Web-Developers-Bruce-Johnson-ebook-full-chapter>

Kapoor, A., Gulli, A., Pal, S., & Chollet, F. (2022). *Deep Learning with TensorFlow and Keras: Build and deploy supervised, unsupervised, deep, and reinforcement learning models* (3rd ed.). Packt Publishing. <https://github.com/PacktPublishing/Deep-Learning-with-TensorFlow-and-Keras-3rd-edition>



- Korbman, R. (2010). *Obesidad en adultos. Aspectos médicos, sociales y psicológicos*. Anales Médicos, vol. 55, pág. 142-146  
<https://www.medigraphic.com/pdfs/abc/bc-2010/bc103f.pdf>
- Lanata, C. (2012). *El problema del sobrepeso y la obesidad en el Perú: la urgencia de una política de Salud Pública para controlarla*. Revista Peruana de Medicina Experimental y Salud Publica.  
[http://www.scielo.org.pe/scielo.php?script=sci\\_arttext&pid=S1726-46342012000300001&lng=es&tlng=es](http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1726-46342012000300001&lng=es&tlng=es).
- López, A., Martínez, A., Aguilera, V., de la Torre, C., Cárdenas, A., Valdés, E., Macías, A., Santoyo, F., & Barragán, M. (2011). *Género, Interacción Social y Consumo de Alimento: "El Efecto Eva"*. Revista mexicana de trastornos alimentarios, vol. 2, pág. 10-23.  
[http://www.scielo.org.mx/scielo.php?script=sci\\_arttext&pid=S2007-15232011000100002&lng=es&tlng=es](http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S2007-15232011000100002&lng=es&tlng=es).
- López, M. (2017). *Factores Asociados a Obesidad en Lima en el Año 2017*. [Tesis doctoral, Universidad Ricardo Palma]. Repositorio de la Universidad Ricardo Palma.  
<https://repositorio.urp.edu.pe/bitstream/handle/20.500.14138/1825/MCarmenLopezVargas.pdf?sequence=1&isAllowed=y>
- Medinaceli, K., & Silva, M. (2021). *Impacto y regulación de la Inteligencia Artificial en el ámbito sanitario*. Revista del Instituto de Ciencias Jurídicas de Puebla, vol. 15, pág. 77-113. <https://www.scielo.org.mx/pdf/rius/v15n48/1870-2147-rius-15-48-77.pdf>
- Menacho, C. (2014). *Modelos de regresión Lineal con redes neuronales*. Anales Científicos, vol. 75, pág. 253-260 <http://dx.doi.org/10.21704/ac.v75i2.961>
- Ministerio de Salud, Centro Nacional de Alimentación y Nutrición. (2021). *Sobrepeso y Obesidad en la Población Peruana*.  
[https://observateperu.ins.gob.pe/images/archivos/sala\\_nutricional/2020/a\\_sobrepeso\\_obesidad\\_poblacion\\_peruana/sobrepeso\\_y\\_obesidad\\_en\\_la\\_poblacion\\_peruana.pdf](https://observateperu.ins.gob.pe/images/archivos/sala_nutricional/2020/a_sobrepeso_obesidad_poblacion_peruana/sobrepeso_y_obesidad_en_la_poblacion_peruana.pdf)



- Moreno, A. (2019). *Inteligencia artificial en salud desde la perspectiva del Instituto de Ingeniería del Conocimiento*. Revista de la Sociedad Española de Informática y Salud, vol. 136, pág. 32-33.  
<https://dialnet.unirioja.es/servlet/articulo?codigo=7086309>
- Municipalidad de San Borja. (2022). *Censo de Salud y Determinantes del Distrito de San Borja [Municipalidad Distrital de San Borja - MSB]*.  
<https://www.datosabiertos.gob.pe/dataset/censo-de-salud-y-determinantes-del-distrito-de-san-borja-municipalidad-distrital-de-san>
- Muñoz, E., Moreno, J., & Torres, D. (2020). *Uso De La Ciencia De Datos Y Herramientas Tecnológicas Para El Mejoramiento En La Prestación Del Cuidado De La Salud*. [Tesis de especialización, Universidad Sergio Arboleda]. Repositorio de la Universidad Sergio Arboleda.  
<https://repository.usergioarboleda.edu.co/bitstream/handle/11232/1718/Inteligencia%20artificial%20en%20salud.pdf?sequence=1&isAllowed=y>
- Naciones Unidas, División de Estadística y Proyecciones Económicas. (2001). *Ingresos y gastos de consumo de los hogares en el marco de SCN y en encuestas a hogares*.  
[https://www.cepal.org/sites/default/files/publication/files/4718/S01010054\\_es.pdf](https://www.cepal.org/sites/default/files/publication/files/4718/S01010054_es.pdf)
- Navarro, C. (2017). *Aproximación a la medicina personalizada mediante el desarrollo de nuevas metodologías en inteligencia artificial*. [Tesis Doctoral, Universidad de Granada] Repositorio de la Universidad de Granada.  
<http://hdl.handle.net/10481/48604>
- Organización Mundial de la Salud. (2021). *El uso cada vez mayor de la IA en el ámbito de la salud plantea oportunidades y desafíos para los gobiernos, los proveedores y las comunidades*. <https://www.who.int/es/news/item/28-06-2021-who-issues-first-global-report-on-ai-in-health-and-six-guiding-principles-for-its-design-and-use>
- Organización Mundial de la Salud. (2024). *Malnutrición*. <https://www.who.int/es/news-room/fact-sheets/detail/malnutrition>



- Organización Mundial de la Salud. (2024). *Obesidad y Sobrepeso*.  
<https://www.who.int/es/news-room/fact-sheets/detail/obesity-and-overweight>
- Ortega, R., Jiménez, Ana., Martínez, Rosa., Lorenzo, Ana., & Lozano, María. (2021). *Problemática nutricional en fumadores y fumadores pasivos*. *Nutrición Hospitalaria*, vol. 38, pag. 31-34. <https://dx.doi.org/10.20960/nh.3794>.
- Pajuel, J., Bartolo, M., Racacha, E., & Bravo, F. (2022). *Situación nutricional del adulto mayor en el Perú, 2005-2013*. *Anales de la Facultad de Medicina*, vol. 83, pág. 42-48. <https://dx.doi.org/10.15381/anales.v83i1.22544>
- Pajuelo, J. (2017). *La obesidad en el Perú*. *Simposio Obesidad*, vol. 78(2), pág. 179-185. <https://doi.org/10.15381/anales.v78i2.13214>
- Papathanail, I., Bruhlmann, J., Vasiloglou, M., Athopoulou, T., Exadaktylos A., Stanga, Z., Munzer, T., & Mougiakakou, S. (2021). *Evaluation of a Novel Artificial Intelligence System to Monitor and Assess Energy and Macronutrient Intake in Hospitalised Older Patients*. National Library of Medicine.  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8706142/>
- Poladi, S. (2018). *Matplotlib 3.0 Cookbook: Over 150 recipes to create highly detailed interactive visualizations using Python (1st Edición)*. Edición Kindle.  
<https://www.amazon.com/Matplotlib-3-0-Cookbook-interactive-visualizations-ebook/dp/B07G878GXM>
- Portal, P., Vasquez, R. (2018). *Ingesta De Carbohidratos Y Su Relación Con La Resistencia Física De Jugadores De Fútbol De Dos Clubes De Fernando Lores – Tamshiyacu*. [Tesis de Licenciatura, Universidad Nacional de la Amazonia Peruana]. Repositorio Universidad Nacional de la Amazonia Peruana.  
[https://repositorio.unapiquitos.edu.pe/bitstream/handle/20.500.12737/5969/Pepe\\_tesis\\_titulo\\_2018.pdf?sequence=1&isAllowed=y](https://repositorio.unapiquitos.edu.pe/bitstream/handle/20.500.12737/5969/Pepe_tesis_titulo_2018.pdf?sequence=1&isAllowed=y)
- Pyle, D. (1999). *Data Preparation for Data Mining*. Morgan Kaufmann Publishers, Inc.  
[https://www.temida.si/~bojan/MPS/materials/Data\\_preparation\\_for\\_data\\_mining.pdf](https://www.temida.si/~bojan/MPS/materials/Data_preparation_for_data_mining.pdf)
- Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2 (2nd ed.)*. Packt



Publishing. <http://radio.eng.niigata-u.ac.jp/wp/wp-content/uploads/2020/06/python-machine-learning-2nd.pdf>

Raschka, S., Yuxi, L., & Mirjalili, V. (2022). *Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python* (D. Dzhulgakov (ed.)). Packt Publishing.

<https://www.packtpub.com/product/machine-learning-with-pytorch-and-scikit-learn/9781801819312>

Reaño, J. (2020). *Aplicación de inteligencia artificial en diagnósticos de enfermedades*. [Tesis de bachiller, Universidad Católica Santo Toribio de Mogrovejo]. Repositorio de Tesis USAT.

[https://tesis.usat.edu.pe/bitstream/20.500.12423/3378/1/TIB\\_Rea%  
c3%bl0Cobe%  
c3%blasJheiner.pdf](https://tesis.usat.edu.pe/bitstream/20.500.12423/3378/1/TIB_Rea%c3%bl0Cobe%c3%blasJheiner.pdf)

Rivera, K. (2022). *Aplicación de la Inteligencia Artificial en la Nutrición Personalizada*. Revista de Investigaciones de la Universidad Nacional del Altiplano, vol. 11, pág. 265-277. <http://doi.org/10.26788/ri.v11i4.3990>

Serrano, A., Soria, E., & Martin, J. (2009-2010). *Redes Neuronales Artificiales* [http://ocw.uv.es/ingenieria-y-arquitectura/1-2/libro\\_ocw\\_libro\\_de\\_redes.pdf](http://ocw.uv.es/ingenieria-y-arquitectura/1-2/libro_ocw_libro_de_redes.pdf)

Troncoso, C. (2017). *Alimentación del adulto mayor según lugar de residencia*. Horizonte Médico, vol. 17, pag. 3.

<http://dx.doi.org/10.24265/horizmed.2017.v17n3.10>

VanderPlas, J. (2016). *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media.

[https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiSv5CM4LmHAXUBuJUCHbjTCnIQFnoECDEQAQ&url=https%3A%2F%2Favys.omu.edu.tr%2Fstorage%2Fapp%2Fpublic%2Fmustafa.kaya%2F135286%2FPython\\_Datascience.pdf&usg=AOvVaw36oePAjFhJ1vwmsqSTIxUQ&cshid=1721620547804637&opi=89978449](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiSv5CM4LmHAXUBuJUCHbjTCnIQFnoECDEQAQ&url=https%3A%2F%2Favys.omu.edu.tr%2Fstorage%2Fapp%2Fpublic%2Fmustafa.kaya%2F135286%2FPython_Datascience.pdf&usg=AOvVaw36oePAjFhJ1vwmsqSTIxUQ&cshid=1721620547804637&opi=89978449)

## ANEXOS

**ANEXO 1.** Resultados de las métricas  $R^2$  y MAE de los entrenamientos por cada de las redes neuronales creadas

### Figura 64

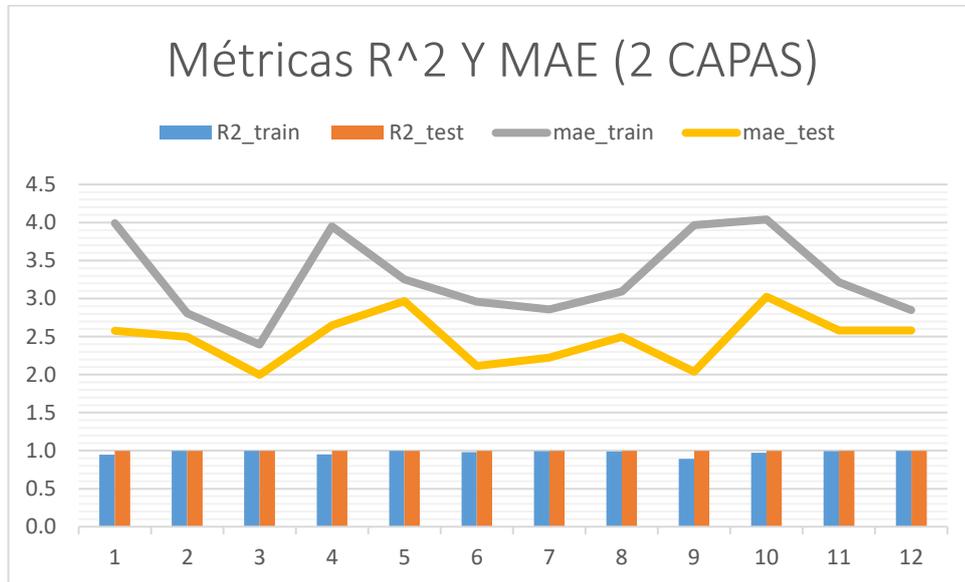
*Resultados de  $R^2$  y MAE de los entrenamientos de la red neuronal (2capas)*

2 capas (32,64)			
R2_train	R2_test	mae_train	mae_test
0.945588444576787	0.996227389806136	3.989723560375820	2.575530416407860
0.997682405759375	0.998056078363938	2.805696672554280	2.497683426283150
0.996637040400951	0.999745232547334	2.395512320243520	1.996484909667960
0.952796306431574	0.998927978928357	3.950419437428840	2.648990160950650
0.997426240383245	0.998346937812651	3.251513107998850	2.965537354363490
0.980462249950415	0.999675994559216	2.959094509549410	2.116543473546830
0.991979972622336	0.999158222719532	2.858971605691310	2.225592349038480
0.991021217206831	0.996881037401547	3.092206650764110	2.498552496228610
0.891548374772163	0.998083941248986	3.966058798006960	2.037143831953650
0.970461481456043	0.998952072919436	4.040498678867520	3.024323097626480
0.993657329807824	0.998759202904795	3.213802244168250	2.582602475353440
0.997482722347846	0.996223226131982	2.851171213673930	2.582291443640360
<b>0.975561982142949</b>	<b>0.998253109611993</b>	<b>3.281222399943570</b>	<b>2.479272952921750</b>

Nota: Elaboración Propia

**Figura 65**

*Grafica de los resultados de R<sup>2</sup> y MAE de los entrenamientos de la red neuronal (2capas)*



Nota: Elaboración Propia

**Figura 66**

*Resultados de R<sup>2</sup> y MAE de los entrenamientos de la red neuronal (3capas)*

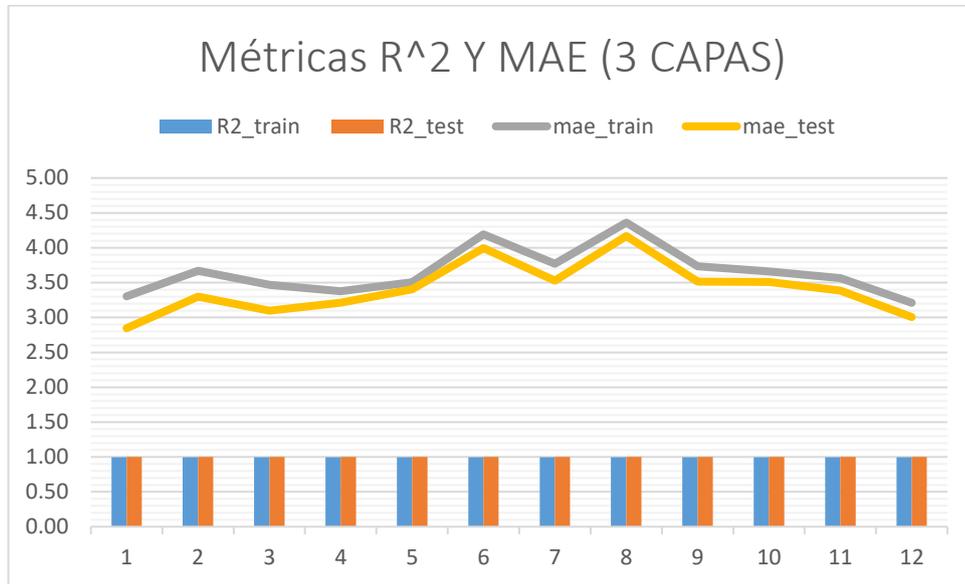
3 capas (32,64,64)			
R2_train	R2_test	mae_train	mae_test
0.996454267821066	0.999581998596950	3.304152893737630	2.846786149697320
0.996995438115336	0.999577524978519	3.670677876178710	3.297319562762210
0.995904484715717	0.999533816228241	3.468751191628110	3.097367072829240
0.997643252685264	0.999439648825052	3.377568441351430	3.213474011617800
0.997431365665644	0.999394874745122	3.505546443482390	3.404513198056850
0.997296755561221	0.999296200290766	4.191834853525030	3.994751394581400
0.997556436365000	0.999410669526104	3.770433394894360	3.530198285108740
0.997695255491333	0.999237582776073	4.360595902198430	4.167995202559220
0.997701517959281	0.999434665836670	3.735650966461030	3.514599889052140
0.997803713324713	0.999383104061649	3.659519558335780	3.510602095477400
0.997711725203294	0.999285189607163	3.563559061891470	3.385693861251470
0.997477094214190	0.999500615642983	3.210719987176530	3.007066391111310
<b>0.997305942260171</b>	<b>0.999422990926274</b>	<b>3.651584214238410</b>	<b>3.414197259508760</b>

Nota: Elaboración Propia

**Figura 67**

*Grafica de los resultados de R<sup>2</sup> y MAE de los entrenamientos de la red neuronal*

*(3capas)*



Nota: Elaboración Propia

**Figura 68**

*Resultados de R<sup>2</sup> y MAE de los entrenamientos de la red neuronal (4capas)*

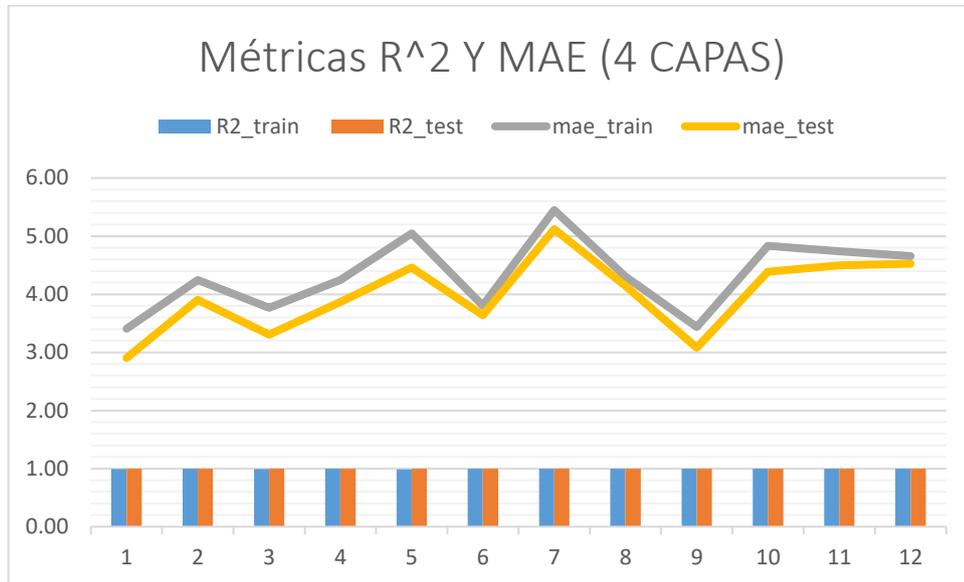
4 capas (32,64,64,128)			
R2_train	R2_test	mae_train	mae_test
0.994199290267690	0.999563858558182	3.407841489742380	2.903786196880250
0.995717730341357	0.999384224472309	4.244753835301600	3.905236805730800
0.994885944525932	0.999502818544342	3.767166930689430	3.305543105394720
0.996949353961072	0.999493544138197	4.242890412929550	3.864770743970210
0.988625580731481	0.999247188422343	5.046824670046950	4.458485865504710
0.997542616023003	0.999410140197023	3.806619121987370	3.639366276678940
0.997015672817602	0.999315264146440	5.448581322630610	5.120092662227250
0.997859963198093	0.999422012823363	4.303869794192790	4.145850141759750
0.997376149709748	0.999623381293099	3.437871565845100	3.081746932516790
0.995388214292421	0.999397224049424	4.832320435397010	4.386330630298490
0.997677346330166	0.999423247965486	4.738155998629970	4.497615405117370
0.996952241189879	0.999357511006555	4.655837224996770	4.523701850066830
<b>0.995849175282370</b>	<b>0.999428367968064</b>	<b>4.327727733532460</b>	<b>3.986043884678840</b>

Nota: Elaboración Propia

**Figura 69**

*Grafica de los resultados de R<sup>2</sup> y MAE de los entrenamientos de la red neuronal*

*(4capas)*



Nota: Elaboración Propia

**Figura 70**

*Resultados de R<sup>2</sup> y MAE de los entrenamientos de la red neuronal (5capas\_1)*

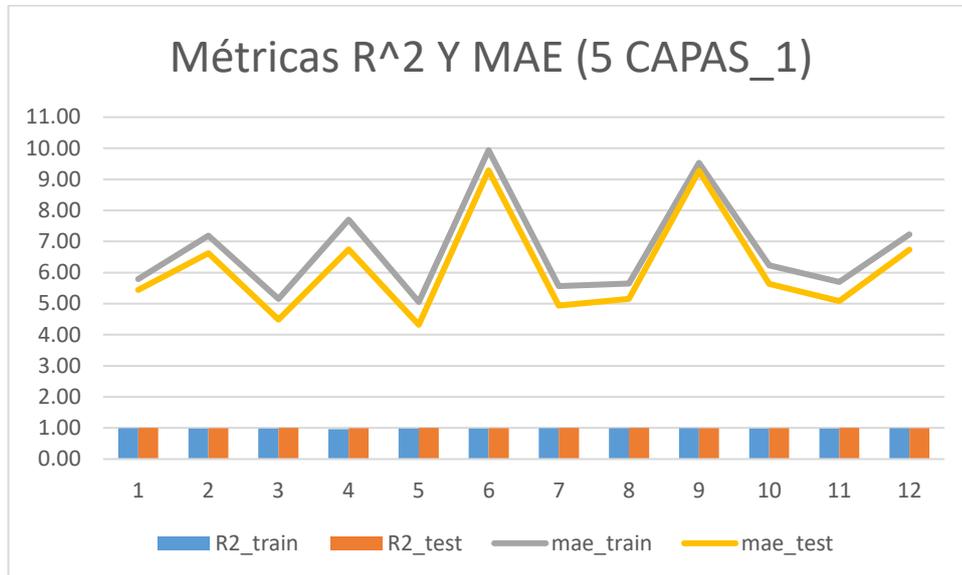
5 capas (32,32,64,128,128)			
R2_train	R2_test	mae_train	mae_test
0.995277173964061	0.998950067931885	5.793429128075610	5.441205133170420
0.979819990028731	0.998636709448668	7.189081461235390	6.624987888193200
0.982565827603766	0.999180287800170	5.152261550421210	4.483853799293460
0.964607863679263	0.998480690608929	7.699904160101290	6.749605984277630
0.980728538974451	0.999221372512414	5.054632087688100	4.320059273629880
0.986154248033291	0.997932586945434	9.937797273728460	9.288212333894380
0.991860607030121	0.999154555342668	5.565431132127580	4.937447108390890
0.989991837453054	0.998978056576119	5.651035603563630	5.157292089502660
0.996175996461471	0.997895372073645	9.528955553028820	9.273262153814980
0.980195130121199	0.998788785915792	6.228148548095430	5.634530681117000
0.987880969421675	0.998993434489945	5.694341900596720	5.084390601877680
0.994137205343343	0.998736860087532	7.224601819065710	6.734502588773230
<b>0.985782949009536</b>	<b>0.998745731644433</b>	<b>6.726635018144000</b>	<b>6.144112469661280</b>

Nota: Elaboración Propia

**Figura 71**

*Grafica de los resultados de R<sup>2</sup> y MAE de los entrenamientos de la red neuronal*

*(5capas\_1)*



Nota: Elaboración Propia

**Figura 72**

*Resultados de R<sup>2</sup> y MAE de los entrenamientos de la red neuronal (5capas\_2)*

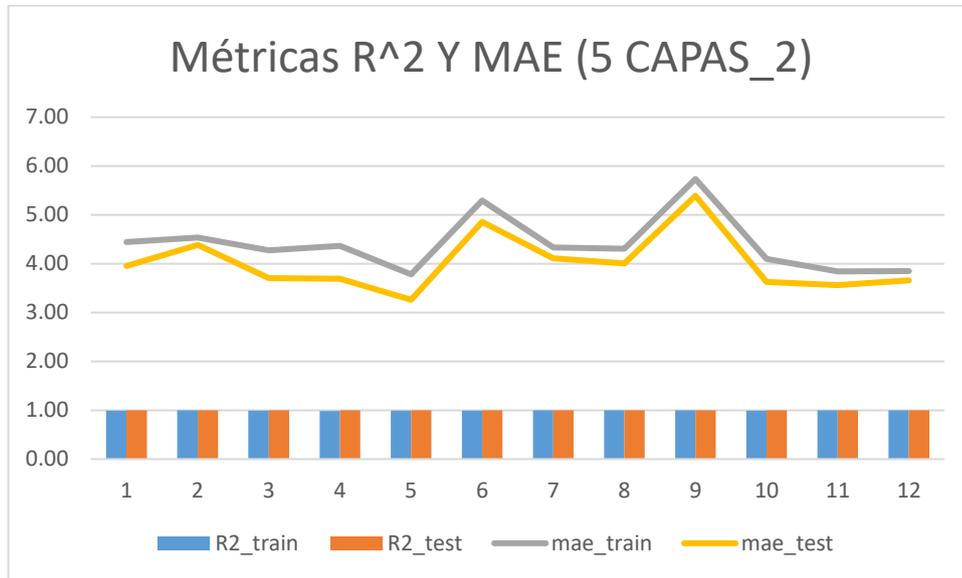
5 capas (32,32,64,64,32)			
R2_train	R2_test	mae_train	mae_test
0.9927757887467880	0.9993931873938290	4.4437540846704600	3.9557662218920800
0.9975889396065360	0.9993984432405200	4.5319228157755600	4.3825289231346600
0.9893070557324580	0.9993923911087080	4.2716946500568300	3.7067707609552800
0.9881652952568990	0.9994823716362340	4.3644835158607500	3.6934288682206600
0.9942358266908660	0.9995587162734710	3.7815116146348600	3.2611954150014900
0.9946180090260860	0.9992650885640400	5.2941605346769800	4.8565433889404000
0.9967226472079810	0.9993556539211410	4.3351238098851200	4.1084573844338800
0.9969320020243180	0.9994273315775030	4.3065153334625300	4.0035550686730400
0.9963430089427600	0.9992431277981600	5.7341231262369200	5.3899569803717600
0.9943635533153780	0.9994446665736800	4.0981332276965700	3.6283832307513900
0.9973230136404620	0.9995226864205650	3.8419700298414700	3.5617043508269700
0.9976840729452960	0.9994993824275660	3.8490950511840100	3.6562791947582500
<b>0.9946716010946520</b>	<b>0.9994152539112850</b>	<b>4.4043739828318400</b>	<b>4.0170474823299900</b>

Nota: Elaboración Propia

### Figura 73

*Grafica de los resultados de R<sup>2</sup> y MAE de los entrenamientos de la red neuronal*

*(5capas\_2)*



Nota: Elaboración Propia

**ANEXO 2.** Códigos en lenguaje Python de los modelos entrenados, base de datos y aplicativo web

[https://drive.google.com/drive/folders/1p0FzoueWzSsDDDjI3e1Bp-mCgBB\\_V\\_2y?usp=drive\\_link](https://drive.google.com/drive/folders/1p0FzoueWzSsDDDjI3e1Bp-mCgBB_V_2y?usp=drive_link)



### ANEXO 3. Declaración jurada de autenticidad de tesis



Universidad Nacional  
del Altiplano Puno



Vicerrectorado  
de Investigación



Repositorio  
Institucional

#### DECLARACIÓN JURADA DE AUTENTICIDAD DE TESIS

Por el presente documento, Yo Luis Angel Curo Roque  
identificado con DNI 71437539 en mi condición de egresado de:

Escuela Profesional,  Programa de Segunda Especialidad,  Programa de Maestría o Doctorado  
Ingeniería Electrónica

informo que he elaborado el/la  Tesis o  Trabajo de Investigación denominada:

“ Diseño de un algoritmo para el cálculo del valor energético en personas de 45 a 75 años en el distrito de

San Borja - Lima mediante inteligencia artificial ”

Es un tema original.

Declaro que el presente trabajo de tesis es elaborado por mi persona y **no existe plagio/copia** de ninguna naturaleza, en especial de otro documento de investigación (tesis, revista, texto, congreso, o similar) presentado por persona natural o jurídica alguna ante instituciones académicas, profesionales, de investigación o similares, en el país o en el extranjero.

Dejo constancia que las citas de otros autores han sido debidamente identificadas en el trabajo de investigación, por lo que no asumiré como tuyas las opiniones vertidas por terceros, ya sea de fuentes encontradas en medios escritos, digitales o Internet.

Asimismo, ratifico que soy plenamente consciente de todo el contenido de la tesis y asumo la responsabilidad de cualquier error u omisión en el documento, así como de las connotaciones éticas y legales involucradas.

En caso de incumplimiento de esta declaración, me someto a las disposiciones legales vigentes y a las sanciones correspondientes de igual forma me someto a las sanciones establecidas en las Directivas y otras normas internas, así como las que me alcancen del Código Civil y Normas Legales conexas por el incumplimiento del presente compromiso

Puno 25 de Julio del 2024

  
\_\_\_\_\_  
FIRMA (obligatoria)



Huella



## ANEXO 4. Autorización para el depósito de tesis en el Repositorio Institucional



Universidad Nacional  
del Altiplano Puno



Vicerrectorado  
de Investigación



Repositorio  
Institucional

### AUTORIZACIÓN PARA EL DEPÓSITO DE TESIS O TRABAJO DE INVESTIGACIÓN EN EL REPOSITORIO INSTITUCIONAL

Por el presente documento, Yo Luis Angel Curo Roque  
identificado con DNI 7147539 en mi condición de egresado de:

Escuela Profesional,  Programa de Segunda Especialidad,  Programa de Maestría o Doctorado

Ingeniería Electrónica

informo que he elaborado el/la  Tesis o  Trabajo de Investigación denominada:

“ Diseño de un algoritmo para el cálculo del valor energético en personas de 45 a 75 años en el distrito de San Borja - Lima mediante inteligencia artificial ”

para la obtención de  Grado,  Título Profesional o  Segunda Especialidad.

Por medio del presente documento, afirmo y garantizo ser el legítimo, único y exclusivo titular de todos los derechos de propiedad intelectual sobre los documentos arriba mencionados, las obras, los contenidos, los productos y/o las creaciones en general (en adelante, los “Contenidos”) que serán incluidos en el repositorio institucional de la Universidad Nacional del Altiplano de Puno.

También, doy seguridad de que los contenidos entregados se encuentran libres de toda contraseña, restricción o medida tecnológica de protección, con la finalidad de permitir que se puedan leer, descargar, reproducir, distribuir, imprimir, buscar y enlazar los textos completos, sin limitación alguna.

Autorizo a la Universidad Nacional del Altiplano de Puno a publicar los Contenidos en el Repositorio Institucional y, en consecuencia, en el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto, sobre la base de lo establecido en la Ley N° 30035, sus normas reglamentarias, modificatorias, sustitutorias y conexas, y de acuerdo con las políticas de acceso abierto que la Universidad aplique en relación con sus Repositorios Institucionales. Autorizo expresamente toda consulta y uso de los Contenidos, por parte de cualquier persona, por el tiempo de duración de los derechos patrimoniales de autor y derechos conexos, a título gratuito y a nivel mundial.

En consecuencia, la Universidad tendrá la posibilidad de divulgar y difundir los Contenidos, de manera total o parcial, sin limitación alguna y sin derecho a pago de contraprestación, remuneración ni regalía alguna a favor mío; en los medios, canales y plataformas que la Universidad y/o el Estado de la República del Perú determinen, a nivel mundial, sin restricción geográfica alguna y de manera indefinida, pudiendo crear y/o extraer los metadatos sobre los Contenidos, e incluir los Contenidos en los índices y buscadores que estimen necesarios para promover su difusión.

Autorizo que los Contenidos sean puestos a disposición del público a través de la siguiente licencia:

Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

En señal de conformidad, suscribo el presente documento.

Puno 24 de Julio del 2024

FIRMA (obligatoria)



Huella