

UNIVERSIDAD NACIONAL DE TRUJILLO

Facultad de Ciencias Físicas y Matemáticas

Escuela Profesional de Informática



**Sistema Basado en Conocimiento Mediante RNA para
Pronósticar las Ventas en el Mall Aventura**

TESIS

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO
INFORMÁTICO**

AUTOR: Lopez Andrade Alex Michael Andre

ASESOR: Mg. Bravo Escalante Jorge David

TRUJILLO - PERÚ

2020

PAUL DENIS SANCHEZ RODRIGUEZ

**SISTEMA BASADO EN CONOCIMIENTO MEDIANTE
RNA PARA PRONÓSTICAR LAS VENTAS EN EL MALL
AVENTURA**

Tesis presentada a la Escuela Profesional de Informática en la Facultad de Ciencias Físicas y Matemáticas de la Universidad Nacional de Trujillo, como requisito parcial para la obtención del grado de Título profesional de Ingeniero Informático

ASESOR: Mg. BRAVO ESCALANTE JORGE DAVID

Trujillo - Perú

2020

HOJA DE APROBACIÓN

Sistema Basado en Conocimiento Mediante RNA para Pronósticar las Ventas en el Mall Aventura

Lopez Andrade Alex Michael Andre

Tesis defendida y aprobada por el jurado examinador:

Prof. Mg. Yenny M. Sifuentes Díaz - Asesor
Departamento de Informática - UNT

Prof. Mg. Ricardo Manuel Guevara Ruiz
Departamento de Informática - UNT

Prof. Mg. Sofía Pedro Huamán
Departamento de Informática - UNT

Trujillo, 2020

Dedico esta tesis a:

Mis padres, por el gran apoyo recibido...

Lopez Andrade Alex Michael Andre

Agradecimientos

Agradezco a:

Mis padres, por el apoyo ...

Lopez Andrade Alex Michael Andre

Resumen

Actualmente ...

Palabras claves: redes neuronales, pronostico, sistemas basado en conocimiento.

Abstract

Currently ...

Keywords: neural networks, , intelligent systems.

Lista de símbolos

Constantes:

- (1) r, \bar{r} Índice que denota regiones.
- (2) n Índice de bienes finales deseados por los consumidores.
- (3) ...

Variables:

- (5) x^r Vector columna que denota la actividad de producción.
- (6) u^r ...

Índice de figuras

2.1. Modelo definidos por KADS.	8
2.2. Red neuronal artificial.	14
2.3. Gráfico de la función sigmoide.	17
2.4. Representación de un red multicapa.	20
2.5. Ecuaciones del backpropagation.	23
3.1. Requerimiento funcional de investigación.	32
3.2. Modelo de diagrama de clases para la investigación de la tesis.	34
3.3. Modelo de estados para la investigación de la tesis.	34
3.4. Modelo de casos de uso para la investigación de la tesis.	35
3.5. Diagrama de componentes de nuestra investigación.	36
3.6. Modelo de secuencia para clasificar automáticamente documentos.	37
3.7. Modelo de secuencia para reporte de documentos.	37
3.8. Diagrama de la base de datos de entidad relación.	38
3.9. Login de la aplicación.	39

3.10. Interfaz principal.	39
3.11. Interfaz de la carga de los documentos.	40
3.12. Interfaz de la búsqueda múltiple de documentos.	41
3.13. Interfaz de reporte de los documentos.	41
3.14. Diagrama de flujo para el desarrollo de la investigación.	42
3.15. Extracción de datos.	43
3.16. Descripción del patrón.	44
3.17. Arquitectura de la aplicación.	44
3.18. Red neuronal en Joone.	45
3.19. Entrenamiento de las 5000 épocas.	46
3.20. Pesos de salida.	47
3.21. Pesos de salida.	48
3.22. Bias.	48
3.23. Inicio de vector característico.	50
3.24. Activando neuronas en la capa de entrada.	50
3.25. Activando neuronas en la capa oculta.	51
3.26. Activando neuronas en la capa de salida.	51
3.27. Resultado de la función sigmoide en la aplicación.	52
4.1. Resultado detallado para la búsqueda, segun criterio subir con éxito	55

4.2. Tabla cruzada: subir con éxito vs. encontró el doc.	55
4.3. Resultado detallado para la búsqueda según criterio sin error ortográfico. . .	57
4.4. Tabla cruzada: sin error ortográfico vs. encontró el doc.	57
4.5. Trabajadores antes de usar la aplicación.	59
4.6. Trabajadores usando la aplicación.	60
A.1. Inicio de la interfaz de la aplicación.	72
A.2. Interfaz de administración de los usuarios.	73
A.3. Interfaz de registro de actividades.	73
A.4. Destino de los documentos.	74
A.5. Tipos de documentos.	74
A.6. Función para cargar muchos documentos a la vez.	75
A.7. La librería POI permite ingresar los tipos de documentos: .doc, .docx y .pdf.	76
A.8. Evaluación del arreglo.	77
A.9. Se muestra los valores de la red neuronal.	78
A.10. Se muestra los nombres de las variables del usuario.	78

Índice de tablas

2.1. Componentes del perceptrón.	19
2.2. Indicadores para en nuestra investigación.	28
3.1. Requerimiento no funcional de nuestra investigación.	33
3.2. Datos del entrenamiento	47
3.3. Vector de aprendizaje.	48
4.1. Tabla de valores predictivos.	54
4.2. Tabla base de comparación de tiempos	58
4.3. Resultados con tiempos, sin usar la aplicación y usando la aplicación	60
A.1. Tabla con los documentos de nuestra población	64
A.2. Encuesta a trabajadores sin la aplicación.	69
A.3. Encuesta a trabajadores con la aplicación.	70
A.4. Destino de los documentos.	71

Índice general

Dedicatoria	I
Agradecimientos	II
Resumen	III
Abstract	IV
Lista de símbolos	V
Índice de Figuras	VIII
Índice de Tablas	IX
1. Introducción	1
1.1. Justificación de la investigación	2
1.2. Formulación del problema	2

1.3.	Hipótesis	3
1.4.	Objetivos	3
1.4.1.	Generales	3
1.4.2.	Específicos	3
1.5.	Estructura de la tesis	4
2.	Materiales y Métodos	6
2.1.	Marco teórico	6
2.1.1.	Metodología Common Kads	7
2.1.1.1.	Descripción	7
2.1.2.	Inteligencia Artificial (IA)	9
2.1.3.	Sistemas inteligentes	10
2.1.3.1.	Capacidades requeridas	11
2.1.4.	Red Neuronal Artificial (RNA)	12
2.1.5.	Tipos de aprendizaje	13
2.1.5.1.	Aprendizaje supervisado	14
2.1.6.	Elementos básicos	14
2.1.6.1.	Función de entrada (input function)	15
2.1.6.2.	Función de activación (activation function)	15
2.1.6.3.	Función de salida (output function)	17

2.1.7.	Perceptrón	18
2.1.7.1.	Perceptrón simple	18
2.1.7.2.	Perceptrón Multicapa	20
2.1.8.	Backpropagation	21
2.1.9.	Definición de regresión	23
2.1.9.1.	Fases del análisis de regresión múltiple	24
2.1.10.	Mysql Worbench	25
2.1.11.	Python	26
2.1.12.	TensorFlow	27
2.2.	Método de la investigación	27
2.2.1.	Tipo de investigación	27
2.2.2.	Variables de la Investigación	28
2.2.2.1.	Variable Dependiente	28
2.2.2.2.	Variable Independiente	28
2.2.3.	Operacionalización de la variable	28
2.3.	Recolección de datos para la elaboración del modelo	29
2.3.1.	Técnica	29
2.3.2.	Población	29
2.3.3.	Muestra	29

2.3.3.1. Muestra por conveniencia	29
2.4. Etapas de la investigación	29
3. Sistema Basado en Conocimiento Mediante RNA paraPronósticar las Ventas en el Mall Aventura	31
3.1. Análisis	31
3.1.1. Requerimiento funcional	31
3.1.2. Requimiento no funcional	33
3.2. Diseño	33
3.2.1. Diagrama de clases	33
3.2.2. Diagrama de modelos de estados	34
3.2.3. Diagrama de casos de uso	35
3.2.4. Diagrama de componentes	36
3.2.5. Diagrama de secuencia	37
3.2.6. Diagrama de la base de datos entidad relación	38
3.2.7. Diseño de interfaces	38
3.2.8. Diseño de algoritmo	42
3.2.9. Extracción de datos	43
3.2.10. Arquitectura de la aplicación	44
3.3. Implementación	45

3.4. Entrenamiento	45
3.5. Funcionamiento de las redes neuronales	49
3.6. Función activación	52
4. Resultados y discusión de la tesis	53
4.1. Resultados computacionales	53
4.1.1. Resultado según criterio subir con éxito	54
4.1.2. Resultado según criterio sin error ortográfico	56
4.2. Tiempo	58
5. Consideraciones finales	61
5.1. Conclusiones	61
5.2. Trabajos futuros	62
Referencias bibliográficas	64
A. Código de la aplicación	64

Capítulo 1

Introducción

Las empresas para elaborar sus planes de mercadeo y su planeación requieren la elaboración inicial de un pronóstico de ventas, el cual les permitirá proyectar las posibles ventas futuras basándose en datos históricos de la misma empresa, con el fin de planear, administrar y controlar los presupuestos necesarios para un buen uso de los recursos que se requerirán para cumplir con las metas propuestas.

El pronóstico de ventas es una herramienta comercial que permite estimar las ventas a futuro, con el fin de establecer metas en un determinado periodo, para su elaboración se tienen en cuenta los resultados históricos y las tendencias de ventas presentadas por el área comercial.

En base a eso, la empresa (restaurant Combiche-Trujillo), mediante sus datos históricos de ventas, desea conocer cuáles serán las ventas que obtendrán en los siguientes periodos posterior a su análisis. Con el apoyo de las redes neuronales, se busca estimar las ganancias

que se obtendrán según el número de ventas pronosticadas , dado que en la actualidad, existen varios restaurantes en la ciudad de Trujillo, y que gracias a la aplicación de dicha herramienta computacional, contribuirá a que la empresa tome medidas de precaución cuando el mercado esté bajo.

1.1. Justificación de la investigación

La importancia de esta investigación desde un punto de vista informático se justifica académicamente, como un aporte para la investigación y una motivación para profundizar en inteligencia artificial, redes neuronales.

Este proyecto se justifica socialmente, ya que no solo beneficiará a la empresa, restaurant Combiche - Trujillo, sino a los comensales que desean degustar las variedades de platos que estan a la venta, y al ser realizado con sistemas basado en conocimiento mediante RNA la aplicación irá mejorando a medida que se vaya usando en el pronóstico de las ventas, ya que tiene la principal ventaja de ir aprendiendo.

1.2. Formulación del problema

El problema de la empresa es establecer medidas de precaución ante las bajas ventas que se puedan tener y para ello se necesita de una herramienta computacional que pronostique, en base a las ventas históricas de la empresa, si hay ganancias o pérdidas de las ventas para

tomar una decisión que beneficie a la empresa. Analizando nuestro problema, nos nace la siguiente pregunta:

¿Cómo pronosticar las ventas de un producto utilizando una red neuronal artificial?

1.3. Hipótesis

Mediante el desarrollo de un sistema basado en conocimiento utilizando R.N.A. se podrá pronosticar las ventas en el Mall Aventura Plaza.

1.4. Objetivos

1.4.1. Generales

Desarrollar un sistema basado en conocimiento mediante RNA para pronosticar las ventas de un producto.

1.4.2. Específicos

- Realizar una investigación bibliográfica para recolectar datos referentes al tema de investigación y analizar los resultados de una encuesta aplicada a un experto en ventas.
- Aplicar la metodología Common Kads especial para sistemas basados en conocimientos.
- Diseñar la arquitectura de nuestro sistema basado en conocimiento.

- Entrenar el software basándonos en el modelo computacional diseñado previamente.
- Analizar y comparar los resultados del software contra otros métodos de pronósticos de ventas (lineal, algorítmico).

1.5. Estructura de la tesis

Nuestro trabajo de investigación tiene los siguientes capítulos:

- En el primer capítulo se desarrolla los aspectos generales así como la justificación, formulación del problema, hipótesis de la investigación y objetivos del presente trabajo de investigación.
- En el segundo capítulo se realiza la recopilación de diferentes conceptos y definiciones que es de mucha importancia a lo largo de toda la investigación, como la metodología de investigación, inteligencia artificial, sistemas inteligentes, redes neuronales, perceptrón, perceptrón multicapa, backpropagation., también se define definiciones de clasificación automática de documentos, así como su representación vectorial y el centroide del conjunto de palabras, similitud aplicando métodos de clasificación automática de documentos, también describimos los tipos de documentos y por último como conceptos finales, consideramos a los gestores, plataformas y lenguajes que utilizaremos así como MySQL Workbench 8.0, NeatBeans IDE 8.2, Java y Joone.

- En el tercer capítulo se analiza, comprende, implementa y optimiza las soluciones de los sistemas inteligentes para la clasificación automática de documentos. Empezando por algunos problemas de algoritmos actuales. También la prueba a seguir de redes neuronales, se diseñan los modelos propuestos y se explica la implementación del sistema.
- En el cuarto capítulo se contrasta los resultados con el prototipo de la aplicación de sistemas inteligentes para la clasificación automática de documentos.
- En el quinto capítulo se presenta las conclusiones de nuestro trabajo de investigación, demostrando los objetivos cumplidos.

Capítulo 2

Materiales y Métodos

2.1. Marco teórico

El siguiente capítulo presenta conceptos importantes que utilizamos como parte fundamental para el desarrollo de nuestra tesis, este material bibliográfico fue investigado por la relación que tiene con los temas de nuestra investigación. La investigación se tornó muy amplia, principalmente en los conceptos claves para nuestro conocimiento científico, el cual ayudó a desarrollar la tesis, inteligencia artificial, sistemas inteligentes, redes neuronales, perceptrón, backpropagation, conceptos importantes ya que sin ellos sería imposible el modelamiento y desarrollo.

2.1.1. Metodología Common Kads

La metodología elegida para el desarrollo del software es la metodología ágil *Common Kads*. Es una metodología diseñada para el análisis y la construcción de sistemas basados en conocimiento (SBC) de forma análoga a los métodos empleados en ingeniería de software. Fue propuesta y desarrollada por un grupo de investigadores pertenecientes a diversos países de la comunidad Europea, a través del programa ESPRIT para la innovación y la aplicación de tecnología informática avanzada. Lo que se pretendía era crear un estándar para la ingeniería del conocimiento y sistemas de conocimiento con el cual se pudieran construir sistemas industriales de calidad a gran escala, en una forma estructurada y controlada. (?).

2.1.1.1. Descripción

En *Common KADS* podemos ver reflejadas algunas ideas que han emergido, no solo de la experiencia en la ingeniería del conocimiento, sino también en el campo de la ingeniería del software en general. Estas ideas se pueden concretar en tres conceptos: modelado, reutilización y gestión del riesgo.

El modelado se refiere la configuración ideal que representa de manera simplificada una teoría, la cual ayuda a comprender las teorías y leyes.

La reutilización se centra en utilizar nuevamente productos ya creados anteriormente una o varias veces las cuales sean necesarias.

Gestión del riesgo se refiere al proceso de identificar, analizar y responder a factores de riesgo a lo largo de la vida de un proyecto, beneficiando los objetivos propuestos.

La metodología *CommonKADS* abarca todo el ciclo de desarrollo de software mediante un número de modelos interrelacionados que capturan los principales rasgos del sistema y de su entorno. El proceso de desarrollo de SBC consiste en rellenar un conjunto de plantillas de los modelos. Asociados a las plantillas, *CommonKADS* define estados de los modelos que caracterizan hitos en el desarrollo de cada modelo. Estos estados permiten la gestión del proyecto, cuyo desarrollo se realiza de una forma cíclica dirigida por los riesgos. (?).

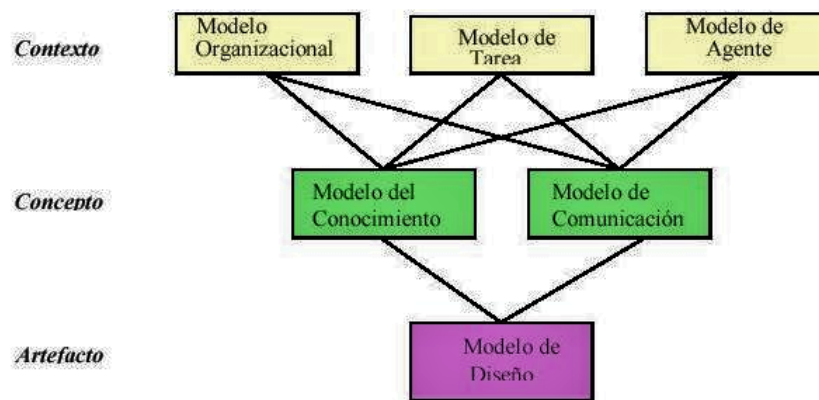


Figura 2.1: Modelo definidos por KADS.
Fuente: ?.

La metodología *CommonKads* tiene tres etapas: el contexto, concepto y artefacto; en cada etapa contienen modelos, en la etapa de contexto tiene modelo, organizacional, modelo de tarea y modelo de agente; en concepto tiene modelo de conocimiento y modelo de comunicación; y en artefacto tiene modelo de diseño.

La relación que fija con UML son los siguientes diagramas: diagrama de clases, diagrama de estados, diagrama de actividades y diagrama de casos de uso. (?)

2.1.2. Inteligencia Artificial (IA)

Empecemos esto con una pregunta clave: ¿Qué es la inteligencia artificial?. Aunque es un término de compleja definición, esta puede ser catalogada como una rama de la ciencia que se dedica al estudio de la forma en que se desarrolla el proceso del pensamiento humano para reproducirlo en un ente artificial. Es por ello que para poder lograrlo, la I.A. centra su estudio en dos áreas: el cuerpo humano (estudio de la inteligencia) y el ordenador electrónico. Teniendo en cuenta que para poder reproducir el pensamiento es necesario conocerlo, como primera meta para la investigación de los científicos fue entender los procesos cognoscitivos de la mente humana. De esta manera definir su modelo de inteligencia, para luego proceder a programarlo en ordenadores, simulando cada uno de los procesos y comprobando los resultados de sus teorías. (?)

Según ?, acuñó la expresión “inteligencia artificial”, y la definió como: “la ciencia e ingenio de hacer máquinas inteligentes, especialmente programas de cómputo inteligentes”.

2.1.3. Sistemas inteligentes

Los sistemas inteligentes radican en un conjunto de herramientas y aplicaciones que agrupados llevan a cabo la recopilación, extracción y formato de información obtenida de distintas fuentes con el único propósito de crear medios inteligentes y artificiales para múltiples usos. Frecuentemente se usa para el soporte en la toma de decisiones, sin embargo, esta no es su única funcionalidad.

Son sistemas que, durante su existencia, aprenden y almacenan para luego actuar continuamente de forma interna o externa, de forma que pueda alcanzar su propósito superándose cada vez. Tiene su propio objetivo esencial y para alcanzarlo, escoge una acción tomada de las experiencias que ha almacenado en su memoria.

Para poder hablar de un sistema inteligente es necesario que exista un entorno con el cual el sistema pueda interactuar y, además, el sistema inteligente debe contener “sentidos” que le permitan recibir comunicaciones de dicho entorno y con ello poder transmitir información. El sistema actúa continuamente y cuenta con una memoria para archivar el resultado de sus acciones. Como se dijo anteriormente, este sistema tiene un objetivo, por lo cual, el conseguirlo significaría el hecho de seleccionar la respuesta adecuada ante cada estímulo. También se caracteriza porque a través de su memoria, durante su existencia, aprende de sus experiencias, logrando así, mejorar tanto su rendimiento como su eficiencia. Por último, es necesario indicar que este sistema consume energía, la cual utiliza en sus procesos internos

y su actuar. (?).

2.1.3.1. Capacidades requeridas

Según ?, un sistema inteligente podrá ser considerado completo, si se incluye dentro de estas diversas funcionalidades, tales como :

- **Inteligencia:** Es la capacidad de alcanzar nuestros objetivos. La inteligencia incluye la capacidad de aprender a lograrlo.
- **Conceptualización:** Un concepto es el elemento primordial del pensamiento. Es el almacenamiento físico, material de información (en neuronas o electrones). Todos los conceptos de la memoria están interrelacionados en red. La capacidad de poder conceptualizar implica el desarrollo de distintos niveles de abstracción.
- **Reglas de actuación:** Una regla de actuación es la consecuencia de una experiencia o el resultado de interpretar la propia memoria, es decir, relaciona situaciones y efectos de la acción.
- **Memoria:** La memoria es un almacenamiento físico de conceptos y reglas de actuación, que incluye la experiencia del sistema.
- **Aprendizaje:** Posiblemente, es la capacidad más importante de un sistema inteligente, ya que este aprende conceptos a partir de la información obtenida a través de los

sentidos. Aprende también reglas de actuación a base de su experiencia, la cual en ocasiones, la actuación se almacena con su valor. Una regla de actuación aumenta en valor si permitió el logro de un objetivo. El aprendizaje incluye la inserción de conceptos abstractos, a base de ejemplos concretos y la creación de conceptos compuestos que contienen los conceptos de partes de un objeto. Entonces el aprendizaje también se define como la capacidad de detectar relaciones (patrones) entre la parte (“situación la parte”, “situación futura”) de una regla de actuación.

2.1.4. Red Neuronal Artificial (RNA)

Una red neuronal es un sistema de procesadores paralelos conectados entre sí en forma de grafo dirigido. Esquemáticamente cada elemento de procesamiento (neuronas) de la red se representa como un nodo. Estas conexiones establecen una estructura jerárquica que tratando de emular la fisiología del cerebro busca nuevos modelos de procesamiento para solucionar problemas concretos del mundo real. Lo importante en el desarrollo de la técnica de las RNA es su útil comportamiento al aprender, reconocer y aplicar relaciones entre objetos y tramas de objetos propios del mundo real. (?)

Las ventajas de la red neuronal son:

1. Aprendizaje adaptativo: Son adaptables debido a la capacidad de autoajuste de los elementos procesales (neuronas) que componen el sistema.

2. Auto organización: La auto organización consiste en la modificación de la red neuronal completa para llevar a cabo un objetivo específico. Cuando las redes neuronales se usan para reconocer ciertas clases de patrones, ellas auto organizan la información usada.
3. Tolerancia a fallos: Las redes neuronales pueden aprender a reconocer patrones con ruido, distorsionado o incompletos. Esta es una tolerancia a fallos respecto a los datos. Las redes pueden seguir realizando su función (con cierta degradación) aunque se destruya parte de la red.
4. Operación en tiempo real: Las redes neuronales se adaptan bien a esto debido a su implementación paralela.
5. Fácil inserción dentro de la tecnología existente: Con las herramientas computacionales existentes (no del tipo computadora), una red puede ser rápidamente entrenada, comprobada, verificada y trasladada a una implementación hardware de bajo coste.

2.1.5. Tipos de aprendizaje

Hay dos métodos de aprendizaje importantes, que son: aprendizaje supervisado y no supervisado, la diferencia entre ambos tipos depende en la existencia o no de un agente externo que controle todo el proceso. (?):

2.1.5.1. Aprendizaje supervisado

El aprendizaje supervisado se caracteriza porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo (supervisor, maestro) que determina la respuesta que debería generar la red a partir de una entrada determinada.

En este tipo de aprendizaje se suelen considerar, a su vez, tres formas de llevarlo a cabo, que dan lugar a los siguientes aprendizajes supervisados: Aprendizaje por corrección de error, aprendizaje por refuerzo y aprendizaje estocástico. (?).

2.1.6. Elementos básicos

A continuación se puede ver, en la Figura 2.2, un esquema de una red neuronal:

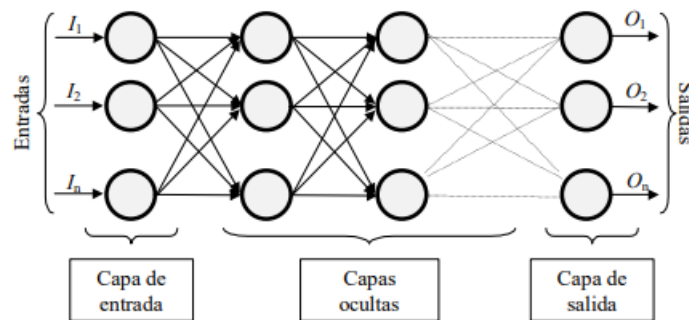


Figura 2.2: Red neuronal artificial.

Fuente: ?.

Esta red neuronal está constituida por neuronas interconectadas y arregladas en tres capas. Los datos ingresan por medio de la “capa de entrada”, pasan a través de la “capa oculta”

y salen por la “capa de salida”. Cabe mencionar que la capa oculta puede estar constituida por varias capas.

2.1.6.1. Función de entrada (input function)

La neurona trata a muchos valores de entrada como si fueran uno solo; esto recibe el nombre de entrada global. Por lo tanto, ahora nos enfrentamos al problema de cómo se pueden combinar estas simples entradas $(in_{i1}, in_{i2}, \dots)$ dentro de la entrada global, gin_i . Esto se logra a través de la función de entrada, la cual se calcula a partir del vector entrada. La función de entrada puede describirse como sigue:

$$input_i = (in_{i1} \bullet w_{i1}) * (in_{i2} \bullet w_{i2} * \dots (in_{in} \bullet w_{in}) \quad (2.1)$$

donde: * representa al operador apropiado (por ejemplo: máximo, sumatoria, productoria, etc.), n al número de entradas a la neurona n_i y w_i al peso. (?).

2.1.6.2. Función de activación (activation function)

Una neurona biológica puede estar activa (excitada) o inactiva (no excitada); es decir, que tiene un “estado de activación”. Las neuronas artificiales también tienen diferentes estados de activación; algunas de ellas solamente dos, al igual que las biológicas, pero otras pueden tomar cualquier valor dentro de un conjunto determinado. (?).

La función activación calcula el estado de actividad de una neurona; transformando la entrada global (menos el umbral, σ_i) en un valor (estado) de activación, cuyo rango normalmente va de (0 a 1) o de (-1 a 1). Esto es así, porque una neurona puede estar totalmente inactiva (0 o -1) o activa (1). La función activación, es una función de la entrada global (gin_i) menos el umbral (σ_i). Las funciones de activación más comúnmente utilizadas se detallan a continuación:

- Función Lineal

con $x = gin_i - \sigma$, y $a > 0$.

Los valores de salida obtenidos por medio de esta función de activación serán: $a \cdot (gin_i - \sigma_i)$, cuando el argumento de $(gin_i - \sigma_i)$ esté comprendido dentro del rango $(-1/a, 1/a)$. Por encima o por debajo de esta zona se fija la salida en 1 o -1, respectivamente. Cuando $a = 1$ (siendo que la misma afecta la pendiente de la gráfica), la salida es igual a la entrada. (?).

- Función Sigmoidea

$$f(x) = \frac{1}{1 + e^{-gx}}, \text{ con } x = gin_i - \sigma, \quad (2.2)$$

Los valores de salida que proporciona esta función están comprendidos dentro de un rango que va de 0 a 1. Al modificar el valor de g se ve afectada la pendiente de la función de activación.

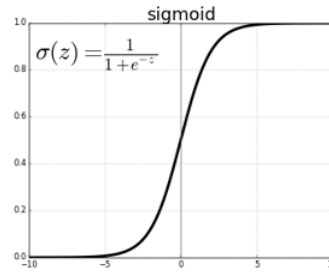


Figura 2.3: Gráfico de la función sigmoide.

Fuente: Elaboración propia.

■ Función Tangente hiperbólica

$$f(x) = \frac{e^{gx} - e^{-gx}}{e^{gx} + e^{-gx}}, \text{ con } x = \text{gin}_i - \sigma, \quad (2.3)$$

Los valores de salida de la función tangente hiperbólica están comprendidos dentro de un rango que va de -1 a 1. Al modificar el valor de g se ve afectada la pendiente de la función de activación.

2.1.6.3. Función de salida (output function)

El último componente que una neurona necesita es la función de salida. El valor resultante de esta función es la salida de la neurona i (out_i); por ende, la función de salida determina que valor se transfiere a las neuronas vinculadas. Si la función de activación está por debajo de un umbral determinado, ninguna salida se pasa a la neurona subsiguiente. Normalmente,

no cualquier valor es permitido como una entrada para una neurona, por lo tanto, los valores de salida están comprendidos en el rango $[0, 1]$ o $[-1, 1]$. También pueden ser binarios 0, 1 o $-1, 1$. (?).

Dos de las funciones de salida más comunes son cuando: la salida es la misma que la entrada (función identidad), o binaria, que devuelve uno cuando el act_i es mayor o igual que el umbral ϵ_i , caso contrario es cero.

2.1.7. Perceptrón

El perceptrón lee los valores de entrada, después suma todas las entradas teniendo en cuenta los pesos y por último el resultado lo introduce en una función de activación que nos genera el resultado final. (?).

2.1.7.1. Perceptrón simple

Se determina los pesos sinápticos y el umbral que proporcione el óptimo ajuste de la entrada con la salida, estas variables, para determinar estas variables se sigue un proceso adaptativo, el cual comienza con valores aleatorios y se van modificando según la diferencia entre los valores deseados y los calculados por la red.

Recordar que el perceptrón sólo es capaz de representar funciones lineales, dado que no dispone de capas ocultas, para esto existen los perceptrones multicapa.

Tabla 2.1: Componentes del perceptrón.

Componentes del perceptrón	Definición
Entradas	Es la información que recibe el perceptrón.
Pesos	Son los valores numéricos que se encargan de establecer la influencia de una entrada en la salida deseada.
Bias	Es un parámetro que tienen algunos modelos de redes neuronales el cual permite encontrar fácilmente la separación entre posibilidades de salida de una red neuronal.
Función de activación	Se encarga de determinar un valor de salida una vez se han procesado cada una de las entradas.

Fuente: Elaboración propia según ?.

En la Tabla 2.1 se representan los componente del perceptrón simple.

El entrenamiento del perceptrón es un proceso iterativo y sigue los siguientes pasos, hasta lograr reducir el error:

- Paso 1: Inicializar los pesos y el bias.
- Paso 2: Calcular las salidas (net) con los pesos y el bias.
- Paso 3: Obtener la salida utilizando la función de activación y calcular cada valor del error.
- Paso 4: Corregir el Bias y los pesos.

2.1.7.2. Perceptrón Multicapa

El Perceptrón multicapa es una red de alimentación hacia delante compuesta por una capa de N neuronas de entrada (sensores), otra capa formada por M neuronas de salida y un número determinado de capas ocultas. (Ver Figura 2.4). El tamaño de éstas dependerán de la dificultad de la correspondencia a implementar. (?)

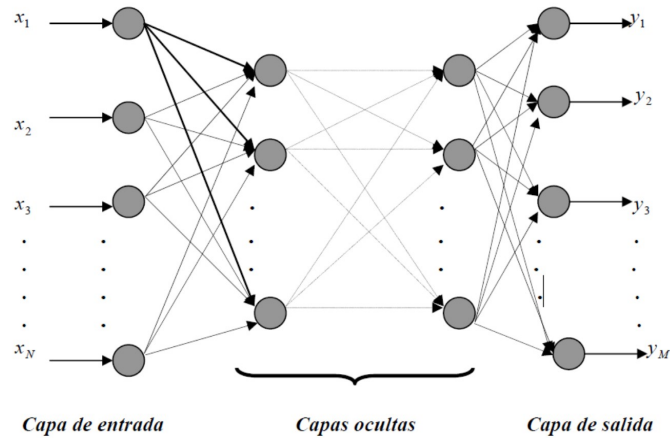


Figura 2.4: Representación de un red multicapa.

Fuente: Alba Morera (2018).

El objetivo que se busca con este tipo de red es el mismo, establecer una correspondencia entre un conjunto de entrada y un conjunto de salidas deseadas, de manera que:

$$(x_1, \dots, x_n) \in \mathbb{R}_N \rightarrow (y_1, \dots, y_n) \in \mathbb{R}_M \quad (2.4)$$

Para ello se dispone de un conjunto de p pares de entrenamiento de manera que sabemos perfectamente que al patrón de entrada (x_1^k, \dots, x_N^k) le corresponde la salida (y_1^k, \dots, y_M^k) , $k = 1, \dots, p$. Así, nuestro conjunto de entrenamiento es:

$$(x_1^k, \dots, x_N^k) \rightarrow (y_1^k, \dots, y_M^k), k=1, \dots, p \quad (2.5)$$

2.1.8. Backpropagation

La red neuronal propaga la señal de los datos de entrada hacia adelante a través de sus parámetros hacia el momento de la decisión, y luego propaga hacia atrás la información sobre el error, a la inversa a través de la red, para que pueda alterar los parámetros. Esto sucede paso a paso:

- La red adivina los datos, utilizando sus parámetros.
- La red se mide con una función de pérdida.
- El error se propaga hacia atrás para ajustar los parámetros equivocados.

La propagación hacia atrás de errores o retropropagación (del inglés backpropagation) es un método de cálculo del gradiente utilizado en algoritmos de aprendizaje supervisado utilizados para entrenar redes neuronales artificiales. El método emplea un ciclo propagación (adaptación de dos fases). Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas siguientes de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Sin embargo, las neuronas de la capa oculta sólo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total.

La importancia de este proceso consiste en que, a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas de tal modo que las distintas neuronas aprenden a reconocer distintas características del espacio total de entrada. Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas

individuales hayan aprendido a reconocer durante su entrenamiento.

$$\delta^L = \nabla_a C \odot \sigma'(z^L) \quad (\text{BP1})$$

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l) \quad (\text{BP2})$$

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l \quad (\text{BP3})$$

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l \quad (\text{BP4})$$

Figura 2.5: Ecuaciones del backpropagation.

Fuente: ?.

Donde: BP1 es para el error en la capa de salida, BP2 es para el error en términos del error en la siguiente capa, BP3 es para la tasa de cambio del costo con respecto a cualquier sesgo en la red y BP4 es para la tasa de cambio del costo con respecto a cualquier peso en la red.

2.1.9. Definición de regresión

Según ?, el término de regresión es uno de los pilares estadísticos más modernos el cual hace referencia al análisis simultaneo de dos o más variables relacionadas entre sí.

Una de las variables se le conoce como variable dependiente (y) y la otra como variable independiente (x).

$$Y = B_0 + B_1X_1 + B_2X_2 + \dots + B_kX_k \quad (2.6)$$

Donde: Y : es la variable dependiente, la cual también es denominada variable respuesta
 X_i : es la variable independiente i , la cual también se llama exploratoria B_i : es el coeficiente del modelo para la variable X_i

Tanto la variable independiente como las independientes deben ser métricas, aunque las independientes también pueden tener valores cualitativos

2.1.9.1. Fases del análisis de regresión múltiple

Según ?, las fases son:

- Identificar problema o área de oportunidad.
- Seleccionar las variables dependientes e independientes.
- Recolectar variables.
- Realizar análisis descriptivo del tipo de relación entre variables.
- Seleccionar método.
- Calcular coeficientes del modelo de regresión lineal múltiple para construir la función.
- Identificar problemas de colinealidad o multicolinealidad.

- Realizar prueba global de la ecuación.
- Efectuar pruebas individuales de los coeficientes.
- Probar cumplimiento de los supuestos del análisis.
- Interpretar coeficientes de determinación, correlación, determinación ajustado y error estándar.
- Analizar los coeficientes de la ecuación de regresión.
- Elaborar pronósticos puntuales y por intervalo.

2.1.10. Mysql Worbench

MySQL Workbench es un software creado por la empresa Sun Microsystems, esta herramienta permite modelar diagramas de Entidad-Relación para bases de datos MySQL. Con esta herramienta se puede elaborar una representación visual de las tablas, vistas, procedimientos almacenados y claves foráneas de la base de datos. Además, es capaz de sincronizar el modelo en desarrollo con la base de datos real. Se puede realizar una ingeniería directa e ingeniería inversa para exportar e importar el esquema de una base de datos ya existente el cual haya sido guardado o hecho copia de seguridad con MySQL Administrador. MySQL Workbench puede generar también el guión necesario para crear la base de datos que se ha

dibujado en el esquema; es compatible con los modelos de base de datos de DBDesigner 4 y soporta las novedades incorporadas en MySQL 5.x. (?).

2.1.11. Python

Python es un lenguaje de programación interpretado simple pero poderoso que cierra la brecha entre la programación de C y la de shell y, por lo tanto, es ideal para la "programación desechable" la creación rápida de prototipos. Su sintaxis se construye a partir de construcciones tomadas de una variedad de otros lenguajes; las más destacadas son las influencias de ABC, C, Modula-3 e Icon. El intérprete de Python se puede ampliar fácilmente con nuevas funciones y tipos de datos implementados en C. Python también es adecuado como lenguaje de extensión para aplicaciones C altamente personalizables, como editores o administradores de ventanas. Python está disponible para varios sistemas operativos, entre los que se encuentran varios tipos de UNIX (incluido Linux), el sistema operativo Apple Macintosh, MS-DOS, MS-Windows 3.1, Windows NT y OS / 2. Es conciso, pero intenta ser exacto y completo. La semántica de los tipos de objetos incorporados no esenciales y de las funciones y módulos incorporados se describe en la Referencia de la biblioteca de Python. (?).

2.1.12. TensorFlow

TensorFlow implementa una visión de la teoría de la probabilidad adaptada al paradigma moderno de aprendizaje profundo de computación diferenciable de extremo a extremo. Basado en dos abstracciones básicas, ofrece bloques de construcción flexibles para el cálculo probabilístico. Las distribuciones proporcionan métodos rápidos y numéricamente estables para generar muestras y calcular estadísticas, por ejemplo, densidad logarítmica. Los bijectors proporcionan transformaciones de seguimiento de volumen componibles con almacenamiento en caché automático. Juntos, permiten la construcción modular de distribuciones y transformaciones de alta dimensión que no son posibles con bibliotecas anteriores (por ejemplo, pixelCNN, flujos autorregresivos y redes residuales reversibles). Son el caballo de batalla detrás de los sistemas de programación probabilísticos profundos como Edward y potencian la inferencia rápida de caja negra en modelos probabilísticos construidos sobre componentes de redes profundas.. (?).

2.2. Método de la investigación

2.2.1. Tipo de investigación

La presente tesis es del tipo experimental.

2.2.2. Variables de la Investigación

2.2.2.1. Variable Dependiente

Pronosticar las ventas en el Mall Aventura Plaza.

2.2.2.2. Variable Independiente

Sistema basado en conocimiento mediante RNA.

2.2.3. Operacionalización de la variable

Nos permiten realizar mediciones y determinar la validez de la hipótesis que fue planteada en la investigación.

Tabla 2.2: Indicadores para en nuestra investigación.

Variable dependiente	Descripción	Indicador	
Pronosticar las ventas en el Mall Aventura Plaza	Técnica inteligente para pronosticar las ventas	Sensibilidad:	$\frac{\text{Verdaderos positivos}}{\text{Verdaderos positivos} + \text{falsos negativos}}$
		Precisión:	$\frac{\text{Verdaderos positivos}}{\text{Verdaderos positivos} + \text{falsos positivos}}$
		Exactitud:	$\frac{\text{Verdaderos positivos} + \text{falsos positivos}}{\text{Verdaderos positivos} + \text{Verdaderos negativos}}$
		Tiempo:	$\frac{\text{Total de documentos}}{\text{Representado en segundos}}$

Fuente: Elaboración propia.

2.3. Recolección de datos para la elaboración del modelo

2.3.1. Técnica

2.3.2. Población

La población para nuestra investigación son las ventas en los últimos 12 meses de diversos productos en el Mall Aventura Plaza - Trujillo.

2.3.3. Muestra

2.3.3.1. Muestra por conveniencia

Permite seleccionar aquellos casos accesibles que acepten ser incluidos. Esto, fundamentado en la conveniente accesibilidad y proximidad de los sujetos para el investigador. (?).

2.4. Etapas de la investigación

Nuestra investigación comprenderá las siguientes etapas:

- La investigación bibliográfica a través de búsqueda de artículos en la web, libros y casos de estudio en relación con el tema de investigación (pronósticos de ventas).
- La aplicación de la metodología del sistema basado en conocimiento (Common Kads).

- Recolección de nuestra población, ventas, los cuales se utilizó en el desarrollo de la investigación.
- Teniendo en cuenta que se necesita entrenar a las redes neuronales, puesto que son que las redes neuronales son heurísticas se requieren un constante ajuste para obtener un óptimo resultado.
- Comparar resultados del software contra otros métodos de pronósticos de ventas (lineal, algorítmico).

Capítulo 3

Sistema Basado en Conocimiento

Mediante RNA paraPronósticar las

Ventas en el Mall Aventura

3.1. Análisis

3.1.1. Requerimiento funcional

La siguiente imagen (ver Figura: 3.1) hace referencia y describen las actividades de nuestra aplicación.



Figura 3.1: Requerimiento funcional de investigación.

Fuente: Elaboración propia.

Usuario-administrador:

- Cargar documento y subir los documentos a clasificar.
- Reportar documentos documentos clasificados, por tipo y por destino.
- Mostrar gráfico de porcentaje de la cantidad de documentos clasificados, por tipo y por destino.

Administrador:

- Administrar permisos y ver los accesos que tiene los usuarios.
- Registrar, nuevo usuario para que tenga acceso a la aplicación.

3.1.2. Requerimiento no funcional

Describe otras prestaciones, características y/o limitaciones de la aplicación.

Tabla 3.1: Requerimiento no funcional de nuestra investigación.

Requerimiento no funcional	Descripción
Usabilidad	Aplicación de fácil manejo, con interfaces gráficas simples y con iconos predictivos.
Eficiencia	Aplicación eficiente, ligera y con respuesta rápida, en 300 milésimas de segundo aproximadamente por documento.
Dependibilidad	Aplicación disponible. Aplicación confiable. Aplicación integro, sin alteraciones. Fácil mantenimiento de la aplicación.
Seguridad	Tiene control para el acceso de información.

Fuente: Elaboración propia.

3.2. Diseño

3.2.1. Diagrama de clases

Modelo el cual seguiremos en el desarrollo del sistema.

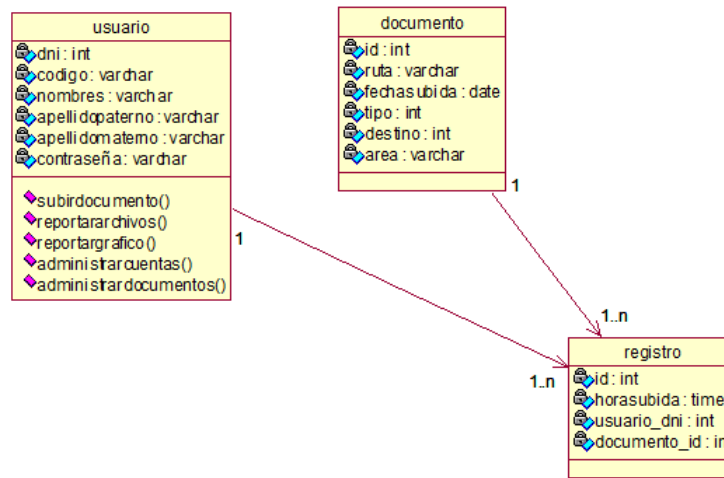


Figura 3.2: Modelo de diagrama de clases para la investigación de la tesis.

Fuente: Elaboración propia.

3.2.2. Diagrama de modelos de estados

Modelo de estados el cual seguiremos en el desarrollo del sistema.

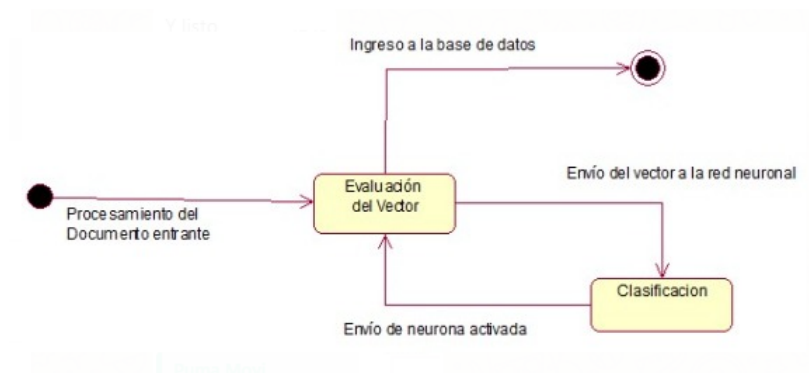


Figura 3.3: Modelo de estados para la investigación de la tesis.

Fuente: Elaboración propia.

3.2.3. Diagrama de casos de uso

Modelo de casos de uso el cual seguiremos en el desarrollo del sistema.

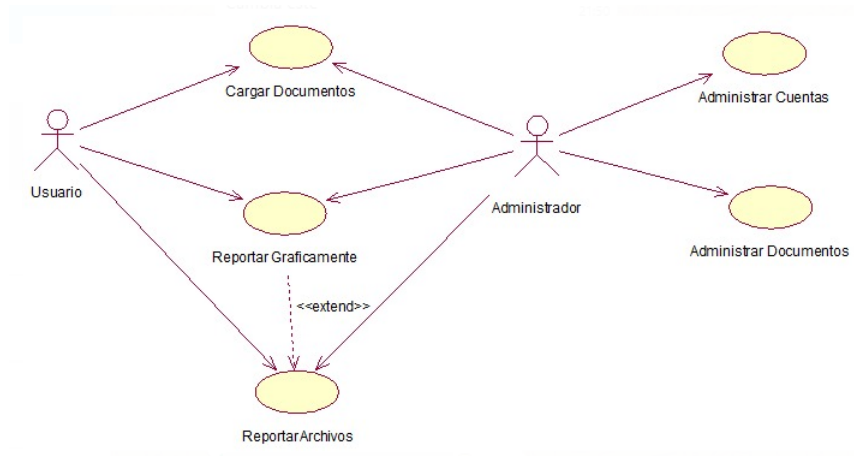


Figura 3.4: Modelo de casos de uso para la investigación de la tesis.

Fuente: Elaboración propia.

- Cargar documentos, permite subir documentos digitalizados a la aplicación.
- Administrar cuentas, permite al administrador administrar las cuentas así como ver el registro de cada usuario y crear usuarios nuevos.
- Administrar documentos, permite al administrador ver la hora y fecha de la carga de documentos por usuario.
- Reporte archivos, permite solicitar los documentos que se subieron a la aplicación.

3.2.4. Diagrama de componentes

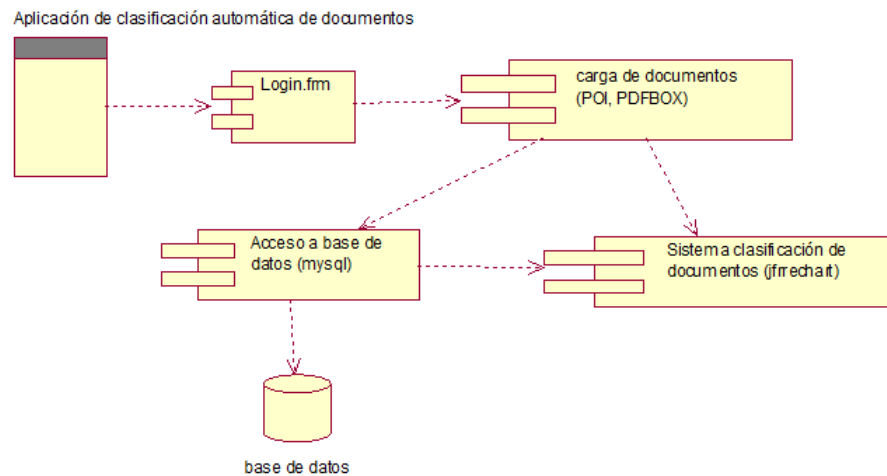


Figura 3.5: Diagrama de componentes de nuestra investigación.

Fuente: Elaboración propia.

3.2.5. Diagrama de secuencia

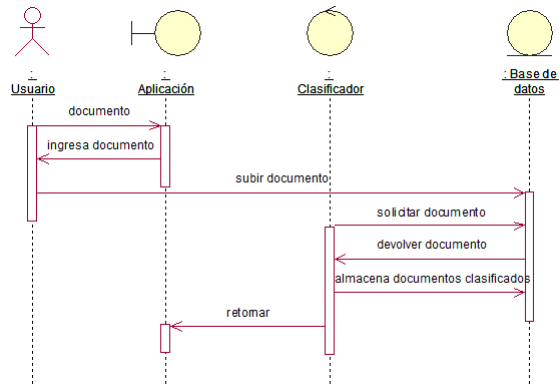


Figura 3.6: Modelo de secuencia para clasificar automáticamente documentos.

Fuente: Elaboración propia.

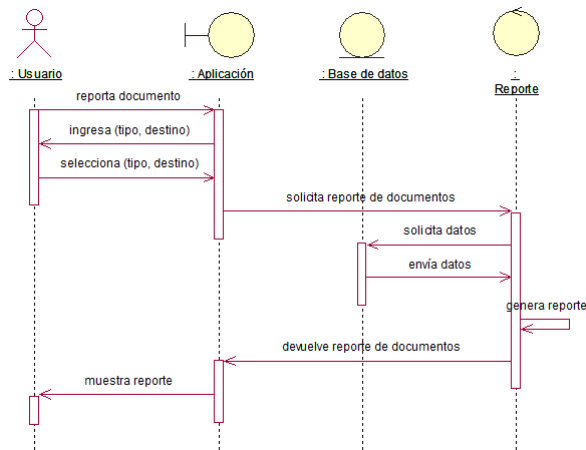


Figura 3.7: Modelo de secuencia para reporte de documentos.

Fuente: Elaboración propia.

3.2.6. Diagrama de la base de datos entidad relación

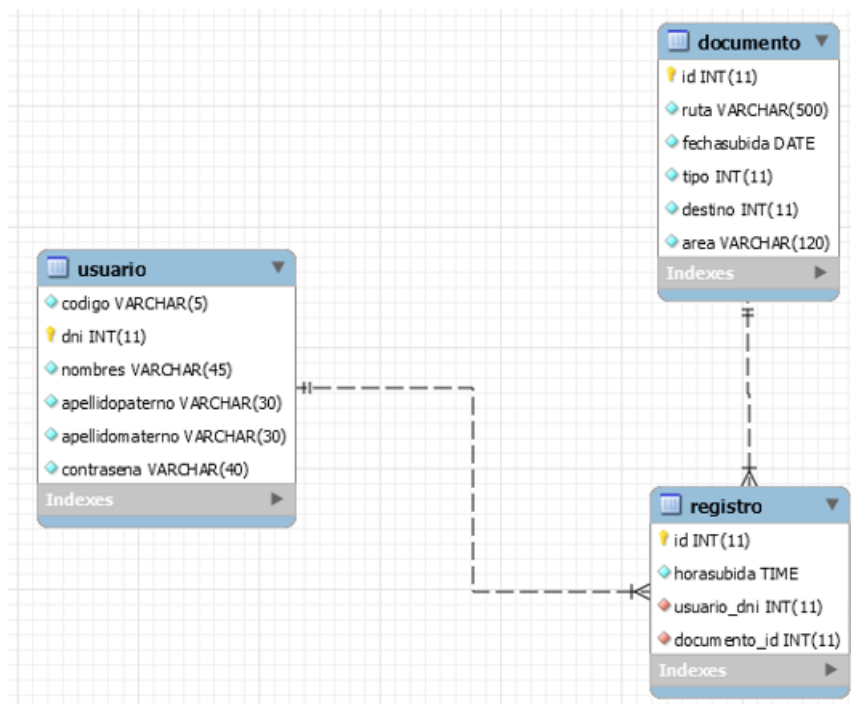


Figura 3.8: Diagrama de la base de datos de entidad relación.

Fuente: Elaboración propia.

3.2.7. Diseño de interfaces

A continuación se muestran las interfaces de la aplicación:

En la Figura 3.9, se muestra el login tanto para el administrador como para los usuarios.



Figura 3.9: Login de la aplicación.

Fuente: Elaboración propia.

En la Figura 3.10, se muestra todas las opciones que tiene el administrador después de haber ingresado su login.



Figura 3.10: Interfaz principal.

Fuente: Elaboración propia.

En la Figura 3.11, se muestra la interfaz de cargar los documentos luego de dar click en la opción de cargar documento de la Figura 3.10.



Figura 3.11: Interfaz de la carga de los documentos.

Fuente: Elaboración propia.

En la Figura 3.12, se muestra una interfaz para seleccionar los múltiples documentos a subir.

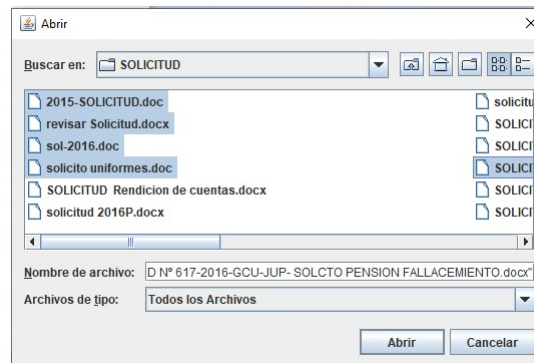


Figura 3.12: Interfaz de la búsqueda múltiple de documentos.

Fuente: Elaboración propia.

En la Figura 3.13, se muestra el reporte de los documentos cargados en la aplicación, listos para su selección y muestra en cualquiera de los formatos (.PDF, .DOCX o .DOC).



Figura 3.13: Interfaz de reporte de los documentos.

Fuente: Elaboración propia.

3.2.8. Diseño de algoritmo

El siguiente diagrama de flujo en el cual mostramos el proceso de implementación computacional que seguimos para el desarrollo.

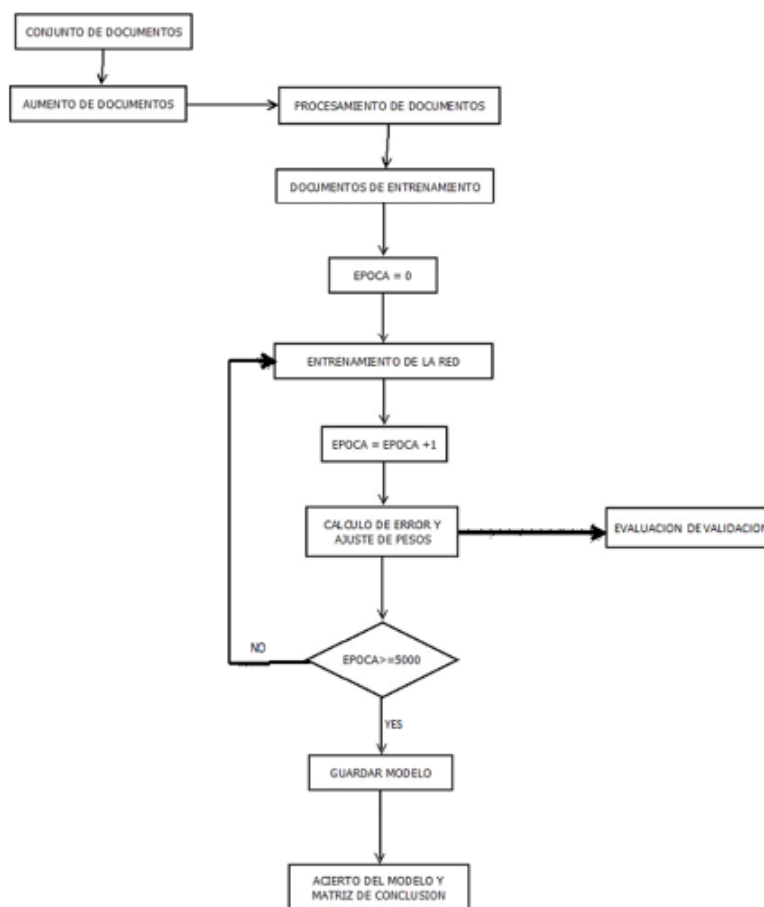


Figura 3.14: Diagrama de flujo para el desarrollo de la investigación.

Fuente: Elaboración propia.

3.2.9. Extracción de datos

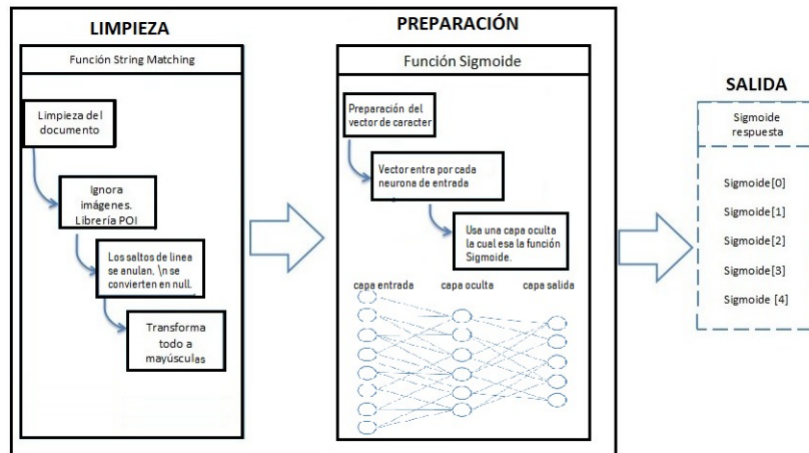


Figura 3.15: Extracción de datos.

Fuente: Elaboración propia.

Limpieza: Extracción de los caracteres necesarios para formar el vector. Con la librería POI eliminamos todas las imágenes y encabezados (marca de agua) del documento, para poder llegar al texto a utilizar. El algoritmo convierte el salto de línea en null, así evitamos espacios en blanco. El algoritmo convierte todas las letras a mayúsculas para una mejor lectura.

Preparación: Se prepara el vector de carácter, el vector entra por cada neurona de entrada. Usa una capa oculta, la cual con la función sigmoide nos da una respuesta y la más aproximada a uno nos dará como respuesta el tipo de documento.

Salida: La función sigmoide nos da como respuesta un valor, para sigmoide[0] el tipo de

documento asignado es informe, para sigmoide[1] el tipo de documento asignado es solicitud, para sigmoide[2] el tipo de documento asignado es memorandum, para sigmoide[3] el tipo de documento asignado es oficio y para sigmoide[4] el tipo de documento asignado es resolución rectoral.

Informe	Solicitud	Memorandum	Oficio	Resolución Rectoral	Multiple	Pedido	Reclamo
1	0	0	0	0	0	1	0

Figura 3.16: Descripción del patrón.

Fuente: Elaboración propia.

3.2.10. Arquitectura de la aplicación

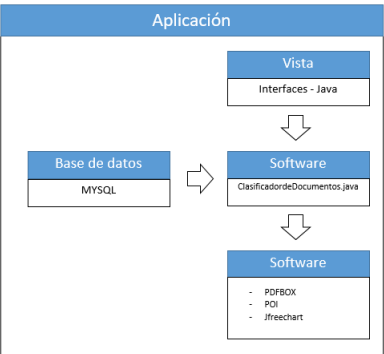


Figura 3.17: Arquitectura de la aplicación.

Fuente: Elaboración propia.

3.3. Implementación

Nos enfocamos en la resolución de la tesis y el desarrollo del sistema inteligente, un software realizado en java en la plataforma de Netbeans 8.1, como gestor de datos utilizaremos Workbench MySQL y Joone como diseñador de la red neuronal, el cual nos mostrará los resultados óptimos de nuestra clasificación de documentos a través de nuestra aplicación. Detallamos cómo y qué conceptos tuvimos en cuenta para el desarrollo de la aplicación, mostramos también los procesos de implementación computacional más importantes que fueron realizados durante el desarrollo de la investigación.

3.4. Entrenamiento

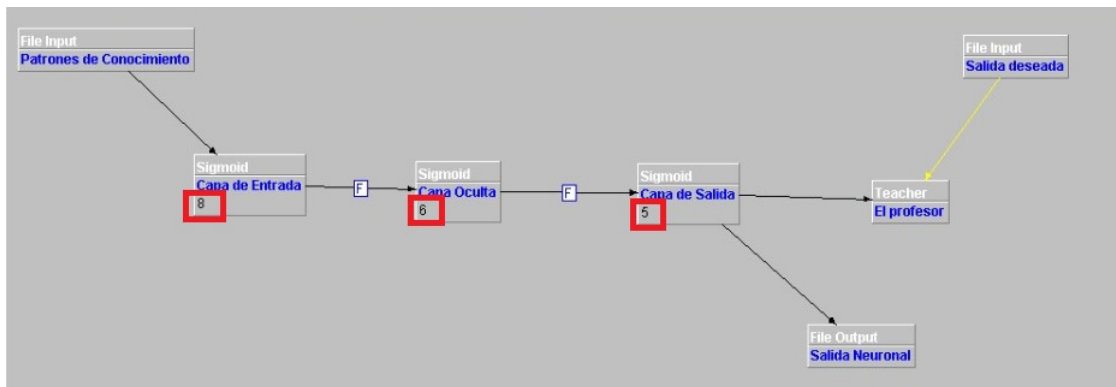


Figura 3.18: Red neuronal en Joone.

Fuente: Elaboración propia.

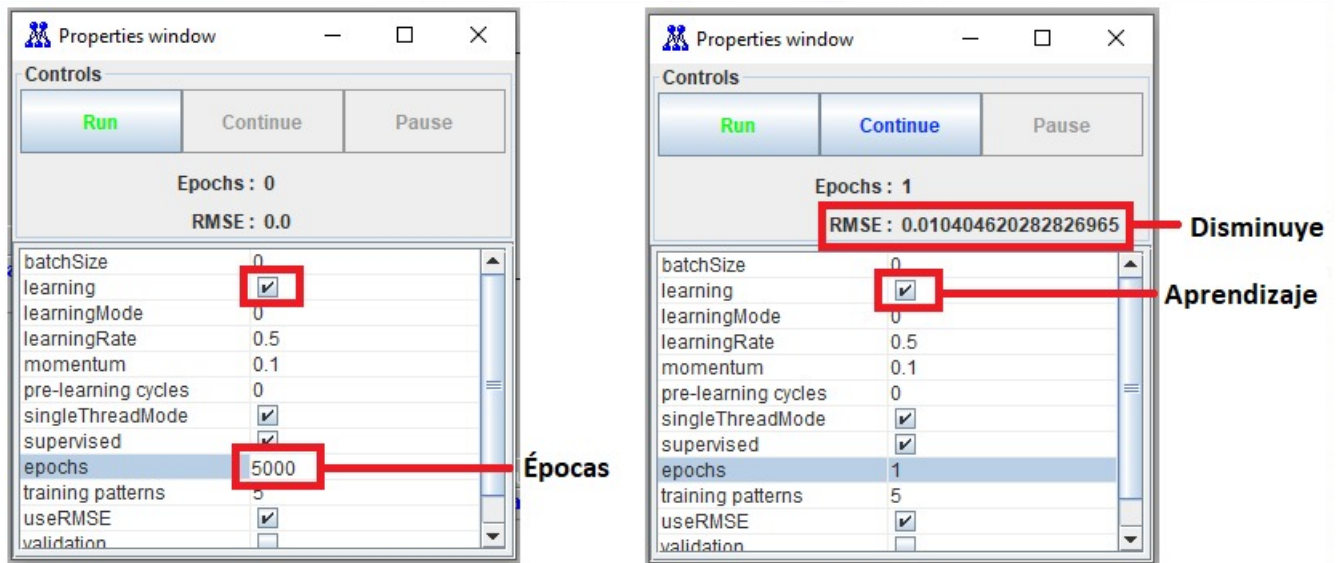


Figura 3.19: Entrenamiento de las 5000 épocas.

Fuente: Elaboración propia.

El cual se hizo con un total de 5000 épocas.

Los datos utilizados fueron extraídos de diversas oficinas de la Universidad Nacional de Trujillo, los cuales conforman la Dataset.

Tabla 3.2: Datos del entrenamiento

Función	Cantidad	Tipo
Entrenamiento	400 documentos	No especificados, tomados solo para esta función
Validación	100 documentos	Oficio = 20 Solicitud = 20 Memorandum = 20 Informe = 20 Resolución rectoral = 20
Muestra	134 documentos	Oficio = 68 Solicitud = 22 Memorandum = 6 Informe = 24 Resolución rectoral = 14

Fuente: Elaboración propia.

Las siguientes tablas muestran nuestras pruebas en el desarrollo de la red.

5.98110098	-4.9181375	-3.2990561	-0.0048796	0.17507329
-3.8951318	-0.0406367	-1.9321498	4.70906128	-1.0464711
-0.9862924	-0.1359934	3.87899577	-6.1465544	-1.0262063
-2.0577522	6.15774414	-3.6682604	-3.2249468	0.40149572
-1.6702981	1.41010313	5.65123345	-3.076806	-4.8624826
6.23790148	-1.151774	-4.9715872	-1.6116612	-0.1031716
-1.3900638	-6.9478075	-0.4391444	-1.4193106	4.76703362
0.18448666	-0.6588261	2.09502072	3.47731552	-7.1811308
-5.7543816	4.1316186	-5.6743162	0.85131516	3.56747955
-0.9523734	-1.2967779	0.40931399	3.95046364	-3.5268163

Figura 3.20: Pesos de salida.

Fuente: Elaboración propia.

Tabla 3.3: Vector de aprendizaje.

Vectores de aprendizaje							
Informe	Solicitud	Memorandum	Oficio	Resolución	Múltiple	Pedido	Reclamo
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0

Fuente: Elaboración propia.

5.98110098	-4.9181375	-3.2990561	-0.0048796	0.17507329
-3.8951318	-0.0406367	-1.9321498	4.70906128	-1.0464711
-0.9862924	-0.1359934	3.87899577	-6.1465544	-1.0262063
-2.0577522	6.15774414	-3.6682604	-3.2249468	0.40149572
-1.6702981	1.41010313	5.65123345	-3.076806	-4.8624826
6.23790148	-1.151774	-4.9715872	-1.6116612	-0.1031716
-1.3900638	-6.9478075	-0.4391444	-1.4193106	4.76703362
0.18448666	-0.6588261	2.09502072	3.47731552	-7.1811308
-5.7543816	4.1316186	-5.6743162	0.85131516	3.56747955
-0.9523734	-1.2967779	0.40931399	3.95046364	-3.5268163

Figura 3.21: Pesos de salida.

Fuente: Elaboración propia.

-0.4386899
-0.5184561
-0.5417869
-0.0894253
-0.4165299
-0.6130218
-0.1370983
-0.0689692
-0.2013986
-0.5464507
-0.4939044
-0.5663579

Figura 3.22: Bias.

Fuente: Elaboración propia.

3.5. Funcionamiento de las redes neuronales

Al cargar un documento se va ingresando el vector característico y se interpreta en la red neuronal. Este es un caso de cómo la red neuronal va a ir funcionando con este documento. Los vectores característicos van a responder cuando se activen y estos van a dar lugar a activar la neurona.

Funciones:

Una de las principales funciones es “CargarDocumentos.java” la cual usa un componente que le permite una selección múltiple, una ventaja del software es que recibe documentos que estén guardados en diferentes formatos, con la librería POIFS nos permite reconocer las extensiones .doc, .docx., .pdf, de la misma forma algún documento que se ingrese en otro tipo de formato o ya sea como imagen (.jpg, .jpeg, .png, .gif) lo leerá como invalido.

La función BuscarDocumentos, se desarrolló buscando las características que tiene un documento para clasificarlo como un tipo específico.

En este caso el documento por evaluar será oficio, la neurona en la capa de entrada se activa cuando una parte contiene la palabra solicitud y al mismo tiempo petición.

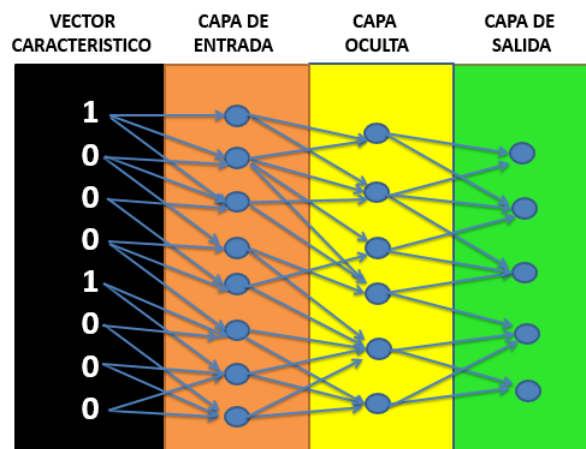


Figura 3.23: Inicio de vector característico.

Fuente: Elaboración propia.

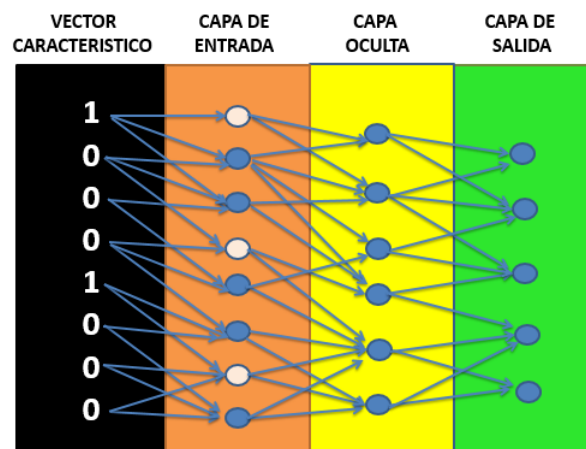


Figura 3.24: Activando neuronas en la capa de entrada.

Fuente: Elaboración propia.

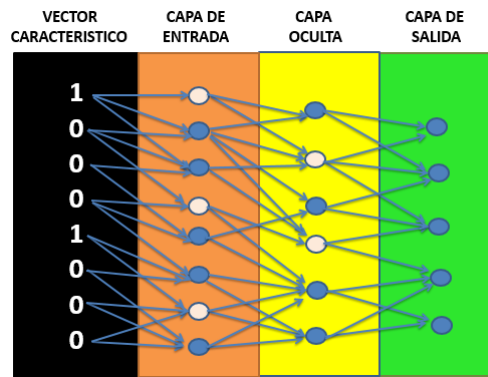


Figura 3.25: Activando neuronas en la capa oculta.

Fuente: Elaboración propia.

Si se activan esos unos, se activan en la capa de entrada esas neuronas.

Y al final en la capa de salida se activa solo una y da la respuesta, cada neurona de salida nos da una respuesta diferente, en este caso la segunda neurona nos da como respuesta una solicitud.

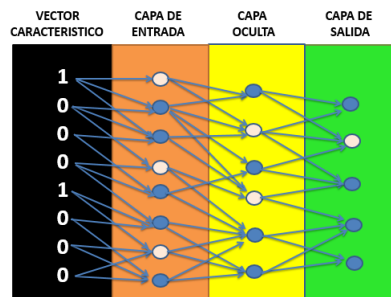


Figura 3.26: Activando neuronas en la capa de salida.

Fuente: Elaboración propia.

3.6. Función activación

La neurona no sólo transmite la entrada que recibe, la función de activación utiliza la suma ponderada de la entrada anterior y la transforma como salida. La función sigmoide denotada por `funcionSigmoidea`, donde se denota la constante exponencial, que es aproximadamente 2.71828.

En nuestra aplicación la función sigmoide nos mostrará el resultado de nuestra búsqueda dando numéricamente la opción que más se acerca al resultado que queremos llegar. En este caso “Solicitud” es el resultado que más se aproxima a 1.

```
Sigmoidal [0]- 0.012531487731386662
Sigmoidal [1]- 0.9959445628183717
Sigmoidal [2]- 0.0020733062955850773
Sigmoidal [3]- 0.007473006424699547
Sigmoidal [4]- 0.0015958380984010267
SOLICITUD
INSERT INTO documento(ruta, fechasubida, tipo, destino, area) VALUES('F:/ANTHONY-2020-NO-BORRAR/'
```



Figura 3.27: Resultado de la función sigmoide en la aplicación.

Fuente: Elaboración propia.

En la Tabla 3.3 nos muestra cuales son los vectores.

Capítulo 4

Resultados y discusión de la tesis

En la aplicación del sistema inteligente conseguimos resultados esperados que van de acuerdo con la hipótesis planteada.

4.1. Resultados computacionales

Con los indicadores de la Tabla: 2.2, se obtuvo los siguientes resultados:

- **Verdadero positivo:** documentos correctos encontrados.
- **Falso positivo:** documentos incorrectos encontrados.
- **Verdadero negativo:** documentos correctos no encontrados.

- **Falso negativo:** documentos incorrectos no encontrados.

4.1.1. Resultado según criterio subir con éxito

A continuación se muestra una tabla:

Tabla 4.1: Tabla de valores predictivos.

	Verdadero (V)	Falso (F)
Positivo (P)	VP	FP
Negativo (N)	VN	FN

Fuente: Elaboración propia.

Donde:

Verdadero, es subir con éxito.

Positivo, es encontró el documento.

Falso, es no subir con éxito.

Negativo, es no encontró el documento.

	OFICIO	INFORME	MEMORANDUM	RESOLUCIÓN RECTORAL	SOLICITUD	OTROS	TOTAL
OFICIO	66	0	0	0	0	0	66
INFORME	0	23	0	0	0	1	24
MEMORANDUM	0	0	4	0	0	0	4
RESOLUCIÓN RECTORAL	0	0	0	11	0	3	14
SOLICITUD	0	0	0	0	19	3	22
OTROS	0	0	0	0	0	4	4
TOTAL	66	23	4	11	19	11	134

Figura 4.1: Resultado detallado para la búsqueda, según criterio subir con éxito

Fuente: Elaboración propia.

Tabla cruzada Subir con éxito *Encontró el doc.

Recuento

		Encontró el doc.		Total
		0	1	
Subir con éxito	0	7	0	7
	1	4	123	127
Total		11	123	134

Figura 4.2: Tabla cruzada: subir con éxito vs. encontró el doc.

Fuente: Elaboración propia.

$$\frac{VP}{VP + FN} = \frac{123}{123 + 7} = \frac{123}{130} = 0,946, \quad 94,6 \% \text{ de sensibilidad.} \quad (4.1)$$

$$\frac{VP}{VP + FP} = \frac{123}{123 + 0} = \frac{123}{123} = 1, \quad 100 \% \text{ de precisión.} \quad (4.2)$$

$$\frac{VP + VN}{TOTAL} = \frac{123 + 4}{134} = \frac{127}{134} = 0,947, \quad 94,7 \% \text{ de exactitud.} \quad (4.3)$$

En los resultados de criterio: Subir con éxito el documento, se obtuvo en el indicador de sensibilidad un 94.6 %, de precisión un 100 % y en exactitud un 94.7 %. Por lo tanto, consideramos nuestra hipótesis como válida.

4.1.2. Resultado según criterio sin error ortográfico

En la tabla de valores predictivos (ver Tabla: 4.1), se detalla a continuación los valores para el criterio sin error ortográfico:

Verdadero, es subir con éxito.

Positivo, es encontró el documento.

Falso, es no subir con éxito.

Negativo, es no encontró el documento.

	OFICIO	INFORME	MEMORANDUM	RESOLUCIÓN RECTORAL	SOLICITUD	OTROS	TOTAL
OFICIO	67	0	0	0	0	0	67
INFORME	0	23	0	0	0	1	24
MEMORANDUM	0	0	4	0	0	2	6
RESOLUCIÓN RECTORAL	0	0	0	11	0	3	14
SOLICITUD	0	0	0	0	19	3	22
OTROS	0	0	0	0	0	1	1
TOTAL	66	23	4	11	19	11	134

Figura 4.3: Resultado detallado para la búsqueda según criterio sin error ortográfico.

Fuente: Elaboración propia.

Tabla cruzada Doc. Sin error ortográfico*Encontró el doc.

Recuento

		Encontró el doc.		Total
		0	1	
Doc. Sin error ortográfico	0	10	1	11
	1	1	122	123
Total		11	123	134

Figura 4.4: Tabla cruzada: sin error ortográfico vs. encontró el doc.

Fuente: Elaboración propia.

$$\frac{VP}{VP + FN} = \frac{122}{122 + 10} = \frac{122}{132} = 0,9242, \quad 92,42 \% \text{ de sensibilidad.} \quad (4.4)$$

$$\frac{VP}{VP + FP} = \frac{122}{122 + 1} = \frac{122}{123} = 0,9918, \quad 99,18 \% \text{ de precisión.} \quad (4.5)$$

$$\frac{VP + VN}{TOTAL} = \frac{122 + 1}{134} = \frac{123}{134} = 0,917, \quad 91,7 \% \text{ de exactitud.} \quad (4.6)$$

En los resultados del criterio: Sin error ortográfico, se obtuvo en el indicador de sensibilidad un 92.42 %, de precisión un 99.18 % y en el de exactitud un 91.7 %. Por lo tanto consideramos nuestra hipótesis como válida.

4.2. Tiempo

En esta parte se miden los resultados que obtuvimos con un grupo de 30 trabajadores, antes y después de utilizar la aplicación.

Tabla 4.2: Tabla base de comparación de tiempos

Tiempo de búsqueda sin la aplicación		Tiempo de búsqueda con la aplicación	
Pasos	Tiempo (segundos)	Pasos	Tiempo (segundos)
Ingresar a (MiPC o Este Equipo)	60	Abrir la aplicación	2.5
Buscar documentos	120	Cargar documentos	2
Abrir el supuesto documento que necesitas	60	Abrir el reporte y ver documento buscado	2
Repetir paso 3 hasta encontrar el documento buscado	300 a más	Abrir documento	0.3

Fuente: Elaboración propia.

Con la Tabla 4.2 se obtuvieron los resultados (ver Tabla: A.2) de los cuales se sacó una

media aritmética de aproximadamente 760.2 segundos, de los cuales 22 trabajadores demoraban aproximadamente 600 segundos en buscar un documento y 8 trabajadores demoraban aproximadamente 1200 segundos, lo que hace un total de 22800 segundos para 30 trabajadores

Se aplicó la Tabla: 4.2 y los resultados (ver Tabla: A.3) fueron muy favorables con un total de 16 personas que tardaron aproximadamente 5 segundos en la búsqueda de documentos y 14 personas que tardaron aproximadamente 10 segundos, dándonos un total de 220 segundos para 30 trabajadores, lo cual nos da un media de 7.33 segundos, con este resultado mostramos la optimización de tiempo al usar la aplicación.

Tabla cruzada APROX 10 MIN*APROX 20 MIN

Recuento

		APROX 20 MIN		
		0	1	Total
APROX 10 MIN	0	0	8	8
	1	22	0	22
Total		22	8	30

Figura 4.5: Trabajadores antes de usar la aplicación.

Fuente: Elaboración propia.

Estos resultados nos muestran que se llegó al objetivo, cumpliendo con optimizar uno de los principales recursos, el tiempo. Con esto concluimos logrando todos los objetivos planteados para nuestra tesis.

Tabla cruzada APROX 5 seg*APROX 10 seg

Recuento

		APROX 10 seg		
		0	1	Total
APROX 5 seg	0	0	14	14
	1	16	0	16
Total		16	14	30

Figura 4.6: Trabajadores usando la aplicación.

Fuente: Elaboración propia.

Tabla 4.3: Resultados con tiempos, sin usar la aplicación y usando la aplicación

Sin aplicación	Con aplicación
760.2 segundos	7.33 segundos

Fuente: Elaboración propia.

Logrando reducir el tiempo de 760.2 segundos a 7.33 segundos, esto significa un ahorro de tiempo en más de 752 segundos en la búsqueda por documento.

Capítulo 5

Consideraciones finales

5.1. Conclusiones

Al finalizar la investigación se pudo cumplir con los objetivos específicos propuestos, los cuales se detallan a continuación:

1. Durante esta investigación se pudo explicar la problemática de la demora en la búsqueda de documentos así como la solución aplicando un software la cual contiene como principal mecanismo el uso de los sistemas inteligentes, en este caso se determinó usar redes neuronales, perceptrón y backpropagation, la cual nos permitió establecer un lapso de tiempo muy reducido al momento de la búsqueda de los documentos.
2. Se cumplió con la implementación de un prototipo para la clasificación de documentos,

la cual es muy sencilla y útil para el usuario de cualquier oficina.

3. En la investigación se logró una sensibilidad del 94.6 %, precisión del 100 % y exactitud del 94.7 % en el criterio de subir con éxito y encontró el documento.
4. En la investigación se logró una sensibilidad del 92.42 %, precisión del 99.18 % y exactitud del 91.7 % en el criterio de documento sin error ortográfico y encontró el documento.
5. Se optimizó reducir el tiempo de 760.2 segundos a 7.33 segundos, esto significa un ahorro de tiempo en más de 752 segundos en la búsqueda por documento.

5.2. Trabajos futuros

Si bien esta investigación cumple con la hipótesis y objetivos específicos planteados, durante la presente tesis, se presentaron algunas ideas que podrían mejorar la aplicación, por ejemplo:

- Diseñar más a detalle la jerarquía de las oficinas de la Universidad Nacional de Trujillo, lo cual se podría llegar hasta cada escuela, como sus jerarquías internas dentro de ellas, creando una clasificación mas minuciosa.
- Proponer diseñar una red local en todas las oficinas de la Universidad Nacional de

Trujillo para su clasificación de los documentos de manera que permita mejorar aun más los tiempos de emisión y recepción de documentos a diferentes oficinas.

- Crear un app móvil y de esta forma tener un alcance a más usuarios.



UNIVERSIDAD NACIONAL DE TRUJILLO

DECLARACION JURADA RR-384-2018/UNT

Los autores suscritos en el presente documento DECLARAMOS BAJO JURAMENTO que somos los responsables legales de la calidad y originalidad del contenido del Proyecto de Investigación Científica, así como, del Informe de Investigación Científica realizado.

Titulado: Aplicación de sistemas inteligentes para la
clasificación automática de documentos

PROYECTO DE INVESTIGACIÓN CIENTÍFICA

PROY DE TRABAJO DE INVESTIGACIÓN (PREGRADO) ()
PROYECTO DE TESIS PREGRADO ()
PROYECTO DE TESIS MAESTRÍA ()
PROYECTO DE TESIS DOCTORADO ()

INFORME FINAL DE INVESTIGACIÓN CIENTÍFICA

TRABAJO DE INVESTIGACIÓN (PREGRADO) ()
TESIS PREGRADO (X)
TESIS MAESTRIA ()
TESIS DOCTORADO ()

El equipo investigador Integrado por:

N°	APELLIDOS Y NOMBRES	FACULTAD/DEPARTAMENTO	CATEGORIA DOCENTE ASESOR	CÓDIGO Docente Numero Matricula del estudiante	Autor Coautor Asesor
01	Contreras Panamango Luis Anthony	FCFYM		052701510	Autor
02	Sanchez Rodriguez Paul Denis	FCFYM		012701110	Autor

Trujillo, 20 de noviembre de 2019

FIRMA

FIRMA

FIRMA

64

DNI

DNI

DNI

70493427

46795930



UNIVERSIDAD NACIONAL DE TRUJILLO

CARTA DE AUTORIZACIÓN DE PUBLICACIÓN DE TRABAJO DE INVESTIGACIÓN EN REPOSITORIO DIGITAL RENATI-SUNEDU RR-384-2018/UNT

Trujillo, 20 de noviembre de 2019.

Los autores suscritos del INFORME FINAL DE INVESTIGACIÓN CIENTÍFICA

Titulado: Aplicación de sistemas inteligentes para la
clasificación automática de documentos

AUTORIZAMOS SU PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL, REPOSITORIO RENATI-SUNEDU, ALICIA-CONCYTEC CON EL SIGUIENTE TIPO DE ACCESO:

- A. Acceso Abierto: ☒
B. Acceso Restringido ☐ (datos del autor y resumen del trabajo)
C. No autorizo su Publicación ☐

Si eligió la opción restringido o NO autoriza su publicación sírvase justificar:

ESTUDIANTES DE PREGRADO: TRABAJO DE INVESTIGACIÓN

☐ TESIS ☒

ESTUDIANTES DE POSGRADO: TESIS MAESTRIA ☐

TESIS DOCTORADO ☐

DOCENTES: INFORME DE INVESTIGACIÓN

☐

El equipo investigador Integrado por:

N°	APELLIDOS Y NOMBRES	FACULTAD	CONDICIÓN DOCENTE (NOMBRADO, CONTRATADO, EMÉRITO, estudiante, OTROS)	CÓDIGO Docente Numero Matricula del estudiante	Autor Coautor Asesor
01	Contreras Romarango Luis Anthony	FCFYM		052701510	Autor
02	Sanchez Rodriguez Paul Denis	FCFYM		012701110	Autor
			65		

FIRMA

DNI

FIRMA

DNI

FIRMA

DNI