



UNIVERSIDAD JOSÉ ANTONIO PÁEZ

**APLICACIÓN PREDICTIVA DE CUERPOS,
FENÓMENOS Y OBJETOS ASTRONÓMICO -
ESPACIALES MEDIANTE EL USO DE
APRENDIZAJE PROFUNDO (DNN, CNN) Y
VISIÓN ARTIFICIAL.**

Autor: Adquinson Farias

Urb. Yuma II, calle n° 3. Municipio San Diego
Teléfono: (0241) 8714240 (master) – Fax: (0241) 8712394

REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN COMPUTACIÓN

**APLICACIÓN PREDICTIVA DE CUERPOS, FENÓMENOS Y
OBJETOS ASTRONÓMICO - ESPACIALES MEDIANTE EL USO
DE APRENDIZAJE PROFUNDO (DNN, CNN) Y VISIÓN
ARTIFICIAL.**

Trabajo de Grado presentado como requisito parcial para optar al título de
INGENIERO EN COMPUTACIÓN

Autor: Adquinson Farías
C.I.: V- 26.917.102
Tutor: Dra. Belkys Araujo
C.I.: V- 6.906.234

San Diego, Octubre de 2019



FI-C -001-2019-2CE

Valencia, 19 de Julio de 2019

Ciudadanos:
Adquinson Farias
C.I: 26.917.102
Presente-

Cumplo con informarle que la Comisión de Trabajo de Grado y Pasantías de la Facultad de Ingeniería en su reunión N° 01-2019 de fecha 19-07-2019 aprobó el proyecto de trabajo de grado titulado **APLICACIÓN PREDICTIVA DE CUERPOS, FENÓMENOS Y OBJETOS ASTRONÓMICOS-ESPACIALES MEDIANTE EL USO DE APRENDIZAJE PROFUNDO (DNN,CNN) Y VISIÓN ARTIFICIAL** Presentado por usted como requisitos para optar al título de Ingeniero en Computación.

Se ratifica la designación del Ing. Belkys Araujo C.I:6.906.234 y la Ing. Alicia De Pizzela C.I: 4.598.880 como Tutores Académicos y Metodológicos que los asesoraran en el desarrollo de este proyecto.

Atentamente,



Prof. Luis Lira
Decano de la Facultad de Ingeniería

c.c. Coordinación de Pasantías y Trabajo de Grado (1).

2
1

Ll/le



REPÚBLICA BOLIVARIANA DE VENEZUELA

UNIVERSIDAD JOSÉ ANTONIO PÁEZ

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA EN COMPUTACIÓN

ACEPTACIÓN DEL TUTOR

Quien suscribe, Dra. Belkys Araujo, portador de la cédula de identidad N°: V-6.906.234, en mi carácter de tutor del trabajo de grado presentado por el ciudadano: Adquinson Farías, portador de la cédula de identidad N°: V- 26.917.102, titulado: **APLICACIÓN PREDICTIVA DE CUERPOS, FENÓMENOS Y OBJETOS ASTRONÓMICO - ESPACIALES MEDIANTE EL USO DE APRENDIZAJE PROFUNDO (DNN, CNN) Y VISIÓN ARTIFICIAL**, presentado como requisito parcial para optar por el título de: Ingeniero en Computación, considero que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del jurado examinador que se designe.

En San Diego, a los 10 días del mes de Octubre del año 2019.

Dra. Belkys Araujo

V- 6.906.234

A mi madre, Doris Mantilla, quién hubo dado todo por verme llegar hasta aquí, y a quién injustamente el tiempo le quitó la oportunidad de verlo.

Hasta que nos veamos nuevamente.

AGRADECIMIENTOS

A mi mamá. Aunque ya no esté físicamente, siempre fue el apoyo e impulso primordial para cumplir con este objetivo. Agradezco enormemente por todo lo que hiciste por mí, y por todos los valores y principios que desde joven me inculcaste. Espero mejorar cada día más, para que aún si no estás, puedas sentirte cada día más, un poco más orgullosa de mí.

A mi papá y hermana. Aquiles y Darlenys Farias. Gracias por haber estado siempre al pendiente, y por haberme prestado el apoyo en cuanto más lo necesité. Sin ustedes no habría podido afrontar el mayor y peor desafío que me ha tocado en la vida.

A mi prima, Norma Mendez. Muchas gracias por las tardes de café que tanto me ayudaron cuando más cansado estaba, y gracias aún más por haberme brindado un gran apoyo, y estar pendiente de mi progreso desde el inicio, hasta el fin.

A mi tutor académico, la Dra. Belkys Araujo. Muchísimas gracias por la guía, los consejos, la colaboración y empatía. Gracias, además, por ser una excelente profesional y recordarnos a mis compañeros y a mí, los valores y principios que debe tener un profesional de la ingeniería.

A mis amigos de la comarca. Víctor, Valeria, Kasen, Miguel Z., Edgar, Anthony, Simón, Chen, Luis, Zuljanny y Robert. Gracias por las risas, las peleas, las ayudas y las invitaciones. Este recorrido definitivamente, no habría sido el mismo sin ustedes. Gracias además, por haber estado en el momento más difícil de mi vida.

A mis compañeros de la promoción XLI. Gracias por haber sido excelentes y dedicados estudiantes. Estoy seguro que el día de mañana serán aún mejores profesionales, y que cada uno de nosotros dirá: “Yo estudié con él”, con el mayor orgullo posible.

A mis profesores. Belkys, Oneida, Jetro, José Luis, Mayerlin, Franibe, Ana, Labarca, Arnulfo, Francisco, George, Wilfredo, Bassam y a los demás, a quienes ya no puedo recordar. Muchas gracias por haber compartido sus conocimientos y enseñanzas aun cuando la situación del país no era la más óptima. Hicieron lo posible y lo que estaba en sus manos por seguir educando y creando profesionales en medio de la crisis.

ÍNDICE GENERAL

| | |
|---|-----|
| CONTENIDO | pp. |
| LISTA DE FIGURAS | xii |
| LISTA DE CUADROS | xiv |
| RESUMEN INFORMATIVO | xv |
| INTRODUCCIÓN | 1 |
| | |
| CAPÍTULO | |
| I. EL PROBLEMA | |
| 1.1. Planteamiento del problema..... | 2 |
| 1.2. Formulación del problema..... | 4 |
| 1.3. Objetivos de la investigación..... | 4 |
| 1.3.1. Objetivo general..... | 4 |
| 1.3.2. Objetivos específicos..... | 4 |
| 1.4. Justificación de la investigación..... | 5 |
| 1.5. Alcance de la investigación..... | 6 |
| 1.6. Limitaciones..... | 6 |
| | |
| II. MARCO TEÓRICO | |
| 2.1. Antecedentes..... | 7 |
| 2.2. Bases teóricas..... | 17 |
| 2.2.1. Volumen de datos..... | 17 |
| 2.2.2. Machine Learning..... | 18 |

| | |
|---|----|
| 2.2.2.1. Redes neuronales artificiales..... | 19 |
| 2.2.2.2. Deep Learning..... | 20 |
| 2.2.2.2.1. Deep Neural Networks (DNN)..... | 21 |
| 2.2.2.2.2. Convolutional Neural Networks (CNN)..... | 22 |
| 2.2.2.3. Función de Activación..... | 23 |
| 2.2.2.4. Kernel..... | 24 |
| 2.2.2.5. Pooling..... | 25 |
| 2.2.2.5.1. Average Pooling..... | 25 |
| 2.2.2.5.2. Max Pooling..... | 25 |
| 2.2.2.5.3. Min Pooling..... | 25 |
| 2.2.2.6. Stride..... | 26 |
| 2.2.2.7. Feature Maps..... | 27 |
| 2.2.3. Fotometría..... | 28 |
| 2.2.4. The Sloan Digital Sky Survey (SDSS)..... | 28 |
| 2.3. Definición de términos..... | 29 |

III. MARCO METODOLÓGICO

| | |
|--------------------------------------|----|
| 3.1. Diseño de la investigación..... | 30 |
| 3.2. Nivel de la investigación..... | 31 |
| 3.3. Tipo de la investigación..... | 31 |
| 3.4. Población y Muestra..... | 32 |
| 3.4.1. Población..... | 32 |

| | |
|---|----|
| 3.4.2. Muestra..... | 32 |
| 3.5. Técnicas e Instrumentos de Recolección de datos..... | 33 |
| 3.5.1. Técnicas..... | 33 |
| 3.5.2. Instrumentos..... | 34 |
| 3.6. Validez del instrumento..... | 34 |
| 3.7. Confiabilidad del instrumento..... | 35 |
| 3.8. Técnicas de Análisis..... | 35 |
| 3.9. Fases de la investigación..... | 35 |

IV. RESULTADOS

| | |
|---|----|
| 4.1. Fase I: Determinar los requerimientos funcionales y no funcionales mediante las técnicas de recolección de datos..... | 39 |
| 4.1.1. Requerimientos funcionales..... | 40 |
| 4.1.1.1. Conjuntos de datos de imágenes..... | 40 |
| 4.1.1.2. Red neuronal convolucional..... | 41 |
| 4.1.1.3. Visualización..... | 41 |
| 4.1.1.4. Clasificación y Predicción..... | 41 |
| 4.1.2. Requerimientos no funcionales..... | 42 |
| 4.1.2.1. Seguridad..... | 42 |
| 4.1.2.2. Rendimiento..... | 42 |
| 4.1.2.3. Portabilidad..... | 42 |
| 4.1.2.4. Usabilidad..... | 43 |

| | |
|---|----|
| 4.1.2.5. Integridad de los datos..... | 43 |
| 4.1.2.6. Confiabilidad..... | 43 |
| 4.2. Fase II: Diseñar los componentes globales y modelos de datos datos que constituyen la aplicación, basados en la experiencia de usuario y los principios de consistencia y durabilidad..... | 44 |
| 4.2.1. Estudio de factibilidad..... | 44 |
| 4.2.2. Estudio de negocio..... | 46 |
| 4.2.2.1. Definición de la arquitectura del sistema..... | 46 |
| 4.2.2.2. Plan de desarrollo..... | 47 |
| 4.2.2.3. Requerimientos prioritarios..... | 48 |
| 4.2.3. Iteración del modelo funcional..... | 49 |
| 4.2.4. Diseño y Construcción de la Iteración..... | 50 |
| 4.2.4.1. Diseño de la Iteración..... | 50 |
| 4.2.4.1.1. Actores del sistema..... | 50 |
| 4.2.4.1.2. Casos de Uso..... | 53 |
| 4.2.4.1.3. Diagramas de Clases..... | 65 |
| 4.2.4.1.4. Diseño de la Interfaz..... | 69 |
| 4.2.4.1.5. Diseño de la Red Neuronal Convolutacional.. | 72 |
| 4.2.4.1.6. Esquema de la base de datos..... | 73 |
| 4.3. Fase III: Construir la aplicación en función de los requerimientos recabados y ejecutar pruebas de acuerdo a la metodología de trabajo..... | 74 |

| | |
|--|-----------|
| 4.3.1. Recursos..... | 74 |
| 4.3.1.1. Humanos..... | 74 |
| 4.3.1.2. Institucionales..... | 74 |
| 4.3.1.3. Tecnológicos..... | 75 |
| 4.3.2. Construcción de la Iteración..... | 75 |
| 4.3.2.1. Lenguaje de programación..... | 76 |
| 4.3.2.2. Paradigma de Programación..... | 76 |
| 4.3.2.3. Interfaz de Usuario..... | 76 |
| 4.3.2.4. Red Neuronal Convolutiva..... | 77 |
| 4.3.2.5. Dataset..... | 78 |
| | |
| V. CONCLUSIONES Y RECOMENDACIONES | |
| 5.1. Conclusiones..... | 79 |
| 5.2. Recomendaciones..... | 82 |
| REFERENCIAS..... | 83 |
| ANEXOS | |
| A. Encuesta..... | 85 |
| B. Procesos realizados por la aplicación..... | 96 |

LISTA DE FIGURAS

| | |
|--|----|
| 2.1: Representación de una red neuronal convolucional..... | 22 |
| 2.2: Funciones de activación usadas comúnmente en las RNA's..... | 23 |
| 2.3: Kernel y el proceso de convolución..... | 24 |
| 2.4: Aplicación del Max pooling y Average pooling..... | 26 |
| 2.5: Mapas de características generados por una convolución..... | 26 |
| 4.1: Arquitectura de la aplicación..... | 46 |
| 4.2: Plan de Desarrollo..... | 47 |
| 4.3: Caso de uso: Crear conjunto de datos..... | 53 |
| 4.4: Caso de uso: Crear Modelos..... | 55 |
| 4.5: Caso de uso: Entrenar Modelo..... | 57 |
| 4.6: Caso de uso: Visualizar mapas de características..... | 59 |
| 4.7: Caso de uso: Visualizar proceso de entrenamiento..... | 61 |
| 4.8: Caso de uso: Detección y Clasificación..... | 63 |
| 4.9: Diagrama de clases: MainApp..... | 65 |
| 4.10: Diagrama de clases: ScreenLayout..... | 66 |
| 4.11: Diagrama de clases: Sender..... | 66 |
| 4.12: Diagrama de clases: OptionsLayout..... | 67 |
| 4.13: Diagrama de clases: Propagador..... | 67 |

| | |
|--|----|
| 4.14: Diagrama de clases: Opciones..... | 68 |
| 4.15: Diagrama de clases: Dataset..... | 68 |
| 4.16: Diagrama de clases: Convnet..... | 69 |
| 4.17: Wireframe de la ventana principal..... | 69 |
| 4.18: Wireframe del sub menú de Entrenamiento..... | 70 |
| 4.19: Wireframe del sub menú Conjunto de datos..... | 70 |
| 4.20: Wireframe del sub menú Modelos..... | 71 |
| 4.21: Wireframe del sub menú Visualización..... | 71 |
| 4.22: Wireframe del sub menú de Clasificación..... | 72 |
| 4.23: Diseño de la red neuronal convolucional estilo LeNet..... | 72 |
| 4.24: Esquema de la base de datos para archivos de evento y log..... | 73 |
| 4.25: Inicializador de la aplicación..... | 76 |
| 4.26: Creación básica de un componente global..... | 77 |
| 4.27: Clase Convnet..... | 78 |
| 4.28: Clase Dataset..... | 78 |

LISTA DE CUADROS

| | |
|---|----|
| 1: Relación entre los ítems y los requerimientos funcionales y no funcionales.. | 43 |
| 2: Estudio de factibilidad de acuerdo con la metodología DSDM..... | 45 |
| 3: Requerimientos prioritarios encontrados en el estudio de negocio..... | 48 |
| 4: Descripción de los actores: Investigador / Usuario..... | 50 |
| 5: Descripción de los actores: Modificador..... | 51 |
| 6: Descripción de los actores: Accionador..... | 51 |
| 7: Descripción de los actores: Visualizador..... | 52 |
| 8: Descripción de los actores: Red Neuronal (nn)..... | 52 |
| 9: Definición de caso de uso: Crear conjunto de datos..... | 53 |
| 10: Definición de caso de uso: Crear modelos..... | 55 |
| 11: Definición de caso de uso: Entrenar modelo..... | 57 |
| 12: Definición de caso de uso: Visualizar mapas de características..... | 59 |
| 13: Definición de caso de uso: Visualizar proceso de entrenamiento..... | 61 |
| 14: Definición de caso de uso: Detección y Clasificación..... | 63 |



REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN COMPUTACIÓN

APLICACIÓN PREDICTIVA DE CUERPOS, FENÓMENOS Y OBJETOS ASTRONÓMICO - ESPACIALES MEDIANTE EL USO DE APRENDIZAJE PROFUNDO (DNN, CNN) Y VISIÓN ARTIFICIAL.

Autor: Adquinson Farías

Tutor: Dra. Belkys Araujo

Fecha: Agosto de 2019

RESUMEN

La astronomía es un área de la investigación que produce grandes volúmenes de datos, un ejemplo de ello es el LSST, que se espera en 2022, pueda llegar a producir hasta 20 TB por noche. Usualmente, para obtener estos datos se usan métodos como la espectroscopia o aquellos basados en fotometría, este último, representa un interés particular debido a que en la última década, la cantidad y calidad de las imágenes obtenidas del espacio se ha incrementado enormemente. Un ejemplo de ello es la galaxia M33, cuya imagen luego del coloreado RGB ocupa 1.67 GB de almacenamiento. Esto, evidentemente, representa un desafío para los investigadores, pues, no sólo deben lidiar con las limitaciones del almacenamiento, sino que también con el proceso de obtención, procesamiento, clasificación y análisis del cuerpo, fenómeno u objeto astronómico que se encuentre en la imagen. Así pues, esta herramienta se presenta una alternativa para procesar y clasificar los fenómenos, cuerpos y objetos astronómicos mediante el uso de aprendizaje profundo y visión artificial, ya que, al eliminar dos pasos intermedios, el investigador podrá centrarse en las tareas de análisis y estudio

Descriptor: Inteligencia Artificial: *Deep Learning*. Datos: Procesamiento.
Astronomía: Esquema de Clasificación Morfológica, SDSS.

INTRODUCCIÓN

El siglo XXI se ha caracterizado por demostrar que no importa el área en la cual se desempeñe una persona o una máquina, siempre es posible obtener datos sin importar el trasfondo. Desde luego, cuando se habla de máquina se asume que son aquellas que operan en una línea de producción, o aquellas que actúan como una interfaz entre lo que se conoce como el control y los procesos físicos. Sin embargo, hoy día las computadoras y los dispositivos móviles son aquellos que producen más volumen de datos por segundo.

Imaginen ahora pues, que la computadora más allá de sólo procesar y generar datos, es también capaz de interpretar la información que resulta de procesar esos datos. Esto es básicamente lo que se conoce como inteligencia artificial, la cual se ha popularizado enormemente en la última década gracias a una rama o sub área de la misma, conocida como Machine Learning o aprendizaje de máquina que, según Samuel (1959), no es más que la capacidad de una máquina para hacer una tarea específica aún sin haber sido programada explícitamente para la misma.

En el mismo orden de ideas, esta capacidad de resolver problemas de cualquier tipo es posible gracias a la implementación de algoritmos que permiten emular la toma de decisiones humana. En este contexto, las redes neuronales artificiales y, una implementación basada en la misma, llamada Deep Learning o aprendizaje profundo forman parte del objeto de estudio pues, según Beale (2019) ha sido una gran y potente herramienta en las áreas del reconocimiento de voz, el procesamiento del lenguaje natural y, especialmente, en el reconocimiento de imágenes.

De lo anterior, el último punto representa el foco de interés pues, la base de este proyecto de investigación, se fundamenta en el procesamiento, detección y clasificación (muchas veces asociado a la predicción) de amplios conjuntos de imágenes de carácter astronómico, especialmente aquellas que se considere pertenecen a los tipos de cuerpos, fenómenos u objetos astronómicos, optimizando así el posterior proceso de análisis y estudio.

CAPÍTULO I

EL PROBLEMA

1.1. Planteamiento del problema

El estudio astronómico es un área de la investigación dónde se utilizan sondas espaciales y potentes telescopios reflectores, por nombrar algunos, para obtener datos de todo aquello que se encuentra en nuestro sistema solar, e incluso, lo que está a millones de años luz de él. Por lo tanto, es casi inevitable pensar en el enorme volumen de datos que pueden llegar a producir estas herramientas diariamente y la consecuente tarea de clasificar y procesar un cúmulo de datos proveniente de distintas partes del universo. Un ejemplo de ello es la Galaxia del Triángulo o Messier 33, cuya imagen luego del coloreado RGB basado en gases ocupa 1.67 GB en almacenamiento de datos.

Esto facilita pues, comprender que el estudio astronómico es un área de la investigación muy demandante, y, por tanto, requiere que las agencias espaciales inviertan y utilicen muy eficientemente sus recursos monetarios, tecnológicos y humanos en pro de la automatización, gestión y el procesamiento de los datos que obtienen diariamente de los telescopios, sondas y estaciones espaciales. Sin embargo, existen muchas organizaciones e instituciones que no poseen los recursos humanos y tecnológicos necesarios para llevar a cabo dichas tareas, es decir, el procesamiento y posterior estudio de estos datos sería arduo, costoso y probablemente imposible.

Tal es el caso de Venezuela, cuyas organizaciones e instituciones científicas llevan varias décadas de atraso tecnológico y teórico respecto de cualquier otro cuerpo de investigación espacial del mundo, es decir, no poseen ni los recursos ni las herramientas para llevar a cabo su trabajo. Esto implica además, que quienes se encuentren trabajando en el área no han tenido acceso a nuevas herramientas tecnológicas de procesamiento de datos, tales como:

Aprendizaje profundo o detección mediante visión artificial, de modo que siguen trabajando de forma ineficiente y obsoleta, arrojando beneficios académicos y sociales casi inexistentes.

En relación con lo anterior, estos algoritmos de aprendizaje automático pertenecen a una clasificación particular denominada como aprendizaje profundo, lo que sugiere que pueden implementarse como aprendizaje supervisado o no supervisado en sus capas de proceso. Además, son capaces de obtener y reconocer conjuntos de características y patrones representativos en base a los criterios asignados por los investigadores. Esto, representa un ahorro evidente de tiempo y esfuerzo necesario en la clasificación y gestión de volúmenes de datos gráficos, de modo que el enfoque recaiga sobre el análisis y estudio de los resultados obtenidos.

Vale destacar, además, la importancia y la necesidad de poseer información relacionada con estos objetos astronómicos, pues, a pesar de que muchas veces suele ser desconocido por la persona promedio, estos objetos tienen eventos y fenómenos asociados a su existencia o ubicación, lo que les confiere la capacidad de afectarnos en mayor o menor medida. Un ejemplo de ello son las fulguraciones solares, que liberan una intensa radiación electromagnética en todo el rango del espectro electromagnético, o el campo gravitatorio de la luna, que atrae el agua de los océanos y afecta las mareas en base a la posición y distancia relativa de la luna.

De cualquier forma, la ausencia de un marco de trabajo común que provea conjuntos de herramientas dedicados al procesamiento, clasificación y gestión de datos gráficos de forma automática, eficiente e inteligente puede ocasionar el retraso de muchos procesos de análisis y estudio dentro de las organizaciones e instituciones, lo que impide la obtención de resultados y el trabajo basado en la reutilización y mejora de los mismos.

En definitiva, la existencia de una aplicación de escritorio que sea fácil de distribuir, compatible con los sistemas operativos más utilizados y que cumpla

con las premisas de: Gestionar y procesar amplios volúmenes de datos gráficos, la detección inteligente mediante visión artificial y la parametrización basada en características relevantes, permitiría a los investigadores de las organizaciones e instituciones el mitigar un conjunto de factores que dificultan la obtención de resultados académicos y sociales basados en el análisis y estudio de los cuerpos, fenómenos y objetos astronómico – espaciales.

1.2. Formulación del problema

Tomando en cuenta la problemática presentada. El agrupar las herramientas y tecnologías más relevantes del procesamiento inteligente y automatizado de datos en una aplicación, puede dar lugar a ciertas interrogantes, especialmente en el cómo se gestionarán los datos desde la misma a medida que, y de forma casi simultánea, la aplicación produce un conjunto de datos lo suficientemente relevante para ser utilizado en el entrenamiento y aprendizaje de un módulo de detección y visión artificial. Entonces,

- a) ¿En qué proporción mejora el uso de visión artificial y métodos de aprendizaje profundo la detección de características y posterior clasificación de cuerpos, fenómenos y objetos de carácter astronómico?

1.3. Objetivos de la investigación

1.3.1. Objetivo General

Desarrollar una aplicación predictiva de cuerpos, fenómenos y objetos Astronómico - Espaciales mediante el uso de Aprendizaje Profundo (DNN, CNN) y Visión Artificial.

1.3.2. Objetivos Específicos

- Determinar los requerimientos funcionales y no funcionales mediante las técnicas de recolección de datos.

- Diseñar los componentes globales y modelos de datos que constituyen la aplicación mediante la experiencia de usuario y los principios de consistencia y durabilidad.
- Construir la aplicación en función de los requerimientos recabados y ejecutar pruebas de acuerdo a la metodología de trabajo.

1.4. Justificación de la investigación

Este trabajo de investigación tiene sus raíces en un área de la investigación científica que, además de ser muy interesante, es muy importante para la sociedad en general, pues, nos aporta una mejor idea de lo que representa todo aquello dentro de nuestro sistema solar y por supuesto, lo que está fuera de él, es decir, las cosas que consideramos que están al infinito y más allá.

Por tanto, el propósito fundamental es el de proveer a todos aquellos que se encargan de ver y estudiar estos cuerpos, fenómenos y objetos un conjunto de herramientas lo suficientemente versátil y completo, es decir, que permita gestionar amplios volúmenes de datos gráficos, la detección de características relevantes en una investigación por medio de visión artificial, o simplemente servir como un punto de partida para la enseñanza de los conceptos básicos, intermedios o claves del área del aprendizaje, detección y el reconocimiento de patrones mediante el uso de ciertos algoritmos de aprendizaje de máquina.

Así pues, beneficiará a todos aquellos investigadores, universitarios, colegas, aficionados, organizaciones, instituciones y entes estatales que deseen hacer uso de estas herramientas con el objeto de procesar, clasificar, analizar, estudiar y visualizar, o de forma resumida, predecir características importantes sobre un conjunto de datos gráfico que contenga cuerpos, fenómenos y objetos del estudio astronómico – espacial. Además, representa una pequeña contribución a la comunidad de código abierto u open source.

1.5. Alcance de la Investigación

Se estima que el alcance del proyecto sean aquellas organizaciones, institutos y particulares que necesiten o deseen procesar, analizar y visualizar amplios volúmenes de imágenes y datos relevantes al estudio de los cuerpos, fenómenos y objetos astronómico - espaciales, por tanto, el alcance general de la investigación puede definirse como global, mientras que el alcance la aplicación puede catalogarse como open source.

1.6. Limitaciones

Una de las principales limitaciones para el desarrollo de este proyecto es la falta de una tarjeta gráfica con soporte CUDA ≥ 3.5 . Esto se debe a que los algoritmos de aprendizaje profundo enfocados al procesamiento de imagen son mucho más eficientes si se ejecutan en un módulo dedicado exclusivamente a los gráficos, como lo es la GPU. Esto significa pues, que todos los procesos envueltos en el procesamiento de gráficos e imágenes verán afectado su rendimiento general.

Otra posible limitante es la velocidad de conexión a internet. Tal y como se expresa inicialmente, el área de las observaciones espaciales consume grandes cantidades de almacenamiento, por lo tanto, para poder obtener los datos correspondientes es necesaria una buena conexión a internet. De no ser posible, sólo se trabajará con una cantidad de datos que sea lo suficientemente significativa.

Finalmente, el tiempo. Este factor es el más importante, pues, la obtención de los datos del conjunto de entrenamiento y el entrenamiento del modelo en sí, son tareas que requieren lapsos de tiempo lo suficientemente amplios.

CAPÍTULO II

MARCO TEÓRICO REFERENCIAL

2.1. Antecedentes

El aprendizaje profundo es desde hace algunos años un área en expansión y, desde entonces, ha sido una gran y potente herramienta en las áreas del reconocimiento de voz, el procesamiento del lenguaje natural y, especialmente, en el reconocimiento de imágenes, lo cual ha facilitado enormemente la tarea de resolver problemas relacionados al área del reconocimiento facial, detección de movimiento, clasificación de objetos e incluso, en muchas tecnologías avanzadas de asistencia al conductor, que abarcan casos como el manejo autónomo, detección de carriles, detección de peatones y estacionamiento autónomo (Beale, 2019).

Brice, M.J. (2019), realizó una investigación titulada “**Classification Of Stars From Redshifted Stellar Spectra Utilizing Machine Learning**”, presentada ante la Universidad Central de Washington (Central Washington University), optando por el título de Máster de Ciencias (MSc). En ella, describe la existencia de múltiples incursiones en la tarea de automatizar la clasificación del espectro estelar, sin embargo, pocos incluyen a the Sloan Digital Sky Survey (SDSS). Así pues, el espectro estelar del SDSS es sometido a métodos de clasificación estándar tales como, los K Vecinos Cercanos, Bosque Aleatorio o la Máquina de Vector Soporte.

Luego, resalta que los datos provenientes del espectro estelar son datos de altas dimensiones y que las mismas, suelen ser reducidas mediante métodos estándar de selección de características, como el Chi – Cuadrado y el Puntaje de Fisher. Estos métodos son utilizados para clasificar el espectro estelar en dos categorías estándar del esquema de clasificación, la clasificación espectral de Harvard (the Harvard Spectral Classification) y las clases luminosas de Morgan Keenan (Morgan Keenan Luminosity Classes).

Así pues, si una estrella clasifica en ambos esquemas, muchas de sus propiedades pueden ser aproximadas con facilidad, al contrario de las aproximaciones directas, que pueden tomar incluso meses de observaciones.

Cabe destacar además que, luego de estudiar y analizar, concluye en una nueva aproximación para extraer el desplazamiento al rojo (redshift) mediante el uso de un modelo de aprendizaje de máquina. Es decir que, para extraer el desplazamiento al rojo del espectro estelar, se deben identificar las líneas de absorción en las longitudes de ondas que componen el desplazamiento al rojo, además, deben ser comparadas con el resto de las longitudes de onda equivalentes y, finalmente comparadas entre ellas mismas. Entonces, los números del desplazamiento al rojo de más frecuencia se extraen y promedian para producir el desplazamiento al rojo final.

Así pues, durante este proceso, estas líneas de absorción son utilizadas para identificar si pertenecen a la clase Harvard o Morgan Keenan. Sin embargo, existe otra aproximación, que compara estos espectros con plantillas, derivando como resultado clases del tipo Harvard o MK. Así pues, es redundante extraer el desplazamiento al rojo luego de la clasificación automática del espectro estelar mediante las aproximaciones usuales, o el clasificar automáticamente espectros estelares que ya hayan sido corregidos por desplazamiento al rojo, puesto que, en ambos casos, la clasificación se efectúa dos veces.

Finalmente, expone que se pueden obtener clasificaciones estelares precisas mediante la combinación de pre procesamiento de datos de dominio específico, selección de características y técnicas de clasificación. Además, comparado al trabajo previo de otros autores, señala dos conclusiones interesantes. La primera es que no se necesita el espectro completo para obtener un alto nivel de precisión, y segundo que, aparte del ajuste de la longitud de onda y el escalado de flujo, cualquier pre procesamiento adicional del espectro,

luego de los procesos presentados por Dawson *et al.*, y Stoughton *et al.*, es innecesario.

Así mismo, Van Amerongen, P. (2018), realizó una investigación titulada “**Discovering exoplanets using Convolutional Neural Networks**”, presentada ante la Universidad de Groninga (University of Groningen), optando por el título de Bachiller universitario en Ciencias (BSc). Inicialmente, describe la existencia de múltiples métodos para detectar exoplanetas y, que, de ellos, el método del tránsito es el más prominente (Deeg, 1998). Este método mide el brillo de una estrella a intervalos regulares, luego, el resultado se grafica como el brillo relativo emitido por una estrella en función del tiempo. De ahí que esta gráfica se conozca como curva de luz.

Sin embargo, este método posee dos desventajas principales. La primera indica que el planeta debe tener una órbita alineada de tal forma que, el mismo, pueda ser posicionado entre la estrella anfitrión y la tierra. La segunda, refiere que debido a la pequeña caída en la curva de luz es fácil obtener falsos positivos. Una caída no necesariamente indica que un exoplaneta esté transitando. De acuerdo con Santerne *et al.*, (2012) el porcentaje de falsos positivos de los exoplanetas candidatos de la misión Kepler es de 35%, por lo cual es necesario una inspección mayor de los exoplanetas candidatos, por ejemplo, con el método de velocidad radial.

Se señala además que, el uso de redes neuronales artificiales en la detección de exoplanetas sería muy útil. Esto se debe a que las curvas de luz utilizadas para representar a una estrella que alberga a un exoplaneta, pueden ser utilizadas como datos de entrenamiento. Luego, se indica que hay muchas formas de procesar curvas de luz usando redes neuronales, por ejemplo, el agrupar los puntos de los datos como un número predefinido de puntos y luego alimentar a la red con esos datos. Estudios sobre estas técnicas han sido realizados por Shallue and Vanderburg (2018), produciendo valiosos resultados.

Así mismo, expone el uso por su parte de una aproximación diferente, esto es, mediante el uso de redes neuronales convolucionales para tratar de localizar potenciales exoplanetas. Es decir, la curva de luz puede ser transformada en una imagen de tamaño predeterminado, la cual será procesada por la red neuronal convolucional, resultado en la posible detección de un exoplaneta. La ventaja de esta técnica es que a la red neuronal se le pueden entregar todos los datos, no es necesario el agrupar. Por otra parte, es necesario generar imágenes de las curvas de luz que sean iguales en tamaño, pues, el tamaño de la entrada de la red convolucional debe ser fijo.

Desde luego, se menciona que no se produjo una solución perfecta para detectar exoplanetas pero que, sirve de punto de partida para próximas investigaciones, incluyendo a las de código abierto. Además, se detectaron dos objetos aún no clasificados durante el análisis de 3000 estrellas, o lo que representa el 1% del total de estrellas analizadas por la misión Kepler, están son: KIC ID 2163434 y KIC ID 28549, así pues, luego de un análisis más cercano se determinó que ambas estrellas mostraban caídas en el brillo, lo que podría implicar presencia de exoplanetas, por lo cual, una mayor investigación podría ser de utilidad para confirmar los resultados.

Además, indica que al parecer la red neuronal es capaz de detectar binarias eclipsantes con facilidad, y que podría ser de utilidad en la búsqueda de ese tipo de objetos si se usa en la misión TESS. Así mismo, se expone que la red neuronal es ajustable y reusable, y que, por tanto, tiene el potencial de ser utilizada para buscar exoplanetas, siempre y cuando se le hagan algunas modificaciones.

Finalmente, concluye que es posible detectar de forma precisa características específicas en imágenes de curvas de luz usando redes neuronales convolucionales, y por tal motivo, la pregunta principal de la investigación permanece sin respuesta. Sin embargo, confiesa que si uno es capaz de producir u obtener más características relacionadas a la presencia de exoplanetas

mediante el uso o aplicación de cierto algoritmo (automático) de reducción de datos en imágenes, es posible definitivamente el detectar exoplanetas de forma precisa, a partir de las curvas de luz y usando redes neuronales convolucionales.

En este mismo orden de ideas, Hocking, A. (2018), realizó una investigación titulada “**Automatic object detection and categorisation in deep astronomical imaging surveys using unsupervised machine learning**”, presentada ante la Universidad de Hertfordshire (University of Hertfordshire), optando por el título de Doctor en Filosofía (DPhil). En la misma, introduce y resume cómo se produce el análisis de imagen en la Astronomía, y describe, además, el origen del sistema de clasificación y su evolución, hasta llegar al modelo actual. Luego, propone y selecciona un modelo de aprendizaje no supervisado y lo implementa, dirigido al ámbito astronómico.

En principio, indica que, en los años 1920, usando el telescopio más grande de su tiempo, el reflector Hooker de 100 pulgadas, Edwin Hubble clasificó las galaxias en dos tipos morfológicos principales, esto es, espirales y elípticas (Hubble, 1926). Él, además, identificó un tercer tipo llamado galaxia irregular y desarrolló un esquema para producir lo que se conoce como el Hubble Tuning Fork o la Secuencia de Hubble (Hubble, 1936). El término “Tuning Fork” es una referencia a la bifurcación de las galaxias espirales en dos grupos, es decir, con o sin barra central. Esta secuencia da la impresión de evolución, a pesar de que no existe evidencia para sustentar esto.

Mientras tanto, los astrónomos sólo eran capaces de clasificar las galaxias en el universo local usando este sistema. Sandage (2005), describió el desarrollo de este esquema de clasificación. Así pues, indica que el esquema morfológico original de Hubble fue desarrollado a partir de placas fotográficas tomadas en Mount Wilson Observatory (Hubble, 1926, 1936). Luego, De Vaucouleurs (1959, 1964) extiende el sistema original de clasificación Hubble para incluir más sub tipos, incluyendo una transición fluida entre los tipos Sc y SBc a

Irregulares. Él introdujo además la nomenclatura para describir a los tipos de galaxias.

Así pues, las espirales sin barras (SA) y los tipos intermedios con barras débiles (SAB), galaxias con anillos (r), sin anillos (s) e intermedias (rs), además de una mayor clasificación referente a los brazos espirales. Por ejemplo, difusa, con brazos rotos y una leve protuberancia. (Sd), Altamente irregular (Im) e Irregular sin protuberancia (Sm), reemplazando la clasificación Irr de Hubble. Holmberg (1958) introdujo, además, el uso de los símbolos de más y menos para denotar divisiones aún más finas. In Van den Bergh (1960a, b), añadió el concepto de orden en los brazos espirales, sumando las etiquetas I, I-II, II, etc. para denotar el incremento de irregularidad.

Así mismo, se describe que estas nuevas clasificaciones indican el incremento luminoso y, que el sistema más moderno cubre un rango de magnitud de cuatro (entre I y V) (Sandage, 2005). También añade que, de forma más reciente, las características morfológicas más detalladas como el nivel de agrupación o clúster y las colas de marea o corriente, son listadas como sub categoría y se encuentran siendo clasificadas por Galaxy Zoo. Luego, se indica que observaciones más recientes han capturado galaxias de alto nivel de desplazamiento al rojo cosmológico. Este último, surge como una consecuencia de los fotones de la luz viajando a través un universo en expansión.

Es decir que, la longitud de onda de los fotones se expande en proporción directa a la expansión del universo por el que viajan, resultando en un desplazamiento al rojo z definido por una ecuación dónde

Por otra parte, describe la existencia de dos categorías principales para la observación astronómica por imagen. La primera, se refiere a las imágenes que abarcan una amplia región del cielo. La segunda, se basa en imágenes que capturan múltiples áreas del cielo y que contienen algún tipo de objeto conocido. Así mismo, señala que los datos utilizados para entrenar el modelo consideran el uso de ambas categorías. Señala también que, actualmente, existen muchas más técnicas desarrolladas con la intención de capturar imágenes de áreas del cielo cada vez más y más grandes.

En relación con lo anterior, estima que quizá, la gran observación más famosa en luz visible es the Sloan Digital Sky Survey, pues ha sido objeto de múltiples estudios. (York *et al.*, 2000; Stoughton *et al.*, 2002; Abazajian *et al.*, 2009; Blanton *et al.*, 2017, SDSS).

Finalmente, argumenta la realización de una técnica de aprendizaje de máquina no supervisado, lo suficientemente eficiente y que utiliza una combinación de GNG, agrupación jerárquica y de etiquetado de componentes conectados, para explorar observaciones astronómicas y automáticamente, segmentar y etiquetar las imágenes que representan los datos. Esta técnica se basa en parches, pues el modelo no procesa las imágenes completas de las galaxias, en cambio, representa las galaxias al combinar múltiples parches pequeños que se solapan entre sí. Cada uno parches solapados es, típicamente, mucho más pequeño que el total de la imagen de la galaxia.

Por ejemplo, un parche podría contener una sección de un bazo espiral, o una sección que represente una característica de superficie de bajo brillo.

Así pues, el desarrollo de esta técnica comienza por considerar qué representaciones de imágenes y algoritmos no supervisados poseen las propiedades necesarias para ser efectivos. Declara que, en el Capítulo 3 identifica tres representaciones de imagen invariantes rotacionalmente, llamadas, espectro de potencia radialmente promediado, RIFT e Intensidad

Rotacional. Es decir, pues, que la invariancia rotacional es una característica esencial que permite agrupar galaxias de un tipo similar independientemente de su orientación.

Consecuentemente, demuestra la implementación y resultados de su técnica en los campos de frontera del telescopio espacial Hubble (HST), al entrenar el algoritmo utilizando las galaxias de un campo (Abell 2744) y, posteriormente, aplicando los resultados a otro campo (MACS0416.1-2403). Además, expone la capacidad del algoritmo de separar limpiamente galaxias del tipo temprana y tardía, aún sin haber utilizado ningún tipo de entrenamiento pre dirigido sobre la base del qué es o no, una galaxia temprana o tardía. Similarmente, presenta el resultado de probar esta técnica en la generalización e, igualmente, en la identificación de una configuración óptima.

Por último, describe los resultados de aplicar la técnica a los resultados del proyecto del telescopio espacial Hubble (HST), denominado como Observación extra galáctica profunda de agrupación cósmica cercana al infrarrojo (CANDELS), y que resultó en la creación de un catálogo de 60000 galaxias etiquetadas y agrupadas en base a su similitud.

Por último, Shi, H. (2016), realizó una investigación titulada “**Galaxy classification with deep convolutional neural networks**”, presentada ante la Universidad de Illinois (University of Illinois at Urbana-Champaign), optando por el título de Máster de Ciencias (MSc) en Ingeniería Eléctrica y Computación.

Antes que nada, es importante describir que el uso de imágenes digitales provenientes de las observaciones astronómicas, representa gran interés en los investigadores de la astronomía al momento de determinar la clase morfológica de una galaxia. Esto es pues, porque los métodos convencionales se basan enormemente en algunas características morfológicas “tomadas a mano”. Mientras, los métodos populares de extracción de características, desarrollados

y usados en el procesamiento de imágenes naturales, tienden a no ser adecuados para las imágenes de galaxias.

Describe, además, que las redes neuronales convolucionales profundas (CNNs) son capaces de aprender características importantes de las imágenes mediante el uso de convolución jerárquica y operaciones de agrupación. Luego, indica que aplica las mejores tecnologías de las redes neuronales convolucionales profundas para clasificar galaxias, esto es pues, para las tareas de regresión y de multi - clasificación. Además, implementa y compara el rendimiento con varios de los algoritmos convencionales de aprendizaje de máquina, utilizados en sub tareas de clasificación.

Sus experimentos demostraron también que la capacidad de las redes neuronales convolucionales es excepcional para estas tareas, pues, son capaces de aprender características representativas automáticamente, alcanzando un alto rendimiento y sobrepasando, además, al reconocimiento humano y a otros métodos de aprendizaje de máquina.

También, ahonda en el apartado de la obtención de los datos para la clasificación de galaxias. Esto es pues, una introducción a The Sloan Digital Sky Survey (SDSS), el cual ha mapeado cerca de un cuarto del cielo entero e incluido millones de imágenes en 5 bandas diferentes del sistema fotométrico sólo en el lanzamiento número 7 de sus datos (SDSS DR7). La observación contiene una inmensa cantidad de información, al igual que datos contenidos en imágenes (más de 40 TB de galaxias y catálogos, etc.). Estos datos son muy difíciles de procesar en conjuntos de grandes volúmenes, así como también, el análisis directo para las investigaciones científicas.

Adicionalmente, describe un proyecto de ciencia público, alojado por Galaxy Zoo. Este proyecto, denominado Galaxy Zoo Project 2 clasificó alrededor de 200.000 galaxias en el modelo de color RGB. Estas galaxias, pertenecen a observaciones del SDSS DR7 y otros tipos de observaciones

similares. Este proyecto obtuvo más de 60 millones de clasificaciones humanas en un lapso de 14 meses. Luego, en diciembre de 2013 Kaggle inicia un desafío mundial para los algoritmos de clasificación automatizada. En el mismo, utiliza al menos la mitad de los datos del proyecto Galaxy Zoo Project 2.

Así bien, en este desafío, el conjunto de datos de entrenamiento consistía de 61.578 imágenes en formato JPEG, y de un tamaño de 424x424 con canales de color RGB. Cada imagen tenía un número de identificación y una galaxia (o una forma parecida) en el centro. Adicionalmente, había 37 etiquetas, resultado de la clasificación de los voluntarios. En relación, la etiqueta se genera al promediar los votos de alrededor de 44 voluntarios, quienes responden 11 preguntas relacionadas con la clasificación de la imagen y también, por técnicas de post procesado para toda la distribución de probabilidad, generando así una estructura de 37 probabilidades.

Cabe considerar, además, la introducción a los métodos de clasificación del capítulo 3. En el mismo, describe los algoritmos convencionales de aprendizaje utilizados en sus tareas de clasificación, asimismo, señala y enfoca la eficiencia algorítmica de cada uno de ellos. Tal es el caso de algoritmos como: El Vecino Cercano (NN), la Regresión Logística, el Bayesiano Ingenuo, los Árboles de Decisión y el Bosque Aleatorio, las Máquinas de Vector Soporte (SVM) y, las Redes Neuronales Convolucionales (CNN's).

Finalmente, demuestra que, aunque los métodos convencionales de clasificación de galaxias obtienen ventaja de las características morfológicas seleccionadas cuidadosamente “a mano”, es posible obtener características lo suficientemente discriminativas a partir del pre procesamiento de píxeles “crudos” para que sean usados con los mejores algoritmos de clasificación del aprendizaje de máquina.

Sin embargo, añade que la extracción de características sigue siendo un paso clave en las tareas de clasificación de imágenes de galaxias y, también,

afirma que las redes neuronales convolucionales pueden aprender automáticamente características significativas y representativas a partir de píxeles crudos, con el fin de mejorar el rendimiento de la clasificación. Luego, expone que en sus resultados experimentales un modelo optimizado de CNN con su representación jerárquica produjo resultados significativamente mejores que aquellos por métodos convencionales, dejando claro que este modelo es adecuado para el estudio de imágenes astronómicas de galaxias.

2.2. Bases teóricas

2.2.1. Volumen de Datos

De acuerdo con González, Muñoz, y Hernández (2018), los conjuntos de datos astronómicos crecen constantemente en tamaño y complejidad. Las nuevas generaciones de unidades integrales de campo (IFU) generan alrededor de 60 GB de datos por noche, mientras, los instrumentos de imagen suelen generar hasta 300 GB por noche.

Tal es el caso de: “The Large Synoptic Survey Telescope” (LSST; Ivezić *et al.*, 2008, citado en González *et al.*, 2018), el cual se está construyendo en Chile y que se espera, comience operaciones completamente en 2022. Así pues, el LSST podría generar hasta 20 TB de datos por noche y, en el transcurso, detectar más de 20 millones de galaxias.

Esto claramente, representa una alerta e indica la necesidad de un método capaz de procesar, clasificar, etiquetar e identificar los grandes volúmenes de datos producidos por las observaciones astronómicas. Jones, Schroeder y Nitschke (2019) declaran que un ojo humano entrenado es muy capaz de detectar manualmente galaxias en imágenes o en zonas de evasión (ZoA) y que, aun así, no implica que las tareas de este proceso consuman menos tiempo. Esto es pues, debido a la vasta naturaleza de los datos recopilados y, por tanto, se hace necesario el uso de un sistema automatizado y preciso para identificar galaxias, especialmente en ZoA.

2.2.2. Machine Learning

El aprendizaje de máquina, mejor conocido como “Machine Learning” es una subdivisión o subconjunto de la inteligencia artificial (AI), que se define comúnmente como la capacidad de una máquina para hacer una tarea específica aún sin haber sido programada explícitamente para la misma (Samuel, 1959). Además, se enfoca en el estudio y combinación de algoritmos y modelos estadísticos bajo una capa de abstracción que facilita el afrontar distintos tipos de problemas, en función de sus características específicas, dando como resultado una implementación basada en el tipo de algoritmo, el tipo de datos utilizado para la entrada y la salida o, simplemente, el tipo de tarea o problema que se intenta resolver.

Similarmente, según Luckert (2016), los programas de computadora tradicionales no ven el resultado de las tareas que ejecutan y por tanto, son incapaces de mejorar su comportamiento. Señala, además, que el campo del aprendizaje de máquina se aplica a este problema en específico, pues envuelve la creación de programas de computadora que son capaces de aprender y por tanto, son capaces de mejorar su rendimiento a medida que acumulan más datos y experiencia.

En relación con lo anterior, es preciso denotar que los algoritmos de aprendizaje de máquina suelen distribuirse o agruparse en categorías de acuerdo a su propósito, formando así la base en la clasificación de los algoritmos. Un ejemplo de ello son las categorías de Aprendizaje supervisado, no supervisado, semi - supervisado, Aprendizaje reforzado, Aprendizaje de características, Aprendizaje por diccionario de dispersión, por Reglas de asociación o por Detección de anomalías.

Así mismo, es importante resaltar que para poder realizar tareas de aprendizaje de máquina es necesario crear un modelo a entrenar. Esto se lleva a cabo a base de conjuntos de datos de entrenamiento, pues esto, les proporcionan a estos modelos la experiencia necesaria para procesar y analizar datos relacionados, aun cuando estos datos no hayan pertenecido al conjunto de entrenamiento originario. Así bien, al igual que con las categorías, existen muchos modelos utilizados en el aprendizaje de

máquina, tal es el caso de los Árboles de decisión, las Redes bayesianas, las Máquinas de vector soporte o las Redes neuronales artificiales.

2.2.2.1. Redes neuronales artificiales (ANNs)

El cerebro humano contiene una inmensa cantidad de neuronas y de células nerviosas, por lo cual cada una de esas células estará siempre conectada a muchas más del mismo tipo. Esto crea una muy compleja red de transmisión de señales donde cada una de estas células neuronales recopila o recibe las entradas de las otras células neuronales a la cuales esté conectada y, en el caso de que alcance cierto umbral, enviará un impulso consecuente a todas las células a la cuales esté conectada (Borglin, 2011).

Esta breve descripción del cerebro humano permite en gran medida entender el concepto de las redes neuronales artificiales, pues, indica que la información se propaga en gran medida a través de una red de elementos interconectados que reaccionan a un cierto impulso en específico, y que esto, además, determinará la consecuente propagación del impulso.

Así pues, a partir de la introducción y la analogía con el cerebro humano, Borglin (2011) describe que las redes neuronales artificiales son un clasificador modelado a partir del cómo trabaja el cerebro humano, es decir, imita el funcionamiento de una neurona mediante el uso del concepto de “perceptron”, que no es más que la unidad básica de referencia en este sistema. El perceptron, puede tomar múltiples entradas ponderadas y resumirlas, luego, determinará si la combinación de la entrada excede cierto umbral y de ser verdadero, el perceptron se activará y, consecuentemente, transmitirá una salida.

Un concepto más formal según Chen (1991), indica que una red neuronal artificial consiste de un conjunto de neuronas, denominadas como la unidad básica o elementos de procesado, una topología de conexión y un algoritmo de aprendizaje. Luego, describe que las señales de entrada se retransmiten a través de la red y que estas pueden venir del entorno externo de la red o, ser una salida de las otras unidades que conforman la red. Adicionalmente, señala que cada representación de una conexión de

entrada entre las distintas unidades o neuronas no es más que un valor ajustable llamado peso o fuerza de la conexión.

Cabe destacar que esta red neuronal artificial se divide en capas, esto es: La primera capa conocida como la capa de entrada. La capa final, también conocida como la capa de salida. Luego, las $n-1$ capas intermedias a la capa de entrada y la capa de salidas, conocidas como las capas escondidas, pues su comportamiento es restringido de forma implícita solamente por los datos de entrenamiento, debido a que los datos no muestran la salida deseada para cada una de esas capas. Así bien, el número de estas n capas define la profundidad de la red (Borovkov, 2017).

De forma análoga, este concepto puede extenderse si se considera a la capa de entrada como las múltiples neuronas que reciben los impulsos de entrada. Luego, la capa escondida, este conjunto de capas se encargan de procesar y redirigir el impulso de entrada hacia la capa de salida y, puede ser desde una hasta n cantidad de capas con m cantidad de neuronas en cada capa. Finalmente, la capa de salida, que no es más que el conjunto de neuronas que reaccionan y representan la respuesta al impulso inicial.

Finalmente, cabe destacar además el concepto de las redes neuronales profundas. Esto es pues, una derivación de las redes neuronales artificiales. Estas son pues, diferentes de los otros métodos populares del aprendizaje de máquina que permiten estructuras poco profundas (*shallow*). Típicamente, las redes neuronales profundas tienen más capas y parámetros, por lo tanto, tienen un mejor potencial para representar entradas más complejas. Estas redes complejas, permiten a las personas construir agentes inteligentes capaces de llevar a cabo complicadas tareas de percepción, e incluso, alcanzar el rendimiento o capacidad de los humanos en algunos casos (Wang, 2017).

2.2.2.2. Deep Learning

El aprendizaje profundo, mejor conocido como Deep Learning es un conjunto de algoritmos que se fundamenta en las redes neuronales artificiales. Forma parte además del sub - campo del machine learning.

Según Rötner (2017), el término es utilizado para describir a un tópico que lleva tiempo siendo tendencia en el aprendizaje de máquina. Adicionalmente, el mismo ha ganado bastante atención pública debido a su rol en la resolución de numerosas tareas. Por ejemplo, AlphaGo (Silver *et al.*, 2016, citado en Rötner) venciendo al campeón humano de GO en 2016.

Similarmente, Gal (2016) indica que la carta de introducción al aprendizaje profundo es la herramienta estadística más simple de todas, es decir, la regresión lineal. Esto es pues, porque el modelo más básico del aprendizaje profundo puede ser descrito como una jerarquía de funciones base parametrizadas (jerarquía que se conoce como red neuronal y, dónde cada vector de características de la jerarquía se conoce como capa). Establece también que la configuración más simple de regresión se podría formar por múltiples funciones base a partir de modelos de regresión y, que, para la clasificación, se formaría además una función logística al final, que agruparía la salida del modelo lineal para obtener un vector de probabilidad.

2.2.2.2.1. Deep Neural Networks (DNN)

Las redes neuronales profundas son un concepto fundamentado en las redes neuronales artificiales, e indica la posibilidad de introducir un mayor número de capas escondidas, en pro de fortalecer el proceso de aprendizaje y la detección de características. En este sentido, la profundidad viene definida por el número de capas escondidas (intermedias) que posea el modelo, luego de la capa de entrada, y antes de la capa de salida.

Esto es pues, cada capa a un mayor nivel de profundidad conecta sus impulsos de salida como la entrada de la capa siguiente, aunque, en los últimos años se ha hecho muy popular la técnica de “propagación hacia atrás” o backpropagation. Este algoritmo, calcula el descenso del gradiente en una red neuronal, tomando los valores calculados de cada nodo (y miniaturizados) y se pasan hacia delante. Luego, la derivada parcial del error respecto de cada parámetro se calcula en una propagación hacia atrás, a través del grafo que define el modelo.

2.2.2.2. Convolutional Neural Networks (CNN)

Karpathy y Johnson (2015) argumentan que las redes neuronales convolucionales son muy similares a las redes neuronales comunes. Están hechas de neuronas que tienen pesos y ordenadas que pueden ser aprendidas, reciben entradas sobre las cuales se realiza un producto punto (matricial) y, opcionalmente, le sigue una expresión de no linealidad.

La diferencia, sin embargo, radica en que este tipo de red neuronal asume en ciertos aspectos (y de forma explícita) que los parámetros de entrada provienen de imágenes. Esto, ciertamente, facilita la extracción de propiedades muy útiles que pudieran ser aprovechadas en el desarrollo de la arquitectura y en el proceso de extracción de características.

En este orden de ideas, es importante expresar que la característica relevante de este tipo de red neuronal radica en el concepto de profundidad. Una imagen digital suele pensarse como un objeto definido en dos dimensiones, es decir, alto y ancho. Sin embargo, también está compuesta por canales de color, que suelen ser tres (uno en caso de escala de grises). Estos canales de color es lo que aprovecha la red neuronal convolucional para crear un arreglo de neuronas tridimensional, pues, los canales de color se solapan y pasan a representar una nueva dimensión, es decir, la profundidad de la región de activación.

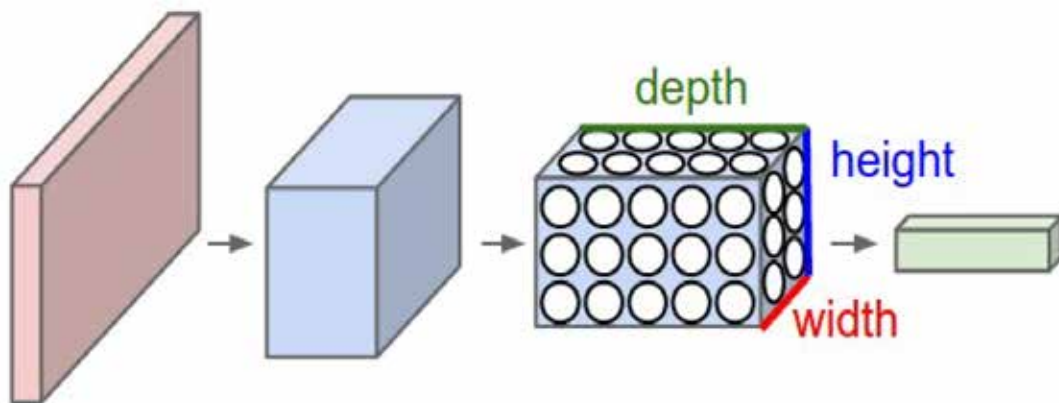


Figura 2.1: Representación de una red neuronal convolucional.

Fuente: Karpathy y Johnson (2015), [cs231n.github.io/convolutional-networks/]

2.2.2.3. Función de Activación

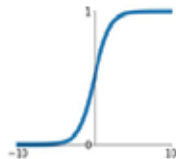
La salida de un nodo, neurona o perceptron suele estar definida según el valor de la entrada y la función de activación aplicada a la misma. Generalmente toma la suma de los pesos de todas las entradas de la capa previa y luego, genera y pasa el valor de salida (típicamente un valor no lineal) a la capa siguiente. Similarmente, puede describirse como la herramienta que le permite a la red neuronal aprender y operar sobre límites de decisión complejos.

Esta función de activación puede variar de capa en capa, y suele ser, a rasgos generales, una función no lineal. Ejemplo de funciones de activación comunes pueden ser:

- a) ReLU (Rectified Linear Unit)
- b) Sigmoide
- c) TanH
- d) Softmax
- e) ELU

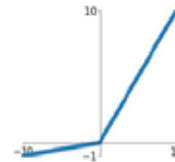
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



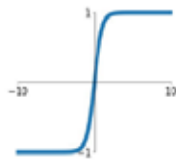
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

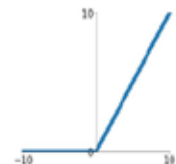


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ReLU

$$\max(0, x)$$



ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

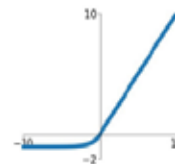


Figura 2.2: Funciones de activación usadas comúnmente en las RNA's.
Fuente: Jain, P. (2019) [towardsdatascience.com/complete-guide-of-activation-functions-34076e95d044/]

2.2.2.4. Kernel

El kernel, filtro convolucional o ventana es una matriz definida con el mismo número de dimensiones que la entrada a procesar, pero con la restricción de que su ancho y alto suelen ser de menor tamaño.

Este filtro, suele moverse de izquierda a derecha y de arriba hacia abajo a través de la imagen, según dos parámetros importantes. Uno de ellos es el tamaño del filtro, el otro se refiere al tamaño del salto. Luego de definir estos parámetros es que se produce el proceso de convolución pues, se produce una multiplicación de matrices entre los valores internos del filtro (inicialización del filtro) y el valor numérico que representa cada pixel dentro del rango del filtro.

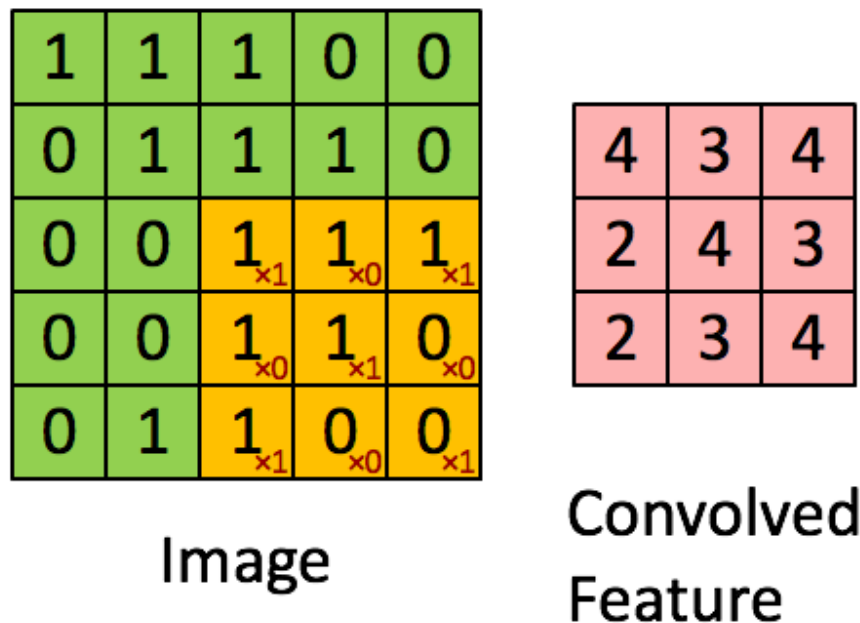


Figura 2.3: Kernel y el proceso de convolución.

El recuadro verde representa la imagen de entrada y los valores asociados. El recuadro amarillo representa el kernel que se mueve a través de la imagen. Los números rojos dentro del recuadro amarillo representan el valor de inicialización del filtro, es decir, son los que se multiplican con los valores de la imagen de entrada. Mientras, el recuadro rojo representa la característica resultante del proceso de convolución.

Fuente: Dettmers, T. (2015) [devblogs.nvidia.com/deep-learning-nutshell-core-concepts/]

2.2.2.5. Pooling

El sub muestreo, agrupación o pooling es una técnica que agrupa un cierto número de valores escalares presentes en el rango del kernel convolucional, para generar una nueva salida con espacialidad reducida y mejor concentración de características. Al hacer esto, permite disminuir o comprimir el tamaño de la entrada para la capa siguiente, y consecuentemente, disminuye la cantidad de memoria necesaria para procesar esa capa.

Esta técnica suele aplicarse de tres formas: La primera es el pooling promediado (average pooling), luego, se tiene el pooling máximo (max pooling) y, finalmente, el pooling mínimo (min pooling). Es importante acotar, además, que al igual que con el kernel, los parámetros de tamaño (dimensionalidad) y salto (stride) afectan el desempeño del sub muestreo o agrupación. Usualmente la elección de uno u otro tipo dependerán del problema a tratar, sin embargo, es necesario mencionar que los métodos más utilizados son el de pooling promedio y el de pooling máximo, esto es pues, porque suelen arrojar mejores resultados en la agrupación de características.

2.2.2.5.1. Average Pooling

La agrupación promediada consiste en tomar el valor promedio generado a partir de los valores que quedan dentro del rango del kernel convolucional. Esto es pues, un grupo de píxeles a los que se les aplica un promedio.

2.2.2.5.2. Max Pooling

Una agrupación máxima selecciona el máximo valor de entre todos aquellos presentes o dentro del rango del kernel convolucional. Esta variante de la agrupación o sub muestreo suele ser la más usada en la mayoría de los problemas que involucran la creación de una red neuronal convolucional.

2.2.2.5.3. Min Pooling

En contraparte a la agrupación por máximos, la agrupación por mínimos selecciona el mínimo valor de entrada de entre todos aquellos presentes o dentro del rango del kernel convolucional.

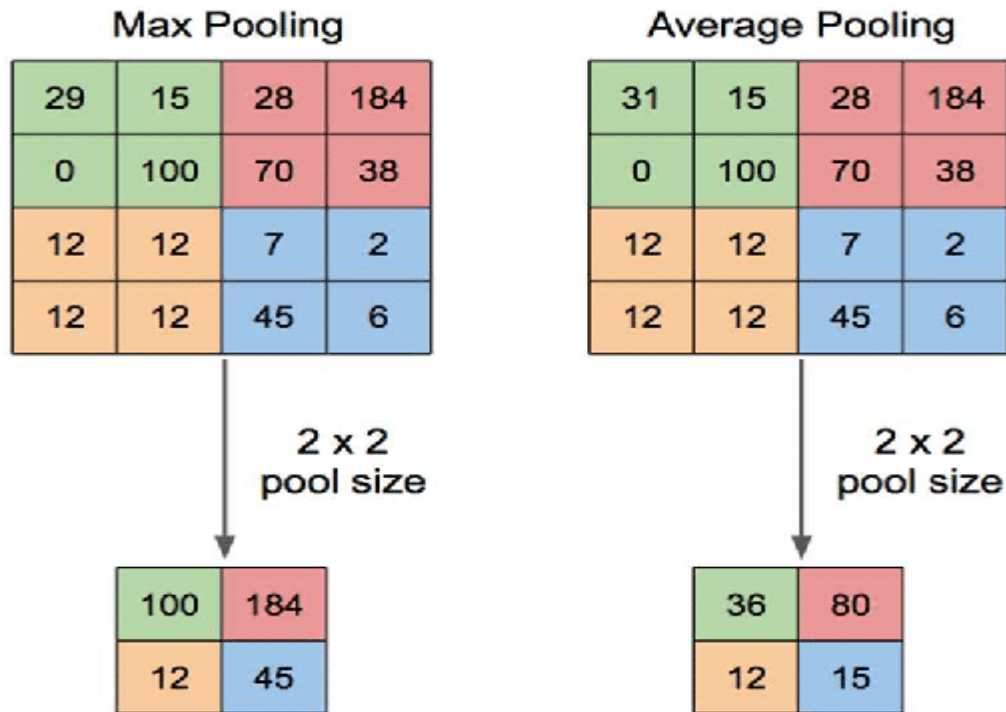


Figura 2.4: Aplicación del Max pooling y Average pooling.

Los recuadros superiores indican los valores de la entrada y luego de aplicar una agrupación con un kernel o filtro de 2x2, según el tipo, se obtiene una agrupación máxima (recuadro inferior izquierdo) o una agrupación promediada (recuadro inferior derecho)

Fuente: Yani, M. *et al* (2019) [researchgate.net/figure/Illustration-of-Max-Pooling-and-Average-Pooling-Figure-2-above-shows-an-example-of-max_fig2_333593451]

2.2.2.6. Stride

Se refiere al tamaño del salto o movimiento de un kernel o ventana convolucional. Usualmente define el movimiento en dos dimensiones más dos factores adicionales, como lo son el número de bache y el número de entradas (nueva dimensión de profundidad). Así pues, cuando se define el tamaño del salto en dos dimensiones (x, y) se asume que el factor de bache y la profundidad son uno (1, x, y, 1). Esto indica que el salto se hará sobre una muestra tomando x píxeles de ancho, con y píxeles de alto, y con profundidad (entrada) uno.

Es importante denotar además, que la entrada al operador convolucional termina siendo de cuatro dimensiones, esto es pues:

- a) Tamaño del Bache
- b) Filas de la imagen (x)
- c) Columnas imagen (y)
- d) Canales entrada (colores)

2.2.2.7. Feature Maps

Las características, o mapas de características representan el resultado del proceso de convolución para una capa determinada de la red neuronal. En estos mapas se evidencia que patrones o características se están reconociendo, en función de los parámetros de convolución, como lo son el tamaño del kernel y el tamaño del salto. En las capas más tempranas los patrones suelen ser bastante simples, van desde agrupaciones de píxeles de color hasta formas geométricas básicas o primitivas. Luego, los patrones van incrementando su complejidad a medida que se incrementa la profundidad en la red neuronal.

Sin embargo, los patrones de las últimas capas tienden a ser más complejos de reconocer para un humano pero, esto no representa problema alguno para la red neuronal, pues, los pesos que hacen referencia a esos patrones le indican exactamente a que está haciendo referencia alguna u otra característica.

En este sentido, es importante acotar que la generación de mapas de características también puede ser mejorada mediante las funciones de activación, por ejemplo: Al reducir ruido y aplicar ReLU es posible mejorar la apreciación de valores relevantes al eliminar el conjunto de valores cuyo umbral de activación sea negativo, reemplazándolos con ceros, y dejando nada más los valores mayores a cero.

Finalmente, cabe destacar que los mapas y filtros consumen recursos, por tanto, un ejemplo de optimización puede conseguirse si se eliminan ciertos mapas en cada capa. De acuerdo con Li *et al.* (2017), es posible recortar los filtros y ciertos mapas de características de un modelo bien entrenado para mejorar la eficiencia computacional.

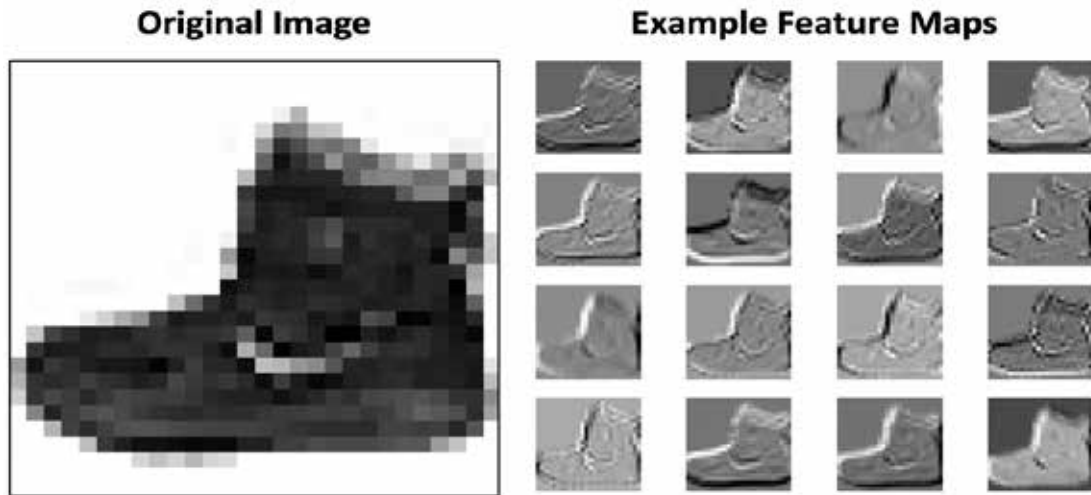


Figura 2.5: Mapas de características generados por una convolución.

Fuente: West, M. (2019) [bouvnet.no/bouvnet-deler/understanding-convolutional-neural-networks-part-1]

2.2.3. Fotometría

La ciencia de medir el flujo que recibimos de los cuerpos celestes es a lo que llamamos fotometría. Esto es pues, el proceso de medición del flujo sobre la base de los distintos tipos de longitud de onda del espectro de radiación (Romanishin, 2006).

Las medidas fotométricas en múltiples bandas se usan para determinar el color de un objeto, por ejemplo, que tan azul, rojo o neutral es el objeto en cuestión comparado con los colores del sol. Además, para la observación de objetos cuya luz débil, la fotometría es el método preferido en contraposición a la espectroscopia, esto se debe a que se pueden obtener mejores proporciones de señal – ruido, debido a que la luz está menos dispersa, cosa que no es lo suficientemente óptima para la espectroscopia (DeMeo, 2010).

2.2.4. The Sloan Digital Sky Survey (SDSS)

Es un proyecto que ha proveído una base de datos sin precedentes, principalmente de observaciones fotométricas de estrellas y galaxias, además, se puede

decir que ha establecido el estándar de facto para las observaciones fotométricas futuras y gran parte del procesamiento de imagen fotométrico futuro (Bessell, 2005).

Esto se debe pues, a su novedoso sistema fotométrico de cinco colores, es decir, un sistema CCD de banda ancha (u'g'r'i'z') que cubre longitudes de onda desde los 3000 hasta los 11 000 Å (Fukugita, 1996).

2.3. Definición de términos

Angstrom: Unidad del sistema internacional de unidades para denotar la longitud de onda.

ANN: Abreviación de Artificial Neuronal Networks, se utiliza para referirse a las Redes Neuronales Artificiales.

CNN: Abreviación de Convolutional Neuronal Networks, se utiliza para referirse a las Redes Neuronales Convolucionales.

DNN: Abreviación de Deep Neuronal Networks, se utiliza para referirse a las Redes Neuronales profundas.

Objeto Astronómico: Es una entidad física significativa, una asociación o estructura que la ciencia ha inquirido que existe en el universo. Así bien, un objeto astronómico puede consistir en un conjunto de cuerpos celestes, por ejemplo, el cinturón de asteroides o un sistema estelar.

CAPÍTULO III

MARCO METODOLÓGICO

El marco metodológico se refiere a metodología del trabajo, incluye el tipo de investigación, las técnicas y los procedimientos que serán utilizados para llevar a cabo la indagación. Es como se realiza el estudio para responder el problema planteado. Sobre el marco de la investigación Sabino (2002) expresa lo siguiente: “La metodología, por otra parte, podrá constituirse en un capítulo especial solo en los casos en que ello se justifique: en investigaciones de campo o de laboratorio, o cuando posean singularidades que obliguen a una exposición razonada y explícita de la misma. En este sentido los rasgos que caracterizan la investigación son:

3.1. Diseño de Investigación

Arias (2012), establece que el diseño de investigación es la estrategia general que adopta el investigador para responder al problema planteado. Representa las estrategias que se deben seguir para el desarrollo de la investigación, es decir, se debe indicar de manera estructurada y funcional las etapas del proceso. En atención al diseño, la investigación se clasifica en: documental, de campo y experimental (p. 27).

Así bien, indica que la investigación documental es un proceso basado en la búsqueda, recuperación, análisis, crítica e interpretación de datos secundarios, es decir, los obtenidos y registrados por otros investigadores en fuentes documentales: impresas, audiovisuales o electrónicas.

Luego, expresa que la investigación de campo es aquella que consiste en la recolección de datos directamente de los sujetos investigados, o de la realidad donde ocurren los hechos (datos primarios), sin manipular o controlar variable alguna, es decir, el investigador obtiene la información pero no altera las condiciones existentes.

Finalmente, señala que la investigación experimental es un proceso que consiste en someter a un objeto o grupo de individuos a determinadas condiciones, estímulos o

tratamiento (variable independiente), para observar los efectos o reacciones que se producen (variable dependiente).

Por consiguiente, el diseño de este proyecto de investigación corresponde al de Investigación de Campo, considerando que se apoya en informaciones que provienen de la realidad o escenarios naturales.

3.2. Nivel de la Investigación

Arias (2012) señala que el nivel de investigación se refiere al grado de profundidad con que se aborda un fenómeno u objeto de estudio. En relación a esto, indica que el nivel de la investigación puede ser de dos tipos, esto es, (a) Exploratoria o (b) Descriptiva. La exploratoria es aquella que se efectúa sobre un tema u objeto desconocido o poco estudiado, por lo que sus resultados constituyen una visión aproximada de dicho objeto, es decir, un nivel superficial de conocimientos. Mientras, la descriptiva consiste en la caracterización de un hecho, fenómeno, individuo o grupo, con el fin de establecer su estructura o comportamiento.

En función de lo anterior, es posible establecer que este proyecto de investigación pertenece al nivel descriptivo.

3.3. Tipo de la Investigación

Según la Universidad Pedagógica Experimental Libertador (UPEL) (2006) un proyecto de investigación puede ser del tipo: (a) Investigación de Campo, (b) Investigación Documental, (c) Proyectos Factibles, (d) Proyecto especiales (p. 11). En este sentido además, establece que un proyecto especial consiste en creaciones tangibles, susceptibles de ser utilizadas como soluciones a problemas demostrados, o que respondan a necesidades e intereses de tipo cultural. El desarrollo de software y hardware, prototipos y productos tecnológicos en general suelen entrar en esta categoría.

En consecuencia, es posible determinar que este trabajo de investigación entra en la categoría de proyecto especial, pues, desarrolla componentes de software para mejorar y optimizar el proceso de detección y clasificación de cuerpos, fenómenos y

objetos astronómicos a partir de imágenes tomadas en los proyectos de investigación espacial.

3.4.Población y Muestra

Según la Universidad Pedagógica Experimental Libertador (UPEL) (2006, p. 25), la población y muestra se describen como el universo afectado por el estudio, el grupo seleccionado, las características, tamaño y metodología seguida para la selección de la muestra o de los sujetos, la asignación de las unidades a grupos o categorías y otros aspectos que se consideren necesarios.

3.4.1. Población

De Barrera (2008), define la población como un: “conjunto de seres que poseen la característica o evento a estudiar y que se enmarcan dentro de los criterios de inclusión” (p.141)

En este contexto, nuestra población será el conjunto de investigadores que pertenecen al Centro de Investigaciones de Astronomía "Francisco J. Duarte" (CIDA).

3.4.2. Muestra

Balestrini (2006), señala que: “una muestra es una parte representativa de una población, cuyas características deben producirse en ella, lo más exactamente posible. (p.141)”

Para Castro (2003), la muestra se clasifica en probabilística y no probabilística. La probabilística, son aquellas donde todos los miembros de la población tienen la misma opción de conformarla, a su vez pueden ser: muestra de azar simple, muestra de azar sistemático, muestra estratificada o por áreas. La no probabilística, la elección de los miembros para el estudio dependerá de un criterio específico del investigador, lo que significa que no todos los miembros de la población tienen igualdad de oportunidad de conformarla. La forma de obtener este tipo de muestra es: muestra intencionada y muestra accidental o sin norma.

Así bien, para este proyecto de investigación en particular, la muestra coincide con la población.

3.5. Técnicas e Instrumentos de recolección de datos

Tamayo y Tamayo (2003), expone que la recolección de los datos depende en gran parte del tipo de investigación y del problema planteado para la misma, y puede efectuarse desde la simple ficha bibliográfica, observación, entrevista, cuestionarios o encuestas y aun mediante ejecución de investigaciones para este fin.

3.5.1. Técnicas

Según Arias (2012) “se entenderá por técnica de investigación, el procedimiento o forma particular de obtener datos o información”. Las técnicas son particulares y específicas de una disciplina, por lo que sirven de complemento al método científico, el cual posee una aplicabilidad general.

Así mismo, enumera que entre las técnicas se tienen: (a) La encuesta, (b) La entrevista, (c) La observación.

- a) La encuesta: Esta técnica puede aplicarse de dos formas, oral o escrita. La forma oral provee los instrumentos de: Guía de encuesta, Grabadora y Cámara de video. Mientras, la escrita hace uso del cuestionario.
- b) La entrevista: Esta técnica también puede aplicarse de dos formas, es decir, puede ser de forma estructurada y no estructurada. La estructurada hace uso de los instrumentos como la guía de entrevista, grabador / cámara de videos. La no estructurada, por otra parte, hace uso de instrumentos como la libreta de notas, grabador / cámara de video.
- c) La observación: Similar a la entrevista, puede ser de tipo estructurada o no estructurada. El tipo estructurado hace uso de instrumentos como la lista de cotejo y la escala de estimación. En cambio, la no estructurada hace uso del diario de campo y las cámaras: fotográficas o de video.
- d) Análisis documental: Esta técnica comprende las fichas, o la computadora y sus unidades de almacenaje.

Tomando en cuenta lo anterior, se indica que la técnica seleccionada para este proyecto de investigación es la encuesta.

3.5.2. Instrumentos

De acuerdo con Arias (2012, p. 68), un instrumento de recolección de datos es cualquier recurso, dispositivo o formato (en papel o digital), que se utiliza para obtener, registrar o almacenar información.

En referencia a lo anterior, es necesario expresar que el instrumento seleccionado será un Cuestionario del tipo Likert, debido a que el diseño de la investigación es de campo y la técnica es la encuesta. Este instrumento hace uso de un conjunto de preguntas o ítems que se deben marcar o no, en función de la opinión de aquel que está siendo sometido al instrumento.

3.6. Validez del Instrumento

Se refiere al grado en que el instrumento de recolección de información mide lo que en realidad se desea medir (Arias, 2012). Ésta, se determina a través con un procedimiento llamado juicio de expertos donde tres (3) especialistas uno (1) en metodología y dos (2) en contenido emitirán una opinión y determinarán si el instrumento puede ser aplicado o no.

3.7. Confiabilidad del Instrumento

Hernández, Fernández y Baptista (2010) la confiabilidad de un instrumento de medición se mide a través de diferentes técnicas que buscan la aplicación repetida al mismo objeto buscando resultados similares, con la finalidad de verificar si la información obtenida es confiable, permitiendo así alcanzar los objetivos de la investigación. En este sentido Ruíz (2002), señala que para estimar la confiabilidad se puede utilizar uno de los siguientes métodos:

1. Confiabilidad de re-aplicación de pruebas (test – retest): Consiste en administrar dos veces la misma prueba a un mismo grupo en un intervalo de tiempo relativamente corto. Se calcula por la fórmula de correlación lineal de Pearson.
2. Confiabilidad de versiones equivalentes (pruebas paralelas): Se utiliza para el caso de dos pruebas que miden el mismo constructo, se debe elaborar dos

versiones de la misma prueba y se administra a la misma muestra. Para calcular el coeficiente de confiabilidad se procede igual a la re-aplicación de pruebas.

3. Confiabilidad de consistencia interna (homogeneidad): Permite determinar el grado en que ítems de una prueba están correlacionados entre sí. Existen diferentes procedimientos para estimar la confiabilidad, pero los más conocidos son:
 - a. Kuder y Richardson 20 (aplicable a pruebas de ítems dicotómicos, es decir, las respuestas son correctas o incorrectas)
 - b. Alpha de Cronbach (no existen respuestas correctas o incorrectas, sino que el sujeto marca el valor que mejor representa su respuesta)
 - c. Dos mitades, corregido por la fórmula de Spearman – Brown

Así pues, para esta investigación se utilizará el método de “Alpha de Cronbach”, pues no requiere que las respuestas sean correctas o incorrectas, y según Hernández, Fernández y Baptista (2010), es un método que puede ser utilizado para cualquier cantidad de alternativas, sean pares o impares, pues el mismo no divide en dos mitades los ítems del instrumento de medición, simplemente se aplica la medición y se calcula el coeficiente. Estos dos factores la hacen ideal pues, para verificar la confiabilidad de un instrumento utilizado para recabar requerimientos funcionales y no funcionales.

3.8. Técnicas de Análisis

Una vez aplicados los instrumentos, la técnica de análisis de información a utilizar serán los gráficos, especialmente un diagrama de barras o un diagrama de dispersión. Esto facilitará pues, el entender y discernir las verdaderas necesidades.

3.9. Fases de la Investigación

Las fases de este proyecto de investigación se estructuran bajo el esquema de tres fases principales. Así bien, cada una de estas fases es esencial para la elaboración del proyecto, pues, cada una aporta la información, experiencia y funcionalidad necesarias para completar el proyecto exitosamente. De tal forma que:

Fase I: “Determinación de los requerimientos funcionales y no funcionales mediante las técnicas de recolección de datos”

Este proceso es esencial y debe ejecutarse antes del desarrollo de una aplicación o componente de software, pues, más allá de considerarse una buena práctica en la industria del desarrollo de software, los autores le definen como el aspecto o proceso inicial en el que se fundamenta el desarrollo de toda aplicación o sistema de software.

En este sentido, Pressman (2010, p. 129) indica que el espectro amplio de tareas y técnicas que llevan a entender los requerimientos se denomina ingeniería de requerimientos. Dicho proceso comienza durante la actividad de comunicación, y continúa en la de modelado, por lo que debe adaptarse a las necesidades del proceso, del proyecto, del producto y de las personas que hacen el trabajo.

Durante esta fase obtener los requerimientos funcionales y no funcionales mediante las técnicas de recolección de datos.

Fase II: “Diseño de los componentes globales y modelos de datos que constituyen la aplicación mediante la experiencia de usuario y los principios de consistencia y durabilidad”:

Se utilizarán las distintas metodologías y modelos de arquitectura para establecer las bases de la aplicación, los modelos de datos.

Fase III: “Construcción de la aplicación en función de los requerimientos recabados y ejecutar pruebas de acuerdo a la metodología de trabajo”

Trata de la construcción de la aplicación en función de los requerimientos funcionales y los patrones de arquitectura de sistemas de información.

CAPÍTULO IV

RESULTADOS

El desarrollo de aplicaciones o sistemas de software se apoya en el uso de metodologías de desarrollo de software que proveen no sólo buenas prácticas de desarrollo, sino también de un enfoque que facilita la estructura básica a seguir para construir y dar mantenimiento a sistemas que deben cumplir ciertas restricciones o funcionalidades.

Así pues, este capítulo demuestra los resultados de aplicar los principios de la metodología de desarrollo ágil: “Método de Desarrollo de Sistemas Dinámicos” (DSDM) en la obtención de los requerimientos funcionales y no funcionales, el diseño de los componentes de la aplicación y la codificación o construcción de la misma. Por tanto, deben describirse las tres fases que componen el desarrollo de un proyecto de software según esta metodología. Esto es pues:

- Pre – Proyecto
- Ciclo de vida del Proyecto
- Post – Proyecto

El pre – proyecto se basa principalmente en la definición de roles dentro del proyecto. En este apartado resaltan las posiciones de “Visionario”, quién es alguien encargado de la identificación temprana de los requerimientos esenciales para la inicialización del proyecto. Similarmente, Pressman (2010, p.71) describe esta fase como el estudio de factibilidad que establece los requerimientos y restricciones básicas del negocio, asociados con la aplicación que se va a construir. En este sentido, la Fase I de los resultados corresponde a la fase de pre – proyecto de la metodología DSDM.

Similarmente, la fase del ciclo de vida del proyecto se divide en varias sub fases. Inicialmente, se realiza la sub fase del estudio de factibilidad, seguida de un estudio del negocio. Luego, en la iteración del modelo funcional se produce un conjunto de prototipos incrementales que demuestran al cliente la funcionalidad, y que sirven

para recabar requerimientos adicionales por medio de la obtención de retroalimentación de los usuarios cuando practican con el prototipo. Así mismo, y de forma simultánea a la sub fase anterior, en el diseño e iteración de la construcción ya se ha diseñado en base a una arquitectura de software y se tiene un plan de desarrollo.

Por último, la sub fase de implementación, cuyo objetivo es colocar los incrementos ya probados en el entorno de trabajo de los usuarios del sistema para determinar futuros requerimientos a desarrollar, haciendo notar que:

- a) El incremento tal vez no sea el de 100% final.
- b) Quizá se pidan cambios cuando el incremento se ponga en su lugar.

En cualquier caso, el trabajo de desarrollo DSDM continúa y vuelve a la actividad de iteración del modelo funcional. En función de lo anterior, es posible denotar que la fase o ciclo de vida del proyecto, tiene una correspondencia directa con las Fases II y III, que serán expuestas en este capítulo.

Además, la fase del post proyecto abarca las actividades que ocurren luego de que el equipo encargado del proyecto se ha disuelto. Esta fase incluye soporte y actividades de mantenimiento, y opcionalmente, una reseña de post implementación para evaluar el sistema en uso. Es de hacer notar que, aunque esta fase no tiene correspondencia directa con ninguna de las fases de resultados expuestas en este capítulo, debe ser descrita a modo de mejorar la comprensión sobre la metodología DSDM.

Es entonces, que cada una de las tres fases expuestas en este capítulo, contribuye con el desarrollo de esta aplicación multiplataforma, que facilita y abstrae el proceso de implementación de una red neuronal artificial (convolucional) para el procesamiento de imágenes provenientes de observaciones de campo profundo (Deep field observations) al esconder la implementación de bajo nivel, mediante componentes gráficos e interfaces de usuario de alto nivel.

4.1. Fase I:

Durante esta fase la técnica usada fue una encuesta de diez preguntas, apoyada en una escala tipo Lickert en formato digital como instrumento, con una escala de apreciación del uno al diez (numérica / nominal). Inicialmente, se quería obtener los requerimientos funcionales y no funcionales en función de la opinión de los investigadores del Centro de Investigaciones de Astronomía (CIDA), sin embargo, el sitio web de la institución es inaccesible y, adicionalmente, se pudo conocer de forma extraoficial que la institución ya no opera como centro de investigación, y que sólo hay personal de seguridad en la misma para el resguardo de los equipos e instrumentos.

En consecuencia, se optó por crear una encuesta online y aplicarla mediante propagación en la web. Para esto, se creó la encuesta con todos sus ítems en un formulario de google forms y posteriormente, se publicó el link o url de la misma en la plataforma social Twitter, con hashtags o etiquetas dedicados a la astronomía o, noticias relacionadas de alguna forma. Así pues, cualquier persona del público objetivo podría revisar los ítems presentes en la encuesta y contestar en función de sus intereses o conocimientos.

En relación con lo anterior, las preguntas de respuesta única (ítems) expuestos en la encuesta son:

- 1) ¿Has escuchado hablar de las redes neuronales artificiales?
- 2) De haber oído de ellas, o no ¿Consideras que una aplicación que facilite la creación de una RNA para la clasificación de imágenes pueda ser útil?
- 3) De estar de acuerdo ¿Debería contar esta aplicación con un esquema de color cómodo, que facilite el uso de la aplicación durante largas horas de uso?
- 4) ¿Consideras que es necesario el visualizar o graficar los datos numéricos de bajo nivel que maneja la aplicación detrás de escenas (backend)?
- 5) ¿Es importante la compatibilidad de una aplicación entre diferentes sistemas operativos (OS)?

- 6) ¿Crees que la interfaz gráfica facilite el uso del backend de bajo nivel para crear RNA's, aunque pueda causar cuellos de botella en la tarea de implementación y problemas de rendimiento?
- 7) ¿Debería mantenerse un registro sobre quién modifica la estructura de la red neuronal, a fin de evitar cambios no autorizados en los parámetros de entrenamiento y en la estructura de la misma?
- 8) ¿Debería contar la aplicación con módulos que separen las actividades de lectura, creación, entrenamiento y predicción de la RNA?
- 9) ¿Considera necesario el poder crear, guardar y leer conjuntos de datos de imágenes mediante la aplicación, evitando el tener que pre procesarlas en cada ejecución de lectura?
- 10) ¿Es importante que la aplicación muestre información comparativa de la clase predicha vs la clase real de la imagen en las tareas de predicción?

4.1.1. Requerimientos funcionales

4.1.1.1. Conjuntos de datos de imágenes

- a) Los conjuntos de datos creados por la aplicación se guardarán como un como un archivo con formato de datos jerárquico (Hierarchical Data Format), en su API HDF5.
- b) Antes de crearse el conjunto de datos deberá especificarse el tamaño aproximado que ocupará el mismo en el disco duro del equipo.
- c) Los conjuntos de datos podrán ser creados de dos formas principales. La forma de archivo "file" es un archivo de texto que contiene la ruta absoluta, nombre de la imagen y clase a la que pertenece. Mientras, la forma de directorio "folder" deberá ser una carpeta raíz, en cuyo interior estén las demás imágenes separadas en carpetas según corresponda la clase, y dónde cada carpeta estará nombrada según la clase que corresponda.

4.1.1.2. Red neuronal convolucional

- a) Los parámetros de configuración, tales como el número de filtros, tamaño del filtro, tamaño del salto, número de unidades, activaciones, etc., serán configurables desde el apartado de Entrenamiento.
- b) Se deberán guardar los parámetros entrenados del modelo como la matriz de pesos (weights) y neuronas constantes (bias).
- c) Se podrá restaurar el modelo guardado para ser usado en tareas de detección y validación.

4.1.1.3. Visualización

- a) Se podrá elegir entre dos opciones de visualización. La primera permitirá apreciar el proceso de detección en los mapas de características de las capas intermedias (hidden layers) y la influencia de los parámetros de convolución, tales como el tamaño del filtro (kernel) y el tamaño del salto (stride) dentro de la aplicación. Por otra parte, la segunda opción permitirá visualizar el proceso de entrenamiento completo (estadístico) del modelo y, dependiendo de los parámetros escogidos para el modelo, también se podrán visualizar imágenes del proceso de detección y mapas de características. Esta opción de visualización abre una ventana emergente del navegador para mostrar los datos requeridos.
- b) Los archivos de visualización o resumen serán almacenados como archivos de eventos (events file) de Tensorflow.

4.1.1.4. Clasificación y Predicción

- a) En el proceso de clasificación y predicción se deberá cargar el modelo pre entrenado antes de evaluar.
- b) Deberá decidirse la estructura de carga de la o las imágenes antes de evaluar el modelo. Puede ser una ruta absoluta hacia el archivo (en caso de una imagen) o una carpeta (directorio). Si el modo de carga es

- directorio, el usuario deberá elegir pasar a la siguiente imagen mediante los controles de la interfaz para continuar con el proceso de validación.
- c) En la ventana de validación, se mostrará información comparativa de la clase predicha por el modelo.

4.1.2. Requerimientos no funcionales

4.1.2.1. Seguridad

- a) La aplicación no permitirá la ejecución de scripts que no se hayan especificado explícitamente en un documento de configuración o requerimientos de módulos (requirements.txt)
- b) El apartado o módulo de configuración de la red neuronal emitirá un pequeño log o registro con una firma o clave en hash que indique la hora, y el nombre del usuario (PC o alias) que modificó los parámetros de configuración.
- c) Verificará que los formatos de archivo son los debidos, además comprobará la transmisión y codificación de la información (exportación) a otros formatos de archivo diferente.
- d) El entrenamiento de la red neuronal requerirá un alias del usuario que esté creando o modificando los parámetros de entrenamiento, etiquetado o preferencias.

4.1.2.2. Rendimiento

- a) La graficación de componentes relacionados con los conjuntos de entrenamiento estarán en función de los parámetros, tamaño del filtro y de las unidades escondidas.
- b) El equipo deberá disponer de al menos 8 GB de memoria RAM.
- c) No se deberán usar imágenes con una resolución mayor a los 500 x 500 píxeles.

4.1.2.3. Portabilidad

- a) La aplicación deberá poder ejecutarse desde cualquier entorno o sistema operativo (Windows, Linux- Like, MacOS) siempre y cuando, la

arquitectura del mismo sea de 64 bits.

4.1.2.4. Usabilidad

- a) Usará componentes gráficos separados de la lógica de programación
- b) Representará los controles gráficos como una secuencia lógica de acciones

4.1.2.5. Integridad de los datos

- a) Los datos serán accedidos y modificados mediante la interfaz gráfica, para disminuir la probabilidad de corrupción de datos.
- b) La codificación y decodificación de los datos se hará mediante los algoritmos previstos para cada formato de archivo específico y se enlazará el resultado con su relación en una base de datos.

4.1.2.6. Confiabilidad

- a) La aplicación hará uso de módulos de aplicación científico y de carácter de código abierto, para mejorar la precisión e implementación de algoritmos de carácter astronómico.

Cuadro 1

Relación entre los ítems y los requerimientos funcionales y no funcionales

| Ítems | Requerimiento funcional | Requerimiento no funcional |
|-----------|--|----------------------------|
| Ítem n° 1 | 4.1.1.2. Red Neuronal Convolutiva 4.1.1.4. Clasificación y Predicción | N/A |
| Ítem n° 2 | N/A | 4.1.2.6. Confiabilidad |
| Ítem n° 3 | N/A | 4.1.2.4. Usabilidad |
| Ítem n° 4 | 4.1.1.3. Visualización | 4.1.2.2. Rendimiento |
| Ítem n° 5 | N/A | 4.1.2.3. Portabilidad |

| | | |
|------------|---|--|
| Ítem n° 6 | 4.1.1.1. Conjuntos de datos de imágenes 4.1.1.3. Visualización 4.1.1.4. Clasificación y Predicción | 4.1.2.2. Rendimiento 4.1.2.4. Usabilidad |
| Ítem n° 7 | 4.1.1.2. Red neuronal convolucional | 4.1.2.1. Seguridad |
| Ítem n° 8 | 4.1.1.2. Red neuronal convolucional 4.1.1.3. Visualización 4.1.1.4. Clasificación y Predicción | 4.1.2.4. Usabilidad |
| Ítem n° 9 | 4.1.1.1. Conjuntos de datos de imágenes | 4.1.2.2. Rendimiento 4.1.2.5. Integridad de los datos |
| Ítem n° 10 | 4.1.1.4. Clasificación y Predicción | N/A |

4.2. Fase 2:

Las cuatro primeras sub fases del ciclo de vida de la metodología DSDM marcaron el flujo de trabajo a seguir para cumplir con el diseño de los componentes globales de la aplicación, es decir:

- Estudio de factibilidad
- Estudio de negocio
- Iteración del modelo funcional
- Diseño y Construcción de la iteración

4.2.1. Estudio de factibilidad

Los objetivos del estudio de factibilidad se basan en establecer si un modelo de desarrollo propuesto reúne los requerimientos de negocio de la organización, y luego se evalúa si la aplicación es un candidato viable para aplicarle el proceso DSDM. En este sentido, los requerimientos del negocio y la evaluación de la solución como

candidato se basan en una apreciación personal que, visto en retrospectiva, coinciden de los factores que motivaron el desarrollo del trabajo de investigación.

Cuadro 2

Estudio de factibilidad de acuerdo con la metodología DSDM

| Factores | Resumen |
|------------------------------|--|
| Propósito del proyecto | Facilitar las tareas de predicción de ciertos tipos de objetos, cuerpos y fenómenos astronómico – espaciales en amplios volúmenes de datos con archivos de tipo jpeg, jpg. |
| Audiencia objetivo | Investigadores, desarrolladores o público general que desee validar el proceso de predicción de una red neuronal convolucional respecto de objetos, cuerpos y fenómenos astronómico – espaciales. |
| Riesgos detectados | Que el usuario no tenga un mínimo de conocimiento del proceso de convolución y haga predicciones con un modelo mal estructurado, restaría credibilidad al propósito de la aplicación. |
| Solución propuesta | Una aplicación que facilite las tareas de crear, parametrizar y entrenar una red neuronal convolucional, que sirva de base para predecir distintos tipos de objetos, cuerpos y fenómenos astronómico – espaciales. |
| Justificación de la solución | La existencia de investigadores, organizaciones e instituciones que estudian la gran cantidad de objetos, cuerpos y fenómenos astronómicos que están plasmados en millones de imágenes, y en inmensos volúmenes de datos gráficos. |

4.2.2. Estudio de negocio

La etapa del estudio de negocio se centra en la definición de la arquitectura del sistema, el plan de desarrollo y una lista de los requerimientos prioritarios que se deben trabajar desde un inicio.

4.2.2.1. Definición de la arquitectura del sistema

La arquitectura del sistema representa a los componentes de software, las reglas que definen la estructura y la relación entre las partes que componen el sistema. Esta estructura se fundamenta, además, en un estilo arquitectónico que ayuda a definir la relación entre los componentes y las funciones que llevan a cabo.

Así bien, la arquitectura de la aplicación está representada por el tipo jerárquico, o de capas jerárquicas. Este estilo arquitectónico define dos funciones principales para cada capa:

- 1) Proveer servicios para las capas superiores.
- 2) Llamar a las funciones de capas más bajas.

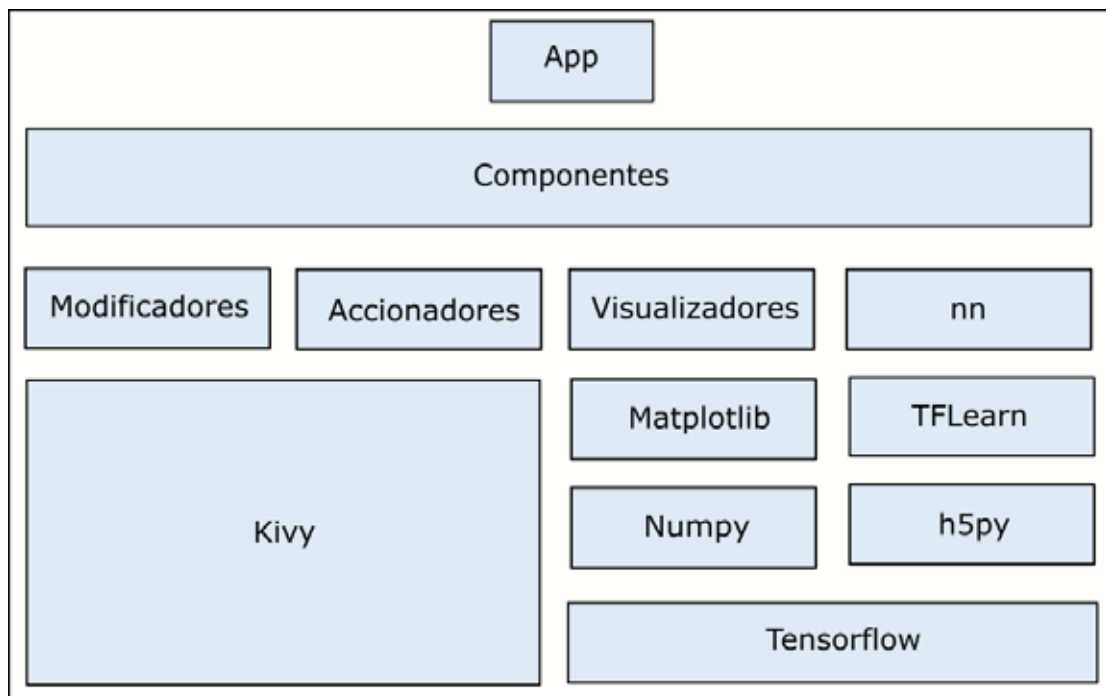


Figura 4.1: Arquitectura de la aplicación

4.2.2.2. Plan de desarrollo

La aplicación siguió un plan de desarrollo enfocado en satisfacer y completar los requerimientos funcionales y no funcionales que fueron recabados mediante las técnicas de recolección de datos. Esto quiere decir pues, que el plan prevé un conjunto de requerimientos prioritarios, es decir, los que agregan valor a la aplicación y deben implementarse cuanto antes y aquellos que, aunque importantes, pueden considerarse complementarios, y pueden implementarse con más calma.

Así pues, se estableció una ruta (roadmap) con los requerimientos que debían completarse en un lapso de tiempo de cuatro meses y, adicionalmente, se estimó el tiempo que tomaría completar cada uno de los hitos (milestones) asociados a cada requerimiento.



Figura 4.2: Plan de Desarrollo

Los recuadros de la izquierda indican el requerimiento o conjunto de requerimientos, mientras que los recuadros de la derecha indican los hitos asociados a ese

requerimiento. Similarmente, las letras dentro de los recuadros de la derecha indican el requerimiento específico al cual se hace referencia.

4.2.2.3. Requerimientos prioritarios

Los requerimientos prioritarios concuerdan en su mayoría con aquellos definidos en el plan de desarrollo, sin embargo, es importante listarlos y definirlos para mejorar la comprensión sobre el valor que agrega cada uno de ellos a la solución propuesta.

Cuadro 3

Requerimientos prioritarios encontrados en el estudio de negocio

| Prioridad | Requerimiento | Resumen |
|-----------|----------------------------|--|
| 1 | Red neuronal convolucional | El requerimiento de red neuronal convolucional abarca la creación y entrenamiento de la red neuronal y su incorporación con la aplicación. En el plan de desarrollo es, de hecho, el requerimiento con mayor tiempo de desarrollo estimado. |
| 1 | Rendimiento | Representa uno de los factores de mayor importancia pues, es sabido que el proceso de entrenamiento de una red neuronal requiere de una cantidad de recursos de hardware considerable. Además, está contemplado como un problema en aquellas organizaciones que no cuentan con equipos lo suficientemente potentes para realizar estas operaciones de cómputo. |
| 2 | Clasificación y Predicción | Después de la creación y entrenamiento de la red neuronal es necesario contrastar que tan exactas son las predicciones hechas por el modelo, pues, ayudan a determinar el acierto de los parámetros de entrenamiento. |

| | | |
|---|---------------|--|
| 2 | Visualización | La visualización se refiere al método de constatar visualmente el proceso de entrenamiento de la red neuronal convolucional, principalmente la forma en la cual los procesos de convolución y sub muestreo (pooling) van determinando el conjunto de características relevantes y sus pesos. |
| 2 | Usabilidad | Representa los principios de diseños aplicados a la construcción de la aplicación, y la experiencia de usuario que la misma puede ofrecer. Dicha experiencia comienza desde el proceso de creación de la red neuronal convolucional, creación del conjunto de datos, la visualización del proceso de entrenamiento y finaliza con el proceso de comprobación del modelo, es decir, la detección y clasificación. |
| 3 | Portabilidad | <p>Es el uno de los aspectos objetivo al momento de plantear la solución, y se debe principalmente a que, según las estadísticas de uso de sistemas operativos, los más usados en el ámbito científico son Linux – Like. Esto beneficiaría al tipo de usuario objetivo como los investigadores científicos, universitarios e incluso instituciones.</p> <p>En contra parte, el sistema operativo más utilizado por usuarios del tipo objetivo como organizaciones, aficionados y universitarios, tiende a ser Windows.</p> |

4.2.3. Iteración del modelo funcional

En esta fase se crean prototipos funcionales a partir de los requerimientos listados en las sub fases anteriores. Estos prototipos se someten a la interacción de clientes o usuarios de forma tal que provean una realimentación que ayude a mejorar, o añadir más requerimientos no funcionales de la aplicación.

Sin embargo, para este aspecto del proceso de desarrollo no se contó (al momento) con usuarios relacionados al área, y que pudiesen probar la aplicación. Por tanto, no se cuenta con una realimentación o requerimientos no funcionales adicionales en esta sub fase del ciclo de vida.

4.2.4. Diseño y Construcción de la Iteración

Esta sub fase del ciclo de vida se dividió en dos etapas. La etapa del diseño de la iteración corresponde su desarrollo a la fase dos de los resultados, mientras, la etapa de construcción de la iteración corresponde con la fase tres de los resultados.

4.2.4.1. Diseño de la Iteración

Esta etapa se basa en describir, fundamentar y diseñar el concepto de la aplicación, incluida su documentación, a partir de los objetivos recabados en las sub fases anteriores y considerando, además, dos de los aspectos clave de esta segunda fase:

- Diseño de los componentes globales basados en la experiencia de usuario
- Diseño de los modelos de datos en función de los principios de consistencia y durabilidad, que son dos de los principios ACID.

4.2.4.1.1. Actores del sistema

Cuadro 4

Descripción de los actores: Investigador / Usuario

| Actor | Descripción |
|------------------------|---|
| Investigador / Usuario | <p>Este actor fungirá como el usuario básico de la aplicación. Podrá crear, entrenar, visualizar y predecir a partir de los resultados provenientes de la red neuronal convolucional.</p> <p>Además, podrá crear conjuntos de datos necesarios para entrenar la red neuronal.</p> |

Cuadro 5

Descripción de los actores: Modificador

| Actor | Descripción |
|-------------|---|
| Modificador | <p>El actor “modificador” se encarga de alterar el estado visual de la aplicación. Esto es pues, la forma en la cual se presentan los menús, botones, widgets y otros componentes de la interfaz gráfica.</p> <p>Funciona, además, como un puente entre la interfaz de usuario y los demás actores, obteniendo la información que generan y adaptándola para mostrarla en pantalla.</p> |

Cuadro 6

Descripción de los actores: Accionador

| Actor | Descripción |
|------------|--|
| Accionador | <p>El actor “accionador” se encarga de propagar y responder a los eventos registrados o activados por otros actores. Es decir, funciona como un escucha o, lo que se conoce como un controlador en la arquitectura MVC.</p> <p>Se encarga, además, de controlar el vínculo de los datos de forma dinámica para controlar el comportamiento del actor “modificador” y “visualizador”.</p> |

Cuadro 7

Descripción de los actores: Visualizador

| Actor | Descripción |
|--------------|--|
| Visualizador | <p>El actor “visualizador” tiene el rol de procesar y transformar los datos dinámicos para presentarlos en un formato legible y entendible por los humanos.</p> <p>Principalmente, procesa flujos de bytes y dependiendo del proceso, los guarda a los archivos de evento o, muestra partes del proceso de entrenamiento en la ventana de la aplicación.</p> |

Cuadro 8

Descripción de los actores: Red Neuronal (nn)

| Actor | Descripción |
|----------------------|---|
| Red Neuronal (nn) | <p>Este actor funge como el intermediario entre la API de bajo nivel (tensorflow) y el actor “accionador”. Es decir, se encarga de obtener el flujo de datos necesario para crear la estructura de la red neuronal, guardarla y restaurarla.</p> <p>Interviene, además, en el proceso de entrenamiento de la red neuronal y provee los flujos de datos al actor “visualizador”.</p> |

4.2.4.1.2. Casos de Uso

Según Sommerville (2011, p.107), los casos de uso identifican las interacciones individuales entre el sistema y sus usuarios u otros sistemas. En su forma más sencilla, un caso de uso identifica a los actores implicados en una interacción, y nombra el tipo de interacción. Cada caso de uso debe documentarse con una descripción textual.

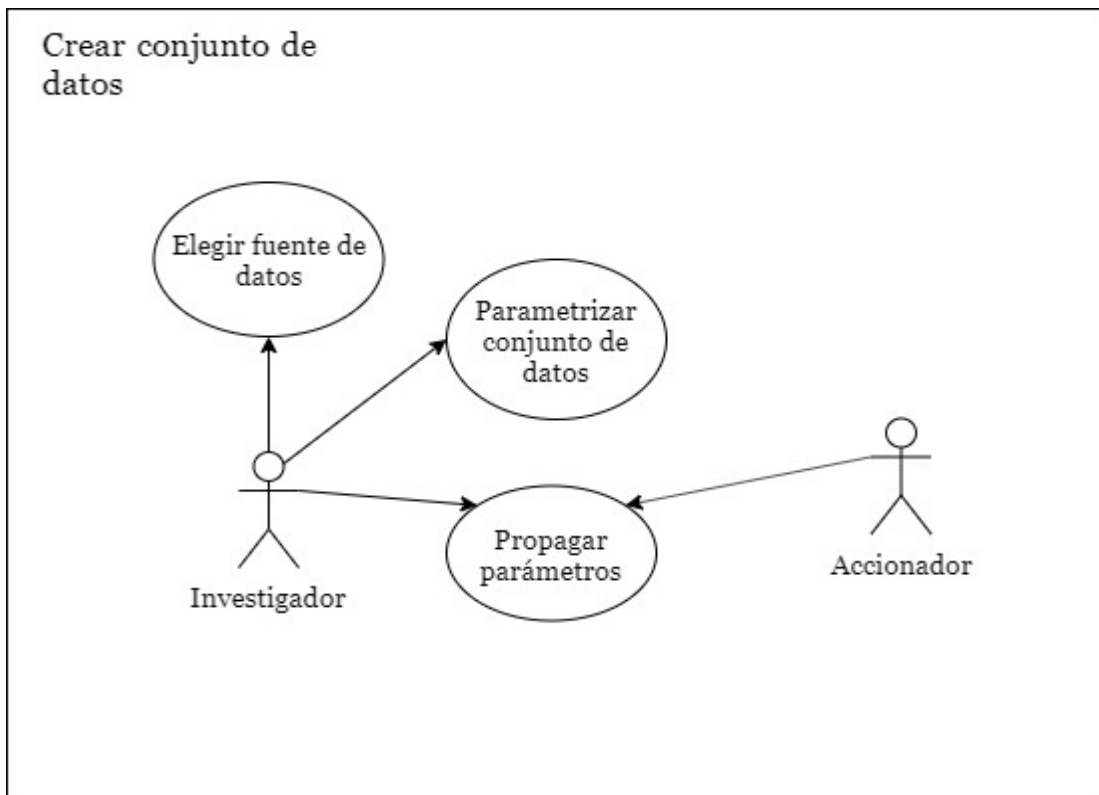


Figura 4.3: Caso de uso; Crear conjunto de datos

Cuadro 9

Definición de caso de uso: Crear conjunto de datos

| Definición del Caso de Uso | |
|----------------------------|-------------------------|
| Nombre de Caso de Uso: | Crear conjunto de datos |
| Actor Principal: | Investigador |

| | |
|--------------------------------------|---|
| Actores Secundarios: | Accionador |
| Objetivo en contexto: | Crear un conjunto de datos jerárquico a partir de imágenes, para eliminar la necesidad de tener que procesar y normalizar las imágenes cada vez que se cargan para ser usadas en los proceso de entrenamiento y validación. |
| Precondiciones: | Deben existir las imágenes en un directorio, ordenado de forma tal que, cada clase identificada sea un directorio único (el nombre de la carpeta debe ser el de la clase), y cada imagen perteneciente a una clase debe estar dentro del directorio de clase correspondiente. |
| Disparador: | El investigador decide crear un conjunto de datos con características espaciales específicas. |
| Condición de Término: | El conjunto de datos se guarda en disco de forma exitosa. |
| Condición de termino Fallida: | El espacio de almacenamiento es insuficiente, o el investigador cancela la operación. |
| Prioridad: | Baja. Puede implementarse para extender las capacidades de la aplicación y facilitar la creación de una red neuronal convolucional. |

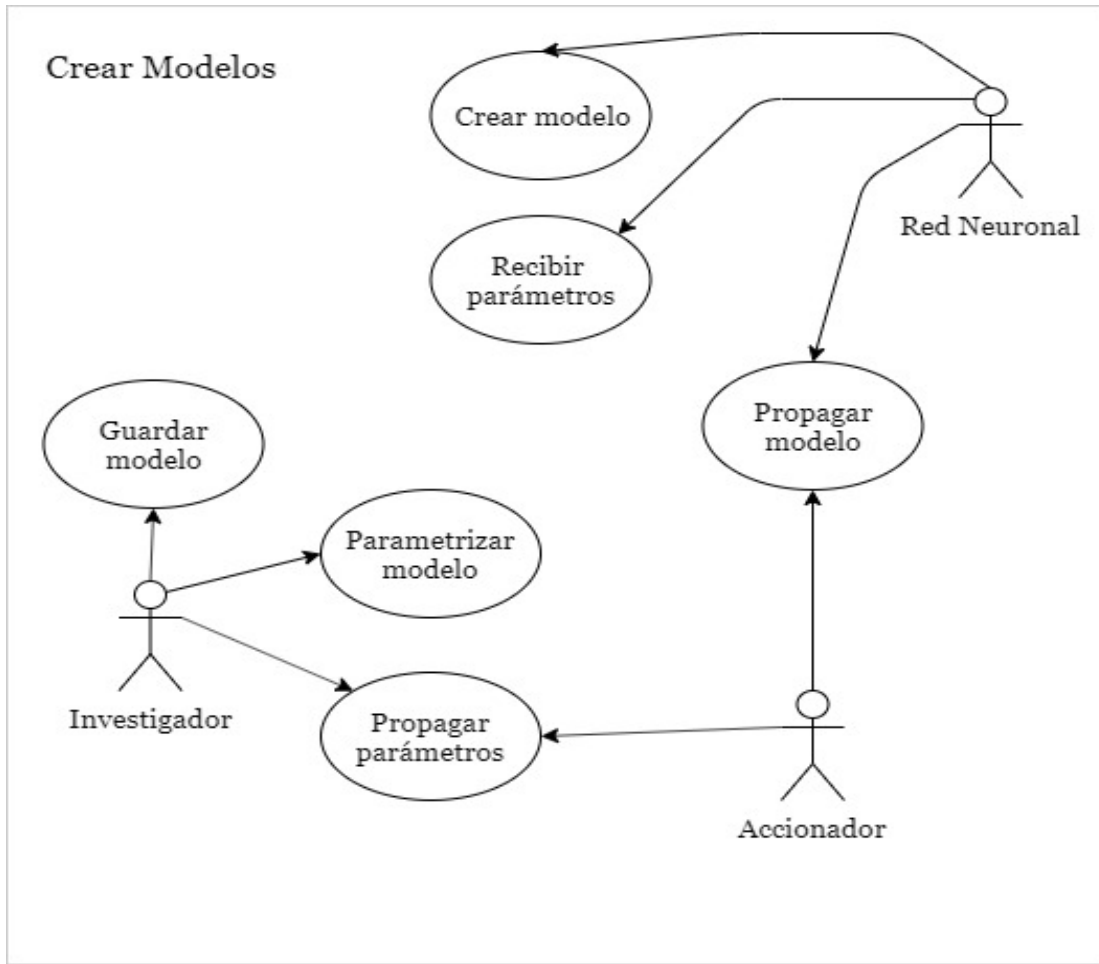


Figura 4.4: Caso de uso; Crear modelos

Cuadro 10

Definición de caso de uso: Crear modelos

| Definición del Caso de Uso | |
|-------------------------------|---------------------------|
| Nombre de Caso de Uso: | Crear modelos |
| Actor Principal: | Red Neuronal |
| Actores Secundarios: | Investigador, Accionador. |

| | |
|--------------------------------------|---|
| Objetivo en contexto: | Crear un modelo de red neuronal convolucional a partir de los parámetros introducidos por el investigador. Esto es pues, que cada capa pueda ser parametrizada en función del rol que cumple dentro del modelo convolucional. |
| Precondiciones: | <p>Debe seleccionarse un conjunto de imágenes con la forma definida en los parámetros. Este conjunto es necesario para inicializar el modelo. Se recomienda tenga al menos el 10% del total, de los datos.</p> <p>Se puede dar el caso también de que se guardaron los parámetros del modelo (no se inicializó), en este caso, debe seleccionarse el archivo con los parámetros del modelo, y luego seleccionar el conjunto de datos de inicialización.</p> |
| Disparador: | El investigador decide crear un modelo de red neuronal convolucional. |
| Condición de Término: | El modelo se guarda en disco, o se crea e inicializar y se guarda la inicialización en disco. |
| Condición de termino Fallida: | El conjunto de datos seleccionado para inicializar con cumple con los factores de forma especificados en la inicialización. |
| Prioridad: | Alta. Debe implementarse obligatoriamente. |

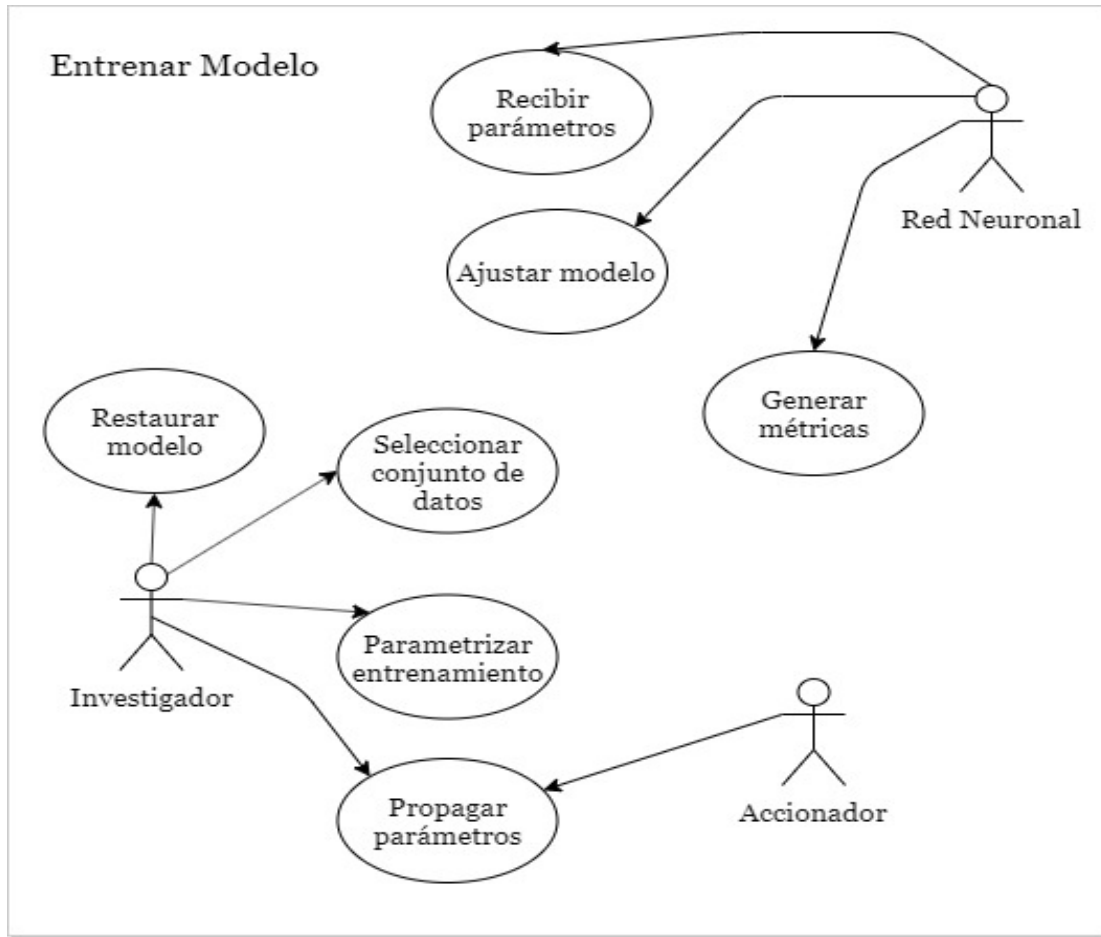


Figura 4.5: Caso de uso; Entrenar Modelo

Cuadro 11

Definición de caso de uso: Entrenar modelo

| Definición del Caso de Uso | |
|-------------------------------|---------------------------|
| Nombre de Caso de Uso: | Entrenar Modelo |
| Actor Principal: | Red Neuronal |
| Actores Secundarios: | Investigador, Accionador. |

| | |
|--------------------------------------|---|
| Objetivo en contexto: | Entrenar el modelo convolucional creado, a partir de la información contenida en los conjuntos de datos de entrenamiento. |
| Precondiciones: | Debe seleccionarse el modelo a entrenar, el conjunto de datos con la forma correcta (definida en el modelo guardado) y parametrizar el entrenamiento. |
| Disparador: | El investigador decide entrenar un modelo de red neuronal convolucional con un conjunto de datos, con forma de entrada igual a la definida en el modelo. |
| Condición de Término: | Se completa el entrenamiento según los parámetros especificados y se guarda la información del modelo (como los pesos, bias, metadata o checkpoints) para poder restaurarlos y reentrenar, en caso de que se requiera. |
| Condición de termino Fallida: | El conjunto de datos seleccionado para el entrenamiento no cumple con los factores de forma especificados en la inicialización. O, los parámetros de entrenamiento no son los adecuados para el equipo de cómputo, lo que produciría un consumo anormal de los recursos del sistema y detendría completamente la ejecución del mismo. |
| Prioridad: | Alta. Debe implementarse obligatoriamente. |

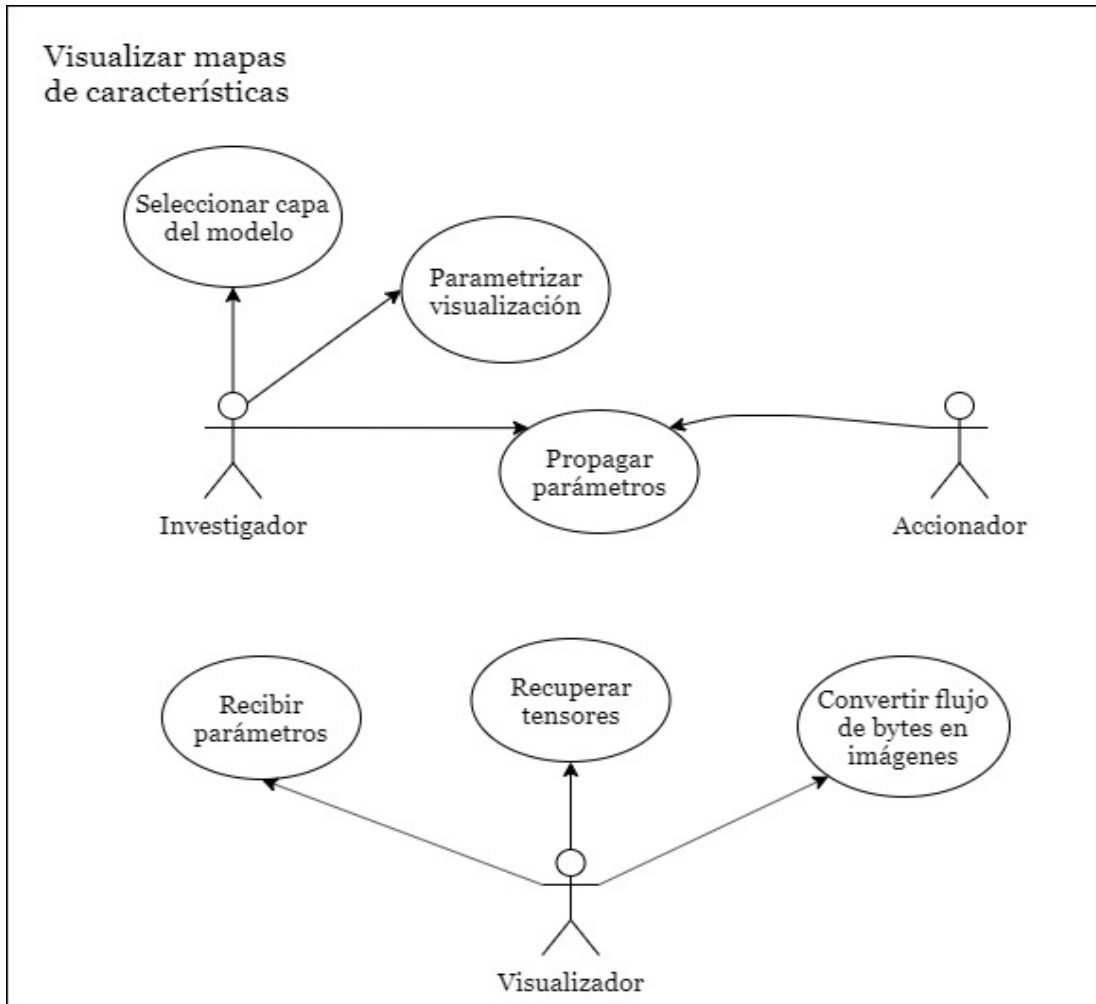


Figura 4.6: Caso de uso; Visualizar mapas de características

Cuadro 12

Definición de caso de uso: Visualizar mapas de características

| Definición del Caso de Uso | |
|-------------------------------|-------------------------------------|
| Nombre de Caso de Uso: | Visualizar mapas de características |
| Actor Principal: | Visualizador |

| | |
|--------------------------------------|---|
| Actores Secundarios: | Investigador, Accionador. |
| Objetivo en contexto: | Visualizar los procesos intermedios del entrenamiento de la red neuronal convolucional. En estos procesos se obtienen los mapas de características o patrones que usa la red neuronal para detectar en los procesos de predicción. |
| Precondiciones: | Debe haberse entrenado y guardado un modelo, luego de haberse seleccionado, debe parametrizarse el proceso de visualización. |
| Disparador: | El investigador decide visualizar la efectividad de los parámetros de entrenamiento, mediante la visualización de las características extraídas por la red neuronal convolucional. Para esto, debe seleccionar el conjunto de datos del cual se desee visualizar. |
| Condición de Término: | Se obtienen los mapas de características según parámetros y se grafican. |
| Condición de termino Fallida: | Que los parámetros de visualización excedan los límites establecidos según la forma del conjunto de datos. Que no se seleccionen un conjunto de datos, o que la forma del conjunto de datos no sea igual a la establecida en los parámetros del modelo convolucional. |
| Prioridad: | Alta. Debe implementarse obligatoriamente. |

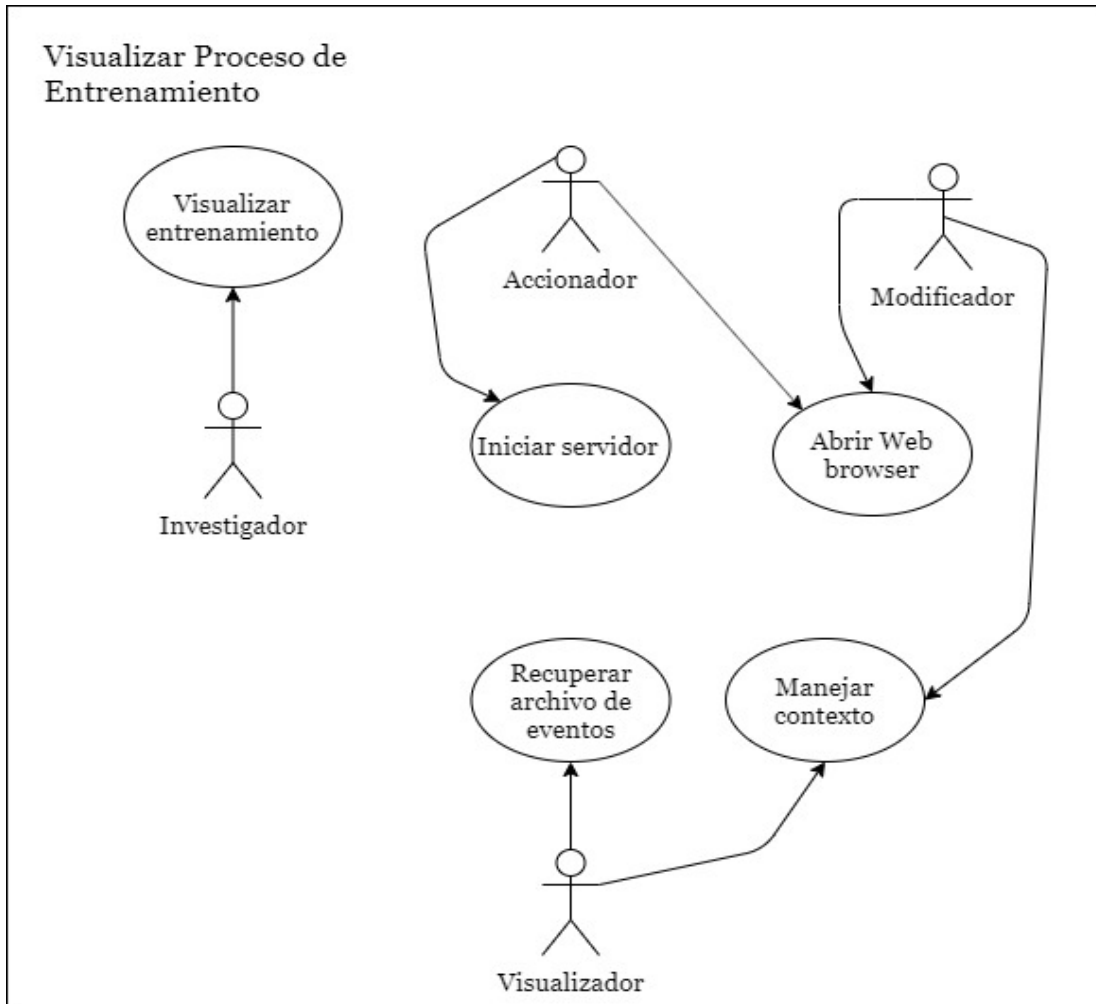


Figura 4.7: Caso de uso; Visualizar Proceso de Entrenamiento

Cuadro 13

Definición de caso de uso: Visualizar proceso de entrenamiento

| Definición del Caso de Uso | |
|-------------------------------|-------------------------------------|
| Nombre de Caso de Uso: | Visualizar proceso de entrenamiento |
| Actor Principal: | Visualizador |

| | |
|--------------------------------------|---|
| Actores Secundarios: | Investigador, Modificador, Accionador. |
| Objetivo en contexto: | Visualizar las métricas resultantes de los procesos matemáticos y estadísticos que definen el aprendizaje y precisión de los parámetros del modelo convolucional. |
| Precondiciones: | Debe haberse entrenado y guardado un modelo. Esto guarda las métricas asociadas en un archivo de eventos. Este archivo deberá estar guardado en un directorio específico, definido como el directorio de registros (/logs). |
| Disparador: | El investigador decide visualizar las métricas de entrenamiento, asociadas al modelo convolucional. |
| Condición de Término: | Se cierra la ventana del navegador web asociada a la sesión iniciada por el modificador. |
| Condición de termino Fallida: | Que el servidor no se inicie correctamente debido a que la ruta que se proporcionó como directorio de registros no contenga archivos de evento. O que la base de datos no tenga metadatos de evento asociado, o que no se encuentre en la ruta un archivo de datos (sqlite3) que contenga eventos. |
| Prioridad: | Alta. Debe implementarse obligatoriamente. |

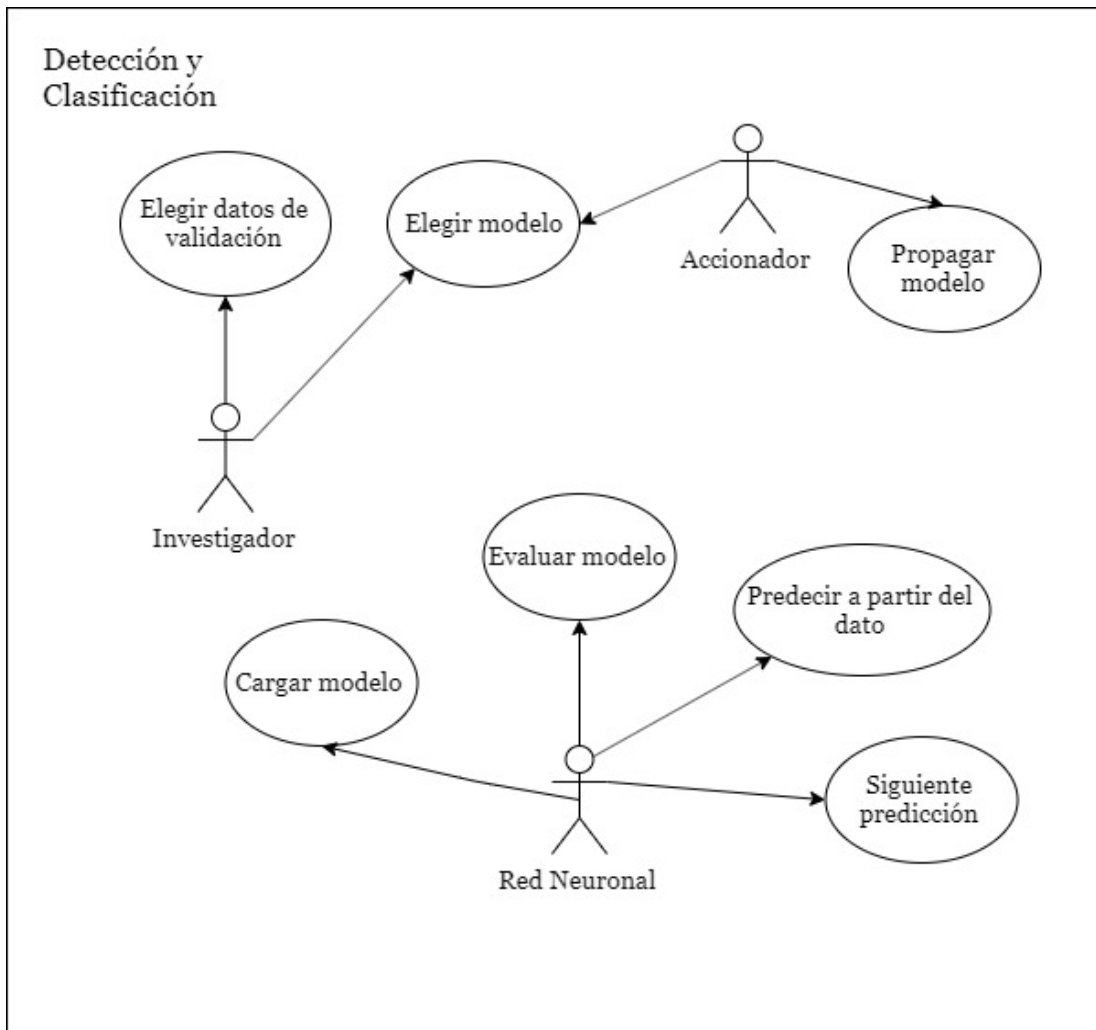


Figura 4.8: Caso de uso; Clasificación y Detección

Cuadro 14

Definición de caso de uso: Detección y Clasificación

| Definición del Caso de Uso | |
|-------------------------------|---------------------------|
| Nombre de Caso de Uso: | Detección y Clasificación |
| Actor Principal: | Red Neuronal |
| Actores Secundarios: | Investigador, Accionador. |

| | |
|---|---|
| <p>Objetivo en contexto:</p> | <p>Predecir clases de cuerpos, fenómenos y objetos astronómico – espaciales, mediante un modelo de red neuronal convolucional entrenado en función de los parámetros establecidos por el investigador en la fase de creación y entrenamiento.</p> |
| <p>Precondiciones:</p> | <p>Debe existir un modelo convolucional entrenado antes de poder predecir la clase de un dato o conjunto de datos. Este modelo se podrá cargar desde el directorio en que se haya guardado luego de entrenar el modelo.</p> |
| <p>Disparador:</p> | <p>El investigador decide probar la efectividad de los parámetros de entrenamiento al evaluar la precisión de la predicción del modelo de red neuronal convolucional al evaluar la clase de una o varias imágenes sin necesidad de que hayan sido procesadas.</p> |
| <p>Condición de Término:</p> | <p>Se predice la clase de la imagen o conjunto de imágenes. Si es un conjunto de imágenes puede cancelarse la operación en medio de la operación para que se detenga la evaluación de los elementos.</p> |
| <p>Condición de termino Fallida:</p> | <p>No se encuentra el modelo a cargar en la ruta especificada. Los datos de validación no son los indicados para el modelo cargado.</p> |
| <p>Prioridad:</p> | <p>Alta. Debe implementarse obligatoriamente.</p> |

4.2.4.1.3. Diagrama de Clases

Los diagramas de clases permiten modelar las clases de un sistema mediante una visión estática que muestra los atributos, operaciones y relaciones que ejecutan las mismas.

Los elementos principales de un diagrama de clase son cajas, que son los íconos utilizados para representar clases e interfaces. Cada caja se divide en partes horizontales. La parte superior contiene el nombre de la clase. La sección media menciona sus atributos. Mientras, la sección baja contiene los comportamientos o métodos de la clase.

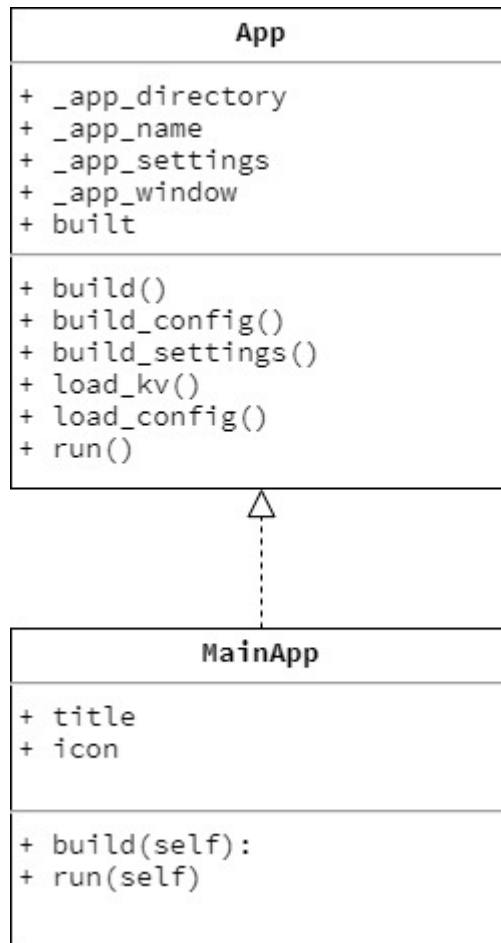


Figura 4.9: Diagrama de clases; MainApp

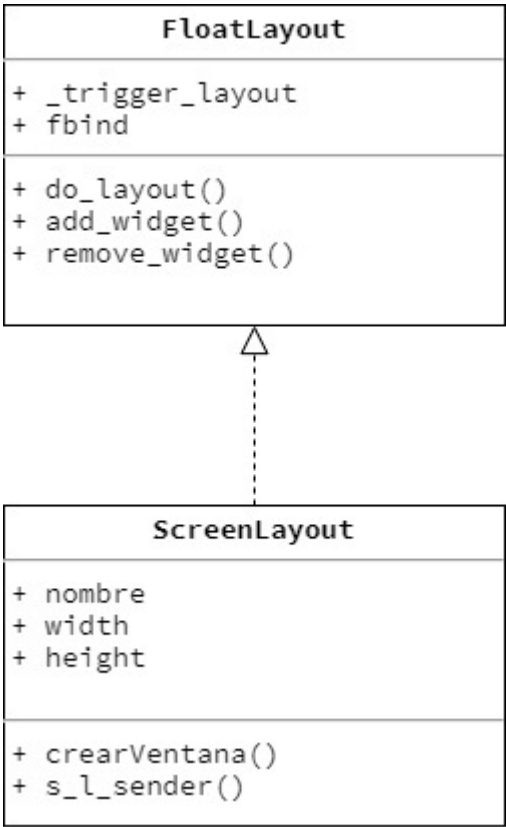


Figura 4.10: Diagrama de clases; ScreenLayout

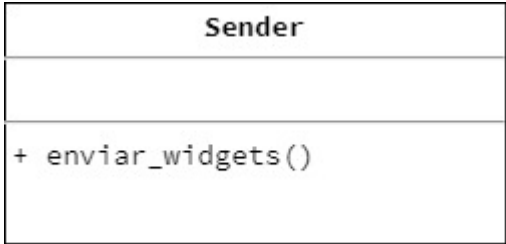


Figura 4.11: Diagrama de clases; Sender

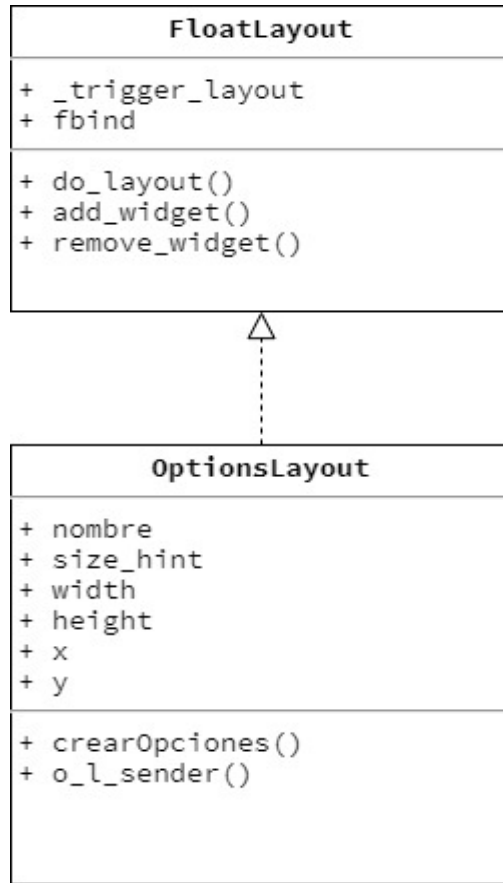


Figura 4.12: Diagrama de clases; OptionsLayout

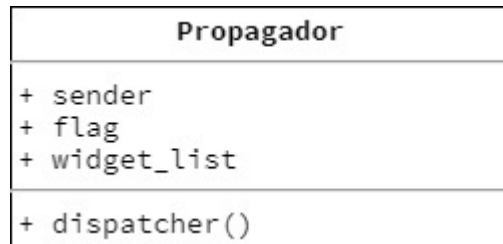


Figura 4.13: Diagrama de clases; Propagador

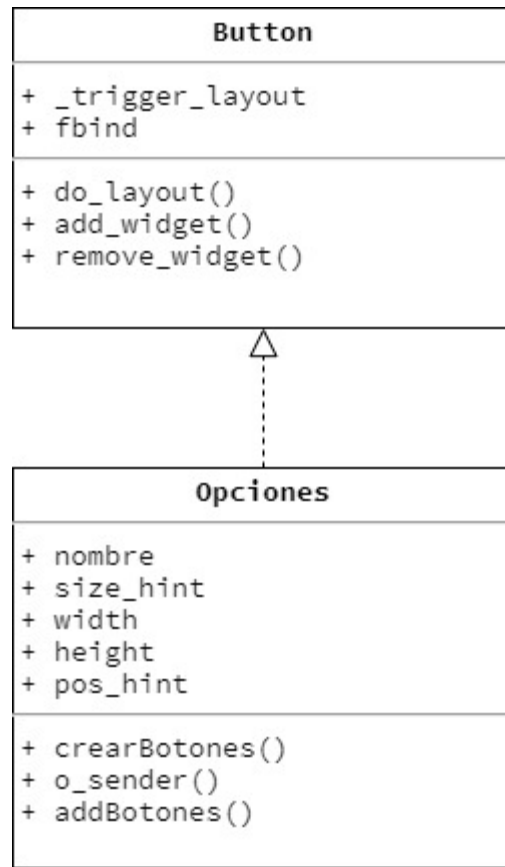


Figura 4.14: Diagrama de clases; Opciones

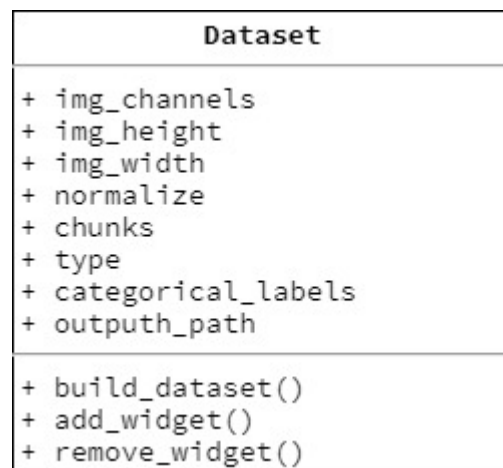


Figura 4.15: Diagrama de clases; Dataset

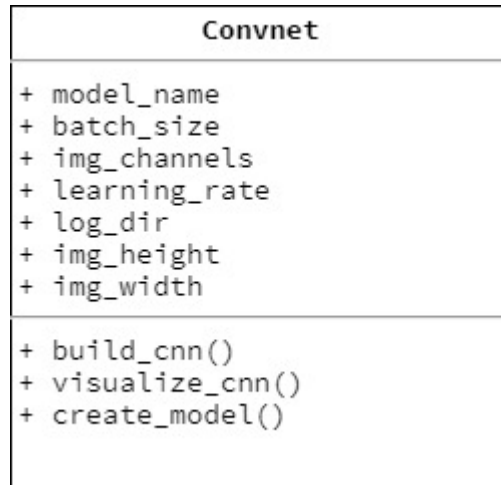


Figura 4.16: Diagrama de clases; Convnet

4.2.4.1.4. Diseño de la Interfaz

Pressman (2010, p. 270) establece que el proceso general de análisis y diseño de la interfaz de usuario comienza con la creación de diferentes modelos del funcionamiento del sistema (según se percibe desde fuera). Se empieza delineando las tareas orientadas al usuario —y a la computadora— que se requieren a fin de obtener el funcionamiento del sistema.



Figura 4.17: Wireframe de la ventana principal



Figura 4.18: Wireframe del sub menú de Entrenamiento



Figura 4.19: Wireframe del sub menú Conjunto de datos



Figura 4.20: Wireframe del sub menú Modelos



Figura 4.21: Wireframe del sub menú Visualización



Figura 4.22: Wireframe del sub menú de Clasificación

4.2.4.1.5. Diseño de la Red Neuronal Convolucional

El diseño de la red neuronal convolucional define la arquitectura en general, especialmente en número de capas, las operaciones que realizan cada una de ellas y los parámetros o resultados que producen las transformaciones entre capas.

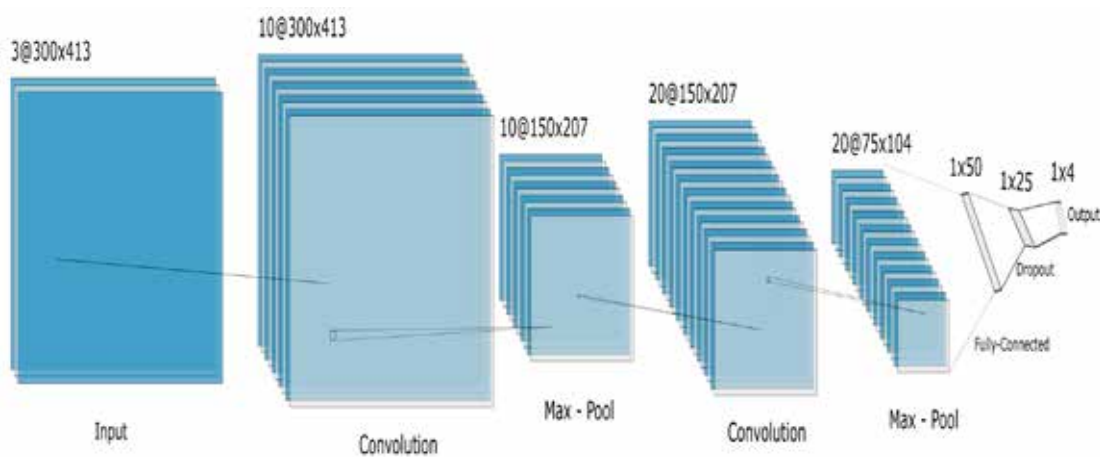


Figura 4.23: Diseño de la red neuronal convolucional estilo LeNet

Creada desde: LeNail, A. (2019) [alexlenail.me/NN-SVG]

4.2.4.1.6. Esquema de la base de datos

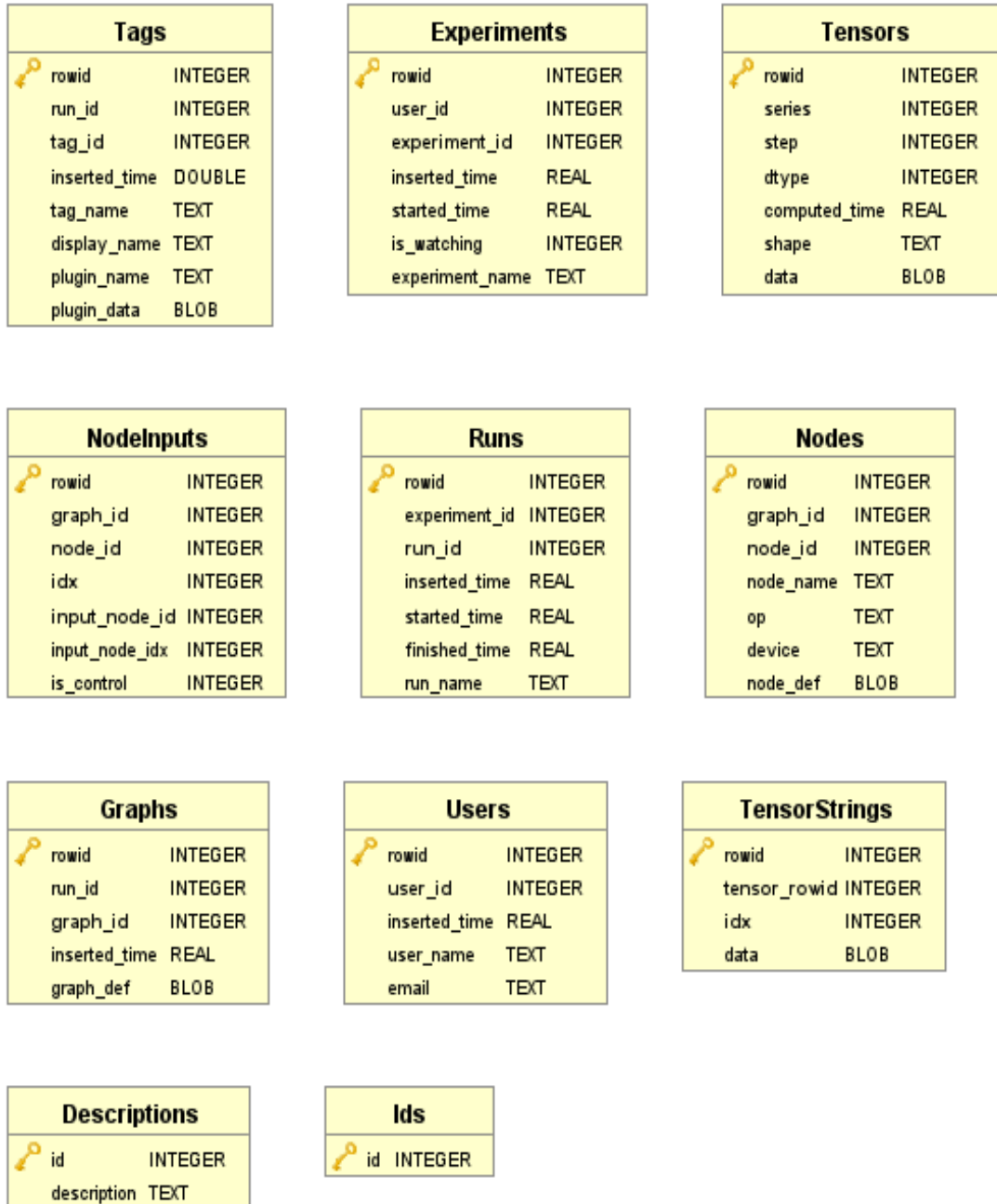


Figura 4.24: Esquema de la base de datos para archivos de evento y log.

4.3. Fase 3:

Esta fase desarrolla la mitad correspondiente a la construcción de la iteración, tomando como referencia al desglose hecho en la fase anterior. Similarmente, expande la sub fase de implementación, siendo esta la última del ciclo de vida del proyecto. Por tanto, esto no es más que la fase de desarrollo y de las pruebas unitarias de la aplicación.

4.3.1. Recursos

Los recursos necesarios para la fase de desarrollo comprenden varios tipos esenciales, estos es:

- Humanos
- Institucionales
- Tecnológicos

4.3.1.1. Humanos

Tienen que ver con todos los sujetos que forman parte del proceso, incluyendo los que forman parte de las unidades de observación, los investigadores, los asistentes y los colaboradores.

Así bien, las personas encargadas de prestar ayuda, guía y colaboración para la exitosa elaboración de este proyecto son:

- 1) Tutor Académico: Dra. Belkys Araujo.
- 2) Tutor Metodológico: Ing. Alicia De Pizzella
- 3) Encargados de las observaciones Astronómicas / Encuestados
- 4) Investigador: Adquinson Farías.

4.3.1.2. Institucionales

Estos recursos denotan a aquellos organismos o instituciones a los cuales se recurre para obtener la información necesaria para el desarrollo del proyecto u aplicación. Así pues, las instituciones u organismos que brindaron información son los siguientes:

- a. The Sloan Digital Sky Survey

- b. Universidad José Antonio Páez
- c. Galaxy Zoo
- d. NASA / IPAC
- e. ESA / Hubble
- f. JPL (Jet Propulsión Laboratory)
- g. HST
- h. Google Developers
- i. Mathworks
- j. NVIDIA Developer

4.3.1.3. Tecnológicos

Los recursos tecnológicos comprenden objetos de hardware como el computador, o recursos de software tales como librerías, binarios, frameworks, API's, entre otros.

a. Hardware

- Computadora tipo Laptop.

a. Software

- Python: 3.6+
- Kivy: 1.11.1
- Tensorflow: 1.14.0
- Tensorboard: 1.14.0
- TFLearn: 0.3.2
- Keras: 2.2.5
- OpenCV: 4.1.1.26
- Numpy: 1.16.4
- Matplotlib: 3.1.1

4.3.2. Construcción de la Iteración

En este apartado se describe el proceso de desarrollo y codificación de la aplicación, además de las tecnologías necesarias o utilizadas para alcanzar dicho estado de integración, entre las funciones de bajo nivel, y aquellas de alto nivel.

4.3.2.1. Lenguaje de Programación

Tal y como se establece en los recursos tecnológicos, el lenguaje de programación utilizado para desarrollar la aplicación es Python, principalmente por su gran poder para el procesamiento de datos, abundantes librerías de aprendizaje de máquina y compatibilidad entre diferentes sistemas operativos.

4.3.2.2. Paradigma de Programación

El paradigma orientado a objetos contempla un conjunto de características y elementos de abstracción necesarios para llevar a cabo la implementación de la aplicación. Adicionalmente, se aprovecha la propiedad de modularidad que permite subdividir la aplicación en partes pequeñas, de forma que cada una sea tan independiente como sea posible de la aplicación en sí, y de las demás partes.

4.3.2.3. Interfaz de Usuario

La codificación de la interfaz de usuario se fundamenta en la librería Kivy, que se encarga de procesar y generar los recursos de bajo nivel que son renderizados y, posteriormente utilizados por el componente o módulo modificador.

```
import kivy
#kivy.require('1.11.1') # replace with your current kivy version !
import os.path
from kivy.app import App
from kivy.core.window import Window
from mainframe import iniciar, set_img
from kivy.modules import inspector

#Referencia de la aplicación principal que hereda de kivy.app -> App
class MainApp(App):

    def build(self):
        self.title = ("AVAI")
        self.icon = set_img()
        Window.maximize()

        mainContainer = iniciar()
        inspector.create_inspector(Window, mainContainer) #python main.py -m inspector

        return mainContainer

if __name__ == '__main__':
    MainApp().run()
```

Figura 4.25: Inicializador de la aplicación

```

class ScreenLayout(FloatLayout):
    """
    Clase extensión del FloatLayout para contener los elementos principales de la aplicación
    """

    def __init__(self, nombre = "MainLayout", **kwargs):
        self.nombre = nombre
        self.width = 1366
        self.height = 765
        super(ScreenLayout, self).__init__(**kwargs)

    def __str__(self):
        return ("ObjName: "+self.nombre)

    def crearVentana(self):
        """
        Crea la ventana con los componentes añadidos a la raíz de widgets de la aplicación
        """

        screen_layout_obj = [ ]

        mainContainer = ScreenLayout()
        resultsContainer = ScreenLayout("resultsLayout", width = 1064, height = 649, size_hint = [None, None],
        topContainer = ScreenLayout("topLayout", width = 1366, height = 32, size_hint = [None, None],
        lowerContainer = ScreenLayout("lowerLayout", width = 1366, height = 22, size_hint = [None, No

```

Figura 4.26: Creación básica de un componente global

4.3.2.4. Red Neuronal Convolutiva

En ese apartado se define una clase que agrupa todo el comportamiento necesario para crear, guardar, entrenar y cargar el modelo convolutivo, definido según los parámetros introducidos mediante la interfaz gráfica. La mayoría de los parámetros serán llevados a la clase mediante el componente Accionador. Por tanto, los atributos principales para esquematizar el modelo son el alto, el ancho y los canales que tendrá la imagen de entrada.

Así pues, adicionalmente al método de construir, la clase Convnet posee dos métodos adicionales:

- a) Visualización: El método de visualización se encarga de llamar a los procesos de bajo nivel, encargados de obtener los mapas de características y procesar las imágenes.
- b) Crear modelo: Este método obtiene los parámetros y crea la definición del modelo convolutivo, lo que permite restaurarlo para entrenarlo después o, entrenarlo inmediatamente en función de esos parámetros.

```

class Convnet(object):

    def __init__(self, input_height, input_width, input_channels):
        self.img_height = input_height
        self.img_width = input_width
        self.img_channels = input_channels

    def build_cnn(self):
        # Build neural network and train

        # En la capa de entrada primero va el alto, y luego el ancho
        # Esto es pues, porque una vez que el archivo se lee las
        # dimensiones se transponen

        # Img H, Img W, Input_layer/X:8
        convnet = input_data( shape=(None, self.img_height, self.img_width, self.img_channels),
                             name = "input_layer")

        # Layer 1, c1/Relu:8

        convnet = conv_2d( incoming = convnet,
                           nb_filter = 10,
                           filter_size = [10, 10],
                           strides = [1, 1, 1, 1],
                           activation="relu",
                           name = "c1")

```

Figura 4.27: Clase Convnet

4.3.2.5. Dataset

En esta clase se encuentra el funcionamiento para crear un conjunto de datos.

```

class Dataset(object):

    def __init__(self, input_height, input_width, input_channels):
        self.img_height = input_height
        self.img_width = input_width
        self.img_channels = input_channels

        # CREACION DEL DATASET.H5
    def build_dataset(self, img_data_dir, img_shape, dt_save_route, mode="folder", categ_labels=True, norma

        # Resultado_en_Bytes = (NUM_IMGS * IMG_WIDTH * IMG_HEIGHT * CHANNELS * CLASSES)

        build_hdf5_image_dataset(img_data_dir, image_shape=(img_shape), mode=mode,
                                output_path=(dt_save_route), categorical_labels=categ_labels, normalize=normalize, chunks=chunks)

        # Para crear el dataset va primero el ancho, luego el alto
        # Aunque la documentación dice lo contrario, una vez que el archivo se lee
        # las dimensiones se transponen

```

Figura 4.28: Clase Dataset

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Una aplicación suele simplificar y hacer más eficiente cualquier tipo de proceso o tarea que se lleve a cabo de forma manual, sin embargo, estas aplicaciones suelen estar programadas de forma explícita, tal que se ejerza un tipo de comportamiento pre definido y contemplado en la estructura de la aplicación.

En contraparte, los modelos de aprendizaje de máquina son programados para “aprender a aprender”, es decir, son programados para ejecutar uno o varios tipos de tareas, sin ningún tipo de explicitud sobre la tarea que van a desempeñar, pero, esto no significa que sean completamente autónomos y que puedan aprender de cualquier tipo de tareas. Para esto, existen diferentes modelos que se ajustan y se adaptan mejor dependiendo del tipo de tarea que vayan a ejecutar.

Desde luego, tal y como demuestra el desarrollo de este proyecto, es posible realizar el proceso de integración entre una aplicación con comportamientos dirigidos y basada en la experiencia de usuario, para que se ajuste a la flexibilidad que ofrecen los modelos de aprendizaje de máquina. Esto, claramente, favorece el proceso de creación de una red neuronal convolucional que pudiera ser utilizada para la detección y clasificación de cuerpos, fenómenos y objetos de carácter astronómico.

Así pues, de forma inicial se vislumbran ciertos beneficios. Primero, la minimización del tiempo necesario para crear los conjuntos de datos, definir la estructura de la red neuronal convolucional e implementar la misma. Similarmente, la dificultad del proceso de implementación del entrenamiento y el de visualización de características disminuye gracias a la abstracción que provee la interfaz de alto nivel, respecto de los procesos de bajo nivel necesarios para estas tareas.

A modo de referencia, y según los resultados obtenidos, el proceso de detección puede dividirse en un conjunto de pasos estructurados completamente necesarios que,

deben cumplirse antes de poder develar el verdadero potencial de detección y clasificación del modelo creado.

Primero, la creación o carga del conjunto de datos que representa la información de la cual se quiere aprender, o predecir. Segundo, la creación o ajuste del modelo en función de los parámetros de entrada, esto es pues, el conjunto de datos. Finalmente, el proceso de entrenamiento y su componente opcional, el proceso de visualización, que ayuda a entender si el modelo está aprendiendo lo que debería aprender, sin necesidad de predecir o detectar sobre un dato nunca antes visto por el modelo, justo después de haber culminado el proceso de entrenamiento.

De forma similar, la interfaz de alto nivel previó de fases de implementación, cada una con un conjunto de pasos únicos que influyeron en el proceso final de integración entre los componentes de bajo y alto nivel. Este proceso de integración, permitió distinguir ciertos aspectos de cada una de las fases desarrolladas. En este sentido, son tres los factores que permiten arrojar conclusiones.

En primer lugar, se distingue que las técnicas de recolección de datos son un método excelente para recabar requerimientos de una población de usuarios distante y dispersa, sin embargo, esto es especialmente cierto cuando se aplican en línea o en la web. Es este sentido, se evidenció que la experiencia de usuario puede ser un factor muy subjetivo, pues, la mayoría de las veces un factor de requerimiento podía satisfacer las demandas de algunos usuarios, descuidando la de otros en el proceso.

Además, los factores de funcionalidad como los requerimientos, deben poseer al menos una ruta pre definida que ayude al futuro usuario a orientarse respecto de que es lo que hará la aplicación y cuál es la funcionalidad prevista. Es decir, se debe tener una pequeña lista de requerimientos, y que sean los usuarios quienes decidan cuáles de ellos deberían ser implementados según un orden de prioridad, al desarrollo de la aplicación. Esto resultaría en que los requerimientos principales se muestren en el primer prototipo previsto en la iteración del modelo funcional, de forma que este mismo prototipo ayude de forma simultánea a recabar nuevos requerimientos mientras, se someten a pruebas

de usuario los requerimientos ya implementados en el prototipo.

De modo similar, el apartado más resaltante de la fase de diseño de componentes es la integración a nivel conceptual que proveen las metodologías de trabajo. Es decir, que el diseño es un paso fundamental para poder desarrollar y construir aplicaciones que cumplan de forma satisfactoria con la funcionalidad prevista, pero, sin descuidar el cómo se proyecta un usuario de cara al manejo de la aplicación. Por lo cual, es importante dar indicadores (títulos en las ventanas, iconos gráficos, código de colores consistente, etc.) que permitan al usuario conocer el contexto del trabajo en curso.

Además, es importante conocer que cuando se habla de integración conceptual, se refiere al cómo, mediante el modelado y diseño se facilita el proceso de “modularización”. Esto, resulta particularmente útil cuando se desarrolla una aplicación pensando en módulos o componentes que puedan utilizarse mediante el paso y recepción de mensajes y llamadas entre módulos, esto es pues, el desarrollo de una aplicación orientada a objetos, con el componente de modularidad.

Finalmente, el desarrollo de la aplicación presenta dos aspectos interesantes. En primer lugar, se evidencia que el desarrollo de componentes de software es un proceso iterativo que está dispuesto a contener errores, independientemente de si el componente en sí es una aplicación, o una red neuronal artificial. Debido a esto, se sienta el argumento que indica qué, las pruebas son necesarias y deberían ser totalmente obligatorias en cada fase de desarrollo de los componentes individuales, así como también en el proceso de integración entre los componentes desarrollados por separado.

Por último, la relación entre las pruebas y los requerimientos suelen estar enfocadas a los requerimientos del tipo funcional, lo cual no está mal. Sin embargo, cuando se considera un aspecto no funcional como la usabilidad, que a su vez forma parte de la experiencia de usuario, es posible denotar la falta de métodos contenidos en las metodologías de desarrollo que permitan hacer pruebas más extensivas, más recurrentes y mejor documentadas a los apartados que demuestran el nivel de la experiencia alcanzado por un requerimiento funcional, o no funcional en específico.

No obstante, es posible que esto se produzca principalmente porque, probar y mejorar los apartados que se consideran como “experiencia de usuario” es más difícil debido a la subjetividad implícita que lleva este apartado, ahora, si se le suma la dificultad que provee el re factorizar o mejorar una aplicación de escritorio, el proceso de pruebas, mejoramiento e implementación podría terminar siendo una espiral sin fin, que generaría grandes cargas de trabajo de forma innecesaria, y con ello, pérdidas de tiempo y dinero.

5.2. Recomendaciones

En relación, con el recabado de requerimientos, es recomendable tener en cuenta que, aunque posible, segmentar la población en una encuesta en línea es menos beneficioso que hacerlo en una aplicada cara a cara, esencialmente porque se pierde una gran cantidad de información que puede ayudar a obtener un mejor panorama de lo que se quiere hacer. En cambio, después de haber visto los resultados, se sugiere es que se divida la encuesta en tópicos, donde cada uno podría por ejemplo, corresponder con una funcionalidad o un proceso en específico.

En lo que concierne al diseño de los componentes, es recomendable escuchar a los usuarios y diseñar para darles el control dentro de la aplicación, de acuerdo a los requerimientos, sin embargo, también es necesario que se establezca un flujo consistente en cada una de las operaciones o funciones que realice la aplicación, a fin de evitar comportamientos indeseados, y minimizar la susceptibilidad a errores en los procesos de comunicación entre la interfaz de alto nivel, y los componentes que realizan las tareas de bajo nivel.

Finalmente, en el caso del desarrollo de la aplicación, se recomienda que la implementación de bajo nivel (procesar las imágenes, tipos de datos y la red neuronal) se haga completamente bajo un mismo esquema sin el uso de interfaces (Keras, TFlearn) para optimizar y hacer más eficiente el manejo de los datos. Además, se puede mejorar la extensibilidad al crear e implementar nuevos modelos de aprendizaje.

REFERENCIAS

- Arias, F. (2012), **El proyecto de investigación. Introducción a la metodología científica.** (6°. ed.) Caracas – Venezuela: Episteme.
- Balestrini, J. (2006), **Cómo se elabora el proyecto de Investigación.** (7°. Ed.) Caracas: BL Consultores Asociados.
- Borglin, J. (2011), “*Methodology of research in bio-medical engineering*”. Trabajo de grado de Maestría en Ingeniería Bio-Médica, *Chalmers University of Technology*, Gothenburg, Suecia.
- Borovkov, A. (2017), “*Methodology of research in physics*”. Trabajo de grado de Maestría de Ciencias en Física, Universidad de Hamburgo, Alemania.
- Brice, Michael J. (2019), “*Methodology of research in philosophy*”. *All Master's Theses*. 1207. Disponible en: <https://digitalcommons.cwu.edu/etd/1207>
- Castro, M. (2003). **El proyecto de investigación y su esquema de elaboración.** (2ª. Ed.). Caracas: Uyapal.
- De Barrera, J. (2006), **Metodología de la investigación. Guía para la comprensión holística de la ciencia.** (4°. ed.) México: Quirón.
- Gal, Y. (2016), “*Methodology of research in philosophy*”. Trabajo de grado de Doctorado en Filosofía, Universidad de Cambridge, Inglaterra.
- González, R.E., Muñoz, R.P., y Hernández, C.A., (2018), “*Methodology of research in philosophy*”.
- Hernández, R., Fernández, C. y Baptista, L. (2010), **Metodología de la Investigación.** (6°. Ed.) México: McGraw-Hill
- Hocking, A. (2018), “*Methodology of research in philosophy*”. Trabajo de grado de Doctorado en Filosofía, Universidad de Hertfordshire, Inglaterra.
- Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P. (2017), “*Methodology of research in philosophy*”. Published as a conference paper at ICLR 2017. arXiv:1608.08710v3 [cs.CV] 10 Mar 2017.

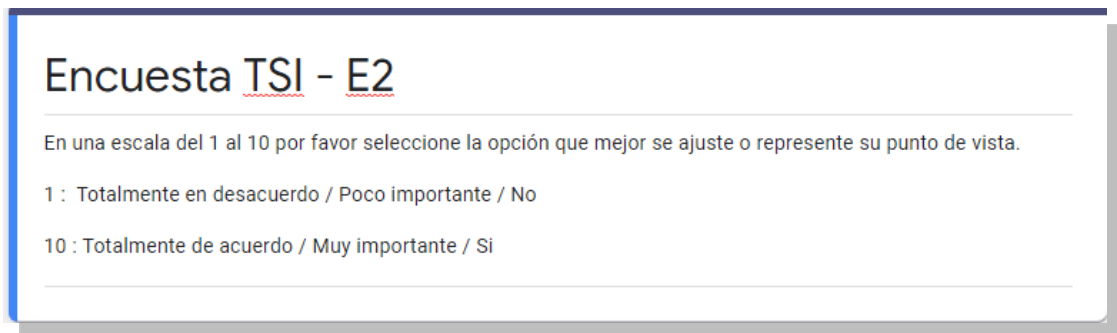
- Pressman, R.S. (2010), **Ingeniería del software. Un enfoque práctico** (7^a. Ed.). México: McGraw-Hill
- Romanishin, W. (2006), “
”. Universidad de Oklahoma, USA.
- Rötner, S. (2017), “
”. Trabajo de grado de Maestría, Universidad Técnica de Dortmund, Alemania.
- Sabino, C. (2002), **El Proceso de Investigación**. (5^o. ed.) Caracas: Panapo, Bogotá: Panamericana, Buenos Aires: Lumen.
- Shi, H. (2016), “
”. Trabajo de grado de Maestría de Ciencias en Ingeniería Eléctrica y Computación, Universidad de Illinois (*at Urbana-Champaign*), USA.
- Sommerville, I. (2011), **Ingeniería de Software**. (9^o. ed.). México: Pearson Educación de México, S.A. de C.V
- Tamayo y Tamayo, M. (2003), **El proceso de la investigación científica**. (4^o. ed.) México: Editorial Limusa.
- Universidad Pedagógica Experimental Libertador (2006). **Manual de trabajo de grado de especialización y maestría y tesis doctorales**. Caracas: FEDUPEL
- Van Amerongen, P. (2018), “
”. Trabajo de grado, Universidad de Groninga, Países Bajos.
- Wang, F. (2017), “
”. Trabajo de grado de Doctorado en Filosofía, *The Australian National University*, Australia.

ANEXO A

Encuesta

Este anexo muestra los resultados obtenidos de las encuestas junto a la imagen de referencia usada (si aplica) para ilustrar la pregunta.

a) Encabezado de la encuesta



Encuesta TSI - E2

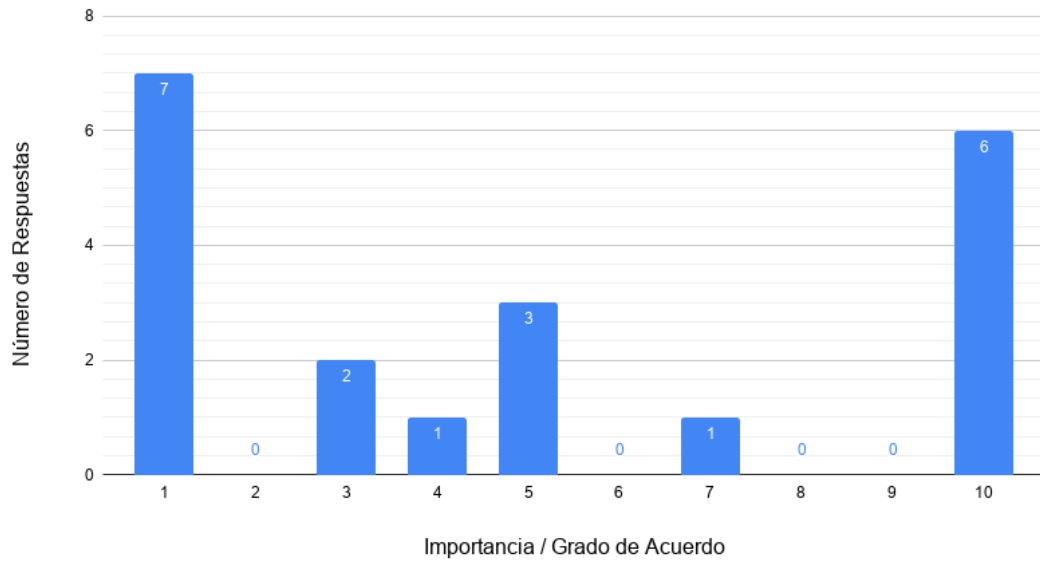
En una escala del 1 al 10 por favor seleccione la opción que mejor se ajuste o represente su punto de vista.

1 : Totalmente en desacuerdo / Poco importante / No

10 : Totalmente de acuerdo / Muy importante / Si

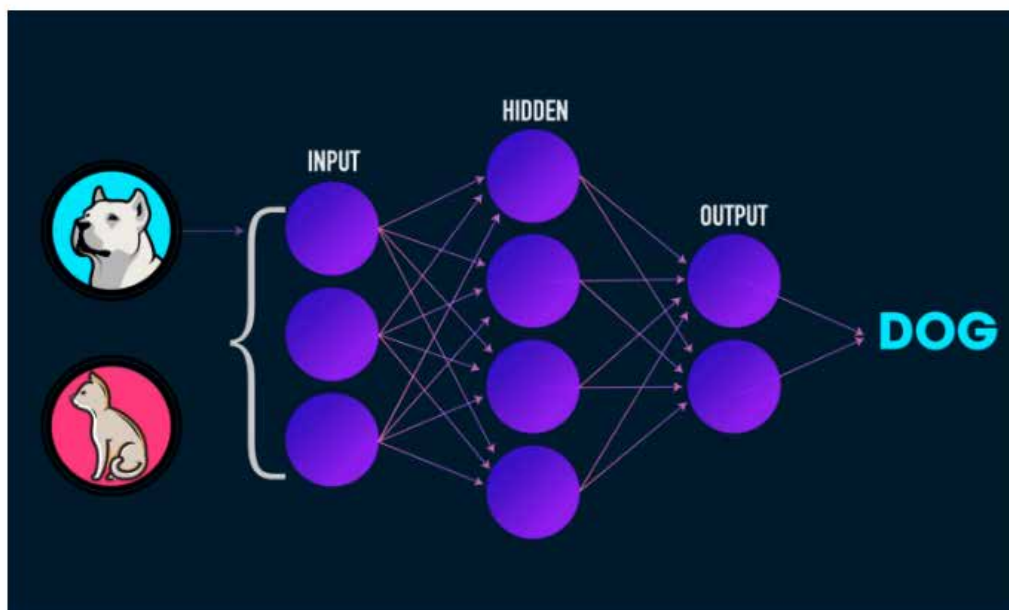
b) Primera pregunta

P-1: ¿Has escuchado hablar de las redes neuronales artificiales?



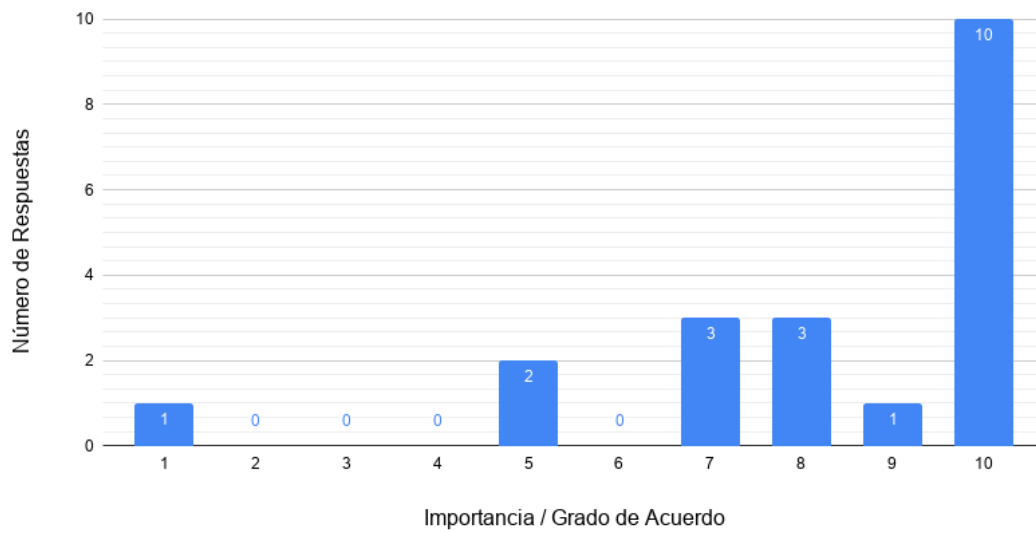
c) Imagen ilustrativa de la primera pregunta

Ejemplo: Red neuronal artificial



d) Segunda pregunta

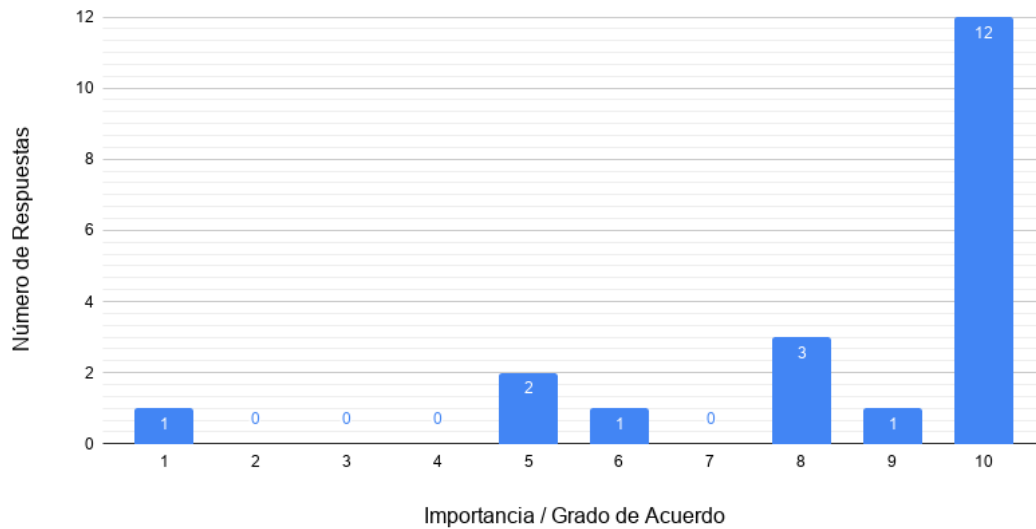
P-2: De haber oído de ellas, o no, ¿Consideras que una aplicación que facilite la creación de una RNA para la clasificación de imágenes pueda ser útil?



Nota: Esta pregunta no tuvo imagen referencial.

e) Tercera pregunta

P-3: De estar de acuerdo ¿Debería contar esta aplicación con un esquema de color cómodo, que facilite el uso de la aplicación durante largas horas de uso?



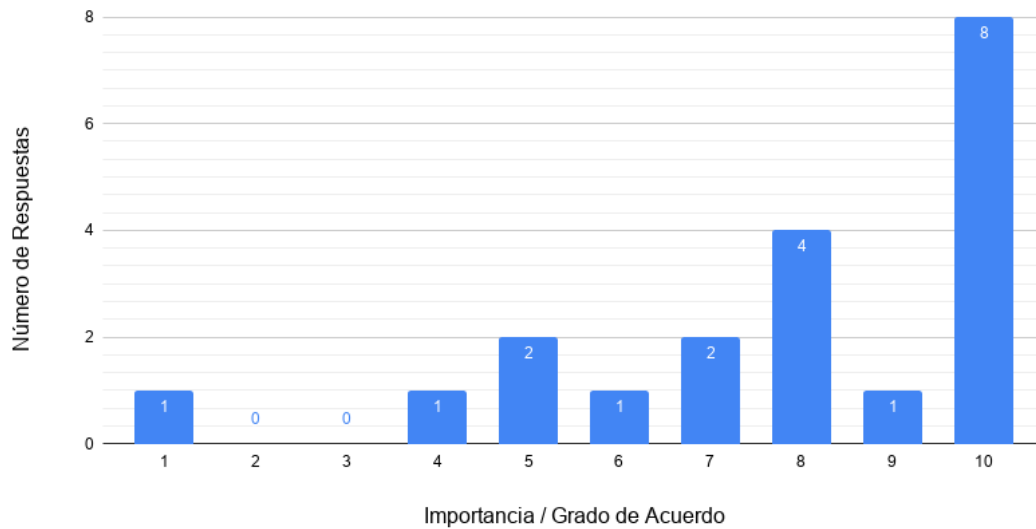
f) Imagen ilustrativa de la tercera pregunta

Ejemplo: Esquema de colores (Darkmode)



g) Cuarta pregunta

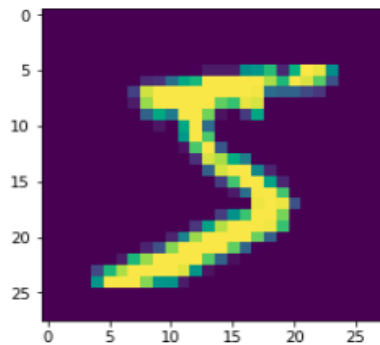
P-4: ¿Consideras que es necesario el visualizar o graficar los datos numéricos de bajo nivel que maneja la aplicación (backend)?



h) Imagen ilustrativa de la cuarta pregunta

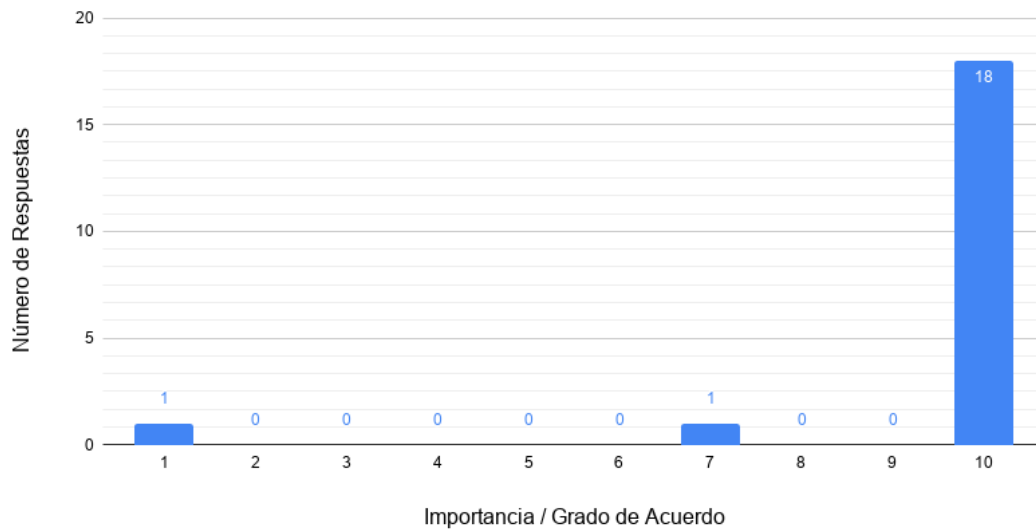
Ejemplo: Gráfico de un elemento del dataset

```
(60000, 28, 28, 1) (60000,)  
(10000, 28, 28, 1) (10000,)  
In [7]: import matplotlib.pyplot as plt  
plt.imshow(x_train[0])  
plt.show()
```



i) Quinta pregunta

P-5: ¿Es importante la compatibilidad de una aplicación entre diferentes sistemas operativos (OS)?



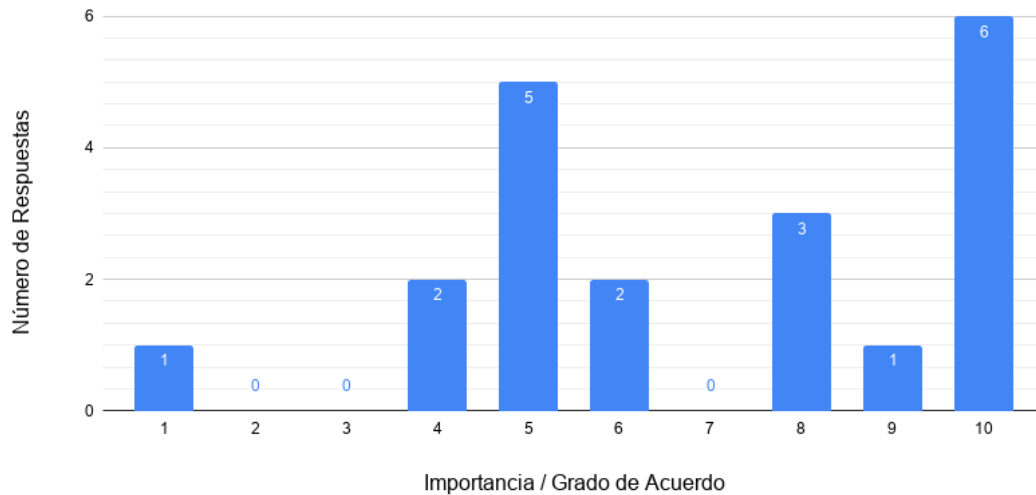
j) Imagen ilustrativa de la quinta pregunta

Ejemplo: Sistemas Operativos



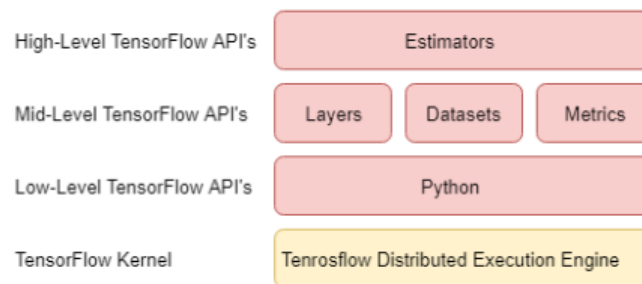
k) Sexta pregunta

P-6: ¿Crees que la interfaz gráfica facilite el uso del backend de bajo nivel para crear RNA's aunque pueda causar cuellos de botella en la tarea de implementación y problemas de rendimiento?



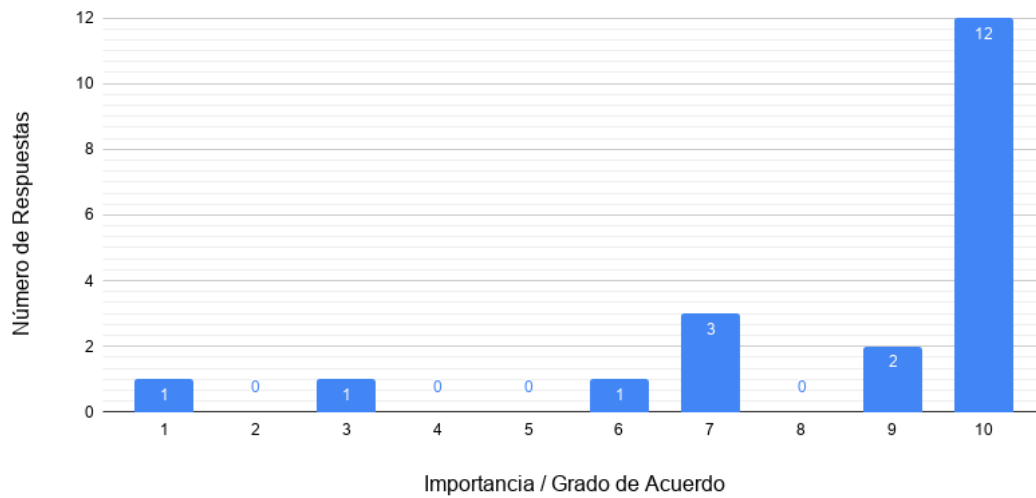
l) Imagen ilustrativa de la sexta pregunta

Ejemplo: Arquitectura de la API de bajo nivel



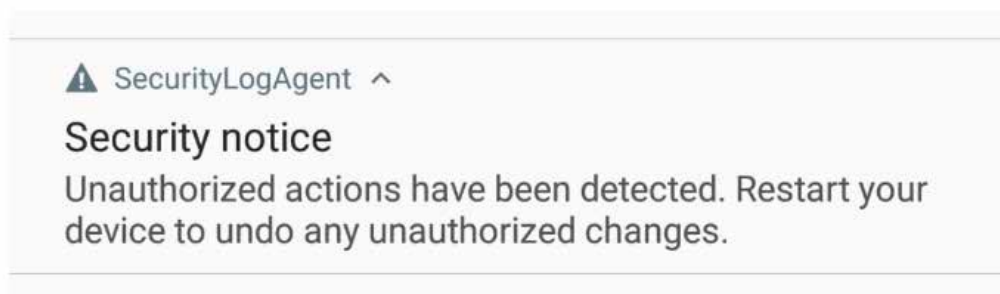
m) Séptima pregunta

P-7: ¿Debería mantenerse un registro sobre quién modifica la estructura de la red neuronal, a fin de evitar cambios no autorizados en los parámetros de entrenamiento y en la estructura de la misma?



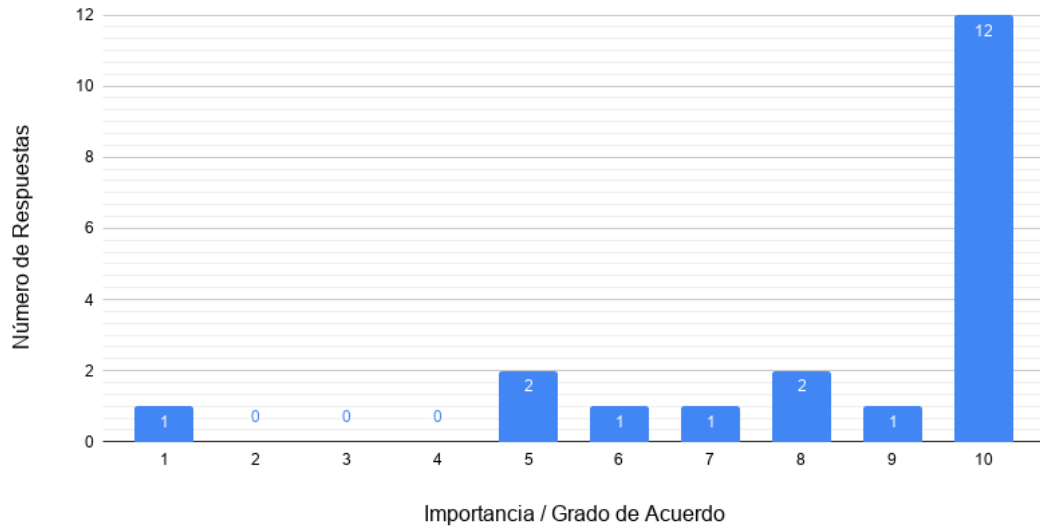
n) Imagen referencial de la séptima pregunta

Ejemplo: Registro de acciones



o) Octava pregunta

P-8: ¿Debería contar la aplicación con módulos que separen las actividades de lectura, creación, entrenamiento y predicción de la RNA ?



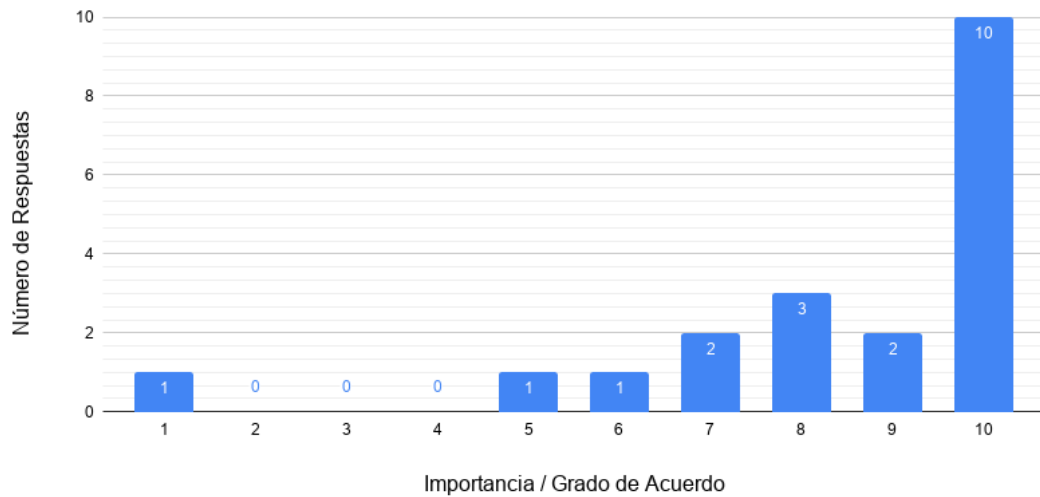
p) Imagen ilustrativa de la octava pregunta

Ejemplo: Modularidad



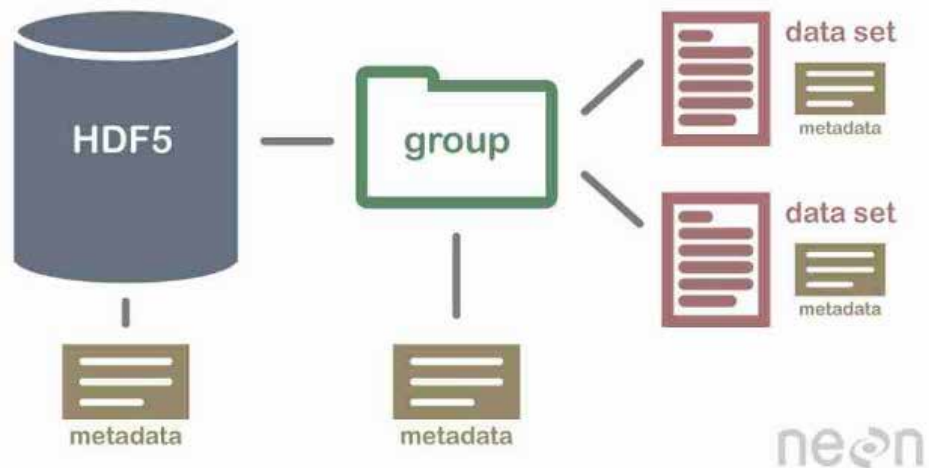
q) Novena pregunta

P-9: ¿Considera necesario el poder crear, guardar y leer conjuntos de datos de imágenes mediante la aplicación, evitando el tener que pre procesarlas en cada ejecución de lectura?



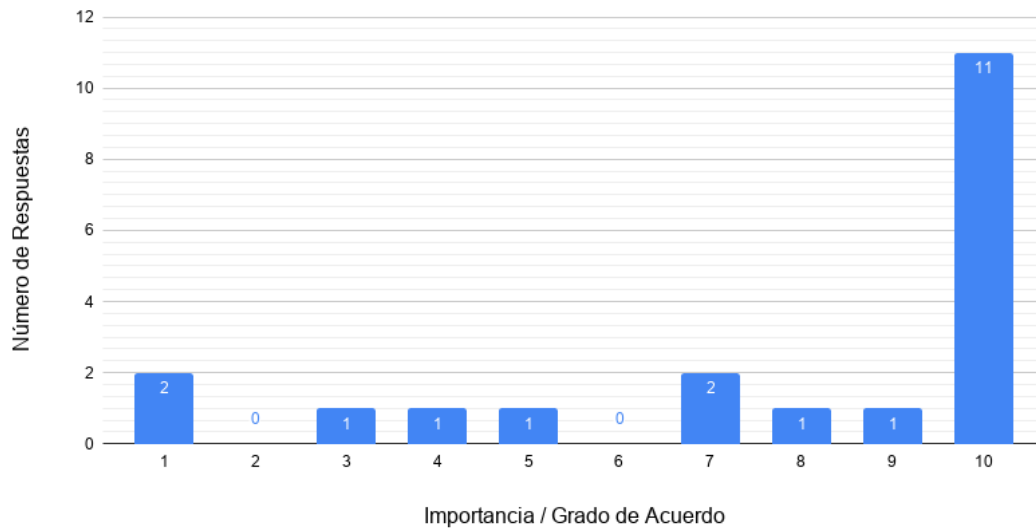
r) Imagen ilustrativa de la novena pregunta

Ejemplo: Creador de datasets (conjunto de datos)




s) Décima pregunta

P-10: ¿Es importante que la aplicación muestre información comparativa de la clase predicha vs la categoría real de la imagen en la tarea de predicción?

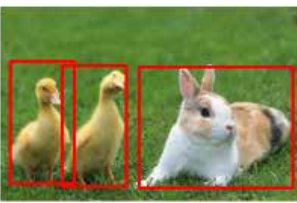


t) Imagen ilustrativa de la décima pregunta

Ejemplo: Tareas de Clasificación, Detección y Segmentación




Classification



- 1: Duck ,
[x1, y1, x2, y2]
- 2: Duck,
[x1, y1, x2, y2]
- 3: Bunny,
[x1, y1, x2, y2]

Object Detection



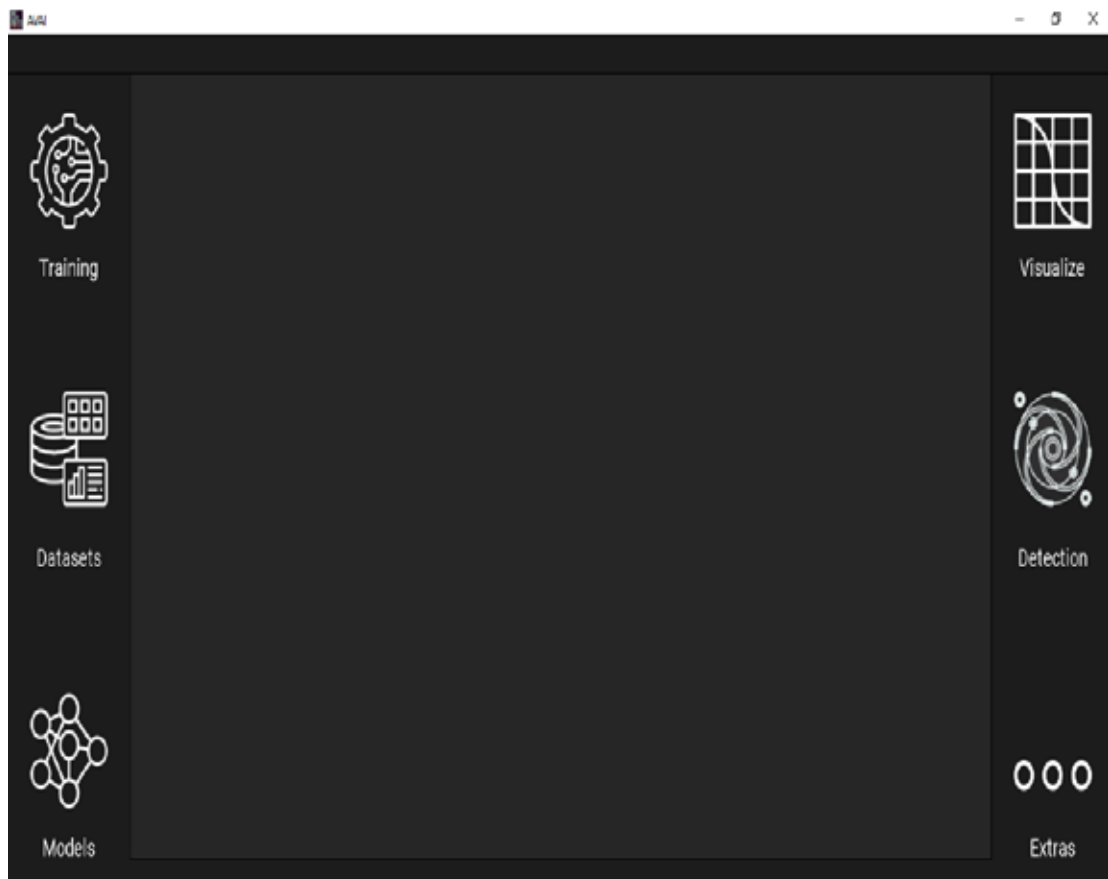
Segmentation

ANEXO B

Procesos realizados por la aplicación

Este anexo muestra los resultados finales del proceso de codificación, entrenamiento y de visualización de la red neuronal convolucional mediante la aplicación.

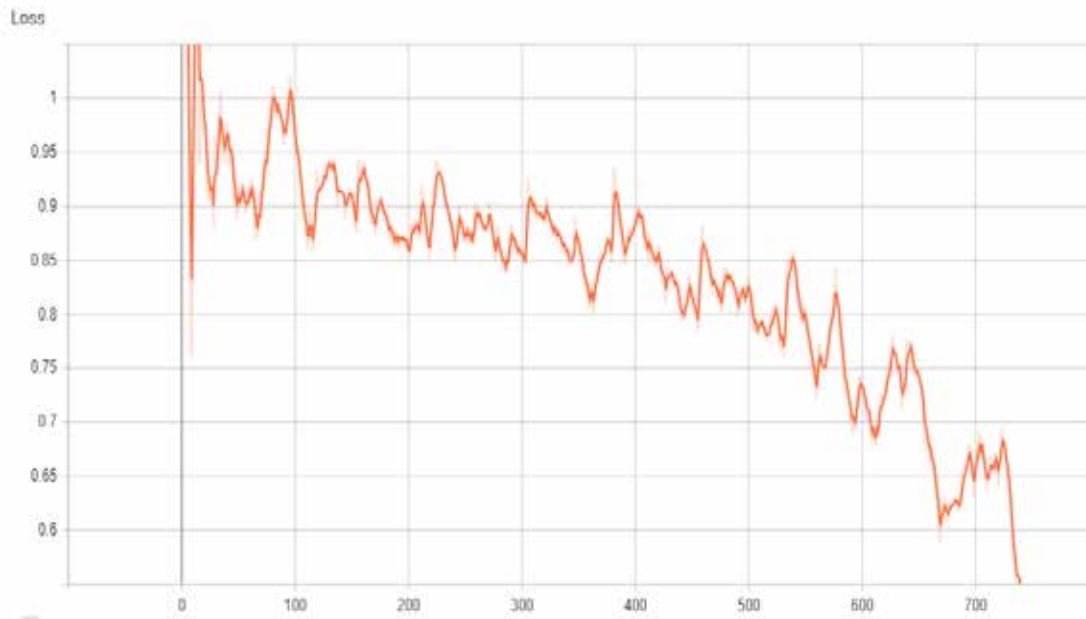
a) Vista principal de la aplicación



b) Precisión del modelo en función del progreso del entrenamiento



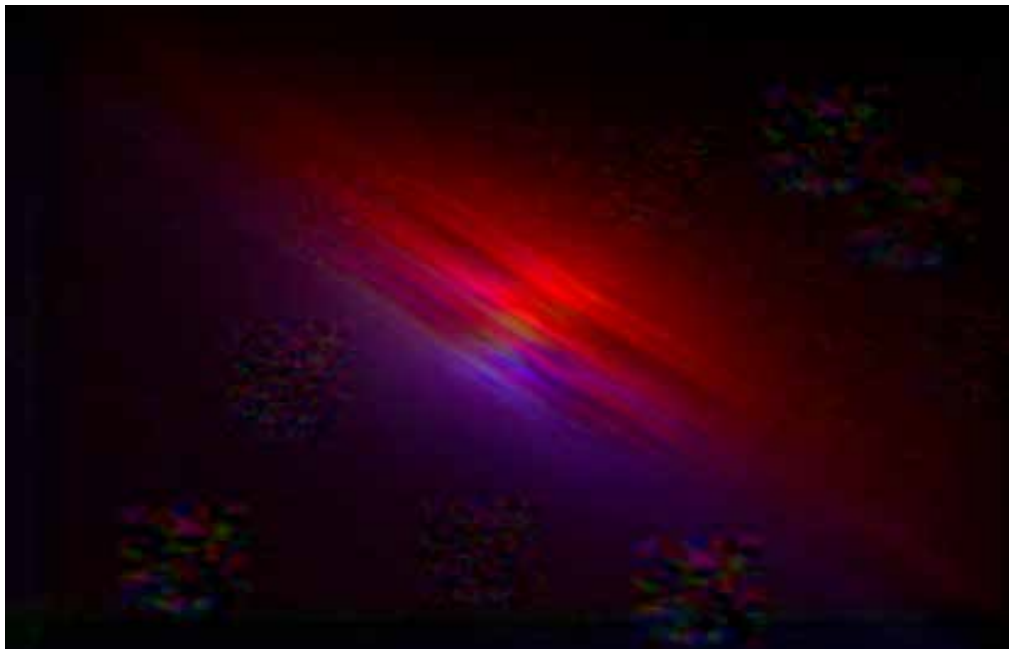
c) Resultados de la función de pérdida



**d) Imagen del conjunto de entrenamiento. Fuente: European Space Agency
NGC 7814.**



e) Feature map de NGC 7814 luego de la convolución en C1.



f) Feature map de NGC 7814 luego de modificar los parámetros de C1.



g) Feature map de NGC 7814 en C2.

