

Universidad Internacional de La Rioja (UNIR)

ESIT

Máster Universitario en Inteligencia Artificial

Computación autonómica para Centro de Datos mediante agentes inteligentes.

Trabajo Fin de Máster

Presentado por: Cabrera Burgos, Xavier Antonio

Director/a: García Martínez, Yamila. PhD

Ciudad: Guayaquil, Ecuador
Fecha: 17 de septiembre de 2020

Resumen

En el ámbito de los centros de datos, la computación autónoma se presenta como la solución a los problemas de gestión de tareas que requieren de personal altamente calificado. Este proyecto tiene como objetivo implementar una red neuronal difusa dentro de un elemento autónomo computacional para inferir si es seguro auto repararse. Se utilizó para este efecto la información recolectada en el periodo comprendido entre mayo de 2018 y mayo de 2020 de las tareas asignadas a ingenieros expertos que debían realizar mantenimientos de reparación correctivos y preventivos a los componentes de tecnologías de información de un centro de datos ubicado en Ecuador. En la información indicada se recolectaron 929 trabajos de mantenimiento, estos parámetros sirvieron para entrenar una red neuronal difusa que puede inferir si una tarea debe ser realizada o no. En la experimentación se modificaron aleatoriamente los datos de las amenazas para simular cambios en el entorno y se comprobó que la Raíz del error cuadrático medio fue 4.23×10^{-2} al utilizar la nueva información desconocida para el modelo ya entrenado. Con estos resultados se puede destacar la importancia de desarrollar sistemas de auto corrección que disminuyan las operaciones manuales en los centros de datos.

Palabras Clave: Centro de Datos, Computación Autónoma, Inteligencia Artificial, Lógica Borrosa

Abstract

In the field of data centers, autonomic computing is presented as the solution to the problems of managing tasks that require highly qualified personnel. This project aims to implement a fuzzy neural network within a computational autonomic element to infer whether it is safe to repair itself. For this purpose, the information collected in the period between May 2018 and May 2020 of the tasks assigned to expert engineers who had to carry out corrective and preventive repair maintenance to the information technology components of a data center located in Ecuador. In the indicated information, 929 maintenance jobs were collected, these parameters were used to train a fuzzy neural network that can infer whether a task should be performed or not. In the experimentation, the threat data was randomly modified to simulate changes in the environment, and it was found that the Root mean square error was 4.23×10^{-2} when using the new unknown information for the already trained model. With these results, the importance of developing self-correction systems that reduce manual operations in data centers can be highlighted.

Keywords: Data Center, Autonomic Computing, Artificial Intelligence, Fully Logic

Índice de contenidos

1. Introducción.....	1
1.1 Motivación	2
1.2 Planteamiento del trabajo	3
1.3 Estructura de la memoria	4
2. Contexto y estado del arte.....	5
2.1 Contexto	5
2.2 Centros de Datos.....	6
2.2.1 Breve introducción al estado actual de los centros de datos	6
2.2.2 Visión arquitectónica de los centros de datos.	7
2.2.3 Computación de Nube como servicio	9
2.2.4 Mantenimiento de Infraestructura	11
2.3 Computación Autónoma.....	12
2.3.1 Fuente de inspiración de la Computación Autónoma	12
2.3.2 Computación autónoma y Computación tradicional	14
2.3.3 Arquitectura de elementos autónomos.....	16
2.4 Inteligencia Artificial	17
2.4.1 Lógica Borrosa	18
2.4.2 Redes Neuronales.....	20
2.4.3 Sistema de Inferencia Neuro Difuso	21
2.5 Estado del Arte	24
3. Objetivos y metodología de trabajo	27
3.1. Objetivo general.....	27
3.2. Objetivos específicos	27
3.3. Metodología del trabajo	27
4. Identificación de requisitos	31
4.1 Situación actual de operaciones del centro de datos	31

5. Descripción de la solución	33
5.1 Agente autonómico Inteligente.....	33
5.2 Definición de Red ANFIS	34
5.3 Escala de evaluación de elementos	35
5.4 Flujo de acción del método propuesto	38
6. Evaluación.....	40
6.1 Conjunto de datos.....	40
6.1.1 Análisis de Datos.....	41
6.2 Creación del Sistema de Inferencia Difusa.	44
6.2 Entrenamiento y pruebas de la Red ANFIS.	45
7. Conclusiones y trabajo futuro	51
7.1. Conclusiones	51
7.2. Líneas de trabajo futuro	52
8. Bibliografía	53
Anexos.....	57
Anexo I. Sistema FIS utilizado en Matlab.....	57
Anexo II. Código de ANFIS en MatLab	60
Anexo. Artículo de investigación	63

Índice de tablas

Tabla 1. Conceptos básicos de la computación autonómica y su comparación con los de la computación tradicional.....	15
Tabla 2. Comparativa de las redes neuronales y los sistemas de control difuso.	21
Tabla 3. Clasificación de Redes Neuronales Difusas	22
Tabla 4. Índice de la madurez Autonómica.....	32
Tabla 5. Entradas de la red ANFIS y sus correspondientes funciones de pertenencia.	34
Tabla 6. Valoración aproximada de los elementos de TRM.....	36
Tabla 7. Valoración aproximada de la disponibilidad de los elementos	37
Tabla 8. Valoración de la criticidad de los elementos	37
Tabla 9. Valoración de las amenazas al sistema.....	38
Tabla 10. Campos de la tabla de control de mantenimientos	40

Índice de figuras

Figura 1. Tendencia del consumo en un centro de Datos	2
Figura 2. Breve esquema del bucle de control MAPE-K de los elementos autonómicos	3
Figura 3. Variables clave del Negocio	5
Figura 4. Comportamiento en la tendencia de utilización de puertos para transporte de información	6
Figura 5. Estimación de capacidad de los discos duros en los centros de datos de Estados Unidos.....	7
Figura 6. TRM clásico de un centro de datos	8
Figura 7. Posicionamiento estructural de la relación en los diferentes servicios de computación en la nube	9
Figura 8. Informe de IaaS proporcionado por Gartner	10
Figura 9. Principales causas de indisponibilidad en los servicios de centro de datos	11
Figura 10. Sistemas y subsistemas del sistema nervioso central	13
Figura 11. Atributos de la Computación Autonómica	14
Figura 12. Estructura de un elemento autonómico.	17
Figura 13. Disciplinas más relevantes de IA.	18
Figura 14. Componentes principales de un Sistema Mamdani de Inferencia Difusa	19
Figura 15. Resultado de la función de pertenencia.....	19
Figura 16. Diagrama comparativo entre una neurona natural y una artificial.	20
Figura 17. Estructura básica de una red neuronal artificial.	21
Figura 18. Proceso de inferencia de un sistema de lógica difusa.	23
Figura 19. Red neuronal tipo ANFIS.....	24
Figura 20. Sistema de Red Neuronal NEFCLASS.	25
Figura 21. Enfoque de la integración de ANFIS en la administración de sistemas de computación de gran escala.....	26
Figura 22. Componentes de un sistema Experto	28
Figura 23. Proceso regular para la creación de un Sistema Experto	29

Figura 24.Elemento autonómico que incluye una red ANFIS en la fase de análisis.....	33
Figura 25.Estructura de la red neuronal ANFIS para la solución propuesta.....	35
Figura 26.Valoración de los elementos de entrada de la red	36
Figura 27.Diagrama de flujo del funcionamiento de alto nivel de la fase de análisis	39
Figura 28.Información del conjunto de Datos a procesar.....	41
Figura 29.Descripción del conjunto de Datos a procesar.....	41
Figura 30.Histogramas de las columnas del conjunto de Datos a procesar	42
Figura 31.Grafico de Correlación entre los elementos del TRM y los mantenimientos realizados.....	43
Figura 32.Correlación de las distintas variables agrupados por si han sido realizados o no	43
Figura 33.Asistente para la creación de Sistemas de Inferencia Difusos de Matlab	44
Figura 34.Opción de generación de sistemas FIS en Matlab.....	45
Figura 35.Carga del FIS previamente creado al workspace de Matlab	45
Figura 36.Entrenamiento de la red ANFIS	46
Figura 37.Carga del conjunto de datos de validación en una variable	47
Figura 38.Resultado de la evaluación de la red ANFIS	47
Figura 39.Comparativa de los resultados obtenidos contra los originales.....	48
Figura 40.Validación del funcionamiento de la red con datos nuevos.....	49
Figura 41.Validación del funcionamiento de la red con datos nuevos.....	50

1. Introducción

Existen muchas definiciones de inteligencia (Moreno et al., 1998), coloquialmente se ha concebido como la capacidad de pensamiento, las ideas, el razonamiento y por supuesto la toma de decisiones es lo que los humanos percibimos como Inteligencia. Es el poder de aprender cosas, asociarlas y sacar conclusiones que nos benefician de manera personal o grupal. En el campo científico se dice que la inteligencia es un bien medible y cuantificable clasificándolas por sus características múltiples, biológicas, de desarrollo y psicométricas (Villamizar & Donoso, 2013). La Inteligencia Artificial (IA) por su parte trata de emular este comportamiento mediante el uso de algoritmos que perciben el entorno en el que se desenvuelven para llevar acciones que simulen el razonamiento y la toma de decisiones.

Esta capacidad humana, la inteligencia, se desarrolla en el cerebro, órgano que forma parte del Sistema Nervioso Central, el cual está compuesto por múltiples sistemas. Uno de estos subsistemas es el llamado Sistema Nervioso Autónomo el cual se encarga de algunas de las funciones humanas que se dan por hecho, pero son vitales para el correcto funcionamiento de todo el cuerpo, el latido del corazón, respiración, sudoración, procesamiento de alimentos.

Los científicos de las ciencias de la computación han tomado este diseño biológico para inspirarse en dos sistemas diferenciados pero inclusivos. Por un lado, la IA que se podría definir como: “Un programa que, en un mundo arbitrario, no se las arreglará peor que un humano” (Dobrev, 2012), y por otro lado la computación autónoma se podría definir como la propuesta de arquitectura computacional para disminuir la carga administrativa de los operadores en los centros de datos (Obienu, 2018). De esta última, su inspiración, el sistema nervioso autónomo empuja a estos sistemas a ser diseñados de tal forma en que actúen y respondan de manera automática y sin intervención de ninguna clase. Basado en estos conceptos se propone una combinación de estrategias de la Computación Autónoma (CA) y la IA para crear sistemas computacionales en vías de la independencia de la intervención humana. La integración de todos estos conocimientos se propone en esta investigación como aporte para dar solución al problema de la ejecución y planeación de los mantenimientos que actualmente se realizan de forma manual por medio de la intervención de expertos necesarios en los centros de datos.

1.1 Motivación

La excesiva complejidad que se tiene en la actualidad en los Centros de Datos puede llegar a desconcertar a sus operadores (Eilam et al., 2006). El manejo de múltiples tecnologías dispersas en ambientes heterogéneos empuja a las organizaciones que manejan estos centros, hacia la necesidad de tener que recurrir a expertos en las diversas líneas tecnológicas que se manejan. Esto solo recarga el gasto operativo de las organizaciones y disminuye la confiabilidad de los sistemas al estar expuesto a errores humanos (Sterritt, 2005). Esta complejidad es una consecuencia del crecimiento abrumador al que están expuestos los centros de datos. En la figura 1 se puede observar la tendencia de crecimiento de un centro de datos.

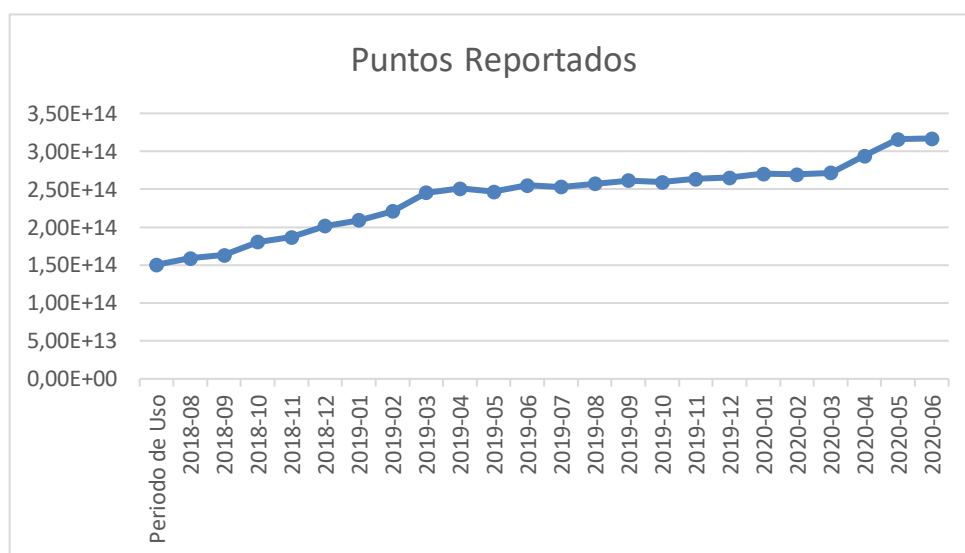


Figura 1. Tendencia del consumo en un centro de Datos

(Elaboración Propia).

En la imagen se puede observar la tendencia del consumo de los últimos 24 meses, en el uso de software de virtualización VMware del centro de datos estudiado (Datacenter Millenium). Este software se mide por puntos en referencia a la memoria RAM usada. Nótese la subida marcada en los meses de abril y mayo de 2020 en donde los servicios de tecnología tuvieron una alta demanda a causa del confinamiento a nivel mundial.

Con este comportamiento, los administradores de TI (Tecnologías de Información) de los Centros de Datos no pueden quedarse sin tomar acción. Con un crecimiento vertiginoso se ven obligados a buscar ayuda en la automatización de sus procesos para lograr atender los crecimientos inesperados. Las interrupciones del servicio, ya sea por falta de recursos tecnológicos o por fallas humanas se deben evitar a toda costa, puesto que muchos de estos en algunos casos son extremadamente críticos, como servicios bancarios, a hospitales, de transporte, cadenas de suministros alimenticios, telecomunicaciones, energía y agua potable.

1.2 Planteamiento del trabajo

La complejidad en la infraestructura tecnológica de los Centros de Datos no va a desaparecer (Fadaeefath Abadi et al., 2020) y con lo expuesto antes es visible, conforme lo observado en la figura 1, que su tendencia en aumento va a continuar. Por lo tanto, el camino del automatismo es una de las múltiples estrategias que se deben utilizar para que la carga operativa de expertos sea trasladada a elementos computacionales autónomos que sean los encargados de solucionar los problemas que pudieran presentarse.

La computación autónoma tiene como base arquitectónica el bucle de control de MAPE-K¹, el cual se utilizará como punto de arranque en la construcción de un agente inteligente al que se le incluirán capacidades de inferencia gracias a la utilización de un tipo de red neuronal difusa: ANFIS²; misma que basa su arquitectura en la combinación de los conceptos de las redes neuronales profundas y las teorías de la lógica borrosa. Con la combinación de estos elementos se pretende que el agente sea capaz de poder determinar cuándo es seguro realizar una auto reparación en sus componentes. La figura 2 muestra brevemente como se interrelaciona cada una de las fases del bucle de control MAPE-K y la relevancia de la fase de análisis dentro del mismo.

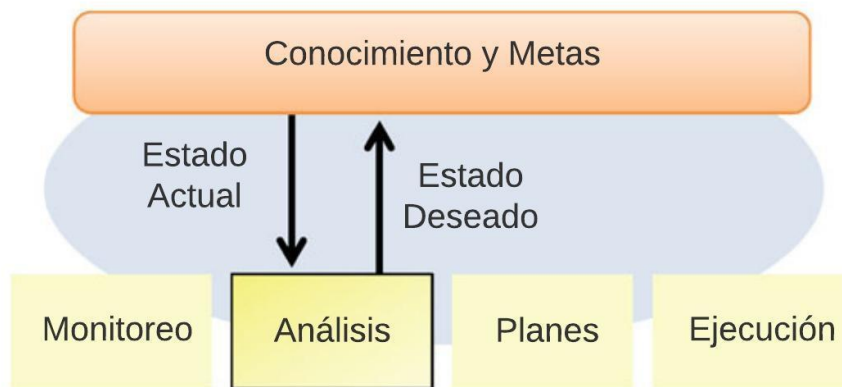


Figura 2. Breve esquema del bucle de control MAPE-K de los elementos autónomos (Lalanda et al., 2013b).

En la parte inferior de la imagen se puede observar las cuatro etapas del bucle de control MAPE-K y el conocimiento en la parte superior. Un detalle adicional es que se muestra como la fase de análisis interactúa con la base de conocimientos. Cada etapa del bucle de control es modificable, en este documento se propondrá una fase de análisis que incluya modelos de IA para procesar el cambio del estado actual al estado deseado.

¹ Monitor Analyze Plan Execute over a Shared Knowledge

² Adaptative neuro fuzzy inference system

1.3 Estructura de la memoria

Este documento se puede dividir en tres secciones principales:

- 1) La primera sección, conformada por el capítulo uno, es una introducción al problema, su análisis y solución propuesta a grandes rasgos de manera introductoria al lector.
- 2) La segunda sección, conformada por los capítulos dos y tres, el contexto y estado del arte que permite mostrar al lector el estado actual de las investigaciones relacionadas con los centros de datos, CA e IA en los campos del Razonamiento y Planificación automáticas en conjunto con los Sistemas Cognitivos Artificiales.

También podríamos declarar parte de esta sección a la metodología que se utilizará para poder poner en marcha la solución, así como los objetivos que se plantearon, de tal forma que se pueda verificar su cumplimiento posterior

- 3) En la tercera parte conformada por los capítulos restantes, se encuentra una descripción detallada de la solución, cómo se establecieron los parámetros necesarios para llevar a cabo el desarrollo de la solución y los resultados obtenidos.

Las conclusiones de lo observado, además de las líneas de trabajo futuro también forman parte de esta sección del documento.

2. Contexto y estado del arte

Para presentar la solución al problema, es necesario contextualizar tres temas importantes: los centros de datos, la computación autonómica y las redes neuronales difusas. A lo largo de los siguientes apartados se irá hilando cada uno de estos puntos para realizar la construcción del escenario sobre el cual se propone que la decisión de realizar un mantenimiento o no, debe ser automática, mejorando la eficiencia y permitiendo que los operadores expertos puedan enfocarse en otros aspectos del negocio.

2.1 Contexto

La industria busca disminuir siempre sus costos operativos (NASDAQ OMX's, 2020). Para ello han recurrido siempre al automatismo de sus procesos, sin embargo, este automatismo mecánico no es suficiente (Georgeff et al., 2014), puesto que las necesidades del negocio siempre escalan y nos conducen hacia la complejidad en donde se tiende a la insuficiencia de recursos. En la figura 3 se puede observar cómo es la relación entre la usabilidad, la complejidad, y las necesidades del negocio.

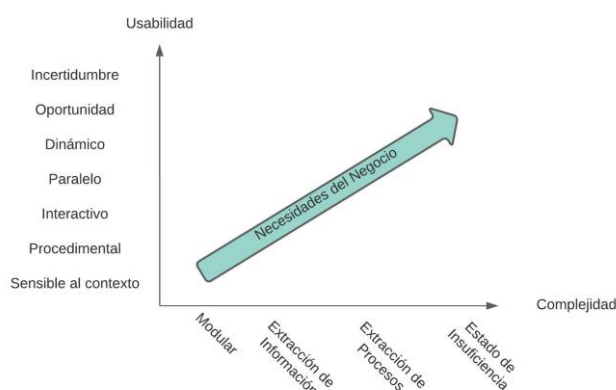


Figura 3. Variables clave del Negocio
(Georgeff et al., 2014).

Se puede apreciar la relación de las necesidades del negocio con el estado de insuficiencia. Mientras las necesidades del negocio aumentan, también el vacío que se va dejando en los bloques de construcción inconclusos por atender las necesidades urgentes. Sean estas de mantenimiento del negocio o de la lucha en la búsqueda de nuevos mercados.

Como una solución a este contexto, y con los avances de la potencia computacional, se propone la utilización de estrategias de IA como una ciencia de soporte que permite llevar un paso más allá este automatismo que es exigido por el dinamismo del negocio, permitiendo obtener resultados de la información que se puede capturar de los sistemas informáticos y

mediante algoritmos computacionales basados en probabilidades matemáticas obtener resultados precisos cuando la demanda de expertos escasea.

2.2 Centros de Datos

Los centros de datos son edificios especializados en los que se concentra un gran volumen de equipos en un ambiente controlado de energía, climatización y accesos. Estos se encuentran en constante crecimiento debido a las necesidades de procesar una cantidad de información cada vez mayor y con mejores velocidades. (Jirovsky & Mehra, 2015). Estos edificios cuentan con niveles de seguridad especiales en donde sus clientes trasladan el aspecto tecnológico de sus negocios para evitar preocuparse de los aspectos como la compra de equipos de tecnología, regulaciones estatales y espacio útil en sus oficinas.

2.2.1 Breve introducción al estado actual de los centros de datos

Una forma de poder medir esa tendencia es observando la cantidad de tráfico en número y velocidad de puertos de los dispositivos conectados a la red, se podría concluir que la necesidad de ingresar datos que puedan ser procesados al interior de los centros de datos exige al negocio la implementación de nuevas tecnologías. En la figura 4 se puede observar cómo ha ido evolucionado la demanda de mayor velocidad en los centros de datos.

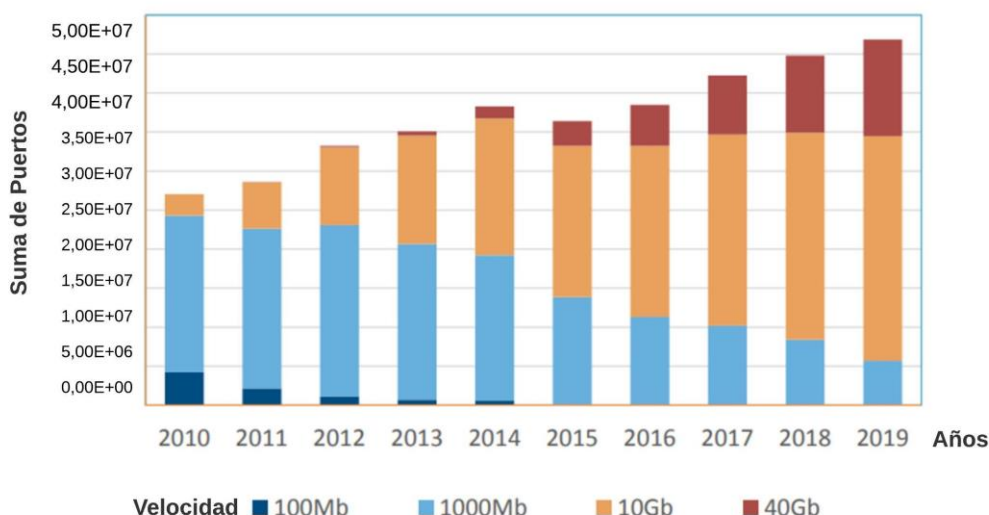


Figura 4. Comportamiento en la tendencia de utilización de puertos para transporte de información

(Jirovsky & Mehra, 2015).

En esta imagen se puede apreciar cómo la demanda de puertos de 100mb disminuye para el 2019 y se incrementa el uso de velocidades al aumentar el uso de puertos de 10 y 40Gb.

Con la imagen anterior se puede observar el aumento de las interfaces en su velocidad y además que también existe una demanda en la cantidad de puertos necesarios. Esto se traduce a un mayor número de equipos conectados en el centro de datos. Es necesario recordar que esto nos conduce a un aumento de la complejidad, dado que la misma aumenta cuando las necesidades del negocio también lo exigen (como lo explicado en la introducción). Además con esta imagen podemos adelantar, que el resto de los componentes arquitectónicos de un centro de datos deben ajustarse a estas necesidades y recibir incrementos, tanto en sus velocidades, como en sus capacidades. Se observan componentes como el almacenamiento que han ampliado su utilización en discos de mayor velocidad y capacidad, dejando a un lado el uso de discos rotacionales y reemplazándolos por los más modernos SSD³ o NVMe⁴ (Arman Shehabi et al., 2016). La figura 5 muestra como la estimación en la capacidad de los discos crece al igual que la demanda de puertos.

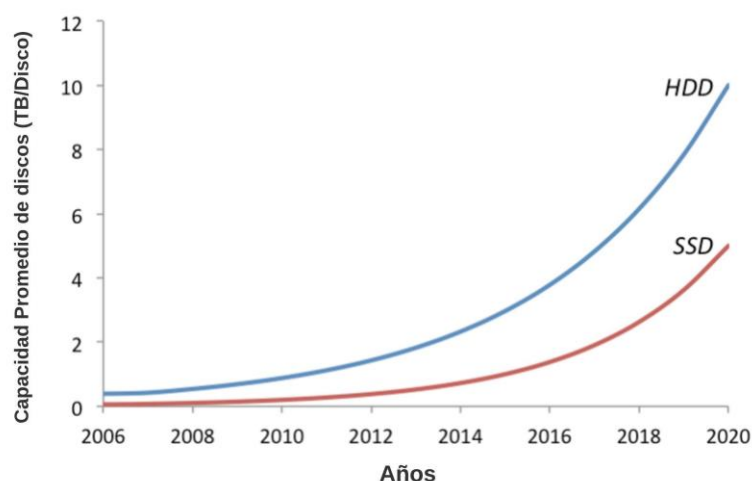


Figura 5. Estimación de capacidad de los discos duros en los centros de datos de Estados Unidos (Arman Shehabi et al., 2016).

Se puede observar que la capacidad de los discos duros tiene un crecimiento sostenido a lo largo del tiempo y que en los últimos años su capacidad ha aumentado, tanto para HDD (Hard Disk Drive) como para SSD (Solid State Drive)

2.2.2 Visión arquitectónica de los centros de datos.

La integración de los componentes antes mencionados, cómputo, almacenamiento y red de comunicaciones, de un centro de datos ya tiene lineamientos de la industria conocidos como

³ Solid State Drive

⁴ Non-Volatile Memory Express

“*best practices*”. Muchos de estos vienen de estándares como ITIL⁵, COBIT⁶, TOGAF⁷ y muchos más que podemos continuar enumerando.

Para la representación de los elementos y su interrelación me he basado en el estándar TOGAF y uno de sus apartados, el TRM⁸ (The Open Group, 2018) el cual permite modelar mediante bloques de construcción la arquitectura de los componentes de tecnología del centro de datos de una manera estándar.

La figura 6 muestra el TRM mencionado con los componentes principales al interior del centro de datos y cómo estos se relacionan.

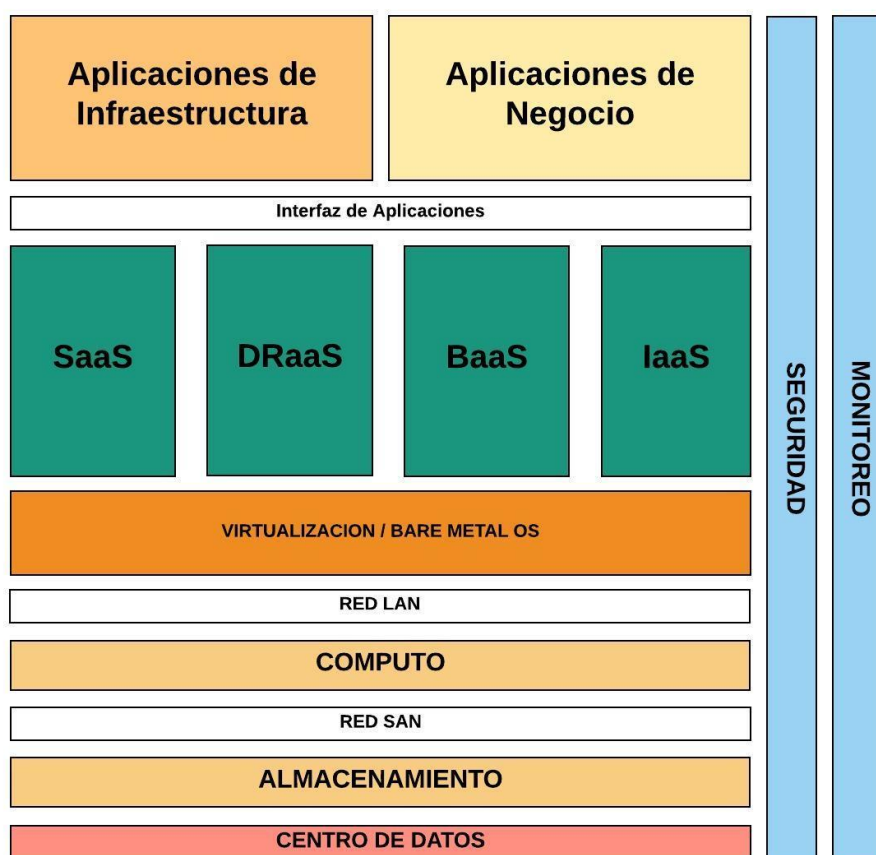


Figura 6. TRM clásico de un centro de datos

(Elaboración Propia).

Con la ayuda de este modelo técnico de referencia es posible visualizar en alto nivel los diferentes componentes de un centro de datos y sus relaciones, en la parte más baja se encuentran componentes como el almacenamiento, cómputo y la red. Componentes críticos de la operación al ser la base de las aplicaciones que están en la parte superior.

⁵ Information Technology Infrastructure Library

⁶ Control Objectives for Information and Related Technologies

⁷ The Open Group Architecture Framework

⁸ Technical Reference Model

En la parte media del TRM, en color verde, se encuentran los servicios clásicos de “*Cloud Computing*”⁹, los llamados “*as a Service*”, de izquierda a derecha, *Software*, *Disaster Recovery*, *Backup* y *Infraestructure*. Sobre este último se ampliará la definición de lo que comprende y cómo se plantea resolver la complejidad de ejecutar o no el mantenimiento de sus componentes mediante el uso de agentes autónomos.

2.2.3 Computación de Nube como servicio

Los recursos de computación que se mostraron en el TRM son limitados, no existe un lugar en el que se pueda concentrar una capacidad infinita de estos recursos. Como solución a este problema existen hoy en día, tecnologías que permiten agrupar los sistemas desde sus componentes básicos de representación (CPU, RAM, Disco Duro y Red), como si fuesen uno solo.

Este tipo de acciones se los conoce con el anglicismo de “*clusterización*” que comprende la suma de varios ordenadores empresariales y que, con la ayuda de software especializado pueden simular que superan sus limitaciones físicas. En definitiva, permiten simular que existe una gran capacidad de recursos dentro de una misma entidad lógica.

A partir de estas agrupaciones nace el primer servicio básico de la computación en la nube, el llamado IaaS¹⁰ (Ibrahim Omer et al., 2014), sobre el cual se soportan otros servicios como SaaS¹¹, PaaS¹², y los demás servicios de computación en la nube. La figura 7 muestra una ejemplificación detallada de cómo es la interrelación de estos servicios. Anteriormente ya se pudo ubicar algunos de estos en el TRM.

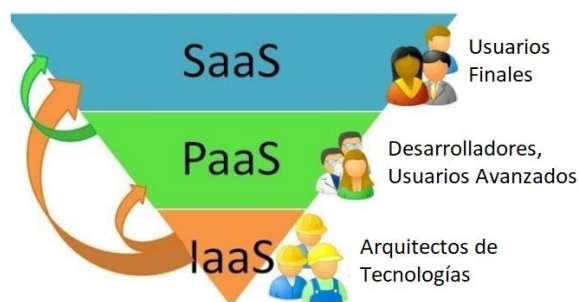


Figura 7. Posicionamiento estructural de la relación en los diferentes servicios de computación en la nube

(Ibrahim Omer et al., 2014).

La agrupación de los componentes de almacenamiento, cómputo y red son la base de los servicios de computación en la nube. De estos, el IaaS es la base sobre la cual los otros servicios reposan.

⁹ Computación en la nube

¹⁰ Infrastructure as a Service

¹¹ Software as a Service

¹² Platform as a Service

En la industria existen los grandes proveedores de servicios de computación en nube como Google, Amazon y Microsoft Azure, quienes dominan la participación de mercado de estos servicios. Sin embargo, esto es, principalmente en Estados Unidos. Paralelamente en los distintos países existen proveedores de nube que proporcionan servicios similares de manera personalizada. Muchas veces estos sufren de crecimiento exponencial y su administración se escapa de sus propias manos. La figura 8 muestra el posicionamiento de los líderes en este aspecto de acuerdo al informe de Gartner 2019. (Gartner Inc, 2019)

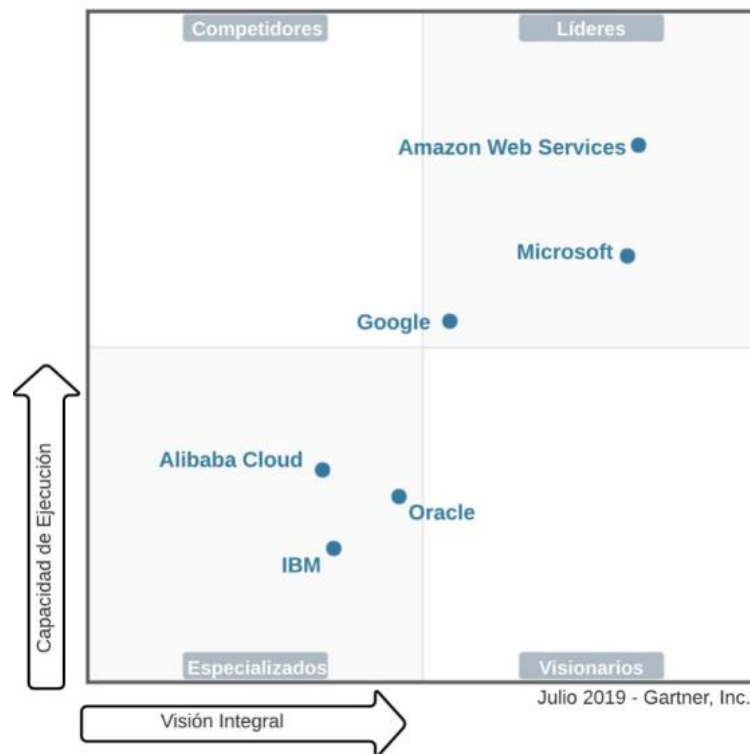


Figura 8. Informe de IaaS proporcionado por Gartner
(Gartner Inc, 2019).

El cuadrante mágico de Gartner es un gráfico que da a conocer quiénes son los principales jugadores en algún ámbito de la tecnología. En esta imagen se visualiza los principales participantes Amazon, Microsoft y Google, en el campo de los servicios de IaaS en la computación de nube.

Uno de los mencionados anteriormente, Amazon, se autoproclama así mismo como una compañía que proporciona además de servicios de IaaS, servicios de IA (US. Securities and Exchange Commission, 2018) y que las líneas de negocio como ventas en línea y servicios de computación en nube son servicios agregados que se apalancan sobre ese servicio y viceversa. Con ese ejemplo la industria de centros de datos ahora está cambiando su estructura para incluir como línea de negocio a la IA y que exista en el ecosistema del resto de servicios tradicionales.

Finalmente, hay que recordar nuevamente, que se debe prestar atención al servicio IaaS, pues es aquí donde lo desarrollado en este documento plantea usar elementos Autónomos que usen estrategias de IA que permitan inferir automáticamente la motivación de las máquinas de auto repararse.

2.2.4 Mantenimiento de Infraestructura

Uno de los desafíos en la operación de centros de datos, para el mantenimiento de los servicios de computación de nube es la planificación y operación de su mantenimiento. Es necesario que los centros de datos establezcan un marco de trabajo que les permita analizar y comprender los niveles de fallos de los distintos componentes y su disponibilidad. (Fadaeefath Abadi et al., 2020)

Generalmente la ejecución de los mantenimientos requiere validaciones transversales y prerequisites que son ejecutados por un operario humano, quien a pesar de tener un manual reglamentario en el cual puede seguir los pasos adecuados para ejecutar un mantenimiento, tiene la facilidad de poder tener una visión holística del entorno y en casos de peligro poder determinar si un mantenimiento es viable o no, con el fin de mantener la integridad del sistema y los servicios. De igual forma es sujeto a tomar una mala decisión y fallar (Wiboonrat, 2020). La figura 9 muestra una colección de las principales causas que provocan caídas de servicio en los centros de datos.

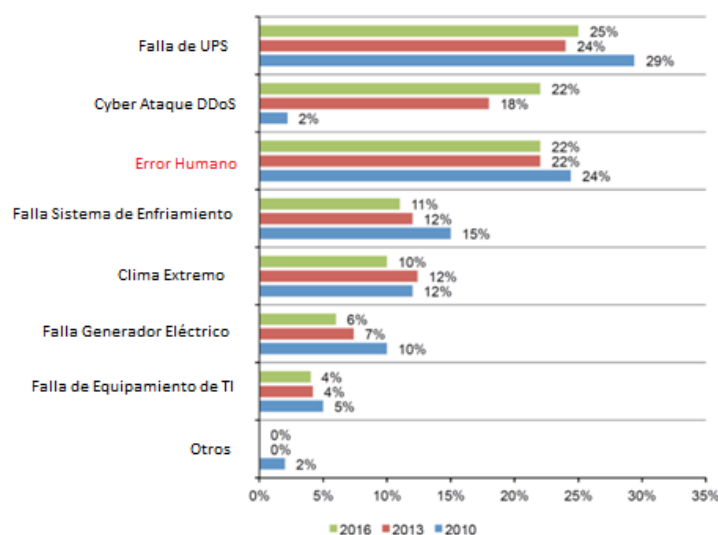


Figura 9. Principales causas de indisponibilidad en los servicios de centro de datos (Wiboonrat, 2020).

En esta imagen se observa los porcentajes de errores humanos. La muestra del año 2010, en donde se podría presumir que la acumulación de conocimiento y experiencia de años anteriores sería un apoyo para bajar ese porcentaje, sin embargo, la realidad es que aumentó, debido al incremento de la complejidad en la infraestructura.

2.3 Computación Autonomica

La computación autónoma es la conceptualización de una computación automática, sin intervención humana o que requiere lo mínimo de ella. Su principal objetivo es liberar a los administradores de TI de la complejidad que los entornos de cómputo pueden llegar a tener (IBM, 2005).

En el mismo sentido la computación autónoma es un proceso que requiere una evolución progresiva dentro de la organización (Brittenham et al., 2007). De la arquitectura, estándares y tecnología que los administradores de TI deben manejar.

En los siguientes apartados se dará una introducción a sus conceptos y el sentido que toma dentro de la solución propuesta.

2.3.1 Fuente de inspiración de la Computación Autonomica

El cuerpo humano y toda su representación siempre ha sido objeto de los más profundos estudios. Una maquinaria biológica y con sus diversos órganos trabajando en sinergia demuestran la perfecta armonía en la que cada uno se complementa. (Manoel et al., 2005) (Jeffrey & David, 2003) (Lalanda et al., 2013a)

Estos órganos se pueden agrupar como sistemas o aparatos dependiendo del tipo de tejido que los conforman (Aragort, 2009). Particularmente, en el estudio de la inteligencia artificial, que tiene como inspiración el tratar de emular las conductas humanas, es necesario enfocarse en dos de estos:

- El sistema nervioso central (SNC)
- El sistema nervioso autónomo (SNA - También llamado vegetativo)

El sistema nervioso central o (SNC) y que tiene como órgano principal al cerebro, es a nivel humano donde se genera la capacidad de pensamiento, las ideas, el razonamiento y por supuesto la toma de decisiones, esto es lo que los humanos hemos definido o calificado como "Inteligencia". Origen de la inspiración necesaria en el hombre de tratar de emular este comportamiento en las máquinas.

Por su parte el sistema nervioso autónomo (SNA), que está relacionado con el SNC está encargado de controlar y estimular las funciones necesarias vitales para la subsistencia del

cuerpo, los latidos del corazón, respiración, procesamiento digestivo, entre otras funciones, de las cuales un mal funcionamiento podría ser muy grave e incluso fatal.

De acuerdo con la figura 10 se puede observar las diferentes subdivisiones del sistema nervioso y las partes de su interacción

Sistema Nervioso	Sistema Nervioso Central (SNC)	Cerebro	Cerebro Anterior	Telencéfalo	Neocortex
					Ganglios Basales
					Sistema Límbico
				Diencéfalo	Tálamo
					Hipotálamo
			Cerebro Medio		
	Sistema Nervioso Periférico	Médula Espinal	Cerebro Posterior	Cerebelo	
				Protuberancia	
				Bulbo	
		Somático			
		Autónomo	Simpático		
			Parasimpático		

Figura 10. Sistemas y subsistemas del sistema nervioso central

(Aragort, 2009).

Leyendo de derecha a izquierda podemos observar cómo el sistema nervioso humano se compone de múltiples sistemas. Del SNC sistema nervioso central que sirve como inspiración para la IA en la parte superior y el autónomo como una subdivisión del sistema nervioso periférico se encarga de las funciones básicas del cuerpo humano.

Este sistema tiene la particularidad de trabajar en sinergia total con el sistema nervioso central, por cuanto cualquier señal de aviso o alarma de mal funcionamiento de alguno de sus órganos gobernados, llega hasta el cerebro, de tal forma que pueda tomar acciones conscientes o inconscientes para su reparación.

En computación un elemento autonómico tiene esa misma particularidad, recibir señales para ser procesadas dentro de un bucle de control. Sobre este particular se tratará en el apartado 2.3.3.

2.3.2 Computación autónoma y Computación tradicional

Uno de los puntos destacados que se deben tener en consideración cuando se habla de computación autónoma, es la automatización que inserta a las operaciones de TI, donde conceptualmente propone el ajuste automático a entornos cambiantes.

La computación autónoma basa su construcción en cuatro grandes atributos de acuerdo con (Jeffrey & David, 2003), los cuales se pueden apreciar en la figura 11.

De acuerdo con (Obienu, 2018), y como ya lo habíamos mencionado con anterioridad, el objetivo final de la arquitectura de la computación autónoma es reducir la intervención de los operadores llevando a cabo las funciones administrativas conforme a políticas predefinidas que se encuentren en el sistema.

Es comprensible que estos cambios no se pueden dar de un momento a otro, se necesita una revolución tecnológica que lo permita. Por otro lado, la inclusión de estos conceptos en los ambientes de computación tradicional debe ser llevado de manera gradual (Manoel et al., 2005) permitiendo que exista cohesión entre los modelos de trabajo sin que se provoque un rechazo a la inclusión de nuevas tecnologías

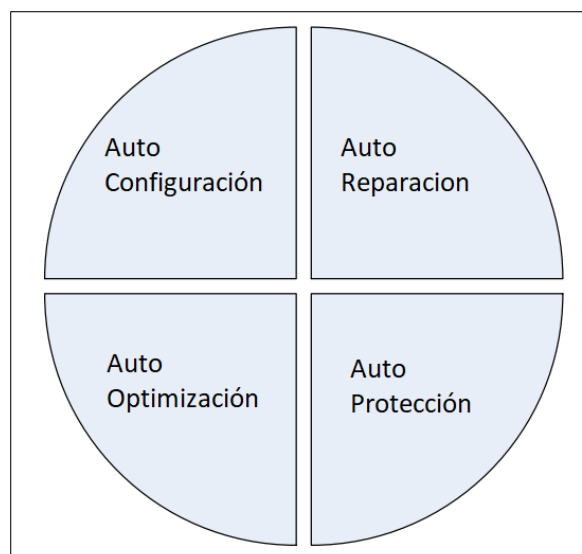


Figura 11. Atributos de la Computación Autónoma

(Manoel et al., 2005)

La computación autónoma tiene cuatro atributos que se pueden observar en la figura. Uno de ellos, es el de la Auto reparación (*"Self Healing"*). Con ese concepto se pretenderá otorgar a los elementos autónomos, capacidades de análisis en función de saber si es seguro o no llevar a cabo una reparación.

A continuación, se mostrará en una tabla la ampliación de estos conceptos y se incluirá el correspondiente concepto de la computación tradicional para introducir al lector en los conceptos de la computación autónoma.

Tabla 1. Conceptos básicos de la computación autónoma y su comparación con los de la computación tradicional.

Concepto	Computación Tradicional	Computación autónoma
Auto Configuración	Centro de Datos Corporativos que tienen múltiples equipos de diversos fabricantes. Instalar, configurar e integrar sistemas consume tiempo	Configuración automatizada de los componentes y sistemas con la ayuda de políticas de alto nivel. El resto de sistema se ajusta perfecta y automáticamente.
Auto Optimización	Sistemas tienen miles de configuraciones manuales, parámetros de ajuste no lineales y su número incrementa en cada nueva versión.	Los componentes y sistemas buscan continuamente oportunidades para mejorar su desempeño y eficiencia.
Auto Reparación	La determinación de problemas es larga, sistemas complejos pueden tomar a los operadores y programadores semanas en lograr encontrar el fallo.	Los sistemas detectan, diagnostican y reparan automáticamente los problemas de software y hardware
Auto Protección	Detección y recuperación de ataques y errores de cascada de forma manual	El sistema se defiende automáticamente contra ataques maliciosos. Utiliza la alerta temprana para anticipar y prevenir fallas en todo el sistema.

(Jeffrey & David, 2003)

2.3.3 Arquitectura de elementos autonómicos

Los elementos de la computación autonómica se podrían explicar cómo componentes de dos niveles: En el nivel superior se encuentra el Administrador Autonómico, mismo que es gobernado por el bucle de control MAPE-K. En el nivel inferior es donde se encuentra el Elemento Administrado que está emitiendo y recibiendo señales hacia los sensores y desde los actuadores del Administrador autonómico. Es importante destacar que el elemento administrado puede ser un solo componente o muchos y en el caso de los centros de datos, estos generalmente son muchos y heterogéneos. (IBM, 2005) La figura 12 nos presenta un diagrama de cuáles son sus componentes y cómo interactuaría con los elementos del centro de datos.

El Administrador Autonómico, tiene cinco secciones principales del ciclo de control MAPE-k (Lalanda et al., 2013b):

- **Monitoreo:** La función de monitoreo recolecta los detalles de los recursos administrados para ser entregados a la función de análisis. Estos datos pueden incluir detalles como cambios de temperatura, estado, eficiencia, desgaste. La información que se recolecta aquí también puede ser pre procesada si fuese necesario, antes de que pase al análisis.
- **Análisis:** La función de análisis debe proporcionar mecanismos que permitan determinar si algún cambio debe ser realizado (siendo este el apoyo de la motivación de este proyecto) y de igual forma poder cancelar el cambio en caso de que pueda atentar contra su propia existencia. Esta función debe ser capaz de poder realizar procesamiento inductivo y razonal de aprendizaje del ambiente de TI e inclusive poder predecir futuros comportamientos.
- **Planeación:** La función de planeación selecciona un procedimiento existente de la base de conocimientos para su ejecución de acuerdo con las entradas recibidas por parte de análisis. Los planes de acción pueden ser estáticos o dinámicos. (Lalanda et al., 2013b)
- **Ejecución:** La función de ejecución se comunica directamente con los actuadores y tienen el lenguaje que estos entienden. Por ejemplo, de ser algún tipo de válvula de presión que entiende un determinado lenguaje de programación, esa información se encuentra en este nivel.

- **Conocimiento:** El conocimiento es una base de datos en los cuales se puede recurrir para conocer de experiencias pasadas, información de expertos, o cualquier otro conocimiento que pueda servir para que el elemento administrado sea operado de manera eficiente por medio del administrador autónomo.

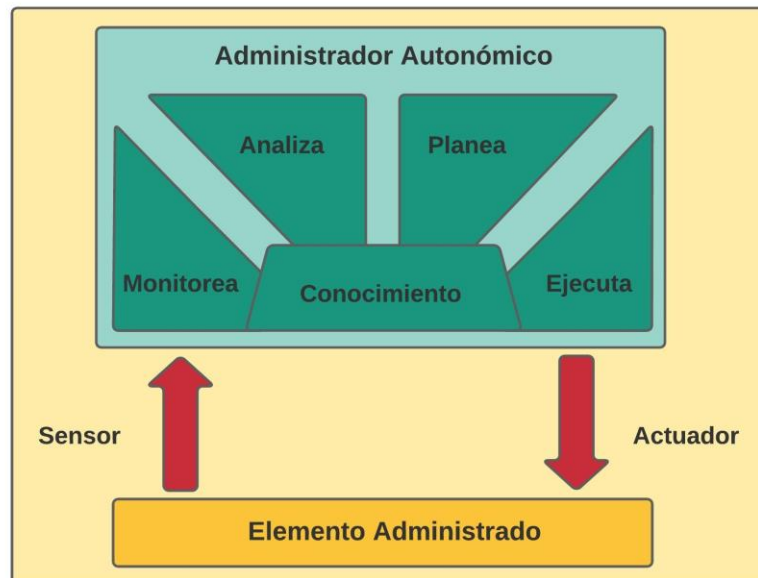


Figura 12. Estructura de un elemento autónomo.

Adaptado de (Jeffrey & David, 2003)

Estructura base de un elemento autónomo. Se puede visualizar los cuatro ciclos del bucle de control de MAPE-K. El bloque que será intervenido en este proyecto es el de análisis.

2.4 Inteligencia Artificial

De la IA podemos plantear la definición: “Un programa que, en un mundo arbitrario, no se las arreglará peor que un humano” (Dobrev, 2012) y podría decirse que abarca las siguientes disciplinas de acuerdo a la figura 13 a continuación (Kraus, 2020).

La inteligencia artificial tiene su inspiración en el sistema nervioso central, del cual el componente más importante es la neurona, de acuerdo con (Velayos & Diéguez, 2015) esta constituye el elemento básico del Sistema Nervioso. Tenemos dos clases de ellas que se pueden clasificar como excitatorias e inhibitorias y utilizan impulsos para transmitirse mensajes entre ellas y comunicarse. Esta es la base de las redes neuronales sobre las cuales hablaremos más adelante como parte del fundamento de este documento de investigación.



Figura 13. Disciplinas más relevantes de IA.

(Kraus, 2020)

La inteligencia artificial es una ciencia que abarca algunas disciplinas como el aprendizaje automático, el aprendizaje profundo y visión por computador. Las reglas del negocio pueden modificar el alcance que se le quiera dar a la IA en una organización.

2.4.1 Lógica Borrosa

La lógica borrosa o difusa, es una declaración matemática de lógica multivaluada presentada por el matemático Lotfi Zadeh en 1965, (L. A. Zadeh, 1965). Esta teoría aproxima de manera matemática los motivos por los cuales las personas podrían tomar una decisión basada en información imprecisa.

Famosos problemas como el del vaso medio lleno o medio vacío, o este cuarto es frío, tibio o caluroso, son clásicos en este campo. Su funcionamiento se basa en reglas preestablecidas que pueden venir del conocimiento de expertos en el tema, o de la simple observación de los fenómenos que pueden ocurrir. A diferencia que, en la lógica booleana, las respuestas nunca serán si y no, sino, estados intermedios entre estos como mucho, poco, muy poco o demasiado.

En los problemas relacionados con la ingeniería se utiliza principalmente para manejar la incertidumbre o porque es imposible tener una respuesta concreta para una determinada situación que siempre ha sido controlada por humanos expertos (Khalifa & Frigui, 2015). La figura 14 muestra la arquitectura tradicional de un sistema de lógica borrosa.

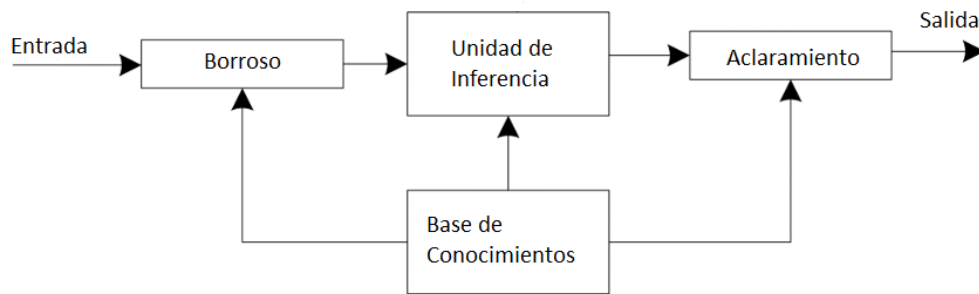


Figura 14. Componentes principales de un Sistema Mamdani de Inferencia Difusa
(Khalifa & Frigui, 2015)

En la entrada de estos sistemas la información borrosa (fuzzyfication) se procesa por medio de reglas precisas alimentadas por expertos que se encuentran en una base de conocimientos. El Sistema de inferencia donde se realiza el razonamiento y una fase de aclaramiento (defuzzyfication) que entregue el valor de las aproximaciones calculadas.

Uno de los objetivos de un sistema de inferencia difusa es poder determinar el grado de pertenencia de un dato dentro de los conjuntos de datos borrosos. Existe para esto ayuda visual con las llamadas “funciones de pertenencia” en la cuales se define un espacio de la pertenencia valuado entre cero y uno. La figura 15 muestra un ejemplo de cómo se aprecia esta función de pertenencia.

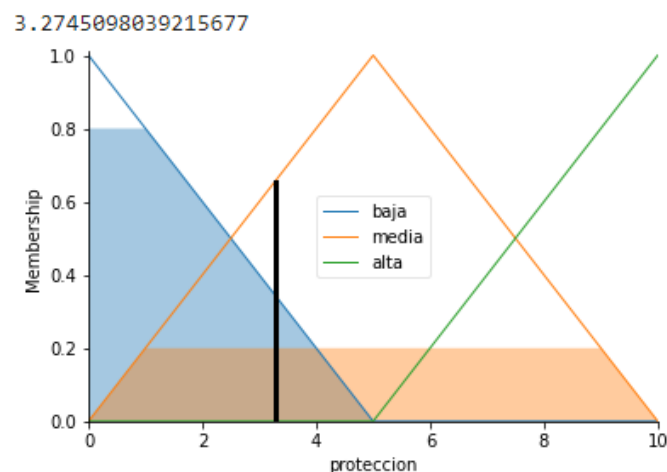


Figura 15. Resultado de la función de pertenencia.
(Elaboración Propia)

En el diagrama se puede ver cómo trabaja una función de pertenencia. Para este ejercicio se está comprobando algunas entradas a medir, entre ellas la protección (redundancia y disponibilidad) de un determinado componente del centro de datos son adecuados. De acuerdo con el resultado observado en la imagen, la protección para ese elemento es “medio baja” y su valor continuo, resultado del aclaramiento (defuzzyfication) es 3.27 en una escala de 1 a 10. Por lo cual podríamos concluir que al elemento de este ejemplo *“le falta”* subir su nivel de protección.

La lógica difusa se utiliza en el campo de la inteligencia artificial como una forma poderosa de modelamiento de problemas computacionales que permitan manejar la incertidumbre y la imprecisión efectivamente. (Lotfi A. Zadeh, 1975). En esta investigación esta es una de las partes fundamentales por cuanto se pretende trasladar el conocimiento de expertos en manejo de reparaciones de los elementos de cómputo de un centro de datos a un sistema neuronal difuso, del cual se ampliará su contenido en el apartado 2.4.3 y 2.5

2.4.2 Redes Neuronales

De acuerdo con (Javier et al., 2009) una red neuronal artificial RNA, es un modelo matemático inspirado en la compleja red de neuronas del sistema nervioso central y como se encuentra estructurado el Cerebro.

La primera propuesta en este sentido fue un esfuerzo multidisciplinario de dos científicos en el año 1943, Warren McCulloch psiquiatra y estudioso de la anatomía del cerebro y Walter Pitts, matemático (McCulloch & Pitts, 1943) . La figura 16 muestra lado a lado una comparativa entre el diseño biológico y el artificial.

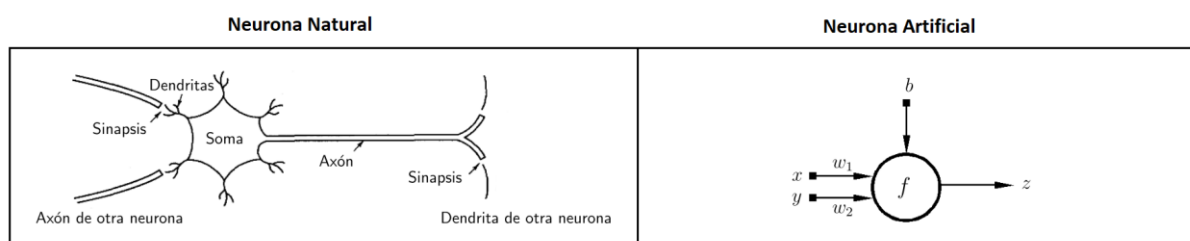


Figura 16. Diagrama comparativo entre una neurona natural y una artificial.

(Javier et al., 2009)

Podemos observar a la izquierda, la neurona natural la cual se puede dividir en tres partes principales, las dendritas, filamentos de conectividad hacia el axón de otra neurona, el soma, parte central de neurona y el axón, componente que se utiliza para conectar hacia las dendritas de otras neuronas. A la derecha, la neurona artificial, que reciben sus entradas en x e y con una salida z que es tomada por la siguiente neurona. En el centro la función de activación f de acuerdo con la actividad que realice la neurona.

El modelo de redes neuronales profundas que se utilizan en la actualidad se trata de grandes agrupaciones de estas neuronas formando diversas capas. Inicialmente es deseable que existan al menos tres capas, una capa de entrada, una capa oculta y una capa de salida. El número de capas final que tenga una red neuronal dependerá del tipo de trabajo que esta realice (Karadurmuş et al., 2019). La figura 17 muestra la forma básica de este tipo de redes.

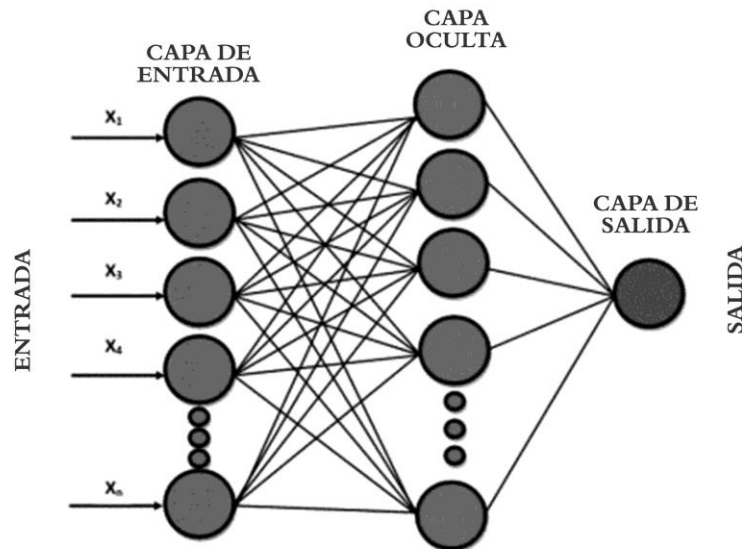


Figura 17. Estructura básica de una red neuronal artificial.
(Karadurmuş et al., 2019)

Nótese como las redes neuronales se componen de múltiples neuronas artificiales ordenadas por capas. Dependiendo del problema que se quiera resolver estas tendrán más o menos neuronas en sus diferentes capas.

2.4.3 Sistema de Inferencia Neuro Difuso

Las redes neuronales y los sistemas de lógica difusa tienen cosas en común (Kruse, 2008). Pueden ser usados ambos sistemas para resolver distintas clases de problemas, sin embargo, tienen sus respectivas ventajas y desventajas. Estas diferencias se pueden entender mejor al observar la tabla 2.

Tabla 2. Comparativa de las redes neuronales y los sistemas de control difuso.

Redes Neuronales	Lógica Difusa
No se requiere un modelo matemático	No se requiere un modelo matemático
Aprenden desde cero	Se requiere conocimiento de expertos
Muchos algoritmos de Aprendizaje	No son capaces de aprender
Comportamiento de Caja Negra	Interpretación e Implementación sencillas

(Kruse, 2008)

Las redes neuronales de inferencia neuro difusa se pueden usar tanto para problemas de clasificación y de regresión. De acuerdo con (Liu & Li, 2004) se podría establecer la siguiente clasificación observada en la tabla 3.

Tabla 3. Clasificación de Redes Neuronales Difusas

Superclase	Clase	Tipo	Década de Aparición
Red Neuro Difusa	Basada en operadores difusos	Redes neuronales de Avance (FeedForward)	1980
		Redes neuronales de Retorno (Feedback)	1990
	Redes neuronales Borrosas	RND Regulares	1990
		RND Mejoradas	1990
	Redes Difusas de Inferencia	Mamdani	1990
		Takagi-Sugeno	1990
		Generalizadas	1990

(Liu & Li, 2004)

De las últimas de la tabla, las Redes Difusas de Inferencia serán las utilizadas en este trabajo de investigación. Las de tipo Mamdani, se promueven como redes simples y de fácil interpretación (Kruse, 2008) son ampliamente usadas en el traslado del conocimiento y experiencia de humanos expertos en un conjunto de reglas que permita la fase de razonamiento dentro de un sistema de inferencia difuso (Mamdani & Assilian, 1975).

Las redes de tipo Takagi-Sugeno se caracterizan por que la propagación sucede en dos pasos, combinando algoritmos de backpropagation con la estimación de los cuadrados medios. En la identificación de requisitos se abordará las redes tipo ANFIS que implementan este modelo y que se utilizaron como la propuesta de solución al problema tratado en esta investigación.

La figura 18 muestra en una vista de alto nivel el proceso de inferencia en un sistema difuso.

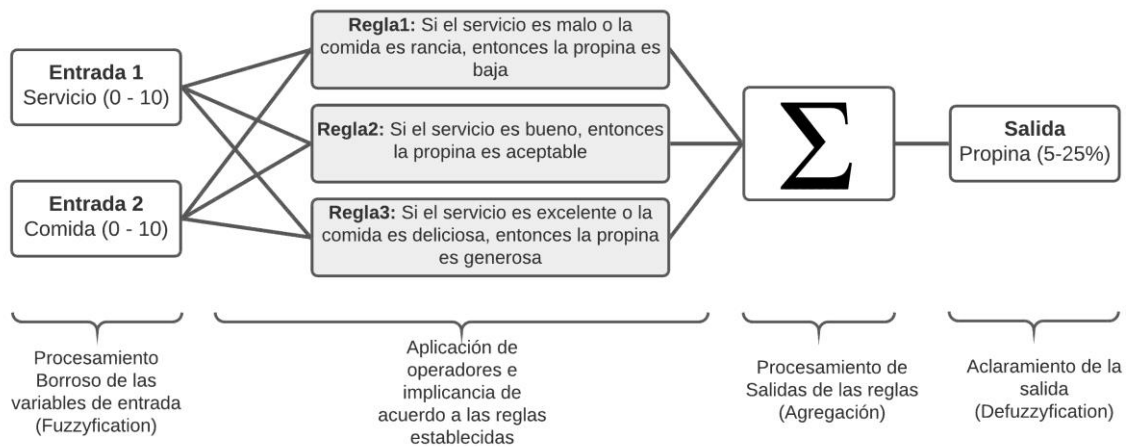


Figura 18. Proceso de inferencia de un sistema de lógica difusa.
Adaptado (Mathworks, 2020)

El clásico problema de una cena para dos, tratado con la conceptualización base del inicio de una red neuronal. En el mismo tenemos una capa de entrada en donde se procesan las variables, una capa intermedia en donde se procesan las reglas del sistema, una capa de agregación de las reglas de la capa anterior y una capa de salida con el resultado final del proceso.

En los sistemas de inferencia difusa se podría decir que el proceso de inferencia se lleva a cabo en 5 partes generalizadas (Mathworks, 2020):

1. **Procesamiento Borroso de las variables de entrada (Fuzzyfication):** Las variables se procesan de acuerdo con una valoración imprecisa de sus clases y se las ingresa en la función de pertenencia correspondiente (Ejem. bajo, medio, alto) o las que exista para el problema que se esté tratando.
2. **Aplicación de un operador (AND / OR) en el antecedente:** A la variable ingresada al sistema y que pasa el procesamiento anterior, se le aplica las reglas de operación, esta es la primera parte del razonamiento en un sistema de tipo Mamdani. Por ejemplo: si es muy gordo o alto, si es flaco y bajo.
3. **Aplicación del método de implicancia:** El consecuente viene acompañado de la preposición “entonces” después de la combinación de antecedentes mediante operadores difusos. Por ejemplo: si es flaco y alto entonces es flexible.
4. **Procesamiento de las salidas:** Este paso es la evaluación combinada (agregación) de las reglas que tenemos en nuestro sistema de inferencia difuso.
5. **Aclaramiento de la salida (Defuzzyfication):** Sobre el resultado de la agregación que se obtuvo en el procesamiento de las salidas, se obtendrá un valor único que

es la evaluación final del proceso de lógica borrosa que se quiera aplicar a algún problema dado.

2.5 Estado del Arte

Durante el inicio de la década de los 90 en el siglo XX, el interés en las redes neuronales fue muy activo, en este aspecto las Redes Neuronales de Inferencia Difusa no se quedaron atrás. En el año 1993 (Jang, 1993) aparece la propuesta de una red integrada llamada ANFIS con el objetivo de combinar lo mejor de los mundos de las redes neuronales y la lógica difusa. La figura 19 muestra un ejemplo de la conceptualización de la red.

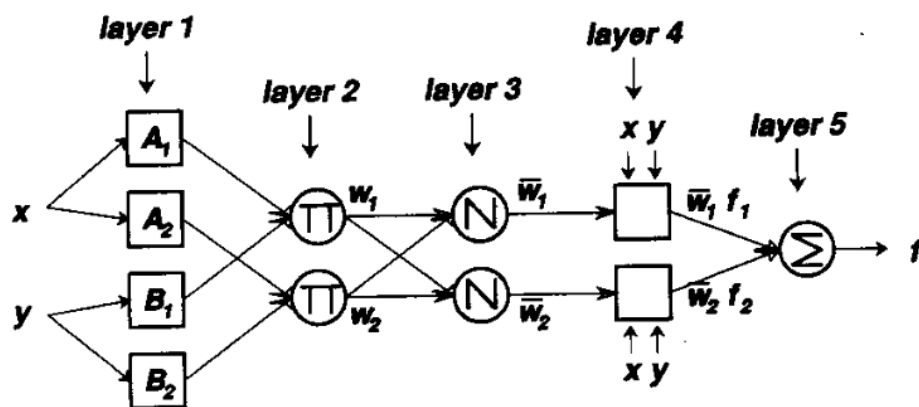


Figura 19.Red neuronal tipo ANFIS.

(Jang, 1993)

Las redes neuronales difusas de tipo ANFIS se componen de 5 capas. La primera capa es la asignación a los diferentes pesos de las funciones de pertenencia de cada variable ingresada. En la capa 2 se procesan los consecuentes de las reglas, en la capa 3 se realiza un proceso de normalización de la salida de la capa anterior, en la capa 4 se procesan los consecuentes de las reglas y finalmente en la capa 5 sucede la agregación de todas las salidas de las capas anteriores.

En el año 1995 se propuso resolver problemas de clasificación mediante redes neuronales difusas (Nauck & Kruse, 1995). La principal aportación fue establecer un modelo neuronal para clasificación derivado del modelo genérico del perceptrón difuso de tres capas y la inclusión de algoritmos de aprendizaje supervisado para adaptar los parámetros de las funciones de pertenencia para múltiples clases. La figura 19 muestra visualmente la estructura de la red.

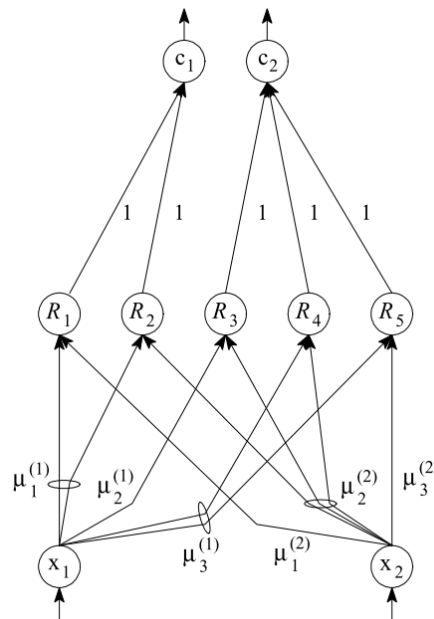


Figura 20. Sistema de Red Neuronal NEFCLASS.

(Nauck & Kruse, 1995)

En la gráfica se puede observar una red neuronal difusa de clasificación llamada NEFCLASS y que tiene tres capas. En la primera capa dos neuronas de entrada, en la segunda capa cinco neuronas que contienen reglas a procesar y una capa de salida con dos clases

La aplicación con redes neuronales para el control de elementos autónomos se observa en el año 2011, (Mousa Al-Zawi et al., 2011) se realizan investigaciones sobre la aplicación de redes neuronales en elementos autónomos que estén enfocados en realizar la auto reparación. En sus experimentos utilizan redes PRNN¹³ modificadas que permitan controlar y mantener el ancho de banda de forma óptima para los clientes que se conectan a un grupo de servidores. Sus hallazgos fueron positivos llegando a la conclusión de que es viable alimentar elementos autónomos con la capacidad de inferencia de redes neuronales.

Posteriormente en el año 2015 (Magradze, 2015), en su tesis doctoral realiza la investigación de combinar elementos de computación autónoma con redes neuronales difusas adaptativas ANFIS que permitan facilitar la administración de los componentes del centro de datos Atlas Tier 2 Center GoeGrid. Pudo llegar a la conclusión de que las redes tipo ANFIS son viables para realizar lo antes expuesto. En la figura 20 se puede observar su propuesta para el sistema de administración en entornos de gran escala.

¹³ Pipeline Recursive Neural Network

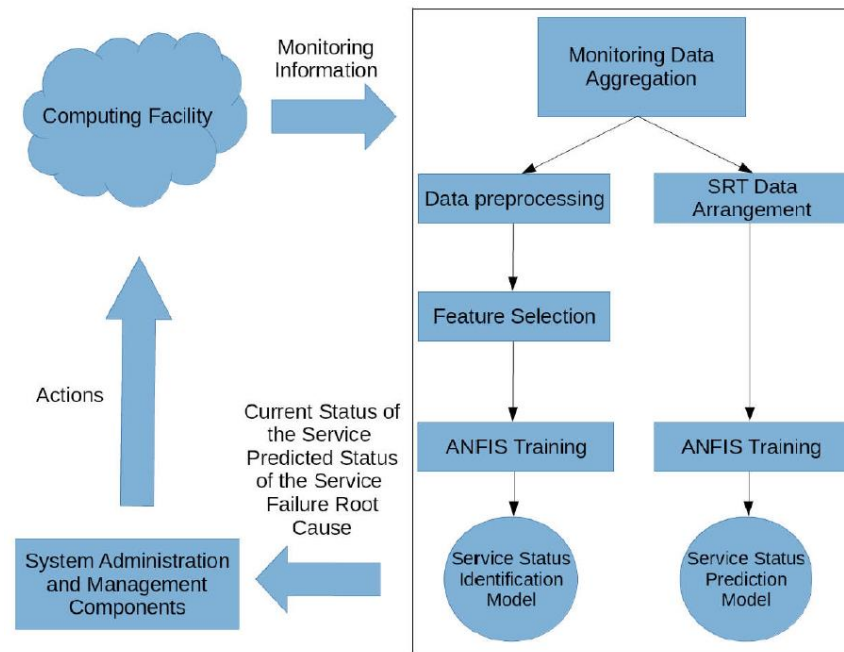


Figura 21. Enfoque de la integración de ANFIS en la administración de sistemas de computación de gran escala
(Magradze, 2015)

En la propuesta se tienen dos redes neuronales difusas tipo ANFIS, en donde la de la derecha se encarga de predecir los tiempos de respuesta de aprovisionamiento de servicios en caliente y la de la izquierda lo referente a la información previamente almacenada.

En el año 2019 (Naseri et al., 2019) en su lugar propone elementos autonómicos que en su fase de planificación de MAPE-K contengan redes neuronales de tipo ANFIS para procesar seis variables de un entorno controlado por humanos, específicamente el de válvulas de control para gas y petróleo. Las variables manejadas son entrada de presión, temperatura, velocidad de flujo, controlador de salida y desplazamiento de la varilla de válvula. En este ejercicio se tiene conocimiento que poder identificar correctamente el tipo de fallo de estas válvulas y asociar los diferentes escenarios de corrección de errores es únicamente del dominio de expertos.

Sus hallazgos demostraron que es viable utilizar redes neuronales difusas tipo ANFIS en este tipo de escenarios, sin embargo, existieron limitaciones que estuvieron relacionadas con la sensibilidad de la industria del gas.

3. Objetivos y metodología de trabajo

Después de colocar en contexto el estado actual de las investigaciones realizadas en el ámbito de la computación autónoma en combinación con la inteligencia artificial y lo alcanzado en los servicios de nube y centro de datos, se plantean los siguientes objetivos y metodología de trabajo.

3.1. Objetivo general

Proponer la utilización de estrategias basadas en la inteligencia artificial dentro del funcionamiento arquitectónico de la computación autónoma permitiendo delegar a estos sistemas las tareas que hoy en día requieren la intervención de un humano debido a la complejidad de los sistemas involucrados.

3.2. Objetivos específicos

- Analizar la estrategia utilizada actualmente para realizar los procesos de mantenimiento a los elementos de un centro de datos.
- Diseñar la metodología a implementar para permitir que un agente inteligente tenga la capacidad de inferir si un mantenimiento es viable o no.
- Implementar la metodología con las distintas herramientas que se tenga a disposición.
- Evaluar los resultados obtenidos del caso base y el logrado.

3.3. Metodología del trabajo

Para alcanzar los objetivos planteados en este TFM se utiliza como metodología de trabajo la teoría que se utiliza para la construcción de sistemas expertos. Los Sistemas Expertos son piezas de software que se utilizan para ayudar a expertos en la toma de decisiones (Mousavi et al., 2017). La estructura básica de un sistema experto se compone generalmente de seis partes (Peijiang & Yongjun, 2013):

- 1) Las interfaces desde la cual se puede intercambiar datos o ingresar nueva información al sistema.

- 2) El sistema que devuelve los resultados al usuario con su respectiva explicación.
- 3) El motor de inferencia que simula el proceso de razonamiento de expertos.
- 4) La base de conocimientos que se utiliza para guardar el conocimiento empírico y las reglas construidas por expertos.
- 5) Una base de datos intermedia que permite trabajar con los resultados inmediatos entregados al motor de inferencia desde las reglas y datos nuevos entregados por los expertos.
- 6) La interfaz de adquisición de conocimientos desde la cual se ingresa la información de los expertos.

La figura 22 muestra gráficamente como se interrelacionan estas partes.

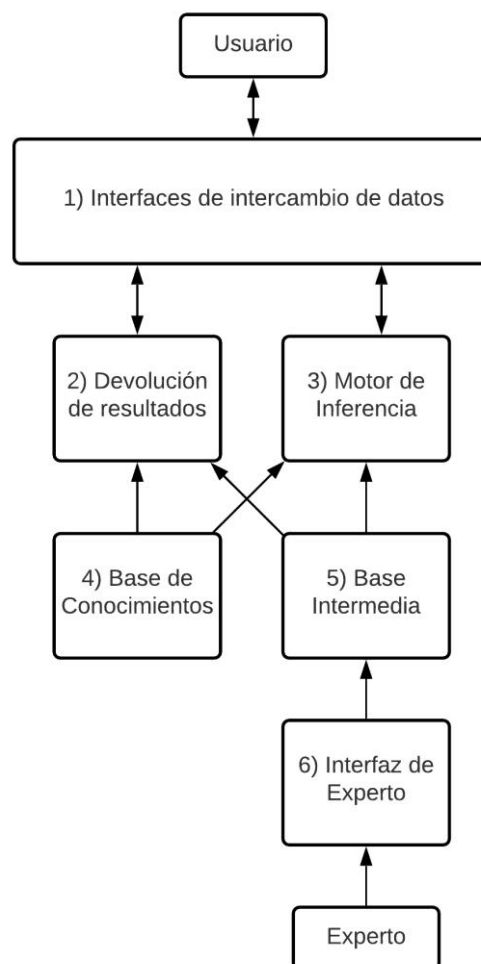


Figura 22. Componentes de un sistema Experto
(Peijiang & Yongjun, 2013)

Nótese en la imagen las flechas de dirección que indican el flujo de información. Si un usuario utiliza el sistema experto, tiene dos caminos: inferir un resultado en base al conocimiento ya establecido o solicitar nuevos resultados en base a datos nuevos. El experto siempre alimenta el sistema con datos nuevos.

Los sistemas expertos se construyen por medio de seis fases habituales que se muestran en la figura 23,

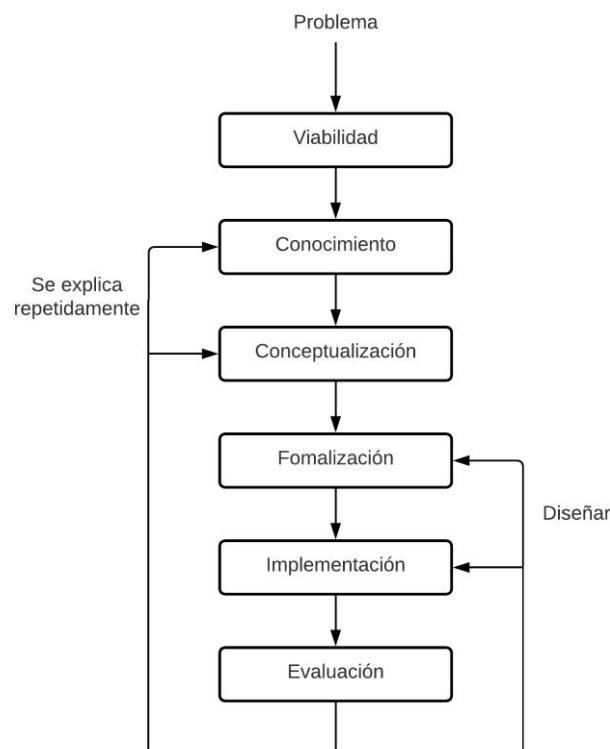


Figura 23. Proceso regular para la creación de un Sistema Experto

Adaptado de (Peijiang & Yongjun, 2013)

En la imagen se puede observar las distintas fases que pueden ser utilizadas para la creación de un sistema experto. Estas se recorren desde el problema y se puede recibir retroalimentación de regreso si en alguna fase es necesario adquirir más conocimiento o replantear el diseño.

Viabilidad: El presente estudio tiene un campo de aplicación muy amplio. La necesidad de especialistas en diferentes ámbitos siempre es necesaria, por eso la utilización de sistemas expertos es requerida en diferentes campos como medicina, telecomunicaciones, ingeniería, banca. Con la ayuda de este tipo de sistemas la toma de decisiones se acelera y facilita.

Conocimiento: Es importante que existan las suficientes fuentes de conocimiento para lo que se desea implementar. En esta fase se ha utilizado la información de los mantenimientos realizados en un centro de datos para el modelamiento de este sistema.

Conceptualización: La conceptualización de este tema tiene como su base los conceptos de la computación autónoma. Se utilizará la fase de análisis del bucle de control MAPE-K

Formalización: Mediante la utilización del ciclo de control MAPE-K y las redes neuronales difusas se plantea un diseño que puede ser reutilizable en diferentes aspectos de los centros de datos como el control de energía o climatización.

Implementación: El modelo desarrollado hará uso de herramientas como Python, Jupyter Notebook y Matlab que permitan la edición y actualización del sistema. Otras herramientas de análisis de datos y programación podrían ser utilizadas en un futuro.

Evaluación: La evaluación será llevada a cabo mediante la comprobación del comportamiento de la red neuronal ante nuevos datos sobre los cuales no ha recibido un entrenamiento previo. El conocimiento adquirido es importante en este punto.

4. Identificación de requisitos

Uno de los objetivos planteados en esta investigación, es el de analizar la situación actual de los centros de datos con referencia a la realización de mantenimientos y correcciones sobre sus componentes, este capítulo sentará las bases de la motivación objeto de la propuesta descrita en los siguientes capítulos.

4.1 Situación actual de operaciones del centro de datos

Dentro de los centros de datos pueden existir diversas formas de realizar los mantenimientos de la infraestructura de los equipos de cómputo. Algunas de estas pueden ser procesos manuales, semi automáticos o automáticos. Esto es un proceso evolutivo dentro de las organizaciones y específicamente para el caso de los elementos de infraestructura de cómputo me he apoyado en el marco de trabajo de la computación autónoma. La transformación autónoma de los centros de datos involucra cinco pasos de acuerdo a la tabla 4 (Manoel et al., 2005).

En el escenario cubierto en este documento se ha realizado un análisis dentro de un centro de datos que se encuentra actualmente entre el Nivel 1 Básico y Nivel 2 Administrado, por cuanto:

1. El personal se encuentra enfocado actualmente en realizar tareas operativas de mantenimiento, administración, creación de componentes relacionados con los servicios.
2. El personal debe especializarse en el uso de los equipos en lugar de especializarse en las necesidades del negocio.
3. El personal toma decisiones de cómo se debe proceder ante ciertos eventos.

Como se puede observar, el centro de datos está ciento por ciento operado por personal con experiencia y su nivel de automatización es muy reducido. Uno de los retos a los que se enfrentan actualmente es realizar el proceso de mantenimiento de la infraestructura, el mismo requiere una planeación cuidadosa por cuanto es vital identificar los componentes transversales y externos que podrían afectar un mantenimiento. Entre estos podríamos identificar: Trabajos de mantenimiento eléctrico en curso, manifestaciones, ataque informático, epidemias, días críticos de operación (fin de mes). Además de la correlación de los elementos base del TRM (visto con anterioridad en 2.2.2) y su impacto en las capas superiores del mismo.

En la investigación de campo se observó que en el centro de datos en cuestión se maneja la planificación mediante hojas de cálculo que son administradas por un coordinador, el mismo que asigna las tareas a realizarse al personal. El personal evalúa en base a su experiencia si el mantenimiento se puede dar y solicita la autorización a su correspondiente jefatura. Este de igual forma revisa el documento realizado por el ingeniero para el procedimiento de mantenimiento y puede autorizar o vetar el mismo, al tener la visibilidad de otros factores, como trabajos simultáneos, amenazas a los procesos del negocio, criticidad de los elementos involucrados. Es de aquí el primer punto de partida de donde se extraerá la información para los sistemas de aprendizaje neuro difusos colocados en la fase de análisis del ciclo de control MAPE-K del elemento autonómico.

Tabla 4. Índice de la madurez Autonómica.

	Básico Nivel 1	Administrado Nivel 2	Predictivo Nivel 3	Adaptativo Nivel 4	Autonómico Nivel 5
Características	Basada en reportes del sistema, documentación de productos y acciones manuales para configurar, optimizar, reparar y proteger los componentes individuales de TI	Software de Administración en sitio que permite la consolidación y automatización de tareas de TI	Componentes de TI y sistemas son capaces de monitorear, correlacionar y analizar el ambiente y las acciones recomendadas	Componentes de TI individuales y colectivamente son capaces de monitorear, correlacionar y analizar el ambiente y tomar acciones con mínima intervención humana	La integración de los componentes de TI se gestiona de forma colectiva y dinámica mediante las reglas y políticas del negocio.
Habilidades	Requiere mucho personal altamente calificado	El personal analiza y toma acciones	El personal aprueba e inicia las acciones	El personal gestiona el rendimiento en función del SLA	El personal se enfoca en satisfacer las necesidades empresariales
Beneficios	Requerimientos básicos son atendidos	Mayor Conocimiento del sistema. Productividad Mejorada	Se reduce la dependencia de conocimientos profundos. Decisiones realizadas de forma rápida y mejor.	Balance entre la interacción humano-máquina. Agilidad y resiliencia de TI	Las políticas empresariales impulsan la gestión de TI. El negocio es ágil y resiliente.
	Manual				Autonómico

(Manoel et al., 2005)

5. Descripción de la solución

En este capítulo se presenta la propuesta de solución para el problema de la gestión de los mantenimientos al interior de los centros de datos. Como se pudo observar en el capítulo anterior, uno de los retos a los que se enfrentó esta investigación es la pequeña cantidad de datos que se tiene para entrenar los modelos. Se eligió las redes tipo ANFIS porque de acuerdo con (Gómez Vargas et al., 2010) este tipo de redes tienen la ventaja de que se pueden generar topologías eficaces cuando se cuenta con un número de registros pequeño.

5.1 Agente autónomo Inteligente

En este proyecto investigativo se sentarán las bases de lo que se debe integrar dentro del elemento autónomo en la fase de análisis del ciclo de control MAPE-K. Esta propuesta coloca una red neuronal difusa adaptativa (ANFIS) que permita inferir si el proceso de mantenimiento (“self-healing”) de un equipo del centro de datos debe proceder o no. La figura 24 muestra el diseño de alto nivel de esta solución.

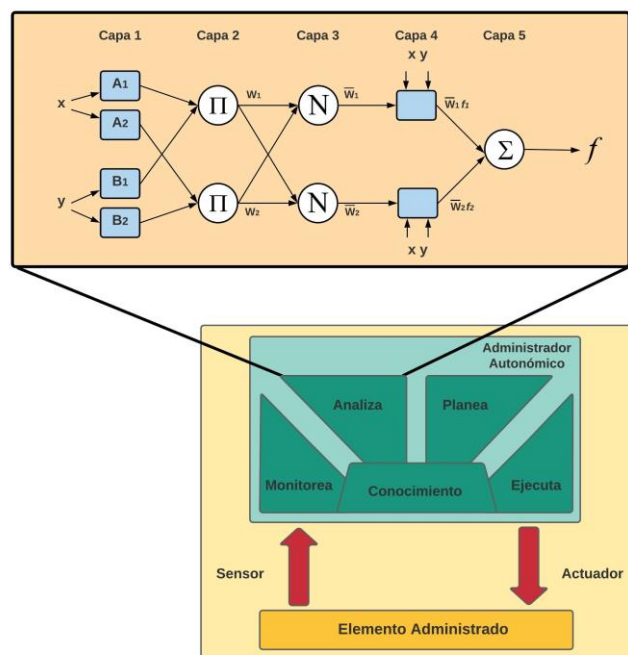


Figura 24. Elemento autónomo que incluye una red ANFIS en la fase de análisis.

Elaboración Propia

En la imagen se realiza una ampliación de lo que sería el interior de la fase de análisis de nuestro elemento autónomo propuesto. La red neuronal de tipo ANFIS recibiría los datos que utilizan los expertos para inferir si las condiciones para realizar un mantenimiento son favorables o no.

5.2 Definición de Red ANFIS

La red neuronal ANFIS al interior de la fase de análisis está definida por las siguientes entradas expuestas en la tabla 5.

Tabla 5. Entradas de la red ANFIS y sus correspondientes funciones de pertenencia.

Ingreso	Funciones de Pertenencia
Posición en el TRM	Infraestructura Plataforma Aplicaciones
Disponibilidad del Elemento	Baja Media Alta
Criticidad del Elemento	Baja Media Alta
Amenaza Externa	Baja Media Alta

Elaboración Propia

La valoración de cada una de las entradas se realizó mediante la colaboración de expertos del centro de datos. Puede llamar la atención en la estructura de ingreso a la red neuronal, el colocar los elementos identificables del TRM y que cumplen funciones específicas de almacenamiento, red y cómputo en una de las entradas difusas. La justificación de esto es porque entre más alto en el TRM se encuentren los elementos, mayor es la redundancia que estos pueden alcanzar, por lo tanto, es más seguro realizar mantenimiento sobre ellos. Además de que existen híbridos de almacenamiento-cómputo que no pueden ser encasillados en alguna categoría específica.

Al ser una red tipo ANFIS de clase Sugeno, tiene una sola salida en la cual se agrupan las funciones de pertenencia para cada una de las funciones de pertenencia de entrada. La red elaborada tiene un total de 81 funciones de pertenencia de salida para la evaluación de cada regla.

En la figura 25 se puede observar la estructura de la red propuesta mediante el uso del software Matlab de MathWorks.

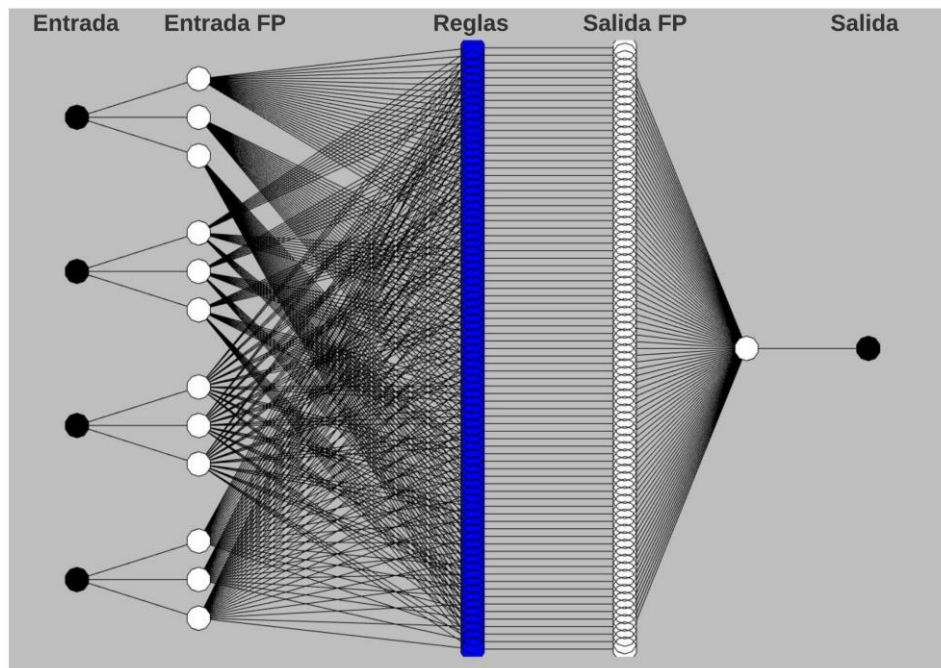


Figura 25. Estructura de la red neuronal ANFIS para la solución propuesta

Elaboración Propia

La estructura del sistema de inferencia difuso tiene 4 entradas que se mostraron en la tabla 5. Cada entrada tiene tres entradas de las funciones de pertenencia para identificar el nivel en el cual un elemento autónomo se encuentra. A continuación, las reglas que procesan esas funciones de pertenencia que lo enrutan hacia una determinada función de salida.

Cada función de pertenencia se ha delimitado de cero a uno y dependiendo del objetivo que se quiera lograr los elementos son ubicados en algún punto de esta escala para que sean clasificados por las funciones de pertenencia.

5.3 Escala de evaluación de elementos

La entrada de valores a la red neuronal está condicionada a las funciones de pertenencia mostradas en la tabla 5. De igual manera los distintos elementos, situaciones o características se encuentran evaluados dentro de una escala elaborada de acuerdo con lo observado en el análisis de campo. La figura 26 muestra cómo se realiza esa valoración tomando como ejemplo al TRM de la sección 2.2.2 y los datos más precisos en la tabla 6.

De esta misma forma las variables restantes, disponibilidad, criticidad y amenaza reciben sus correspondientes valoraciones que serán tratadas a continuación en las tablas 7, 8 y 9.

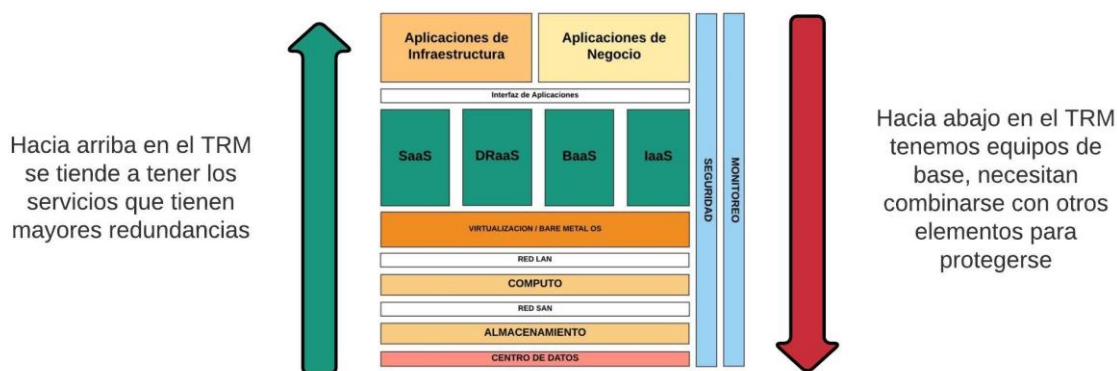


Figura 26. Valoración de los elementos de entrada de la red

Elaboración Propia

En este ejemplo se puede observar cómo funciona la escala de medición de los elementos del TRM antes de su ingreso a la red neuronal. Los equipos en la base del TRM como los de almacenamiento o cómputo tienen un valor más bajo al ingreso de la red cercano a cero, mientras los que están en la parte superior se acercan a uno.

Si el equipo se encuentra en la capa más baja del TRM se lo identifica con la valoración más baja de la escala, viceversa para los que están en el tope del TRM.

Tabla 6. Valoración aproximada de los elementos de TRM

Valor	Elemento TRM
10	Aplicaciones de negocio.
9	Aplicaciones de Infraestructura y Monitoreo.
8	Servicios de Cloud, IaaS, SaaS, DRaaS.
7	Equipos de Red y comunicaciones. Equipos de Seguridad Informática.
6	Sistemas operativos de usuario final, contenedores.
5	Sistemas operativos de Virtualización, Middleware.
4	Administración de Hardware de equipos de cómputo.
3	Componentes internos de conexión entre cómputo y almacenamiento.
2	Equipos híbridos de almacenamiento. Por ejemplo: servidores con discos
1	Equipos de Almacenamiento de datos.

Elaboración Propia

Los equipos tienen diferentes niveles de disponibilidad. La misma depende de la cantidad de protecciones que se puedan tener, fuentes de energía redundantes, múltiples puertos de conexión a red, duplicidad en sus elementos, copia de su información y configuración en línea.

Tabla 7. Valoración aproximada de la disponibilidad de los elementos

Valor	Disponibilidad
10	Elementos clusterizados duplicados.
9	Elementos clusterizados (grupo de equipos que funcionan como uno).
8	Elementos duplicados.
7	Elementos con redundancia externa.
6	Elementos con redundancia interna.
5	Elementos con redundancia reducida.
4	Elementos de crecimiento vertical con redundancia
3	Elementos de crecimiento vertical sin redundancia
2	Elementos sin soporte del fabricante
1	Elementos sin redundancia de ningún tipo.

Elaboración Propia

La criticidad de los equipos está asociada con su posición en el TRM y los servicios que ofrecen. Por ejemplo, un equipo que se encuentra en la parte más baja del TRM tendrá una criticidad más alta (equipo de almacenamiento). Por otro lado, un equipo que se encarga de entregar la hora de red a otros elementos podría tener una criticidad más baja porque es posible desviar ese tipo de servicios a un proveedor externo. La valoración de la criticidad de un componente se puede realizar mediante un análisis de riesgos de la infraestructura involucrada. La tabla 8 muestra esta primera aproximación.

Tabla 8. Valoración de la criticidad de los elementos

Valor	Criticidad
9-10	Muy Alta
7-8	Alta
5-6	Media
3-4	Baja
1-2	Despreciable

Elaboración Propia

En la entrada de la última variable de la red neuronal se ha calificado para este trabajo como “amenaza”, que podría interpretarse como una valoración del riesgo a los que estos sistemas

están expuestos. Se mantiene la etiqueta de amenaza por cuanto al hablar de un agente inteligente podríamos pensar en la posibilidad futura de que él pueda modificarse a sí mismo e inclusive llegar a otras instancias, como la de la autoconservación. Eso se verá conforme a la evolución en la madurez autonómica de la organización. Se delega el manejo del riesgo a una capa superior y sobre el cual este agente más bien está siendo alimentado por sistemas que le están informando las condiciones externas que podrían afectarle. La tabla 9 muestra la valoración de las amenazas externas identificadas y que han impedido un mantenimiento en algún momento del tiempo.

Tabla 9. Valoración de las amenazas al sistema

Valor	Amenaza Externa
10	Ataque Informático
8.3333	Manifestaciones (Caos Social)
6.6666	Epidemias (Personal que podría enfermar)
5	Días Importantes (Feriados, Fin de mes, Quincenas)
3.3333	Incompatibilidad con otros equipos dependientes
1.6666	Daños de otros componentes

Elaboración Propia

Lo interesante de este sistema es la gran flexibilidad que aporta, pudiendo tener flexibilidad para manejar los valores que puedan ingresar desde la fase de monitoreo como valores intermedios entre los mismos y que podrían significar circunstancias combinadas, como que es posible que se vaya a ejecutar un mantenimiento sobre un equipo que puede tener cierta incompatibilidad en sus componentes al aplicar los parches y que esto sea realizado en un día importante, el sistema podría ser capaz de recibir cualquier valoración dentro de este rango.

5.4 Flujo de acción del método propuesto

La aplicación del método propuesto está basada en el flujo de control de MAPE-K para la sección de análisis del elemento autonómico. Anteriormente dentro del elemento autonómico se colocó en la figura 22 una representación de alto nivel de la solución propuesta, que es incluir una red ANFIS dentro de esa parte del flujo de control. Esta red puede o no ser activada por las acciones que la fase de monitoreo envíe a la de análisis.

El flujo de acción del método propuesto se expone en la figura 27.

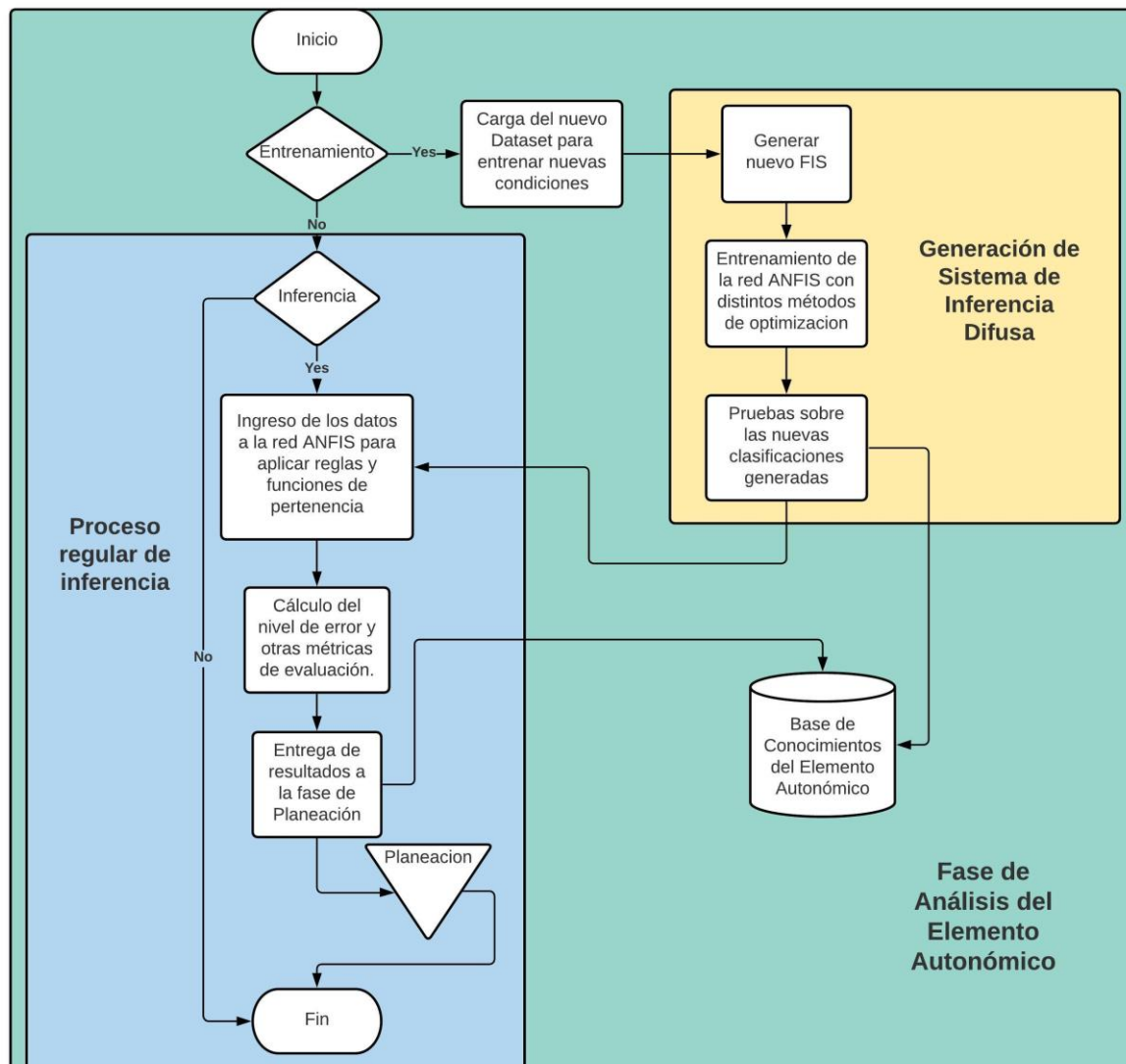


Figura 27. Diagrama de flujo del funcionamiento de alto nivel de la fase de análisis

Elaboración Propia

El diagrama de flujo presenta dos bloques principales. A la izquierda dentro del recuadro azul el proceso normal de inferencia del cual el elemento autónomo procesará las señales enviadas desde la fase de monitoreo y que serán a su vez, enviadas a la fase de planificación. A la derecha, en el recuadro amarillo, el proceso de entrenamiento y aprendizaje de la red en el caso de que amerite realizarlo nuevamente

6. Evaluación

Para evaluar el funcionamiento de lo anteriormente propuesto se tuvo la colaboración de los datos de mantenimientos realizados por el centro de datos estudiado (Datacenter Millenium) a lo largo de dos años, específicamente desde Mayo de 2018 hasta Junio de 2020. El mismo contaba con un total de 929 mantenimientos realizados en diversas categorías entre los que había mantenimientos realizados y cancelados por distintos motivos. Es aquí donde el elemento autónomo inteligente propuesto procede a reemplazar la decisión de un experto y presenta resultados tangibles a los administradores del centro de datos.

6.1 Conjunto de datos.

El conjunto de datos trabajado tiene siete columnas que se detallan a continuación en la tabla 10.

Tabla 10. Campos de la tabla de control de mantenimientos

Columna	Tipo de Dato
Tarea	Identificativo de la tarea de mantenimiento realizada dentro del sistema de control (Int)
Responsable	Nombre del Ingeniero responsable del mantenimiento (String)
Tipo de Mantenimiento	Los mantenimientos se encuentran clasificados en diferentes tipos de acuerdo con las acciones que deben realizarse. Actualización de Firmware de Equipo, Cambio de Piezas, Ajuste de Configuración. (String)
Descripción	Comentario del equipo a ser intervenido en el tipo de mantenimiento antes seleccionado. Identificación de este como Almacenamiento, Cómputo, Red o algún otro elemento del TRM. (String)
Fecha	Tiempo en el cual el mantenimiento debe ser realizado (Date)
Realizado	Variable binaria que indica si el mantenimiento fue realizado o no (1 o 0) (Int)
Cancelación	En caso de que el mantenimiento haya sido cancelado se ingresa una descripción de los motivos por los cuales no pudo ser realizado (String)

Elaboración Propia

6.1.1 Análisis de Datos

Sobre esta data origen se utilizaron las columnas de Tipo de mantenimiento, Descripción, Realizado y Cancelación para tener la base de los datos que serán ingresados en el sistema difuso. Estos datos fue necesario adecuarlos para ajustarlos a la entrada de lo que sería la red ANFIS propuesta en la sección 5.2. La figura 28 muestra la información del conjunto de datos a procesar.

```
datosMantenimientos.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 929 entries, 0 to 928
Data columns (total 5 columns):
#   Column              Non-Null Count  Dtype
---  -
0   TRM                  929 non-null    float64
1   DISPONIBILIDAD       929 non-null    float64
2   CRITICIDAD           929 non-null    float64
3   AMENAZA              929 non-null    float64
4   REALIZADO            929 non-null    int64
dtypes: float64(4), int64(1)
memory usage: 36.4 KB
```

Figura 28. Información del conjunto de Datos a procesar

Elaboración Propia

La información relevante al conjunto de datos que se utilizará para entrenar el sistema que analizará si la realización de un mantenimiento es viable o no. Contiene 5 columnas con valores normalizados de los expuestos en las tablas 6, 7, 8 y 9. La última columna "realizado" indica si el mantenimiento fue realizado o no y servirá para entrenar al sistema.

La figura 29 muestra una descripción de los datos a utilizarse:

```
datosMantenimientos.describe()
```

	TRM	DISPONIBILIDAD	CRITICIDAD	AMENAZA	REALIZADO
count	929.000000	929.000000	929.000000	929.000000	929.000000
mean	0.676276	0.923064	0.905967	0.353008	0.937567
std	0.214761	0.150466	0.139712	0.088656	0.242070
min	0.255500	0.333300	0.333300	0.333300	0.000000
25%	0.456600	0.888800	0.855500	0.333300	1.000000
50%	0.856600	0.999900	0.999900	0.333300	1.000000
75%	0.856600	0.999900	0.999900	0.333300	1.000000
max	0.999900	0.999900	0.999900	0.999700	1.000000

Figura 29. Descripción del conjunto de Datos a procesar

Elaboración Propia

El conjunto de datos contiene 929 elementos. Estos posteriormente serán divididos en conjunto de entrenamiento y test.

En la figura 30 con la ayuda de histogramas se puede visualizar de forma gráfica el comportamiento de las frecuencias de los datos por analizar.

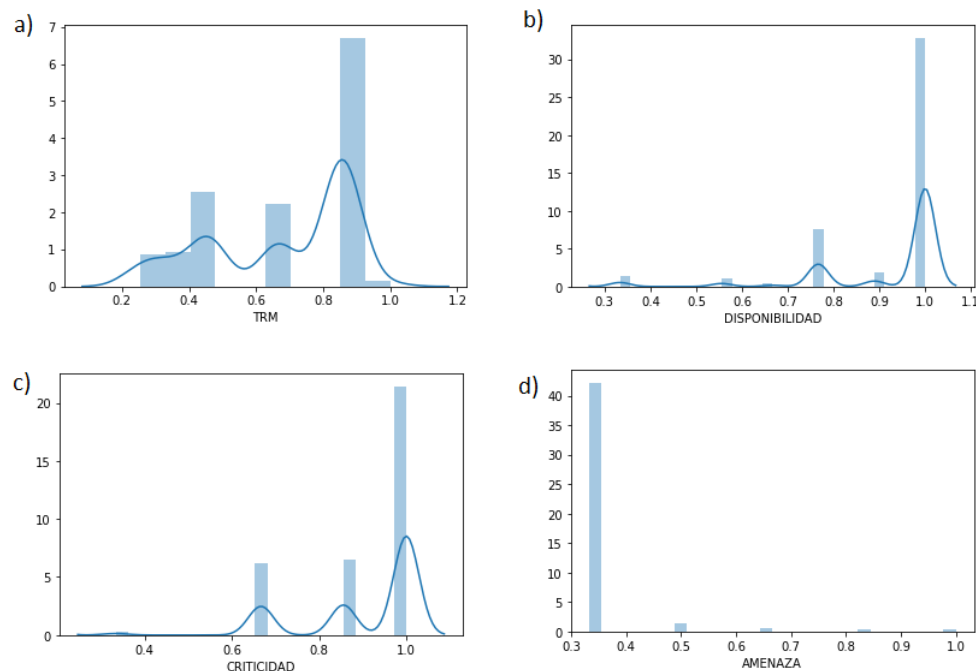


Figura 30. Histogramas de las columnas del conjunto de Datos a procesar

Elaboración Propia

- En TRM se puede observar que la mayor cantidad de elementos a dar mantenimiento se encuentran en las capas superiores (aplicaciones). El mantenimiento al hardware de Storage es menor.
- La disponibilidad de los elementos es alta. Esto se debe a que se encuentran adecuadamente protegidos con distintas capas de redundancias.
- Los servicios ejecutados en los centros de datos son críticos. Esto no escapa de la realidad al indicarse que la mayoría de los elementos en el conjunto de datos tienen una criticidad alta.
- Durante el tiempo de estudio, años 2018, 2019 y 2020, existió un número bajo de amenazas que puedan impedir los mantenimientos a realizarse. Sin embargo, si existen puntos en los cuales por diversas causas los mantenimientos se tuvieron que cancelar.

Al margen de si un elemento tiene disponibilidad o criticidad alta, en las operaciones del centro de datos los mantenimientos a los equipos deben realizarse para asegurar la calidad del servicio. Es importante conocer cuántos elementos pertenecientes al TRM se le han realizado mantenimientos exitosos, la figura 31 muestra lo expuesto.

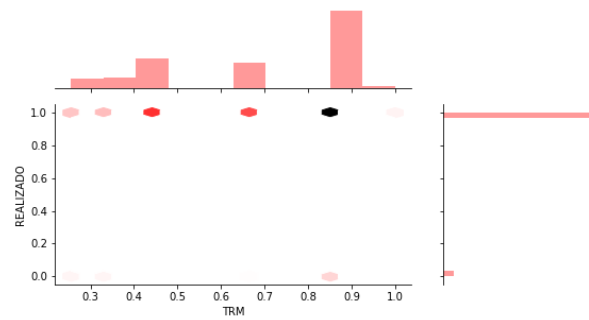


Figura 31. Grafico de Correlación entre los elementos del TRM y los mantenimientos realizados

Elaboración Propia

Como se puede observar, antes de su la mayor parte de elementos de centro de datos tienen cumplidos sus mantenimientos (histograma y valor cercano a 1). Mientras aquellos a los que no se les ha realizado son muy pocos.

Finalmente, en la figura 32.

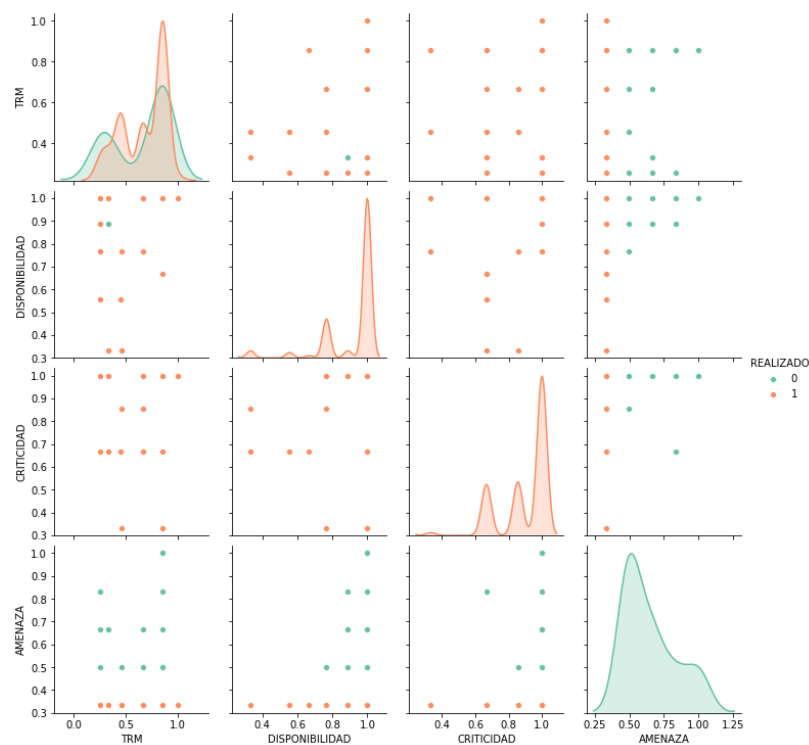


Figura 32. Correlación de las distintas variables agrupados por si han sido realizados o no

Elaboración Propia

Extensión de la figura 31 ahora en su grafico de dispersión e histogramas para el resto de las columnas del conjunto de datos.

6.2 Creación del Sistema de Inferencia Difusa.

Luego del análisis de los datos a ingresarse, de acuerdo con el flujo de la figura 25, corresponde la creación del Sistema de Inferencia Difuso FIS (Fuzzy Inference System) y el entrenamiento de los datos. Para el efecto se ha hecho uso del Software Matlab, mismo que tiene una amplia colección de herramientas para la aplicación de técnicas de lógica difusa. Dentro del software de Matlab se dispone de un asistente gráfico que permite la creación del sistema difuso, mediante el comando “neuroFuzzyDesigner”. La figura 33 muestra cómo se visualiza el mismo.

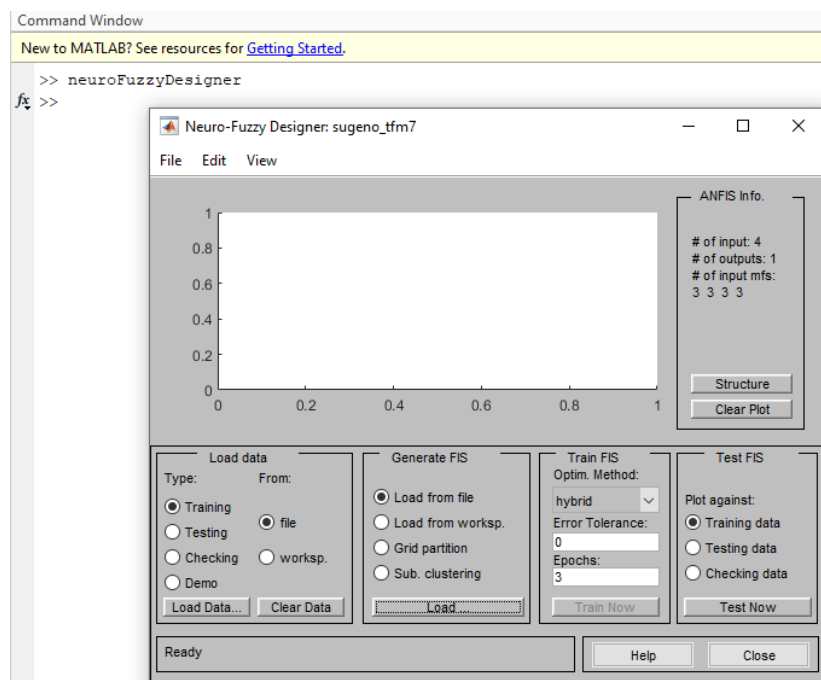


Figura 33. Asistente para la creación de Sistemas de Inferencia Difusos de Matlab

Elaboración Propia

Para la creación del sistema FIS base se carga la data de entrenamiento y mediante la utilización del recuadro “Generate FIS” se selecciona la opción “Grid Partition” y luego “Generate”. La figura 34 muestra cómo se visualiza la opción indicada.

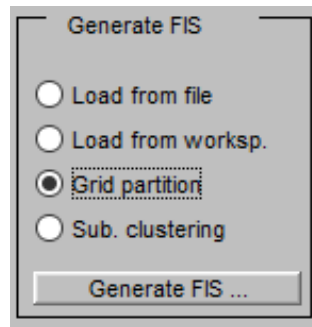


Figura 34. Opción de generación de sistemas FIS en Matlab

Elaboración Propia

La salida de este resultado se puede apreciar en la figura 22.

6.2 Entrenamiento y pruebas de la Red ANFIS.

Para el entrenamiento de la red ANFIS primero se debe realizar la carga del FIS anteriormente creado. La figura 35 muestra este proceso.

```
>> fis = readfis('sugeno_tfm9')

fis =

    sugfis with properties:

        Name: "sugeno_tfm9"
        AndMethod: "prod"
        OrMethod: "probor"
        ImplicationMethod: "prod"
        AggregationMethod: "sum"
        DefuzzificationMethod: "wtaver"
        Inputs: [1x4 fisvar]
        Outputs: [1x1 fisvar]
        Rules: [1x81 fisrule]
        DisableStructuralChecks: 0

    See 'getTunableSettings' method for parameter optimization.
```

Figura 35. Carga del FIS previamente creado al workspace de Matlab

Elaboración Propia

Se puede observar que, en el momento de la carga, Matlab va indicando ciertos parámetros del sistema FIS cargado como el nombre, su método de aclaramiento y el número de reglas.

En Matlab el entrenamiento de la red ANFIS se lleva a cabo de la siguiente forma conforme lo observado en la figura 36.

```

ANFIS info:
  Number of nodes: 193
  Number of linear parameters: 81
  Number of nonlinear parameters: 24
  Total number of parameters: 105
  Number of training data pairs: 309
  Number of checking data pairs: 620
  Number of fuzzy rules: 81

Start training ANFIS ...

  1    0.00639606    0.0393572
  2    0.00532296    0.0368396
  3    0.0043081     0.034461
  4    0.00337409    0.0322499
  5    0.00254186    0.0302347
Step size increases to 0.011000 after epoch 5.
  6    0.00182812    0.028444
  7    0.00119289    0.0267695
  8    0.000806862   0.0257511
  9    0.00123623    0.0270551
 10    0.00076275    0.0257289

Designated epoch number reached --> ANFIS training completed at epoch 10.

Minimal training RMSE = 0.000763
Minimal checking RMSE = 0.0257289

```

Figura 36. Entrenamiento de la red ANFIS

Elaboración Propia

La red ANFIS se entrena en el sistema FIS con los datos de entrenamiento y test cargados. El RMSE obtenido en esta etapa es de 2.57×10^{-2} , por lo cual se puede concluir que los valores de entrenamiento y test son bastante cercanos.

De los datos recolectados se separaron dos conjuntos de datos, uno para entrenamiento y otro para validación. A continuación, se carga el conjunto de datos de validación en una variable para que posteriormente sea evaluada.

```
>> x = datosChk(:,1:4)

x =

    0.2566    0.8888    0.9999    0.3333
    0.4566    0.3333    0.8555    0.3333
    0.8566    0.9999    0.9999    0.3333
    0.6666    0.9999    0.9999    0.3333
    0.8566    0.9999    0.9999    0.3333
    0.8566    0.9999    0.9999    0.3333
    0.4566    0.3333    0.8555    0.3333
    0.8566    0.9999    0.9999    0.3333
```

Figura 37. Carga del conjunto de datos de validación en una variable

Elaboración Propia

Esta imagen muestra los primeros registros del conjunto de datos. Como se puede apreciar solo consta de cuatro columnas sin la columna de si fue realizado o no. Este es el valor que ahora la red ANFIS procederá a predecir.

La validación del modelo se realiza evaluando el sistema FIS contra los datos de validación. El mismo comando envía ya los valores de evaluación como predicciones de la red Neuronal. La figura 38 muestra un extracto de este particular

```
>> anfisOutput = evalfis (fis, x)

anfisOutput =

    1.0001
    1.0000
    1.0000
    0.9995
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
```

Figura 38. Resultado de la evaluación de la red ANFIS

Elaboración Propia

Los valores inferidos mediante este sistema se proceden a cargar al conjunto de datos para la respectiva validación.

Los valores de prueba provienen del mismo conjunto de datos original pero previamente particionado al 30%. Sobre este conjunto de datos ya se tenía los resultados iniciales de los mantenimientos realizables o no realizables, por lo que es posible comparar el funcionamiento de la red neuronal.

La figura 39 muestra un extracto de los resultados obtenidos en la primera prueba y su comparación con los datos originales.

TRM	DISPONIBILIDAD	CRITICIDAD	AMENAZA	REALIZABLE	PREDICHO ANFIS
0,2566	0,8888	0,9999	0,3333	1	1,0001
0,6666	0,9999	0,9999	0,3333	1	0,9995
0,2566	0,8888	0,9999	0,3333	1	1,0001
0,3333	0,9999	0,9999	0,3333	1	0,9999
0,6666	0,9999	0,9999	0,3333	1	0,9995
0,3333	0,9999	0,9999	0,3333	1	0,9999
0,6666	0,9999	0,9999	0,3333	1	0,9995
0,8566	0,9999	0,9999	0,9997	0	0,1281
0,3333	0,9999	0,9999	0,3333	1	0,9999
0,8566	0,9999	0,9999	0,9997	0	0,0005
0,3333	0,9999	0,9999	0,3333	1	0,9999
0,6666	0,9999	0,9999	0,3333	1	0,9995
0,3333	0,9999	0,9999	0,3333	1	0,9999
0,3333	0,9999	0,9999	0,3333	1	0,9999
0,2566	0,8888	0,9999	0,3333	1	1,0001
0,4555	0,5555	0,6666	0,3333	1	0,9999
0,6666	0,9999	0,9999	0,3333	1	0,9995
0,8566	0,9999	0,9999	0,9997	0	0,0005
0,2566	0,8888	0,9999	0,3333	1	1,0001

Figura 39.Comparativa de los resultados obtenidos contra los originales

Elaboración Propia

Los campos resaltados en amarillo son los trabajos que tuvieron que cancelarse por algún motivo. Estos trabajos cancelados tienen un nivel de amenaza superior en relación con otros trabajos. La evaluación mediante la red ANFIS predice adecuadamente cuando un trabajo debe ser cancelado o no.

La siguiente prueba que se realizó con la red ANFIS es modificar los datos de amenaza. Esto es, mantener los datos de TRM, Disponibilidad, Criticidad, y solamente modificar de forma aleatoria ciertos valores de amenaza para saber si la red neuronal puede procesar adecuadamente valores nuevos sobre los cuales no tiene información previa.

La figura 40 muestra el resultado de esta modificación.

AMENAZA ORIGINAL	AMENAZA MODIFICADA	REALIZABLE	PREDICHO ANFIS	PREDICHO AM
0,3333	0,3333	1	1,0001	1,0001
0,3333	0,3333	1	0,9995	0,9995
0,3333	0,3333	1	1,0001	1,0001
0,3333	0,3333	1	0,9999	0,9999
0,3333	0,3333	1	0,9995	0,9995
0,3333	0,3333	1	0,9999	0,9999
0,3333	0,3333	1	0,9995	0,9995
0,9997	0,0957	0	0,1281	3,6229
0,3333	0,3333	1	0,9999	0,9999
0,9997	0,9997	0	0,0005	0,0005
0,3333	0,1658	1	0,9999	1,7473
0,3333	0,9778	1	0,9995	-0,1616
0,3333	0,3333	1	0,9999	0,9999
0,3333	0,3333	1	0,9999	0,9999
0,3333	0,3333	1	1,0001	1,0001
0,3333	0,3333	1	0,9999	0,9999
0,3333	0,3333	1	0,9995	0,9995
0,9997	0,9997	0	0,0005	0,0005
0,3333	0,3333	1	1,0001	1,0001

Figura 40. Validación del funcionamiento de la red con datos nuevos

Elaboración Propia

Este cuadro valida que el sistema propuesto es efectivo con referencia a lo que se desea lograr. En el caso de que la amenaza original es modificada con valores menores, la red ahora cataloga el trabajo como "realizable" (valores superiores a uno). Mientras que una amenaza menor que fue modificada con un valor mayor ahora marca un mantenimiento como no realizable, puesto que considera que las condiciones son adversas y es más seguro cancelarlo.

Finalmente, la figura 41 nos muestra una comparativa del error del modelo entrenado con los datos originales y los datos modificados.

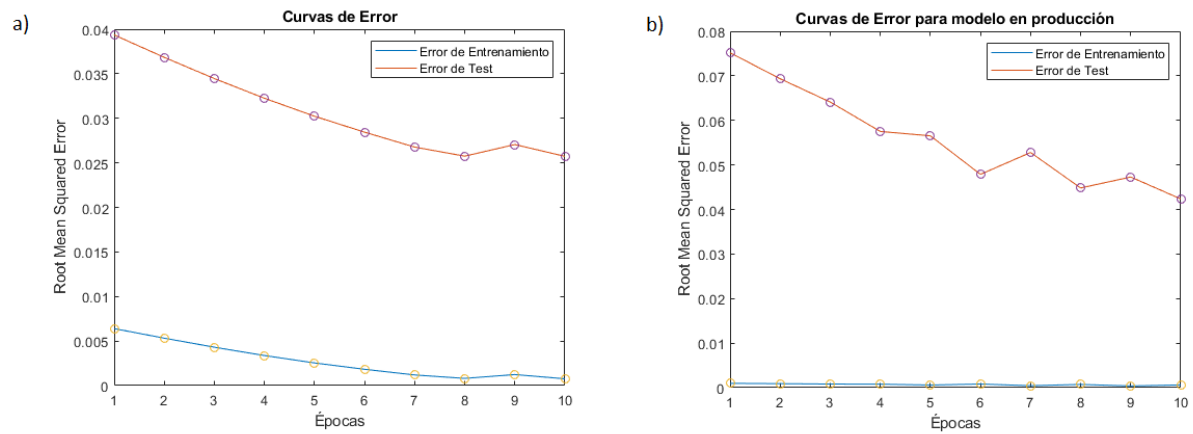


Figura 41. Validación del funcionamiento de la red con datos nuevos

Elaboración Propia

La comparativa de las curvas de error nos permiten observar que el modelo puede generalizar el ingreso de nueva información. a) En esta imagen con los datos originales se puede ver la curva de error de prueba llegar hasta el valor de 2.25×10^{-2} . b) En esta imagen se ve el comportamiento del modelo al clasificar nuevos datos, su margen de error es de 4.23×10^{-2}

7. Conclusiones y trabajo futuro

Llegado a este punto se presentan las conclusiones y trabajo futuro de esta investigación. Se detallará lo alcanzado con respecto a cada uno de los objetivos planteados y se dejará trazado desde la línea de esta investigación el camino que se debe continuar.

7.1. Conclusiones

La solución propuesta en esta investigación utiliza estrategias de inteligencia artificial que permiten llevar a las organizaciones hacia una madurez autonómica.

La utilización de sistemas como este que combinan los conceptos fundacionales de la computación autonómica (Hassan et al., 2006) y los de lógica difusa por el lado de la inteligencia artificial (Liu & Li, 2004), abre las puertas a nuevas condiciones que permitan a los expertos poderse enfocar en otros aspectos del negocio, disminuyendo la complejidad de los activos de tecnología y aumentando la eficiencia.

Los datos analizados sobre los mantenimientos realizados en este trabajo investigativo demuestran que actualmente se sigue delegando la decisión final de realizar un mantenimiento a las manos de expertos que puedan determinar el momento adecuado para llevarlos a cabo.

La metodología propuesta en este documento se basa en la utilización de herramientas como el TRM de TOGAF, Computación Autonómica, Lógica Difusa, Redes ANFIS en el alto nivel y Matlab, Python y Jupyter Notebook en el bajo nivel para procesar los datos. La flexibilidad del sistema permite la modificación de ingreso de parámetros para la inclusión o retiro de estos, de acuerdo con las condiciones en las que se necesite ejecutar el sistema propuesto.

La red ANFIS implementada consta de 4 entradas, las cuales se forjaron desde los datos origen de las 929 tareas de mantenimiento analizadas. Estas entradas denotan la posición del elemento en el TRM, disponibilidad, criticidad y amenaza que se utilizaron para inferir si el mantenimiento es realizable o no, al atravesar las 3 funciones de pertenencia de cada entrada y las 81 reglas establecidas en el sistema FIS. Además, se utilizó el método de cálculo de gradiente “*backpropagation*” y al final una capa de salida que retorna el valor “*defuzzyficado*” entre 0 y 1.

En la evaluación de los resultados, la red ANFIS entrenada demostró que pudo clasificar nuevos datos de mantenimientos de forma correcta, siendo el valor de amenaza el de mayor

variabilidad y peso para la toma de decisiones. El resultado del RMSE de la red original con respecto a la red que clasificó los datos desconocidos al entrenamiento inicial tiene una diferencia de 1.83×10^{-2} .

Por lo tanto, la contribución de este trabajo en ayudar a los expertos al eliminar los procesos manuales y facilitando la planeación de mantenimientos de manera automatizada se considera cumplida. Además, este pasa a ser un método muy eficiente para ser utilizado por los demás miembros de la organización que no cuenten con la experiencia suficiente, pasarían a confiar en el sistema y el mismo les proporcionaría los resultados adecuados, puesto que ya cuenta con información de experiencia de expertos en su base de conocimientos. En resumen, se puede trabajar más rápido, se evitan los errores humanos y se consolida el conocimiento y la experiencia en un solo sistema.

7.2. Líneas de trabajo futuro

De cara a la implementación de elementos autónomos dentro de los centros de datos es importante continuar la investigación en este ámbito de la inteligencia artificial sumando más experiencia desde centros de datos alrededor del mundo que permitan analizar la estrategia utilizada actualmente en sus operaciones.

La metodología propuesta en este estudio tiene limitantes, como el aspecto de la obtención de datos. Estos deberían ser tratados como investigaciones posteriores en el que se estudie cada fase del bucle de control MAPE-K indicado en la figura 12 (Lalanda et al., 2013b). Otros trabajos deben enfocarse en la recopilación de datos automatizada desde el monitoreo y que es enviada a la fase de análisis propuesta en este trabajo.

De igual forma los valores que se obtienen en la fase de análisis deben ser procesados en la fase de planeación y posterior ejecución, del elemento autónomo para de esta forma completar el ciclo del bucle de control indicado. Automatizando así, todo el proceso del elemento autónomo desarrollado para la automatización de la auto reparación (*“self-healing”*) en el centro de datos.

Como este es un proceso que exige a las organizaciones ir ganando madurez tecnológica, esperamos que la inclusión de este tipo de tecnologías sea realizada de manera controlada, conforme se pueda seguir recolectando más información y el sistema aquí propuesto pueda ir aprendiendo nuevas condiciones que permitan en el mediano o largo plazo delegar enteramente estas funciones desde los operadores del centro de datos al sistema experto.

8. Bibliografía

- Aragort, M. (2009). *El sistema Nervioso Central*. El Cid Editor. <https://bv.unir.net:3555/es/ereader/unir/28486?page=6>.
- Arman Shehabi, Sarah Josephine Smith, Dale A Sartor, Richard E Brown, Magnus Herrlin, Jonathan G Koomey, Eric R Masanet, Nathaniel Horner, Inês Lima Azevedo, & William Lintner. (2016). *United States Data Center Energy Usage Report | Energy Technologies Area*. June, 65. <https://eta.lbl.gov/publications/united-states-data-center-energy>
- Brittenham, P., Cutlip, R. R., Draper, C., Miller, B. A., Choudhary, S., & Perazolo, M. (2007). IT service management architecture and autonomic computing. *IBM Systems Journal*, 46(3), 565–581. <https://doi.org/10.1147/sj.463.0565>
- Dobrev, D. (2012). *A Definition of Artificial Intelligence*. 1–7. <http://arxiv.org/abs/1210.1568>
- Eilam, T., Kalantar, M. H., Konstantinou, A. V., Pacifici, G., Pershing, J. A., & Agrawal, A. (2006). Managing the configuration complexity of distributed applications in internet data centers. *IEEE Communications Magazine*, 44(3), 166–177. <https://doi.org/10.1109/MCOM.2006.1607881>
- Fadaeefath Abadi, M., Haghighat, F., & Nasiri, F. (2020). Data center maintenance: applications and future research directions. *Facilities*. <https://doi.org/10.1108/F-09-2019-0104>
- Gartner Inc. (2019). *Informe de infraestructura como servicio en la nube de Gartner*. <https://cloud.google.com/gartner-cloud-infrastructure-as-a-service/thank-you?hl=es>
- Georgeff, M., Pell, B., Pollack, M., Tambe, M., & Wooldridge, M. (2014). *The Belief-Desire-Intention Model of Agency The Belief-Desire-Intention Model of Agency*. February 1970. <https://doi.org/10.1007/3-540-49057-4>
- Gómez Vargas, E., Obregón Neira, N., & Socarras Quintero, V. (2010). Application of neuro-fuzzy ANFIS model vs neural network, to the predictive monthly mean flow problem in the Bogotá river in Villapinzón. *Tecnura*, 14(27), 18–29.
- Hassan, S., McSherry, D., & Bustard, D. (2006). Autonomic self healing and recovery informed by environment knowledge. *Artificial Intelligence Review*, 26(1–2), 89–101. <https://doi.org/10.1007/s10462-007-9033-6>

IBM. (2005). *An architectural blueprint for autonomic computing*.

Ibrahim Omer, S. M., A.Mustafa, A. B., & Elmahdi Alghali, F. A. (2014). Advantages of Autonomic Computing over Cloud Computing Comparative Analysis. *IOSR Journal of Electrical and Electronics Engineering*, 9(6), 56–60. <https://doi.org/10.9790/1676-09635660>

Jang, J. S. R. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3), 665–685. <https://doi.org/10.1109/21.256541>

Javier, C., Germ, T., Una, R., & Neuronal, R. (2009). Redes Neuronales Artificiales Introducción Modelo neuronal de McCulloch-Pitts. *REVISTA DE EDUCACIÓN MATEMÁTICA*, 24(3), 22–30.

Jeffrey, O., & David, M. (2003). The Vision of Autonomic Computing. *Computer*, 18(January), 41–50. <https://doi.org/10.1046/j.1365-2745.2002.00730.x>

Jirovsky, P., & Mehra, R. (2015). *The New Need for Speed in the Datacenter Network The 3rd Platform creates the underpinnings for business process business model transformation : □ How enterprises engage with Inversion of business from make-sell to sense-and-respond*.

Karadurmuş, E., Taşkın, N., Göz, E., & Yüceer, M. (2019). Prediction of Bromate Removal in Drinking Water Using Artificial Neural Networks. *Ozone: Science and Engineering*, 41(2), 118–127. <https://doi.org/10.1080/01919512.2018.1510763>

Khalifa, A. B., & Frigui, H. (2015). Multiple Instance Mamdani Fuzzy Inference. *The International Journal of Fuzzy Logic and Intelligent Systems*, 15(4), 217–231. <https://doi.org/10.5391/ijfis.2015.15.4.217>

Kraus, R. (2020). *What is Machine Learning?* <https://www.redhat.com/ja/blog/what-machine-learning>

Kruse, R. (2008). Fuzzy neural network. *Scholarpedia*, 3(11), 6043. <https://doi.org/10.4249/scholarpedia.6043>

Lalanda, P., McCann, J. a., & Diaconescu, A. (2013a). Autonomic Computing. In *Technology* (Issue April). Springer London. <https://doi.org/10.1007/978-1-4471-5007-7>

Lalanda, P., McCann, J. a., & Diaconescu, A. (2013b). Autonomic Computing. In *Technology*

- (Issue April). Springer London. <https://doi.org/10.1007/978-1-4471-5007-7>
- Liu, P., & Li, H. (2004). *Fuzzy Neural Network Theory and Application* (Vol. 59). World Scientific.
- Magradze, E. (2015). *Monitoring and Optimization of ATLAS Tier 2 Center GoeGrid Dissertation*.
- Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1), 1–13. [https://doi.org/10.1016/S0020-7373\(75\)80002-2](https://doi.org/10.1016/S0020-7373(75)80002-2)
- Manoel, E., Nielson, M., Salahshour, a, Kvl, S., & Sudarshanan, S. (2005). Problem determination using self-managing autonomic technology. *Contract*. <http://www.redbooks.ibm.com/redbooks/pdfs/sg246665.pdf>
- Mathworks. (2020). *Fuzzy Inference Process - MATLAB & Simulink - MathWorks América Latina*. <https://la.mathworks.com/help/fuzzy/fuzzy-inference-process.html>
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*. <https://doi.org/10.1007/BF02478259>
- Moreno, C. M., Vicente, E. S., & Martínez, C. E. (1998). Revision historica del concerto de inteligencia: Una aproximacion a la inteligencia emocional. *Revista Latinoamericana de Psicologia*, 30(1), 11–30.
- Mousa Al-Zawi, M., Al-Jumeily, D., Hussain, A., & Taleb-Bendiab, A. (2011). Autonomic computing: Applications of self-healing systems. *Proceedings - 4th International Conference on Developments in ESystems Engineering, DeSE 2011*, 381–386. <https://doi.org/10.1109/DeSE.2011.22>
- Mousavi, -Seyed Muhammad Hossein, Mirinezhad, -S. Younes, & Lyashenko, -Vyacheslav. (2017). An Evolutionary-Based Adaptive Neuro-Fuzzy Expert System as a Family Counselor before Marriage with the Aim of Divorce Rate Reduction. *Conference: 2nd International Conference on Research Knowledge Base in Computer Engineering and IT, August*, 1–10. www.C-IT.ir
- NASDAQ OMX's. (2020). OPSANI INTRODUCES FREE CLOUD COST REDUCTION SERVICES FOR CLOUD APPS: Project "Vital" Helps Companies Lower Cloud Bills By Up to 70 Percent at No Cost for Three Months. *News Release Distribution Channel*.

- Naseri, H., Shahidinejad, A., & Ghobaei-Arani, M. (2019). An intelligent autonomous system for condition-based maintenance-Case study: Control valves. *Journal of Information Systems and Telecommunication*, 7(2), 144–154.
- Nauck, D., & Kruse, R. (1995). NEFCLASS - a neuro-fuzzy approach for the classification of data. *Proceedings of the ACM Symposium on Applied Computing, September 2013*, 461–465. <https://doi.org/10.1145/315891.316068>
- Obieniu, A. C. (2018). *Autonomic Computing : An Evidence of Better Management Technology. February*, 941–954.
- Peijiang, C., & Yongjun, M. (2013). Research and Implementation of Vehicle Fault Diagnosis Expert System. *INTERNATIONAL JOURNAL ON Advances in Information Sciences and Service Sciences*, 5(7), 28–36. <https://doi.org/10.4156/aiss.vol5.issue7.4>
- Sterritt, R. (2005). Autonomic computing. *Innovations in Systems and Software Engineering*, 1(1), 79–88. <https://doi.org/10.1007/s11334-005-0001-5>
- The Open Group. (2018). *TOGAF Standar Version 9.2*. 1–532.
- US. Securities and Exchange Commision. (2018). *Amazon EX-99.1*. EDGAR. <https://www.sec.gov/Archives/edgar/data/0001018724/000119312518121161/d456916dex991.htm>
- Velayos, J. L., & Diéguez, G. (2015). *Anatomía y Fisiología del sistema nervioso central*. CEU Ediciones.
- Villamizar, G., & Donoso, R. (2013). Revisión Histórica Definitions and Theories About Intelligence . *Psicogente*, 16, 407–423.
- Wiboonrat, M. (2020). Human Factors Psychology of Data Center Operations and Maintenance. *2020 6th IEEE International Conference on Information Management, ICIM 2020*, 167–171. <https://doi.org/10.1109/ICIM49319.2020.244692>
- Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*. [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)
- Zadeh, Lotfi A. (1975). *Fuzzy logic and approximate reasoning*. <https://doi.org/10.1007/bf00485052>

Anexos

Anexo I. Sistema FIS utilizado en Matlab

```
[System]
Name='sugeno_tfm9'
Type='sugeno'
Version=2.0
NumInputs=4
NumOutputs=1
NumRules=81
AndMethod='prod'
OrMethod='probor'
ImpMethod='prod'
AggMethod='sum'
DefuzzMethod='wtaver'

[Input1]
Name='trm'
Range=[0 1]
NumMFs=3
MF1='infraestructura':'gaussmf',[0.212330450072005 0]
MF2='plataforma':'gaussmf',[0.2123 0.5]
MF3='aplicaciones':'gaussmf',[0.212330450072005 1]

[Input2]
Name='disponibilidad'
Range=[0 1]
NumMFs=3
MF1='dis_baja':'gaussmf',[0.212330450072005 0]
MF2='dis_media':'gaussmf',[0.2123 0.5]
MF3='dis_alta':'gaussmf',[0.212330450072005 1]

[Input3]
Name='criticidad'
Range=[0 1]
NumMFs=3
MF1='cri_baja':'gaussmf',[0.212330450072005 0]
MF2='cri_media':'gaussmf',[0.212330450072005 0.5]
MF3='cri_alta':'gaussmf',[0.212330450072005 1]

[Input4]
Name='amenaza'
Range=[0 1]
NumMFs=3
MF1='ame_baja':'gaussmf',[0.212330450072005 0]
MF2='ame_media':'gaussmf',[0.212330450072005 0.5]
MF3='ame_alta':'gaussmf',[0.212330450072005 1]

[Output1]
Name='realizado'
Range=[0 1]
NumMFs=81
MF1='out1mf1':'constant',[0]
MF2='out1mf2':'constant',[0]
MF3='out1mf3':'constant',[0]
```

```
MF4='out1mf4': 'constant', [0]
MF5='out1mf5': 'constant', [0]
MF6='out1mf6': 'constant', [0]
MF7='out1mf7': 'constant', [0]
MF8='out1mf8': 'constant', [0]
MF9='out1mf9': 'constant', [0]
MF10='out1mf10': 'constant', [0]
MF11='out1mf11': 'constant', [0]
MF12='out1mf12': 'constant', [0]
MF13='out1mf13': 'constant', [0]
MF14='out1mf14': 'constant', [0]
MF15='out1mf15': 'constant', [0]
MF16='out1mf16': 'constant', [0]
MF17='out1mf17': 'constant', [0]
MF18='out1mf18': 'constant', [0]
MF19='out1mf19': 'constant', [0]
MF20='out1mf20': 'constant', [0]
MF21='out1mf21': 'constant', [0]
MF22='out1mf22': 'constant', [0]
MF23='out1mf23': 'constant', [0]
MF24='out1mf24': 'constant', [0]
MF25='out1mf25': 'constant', [0]
MF26='out1mf26': 'constant', [0]
MF27='out1mf27': 'constant', [0]
MF28='out1mf28': 'constant', [0]
MF29='out1mf29': 'constant', [0]
MF30='out1mf30': 'constant', [0]
MF31='out1mf31': 'constant', [0]
MF32='out1mf32': 'constant', [0]
MF33='out1mf33': 'constant', [0]
MF34='out1mf34': 'constant', [0]
MF35='out1mf35': 'constant', [0]
MF36='out1mf36': 'constant', [0]
MF37='out1mf37': 'constant', [0]
MF38='out1mf38': 'constant', [0]
MF39='out1mf39': 'constant', [0]
MF40='out1mf40': 'constant', [0]
MF41='out1mf41': 'constant', [0]
MF42='out1mf42': 'constant', [0]
MF43='out1mf43': 'constant', [0]
MF44='out1mf44': 'constant', [0]
MF45='out1mf45': 'constant', [0]
MF46='out1mf46': 'constant', [0]
MF47='out1mf47': 'constant', [0]
MF48='out1mf48': 'constant', [0]
MF49='out1mf49': 'constant', [0]
MF50='out1mf50': 'constant', [0]
MF51='out1mf51': 'constant', [0]
MF52='out1mf52': 'constant', [0]
MF53='out1mf53': 'constant', [0]
MF54='out1mf54': 'constant', [0]
MF55='out1mf55': 'constant', [0]
MF56='out1mf56': 'constant', [0]
MF57='out1mf57': 'constant', [0]
MF58='out1mf58': 'constant', [0]
MF59='out1mf59': 'constant', [0]
MF60='out1mf60': 'constant', [0]
MF61='out1mf61': 'constant', [0]
MF62='out1mf62': 'constant', [0]
MF63='out1mf63': 'constant', [0]
MF64='out1mf64': 'constant', [0]
```

```

MF65='out1mf65': 'constant', [0]
MF66='out1mf66': 'constant', [0]
MF67='out1mf67': 'constant', [0]
MF68='out1mf68': 'constant', [0]
MF69='out1mf69': 'constant', [0]
MF70='out1mf70': 'constant', [0]
MF71='out1mf71': 'constant', [0]
MF72='out1mf72': 'constant', [0]
MF73='out1mf73': 'constant', [0]
MF74='out1mf74': 'constant', [0]
MF75='out1mf75': 'constant', [0]
MF76='out1mf76': 'constant', [0]
MF77='out1mf77': 'constant', [0]
MF78='out1mf78': 'constant', [0]
MF79='out1mf79': 'constant', [0]
MF80='out1mf80': 'constant', [0]
MF81='out1mf81': 'constant', [0]

```

[Rules]

```

1 1 1 1, 1 (1) : 1
1 1 1 2, 2 (1) : 1
1 1 1 3, 3 (1) : 1
1 1 2 1, 4 (1) : 1
1 1 2 2, 5 (1) : 1
1 1 2 3, 6 (1) : 1
1 1 3 1, 7 (1) : 1
1 1 3 2, 8 (1) : 1
1 1 3 3, 9 (1) : 1
1 2 1 1, 10 (1) : 1
1 2 1 2, 11 (1) : 1
1 2 1 3, 12 (1) : 1
1 2 2 1, 13 (1) : 1
1 2 2 2, 14 (1) : 1
1 2 2 3, 15 (1) : 1
1 2 3 1, 16 (1) : 1
1 2 3 2, 17 (1) : 1
1 2 3 3, 18 (1) : 1
1 3 1 1, 19 (1) : 1
1 3 1 2, 20 (1) : 1
1 3 1 3, 21 (1) : 1
1 3 2 1, 22 (1) : 1
1 3 2 2, 23 (1) : 1
1 3 2 3, 24 (1) : 1
1 3 3 1, 25 (1) : 1
1 3 3 2, 26 (1) : 1
1 3 3 3, 27 (1) : 1
2 1 1 1, 28 (1) : 1
2 1 1 2, 29 (1) : 1
2 1 1 3, 30 (1) : 1
2 1 2 1, 31 (1) : 1
2 1 2 2, 32 (1) : 1
2 1 2 3, 33 (1) : 1
2 1 3 1, 34 (1) : 1
2 1 3 2, 35 (1) : 1
2 1 3 3, 36 (1) : 1
2 2 1 1, 37 (1) : 1
2 2 1 2, 38 (1) : 1
2 2 1 3, 39 (1) : 1
2 2 2 1, 40 (1) : 1
2 2 2 2, 41 (1) : 1
2 2 2 3, 42 (1) : 1

```

```

2 2 3 1, 43 (1) : 1
2 2 3 2, 44 (1) : 1
2 2 3 3, 45 (1) : 1
2 3 1 1, 46 (1) : 1
2 3 1 2, 47 (1) : 1
2 3 1 3, 48 (1) : 1
2 3 2 1, 49 (1) : 1
2 3 2 2, 50 (1) : 1
2 3 2 3, 51 (1) : 1
2 3 3 1, 52 (1) : 1
2 3 3 2, 53 (1) : 1
2 3 3 3, 54 (1) : 1
3 1 1 1, 55 (1) : 1
3 1 1 2, 56 (1) : 1
3 1 1 3, 57 (1) : 1
3 1 2 1, 58 (1) : 1
3 1 2 2, 59 (1) : 1
3 1 2 3, 60 (1) : 1
3 1 3 1, 61 (1) : 1
3 1 3 2, 62 (1) : 1
3 1 3 3, 63 (1) : 1
3 2 1 1, 64 (1) : 1
3 2 1 2, 65 (1) : 1
3 2 1 3, 66 (1) : 1
3 2 2 1, 67 (1) : 1
3 2 2 2, 68 (1) : 1
3 2 2 3, 69 (1) : 1
3 2 3 1, 70 (1) : 1
3 2 3 2, 71 (1) : 1
3 2 3 3, 72 (1) : 1
3 3 1 1, 73 (1) : 1
3 3 1 2, 74 (1) : 1
3 3 1 3, 75 (1) : 1
3 3 2 1, 76 (1) : 1
3 3 2 2, 77 (1) : 1
3 3 2 3, 78 (1) : 1
3 3 3 1, 79 (1) : 1
3 3 3 2, 80 (1) : 1
3 3 3 3, 81 (1) : 1

```

Anexo II. Código de ANFIS en MatLab

```

% Carga del sistema de inferencia difusa
fisFinal = readfis('sugeno_tfm_final');

```

```

%Grafica las Funciones de Pertenencia
figure
subplot(2,2,1)
plotmf(fisFinal,'input',1)
subplot(2,2,2)
plotmf(fisFinal,'input',2)
subplot(2,2,3)
plotmf(fisFinal,'input',3)
subplot(2,2,4)

```

```
plotmf(fisFinal,'input',4)
```

```
% Parametros de entrenamiento red Neuronal
epocasMax=50;
errorObj=0;
stepInicial=0.01;
stepDecremento=0.9;
stepIncremento=1.1;
opcionesEntrenamiento=[epocasMax errorObj stepInicial stepDecremento
stepIncremento];

mostrarInfo=true;
mostrarError=true;
mostrarStep=true;
mostrarResultado=true;
mostrarOpciones=[mostrarInfo mostrarError mostrarStep mostrarResultado];

% Se utiliza 1 es método Híbrido, con 0 es Backpropagation
metodoOptimizacion=1;
```

```
%Configura las opciones de entrenamiento
opcionesGenerales =
anfisOptions('InitialFIS',fisFinal,'ValidationData',datosTest);
```

```
% Entrenamiento del FIS con una red ANFIS con los datos de entrenamiento
[anfisEntrenada1,error1,ss,anfisEntrenada2,error2] =
anfis(datosEntrenamiento,opcionesGenerales,opcionesEntrenamiento,mostrarOpciones,[],metodoOptimizacion);
```

```
% Funciones de Pertenencia del sistema Entrenado
figure
subplot(2,2,1)
plotmf(anfisEntrenada2,'input',1)
subplot(2,2,2)
plotmf(anfisEntrenada2,'input',2)
subplot(2,2,3)
plotmf(anfisEntrenada2,'input',3)
subplot(2,2,4)
plotmf(anfisEntrenada2,'input',4)
```

```
% Gráficas de Evaluación
figure
plot([error1 error2])
```

```
hold on
plot([error1 error2], 'o')
legend('Error de Entrenamiento', 'Error de Test')
xlabel('Épocas')
ylabel('Root Mean Squared Error')
title('Curvas de Error')
```

```
%Comparar los valores predichos.
salidaAnfis = evalfis(anfisEntrenada2,[datosEntrenamiento(:,1:4);
datosTest(:,1:4)]);
datosOriginal = cat(1,datosEntrenamiento(:,5),datosTest(:,5))
figure
plot(1:929,[datosOriginal salidaAnfis])
axis([400 800 -0.3 1.2])
legend('Entrenamiento', 'Predicho')
ylabel("Realizados")
xlabel('Mantenimientos')
title('Predicción de Mantenimientos Realizados')
```

```
% Evaluación con nuevos datos aleatorios de amenaza
datosNuevosTemp1 = datosTest(:,4)
datosPerm = datosNuevosTemp1(randperm(length(datosNuevosTemp1)))
datosNuevosTemp2 = datosTest(:,1:3)
datosNuevos = cat(2,datosNuevosTemp2,datosPerm)
datosAnfisCheckEvalR = evalfis (anfisEntrenada2, datosNuevos);
datosNuevosFull = cat(2,datosNuevos,datosAnfisCheckEvalR)
```

```
% Evaluación de nuevos Datos
opcionesGenerales1 =
anfisOptions('InitialFIS',anfisEntrenada2, 'ValidationData',datosNuevosFull);
[anfisEntrenadaNew1,errorNew1,ss1,anfisEntrenadaNew2,errorNew2] =
anfis(datosTest,opcionesGenerales1,opcionesEntrenamiento,mostrarOpciones,[],me
todoOptimizacion);
salidaAnfis2 = evalfis(anfisEntrenadaNew2,[datosTest(:,1:4);
datosNuevosFull(:,1:4)]);
datosOriginal2 = cat(1,datosTest(:,5),datosNuevosFull(:,5));

figure
plot(1:1240,[datosOriginal2 salidaAnfis2])
axis([1 1240 -0.3 1.2])
legend('Entrenamiento', 'Predicho')
ylabel("Realizados")
xlabel('Mantenimientos')
title('Predicción de Mantenimientos Realizados')
```

```
% Gráficas de Evaluación del modelo clasificando nueva información.  
figure  
plot([errorNew1 errorNew2])  
hold on  
plot([errorNew1 errorNew2], 'o')  
legend('Error de Entrenamiento', 'Error de Test')  
xlabel('Épocas')  
ylabel('Root Mean Squared Error')  
title('Curvas de Error para modelo en producción')
```

Anexo. Artículo de investigación

Computación Autónoma para Centro de Datos mediante Agentes Inteligentes

Xavier Antonio Cabrera Burgos

Universidad Internacional de la Rioja, Logroño (España)



Fecha

RESUMEN

En el ámbito de los centros de datos, la computación autónoma se presenta como la solución a los problemas de gestión de tareas que requieren de personal altamente calificado. Este proyecto tiene como objetivo implementar una red neuronal difusa dentro de un elemento autónomo computacional para inferir si es seguro auto repararse. Se utilizó para este efecto la información recolectada en el periodo comprendido entre mayo de 2018 y mayo de 2020 de las tareas asignadas a ingenieros expertos que debían realizar mantenimientos de reparación correctivos y preventivos a los componentes de tecnologías de información de un centro de datos ubicado en Ecuador. En la información indicada se recolectaron 929 trabajos de mantenimiento, estos parámetros sirvieron para entrenar una red neuronal difusa que puede inferir si una tarea debe ser realizada o no. En la experimentación se modificaron aleatoriamente los datos de las amenazas para simular cambios en el entorno y se comprobó que la Raíz del error cuadrático medio fue 4.23×10^{-2} al utilizar la nueva información desconocida para el modelo ya entrenado. Con estos resultados se puede destacar la importancia de desarrollar sistemas de auto corrección que disminuyan las operaciones manuales en los centros de datos.

PALABRAS CLAVE

Centro de Datos, Computación Autónoma, Inteligencia Artificial, Lógica Borrosa.

I. INTRODUCCIÓN

EXISTEN muchas definiciones de inteligencia[1]. Coloquialmente se ha concebido como la capacidad de pensamiento, las ideas, el razonamiento y por supuesto la toma de decisiones es lo que los humanos percibimos como Inteligencia. En el campo científico se dice que la inteligencia es un bien medible y cuantificable clasificándolas por sus características psicométricas, biológicas, de desarrollo y múltiples [2]. La Inteligencia Artificial (IA) por su parte trata de emular este comportamiento mediante el uso de algoritmos que perciben el entorno en el que se desenvuelven para llevar acciones que simulen el razonamiento y la toma de decisiones. Los científicos de las ciencias de la computación han tomado este diseño biológico para inspirarse en dos sistemas diferenciados pero inclusivos. Por un lado, la IA que se podría definir como: “Un programa que, en un mundo arbitrario, no se las arreglará peor que un humano” [3] y por otro lado la computación autónoma se podría definir como la propuesta de arquitectura computacional para disminuir la carga administrativa de los operadores en los centros de datos [4].

La alta complejidad en la infraestructura tecnológica de los Centros de Datos no va a desaparecer [5], y con lo expuesto antes, es visible, que su tendencia en aumento va a continuar. Por lo tanto, el camino del automatismo es una de las múltiples estrategias que se deben utilizar para que la carga operativa de expertos sea trasladada a elementos computacionales autónomos que sean los encargados de solucionar los problemas

que pudieran presentarse. En la figura 1 se puede observar la tendencia de crecimiento de un centro de datos.

La computación autónoma tiene como base arquitectónica el

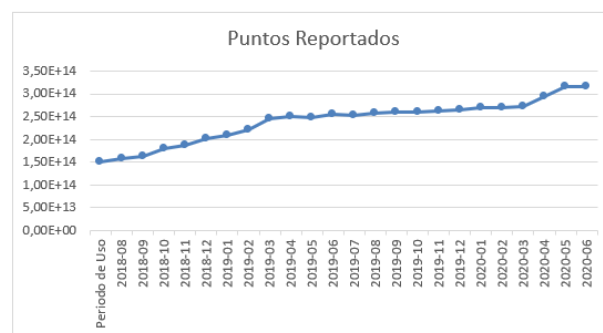


Figura 1. Tendencia del consumo en un centro de Datos. En la imagen se puede observar la tendencia del consumo de los últimos 24 meses, en el uso de software de virtualización VMware del centro de datos estudiado (Datacenter Millenium). Este software se mide por puntos en referencia a la memoria RAM usada. Nótese la subida marcada en los meses de abril y mayo de 2020 en donde los servicios de tecnología tuvieron una alta demanda a causa del confinamiento a nivel mundial

bucle de control de MAPE-K (Monitor-Analyze-Plan-Execute over a Shared Knowledge), el cual se utilizará como base de un agente inteligente al que se le incluirán capacidades de IA gracias

a la utilización de una red neuronal difusa (Neuro fuzzy inference system), misma que basa su arquitectura en la combinación de los conceptos de las redes neuronales profundas y las teorías de la lógica borrosa. Con la combinación de estos elementos se pretende que el agente sea capaz de poder determinar cuándo es seguro realizar una auto reparación en sus componentes.

II. ESTADO DEL ARTE

La industria busca arduamente disminuir siempre sus costos operativos [6]. Para ello han recurrido siempre a la automatización de sus procesos, sin embargo, esta es mecánica y no es suficiente [7], puesto que las necesidades del negocio siempre escalan y nos conducen hacia la complejidad en donde siempre se tiende a la incompletitud. Como una solución a este contexto, y con el avance de la potencia computacional, se propone la utilización de estrategias de IA como una ciencia de soporte que permite llevar un paso más allá este automatismo que es exigido por el dinamismo del negocio, permitiendo obtener resultados de la información que se puede capturar de los sistemas informáticos y mediante algoritmos computacionales basados en probabilidades matemáticas obtener resultados precisos cuando la demanda de expertos escasea.

En el centro de datos los elementos se agrupan por su función por medio de la ayuda de un TRM (Technical Reference Model) [8] el cual permite modelar mediante bloques de construcción la arquitectura de los componentes de tecnología del centro de datos de una manera estándar. La figura 2 muestra el TRM mencionado y como los componentes principales al interior del centro de datos y cómo estos se relacionan.

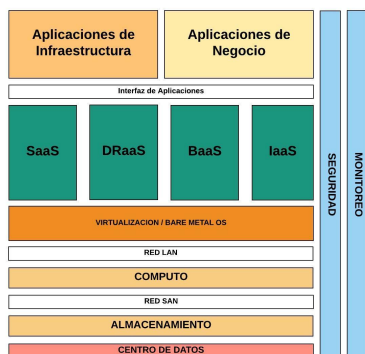


Fig. 2. TRM clásico de un centro de datos Con la ayuda de este modelo técnico de referencia es posible visualizar en alto nivel los diferentes componentes de un centro de datos y sus relaciones, en la parte más baja se encuentran los componentes base, como el almacenamiento, cómputo y la red. Componentes críticos de la operación al ser la base de las aplicaciones que están en la parte superior.

En este trabajo se utilizan elementos de la computación autónoma que utilizan redes neuronales. Los elementos de la computación autónoma se podrían explicar cómo componentes de dos niveles: En el nivel superior se encuentra el Administrador Autónomo, mismo que es gobernado por el bucle de control MAPE-K. En el nivel inferior es donde se encuentra el Elemento Administrado que está emitiendo y recibiendo señales hacia los sensores y desde los actuadores del Administrador autónomo. Es importante destacar que el elemento administrado puede ser un solo componente o muchos y en el caso de los centros de datos, estos generalmente son muchos y heterogéneos [9]. La figura 3 nos presenta un diagrama de cuáles son sus componentes y cómo interactuaría con los elementos del centro de datos.

Dentro de estos elementos autónomos se pueden insertar diversos sistemas, en este caso podrían ser de IA. Los sistemas de inferencia neuro difusos y las redes neuronales tienen cosas en

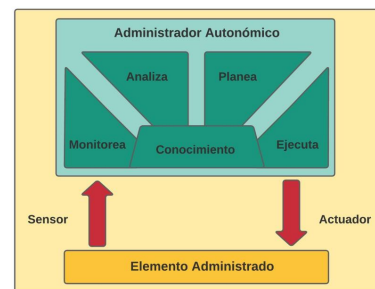


Fig. 3. Estructura de un elemento autónomo. Adaptado de [10] Estructura base de un elemento autónomo. Se puede visualizar los cuatro ciclos del bucle de control de MAPE-K. Se debe prestar especial atención al bloque que será intervenido en este proyecto, el de análisis.

común [11]. La figura 4 muestra una vista de alto nivel del proceso de inferencia en un sistema difuso.

Durante el inicio de la década de los 90 en el siglo XX, el interés en las redes neuronales fue muy activo, en este aspecto las Redes Neuronales de Inferencia Difusa no se quedaron atrás. En el año 1993 [12] aparece la propuesta de una red integrada llamada ANFIS (Adaptive-network-Based Fuzzy Inference System) con el objetivo de combinar lo mejor de los mundos de las redes neuronales y la lógica difusa.

La aplicación con redes neuronales para el control de elementos autónomos se observa en el año 2011 [13], se realizan investigaciones sobre la aplicación de redes neuronales en elementos autónomos que estén enfocados en realizar la auto reparación. En sus experimentos utilizan redes PRNN (Pipeline Recursive Neural Network) modificadas que permitan controlar y mantener el ancho de banda de forma óptima para los clientes que se conectan a un grupo de servidores. Sus hallazgos fueron positivos llegando a la conclusión de que es viable alimentar elementos autónomos con la capacidad de inferencia de redes neuronales.

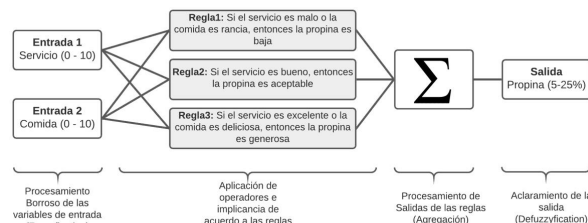


Fig. 4. Proceso de inferencia de un sistema de lógica difusa. [14] El clásico problema de una cena para dos, tratado con la conceptualización base del inicio de una red neuronal. En el mismo tenemos una capa de entrada en donde se procesan las variables, una capa intermedia en donde se procesan las reglas del sistema, una capa de agregación de las reglas de la capa anterior y una capa de salida con el resultado final del proceso.

En el año 2019 [15] en su lugar propone elementos autónomos que en su fase de planificación de MAPE-K contengan redes neuronales de tipo ANFIS para procesar seis variables de un entorno controlado por humanos, específicamente el de válvulas de control para gas y petróleo. Las variables manejadas son entrada de presión, temperatura, velocidad de flujo, controlador de salida y desplazamiento de la varilla de válvula. En este ejercicio se tiene conocimiento que poder identificar correctamente el tipo de fallo de estas válvulas y asociar los diferentes escenarios de corrección de errores es únicamente del dominio de expertos.

Sus hallazgos demostraron que es viable utilizar redes neuronales difusas tipo ANFIS en este tipo de escenarios, sin

embargo, existieron limitaciones que estuvieron relacionadas con la sensibilidad de la industria del gas.

III. OBJETIVOS Y METODOLOGÍA

El objetivo general del trabajo es proponer la utilización de estrategias basadas en la inteligencia artificial dentro del funcionamiento arquitectónico de la computación autónoma permitiendo delegar a estos sistemas las tareas que hoy en día requieren la intervención de un humano debido a la complejidad de los sistemas involucrados.

- Los objetivos específicos:
- Analizar la estrategia utilizada actualmente para realizar los procesos de mantenimiento a los elementos de un centro de datos.
 - Diseñar la metodología a implementar para permitir que un agente inteligente tenga la capacidad de inferir si un mantenimiento es viable o no.
 - Implementar la metodología con las distintas herramientas que se tenga a disposición.
 - Evaluar los resultados obtenidos del caso base y el logrado.

Para alcanzar los objetivos planteados en esta investigación se utiliza como metodología de trabajo la teoría que se utiliza para la construcción de sistemas expertos. Los sistemas expertos son piezas de software que se utilizan para ayudar a expertos [16].

IV. CONTRIBUCIÓN

En este proyecto investigativo se sentarán las bases de lo que se debe integrar dentro del elemento autónomo en la fase de análisis del ciclo de control MAPE-K. Esta propuesta coloca una red neuronal difusa adaptativa (ANFIS) que permita inferir si el proceso de mantenimiento (“self-healing”) de un equipo del centro de datos debe proceder o no. La figura 5 muestra el diseño de alto nivel de esta solución.

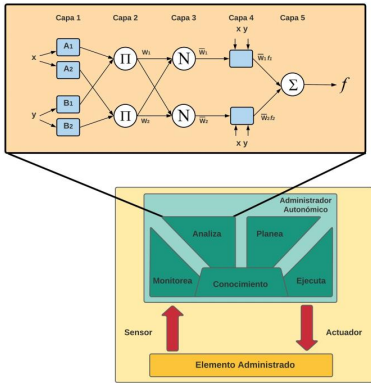


Fig. 5. Elemento autónomo que incluye una red ANFIS en la fase de análisis. En la imagen se realiza una ampliación de lo que sería el interior de la fase de análisis de nuestro elemento autónomo propuesto. La red neuronal de tipo ANFIS recibiría los datos que utilizan los expertos para inferir si las condiciones para realizar un mantenimiento son favorables o no.

La red neuronal ANFIS al interior de la fase de análisis está definida por las siguientes entradas expuestas en la tabla 1.

La valoración de cada una de las entradas se realizó mediante la colaboración de expertos del centro de datos. Puede llamar la atención en la estructura de ingreso a la red neuronal, el colocar

los elementos identificables del TRM y que cumplen funciones específicas de almacenamiento, red y cómputo en una de las entradas difusas. La justificación de esto es porque entre más alto en el TRM se encuentren los elementos, mayor es la redundancia que estos pueden alcanzar, por lo tanto, es más seguro realizar mantenimiento sobre ellos. Además de que existen híbridos de almacenamiento-cómputo que no pueden ser encasillados en

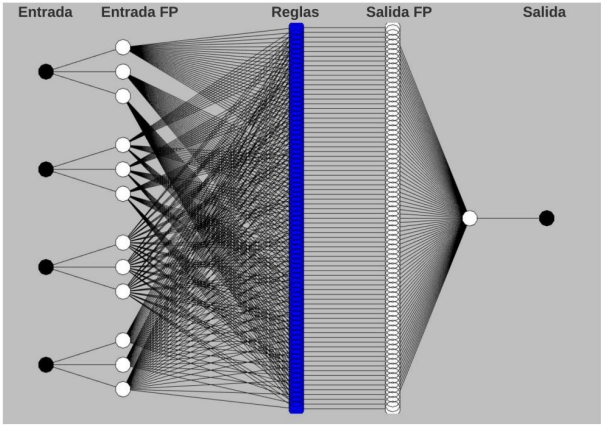


Fig. 6. Estructura de la red neuronal ANFIS para la solución propuesta. La estructura del sistema de inferencia difuso tiene 4 entradas que se mostraron en la tabla 5. Cada entrada tiene tres entradas de las funciones de pertenencia para identificar el nivel en el cual un elemento autónomo se encuentra. A continuación, las reglas que procesan esas funciones de pertenencia que lo enrutan hacia una determinada función de salida.

alguna categoría específica.

Al ser una red tipo ANFIS de clase Sugeno, tiene una sola salida en la cual se agrupan las funciones de pertenencia para cada

TABLA I
ENTRADAS DE LA RED ANFIS

INGRESO A LA RED	FUNCIONES DE PERTENENCIA
Posición en el TRM	Infraestructura
	Plataforma
	Aplicaciones
Disponibilidad del Elemento	Baja
	Media
	Alta
Críticidad del Elemento	Baja
	Media
	Alta
Amenaza Externa	Baja
	Media
	Alta

una de las funciones de pertenencia de entrada. La red elaborada tiene un total de 81 funciones de pertenencia de salida para la evaluación de cada regla. En la figura 6 se puede observar la estructura de la red propuesta mediante el uso del software MatLab de MathWorks.

Cada función de pertenencia se ha delimitado de cero a uno y dependiendo del objetivo que se quiera lograr los elementos son ubicados en algún punto de esta escala para que sean clasificados por las funciones de pertenencia.

La aplicación del método propuesto está basada en el flujo de control de MAPE-K para la sección de análisis del elemento autónómico. Anteriormente dentro del elemento autónómico se colocó en la figura 5 una representación de alto nivel de la solución propuesta, que es incluir una red ANFIS dentro de esa parte del flujo de control. Esta red puede o no ser activada por las acciones que la fase de monitoreo envíe a la de análisis

El flujo de acción del método propuesto se expone en la figura 7.

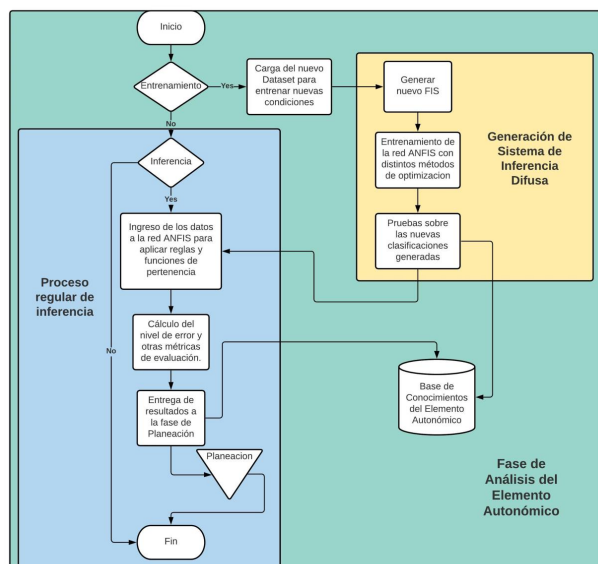


Fig. 7. Diagrama de flujo del funcionamiento de alto nivel de la fase de análisis. El diagrama de flujo presenta dos bloques principales. A la izquierda dentro del recuadro azul el proceso normal de inferencia del cual el elemento autónómico procesará las señales enviadas desde la fase de monitoreo y que serán a su vez, enviadas a la fase de planificación. A la derecha, en el recuadro amarillo, el proceso de entrenamiento y aprendizaje de la red en el caso de que amerite realizarlo nuevamente

V. EVALUACIÓN Y RESULTADOS

Para evaluar el funcionamiento de lo anteriormente propuesto se tuvo la colaboración de los datos de mantenimientos realizados por el centro de datos estudiado (Datacenter Millenium) a lo largo de dos años, específicamente desde Mayo de 2018 hasta Junio de 2020. El mismo contaba con un total de 929 mantenimientos realizados de diversas categorías entre los que había mantenimientos realizados y cancelados por diversos motivos. Es aquí donde el elemento autónómico inteligente propuesto tiene la intención de reemplazar la decisión de un experto y que el sistema pueda cubrir esa decisión.

Sobre esta información origen se utilizaron las columnas de Tipo de mantenimiento, Descripción, Realizado y Cancelación para tener la base de los datos que serán ingresados en el sistema difuso. Estos datos fue necesario adecuarlos para ajustarlos a la entrada de lo que sería la red ANFIS propuesta en la sección anterior.

Al margen de si un elemento tiene disponibilidad o criticidad alta, en las operaciones del centro de datos los mantenimientos a los equipos deben realizarse para asegurar la calidad del servicio. Es de especial interés conocer cuántos elementos pertenecientes al TRM se le han realizado mantenimientos exitosos. La figura 8 nos da una relación de los mantenimientos realizados y los que no.

Luego del análisis de los datos a ingresarse, de acuerdo con el flujo de la figura 7, corresponde la creación del Sistema de

Inferencia Difuso FIS (Fuzzy Inference System) y el entrenamiento de los datos. Para el efecto se ha hecho uso del Software Matlab, mismo que tiene una amplia colección de herramientas para la aplicación de técnicas de lógica difusa. Dentro del software de MatLab se dispone de un asistente gráfico que permite la creación del sistema, mediante el comando “neuroFuzzyDesigner”.

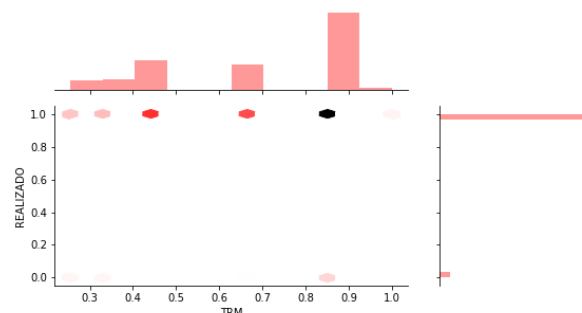


Fig. 8. Gráfico de Correlación entre los elementos del TRM y los mantenimientos realizados. Como se puede observar, antes de su la mayor parte de elementos de centro de datos tienen cumplidos sus mantenimientos (histograma y valor cercano a 1). Mientras aquellos a los que no se les ha realizado son muy pocos.

De los datos recolectados se separaron dos conjuntos de datos, uno para entrenamiento y otro para validación

Para el entrenamiento de la red ANFIS primero se debe realizar la carga del FIS anteriormente creado y ejecutarlo mediante el comando de entrenamiento. La figura 9 muestra el resultado del entrenamiento.

```
ANFIS info:
Number of nodes: 193
Number of linear parameters: 81
Number of nonlinear parameters: 24
Total number of parameters: 105
Number of training data pairs: 309
Number of checking data pairs: 620
Number of fuzzy rules: 81

Start training ANFIS ...

1    0.00639606    0.0393572
2    0.00532296    0.0368396
3    0.0043081    0.034461
4    0.00337409    0.0322499
5    0.00254186    0.0302347
Step size increases to 0.011000 after epoch 5.
6    0.00182812    0.028444
7    0.00119289    0.0267695
8    0.000806862    0.0257511
9    0.00123623    0.0270551
10   0.00076275    0.0257289

Designated epoch number reached --> ANFIS training completed at epoch 10.

Minimal training RMSE = 0.000763
Minimal checking RMSE = 0.0257289
```

Fig. 9. Entrenamiento de la red ANFIS. La red ANFIS se entrena en el sistema FIS con los datos de entrenamiento y test cargados. El RMSE obtenido en esta etapa es de 2.57×10^{-2} , por lo cual se puede concluir que los valores de entrenamiento y test son bastante cercanos.

Evaluación 1

Los valores de prueba provienen del mismo conjunto de datos original pero previamente particionado al 30%. Sobre este conjunto de datos ya se tenía los resultados iniciales de los mantenimientos realizables o no realizables, por lo que es posible comparar el funcionamiento de la red neuronal.

La figura 10 muestra un extracto de los resultados obtenidos en la primera prueba y su comparación con los datos originales

TRM	DISPONIBILIDAD	CRITICIDAD	AMENAZA	REALIZABLE	PREDICHO ANFIS
0,2566	0,8888	0,9999	0,3333	1	1,0001
0,6666	0,9999	0,9999	0,3333	1	0,9995
0,2566	0,8888	0,9999	0,3333	1	1,0001
0,3333	0,9999	0,9999	0,3333	1	0,9999
0,6666	0,9999	0,9999	0,3333	1	0,9995
0,3333	0,9999	0,9999	0,3333	1	0,9999
0,6666	0,9999	0,9999	0,3333	1	0,9995
0,8566	0,9999	0,9999	0,9997	0	0,1281
0,3333	0,9999	0,9999	0,3333	1	0,9999
0,8566	0,9999	0,9999	0,9997	0	0,0005
0,3333	0,9999	0,9999	0,3333	1	0,9999
0,6666	0,9999	0,9999	0,3333	1	0,9995
0,3333	0,9999	0,9999	0,3333	1	0,9999
0,3333	0,9999	0,9999	0,3333	1	0,9999
0,2566	0,8888	0,9999	0,3333	1	1,0001
0,4555	0,5555	0,6666	0,3333	1	0,9999
0,6666	0,9999	0,9999	0,3333	1	0,9995
0,8566	0,9999	0,9999	0,9997	0	0,0005
0,2566	0,8888	0,9999	0,3333	1	1,0001

Fig. 10. Comparativa de los resultados obtenidos contra los originales. Los campos resaltados en amarillo son los trabajos que tuvieron que cancelarse por algún motivo. Es importante destacar que estos trabajos cancelados tienen un nivel de amenaza superior en relación con otros trabajos. La evaluación mediante la red ANFIS predice adecuadamente cuando un trabajo debe ser cancelado o no.

Evaluación 2

La siguiente prueba que se realizó con la red ANFIS es modificar los datos de amenaza. Esto es, mantener los datos de TRM, Disponibilidad, Criticidad, y solamente modificar de forma aleatoria ciertos valores de amenaza para saber si la red neuronal puede procesar adecuadamente valores nuevos sobre los cuales no tiene información previa. La figura 11 muestra el resultado de esta modificación.

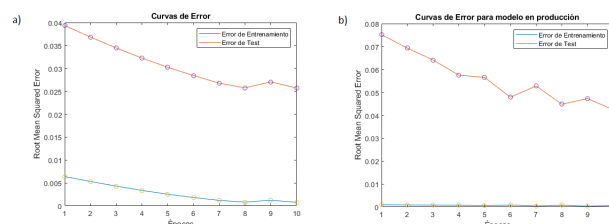
AMENAZA ORIGINAL	AMENAZA MODIFICADA	REALIZABLE	PREDICHO ANFIS	PREDICHO AM
0,3333	0,3333	1	1,0001	1,0001
0,3333	0,3333	1	0,9995	0,9995
0,3333	0,3333	1	1,0001	1,0001
0,3333	0,3333	1	0,9999	0,9999
0,3333	0,3333	1	0,9995	0,9995
0,3333	0,3333	1	0,9999	0,9999
0,3333	0,3333	1	0,9995	0,9995
0,9997	0,0957	0	0,1281	3,6229
0,3333	0,3333	1	0,9999	0,9999
0,9997	0,9997	0	0,0005	0,0005
0,3333	0,1658	1	0,9999	1,7473
0,3333	0,9778	1	0,9995	-0,1616
0,3333	0,3333	1	0,9999	0,9999
0,3333	0,3333	1	0,9999	0,9999
0,3333	0,3333	1	1,0001	1,0001
0,3333	0,3333	1	0,9999	0,9999
0,3333	0,3333	1	0,9995	0,9995
0,9997	0,9997	0	0,0005	0,0005
0,3333	0,3333	1	1,0001	1,0001

Fig. 11. Validación del funcionamiento de la red con datos nuevos. Este cuadro valida que el sistema propuesto es efectivo con referencia a lo que se desea lograr. En el caso de que la amenaza original es modificada con valores menores, la red ahora cataloga el trabajo como “realizable” (valores superiores a uno). Mientras que una amenaza menor que fue modificada con un valor mayor al que tenía lo marca un mantenimiento como no realizable, ya que ahora considera que las condiciones son adversas y es más seguro cancelarlo.

VI. DISCUSIÓN

Luego de la experimentación con los datos ingresados a la red neuronal, primero en el entrenamiento original sin modificación de sus valores y luego los nuevos datos con información desconocida para la red y su procesamiento, se observa que el comportamiento en ambas situaciones es muy similar.

Eso queda demostrado en las curvas de error que se generaron para cada uno de los experimentos. La figura 12 muestra el resultado del RMSE (Root Mean Square Error) son similares.



VII. CONCLUSIONES

La solución propuesta en esta investigación utiliza estrategias de inteligencia artificial que permiten llevar a las organizaciones hacia una madurez autónoma.

La utilización de sistemas como este que combinan los conceptos fundacionales de la computación autónoma [17] y los de lógica difusa por el lado de la inteligencia artificial [18], abre las puertas a nuevas condiciones que permitan a los expertos poderse enfocar en otros aspectos del negocio, disminuyendo la complejidad de los activos de tecnología y aumentando la eficiencia.

Los datos analizados sobre los mantenimientos realizados en este trabajo investigativo demuestran que actualmente se sigue delegando la decisión final de realizar un mantenimiento a las manos de expertos que puedan determinar el momento para llevarlos a cabo.

La metodología propuesta en este documento se basa en la utilización de herramientas como el TRM de TOGAF, Computación Autónoma, Lógica Difusa, Redes ANFIS en el alto nivel y Matlab, Python y Jupyter Notebook en el bajo nivel para procesar los datos. La flexibilidad del sistema permite la modificación de ingreso de parámetros para la inclusión o retiro de estos, de acuerdo con las condiciones en las que se necesite ejecutar el sistema propuesto.

La red ANFIS implementada consta de 4 entradas, las cuales se forjaron desde los datos origen de las 929 tareas de mantenimiento analizadas. Estas entradas denotan la posición del elemento en el TRM, disponibilidad, criticidad y amenaza que se utilizaron para inferir si el mantenimiento es realizable o no, al atravesar las 3 funciones de pertenencia de cada entrada y las 81 reglas establecidas en el sistema FIS. Además, se utilizó el método de cálculo de gradiente “backpropagation” y al final una capa de salida que retorna el valor “defuzzyficado” entre 0 y 1.

En la evaluación de los resultados, la red ANFIS entrenada demostró que pudo clasificar nuevos datos de mantenimientos de forma correcta, siendo el valor de amenaza el de mayor variabilidad y peso para la toma de decisiones. El resultado del RMSE de la red original con respecto a la red que clasificó los datos desconocidos al entrenamiento inicial tiene una diferencia de 1.83×10^{-2} .

Por lo tanto, la contribución de este trabajo en ayudar a los expertos al eliminar los procesos manuales y facilitando la planeación de mantenimientos de manera automatizada se considera cumplida. Además, este pasa a ser un método muy eficiente para ser utilizado por los demás miembros de la organización que no cuenten con la experiencia suficiente, pasarían a confiar en el sistema y el mismo les proporcionaría los resultados adecuados, puesto que ya cuenta con información de

experiencia de expertos en su base de conocimientos. En resumen, se puede trabajar más rápido, se evitan los errores humanos y se consolida el conocimiento y la experiencia en un solo sistema.

La metodología propuesta en este estudio tiene limitantes, como el aspecto de la obtención de datos. Estos deberían ser tratados como investigaciones posteriores en el que se estudie cada fase del bucle de control MAPE-K [18]. Otros trabajos deben enfocarse en la recopilación de datos automatizada desde el monitoreo y que es enviada a la fase de análisis propuesta en este trabajo.

De igual forma los valores que se obtienen en la fase de análisis deben ser procesados en la fase de planeación y posterior ejecución, del elemento autonómico para de esta forma completar el ciclo del bucle de control indicado. Automatizando así, todo el proceso del elemento autonómico desarrollado para la automatización de la auto reparación (“self-healing”) en el centro de datos

REFERENCIAS

- [1] Moreno, C. M., Vicente, E. S., & Martínez, C. E. (1998). Revision histórica del concepto de inteligencia: Una aproximación a la inteligencia emocional. *Revista Latinoamericana de Psicología*, 30(1), 11–30.
- [2] G. Villamizar and R. Donoso, “Revisión Histórica Definitions and Theories About Intelligence .,” *Psicogente*, vol. 16, pp. 407–423, 2013.
- [3] D. Dobrev, “A Definition of Artificial Intelligence,” pp. 1–7, 2012.
- [4] A. C. Obien, “Autonomic Computing : An Evidence of Better Management Technology,” no. February, pp. 941–954, 2018.
- [5] M. Fadaeefath Abadi, F. Haghighat, and F. Nasiri, “Data center maintenance: applications and future research directions,” *Facilities*, 2020, doi: 10.1108/F-09-2019-0104.
- [6] NASDAQ OMX’s, “OPSANI INTRODUCES FREE CLOUD COST REDUCTION SERVICES FOR CLOUD APPS: Project ‘Vital’ Helps Companies Lower Cloud Bills By Up to 70 Percent at No Cost for Three Months,” News Release Distribution Channel, 2020.
- [7] M. Georgeff, B. Pell, M. Pollack, M. Tambe, and M. Wooldridge, “The Belief-Desire-Intention Model of Agency The Belief-Desire-Intention Model of Agency,” no. February 1970, 2014, doi: 10.1007/3-540-49057-4.
- [8] The Open Group, “TOGAF Standar Version 9.2,” pp. 1–532, 2018.
- [9] IBM, “An architectural blueprint for autonomic computing,” 2005.
- [10] O. Jeffrey and M. David, “The Vision of Autonomic Computing,” *Computer (Long. Beach. Calif.)*, vol. 18, no. January, pp. 41–50, 2003, doi: 10.1046/j.1365-2745.2002.00730.x.
- [11] R. Kruse, “Fuzzy neural network,” *Scholarpedia*, vol. 3, no. 11, p. 6043, 2008, doi: 10.4249/scholarpedia.6043.
- [12] J. S. R. Jang, “ANFIS: Adaptive-Network-Based Fuzzy Inference System,” *IEEE Trans. Syst. Man Cybern.*, vol. 23, no. 3, pp. 665–685, 1993, doi: 10.1109/21.256541.
- [13] M. Mousa Al-Zawi, D. Al-Jumeily, A. Hussain, and A. Taleb-Bendiab, “Autonomic computing: Applications of self-healing systems,” *Proc. - 4th Int. Conf. Dev. eSystems Eng. DeSE 2011*, pp. 381–386, 2011, doi: 10.1109/DeSE.2011.22.
- [14] Mathworks, “Fuzzy Inference Process - MATLAB & Simulink - MathWorks América Latina,” 2020. [Online]. Available: <https://la.mathworks.com/help/fuzzy/fuzzy-inference-process.html>. [Accessed: 08-Sep-2020].
- [15] H. Naseri, A. Shahidinejad, and M. Ghobaei-Arani, “An intelligent autonomous system for condition-based maintenance-Case study: Control valves,” *J. Inf. Syst. Telecommun.*, vol. 7, no. 2, pp. 144–154, 2019.
- [16] Seyed Muhammad Hossein Mousavi, -S. Younes Mirinezhad, and -Vyacheslav Lyashenko, “An Evolutionary-Based Adaptive Neuro-Fuzzy Expert System as a Family Counselor before Marriage with the Aim of Divorce Rate Reduction,” *Conf. 2nd Int. Conf. Res. Knowl. Base Comput. Eng. IT*, no. August, pp. 1–10, 2017.
- [17] S. Hassan, D. McSherry, and D. Bustard, “Autonomic self healing and recovery informed by environment knowledge,” *Artif. Intell.*

Rev., vol. 26, no. 1–2, pp. 89–101, 2006, doi: 10.1007/s10462-007-9033-6.

- [18] P. Lalanda, J. a. McCann, and A. Diaconescu, *Autonomic Computing*, no. April. London: Springer London, 2013.