

UNIVERSIDAD NACIONAL DE TRUJILLO

Facultad de Ciencias Físicas y Matemáticas

Escuela Profesional de Informática



**Sistema Basado en Conocimiento Mediante RNA para
Pronósticar las Ventas del Restaurante Combiche 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

2021

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

LOPEZ ANDRADE ALEX MICHAEL ANDRE

**SISTEMA BASADO EN CONOCIMIENTO MEDIANTE
RNA PARA PRONÓSTICAR LAS VENTAS DEL
RESTAURANTE COMBICHE EN EL MALL AVENTURA
EN TRUJILLO**

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

2021

III

HOJA DE APROBACIÓN

Sistema Basado en Conocimiento Mediante RNA para Pronósticar las Ventas del Restaurante Combiche en el Mall Aventura en Trujillo

Lopez Andrade Alex Michael Andre

Tesis defendida y aprobada por el jurado examinador:

Prof. Mg. Bravo Escalante Jorge David - Asesor
Departamento de Informática - UNT

Prof. Dr. Jorge Luis Gutierrez Gutierrez
Departamento de Informática - UNT

Prof. Mg. José Arturo Díaz Pulido
Departamento de Informática - UNT

Trujillo, 2021

Dedico esta tesis a:

A Dios, por siempre cuidarme y bendecirme en cada paso que doy.

A mis abuelos, Ruperto López y Maria de los Angeles Marin por encaminarme en el sendero del bien.

A mis Padres, por siempre estar presente en cada momento de la vida brindandome su apoyo incondicional.

A mi Novia, Jhanyra Chacon por ser el motivo de querer superarme profesionalmente hacia un futuro mejor.

Lopez Andrade Alex Michael Andre

Agradecimientos

Agradezco a:

Mis Padres, por su apoyo y consejo en este proceso de la investigación.

Mi Novia, Jhanyra Chacon por impulsarme en culminar esta etapa de mi vida.

A mi compañeros Paul Rodriguez y Pedro Alvarez por su asesoria en este proyecto de investigación.

A mis primos Ivan López y Segundo Novoa por su apoyo desde que inicio mi vida universitaria.

A la Milo por alegrarme todos los días.

Lopez Andrade Alex Michael Andre

Resumen

Actualmente, el área de de inteligencia artificial se aplica a múltiples actividades empresariales desde el área de automatización hasta pronósticos para la toma de decisiones, el área de ventas en las empresas se apoyan en tecnologías informáticas para el almacenamiento de información de manera que se puede determinar sí van aumentando o disminuyendo en determinado tiempo.

El presente trabajo de información brinda un sistema de conocimiento basado en Redes Neuronales que ha partir de la información sobre ventas que se tiene, pueda pronosticar el futuro de estas. Se diseñó el sistema de conocimiento basado para pronosticar ventas usando como herramienta principal las redes neuronales.

Implementamos una aplicación para acceder utilizando una interfaz gráfica que nos permite entrenar nuestra red neuronal y pronosticar futuras ventas. Se concluyó que el sistema logra pronosticar eficientemente las ventas en el restaurante Combiche del mall Aventura Plaza en Trujillo.

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

Abstract

Actually, the artificial intelligence is applied to multiple business activities from the area. Since automation to forecasting for decision making, the sales area in companies rely on computer technologies for storing information so that they can determine if they are increasing or decreasing in a certain time.

This information work provides a knowledge system based on Neural Networks that has from the information on sales that you have, can forecast the future of these. The knowledge-based system was designed to forecast sales using neural networks as the main tool.

I implement an application to access using a graphical interface that allows us to train our neural network and forecast future sales. It was concluded that the system manages to efficiently forecast sales at the Combiche restaurant in the Aventura Plaza Mall in Trujillo.

Keywords: neural networks, , intelligent systems.

Índice de figuras

2.1. Red neuronal artificial.	12
2.2. Gráfico de la función sigmoide.	15
2.3. Representación de un red multicapa.	19
2.4. Ecuaciones del backpropagation.	21
3.1. Requerimiento funcional de investigación.	31
3.2. Modelo de diagrama de clases para la investigación de la tesis.	32
3.3. Modelo de estados para la investigación de la tesis.	33
3.4. Modelo de casos de uso para la investigación de la tesis.	33
3.5. Modelo de casos de uso para la investigación de la tesis.	34
3.6. Diagrama de componentes de nuestra investigación.	35
3.7. Modelo de secuencia para clasificar automáticamente documentos.	36
3.8. Ventana Principal del Sistema.	37
3.9. Diagrama de flujo para el desarrollo de la investigación.	39
3.10. Red Neuronal utilizada en la investigación.	40

3.11. Red Neuronal utilizada en la investigación.	41
3.12. Extracción de datos.	42
3.13. Descripción del patrón.	43
3.14. Arquitectura de la aplicación.	43
4.1. Contrastación por método de Chi-Cuadrado en Geogebra.	76
B.1. Sistema de Ventas de Combiche del cual se obtuvo el dataset.	111

Índice de tablas

2.1. Componentes del perceptrón.	17
3.1. Requerimiento no funcional del sistema.	32

Índice general

Dedicatoria	I
Agradecimientos	II
Resumen	III
Abstract	IV
Índice de Figuras	VI
Índice de Tablas	VII
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. Inteligencia Artificial (IA)	7
2.1.2. Sistemas inteligentes	7
2.1.2.1. Capacidades requeridas	9
2.1.3. Red Neuronal Artificial (RNA)	10
2.1.4. Tipos de aprendizaje	11
2.1.4.1. Aprendizaje supervisado	12
2.1.5. Elementos básicos	12
2.1.5.1. Función de entrada (input function)	13
2.1.5.2. Función de activación (activation function)	13
2.1.5.3. Función de salida (output function)	16
2.1.6. Perceptrón	16
2.1.6.1. Perceptrón simple	17
2.1.6.2. Perceptrón Multicapa	18
2.1.7. Backpropagation	20

2.1.8.	Definición de regresión	22
2.1.8.1.	Fases del análisis de regresión múltiple	22
2.1.9.	Python	24
2.1.10.	TensorFlow	24
2.2.	Método de la investigación	25
2.2.1.	Tipo de investigación	25
2.2.2.	Variables de la Investigación	25
2.2.2.1.	Variable Dependiente	25
2.2.2.2.	Variable Independiente	26
2.2.3.	Operacionalización de la variable	26
2.3.	Recolección de datos para la elaboración del modelo	26
2.3.1.	Técnica	26
2.3.2.	Población	26
2.3.3.	Muestra	27
2.4.	Etapas de la investigación	27

3.	Sistema Basado en Conocimiento Mediante RNA para Pronósticar las Ventas	
	del Restaurante Combiche en el Mall Aventura	29
3.1.	Análisis	30
3.1.1.	Requerimiento funcional	30

3.1.2. Requiemiento no funcional	31
3.2. Diseño	32
3.2.1. Diagrama de clases	32
3.2.2. Diagrama de modelos de estados	33
3.2.3. Diagrama de casos de uso	33
3.2.4. Diagrama de casos de uso del negocio	34
3.2.5. Diagrama de componentes	35
3.2.6. Diagrama de secuencia	36
3.2.7. Diseño de interfaces	37
3.2.8. Diseño de algoritmo	38
3.2.9. Diseño de Red Neuronal	40
3.2.10. Extracción de datos	42
3.2.11. Arquitectura de la aplicación	43
3.3. Implementación	43
3.4. Funcionamiento de las redes neuronales	44
3.5. Función activación	45
3.6. Funciones del sistema:	46
3.7. Código Fuente:	46
4. Resultados y discusión de la tesis	74

4.1. Resultados computacionales	74
4.2. Discusión de Resultados	75
5. Consideraciones finales	77
5.1. Conclusiones	77
5.2. Trabajos futuros	78
Referencias bibliográficas	79
A. Dataset de entrenamiento	81
B. Segundo apendice	110

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 RUP 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 los resultados obtenidos con el software.

1.5. Estructura de la tesis

En presente 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 el vector característico, la extracción de datos y por último como conceptos finales, consideramos a los gestores, plataformas y lenguajes de programación que utilizaremos así como Visual Studio Code, Tensorflow, Python.
- 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. 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. (Prince, 2009).

Según McCarthy (1989), acuñó la expresión “inteligencia artificial”, y la definió como: “la ciencia e ingenio de hacer máquinas inteligentes, especialmente programas de cómputo inteligentes”.

2.1.2. 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 múlti-

ples 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. (Cruz, 2012).

2.1.2.1. Capacidades requeridas

Según Cruz (2012), 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 oca-

siones, 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.

2.1.3. 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. (Freeman and Skapura, 1991)

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.

2.1.4. 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. (Matich, 2001):

2.1.4.1. 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. (Matich, 2001).

2.1.5. Elementos básicos

A continuación se puede ver, en la Figura 2.1, un esquema de una red neuronal:

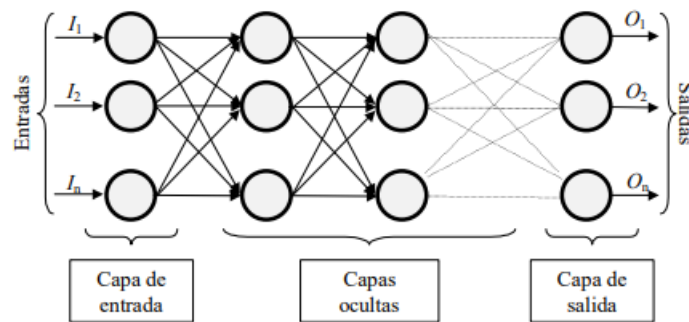


Figura 2.1: Red neuronal artificial.

Fuente: Erlinb (2017).

Esta red neuronal está constituida por neuronas interconectadas y arregladas en tres ca-

pas. 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.

2.1.5.1. 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. (Matich, 2001).

2.1.5.2. 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. (Matich, 2001).

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. (Matich, 2001).

- 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.

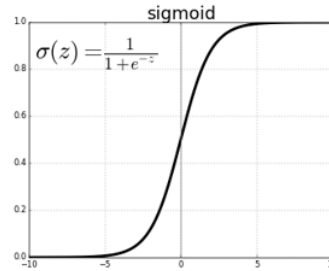


Figura 2.2: Gráfico de la función sigmoide.

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.

2.1.5.3. 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 $-1, 1$. (Matich, 2001).

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.

2.1.6. 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. (Espitia Corredor, Daniel Eduardo, 2017).

2.1.6.1. 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 Espitia Corredor, Daniel Eduardo (2017).

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.

2.1.6.2. 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 2.3). El tamaño de éstas dependerán de la dificultad de la correspondencia a implementar. (Munt, 2018).

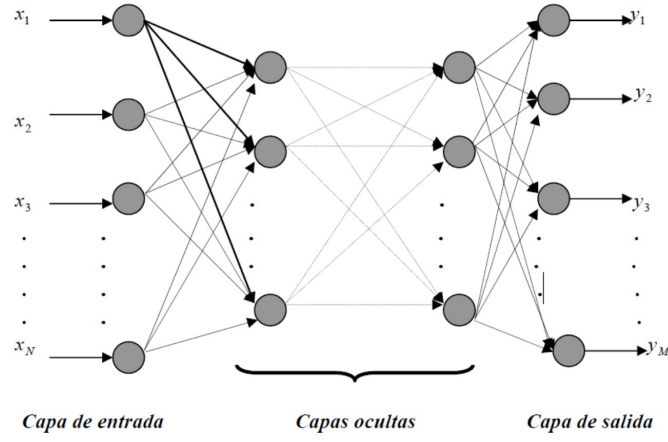


Figura 2.3: Representación de un red multicapa.

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:

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

2.1.7. 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 2.4: Ecuaciones del backpropagation.

Fuente: Michael A. Nielsen (2015).

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.

2.1.8. Definición de regresión

Según Pat Fernández et al. (2013), 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

2.1.8.1. Fases del análisis de regresión múltiple

Según Pat Fernández et al. (2013), 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.

2.1.9. 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" y 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. (Rossum, 1995).

2.1.10. 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.. (Dillon et al., 2017).

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.

2.3. Recolección de datos para la elaboración del modelo

2.3.1. Técnica

Observación, fue la técnica aplicada para la recolección de datos, específicamente la observación estructurada. Indicando la cantidad de ventas realizadas en la fecha respectiva.

2.3.2. Población

La población para nuestra investigación son las ventas del 2018 hasta el 2020 en el Restaurante Combiche en el Mall Aventura Plaza - Trujillo.

2.3.3. Muestra

Este conjunto de datos está conformado por un dataset de 968 registros, que representan las ventas durante el período enero del 2018 hasta agosto del 2020. Se optó por usar un muestreo aleatorio simple con un nivel de confianza del 95 % y un error de muestreo del 5 %.

$$n = \frac{z^2 pq}{e^2} \quad (2.7)$$

En el que:

- n = Tamaño de muestra.
- z = Coeficiente de confiabilidad 95 % al que corresponde (1.96).
- pq = Varianza de la población, ponemos la varianza mayor posible porque a mayor varianza hará falta una muestra mayor (0.25).
- e = Error muestral 5 % (0.5).

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 RUP.
- 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 con la información del dataset.

Capítulo 3

Sistema Basado en Conocimiento

Mediante RNA para Pronósticar las

Ventas del Restaurante Combiche 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 del sistema.

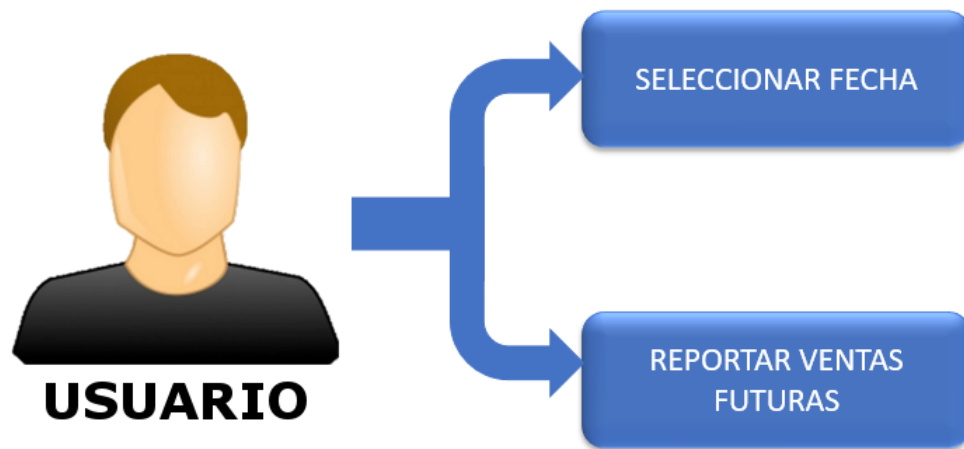


Figura 3.1: Requerimiento funcional de investigación.

Fuente: Elaboración propia.

Usuario:

- Seleccionar la fecha a pronósticar.
- Reportar futuras ventas gráficamente.

3.1.2. Requerimiento no funcional

Describe otras prestaciones, características y/o limitaciones del sistema.

Tabla 3.1: Requerimiento no funcional del sistema.

Requerimiento no funcional	Descripción
Usabilidad	Aplicación de fácil manejo, con interfaz gráfica sencilla.
Eficiencia	Aplicación eficiente, ligera y con respuesta rápida.
Dependibilidad	Aplicación disponible. Aplicación confiable. Aplicación integro, sin alteraciones. Fácil mantenimiento de la aplicación.

Fuente: Elaboración propia.

3.2. Diseño

3.2.1. Diagrama de clases

Modelo el cual seguiremos en el desarrollo del sistema.

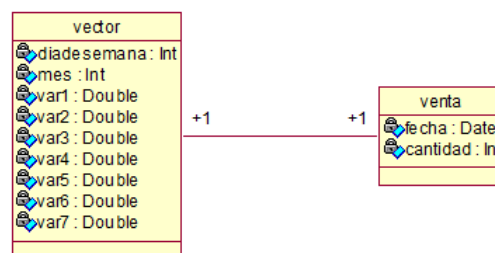


Figura 3.2: Modelo de diagrama de clases para la investigación de la tesis.

Fuente: Elaboración propia.

3.2.2. Diagrama de modelos de estados

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

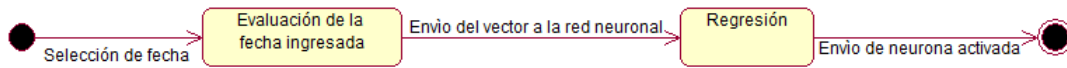


Figura 3.3: Modelo de estados para la investigación de la tesis.

Fuente: Elaboración propia.

3.2.3. Diagrama de casos de uso

Modelo de caso de uso.

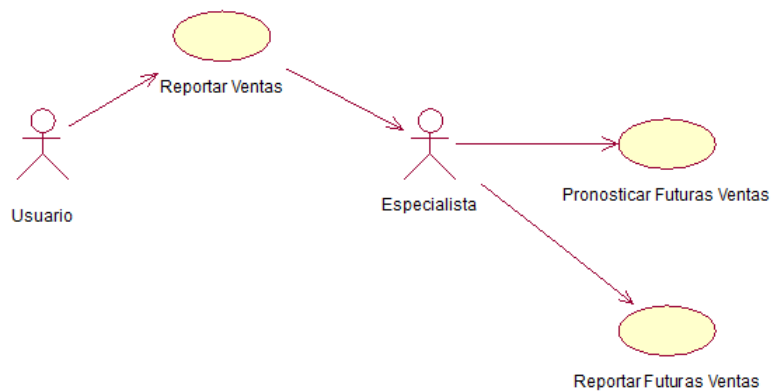


Figura 3.4: Modelo de casos de uso para la investigación de la tesis.

Fuente: Elaboración propia.

- Reportar Ventas, el usuario reporta las ventas a un especialista.
- Pronosticar Futuras Ventas, el especialista pronostica las futuras ventas a partir de los reportes.
- Reportar Futuras ventas, el especialista brinda el pronóstico de las futuras ventas en un reporte.

3.2.4. Diagrama de casos de uso del negocio

Modelo de caso de uso de negocio.

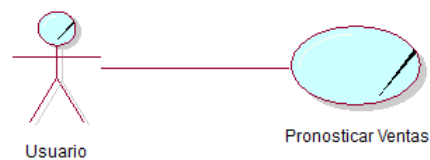


Figura 3.5: Modelo de casos de uso para la investigación de la tesis.

Fuente: Elaboración propia.

- Pronosticar Ventas, El usuario pronostica directamente las futuras ventas mediante un sistema basado en conocimiento.

3.2.5. Diagrama de componentes

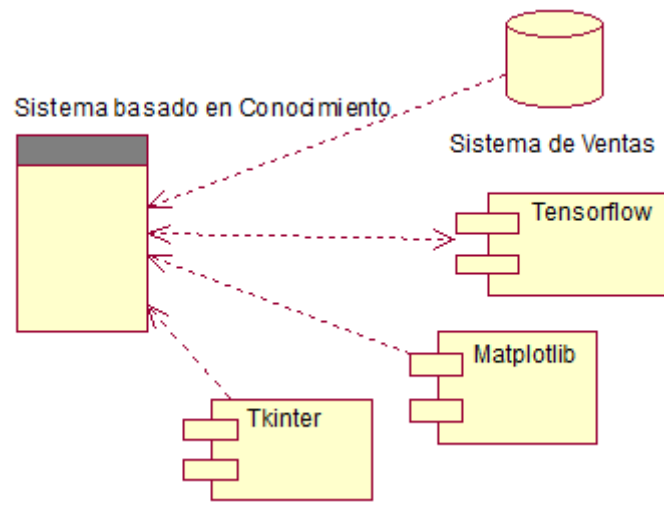


Figura 3.6: Diagrama de componentes de nuestra investigación.

Fuente: Elaboración propia.

3.2.6. Diagrama de secuencia

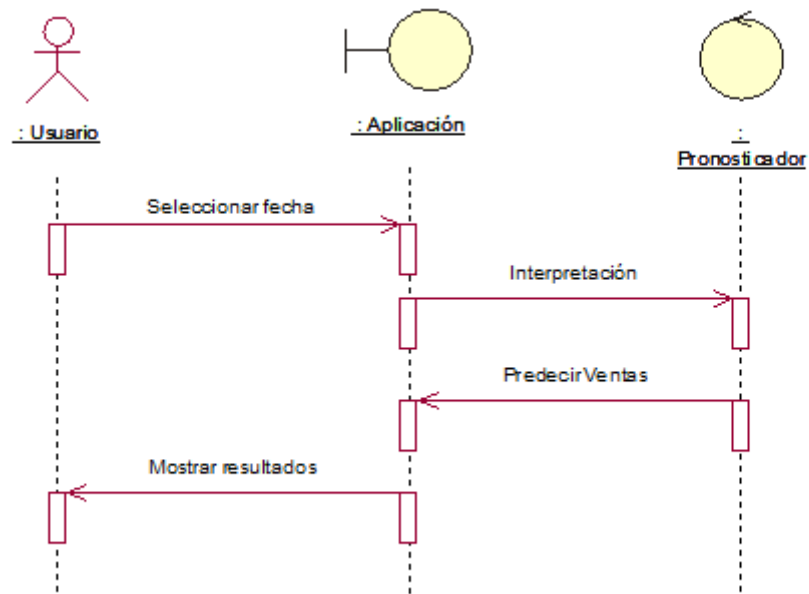


Figura 3.7: Modelo de secuencia para clasificar automáticamente documentos.

Fuente: Elaboración propia.

3.2.7. Diseño de interfaces

A continuación se muestran la interfaz del sistema:

En la Figura 3.8, se muestra la ventana principal.

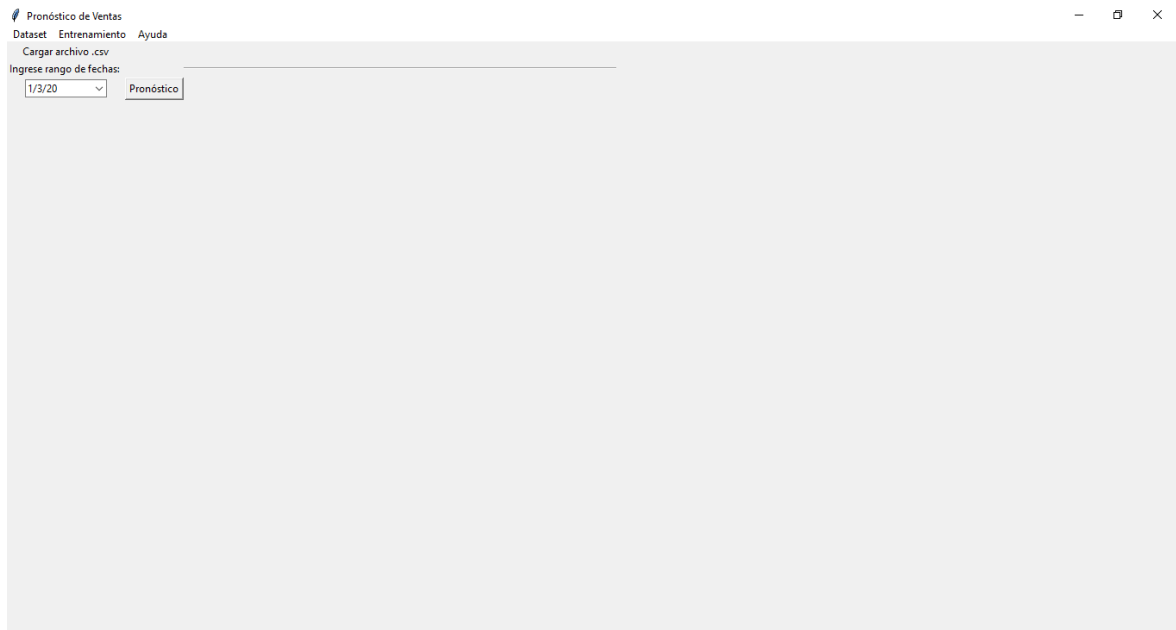


Figura 3.8: Ventana Principal del Sistema.

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.

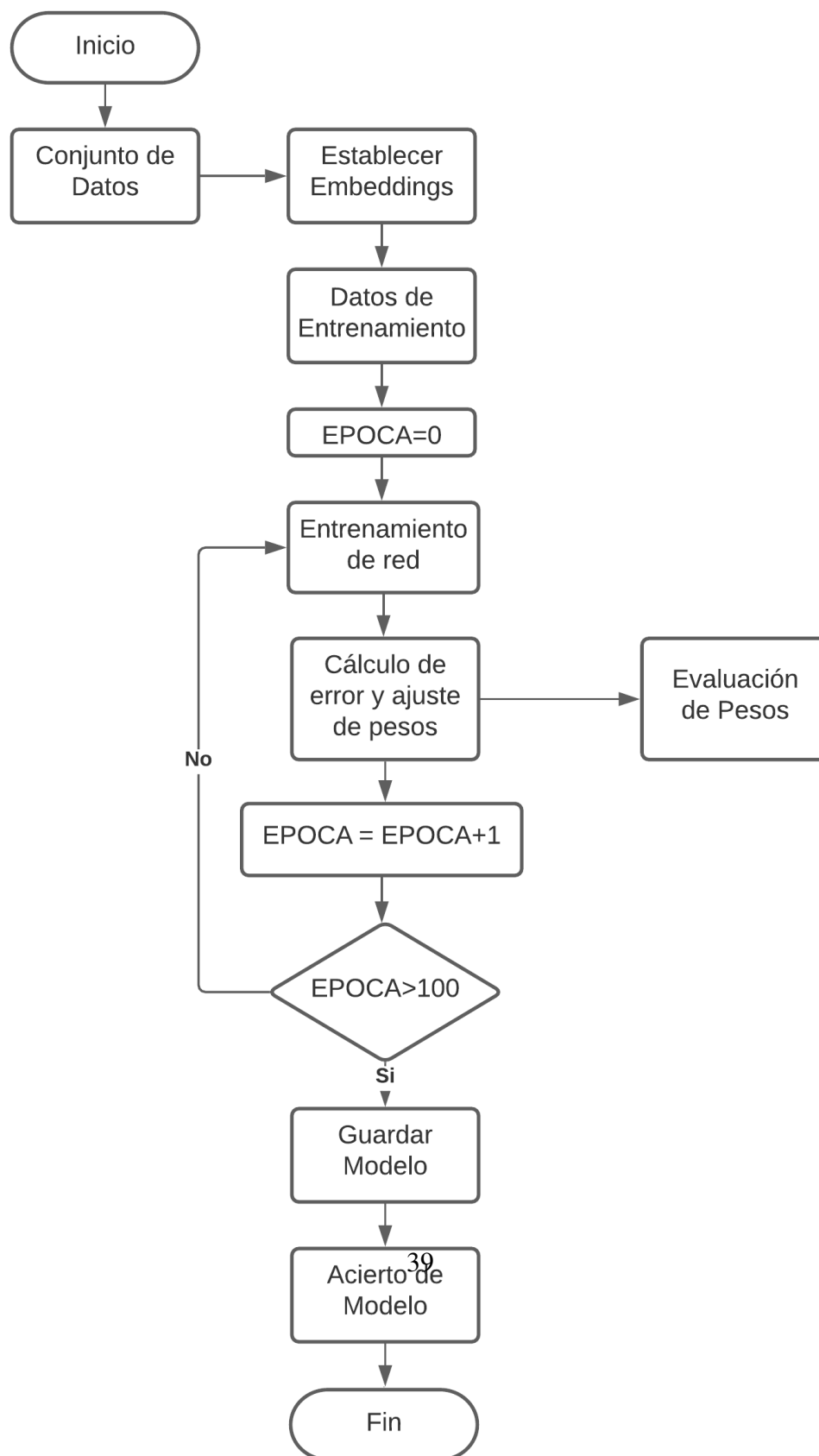


Figura 3.9: Diagrama de flujo para el desarrollo de la investigación.

Fuente: Elaboración propia.

3.2.9. Diseño de Red Neuronal

El siguiente gráfico es el modelo de neurona utilizada en la red neuronal de la investigación.

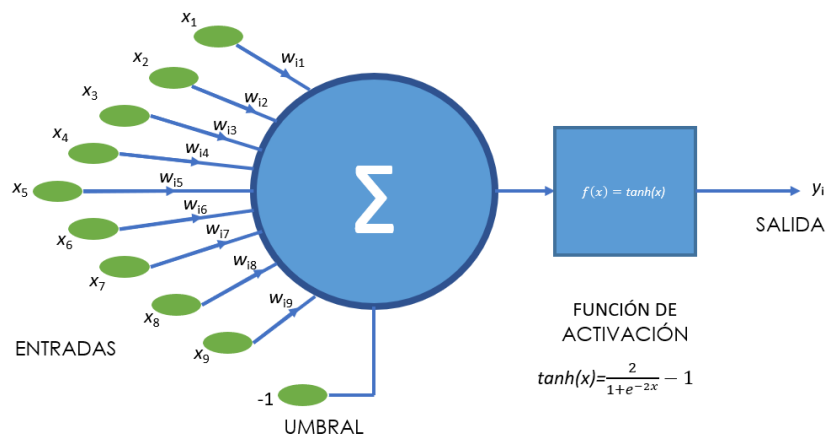


Figura 3.10: Red Neuronal utilizada en la investigación.

Fuente: Elaboración propia.

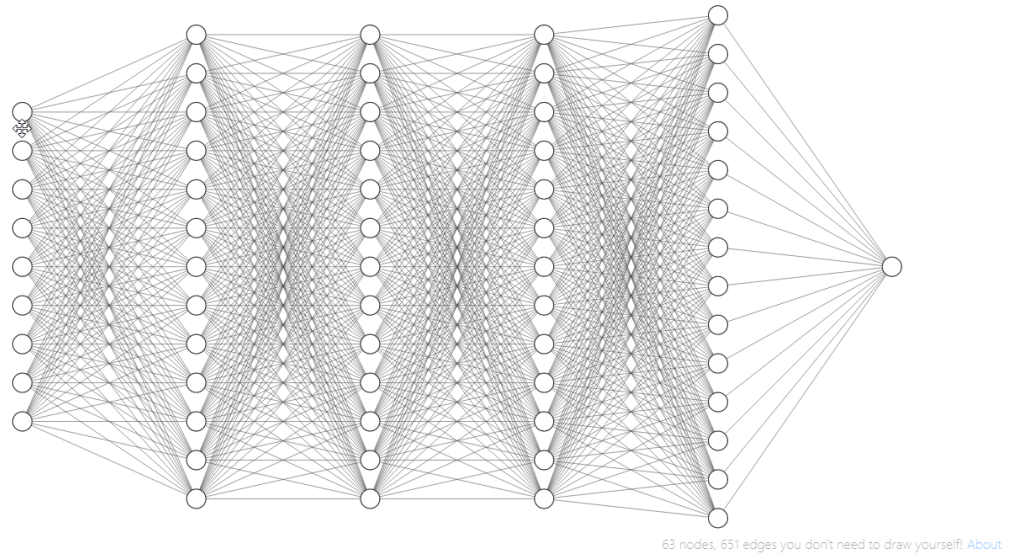


Figura 3.11: Red Neuronal utilizada en la investigación.

Fuente: Elaboración propia.

3.2.10. Extracción de datos

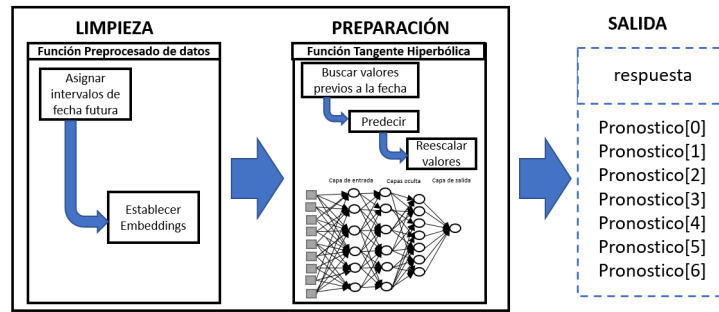


Figura 3.12: Extracción de datos.

Fuente: Elaboración propia.

Preprocesado: Selección de las fechas previas(15 días) a la fecha que se quiere pronosticar, de esta manera se tiene los valores de ventas previas a dicha semana, se prepara las fechas futuras(1 semana) a partir de la fecha seleccionada.

Preparación: Usando los valores de ventas y fechas se le establece sus respectivos valores de Embeddings, se re-escalan los valores para luego predecir en la red neuronal pre-entrenada y se re-escala los valores obtenidos.

Salida: La función tangente hiperbólica nos da como respuesta un vector, el cual indica los valores de las ventas en la próxima semana iniciando en la fecha indicada.

En este vector se puede visualizar que los primeros dos valores son extraídos del día de la semana y el mes de la fecha, mientras que los otros siete indican los valores de las fechas siguiente pero re-escalado.

5	1	12	8	3	19	7	10	15
---	---	----	---	---	----	---	----	----

Figura 3.13: Descripción del patrón.

Fuente: Elaboración propia.

3.2.11. Arquitectura de la aplicación

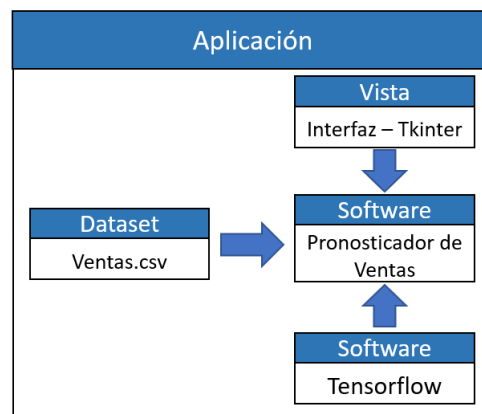


Figura 3.14: Arquitectura de la aplicación.

Fuente: Elaboración propia.

3.3. Implementación

Nos enfocamos en la resolución de la tesis y el desarrollo del sistema basado en conocimiento, un software realizado en python independiente de la IDE, usando tensorflow para el

entrenamiento y predicción de la red neuronal feedforward, el cual nos mostrará los resultados óptimos de nuestra pronóstico a través de nuestro sistema.

En el proceso de entrenamiento de la red neuronal se hizo un total de 100 épocas. Los datos utilizados fueron extraídos de la base de datos del restaurante Combiche en el Mall Aventura y almacenados en un dataset (en formato .csv).

3.4. Funcionamiento de las redes neuronales

Al seleccionar la fecha, se establece vectores de fechas previas para predecir la semana siguiente, se establecen embeddings y se interpreta en la red neuronal. Los vectores que se obtienen en la predicción son re-escalados para poder entenderlos.

3.5. Función activación

Las redes neuronales feedforward tienen la característica de ir hacia adelante, no presentan bucles. La función de activación nos indica el valor de venta de la fecha siguiente a las fechas ingresadas. Considerando que dicho valor aún necesita re-escalarsse.

3.6. Funciones del sistema:

Una de las funciones es mostrar gráficamente la pérdida(valores de la función de coste con los datos de entrenamiento).

Además, reporta gráficamente los valores de testing y el pronóstico para las fechas a probar. Principalmente, muestra gráficamente los valores del pronóstico de la semana de la fecha seleccionada.

3.7. Código Fuente:

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4  import matplotlib.pyplot as plt
5  from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
6  from matplotlib.figure import Figure
7
8  import pandas as pd
9  import numpy as np
10 import matplotlib.pyplot as plt
11 import matplotlib
12
```

```

13 plt.rcParams['figure.figsize'] = (16, 9)
14 plt.style.use('fast')
15
16 from keras.models import Sequential, load_model
17 from keras.layers import Dense, Activation, Flatten
18 from sklearn.preprocessing import MinMaxScaler
19
20 from keras.layers import Input, Embedding, Dense, Flatten, Dropout,
    concatenate, LSTM
21 from keras.layers import BatchNormalization, SpatialDropout1D
22 from keras.callbacks import Callback
23 from keras.models import Model
24 from keras.optimizers import Adam
25
26
27 import tkinter as tk
28 from tkinter.filedialog import askopenfilename
29 from tkinter.messagebox import showerror
30 from tkinter import ttk
31 from tkinter import HORIZONTAL
32 from tkinter import StringVar, Label, Button
33 from tkcalendar import Calendar, DateEntry

```



```

34
35 import time
36 import datetime
37 from datetime import date , timedelta
38 import matplotlib.pyplot as plt
39 import openpyxl
40 import xlswriter
41 import os.path
42 from os import path
43 from shutil import copyfile
44
45
46 class Application(tk.Frame):
47     def __init__(self , parent , *args , **kwargs):
48         tk.Frame.__init__(self , parent , *args , **kwargs)
49         self.parent = parent
50         self.fileSelect = ""
51         self.parent.title(" Pron stico de Ventas")
52         self.menubar = tk.Menu(parent)
53         self.filemenu = tk.Menu(self.menubar , tearoff = 0)
54         self.filemenu.add_command(label = "Cargar data", command = self.
            selectFile)

```

```

55
56     self.filemenu.add_separator()
57     #self.filemenu.add_command(label = "Guardar reporte", command =
        self.saveReport)
58     #self.filemenu.add_separator()
59     #self.filemenu.add_command(label = "Guardar model", command =
        self.saveModel)
60     #self.filemenu.add_separator()
61     self.filemenu.add_command(label = "Salir", command = root.quit)
62     self.menubar.add_cascade(label = "Dataset", menu = self.filemenu)
63     self.editmenu = tk.Menu(self.menubar, tearoff=0)
64
65     self.editmenu.add_command(label = "Entrena Red Neuronal", command
        = self.entrenamiento)
66
67     self.menubar.add_cascade(label = "Entrenamiento", menu = self.
        editmenu)
68
69     self.helpmenu = tk.Menu(self.menubar, tearoff=0)
70     self.helpmenu.add_command(label = "Acerca de")
71     self.menubar.add_cascade(label = "Ayuda", menu = self.helpmenu)
72

```

```

73         self.parent.config(menu = self.menubar)
74
75         self.text1 = StringVar()
76         self.text1.set('Cargar archivo .csv')
77
78         self.texto1 = ttk.Label(self.parent, textvariable=self.text1)
79         self.texto1.grid(row=0)
80
81         self.labelR = Label(self.parent, text="Ingrese rango de fechas: ")
82         self.labelR.grid(row=1, column=0)
83
84         self.cal1 = DateEntry(self.parent, dateformat=3,width=12,
85                                background='darkblue',foreground='white', borderwidth=4,year
86                                =2020,month=3,day=1)
87
88         self.cal1.grid(row=2, column=0)
89
90         self.cal2 = DateEntry(self.parent, dateformat=3,width=12,
91                                background='darkblue',foreground='white', borderwidth=4,year
92                                =2020,month=2,day=29)
93
94         self.cal2.grid(row=1, column=2)

```

```

90         self.butInter = Button(self.parent, text="Pron stico", command
           = self.graficar_predicciones)
91     self.butInter.grid(row=2, column=1)
92
93     ttk.Separator(self.parent, orient=HORIZONTAL).grid(row=1, column
           =3, columnspan=4, ipadx=250)
94
95     self.lf = ttk.Labelframe(self.parent, text='Ventas')
96     self.lf.grid(row=3, column=0, sticky='nwes', padx=3, pady=3)
97
98     def saveReport(self):
99         if path.exists('report.png'):
100             workbook = xlswriter.Workbook('reporte.xlsx')
101             worksheet = workbook.add_worksheet()
102             workbook.close()
103             wb = openpyxl.load_workbook('reporte.xlsx')
104             ws = wb.active
105
106             img = openpyxl.drawing.image.Image('report.png')
107             #img.anchor(ws.cell('A1'))
108
109             ws.add_image(img)

```

```

110         wb.save('reporte.xlsx')
111
112
113     def saveModel(self):
114         print("GUARDADO")
115
116     def graficar_predicciones(self):
117         ##INTERPRETACION
118
119         for widget in self.lf.wininfo_children():
120             widget.destroy()
121
122         PASOS = 7
123         fechaIni = str(self.cal1.get_date() - datetime.timedelta(days=15)
124             )
125         fechaFin = str(self.cal1.get_date() - datetime.timedelta(days=1))
126         #fechaFin = str(self.cal2.get_date())
127         print(fechaFin)
128         print(fechaIni)
129         self.df = pd.read_csv('temp.csv', parse_dates=[0], header=None,
130             index_col=0, names=['fecha', 'unidades'])
131         self.df['weekday']=[x.weekday() for x in self.df.index]

```

```

130     self.df['month']=[x.month for x in self.df.index]
131     ultimosDias = self.df[fechaIni:fechaFin]
132     print("ULTIMOS DIAS")
133     print(ultimosDias)
134     values = ultimosDias['unidades'].values
135
136     values = values.astype('float32')
137     scaler = MinMaxScaler(feature_range=(-1, 1))
138
139     values=values.reshape(-1, 1)
140     scaled = scaler.fit_transform(values)
141
142     reframed = self.series_to_supervised(scaled, PASOS, 1)
143     reframed.reset_index(inplace=True, drop=True)
144
145     contador=0
146     reframed['weekday']=ultimosDias['weekday']
147     reframed['month']=ultimosDias['month']
148
149     for i in range(reframed.index[0], reframed.index[-1]):
150         reframed['weekday'].loc[contador]=ultimosDias['weekday'][i+8]
151         reframed['month'].loc[contador]=ultimosDias['month'][i+8]

```

```

152         contador=contador+1
153     reframed.head()
154
155     reordenado=reframed[ ['weekday','month','var1(t-7)','var1(t-6)','
        var1(t-5)','var1(t-4)','var1(t-3)','var1(t-2)','var1(t-1)'] ]
156     reordenado.dropna(inplace=True)
157     values = reordenado.values
158     x_test = values[5:,:]
159     x_test = x_test.reshape((x_test.shape[0], 1, x_test.shape[1]))
160
161
162     ultDiaSemana = reordenado.weekday[len(reordenado)-1]
163
164     def agregarNuevoValor(x_test,nuevoValor,ultDiaSemana):
165         for i in range(x_test.shape[2]-3):
166             x_test[0][0][i+2] = x_test[0][0][i+3]
167             ultDiaSemana=ultDiaSemana+1
168             if ultDiaSemana>6:
169                 ultDiaSemana=0
170                 x_test[0][0][0]=ultDiaSemana
171                 x_test[0][0][1]=12
172                 x_test[0][0][x_test.shape[2]-1]=nuevoValor

```

```

173         return x_test , ultDiaSemana
174
175     results=[]
176     if not path.exists("model.h5"):
177         print("No existe el modelo pre-entrenado")
178         return False
179
180     self.model = load_model('model.h5')
181
182     for i in range(7):
183         dia=np.array([x_test[0][0][0]])
184         mes=np.array([x_test[0][0][1]])
185         valores=np.array([x_test[0][0][2:9]])
186         parcial=self.model.predict([dia , mes, valores])
187         results.append(parcial[0])
188         print('pred',i , x_test)
189         x_test , ultDiaSemana=agregarNuevoValor(x_test , parcial[0] ,
190             ultDiaSemana)
191
192     adimen = [x for x in results]
193     inverted = scaler.inverse_transform(adimen)
194     #inverted

```



```

194     prediccionProxSemana = pd.DataFrame(inverted)
195     prediccionProxSemana.columns = ['pronostico']
196     prediccionProxSemana.plot()
197     prediccionProxSemana.to_csv('pronostico.csv')
198
199     predicciones = prediccionProxSemana.to_numpy()
200     y_values=[]
201     for i in range(len(predicciones)):
202         y_values.append(predicciones[i][0])
203
204     x_values = []
205     for i in range(len(predicciones)):
206         x_values.append(datetime.datetime.strptime(str(self.call.
                get_date() + datetime.timedelta(days=i)), "%Y-%m-%d").date
                ())
207
208     fig = plt.figure(figsize=(5,5))
209     dates = matplotlib.dates.date2num(x_values)
210     matplotlib.pyplot.plot_date(dates, y_values)
211     plt.plot(x_values, y_values)
212     plt.gcf().autofmt_xdate()
213     plt.title('Predicci n de la semana')

```

```

214         print(prediccionProxSemana)
215         canvas = FigureCanvasTkAgg(fig , master=self.lf)
216         canvas.draw()
217         canvas.get_tk_widget().grid(row=3, column=0)
218
219     def selectFile(self):
220         fname = askopenfilename(filetypes=(("Archivo Dataset", "*.csv"),
221                                           ("Todos los archivos", "*.*")
222                                           ))
223
224         if fname:
225             try:
226                 self.text1.set(fname)
227                 self.fileSelect = fname
228             except:
229                 print("")
230
231     def series_to_supervised(self,data , n_in=1, n_out=1, dropnan=True):
232         n_vars = 1 if type(data) is list else data.shape[1]
233         df = pd.DataFrame(data)
234         cols , names = list() , list()
235         # input sequence (t-n, ... t-1)
236         for i in range(n_in , 0 , -1):

```

```

235         cols.append(df.shift(i))
236         names += [( 'var %d(t-%d)' % (j+1, i)) for j in range(n_vars)]
237     # forecast sequence (t, t+1, ... t+n)
238     for i in range(0, n_out):
239         cols.append(df.shift(-i))
240         if i == 0:
241             names += [( 'var %d(t)' % (j+1)) for j in range(n_vars)]
242         else:
243             names += [( 'var %d(t+%d)' % (j+1, i)) for j in range(
                n_vars)]
244     # put it all together
245     agg = pd.concat(cols, axis=1)
246     agg.columns = names
247     # drop rows with NaN values
248     if dropnan:
249         agg.dropna(inplace=True)
250     return agg
251
252 def agregarNuevoValor(self, x_test, nuevoValor):
253     for i in range(x_test.shape[2]-3):
254         x_test[0][0][i+2] = x_test[0][0][i+3]
255     ultDiaSemana=ultDiaSemana+1

```

```

256         if ultDiaSemana>6:
257             ultDiaSemana=0
258         x_test[0][0][0]=ultDiaSemana
259         x_test[0][0][1]=12
260         x_test[0][0][x_test.shape[2]-1]=nuevoValor
261         return x_test ,ultDiaSemana
262
263     def crear_modeloEmbeddings(self):
264         PASOS=7
265         emb_dias = 2
266         emb_meses = 4
267
268         in_dias = Input(shape=[1], name = 'dias')
269         emb_dias = Embedding(7+1, emb_dias)(in_dias)
270         in_meses = Input(shape=[1], name = 'meses')
271         emb_meses = Embedding(12+1, emb_meses)(in_meses)
272
273         in_cli = Input(shape=[PASOS], name = 'cli')
274
275         fe = concatenate([(emb_dias), (emb_meses)])
276
277         x = Flatten()(fe)

```

```

278     x = Dense(PASOS, activation='tanh')(x)
279     outp = Dense(1, activation='tanh')(x)
280     model = Model(inputs=[in_dias, in_meses, in_cli], outputs=outp)
281
282     model.compile(loss='mean_absolute_error',
283                   optimizer='adam',
284                   metrics=['MSE'])
285
286     model.summary()
287     return model
288
289 def entrenamiento(self):
290     if "csv" in self.fileSelect:
291         self.df = pd.read_csv(self.fileSelect, parse_dates=[0],
292                               header=None, index_col=0, names=['fecha', 'unidades'])
293         copyfile(self.fileSelect, './temp.csv')
294         self.df.head()
295
296         self.df['weekday']=[x.weekday() for x in self.df.index]
297         self.df['month']=[x.month for x in self.df.index]
298         self.df.head()
299         self.df.describe()

```

```

299
300
301     print( self . df . head () )
302     PASOS=7
303     values = self . df [ ' unidades ' ] . values
304
305     values = values . astype ( ' float32 ' )
306     scaler = MinMaxScaler ( feature_range = ( -1 , 1 ) )
307
308     values = values . reshape ( -1 , 1 )
309
310     scaled = scaler . fit_transform ( values )
311
312     reframed = self . series_to_supervised ( scaled , PASOS , 1 )
313     reframed . reset_index ( inplace = True , drop = True )
314
315     contador=0
316     reframed [ ' weekday ' ] = self . df [ ' weekday ' ]
317     reframed [ ' month ' ] = self . df [ ' month ' ]
318
319     for i in range ( reframed . index [ 0 ] , reframed . index [ -1 ] ) :
320         reframed [ ' weekday ' ] . loc [ contador ] = self . df [ ' weekday ' ] [ i + 8 ]

```

```

321         reframed['month'].loc[contador]=self.df['month'][i+8]
322         contador=contador+1
323     reframed.head()
324
325     reordenado=reframed[ ['weekday','month','var1(t-7)','var1(t
        -6)','var1(t-5)','var1(t-4)','var1(t-3)','var1(t-2)','var1
        (t-1)','var1(t)'] ]
326     reordenado.dropna(inplace=True)
327
328     training_data = reordenado.drop('var1(t)',axis=1)#.values
329     target_data=reordenado['var1(t)']
330     #training_data.head()
331     valid_data = training_data[len(values)-30:len(values)]
332     valid_target=target_data[len(values)-30:len(values)]
333
334     training_data = training_data[0:len(values)-30]
335     target_data=target_data[0:len(values)-30]
336     print(training_data.shape , target_data.shape , valid_data.shape ,
        valid_target.shape)
337
338
339     EPOCHS=100

```

```

340
341     self.model = self.crear_modeloEmbeddings()
342
343     continuas=training_data[['var1(t-7)', 'var1(t-6)', 'var1(t-5)',
344                             'var1(t-4)', 'var1(t-3)', 'var1(t-2)', 'var1(t-1)']]
345
346     valid_continuas=valid_data[['var1(t-7)', 'var1(t-6)', 'var1(t-5)',
347                                 'var1(t-4)', 'var1(t-3)', 'var1(t-2)', 'var1(t-1)']]
348
349     history=self.model.fit([training_data['weekday'],
350                             training_data['month'], continuas], target_data, epochs=
351                             EPOCHS, validation_data=(valid_data['weekday'], valid_data[
352                             'month'], valid_continuas, valid_target))
353
354
355     self.model.save('model.h5')
356
357     results=self.model.predict([valid_data['weekday'], valid_data[
358                                 'month'], valid_continuas])
359
360     plt.show()
361
362     fig = plt.figure(figsize=(6, 5))
363
364     plt.plot(history.history['loss'], label='loss')
365
366     plt.title('loss')
367
368     plt.plot(history.history['val_loss'], label='val loss')
369
370     plt.title('validate loss')

```



```

356         plt.legend(loc='best')
357     #plt.show()
358     canvas = FigureCanvasTkAgg(fig, master=self.lf)
359     canvas.draw()
360     canvas.get_tk_widget().grid(row=3, column=1)
361     print("VALIDATE ")
362     compara = pd.DataFrame(np.array([valid_target, [x[0] for x in
        results]])).transpose()
363     compara.columns = ['real', 'prediccion']
364
365     inverted = scaler.inverse_transform(compara.values)
366
367     compara2 = pd.DataFrame(inverted)
368     compara2.columns = ['real', 'prediccion']
369     compara2['diferencia'] = compara2['real'] - compara2['
        prediccion']
370
371     print(compara2['real'])
372
373     x_vals = []
374     x_vals = self.df.index[len(values)-30:len(values)].to_numpy()
375     x_values = []

```

```

376         for x in x_vals:
377             x_values.append(str(datetime.datetime.strptime(str(x)
378                                     [:10], '%Y-%m-%d').strftime('%d-%m-%Y')))
379
380         print(x_values)
381
382         fig = plt.figure(figsize=(6, 5))
383
384         compara2['real'].plot()
385
386         #plt.plot(x_values, compara2['real'][0])
387
388         compara2['prediccion'].plot()
389
390         #plt.plot(x_values, compara2['prediccion'][0])
391
392         canvas = FigureCanvasTkAgg(fig, master=self.lf)
393
394         canvas.draw()
395
396         canvas.get_tk_widget().grid(row=3, column=2)
397
398
399 if __name__ == "__main__":
400     root = tk.Tk()
401
402     w, h = root.winfo_screenwidth(), root.winfo_screenheight()
403
404     root.wm_state('zoomed')
405
406     Application(root).pack(side="top", fill="both", expand=True)
407
408     root.mainloop()

```

Capítulo 4

Resultados y discusión de la tesis

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

4.1. Resultados computacionales

La presente investigación tiene como finalidad pronósticar las ventas de la siguiente semana utilizando un set de datos que entrena a una red neuronal fast forward en un sistema basado en conocimiento. Se consultó el pronóstico para la semana del 23 de febrero del 2020, tomando en cuenta que es semanas antes de la cuarentena.

En la tabla 4.1 se muestra el dataset con los valores de venta observados y la fecha, además

de una columna con los valores de venta pronosticados por el sistema basado en conocimiento.

AÑO	MES	DIA	VENTAS OBSERVADAS	PREDICCIÓN
2020	2	23	5	1
2020	2	24	0	5
2020	2	25	16	7
2020	2	26	5	5
2020	2	27	3	3
2020	2	28	2	4
2020	2	29	5	5

4.2. Discusión de Resultados

Para la contrastación se utilizó el método Chi-Cuadrado. Utilizando la Tabla 4.1, podemos decir:

$$df = 6$$

$$Chi - Square = 11,4039$$

$$p - value = 0,0766682$$

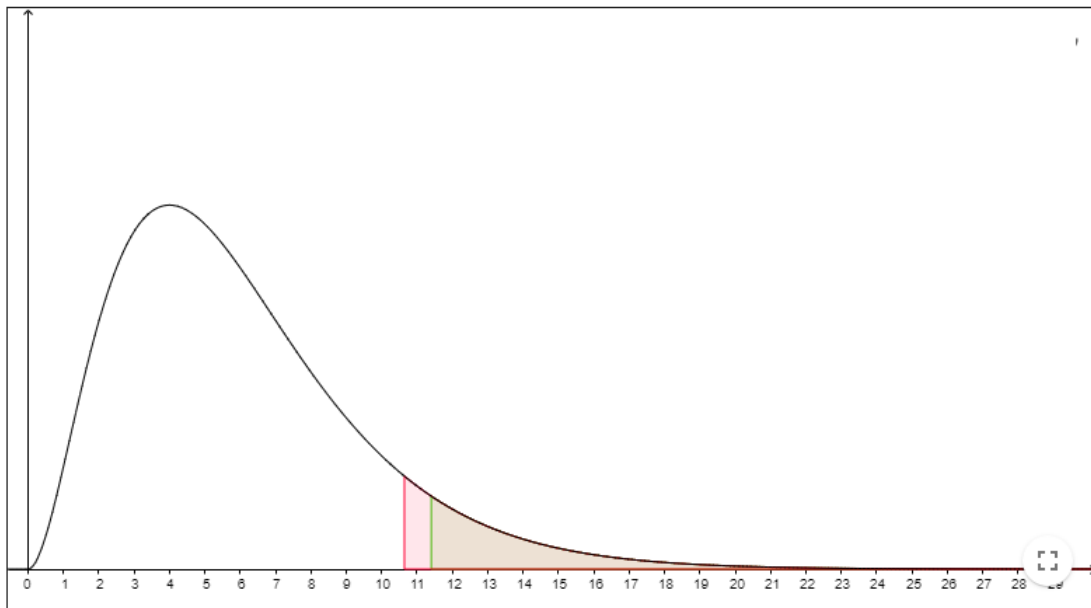


Figura 4.1: Contrastación por método de Chi-Cuadrado en Geogebra.

Fuente: Elaboración propia.

Con los resultados obtenidos por la contrastación, se obtiene el $p\text{-value} = 0.0766682$, valor inferior a 0.1 por lo tanto, llegamos a la conclusión que la hipótesis es estadísticamente significativa.

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 pronosticación de ventas así como su solución aplicando un sistema basado en conocimiento, para este caso se determinó usar redes neuronales fast forward, la cual nos permitió pronosticar las ventas de la próxima semana.
2. Se cumplió con la implementación de un prototipo del sistema basado en conocimiento para pronósticos, la cual es muy útil para la toma decisiones.

3. En la investigación se logró descartar la hipótesis nula con un nivel de significancia del 90 %.

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:

- Utilizar datasets con una mayor cantidad de registros, para un mejor entrenamiento de la red neuronal y así una obtener una predicción más eficaz.
- Proponer automatización del sistema o módulo de compras al conectarlo con el sistema basado en conocimiento para establecer las compras que se realizaran para esa semana de acuerdo a las ventas pronosticadas.
- Crear un app móvil para acceder al sistema basado en conocimiento de manera remota.

Referencias bibliográficas

Cruz, A. (2012). Definiciones de sistemas inteligentes y 5 ejemplos de sistemas inteligentes. url=<http://www.academia.edu/10376257/Definiciones-de-sistemas-inteligentes-y-5-ejemplos-de-sistemas-inteligentes> . [Online; accedado Octubre 05, 2019].

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., and Saurous, R. A. (2017). Tensorflow distributions. *arXiv preprint arXiv:1711.10604*.

Erlinb (2017). Redes neuronales: Conceptos básicos y aplicaciones. url=www.perfectcodex.com/redneuronal/2017/12/27/conceptos-basicos/. [Online; accedado Octubre 01, 2019].

Espitia Corredor, Daniel Eduardo (2017). Entrenamiento del perceptron. url=www.platzi.com/tutoriales/1352-ia/2619-entrenamiento-del-perceptron/. [Online; accedado Octubre 06, 2019].

- Freeman, J. A. and Skapura, D. M. (1991). *Neural networks: algorithms, applications, and programming techniques*. Addison Wesley Longman Publishing Co., Inc.
- Matich, D. J. (2001). *Redes neuronales: Conceptos básicos y aplicaciones*. Universidad Tecnológica Nacional, México.
- McCarthy, J. (1989). Artificial intelligence, logic and formalizing common sense. In *Philosophical logic and artificial intelligence*, pages 161–190. Springer.
- Michael A. Nielsen (2015). *Redes neuronales y aprendizaje profundo: Cómo funciona el algoritmo de backpropagation*. url=www.neuralnetworksanddeeplearning.com/chap2.html. [Online; accedado Octubre 01, 2019].
- Munt, A. M. (2018). *Introducción a los modelos de redes neuronales artificiales el perceptrón simple y multicapa*. Universidad de Zaragoza, España.
- Pat Fernández, L. A., Menchaca, M., Fernández, P., Martínez Luis, D., et al. (2013). *Introducción a los modelos de regresión*. Number EE/519.536 I5.
- Prince, B. D. (2009). *Inteligencia artificial. ¿qué es?* url=<http://inteligencia-artificial-delabarrita.blogspot.pe/2009/10/en-que-consiste-y-en-que-se-fundamenta.html>. [Online; accedado Octubre 04, 2019].
- Rossum, G. (1995). *Python reference manual*.

Apéndice A

Dataset de entrenamiento

Año	Mes	Día	Ventas Observadas
2018	1	10	1
2018	1	11	0
2018	1	12	0
2018	1	13	0
2018	1	14	0
2018	1	15	0
2018	1	16	0
2018	1	17	0
2018	1	18	0
2018	1	19	0
2018	1	20	0
2018	1	21	0
2018	1	22	0
2018	1	23	0
2018	1	24	0
2018	1	25	0
2018	1	26	0

2018	1	27	0
2018	1	28	0
2018	1	29	0
2018	1	30	0
2018	1	31	0
2018	2	1	1
2018	2	2	0
2018	2	3	0
2018	2	4	0
2018	2	5	0
2018	2	6	0
2018	2	7	0
2018	2	8	0
2018	2	9	0
2018	2	10	0
2018	2	11	0
2018	2	12	0
2018	2	13	0
2018	2	14	0
2018	2	15	0
2018	2	16	0
2018	2	17	0
2018	2	18	0
2018	2	19	0
2018	2	20	0
2018	2	21	0
2018	2	22	0
2018	2	23	0
2018	2	24	0
2018	2	25	0
2018	2	26	0
2018	2	27	0
2018	2	28	0
2018	3	1	0
2018	3	2	0

2018	3	3	0
2018	3	4	0
2018	3	5	0
2018	3	6	0
2018	3	7	0
2018	3	8	0
2018	3	9	0
2018	3	10	0
2018	3	11	0
2018	3	12	0
2018	3	13	0
2018	3	14	0
2018	3	15	0
2018	3	16	0
2018	3	17	0
2018	3	18	0
2018	3	19	0
2018	3	20	0
2018	3	21	0
2018	3	22	0
2018	3	23	0
2018	3	24	0
2018	3	25	0
2018	3	26	0
2018	3	27	0
2018	3	28	0
2018	3	29	0
2018	3	30	0
2018	3	31	0
2018	4	1	0
2018	4	2	0
2018	4	3	9
2018	4	4	26
2018	4	5	18
2018	4	6	9

2018	4	7	20
2018	4	8	32
2018	4	9	23
2018	4	10	23
2018	4	11	22
2018	4	12	15
2018	4	13	43
2018	4	14	23
2018	4	15	26
2018	4	16	30
2018	4	17	18
2018	4	18	19
2018	4	19	24
2018	4	20	18
2018	4	21	17
2018	4	22	32
2018	4	23	24
2018	4	24	27
2018	4	25	22
2018	4	26	27
2018	4	27	16
2018	4	28	23
2018	4	29	32
2018	4	30	0
2018	5	1	20
2018	5	2	27
2018	5	3	20
2018	5	4	12
2018	5	5	13
2018	5	6	33
2018	5	7	0
2018	5	8	20
2018	5	9	30
2018	5	10	19
2018	5	11	17

2018	5	12	22
2018	5	13	21
2018	5	14	0
2018	5	15	14
2018	5	16	17
2018	5	17	20
2018	5	18	24
2018	5	19	11
2018	5	20	29
2018	5	21	0
2018	5	22	16
2018	5	23	23
2018	5	24	14
2018	5	25	10
2018	5	26	20
2018	5	27	20
2018	5	28	0
2018	5	29	23
2018	5	30	26
2018	5	31	39
2018	6	1	23
2018	6	2	29
2018	6	3	39
2018	6	4	10
2018	6	5	15
2018	6	6	35
2018	6	7	25
2018	6	8	36
2018	6	9	41
2018	6	10	16
2018	6	11	20
2018	6	12	11
2018	6	13	19
2018	6	14	18
2018	6	15	20

2018	6	16	30
2018	6	17	35
2018	6	18	42
2018	6	19	8
2018	6	20	15
2018	6	21	24
2018	6	22	17
2018	6	23	34
2018	6	24	31
2018	6	25	21
2018	6	26	9
2018	6	27	32
2018	6	28	30
2018	6	29	23
2018	6	30	35
2018	7	1	28
2018	7	2	15
2018	7	3	11
2018	7	4	26
2018	7	5	23
2018	7	6	24
2018	7	7	21
2018	7	8	31
2018	7	9	19
2018	7	10	11
2018	7	11	22
2018	7	12	22
2018	7	13	17
2018	7	14	20
2018	7	15	25
2018	7	16	13
2018	7	17	11
2018	7	18	21
2018	7	19	11
2018	7	20	25

2018	7	21	25
2018	7	22	30
2018	7	23	13
2018	7	24	14
2018	7	25	19
2018	7	26	17
2018	7	27	19
2018	7	28	28
2018	7	29	40
2018	7	30	20
2018	7	31	9
2018	8	1	16
2018	8	2	16
2018	8	3	15
2018	8	4	29
2018	8	5	24
2018	8	6	22
2018	8	7	9
2018	8	8	17
2018	8	9	9
2018	8	10	11
2018	8	11	21
2018	8	12	23
2018	8	13	21
2018	8	14	5
2018	8	15	25
2018	8	16	19
2018	8	17	20
2018	8	18	21
2018	8	19	29
2018	8	20	38
2018	8	21	6
2018	8	22	24
2018	8	23	20
2018	8	24	20

2018	8	25	20
2018	8	26	38
2018	8	27	32
2018	8	28	12
2018	8	29	13
2018	8	30	25
2018	8	31	28
2018	9	1	33
2018	9	2	33
2018	9	3	30
2018	9	4	10
2018	9	5	16
2018	9	6	20
2018	9	7	18
2018	9	8	28
2018	9	9	26
2018	9	10	19
2018	9	11	4
2018	9	12	20
2018	9	13	25
2018	9	14	20
2018	9	15	25
2018	9	16	23
2018	9	17	17
2018	9	18	8
2018	9	19	20
2018	9	20	19
2018	9	21	20
2018	9	22	21
2018	9	23	28
2018	9	24	22
2018	9	25	7
2018	9	26	20
2018	9	27	22
2018	9	28	29

2018	9	29	23
2018	9	30	36
2018	10	1	21
2018	10	2	9
2018	10	3	15
2018	10	4	17
2018	10	5	20
2018	10	6	19
2018	10	7	28
2018	10	8	24
2018	10	9	0
2018	10	10	21
2018	10	11	13
2018	10	12	20
2018	10	13	14
2018	10	14	34
2018	10	15	22
2018	10	16	5
2018	10	17	15
2018	10	18	16
2018	10	19	11
2018	10	20	13
2018	10	21	22
2018	10	22	0
2018	10	23	26
2018	10	24	15
2018	10	25	22
2018	10	26	15
2018	10	27	16
2018	10	28	13
2018	10	29	0
2018	10	30	28
2018	10	31	12
2018	11	1	19
2018	11	2	26

2018	11	3	21
2018	11	4	22
2018	11	5	0
2018	11	6	22
2018	11	7	15
2018	11	8	13
2018	11	9	17
2018	11	10	24
2018	11	11	18
2018	11	12	0
2018	11	13	23
2018	11	14	14
2018	11	15	19
2018	11	16	20
2018	11	17	19
2018	11	18	24
2018	11	19	0
2018	11	20	27
2018	11	21	18
2018	11	22	19
2018	11	23	26
2018	11	24	17
2018	11	25	16
2018	11	26	0
2018	11	27	27
2018	11	28	14
2018	11	29	11
2018	11	30	19
2018	12	1	13
2018	12	2	24
2018	12	3	0
2018	12	4	20
2018	12	5	21
2018	12	6	16
2018	12	7	16

2018	12	8	26
2018	12	9	23
2018	12	10	0
2018	12	11	35
2018	12	12	25
2018	12	13	14
2018	12	14	18
2018	12	15	28
2018	12	16	32
2018	12	17	0
2018	12	18	29
2018	12	19	13
2018	12	20	17
2018	12	21	17
2018	12	22	20
2018	12	23	26
2018	12	24	0
2018	12	25	28
2018	12	26	36
2018	12	27	16
2018	12	28	20
2018	12	29	18
2018	12	30	28
2018	12	31	37
2019	1	1	21
2019	1	2	0
2019	1	3	38
2019	1	4	21
2019	1	5	21
2019	1	6	20
2019	1	7	0
2019	1	8	16
2019	1	9	16
2019	1	10	19
2019	1	11	21

2019	1	12	18
2019	1	13	19
2019	1	14	25
2019	1	15	15
2019	1	16	12
2019	1	17	6
2019	1	18	13
2019	1	19	16
2019	1	20	27
2019	1	21	27
2019	1	22	17
2019	1	23	13
2019	1	24	14
2019	1	25	14
2019	1	26	13
2019	1	27	22
2019	1	28	27
2019	1	29	9
2019	1	30	12
2019	1	31	18
2019	2	1	10
2019	2	2	15
2019	2	3	20
2019	2	4	25
2019	2	5	17
2019	2	6	12
2019	2	7	21
2019	2	8	14
2019	2	9	6
2019	2	10	15
2019	2	11	25
2019	2	12	9
2019	2	13	13
2019	2	14	20
2019	2	15	17

2019	2	16	10
2019	2	17	20
2019	2	18	25
2019	2	19	9
2019	2	20	5
2019	2	21	20
2019	2	22	8
2019	2	23	13
2019	2	24	16
2019	2	25	21
2019	2	26	13
2019	2	27	11
2019	2	28	14
2019	3	1	11
2019	3	2	11
2019	3	3	10
2019	3	4	25
2019	3	5	21
2019	3	6	11
2019	3	7	12
2019	3	8	9
2019	3	9	10
2019	3	10	21
2019	3	11	15
2019	3	12	11
2019	3	13	9
2019	3	14	13
2019	3	15	11
2019	3	16	18
2019	3	17	23
2019	3	18	19
2019	3	19	3
2019	3	20	5
2019	3	21	12
2019	3	22	9

2019	3	23	10
2019	3	24	21
2019	3	25	16
2019	3	26	15
2019	3	27	7
2019	3	28	9
2019	3	29	7
2019	3	30	16
2019	3	31	20
2019	4	1	17
2019	4	2	0
2019	4	3	8
2019	4	4	9
2019	4	5	13
2019	4	6	5
2019	4	7	25
2019	4	8	23
2019	4	9	0
2019	4	10	0
2019	4	11	11
2019	4	12	8
2019	4	13	14
2019	4	14	12
2019	4	15	18
2019	4	16	7
2019	4	17	4
2019	4	18	18
2019	4	19	16
2019	4	20	19
2019	4	21	8
2019	4	22	19
2019	4	23	6
2019	4	24	9
2019	4	25	14
2019	4	26	7

2019	4	27	8
2019	4	28	19
2019	4	29	19
2019	4	30	7
2019	5	1	11
2019	5	2	17
2019	5	3	6
2019	5	4	7
2019	5	5	7
2019	5	6	21
2019	5	7	11
2019	5	8	10
2019	5	9	7
2019	5	10	9
2019	5	11	8
2019	5	12	14
2019	5	13	19
2019	5	14	10
2019	5	15	12
2019	5	16	7
2019	5	17	0
2019	5	18	3
2019	5	19	15
2019	5	20	19
2019	5	21	14
2019	5	22	8
2019	5	23	15
2019	5	24	6
2019	5	25	5
2019	5	26	11
2019	5	27	20
2019	5	28	10
2019	5	29	8
2019	5	30	19
2019	5	31	17

2019	6	1	11
2019	6	2	18
2019	6	3	14
2019	6	4	12
2019	6	5	8
2019	6	6	7
2019	6	7	7
2019	6	8	8
2019	6	9	16
2019	6	10	21
2019	6	11	0
2019	6	12	5
2019	6	13	18
2019	6	14	15
2019	6	15	15
2019	6	16	15
2019	6	17	11
2019	6	18	0
2019	6	19	7
2019	6	20	9
2019	6	21	4
2019	6	22	12
2019	6	23	16
2019	6	24	25
2019	6	25	9
2019	6	26	10
2019	6	27	9
2019	6	28	6
2019	6	29	6
2019	6	30	23
2019	7	1	12
2019	7	2	10
2019	7	3	5
2019	7	4	8
2019	7	5	9

2019	7	6	15
2019	7	7	15
2019	7	8	22
2019	7	9	5
2019	7	10	10
2019	7	11	7
2019	7	12	7
2019	7	13	8
2019	7	14	16
2019	7	15	24
2019	7	16	8
2019	7	17	10
2019	7	18	17
2019	7	19	12
2019	7	20	8
2019	7	21	20
2019	7	22	18
2019	7	23	7
2019	7	24	7
2019	7	25	7
2019	7	26	6
2019	7	27	7
2019	7	28	10
2019	7	29	12
2019	7	30	7
2019	7	31	3
2019	8	1	8
2019	8	2	6
2019	8	3	8
2019	8	4	18
2019	8	5	13
2019	8	6	6
2019	8	7	8
2019	8	8	7
2019	8	9	2

2019	8	10	5
2019	8	11	12
2019	8	12	12
2019	8	13	6
2019	8	14	9
2019	8	15	5
2019	8	16	3
2019	8	17	4
2019	8	18	14
2019	8	19	22
2019	8	20	4
2019	8	21	4
2019	8	22	7
2019	8	23	6
2019	8	24	4
2019	8	25	15
2019	8	26	20
2019	8	27	6
2019	8	28	13
2019	8	29	4
2019	8	30	7
2019	8	31	5
2019	9	1	15
2019	9	2	16
2019	9	3	4
2019	9	4	4
2019	9	5	7
2019	9	6	1
2019	9	7	16
2019	9	8	13
2019	9	9	19
2019	9	10	3
2019	9	11	1
2019	9	12	11
2019	9	13	3

2019	9	14	3
2019	9	15	18
2019	9	16	13
2019	9	17	5
2019	9	18	5
2019	9	19	3
2019	9	20	6
2019	9	21	7
2019	9	22	12
2019	9	23	16
2019	9	24	5
2019	9	25	5
2019	9	26	8
2019	9	27	10
2019	9	28	5
2019	9	29	11
2019	9	30	0
2019	10	1	21
2019	10	2	7
2019	10	3	6
2019	10	4	10
2019	10	5	5
2019	10	6	7
2019	10	7	13
2019	10	8	3
2019	10	9	5
2019	10	10	5
2019	10	11	10
2019	10	12	11
2019	10	13	10
2019	10	14	12
2019	10	15	6
2019	10	16	3
2019	10	17	4
2019	10	18	6

2019	10	19	3
2019	10	20	7
2019	10	21	9
2019	10	22	2
2019	10	23	6
2019	10	24	17
2019	10	25	15
2019	10	26	13
2019	10	27	26
2019	10	28	13
2019	10	29	22
2019	10	30	17
2019	10	31	9
2019	11	1	13
2019	11	2	14
2019	11	3	17
2019	11	4	20
2019	11	5	13
2019	11	6	0
2019	11	7	17
2019	11	8	11
2019	11	9	15
2019	11	10	20
2019	11	11	9
2019	11	12	14
2019	11	13	7
2019	11	14	8
2019	11	15	11
2019	11	16	15
2019	11	17	25
2019	11	18	12
2019	11	19	16
2019	11	20	13
2019	11	21	15
2019	11	22	13

2019	11	23	10
2019	11	24	19
2019	11	25	10
2019	11	26	12
2019	11	27	5
2019	11	28	14
2019	11	29	9
2019	11	30	11
2019	12	1	13
2019	12	2	20
2019	12	3	10
2019	12	4	1
2019	12	5	12
2019	12	6	6
2019	12	7	15
2019	12	8	9
2019	12	9	15
2019	12	10	7
2019	12	11	8
2019	12	12	5
2019	12	13	7
2019	12	14	7
2019	12	15	8
2019	12	16	17
2019	12	17	5
2019	12	18	4
2019	12	19	14
2019	12	20	11
2019	12	21	11
2019	12	22	20
2019	12	23	11
2019	12	24	9
2019	12	25	10
2019	12	26	11
2019	12	27	7

2019	12	28	12
2019	12	29	8
2019	12	30	8
2019	12	31	5
2020	1	1	9
2020	1	2	17
2020	1	3	13
2020	1	4	0
2020	1	5	17
2020	1	6	11
2020	1	7	7
2020	1	8	11
2020	1	9	12
2020	1	10	10
2020	1	11	8
2020	1	12	9
2020	1	13	0
2020	1	14	6
2020	1	15	6
2020	1	16	11
2020	1	17	8
2020	1	18	11
2020	1	19	4
2020	1	20	8
2020	1	21	0
2020	1	22	10
2020	1	23	9
2020	1	24	8
2020	1	25	5
2020	1	26	4
2020	1	27	12
2020	1	28	0
2020	1	29	8
2020	1	30	10
2020	1	31	9

2020	2	1	2
2020	2	2	7
2020	2	3	12
2020	2	4	0
2020	2	5	4
2020	2	6	0
2020	2	7	6
2020	2	8	4
2020	2	9	10
2020	2	10	6
2020	2	11	0
2020	2	12	3
2020	2	13	6
2020	2	14	4
2020	2	15	10
2020	2	16	15
2020	2	17	6
2020	2	18	4
2020	2	19	0
2020	2	20	3
2020	2	21	3
2020	2	22	3
2020	2	23	5
2020	2	24	0
2020	2	25	16
2020	2	26	5
2020	2	27	3
2020	2	28	2
2020	2	29	5
2020	3	1	6
2020	3	2	7
2020	3	3	0
2020	3	4	4
2020	3	5	1
2020	3	6	3

2020	3	7	2
2020	3	8	6
2020	3	9	5
2020	3	10	3
2020	3	11	6
2020	3	12	0
2020	3	13	3
2020	3	14	6
2020	3	15	3
2020	3	16	0
2020	3	17	0
2020	3	18	0
2020	3	19	0
2020	3	20	0
2020	3	21	0
2020	3	22	0
2020	3	23	0
2020	3	24	0
2020	3	25	0
2020	3	26	0
2020	3	27	0
2020	3	28	0
2020	3	29	0
2020	3	30	0
2020	3	31	0
2020	4	1	0
2020	4	2	0
2020	4	3	0
2020	4	4	0
2020	4	5	0
2020	4	6	0
2020	4	7	0
2020	4	8	0
2020	4	9	0
2020	4	10	0

2020	4	11	0
2020	4	12	0
2020	4	13	0
2020	4	14	0
2020	4	15	0
2020	4	16	0
2020	4	17	0
2020	4	18	0
2020	4	19	0
2020	4	20	0
2020	4	21	0
2020	4	22	0
2020	4	23	0
2020	4	24	0
2020	4	25	0
2020	4	26	0
2020	4	27	0
2020	4	28	0
2020	4	29	0
2020	4	30	0
2020	5	1	0
2020	5	2	0
2020	5	3	0
2020	5	4	0
2020	5	5	0
2020	5	6	0
2020	5	7	0
2020	5	8	0
2020	5	9	0
2020	5	10	0
2020	5	11	0
2020	5	12	0
2020	5	13	0
2020	5	14	0
2020	5	15	7

2020	5	16	4
2020	5	17	0
2020	5	18	6
2020	5	19	4
2020	5	20	8
2020	5	21	8
2020	5	22	6
2020	5	23	7
2020	5	24	0
2020	5	25	4
2020	5	26	8
2020	5	27	3
2020	5	28	4
2020	5	29	8
2020	5	30	4
2020	5	31	0
2020	6	1	2
2020	6	2	3
2020	6	3	6
2020	6	4	6
2020	6	5	3
2020	6	6	3
2020	6	7	0
2020	6	8	6
2020	6	9	6
2020	6	10	5
2020	6	11	5
2020	6	12	5
2020	6	13	6
2020	6	14	0
2020	6	15	13
2020	6	16	4
2020	6	17	1
2020	6	18	7
2020	6	19	4

2020	6	20	5
2020	6	21	6
2020	6	22	6
2020	6	23	4
2020	6	24	3
2020	6	25	2
2020	6	26	6
2020	6	27	5
2020	6	28	0
2020	6	29	7
2020	6	30	4
2020	7	1	4
2020	7	2	7
2020	7	3	4
2020	7	4	6
2020	7	5	7
2020	7	6	9
2020	7	7	6
2020	7	8	3
2020	7	9	7
2020	7	10	3
2020	7	11	5
2020	7	12	6
2020	7	13	6
2020	7	14	0
2020	7	15	4
2020	7	16	7
2020	7	17	4
2020	7	18	5
2020	7	19	4
2020	7	20	9
2020	7	21	1
2020	7	22	0
2020	7	23	3
2020	7	24	4

2020	7	25	4
2020	7	26	7
2020	7	27	0
2020	7	28	12
2020	7	29	7
2020	7	30	0
2020	7	31	3
2020	8	1	6
2020	8	2	7
2020	8	3	8
2020	8	4	2
2020	8	5	1
2020	8	6	0
2020	8	7	2
2020	8	8	2
2020	8	9	6
2020	8	10	3
2020	8	11	0
2020	8	12	2
2020	8	13	7
2020	8	14	8
2020	8	15	4
2020	8	16	0
2020	8	17	10
2020	8	18	4
2020	8	19	5
2020	8	20	4
2020	8	21	5
2020	8	22	6
2020	8	23	4
2020	8	24	0
2020	8	25	8
2020	8	26	6
2020	8	27	5
2020	8	28	6

2020	8	29	4
2020	8	30	9
2020	8	31	0
2020	9	1	6
2020	9	2	3
2020	9	3	3

Apéndice B

Segundo apendice

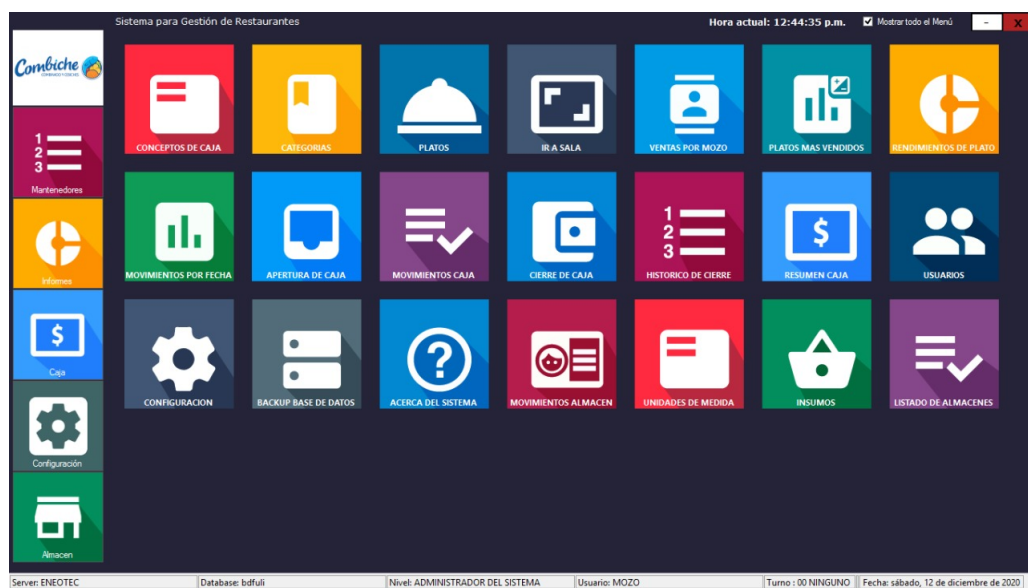


Figura B.1: Sistema de Ventas de Combiche del cual se obtuvo el dataset.

Fuente: Elaboración propia.