



Desarrollo de un aplicativo móvil que optimice el diagnóstico de humedad en el Laurel mediante Aprendizaje Profundo en el aserradero “San Martín” ubicado en la parroquia de Pastocalle provincia de Cotopaxi

Muñoz Quinatoa, Ricardo Fabricio y Navas Cifuentes, Jonathan Gabriel

Departamento de Eléctrica y Electrónica

Carrera de Ingeniería en Software

Trabajo de titulación, previo a la obtención del título de Ingeniero en Software

M. Sc. Escobar Sánchez, Milton Eduardo

Latacunga – Ecuador

25 de agosto de 2021



DEPARTAMENTO DE ELECTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA DE SOFTWARE

CERTIFICACIÓN

Certifico que el trabajo de titulación, **“Desarrollo de un aplicativo móvil que optimice el diagnóstico de humedad del Laurel mediante Aprendizaje Profundo en el aserradero “San Martín” ubicado en la parroquia de Pastocalle provincia de Cotopaxi”** fué realizado por los señores **Muñoz Quinatoa, Ricardo Fabricio y Navas Cifuentes, Jonathan Gabriel** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, agosto del 2021



Firmado electrónicamente por:

**MILTON
EDUARDO
ESCOBAR
SANCHEZ**

.....

Escobar Sánchez, Milton Eduardo Mg.

C. C.: 1710557545

Informe Urkund



Document Information

Analyzed document	MuñozRicardo_NavasJonathan.pdf (D111676945)
Submitted	8/25/2021 5:14:00 PM
Submitted by	Lorena Ibarra
Submitter email	lorenadibarra@uta.edu.ec
Similarity	4%
Analysis address	lorenadibarra.uta@analysis.urkund.com



Firmado electrónicamente por:

**MILTON
EDUARDO
ESCOBAR
SANCHEZ**

.....

Escobar Sánchez, Milton Eduardo Mg.

C. C.: 1710557545



DEPARTAMENTO DE ELECTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA DE SOFTWARE

RESPONSABILIDAD DE AUTORÍA

Nosotros, **Muñoz Quinatoa, Ricardo Fabricio** con cédula de ciudadanía n° 1724256696 y **Navas Cifuentes, Jonathan Gabriel** con cédula de ciudadanía n° 1725635310, declaramos que el contenido, ideas y criterios del trabajo de titulación: **Desarrollo de un aplicativo móvil que optimice el diagnóstico de humedad del Laurel mediante Aprendizaje Profundo en el aserradero “San Martín” ubicado en la parroquia de Pastocalle provincia de Cotopaxi** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, agosto del 2021

Muñoz Quinatoa, Ricardo Fabricio

C.C.: 1724256696

Navas Cifuentes, Jonathan Gabriel

C.C.: 1725635310



DEPARTAMENTO DE ELECTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA DE SOFTWARE

AUTORIZACIÓN

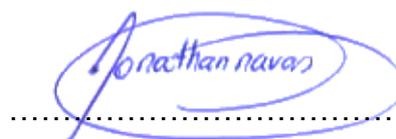
Nosotros, **Muñoz Quinatoa, Ricardo Fabricio** con cédula de ciudadanía n° 1724256696 y **Navas Cifuentes, Jonathan Gabriel** con cédula de ciudadanía n° 1725635310, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Desarrollo de un aplicativo móvil que optimice el diagnóstico de humedad del Laurel mediante Aprendizaje Profundo en el aserradero “San Martín” ubicado en la parroquia de Pastocalle provincia de Cotopaxi** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Latacunga, agosto del 2021



Muñoz Quinatoa, Ricardo Fabricio

C.C.: 1724256696



Navas Cifuentes, Jonathan Gabriel

C.C.: 1725635310

Dedicatoria

Esta tesis está dedicada a:

A Dios, quien me dio la oportunidad de existir brindándome salud, bendición y la guía necesaria para todo el trayecto de mi vida.

A mi madre la Sra. Esthela Quinatoa, quien ha sido un pilar fundamental en el transcurso de mi vida proporcionándome cariño, fuerza y constancia. Motivándome a seguir adelante a pesar de las adversidades.

A mi padre el Sr. Francisco Muñoz, por enseñarme el valor del esfuerzo y la dedicación, dándome un ejemplo de cómo ser un sujeto de bien.

A mi hermana la Srta. Shirley Muñoz, por influir positivamente en las decisiones difíciles que tenido que tomar, además por el apoyo brindado incondicionalmente.

A mis amigos el Sr. Bryan Correa y la Srta. Abigail Sinaluisa, quienes siempre han estado pendientes de mí, y constantemente me animado, motivándome para seguir adelante.

A todos aquellos individuos que dudaron y no creyeron en mí, aquellos que esperaban que fracasara y no llegara a la cúspide de mis estudios. Subestimándome al no creer que fuese capaz de lograrlo. Pensando que me rendiría en las encrucijadas de la vida.

“Me convertiré en un héroe del que puedas estar orgulloso.”

Endeavour

Sr. Ricardo Muñoz

Dedicatoria

Desde el fondo de mi corazón y de mi alma es un honor poder dedicar el presente trabajo de titulación a mi madre la Sra. Tatiana Cifuentes quien ha sido mi motor de motivación, superación, así como mi ejemplo de vida. A mi padre el Sr. César Navas por apoyarme siempre y demostrarme que el infinito es el límite. A mis hermanos los Sres. Jordy Navas y Kevin Navas quienes han estado para mi incondicionalmente.

A mi bis abuelita la Sra. Rosita y mi tía la Sra. Delia infinitas gracias por cada sonrisa, cada picardía las cuales siempre las guardaré en mi mente y en mi corazón desde donde estén muchas gracias de corazón.

A mis abuelitos/as la Sra. Gladys Olivo, el Sr. Jorge Washington Cifuentes, y la Sra. Carmen Venegas por llenarme de toda su sabiduría, amor y enseñarme que los pequeños momentos son aquellos instantes que enriquecen el alma y el corazón.

A la Tem. Tania Cifuentes por llegar a ser más que mi tía es decir mi segunda madre, queriéndome desde que nací. Apoyándome y motivándome en cada circunstancia que he tenido a lo largo de mi vida.

A la Sra. Susana Cifuentes por quererme tanto y también ser como una madre además de estar siempre pendiente de mi bienestar sin importar el día o la hora.

Al Ing. Jorge Cifuentes le agradezco por inculcarme desde pequeño esa ambición por aprender muchas cosas, así como la pasión por el fútbol y el ídolo del Ecuador.

Al Sr. Alfredo Cifuentes le estoy agradecido por enseñarme que todo lo que se hace con el corazón tiene mucho valor en la vida y llega a ser que los sueños se hagan realidad.

A la familia Navas Pallo por acogerme en su hermoso hogar, brindarme su cariño y estar pendientes de que me encuentre bien y no me pase nada. Así como al resto de la familia Navas a la cual siempre estaré totalmente agradecido.

Y finalmente a mis amigos los Sres. Santiago Collaguazo, Ricardo Muñoz, Jhonatan Salazar, Cristian Tapia, Sebastián Chiriboga, Jerson Morocho, Gonzalo Tapuy, Mauro Morales, Disney León y Oscar Toaquiza por brindarme una amistad sincera, así como a mis demás amigos.

Sr. Jonathan Navas

Agradecimiento

Un trabajo de investigación es siempre fruto de ideas, proyectos y esfuerzos previos que corresponden a otras personas.

Deseo expresar mi más sincero agradecimiento al director de esta tesis, Mgs. Milton Escobar, por la dedicación y apoyo que ha brindado durante la duración de este trabajo, por la dirección, el rigor constante y el respecto a mis sugerencias e ideas lo que ha facilitado a las mismas. Gracias por la confianza ofrecida desde que se dio inicio a la construcción de la misma.

De igual manera doy Gracias a mi familia, a mis padres y a mi hermana, por brindarme el cimiento necesario para forjar la persona que soy ahora, siendo ellos la motivación necesaria para no rendirme en momentos de oscuridad.

Así mismo, agradezco a mi compañero de catedra el sr. Jonathan Navas por su apoyo personal y humano, con quien he luchado codo a codo para hacer realidad esta tesis, y además de compartir proyectos e ilusiones durante todos estos años.

Y finalmente gracias a mis amigos, que siempre me han prestado un gran apoyo moral y humano, necesarios en los momentos difíciles de este proyecto.

“Solo con el esfuerzo no es suficiente.”

Elvis De la Cruz Peña

Sr. Ricardo Muñoz

Agradecimiento

A la Universidad de las Fuerzas Armadas ESPE, por permitirme estudiar y forjarme como profesional a lo largo de mi travesía universitaria, obteniendo conocimientos y excelentes experiencias que me permitirán aportar con ese granito de arena para construir un mejor Ecuador.

Una vez finalizado el presente trabajo investigativo agradezco a mi director de tesis y amigo Mg. Milton Escobar por permitirme demostrar que las grandes cosas se las construyen con pasos pequeños. Además, por aceptar ser nuestro guía académico en este proceso. A mi compañero al Sr Ricardo Muñoz por formar parte de cada reto que hemos tenido a lo largo de nuestra carrera universitaria, demostrando que por más difícil que sea la situación el esfuerzo y trabajo duro hacen que lo imposible se vuelva posible.

A la Mg. Ximena López, quien más que una docente la considero una madre pues sus consejos y sabiduría me ha permitido crecer personal y profesionalmente. Además de inculcarme que a donde vaya siempre tengo que dar lo mejor de mí, dejando mi nombre y el de la universidad muy en alto.

Al Dr. José Luis Carrillo, por enseñarme que tengo que esforzarme por destacar y no formar parte del montón.

Al Mg. Diego Velasco, le agradezco por ser un docente que te motiva a ir dos pasos más allá y siempre estar preparados para alguna eventualidad.

También me gustaría agradecer a toda mi familia nuevamente por formar parte este proceso y logro universitario, además de continuar siendo mi motor de motivación y recordarme que la familia es lo más importante.

Finalmente, me gustaría agradecer a los Ingenieros Nicolás Schürmann, Fernando Herrera y Freddy Vega por ser grandes mentores en el área tecnológica además de brindarme sus conocimientos a través de los cursos realizados en plataformas virtuales, impulsando mi desarrollo personal y profesional a un nivel de mejor calidad.

“Nunca pares de aprender”

Platzi

Sr. Jonathan Navas

Tabla de contenido

Carátula.....	1
Certificación	2
Informe Urkund	3
Responsabilidad de autoría.....	4
Autorización	5
Dedicatoria.....	6
Dedicatoria.....	7
Agradecimiento.....	8
Agradecimiento.....	9
Tabla de contenido	10
Índice de tablas	14
Índice de figuras	15
Índice de ecuaciones.....	19
Resumen	20
Abstract.....	21
Generalidades	22
Planteamiento y Formulación del Problema.....	25
<i>Preguntas de Investigación.....</i>	28
Justificación e Importancia.....	28
Objetivos.	29
<i>Objetivo General.....</i>	29
<i>Objetivos Específicos.....</i>	30
Hipótesis.	30
Variables de Investigación.....	30
Marco Teórico	32
Introducción.....	32
Revisión Sistemática.	32
Antecedentes Históricos.....	53
<i>Evolución de métodos y técnicas de uso en la madera.....</i>	53
<i>Evolución de métodos y técnicas utilizadas para la medición del nivel de humedad en la madera.....</i>	54
Madera.....	60
<i>Introducción.....</i>	60
<i>Características de la madera.....</i>	60
<i>Componentes Físicos de la madera.....</i>	61

<i>Componentes Químicos de la madera</i>	62
<i>Enfermedades en la madera</i>	63
<i>Tipos de la madera</i>	65
<i>Humedad en la Madera</i>	67
<i>Uso sugerido de la madera de acuerdo al nivel de su humedad</i>	70
Laurel	71
<i>Introducción</i>	71
<i>Propiedades</i>	72
<i>Beneficios y Usos</i>	73
Software	74
<i>Introducción</i>	74
<i>Tipos de Software</i>	74
<i>Ciclo de vida del software</i>	76
<i>Arquitectura de Software</i>	85
<i>Calidad del Software</i>	89
Aplicaciones Móviles	94
<i>Introducción</i>	94
<i>Historia de las aplicaciones móviles</i>	94
<i>Sistemas operativos móviles</i>	95
<i>Tipos de aplicaciones móviles</i>	97
Metodologías para el desarrollo de software	98
<i>Introducción</i>	98
Pruebas de Software	107
<i>Introducción</i>	107
Inteligencia Artificial	109
<i>Introducción</i>	109
<i>Etapas de la Inteligencia Artificial</i>	110
<i>Tipos de la Inteligencia Artificial</i>	111
<i>Ramas de la Inteligencia Artificial</i>	112
<i>Visión Artificial o visión por Computadora</i>	120
<i>Redes Neuronales Convolucionales (CNN)</i>	122
<i>Arquitectura de una red neuronal convolucional (CNN)</i>	127
<i>Métricas de Evaluación de algoritmos de IA</i>	128
Herramientas de desarrollo de Software	131
<i>Introducción</i>	131
<i>Tipos de herramientas para el desarrollo de software</i>	132
<i>GitHub</i>	133

<i>Anaconda</i>	134
<i>Android Studio</i>	134
<i>Draw.io</i>	136
<i>Figma</i>	136
<i>Teachable Machine</i>	137
Implementación y prueba de un modelo de red neuronal convolucional	145
Introducción.....	145
Etapas	
Etapas 1. Recolección y captura de imágenes	146
Fase 1. Obtención de información de contenido de humedad	147
Fase 2. Selección del método de detección del nivel de humedad	150
Fase 3. Generación del dataset	152
Etapas 2. Preprocesamiento de imágenes del dataset	157
Fase 1. Ajuste del dataset	158
Fase 2. División del dataset	161
Etapas 3. Codificación manual del algoritmo	162
Fase 1. Preparación del entorno de desarrollo	163
Fase 2. Definición de las arquitecturas de redes neuronales convolucionales (CNN)	165
Fase 3. Importación de librerías	182
Fase 4. Definición de funciones	184
Fase 5. Carga y ajuste de datos	185
Fase 6. Creación de grupos de entrenamiento y pruebas	186
Fase 7. Ajuste de Hiperparámetros	187
Fase 8. Entrenamiento del modelo	189
Fase 9. Inspección de resultados de codificación manual	190
Etapas 4. Uso de Teachable Machine	191
Fase 1. Recopilación y agrupamiento de datos	192
Fase 2. Ajuste de hiperparámetros	192
Fase 3. Entrenamiento del modelo	193
Fase 4. Inspección de resultados	194
Etapas 5. Comparación de resultados entre la codificación manual y la herramienta Teachable Machine	195
Fase 1. División de datos de entrenamiento y pruebas	196
Fase 2. Ajuste de hiperparámetros para los modelos de CNN	196
Fase 3. Ejecución de modelos de redes neuronales convolucionales con datos de entrenamiento	198
Fase 4. Ejecución de modelos de clasificación con datos de pruebas	199
Etapas 6. Exportación del modelo de red neuronal convolucional	205

<i>Fase 1. Selección del tipo de exportación de modelo.</i>	206
<i>Fase 2. Selección del tipo de conversión del modelo.</i>	206
<i>Fase 3. Selección de la ruta donde se guarda el modelo.</i>	207
Análisis, diseño y desarrollo de un aplicativo móvil para el diagnóstico de humedad en la madera del Laurel.	208
Etapas	
Etapas 1. Levantamiento de requisitos de software.	208
<i>Requisitos Funcionales.</i>	211
<i>Requisitos no Funcionales.</i>	212
<i>Requisitos de Interfaz.</i>	213
Etapas 2. Diseño de arquitectura del aplicativo móvil.	214
<i>Diagrama de Caso de Uso.</i>	216
<i>Diagrama de Secuencia.</i>	217
<i>Diagrama de Paquetes.</i>	218
Etapas 3. Desarrollo del aplicativo móvil.	219
<i>Paso 1. Descarga de Código Base.</i>	220
<i>Paso 2. Configuración del Entorno.</i>	223
<i>Paso 3. Implementación del modelo de Red Neuronal.</i>	224
<i>Paso 4. Desarrollo de Interfaces.</i>	227
Implementación, pruebas y validación del aplicativo móvil para el diagnóstico de humedad en la madera del Laurel.	232
Etapas	
Etapas 1. Implementación del aplicativo móvil en el aserradero “San Martín”.	233
Etapas 2. Pruebas del Aplicativo Móvil.	235
Etapas 3. Validación de los resultados obtenidos.	246
<i>Aplicación de Encuesta de Usabilidad SUS.</i>	247
Conclusiones y Recomendaciones.	250
Conclusiones.	250
Recomendaciones.	252
Bibliografía	254
Anexos	264

Índice de tablas

Tabla 1 Sinónimos y términos relacionados de la búsqueda de artículos científicos utilizando la estrategia PICOC.....	35
Tabla 2 Actividades correspondientes al proceso de planificación e insumos.....	36
Tabla 3 Artículos hallados, seleccionados y precisión correspondiente a la cadena de búsqueda.	37
Tabla 4 Tipos de Investigaciones.....	38
Tabla 5 Área de Investigación.....	40
Tabla 6 Tabla comparativa de investigaciones extraídas de la revisión sistemática.....	50
Tabla 7 Tabla con características específica de las investigaciones halladas en la revisión sistemática.....	51
Tabla 8 Métodos de diagnóstico de contenido de humedad en la madera	59
Tabla 9 Usos recomendados de la madera en función al nivel de humedad.....	71
Tabla 10 Prototipos de Dataset	157
Tabla 11 Propiedades de una imagen.....	160
Tabla 12 Arquitectura Wood del Modelo 1.....	167
Tabla 13 Arquitectura Forest del Modelo 2.....	173
Tabla 14 Arquitectura MobileNet del modelo 3.....	178
Tabla 15 Ajuste de hiperparámetros.....	189
Tabla 16 Resultados de código manual.....	190
Tabla 17 Ajuste de hiperparámetros.....	194
Tabla 18 Ajuste de hiperparámetros del artículo científico.....	197
Tabla 19 Ajuste de hiperparámetros.....	198
Tabla 20 Comparativa de los mejores resultados de la exactitud (Accuracy), según su división de datos y ajuste de hiperparámetros en la fase de entrenamiento.....	199
Tabla 21 Comparativa de los mejores resultados de la F1, según su división de datos y ajuste de hiperparámetros en la fase de pruebas.....	204
Tabla 22 Casos de prueba.....	237
Tabla 23 Clases Agrupadas por el porcentaje de Humedad del Laurel.....	239
Tabla 24 Pruebas del aplicativo móvil	244
Tabla 25 Tabla de Tabulación de los valores en la encuesta SUS	248

Índice de figuras

Figura 1 Estructura del marco teórico	32
Figura 2 Pasos para realizar una revisión sistemática corta	33
Figura 3 Publicaciones halladas por año utilizando la cadena de búsqueda.....	41
Figura 4 Secado de Madera Aserrada	68
Figura 5 Clasificación de los tipos de Software según su función.....	75
Figura 6 Ciclo de vida Lineal.....	77
Figura 7 Ciclo de vida en Cascada.....	77
Figura 8 Ciclo de vida en V.....	78
Figura 9 Ciclo de vida tipo Sashimi.....	79
Figura 10 Ciclo de vida en Cascada con subproyectos.....	80
Figura 11 Ciclo de vida iterativo.....	81
Figura 12 Ciclo de vida por prototipos.....	81
Figura 13 Ciclo de vida evolutivo	82
Figura 14 Ciclo de vida incremental.....	83
Figura 15 Ciclo de vida en espiral.....	84
Figura 16 Ciclo de vida orientado a objetos	85
Figura 17 Arquitectura Monolítica	87
Figura 18 Arquitectura de Microservicios.....	88
Figura 19 Uso de tráfico web por un dispositivo móvil.....	96
Figura 20 Metodología diseño rápido de aplicaciones RAD.....	101
Figura 21 Metodología Programación Extrema XP	103
Figura 22 Organización de personal según la Metodología Crystal.....	105
Figura 23 Fases de la Metodología Mobile-D.....	107
Figura 24 Rendimiento del Deep Learning vs Machine Learning.....	114
Figura 25 Redes Neuronales para el reconocimiento de rostros	115
Figura 26 Estructura de una red neuronal	116
Figura 27 Entrenamiento desde cero vs Entrenamiento usando Transfer Learning.....	117
Figura 28 Estrategias recomendadas de uso en Transfer Learning	118
Figura 29 Flujo de la visión por computadora clásica	120
Figura 30 Flujo actual de la visión por computadora.....	121
Figura 31 Extracción de características de una imagen digital.....	121
Figura 32 Conversión de imágenes a una matriz numérica	123
Figura 33 Caracterización de una imagen en rgb.....	123
Figura 34 Multiplicación de la imagen de entrada por el kernel.....	124
Figura 35 Subsampling utilizando Max-Pooling.....	125
Figura 36 Proceso de la primera convolución	126
Figura 37 Proceso de la segunda convolución	126
Figura 38 Arquitectura de una CNN.....	128
Figura 39 Matriz de Confusión	129
Figura 40 Interfaz de GitHub.....	133
Figura 41 Suite de Anaconda	134
Figura 42 Interfaz Android Studio	135
Figura 43 Interfaz Draw.io.....	136
Figura 44 Interfaz de Figma.....	137

Figura 45 <i>Proceso de Extracción de Conocimiento mediante la Transferencia de Aprendizaje.</i>	138
Figura 46 <i>Tipos de Proyectos Teachable Machine</i>	139
Figura 47 <i>Interfaz Clasificador de Imágenes</i>	140
Figura 48 <i>Clasificación de animales usando Teachable Machine</i>	140
Figura 49 <i>Configuración de Parámetros Teachable Machine</i>	141
Figura 50 <i>Pruebas del modelo para la clasificación de imágenes</i>	142
Figura 51 <i>Resultados de la precisión por clase y la matriz de confusión clasificación de animales</i>	143
Figura 52 <i>Gráfica de precisión y pérdida por época</i>	143
Figura 53 <i>Exportación del modelo generado a través de Teachable Machine</i>	144
Figura 54 <i>Estructura del Capítulo III</i>	145
Figura 55 <i>Recolección y captura de imágenes en función al nivel de humedad</i>	146
Figura 56 <i>Fuentes de información</i>	148
Figura 57 <i>Selección del método de detección del nivel de humedad</i>	151
Figura 58 <i>Generación del dataset</i>	153
Figura 59 <i>Inspección visual</i>	155
Figura 60 <i>Método eléctrico</i>	156
Figura 61 <i>Pre procesamiento de imágenes del dataset</i>	158
Figura 62 <i>Ajuste del dataset</i>	159
Figura 63 <i>División del Dataset</i>	161
Figura 64 <i>Codificación manual del algoritmo</i>	163
Figura 65 <i>Preparación del entorno de desarrollo</i>	164
Figura 66 <i>Definición de la arquitectura de CNN</i>	166
Figura 67 <i>Arquitectura Wood</i>	168
Figura 68 <i>Implementación de la Arquitectura Wood en el Modelo 1</i>	169
Figura 69 <i>Entrada Modelo 1</i>	170
Figura 70 <i>Extracción de características modelo 1</i>	170
Figura 71 <i>Salida Modelo 1</i>	171
Figura 72 <i>Arquitectura Forest</i>	173
Figura 73 <i>Implementación de la Arquitectura Forest en el Modelo 2</i>	174
Figura 74 <i>Entrada Modelo 2</i>	175
Figura 75 <i>Extracción de características Modelo 2</i>	175
Figura 76 <i>Salida Modelo 2</i>	177
Figura 77 <i>Arquitectura MobileNet</i>	178
Figura 78 <i>Implementación de la Arquitectura MobileNet en el Modelo 3</i>	179
Figura 79 <i>Entrada modelo 3</i>	180
Figura 80 <i>Extracción de características modelo 3</i>	181
Figura 81 <i>Salida Modelo 3</i>	182
Figura 82 <i>Librerías del código manual</i>	183
Figura 83 <i>Definición de funciones</i>	184
Figura 84 <i>Carga y ajuste de datos</i>	186
Figura 85 <i>Creación de grupos de entrenamiento y pruebas</i>	186
Figura 86 <i>Ajuste de hiperparámetros</i>	188
Figura 87 <i>Entrenamiento del modelo</i>	190
Figura 88 <i>Uso de Teachable Machine</i>	191
Figura 89 <i>Recopilación y agrupamiento de datos</i>	192
Figura 90 <i>Ajuste de hiperparámetros</i>	193

Figura 91 <i>Entrenamiento del modelo.</i>	194
Figura 92 <i>Comparación de resultados.</i>	195
Figura 93 <i>Métrica Precisión.</i>	202
Figura 94 <i>Métrica Recall.</i>	203
Figura 95 <i>Exportación del modelo CNN.</i>	205
Figura 96 <i>Selección del tipo de exportación del modelo.</i>	206
Figura 97 <i>Selección del tipo de conversión del modelo.</i>	207
Figura 98 <i>Archivos que contienen un modelo entrenado.</i>	207
Figura 99 <i>Etapas para el desarrollo del aplicativo móvil propuesto.</i>	208
Figura 100 <i>Levantamiento de requisitos de software.</i>	209
Figura 101 <i>Especificación de requisitos de software (ERS).</i>	211
Figura 102 <i>Requisitos de Interfaz</i>	213
Figura 103 <i>Interfaces del aplicativo móvil para el diagnóstico de humedad del laurel.</i>	214
Figura 104 <i>Arquitectura de Software del aplicativo móvil.</i>	216
Figura 105 <i>Diagrama de Caso de Uso Aplicativo móvil.</i>	217
Figura 106 <i>Diagrama de Secuencia del aplicativo móvil.</i>	218
Figura 107 <i>Diagrama de Paquetes del aplicativo móvil.</i>	218
Figura 108 <i>Desarrollo del Aplicativo Móvil.</i>	219
Figura 109 <i>Repositorio Github código fuente base Tensorflow Android</i>	220
Figura 110 <i>Repositorio raíz del proyecto de TensorFlow</i>	221
Figura 111 <i>Descarga Manual de repositorios en Github</i>	222
Figura 112 <i>Selección del Path para almacenar el código fuente</i>	222
Figura 113 <i>Descarga del repositorio Utilizando Git Clone.</i>	223
Figura 114 <i>Interfaz Inicial Android Studio</i>	223
Figura 115 <i>Estructura del aplicativo móvil en Android Studio.</i>	224
Figura 116 <i>Importación del Modelo de Red Neuronal Convolutiva (CNN) en el proyecto del aplicativo móvil.</i>	225
Figura 117 <i>Reemplazo del modelo de red neuronal en el archivo ClassifierQuantizedMobileNet.java</i>	225
Figura 118 <i>Error no version NDK.</i>	226
Figura 119 <i>Ejecución Aplicativo móvil en un dispositivo virtual.</i>	227
Figura 120 <i>Traducción del Menú principal o home en aplicativo Móvil</i>	228
Figura 121 <i>Interfaz Principal Ejecutándose en el aplicativo móvil físico</i>	229
Figura 122 <i>Interfaz Información ejecutándose en un dispositivo virtual.</i>	230
Figura 123 <i>Interfaz Contacto ejecutándose en un dispositivo virtual.</i>	231
Figura 124 <i>Estructura Capítulo V.</i>	232
Figura 125 <i>Secado de madera en el aserradero "San Martín"</i>	233
Figura 126 <i>Fachada exterior aserradero "San Martín"</i>	234
Figura 127 <i>Implementación Aplicativo Móvil</i>	234
Figura 128 <i>Modelo sugerido para pruebas funcionales de aplicaciones software.</i> ..	236
Figura 129 <i>Evaluación de casos de prueba.</i>	238
Figura 130 <i>Pasos para realizar las pruebas del aplicativo móvil.</i>	240
Figura 131 <i>Selección de muestra</i>	240
Figura 132 <i>Medición a través del método eléctrico.</i>	241
Figura 133 <i>Configuración y Selección del modelo de Red Neuronal Convolutiva (CNN) en el Aplicativo Móvil.</i>	242
Figura 134 <i>Medición del contenido de humedad utilizando el aplicativo móvil.</i>	243

Figura 135 <i>Resultados de las Pruebas del Aplicativo móvil</i>	246
Figura 136 <i>Muestra con Moho</i>	247
Figura 137 <i>Escala de Usabilidad</i>	249

Índice de ecuaciones

Ecuación 1 <i>Humedad en la madera</i>	69
Ecuación 2 <i>Porcentaje de humedad</i>	69
Ecuación 3 <i>Fórmula de la métrica Precisión</i>	129
Ecuación 4 <i>Fórmula de la ecuación Sensitivity</i>	130
Ecuación 5 <i>Fórmula de la ecuación de Especificidad</i>	130
Ecuación 6 <i>Ecuación de F1 Score</i>	131
Ecuación 7 <i>Cálculo grado de usabilidad</i>	248

Resumen

La presente investigación tiene como propósito el desarrollo de un aplicativo móvil que optimiza el diagnóstico de contenido de humedad en la madera del laurel para el aserradero “San Martín”, mediante la implementación, prueba y comparación de 4 modelos de redes neuronales convolucionales (CNN). Mejorando la efectividad en el proceso de selección de materia prima de dicha madera. Realizándose una revisión sistemática, obteniéndose información de artículos científicos actuales, lo cual permite determinar la problemática a solucionar y su enfoque en posibles soluciones. Continuándose con el desarrollo y validación de un aplicativo móvil el mismo que está compuesto de 4 etapas: 1. Levantamiento de requisitos de software (estándar IEEE 830), 2. Diseño de arquitectura de software (monolítica), 3. Desarrollo del aplicativo móvil (metodología RAD), y 4. Pruebas y validación (encuesta SUS). Determinándose que el modelo de redes neuronales convolucionales (CNN) que presenta los mejores resultados en la clasificación y diagnóstico de contenido de humedad del laurel es la herramienta Google Teachable Machine obteniendo un 93.75% de efectividad. Además, cabe destacar que se obtiene un 84.25% del grado de usabilidad en el aplicativo móvil desarrollado, reflejando la calidad del mismo. Siendo una herramienta muy importante para los artesanos y expertos del aserradero.

Palabras clave:

- **MADERA LAUREL**
- **APRENDIZAJE PROFUNDO**
- **REDES NEURONALES**
- **VISIÒN POR COMPUTADORA**
- **SOFTWARE GOOGLE TEACHABLE MACHINE**

Abstract

The purpose of this research is the development of a mobile application that optimizes the diagnosis of moisture content in laurel wood for the "San Martin" sawmill, through the implementation, testing and comparison of 4 convolutional neural network (CNN) models. Improving the effectiveness in the process of selection of raw material of such wood. A systematic review was carried out, obtaining information from current scientific articles, which allows determining the problem to be solved and its approach to possible solutions. The development and validation of a mobile application is composed of 4 stages: 1. Software requirements gathering (IEEE 830 standard), 2. Software architecture design (monolithic), 3. It was determined that the model of convolutional neural networks (CNN) that presents the best results in the classification and diagnosis of moisture content of laurel is the Google Teachable Machine tool, obtaining 93.75% effectiveness. In addition, it should be noted that 84.25% of the degree of usability is obtained in the mobile application developed, reflecting the quality of the same. This is a very important tool for sawmill artisans and experts.

Key words:

- **MOISTURE IN LAUREL WOOD**
- **DEEP LEARNING**
- **CONVOLUTIONAL NEURAL NETWORKS**
- **COMPUTER VISION**
- **SOFTWARE GOOGLE TEACHABLE MACHINE**

Capítulo I

1. Generalidades

Desde el principio de la existencia de los seres humanos, la madera ha sido el material con mejores características físicas y químicas. La misma que ha sido utilizada como materia prima para la construcción de barcos, casas, armas de corto alcance, así como elaboración de muebles en aserraderos e industrias madereras (Cárdenas 2014).

En el mundo de la madera existen dos grandes campos a considerar: El sector forestal aprovecha dos tipos de recursos: maderables y/o no maderables. El primero es un recurso forestal que pueden ser transformado para el uso en la cotidianidad humana y forman parte de actividades económicas. El segundo es un bien de origen forestal excluyente de la madera que son usados para finalidades humanas como la alimentación, medicina y forraje (Hidalgo Broncano 2013).

El sector industrial maderero se enfoca en la siembra de árboles, apeo, y transformación de la materia prima(madera), pasando por procesos técnicos como el corte y moldeo para la posterior fabricación de productos derivados de madera a gran escala (Hidalgo Broncano 2013).

El proceso de transformación de manera artesanal consiste en la utilización de herramientas tradicionales y rudimentarias que requieren un mayor esfuerzo físico y mental, y que demanda tiempo y recursos para su aplicación. En contraste, el proceso industrial consiste en la utilización de herramientas sofisticadas y maquinaria pesada que permiten generar una mayor producción en el menor tiempo posible (Hidalgo Broncano 2013).

En el campo de la industria maderera ha existido un gran avance tecnológico desde los años 70, donde se ha buscado ubicar factores para mejorar el proceso de transformación de la madera uno de ellos es la humedad (factor clave para controlar el secado de la madera comparando la humedad relativa del aire y la humedad de la madera, para el secado perfecto), lo que facilita la selección de la materia prima óptima para su posterior utilización (Hidalgo Broncano 2013).

La madera puede ser secada de dos formas, siendo estas: 1) desecación natural y 2) artificial. La primera manera consiste en la deshidratación natural de una pieza de madera, eliminando cantidades pequeñas de agua en su estructura. Cabe destacar que este secado se lo realiza al aire libre por lo tanto su secado dependerá factores climáticos como el aire y el viento. Llegando a ser un secado lento. La segunda comprende la utilización de estructuras mecánicas para la eliminación y reducción de agua en la estructura maderera. La máquina más utilizada para este proceso es el horno industrial, en el que se introduce una pieza de madera y se lo seca configurando la corriente aire necesaria siendo un método de secado rápido (Hidalgo Broncano 2013).

Para establecer el contenido de humedad presente en la madera existen dos tipos de métodos: los no destructivos (método de inspección visual), y los destructivos (método de gravímetro y el método eléctrico), generando una mejor predicción del nivel de humedad en la materia prima y por ende productos con mayor calidad.

El método de inspección visual se basa en el empirismo de la persona, siendo la técnica más sencilla de todas y debe ser el primer paso en la evaluación de cualquier estructura de madera tomando características físicas en consideración como: el color, los anillos de crecimiento y la textura. El método de

inspección visual termina siendo de mucha utilidad debido a que no se requiere de equipos sofisticados para medir el nivel de humedad (Mata Jiménez 2011).

El método eléctrico consiste en la medición del contenido de humedad utilizando dispositivos que hacen uso de la resistencia de la madera al paso de corriente eléctrica, aquello provoca que las mediciones de humedad sean inmediatas y de forma no destructiva. Sin embargo, presentan un rango de confiabilidad del 6 al 25% del contenido de humedad presente en la madera (Foglia Córdoba 2005).

El método del gravímetro, consiste en el proceso de desecación de madera a través del uso de hornos industriales. Se dice que es el más preciso para el diagnóstico de contenido de humedad en la madera, ya que se utiliza cálculos matemáticos para evaluar la cantidad de agua en una pieza de madera (Mata Jiménez 2011).

El gran desarrollo que está viviendo la humanidad conjuntamente con la tecnología y la Inteligencia artificial en los últimos años, hace que cada vez existan más herramientas hechas con algoritmos inteligentes que utilizan los humanos en la vida cotidiana. El reconocimiento de imágenes forma parte del desarrollo de nuevas técnicas de Deep Learning las cuales han permitido tener un rendimiento muy superior al de los humanos en tareas de clasificación y detección de imágenes (Balboa Huanca et al. s. f.).

Con el propósito de mejorar las técnicas de medición del nivel de humedad, la tecnología (dispositivos móviles) y la inteligencia artificial mediante el uso de aprendizaje profundo puede dar origen a la creación de un nuevo método de medición no destructivo basándose en características físicas como: anillos de crecimiento, dimensión, color y textura de un corte de madera (laurel), dando como resultado un proceso optimizado del mismo, teniendo en cuenta que

optimizar significa conseguir los mejores resultados posibles, llegando a ser el mejor aporte al sector maderero.

De acuerdo con un estudio estadístico realizado por el Instituto Boliviano de Comercio Exterior para el año 2015. Se establece que el crecimiento en exportación maderera a nivel internacional es del 43% en países de primer mundo, sumando un total de \$232 000 millones de dólares, generando así un aumento de aproximadamente 8.3% a nivel mundial en un periodo de cinco años (Instituto Boliviano de Comercio Exterior 2016).

En Latinoamérica el crecimiento maderero también es notorio, al haber alcanzado en el año 2015 una cifra superior a \$14 000 000 por motivos de exportaciones. Por tal razón dicha región se ubica entre los territorios con más riqueza maderera (Instituto Boliviano de Comercio Exterior 2016).

Y finalmente en el Ecuador, durante el año 2015, el sector maderero exportó alrededor de \$344 millones y un total de 683 000 toneladas de madera, así como tableros, muebles, acabados para la construcción, papel y cartón (Instituto Boliviano de Comercio Exterior, 2016).

Además, la producción de madera es de aproximadamente 421 000 toneladas y está concentrada principalmente en variedades de madera fina, madera regular, madera para construcción y de pallets (estibas) y otros, generando alrededor de 90,000 puestos de trabajo beneficiando directamente a 235,000 familias (PROECUADOR 2015).

1.1. Planteamiento y Formulación del Problema.

De acuerdo, con la información entregada por el Instituto de Promoción de Exportaciones e Inversiones (PROECUADOR 2015) a lo largo de las últimas

cuatro décadas en la Provincia de Cotopaxi y en la parroquia de Pastocalle en particular, se ha ido conformando un complejo artesanal-industrial muy interesante en torno a la actividad maderera, específicamente en la fabricación de productos derivados de la misma siendo estos muebles (cocina, baño, comedor), carrocerías, y casas para animales de crianza.

Las tradiciones de los artesanos en la parroquia de Pastocalle han ido evolucionando con el pasar de los años, generando cada vez mejores técnicas de producción maderera, esto explica la constante creación de un gran número de negocios que van desde pequeños talleres artesanales hasta unidades de producción de tipo PYMES que son las pequeñas y medianas empresas.

El aserradero “San Martín” ubicado en la ciudad de Latacunga Provincia de Cotopaxi parroquia Pastocalle, ofrece a sus clientes una alta variedad de productos realizados con madera como: muebles de sala, cocina, comedor, baños, etc. Donde su principal objetivo es brindar a través de sus trabajos una alta calidad a sus clientes, motivo por el cual debe existir un óptimo diagnóstico de humedad en su materia prima (madera).

El nivel de humedad en la madera es uno de los parámetros más importantes a considerar en el proceso de selección de la madera, donde el artesano en base a su empirismo escoge la materia prima idónea para su posterior fabricación, ya que el nivel de humedad repercute directamente en la calidad final del producto, por lo tanto, si la madera es muy húmeda presenta deformaciones y debilidad, además de ser muy vulnerable a enfermedades (polilla, moho), mientras que si la madera es muy seca presenta porosidad y rajaduras por lo que tiende a reducir el tiempo de vida útil del producto.

Durante la transformación de la madera se genera una alta cantidad de residuos de materia prima, lo que produce un gran impacto ambiental, una de las

razones es la humedad (su medición se realiza de una manera artesanal y no técnica), lo cual repercute en la producción excesiva de residuos y produce pérdidas económicas para el artesano, debido a que la madera erróneamente seleccionada es desechada y destinada para otros usos (leña, aserrín), reflejándose posteriormente en la dificultad de realizar el proceso de fabricación, o incluso llegando a detener dicho proceso, generando un cliente insatisfecho.

El proceso de fabricación de productos derivados de la madera es muy importante en la parroquia de Pastocalle donde los talleres artesanales no cuentan con instrumentos informáticos de última tecnología pues dichos talleres se ven afectados directamente por lo siguiente:

- La variación de voltaje al utilizar maquinaria industrial.
- Exceso de polvo generado por las maquinarias durante el proceso de transformación de la materia prima (madera).
- Diseños de talleres no aptos para poseer equipos informáticos.

Por estas razones es recomendable hacer uso de dispositivos móviles tales como: celulares o tabletas, pues estos no requieren de cuidados especiales y son fáciles de transportar, además que pueden ser utilizados sin problema alguno en talleres industriales.

El presente trabajo de investigación tiene como propósito desarrollar un aplicativo móvil utilizando aprendizaje profundo que permita el diagnóstico del contenido de humedad existente en la madera (laurel), a través de la captura de imágenes (corte de laurel) los artesanos del aserradero "San Martín" obtendrán el nivel de humedad óptimo para la transformación de materia prima, permitiendo la utilización de la madera (laurel) de forma eficiente, previniendo el desperdicio del recurso maderero que influye directamente en el impacto ambiental y la calidad del producto final.

1.1.1. Preguntas de Investigación.

¿Cuál es la base teórica en la cual se basará el desarrollo de la presente investigación haciendo énfasis en el análisis del diagnóstico de humedad de la madera (laurel)?

¿Cuál es el procedimiento para la construcción de un modelo de red neuronal convolucional (CNN)?

¿Cuáles son los lineamientos necesarios para desarrollar el aplicativo móvil propuesto en la presente investigación, basándose en la metodología de desarrollo RAD y el estándar IEEE 830 para la especificación de requerimientos?

¿Cómo se puede implementar un aplicativo móvil que optimice el diagnóstico de humedad del laurel mediante aprendizaje profundo?

¿Cuáles son los pasos necesarios para validar y probar la información de salida del aplicativo móvil con respecto al diagnóstico de humedad en la madera (laurel)?

1.2. Justificación e Importancia.

En la actualidad en el sector maderero de la provincia de Cotopaxi específicamente en la parroquia Pastocalle no existe una técnica adecuada para determinar el contenido de humedad de la madera, ya que la mayoría de los aserraderos sólo lo hacen de la manera tradicional. Motivo por el cual la medición del nivel de humedad no es precisa ya que se la realiza únicamente en base a la experiencia del artesano lo que conlleva a una mala selección de materia prima, obteniendo como consecuencia productos de mala calidad y sin garantía, afectando directamente en la satisfacción de sus clientes.

Por esta razón las empresas de este cantón buscan de una u otra forma solucionar estos problemas:

- Pérdida de tiempo y dinero. - Como consecuencia de un diagnóstico erróneo en el nivel de humedad de la materia prima (madera), se generan productos con una mala calidad. Repercutiendo en el trabajo del artesano incluso obligándolo a volverlo a realizar, perdiendo tiempo y dinero.
- Contaminación ambiental. - El desperdicio innecesario de la madera como resultado de una mala selección de la materia prima.
- Portabilidad tecnológica. - Debido a que los talleres no poseen en sus instalaciones los medios necesarios para emplear equipos informáticos tradicionales.

Debido a las problemáticas que se mencionaron es necesario que los artesanos cuenten con una herramienta tecnológica portable que les permita medir el contenido de humedad en la madera de una forma rápida y sencilla, beneficiando directamente a los trabajadores del aserradero “San Martín” así como sus clientes, debido a que sus productos elaborados contarán con garantía y calidad, permitiendo reducir el impacto ambiental y económico, como consecuencia de una efectiva medición del nivel de humedad en la materia prima.

1.3. Objetivos.

1.3.1. Objetivo General.

- Desarrollar un aplicativo móvil que optimice el diagnóstico de humedad del laurel mediante aprendizaje profundo en el aserradero “San Martín” ubicado en la parroquia Pastocalle provincia de Cotopaxi.

1.3.2. Objetivos Específicos.

- Elaborar el marco teórico con los conceptos fundamentales y necesarios para el desarrollo del aplicativo móvil.
- Implementar y probar un modelo de red neuronal enfocado en una arquitectura de redes neuronales convolucionales (CNN).
- Analizar, diseñar y desarrollar un aplicativo móvil, basado en el estándar IEEE 830 y la metodología de desarrollo RAD (Desarrollo Rápido de Aplicaciones).
- Implementar el aplicativo móvil en el aserradero “San Martín” ubicado en la ciudad de Latacunga, provincia de Cotopaxi parroquia Pastocalle.
- Probar y validar el aplicativo móvil mediante pruebas de funcionalidad (precisión del modelo de red neuronal convolucional CNN) y la valoración de expertos maderos a través de la encuesta SUS en el aserradero “San Martín” ubicado en la ciudad de Latacunga, provincia de Cotopaxi parroquia Pastocalle.

1.4. Hipótesis.

¿Si se desarrolla un aplicativo móvil con aprendizaje profundo, entonces se optimizará el diagnóstico de humedad del laurel en el aserradero “San Martín” ubicado en la parroquia Pastocalle provincia de Cotopaxi?.

1.5. Variables de Investigación.

- Variable Independiente: desarrollo de un aplicativo móvil con aprendizaje profundo

- Variable Dependiente: optimización del diagnóstico de humedad del laurel en el aserradero “San Martín” ubicado en la parroquia de Pastocalle provincia de Cotopaxi.
- Indicadores:
 - Nivel de humedad detectado en la madera.
 - Tiempo de análisis para el diagnóstico de contenido de humedad.
 - Porcentaje de exactitud en la precisión del nivel de humedad en la madera.
 - Grado de satisfacción del artesano.

Capítulo II

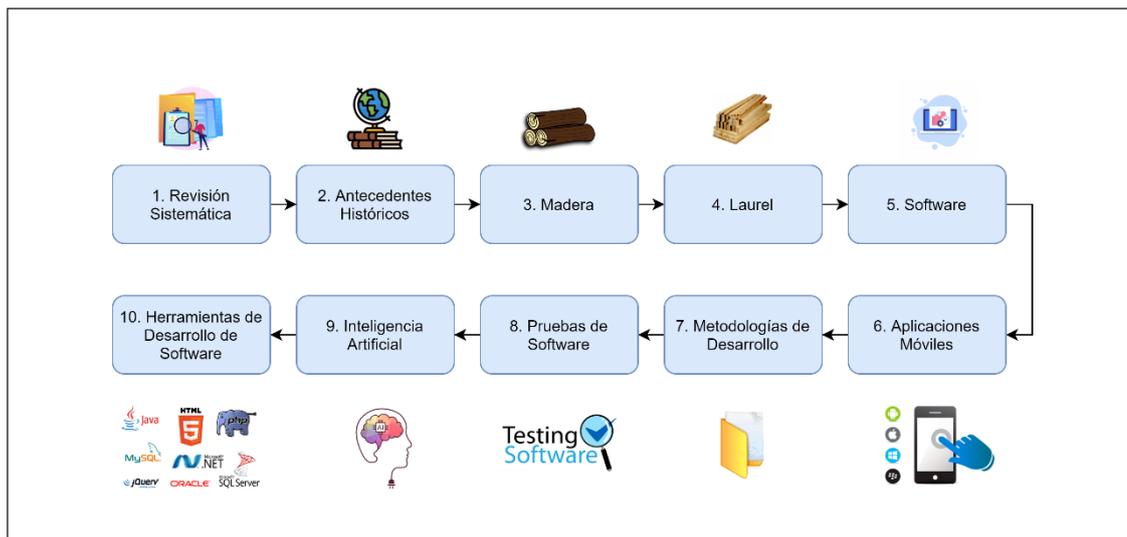
2. Marco Teórico

2.1. Introducción.

El presente capítulo contempla en detalle la estructura del marco teórico, como se puede observar en la Figura 1. A continuación, se especifica de forma resumida sus componentes: 1) revisión sistemática, 2) antecedentes históricos, 3) la madera, 4) laurel, 5) software, 6) aplicaciones móviles, 7) metodologías de desarrollo, 8) pruebas de software, 9) inteligencia artificial y 10) herramientas de desarrollo de software.

Figura 1

Estructura del marco teórico



Nota. En esta figura se muestra la estructura del Marco Teórico.

2.2. Revisión Sistemática.

La revisión sistemática (RS) es la recopilación y sintetización de información científica sobre algún tema en específico. Para ello se analiza aspectos de tipo cuantitativos y cualitativos de investigaciones realizadas a

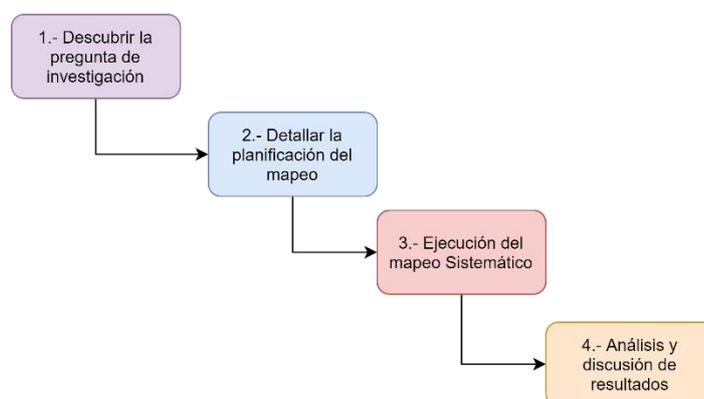
través de los años con el objetivo de minimizar los sesgos y las limitaciones de información. Las RS son recomendadas cuando se necesita conocer la actualidad de un tema de investigación en específico (Araujo 2011).

La generación de un mapeo sistemático requiere una guía metodológica. Según (Escobar-Sánchez y Carrillo-Medina 2017) consiste en realizar los siguientes pasos: 1) Describir la pregunta de la investigación 2) detallar la planificación del mapeo 3) ejecución del mapeo sistemático y análisis y discusión de resultados.

En la figura 2 se puede visualizar el proceso para realizar una revisión sistemática corta. Ya que el realizar una revisión sistemática completa conlleva más tiempo es decir un tema de investigación aparte. Razón por la cual su principal enfoque es obtener bibliografía actual (dos últimos años) de proyectos científicos de una manera técnica y rápida. Información importante y relevante en el campo de la inteligencia artificial en el área de visión por computadora utilizando como herramienta el Deep learning, específicamente a través del uso de redes neuronales convolucionales (CNN) para la clasificación de imágenes.

Figura 2

Pasos para realizar una revisión sistemática corta.



Nota. Esta figura muestra los pasos para realizar una revisión sistemática corta.

1) Pregunta de la Investigación.- Tiene como objetivo la recolección de información a través de fuentes bibliográficas, la cual estará sujeta a configuración de parámetros para que la interrogante sea lo más precisa posible, es por ello que la formulación de la pregunta de investigación arroja una cadena de búsqueda que será de utilidad durante la revisión sistemática (Bessaa et al. 2013).

La efectividad de una cadena de búsqueda depende directamente de los criterios de aceptación, así como exclusión. Por lo tanto, la pregunta en la presente investigación es la siguiente:

¿Cuáles son las investigaciones en el área de inteligencia artificial utilizando algoritmos de deep learning y redes neuronales convolucionales para la clasificación de imágenes en los dos últimos años?

2) Planificación del mapeo sistemático. – Involucra un conjunto de actividades organizadas de manera secuencial con el objetivo de que la investigación sea lo más precisa posible. Por tal razón es necesario que la selección de fuentes bibliográficas sea de origen confiable como por ejemplo revistas científicas, publicaciones, etc. Además, que también se debe seleccionar el idioma de origen de los recursos antes mencionados, así como el motor de búsqueda que se desee utilizar (Escobar-Sánchez y Carrillo-Medina 2017).

Como herramienta de motor de búsqueda se define a Google Scholar, además el idioma seleccionado para los recursos es el inglés, también se debe considerar utilizar la estrategia PICOC propuesta por (Bessaa et al. 2013) que corresponde a Población, Intervención, Contexto, Comparación, Output, Contexto.

- Población: Clasificación de Imágenes.
- Intervención: Redes Neuronales Convolucionales.

- Contexto: Deep Learning.
- Comparación: Algoritmos de Deep learning para la clasificación de imágenes.
- Salida: Estadística de artículos científicos que hayan utilizado en sus investigaciones Deep learning para la clasificación de imágenes.

Teniendo en cuenta los términos de PICOC, población, intervención y el contexto, se procede a definir la estructura de la cadena de investigación.

Utilizándose el término de Image clasification para la población, para la intervención se utiliza el término de Convolutional Neural Networks, para el contexto se usa el termino Deep learning. Los sinónimos relacionados con los términos de la pregunta de investigación que son utilizados se pueden observar en la tabla 1.

Tabla 1

Sinónimos y términos relacionados de la búsqueda de artículos científicos utilizando la estrategia PICOC

Estrategia PICOC	Término	Sinónimo / Término relacionado
Población	Image Clasification	
Intervención	Convolutional Neural Networks	CNN
Contexto	Deep learning	

Nota. Esta tabla muestra los sinónimos y términos que son relacionados con la búsqueda de artículos científicos utilizando la estrategia PICOC.

Durante la generación de la cadena de búsqueda se puede generar sub cadenas, utilizando operadores lógicos entre los componentes de PICOC establecidos anteriormente (Población, Intervención, Contexto, Comparación y Salida). Permitiéndose relacionar estos términos con sus respectivos sinónimos.

Una correcta planificación requiere de actividades necesarias para la búsqueda de artículos científicos que están relacionados con el propósito de esta

investigación. En los cuales se debe determinar los criterios de aceptación y los de exclusión como se ilustra en la tabla 2.

Tabla 2

Actividades correspondientes al proceso de planificación e insumos.

#	Actividad	Insumos
1	Selección de Fuentes Bibliográficas para la búsqueda de artículos científicos.	Fuentes: Artículos Científicos, Tesis. Motor de Búsqueda: Google Scholar. Idioma de Búsqueda: inglés.
2	Definición de los criterios de inclusión y de exclusión.	¿Cuáles son las investigaciones en el área de inteligencia artificial utilizando algoritmos de Deep Learning y redes neuronales convolucionales para la clasificación de imágenes en los dos últimos años?
3	Preselección de los recursos.	Uso de términos para la cadena de búsqueda, así como la exclusión de términos no deseados.
4	Selección de material científico (artículos).	Se toma en cuenta solo el material que cumpla con los criterios de búsqueda y permitan obtener aportes importantes a la investigación.
5	Definición de la cadena de búsqueda	cadena_búsqueda = "Image Clasification" AND "Convolutional Neural Networks" OR "CNN" AND "Deep learning"
6	Extracción de Información	Información general del material científico obtenido producto de la investigación

Nota. Esta tabla presenta las actividades correspondientes al proceso de planificación e insumos.

3) Ejecución del Mapeo Sistemático. - Cumplimiento de las tareas planificadas para la búsqueda de artículos científicos, para lo cual se empieza ingresando la cadena (cadena_búsqueda) en el motor de búsqueda Google Scholar. Los resultados de la búsqueda obtenida según los filtros configurados son el título, el resumen y las palabras claves de cada artículo, dando como respuesta 10 investigaciones sin embargo al aplicar los criterios de aceptación y exclusión se genera un resultado de 8 artículos que servirán como base de conocimiento para la presente investigación, como se presenta en la tabla 3.

Tabla 3

Artículos hallados, seleccionados y precisión correspondiente a la cadena de búsqueda

Motor de búsqueda	Artículos Encontrados	Artículos Escogidos	Precisión
Google Scholar	10	8	80%
Total	10	8	

Nota. Esta tabla muestra la cantidad de artículos hallados, seleccionados, así como la precisión correspondiente con la cadena de búsqueda.

Gracias a uso de la técnica de PICOC resulta más fácil y práctico el poder hallar información importante que sirva como respaldo científico para la presente investigación. Permittiéndose hallar información mucho más centralizada a través de los usos de los parámetros considerados en esta técnica. Como se pudo visualizar en la tabla 3. Obteniéndose un 80% de precisión en la búsqueda (8/10 artículos) valor aceptable considerando que el área de investigación es relativamente actual (dos últimos años).

Luego de determinar que recursos son excluidos por no generar aportes de valor a la presente investigación se procede a seleccionar 8 artículos que cumplen con los criterios de valor. Analizándose las siguientes características: 1) tipo de investigación, 2) cantidad de artículos encontrados, 3) porcentaje de

búsqueda, 4) título según tesis o artículo científico, 5) año de publicación, 6) referencia como se muestra en la tabla 4.

Tabla 4

Tipos de investigaciones

Tipo de Investigación	Cantidad de investigaciones	Porcentaje	Título	Año	Referencias
Caso de Estudio	8	100%	1.- Development and application of education program on understanding artificial intelligence and social impact.	2020	(Kim et al. 2020)
			2.- Teachable machine: approachable web-based tool for exploring machine learning classification.	2020	(Carney et al. 2020).
			3.- Feasibility study of google's teachable machine in diagnosis of tooth-	2020	(Jeong 2020).

Tipo de Investigación	Cantidad de investigaciones	Porcentaje	Título	Año	Referencias
			marked tongue.		
			4.- Indonesian sign language (bisindo) as means to visualize basic graphic shapes using teachable machine.	2020	(Harditya 2020).
			5.- Transfer learning for road-based location classification of non-residential property.	2021	(Mizgirev et al. 2021).
			6.- Computer vision and building envelopes.	2021	(Anani-Manyo 2021).
			7.- Implementation of a deep learning model for automated classification of aedes aegypti (linnaeus	2021	(Ong et al. 2021)

Tipo de Investigación	Cantidad de investigaciones	Porcentaje	Título	Año	Referencias
) and aedes albopictus (skuse) in real time.		
			8.-Insect detection using a machine learning model	2021	(Gupta y Homchan 2021).
Total	8	100%			

Nota. Esta tabla muestra los resultados utilizando la cadena de búsqueda, así como la filtración de artículos científicos.

Como se pudo observar en la tabla 4 las investigaciones pertenecen a los dos últimos años de acuerdo a la cadena de búsqueda preestablecida anteriormente. Obteniéndose distintas áreas de estudio como se puede observar en la tabla 5.

Tabla 5

Área de Investigación

Área de investigación	Cantidad de investigaciones	Porcentaje	Uso de Inteligencia Artificial	Referencias
Arquitectura	1	12.5%	SI	(Anani-Manyo 2021)
Biodiversidad	2	25%	SI	(Ong et al. 2021), (Gupta y Homchan 2021)
Comunicación	1	12.5%	SI	(Harditya 2020)
Educación	2	25%	SI	(Kim et al. 2020),

Área de investigación	Cantidad de investigaciones	Porcentaje	Uso de Inteligencia Artificial	Referencias
				(Carney et al. 2020)
Salud Oral	1	12.5%	SI	(Jeong 2020)
Sector Inmobiliario	1	12.5%	SI	(Mizgirev et al. 2021)
Total	8	100%	/	

Nota. Esta tabla muestra a que área corresponde cada investigación seleccionada previamente en la tabla 4.

En la figura 3 se puede observar la clasificación de las investigaciones halladas y representadas en la tabla 4 de acuerdo al año en que fueron publicadas, obteniéndose un total de 4 publicaciones para el año 2020 que corresponden a un 50% y el otro 50% para el año 2021 respectivamente.

Figura 3

Publicaciones halladas por año utilizando la cadena de búsqueda



Nota. Esta figura muestra la cantidad de publicaciones ya sean artículos, tesis o libros que estén dentro del campo investigativo al usar la cadena de búsqueda.

4) Resumen, análisis y discusión de resultados. – Presenta un resumen de las investigaciones halladas en la revisión sistemática permitiendo extraer su información más importante, para su posterior revisión y discusión.

Para lo cual se realizan las siguientes fases: a) resumen, b) análisis y discusión de resultados.

a) Resumen. - En esta fase se leen cada uno de los 8 artículos seleccionados previamente en el Capítulo II (Fase 3. Ejecución del mapeo sistemático) extrayéndose lo más importante. Cabe destacar que se utilizó fundamentalmente el abstract y las conclusiones. A continuación, se listan los resúmenes correspondientes.

1) Development and application of education program on understanding artificial intelligence and social impact.

La tecnología ha tenido un impacto muy importante en la evolución de la humanidad pues ha permitido facilitar la vida de las personas, así como ha permitido crear grandes problemas para la sociedad, es por ello que en esta investigación se busca hallar una manera didáctica y práctica de enseñanza a los estudiantes de secundaria con respecto a la influencia de la inteligencia artificial en la ética de la sociedad en ejemplos de empresas como Uber y YouTube (Kim et al. 2020).

Los investigadores de este artículo determinaron una actividad donde los estudiantes generaron un dataset de perros y gatos. Posteriormente con el uso de la herramienta Google Teachable Machine se creó un modelo de aprendizaje automático. De esta manera, se pudo concluir que la satisfacción y el interés de los alumnos por aprender temas de inteligencia artificial es alta. Haciendo énfasis en la ética y su importancia en la sociedad moderna (Kim et al. 2020).

2) Teachable machine: approachable web-based tool for exploring machine learning classification.

Con el avance tecnológico en los últimos años áreas como la educación han permitido crear nuevas investigaciones que han aportado al sector científico,

es por ellos que en este artículo tiene como objetivo analizar la facilidad y uso de Teachable Machine como herramienta para enseñanza de inteligencia artificial a los estudiantes de una forma didáctica (Carney et al. 2020).

En esta investigación se generó un dataset de 863 imágenes las cuales corresponden a tres clases de bananas: inmaduras, maduras y muy maduras para que los estudiantes comprendan fácilmente la gran utilidad que puede tener este tipo de investigaciones en el mundo real. Utilizando las siguientes configuraciones de hiperparámetros 50 épocas, batch_size de 16 y un learning rate de 0.0001 se obtuvo un modelo clasificador de imágenes con un 100% de precisión (Carney et al. 2020).

Lo más importante en este artículo es que Google Teachable Machine no solo es una herramienta de apoyo educativo sino un instrumento muy poderoso a tener en cuenta para investigaciones que requieran de algoritmos de aprendizaje automático en distintas áreas (Carney et al. 2020).

3) Feasibility study of google's teachable machine in diagnosis of tooth-marked tongue.

La medicina, así como la tecnología han ido avanzando a pasos agitados pues estas ramas van de la mano desde el origen de los tiempos, es por ello que resulta muy importante tener avances que aporten a la comunidad científica, además es de conocimiento general que la salud y la higiene bucal es un factor muy significativo a considerar, pues se sabe que la boca es la primera puerta de acceso al sistema digestivo por lo que es necesario tomar medidas preventivas para enfermedades (Jeong 2020).

Tomando en cuenta lo anterior esta investigación propone la clasificación de imágenes para diagnosticar lenguas marcadas por dientes. Donde esta puede tener Macroglosia el cual consiste en un trastorno físico en el que la lengua es

más grande de lo normal generando consecuencias graves a futuro. Por tal razón se plantea el uso de Google Teachable Machine para creación de un modelo de aprendizaje automático (Jeong 2020).

En esta investigación se generó un dataset de 1250 imágenes divididas entre las siguientes clases: lenguas marcas por dientes con un total de 634 muestras y lenguas no marcadas por dientes con 491 muestras respectivamente, además de la configuración de hiperparámetros con 75 épocas, un batch_size de 128 y un learning rate de 0.0001 obteniendo como resultado una precisión de 92.1% (Jeong 2020).

Para finalizar, se concluye que los resultados de precisión de este modelo son superiores a los resultados en investigaciones pasadas en el área de higiene dental los cuales fueron generados a través de modelos manuales de redes neuronales convolucionales por tal motivo es recomendable usar Teachable Machine en futuras investigaciones pues no requiere de altos recursos computacionales además de generar mejores resultados (Jeong 2020).

4) Indonesian sign language (Bisindo) as means to visualize basic graphic shapes using teachable machine.

La comunicación ha sido un pilar fundamental en la evolución de los seres vivos y más aún en los seres humanos, pues es más que una necesidad poder comunicarnos con otras personas, sin embargo, para personas con discapacidad especialmente para personas sordo mudas resulta muy difícil poder establecer un mecanismo de comunicación. Por tal razón el objetivo de esta investigación es generar un medio de comunicación a través del uso de la tecnología, la inteligencia artificial y los modelos de aprendizaje automático (Harditya 2020).

En este artículo se destaca el uso de lenguaje de señas Bisindo o IndoSI. Frente a esta problemática se propone la adaptación del lenguaje de señas Bisindo a una manera digital utilizando el servicio de Google Teachable Machine para que por medio de posturas de la persona un software le permita a un/a usuario sordomudo/a establecer una comunicación sin problema alguno (Harditya 2020).

El modelo de aprendizaje automático toma como referencia distintas posiciones del cuerpo humano dando como resultado una herramienta de comunicación muy efectiva. Finalmente, la investigación exhorta a la comunidad científica a realizar más investigaciones para que las personas con discapacidad puedan comunicarse y no tener problemas en la sociedad, además de mencionar que la Herramienta Teachable Machine fué muy útil en la creación de modelos de aprendizaje automático (Harditya 2020).

5) Transfer learning for road-based location classification of non-residential property.

A lo largo de la historia la ubicación de las propiedades siempre ha tenido un enfoque económico muy importante, pues ha sido necesario clasificar dichas propiedades según su sector para generar un valor comercial y esta sea rentable, además de facilitar la toma de decisiones de gestión en el ámbito de las finanzas y préstamos. Por tal razón esta investigación se centra en la utilización de un servicio de Google llamado Teachable Machine para la generación de un modelo de aprendizaje automático que permita clasificar propiedades no residenciales tomando en consideración su ubicación respecto a las carreteras principales (Mizgirev et al. 2021).

En esta investigación se propone la generación de un dataset de 2363 imágenes extraídas de la ciudad de Moscow, a través de un mapa de calor del

precio real de cada propiedad según el sector. Posteriormente se procede a dividir dichas imágenes en 3 categorías o clases: red line (978), minor road (466), inside quarter (919) respectivamente (Mizgirev et al. 2021).

Lo más importante en esta investigación es que se sugiere que las clases contemplen el mismo número de muestras, pues al no tener un equilibrio en la primera prueba que se realizó el algoritmo tuvo una precisión del 84% mientras que cuando se dejó las dos clases con mayores muestras en equilibrio y se obtuvo una precisión del 93.4% (Mizgirev et al. 2021).

Se concluye que el uso de Teachable Machine puede resolver un problema de clasificación de propiedades y es recomendado para usar en múltiples investigaciones al no necesitar de un gran consumo de recursos y programación como lo requeriría realizar el modelo de aprendizaje automático de forma manual (Mizgirev et al. 2021).

6) Computer vision and building envelopes.

La inteligencia artificial (IA) ha venido revolucionando en los últimos años de forma impresionante la vida cotidiana de los seres humanos, pues cada vez es más común ver que algoritmos de visión por computadora tienen más aplicaciones en un entorno real, llegando a ser muy importante en el sector de la arquitectura, pues dicha área necesita cada vez de mayor precisión y evolución. Por estas razones resulta trascendental conocer que los techos de los edificios deben ser revisados y evaluados de forma periódica, por lo tanto, esta investigación propone el uso de un algoritmo de redes neuronales convolucionales (CNN) y la utilización de una herramienta intuitiva en la generación de modelos de aprendizaje automático para la clasificación de imágenes (Anani-Manyo 2021).

El investigador propone la generación de un dataset de 707 imágenes las cuales la divide en dos sub investigaciones, la primera corresponde a la clasificación de techos teniendo en cuenta el tipo de material (asfalto, arcilla, concreto y metal) y la otra la clasificación de techos por su estado dañado y no dañado) (Anani-Manyo 2021).

El autor de esta investigación desarrolla un algoritmo para clasificación de imágenes de forma manual utilizando CNN obteniendo una precisión de 64% mientras que utilizando la herramienta Teachable Machine obtuvo una precisión del 84% dando como resultado que la herramienta presenta mejores valores con respecto a la clasificación de techos por tipo de material, mientras que en la otra investigación el autor verifica si el techo de un edificio está o no dañado ambos algoritmos presentan una precisión del 100%. Por tal razón se menciona que es si factible usar ambos métodos para resolver problemas de clasificación de imágenes en un entorno real (Anani-Manyo 2021).

7) Implementation of a deep learning model for automated classification of *Aedes Aegypti* (Linnaeus) and *Aedes Albopictus* (Skuse) in real time.

La identificación de especies a lo de la historia ha sido una tarea muy complicada en el área de las ciencias de la vida y más aún para los profesionales de dicha área, es por ello que en esta investigación se propone poder clasificar a las especies de mosquitos *Aedes aegypti* (Linnaeus) y *Aedes albopictus* (Skuse) a través de un modelo de Deep learning y redes neuronales convolucionales profundas, para ser usados posteriormente en un hardware que permita leer dicho modelo y clasificar ambas especies (Ong et al. 2021).

En esta investigación se genera un dataset de 4120 imágenes entre (*Linnaeus* y *Skuse*) con más de 12 días de vida teniendo en cuenta

características como la especie, la cabeza y el tórax. Un profesional tiene una precisión de clasificación de dichas especies entre 80 y 97% es por ello que el grupo de investigación utilizó como herramienta para generar el modelo de aprendizaje automático a Google Teachable Machine realizando distintas configuraciones para obtener los mejores resultados, al dataset de imágenes se divide entre 2884 imágenes de entrenamiento y 1236 para pruebas así como hiperparámetros 30 épocas, batch_size de 16 y un learning rate de 0.001, como consecuencia de dichas configuraciones se logró una precisión del 98.3% llegando a ser mucho más precisa comprándola con la clasificación de un profesional de forma manual en el área (Ong et al. 2021).

Para finalizar la investigación se concluye, que este estudio permitirá generar un marco de referencia para futuros estudios en el área de las ciencias de la vida con muchas posibilidades de implementaciones en distintos tipos de hardware (Ong et al. 2021).

8) Insect detection using a machine learning model.

A través de la historia la humanidad ha buscado hallar características físicas en las especies para poder agruparlas de una manera que permita una correcta identificación. Es por ello que en esta investigación se pretende poder clasificar especies de grillos (*Acheta domesticus*, *Gryllus Bimaculatus*) con importancia económica, para lo cual se utilizó una plataforma basada en interfaz gráfica llamada Teachable machine de Google en donde se pueden crear modelos de aprendizaje automático de una manera muy eficaz (Gupta y Homchan 2021).

Se generó un dataset de 2646 imágenes de las especies, tomando en cuenta la especie y su sexo, posteriormente se dividió las imágenes en dos grupos 2247 imágenes para entrenamiento y 399 para pruebas del modelo de

aprendizaje automático con la siguiente configuración de hiperparámetros: 100 épocas, batch_size de 16 (Gupta y Homchan 2021).

Como resultado del entrenamiento del modelo se obtuvo una precisión del 100% en la identificación de especies por sexo, además dicho modelo pudo ser implementado en un aplicativo móvil para cumplir la funcionalidad de clasificación de especies de grillos. Finalmente, la investigación concluye que existe una amplia apertura para que profesionales en el campo de las ciencias de la vida puedan desarrollar más modelos de aprendizaje automático que permita clasificar muchas más especies (Gupta y Homchan 2021).

b) **Análisis.** – En esta fase se procede a leer y estudiar detenidamente los resúmenes de los artículos científicos obtenidos en la fase anterior (Fase a.- Resumen). Presentándose aquellas características más relevantes de dichos estudios tomando en consideración: 1) objetivo, 2) Aplicabilidad y 3) cantidad de imágenes que se utilizaron en la elaboración del dataset correspondiente. Así como también aspectos específicos tales como: 1) Fecha de publicación, 2) Tipo de recurso, 3) País, 4) Área de investigación, 5) Utilización de la herramienta Google Teachable Machine, y 7) Codificación manual de algoritmos de Deep Learning como se pueden observar respectivamente en las tablas 6 y 7.

Tabla 6

Tabla comparativa de investigaciones extraídas de la revisión sistemática.

N°	Objetivo	Aplicabilidad	Cantidad de imágenes usadas
1	Clasificación de Imágenes para la detección de animales (perros y gatos).	Concientización de los estudiantes y la importancia de la inteligencia artificial de forma didáctica a través de la generación de un clasificador de perros y gatos	No se menciona
2	Clasificación de Imágenes para la detección de bananas y su nivel de madurez como material de aprendizaje.	Manejo de la Herramienta Teachable Machine como instrumento didáctico para la enseñanza de inteligencia artificial.	No se menciona
3	Clasificación de Imágenes para el diagnóstico de lenguas marcadas por los dientes.	Realización de un diagnóstico de lenguas marcadas por dientes a través del uso de un modelo inteligente creado en la herramienta Teachable Machine.	Total 1250 Entrenamiento 1125 - (90%) Pruebas 125 - (10%)
4	Clasificación de Objetos e Imágenes para la generación del lenguaje de señas BISINDO.	Creación de un modelo de redes neuronales convolucionales para la generación de un lenguaje de señas usando Teachable Machine.	No se menciona
5	Clasificación de Imágenes para la ubicación de propiedades residenciales según su sector con respecto a las carreteras.	Uso de la herramienta Teachable machine para la generación de un modelo inteligente para la clasificación de propiedades no residenciales.	Total 5000 Entrenamiento 4250 (85%) – Pruebas 750 – (15%)
6	Clasificación de Imágenes para la detección de materiales de construcción, evaluación de detalles y la identificación de cubiertas de edificios	Comparativa de un algoritmo de redes neuronales convolucionales vs el uso de Teachable machine para la generación de un modelo inteligente en la clasificación de materiales de construcción.	Total 707 Entrenamiento 600 (85%) – Pruebas 107 – (15%)
7	Clasificación de Imágenes para la detección de Insectos (Aedes aegypti y Aedes albopictus).	Implementación de un modelo inteligente para la clasificación de insectos a través del uso de un hardware clasificador.	Total 4120 Entrenamiento 2284 – (70%) Pruebas 1236 – (30%)

N°	Objetivo	Aplicabilidad	Cantidad de imágenes usadas
8	Clasificación de Imágenes para la detección de Insectos (Acheta domesticus y Gryllus bimaculatus).	Generación de un modelo de aprendizaje automático para la clasificación de insectos.	Total 2646 Entrenamiento 2247 – (85%) Pruebas 399 – (15%)

Nota. Esta tabla muestra un extracto de las investigaciones halladas a través de la revisión sistemática.

Tabla 7

Tabla con características específicas de las investigaciones halladas en la revisión sistemática.

N°	Fecha de publicación	Tipo	País	Área de Investigación	Uso de Teachable Machine	Codificación manual de algoritmos de Deep Learning
1	Marzo 2020	Paper	Corea	Educación	SI	NO
2	Abril 2020	Paper	EEUU	Educación	SI	NO
3	Noviembre 2020	Paper	Corea	Salud Oral	SI	NO
4	Diciembre 2020	Paper	Indonesia	Comunicación	SI	NO
5	Abril 2021	Paper	Rusia	Sector Inmobiliario	SI	NO
6	Abril 2021	Tesis de Maestría	EEUU	Arquitectura	SI	SI
7	Mayo 2021	Paper	Malasia	Biodiversidad	SI	NO
8	Mayo 2021	Paper	Tailandia	Biodiversidad	SI	NO

Nota. Esta tabla muestra de forma detallada características específicas de las investigaciones halladas en la revisión sistemática.

- c) Discusión de resultados.** – En esta fase se procede a contrastar los resultados de las tablas 6 y 7 los mismos que fueron obtenidos del análisis de los artículos científicos hallados en la etapa anterior (Fase b.- Análisis). Llegándose a concluir los siguientes resultados:

- Los recursos hallados son actuales, de gran impacto investigativo, así como son fuentes confiables de información. Considerándose como respaldo científico para la presente investigación.
- Los dataset utilizados en su mayoría recomiendan que para tener una alta precisión en la clasificación de imágenes a través del uso de redes neuronales convolucionales es necesario balancear sus clases.
- La cantidad mínima de imágenes para la generación de los dataset de entrenamiento y pruebas utilizados establecen como mínimo utilizar 707 imágenes.
- Google Teachable Machine permite la generación de modelos de redes neuronales convolucionales garantizando una alta precisión (84% - 100%) en la predicción de clasificación de imágenes. Siendo una herramienta muy interesante a tomar en cuenta para la construcción del marco teórico de esta investigación por sus buenos resultados vistos en los artículos científicos seleccionados.

Cabe destacar que un factor de suma importancia y trascendencia para el éxito de una investigación radica en realizar una revisión sistemática.

Herramienta metodológica que permite tener una visión adecuada al momento de realizar un proyecto de investigación. Presentándonos la situación actual de la problemática a solucionar y un enfoque de sus posibles soluciones.

2.3. Antecedentes Históricos.

2.3.1. Evolución de métodos y técnicas de uso en la madera.

Desde el origen de los tiempos la madera ha sido el material predilecto en la humanidad, debido a sus bondades físicas que le han permitido ser considerado como combustible, material para la construcción, elaboración de armas (lanzas), así como para la fabricación de flotas de barcos (Cárdenas 2014).

En medida que la humanidad fué evolucionando, el uso de la madera era cada vez más necesario, es por ello que en el antiguo Egipto hace más de 3000 años antes de Cristo, los artesanos madereros generaron la técnica de la “laminación de la madera”, la cual consistía en unir varios pedazos de madera de tamaño reducido, para crear objetos mucho más grandes (Cárdenas 2014).

Aproximadamente a inicios del siglo XIX aparece la técnica del “enchapado”, la cual consistía en la superposición de láminas de madera en otros objetos, esta técnica fue usada para la decoración de muebles en ese entonces ya que tenía como objetivo mejorar la presentación de productos elaborados con madera. Esta técnica en primer lugar fue considerada para uso artesanal ya que requería de cierto conocimiento previo en la madera y su trabajo dificultoso de corte y pegado (Cárdenas 2014).

A finales del siglo XIX surge la revolución industrial en el campo maderero, convirtiendo a la madera en un material duradero, fuerte y versátil. A partir de dicha revolución la madera fue considerada como la materia prima predilecta, siendo empleada para la construcción de aeronaves, vehículos, embarcaciones, entre otros (Cárdenas 2014).

Durante el siglo XX aparece la técnica del “contrachapado” que consistió en la unión de varias láminas de madera, mejorando la técnica del enchapado

tradicional, pues el contrachapado permitía que la madera sea más flexible facilitando el trabajo de moldeado por parte del artesano (Cárdenas 2014).

2.3.2. Evolución de métodos y técnicas utilizadas para la medición del nivel de humedad en la madera.

La madera posee propiedades naturales de absorción de cantidades considerables de agua en su estructura. Se conoce que existen tres formas en las que el agua se puede mostrar, siendo estas: 1) Constitución, 2) Impregnación y 3) Libre. A continuación, se listan dichas formas de presentación de agua (Bermejo, Félix 2014):

- 1) Constitución: Se encuentra ubicada en las propiedades químicas de una estructura maderera.
- 2) Impregnación: Se encuentra ubicada en las paredes de la madera.
- 3) Libre: Se encuentra ubicada alrededor del lumen de una estructura maderera.

La humedad que se encuentra situada en la madera existe debido a la relación de partículas de agua y el peso natural de la madera. Razón por la cual se establece que dicho nivel de humedad debe ser expresado en porcentaje. Además, la madera puede ser clasificada en: 1) madera verde, 2) madera con humedad, 3) madera seca a través del aire y 4) madera seca mediante hornos. A continuación, se describe brevemente dicha clasificación (Bermejo, Félix 2014):

- Madera verde: Se dice que existe este tipo de madera, cuando dicha humedad supera el 30% hasta llegar a un 70%, dependiendo de la familia que pertenezca.

- Madera con humedad: Comprende la aparición de muestras de madera con humedad entre un rango superior al 20% y menor del 30% en diferentes familias madereras.
- Madera seca a través del aire: Comprende la aparición de muestras de madera con un porcentaje inferior al 18% y superior al 13%.
- Madera seca en cámara: Comprende la aparición de muestras de madera con un porcentaje inferior al 12%

A continuación, se presentan los 3 métodos más utilizados para el proceso de medición y diagnóstico de contenido de humedad en la madera: 1) Método de Inspección Visual, 2) Método del gravímetro y 3) Método Eléctrico.

2.3.2.1. Método de inspección visual

Este método tiene como principio el empirismo de la persona, siendo la técnica más sencilla de todas y debe ser el primer paso en la evaluación de cualquier estructura de madera tomando características físicas en consideración como: el color, los anillos de crecimiento y la textura. El método de inspección visual termina siendo de mucha utilidad debido a que no se requiere de equipos sofisticados para medir el nivel de humedad (Mata Jiménez 2011).

Con esta técnica el artesano maderero puede realizar un reconocimiento de detalles, integridad, deficiencias y daños en la estructura de la madera sin embargo al tratarse de una técnica subjetiva, la medición depende de la agudeza visual, así como su experiencia, por lo tanto, dicha revisión arroja datos cualitativos mas no cuantitativos, además de limitarse a únicamente a la superficie externa de la madera sin tomar en cuenta posibles enfermedades que puede padecer internamente como pudriciones, hongos.

2.3.2.2. Método del gravímetro.

Consiste en el proceso de desecación de madera a través del uso de hornos industriales. Se dice que es el más preciso para el diagnóstico de contenido de humedad en la madera, ya que se utiliza cálculos matemáticos para evaluar la cantidad de agua en una pieza de madera (Mata Jiménez 2011). A continuación, se detallan las actividades principales que este método utiliza para la medición de humedad:

1) Obtención de muestras. - Actividad que permite tomar una fracción del pedazo de madera para poder ser analizada, posteriormente dicha muestra debe tener una dimensión superior a 50cm debido a que esta medida representa el nivel de humedad sin pérdida, cabe mencionar que la muestra no debe tener defectos, daños, etc. Además de que los cortes deben hacerse con cuchillas previamente afiladas para evitar el calentamiento del material que podría resultar en pérdida de humedad.

2) Medición del peso inicial. – Actividad que permite pesar muestras de madera en balanzas con una precisión de 0.1gr, teniendo en consideración que la sensibilidad en la medición dependerá del tamaño de la muestra de madera es decir mientras más pequeña sea la muestra se deberá aumentar la sensibilidad de la balanza.

3) Secado de las muestras. - Actividad que permite el sacado de la madera para lo cual dichas muestras se introducen en hornos. Además, es recomendable que se disponga de hornos con termostato regulable esto debido a que la temperatura ideal para secar la madera es 103 ± 2 °C aproximadamente. Sin embargo, si la temperatura sobrepasa los 105°C este puede perder peso significativamente, no solo por la cantidad de agua sino por la evaporación de otras sustancias que contiene la madera.

4) Medición de peso constante. - Actividad que permite evaluar de forma constante a la madera hasta alcanzar un peso ideal. Una vez que la humedad es removida de la madera esta se debe enfriar rápidamente en un ambiente seco para impedir que vuelva a adquirir humedad.

2.3.2.3. Método eléctrico.

Método que consiste en la medición del contenido de humedad utilizando dispositivos que hacen uso de la resistencia de la madera al paso de corriente eléctrica, aquello provoca que las mediciones de humedad sean inmediatas y de forma destructiva al utilizar puntas de acero que son introducidas en la estructura maderera. Sin embargo, presentan un rango de confiabilidad del 6 al 25% del contenido de humedad presente en la madera (Foglia Córdoba 2005).

Según (Mata Jiménez 2011), manifiesta que el método eléctrico se subdivide en dos técnicas: 1) Resistencia Eléctrica y 2) Constante eléctrica las cuales se presentan a continuación:

1) Resistencia Eléctrica: La madera es considerada como aislante de energía eléctrica, no obstante, también cuenta con variaciones de conductividad eléctrica. Estos tipos de variaciones son comúnmente relacionados con la orientación que tiene sus fibras, la temperatura del clima y el contenido de humedad presente en la madera. Por tal motivo este método es muy usado para la medición del contenido de humedad. Esta técnica tiene como principio a la resistencia eléctrica de los materiales (madera). Se debe tener en cuenta que mientras más cantidad de agua existe en el material, este dispone de una menor resistencia eléctrica y por ende una mayor intensidad de corriente viajando a través de la madera.

La temperatura es otro factor importante en la medición de humedad en la madera, debido a que la resistencia eléctrica tiende a reducirse conforme

aumenta la temperatura de la madera. Por otro lado, el Xilohigrómetro es una herramienta que tiene como objetivo medir la resistencia eléctrica, a través de la ubicación de dos electrodos en la superficie de la madera. Para finalizar esta técnica se puede hacer tantas veces como el artesano desea sin embargo dicho material mientras más mediciones se realice va a tener más perforaciones en su estructura.

2) Constante Eléctrica: Este método se basa en la constante dieléctrica que está presente en la madera, es decir, hace referencia a la habilidad de la madera en absorber y contener energía, dependiendo directamente de la densidad que dispone la madera. Durante este método se debe utilizar instrumentos que se sitúen en la madera de forma externa, posteriormente se debe comparar la constante dieléctrica en superficies de madera totalmente seca, que tiende a ser baja con la del objeto de medición, sin embargo, este método tiene una precisión algo menor si es comparada con el método de resistencia eléctrica.

Cabe destacar que de los métodos antes mencionados también existen: radiación infrarroja, microondas y por destilación. Sin embargo, ninguno de estos métodos son utilizados en la industria de la madera. Para lo cual se selecciona a los métodos de inspección visual, método gravimétrico y el método eléctrico en esta investigación. Cuya justificación se presenta de manera resumida como se observa en la tabla 8.

Tabla 8*Métodos de diagnóstico de contenido de humedad en la madera*

	Método de Inspección Visual	Método del gravímetro	del Método Eléctrico
Concepto	El método de inspección visual tiene como principio el empirismo de la persona, siendo la técnica más sencilla de todas y debe ser el primer paso en la evaluación de cualquier estructura de madera.	El método del gravímetro, consiste en el proceso de desecación de la madera a través del uso de hornos industriales. Se dice que es el más preciso para el diagnóstico de contenido de humedad en la madera, ya que se utiliza cálculos matemáticos para evaluar la cantidad de agua en una pieza de madera.	Método que consiste en la medición del contenido de humedad utilizando dispositivos que hacen uso de la resistencia de la madera al paso de corriente eléctrica, aquello provoca que las mediciones de humedad sean inmediatas y de forma destructiva.
Derivación de técnicas por Método	Ninguna	Ninguna	<p>Resistencia eléctrica: es una técnica que se basa en la resistencia eléctrica que está presente en la madera.</p> <hr/> <p>Constante dieléctrica: es una técnica que se basa en su habilidad que tiene la madera para absorber y almacenar energía.</p>
Aparición	Siglo XIX	Principios del siglo XX	Principios del siglo XX

	Método de Inspección Visual	Método gravímetro	del Método Eléctrico
Formas de conocimiento	Experiencia	Científico	Científico
Formas de medición	Apreciación visual	Fórmula matemática	Instrumentos Eléctricos

Nota. Esta tabla muestra los métodos de medición de humedad en la madera.

2.4. Madera.

2.4.1. Introducción.

La madera es un material que se encuentra comúnmente en la naturaleza, contiene propiedades de flexibilidad y dureza, lo que le permite ser considerado un excelente material predilecto para la construcción. Durante todos los años crecen mediante un sistema de capas con forma circular (Uriarte, Julia Máxima 2020).

Esta especie es de origen vegetal al ser un material biológico, por lo que su estructura tiene como objetivo transportar el agua, proteínas y demás sustancias a través de sus fibras desde las raíces hasta punta más alta. Además, ha contribuido notablemente en la vida cotidiana de la humanidad, formando parte de la constante evolución (Aguilar Pozzer y Guzowski 2017).

2.4.2. Características de la madera.

En términos generales la madera presenta una serie de características físicas y químicas que le permite ser considerada como el material predilecto en el sector industrial (Queipo et al. 2013). Entre ellas cabe destacar:

- No se requiere de maquinaria sofisticada para transformar este material.
- Es un recurso de bien renovable, reciclable, y biodegradable.
- Su estructura es ligera, resistente y fácilmente transportable.

- Es muy utilizada en la fabricación de muebles debido a su fácil maleabilidad.
- Presenta una respuesta predecible ante situaciones de fuego.
- Resulta un buen aislante para el calor y en términos eléctricos.

2.4.3. Componentes Físicos de la madera.

Una de las características más importantes que posee la madera en su estructura son las propiedades físicas las mismas que la hacen ser considerada como materia prima para la construcción de muebles de exteriores, interiores así como en la ebanistería (Arquitectes 2018). A continuación, se describen sus componentes físicos más importantes:

- **El Color:** Es la pigmentación que se encuentra en la madera, como resultado de la presencia de sustancias que dan color a las fibras, dicha coloración permite el poder clasificarlas y diferenciarlas entre varias especies madereras, además se dice que mientras más oscuro es el color de la madera este tiende a ser más duradero y resistente.
- **Olor:** Es producido por las sustancias como resinas, aceites naturales, en muchos casos estos contienen olores algo peculiares.
- **Textura:** Es la relación que existe con las fibras en la madera, y estas dependen del tamaño, forma, y color del árbol. Por tal razón se dice que cada árbol tiene una estructura única.
- **Orientación de las Fibras:** Comprende al comportamiento estructural de las fibras de acuerdo a la dirección que siguen, los caminos leñosos longitudinales y posee una gran importancia en la manejabilidad de la madera.
- **La corteza:** Representa la capa exterior del tronco de madera, y esta se encuentra formada por células muertas, además de ser impermeable y muy resistente. El objetivo principal de la corteza es poder defender al árbol de agentes externos que puedan dañar su estructura.

- El Cámbium: Corresponde a la formación de capas (xilema, floema) en donde la xilema se encarga de crecer hacia el interior del tronco es también conocido como anillos de crecimiento y estos se generan en cada ciclo vegetativo de la especie, mientras que el floema se encarga de crecer hacia el exterior generando la corteza del árbol.
- Albura: Es la zona más joven de los anillos de crecimiento del tronco, por esta circulan las sustancias como la sabia, nutrientes a través del tronco por todo el árbol.
- Duramen: Corresponde a la zona con más anillos es decir la zona más antigua y céntrica del tronco, esta tiende a ser la más resistente, por lo tanto, las sustancias ya no pueden circular en esta área del tronco.
- La médula: Es la zona cercana al duramen y esta es el residuo del tallo una vez cortado el tronco donde en algún momento fue el núcleo de dicho árbol.
- Anillos de crecimiento: Comprende a las envolturas del tronco y estas se forman a través del tiempo, teniendo una forma cilíndrica además de ser cada vez más oscura conforme pasan los años.

2.4.4. Componentes Químicos de la madera.

La madera está compuesta de distintos elementos químicos los mismos que se encuentran presentes en toda la estructura física de la madera estos son: el carbono, hidrogeno, oxigeno, así como el nitrógeno. La mezcla de estos componentes dan origen a la creación de elementos como la lignina, celulosa y los taninos (Paz Fong 2008). A continuación, se describen los elementos químicos más importantes:

- Lignina: Son un grupo de compuestos químicos que se encargan de la generación de paredes celulares, presentes en las plantas para producir madera.

- Celulosa: Es un compuesto químico, rígido, insoluble que se encarga de la generación de células de glucosa.
- Taninos: Corresponde a un compuesto químico, que al tomar contacto con el oxígeno presenta alteraciones en su estructura química, puede ser utilizado como combustible a una temperatura superior de 199 °C.

2.4.5. Enfermedades en la madera.

Esta especie como cualquier organismo vivo se encuentra expuesta a muchas enfermedades o patologías en el medio ambiente, a causa de componentes externos y obviamente con el deterioro natural de la madera a través del paso de los años (Todo en Madera 2016).

La madera puede presentar en su estructura física ciertas características visibles, que da una idea de la existencia de al menos una o muchas enfermedades, lo ideal es poder realizar un diagnóstico temprano de la misma, con el fin de no perder dicho material a causa de hongos, podredumbres, así como parásitos (Todo en Madera 2016). A continuación, se explica brevemente las 5 enfermedades más importantes y posibles de presentarse en la madera:

2.4.5.1. La Pudrición.

Es la enfermedad más común que ataca a la madera y además de ser la más complicada en poder ser curada (Todo en Madera 2016). Esto se da debido a que un hongo infecta a la madera generando pudrición, lo que al pasar del tiempo conlleva consecuencias como la desintegración de la madera cabe mencionar que existen dos tipos de pudrición 1) Pudrición Blanca y 2) Pudrición azul a continuación se describen dichos tipos de pudrición:

1) Pudrición Blanca. - Aparece comúnmente cuando la madera se encuentra expuesta a la humedad y por lo tanto brota en una forma de masa de color blanca.

2) Pudrición Azul. – Se produce muy seguido en especies como el pino, sin embargo, el tipo de color dependerá de cada especie de árboles.

2.4.5.2. El Moho.

Se encuentra comúnmente en la madera, tiene como origen la introducción de un hongo en la corteza del tronco de madera, además de que la exposición de la madera en ambientes húmedos hace que dicho material quede totalmente inutilizable en un periodo de tiempo muy corto, el Moho es muy fácil de detectar y no se requiere de dispositivos avanzados para hacer dicho diagnóstico (Todo en Madera 2016).

2.4.5.3. Las Bacterias

Son organismos unicelulares muy pequeños, que han sobrevivido a lo largo de la evolución de la vida y es por ello que es normal observar que la madera puede ser atacada por bacterias, normalmente esto se da en ambientes donde la temperatura es húmeda haciendo que la corteza del tronco del árbol se vuelva cada vez más frágil. Los artesanos que trabajan con madera y esta se encuentra envuelta de bacterias, es necesario iniciar un proceso de desinfección, sin embargo, dicho proceso resulta ser sumamente lento y en caso de no tratar dicha madera contaminada esta se dañara hasta el punto de no servir para nada y ser desechada posteriormente (Todo en Madera 2016).

Las consecuencias de no tratar la madera a tiempo hacen que dichas bacterias causen daños irreversibles, normalmente dichas bacterias atacan químicamente haciendo que otros organismos puedan ser un problema muy importante para la madera, es por ello que se recomienda realizar revisiones y diagnósticos de forma periódica en aserraderos para no perder la materia prima a causa de las bacterias (Todo en Madera 2016).

2.4.5.4. La Termita.

Son animales de tamaño muy reducido, sin embargo, producen grandes daños en la estructura de la madera, en el Ecuador no genera problemas de gran impacto sin embargo en el exterior puede llegar a ser un problema muy serio a considerar, debido a que estos animales se comen la madera vieja haciéndola más vulnerable. Por tal motivo la madera puede tener más enfermedades (Todo en Madera 2016).

2.4.5.5. La Carcoma o Polilla.

Según (Todo en Madera 2016), alega que la carcoma es el producto de la ubicación de huevos de insectos en la estructura de la madera, el bicho de la carcoma construye una especie de cueva adentro de la estructura maderera por tal motivo es difícil conocer cuando ya están los huevecillos depositados, sin embargo los bichos producen sonidos muy peculiares que pueden servir de alerta para poder tomar medidas como por ejemplo inyectar sustancias ácidas en la estructura para matar dicha enfermedad producida por estos bichos.

2.4.6. Tipos de la madera.

Esta especie presenta características físicas que le permite ser considerado como la materia prima predilecta para la construcción de muebles. Además existen dos tipos de maderas que son muy utilizadas en la industria: 1) Madera Serrada y 2) Madera Maciza (Sánchez 2020). A continuación, se describe dichos tipos de madera.

1) Madera aserrada. – Es utilizada para elaborar materiales de construcción como vigas, columnas y son presentados en forma de tablón, cuartón y tabla. El laurel, es muy utilizado para la elaboración de muebles, artesanías, así como encofrados que son moldes que se emplean para ser rellenos por hormigón (Modenese s. f.).

2) Madera maciza. - Es aquella que procede directamente desde el tronco del árbol como por ejemplo el Roble, Cerezo, Nogal. Es por ello que dicha madera posee un nivel de calidad superior si se la compara con la madera aserrada (Modenese s. f.).

Para un mejor entendimiento la madera posee una serie de tipologías las mismas que son útiles cuando se desea trabajar en la fabricación de muebles o construcción (Modenese s. f.). A continuación, se describe las tipologías más utilizadas en el ámbito maderero:

- Viga y Columna: Es la madera maciza, con figuras rectangulares o circulares, que es muy utilizada como estructura en el ámbito de la construcción debido a sus fortalezas físicas.
- Tabla: La tabla tiene usos más enfocados a la decoración y construcción de encofrados, normalmente posee una dimensión de 2cm grosor x 22cm de ancho x 240cm de largo.
- Tablón: Es un tipo de madera de medidas 4cm de grosor x 25cm de ancho x 240cm de largo y tienen utilidades similares a las de la tabla.
- Tablero: Son un conjunto de tablas, pegadas sobre dos cuarterones en forma paralela, con una dimensión de 40cm alto x 80 cm de largo.
- Cuartón: Es un tipo de madera proveniente del Eucalipto, con dimensiones de 6cm de grosor x 4cm de ancho x 240cm de largo y son muy utilizados para crear tableros.
- Listón: Es un producto de madera proveniente del Eucalipto, con una dimensión de 2cm de grosor x 2cm de ancho x 250cm de largo comúnmente utilizados en la alineación y sujeción de tejados de techos.
- Pingo: Es una madera circular con un diámetro aproximado de 9cm o 14cm y es usada como soporte temporal de estructuras de construcción.

2.4.7. Humedad en la Madera.

La humedad es la medida utilizada para identificar la cantidad de agua que posee un elemento. Además de ser un componente natural que existe en la atmósfera y tiene origen entre la relación del vapor del agua existente en el aire. Se dice que mientras mayor cantidad de vapor de agua existe en el ambiente también aumenta la cantidad humedad (Airthings 2021).

Un factor climatológico es la humedad, el cual se define como el vapor de agua que es contenido en la atmósfera. Se conoce que en el planeta Tierra está conformado por dos terceras partes de agua provenientes de océanos, ríos y lagos. En donde la evaporación del agua genera nubes y estas juegan un papel muy importante en la generación de humedad (Conceptodefinicion.de 2021).

Existen una infinidad de conceptos referentes a la humedad en la madera a continuación se detallan los más trascendentales.

Se conoce que la humedad en una pieza de madera aparece cuando en sus propiedades físicas existe partículas de agua. Esto ocurre debido a que la madera es un material higroscópico lo que indica que puede absorber agua a través de las raíces (PuulInfo 2018).

La humedad aparece cuando un árbol es cortado. Este posee niveles altos de agua en su estructura y depende de factores como: la región de procedencia, la especie forestal, así como la época del año para poder determinar el nivel de contenido de humedad. Además, existen dos tipos de madera las cuales son clasificadas de acuerdo a la presencia de humedad: 1) Madera liviana, 2) Madera pesada. A continuación, se describen brevemente:

1) Las maderas livianas. - son aquellas que mayormente contienen niveles altos de humedad debido a la presencia de porosidad en su estructura (Gamez Penadillo 2009).

2) Las maderas pesadas. – corresponden a la materia prima que posee menor cantidad de agua con respecto a las maderas livianas debido a que en sus propiedades físicas carecen de porosidad (Gamez Penadillo 2009).

En la Figura 4, se muestra el proceso de secado tradicional de diferentes especies madereras las cuales deben ser ubicadas en parrillas por un periodo de tiempo de 10 a 12 meses. De esta forma se elimina el nivel de humedad excedente. Es por ello que se utilizan torres para ubicar dicho material y este vaya perdiendo el nivel de humedad progresivamente con el pasar del tiempo (Modenese s. f.).

Figura 4

Secado de Madera Aserrada



Nota: En esta Figura se muestra el proceso de secado de madera aserrada en parrillas, fotografía tomada en el aserradero “San Martín”.

El contenido de humedad presente en la madera puede ser calculado de forma matemática utilizando a la masa del agua y la masa anhidrica respectivamente.

2.4.7.1. Cálculo de la humedad en la madera.

La humedad presente en la madera se puede calcular mediante el uso de la Ecuación 1, considerando que el contenido de humedad es la división entre la masa del agua y la masa en un estado anhidro (Hidalgo Broncano 2013).

Ecuación 1

Humedad en la madera

$$CH = \frac{\text{masa agua}}{\text{masa anhidra}}$$

Nota: En esta ecuación se presenta la fórmula para el cálculo de contenido de humedad en la madera.

En donde, la masa del agua corresponde a la masa inicial de la madera menos la masa anhidra tal cual se puede observar en la Ecuación 2.

Ecuación 2

Porcentaje de Humedad

$$CH = \frac{PG - PO}{PO} \times 100\%$$

Nota: En esta ecuación se presenta la fórmula correspondiente para el cálculo del porcentaje de contenido de humedad en la madera.

- CH: Es la cantidad de humedad expresado en porcentaje de una muestra de madera.
- PG: Es la masa inicial que tiene una muestra de madera y esta es expresada en gramos (gr).
- PO: Es la masa anhidra de una muestra de madera y es expresada en gramos (gr).

De acuerdo con la ecuación 2, se pudo visualizar que la cantidad de humedad de la madera es expresa en porcentaje, Razón por la cual resulta

necesario dividir la diferencia de PG – PO para el PO, y este valor multiplicarlo por un 100%.

La madera en su estructura puede estar compuesto por tres diferentes tipos de agua que son: 1) libre, 2) higroscópica y 3) de constitución.

1) Libre. - Aquella que se encuentra en las cavidades celulares de la madera, la cantidad de agua que esta posee depende de la porosidad que tenga dicha madera, además el proceso de secado resulta muy fácil debido a que el agua sale por los poros mediante la evaporación del agua. En este punto la madera se encontrará en un punto de saturación de fibras, obteniéndose un nivel de humedad alrededor de 28% - 30% (Hidalgo Broncano 2013).

2) Higroscópica.- Aquella que se ubica en las paredes celulares de la estructura física del árbol, también es conocida como agua retenida y es transportada por las fibras del árbol, cuando el agua libre es eliminada de las paredes celulares la madera tiende a sufrir alteraciones en sus dimensiones (Hidalgo Broncano 2013).

3) De constitución. - Aquella que se presenta en la materia celular de la madera y que no puede ser removida por ningún método tradicional de secado. En caso de requerir eliminar dicha agua se tiene como resultado la destrucción parcial en la estructura de la madera (Hidalgo Broncano 2013).

2.4.8. *Uso sugerido de la madera de acuerdo al nivel de su humedad.*

El nivel de humedad que tiene la madera en su estructura es un factor muy importante en el proceso de selección de materia prima. Es por ello que resulta recomendable usar una tabla como herramienta de referencia para una correcta selección de madera en función al nivel de humedad como se puede muestra en la Tabla 9 (Bermejo, Félix 2014).

Tabla 9*Usos recomendados de la madera en función al nivel de humedad*

Usos de la Madera	Porcentaje de humedad requerida
Madera utilizada para la fabricación de muebles para interiores	6% - 10%
Madera utilizada para la construcción de muebles de exteriores, los cuales no van a estar exhibidos a la intemperie.	10% - 14%
Madera utilizada para la construcción de muebles para exteriores los cuales si van a estar exhibidos a la intemperie.	12% - 17%
Madera utilizada para la elaboración de pisos.	8% - 10%
Madera utilizada para la elaboración de estructuras para cocinas	8% - 10%

Nota. En esta tabla se muestra los usos recomendados de la madera en función al nivel de humedad que esta contiene. Tomado de (Bermejo, Félix 2014).

2.5. Laurel.

2.5.1. Introducción.

El Laurel pertenece a la familia "Boraginaceae" y es muy popular debido a que su madera posee altos niveles de calidad, fácil secado, una buena dureza además de tener un rápido crecimiento siendo muy utilizado en la industria maderera. La demanda de esta especie es muy alta en el Ecuador. (Ecuador Forestal 2012).

Además, posee las siguientes características y dimensiones: 40m de alto, 20m de fuste maderable y un diámetro aproximado de 1m, crece en temperaturas aproximadas entre 20°C y 25 °C. La copa del laurel proyecta una sombra circular con tendencia a ser cilíndrica. Sus hojas son ásperas de color

verdoso y opacas por la parte superior. Posee flores de color blanco con un olor y su tamaño varía desde 5cm a 30cm (Ecuador Forestal 2012).

Esta especie crece en países como México, Bolivia, Brasil, Perú y por supuesto en el Ecuador en la región Costa y Amazonia. Una de las propiedades más interesantes del Laurel es que tiene la capacidad de auto podarse dependiendo de la estación climatológica que este se encuentre. Otra característica muy importante a destacar es que gracias al peso y forma de las semillas el Laurel puede plantarse de forma muy abundante (Ecuador Forestal 2012).

2.5.2. *Propiedades.*

El laurel fue muy utilizado desde tiempos muy antiguos exactamente en el Imperio Romano y Griego lo utilizaron como símbolo de status social y poder. Sin embargo, en ese entonces ya se conocía de las bondades que este ofrecía para la población como ser repelente natural de insectos.(Mannise 2020).

Posee muchas utilidades, entre ellas en el sector gastronómico se las utilizaban como condimento debido a su alto potencial digestivo en recetas como: salsas, guisos, especias para las legumbres siendo actualmente muy apetecido en países como México y el mercado Europeo (Mannise 2020).

En el sector médico, las hojas aportan de vitaminas A, C y varias del grupo B como la vitamina B2 o también llamada Riboflavina, Además es una fuente importante de calcio y hierro muy recomendada para el consumo de personas adultas, también sirve como producto diurético, estimula el apetito y ayuda a aliviar los cólicos, el dolor de estómago y la flatulencia (Mannise 2020).

Cabe destacar que es un material de suma importancia para el sector de la industria la maderera. Además es muy utilizado debido a su fácil crecimiento,

alta maleabilidad, buena resistencia, muy ligera y su velocidad de secado convirtiéndolo en un material muy requerido en la construcción de muebles (Mannise 2020).

2.5.3. Beneficios y Usos.

El laurel es utilizado de múltiples formas, debido a que posee propiedades físicas y químicas. A continuación se describen sus principales usos (Mannise 2020):

- **Uso medicinal:** Posee efectos bactericidas por lo tanto su uso es recomendable en personas con enfermedades como: gripe, bronquitis, faringitis. Además de ser un buen estimulante en caso de fatiga y reducción de estrés. Finalmente acelera la sanación de heridas ya que el aceite de Laurel genera propiedades antibacterianas.
- **Gastronomía:** Diurético por excelencia y la eliminación de líquidos del cuerpo humano.
- **Materia Prima:** La madera es muy utilizada en la fabricación de muebles decorativos, incluso en la construcción de casas.
- **Construcción:** Se utiliza para la fabricación de tirantes, vigas, ventanas, puertas, estructuras.
- **Producción de muebles:** Usado para fabricar muebles como roperos, closets, silla, cómodas, casas para animales de crianza.
- **Esculturas:** Obras de arte se han realizado utilizando la madera del laurel a través del tiempo.
- **Biocombustible:** El laurel posee propiedades en su estructura que le permite ser considerado un biocombustible.
- **Papelería:** Se utiliza para la fabricación de productos como cuadernos, papel, carpetas, etc.

- Calefacción: Por las propiedades físicas de secado rápido de la madera puede ser utilizada como leña.

2.6. Software.

2.6.1. Introducción.

El concepto de software posee distintas interpretaciones por ser algo intangible y susceptible de cambios. Razón por la cual a continuación se presentan las definiciones más relevantes.

El software es aquella parte intangible y lógica de la informática, que forma parte de muchos dispositivos como celulares, tabletas, portátiles y hacen que el hardware funcione de alguna determinada. Además, está compuesto de programas, aplicaciones y algoritmos para realizar distintas acciones (Raffino 2021).

El termino software proviene del inglés, y hace referencia al soporte lógico de un sistema informático. Por lo tanto es la parte no física que tiene relación a uno o muchos conjuntos de programas y que manejan una cifra importante de datos, reglas e instrucciones para comunicarse con el hardware (CISSET 2019).

El software es el conjunto de instrucciones, que se incorporan en un sistema informático para realizar determinadas acciones. Además es totalmente necesario para que un sistema pueda funcionar, ya que es el que emite ordenes e indica que es lo que toca hacer cada componente (Importancia.org 2013).

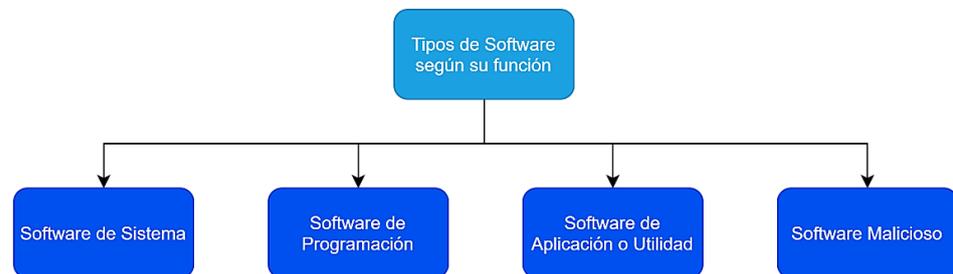
2.6.2. Tipos de Software.

En la actualidad existen muchas formas de clasificar al software. Utilizándose para una un enfoque diferente, su manera de clasificar más común es de acuerdo a su funcionalidad. En la figura 5 se puede visualizar este tipo de

clasificación: 1) Software de sistema, 2) Software de Programación, 3) Software de aplicación y 4) Software Malicioso (TecnoMagazine 2018).

Figura 5

Clasificación de los tipos de Software según su función



Nota. La figura 3 representa la clasificación de los tipos de software según su función. Tomado de (TecnoMagazine 2018).

A continuación se describe de manera breve la clasificación del software según su funcionalidad (TecnoMagazine 2018):

1) Software de Sistema. - Es aquel que permite que el hardware funcione y está incorporado de forma interna en un dispositivo. Varios ejemplo claros son: Microsoft Windows, Gnu/Linux, Mac OS, Android, IOS, drivers de sistema, servidores de Datos, etc. (TecnoMagazine 2018).

2) Software de Programación. - Corresponde a las herramientas suministradas hacia desarrolladores o programadores para la creación de nuevas aplicaciones. Algunos ejemplos de este tipo son los siguientes: editores de texto, compiladores, entornos de desarrollo integrado (IDE) entre otros (TecnoMagazine 2018).

3) Software de Aplicación. - También conocido como software de utilidad, son aquellas aplicaciones, programas, así como herramientas que el usuario común utiliza a diario para realizar alguna acción determinada. Las aplicaciones más conocidas las siguientes: Microsoft Office Word, Excel, Skype, Software

contable, medico, administrativo o a su vez implementaciones como Facebook, Instagram, etc. (TecnoMagazine 2018).

4) Software Malicioso. – Denominado también “Malware” software que tiene como objetivo hacer daño a personas, empresas, gobiernos, etc. Muchos de estos malware o virus no son fáciles de detectar por un usuario común, y requieren de conocimientos técnicos en seguridad informática para poder mitigarlos y eliminarlos (TecnoMagazine 2018).

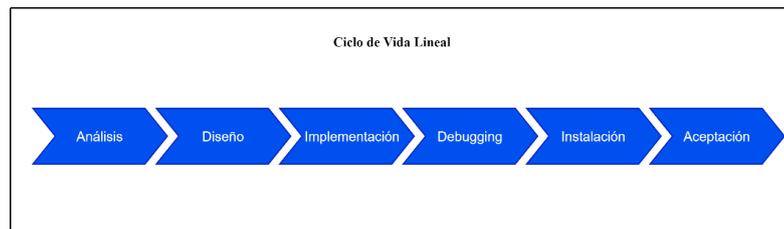
2.6.3. Ciclo de vida del software.

Es un proceso metodológico que facilita la organización de un proyecto software partes pequeñas o fases. Su principal objetivo es mejorar la calidad del mismo, así como la productividad de quienes están involucrados en este. Cabe mencionar que no existe una metodología estándar, esta depende del tipo de proyecto que se maneje (Cantone 2006). A continuación, se presenta una breve descripción de los principales ciclos de vida utilizados en la industria del desarrollo de Software:

1) Lineal. - En la Figura 6, se puede observar este ciclo de vida, el cual consta de las siguientes 6 etapas: 1) análisis, 2) diseño, 3) implementación, 4) debugging, 5) instalación y 6) aceptación. Su principal característica es que sus etapas son secuenciales, es decir cada vez que se finaliza un proceso se continua con el siguiente. Facilitándose la estimación de tiempo y esfuerzo. Su principal desventaja es que al momento de existir un fallo en las etapas iniciales este fallo repercute en las otras etapas lo cual se ve reflejado en un mayor costo al finalizar el proyecto (Cantone 2006).

Figura 6

Ciclo de vida Lineal

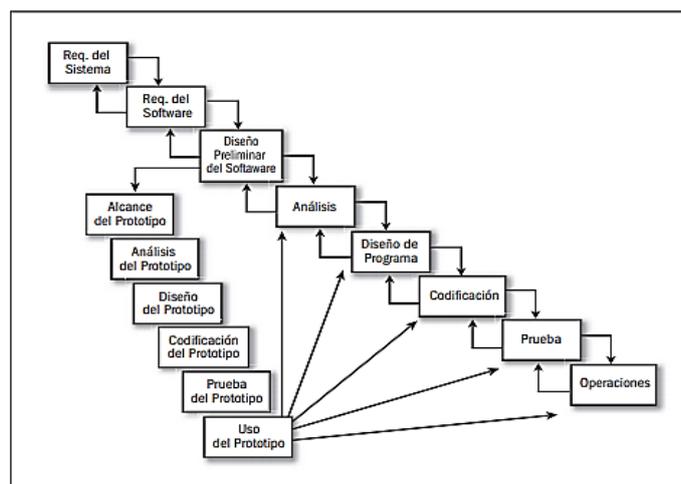


Nota. En esta figura se muestra el proceso del ciclo de vida lineal. Tomado de (Cantone 2006:22).

2) Cascada. - En la figura 7, se puede visualizar este ciclo de vida. El cual fue creado en el año 1970 por Winston Royce. Su principal característica es permitir tener iteraciones entre las fases o procesos, es decir se requiere un análisis exhaustivo por cada una de sus fases antes de proceder con el siguiente proceso. Su principal desventaja es ser un poco flexible (Cantone 2006).

Figura 7

Ciclo de vida en Cascada



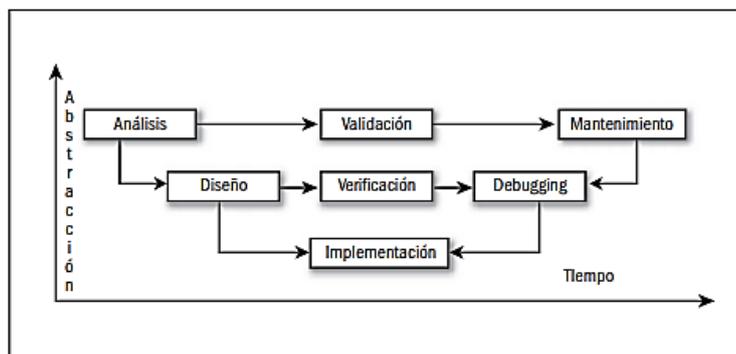
Nota. Esta figura muestra las fases del ciclo de vida en Cascada. Tomado de (Cantone 2006:23).

3) V.- En la figura 8, se puede ver este ciclo de vida. El mismo que fue creado por Alan Davis. Está fundamentado en el ciclo de vida en cascada. Cabe

destacar que se añadieron dos sub fases de retroalimentación: 1) validación que se encuentra entre el análisis y el mantenimiento y 2) verificación que está ubicado entre el diseño y el debugging. La principal ventaja de este ciclo de vida radica en los beneficios de conocer muy bien los requisitos de software.(Cantone 2006).

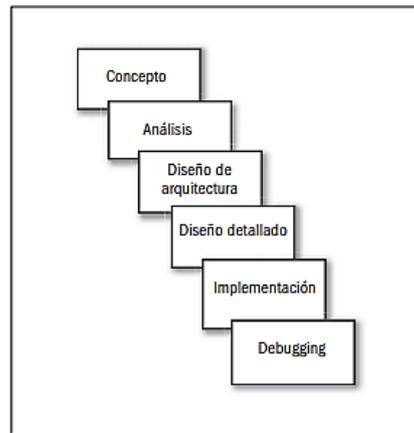
Figura 8

Ciclo de vida en V



Nota. En esta figura se presenta al ciclo de vida en V. Tomado de (Cantone 2006:24).

4) Sashimi. - En la figura 9, se puede observar este ciclo de vida. El cual está inspirado en cascada. Su principal característica es permitir sobreponer las fases de desarrollo, obteniéndose mayor eficiencia al existir retroalimentación temprana. Su principal desventaja es ser muy complejo al gestionar el tiempo entre fases (Cantone 2006).

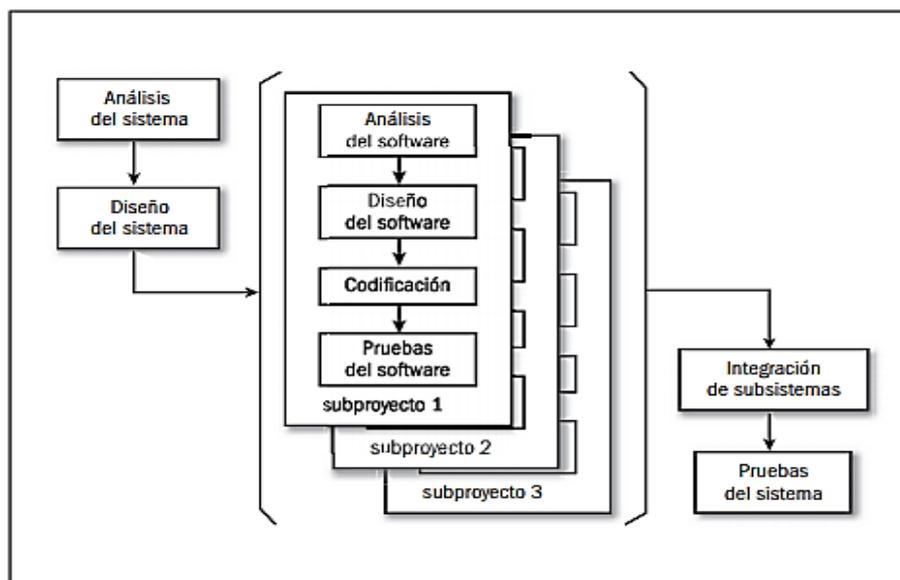
Figura 9*Ciclo de vida tipo Sashimi*

Nota. En esta figura se presenta al ciclo de vida de tipo Sashimi. Obtenido de (Cantone 2006:25).

5) Cascada con subproyectos. - En la figura 10, se puede observar este ciclo de vida. El cual está fundamentado en cascada. Cabe destacar que se divide las fases de desarrollo en sub etapas, para poder desarrollarlas de forma paralela. Su principal desventaja radica en la dependencia de tareas previas para poder continuar con otra (Cantone 2006).

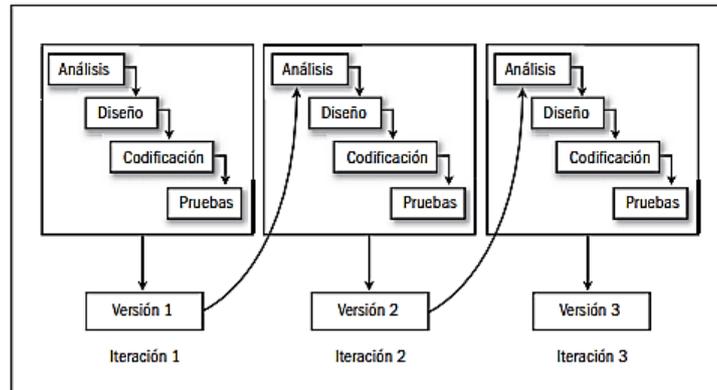
Figura 10

Ciclo de vida en Cascada con subproyectos



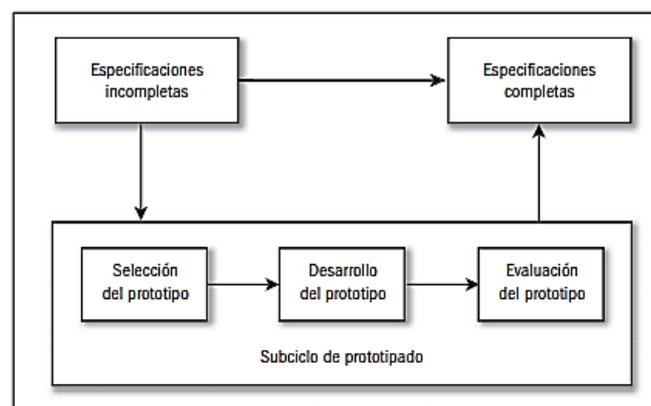
Nota. Esta figura muestra el ciclo de vida en cascada con prototipos. Tomado de (Cantone 2006:26).

6) Iterativo. - En la figura 11, se puede mostrar este ciclo de vida. El cual está inspirado en el de cascada. Su principal característica es mitigar los riesgos de fallos entre requerimientos y el producto final a través de iteraciones. Cabe destacar que al final de cada repetición el cliente obtiene un avance del producto. Su principal ventaja es eliminar fallos en cada entrega del software al cliente (Cantone 2006).

Figura 11*Ciclo de vida Iterativo*

Nota. Esta figura representa el ciclo de vida Iterativo. Obtenido de (Cantone 2006:27).

7) Prototipos. - En la figura 12, se puede visualizar este ciclo de vida. El cual basado en el ciclo de vida iterativo. Su principal característica es generar prototipos pequeños que validen los requerimientos del cliente en etapas tempranas del desarrollo de software. Cabe destacar que este ciclo de vida es recomendable implementarlo al desear utilizar una nueva tecnología o a su vez poco probada (Cantone 2006).

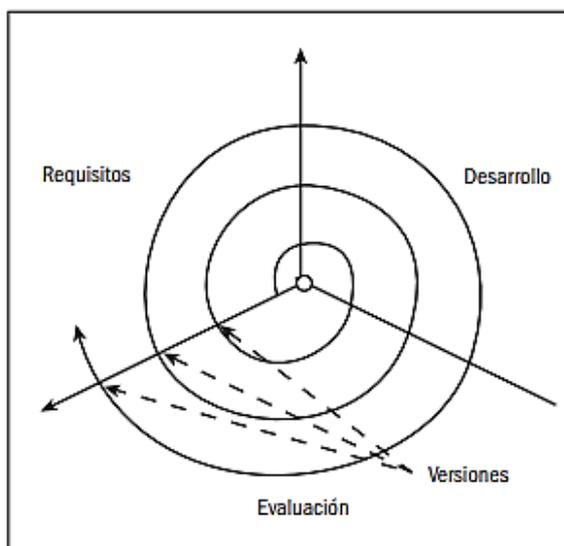
Figura 12*Ciclo de vida por prototipos*

Nota. Esta figura muestra el ciclo de vida por prototipos. Tomado de (Cantone 2006:28).

8) Evolutivo. - En la figura 13, se puede observar este ciclo de vida. El cual es utilizado cuando los requisitos de software no están totalmente definidos o son cambiantes. Su principal ventaja es la facilidad de implementación por parte del equipo de desarrollo de software (Cantone 2006).

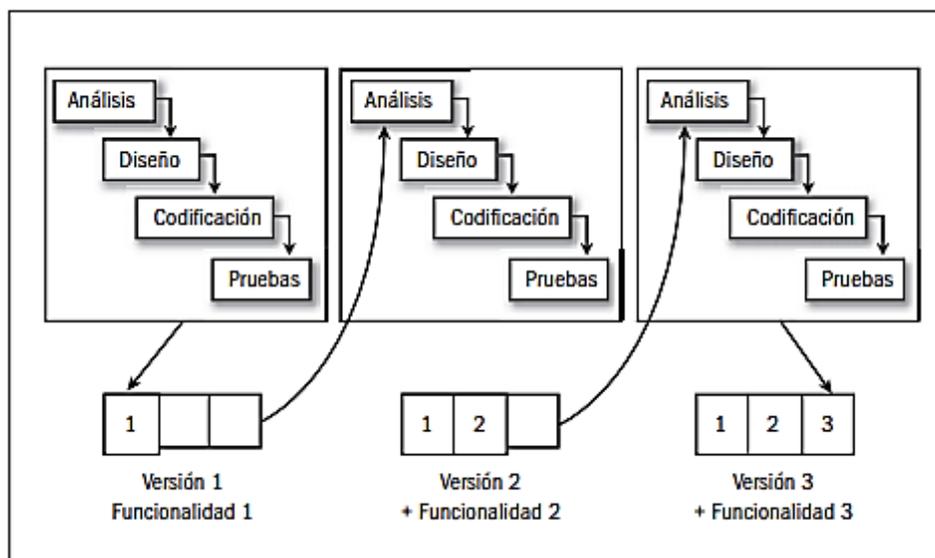
Figura 13

Ciclo de vida evolutivo



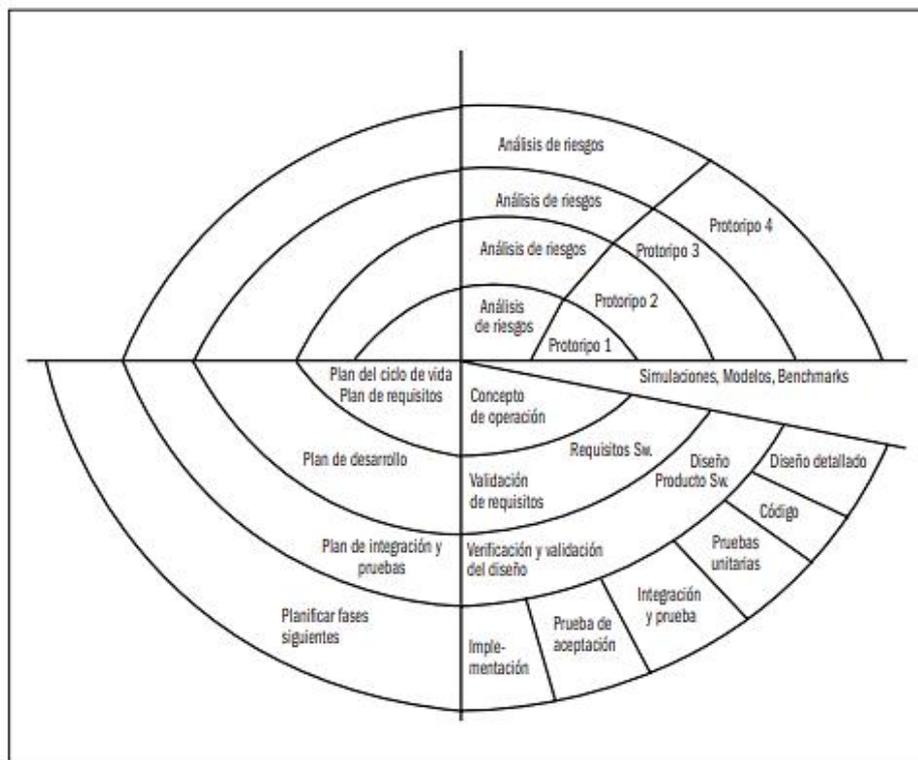
Nota. Esta figura muestra el ciclo de vida evolutivo. Tomado de (Cantone 2006:29).

9) Incremental. - En la figura 14, se puede mirar este ciclo de vida. El cual se fundamenta en la implementación periódica de funcionalidades a través de los requerimientos de software. Cabe destacar que esta técnica permite desarrollar funcionalidades de forma independiente para ser unidas posteriormente. Su principales ventajas son: reducir riesgos, y eliminar fallos de forma sencilla (Cantone 2006).

Figura 14*Ciclo de vida incremental*

Nota. Esta figura muestra el ciclo de vida incremental. Tomado de (Cantone 2006:30).

10) Espiral. - En la figura 15, se puede visualizar este ciclo de vida. El cual está inspirado en el ciclo de vida de prototipos y fue creado en el año de 1988 por Boehm. Su principal característica es permitir generar ciclos en las etapas de desarrollo de software, es decir se crea una serie de repeticiones hasta obtener un producto de calidad. Cabe destacar que es ciclo está compuesto de 4 actividades: 1) planificación, 2) análisis de riesgo, 3) implementación y 4) evaluación. Su principal desventaja es el consumo excesivo de recursos por cada interacción del espiral (Cantone 2006).

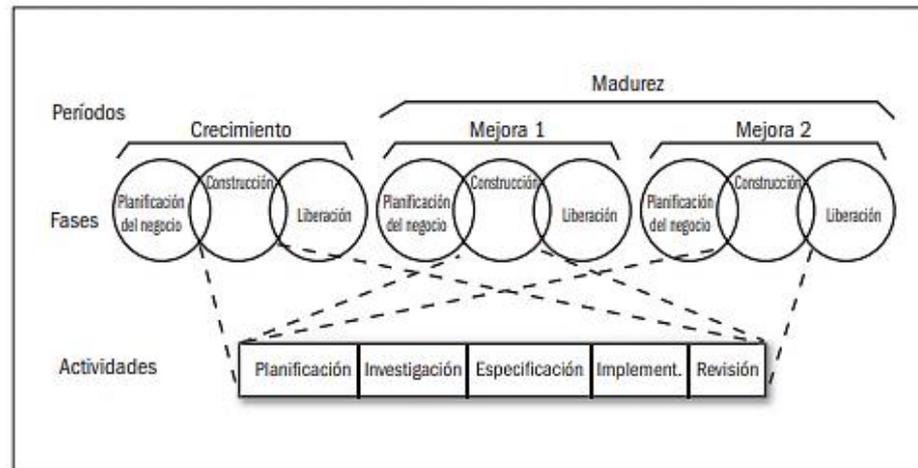
Figura 15*Ciclo de vida en espiral*

Nota. Esta figura representa el ciclo de vida en espiral. Tomado de (Cantone 2006:32).

11) Orientado a objetos. - En la figura 16, se puede mirar este ciclo de vida. El cual fue creado en los años 90 debido a la aparición del paradigma orientado a objetos. Cabe destacar que cada objeto posee características y propiedades que son considerados como atributos y el comportamiento de dichos objetos son llamados métodos. Su principal característica es la fácil abstracción de requerimientos por parte del cliente. (Cantone 2006).

Figura 16

Ciclo de vida orientado a objetos



Nota. Esta imagen muestra el ciclo de vida orientado a objetos. Obtenido de (Cantone 2006:34).

2.6.4. Arquitectura de Software.

El término arquitectura de software es la manera en la que un equipo de desarrollo de software estructura un proyecto. Sus dos principales objetivos son: 1) mejorar el rendimiento y 2) satisfacer las necesidades de los clientes a través del uso de estándares de calidad (Cervantes 2020). Área muy importante dentro de la informática ya que esta permite una correcta estructuración de componentes de un producto, minimizando sus dificultades y facilitando su mantenimiento (García 2020).

La selección de una eficaz arquitectura es independiente de la metodología de desarrollo de software que se maneje. Está compuesta por las siguientes etapas: 1) requerimientos, 2) diseño, 3) documentación y 4) evaluación. A continuación, se detallan brevemente estas etapas (Cervantes 2020):

1) Requerimientos. - Se enfoca en la toma de requisitos por parte del cliente. Así como la correcta priorización que estos tendrán dentro del desarrollo de software. Los requisitos funcionales y las restricciones permitirán garantizar los niveles de calidad de un producto software.

2) Diseño. – Esta orientada en la definición de la arquitectura de software. Se utiliza la experiencia del equipo de desarrollo para la selección de patrones de diseño que se adapten a la problemática a solucionar.

3) Documentación. – Esta dirigida a la elaboración de documentos formales. Los mismos que son utilizados por el equipo de desarrollo en cada una de las etapas del proyecto, requiriéndose una buena comunicación entre sus miembros lo que permitirá detectar algún inconveniente en el diseño de la arquitectura de software del proyecto propuesto.

4) Evaluación. – Esta encaminada a la elaboración de un plan para mitigar los riesgos en cada una de las etapas que conforman el diseño de la arquitectura de software. Lo que permite mitigar dichos riesgos a través de evaluaciones continuas.

2.6.4.1. Tipos de arquitectura de software.

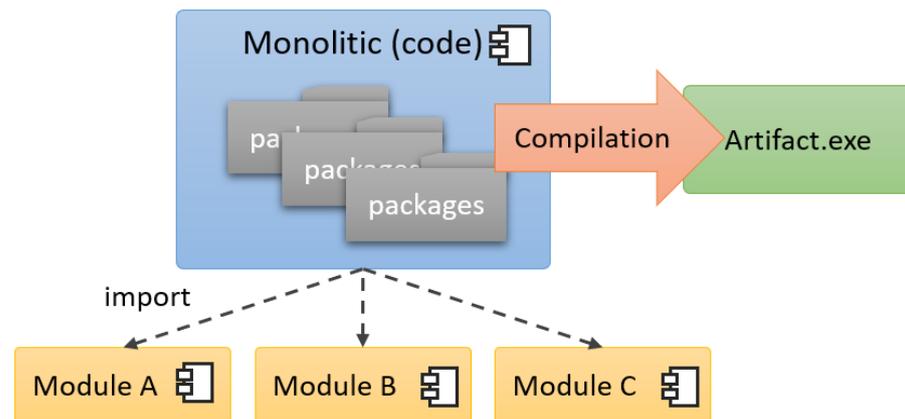
Los proyectos de software pueden requerir de arquitecturas diferentes. Razón por la cual es recomendable realizar un análisis previo de acuerdo a los requerimientos de software a desarrollarse. Los dos tipos de arquitecturas de software más importantes son: 1) Monolíticas y 2) Microservicios (GreenSQA 2019). A continuación, se describen de forma breve cada una de ellas.

1) Monolítica. - Aquella arquitectura en la que todos sus componentes se encuentran agrupados en un mismo producto, empaquetándose en una sola pieza. De manera que todas sus funcionalidades comparten los recursos y

memoria que un software necesite (Blancarte, 2020). En la Figura 17, se observa la estructura de una arquitectura monolítica. Este tipo de arquitectura es muy utilizada cuando la aplicación a desarrollar no es de gran tamaño sin embargo cada paquete o componente debe estar bien estructurado para tener un código fuente claro y conciso (Blancarte, 2020).

Figura 17

Arquitectura Monolítica



Nota. Esta figura muestra la estructura básica de una arquitectura monolítica. Tomado de (Blancarte 2020).

Las 3 principales ventajas de esta arquitectura son (Blancarte 2020):

- 1) Escalabilidad. - Característica que permite comprender el crecimiento (horizontal y vertical) de un producto software.
- 2) Rendimiento. - Funcionalidad que permite evaluar la velocidad de respuesta de un sistema software.
- 3) Fáciles de probar y detectar fallos. - Técnica que consiste en la revisión de las funcionalidades implementadas en un producto de acuerdo al documento de requerimientos de software.

Cabe destacar que este tipo de arquitectura presenta el inconveniente de que si el servidor de aplicaciones falla todo el aplicativo también fallará llegando

a ser incómodo para los usuarios que utilizan el sistema software (Blancarte 2020).

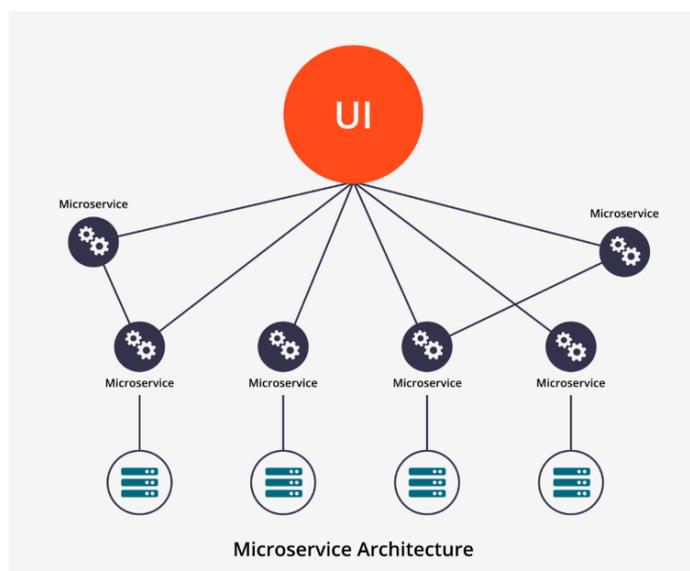
2) Microservicios. – Arquitectura en la que se aísla de forma independiente cada componente llamándose también servicio de un aplicativo software. Aplicando el concepto de bajo acoplamiento y alta cohesión (Chakray, 2018).

Dichos servicios se los realiza a través de API's lo que favorece al equipo de desarrollo al momento de trabajar en múltiples requerimientos de software de forma independiente (Ilimit, 2020). Lo que se ve reflejado en la productividad y rapidez con la que se desarrolla un producto software.

En la figura 18 se puede observar la arquitectura basada en microservicios. La cual está compuesta de servicios y bases de datos que trabajan de forma independiente (Malav, 2017).

Figura 18

Arquitectura de Microservicios



Nota. Esta figura representa la arquitectura básica basada en microservicios. Tomado de (Malav 2017).

Los 3 principales beneficios de esta arquitectura son:

1) Seguridad. - Característica que se ve reflejada al separar cada servicio del sistema de forma independiente. De tal manera que al ocurrir un fallo este solo afectara a un servicio en específico (Malav, 2017).

2) Escalabilidad. – Facilidad de implementación de nuevas funcionalidades en un producto software (Malav, 2017).

3) Flexibilidad. – Reducción de la complejidad de mantenimiento del software al incorporar más fuentes de información (Malav, 2017).

2.6.5. Calidad del Software.

La calidad del software se define como el nivel de satisfacción de un cliente a través del cumplimiento de requerimientos de desarrollo en un producto. Utilizando eficazmente los recursos, herramientas, metodologías, así como ciclos de vida haciendo que dicho producto sea lo más confiable posible (Callejas-Cuervo, Alarcón-Aldana, y Álvarez-Carreño 2016). Además, permite establecer la concordancia entre la especificación de requerimientos de un producto software y su rendimiento. Utilizando estándares de desarrollo para que el producto sea desarrollado profesionalmente (Pressman 2010).

Para que un producto software pueda ser considerado de calidad debe pasar una serie de criterios los mismos que pueden ser medidos una vez se acabe el desarrollo del aplicativo. Por lo que resulta ser un proceso muy costoso para una empresa en el caso de que se detectara un fallo en la etapa final (García León y Beltrán Benavides 1995).

Existen muchas formas de medir la calidad de un software entre ellas la más común es la medición de errores que tiene un sistema. En la actualidad

existen múltiples mecanismos para probar un aplicativo entre ellas están los test: unitarios, de integración y funcionales (García León y Beltrán Benavides 1995).

El aseguramiento de la calidad de un producto software debe ser responsabilidad de cada equipo de desarrollo. Es por ello que surge la necesidad de utilizar estándares de software para maximizar la calidad de un producto (García León y Beltrán Benavides 1995).

2.6.5.1. Estándares del Software.

Un estándar es un conjunto de normas, reglas y normativas que están documentadas en forma de requerimientos con el objetivo de determinar alguna acción. Representando un modelo que puede ser tomado como referencia (Pérez Porto y Merino 2017).

Además, forman parte en cada fase del desarrollo de software como son el análisis, diseño, implementación y pruebas. Esto permite que el producto final cuente con reglas que mejoren su eficiencia y garanticen la calidad del producto. A continuación, se lista una descripción breve de los principales estándares que se utiliza en el desarrollo de Software (Arciniega 2017).

1) ISO / IEC. - Las normas ISO son una recopilación de documentos que especifican una serie de requerimientos que son utilizados por empresas, y/o profesionales para garantizar que los productos o servicios que ofrezcan cumplan con su objetivo (Arciniega 2017).

A continuación, se detallan las principales normas ISO utilizadas en la industria del desarrollo de software:

a) ISO 12207.- Tiene por objetivo regular el proceso de adquisición de software ya sea para desarrolladores y/o proveedores. Dicho estándar

comprende 17 procesos agrupados en 3 categorías (principales, de apoyo y de organización) (Arciniega 2017).

b) ISO 9000.- Tiene como principal característica gestionar y asegurar la calidad de productos y servicios.

c) ISO 9001.- que se aplica en los sistemas de calidad (SGC), el cual se enfoca en los procesos de administración de calidad basándose en 8 principios: 1) El enfoque al cliente, 2) Liderazgo en la organización, 3) La participación del personal de la organización, 4) El enfoque apoyado en procesos, 5) El enfoque de sistema para la gestión en la organización, 6) Procesos de mejora continua, 7) El enfoque basado en hechos para la toma de decisiones en la organización y 8) Relaciones beneficiosas para el proveedor y la organización (Arciniega 2017).

2) SPICE. - Este es un estándar con apoyo internacional para normalizar la evaluación de procesos de software. Teniendo tres enfoques principales que son: 1) transferencia tecnológica, 2) ensayos en la industria tecnológica de forma emergente, y 3) preparación de evaluaciones de procesos de software (Arciniega 2017).

3) CMMI. - Es un modelo que provee de elementos necesarios para una mejora en el proceso de desarrollo de software utilizando un marco de referencia. Evaluándose la efectividad en cada uno de los procesos que tenga una organización. Este modelo establece seis modelos de madurez que son: 1) nivel 0 o incompleto, 2) nivel 1 o inicial, 3) nivel 2 o repetible, 4) nivel 3 o definido, 5) nivel 4 o administrado y 6) nivel 5 o también llamado optimizado (Arciniega 2017).

4) PSP. – Conocido como el proceso de personal del software consiste en un proceso de autorreconocimiento. El cual permite estimar el tiempo que tarda un individuo en desarrollar un aplicativo software. Facilitándose la

estimación en costo de un proyecto. Este modelo es aplicado para programas de pequeña escala y con menos de 1000 líneas de código (Arciniega 2017).

5) TSP. - Denominado como Team Software Process. Es un estándar que enfatiza el trabajo colaborativo en una organización. Debido a que considera que los equipos no se forman por arte de magia, no existe pasos recomendados para formar equipos, así como se debe comprender las habilidades, fortalezas y debilidades de las personas en un equipo, para organizarlos de la mejor manera (Arciniega 2017).

6) Moprosoft. - Es una normativa de origen Mexicana que se basa en los procesos de la industria del software, por lo tanto, se obtiene un alza de nivel administrativo y por ende se generan productos de calidad alcanzando competitividad entre organizaciones a nivel internacional (Arciniega 2017).

7) IEEE 830.- Es un estándar muy utilizado en la industria del desarrollo de software en la etapa de análisis. También llamado como "ERS". Además son un conjunto de recomendaciones para que el cliente y la organización puedan cumplir a totalidad con los acuerdos mutuos en un documento formal (Arciniega 2017).

2.6.5.2. Encuesta SUS.

Las encuestas son herramientas muy utilizadas en el campo de la investigación las cuales están compuestas de una serie de preguntas (abiertas o cerradas) con el objetivo de extraer cierta información en una determinada muestra de personas. Los datos obtenidos a través de la aplicación de esta herramienta suelen ser representados de forma gráfica a través de ilustraciones estadísticas (Westreicher 2021).

La Escala de Usabilidad de un Sistema (SUS) es una herramienta metodológica muy utilizada y fácil de aplicar al momento de medir el grado de usabilidad de un producto software. Además es muy utilizada para medir la experiencia de usuario con respecto a un aplicativo (Devin 2017).

Esta herramienta contiene 10 preguntas simples con una escala de calificación del 1 al 5. En donde 1 significa que el usuario está en total desacuerdo, mientras que 5 significa que está totalmente de acuerdo. Y puede ser aplicada a un número extenso de personas (Devin 2017).

A continuación, se presenta los pasos necesarios para aplicar la encuesta de usabilidad de un sistema SUS (Devin 2017):

1) Preparar la encuesta. – Esta herramienta consta de 10 preguntas simples con una escala de calificación del 1 al 5. En donde 1 significa que el usuario está en total desacuerdo, mientras que 5 significa que está totalmente de acuerdo.

2) Ejecutar. – La encuesta debe ser aplicada a un determinado número de personas, no existe límite recomendado para la ejecución de dicha herramienta.

3) Tabular los datos. – Los datos obtenidos de las 10 preguntas a través de la ejecución de la encuesta deben ser tabulados de forma numérica en el rango de 1 a 5.

4) Calcular el grado de usabilidad. - Existe una fórmula matemática para poder calcular el grado de usabilidad de un sistema. El cual realiza una sumatoria de los resultados obtenidos en las preguntas impares (1,3,5,7 y 9) y se le resta 1. Para las preguntas pares (2,4,6,8 y 10) se debe restar 5 menos el valor asignado por el usuario. Finalmente, al realizar dicha sumatoria se debe multiplicar dicho valor por 2.5.

5) Comparación de resultados.- Si al realizar la operación en el paso 4 el test de usabilidad presenta un valor inferior a 50 se considera que el producto software es fallido, si el valor está por encima de 68 se comprende que el sistema esta correcto y si supera un valor de 80 es un sistema sobresaliente (Devin 2017).

2.7. Aplicaciones Móviles.

2.7.1. Introducción.

Una aplicación móvil o app, es un tipo de software que se puede usar en dispositivos móviles, tablets y pc después de ser instaladas. Su objetivo es ayudar al usuario a realizar alguna tarea específica. En la actualidad hay muchas apps con funcionalidades distintas como son: las redes sociales, salud y fitness, ocio y entretenimiento, etc. (La Mina Digital 2019).

Este tipo de software es diseñado específicamente para dispositivos móviles como celulares, tabletas inteligentes, relojes, etc. A pesar de no requerir muchos recursos estas pueden realizar operaciones en fracciones de tiempo muy cortos. Con el pasar de los años cada vez resulta más frecuente observar que la población las utilizan mucho más (Mroczkowska 2021).

Estas herramientas ofrecen un gran rendimiento a los usuarios sin embargo si se las compara con aplicaciones de escritorio, las apps móviles tienen un objetivo específico por lo tanto sus funcionalidades son limitadas con respecto a las aplicaciones de escritorio y estos pueden ser de tipos como son: juegos, aplicaciones de utilidad, entre otros (Herazo 2020).

2.7.2. Historia de las aplicaciones móviles.

Su historia aparece a finales de los años 90, llegando a facilitar tareas digitales a los usuarios de ese entonces con tan sencillas como una calculadora,

calendarios o agendas ya que dichas aplicaciones venían pre instaladas en los celulares de esa época (Escribano 2018).

El famoso juego del Tetris apareció en junio del año 1984, y fue creado por programadores de origen Ruso llegando a ser un videojuego muy conocido y jugado con el pasar de los años. En aquel tiempo también nació el videojuego llamado Snake y lo traía incorporado la marca Nokia, fue desarrollado en el año de 1997, el éxito que tuvo este videojuego hizo que para el año 2009 más de 350 millones de celulares ya tenían pre instalado en su sistema (Escribano 2018).

El objetivo de las aplicaciones móviles era facilitar y utilizar de mejor manera el tiempo de los usuarios sin importar su profesión ya sean usuarios comunes, profesionales y altos ejecutivos. Sin embargo, con el pasar del tiempo estas aplicaciones fueron evolucionando en el ámbito del ocio y entretenimiento en donde obtuvieron más acogida por parte de los usuarios (Escribano 2018).

Por el año 2010 las aplicaciones móviles estuvieron más enfocadas en el sector multimedia como reproductores de audio y video, así como los videojuegos. Con el avance de las redes telefónicas y el internet dichas aplicaciones se empezaron a utilizar en redes sociales. En la actualidad hay una infinidad de aplicaciones dependiendo de su sistema operativo y cada vez resulta más necesario usar una aplicación móvil que optimice algún proceso en la vida de los usuarios (Escribano 2018).

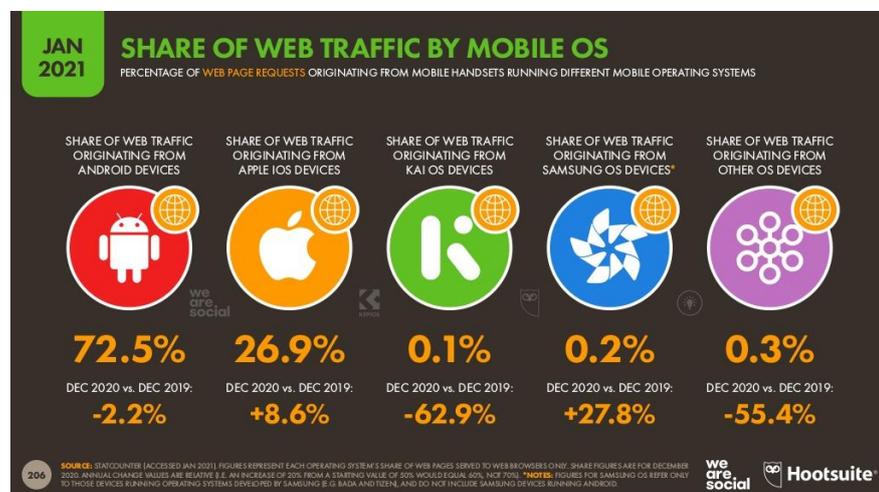
2.7.3. Sistemas operativos móviles.

Un sistema operativo es un conjunto de instrucciones lógicas que tienen como funcionalidad principal permitir correr programas, manejando una serie de tareas de forma simultánea así como componentes físicos como lo son teclados, mouse etc. (Cilsa 2021).

Las aplicaciones móviles como todo tipo de software requieren de un sistema operativo para poder ser ejecutado. A lo largo de la historia han existido muchas empresas en el área tecnológica las cuales han ido creando su propio sistema operativo (Albornoz 2019) como Apple, Google, Huawei los cuales trabajan constantemente para que sus sistemas operativos sean mejores que la competencia. En la figura 19 se puede observar el uso del tráfico web a través de aplicaciones móviles en los diferentes sistemas Operativos, concluyéndose que los más importantes son Android y IOS con un 72.5% y 26.9% de utilización respectivamente (Albornoz 2019).

Figura 19

Uso de tráfico web por un dispositivo móvil



Nota. Esta figura muestra el uso del tráfico web a través de los diferentes sistemas operativos de dispositivos móviles. Tomado de (We are social 2021:206).

Es por ello que los sistemas operativos son los encargados de poder ofrecer un ecosistema a los dispositivos móviles para que puedan funcionar sin problemas. Sin embargo, un mal funcionamiento en el sistema operativo puede incurrir en la producción de fallas muy graves incluso echando a perder al dispositivo móvil (Albornoz 2019).

2.7.4. Tipos de aplicaciones móviles.

De acuerdo con (Herrera 2018), este tipo de software ha venido evolucionando con el pasar de los años, permitiendo que cada vez las apps sean más completas y llenas de funcionalidades, buscando dar solución a varias problemáticas del mundo real. El sector del desarrollo de aplicaciones móviles es cada vez más solicitado por la industria tecnológica. Generando cada vez mayores plazas de trabajo a nivel mundial es por ello que a continuación se presenta los tres tipos de aplicaciones móviles 1) aplicaciones nativas, 2) aplicaciones web y 3) aplicaciones híbridas:

1) Aplicaciones nativas. - Son aquellas que son desarrolladas de forma particular para cada sistema operativo. Cada plataforma tecnológica posee su propio sistema operativo. Los sistemas operativos más conocidos son Android de Google, así como IOS de Apple (Herrera 2018).

Este software hacen un correcto uso del hardware por lo que se adaptan al 100% a los componentes y funcionalidades que se desee implementar, sin embargo, el desarrollo nativo implica mayor coste de producción en caso de desear un producto en distintas plataformas (Herrera 2018).

Poseen un mejor performance, así como una mayor velocidad de arranque y ejecución de subprocesos por lo que generalmente ofrecen una mejor experiencia al usuario que la utiliza. Los lenguajes de programación utilizados en aplicaciones nativas dependen de la plataforma es decir si se requiere una aplicación móvil para Android se puede utilizar Java o Kotlin mientras que para IOS se utiliza Swift (Herrera 2018).

2) Aplicaciones web. - Estos tipos de aplicaciones son construidas con tecnologías como Html, Css y Javascript utilizando conceptos de diseño responsivo para que se pueda adaptar a distintos tipos de pantallas. Es

necesario la utilización de un navegador web para poder visualizar estas aplicaciones. El costo de desarrollo suele ser un poco más barato que un desarrollo nativo y puede correr en distintas plataformas. Las desventajas pueden ser relacionadas con la velocidad y el rendimiento de dichas apps, así como que no es posible utilizar los recursos del hardware de los dispositivos móviles (Herrera 2018).

3) Aplicaciones híbridas. - Las aplicaciones híbridas con el avance de las tecnologías de desarrollo de software han ido siendo una solución muy conveniente, debido a que integra funcionalidades de las aplicaciones nativas, así como utilidades de las aplicaciones web. Haciendo que cada vez sea más rápido el desarrollo de aplicaciones multiplataforma con un muy buen rendimiento sin embargo sigue siendo inferior al de una aplicación desarrollada de forma nativa entre estas tecnologías se tiene a React Native, Flutter, Ionic (Herrera 2018).

2.8. Metodologías para el desarrollo de software.

2.8.1. Introducción.

Según (Software DELSOL 2019) la metodología es una técnica que se utiliza cuando se desea realizar alguna investigación de forma indagatoria con el objetivo de obtener resultados coherentes y eficaces, para ello se debe seguir una serie de estrategias durante el proceso de investigación. Además, son un conjunto de reglas, lineamientos y procedimientos para realizar alguna acción que requiera de ciertas habilidades y conocimientos específicos. Durante este procedimiento se definen roles, actividades y artefactos conjuntamente con buenas prácticas y técnicas recomendadas (Maida y Pacienza 2015).

Este tipo de guías metodológicas para el desarrollo de software aluden a la utilización de múltiples herramientas, métodos y técnicas para la elaboración

profesional de un producto software. Debido a que estos recursos buscan optimizar dicho proceso, guiando la planificación del grupo de trabajo. En el campo del desarrollo existen dos tipos de metodologías que son tradicionales y ágiles (Maida y Pacienza 2015).

A continuación, se describen de forma breve las dos metodologías utilizadas para el desarrollo de software 1) Tradicionales, y 2) Ágiles.

1) Metodologías Tradicionales. - También llamadas clásicas. Son conocidas por ser guías metodológicas molestas, dado que su objetivo principal es tener elaborados documentos técnicos por cada fase del proceso de desarrollo de software. Haciendo que el esfuerzo por documentar sea realmente intenso y tedioso. También se conoce que en estas herramientas realizar un cambio implica mayor costo de desarrollo es por ello que no es flexible ante el cambio por lo tanto no son recomendadas cuando los requisitos del proyecto no están del todo definidos (Maida y Pacienza 2015).

Las organizaciones que utilizan metodologías tradicionales distribuyen las etapas o fases de forma lineal, lo que conlleva a que para continuar con una siguiente fase se debe terminar primero la anterior. Tampoco es permitido volver a una etapa ya aprobada y realizar algún cambio. Por lo que es fácil inferir que estas guías no se adaptan a los cambios requeridos por parte del cliente (Santander 2020).

A continuación, se da una breve descripción de las principales metodologías tradicionales:

a) Waterfall o Cascada. - Metodología muy conocida debido a que utiliza el ciclo de vida también llamado Cascada y consiste en la división de las fases del desarrollo del software (Análisis del sistema, Análisis de requisitos, diseño, codificación, pruebas y mantenimiento). Los resultados de los avances en esta

metodología se los puede visualizar conforme avance el proyecto. De existir un error se lo podrá identificar en etapas avanzadas como se puede observar en el Capítulo II (2.6.3 Ciclos de vida del Software figura 7) repercutiendo en el incremento de costo final del proyecto de software (Santander 2020) .

b) Prototyping. - Esta guía metodológica es basada en la elaboración de prototipos de tal forma que los clientes pueden observar rápidamente como está siendo desarrollado su producto. Además, en caso de requerir algún cambio o corrección de fallos resulta más factible realizarlo en desarrollos tempranos. La filosofía de esta metodología es considerada como prueba y error hasta obtener un producto en el que el cliente se encuentre satisfecho como se puede observar en el Capítulo II (2.6.3 Ciclos de vida del Software figura 12) (Santander 2020).

c) Incremental. - Se fundamenta en la entrega constante de funcionalidades al cliente. Cabe destacar que los avances se los puede observar en etapas avanzadas tal cual se representa en el Capítulo II (2.6.3 Ciclos de vida del Software figura 14) (Santander 2020).

d) Espiral. - Esta metodología se fundamenta en la de cascada y prototipos. Su principal característica es iterar las veces que sean necesarias hasta generar un producto de calidad. Además, está constituida por 4 fases que corresponden a 1) planificación, 2) análisis de riesgo, 3) desarrollo del prototipo y 4) evaluación del cliente. De tal manera se conoce que mientras más cerca se encuentra en las fases del centro de la espiral el proyecto tiende a ser más robusto como se puede observar en el Capítulo II (2.6.3 Ciclos de vida del Software figura 15)(Santander 2020).

e) RAD. - En la figura 20 se muestra esta metodología de desarrollo. La cual nace de la necesidad de tener una guía flexible, rápida y que permita desarrollar aplicativos software de calidad. Está compuesta 4 fases que son las

siguientes: 1) planificación, 2) diseño y retroalimentación del usuario, 3) construcción, y 4) transición (Santander 2020).

Figura 20

Metodología diseño rápido de aplicaciones RAD



Nota. Esta figura muestra las fases de la metodología de desarrollo RAD. Tomado de (DiagramasUML 2019).

El cliente tiene una participación importante durante cada una de las etapas que contemplan esta metodología de desarrollo de software, sin embargo, la probabilidad de generar errores es mayor debido a que no se dispone de mucho tiempo para desarrollar una funcionalidad. Mientras menor sea la cantidad de iteraciones que se realice al aplicativo software es mejor para el grupo de trabajo.

2) Metodologías Ágiles. - El término “ágil” va mucho más allá que una metodología de desarrollo de aplicaciones software en donde se requiere de flexibilidad y rapidez. Comprende una forma de desarrollar productos software de calidad respondiendo a cada necesidad del cliente de forma rápida (Tena 2018).

Aparecen para solucionar los inconvenientes que las metodologías tradicionales podrían tener. Ya que su enfoque es la gran adaptabilidad al cambio, el trabajo continuo con el cliente, mejora en la toma de decisiones del equipo de desarrollo debido que su proceso está centrado en el producto antes que a la documentación excesiva que tienen las metodologías tradicionales. (Maida y Pacienza 2015).

Durante un manifiesto ágil los tiempos de entrega son reducidos, lo cual permite: planificar, crear y comprobar los resultados de forma rápida, además de mejorar la calidad del producto. Existe mayor compromiso entre el cliente y los desarrolladores. Además existe una mejor asignación de recursos lo que implica en mayor producción de la organización (Tena 2018).

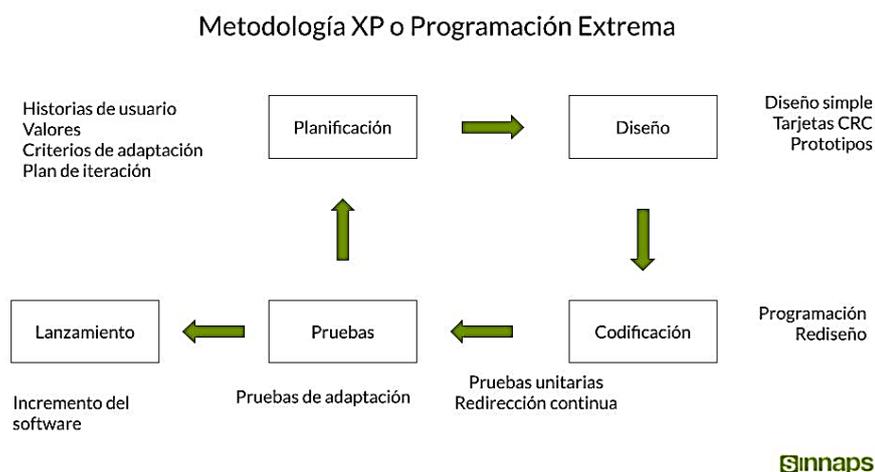
Cuando se desarrolla un proyecto aplicando alguna metodología ágil, el equipo de desarrollo se encuentra motivado, ya que les permite ser auto organizados por lo que incrementa la productividad laboral, y así cumplir a cabalidad con las metas propuestas en cada entrega (Maida y Pacienza 2015).

A continuación, se describe de forma breve las metodologías ágiles más importantes para el desarrollo de software:

a) Programación Extrema. - Nace a finales de los años 90 y fue creada por Kent Beck, Ron Jeffries y Ward Cunningham. Este tipo de metodología marcó un antes y un después en el proceso de ingeniería de software. Debido a que está centrada en promover el trabajo en equipo y de esta manera mejorar las relaciones interpersonales. Por lo tanto, los equipos de desarrollo aprendían de mejor forma, fomentando un ambiente de trabajo agradable (Maida y Pacienza 2015).

Xp es una metodología de desarrollo de software liviana, ofrece un conjunto de buenas prácticas, garantiza la satisfacción del cliente a través de un producto desarrollado con estándares de calidad. Sin embargo, no es recomendada usar en proyectos de largo plazo ya que se requiere de mucho esfuerzo (Digital Guide IONOS 2019).

En la figura 21, se muestra a las 5 fases de desarrollo en la metodología xp: 1) planificación, 2) diseño, 3) codificación, 4) pruebas y 5) lanzamiento (Digital Guide IONOS 2019).

Figura 21*Metodología Programación Extrema XP*

Nota. Esta figura muestra las fases de la Metodología de Programación Extrema. Tomado de (Sinnaps 2020).

b) Scrum. - Aparece en el año de 1980 solucionando procesos industriales en Japón y Estados Unidos llegando a ser un modelo exitoso. Con el paso de los años se convierte en una metodología flexible para la gestión del desarrollo de software, basado en el empirismo de las personas (Maida y Pacienza 2015).

Scrum está basado en 3 pilares los cuales se detallan brevemente a continuación (Maida y Pacienza 2015):

1) La transparencia. - Invita a que todos los involucrados en la organización sepan que es lo que se está realizando en cada artefacto.

2) La inspección. - Propone revisar de forma periódica el progreso para detectar de forma oportuna algún inconveniente.

3) La adaptación. - Menciona que el equipo de trabajo se ajusta a algún cambio requerido por parte del cliente y así conseguir el éxito (Abellán 2020).

Además, cabe destacar que los equipos Scrum son auto organizados, razón por la cual son multifuncionales sin embargo es necesario aclarar que existen tres roles (Maida y Pacienza 2015):

1) Product owner. - Encargado de transmitir las necesidades que posee el cliente de una determinada organización y por lo tanto conoce muy bien la idea de negocio.

2) El Scrum master. - Responsable de adoptar correctamente la guía Scrum, así como facilitar el los eventos a cada integrante del equipo de trabajo.

3) Equipo de desarrollo.- Encargados de realizar tareas de acuerdo a una planificación hecha durante los eventos de scrum (Abellán 2020).

c) Crystal. - Es una metodología ágil centrada en la cantidad de personas que componen un equipo de trabajo, así como la reducción de artefactos producidos durante el desarrollo de software. Por lo tanto su uso es recomendado para grandes organizaciones (Rruthers 2018).

Características como el tamaño de un equipo de desarrollo, comunicación entre los integrantes de un equipo de programadores, el espacio de trabajo, así como el espacio físico que cuenta la organización hacen que los desarrolladores sean un pilar muy importante en esta metodología (Rruthers 2018).

En la figura 23, se muestra a la organización del personal en esta metodología. Dividiéndoles de una manera que se considere la cantidad de personal y la complejidad asimilando a los minerales por su color y dureza (Rruthers 2018).

Figura 22

Organización de personal según la Metodología Crystal



Nota. Esta figura muestra la cantidad de personas requeridas para realizar algún tipo de proyecto software. Tomado de (Rruthers 2018).

Crystal está compuesto por 7 roles que son detallados brevemente a continuación (Rruthers 2018):

- 1) Patrocinador. - Persona encargada de realizar la ejecución del proyecto software, así como la entrega del capital necesario.
- 2) Diseñador Principal. - Mejor desarrollador que tiene la organización y por ende el encargado de guiar al resto de programadores.
- 3) Usuario experto. - Persona que aporta a la organización con su experiencia e información para obtener un mejor producto.
- 4) Diseñador programador. - Persona que se encarga de obedecer lo que dice el diseñador principal.
- 5) Experto en negocios. – Persona que conoce de negocios y debe establecer políticas o estrategias para que siga la organización.
- 6) Verificador. - Persona que genera reportes en caso de producirse un error en el sistema.
- 7) Escritor. – Persona realiza el manual de usuario del sistema desarrollado (Rruthers 2018).

d) Kanban. – El término “Kanban” proviene de origen japonés formada por Kan que significa visual y Ban que es tarjeta haciendo referencia a tarjetas visuales. Tiene por objetivo gestionar de forma eficiente cada tarea que se produce en el desarrollo de software hasta su finalización. Es una metodología muy sencilla de implementar ya que se utiliza un tablero también llamado Kanban (Redacción APD 2021).

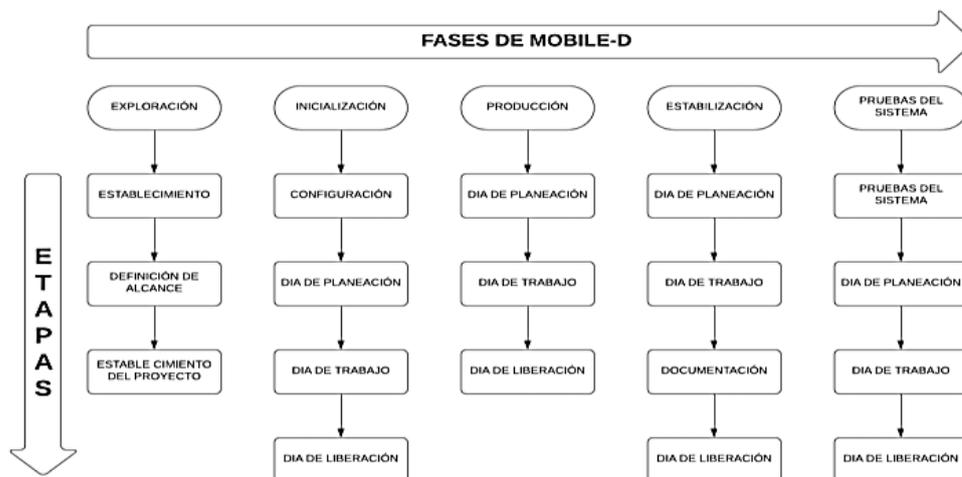
Esta metodología es muy recomendable usar cuando los requisitos de un sistema cambian de forma imprevista, las estaciones de trabajo se organizan de forma independiente y finalmente cuando la organización busca una optimización en diferentes procesos. La entrega continua, la transparencia de trabajo, y un constante seguimiento hace que Kanban sea una metodología ágil muy conveniente de aplicar en una organización (IMT Platform 2017).

e) Mobile-D. – Es una metodología ágil para el desarrollo de aplicaciones móviles que fue creada por Pekka Abrahamsson, así como su equipo de trabajo finlandés “Valtion Teknillinen Tutkimuskeskus” en el año 2004, está orientada al trabajo rápido mediante ciclos cortos. Además, el límite máximo de 10 personas, lo que hace que esta metodología sea recomendada para organizaciones pequeñas (Zambrano García 2017).

En la figura 23, se muestran las 5 fases que corresponden a esta metodología, las cuales son las siguientes (Zambrano García 2017): 1) exploración, 2) inicialización, 3) producción, 4) estabilización, y 5) las pruebas. Cabe destacar que esta metodología es una inspiración de las metodologías ágiles como XP y Crystal.

Figura 23

Fases de la Metodología Mobile-D



Nota. Esta figura muestra las fases de la metodología Mobile-D. Tomado de (Muñoz 2020).

2.9. Pruebas de Software.

2.9.1. Introducción.

Las pruebas de software son herramientas de verificación y aseguramiento de la calidad de un producto. El propósito de las pruebas es desarrollar un producto siguiendo estándares y lineamientos que permitan erradicar problemas durante el desarrollo de un sistema (Maida y Pacienza 2015).

Un producto software debe pasar por una serie de pruebas antes de ser lanzado a producción y cada una debe ser diferente, verificando múltiples perspectivas o escenarios del aplicativo desarrollado. Siendo las más importantes las siguientes: pruebas de rendimiento, pruebas de escalabilidad, pruebas de integración, y pruebas unitarias. (Lee 2020).

El software como cualquier otro producto debe ser probado con un conjunto de técnicas y procesos para validar su correcto funcionamiento.

Además de ser una parte sumamente trascendental en el ciclo de vida del para el desarrollo de software. Las pruebas son una forma de aseguramiento de que los requerimientos del cliente funcionan tal cual lo esperado. Existen dos tipos de pruebas de software que son (Lee 2020):

1) Pruebas funcionales. - Su principal característica es verificar los detalles más importantes de la idea de negocio, la usabilidad y obviamente las funcionalidades aplicadas al producto desarrollado. Por tal motivo el objetivo es garantizar que las funcionalidades se comporten según lo planificado por el equipo de desarrollo y el documento de especificación de requerimientos. Dentro de las pruebas funcionales las más conocidas son (Vargas 2019):

a) Pruebas unitarias. - Verifican el correcto funcionamiento de cada parte o funcionalidad mínima y tenga como salida los resultados esperados. Haciendo énfasis en los puntos críticos que podría tener problemas el aplicativo software.

b) Pruebas de integración. - Son utilizadas para probar cada componente desarrollado de forma individual se complementa con los demás módulos. Además, se la realiza de forma automática.

c) Pruebas de regresión. - Los desarrolladores por lo general buscan refactorizar su código, a veces introducen algún daño en el software de forma involuntaria durante este proceso, por lo que estas pruebas tienen por objetivo encontrar errores que hayan sido introducidos.

d) Pruebas de aceptación. - Una vez que ya se hayan completado cada una de las pruebas antes mencionadas se procede a realizar la prueba con el cliente en donde se verifica que el aplicativo cumpla con cada ítem que está presente en el documento de especificación de requerimientos.

2) Pruebas no funcionales. – Su principal característica es verificar los componentes de un aplicativo software como las pruebas de carga, de rendimiento, de estrés, etc. Estos tipos de pruebas son mecanismos de control y aseguramiento de una buena experiencia del cliente al usar el aplicativo software las principales 3 pruebas no funcionales son (Soto Morales 2021):

a) Pruebas de carga. - Se realizan con el objetivo de verificar cual es el comportamiento del aplicativo con una cantidad moderada de peticiones para poder ser comparadas con un comportamiento esperado o planificado.

b) Pruebas de rendimiento. - Son desarrolladas para medir los tiempos de respuesta del aplicativo dependiendo de la cantidad de peticiones que está teniendo durante la prueba.

c) Pruebas de estrés. - Estas pruebas son muy comunes para comprender cuantas peticiones puede soportar el aplicativo desarrollado. Por lo que durante esta prueba los límites son excesivamente superiores con respecto a las pruebas de carga y rendimiento.

2.10. Inteligencia Artificial.

2.10.1. Introducción.

La Inteligencia Artificial (IA) nace en el año de 1956 por Minsky y McCarthy, el cual pretende resolver problemas de métodos simbólicos, sin embargo, al pasar de los años cada vez es más frecuente escuchar las múltiples aplicaciones en el mundo real de la IA, y esto es gracias al avance de la tecnología y el aumento considerable de datos e información (Álvarez 2016).

Según (Frankenfield 2021) la Inteligencia Artificial (IA) es una rama de la tecnología con un fuerte origen en las matemáticas, la lógica y la biología. El cual

tiene por objetivo imitar acciones como lo hacen los humanos. A través del uso de algoritmos informáticos inteligentes.

Los algoritmos informáticos utilizados en la IA están enfocados generalmente a la toma de decisiones en mucho de los casos se lo realiza en tiempo real no como las máquinas tradicionales, que reaccionan de forma mecánica o predeterminada. En la IA se hace uso de sensores, datos digitales así como entradas remotas para la captura de información para que posteriormente actúe en función a su modelo de entrenamiento, tomando decisiones en cuestión de milisegundos (West 2018).

2.10.2. Etapas de la Inteligencia Artificial.

La Inteligencia Artificial (IA), al pasar de los años evoluciona a pasos agigantados. En donde existen muchas áreas en las cual se puede aplicar algoritmos inteligentes para solucionar alguna problemática. Por tal razón ha surgido la necesidad de separarla en las siguientes 3 etapas:

1) IA Estrecha (ANI). - Es aquella inteligencia que está siendo aplicada a máquinas en las que ya existe un conjunto de reglas predeterminadas por lo tanto no existe ninguna capacidad de razonamiento por parte de la máquina. Normalmente se ve utilizada en gestores GPS, así como en asistentes virtuales como Siri, Alexa en donde se requiere de respuesta inmediata (Rendón Molina, 2019).

2) Inteligencia Artificial General (AGI). - Es la evolución de la inteligencia artificial en donde las maquinarias ya poseen la capacidad de pensar y tomar decisiones en base a su experiencia tal cual lo realizan los humanos. Sin embargo, hasta la actualidad no existe alguna maquinaria que pueda tomar decisiones al 100% (Mejía Llano, 2020).

3) Super IA (ASI). - Teóricamente esta etapa de la IA comprende que la capacidad de las máquinas superará a la de los humanos. En la actualidad la ASI solo forma parte de la ciencia ficción, no obstante, se cree que en un futuro es posible que se de este escenario (Lateef, 2019).

2.10.3. Tipos de la Inteligencia Artificial.

La Inteligencia Artificial (IA) se recomienda ser clasificada de acuerdo a su tipo de funcionalidad entre ellas están:

1) Maquinas Reactivas. - Este tipo de IA es aplicado a máquinas en las que su funcionamiento está enfocado al procesamiento de datos en tiempo real, es decir la maquinaria no puede inferir basándose en datos pasados. Un claro ejemplo es cuando la empresa IBM generó un algoritmo de ajedrez el cual venció al campeón del mundo Garry Kasparov (Lateef, 2019).

2) Memoria limitada. - Esta IA utiliza gran cantidad de datos para la toma de decisiones basados en hechos pasados. Un claro ejemplo son los autos que se manejan a través de algoritmos inteligentes, los cuales utilizan sensores en su estructura para detectar objetos y tomar decisiones inmediatas (Lateef, 2019).

3) Teoría de la mente. - Es un tipo de IA muy avanzada que tiene como objetivo desarrollar máquinas basadas en la inteligencia emocional de las personas para poder comprender de mejor manera los pensamientos humanos. Sin embargo, aún en la actualidad no está bien desarrollado este tipo. Pero existen numerosas investigaciones que en un futuro permitan tener mejores resultados (Lateef, 2019).

4) Autoconciencia. - La evolución de la IA permitirá que las máquinas tengan autoconciencia es sorprendente pensar que en un futuro no muy lejano la superinteligencia sea posible gracias a la autoconciencia. Según Elon Musk y

Hawkings advierten que la evolución de la IA estará dando pasos agigantados en los próximos años (Lateef, 2019).

2.10.4. Ramas de la Inteligencia Artificial.

La Inteligencia Artificial (IA) puede ser aplicada para resolver problemas del mundo real utilizando diferentes procesos y técnicas, siendo las más importantes las siguientes:

1) Aprendizaje automático (Machine Learning). - Rama de la IA en la que las máquinas aprenden basados en hechos e información sin la necesidad de que un humano la programe previamente. Por lo que resulta muy eficiente realizar acciones como descifrar y estimar información a través de un conjunto de datos. En la actualidad los autos autónomos son conducidos a través de algoritmos inteligentes de machine Learning. Dentro del machine learning existen 3 tipos de aprendizajes que son: supervisados, no supervisados y por refuerzo (Tyagi 2021).

a) Aprendizaje supervisado. - En este tipo de aprendizaje las personas suministran la información de forma etiquetada al algoritmo, de tal forma que siempre se conoce las respuestas correctas e incorrectas. Son muy utilizados como clasificadores de correos y spam (Tyagi 2021).

b) Aprendizaje no supervisado. - En este aprendizaje se suministra datos no etiquetados para que el algoritmo encuentre relaciones entre los datos y pueda tomar decisiones. Es muy utilizado cuando es necesario entender gran cantidad de datos históricos (Tyagi 2021).

c) Aprendizaje por refuerzo. - Tiene similitud con el aprendizaje supervisado sin embargo en este tipo de aprendizaje no se suministra información etiquetada por lo que no se sabe con exactitud cuando el algoritmo

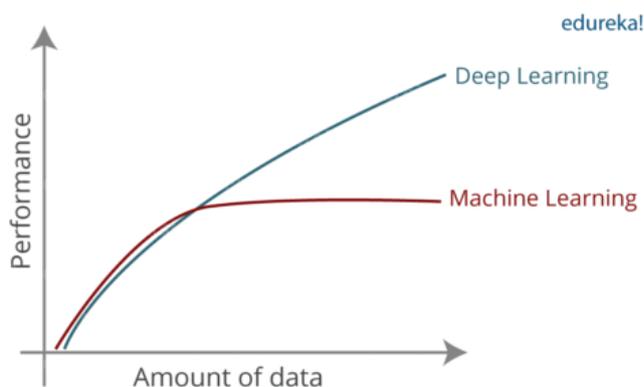
está presentado errores en su interior. Por tal motivo el aprendizaje por refuerzo aprende de acuerdo a sus éxitos y fallos (Tyagi 2021).

2) Aprendizaje profundo (Deep Learning). – O también llamado aprendizaje profundo es una sub categoría del Machine Learning el cual hace uso de redes neuronales similares a las del humano con el objetivo de realizar acciones como el reconocimiento de voz, reconocimiento de imágenes a través de la vision por computadora y el procesamiento del lenguaje natural (Banafa 2016).

En la figura 24 se muestra el rendimiento del Deep Learning vs el Machine Learning. En los últimos años es más frecuente escuchar como el Deep Learning ha permitido lograr avances tecnológicos de forma impresionante siendo utilizados como traductores de idiomas. El aprendizaje profundo hace uso de una gran cantidad de datos para la toma de decisiones, llegando a ser más efectivo en mucho de los casos con respecto al Machine Learning tradicional (Banafa 2016).

Figura 24

Rendimiento del Deep Learning vs Machine Learning



Nota. Esta figura muestra el rendimiento de los algoritmos de Deep Learning en comparación con los de Machine Learning tradicionales. Tomado de (Bakshi 2017).

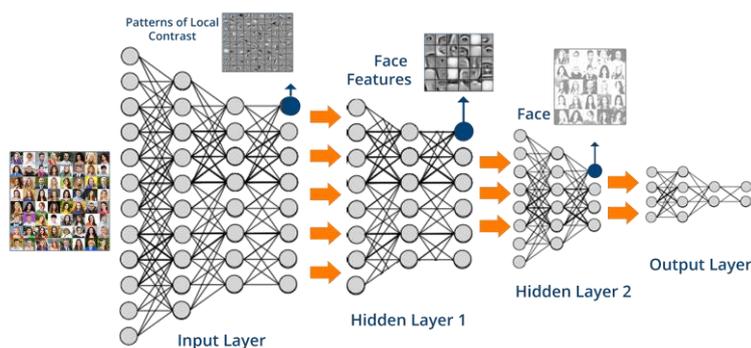
El aprendizaje profundo imita el comportamiento del cerebro humano por tal motivo su aprendizaje está centrado en la extracción de características de una

gran cantidad de datos, para basarse en experiencias tal cual lo hacen los humanos. En la Figura 24 se puede observar que el Deep Learning tiene un mejor rendimiento con mayor cantidad de datos con respecto a la del Machine Learning (Bakshi 2017).

En la figura 25, se muestra un modelo de arquitectura para el reconocimiento de rostros. Dicho reconocimiento es un ejemplo claro del uso del aprendizaje profundo y las redes neuronales convolucionales (CNN) en la vida real. Basándose en el manejo de imágenes que representan las propiedades físicas humanas como son: las orejas, la nariz, y los ojos. Permittedose la clasificación de imágenes, en este caso el reconocimiento de rostros (Bakshi 2017).

Figura 25

Redes Neuronales para el reconocimiento de rostros



Nota. Esta figura muestra la estructura de las redes neuronales para el reconocimiento de rostros. Tomado de (Bakshi 2017).

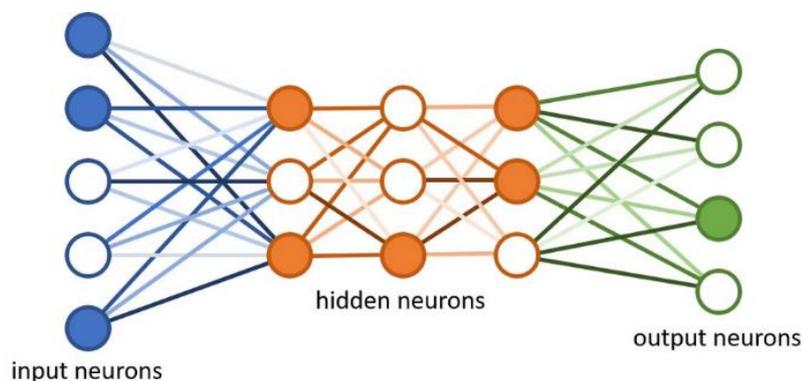
Dentro del campo del Aprendizaje Profundo existen dos formas en las que una red neuronal puede ser entrenada (MathWorks 2021):

a) Entrenamiento desde cero. - En la figura 26, se muestra una arquitectura de red neuronal (NN). La cual está compuesta de las siguientes capas: 1) entrada, 2) ocultas y 3) salida. Cabe destacar que es importante para

el entrenamiento de una red neuronal poseer una gran cantidad de información o datos, repercutiendo en su posterior aprendizaje (MathWorks 2021).

Figura 26

Estructura de una red neuronal

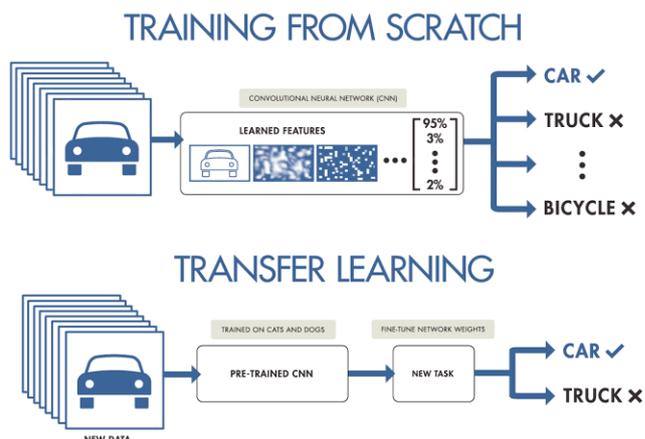


Nota. Esta figura muestra la estructura básica de una red neuronal. Tomado de (Román, 2019).

b) Transferencia de aprendizaje. - En la figura 27, se puede observar la diferencia entre el entrenamiento de una red neuronal desde cero vs el uso de Transferencia de aprendizaje (Transfer Learning). Técnica muy poderosa que ha venido tomando fuerza en los últimos años, pues su esencia está en utilizar la gran cantidad de datos de forma general, para poder resolver un problema en específico. Es decir, se busca modificar los patrones ya entrenados a través de algoritmos muy complejos para reconocer particularidades especiales en una problemática más simple (Shah 2019).

Figura 27

Entrenamiento desde cero vs Entrenamiento usando Transfer Learning



Nota. Esta figura muestra la diferencia entre un algoritmo entrenado desde cero con uno que utiliza Transferencia de aprendizaje. Tomado de (Shah 2019).

La transferencia de aprendizaje empieza utilizando algoritmos muy complejos como AlexNet, GoogleNet, o ResNet teniendo millones de datos de entrada para posteriormente crear nuevas clases con el objetivo de solucionar una problemática nueva. Se requiere pocos ajustes en los hiper parámetros para obtener los mejores resultados. Por lo tanto si se requiere de un clasificador de animales utilizaría unos 1000 imágenes ya no millones como se requeriría en una red neuronal desde cero (MathWorks 2021).

Finalmente, debido a que la transferencia de aprendizaje ya no requiere de los procesos complejos como lo requeriría una la red neuronal desde cero se obtiene mejores resultados haciendo que sea factible usar para problemas de clasificación de imágenes (Shah 2019).

Existen tres categorías de transferencia de aprendizaje los cuales dependen del dominio del problema, así como la cantidad de datos que estos utilizan. A continuación se presenta dichas categorías (Shah 2019):

- 1) Aprendizaje de transferencia inductivo: Es utilizado cuando existe una problemática muy similar a la del dominio del problema. Sin embargo,

las tareas de clasificación son diferentes. Los algoritmos de IA se perfeccionan hasta el punto de dar solución a las tareas del destino (Shah 2019).

- 2) Aprendizaje de transferencia no supervisado: Esta técnica es muy parecida a la inductiva, y la diferencia está en que aquí no existe datos etiquetados como en el inductivo (Shah 2019).
- 3) Aprendizaje por transferencia transductivo: Esta técnica es muy diferente de las antes dos ya mencionadas, debido a que los datos de origen y destino son distintos (Shah 2019).

En la figura 28, se muestra las estrategias recomendadas para el uso de transfer learning. Una vez entendido las tres clasificaciones de los tipos de aprendizaje por transferencia se recomienda una serie de estrategias para seleccionar un tipo de aprendizaje de acuerdo a la problemática que de obtenga.

Figura 28

Estrategias recomendadas de uso en Transfer Learning

Learning Strategy	Related Areas	Source & Target Domains	Source Domain Labels	Target Domain Labels	Source & Target Tasks	Tasks
Inductive Transfer Learning	Multi-task Learning	The Same	Available	Available	Different but Related	Regression Classification
	Self-taught Learning	The Same	Unavailable	Available	Different but Related	Regression Classification
Unsupervised Transfer Learning		Different but Related	Unavailable	Unavailable	Different but Related	Clustering Dimensionality Reduction
Transductive Transfer Learning	Domain Adaptation, Sample Selection Bias & Co-variate Shift	Different but Related	Available	Unavailable	The Same	Regression Classification

Nota. Esta figura muestra una serie de recomendaciones para el uso estratégico de una técnica de transfer learning. Tomado de (Shah 2019).

- 3) El procesamiento de lenguaje natural (NLP). - Esta técnica de Inteligencia Artificial (IA) hace mención a la posibilidad de que las máquinas pueda comprender palabras ya sea de forma oral o escritas. Los chatbots son un

claro ejemplo de la aplicabilidad de esta rama de la IA, también pueden ser clasificadores de spam muy usados con Machine Learning (Ovrmind 2019).

Empresas como Twitter utiliza algoritmos de procesamiento de lenguaje natural para detectar posibles amenazas terroristas a través de su red social. Mientras que Amazon la utiliza para poder entender las necesidades del cliente en sus comentarios y así de esta manera mejorar la experiencia de usuario (Lateef 2019).

4) Robótica. - Es una ciencia relacionada con diferentes campos tecnológicos la cual tiene como objetivo diseñar y producir maquinas con diferentes capacidades para realizar tareas de automatización de procesos comúnmente usados en la industria (EDS 2021).

Cada vez es más común poder observar robots en la vida diaria de las personas en su gran mayoría son utilizados en el sector industrial en donde se requiere de procesos totalmente automatizados como por ejemplo en el armado de autos. Actualmente Sophia es una robot con inteligencia artificial que fue creada para poder interactuar con los humanos (Lateef 2019).

5) Lógica difusa. – Es una rama de la IA fundamentada en el lenguaje booleano. Es muy utilizada en el campo de la medicina debido a que se requiere de un procesamiento booleano para la toma de decisiones. También es muy empleado en los autos con transmisiones automáticas (Lateef 2019).

6) Sistemas expertos. – Son productos software para la toma de decisiones de tal forma que se asemeja a la de un profesional en el área. Los algoritmos de Inteligencia Artificial contienen el suficiente conocimiento almacenado a través de una gran cantidad de datos (Tecnologías de la Información 2018).

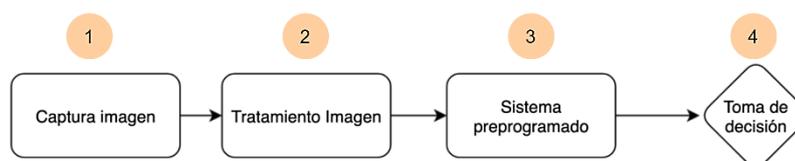
Los sistemas expertos cuentan con cinco componentes que son: la base de conocimiento, el motor de inferencia, el módulo de adquisición de conocimiento, la interfaz de usuario y finalmente el módulo de explicación (Tecnologías de la Información 2018).

2.10.5. Visión Artificial o visión por Computadora.

Esta técnica nace de la necesidad de extraer características físicas en imágenes, para poder ser procesadas a través de algoritmos de Deep Learning y permitir el reconocimiento de objetos o una serie de patrones. En la figura 29, se presenta al flujo necesario para clasificación utilizando la vision por computadora clasifica el cual está compuesto de las siguientes partes 1) captura de imagen, 2) tratamiento de imagen, 3) sistema preprogramado y 4) toma de decisión. En los últimos años este enfoque ha ido evolucionando pasando desde lectores de códigos de barras (Datision 2020).

Figura 29

Flujo de la visión por computadora clásica.



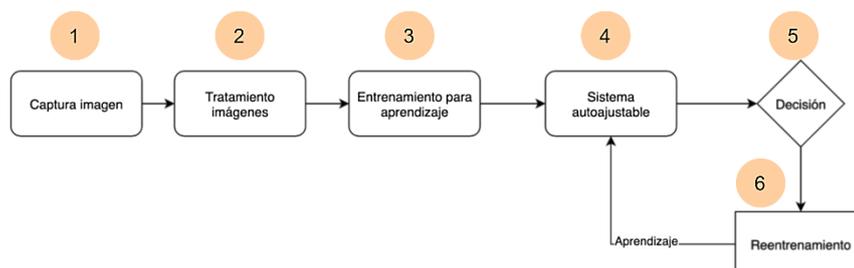
Nota. Esta figura muestra el flujo que sigue la vision por computadora clásica o tradicional. Tomado de (Datision 2020).

En la figura 30, se ilustra al flujo actual de la visión por computadora. La cual está compuesta de las siguientes partes: 1) captura de imagen, 2) tratamiento de imágenes, 3) entrenamiento para aprendizaje, 4) sistema ajustable, 5) decisión y 6) reentrenamiento. Conforme ha ido evolucionando la tecnología y la inteligencia artificial, la visión por computadora igualmente ha ido presentando avances muy importantes gracias a la utilización del aprendizaje

profundo en donde los resultados son exponencialmente mejores, pues ahora el flujo de trabajo es más completo (Datision 2020).

Figura 30

Flujo actual de la visión por computadora

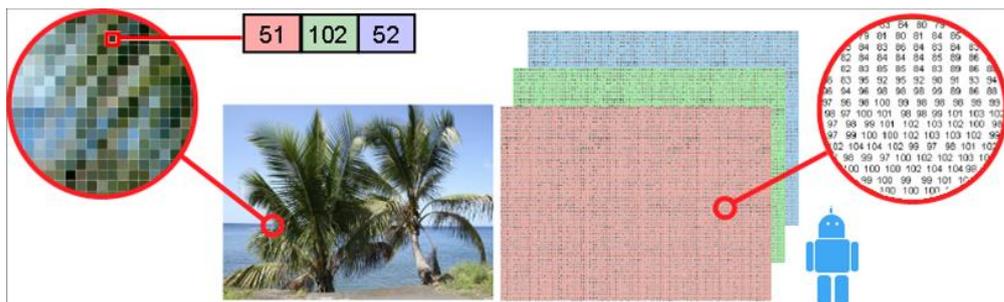


Nota. Esta figura representa el flujo de trabajo actual de la visión por computadora. Tomado de (Datision 2020).

En la figura 31 se observa la extracción de características de una imagen digital. La visión por computadora cada vez implica un reto muy interesante para la tecnología y sobre todo la detección de objetos pues la dificultad de esta técnica radica en la existencia de variaciones de una misma clase, por ejemplo, si se desea detectar autos hay de muchas marcas, colores, tamaños etc. Esto se produce debido a una imagen es transformada a matrices numéricas que están compuestas por valores de forma multidimensional (Lapedriza 2012).

Figura 31

Extracción de características de una imagen digital



Nota. Esta figura muestra el proceso de extracción de características de una imagen digital. Tomado de (Lapedriza 2012).

2.10.6. Redes Neuronales Convolucionales (CNN).

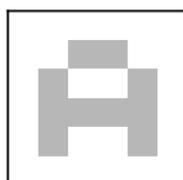
Las Redes Neuronales Convolucionales o también llamadas CNN, son una variante de las redes neuronales artificiales en la cual su objetivo es imitar el comportamiento del ojo humano a través de un cerebro biológico. Son muy utilizadas en el campo de la vision artificial para la clasificación y segmentación de imágenes (Barrios 2019).

De acuerdo con (Bagnato 2019), las cnn son un tipo de red inteligente que procesa capas para imitar el funcionamiento del córtex humano y así identificar características de entrada. Además, contienen muchas capas ocultas las cuales permiten detectar líneas, figuras y curvas. De esta manera el algoritmo se va especializando.

En la figura 32, se muestra la conversión de imágenes a una matriz numérica. Para que una red neuronal pueda aprender por si sola. Esta requiere de una gran cantidad de imágenes, por lo tanto, si se desea clasificar plantas estas deben tener al menos 10000 fotos de girasoles, otras 10000 de rosas y así sucesivamente por cada clase que se desee entrenar al modelo (Bagnato 2019).

Figura 32

Conversión de imágenes a una matriz numérica



Una imagen...

		0.6	0.6		
	0.6			0.6	
	0.6	0.6	0.6	0.6	
	0.6			0.6	

...es una matriz de píxeles.

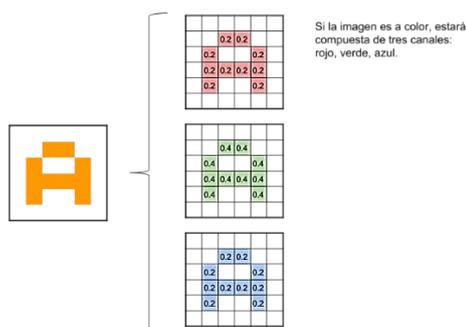
El valor de los píxeles va de 0 a 255 pero se normaliza para la red neuronal de 0 a 1

Nota. Esta imagen muestra el proceso de extracción de características de una letra para convertirla en una matriz numérica. Tomado de (Bagnato 2019).

En la figura 33 se muestra la caracterización de una imagen en rgb (red, green, blue). Como se pudo observar en la figura 32 la imagen cuenta con un tamaño simétrico de 28px x 28px lo que equivale aproximadamente a 784 neuronas (inputs). Considerando que la imagen se encuentra en escala de grises posee un canal de color si la imagen estuviera con colores tuviera la siguiente configuración $28\text{px} \times 28\text{px} \times 3 = 2352$ neuronas (inputs) (Bagnato 2019).

Figura 33

Caracterización de una imagen en rgb

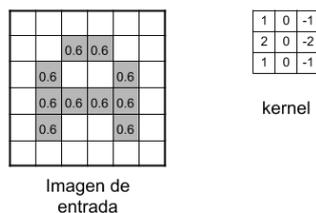


Nota. Esta imagen muestra la extracción de características de una imagen amarilla en tres matrices de colores rojo, verde y azul. Tomado de (Bagnato 2019).

En la figura 34, se ilustra el proceso de multiplicación de la imagen de entrada por un kernel. Las convoluciones aparecen cuando se toma grupos de pixeles cercanos es decir se toma la imagen de entrada y se realiza multiplica utilizando un producto escalar con una matriz llamada kernel en donde inicialmente toma valores aleatorios. Después se multiplica la imagen de entrada con el kernel obteniendo una nueva matriz que será la nueva capa oculta que tendrá la neurona (Bagnato 2019).

Figura 34

Multiplicación de la imagen de entrada por el kernel



Nota. Esta figura muestra el proceso de multiplicación de valores de la matriz de la imagen de entrada con la matriz del kernel para formar una nueva capa oculta. Tomado de (Bagnato 2019).

De acuerdo con la figura 34 el proceso de la multiplicación entre los pixeles de la imagen y el kernel varias veces es llamada como filtros. Por lo tanto, en la primera convolución se podría tener 32 filtros que equivale a 32 matrices de salida este proceso es también llamado como feature mapping. Para saber cuál es la cantidad de neuronas en la primera capa oculta se debe realizar la siguiente operación: $28px * 28px * 1px * 32 \text{ filtros} = 25088 \text{ neuronas}$ (Bagnato 2019).

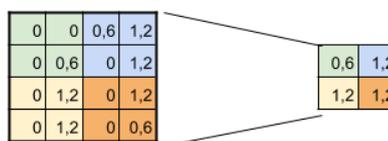
Conforme se va realizando el desplazamiento del kernel se va generando una nueva matriz en la que cada nueva imagen va dibujando características de la imagen original permitiendo en un futuro poder clasificar si una imagen

corresponde a una determinada ya sean flores, perros, gatos, etc (Bagnato 2019).

En la figura 35, se observa el proceso de reducción de Subsampling con Max Pooling. La cantidad de neuronas deben ir disminuyendo conforme más capas ocultas se obtiene debido a que mientras más neuronas tiene la red neuronal esta requiere de un mayor procesamiento computacional. Por lo tanto tomando las 25088 neuronas de la primera capa oculta se las puede ir reduciendo aplicando Max-Pooling (Bagnato 2019).

Figura 35

Subsampling utilizando Max-Pooling



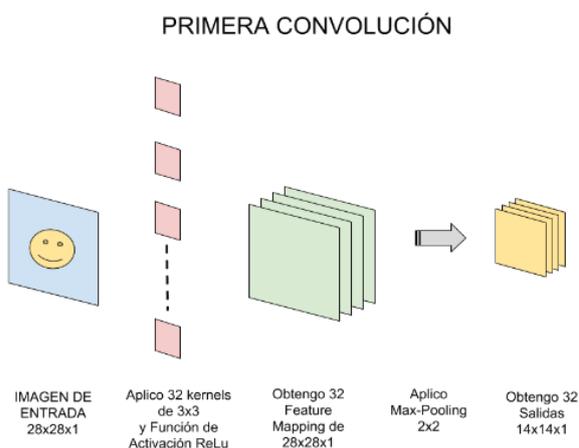
SUBSAMPLING:
 Aplico Max-Pooling de 2x2
 y reduzco mi salida a la mitad

Nota. Esta figura muestra la aplicación de SubSampling utilizando Max-Pooling para reducir el tamaño de la matriz. Tomado de (Bagnato 2019).

En la figura 36, se mira el proceso que se realiza en la primera convolución de una red neuronal. Max-Pooling busca reducir el tamaño de las imágenes tomando más de 2 píxeles en vez de uno. De esta forma va recorriendo de izquierda a derecha, así como arriba hacia abajo. Este proceso se lo realiza por todas matrices de tal forma que inicialmente se tenía un total de 25088 neuronas con un tamaño de 28px * 28px * 32 filtros a una de 14px * 14px * 32 filtros quedando con aproximadamente 6272 neuronas con respecto a la capa oculta anterior (Bagnato 2019).

Figura 36

Proceso de la primera convolución

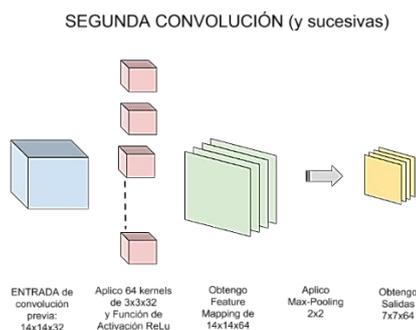


Nota. Esta figura muestra el proceso de conversión durante la primera convolución en una red neuronal. Tomado de (Bagnato 2019).

En la figura 36 se pudo observar el proceso que se realiza en la primera convolución en la red neuronal. Dicha convolución tiene como objetivo detectar características ya sean líneas o curvas, conforme se van desarrollando más convoluciones estas van detectando figuras más completas como se lo puede ver en la figura 37 en donde la salida de la segunda convolución es 7px * 7px * 64 filtros (Bagnato 2019).

Figura 37

Proceso de la segunda convolución.



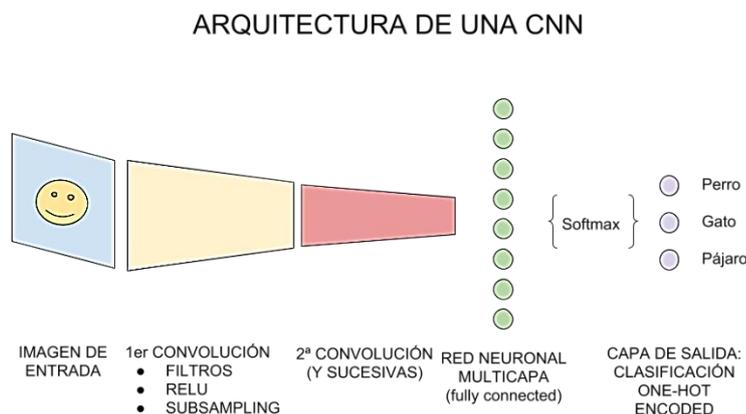
Nota. Esta figura representa el proceso que se realiza durante la segunda convolución en la que la capa de entrada es la capa oculta de la anterior. Tomado de (Bagnato 2019).

Para la tercera convolución se toma la salida de $7\text{px} \times 7\text{px} \times 64$ filtros de la anterior capa oculta como entrada de la nueva convolución. Para posteriormente aplicar el kernel obteniendo un feature mapping de $7\text{px} \times 7\text{px} \times 128$ filtros y al aplicar el Max-Pooling de 2×2 se obtiene una salida de $3\text{px} \times 3\text{px} \times 128$ filtros llegando a ser la última convolución (Bagnato 2019).

2.10.7. Arquitectura de una red neuronal convolucional (CNN).

Toda red neuronal convolucional (CNN) requiere de cierta arquitectura con el objetivo de obtener la mejor solución a una problemática en común (Bagnato 2019).

En la figura 38, se puede observar la arquitectura de una red neuronal convolucional. La cual está compuesta de las siguientes 4 partes: 1) entrada (input-s), 2) convoluciones (filtros, relu, subsampling), 3) función de activación (soft-max) y 4) salida (One-hot-Encoding). Las redes neuronales convolucionales pasan por varios procesos para poder ser entrenadas y así detectar características desde las más simples hasta las más complejas. Por lo tanto, después de realizar varias sucesiones de las capas ocultas a través de las convoluciones, se requiere de una función llamada de "activación" (soft-max, sigmoid, relu, etc). La cual tiene por objetivo conectar la capa de salida de las convoluciones con las clases que se tengan en el algoritmo que también son llamadas como One-Hot-Encoding. (Bagnato 2019).

Figura 38*Arquitectura de una CNN*

Nota. Esta figura muestra la arquitectura básica de una red neuronal convolucional. Tomado de (Bagnato 2019).

2.10.8. Métricas de Evaluación de algoritmos de IA.

De acuerdo con (Pmoinformatica 2018b) las matrices de software son unidades de medición aplicadas durante algún proceso en el desarrollo de software. Con el objetivo de hallar valores que sirvan para la toma de decisiones, una de las métricas más comunes es la precisión (accuracy).

En el campo de la Inteligencia Artificial existen una serie de métricas que permiten evaluar varios criterios de un algoritmo inteligente, con el objetivo de visualizar de forma matemática como se ha ido comportando, dando la posibilidad de poder realizar algún reajuste de ser necesario (Barrios 2019). Las más conocidas son:

1) Precisión. - Métrica utilizada para medir el porcentaje de predicciones que son correctas del modelo de red neuronal. Esta métrica debe utilizarse cuando las clases utilizadas en el algoritmo están balanceadas. Es decir que todas las clases deben tener la misma cantidad de elementos. Sin embargo, no es una métrica fiable cuando las clases no están balanceadas. Pudiendo

presentarse sobreajuste (overfitting) y desajuste (underfitting). En la ecuación 3 se observa el cálculo de la precisión, la cual es el resultado de sumar los verdaderos negativos (TN) más los verdaderos positivos (TP) y dividir sus resultado para la sumatoria de los verdaderos negativos (TN), falsos positivos (FP), falsos negativos(FN) y verdaderos positivos(TP) (Sancho 2021).

Ecuación 3

Fórmula de la métrica Precisión

$$Precision = \frac{TN + TP}{TN + TP + FN + FP}$$

Nota. Esta ecuación representa a la fórmula matemática para el cálculo del accuracy. Tomado de (Sancho 2021).

2) Matriz de confusión. - En la figura 39, se puede observar esta herramienta. La cual es muy utilizada para verificar el rendimiento de un algoritmo de inteligencia artificial. Tomándose en cuenta los valores de predicción vs los valores reales (Barrios Arce 2019).

Figura 39

Matriz de Confusión

VALORES PREDICCIÓN	Verdaderos positivos	Falsos Positivos
	Falsos Negativos	Verdaderos Negativos
	VALORES REALES	

Nota. Esta figura muestra a la estructura de una matriz de confusión. Tomado de (Barrios Arce 2019).

3) Recall/TPR/Sensitivity. - Presenta la división de las predicciones correctas con respecto al total de casos positivos del dataset o conjunto de datos. En la ecuación 4 se puede visualizar el cálculo de la métrica de recall, la cual es el resultado de dividir los verdaderos positivos TP, entre la adición de verdaderos positivos TP y falsos negativos FN.

Ecuación 4

Formula de la ecuación Sensitivity

$$Sensitivity = \frac{TP}{TP + FN}$$

Nota. Esta ecuación muestra la fórmula matemática para el cálculo de la sensibilidad o Recall. Tomado de (Sancho, 2021).

4) La especificidad. - Muestra de todos los valores negativos, cuantos son predichos como negativos. Es decir, es la relación de predicciones negativas con respecto a la cantidad de casos negativos existentes. En la ecuación 6 se puede mostrar el cálculo de la especificidad, la cual es el resultado de dividir los verdaderos negativos (TN) entre la suma de los verdaderos negativos(TN) y falsos negativos (FN) (Sancho 2021).

Ecuación 5

Fórmula de la ecuación de Especificidad

$$Especificidad = \frac{TN}{TN + FP}$$

Nota. Esta ecuación muestra la fórmula matemática para el cálculo de la especificidad. Tomado de (Sancho 2021).

Cuando las clases no se encuentran balanceadas es recomendable usar precisión o sensitivity en vez de la formula del Accuracy, también es muy recomendable usar la métrica F1 (Sancho 2021).

5) F1 score. - Aparece para ser utilizada cuando las clases se encuentran desbalanceadas como por ejemplo la clase 1 tiene 85 muestras mientras que la clase 2 tiene 10 muestras, Por lo tanto, F1 une métricas como Precisión y Recall

en una sola. En la ecuación 7 se puede observar el cálculo de la F1 score, la cual es el resultado de multiplicar la constante de valor 2 por el producto entre el recall y la precisión entre la el producto de recall más precisión (Sancho 2021).

Ecuación 6

Ecuación de F1 Score

$$F1 = \frac{2 * (Recall * Presicion)}{(Recall + Presicion)}$$

Nota. Esta ecuación muestra la unión de las fórmulas de Recall y precisión para el cálculo de F1. Tomado de (Barrios Arce 2019).

De acuerdo a la fórmula de F1 Score se obtienen cuatro posibles salidas que son las siguientes (Barrios Arce 2019):

- Alta precisión y alto recall: Quiere decir que el modelo maneja de forma correcta cada clase.
- Alta precisión y bajo recall: Se dice que el modelo no detecta del todo una clase, sin embargo, cuando lo hace la confiabilidad es muy alta.
- Baja precisión y alto recall: El modelo detecta bien la clase sin embargo también detecta los valores de otra clase lo que no hace confiable.
- Baja precisión y bajo recall: El modelo no tiene la capacidad de clasificar correctamente.

2.11. Herramientas de desarrollo de Software.

2.11.1. Introducción.

En el campo del desarrollo de software son muchas las herramientas que permiten desarrollar productos estos pueden ser compiladores, ide, editores de

código, depuradores, además de herramientas para el versionamiento de código (Velneo 2019).

El proceso de desarrollo de software requiere de múltiples herramientas que permitan a los programadores desarrollar software de forma rápida y eficaz, sin importar las tecnologías o los lenguajes que estén siendo utilizados (Da Silva 2021).

Es por ello que a continuación se describen a los tipos de herramientas utilizadas para el desarrollo de software.

2.11.2. Tipos de herramientas para el desarrollo de software.

- **Compilador:** Es una herramienta de software utilizada para la conversión de código escrito en lenguajes de alto nivel a lenguajes de máquina comúnmente llamados como machine code (European Valley 2018).
- **Editor de código:** Son herramientas el cual su función está centrada en permitir editar código fuente en múltiples lenguajes de programación, actualmente cada vez son más robustos permitiendo incluir plugin para mejorar la productividad del desarrollador (HostGator 2021)
- **Depuradores:** Los depuradores son programas que verifican la existencias de fallos o errores de programación sobre algún lenguaje de programación en específico el proceso de debugeo de errores se lo realiza cuando el algoritmo o programa se encuentra en ejecución (Digital Guide IONOS 2020).
- **Versionamiento de Código:** En la actualidad es muy común escuchar el termino de versionamiento de código, esta herramienta permite poder compartir código fuente entre desarrolladores y organización además de llevar un control de cada versión que es desarrollada con respecto algún producto software. Las herramientas de versionamiento más comunes son Github así como BitBucket (García 2019).

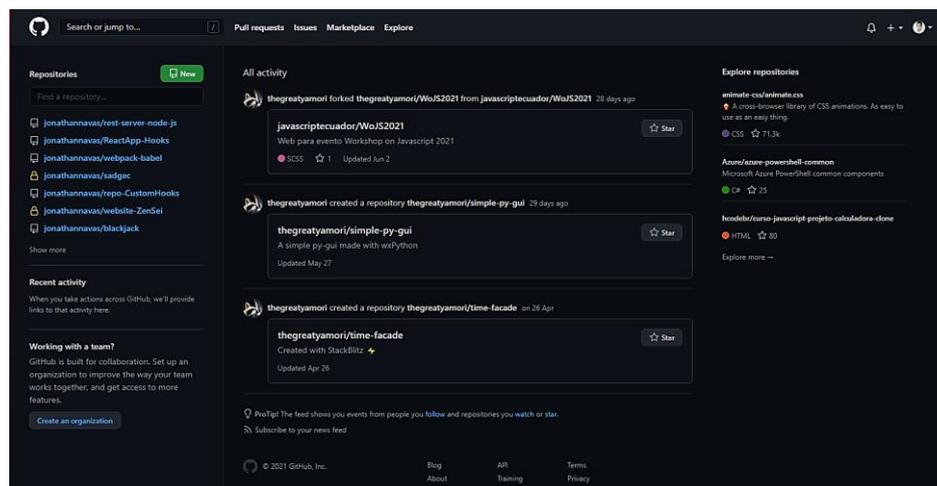
2.11.3. GitHub.

GitHub es una herramienta muy poderosa, utilizada para el control y versionamiento de código fuente, está compuesta de herramientas que permiten la colaboración 24/7 de forma remota. Github fué adquirida por la empresa Microsoft en el año 2018 y está basada en la filosofía de versionamiento de Git creada por Linus Torvalds (Fernández 2019).

En la figura 40, se presenta la interfaz de Github la cual es muy simple e intuitiva además cuenta con una amplia documentación de cada funcionalidad que ofrece a los desarrolladores de software. Existen dos formas de publicación de proyectos estos pueden ser de código abierto ubicados en repositorios públicos mientras que si se desea subir un proyecto del que requiera solo poder ver una cierta organización se la puede subir en repositorios privados (Fernández 2019).

Figura 40

Interfaz de GitHub



Nota. Esta figura muestra la interfaz principal de Github. Tomado de (Github 2021).

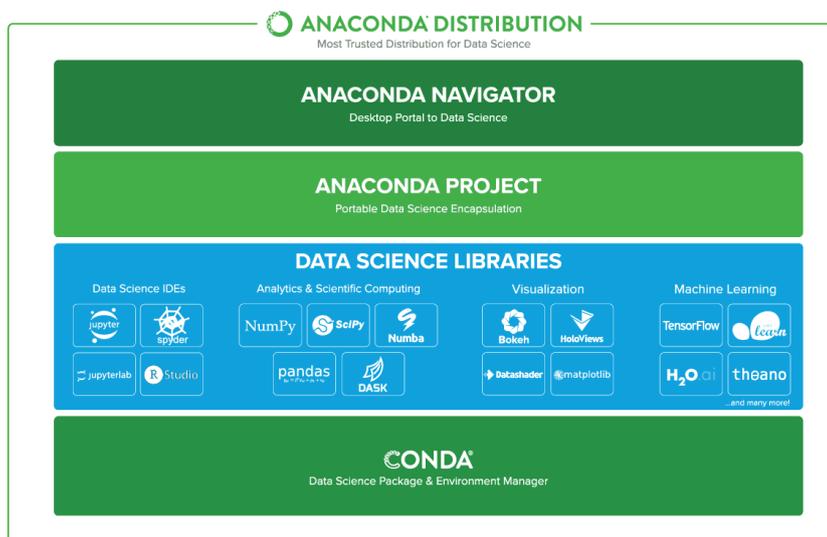
2.11.4. Anaconda.

Anaconda es una suite muy completa de Python que contiene librerías para resolver problemas de ingeniería, matemáticas, estadística, así como temas de inteligencia artificial, está disponible para las plataformas Windows, Mac Os y Linux (Toro 2017).

La figura 41, ilustra la suite de Anaconda que está compuesta por cuatro tipos de soluciones tecnológicas que son Anaconda Navigator, Anaconda Project, librerías para la ciencia de datos y finalmente su componente Conda. Todas ellas se incluyen directamente en la instalación de aplicativo software (Toro 2017).

Figura 41

Suite de Anaconda



Nota. Esta figura muestra los componentes de la suite de Anaconda. Tomado de (Toro 2017).

2.11.5. Android Studio.

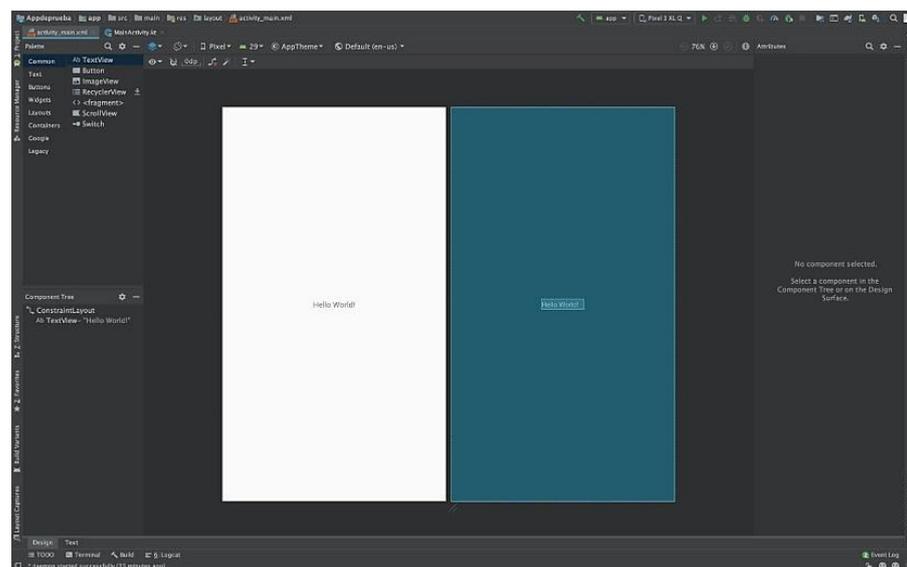
Es el entorno de desarrollo integrado (IDE) más utilizado para la elaboración de aplicaciones móviles del sistema operativo Android. Cuenta con herramientas y servicios necesarios para que un programador pueda crear

aplicaciones móviles de forma rápida y sencilla. Es un software multiplataforma lo que quiere decir que puede ser instalado en sistemas operativos como Windows, Mac OS y Linux. Este IDE permite el uso de lenguajes de programación como Java, Kotlin y gracias a su Android Ndk es posible programar también con lenguajes como C++ o C (Collado 2021).

En la figura 42, se muestra la interfaz de Android Studio la cual cuenta con un emulador de Android muy útil para cuando el desarrollador necesita probar su aplicación en tiempo real, sin embargo, es necesario mencionar que para que el emulador presente un buen rendimiento el ordenador donde está instalado debe contar con buenas especificaciones técnicas, por otro lado la interfaz que posee es muy simple e intuitiva (Collado 2021).

Figura 42

Interfaz Android Studio



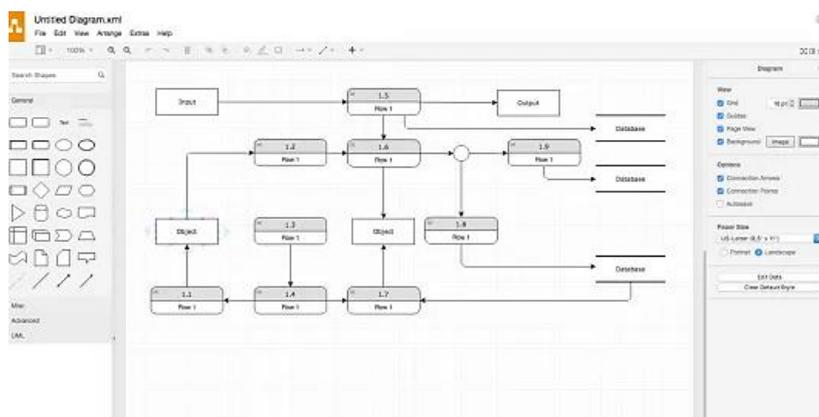
Nota. Esta figura muestra la interfaz del IDE Android Studio. Tomado de (Collado 2021).

2.11.6. Draw.io.

En la figura 43, se muestra la interfaz principal de Draw.io. Es una herramienta, muy utilizada para la creación y edición de diagramas UML, esquemas de red, flujogramas, mapas conceptuales, así como diagramas de Venn. Es un aplicativo web muy sencillo de utilizar y permite generar diagramas muy profesionales (Mancomún 2021).

Figura 43

Interfaz Draw.io



Nota. Esta figura muestra la interfaz de Draw.io Tomado de (Mancomún 2021).

Draw.io viene compuesto de una serie de plantillas predeterminadas por lo que si el usuario no conoce de la generación de algún diagrama en especial dicho software lo guía de forma rápida. Tiene la capacidad de exportar a múltiples formatos de imagen como png y jpg de forma rápida siendo una aplicación muy interesante para utilizar (Calvo 2018).

2.11.7. Figma.

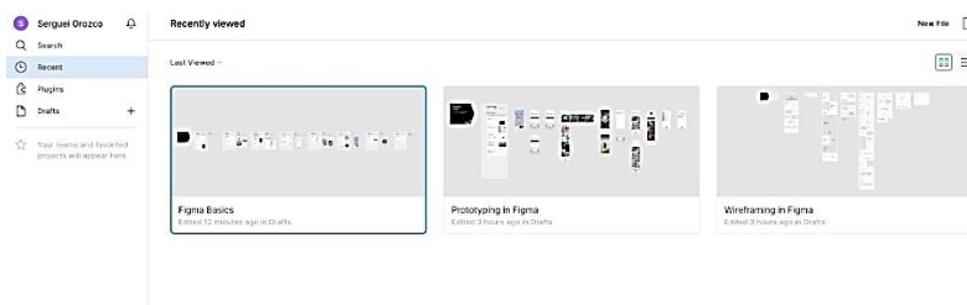
Es una herramienta muy utilizada para la creación y diseño de interfaces. Apareció aproximadamente a finales del año 2015. Y gracias a sus desarrolladores en la actualidad permite tener una colaboración en tiempo real.

Figma ofrece a sus usuarios la posibilidad de ser utilizada tanto en Windows, Linux y Mac (Mejía 2021).

En la figura 44, se presenta a la interfaz de Figma la cual es muy sencilla e intuitiva lo que la pone por encima de su competencia. La comunidad de esta aplicación web y de escritorio constantemente se encuentra generando recursos de forma gratuita y de paga lo que la hace tener una comunidad muy activa (Mejía 2021).

Figura 44

Interfaz de Figma



Nota. Esta figura muestra la interfaz de Figma. Tomado de (Mejía 2021).

La función de prototipos es muy utilizada pues gracias a esta funcionalidad es posible presentar al cliente una versión visual de como estará compuesto su producto software de cierta forma es muy conveniente pues de ser necesario es fácil poder tomar correctivos antes de pasar a la etapa de desarrollo de algún producto software (Mejía 2021).

2.11.8. Teachable Machine.

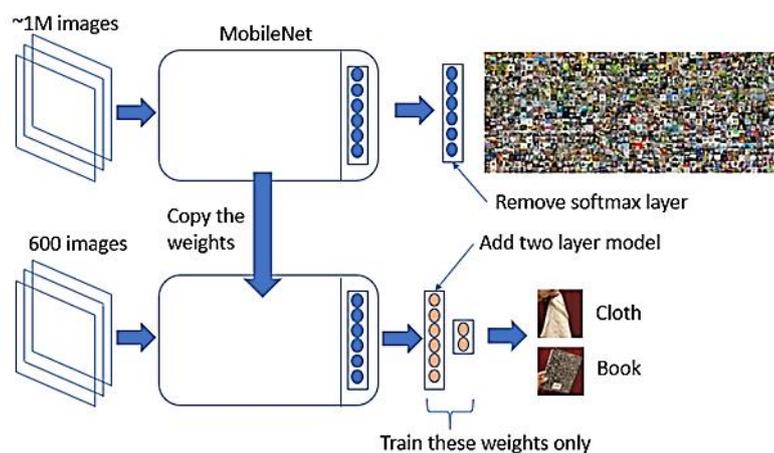
Teachable Machine es una herramienta de Google que revoluciona de forma impresionante la creación de modelos con inteligencia artificial. La primera versión fue creada en el año 2017. Sin embargo ha evolucionado de tal manera

que en la actualidad es una herramienta poderosa y muy utilizada en distintas investigaciones científicas (Teachable Machine 2017).

En la figura 45, se muestra el proceso de extracción de conocimiento a través de la herramienta Teachable Machine. La cual utiliza el modelo de MobileNet que posee millones de imágenes agrupadas por 28 capas convolucionales para la clasificación de imágenes. MobileNet usa un dataset llamado ImageNet el cual está compuesto por más de 1000 clases de objetos. Después se procede a eliminar la capa softmax y se reemplaza con la capa del dominio del nuevo problema a solucionar. De esta manera resulta muy fácil crear modelos inteligentes a través de modelos ya pre entrenados. Además de no requerir muchas imágenes como se necesitaría si se deseara crear un modelo desde cero (Balasubramaniam 2019).

Figura 45

Proceso de extracción de conocimiento mediante la transferencia de aprendizaje



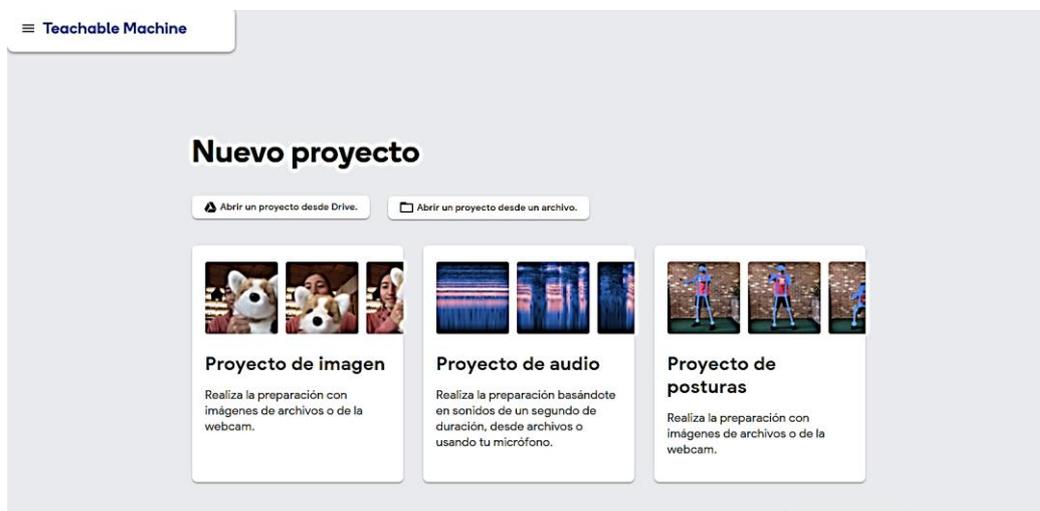
Nota. Esta figura muestra cual es el procedimiento para extraer características de un modelo ya pre entrado y ajustarlo a una nueva problemática. Tomado de (Balasubramaniam 2019).

En la figura 46, se presenta los tipos de proyectos que pueden ser realizados a través de la herramienta Teachable Machine. Esta herramienta

permite crear modelos inteligentes para: clasificación de imágenes, clasificación de audios e identificación de poses de una persona (Teachable Machine 2017).

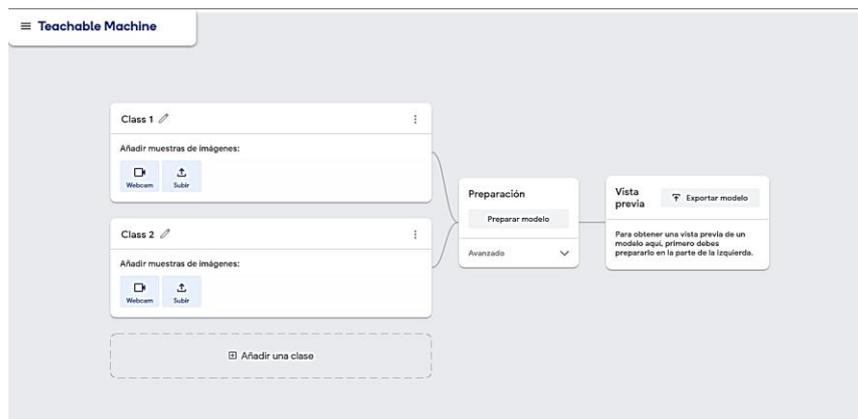
Figura 46

Tipos de proyectos Teachable Machine



Nota. Esta figura muestra las tres posibilidades de proyectos que pueden generarse con la herramienta Teachable Machine. Tomado de (Teachable Machine 2017).

En la figura 47, se muestra la interfaz para crear proyectos de clasificación de imágenes. La interfaz sencilla e intuitiva que posee esta herramienta hace que no se requiera de altos conocimientos en programación, así como en inteligencia artificial para generar un modelo inteligente. El clasificador de imágenes es un módulo muy utilizado en investigaciones de revistas científicas pues solo se requiere suministrar las clases con su respectivo conjunto de imágenes (Teachable Machine 2017).

Figura 47*Interfaz Clasificador de Imágenes*

Nota. Esta figura muestra la interfaz para la creación de un modelo clasificador de imágenes muy potente. Tomado de (Teachable Machine 2017).

En la figura 48, se visualiza un ejercicio didáctico en el cual se genera un clasificador de animales. Para lo cual utilizándose las siguientes tres clases: 1) perros, 2) gallinas y 3) mariposas. Generándose un dataset de 1000 imágenes por cada clase respectivamente. Obteniéndose un total de 3000 imágenes (Teachable Machine 2017).

Figura 48*Clasificación de animales usando Teachable Machine*

Nota. Esta figura muestra la configuración de imágenes de las clases mariposas, perros y gallinas usando Teachable Machine para la clasificación de imágenes. Tomado de (Teachable Machine 2017).

En la figura 49, se muestra la configuración de parámetros necesarios para crear un modelo inteligente. A continuación, se describen los 3 parámetros necesarios (Teachable Machine 2017):

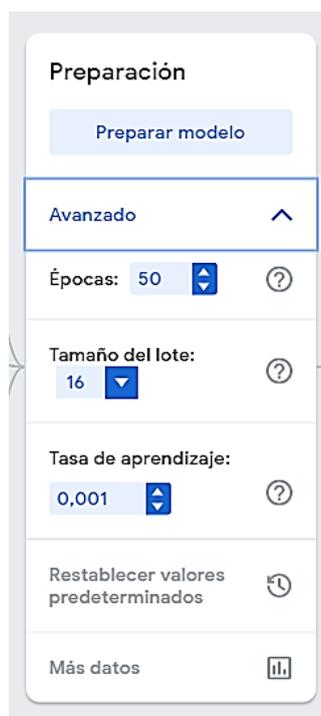
1) Épocas. - Número de iteraciones, en las que se recorre un conjunto de datos para la fase de entrenamiento.

2) Batch size (tamaño del lote). - Corresponde a cuantas veces se divide el dataset para completar una época.

3) Learning rate (tasa de aprendizaje).- Parámetro de ajuste que determina el tamaño de paso de cada interacción en la función de pérdida (Teachable Machine 2017).

Figura 49

Configuración de parámetros Teachable Machine



The image shows a mobile application interface for configuring a machine learning model. At the top, there is a section titled 'Preparación' (Preparation) with a 'Preparar modelo' (Prepare model) button. Below this is an 'Avanzado' (Advanced) section, which is currently expanded. It contains three settings, each with a dropdown menu and a help icon (question mark): 'Épocas' (Epochs) set to 50, 'Tamaño del lote' (Batch size) set to 16, and 'Tasa de aprendizaje' (Learning rate) set to 0,001. At the bottom of the advanced section is a button labeled 'Restablecer valores predeterminados' (Reset to default values) with a circular arrow icon. Below the advanced section is a 'Más datos' (More data) button with a bar chart icon.

Nota. Esta figura muestra la configuración de parámetros para la creación de un modelo inteligente el cual debe ser parametrizada hasta que ofrezca los mejores resultados.

En la figura 50, se observa las pruebas realizadas al modelo entrenado. Una vez presionado el botón para entrenar el modelo Teachable Machine procede a entrenar el modelo con las imágenes suministradas previamente, además de dividir el dataset de la siguiente manera 85% para entrenamiento y 15% de imágenes para las pruebas de forma automática. Posteriormente se puede probar el modelo generado utilizando imágenes distintas que el modelo no conozca para verificar la precisión del modelo (Teachable Machine 2017).

Figura 50

Pruebas del modelo para la clasificación de imágenes.

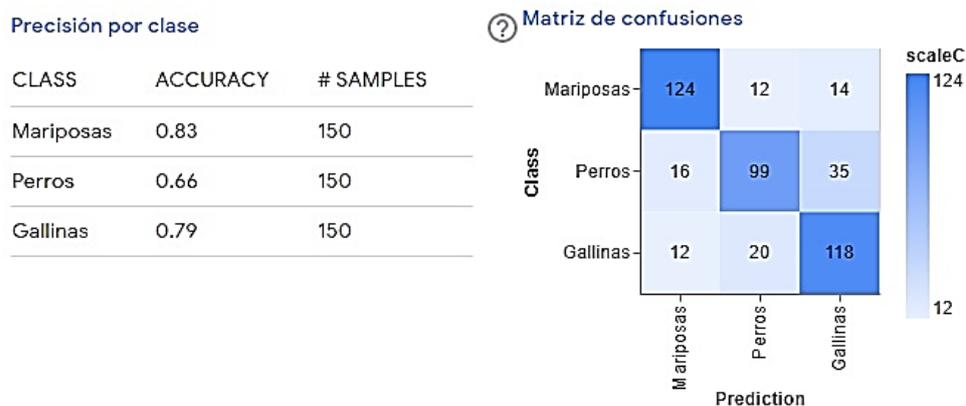


Nota. Esta figura muestra la prueba para verificar la precisión del modelo generado a través de teachable machine. Tomado de (Teachable Machine 2017).

En las figuras 51 y 52 se muestran los resultados del modelo de redes neuronales convolucionales (CNN) generado. Para lo cual se presenta la precisión por clase, la matriz de confusión y las gráficas de pérdida por época respectivamente (Teachable Machine 2017).

Figura 51

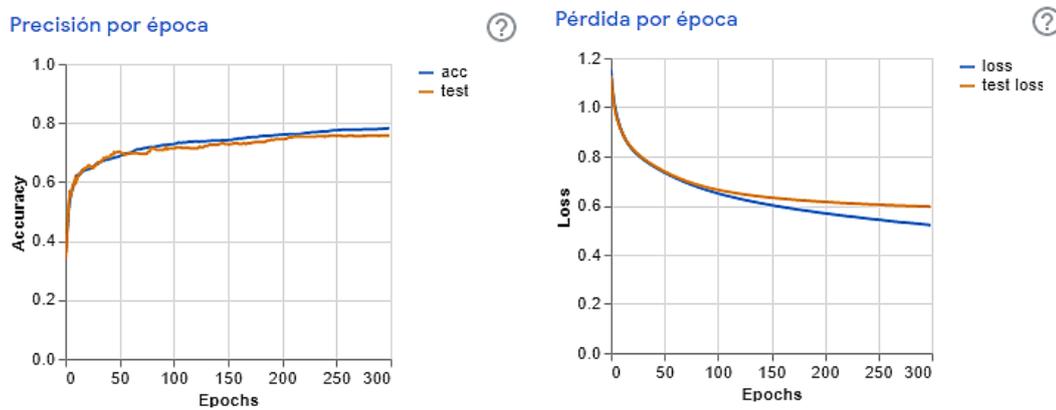
Resultados de la precisión por clase y la matriz de confusión clasificación de animales



Nota. Esta figura muestra los resultados obtenidos del entrenamiento del modelo para la clasificación de animales. Tomado de (Teachable Machine 2017).

Figura 52

Gráfica de precisión y pérdida por época

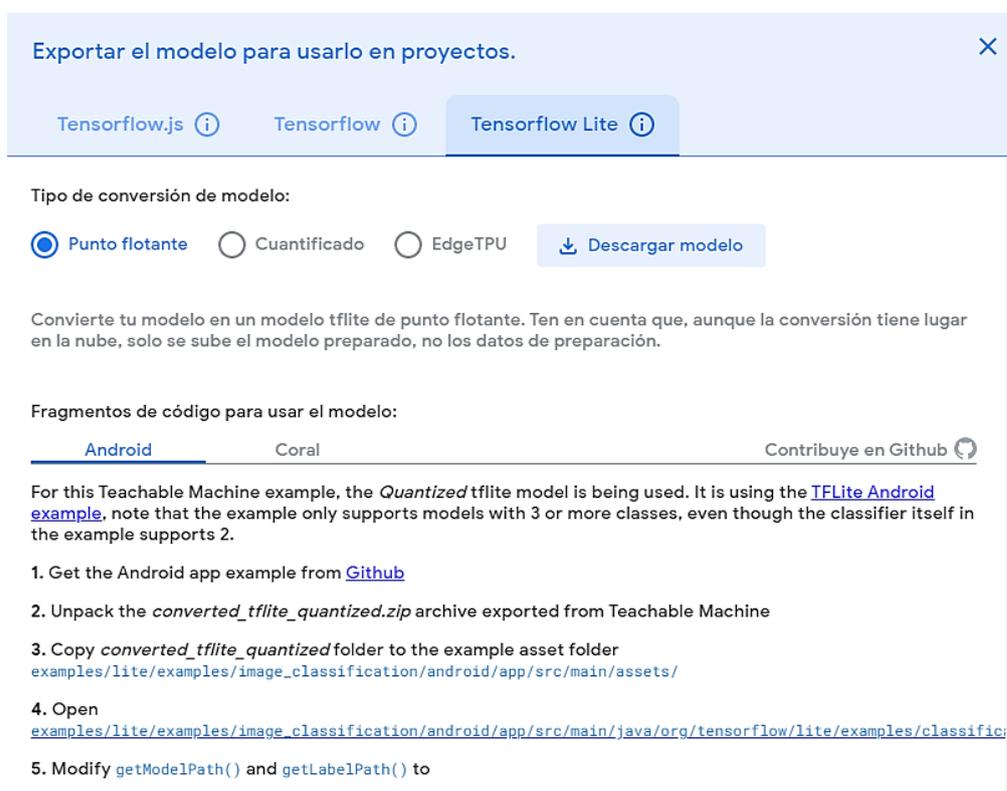


Nota. Esta figura muestra los resultados de la precisión y pérdida por época. Tomado de (Teachable Machine 2017).

En la figura 53, se muestra el proceso de exportación del modelo generado. Para lo cual se visualiza las opciones de exportación que tiene esta herramienta siendo estas: 1) tensorflow, 2) tensorflow.js, y 3) tensorflow lite.

Figura 53

Exportación del modelo generado a través de Teachable Machine



Nota. Esta figura muestra las posibilidades que tiene una persona para exportar el modelo generado a través de Teachable Machine. Tomado de (Teachable Machine 2017).

Capítulo III

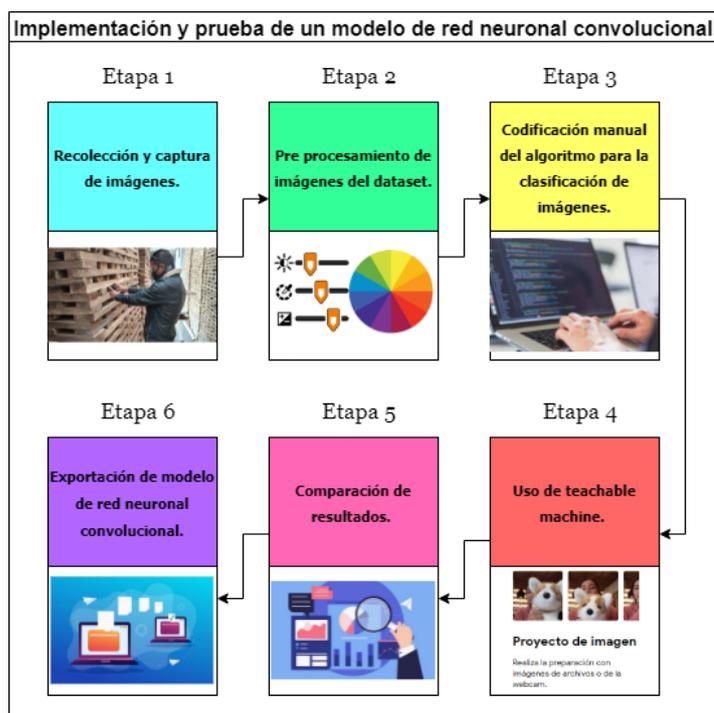
3. Implementación y prueba de un modelo de red neuronal convolucional.

3.1. Introducción.

El presente capítulo contempla en detalle la estructura necesaria para implementar y probar un modelo de red neuronal enfocado en una arquitectura de redes neuronales convolucionales (CNN). En la Figura 54 se puede observar los 6 componentes que la conforman los cuales son los siguientes: 1) Recolección y captura de imágenes, 2) Pre procesamiento de imágenes del dataset, 3) Codificación manual del algoritmo de clasificación de imágenes, 4) Uso de Teachable Machine, 5) Comparación de resultados, 6) Exportación del modelo.

Figura 54

Estructura del Capítulo III



Nota. Esta figura muestra la estructura del Capítulo III, el cual corresponde a la implementación y prueba de un modelo de red neuronal enfocado en una arquitectura de redes neuronales convolucionales.

A continuación, se explican las 6 etapas que contemplan los componentes que conforman la implementación y prueba de un modelo de red neuronal convolucional.

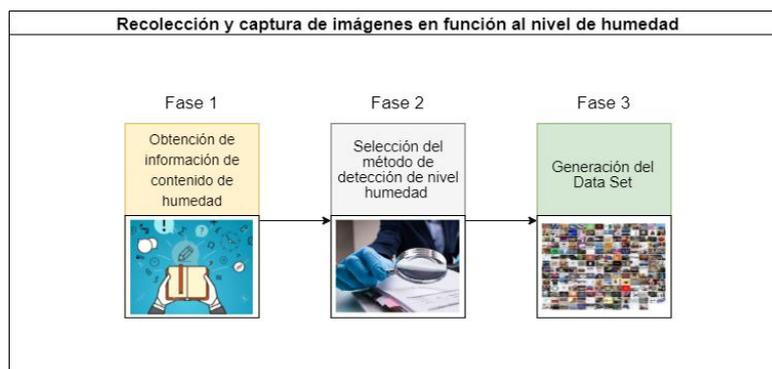
3.2. Etapa 1. Recolección y captura de imágenes.

En esta etapa se realiza la recolección de información acerca del contenido de humedad en la madera Laurel. A través del uso de las principales fuentes información abiertas (OSINT, WEBINT, IMINT, HUMINT y VIRTUAL HUMINT) con el fin de adquirir conocimiento e información (orientación del experto - manuales) que permita determinar la selección de un método adecuado para la detección de su humedad. Además, también se elabora la captura de imágenes de una muestra de madera Laurel en base a su humedad con el objetivo de crear un dataset de madera Laurel.

En la figura 55 se visualiza las fases que intervienen en la recolección y captura de imágenes de madera Laurel, las mismas que fueron planteadas de la siguiente manera: 1) Obtención de información de contenido de humedad, 2) Selección del método de detección de nivel humedad y 3) Generación del Data Set.

Figura 55

Recolección y captura de imágenes en función al nivel de humedad.



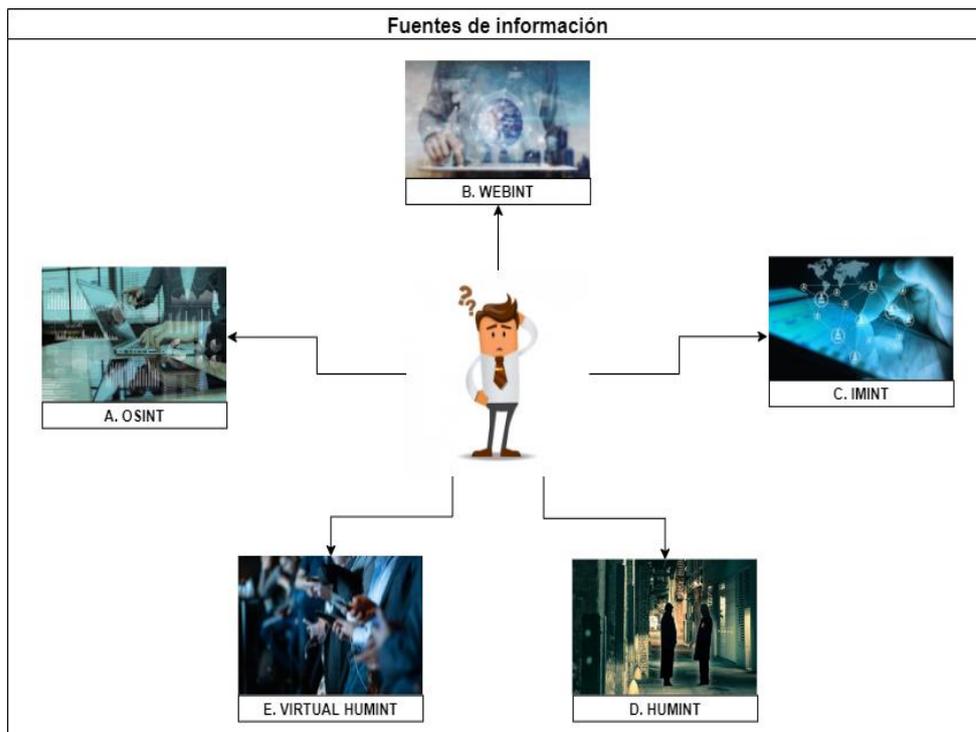
Nota. En esta figura se muestra las fases que se deben emplear para realizar la recolección y captura de imágenes en función al nivel de humedad.

A continuación, se explican las 3 fases que intervienen en la recolección y captura de imágenes.

3.2.1. Fase 1. Obtención de información de contenido de humedad.

En esta fase se ejecuta la recolección de información acerca de contenido de humedad mediante las fuentes de información abierta, con el fin de determinar la influencia que tiene la humedad en los procesos de transformación de la madera laurel.

En la figura 56 se observa las fuentes de información abiertas que se utilizan: A) OSINT, B) WEBINT, C) IMINT, D) HUMINT y E) VIRTUAL HUMINT. Dichas fuentes permiten definir conceptos utilizados en el sector maderero. Permitiendo adquirir conocimientos sobre la madera de Laurel, determinando métodos y técnicas existentes enfocadas en la detección de contenido de su humedad.

Figura 56*Fuentes de información*

Nota. En esta figura se muestra las fuentes de información abiertas que se utilizaron para esta fase.

A continuación, se explican brevemente los 5 tipos de fuentes de información abiertas que se utilizan en la presente investigación:

1. OSINT. - Información conocida como Inteligencia de Fuentes Abiertas. Permite buscar datos disponibles sobre personas, hechos, empresas o instituciones, haciendo que el cumulo de datos se convierta en conocimiento (LISA Institute 2020b).

La presente fuente se utiliza para acceder a documentos que contengan datos útiles acerca de los procesos que intervienen en la transformación de la madera. Para lo cual se realizó una búsqueda de datos en los siguientes principales medios de información siendo estos: bibliotecas e internet. Sin embargo, no se obtuvo información

relevante, dado que existe una cantidad limitada de informes acerca de dicho tema.

2. WEBINT. - Información denominada fuentes digitales especializadas o restringidas. Permiten hacer una indagación en documentos que están en un estado oculto o restringido, es decir que no todos los usuarios pueden acceder a dicha información (Comunycarse 2020).
La presente fuente se utiliza para acceder a datos técnicos acerca de métodos de detección de contenido de humedad. Para lo cual se hizo uso del navegador Brave y el motor de búsqueda DuckDuckGo, permitiendo dirigir la búsqueda a un servidor Tor. Ocultando la dirección IP de un ordenador a los sitios web que se acceso.
3. IMINT. - Información nombrada como Inteligencia de imágenes. Permite recabar datos gráficos como ayuda a la toma de decisiones por parte de instituciones y empresas (LISA Institute 2019).
La presente fuente se utiliza para determinar la existencia de datasets de imágenes de madera de laurel. Para lo cual se realizó una indagación breve en los principales sitios web que proporcionan datasets siendo estos: Kaggle, UCI Machine Learning Repository, VisualData, entre otros. Sin embargo, luego de realizar dicha búsqueda se concluyó que ningún dataset satisfacía la necesidad del proyecto, por lo tanto, se creó uno manualmente.
4. HUMINT. - Información denominada Inteligencia de Fuentes Humanas elaborada en base a la información que se ha obtenido o ha sido proporcionada por personas (LISA Institute 2020a).
La presente fuente se utilizó para adquirir información acerca del manejo de los métodos de detección en la madera laurel. Para lo cual se realizaron reuniones diarias entre los tesisistas y el experto en la madera. Tratándose temas relacionados con las formas de

emplear dichos métodos en una muestra determinada. Cabe recalcar que dichas reuniones se cancelaron a mitad de esta fase. Debido a la pandemia global causada por el covid-19 impidiendo el normal desenvolvimiento de dichas juntas. Presentándose como una adecuada alternativa de solución el uso de Virtual Humint (fuente de información abierta).

5. VIRTUAL HUMINT. – fuente de información abierta que permite obtener datos acerca del contenido de humedad de la madera laurel. A través de la modalidad digital o virtual (Díaz Caneja 2020).

La presente fuente se utilizó como forma alternativa para mantener la comunicación entre los tesisistas de la presente investigación y el experto en la madera del aserradero “San Martín”. Para lo cual se hizo uso de la plataforma zoom, permitiéndose ejecutar las reuniones a través del video chat entre 3 participantes (2 tesisistas y 1 experto en la madera). Convirtiéndose en la principal fuente de información, debido a la facilidad que proporciona esta fuente al momento de adquirir información acerca del contenido de humedad.

3.2.2. Fase 2. Selección del método de detección del nivel de humedad.

En esta fase se realiza la selección de un método de detección del nivel de humedad en la madera Laurel en base a las 3 siguientes características: 1) experiencia, 2) factibilidad y 3) tiempo, las mismas que se detallan a continuación:

1. Experiencia. - Información basada en conocimiento adquiridos a través del tiempo mediante técnicos especializados en la selección, procesamiento y tratamiento de la madera tipo Laurel.

2. Factibilidad. - Condiciones que posee el aserradero “San Martín” para poder aplicar una técnica de detección de humedad, tales como: el número de muestras de Laurel disponibles y el acceso a las parrillas de madera de Laurel.
3. Tiempo. - Duración que toma en aplicar una técnica de medición de humedad del Laurel en el aserradero “San Martín”.

Luego de analizar las características antes mencionadas y a la adquisición de información relacionada con los métodos de detección de humedad, datos que fueron obtenidos previamente en el Capítulo III (3.2.1 Fase 1. -Obtención de Información de contenido de humedad) se definieron las técnicas de detección correspondientes. En la figura 57 se puede observar la selección de los dos métodos más utilizados en el aserradero “San Martín” para la detección de humedad.

Figura 57

Selección del método de detección del nivel de humedad



Nota. En esta figura se muestra los métodos de detección de humedad que se utilizan en el aserradero “San Martín”.

A continuación, se explica los dos métodos que se seleccionan para la presente investigación: 1) Inspección visual, y 2) Método eléctrico.

1. Método 1 (Inspección visual). - Método que consiste en hacer un análisis visual de una muestra de madera de laurel, a través de características físicas como: color, textura, anillos de crecimiento y dimensión, permitiendo detectar el nivel de humedad de una muestra de madera laurel.
2. Método 2 (Método eléctrico). - Técnica que consiste en incrustar dos barras de metal en una muestra madera laurel, con el objetivo de medir la resistencia eléctrica que tiene dicha muestra para luego deducir el porcentaje de humedad que contiene la misma.

Para esta investigación se utilizan los dos métodos antes mencionados, con el propósito de formar un dataset con imágenes de muestras de madera laurel que permitan diferenciar el porcentaje de humedad que tengan dichas muestras.

3.2.3. Fase 3. Generación del dataset.

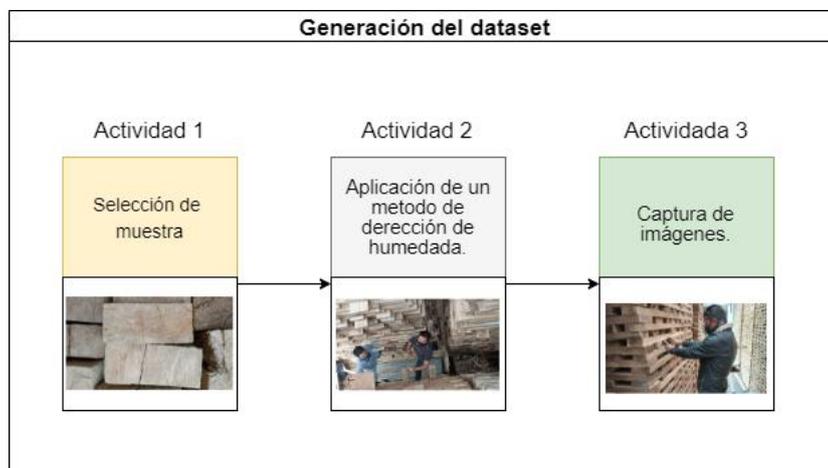
En esta fase se realiza la creación de un dataset de la madera de laurel, mediante la selección de muestras de la misma. Permitiendo escoger aquellas muestras que más se ajustan a las características de una técnica de detección de humedad. Posteriormente se aplican los 2 métodos de detección de humedad que fueron definidos en el Capítulo III (3.2.2 Fase 2.- Selección del método de detección del nivel de humedad) con la finalidad de averiguar el contenido de dicha humedad. Finalizando con la captura de sus imágenes, permitiendo el almacenamiento gráfico de dichos datos.

En la figura 58 se visualiza aquellas actividades que permiten la creación del dataset de madera laurel en función de su humedad, las mismas que fueron

plateados de la siguiente manera: 1) Selección de la muestra, 2) Aplicación de un método de detección de humedad y 3) Captura de imágenes.

Figura 58

Generación del dataset



Nota. En esta figura se muestra los pasos necesarios para crear un dataset.

A continuación, se muestra las actividades que intervienen en la generación del dataset.

1. Actividad 1 (Selección de muestra). - Consiste en escoger las mejores muestras de entre las varias existentes, con el fin de obtener especímenes de calidad.

Para esta investigación esta actividad consiste en discriminar muestras que no cumplan con las características físicas (anillos de crecimiento, dimensión, textura, color) mediante la supervisión de un experto en la madera laurel. Para lo cual se nos provee de un técnico en la madera por parte del aserradero "San Martín", quien proporciona las indicaciones necesarias para seleccionar dichas muestras, enfocándose en las peculiaridades visuales u ópticas.

Además, cabe destacar que dicho aserradero posee en su propiedad alrededor de 100.000 ejemplares de madera, las mismas que se

discriminaron hasta llegar a un total de 10.000 ejemplares en óptimas condiciones para la ejecución de la siguiente actividad.

2. Actividad 2 (Aplicación del método de detección de humedad). - Una vez escogidas las muestras que fueron definidas en la actividad 1 (Selección de una muestra) se procede a aplicar dos métodos de detección de humedad definido en el Capítulo III (3.2.2. Fase 2.- Selección del método de detección del nivel de humedad.), con el objetivo de conocer qué porcentaje de humedad posee dicha muestra. Para lo cual se necesitó la supervisión de un técnico en madera. Aplicándose dichos métodos los cuales se explican a continuación:

1. Método de inspección visual. - En la figura 59 se puede visualizar la aplicación de este método. El mismo que requirió necesario dirigirse al lugar donde se encontraban las parrillas de madera. Posteriormente se analiza las características físicas (anillos de crecimiento, textura, dimensión y color) que contiene cada muestra de madera a través del uso del sentido de la vista. Tomándose en cuenta el tiempo transcurrido en el secado de la madera. Facilitando la de detección de humedad en dicha muestra. Cabe recalcar que dichas muestras se utilizadas en la siguiente actividad(Actividad 3 – Captura de imágenes) permitiéndose generara el dataset 1.

Figura 59*Inspección visual*

Nota. En esta figura se muestra la aplicación del método de inspección visual.

2. Método eléctrico. - En la figura 60 se puede visualizar la aplicación de este método. El mismo que requirió obligatoriamente trasladarse al sitio donde se encontraban las parrillas de madera. Posteriormente se incrusta dos barras de metal en cada muestra de madera, permitiendo medir la resistencia eléctrica que posee dichas muestras, determinando su contenido de humedad. Cabe destacar que dichas muestras son utilizadas en la siguiente actividad (Actividad 3 – Captura de imágenes) permitiéndose formar el dataset 2.

Figura 60

Método eléctrico



Nota. En esta figura se muestra la aplicación del método eléctrico.

3. Actividad 3 (Captura de imágenes). – en esta actividad se utilizó el contenido de humedad identificado por cada muestra. Dichas muestras provienen de la actividad 2 (Aplicación del método de detección de humedad). Procediéndose a tomar fotografías a través de una cámara fotográfica o dispositivo móvil. Creando dos conjuntos de datos, dataset 1 a través del método de inspección visual y, dataset 2 mediante el método eléctrico. Cabe destacar que dichos métodos fueron definidos en el Capítulo III (3.2.2. Fase 2.- Selección del método de detección del nivel de humedad). Obteniéndose un total de 5000 imágenes por cada dataset (dataset 1 y dataset 2). Cabe recalcar que cada dataset debe tener las siguientes características fotográficas: distancia, resolución y vistas. Estableciéndose diferencias significativas en la calidad de imágenes entre los dos datasets. Como se pueden observar en la tabla 10.

Tabla 10*Prototipos del dataset*

Característica		Dataset 1	Dataset 2
Nombre	Descripción		
Distancia	Intervalo desde la cámara fotográfica hacia la muestra de madera.	10 cm	5 cm
Resolución de la imagen	Número de píxeles por pulgada que contiene una imagen.	4 píxeles	8 píxeles
Vistas	Proyecciones ortogonales de un objeto sobre los planos.	<ul style="list-style-type: none"> • Frontal 	<ul style="list-style-type: none"> • Frontal • Derecha • Izquierda • Posterior • Inferior

Nota. En esta tabla se muestra las características que posee un prototipo de dataset.

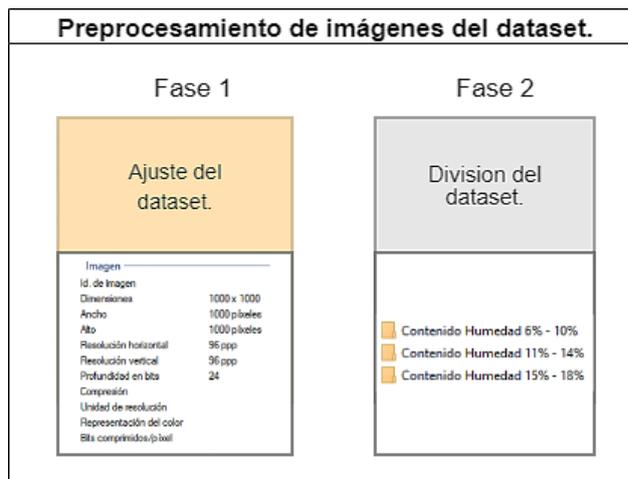
3.3. Etapa 2. Preprocesamiento de imágenes del dataset.

En esta etapa se realiza el preprocesamiento de imágenes de madera laurel que fueron obtenidos en el Capítulo III (3.2. Etapa 1. Recolección y captura de imágenes en función al nivel de humedad). A través de los procesos de tratamiento de imágenes (filtrado, limpieza y anotación) se descartan ilustraciones con fallas visuales y se les añade un nombre en función de las clases (seco, amorochado y chapoteado) e intervalos de porcentaje de humedad (6% - 10%, 11% - 14% y 15% - 18%), con el propósito de categorizar el dataset de madera de Laurel.

En la figura 61 se visualiza las fases que intervienen en el preprocesamiento de imágenes del dataset, las mismas que fueron plateados de la siguiente manera: 1) Ajuste del dataset y 2) División del dataset.

Figura 61

Pre procesamiento de imágenes del dataset



Nota. En esta figura se muestra las fases que se necesitan para el preprocesamiento de imágenes del dataset.

A continuación, se muestra las fases que intervienen en el preprocesamiento de imágenes del dataset.

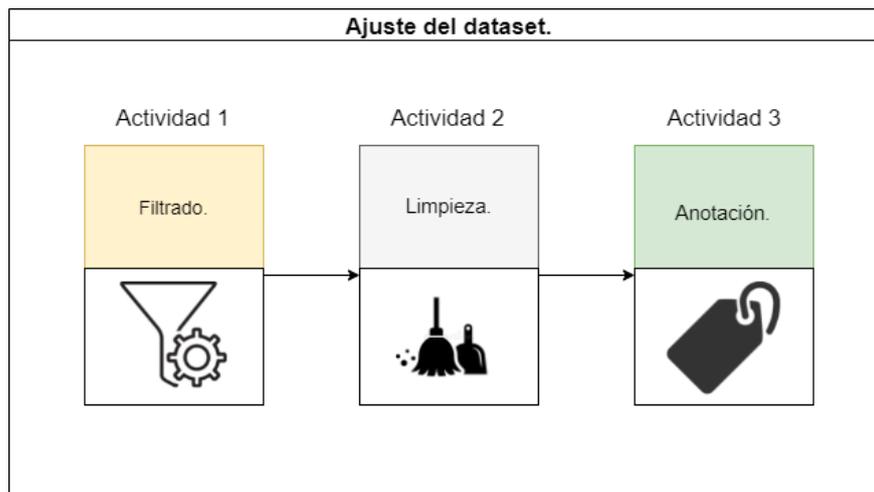
3.3.1. Fase 1. Ajuste del dataset.

En esta fase se realiza el ajuste del dataset de la madera Laurel. Mediante el filtrado de imágenes, permitiendo desechar aquellas que posean defectos visuales. También se efectúa la limpieza de dichas imágenes a través de su edición, cuando estas presentan fallos mínimos de visualización. Y finalmente la anotación que consiste en asignar un nombre a una ilustración.

En la figura 62 se visualiza las actividades que intervienen en el ajuste de un dataset de madera Laurel, las mismas que fueron planteados de la siguiente manera: 1) Filtrado, 2) Limpieza y 3) Anotación.

Figura 62

Ajuste del dataset



Nota. En esta figura se muestra los procesos que se sigue para ajustar un dataset.

A continuación, se muestra las tres actividades que intervienen en el ajuste de dataset (dataset1 y dataset 2).

1. Actividad 1 (Filtrado). – Aquí se descarta imágenes con defectos de visualización como: baja resolución, borrosidad y mala iluminación (Velásquez 2020). Permittedose la eliminación de imágenes de madera que poseen fallas visuales, mediante las indicaciones planteadas por el experto en la madera laurel. Discriminándose aquellas imágenes que tienen fallas ópticas, mediante el uso del sentido de la vista. Por esta razón llegándose a eliminar un 75% del total de las imágenes capturadas.
2. Actividad 2 (Limpieza). – Aquí se edita las imágenes con un error mínimo de visualización con el fin de no descartarlas (Velásquez 2020).
Corrigiendo fallos ópticos de imágenes de madera Laurel. Para lo cual los tesistas hacen uso de la herramienta Gimp para editar dichas imágenes, permitiendo eliminar errores visuales tales como: resolución, borrosidad y enfoque. Haciendo que estas imágenes sean aptas para construir un dataset.

3. Actividad 3 (Anotación). - Consiste en asignarle un nombre a una imagen dependiendo de la necesidad (Velásquez 2020). Para lo cual se asignó un nombre a una muestra de madera laurel, mediante la creación de una nomenclatura de escritura compuesta por: el número de imagen, el porcentaje de humedad y la extensión de dicha muestra (png, jpg, fif, etc). Obteniendo como resultado imágenes marcadas de la siguiente manera 1_11-14.png, con el propósito de distinguir a que clase pertenece cada una de las muestras de madera. Cabe recalcar que el nombre de dichas muestras son cambiantes, debido a que la estructura de la nomenclatura antes mencionada crea un nombres diferentes para cada una de las muestras, es decir son únicas e irremplazables.

Después de aplicar las tres actividades antes mencionadas (filtrado, limpieza y anotación) a los dos datasets (dataset 1 y dataset 2) de madera laurel, de 10.000 imágenes obtenidas en el aserradero “San Martin” se descartaron 7000 imágenes. Obteniéndose 1500 imágenes de calidad por cada dataset con sus correspondientes propiedades como se visualiza en la tabla 11. Cabe destacar que esto beneficiara a la red neuronal en una etapa posterior de entrenamiento debido a que se posee imágenes con características físicas perfectas en la entrada del proceso de convolución.

Tabla 11

Propiedades de una imagen.

Propiedades de una imagen	
Ancho	4096 pixeles
Alto	2304 pixeles
Profundidad	24 bytes (A color)
Nomenclatura de escritura	1_11-14.png, 2-11-14.png,

Nota. En esta tabla se muestra los requisitos mínimos que deben cumplir una imagen.

3.3.2. Fase 2. División del dataset.

En esta fase se realiza la división de los dos datasets (dataset 1 y dataset 2) que fueron obtenidos en el Capítulo III (3.3.1 Fase 1.- Ajuste del dataset). Mediante la creación de 3 clases: 1) seco, 2) amorochado, y 3) chapoteado. Terminología utilizada por los expertos madereros para establecer el contenido de humedad. Permittedose categorizar las imágenes de los dos datasets (dataset 1 y dataset 2).

En la figura 63 se visualiza las clases creadas en la división del dataset de madera Laurel, las mismas que fueron plateados de la siguiente manera: 1) Seco, 2) Amorochado y 3) Chapoteado.

Figura 63

División del dataset



Nota. En esta figura se muestra las clases en las que se dividió el dataset.

A continuación, se muestra las tres clases que se establecieron como etiquetas en los 2 datasets (dataset 1 y dataset 2). Que posteriormente permitirán categorizar las imágenes y así diagnosticar el contenido de humedad del laurel.

1. Clase 1 (Seco). - Conjunto de imágenes que contienen muestras de madera Laurel en un estado seco, es decir, que la muestras poseen una cantidad mínima agua en su estructura entre 6% y 10% de humedad (apreciación de expertos madereros).
2. Clase 2 (Amorachado). - Conjunto de imágenes que contienen muestras de madera Laurel en un estado semiseco, es decir, que la muestras poseen una cantidad pequeña de agua en su estructura entre 11% y 14% de humedad (apreciación de expertos madereros).
3. Clase 3 (Chapoteado). - Conjunto de imágenes que contienen muestras de madera Laurel en un estado chorreante, es decir, que la muestras poseen una cantidad considerable agua en su estructura entre 15% y 18% de humedad (apreciación de expertos madereros).

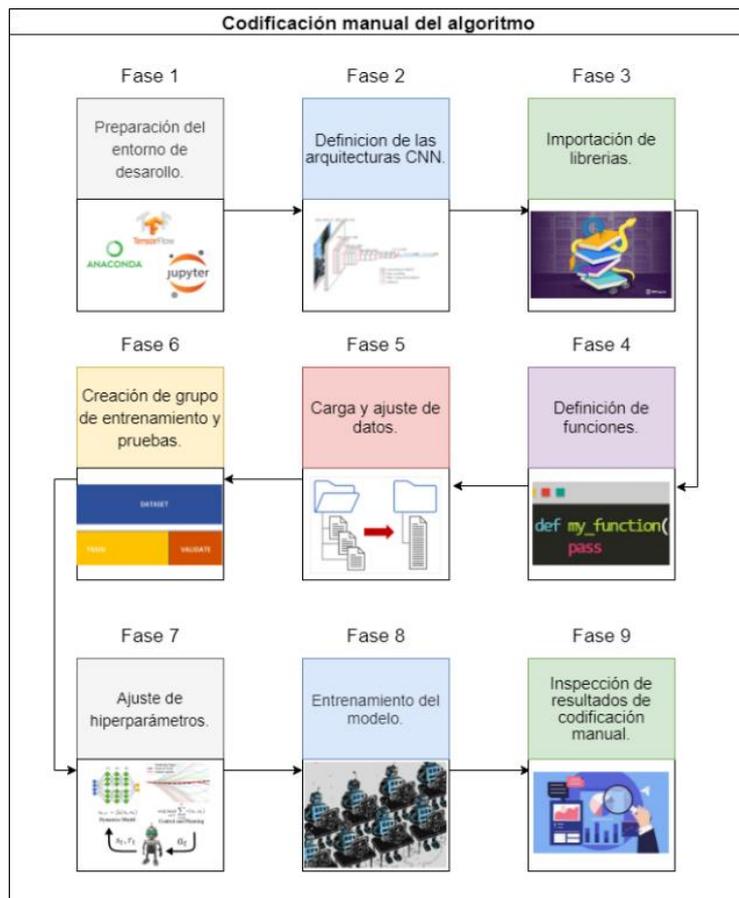
3.4. Etapa 3. Codificación manual del algoritmo.

En esta etapa se realiza la codificación manual del algoritmo para la clasificación de imágenes madera Laurel. A través de la preparación del entorno de desarrollo el cual permite la instalación software necesario para la creación de código fuente.

En la figura 64 se visualiza las fases que intervienen en la codificación manual del algoritmo para la clasificación de imágenes de madera Laurel, las mismas que fueron plateados de la siguiente manera: 1) Preparación del entorno de desarrollo, 2) Definición de las arquitecturas CNN, 3) Importación de librerías, 4) Definición de funciones, 5) Carga y ajuste de datos, 6) creación de grupos de entrenamiento y pruebas, 7) Ajuste de hiperparámetros, 8) Entrenamiento del modelo y 9) inspección de resultados de codificación manual.

Figura 64

Codificación manual del algoritmo



Nota. En esta figura se muestra las fases que intervendrán en la codificación del algoritmo.

A continuación, se muestra las 9 fases que intervienen en la codificación manual del algoritmo.

3.4.1. Fase 1. Preparación del entorno de desarrollo.

Fase que permite la preparación del entorno de desarrollo software, mediante la instalación de herramientas técnicas, en un ordenador que cumpla los siguientes requisitos mínimos de instalación: 1) Sistema Operativo de 64 bits, 2) Memoria ram 8gb y 3) Tarjeta de video Nvidia 4gb. Permitiendo el desarrollo del código manual en base al aprendizaje profundo.

En la figura 65 se visualiza el software requerido para la preparación del entorno de desarrollo. Utilizándose como lenguaje de programación Python(a), Anaconda como entorno de desarrollo integrado (IDE)(b) y librería Tensorflow(c).

Figura 65

Preparación del entorno de desarrollo



Nota. En esta figura se muestra los programas necesarios para el desarrollo del algoritmo.

A continuación, se detallan los recursos que debe tener el entorno de trabajo.

a) Python. - Denominado un lenguaje de programación multiparadigma, es decir que soporta varias formas de programación tales como: orientación a objetos, imperativa, etc. (Programo Ergo Sum 2021). Para esta investigación se descarga e instala la versión 3.6 de 64 bits. Permitiendo acceder a múltiples librerías entre las más importantes: numpy, os, cv2, itertools, matplotlib tensorflow, etc.

b) Anaconda. - Distribución de Python de código abierto que permite realizar operaciones de ciencia de datos y de aprendizaje automático tales como: el procesamiento de grandes volúmenes de información, análisis predictivo y

cómputos científicos (Toro 2017). Para esta investigación se descargó e instaló esta plataforma de forma local en su versión 2.0 de 64 bits, esta versión incluye Jupyter que es la interfaz gráfica de entorno de desarrollo y Anaconda Navigator que es el gestor de paquetes.

c) Tensorflow. - Es una biblioteca de código abierto de extremo a extremo para el aprendizaje automático, desarrollado por Google para satisfacer sus necesidades de sistemas capaces de construir y entrenar redes neuronales (TensorFlow 2021). Para esta investigación se instaló esta librería en su versión 2.0 ya que es la más estable al momento de desarrollar código fuente.

3.4.2. Fase 2. Definición de las arquitecturas de redes neuronales convolucionales (CNN).

En esta fase se definen las arquitecturas para modelos de redes neuronales convolucionales (CNN), obteniéndose dos arquitecturas mediante la revisión sistemática en el Capítulo II (2.2. Revisión sistemática). Dichas arquitecturas fueron modificadas para poder dar solución a la presente investigación (diagnóstico de contenido de humedad del laurel). Generando nuestras propias arquitecturas de redes neuronales convolucionales (CNN) a las cuales las denominamos de la siguiente manera: 1) Wood y 2) Forest.

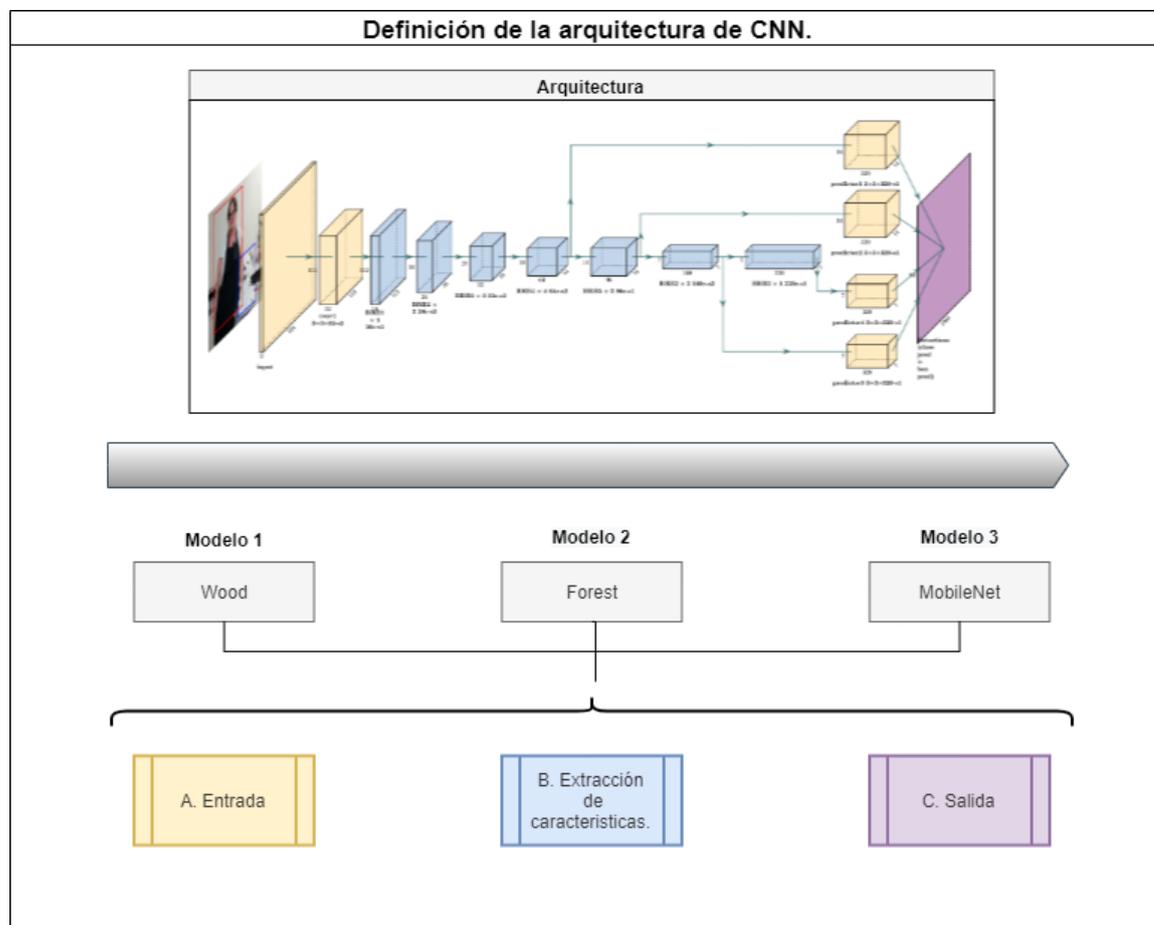
Además, es necesario recalcar que se replicó la arquitectura denominada MobileNet en el modelo 3, dicha arquitectura está diseñado para usarse en aplicaciones móviles. Permittiéndose ajustarse perfectamente a la problemática de investigación (diagnóstico de contenido de humedad del laurel).

En esta fase se definió varias arquitecturas de redes neuronales convolucionales a través del conocimiento y experimentación. Proponiéndose como se observa en la Figura 66 las siguientes modelos: 1) Modelo 1

(Implementación de la arquitectura Wood), 2) Modelo 2 (Implementación de la arquitectura Forest) y 3) Modelo 3 (Implementación de la arquitectura MobileNet).

Figura 66

Definición de la arquitectura de CNN



Nota. En esta figura se muestra las partes que conforma un modelo de red neuronal convolucional.

3.4.2.1. Modelo 1 (Implementación arquitectura Wood).

El modelo 1 está basado en la arquitectura Wood propuesta por los integrantes de esta investigación, cumple con una de las tareas de visión por computadora móviles, denominada clasificación de imágenes. Consiste en extraer características para identificar un tipo de imagen. Wood está b en redes neuronales convolucionales (CNN) que utilizan de múltiples capas, las cuales

actúan como filtros que poseen una o más dimensiones. Dichas capas forman un bloque de convolución.

Cabe destacar que cada uno de estos bloques de convolución disminuyen su dimensión, para aumentar la complejidad de captura de características. Por ejemplo, el bloque 1 permite detectar líneas y curvas, el bloque 2 además de la líneas y curvas añade colores hasta llegar al último bloque en este caso el bloque 3. Es importante destacar que la complejidad de la captura de características depende del número de filtros que se utilice una capa convolucional, es decir que, entre más filtros se utilice mayor es la capacidad de detección de peculiaridades. Aumentado su requerimiento computacional.

Un bloque de convolución se estructura por: dimensión, capas convolucionales (Conv2D, Maxpooling, Flatten, Dropout y Dense) y una función de activación (Relu, Softmax), que en combinación permiten formar la arquitectura Wood, la misma que se visualiza en la tabla 12.

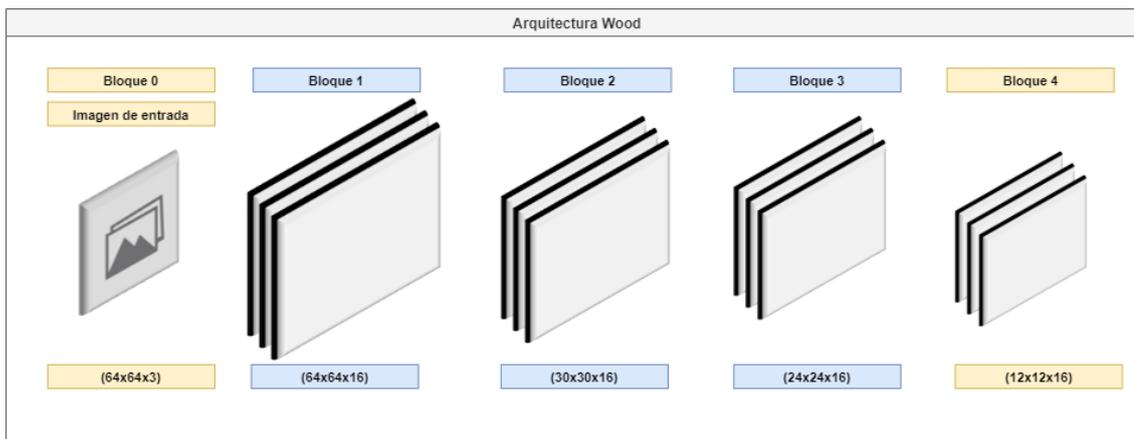
Tabla 12

Arquitectura Wood del Modelo 1

Bloque de la arquitectura Wood (Operación de convolución)								
N. º	Dimensió n	Capas Convolucionales					Función de activación	
		Conv2 D	Maxpoolin g	Flatte n	Dropou t	Dens e	Rel u	Softma x
1	(64x64x3)	2	1	0	0	0	1	0
2	(30x30x16)	1	1	0	0	0	1	0
3	(24x24x16)	1	1	0	0	0	1	0
4	(12x12x16)	0	0	1	1	1	0	1

Nota. Esta tabla contiene los componentes que forman la arquitectura Wood.

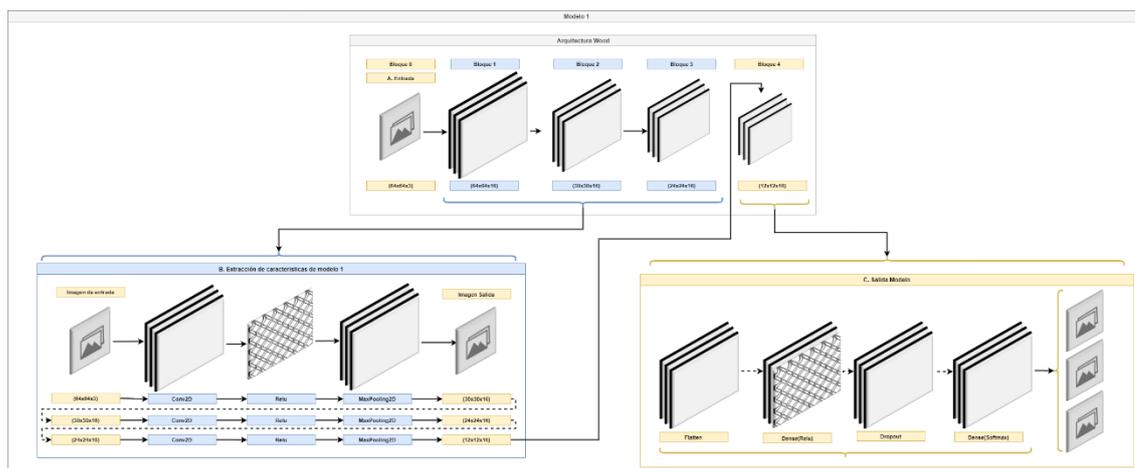
En la figura 67 se visualiza la dimensión de los bloques convolucionales obtenidos al utilizar la arquitectura Wood propuesta en esta investigación. Dichas dimensiones son las siguientes: bloque 1 64x64x16, bloque 2 30x30x16, bloque 3 24x24x16 y bloque 4 12x12x16.

Figura 67**Arquitectura Wood**

Nota. En esta figura se visualiza la dimensión de los bloques de convolución de la arquitectura Wood.

Wood, se caracteriza por utilizar en la capa de entrada imágenes de pequeña magnitud en comparación a otras arquitecturas. Consumiendo una cantidad reducida de recursos del ordenador y ser relativamente rápida en el entrenamiento de un modelo. Debido a que la extracción de características en las neuronas convolucionales (CNN) reducen a una mínima cantidad la dimensión del muestreo. Permitiendo que el reconocimiento de patrones tenga una complejidad simple.

En la Figura 68 se observa las etapas necesarias para la implementación de la arquitectura Wood (Modelo 1). Dichas etapas son las siguientes: A) entrada, B) extracción de características (capas ocultas) y C) salida.

Figura 68*Implementación de la Arquitectura Wood en el Modelo 1*

Nota. En esta figura se visualiza la composición de la arquitectura Wood en el modelo 1.

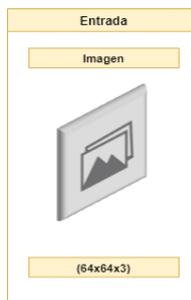
A continuación, se explica brevemente las etapas para la implementación de la arquitectura Wood en el modelo 1.

1. Entrada del modelo 1.

En la figura 69 se visualiza el ingreso de una imagen con sus respectivas dimensiones. Es decir, la entrada o capa sensorial de esta arquitectura permite el ingreso de imágenes a color de tres canales RGB (Red, Green, Blue) con un tamaño 64x64 (alto x ancho). Cabe destacar que estos canales RGB corresponden a la profundidad de la imagen es decir 64x64x3 (alto x ancho x profundidad).

Figura 69

Entrada del Modelo 1



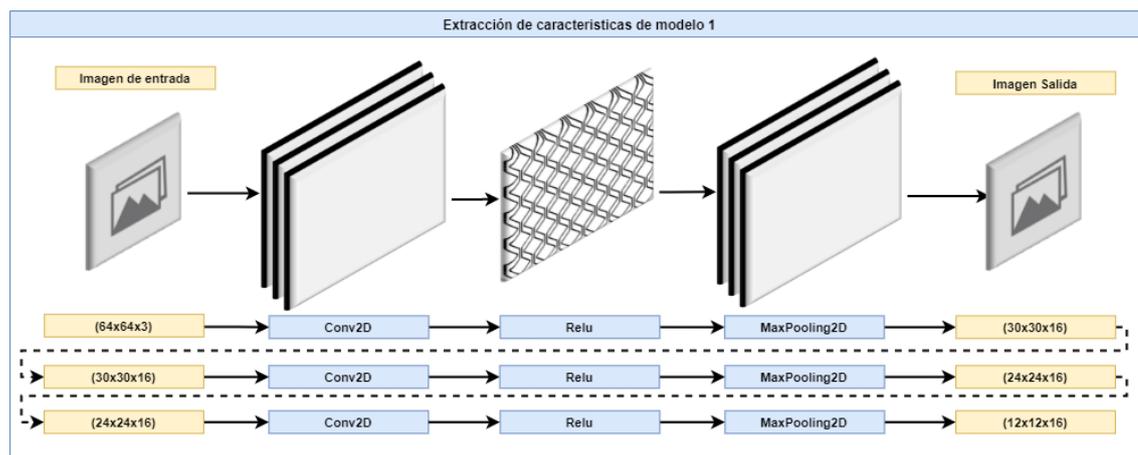
Nota. En esta figura se visualiza la entrada del modelo 1.

2. Extracción de características del modelo 1.

En la figura 70 se visualiza el proceso de convolución mediante la extracción de características a través de capas de convolución (Conv2D y Maxpooling) y una función de activación (Relu)

Figura 70

Extracción de características modelo 1



Nota. En esta figura se visualiza la extracción de características del modelo 1.

La extracción de características en esta arquitectura hace el uso de capas convolucionales (CNN) que suelen constar de más de un filtro. Dichos filtros manejan bloques de convolución, permitiendo generar un mapa de características tridimensional. Se sabe que un mapa de características es tridimensional cuando se utiliza una imagen con una escala a color.

El mapa de características tridimensional se crea en la salida de la capa Conv2D mediante una función de activación Relu, que será igual al número de filtros dicha capa. Entonces la salida de dicha capa formaría parte de una nueva imagen cuyo número de canales (profundidad de la imagen) es igual al número de filtros de la dicha capa, se reduce la dimensión de dicha imagen mediante la capa Maxpooling2D, dando lugar a un bloque de convolución. Esta nueva imagen se introduce en el nuevo bloque de convolución, formando parte de la entrada de la siguiente capa Conv2D, sabiendo que a esta capa se le suma un término independiente (filtros) y una función de no linealidad (función de activación) que dará lugar a lo que llamamos mapa de características.

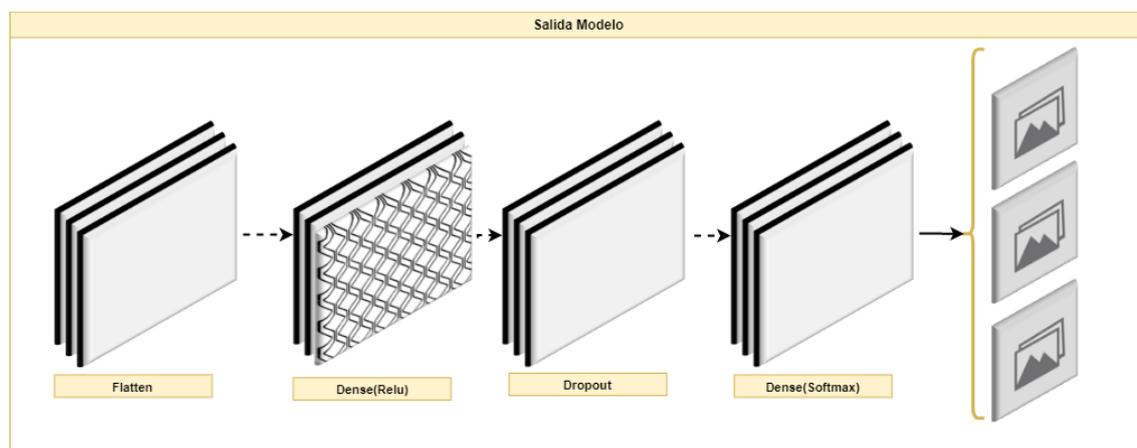
3. Salida del modelo 1.

La salida en esta arquitectura hace el uso del último proceso de convolución mediante las capas de convolución: flatten, dense, dropout. Generando neuronas de salida que utiliza una función de activación (softmax) para predecir cada clase obtenida en el Capítulo III (3.3.2 División del Dataset).

En la figura 71 se visualiza el último proceso de convolución para determinar una clase (seco, amorocho, chapoteado).

Figura 71

Salida Modelo 1



Nota. En esta figura se visualiza las salidas del modelo 1.

3.4.2.2. Modelo 2 (Implementación arquitectura Forest).

El modelo 2 está basada en la arquitectura Forest propuesta por los integrantes de esta investigación, cumple con una de las tareas de visión por computadora móviles, denominada clasificación de imágenes. Consiste en extraer características para identificar un tipo de imagen. Forest está basada en redes neuronales convolucionales (CNN) que radican en múltiples capas, las cuales actúan como filtros que poseen una o más dimensiones. Dichas capas forman un bloque de convolución.

Cabe destacar que cada uno de estos bloques de convolución disminuyen su dimensión, para aumentar la complejidad de captura de características. Por ejemplo, el bloque 1 permite detectar líneas y curvas, el bloque 2 además de la líneas y curvas añade colores hasta llegar al último bloque en este caso el bloque 3. Es importante destacar que la complejidad de la captura de características depende del número de filtros que se utilice una capa convolucional, es decir que, entre más filtros se utilice mayor es la capacidad de detección de peculiaridades. Aumentado su requerimiento computacional.

Un bloque de convolución se estructura por: dimensión, capas convolucionales (Conv2D, Maxpooling, Flatten, Dropout y Dense) y una función de activación (Relu, Softmax), que en combinación permiten formar la arquitectura Forest, la misma que se visualiza en la tabla 13.

Tabla 13

Arquitectura Forest del Modelo 2.

Bloque de la arquitectura Forest (Operación de convolución)								
N. o	Dimensión	Capas Convolucionales					Función de activación	
		Conv2 D	Maxpoolin g	Flatte n	Dropo u	Dens e	Rel u	Softma x
1	(200x200x16)	2	1	0	0	0	1	0
2	(118x118x32)	1	1	0	0	0	1	0
3	(52x52x64)	1	1	0	0	0	1	0
4	(26x26x128)	0	0	1	1	1	0	1

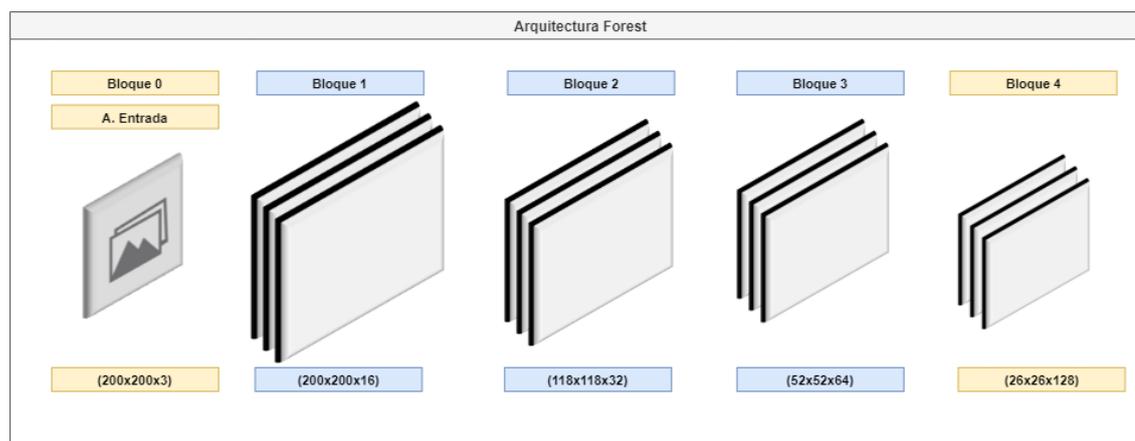
Nota. Esta tabla contiene los componentes que forman la arquitectura Forest.

En la figura 72 se visualiza la dimensión de los bloques convolucionales obtenidos al utilizar la arquitectura Forest propuesta en esta investigación.

Dichas dimensiones son las siguientes: bloque 1 200x200x16, bloque 2 118x118x32, bloque 3 52x52x64 y bloque 4 28x28x128.

Figura 72

Arquitectura Forest.



Nota. En esta figura se visualiza la dimensión de los bloques de convolución de la arquitectura Forest.

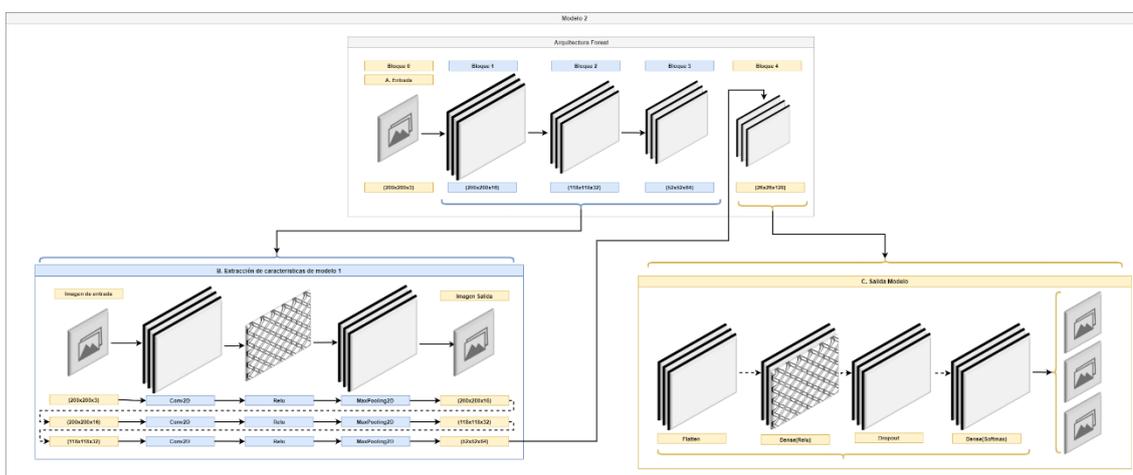
Forest, se caracteriza por utilizar en la capa de entrada imágenes de magnitud grande similar a la arquitectura MobileNet, consumir una cantidad considerable de recursos del ordenador y tomar una cantidad alta de tiempo en

el entrenamiento de un modelo, esto se debe a que en la extracción de características las neuronas convolucionales reducen notablemente la dimensión del muestreo haciendo que el reconocimiento de patrones tenga una complejidad media.

En la Figura 73 se visualiza la arquitectura Forest del modelo 2 la cual consta de las siguientes partes: A) entrada, B) extracción de características (capas ocultas) y C) salida.

Figura 73

Implementación de la Arquitectura Forest en el Modelo 2



Nota. En esta figura se visualiza la composición de la arquitectura del modelo 2.

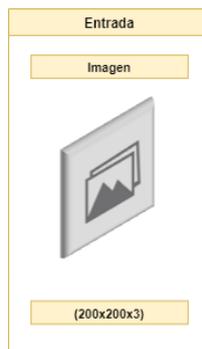
A continuación, se explica brevemente las etapas para la implementación de la arquitectura Forest en el modelo 2.

1. Entrada del modelo 2.

En la figura 74 se visualiza el ingreso de una imagen con sus respectivas dimensiones. Es decir, la entrada o capa sensorial de esta arquitectura permite el ingreso de imágenes a color de tres canales RGB (Red, Green, Blue) con un tamaño 200x200 (alto x ancho). Cabe destacar que estos canales RGB corresponden a la profundidad de la imagen es decir 200x200x3 (alto x ancho x profundidad).

Figura 74

Entrada Modelo 2



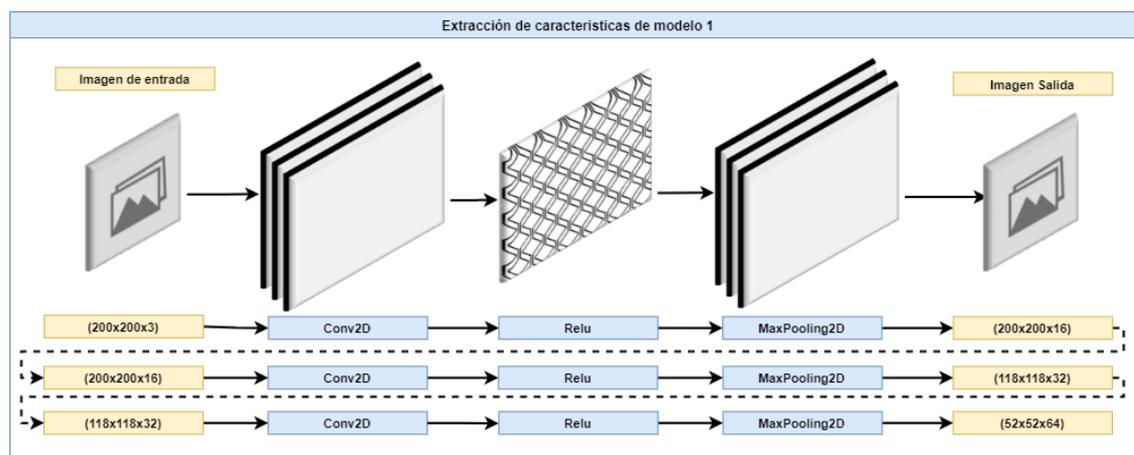
Nota. En esta figura se visualiza la entrada del modelo 2.

2. Extracción de características del modelo 2.

En la figura 75 se visualiza el proceso de convolución mediante la extracción de características a través de una capa de convolución (Conv2D, Maxpooling) y una función de activación (Relu).

Figura 75

Extracción de características Modelo 2



Nota. En esta figura se visualiza la extracción de características del modelo 2.

La extracción de características en esta arquitectura hace el uso de capas convolucionales (CNN) que suelen constar de más de un filtro. Dichos filtros manejan bloques de convolución, permitiendo generar un mapa de

características tridimensional. Se sabe que un mapa de características es tridimensional cuando se utiliza una imagen con una escala a color.

El mapa de características tridimensional se crea en la salida de la capa Conv2D mediante una función de activación Relu, que será igual al número de filtros dicha capa. Entonces la salida de dicha capa formaría parte de una nueva imagen cuyo número de canales (profundidad de la imagen) es igual al número de filtros de la dicha capa, se reduce la dimensión de dicha imagen mediante la capa Maxpooling2D, dando lugar a un bloque de convolución. Esta nueva imagen se introduce en el nuevo bloque de convolución, formando parte de la entrada de la siguiente capa Conv2D, sabiendo que a esta capa se le suma un término independiente (filtros) y una función de no linealidad (función de activación) que dará lugar a lo que llamamos mapa de características.

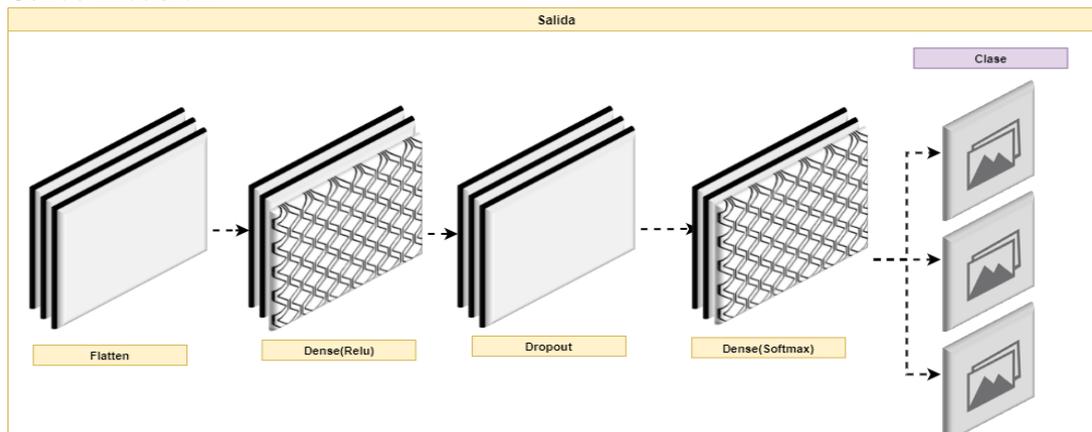
3. Salida del modelo 2.

La salida en esta arquitectura hace el uso del último proceso de convolución mediante las capas de convolución: flatten, dense, dropout y la función de activación softmax, debido a que dicho proceso genere una neurona de salida que utiliza una función de activación (softmax) para predecir cada clase obtenida en el Capítulo III (3.3.2 División del Dataset).

En la figura 76 se visualiza el último proceso de convolución para determinar una clase (seco, amorchado, chapoteado).

Figura 76

Salida Modelo 2



Nota. En esta figura se visualiza la salida de la arquitectura del modelo 2.

3.4.2.3. Modelo 3 (Implementación arquitectura MobileNet).

El modelo 3 está basada en la arquitectura MobileNet propuesta por Google, para tareas de visión por computadora móviles denomina clasificación de imágenes. MobileNet se basa en una arquitectura simplificada que utiliza convoluciones separables en profundidad, para construir redes neuronales profundas y ligeras, es decir, la capa convolucional original se divide en dos capas convolucionales, donde cada filtro de la capa convolucional anterior solo convoluciona con cada canal de la entrada, y luego la última capa convolucional es responsable de la combinación, dichas convoluciones separables forman un bloque convolucional.

Cabe destacar que cada uno de estos bloques de convolución disminuyen su dimensión, para aumentar la complejidad de captura de características. Por ejemplo, el bloque 1 permite detectar líneas y curvas, el bloque 2 además de la líneas y curvas añade colores hasta llegar al último bloque en este caso el bloque 5. Es importante destacar que la complejidad de la captura de características depende del número de filtros que se utilice una capa convolucional, es decir que, entre más filtros se utilice mayor es la capacidad de detección de peculiaridades. Aumentado su requerimiento computacional.

Un bloque de convolución se estructura por: dimensión, capas convolucionales (Conv2D, Maxpooling, Flatten, Dropout y Dense) y una función de activación (Relu, Softmax), que en combinación permiten formar la arquitectura Forest, la misma que se visualiza en la tabla 14.

Tabla 14

Arquitectura MobileNet del modelo 3

Bloque de la arquitectura MobileNet (Operación de convolución)										
N.º	Dimensión	Capas Convolucionales							Función de activación	
		Conv 2D	Zero Padding 2d	Depthwise Conv2D	Batch Normalization	Flatten	Dropout	Dense	Relu	Softmax
1	(224x224x3)	2	1	1	2	0	0	0	0	0
2	(112x112x16)	1	1	1	2	0	0	0	1	0
3	(56x56x32)	1	1	1	2	0	0	0	1	0
4	(28x28x64)	1	1	1	2	0	0	0	1	0
5	(14x14x128)	1	1	1	2	0	0	0	1	0
6	(7x7x256)	0	0	0	0	1	1	1	0	1

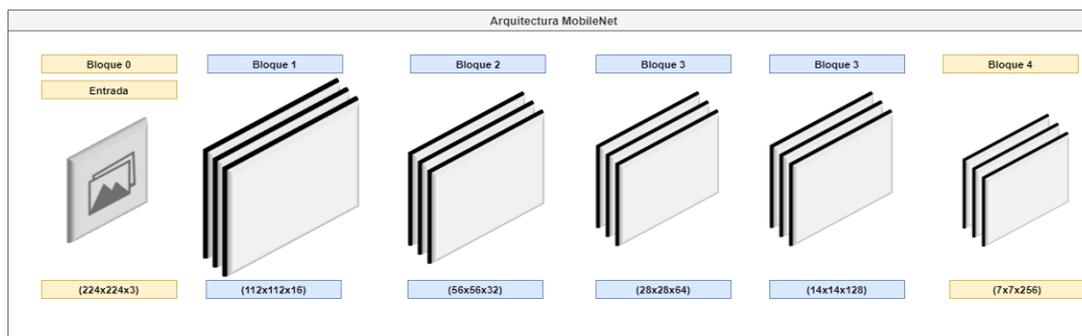
Nota. Esta tabla contiene las partes que componen a la arquitectura del modelo 3.

En la figura 77 se visualiza la dimensión de los bloques convolucionales obtenidos al utilizar la arquitectura MobileNet propuesta en esta investigación.

Dichas dimensiones son las siguientes: bloque 1 112x112x16, bloque 2 56x56x32, bloque 3 28x28x64, bloque 4 14x14x128 y bloque 5 7x7x256.

Figura 77

Arquitectura MobileNet.



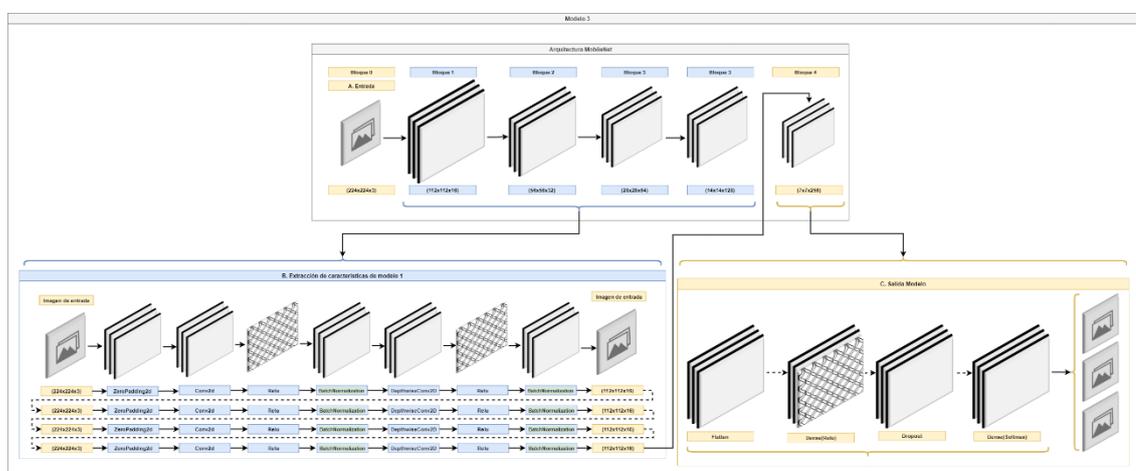
Nota. En esta figura se visualiza la dimensión de los bloques de convolución de la arquitectura Forest.

MobileNet, se caracteriza por utilizar en la capa de entrada imágenes de magnitud grande similar a la arquitectura Forest, consumir todos los recursos del ordenador y tomar una cantidad excesiva de tiempo en el entrenamiento de un modelo, esto se debe a que en la extracción de características las neuronas convolucionales reducen al máximo la dimensión del muestreo haciendo que el reconocimiento de patrones tenga una complejidad alta.

En la Figura 78 se visualiza la arquitectura Wood del modelo 3 la cual consta de las siguientes partes: A) entrada, B) extracción de caracterizas (capas ocultas) y C) salida.

Figura 78

Implementación de la Arquitectura MobileNet en el Modelo 3.



Nota. En esta figura se visualiza la composición de la arquitectura del modelo 3.

A continuación, se explica brevemente las etapas para la implementación de la arquitectura MobileNet en el modelo 3.

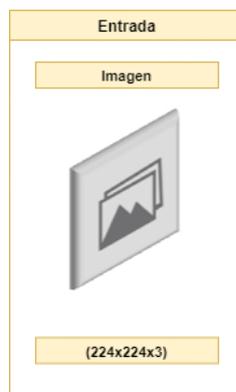
1. Entrada del modelo 3.

En la figura 79 se visualiza el ingreso de una imagen con sus respectivas dimensiones. Es decir, la entrada o capa sensorial de esta arquitectura permite el

ingreso de imágenes a color de tres canales RGB (Red, Green, Blue) con un tamaño 224x224(alto x ancho). Cabe destacar que estos canales RGB corresponden a la profundidad de la imagen es decir 224x224x3 (alto x ancho x profundidad).

Figura 79

Entrada modelo 3



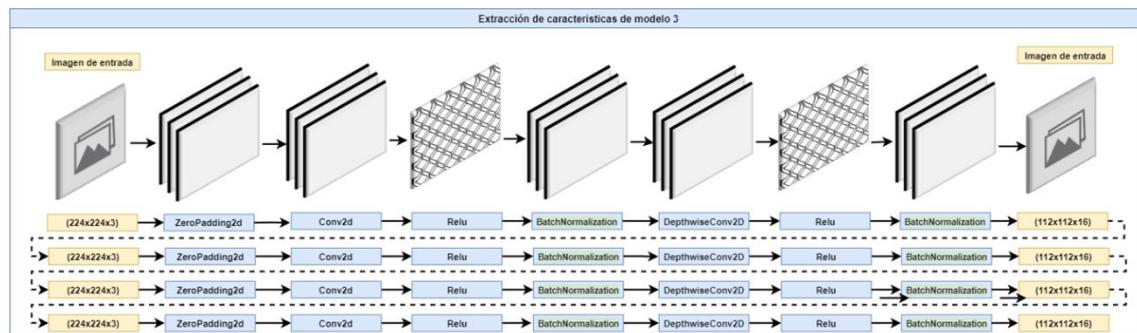
Nota. En esta figura se visualiza la entrada del modelo 3.

2. Extracción de características del modelo 3.

En la figura 80 se visualiza el proceso de convolución mediante la extracción de características (convolución separable) a través de la convolución por profundidad (DepthwiseConv2D y BatchNormalization) con su función de activación (Relu) y la convolución por punto (Conv2D, ZeroPadding2D y BatchNormalization) con su función de activación (Relu).

Figura 80

Extracción de características modelo 3



Nota. En esta figura se visualiza la extracción de características del modelo 3.

La extracción de características en esta arquitectura hace el uso de la convolución separable en profundidad, consiste en dos operaciones de convolución: la convolución en profundidad, y la convolución por punto, generando un mapa de características tridimensional por cada una de estas.

La convolución en profundidad genera el primer mapa de características tridimensional en la salida de la capa Conv2D, mediante una función de activación Relu, dicho mapa de características será igual al número de filtros dicha capa. Entonces la salida de dicha capa formaría parte de una nueva imagen cuyo número de canales (profundidad de la imagen) es igual al número de filtros de la dicha capa, reduciendo la dimensión de dicha imagen mediante la capa BatchNormalization.

La convolución por punto forja el segundo mapa de características haciendo uso de la nueva imagen como entrada en la capa DepthwiseConv2D, la cual mediante la función de activación Relu, dicho mapa de características será igual al número de filtros dicha capa. Entonces la salida de dicha capa formaría parte de una nueva imagen cuyo número de canales (profundidad de la imagen) es igual al número de filtros de la dicha capa, reduciendo la dimensión

de dicha imagen mediante la capa BatchNormalization, dando lugar a un bloque de convolución.

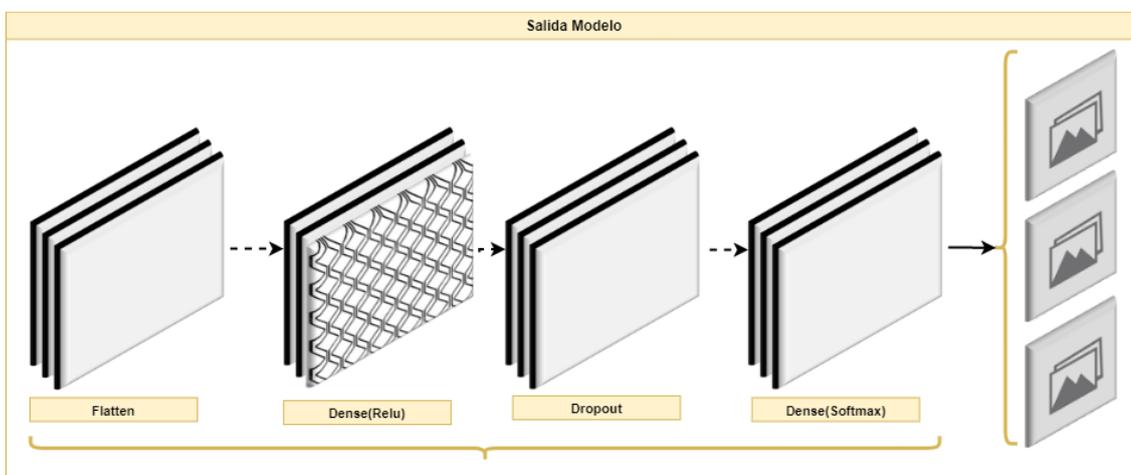
3. Salida del modelo 3.

La salida en esta arquitectura hace el uso del último proceso de convolución mediante las capas de convolución: flatten, dense, dropout y la función de activación softmax, debido a que dicho proceso genere una neurona de salida que utiliza una función de activación (softmax) para predecir cada clase obtenida en el Capítulo III (3.3.2 División del Dataset).

En la figura 81 se visualiza el último proceso de convolución para determinar una clase (seco, amorchado y chapoteado).

Figura 81

Salida Modelo 3



Nota. En esta figura se visualiza la salida del modelo 3.

3.4.3. Fase 3. Importación de librerías.

Esta fase se realiza la importación librerías en el código manual, con el propósito de acceder a funcionalidades que facilitarían la elaboración de este.

Para lo cual se crearon los siguientes cuatro tipos de librerías: tensorflow,

modelo, ayuda y métricas. Permitiendo categorizar los paquetes que intervienen el código manual.

En la Figura 82 se visualiza los tipos de librerías que intervienen en la codificación manual, las mismas que fueron plateadas de la siguiente manera: A) Librerías tensorflow, B) Librerías de modelo, C) Librerías de ayuda y D) Librerías para métricas.

Figura 82

Librerías del código manual

```
# A) Librerías TensorFlow
import keras.backend as K
from tensorflow.python.keras.optimizers import TFOptimizer
from keras.optimizers import SGD
from keras.callbacks import TensorBoard, ModelCheckpoint
from keras.utils import np_utils
from keras.models import load_model
# B) Librerías de modelo
from keras.models import Sequential
from keras.layers import Flatten, Input, Add, Dense,
Activation, ZeroPadding2D, BatchNormalization, Flatten,
Conv2D, AveragePooling2D, MaxPooling2D, GlobalMaxPooling2D,
Dropout
# C) Librerías de ayuda
import cv2
import numpy as np
import os
import itertools
import matplotlib.pyplot as plt
# D) Librerías para métricas
from sklearn.metrics import confusion_matrix, f1_score,
roc_curve, precision_score, recall_score, accuracy_score,
roc_auc_score
from sklearn import metrics
from mlxtend.plotting import plot_confusion_matrix
```

Nota. Esta esta figura se muestra las librerías que se utilizaron para elaborar el código manual.

A continuación, se muestra los tipos de librerías que se crearon en el código manual.

1. Tensorflow. - Importación de funcionalidad correspondiente a la manipulación de tareas de visión por computadora en este caso clasificación de imágenes (tensorflow y keras).

2. Modelo. - Importación de funcionalidad destinada a la construcción del modelo a través de capas de convolución (Flatten, Input, Add, Dense, Activation, ZeroPadding2D, BatchNormalization, Flatten, Conv2D, AveragePooling2D, MaxPooling2D, GlobalMaxPooling2D y Dropout).
3. Ayuda. - Importación de funcionalidad perteneciente a cargar y ajustar datos (dataset de madera Laurel), de acuerdo con los requerimientos de cada arquitectura propuesta en el Capítulo III (3.4.2. Fase 2. - Definición de las arquitecturas de CNN).
4. Métricas. - Importación de funcionalidad concerniente al cálculo de las métricas de evaluación (accuracy, matriz de confusión, precisión, recall y f1).

3.4.4. Fase 4. Definición de funciones.

En esta fase se realiza en la construcción de funciones que permitan disminuir la complejidad del código fuente. Para lo cual se crearon las siguientes tres funciones: grafico, modelo y métrica. Permitiendo dividir el código manual en pequeñas porciones, de tal manera que se reduzca la dificultad de dicho código facilitando la codificación del algoritmo de clasificación de imágenes.

En la Figura 83 se visualiza las funciones que intervienen en la codificación manual, las mismas que fueron plateadas de la siguiente manera: A) Grafico, B) Modelo, y C) Métrica.

Figura 83

Definición de funciones.

```
#Definir Funciones
# A) Graficos
def plotTraining(hist, epochs, typeData):
# B) Modelo
def modelConvolutionalNeuralNetwork():
# C) Metricas
def metric(y_real,y_pred, digits):
```

Nota. Esta esta figura se muestra las funciones que se utilizaron para elaborar el código manual.

A continuación, se muestra las tres funciones que se crearon en el código manual:

1. Gráfico. - Función correspondiente a la construcción de gráficos de resultados en base al número de épocas y la métrica de evaluación.
2. Modelo. - Función destinada a la construcción del modelo de red neuronal convolucional en base a las arquitecturas propuestas en el Capítulo III (3.4.2. Fase 2 Definición de las arquitecturas de CNN).
3. Métricas. - Función perteneciente a la representación y cálculo de resultados de cada métrica propuesta en el Capítulo III (3.4.3 Fase 3 Literal D. – Métricas).

3.4.5. Fase 5. Carga y ajuste de datos.

En esta fase se realiza la carga y ajuste de datos del dataset de madera Laurel mediante las librerías definidas en el Capítulo III (3.4.3 Fase 3 Literal C. – librerías de ayuda). Para lo cual se dio la lectura a las imágenes de madera laurel pertenecientes a una clase definida en el Capítulo III (3.3.2. División del Dataset), mediante la copia de la ruta donde se alojan dichas imágenes. Consecuentemente se ajusta la dimensión de dichas imágenes en base a los requerimientos de las arquitecturas propuestas en el Capítulo III (3.4.2 Fase 2.- Definición de las arquitecturas de CNN)

En la figura 84 se observa el ajuste y carga de datos de los dos datasets (dataset1, dataset2) que contienen muestras de madera laurel.

Figura 84

Carga y ajuste de datos

```
#Imágenes Contenido de Humedad 6%-10%
#Imágenes Pruebas
ch_06_10_folder_test_path="C:/Users/Work/DataSet_Madera/Contenido Humedad 6% - 10%/test_image"
ch_06_10_test=[]
img_size=224
for img in os.listdir(ch_06_10_folder_test_path):
    img = cv2.imread(os.path.join(ch_06_10_folder_test_path,img))
    #img_gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    img_resize = cv2.resize(img,(img_size,img_size))
    ch_06_10_test.append(img_resize)
```

Nota. En esta figura se muestra la forma en que se carga y se ajusta los datos en el código manual.

3.4.6. Fase 6. Creación de grupos de entrenamiento y pruebas.

En esta fase se realiza en la creación de grupos de entrenamiento y pruebas que permitan dividir el dataset de madera que fue generado en el Capítulo III (3.2.3 Fase 3 Generación del Dataset). Para lo cual se establecieron dos grupos de datos: entrenamiento y prueba. Permitiendo dividir las muestras de madera laurel que contienen los dos datasets en una forma porcentual, de tal manera que el 85% de los dos datasets pertenecieron al grupo de entrenamiento, mientras que el 15% restante se asignó al grupo de pruebas.

En la Figura 85 se visualiza las dos categorías creadas para los modelos de redes neuronales convolucionales (CNN), las mismas que fueron plateadas de la siguiente manera: A) Grupo de entrenamiento y B) Grupo de prueba.

Figura 85

Creación de grupos de entrenamiento y pruebas.

```
# A) Grupo de entrenamiento
#Unir e Imprimir el numero total de imagenes del dataset
images_train = np.concatenate([ch_06_10_train,ch_11_14_train,ch_15_18_train])
print(len(images_train))
Images_Train = np.array(images_train)
print(Images_Train.shape)
# B) Grupo de prueba
#Unir e Imprimir el numero total de imagenes del dataset
images_test = np.concatenate([ch_06_10_test,ch_11_14_test,ch_15_18_test])
print(len(images_test))
Images_Test = np.array(images_test)
print(Images_Test.shape)
```

Nota. En esta figura se muestra la creación de grupo de entrenamiento y pruebas para el código manual.

A continuación, se muestra los grupos creados en el código manual.

1. Grupo de entrenamiento. - Grupo correspondiente a la fase de entrenamiento de un modelo de red neuronal convolucional, que permite el almacenar datos (muestras de madera Laurel) de tipo entrenamiento mediante la fase 5 (Carga y ajuste de datos).
2. Grupo de prueba. - Grupo correspondiente a la fase de pruebas de un modelo de red neuronal convolucional, que permite el almacenar datos (muestras de madera Laurel) de tipo prueba mediante la fase 5 (Carga y ajuste de datos).

3.4.7. Fase 7. Ajuste de Hiperparámetros.

En esta fase se efectúa el ajuste de hiperparámetros de un modelo de red neuronal convolucional. Para lo cual se modifican los valores por defecto de dichos hiperparámetros tales como: entrada, época, tamaño de lote y tasa de aprendizaje. Con el objetivo de obtener los mejores resultados posibles en un modelo de red neuronal convolucional (CNN).

En la Figura 86 se visualiza el ajuste de hiperparámetros en la codificación manual, las mismos que fueron plateados de la siguiente manera: A) Entrada, B) Tasa de aprendizaje, C) Tamaño de lote y D) Época.

Figura 86

Ajuste de hiperparámetros

```
# A) Entrada(input_shape)
model.add(Conv2D(filters=16, kernel_size=(3,3),
                 input_shape=(224,224,3),
                 activation='relu'))
# B) Tasa de aprendizaje(Lr)
adadelata=Adadelata(lr=0.001)
wood_model=modelConvolutionalNeuralNetwork()
wood_model.compile(loss='sparse_categorical_crossentropy',
                  optimizer=adadelata,
                  metrics=['accuracy'])
wood_model.summary()
# C) Tamaño de Lote(batch_size)
# D) Epoca(epochs)
madera_model = wood_model.fit(Images_Train, Labels_Train,
                              batch_size=32 ,epochs=50,
                              verbose=1,
                              validation_data=(Images_Test, Labels_Test),
                              shuffle=True)
```

Nota. En esta figura se muestra el ajuste de hiperparámetros para el código manual.

A continuación, se detalla los hiperparámetros que se crearon en el código manual.

1. Entrada. - Hiperparámetro correspondiente a la dimensión de una imagen (alto, ancho y profundidad).
2. Tasa de aprendizaje. - Hiperparámetro referente a la longitud del paso que da cada iteración mientras se realiza el proceso de entrenamiento.
3. Tamaño de lote. - Hiperparámetro perteneciente al número de datos de entrenamiento (muestras de madera laurel) en el proceso de entrenamiento.
4. Época. - Hiperparámetro concerniente al número de iteraciones que debe ejecutar el proceso de entrenamiento.

Los valores modificados por cada modelo de red neuronal convolucional se muestran la tabla 15.

Tabla 15*Ajuste de hiperparámetros.*

Modelo	Hiperparámetros			Época
	Tasa de aprendizaje	Entrada (Dimensión de Imagen)	Tamaño de lote	
Modelo 1	0.00001	64x64	32	400
	0.001	64x64	16	300
Modelo 2	0.00001	200x200	32	400
	0.001	200x200	16	300
Modelo 3	0.00001	224x224	32	400
	0.001	224x224	16	300

Nota. En esta tabla se muestra los valores modificados por cada modelo de red neuronal convolucional.

3.4.8. Fase 8. Entrenamiento del modelo.

Esta fase comprende al entrenamiento de un modelo de red neuronal convolucional mediante el ajuste de hiperparámetros definido en la fase 7 (Ajuste de hiperparámetros). Para lo cual se adiestra los modelos definidos el Capítulo III (3.4.2. Fase 2. Definición de las arquitecturas de redes neuronales convolucionales) identificando patrones o características en una imagen, permitiendo determinar el contenido de humedad en la madera Laurel. De tal forma que se utilizó una media entre 300 y 400 épocas para el entrenamiento de dichos modelos.

En la figura 87 se visualiza el entrenamiento de los modelos propuestos en esta investigación utilizando una media entre 300 y 400 épocas tal.

Figura 87

Entrenamiento del modelo.

```

Train on 1275 samples, validate on 225 samples
Epoch 1/300
1275/1275 [=====] - 6s 5ms/step - loss: 1.1120 - accuracy: 0.2871 - val_loss: 1.1008 - val_accuracy: 0.4222
Epoch 2/300
1275/1275 [=====] - 6s 5ms/step - loss: 1.1120 - accuracy: 0.3145 - val_loss: 1.1004 - val_accuracy: 0.4222
Epoch 3/300
1275/1275 [=====] - 5s 4ms/step - loss: 1.1152 - accuracy: 0.2965 - val_loss: 1.1001 - val_accuracy: 0.4044
Epoch 4/300
1275/1275 [=====] - 5s 4ms/step - loss: 1.1070 - accuracy: 0.3239 - val_loss: 1.0998 - val_accuracy: 0.4133
Epoch 5/300
1275/1275 [=====] - 6s 5ms/step - loss: 1.1105 - accuracy: 0.3208 - val_loss: 1.0995 - val_accuracy: 0.4178
Epoch 6/300
1275/1275 [=====] - 6s 4ms/step - loss: 1.1061 - accuracy: 0.3169 - val_loss: 1.0993 - val_accuracy: 0.4222
Epoch 7/300

```

Nota. En esta figura se muestra el entrenamiento del modelo en el código manual.

3.4.9. Fase 9. Inspección de resultados de codificación manual.

En esta fase se realiza la inspección de resultados obtenidos mediante el ajuste de hiperparámetros definido en la fase 7 (Ajuste de hiperparámetros). Para lo cual si hizo uso de la métrica de evaluación exactitud (Accuracy), permitiendo determinar la eficiencia de un modelo de red neuronal convolucional (CNN).

Los resultados obtenidos al entrenar un modelo de red neuronal convolucional (CNN) con ajuste de hiperparámetros se puede observar en la tabla 16.

Tabla 16

Resultados de código manual

Comparativa de los resultados de la exactitud (Accuracy), según su división de datos y ajuste de hiperparámetros.						
Modelo	División	Hiperparámetros			Época	Exactitud (Accuracy)
		Tasa de aprendizaje	Entrada (Dimensión de Imagen)	Tamaño de lote		
Modelo 1	Porcentual (85%-15%)	0.00001	64x64	32	400	71.11%
		0.001	64x64	16	300	72.88%
Modelo 2	Porcentual (85%-15%)	0.00001	200x200	32	400	74.66%
		0.001	200x200	16	300	76.00%
Modelo 3	Porcentual (85%-15%)	0.00001	224x224	32	400	79.55%
		0.001	224x224	16	300	80.44%

Nota. En esta tabla se muestra los resultados obtenidos por cada modelo de red neuronal convolucional.

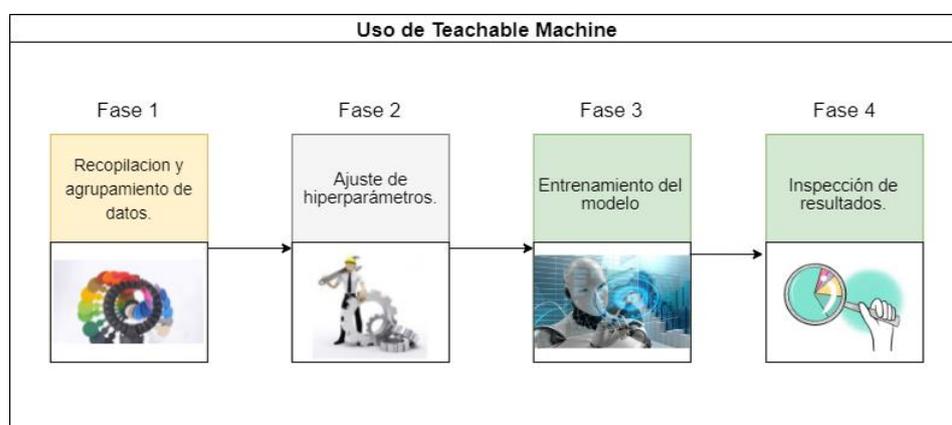
3.5. Etapa 4. Uso de Teachable Machine.

En esta etapa se utiliza teachable machine, una herramienta de Inteligencia Artificial desarrollada por Google, que permite la creación de modelos de aprendizaje automático. A través de la recopilación y agrupamiento de datos, que consiste en reunir una cierta cantidad de información (imágenes de madera laurel) acerca de algún tema en específico (contenido de humedad) para categorizarlos en base a un patrón (anillos de crecimiento, dimensión, textura y color), permitiendo ajustar los hiperparámetros mediante la manipulación de parámetros en el modelo de red neuronal convolucional (CNN), para el entrenamiento del modelo y obtención de resultados.

En la figura 88 se visualiza las fases que intervienen en el manejo de la herramienta Teachable Machine, las mismas que fueron planteadas de la siguiente manera: 1) Obtención de información de contenido de humedad, 2) Selección del método de detección de nivel de humedad y 3) Generación del Data Set.

Figura 88

Uso de Teachable Machine



Nota. En esta figura se muestra los procesos que se realizan al utilizar Teachable Machine.

A continuación, se muestra las 4 fases que intervienen en el uso del teachable machine.

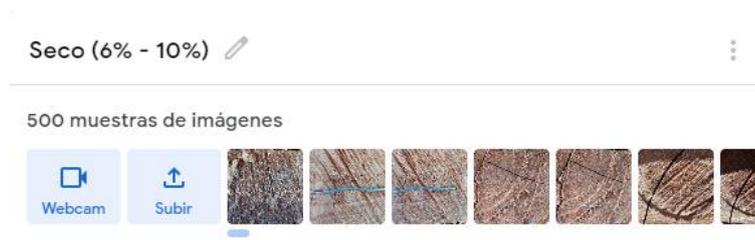
3.5.1. Fase 1. Recopilación y agrupamiento de datos.

En esta fase se utiliza los dos datasets generados en el Capítulo III (3.2.3.- Generación del dataset), con el fin de utilizar las clases que se definieron en el Capítulo III (3.3.2. División del Dataset). Para lo cual se utilizó la ventana emergente que proporciona teachable machine, permitiendo realizar la creación de clases: seco, amorochado y chapotead. Mediante la carga de imágenes de muestras de madera laurel.

En la figura 89 se visualiza la creación de la clase “Seco” mediante la cara de imágenes de un dataset de madera laurel.

Figura 89

Recopilación y agrupamiento de datos.



Nota. En esta figura se muestra el uso de herramienta al utilizar el dataset de madera Laurel.

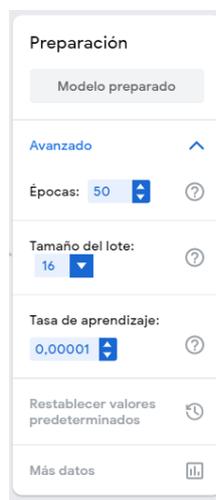
3.5.2. Fase 2. Ajuste de hiperparámetros.

Esta fase se realiza el ajuste de hiperparámetros en la herramienta teachable machine. Para lo cual se modifican los valores por defecto que proporciona la herramienta teachable machine tales como: época, tamaño de lote y tasa de aprendizaje. Con el objetivo de obtener los mejores resultados posibles en un modelo de red neuronal convolucional (CNN).

En la figura 90 se visualiza los hiperparámetros que se modificaron mediante la experimentación.

Figura 90

Ajuste de hiperparámetros.



Nota. En esta figura se muestra cómo se ajusta los hiperparámetros.

3.5.3. Fase 3. Entrenamiento del modelo.

Esta fase comprende al entrenamiento de un modelo de red neuronal convolucional, mediante el ajuste de hiperparámetros definidos en la fase 2 (Ajuste de hiperparámetros). Para lo cual se adiestra el modelo proporcionado por la herramienta teachable machine identificando patrones o características en una imagen, permitiendo determinar el contenido de humedad en la madera Laurel. De tal forma que se utilizó una media entre 100 y 50 épocas para el entrenamiento de dichos modelos.

En la figura 91 se visualiza el número de épocas que se utiliza para el entrenamiento de un modelo de red neuronal convolucional (CNN).

Figura 91

Entrenamiento del modelo.



Nota. En esta figura se muestra el proceso de entrenamiento del modelo de red neuronal utilizando la herramienta Google teachable Machine.

3.5.4. Fase 4. Inspección de resultados.

En esta fase se realiza la inspección de resultados obtenidos mediante el ajuste de hiperparámetros definidos previamente en el Capítulo III (3.5.3 Fase 2. Ajuste de hiperparámetros). Para lo cual se hizo uso de la métrica de evaluación exactitud (Accuracy), permitiendo determinar la eficiencia de un modelo de red neuronal convolucional (CNN).

Los resultados obtenidos al entrenar un modelo de red neuronal convolucional con ajuste de hiperparámetros se visualiza en la tabla 17.

Tabla 17

Ajuste de hiperparámetros.

Comparativa de los resultados de la exactitud (Accuracy), según su división de datos y ajuste de hiperparámetros.						
Modelo	División	Hiperparámetros				Exactitud (Accuracy)
		Tasa de aprendizaje	Entrada (Dimensión de Imagen)	Tamaño de lote	Época	
Teachable Machine	Porcentual (85%-15%)	0.00001	224x224	32	100	91.23%
		0.001	224x224	16	50	94.25%

Nota. En esta tabla se muestra los resultados obtenidos por cada modelo de red neuronal convolucional.

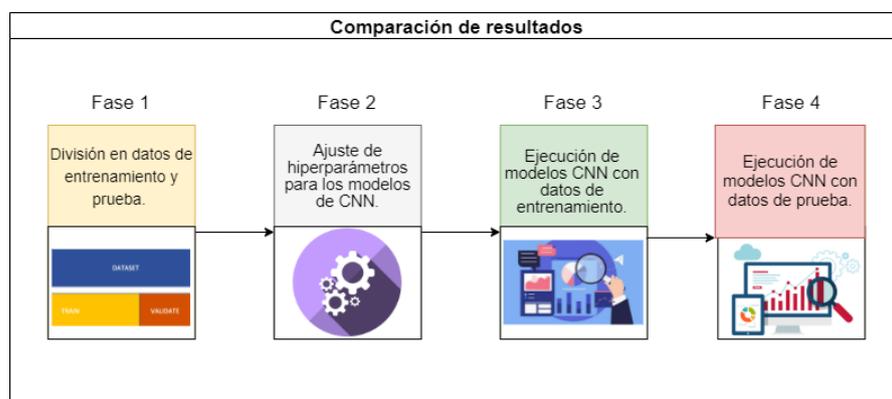
3.6. Etapa 5. Comparación de resultados entre la codificación manual y la herramienta Teachable Machine.

En esta etapa se realizó la comparación de resultados entre la codificación manual y la herramienta Teachable Machine. A través del uso de la misma división de datos (conjunto de muestras de madera Laurel), con los mismos hiperparámetros que se utilizaron en la fase de ajuste y las mismas métricas que se emplearon en la ejecución de modelos de redes neuronales convolucionales (CNN).

En la figura 92 se visualiza las fases que intervienen en la comparación de resultados, dichas fases son las siguientes: 1) División de datos de entrenamiento y prueba 2) Ajuste de hiperparámetros para modelos de redes neuronales convolucionales (CNN), 3) Ejecución de modelos de redes neuronales convolucionales (CNN) con datos de entrenamiento y 4) ejecución de modelos de redes neuronales convolucionales (CNN) con datos de prueba.

Figura 92

Comparación de resultados.



Nota. En esta figura se muestra la precisión por época de entrenamiento.

A continuación, se muestra las 4 fases que intervienen en la comparación de resultados entre el código manual y la herramienta teachable machine.

3.6.1. Fase 1. División de datos de entrenamiento y pruebas.

En esta fase se emplea la división de datos de entrenamiento y pruebas, estableciéndose un porcentaje para cada uno de ellos respectivamente. Permitiendo evitar el sobreajuste (overfitting) o subajuste (underfitting). Realizándose una división de 85% para los datos de entrenamiento y un 15% para los datos de pruebas respectivamente. Es decir, se utilizan 1275 imágenes con contenido de humedad como conjunto de datos de entrenamiento y 225 imágenes con contenido de humedad como conjunto de datos de pruebas.

3.6.2. Fase 2. Ajuste de hiperparámetros para los modelos de CNN.

Esta fase comprende el ajuste de hiperparámetros para modelos de redes neuronales convolucionales (CNN). Conociéndose que, tanto el código manual como la herramienta comparten los mismo hiperparámetros definidos en el Capítulo III (3.4.7 Fase 7 Ajuste hiperparámetros código manual, 3.5.2. Fase 2 Ajuste hiperparámetros teachable machine).

Además, cabe destacar es una de las actividades más complicadas de efectuar, debido a que se ejecuta la experimentación en los hiperparámetros antes mencionados con el fin de mejorar los resultados obtenidos de cada modelo CNN. Dicha complejidad radica en que si se cambian los parámetros de una forma incorrecta posiblemente los resultados obtenidos tengan un sobreajuste (overfitting) o subajuste (underfitting), llegando a tomar una gran cantidad de tiempo volverlos a ajustar.

Para esta investigación se ajustó los hiperparámetros en base al artículo científico "Implementation of a deep learning model for automated classification of *Aedes aegypti* (Linnaeus) and *Aedes albopictus* (Skuse) in real time" (Ong et al. 2021). Siendo este uno de los resultados de la revisión sistemática, mostrando la

forma más adecuada de ajustar parámetros en un modelo de red neuronal convolucional (CNN).

El ajuste de hiperparámetros y resultados obtenidos en el artículo antes mencionado se visualiza en la tabla 18.

Tabla 18

Ajuste de hiperparámetros del artículo científico.

Número de épocas	Learning rate			
	0.00001	0.0001	0.001	0.01
20	93.30%	92.68%	92.73%	93.75%
30	92.61%	96.17%	98.06%	93.78%
40	94.15%	95.04%	97.14%	95.14%
50	93.92%	96.17%	96.93%	95.85%
60	94.26%	96.55%	96.39%	94.22%
70	94.66%	95.75%	96.28%	95.17%
80	94.64%	95.27%	95.80%	94.91%

Nota. Esta tabla contiene los valores utilizados para la configuración de cada hiperparámetro en el artículo científico. Tomado de (Ong et al. 2021).

Los valores que se utilizaron para ajustar los hiperparámetros en cada modelo de red neuronal convolucional (CNN) de esta investigación, se visualizan en la tabla 19.

Tabla 19

Ajuste de hiperparámetros.

Ajuste de hiperparámetros					
Nombre	Tasa de aprendizaje	Numero de épocas	Tamaño del lote	Capa de Entrada (Tamaño de imagen)	Capa de Salida
Modelo 1	0.00001	400	32	64*64	3
Wood	0.001	300	16	64*64	3
Modelo 2	0.00001	400	32	220*220	3
Forest	0.001	300	16	220*220	3
Modelo 3	0.00001	400	32	224*224	3
Mobilenet	0.001	300	16	224*224	3
Teachable machine	0.00001	100	32	224*224	3
	0.001	50	16	224*224	3

Nota. Esta tabla contiene los valores utilizados para la configuración de cada hiperparámetro.

3.6.3. Fase 3. Ejecución de modelos de redes neuronales convolucionales con datos de entrenamiento.

En esta parte se resumen los resultados obtenidos en la fase de entrenamiento utilizándose como conjunto de entrenamiento un grupo de datos que contiene 1275 muestras de madera Laurel, equivalentes al 85% del conjunto de contenido de humedad total, datos que fueron divididos previamente en el Capítulo III (3.6.1. Fase 1. - División de datos de entrenamiento y pruebas).

En la tabla 5 y 6 que se encuentra en el Anexo I sección III (Comparativa de resultados del entrenamiento de los modelos de CNN según división de datos y la tasa de aprendizaje.) donde se observan resultados obtenidos por cada modelo de red neuronal convolucional propuesto en esta investigación. Al momento de ejecutar la fase de entrenamiento (Seco, Amorochado y Chapoteado) y utilizar el ajuste de hiperparámetros se obtuvo unos varios valores en la métrica de exactitud (Accuracy), los mismos que se resumen a continuación como se puede observar en la Tabla 20.

Tabla 20

Comparativa de los mejores resultados de la exactitud (Accuracy), según su división de datos y ajuste de hiperparámetros en la fase de entrenamiento.

Comparativa de los mejores resultados de la exactitud (Accuracy), según su división de datos y ajuste de hiperparámetros en la fase de entrenamiento.						
Modelo	División	Hiperparámetros				Exactitud (Accuracy)
		Tasa de aprendizaje	Entrada (Dimensión de Imagen)	Tamaño de lote	Época	
Modelo 1	Porcentual (85%-15%)	0.00001	64x64	32	400	68.95%
		0.001	64x64	16	300	69.61%
Modelo 2	Porcentual (85%-15%)	0.00001	200x200	32	400	72.86%
		0.001	200x200	16	300	73.43%
Modelo 3	Porcentual (85%-15%)	0.00001	224x224	32	400	76.84%
		0.001	224x224	16	300	77.17%
Teachable Machine	Porcentual (85%-15%)	0.00001	224x224	32	100	91.23%
		0.001	224x224	16	50	94.25%

Nota. Esta tabla contiene los resultados de la métrica Accuracy por cada modelo de red neuronal convolucional.

Lográndose una mayor tasa de predicción con un 94.25% al utilizar la herramienta teachable machine, teniendo una diferencia de 17.08% en comparación al modelo 3 con un 77.17%, sabiendo que teachable machine genera el mejor modelo de red neuronal en base a los resultados obtenidos.

3.6.4. Fase 4. Ejecución de modelos de clasificación con datos de pruebas.

En esta etapa se realizará una evaluación y comparación de los resultados obtenidos por los algoritmos de clasificación propuestos en esta investigación para determinar el porcentaje contenido de humedad. Para esta fase es necesario considerar las siguientes métricas principales: 1) Matriz de confusión, 2) Exactitud, 3) Precisión 4) Recall, y 5) F1.

A continuación, se analizan los resultados obtenidos en cada una de las métricas de evaluación de los modelos de redes neuronales convolucionales (CNN) de acuerdo con cada una de las características mencionadas anteriormente:

1. Métrica 1 (Matriz de Confusión). - métrica que permite representar gráficamente a través de una matriz, la exactitud que los modelos de clasificación obtuvieron al momento de ejecutar sus pruebas. Estableciendo un resultado en la predicción, es decir si el modelo acertó en saber si una clase (Seco, Amorochado, Chapoteado) es correcta o incorrecta (Escobar-Sánchez 2020).

En las tablas 1,2,3,4,5,6,7, y 8 las mismas que se encuentran alojadas en el Anexo 1 Sección IV se puede visualizar las matrices de confusión que se obtuvieron en esta investigación de acuerdo con cada modelo de clasificación según su tasa de aprendizaje y división del conjunto de datos de muestras de madera laurel correspondientes.

Dichas tablas se usaron para realizar la tabulación de los datos obtenidos por esta métrica como se puede visualizar en las tablas 7,8,9, y 10 que se encuentran adjuntas en el Anexo I sección V. Esta información será utilizada en las métricas restantes para determinar cuál de los algoritmos de clasificación propuestos en esta investigación son los mejores en su predicción.

2. Métrica 2 (Exactitud).- métrica que permite conocer el número de aciertos que los modelos de redes neuronales convolucionales obtuvieron para predecir si una clase (Seco, Amorochado, Chapoteado) es correcta o incorrecta (Escobar-Sánchez 2020).

En la tabla 11 y 12 que se adjunta en el Anexo I sección VI se pueden visualizar los resultados obtenidos de la exactitud de acuerdo

con la tasa de aprendizaje, al método de división y la cantidad del conjunto de muestras de madera Laurel establecido.

3. Métrica 3 (Precisión).- en la Figura 93 se puede visualizar los valores considerados dentro de la matriz de confusión para poder obtener la métrica de la precisión la cual permite establecer la precisión de la predicción que realizan cada uno de los modelos de clasificación enfocándose en el número de predicciones que se realizaron correctamente, tomando en cuenta cada una de las clases predichas en este Seco, Amorochado y Chapoteado (true positives) y Seco, Amorochado y Chapoteado (true negatives). Es decir, se obtiene dividiendo el valor de los verdaderos positivos y verdaderos negativos (porcentaje de humedad predichas correctamente) para la suma del total del valor de los verdaderos positivos y verdaderos negativos respectivamente (Escobar-Sánchez 2020).

Figura 93

Métrica Precisión.

Matriz de confusión (Métrica Precisión)											
		Valor Real									
Valor Predicho	(-)	<table border="1"> <tr> <td>Verdaderos Negativos VN</td> <td>Falsos Negativos FN</td> </tr> <tr> <td colspan="2"> $Precisión = TP / (TP + FP)$ $Precisión = TN / (TN + FN)$ </td> </tr> <tr> <td>Falsos Positivos FP</td> <td>Verdaderos Positivos VP</td> </tr> <tr> <td></td> <td></td> <td>(-) (+)</td> </tr> </table>	Verdaderos Negativos VN	Falsos Negativos FN	$Precisión = TP / (TP + FP)$ $Precisión = TN / (TN + FN)$		Falsos Positivos FP	Verdaderos Positivos VP			(-) (+)
	Verdaderos Negativos VN	Falsos Negativos FN									
$Precisión = TP / (TP + FP)$ $Precisión = TN / (TN + FN)$											
Falsos Positivos FP	Verdaderos Positivos VP										
		(-) (+)									
(+)											

Nota. En esta figura se muestra las fórmulas para el cálculo de la métrica de precisión.

En la tabla 13 y 14 que se adjuntan en el Anexo I sección VII se pueden visualizar los resultados obtenidos de la métrica precisión de acuerdo con la tasa de aprendizaje, al método de división y la cantidad del conjunto de muestra de madera Laurel. Esta información será utilizada en las métricas F1 para determinar cuál de los algoritmos de clasificación propuestos en esta investigación son los mejores en su predicción.

- Métrica 4 (Recall). - en la Figura 94 se puede visualizar los valores considerados dentro de la matriz de confusión para poder obtener la métrica recall la cual permite determinar el porcentaje de humedad identificadas de manera correcta como positivos en comparación al total de positivos verdaderos (true positive). Es decir, se lo obtiene

dividiendo el valor de los verdaderos positivos y verdaderos negativos (Seco, Amorocho y Chapoteado predichas correctamente) para la suma del valor de verdaderos positivos y negativos respectivamente (Escobar-Sánchez 2020).

Figura 94

Métrica Recall.

Matriz de confusión (Métrica Recall)											
		Valor Real									
Valor Predicho	(-)	<table border="1"> <tr> <td>Verdaderos Negativos VN</td> <td>Falsos Negativos FN</td> </tr> <tr> <td colspan="2"> $Recall = TP / (TP + FN)$ $Recall = TN / (TN + FP)$ </td> </tr> <tr> <td>Falsos Positivos FP</td> <td>Verdaderos Positivos VP</td> </tr> <tr> <td></td> <td></td> <td>(-) (+)</td> </tr> </table>	Verdaderos Negativos VN	Falsos Negativos FN	$Recall = TP / (TP + FN)$ $Recall = TN / (TN + FP)$		Falsos Positivos FP	Verdaderos Positivos VP			(-) (+)
	Verdaderos Negativos VN	Falsos Negativos FN									
$Recall = TP / (TP + FN)$ $Recall = TN / (TN + FP)$											
Falsos Positivos FP	Verdaderos Positivos VP										
		(-) (+)									
(+)											

Nota. En esta figura se muestra las fórmulas para el cálculo de la métrica de recall.

En la tabla 15 y 16 que se adjuntan en el Anexo I sección VIII se pueden visualizar los resultados obtenidos de la métrica recall de acuerdo con la tasa de aprendizaje, al método de división y la cantidad del conjunto de noticias establecido. Esta información será utilizada en las métricas F1 para determinar cuál de los algoritmos de clasificación propuestos en esta investigación son los mejores en su predicción.

- Métrica 5 (F1).- Es una métrica muy importante para esta investigación ya que permite realizar una comparación de modelos de redes neuronales convolucionales, establece una media aritmética entre la

métrica Precisión y Recall. Calculándose de la siguiente manera $2 * \text{Precisión} * \text{Recall} / (\text{Precisión} + \text{Recall})$ (Escobar-Sánchez 2020).

Dichos resultados son obtenidos basándose en la información recolectada al aplicar las métricas de evaluación (Precisión y Recall), en el entrenamiento de los modelos de redes neuronales convolucionales propuestos en esta investigación. A continuación, se presenta un resumen de los tres modelos de redes neuronales convolucionales y teachable machine propuestos en esta investigación que consiguieron los mejores resultados al aplicar la métrica F1 (media armónica) en las pruebas. Utilizando división porcentual (85% - 15%), tasa de aprendizaje (0.001) y tamaño de lote (16), como se observa en la tabla 21.

Tabla 21

Comparativa de los mejores resultados de la F1, según su división de datos y ajuste de hiperparámetros en la fase de pruebas.

Comparativa de los mejores resultados de la F1, según su división de datos y ajuste de hiperparámetros en la fase de pruebas.						
Modelo	División	Hiperparámetros		Seco (6% - 10%) de humedad	Amorocho (11% - 14%) de humedad	Chapoteado (15% - 18%) de humedad
		Entrada (Dimensión de Imagen)	Época			
Modelo 1	Porcentual	64x64	300	73.33%	72.00%	73.33%
Modelo 2	Porcentual	200x200	300	76.00%	76.00%	76.00%
Modelo 3	Porcentual	224x224	300	79.73%	80.53%	81.08%
Teachable Machine	Porcentual	224x224	50	94.11%	94.59%	97.68%

Nota. Esta tabla contiene los resultados de la métrica F1 por cada modelo de red neuronal convolucional.

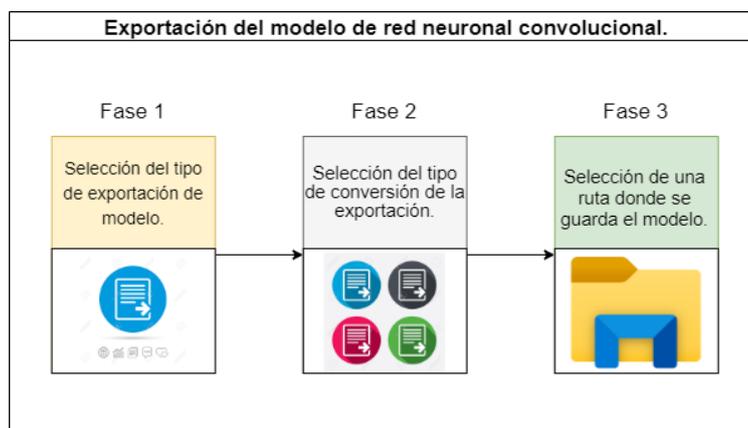
3.7. Etapa 6. Exportación del modelo de red neuronal convolucional.

En esta etapa se ejecuta la exportación del modelo de red neuronal convolucional una vez finalizada la etapa 4 (Uso de Teachable Machine). Mediante la selección del tipo de exportación de modelo CNN, permitiendo escoger unos de los tipos de transacción (Tensorflow.js, Tensorflow, Tensorflow lite) para consecuentemente escoger el tipo de conversión del modelo (punto flotante, cuantificado y Edge Tpu) y finalmente escoger la ruta donde se desea guardar los archivos pertenecientes al modelo de red neuronal convolucional.

En la figura 95 se visualiza las fases que intervienen en la exportación del modelo de red neuronal convolucional, las mismas que fueron plateados de la siguiente manera: 1) Selección del tipo de exportación de modelo, 2) Selección de tipo de conversión de la exportación y 3) Ejecución de modelos CNN con datos de entrenamiento.

Figura 95

Exportación del modelo CNN.



Nota. En esta figura se visualiza las fases que intervienen en la exportación de un modelo CNN.

A continuación, se muestra las 3 fases que intervienen en la exportación del modelo de red neuronal convolucional (CNN).

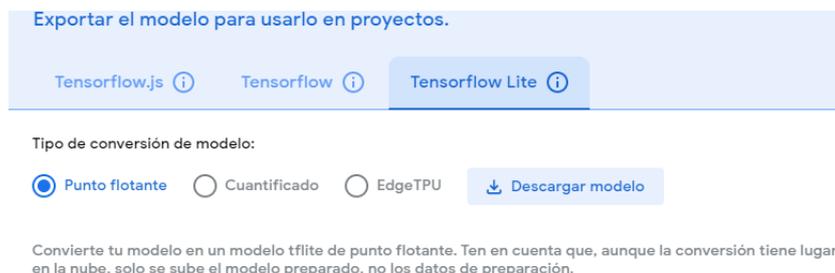
3.7.1. Fase 1. Selección del tipo de exportación de modelo.

En esta fase se realiza la selección de tipo de exportación, entendiéndose como exportar a forzar a una aplicación a crear un archivo con una extensión determinada, permitiendo el uso de dicho archivo en otros aplicativos. Para lo cual se escogió el tipo de exportación tensorflow Lite, dado que dicho tipo de exportación es el único que permite transformar modelos compatibles para dispositivos móviles.

En la figura 96 se visualiza los tres tipos de exportación que proporciona la herramienta teachable machine, los mismos que fueron plateadas de la siguiente manera: Tensorflow.js, Tensorflow y Tensorflow Lite.

Figura 96

Selección del tipo de exportación del modelo.



Nota. En esta figura se visualiza las opciones disponibles para la exportación de un modelo.

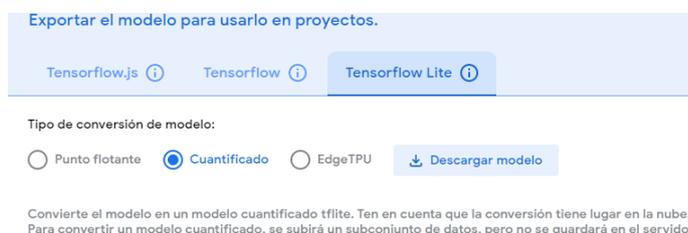
3.7.2. Fase 2. Selección del tipo de conversión del modelo.

Una vez efectuada la fase 1 (Selección del tipo de exportación de modelo), seleccionamos el tipo de conversión del modelo. Para lo cual se escogió el tipo de archivo cuantificado, permitiendo el uso de dicho tipo de archivo en la herramienta Android Studio. Cabe destacar que dicho tipo de archivo es el más apto para elaborar las etapas del desarrollo del aplicativo móvil.

En la figura 97 se observa los tipos tres tipos de conversión de modelo, los mismos que fueron plateadas de la siguiente manera: Punto flotante, Cuantificado y EdgeTPU.

Figura 97

Selección del tipo de conversión del modelo.



Nota. En esta figura se visualiza las opciones disponibles para la conversión de un modelo.

3.7.3. Fase 3. Selección de la ruta donde se guarda el modelo.

Finalmente, en esta fase se realiza la selección de la ruta donde se desea guardar el modelo generado en la fase 2 (Selección del tipo de conversión del modelo). Para lo cual se oprime la opción descargar modelo, desplegándose una ventana emergente que permite escoger la ruta "C:\Users\Work\Downloads\Modelo Entrenado" para guardar los archivos tflite generados por esta herramienta, una vez decidida la ruta inmediatamente se guardan los archivos en un formato rar o comprimidos.

En la figura 98 se visualiza la ejecución de una ventana de ayuda en la que se puede seleccionar una ruta de guardado.

Figura 98

Archivos que contienen un modelo entrenado.

Nombre	Fecha de modificación	Tipo	Tamaño
converted_tflite.zip	12/7/2021 21:49	Archivo WinRAR Z...	2.096 KB
converted_tflite_quantized.zip	12/7/2021 21:47	Archivo WinRAR Z...	831 KB

Nota. En esta figura se visualiza el archivo generado por el conversor de teachable machine.

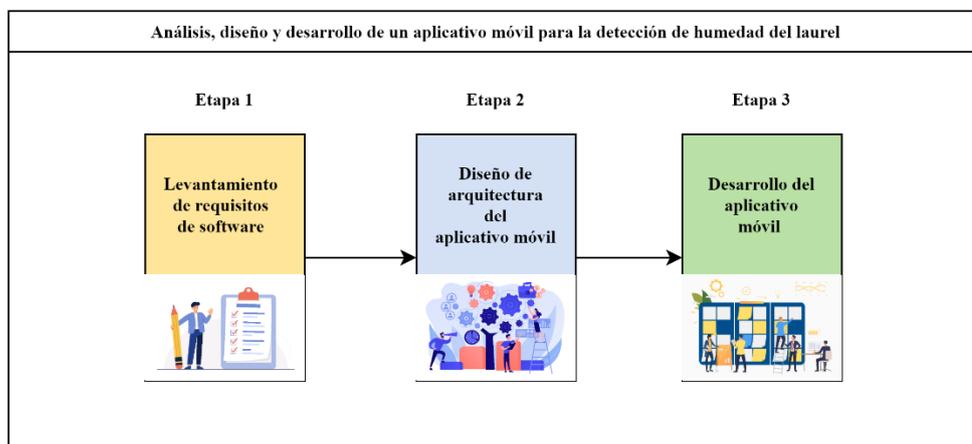
Capítulo IV

4. Análisis, diseño y desarrollo de un aplicativo móvil para el diagnóstico de humedad en la madera del Laurel.

En el presente capítulo se realiza el análisis, diseño y desarrollo del aplicativo móvil propuesto en esta investigación, basándose en el estándar IEEE 830 y la metodología de desarrollo RAD. A continuación, se muestra en la figura 99 las etapas necesarias para el desarrollo del aplicativo móvil las cuales están compuestas de la siguiente manera: 1) Levantamiento de requisitos, 2) Diseño de la arquitectura del aplicativo móvil, 3) Desarrollo del aplicativo Móvil.

Figura 99

Etapas para el desarrollo del aplicativo móvil propuesto en esta investigación.



Nota. En esta figura se observa las etapas del presente capítulo.

A continuación, se describen las 3 etapas que contempla este capítulo:

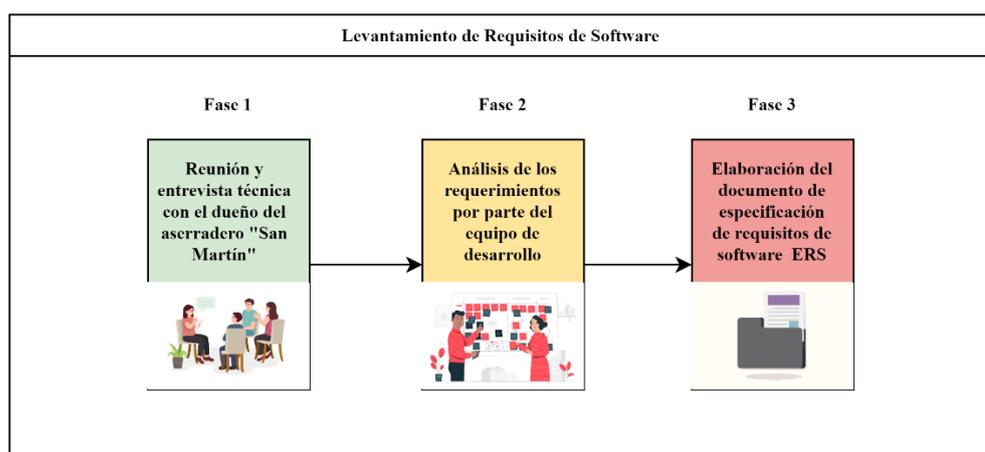
4.1. Etapa 1. Levantamiento de requisitos de software.

En la figura 100, se muestra el proceso requerido para la elaboración del documento de especificación de requerimientos ERS. Basándose en el estándar IEEE 830. El cual permite obtener la información necesaria para realizar la medición del contenido de humedad del laurel. Dicha información fue aprobada

por parte del Sr. Carlos Efraín Castro Guerrero, dueño del aserradero “San Martín” (Auspiciante). Para lo cual se desarrollaron las siguientes fases que se presenta a continuación: 1) Reunión y entrevista técnica con el dueño del aserradero, 2) Análisis de los requerimientos por parte del equipo de desarrollo, y 3) Elaboración del documento de Especificación de Requisitos de Software (ERS) según el estándar IEEE 830.

Figura 100

Levantamiento de requisitos de software



Nota. Esta figura muestra las fases necesarias para generar un documento de especificación de requisitos de software.

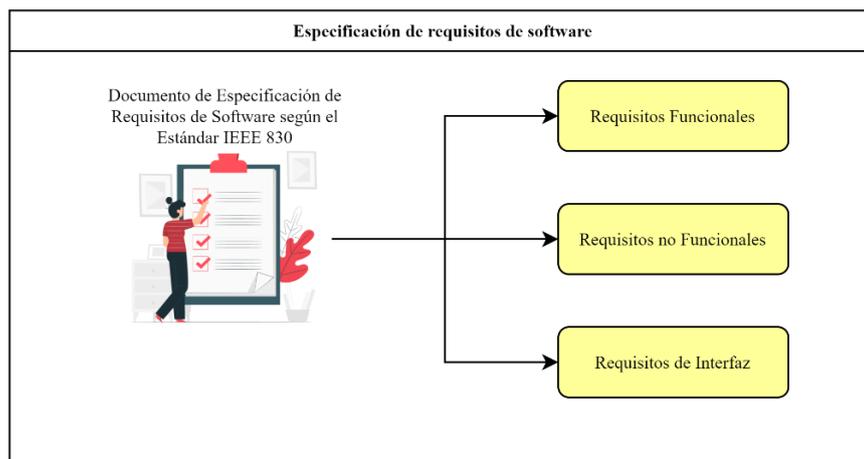
A continuación, se describe de forma detallada las fases necesarias para la generación de un documento de especificación de requisitos de software (ERS):

- Fase 1 (Reuniones y entrevistas técnicas). - Se realiza una charla con el Sr. Carlos Efraín Castro Guerrero, dueño del aserradero “San Martín” y el Sr. Francisco Trajano Muñoz Pila experto artesano maderero. Exponiéndose los inconvenientes que tienen sus trabajadores en el proceso de selección de materia prima, lo cual está basado en función al nivel de humedad de la madera del Laurel. De tal manera que se solicita el desarrollo de una herramienta

tecnológica que sea portable y permita medir de forma rápida y sencilla el nivel de humedad de la madera del Laurel. Beneficiando directamente a los trabajadores de dicho aserradero pues una correcta selección de materia prima influirá directamente en la calidad de fabricación de productos derivados de la madera.

- Fase 2 (Análisis de los requerimientos). – Una vez desarrollada la fase 1, el grupo de desarrollo obtiene el conocimiento necesario para poder proponer una solución viable que se vea reflejada a través de la generación de un documento de especificación de requisitos de software (ERS).
- Fase 3 (Elaboración del documento ERS). – Como resultado de la fase 2, se obtiene el documento de especificación de requisitos de software que está desarrollado según lo establecido con el estándar IEEE 830 y los acuerdos previos con el Sr. Carlos Efraín Castro Guerrero. Estableciéndose responsabilidades tanto para el equipo de desarrollo (tesistas) como para el dueño del aserradero “San Martín”, facilitando sus instalaciones para la adquisición de información que se requiera. Sirviendo como soporte para el desarrollo del aplicativo móvil que mida el nivel de humedad de la madera a través de aprendizaje profundo. El documento de especificación de requerimientos, se describe de forma detallada en el Anexo 2 Documento de especificación de requisitos de software.

En la figura 101, se observa la clasificación de requerimientos según su funcionalidad los cuales son: 1) requisitos funcionales, 2) requisitos no funcionales y 3) requisitos de interfaz (Pmoinformatica 2018a).

Figura 101*Especificación de requisitos de software ERS*

Nota. En esta figura se presenta la clasificación de requisitos en el documento de especificación de requisitos de software.

4.1.1. Requisitos Funcionales.

Para poder desarrollar el aplicativo móvil que mida el nivel de humedad en la madera del laurel a través de aprendizaje profundo se requiere de un análisis previo expresado en las fases 3 fases mencionadas anteriormente: fase 1 (Reuniones y entrevistas técnicas), fase 2 (Análisis de los requerimientos) y fase 3 (Elaboración del documento ERS). Generando un documento de especificación de requisitos de software (ERS) que extrae las funcionalidades y el comportamiento deseado de dicho aplicativo.

A continuación, se detallan de forma general los requisitos funcionales que fueron extraídos para el desarrollo del aplicativo móvil de esta investigación.

- 1) Gestión de información: Se permitirá visualizar la información necesaria para poder manipular correctamente el aplicativo móvil.
- 2) Gestión de medidor de humedad: Se podrá medir el nivel de humedad de la madera del laurel en tiempo real, a través del uso de la cámara del dispositivo móvil, indicando el porcentaje al que pertenece.

- 3) Gestión de contacto: Se permitirá visualizar información de contacto del aserradero “San Martín”.

Los requisitos funcionales de este aplicativo software son detallados de forma más específica en el Anexo 2 Documento de especificación de requisitos de software (Apartado 3 Requisitos Específicos. - en la sección 3.1 Requisitos Funcionales).

4.1.2. Requisitos no Funcionales.

La calidad del software es un aspecto muy importante a tomar en consideración es por ello que los requisitos no funcionales son tan importantes. En este apartado se presentan las restricciones que va a tener el sistema o producto software como por ejemplo la seguridad de las aplicaciones, la usabilidad, uso de tecnologías, etc. (Pmoinformatica 2018a).

A continuación, se presenta los requisitos no funcionales que fueron extraídos de la fase 1 (Reuniones y entrevistas técnicas) y fase 3 (Elaboración del documento ERS):

- 1) Fiabilidad. – Se enfoca en que el aplicativo móvil pueda funcionar correctamente según lo planificado eliminando cualquier tipo de falla en su diagnóstico de nivel de humedad en la madera.
- 2) Disponibilidad. – El aplicativo móvil debe estar listo para funcionar a cualquier hora sin importar horarios.
- 3) Portabilidad. – La app móvil debe poder ser instalado en dispositivos con sistema operativo Android.

Los requisitos no funcionales de este aplicativo móvil son detallados de forma más específica en el Anexo 2 Documento de especificación de requisitos

de software (Apartado 3 Requisitos Específicos. - en la sección 3.2 Requisitos No Funcionales).

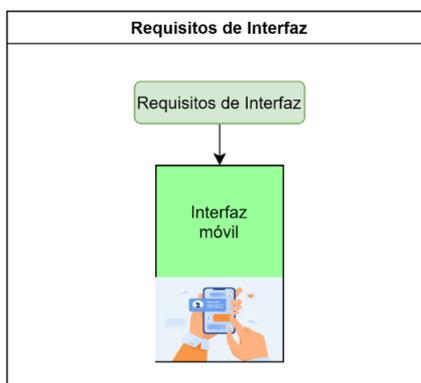
4.1.3. Requisitos de Interfaz.

Según (Gesuiti 2017) las interfaces juegan un rol muy importante en el desarrollo de aplicativos pues de estas dependen que tan usado puede llegar a ser un producto software. Las interfaces deben presentar una buena usabilidad, el cual tiene por objetivo facilitar el uso de alguna acción dentro de un software. Es por ello que en el desarrollo del aplicativo móvil en esta investigación se enfoca en dar una buena experiencia de usuario a los trabajadores del aserradero “San Martin” en el proceso de medición del contenido de humedad del laurel.

En la figura 102, se puede observar los requisitos de interfaz los cuales fueron extraídos en la etapa 1 (Levantamiento de requisitos de software), para lo cual es necesario presentar todos los componentes visuales de los que el aplicativo móvil estará compuesto.

Figura 102

Requisitos de Interfaz



Nota. Esta figura muestra la estructura de los requisitos de interfaz de esta investigación.

En la figura 103 se puede visualizar las interfaces del aplicativo móvil. Cada interfaz móvil está compuesta de elementos visuales agrupados de forma secuencial para permitir una buena usabilidad. Además de agilizar el proceso de medición de humedad en la madera. Para el diseño y maquetación de las interfaces se utilizó el software Figma basándose en el documento de especificación de requisitos el mismo que está disponible en el Anexo 2 Documento de especificación de requisitos de software (Apartado 3 Requisitos Específicos. - en la sección 3.3 Requisitos de Interfaz).

Figura 103

Interfaces del aplicativo móvil para el diagnóstico de humedad del laurel.



Nota. En esta figura se muestran las interfaces para el aplicativo móvil en esta investigación.

4.2. Etapa 2. Diseño de arquitectura del aplicativo móvil.

La finalidad de esta etapa radica en la definición de la arquitectura del cual estará compuesta el aplicativo móvil para el diagnóstico de contenido de humedad del laurel a través de aprendizaje profundo. La información requerida en esta etapa fue extraída a través del levantamiento de requisitos, así como la reunión previa con el dueño del aserradero “San Martín” y el Sr. Carlos Efraín Castro Guerrero que es el experto artesano maderero.

Para el desarrollo del aplicativo móvil de esta investigación se hace uso de una arquitectura monolítica el cual consiste en tener todas las funcionalidades

y servicios que se requiera de forma interna. Sin necesitar de dependencias externas, de tal manera que cada funcionalidad comparte recursos y memoria conjuntamente. Otra ventaja de usar esta arquitectura es que los aplicativos se vuelven muy fáciles de desarrollar además el despliegue como la ejecución se lo realiza de forma rápida y sin complicaciones (Ilimit 2020).

En la figura 104, se puede observar la arquitectura necesaria para el desarrollo del aplicativo móvil. Para lo cual se presenta los siguientes 4 componentes:

1) Modelo de red neuronal convolucional (CNN). - Archivo "model.tflite" exportado previamente en el Capítulo III (Apartado 3.7. Etapa 6. Exportación del modelo de red neuronal convolucional CNN).

2) Backend. - Permite la utilización de librerías de tensorflow para la implementación e interpretación de un modelo de red neuronal convolucional (CNN), utilizando el lenguaje de programación Java.

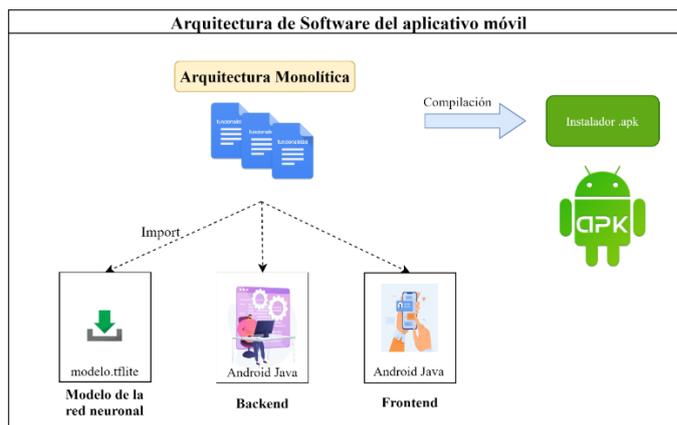
3) Frontend. - Permite visualizar el proceso de diagnóstico del contenido de humedad en el laurel en tiempo real, a través de la utilización del lenguaje de programación Java.

4) Instalador. - Es el archivo instalable del aplicativo móvil, el mismo que está compuesto de todas las especificaciones las cuales se encuentran detalladas en el Anexo 2 Documento de especificación de requisitos de software.

Además, se requiere de diagramas de caso de uso, de secuencia y de paquetes para representar las funcionalidades que va a poseer el aplicativo móvil propuesto en esta investigación.

Figura 104

Arquitectura de software del aplicativo móvil



Nota. Esta figura muestra la arquitectura de software para el aplicativo móvil de esta investigación. Tomado de (Blancarte 2020).

4.2.1. Diagrama de Caso de Uso.

Aquí se diseñan los diagramas de caso de uso, que permiten comprender de forma gráfica las funcionalidades que debe poseer el aplicativo software en desarrollo. Dichas funcionalidades son las siguientes: 1) Gestión de información, 2) Gestión de medidor de humedad y 3) gestión de contacto. Como se puede visualizar en el Anexo 2 (Documento de especificación de requisitos de software). A continuación, se describe de forma detallada a cada uno de los ítems antes mencionados.

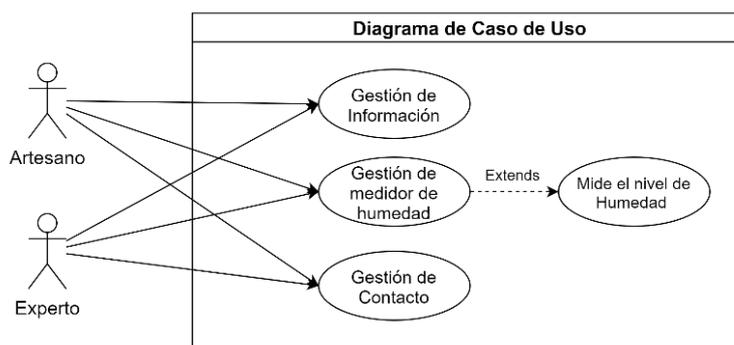
1. Gestión de información. – Permite visualizar información teórica introductoria para que artesano pueda utilizar el aplicativo móvil correctamente.
2. Gestión de medidor de humedad. – Permite realizar el proceso de medición de humedad en la madera del laurel en tiempo real, utilizando un dispositivo Android y la cámara enfocando hacia el corte de madera que se desea medir.

3. Gestión de contacto. – Permite visualizar la información de contacto del aserradero “San Martín” que es la empresa auspiciante de esta investigación.

En la figura 105, se puede visualizar el diagrama de caso de uso de acuerdo a un análisis previo, el cual se encuentra detallado en Anexo 2 (Documento de especificación de requisitos de software) sección 2.2.

Figura 105

Diagrama de Caso de Uso Aplicativo móvil



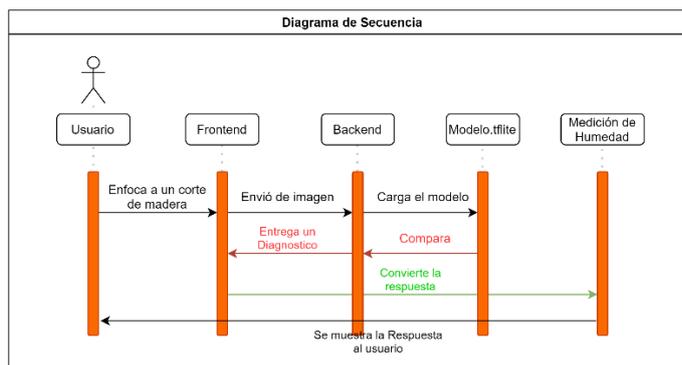
Nota. En esta figura se muestra el diagrama de caso de uso para el desarrollo del aplicativo móvil de esta investigación.

4.2.2. Diagrama de Secuencia.

En la figura 106, se procede a generar el diagrama de secuencia el cual tiene como objetivo indicar de forma gráfica como es la interacción de los involucrados en el proceso de diagnóstico de contenido de humedad de la madera del laurel en el aplicativo móvil. Para lo cual se considera 2 objetos: 1) usuario y 2) el aplicativo móvil.

Figura 106

Diagrama de Secuencia del aplicativo móvil



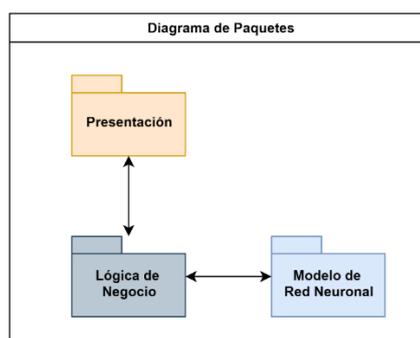
Nota. En esta figura se observa el diagrama de secuencia que se utiliza para el desarrollo del aplicativo móvil en esta investigación.

4.2.3. Diagrama de Paquetes.

El principal objetivo de este tipo de diagrama es permitir visualizar de forma clara y precisa la estructura jerárquica de distintos componentes o módulos que componen un aplicativo software (DiagramasUML 2021). En la figura 107 se muestra el diagrama de paquetes para el desarrollo del aplicativo móvil propuesta en esta investigación. La cual posee los siguientes 3 paquetes: 1) Presentación, 2) lógica de negocio y 3) modelo de red neuronal convolucional (CNN).

Figura 107

Diagrama de paquetes del aplicativo móvil



Nota. En esta figura se muestra el diagrama de paquetes que se utiliza para el desarrollo del aplicativo móvil de esta investigación.

En la figura 107, se pudo visualizar los 3 paquetes que conforman el aplicativo móvil. A continuación, se detallan dichos paquetes:

1) Presentación: Presenta al usuario de forma numérica el nivel de humedad de la madera del laurel en un determinado corte.

2) Lógica de Negocio: Utiliza a los requisitos funcionales definidos en el documento de especificación de requisitos de software. Para cumplir con las funcionalidades y hacer que el aplicativo funcione según lo esperado.

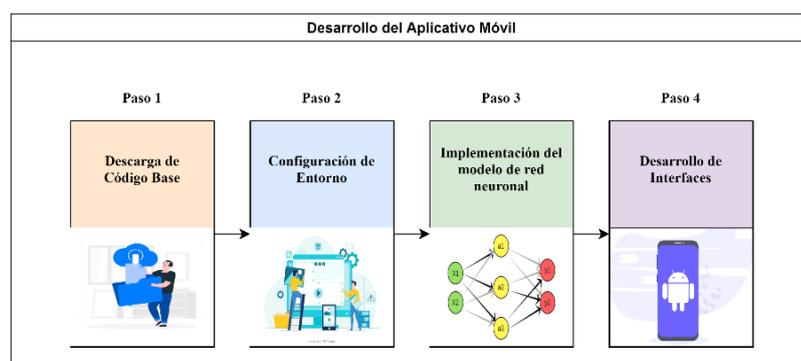
3) Modelo de red neuronal: Responsable de manejar al modelo de red neuronal entrenado en el capítulo III para realizar diagnósticos de humedad en tiempo real.

4.3. Etapa 3. Desarrollo del aplicativo móvil.

En esta etapa se describe al proceso de desarrollo del aplicativo móvil para esta investigación. En la Figura 108 se puede observar los 4 pasos que componen esta etapa: 1) Descarga de Código Base, 2) Configuración del Entorno, 3) Implementación del modelo de red neuronal, y 4) Desarrollo de Interfaces.

Figura 108

Desarrollo del Aplicativo Móvil



Nota. En esta figura se puede observar los pasos necesarios para el desarrollo del aplicativo móvil de esta investigación.

A continuación, se describen los 4 pasos necesarios para desarrollar el aplicativo móvil de esta investigación.

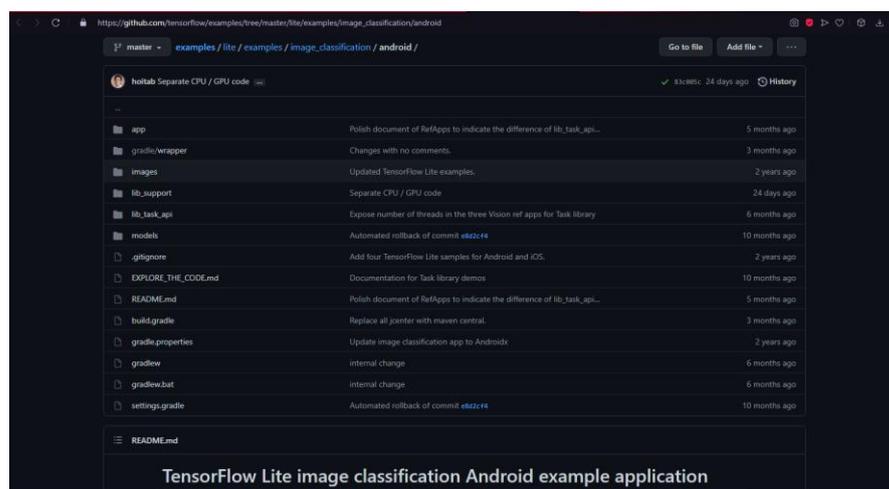
4.3.1. Paso 1. Descarga de Código Base.

De acuerdo con los resultados obtenidos en el Capítulo III (3.7 Etapa 6. Exportación del modelo de red neuronal convolucional) el modelo de red neuronal convolucional (CNN) que presenta los mejores resultados es la herramienta Google Teachable Machine. Razón por la cual se utiliza dicho modelo. En la Figura 109, se puede observar el repositorio de Github el cual posee el código fuente necesario para poder utilizar el modelo de red neuronal convolucional (CNN). Dicho repositorio se encuentra en la siguiente dirección web:

https://github.com/tensorflow/examples/tree/master/lite/examples/image_classification/android

Figura 109

Repositorio Github código fuente base Tensorflow Android

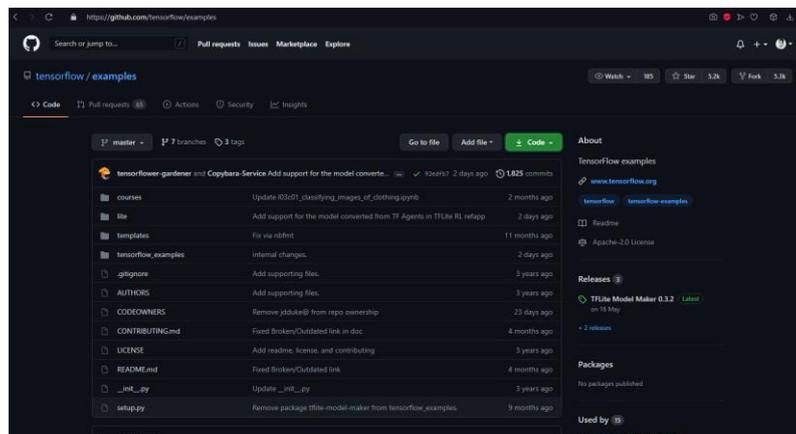


Nota. Esta figura muestra el repositorio de código fuente base alojado en Github recomendado para desarrollar aplicaciones móviles con modelos de redes neuronales utilizando Tensorflow.

Una vez hallado el código fuente se procede a dirigir a la carpeta raíz de ejemplos (examples) para poder descargar dicho proyecto. En la figura 110, se visualiza el contenido correspondiente a la carpeta examples del repositorio de Github del proyecto.

Figura 110

Repositorio raíz del proyecto de Tensorflow

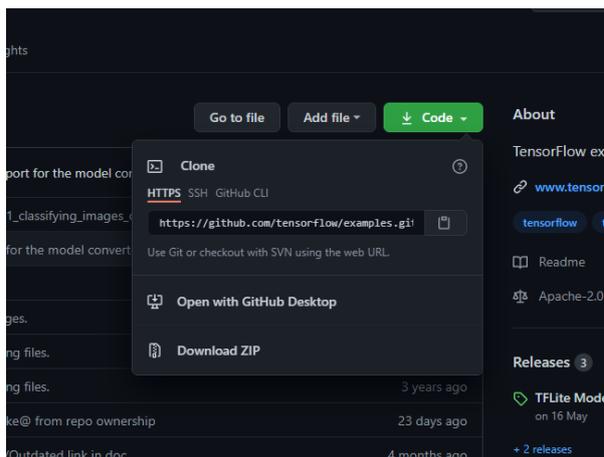


Nota. Esta figura se puede observar el repositorio raíz de los ejemplos de Tensorflow.

Después de ubicarse en la raíz del proyecto existen dos formas de descargar el código fuente. En la Figura 111 se puede observar la primera forma para descargar un proyecto desde los repositorios de Github. El cual consiste en dar click izquierdo en el botón de code y descargarlo en un archivo .zip.

Figura 111

Descarga Manual de Repositorios en Github.

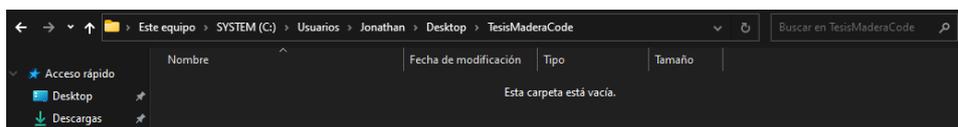


Nota. En esta figura se puede observar el método de descarga manual de un proyecto alojado en los repositorios de Github.

Existe otra forma de descargar proyectos alojados en los repositorios de Github el cual se lo realiza a través de la utilización del siguiente comando de consola git clone "url_repositorio". Este método requiere de la instalación del software de Git. A continuación, es necesario ubicar la carpeta en donde se desea descargar dicho proyecto. En la Figura 112, se puede identificar la ruta en donde este repositorio va a ser descargado, en este caso será en el escritorio de Windows. C:\Users\Jonathan\Desktop\TesisMaderaCode

Figura 112

Selección del Path para almacenar el código fuente.



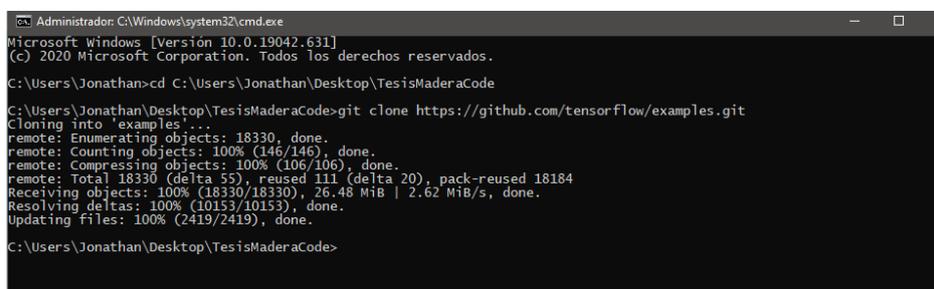
Nota. En esta figura se puede observar el Path o dirección de carpeta en el cual se va a descargar el código fuente.

Una vez creada la carpeta que va a almacenar el código fuente se abre un terminal de Windows y se dirige al Path ubicado anteriormente con el comando cd C:\Users\Jonathan\Desktop\TesisMaderaCode y posteriormente se

utiliza el comando git clone + la dirección url del proyecto. En la Figura 113 se puede observar la ubicación del Path, así como la descarga del proyecto utilizando el comando de Github mencionado previamente.

Figura 113

Descarga del repositorio utilizando Git Clone.



```

Administrador: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19042.631]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Jonathan>cd C:\Users\Jonathan\Desktop\TesisMaderaCode

C:\Users\Jonathan\Desktop\TesisMaderaCode>git clone https://github.com/tensorflow/examples.git
Cloning into 'examples'...
remote: Enumerating objects: 18330, done.
remote: Counting objects: 100% (146/146), done.
remote: Compressing objects: 100% (106/106), done.
remote: Total 18330 (delta 55), reused 111 (delta 20), pack-reused 18184
Receiving objects: 100% (18330/18330), 26.48 MiB | 2.62 MiB/s, done.
Resolving deltas: 100% (10153/10153), done.
Updating files: 100% (2419/2419), done.

C:\Users\Jonathan\Desktop\TesisMaderaCode>

```

Nota. En esta figura se puede observar el proceso de descarga del repositorio utilizando el comando git clone de GitHub.

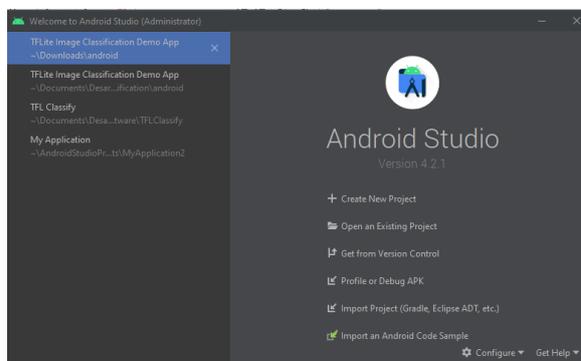
4.3.2. Paso 2. Configuración del Entorno.

Para este paso se requiere tener instalado el IDE Android Studio el cual servirá para poder realizar el desarrollo del aplicativo móvil de esta investigación.

En la figura 114 se observa la interfaz inicial de IDE Android estudio.

Figura 114

Interfaz inicial Android Studio

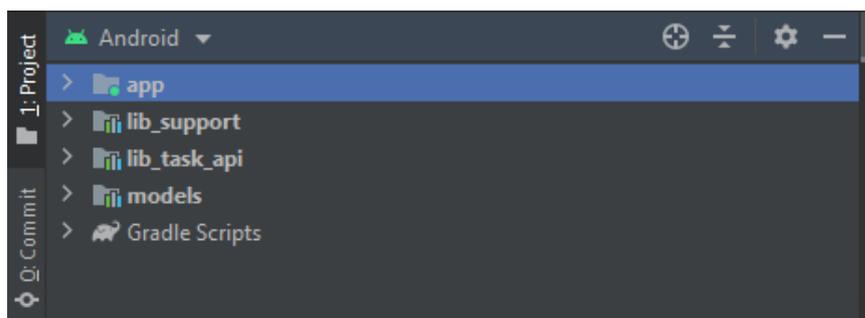


Nota. En esta figura se muestra la interfaz inicial del IDE de desarrollo Android Studio.

Posteriormente se debe abrir el proyecto ubicado en la ruta
 C:\Users\Jonathan\Desktop\TesisMaderaCode\examples\lite\examples\image_classification\android. En la Figura 115, se puede observar la estructura del proyecto que va a ser utilizado como código base suministrado por Tensorflow en la presente investigación.

Figura 115

Estructura del aplicativo móvil en Android Studio.



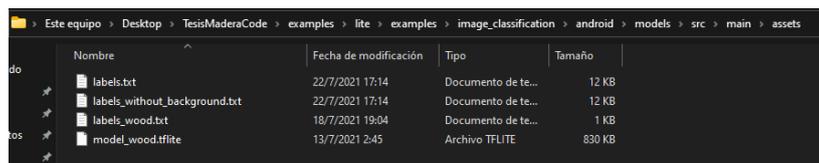
Nota. Esta figura muestra la estructura del aplicativo móvil en Android Studio.

4.3.3. Paso 3. Implementación del modelo de Red Neuronal.

Para este paso se requiere de tener dos archivos los cuales fueron exportados previamente en el Capítulo III (3.7. Etapa 6. Exportación modelo de red neuronal convolucional) siendo estos: “model_wood.tflite” y “labels_wood.txt”. En la Figura 116 se puede mirar los archivos descargados correspondientes al modelo y sus etiquetas los cuales deben ser copiados a la carpeta de modelos ubicados en la ruta
 C:\Users\Jonathan\Desktop\TesisMaderaCode\examples\lite\examples\image_classification\android\models\src\main\assets.

Figura 116

Importación del Modelo de Red Neuronal Convolutiva (CNN) en el proyecto del aplicativo móvil.



Nota. En esta ilustración se puede observar la carpeta en donde se debe pegar los archivos descargados del modelo de red neuronal convolutiva (CNN) en la herramienta Teachable Machine.

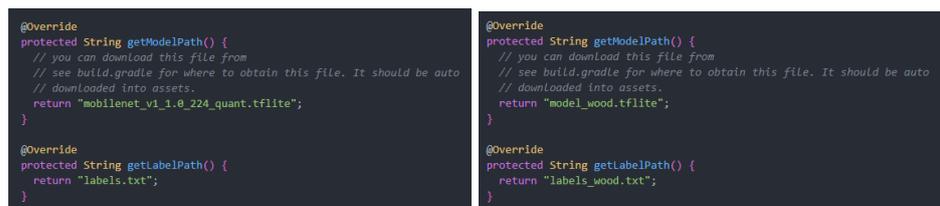
De acuerdo con la documentación suministrada por la herramienta Google Teachable Machine se establece la necesidad de sobrescribir la ruta de los archivos anteriormente pegados como se pudo observar en la Figura 116. Para lo cual nos dirigimos al archivo “ClassifierQuantizedMobileNet.java” que se encuentra ubicado en

C:\Users\Jonathan\Desktop\TesisMaderaCode\examples\lite\examples\image_classification\android\lib_support\src\main\java\org\tensorflow\lite\examples\classification\tflite.

En la figura 117, se muestra una comparación de un antes y un después en el proceso de sobrescritura de la ruta. Pudiéndose observar que se utilizan los archivos “model_wood.tflite” y “labels_wood.txt” respectivamente.

Figura 117

Reemplazo del modelo de red neuronal en el archivo ClassifierQuantizedMobileNet.java

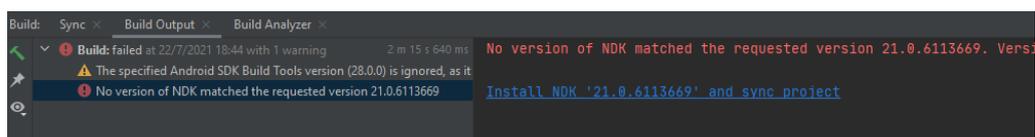


Nota. Esta figura muestra el cambio de nombre del archivo correspondiente al modelo que se va a utilizar en el aplicativo móvil de esta investigación.

Después de haber seleccionado los archivos del modelo de red neuronal se procede a ejecutar el proyecto ya sea en dispositivos físicos o virtuales seleccionando en la barra de opciones de Android Studio build/select build variant el support debug. En la figura 118, se observa que al ejecutar por primera vez posiblemente saldrá el siguiente error “No versión of NDK matched” para lo cual se debe instalar el NDK solicitado y hacer un rebuild del proyecto solo en caso de que aparezca dicho error.

Figura 118

Error no version NDK

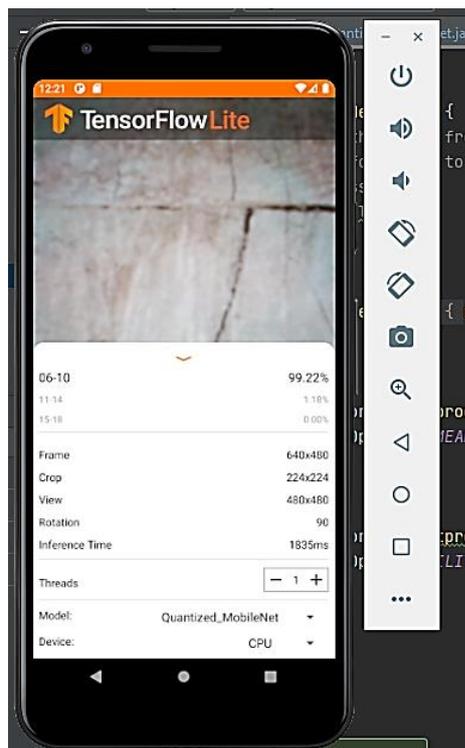


Nota. Esta figura muestra un posible error al ejecutar el proyecto de forma local en un dispositivo virtual.

Una vez que corre el aplicativo móvil se procede a seleccionar el modelo de red neuronal el cual se llama “Quantized_MobileNet” exportado en el Capítulo III (3.7 Etapa 6. Exportación del modelo de red neuronal convolucional CNN) cargándose las configuraciones necesarias permitiéndose un correcto funcionamiento. En la figura 119, se puede observar correr el aplicativo móvil con dichas configuraciones. Permittiéndose medir el contenido de humedad del Laurel a través del aprendizaje profundo (redes neuronales convolucionales CNN).

Figura 119

Ejecución aplicativo móvil en un dispositivo virtual



Nota. Esta figura muestra la ejecución del aplicativo móvil con el código fuente base suministrado por Tensorflow.

4.3.4. Paso 4. Desarrollo de Interfaces.

En este paso se procede a desarrollar las 3 interfaces propuestas en los requisitos de interfaz obtenidos en el Capítulo IV (4.1.3 Requisitos de Interfaz) las cuales son las siguientes:

1) Home o Principal. - encargada de realizar el proceso de medición de contenido de humedad del Laurel.

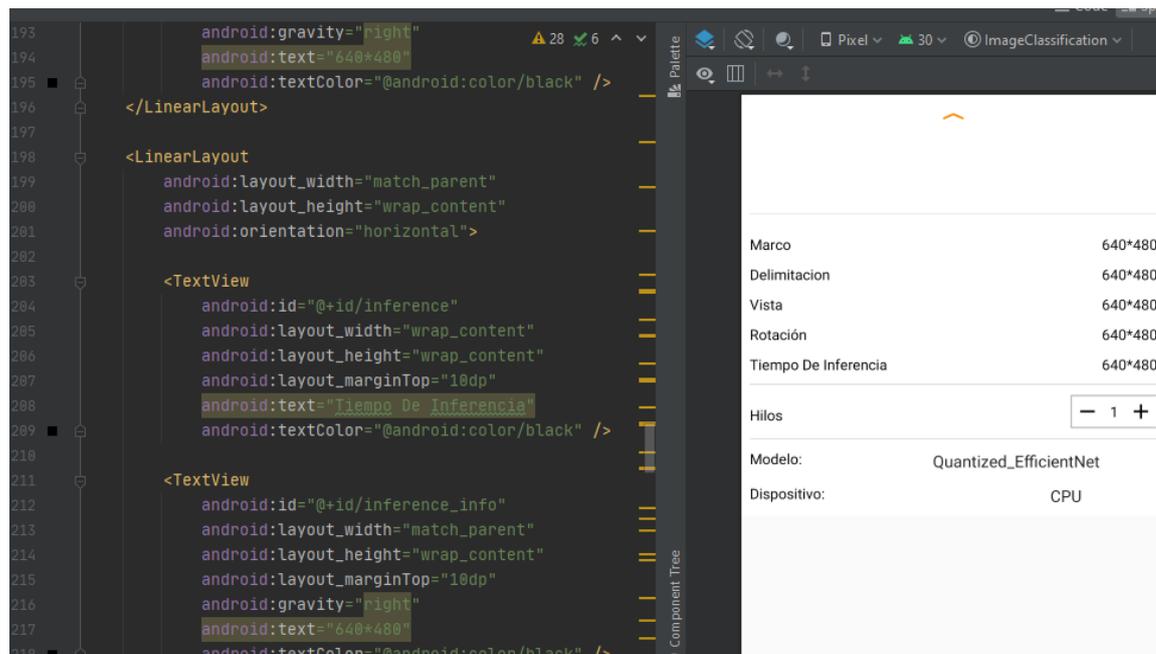
En la Figura 120 se puede visualizar el código utilizado para traducir el menú de la interfaz principal o home que se encuentra por defecto en el idioma Inglés. El mismo que se encuentra en el archivo

“tfe_ic_layout_bottom_sheet.xml”, ubicado en la ruta:

C:\Users\Jonathan\Desktop\TesisMaderaCode\examples\lite\examples\image_classification\android\app\src\main\res\layout.

Figura 120

Traducción del menú principal o home en el aplicativo móvil.

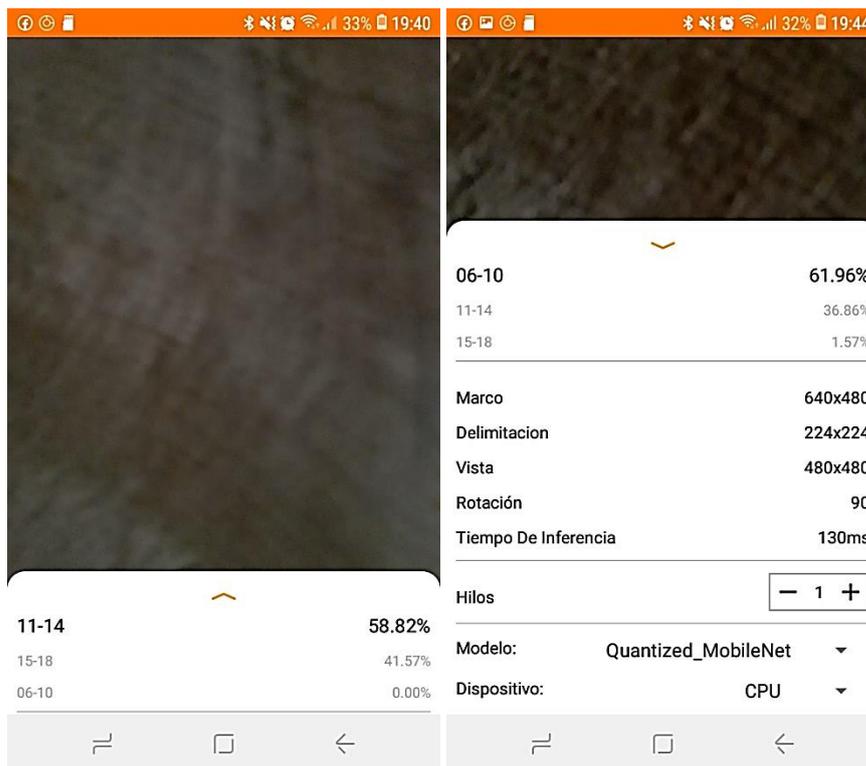


Nota. Esta figura se muestra la traducción del menú principal del aplicativo móvil.

Posteriormente se continua el desarrollo de la interfaz principal que mide el contenido de humedad de la madera. En la figura 121 se presenta a la interfaz completada basándose en el mockup previamente diseñado en el Capítulo IV (4.1.3 Requisitos de Interfaz) haciendo hincapié en que el aplicativo debe ser lo más sencillo y usable para los artesanos del aserradero “San Martín” los cuales serán quienes usen a diario el aplicativo móvil.

Figura 121

Interfaz principal ejecutándose en el aplicativo móvil físico



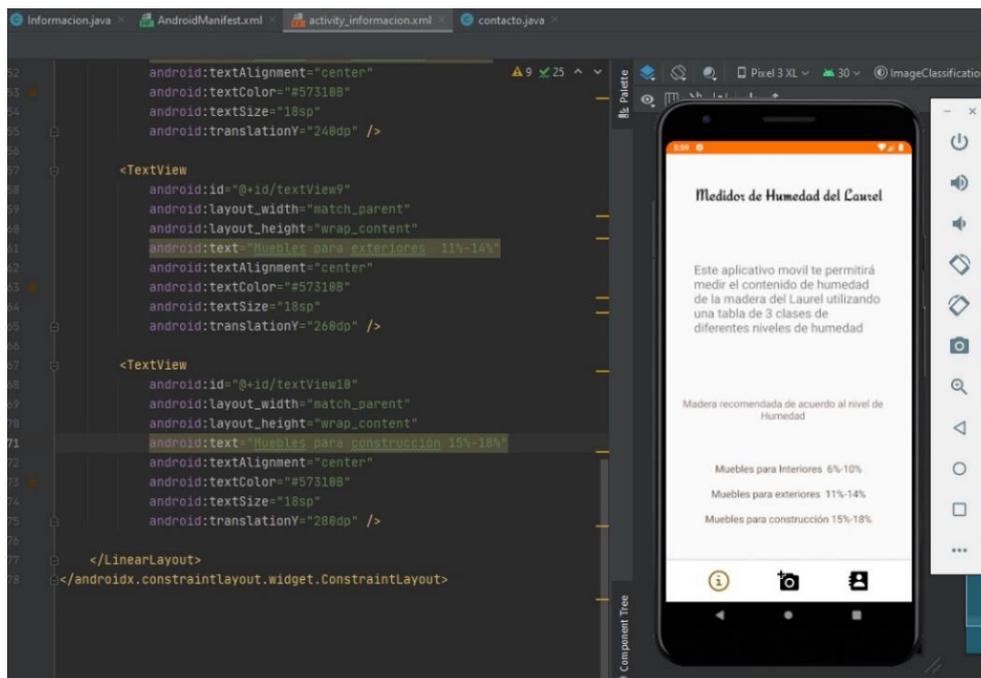
Nota. En esta figura se puede observar la ejecución de la interfaz principal de la aplicación móvil, es decir el módulo de gestión de medidor de humedad.

2) Información. - enfocada en suministrar información teórica a través de una tabla que contendrá las tres clases de porcentaje de humedad de la madera del laurel.

En la Figura 122, se puede observar al desarrollo de la interfaz de información el mismo que se encuentra en el archivo con nombre "activity_informacion" el cual se está ubicado en la ruta C:\Users\Jonathan\Desktop\TesisMaderaCode\examples\lite\examples\image_classification\android\app\src\main\res\layout.

Figura 122

Interfaz Información ejecutándose en un dispositivo virtual



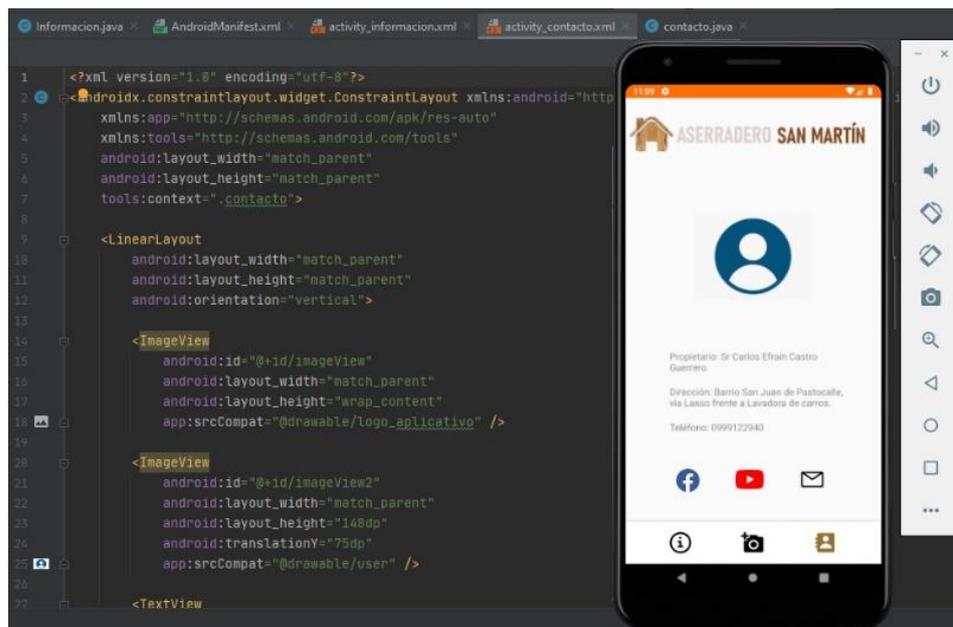
Nota. Esta figura muestra la interfaz de información del aplicativo móvil de esta investigación.

3) Contacto. - orientada en mostrar los datos de contacto del Sr. Carlos Efraín Castro Guerrero propietario del aserradero San Martín.

En la Figura 123, se puede observar al desarrollo de la interfaz de información el mismo que se encuentra en el archivo con nombre "activity_contacto" el cual se está ubicado en la ruta C:\Users\Jonathan\Desktop\TesisMaderaCode\examples\lite\examples\image_classification\android\app\src\main\res\layout.

Figura 123

Interfaz de Contacto ejecutándose en un dispositivo virtual.



Nota. Esta figura muestra la interfaz de contacto del aplicativo móvil desarrollado para esta investigación.

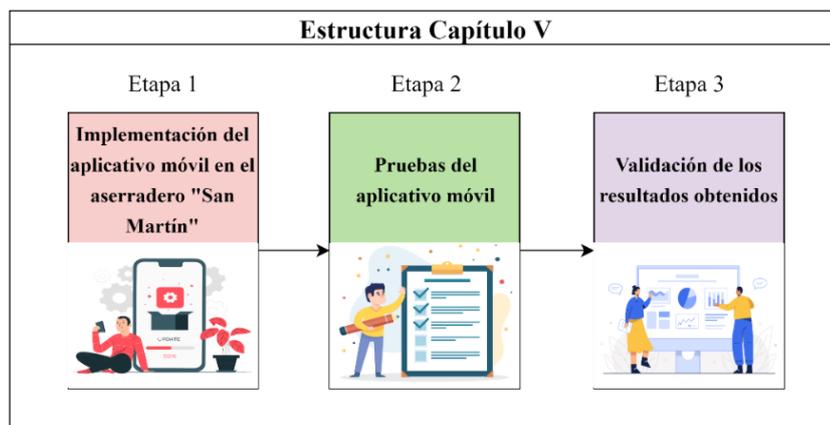
Capítulo V

5. Implementación, pruebas y validación del aplicativo móvil para el diagnóstico de humedad en la madera del Laurel.

En este capítulo se implementa, prueba y valida el aplicativo móvil para el medir el contenido de humedad en el Laurel a través de aprendizaje profundo. Como se puede visualizar en la Figura 124. Este capítulo está compuesto por 3 etapas. A continuación, se listan: 1) Implementación del aplicativo móvil en el aserradero "San Martín", 2) Pruebas del aplicativo móvil y 3) Validación de los resultados obtenidos.

Figura 124

Estructura Capítulo V



Nota. Esta figura muestra la estructura del Capítulo V.

A continuación, se describen las etapas necesarias para implementar, probar y validar el aplicativo móvil para el diagnóstico del contenido de humedad en el laurel.

5.1. Etapa 1. Implementación del aplicativo móvil en el aserradero “San Martín”.

El proceso de secado del Laurel se lo realiza a través de la superposición de cortes de madera (tablones, cuartones) en forma vertical o también denominado parrillas con una altura aproximada de 25m. En la Figura 125 se puede observar cómo se seca la madera a través del tiempo.

Figura 125

Secado de madera en el aserradero “San Martín”



Nota. Esta figura muestra al proceso de secado de la madera del Laurel. Imagen tomada en el aserradero “San Martín”.

El aserradero “San Martín” ubicado en la parroquia de Pastocalle tiene una extensión de aproximadamente $1000m^2$. Además, posee 20 torres de madera del Laurel apiladas como se pudo observar en la figura 125. En la figura 126 se muestra la fachada exterior de dicho establecimiento.

Figura 126

Fachada exterior aserradero “San Martín”.



Nota. Esta figura muestra la vista frontal exterior del aserradero “San Martín” ubicado en la parroquia de Pastocalle provincia de Cotopaxi.

En la figura 127, se muestra al aplicativo instalado. La app móvil para medir el contenido de humedad del Laurel fue implementada en 2 dispositivos móviles. En el dispositivo 1 posee la versión de Android 8.1.0 y el dispositivo 2 posee Android 10, ambos superan la versión mínima permitida como se puede observar en el Anexo 2 Documento de especificación de requisitos de software (Apartado 3 Requisitos Específicos. - en la sección 3.2 Requisitos No Funcionales).

Figura 127

Implementación del aplicativo móvil



Nota. Esta figura muestra a la implementación del aplicativo móvil en dos dispositivos Android.

5.2. Etapa 2. Pruebas del Aplicativo Móvil.

De acuerdo con (Escobar-Sánchez y Fuertes 2015), todo aplicativo software de calidad, debe cumplir con la utilización de pruebas funcionales, sugiriéndose un modelo que está compuesto de 4 fases, las mismas que se listan a continuación: 1) Especificación, 2) Planificación, 3) Ejecución y 4) Evaluación de los resultados.

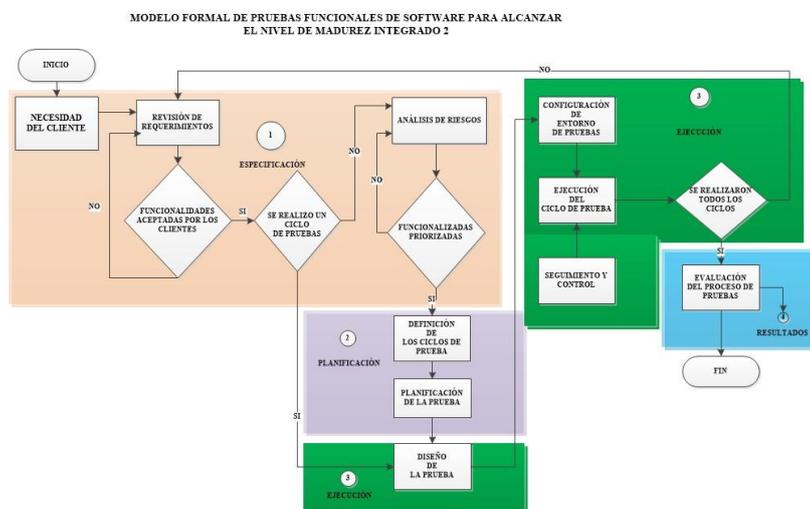
En la figura 128 se puede observar el procedimiento sugerido por (Escobar-Sánchez y Fuertes 2015) para la correcta aplicación de pruebas funcionales. Cabe destacar que dicho modelo permite validar las funcionalidades del aplicativo móvil a través de ciclos y casos de prueba, mitigando los riesgos de posibles fallos. A Continuación, se describen las 4 fases antes mencionadas.

1. Especificación. – Fase en la que se analiza las funcionalidades más importantes del aplicativo móvil, con el fin de establecer el alcance de las pruebas de software y verificar si existiese algún fallo. Cabe destacar que si los artesanos madereros (usuarios) se encuentran de acuerdo con el presente análisis, se continua con el siguiente paso (planificación), caso contrario se debe evaluar las inconformidades suscitadas por dichos artesanos.
2. Planificación. – Fase en la que se realiza una planificación de tiempo para realizar correcciones del aplicativo móvil, estableciéndose ciclos de prueba.
3. Ejecución. – Fase en la que se desarrolla las pruebas de una version del aplicativo móvil, según lo planificado de la fase anterior.
4. Evaluación de los resultados. – Fase en la que se menciona el número de fallos, los cuales son corregidos por los desarrolladores de esta investigación en caso de existir. Obteniéndose una version final del aplicativo móvil libre de errores, el cual cumpla con los requerimientos de

los artesanos madereros. Garantizándose un desarrollo de software con calidad.

Figura 128

Modelo sugerido para pruebas funcionales de aplicaciones software



Nota. Esta figura muestra el modelo formal de pruebas funcionales de software para alcanzar el nivel de madurez integrado 2. Tomado de (Escobar-Sánchez y Fuertes 2015).

A continuación, se describe el proceso realizado para la ejecución de las pruebas del aplicativo móvil de esta investigación. Tomado de (Escobar-Sánchez y Fuertes 2015).

Pruebas de Aceptación. - Pruebas que permiten verificar el correcto funcionamiento de las 3 gestiones especificadas previamente en el Anexo 2 Documento de especificación de requisitos de software. Dichas pruebas se convierten en 3 casos de pruebas correspondientes al desarrollo del aplicativo móvil.

Cabe destacar que las pruebas se las realiza conjuntamente con el Sr. Francisco Trajano Muñoz Pila experto artesano quien cuenta con la experiencia necesaria para la toma de decisiones técnicas, respecto al correcto funcionamiento del aplicativo móvil. Para lo cual los 3 casos de prueba se lo

evalúan en un periodo de tres días, efectuándose el día miércoles 21 de julio del 2021 en el aserradero “San Martin”, como se puede visualizar en la Tabla 22.

Tabla 22

Casos de prueba

Caso de Prueba - Días	1	2	3
Gestión de información			
Gestión de medidor de humedad			
Gestión de contacto			

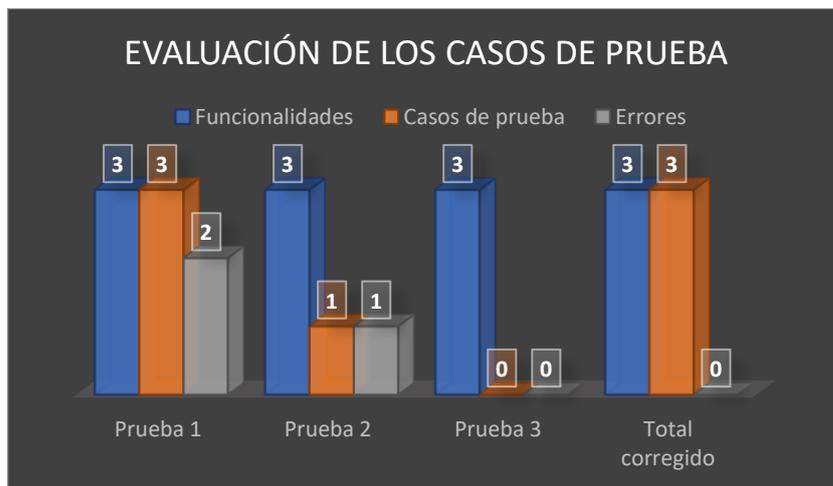
Nota. En esta tabla se muestra la planificación de los casos de prueba.

Como se pudo observar en la tabla 22, los casos de prueba son validados por el experto artesano desde el miércoles 21 de julio hasta el viernes 24 de julio del 2021 respectivamente.

En la figura 129, se puede visualizar la verificación y evaluación de las funcionalidades del aplicativo móvil, las cuales son testeadas en su totalidad por el Sr. Francisco Trajano Muñoz Pila experto artesano. Determinándose que para la prueba 1 existe un fallo en una funcionalidad (Gestión de medidor de humedad), la cual no cumple con la aceptación de dicho experto maderero, siendo el 33% de fallos en el aplicativo móvil, por tal razón los desarrolladores realizan las respectivas correcciones. En la prueba 2 se verifica que el fallo encontrado en la prueba anterior este totalmente corregido llegándose a aceptar dicha funcionalidad. Pasando a la prueba 3 la cual ya no cuenta con ningún fallo. Cumpliéndose a cabalidad con lo establecido en el Anexo 2 Documento de especificación de requisitos de software mitigándose cualquier tipo de error, gracias a la correcta implementación del modelo propuesto por (Escobar-Sánchez y Fuertes 2015).

Figura 129

Evaluación de casos de prueba.



Nota. Esta figura muestra las 3 pruebas realizadas a las funcionalidades del aplicativo móvil.

Una vez aprobado el aplicativo móvil por el Sr. Francisco Trajano Muñoz Pila experto artesano se elabora el documento de aceptación y finalización el mismo que se encuentra en el Anexo 6 (Carta de Aceptación y finalización del proyecto de titulación Aserradero San Martín), A continuación, se procede a realizar pruebas técnicas en el aserradero “San Martín”, con el objetivo de validar la efectividad del algoritmo de red neuronal convolucional (CNN), implementado previamente en el Capítulo IV (Apartado 4.3.3. Paso 3. Implementación del modelo de red neuronal convolucional CNN). Cabe destacar que para el desarrollo de esta etapa se utiliza las 3 clases de madera mencionadas previamente el Capítulo III (Apartado 3.2. Etapa 1. Recolección y captura de imágenes) siendo estas clases: 1) Seco (6%-10% de humedad), 2) Amorochado (11%-14% de humedad), y 3) Chapoteado (15%-18% de humedad) como se puede observar en la Tabla 23.

Tabla 23

Clases Agrupadas por el porcentaje de humedad del Laurel.

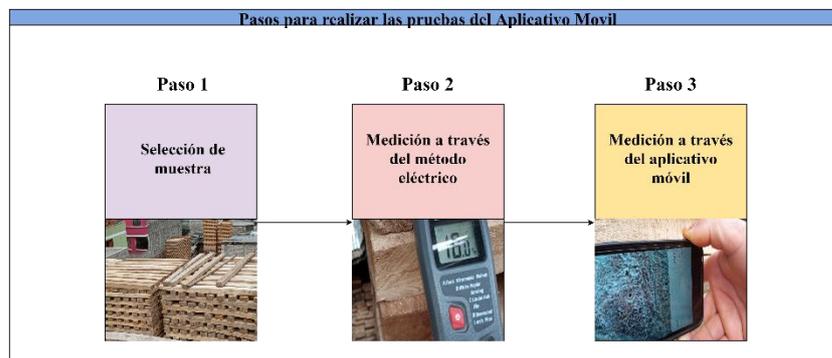
Clase	Imagen
<p>Seco 6%-10% de humedad</p>	
<p>Amorochado 11%-14% de humedad</p>	
<p>Chapoteado 15%-18% de humedad</p>	

Nota. Esta tabla muestra las 3 clases de madera agrupadas por el nivel de humedad del Laurel.

En la Figura 130 se puede observar los pasos necesarios para probar el aplicativo móvil. Entre ellos se lista a continuación: 1) Selección de muestra, 2) Medición a través del método eléctrico y 3) Medición a través del aplicativo móvil.

Figura 130

Pasos para realizar las pruebas del aplicativo móvil.



Nota. Esta figura muestra los pasos necesarios para probar el aplicativo móvil.

A continuación, se detalla los pasos necesarios para probar el aplicativo móvil.

Paso 1. Selección de muestra. – En la Figura 131, se puede observar el proceso de selección de muestra. Para lo cual se escoge la pieza de madera del laurel. Mediante indicaciones dadas por el experto artesano basándose en su experiencia. Considerando sus características físicas como el color y su textura.

Figura 131

Selección de muestra



Nota. En esta figura se presenta el proceso de selección de muestra de madera del Laurel.

Paso 2. Medición a través del método eléctrico. – En la figura 132, se observa cómo se introduce el higrómetro en la muestra de madera, el cual a través del método de resistencia eléctrica mide el contenido de humedad, llegando a ser un método destructivo por ingresar 2 puntas de acero que perfora dicho material. Dando como resultado el porcentaje de humedad del Laurel.

Figura 132

Medición a través del método eléctrico.

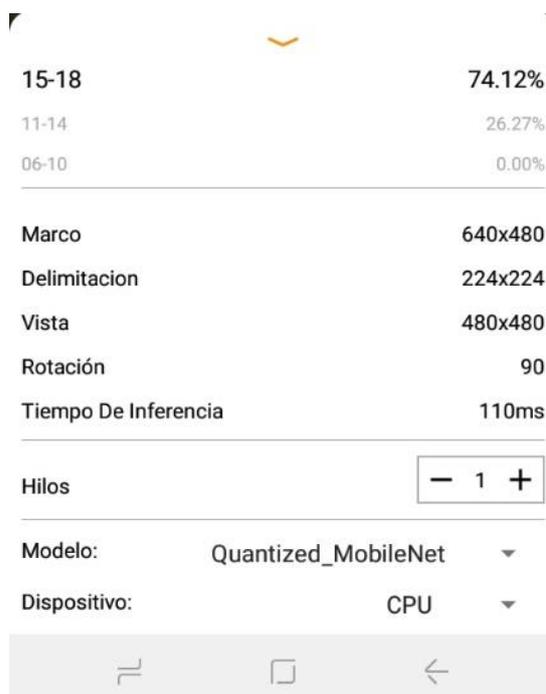


Nota. Esta figura muestra la medición de humedad del Laurel utilizando el higrómetro y el método eléctrico.

Paso 3. Medición a través del aplicativo móvil. – En la figura 133, se procede a utilizar el aplicativo móvil desarrollado en esta investigación, seleccionando la opción “Quantized_MobileNet” que corresponde al modelo generado previamente en el Capítulo III (3.7. Etapa 6. Exportación del modelo de red neuronal).

Figura 133

Configuración y Selección del modelo de Red Neuronal Convolutacional (CNN) en el aplicativo móvil.

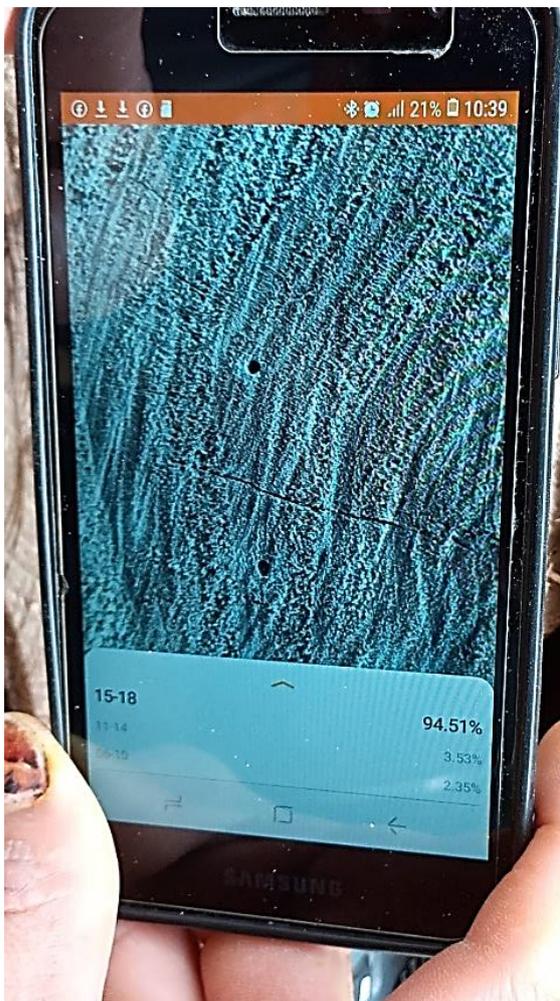


Nota. Esta imagen muestra la interfaz de configuración y selección del modelo de red neuronal del aplicativo móvil.

Una vez seleccionado el modelo de red neuronal convolutacional (CNN) en el aplicativo móvil se debe acercar el dispositivo a un rango de distancia entre 5 y 10 cm aproximadamente, esperando que se enfoque correctamente la muestra a medir. En la figura 134, se observa el proceso de medición de humedad del Laurel utilizando el aplicativo móvil.

Figura 134

Medición del contenido de humedad utilizando el aplicativo móvil.



Nota. Esta figura muestra la utilización del aplicativo móvil para medir del contenido de humedad de la madera del Laurel.

El proceso de pruebas: 1) Selección de muestra, 2) medición del nivel de humedad a través del método eléctrico y 3) medición del nivel de humedad mediante del aplicativo móvil, se lo realiza en 16 ocasiones las mismas que se encuentran disponibles en el Anexo 3 (Aplicación de las pruebas del aplicativo móvil). Debido al peligro existente al no contar con un arnés de seguridad, que permita realizar más mediciones.

Como se pudo observar en la figura 130, existen 3 pasos necesarios para poder probar el aplicativo móvil. Los mismos que solo se repiten en 16 ocasiones y están ubicadas en el Anexo 3 (Aplicación de las pruebas del aplicativo móvil), debido a la poca seguridad que se tiene al treparse en las parrillas de madera y no contar con al menos un arnés, siendo un peligro constante de posible caída para los desarrolladores de esta investigación. Para la demostración de dicho proceso se selecciona una muestra de cada clase como se puede observar en la Tabla 24.

Tabla 24

Pruebas del aplicativo Móvil

Valor # Esperado	Valor Medido	Método Eléctrico	Aplicativo Móvil	Salida
1 Seco 6%-10% De humedad	9.2%			Ok

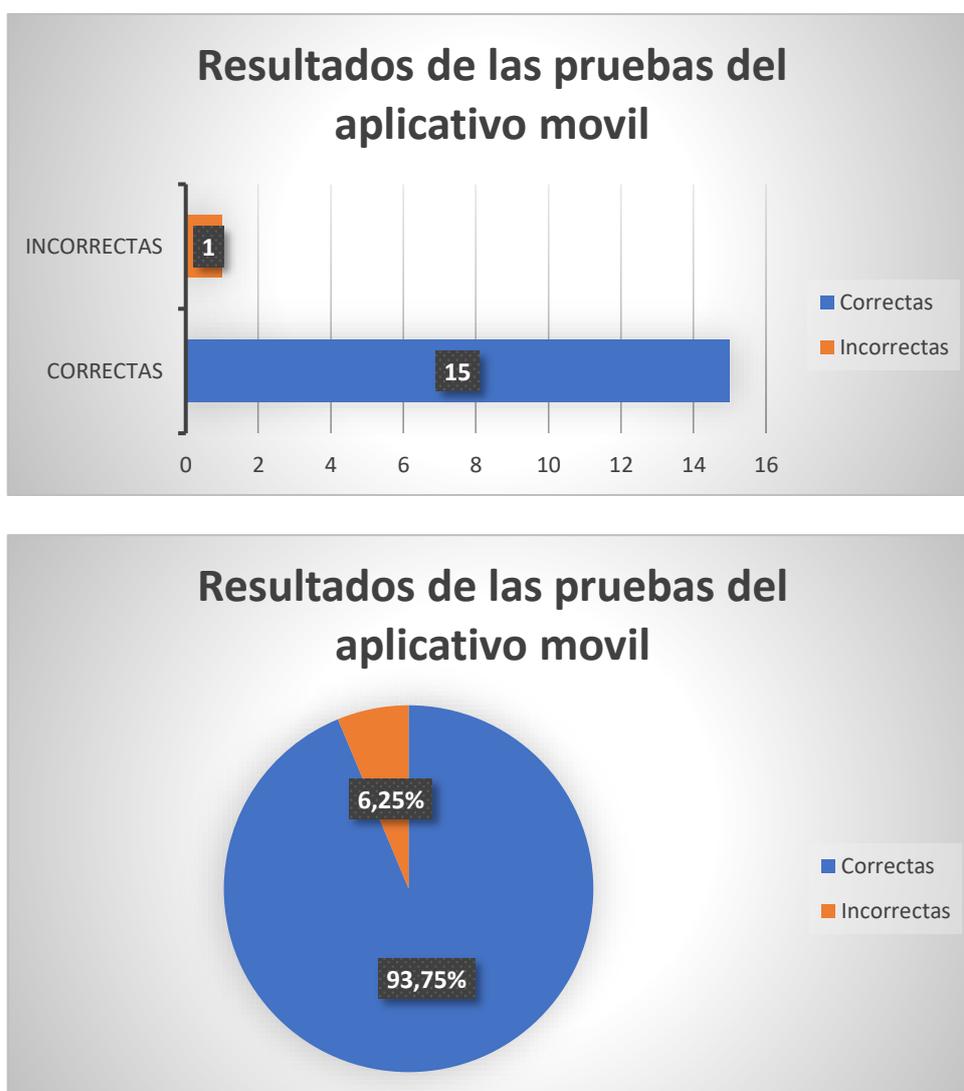
Valor # Esperado	Valor Medido	Método Eléctrico	Aplicativo Móvil	Salida
Amorochado o 2 11%-14% de humedad	12.8 %			Ok
Chapotado o 4 15%-18% de humedad	17.3 %			Ok

5.3. Etapa 3. Validación de los resultados obtenidos.

En la figura 135, se muestra los resultados de las pruebas realizadas al aplicativo móvil. Una vez aplicadas 16 pruebas para medir el contenido de humedad de la madera del Laurel en el aserradero “San Martín”, se obtiene que 15 pruebas fueron exitosas correspondientes al 93.75%. Mientras que en 1 sola prueba hubo un error en la medición (debido a la presencia de moho) siendo el 6.25% de error respectivamente.

Figura 135

Resultados de las pruebas del aplicativo móvil.



Nota. Esta gráfica muestra los resultados de las pruebas del aplicativo móvil.

Las muestras de madera que son seleccionadas para el proceso de medición de humedad deben estar libres de imperfecciones como: suciedad, grietas y enfermedades ya que dicha medición puede llegar a ser errónea como se pudo observar en la prueba número 12 ubicada en la tabla del Anexo 3 (Aplicación de las pruebas del aplicativo móvil). En la figura 136 se puede visualizar la muestra que contiene en su estructura la enfermedad del Moho.

Figura 136

Muestra con Moho



Nota. En esta imagen se observa la muestra utilizada en la prueba número 12 la cual arroja un resultado incorrecto a causa de la existencia de moho.

5.3.1. Aplicación de Encuesta de Usabilidad SUS

Para evaluar el nivel de usabilidad del aplicativo móvil resulta necesario la aplicación de encuestas S.U.S o también llamado Escala de Usabilidad de un Sistema a 10 expertos madereros del aserradero “San Martín” para lo cual se elaboró una encuesta 10 preguntas las mismas que se encuentran ubicadas en el Anexo 4 (Encuesta SUS).

Después de ejecutar las encuestas a los expertos madereros se tabularon dichos resultados, los mismo que se encuentran ubicados en el Anexo

5 (Aplicación de la encuesta SUS). Para lo cual se toma en cuenta las ponderaciones de las 10 preguntas de esta encuesta como se puede observar en la tabla 25.

Tabla 25

Tabla de tabulación de los valores en la encuesta SUS

N.-	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10
1	5,00	1,00	5,00	5,00	5,00	1,00	5,00	1,00	5,00	5,00
2	5,00	2,00	5,00	4,00	5,00	1,00	5,00	2,00	5,00	1,00
3	5,00	2,00	5,00	5,00	5,00	3,00	5,00	2,00	5,00	1,00
4	5,00	2,00	5,00	2,00	5,00	2,00	5,00	2,00	5,00	2,00
5	4,00	2,00	5,00	2,00	5,00	2,00	4,00	3,00	4,00	2,00
6	4,00	1,00	4,00	1,00	5,00	1,00	4,00	1,00	5,00	1,00
7	5,00	2,00	4,00	2,00	5,00	2,00	4,00	1,00	5,00	1,00
8	4,00	2,00	4,00	1,00	4,00	2,00	5,00	1,00	5,00	1,00
9	4,00	1,00	3,00	1,00	4,00	1,00	4,00	2,00	5,00	3,00
10	5,00	2,00	5,00	3,00	4,00	1,00	5,00	2,00	4,00	2,00
Promedio	4,60	1,70	4,50	2,60	4,70	1,60	4,60	1,70	4,80	1,90

Nota. Esta tabla muestra los valores obtenidos a través del proceso de tabulación de cada pregunta en la encuesta SUS.

Para el cálculo de grado de usabilidad de este aplicativo móvil mediante la utilización de la encuesta SUS, se procede a aplicar el siguiente proceso matemático como se observa en la ecuación 7.

Ecuación 7

Cálculo grado de usabilidad

$$SUS = ((4.60 - 1) + (5 - 1.70) + (4.50 - 1) + (5 - 2.60) + (4.70 - 1) + (5 - 1.60) + (4.60 - 1) + (5 - 1.70) + (4.80 - 1) + (5 - 1.90)) * 2.5$$

$$SUS = (84.25)$$

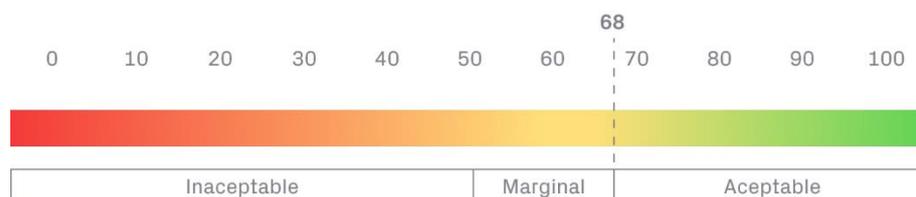
Nota. Esta ecuación sirve para calcular el grado de usabilidad de un producto.

En la figura 137, se puede observar la escala de usabilidad utilizada para evaluar los sistemas software por lo tanto una vez aplicada la encuesta, la

tabulación de información y finalmente el cálculo de grado de usabilidad se obtiene un resultado de 84.25% lo que equivale a que el aplicativo móvil es un sistema sobresaliente.

Figura 137

Escala de Usabilidad



Nota. Esta figura muestra el grado de usabilidad mediante la aplicación de la encuesta SUS. Tomado de (Busquets 2021).

Capítulo VI

6. Conclusiones y Recomendaciones.

En el presente capítulo se detallan las conclusiones y recomendaciones obtenidas en el desarrollo de la investigación.

6.1. Conclusiones.

- Se logró cumplir con el objetivo general planteado, desarrollar un aplicativo móvil que optimice el diagnóstico de humedad en el laurel mediante aprendizaje profundo en el aserradero “San Martín” ubicado en la parroquia Pastocalle provincia de Cotopaxi.
- El desarrollo de una revisión sistemática es fundamental y necesaria como etapa inicial en un proyecto de investigación. Permite obtener recursos científicos actuales, de gran impacto investigativo, que son fuentes confiables de información. Coadyuvando a la consecución de una solución confiable, eficiente y eficaz.
- La construcción del marco teórico permitió entender los conceptos fundamentales con respecto a la madera, técnicas de medición de contenido de humedad, aplicaciones móviles, inteligencia artificial, y técnicas de aprendizaje profundo. Lo que contribuyó notablemente en la elaboración y consecución de la presente investigación.
- El modelo de redes neuronales convolucionales (CNN) que presenta los mejores resultados en la clasificación y diagnóstico de contenido de humedad en el laurel es generado por la herramienta Google Teachable Machine, obteniendo un 97.68% de efectividad.
- La construcción un conjunto óptimo de datos(dataset) de imágenes de madera del laurel, conlleva la eliminación de factores externos (basura, polvo, moho)

así como también una adecuada configuración fotográfica en lo referente a la iluminación y su contraste, repercutiendo en la calidad de la misma.

- La correcta aplicación de las métricas de evaluación (Matriz de confusión, F1 score y precisión) para modelos de redes neuronales convolucionales (CNN). Permite determinar matemáticamente los porcentajes de aprendizaje, evaluando su efectividad. Estableciéndose que la métrica F1 es la más adecuada debido a que facilita comparar el rendimiento combinado de la precisión y la exhaustividad de valores obtenidos.
- La utilización de la metodología de desarrollo rápido de aplicaciones (RAD) es de gran utilidad para el desarrollo del aplicativo móvil propuesto en esta investigación. Porque permite la entrega de prototipos funcionales, utilizando criterios de calidad (usabilidad), en el menor tiempo posible.
- La utilización del estándar IEEE 830 facilita la extracción de características y necesidades tecnológicas del aserradero “San Martin”. Caracterizando las funcionalidades de forma coherente y ordenada. Formalizando también los acuerdos entre los desarrolladores y clientes.
- La recopilación de características y necesidades tecnológicas del aserradero “San Martin” son extraídas a través de la utilización del estándar de especificación de requisitos software IEEE 830. Porque permite establecer un acuerdo con el dueño de dicho establecimiento respecto a las funcionalidades que tendría el aplicativo móvil de esta investigación.
- El correcto levantamiento de requisitos de software mediante el estándar IEEE 830, se ve reflejado en el grado de satisfacción y facilidad de uso del aplicativo móvil de los artesanos y trabajadores del aserradero “San Martin”.
- La arquitectura monolítica que se utiliza en la presente investigación, permite generar un producto software de calidad destacándose por su eficiencia al

ser un aplicativo en tiempo real, autosuficiente y no requerir de servicios externos.

- La herramienta Google Teachable Machine permite a través de su repositorio de Github descargar el código fuente necesario, facilitando la implementación de modelos de redes neuronales convolucionales (CNN) para aplicaciones móviles.
- La guía técnica de un experto maderero permite una correcta ejecución y validación de la información generada como conjunto de datos(dataset), haciendo énfasis en la veracidad de este tema de investigación.
- Las pruebas de funcionalidad permiten comprobar el comportamiento que el aplicativo móvil presenta al interactuar con sus usuarios (artesanos), garantizando que las características y funcionalidades de dicho aplicativo se comporten según lo esperado.
- El aplicativo móvil desarrollado en esta investigación sobre el sistema operativo Android, permite realizar el diagnóstico del contenido de humedad en el laurel en tiempo real, con una efectividad 93.75%. Llegando a ser una herramienta tecnológica confiable para los trabajadores del aserradero "San Martín". Optimizando el proceso de selección de materia prima.
- La utilización de las encuestas de usabilidad de un sistema denominado (SUS), permiten de manera sencilla conocer del grado de satisfacción que los usuarios perciben al utilizar un aplicativo móvil. Obteniéndose un 84.25% lo que corresponde a ser un aplicativo móvil excelente.

6.2. Recomendaciones.

- Se recomienda expandir el alcance de este aplicativo móvil en más especies madereras, con el objetivo de optimizar el proceso de selección de materia prima. Reduciéndose el impacto ambiental generado a causa del desperdicio innecesario de dicha madera.

- Se recomienda generar un conjunto de datos(dataset), utilizándose una cámara digital profesional o su vez un dispositivo móvil con una capacidad superior a 12 megapíxeles para una óptima captura de imágenes.
- Se recomienda realizar el ajuste de hiperparámetros en la etapa de entrenamiento, dado que nos permite verificar los resultados que arrojan dichos hiperparámetros, evitando el underfitting y el overfitting.
- Se recomienda utilizar la métrica de evaluación (F1 Score) para modelos de redes neuronales convolucionales (CNN), pues esta ofrece los mejores resultados de efectividad en comparación con la métrica de precisión(acurracy). Dado que se realiza un cálculo matemático entre las métricas acurracy y recall.
- Se recomienda la implementación de la metodología de desarrollo rápido de aplicaciones (RAD) para productos software los cuales deben ser desarrollados en periodos de tiempo cortos y cuentan con la guía del cliente en cada fase de desarrollo.
- Se recomienda que la pieza de madera del laurel que se desee medir el contenido de humedad no disponga en su estructura: enfermedades, grietas, y suciedad pues estas pueden afectar directamente en la precisión del aplicativo móvil.

7. Bibliografía

- Abellán, Encarna. 2020. «Scrum: qué es y cómo funciona esta metodología». Recuperado 22 de junio de 2021 (<https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>).
- Aguilar Pozzer, Jorge, y Estela Guzowski. 2017. «Materiales y materias primas. Madera». *VirtualPro.co*. Recuperado 10 de junio de 2021 (<https://www.virtualpro.co/biblioteca/materiales-y-materias-primas-madera>).
- Airthings. 2021. «¿Qué es la humedad?» *Airthings*. Recuperado 15 de junio de 2021 (<https://www.airthings.com/es/what-is-humidity>).
- Albornoz, Félix. 2019. «Mejores Sistemas Operativos para Móviles 2021». *Internet Paso a Paso*. Recuperado 17 de junio de 2021 (<https://internetpasoapaso.com/mejores-sistemas-operativos-moviles/>).
- Álvarez, Raúl. 2016. «Marvin Minsky, el padre de la inteligencia artificial, muere a los 88 años». *Xataka*. Recuperado 23 de junio de 2021 (<https://www.xataka.com/robotica-e-ia/marvin-minsky-el-padre-de-la-inteligencia-artificial-muere-a-los-88-anos>).
- Anani-Manyo, Nina K. 2021. «Computer Vision and Building Envelopes». Kent State University.
- Araujo, Miguel. 2011. «Las revisiones sistemáticas». *Medwave* 11(11). doi: 10.5867/medwave.2011.11.5220.
- Arciniega, Fernando. 2017. «Normas y Estándares de calidad para el desarrollo de Software». *Fernando Arciniega*. Recuperado 17 de junio de 2021 (<http://fernandoarciniega.com/normas-y-estandares-de-calidad-para-el-desarrollo-de-software/>).
- Arquitectes, I. T. E. 2018. «Estructuras de madera I/III. El material y su capacidad estructural». *ITE ARQUITECTES*. Recuperado 10 de junio de 2021 (<https://itearquitectes.com/2018/09/18/estructuras-de-madera-el-material-y-las-estructuras/>).
- Bagnato, Juan Ignacio. 2019. «Convolutional Neural Networks: La Teoría explicada en Español | Aprende Machine Learning». *Aprende Machine Learning*. Recuperado 24 de junio de 2021 (<https://www.aprendemachinelarning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>).
- Bakshi, Ashish. 2017. «What Is Deep Learning | Getting Started With Deep Learning». *Eureka*. Recuperado 24 de junio de 2021 (<https://www.edureka.co/blog/what-is-deep-learning>).
- Balasubramaniam, Kartik. 2019. «Have You Taught Your Machine Yet?» *Medium*. Recuperado 28 de junio de 2021 (<https://towardsdatascience.com/have-you-taught-your-machine-yet-45540b7e646b>).
- Balboa Huanca, Maritza Marleny, Minaya Rivera, Iberth Alexander, Muñoz Ulfe, Patrick Axel, Pacora Marrón, Pedro Eduardo, y Quintos Muñoz, Carmen Analí.

- s. f. «Reconocimiento de imágenes a través de inteligencia artificial». *Comunidad Huawei Enterprise*. Recuperado 9 de junio de 2021 (<https://forum.huawei.com/enterprise/es/reconocimiento-de-im%C3%A1genes-a-trav%C3%A9s-de-inteligencia-artificial/thread/733915-100325>).
- Banafa, Ahmed. 2016. «¿Qué es el aprendizaje profundo?» *OpenMind*. Recuperado 24 de junio de 2021 (<https://www.bbvaopenmind.com/tecnologia/mundo-digital/que-es-el-aprendizaje-profundo/>).
- Barrios Arce, Juan Ignacio. 2019. «La matriz de confusión y sus métricas – Inteligencia Artificial –». *Juan Barrios*. Recuperado 24 de junio de 2021 (<https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/>).
- Barrios, Juan. 2019. «Redes neuronales convolucionales son un tipo de redes neuronales». *Juan Barrios*. Recuperado 24 de junio de 2021 (<https://www.juanbarrios.com/redes-neurales-convolucionales/>).
- Bermejo, Félix. 2014. *Humedad en la Madera*. Madrid: Tableros y Molduras Félix Bermejo SL.
- Bessaa, B., M. Belhadj Aissa, R. Amara, y A. Belhadj Aissa. 2013. «Spatial Indexing of Static Maps for Navigation in Online GIS: Application for Tourism Web GIS». *International Journal of Computer Applications in Technology* 47(2/3):189. doi: 10.1504/IJCAT.2013.054351.
- Blancarte, Oscar. 2020. «Arquitectura de Monolítico». *Reactive Programming*. Recuperado 8 de julio de 2021 (<https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/monolitico>).
- Busquets, Cris. 2021. «Cómo medir la usabilidad con un SUS». *uiFromMars*. Recuperado 21 de julio de 2021 (<https://www.uifrommars.com/como-medir-usabilidad-que-es-sus/>).
- Callejas-Cuervo, Mauro, Andrea Catherine Alarcón-Aldana, y Ana María Álvarez-Carreño. 2016. «Modelos de calidad del software, un estado del arte». *Entramado* 13(1):236-50. doi: <https://doi.org/10.18041/entramado.2017v13n1.25125>.
- Calvo, Beatriz. 2018. «Tutorial Draw.io - La mejor herramienta para hacer diagramas». *Diseño web y Branding en Alicante*. Recuperado 28 de junio de 2021 (<https://beatrizcalvo.com/tutorial-draw-io-herramienta-diagramas/>).
- Cantone, Dante. 2006. *La Biblia del Programador / Implementación y Debugging*. 1.^a ed. Creative Andina Corp.
- Cárdenas, Ávila. 2014. «Tecnología de los materiales». Recuperado 9 de junio de 2021 (<http://tecnolouap.blogspot.com/2014/12/maderas.html>).
- Carney, Michelle, Barron Webster, Irene Alvarado, Kyle Phillips, Noura Howell, Jordan Griffith, Jonas Jongejan, Amit Pitaru, y Alexander Chen. 2020. «Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification». Pp. 1-8 en.

- Cervantes, Humberto. 2020. «Arquitectura de Software». *SG Buzz*. Recuperado 16 de julio de 2021 (<https://sg.com.mx/revista/27/arquitectura-software>).
- Cilsa. 2021. «¿Qué es un sistema operativo? | Desarrollar Inclusión». *Desarrollar Inclusión | Portal de tecnología inclusiva de CILSA*. Recuperado 1 de agosto de 2021 (<https://desarrollarinclusion.cilsa.org/tecnologia-inclusiva/que-es-un-sistema-operativo/>).
- CISET. 2019. «Software - Concepto y tipos». Recuperado 16 de junio de 2021 (<https://www.ciset.es/glosario/480-software>).
- Collado, Christian. 2021. «Android Studio, guía de iniciación: qué es, cómo descargar e instalar, y 4 cosas que puedes hacer con él». *Andro4all*. Recuperado 28 de junio de 2021 (<https://andro4all.com/guias/android/android-studio-descargar-instalar-guia-trucos>).
- Comunycarse. 2020. «Inteligencia Web (Webint)». *Comunycarse.com*. Recuperado 9 de agosto de 2021 (<https://www.comunycarse.com/es/inteligencia-web-webint/>).
- Conceptodefinicion.de. 2021. «¿Qué es Humedad?» Su Definición y Significado [2021]». *Concepto de - Definición de*. Recuperado 15 de junio de 2021 (<https://conceptodefinicion.de/humedad/>).
- Da Silva, Douglas. 2021. «Las 5 herramientas de software comúnmente utilizadas». *Zendesk*. Recuperado 24 de junio de 2021 (<https://www.zendesk.com.mx/blog/herramientas-software-comunmente-utilizadas/>).
- Datision. 2020. «La visión por computador como palanca para la transformación en Industria 4.0». *Datision*. Recuperado 24 de junio de 2021 (<https://datision.com/blog/vision-por-computador-industria40/>).
- Devin, Fabio. 2017. «Sistema de Escalas de Usabilidad: ¿qué es y para qué sirve? | UXpañol». Recuperado 20 de julio de 2021 (<https://uxpanol.com/teoria/sistema-de-escalas-de-usabilidad-que-es-y-para-que-sirve/>).
- DiagramasUML. 2019. «Desarrollo rápido de aplicaciones (RAD): ¿Qué es y cómo funciona?». *DiagramasUML.com*. Recuperado 22 de junio de 2021 (<https://diagramasuml.com/desarrollo-rapido-de-aplicaciones-rad-que-es-y-como-funciona/>).
- DiagramasUML. 2021. «Diagrama de paquetes». *Diagramas UML*. Recuperado 8 de julio de 2021 (<https://diagramasuml.com/paquetes/>).
- Díaz Caneja, José Manuel. 2020. «HUMINT y Virtual HUMINT: dos mundos paralelos-I+L». *Inteligencia más Liderazgo*. Recuperado 9 de agosto de 2021 (<https://inteligenciayliderazgo.com/humint/humint-y-virtual-humint-dos-mundos-paralelos/>).
- Digital Guide IONOS. 2019. «Extreme Programming: desarrollo ágil llevado al extremo». *IONOS Digitalguide*. Recuperado 22 de junio de 2021 (<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/extreme-programming/>).

- Digital Guide IONOS. 2020. «Depuradores: importantes herramientas para detectar errores de software». *IONOS Digitalguide*. Recuperado 24 de junio de 2021 (<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/depurador/>).
- Ecuador Forestal. 2012. «Ficha Técnica N° 4: LAUREL». Recuperado 16 de junio de 2021 (<https://ecuadorforestal.org/fichas-tecnicas-de-especies-forestales/ficha-tecnica-no-4-laurel/>).
- EDS. 2021. «¿Qué es la Robótica y para qué sirve? ▷ ¿Qué es un robot? | EDS». *EDS Robotics*. Recuperado 1 de agosto de 2021 (<https://www.edsrobotics.com/blog/que-es-la-robotica/>).
- Escobar-Sánchez, Milton Eduardo. 2020. «Comparativa del uso de Algoritmos de Aprendizaje Automático en la detección de noticias falsas contrastando la información de uno de los principales medios digitales de la ciudad de Quito [Tesis de maestría no publicada]». Universidad Internacional de la Rioja (UNIR), Latacunga, Ecuador.
- Escobar-Sánchez, Milton Eduardo, y Jose Luis Carrillo-Medina. 2017. «Expert system for the control of physiological signals of aircraft pilots: A systematic mapping study».
- Escobar-Sánchez, Milton Eduardo, y Walter Fuertes. 2015. «Modelo formal de pruebas funcionales de software para alcanzar el Nivel de Madurez Integrado 2». *REVISTA FACULTAD DE INGENIERÍA* 24:31. doi: 10.19053/01211129.3549.
- Escribano, David. 2018. «Esta es la historia de las aplicaciones móviles». *Skyscanner Espana*. Recuperado 17 de junio de 2021 (<https://www.skyscanner.es/noticias/esta-es-la-historia-de-las-aplicaciones-moviles>).
- European Valley. 2018. «¿Qué es un Compilador en programación? – Blog Europeanvalley». *European Valley*. Recuperado 24 de junio de 2021 (<https://www.europeanvalley.es/noticias/que-es-un-compilador-en-programacion/>).
- Fernández, Yúbal. 2019. «Qué es Github y qué es lo que le ofrece a los desarrolladores». *Xataka*. Recuperado 25 de junio de 2021 (<https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>).
- Foglia Córdoba, Rafael. 2005. «Conceptos básicos sobre el secado de la madera | Revista Forestal Mesoamericana Kurú».
- Frankenfield, Jake. 2021. «How Artificial Intelligence Works». *Investopedia*. Recuperado 23 de junio de 2021 (<https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp>).
- Gamez Penadillo, Joel. 2009. «Evaluación de las características físicas del *Calycophyllum spruceanum* (Benth) Hooker F. ex Chuman "Capirona" a tres niveles del fuste, en la zona de Tingo María.» *Universidad Nacional Agraria de la Selva* 80.

- García, Andrea. 2019. «Las Herramientas de Versionamiento de Código». *Medium*. Recuperado 24 de junio de 2021 (https://medium.com/@andreagarcia_94938/las-herramientas-de-versionamiento-de-c%C3%B3digo-8a96a88d1702).
- García, Fernán. 2020. «¿Qué es la arquitectura del software?» *Blog de arsys.es*. Recuperado 16 de julio de 2021 (<https://www.arsys.es/blog/arquitectura-software/>).
- García León, Delba, y Alfa Beltrán Benavides. 1995. «Un enfoque actual sobre la calidad del software». *ACIMED* 3(3):40-42.
- Gesuitti, Mauro David. 2017. «¿Qué es la usabilidad en Software?» *NeoComplexx*. Recuperado 7 de julio de 2021 (<https://neocomplexx.com/es/2017/06/27/que-es-la-usabilidad-en-software/>).
- GitHub. 2021. «GitHub: Where the World Builds Software». *GitHub*. Recuperado 25 de junio de 2021 (<https://github.com/>).
- GreenSQA. 2019. «Arquitectura de Microservicios vs Arquitectura Monolítica | GreenSQA | Pruebas de Software y Aseguramiento de la calidad». *GreenSQA*. Recuperado 16 de julio de 2021 (<https://greensqa.com/arquitectura-de-microservicios-vs-arquitectura-monolitica/>).
- Gupta, Yash Munnalal, y Somjit Homchan. 2021. «Insect Detection Using a Machine Learning Model». *Nusantara Bioscience* 13(1). doi: 10.13057/nusbiosci/n130110.
- Harditya, Arya. 2020. «Indonesian Sign Language (BISINDO) As Means to Visualize Basic Graphic Shapes Using Teachable Machine».
- Herazo, Luis. 2020. «¿Qué es una aplicación móvil? | Anincubator - Blog». *Anincubator Website*. Recuperado 17 de junio de 2021 (<https://anincubator.com/que-es-una-aplicacion-movil/>).
- Herrera, Nicolás. 2018. «¿Aplicaciones Nativas, Web o Híbridas? ¿Cuál es la diferencia?» *Blue Latam*. Recuperado 17 de junio de 2021 (<https://bluelatam.com/aplicaciones-nativas-web-o-hibridas-cual-es-la-diferencia/>).
- Hidalgo Broncano, William Oswaldo. 2013. «Diseño de una Cámara de Secado de Madera para la Fábrica de Muebles Hidalgo».
- HostGator. 2021. «Editor de código: echa un vistazo a las 10 mejores opciones». *HostGator*. Recuperado 19 de agosto de 2021 (<https://www.hostgator.mx/blog/mejores-opciones-de-editor-de-codigo/>).
- Ilimit. 2020. «Arquitecturas monolíticas o arquitectura de microservicios». *Ilimit*. Recuperado 8 de julio de 2021 (<https://www.ilimit.com/blog/arquitecturas-monoliticas-o-arquitectura-de-microservicios-ventajas-e-inconvenientes/>).
- Importancia.org. 2013. «Software». *Importancia*. Recuperado 16 de junio de 2021 (<https://www.importancia.org/software.php>).

- IMT Platform. 2017. «Kanban: por qué es ágil y en qué ventaja a SCRUM - ITM Platform». *ITM Platform | Projects, Programs & Portfolio*. Recuperado 22 de junio de 2021 (<https://www.itmplatform.com/es/blog/kanban-por-que-es-agil-y-en-que-ventaja-a-scrum/>).
- Instituto Boliviano de Comercio Exterior. 2016. «IBCE :: América Latina, un proveedor potencial de madera». *IBCE*. Recuperado 9 de junio de 2021 (<https://www.ibce.org.bo/informacion-estudios-mercado-descarga.php?id=119>).
- Jeong, Hyunja. 2020. «Feasibility Study of Google's Teachable Machine in Diagnosis of Tooth-Marked Tongue». *Journal of Dental Hygiene Science* 20(4):206-12. doi: 10.17135/jdhs.2020.20.4.206.
- Kim, Han Sung, Soojin Jun, SeongYune Choi, y Sungae Kim. 2020. «Development and Application of Education Program on Understanding Artificial Intelligence and Social Impact». *The Journal of Korean Association of Computer Education* 23(2):21-29. doi: 10.32431/kace.2020.23.2.003.
- La Mina Digital. 2019. «Qué son las Apps y sus Características». *La Mina Digital*. Recuperado 17 de junio de 2021 (<https://www.laminadigital.es/que-son-las-apps-origen/>).
- Lapedriza, Ágata. 2012. «La visión por computador: Una disciplina en auge». *Tecnología++*. Recuperado 24 de junio de 2021 (<http://informatica.blogs.uoc.edu/la-vision-por-computador-una-disciplina-en-auge/>).
- Lateef, Zulaikha. 2019. «What Are The Types Of Artificial Intelligence? | Branches of AI». *Edureka*. Recuperado 24 de junio de 2021 (<https://www.edureka.co/blog/types-of-artificial-intelligence/>).
- Lee, Glenn. 2020. «Tipos de pruebas de software: diferencias y ejemplos». *LoadView*. Recuperado 23 de junio de 2021 (<https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/>).
- LISA Institute. 2019. «Inteligencia de imágenes (IMINT): misión, función y salidas profesionales». *LISA Institute*. Recuperado 9 de agosto de 2021 (<https://www.lisainstitute.com/blogs/blog/inteligencia-imagenes-imint>).
- LISA Institute. 2020a. «HUMINT: Ejemplos, tipos y motivaciones de las Fuentes Humanas». *LISA Institute*. Recuperado 9 de agosto de 2021 (<https://www.lisainstitute.com/blogs/blog/humint-ejemplos-tipos-fuentes-humanas>).
- LISA Institute. 2020b. «OSINT (Inteligencia de Fuentes Abiertas): tipos, métodos y salidas profesionales». *LISA Institute*. Recuperado 9 de agosto de 2021 (<https://www.lisainstitute.com/blogs/blog/osint-inteligencia-fuentes-abiertas>).
- Maida, Esteban Gabriel, y Julián Pacienza. 2015. «Metodologías de desarrollo de software».
- Malav, Bhagwati. 2017. «Microservices vs Monolithic architecture | by Bhagwati Malav | Hash#Include | Medium». *Medium*. Recuperado 16 de julio de 2021

(<https://medium.com/startlovingyourself/microservices-vs-monolithic-architecture-c8df91f16bb4>).

- Mancomún. 2021. «Draw.io | Mancomún». *Mancomun*. Recuperado 28 de junio de 2021 (<https://www.mancomun.gal/es/solucion-tic/draw-io/>).
- Mannise, Raúl. 2020. «Laurel, propiedades, beneficios y contraindicaciones». *EcoCosas*. Recuperado 16 de junio de 2021 (<https://ecocosas.com/plantas-medicinales/laurel/>).
- Mata Jiménez, Jaime de la. 2011. «Influencia de la humedad de la madera en la evaluación de las propiedades mecánicas del pino silvestre mediante técnicas no destructivas». Recuperado 9 de junio de 2021 (<http://oa.upm.es/36492/>).
- MathWorks. 2021. «Deep Learning: Tres cosas que es necesario saber». Recuperado 24 de junio de 2021 (<https://es.mathworks.com/discovery/deep-learning.html>).
- Mejía, Miguel. 2021. «¡Dale vida a tu producto digital con Figma!» <https://www.crehana.com>. Recuperado 28 de junio de 2021 (<https://www.crehana.com/blog/disenio-grafico/que-es-figma/>).
- Mizgirev, Lev, Edward Galiaskarov, Irina Astrakhantseva, Sergey Bobkov, y Roman Astrakhantsev. 2021. «Transfer learning for road-based location classification of non-residential property *». 16.
- Modenese, Paolo. s. f. «Madera Aserrada: Tipos y Características». *Manual de Obra*. Recuperado 15 de junio de 2021 (<https://www.manualdeobra.com/blog/maderamaciza>).
- Mroczkowska, Agnieszka. 2021. «What Is a Mobile App? | App Development Basics for Businesses». *Droids On Roids*. Recuperado 17 de junio de 2021 (<https://www.thedroidsonroids.com/blog/what-is-a-mobile-app-app-development-basics-for-businesses>).
- Muñoz, Cristian. 2020. «Aplicación de la metodología Mobile-D en el desarrollo de una app móvil para gestionar citas médicas del centro JEL Riobamba». Universidad Nacional Del Chimborazo.
- Ong, Song-Quan, Hamdan Ahmad, Gomesh Nair, Pradeep Isawasan, y Abdul Hafiz Ab Majid. 2021. «Implementation of a deep learning model for automated classification of *Aedes aegypti* (Linnaeus) and *Aedes albopictus* (Skuse) in real time». *Scientific Reports* 11. doi: 10.1038/s41598-021-89365-3.
- Ovrmind. 2019. «Conoce las 10 principales ramas de la inteligencia artificial | Ovrmind Agencia de Marketing Digital». Recuperado 24 de junio de 2021 (<https://ovrmind.com/ramas-inteligencia-artificial/>).
- Paz Fong, Francisco José. 2008. «Determinación de la composición química de la madera obtenida del primer clareo en árboles de melina, de una plantación proveniente del departamento de Izabal». 122.
- Pérez Porto, Julián, y María Merino. 2017. «Definición de estándar — Definicion.de». *Definición.de*. Recuperado 17 de junio de 2021 (<https://definicion.de/estandar/>).

- Pmoinformatica. 2018a. «Documento de requerimientos de software». *Pmoinformatica*. Recuperado 7 de julio de 2021 (http://www.pmoinformatica.com/2018/04/documento-de-requerimientos-de-software_37.html).
- Pmoinformatica. 2018b. «Métricas del software: Uso y ventajas». *Pmoinformatica*. Recuperado 24 de junio de 2021 (<http://www.pmoinformatica.com/2018/04/metricas-del-software-ventajas.html>).
- Pressman, Roger S. 2010. *Ingeniería del Software - Un enfoque práctico*. 7.^a ed.
- PROEcuador. 2015. *Perfil sectorial de forestal para el inversionista*. Quito.
- Programo Ergo Sum. 2021. «¿Qué es Python? - Curso de iniciación a la programación con Python en Raspberry Pi». *Programo Ergo Sum*. Recuperado 9 de agosto de 2021 (<https://www.programoergosum.com/cursos-online/raspberry-pi/244-iniciacion-a-python-en-raspberry-pi/que-es-python>).
- PuulInfo. 2018. «Propiedades de humedad de la madera». *Wood Products*. Recuperado 1 de agosto de 2021 (<https://www.woodproducts.fi/es/content/propiedades-de-humedad-de-la-madera>).
- Queipo, Juan, Rodrigo Gonzalez, Mariana Llinares, Carlos Villagra, y Virginia Gallego. 2013. «Características de la madera como material de construcción». *Publicaciones Digitales Técnicas*. Recuperado 10 de junio de 2021 (<http://publiditec.com/blog/caracteristicas-de-la-madera-como-material-de-construccion/>).
- Raffino, María Estela. 2021. «Software - Concepto, tipos, ejemplos, software libre y hardware». Recuperado 16 de junio de 2021 (<https://concepto.de/software/>).
- Redacción APD. 2021. «Metodología Kanban: en qué consiste y cómo utilizarla». *APD España*. Recuperado 22 de junio de 2021 (<https://www.apd.es/metodologia-kanban/>).
- Román, Victor. 2019. «Introducción al Machine Learning: Una Guía Desde Cero». *Medium*. Recuperado 24 de junio de 2021 (<https://medium.com/datos-y-ciencia/introduccion-al-machine-learning-una-gu%C3%ADa-desde-cero-b696a2ead359>).
- Rruthers, Juanca. 2018. «Metodología Ágil: Crystal». *Ingeniería de Software I*. Recuperado 22 de junio de 2021 (<https://isi2018.wordpress.com/2018/04/09/metodologia-agil-crystal/>).
- Sánchez, Javier. 2020. «TIPOS de MADERA: Características y Clasificación con Ejemplos». *ecologiaverde.com*. Recuperado 1 de agosto de 2021 (<https://www.ecologiaverde.com/tipos-de-madera-caracteristicas-y-clasificacion-1223.html>).
- Sancho, Fernando. 2021. «Medir la eficacia de un aprendizaje». *Fernando Sancho Caparrini*. Recuperado 24 de junio de 2021 (<http://www.cs.us.es/~fsancho/?e=231>).

- Santander. 2020. «Metodologías de desarrollo de software: ¿qué son?» *Santander | Becas*. Recuperado 18 de junio de 2021 (<https://blog.becas-santander.com/es/metodologias-desarrollo-software.html>).
- Shah, Kanish. 2019. «A Quick Overview to the Transfer Learning and It's Significance in Real World Applications». *Medium*. Recuperado 24 de junio de 2021 (<https://medium.com/towards-tech-intelligence/a-quick-overview-to-the-transfer-learning-and-its-significance-in-real-world-applications-790fb57debad>).
- Sinnaps. 2020. «Metodología XP o Programación Extrema: ¿Qué es y cómo aplicarla?» *Gestor de proyectos online*. Recuperado 22 de junio de 2021 (<https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp>).
- Software DELSOL. 2019. «▷ Metodología ¿Qué es?» *Software DELSOL*. Recuperado 18 de junio de 2021 (<https://www.sdelsol.com/glosario/metodologia/>).
- Soto Morales, Víctor Manuel. 2021. «Conoce qué son las pruebas no funcionales de software». *Academia Pragma*. Recuperado 23 de junio de 2021 (<https://www.pragma.com.co/blog/conoce-que-son-las-pruebas-no-funcionales-de-software>).
- Teachable Machine. 2017. «Teachable Machine». *Teachable Machine*. Recuperado 28 de junio de 2021 (<https://teachablemachine.withgoogle.com/>).
- Tecnologías de la Información. 2018. «Sistemas Expertos: Definición, Aplicaciones y Ejemplos». *Tecnologías de la Información*. Recuperado 24 de junio de 2021 (<https://www.tecnologias-informacion.com/sistemas-expertos.html>).
- TecnoMagazine. 2018. «Tipos de Software: Clasificación, Características y Ejemplos». Recuperado 16 de junio de 2021 (<https://tecnomagazine.net/tipos-de-software/>).
- Tena, María. 2018. «Metodología "Agile". La revolución de las formas de trabajo». *BBVA NOTICIAS*. Recuperado 22 de junio de 2021 (<https://www.bbva.com/es/metodologia-agile-la-revolucion-las-formas-trabajo/>).
- TensorFlow. 2021. «TensorFlow». *TensowFlow*. Recuperado 9 de agosto de 2021 (<https://www.tensorflow.org/?hl=es-419>).
- Todo en Madera. 2016. «Enfermedades de la madera». *Carpinteros Valencia*. Recuperado 10 de junio de 2021 (<https://www.carpinterosvalencia.es/enfermedades-la-madera/>).
- Toro, Luigys. 2017. «Anaconda Distribution: La Suite más completa para la Ciencia de datos con Python». *Desde Linux*. Recuperado 28 de junio de 2021 (<https://blog.desdelinux.net/ciencia-de-datos-con-python/>).
- Tyagi, Neelam. 2021. «6 Major Branches of Artificial Intelligence (AI) | Analytics Steps». *Analytic Steps*. Recuperado 24 de junio de 2021 (<https://www.analyticssteps.com/blogs/6-major-branches-artificial-intelligence-ai>).

- Uriarte, Julia Máxima. 2020. «Madera: composición, estructura, usos y características». *Caracteristicas.co*. Recuperado 10 de junio de 2021 (<https://www.caracteristicas.co/madera/>).
- Vargas, Connie. 2019. «Tipos de pruebas funcionales de software testing • Trycore». *Trycore*. Recuperado 23 de junio de 2021 (<https://trycore.co/transformacion-digital/tipos-de-pruebas-funcionales/>).
- Velásquez, Sebastián. 2020. «Lo que debes saber para crear un dataset | Avances AI». Recuperado 9 de agosto de 2021 (<https://www.avances.ai/proceso-creacion-datasets/>).
- Velneo. 2019. «Top 7 herramientas para desarrolladores de software en 2019». *Velneo*. Recuperado 24 de junio de 2021 (<https://velneo.es/herramientas-software-2019/>).
- We are social. 2021. «Digital 2021». *We Are Social*. Recuperado 17 de junio de 2021 (<https://wearesocial.com/digital-2021>).
- West, Darrell M. 2018. «What Is Artificial Intelligence?» *Brookings*. Recuperado 23 de junio de 2021 (<https://www.brookings.edu/research/what-is-artificial-intelligence/>).
- Westreicher, Guillermo. 2021. «Encuesta - Qué es, definición y concepto | 2021 | Economipedia». *Economipedia*. Recuperado 19 de agosto de 2021 (<https://economipedia.com/definiciones/encuesta.html>).
- Zambrano García, Wilmer Alberto. 2017. «Estudio comparativo de metodologías de desarrollo ágil en base al desarrollo de una aplicación móvil, modalidad proyecto de investigación, para la obtención del Título de Ingeniero Informático.»

Anexos