



**FACULTAD DE POSTGRADO
TRABAJO FINAL DE GRADUACIÓN**

**INTELIGENCIA ARTIFICIAL COMO SOPORTE EN LA
GESTIÓN Y OPTIMIZACIÓN DE LAS TECNOLOGÍAS DE
INFORMACIÓN**

(CASO DATAWAREHOUSE TIGO HONDURAS)

SUSTENTADO POR:

MAX JHAZIEL MONDRAGON ESPINOZA

**PREVIA INVESTIDURA AL TÍTULO DE
MÁSTER EN GESTIÓN DE TECNOLOGÍAS DE
INFORMACIÓN**

TEGUCIGALPA, FRANCISCO MORAZÁN, HONDURAS, C.A.

ABRIL 2019

UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

UNITEC

FACULTAD DE POSTGRADO

AUTORIDADES UNIVERSITARIAS

RECTOR

MARLON ANTONIO BREVÉ REYES

SECRETARIO GENERAL

ROGER MARTÍNEZ MIRALDA

DECANO DE LA FACULTAD DE POSTGRADO

CLAUDIA MARÍA CASTRO VALLE

**INTELIGENCIA ARTIFICIAL COMO SOPORTE EN LA
GESTIÓN Y OPTIMIZACIÓN DE LAS TECNOLOGÍAS DE
INFORMACIÓN**

(CASO DATAWAREHOUSE TIGO HONDURAS)

**TRABAJO PRESENTADO EN CUMPLIMIENTO DE LOS
REQUISITOS EXIGIDOS PARA OPTAR AL TÍTULO DE
MÁSTER EN
GESTIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN**

ASESOR

JORGE RAÚL MARADIAGA CHIRINOS

MIEMBROS DE LA TERNA:

FREDIS MEDINA

IDALIA CARCAMO

JUAN CARLOS ALMENDAREZ

INTELIGENCIA ARTIFICIAL COMO SOPORTE EN LA GESTIÓN Y OPTIMIZACIÓN DE LAS TECNOLOGÍAS DE INFORMACIÓN (CASO DATA WAREHOUSE TIGO HONDURAS)

Max Jahaziel Mondragón Espinoza

Resumen

El presente trabajo expone el proceso de aplicación de tecnologías de inteligencia artificial, concretamente redes neuronales; a los procesos de optimización y eficiencia en las actividades de un departamento de Tecnologías de Información, aplicación en Data Warehouse. Sumado a la búsqueda de la mejora continua y optimización de los riesgos intrínsecos de la operativa. La iniciativa surge como respuesta a la necesidad de hacer más eficientes los procesos, reducir el impacto de las fallas y mejorar los tiempos de entrega pactados en acuerdos de distintos niveles. Todo esto con la limitante de recursos (tiempo y personal), considerando la posible utilidad de tecnologías disruptivas como ser la Inteligencia Artificial.

Palabras clave: (Eficiencia, Inteligencia Artificial, Optimización, Redes Neuronales).

GRADUATE SCHOOL

ARTIFICIAL INTELLIGENCE AS SUPPORT FOR THE MANAGEMENT AND OPTIMIZATION OF INFORMATION TECHNOLOGIES (CASE DATA WAREHOUSE TIGO HONDURAS)

Max Jahaziel Mondragón Espinoza

Abstract

The present work shows the process of application of artificial intelligence technologies, specifically neuronal networks; to the processes of optimization and efficiency in the activities performed by a department of Information Technologies, application in Data Warehouse. As well as the search for continuous improvement and optimization of the intrinsic risks of the operation. The initiative rises as a response to the need to improve processes and make them more efficient, reduce the impact of failures and improve delivery times promised in agreements at different levels. All this with the limitation of resources (time and personnel), considering the possible usefulness of disruptive technologies such as Artificial Intelligence.

Keywords: (Artificial Intelligence, Efficiency, Neural Networks, Optimization).

DEDICATORIA

Este trabajo, y todo el esfuerzo que ha llevado, lo dedico a mi padre: José Max Mondragón Canales, a mi madre: María Luisa Espinoza Mendoza y a mis hermanas: Bitia Milalai Mondragón Espinoza y Miriam Ariel Mondragón Espinoza.

Sin su apoyo, motivación, paciencia y guía; este logro no habría sido posible. Todo el tiempo invertido sumado a esas noches de desvelo y los momentos gritando Eureka! en las pruebas y experimentos finales.

Con amor y el más profundo respeto, gracias.

AGRADECIMIENTOS

La educación es uno de los más importantes pilares que sustentan al individuo y a la sociedad. Es la semilla que, con el tiempo, se transforma en éxito y por ende el más grande regalo que un padre puede dar a sus hijos.

Doy gracias primeramente a Dios por la vida, seguido de mis padres ya que desde mi infancia han trabajado sin descanso para darme a mí y a mis hermanas ese valioso regalo que es la educación. No solo en lo académico, sino también valores como el respeto, tolerancia, disciplina y perseverancia. Motivación que me han impulsado constantemente a fijar metas como la que con este trabajo se cumple.

Agradezco a mi abuela, Celaura Canales, quien siempre me ha apoyado en todo cuanto puede.

Agradezco a mis maestros, ya que compartiendo su experiencia y conocimiento hacen de Honduras un mejor país.

Agradezco a mis amigos, que durante mis estudios de maestría no tuvimos viernes de ocio.

A las autoridades académicas de UNITEC y mis compañeros de labores en Tigo.

Y agradezco también a mis colegas de estudios, que se han mantenido constantes día a día durante este desafío y que han demostrado ser profesionales dignos de este título.

ÍNDICE DE CONTENIDO

DEDICATORIA	IX
AGRADECIMIENTOS	X
ÍNDICE DE CONTENIDO	XI
ÍNDICE DE FIGURAS.....	XIV
ÍNDICE DE TABLAS	XVI
Capítulo I - PLANTEAMIENTO DE LA INVESTIGACIÓN.....	1
1.1 Introducción	1
1.2 Antecedentes del Problema	2
1.3 Definición del problema.....	4
1.4 Objetivos del proyecto	4
1.4.1 Objetivo General.....	4
1.4.2 Objetivos Específicos.....	4
1.5 Justificación.....	5
Capítulo II - MARCO TEÓRICO.....	6
2.1 Generación de Valor.....	6
2.2 COBIT	7
2.3 Optimización del Riesgo	9
2.4 ITIL	11

2.5	Actualidad en empresa piloto.....	12
2.6	Técnicas de Análisis de Datos.....	16
2.6.1	Análisis de Correlaciones	16
2.6.2	Análisis de Regresión	16
2.6.3	Minería de Datos.....	16
2.6.4	Clusterización de Datos	17
2.6.5	Algoritmos Matemáticos.....	17
2.7	Inteligencia Artificial	19
2.8	Redes neuronales.....	20
2.9	Aprendizaje y entrenamiento	24
2.10	Aprendizaje profundo (Deep Learning).....	25
2.11	Aplicación de Redes Neuronales.....	28
Capitulo III - METODOLOGÍA DE LA INVESTIGACIÓN		29
3.1	Enfoque de la investigación	29
3.2	Alcance de la investigación.....	30
3.3	Técnicas e instrumentos a utilizar	33
3.4	Fuentes de Información.....	33
3.5	Definición de Variables.....	34
Capitulo IV – ANÁLISIS Y RESULTADOS		36
4.1.	Preparación de modelo de pruebas.....	36

4.2.	Compuertas Lógicas.....	38
4.2.1.	Saturación de la red.....	45
4.3.	Experimento 1. Detección de procesos ineficientes.....	46
4.3.1.	Recopilación de datos	46
4.3.2.	Muestra de datos	49
4.3.3.	Preparación de sistema neural.....	51
4.3.4.	Pruebas funcionales	53
4.3.5.	Aplicación de resultados Experimento 1	56
4.4.	Experimento 2. Detección de fallos.	63
4.4.1.	Muestra de datos	64
4.4.2.	Preparación del sistema neural.....	67
4.4.3.	Pruebas funcionales	68
4.5.	Referente a metodología y resultados.	71
Capitulo V – CONCLUSIONES Y RECOMENDACIONES.....		72
5.1.	Conclusiones	72
5.2.	Recomendaciones.....	74
Bibliografía.....		75

ÍNDICE DE FIGURAS

Figura 1. Componentes de la creación de Valor.	8
Figura 2. Ciclo de Vida de los Servicios en TI.	11
Figura 3. Alerta enviada vía SMS por evento de fallo en un proceso determinado.	13
Figura 4. Comparativo de precisión entre Redes Neuronales a partir de los años 90.....	18
Figura 5. Estructura general de una Neurona Biológica.	21
Figura 6. Función matemática de procesamiento.	21
Figura 7. Funcionamiento de una neurona biológica.....	22
Figura 8. Representación matemática del funcionamiento de una neurona.....	23
Figura 9. Red de neuronas interconectadas.....	24
Figura 10. Representación de Propagación hacia atrás.....	26
Figura 11. Ejemplo de Descenso de Gradiente.....	27
Figura 12. Aplicaciones de Redes Neuronales.	28
Figura 13. Función Sigmoide.....	37
Figura 14. Compuertas Lógicas.	39
Figura 15. Red propuesta.	39
Figura 16. Proyecto de solución a Compuertas Lógicas.....	40
Figura 17. Prueba AND.	40
Figura 18. Prueba OR.	41
Figura 19. Prueba NAND.	41
Figura 20. Prueba NOR.....	42
Figura 21. Prueba XOR.....	42
Figura 22. Prueba XNOR.....	43

Figura 23. Grafica del Error durante 3 millones de iteraciones.....	44
Figura 24. Curva normal con división de cuartiles.....	48
Figura 25. Mejoras agregadas a programa generador de red neural.....	52
Figura 26. Diseño de red 5x9x4x1 propuesto para experimento 1.....	53
Figura 27. Datos de entrenamiento.....	54
Figura 28. Salida del programa.....	54
Figura 29. Optimización proceso 1.....	59
Figura 30. Optimización proceso 2.....	60
Figura 31. Optimización proceso 3.....	61
Figura 32. Impacto de los cambios en el tiempo de entrega de información.....	62
Figura 33. Interconexión de un Data Warehouse con múltiples sistemas.....	63
Figura 34. Ejemplo inicial de muestra de datos de entrada.....	65
Figura 35. Ejemplo de datos de entrada para la red.....	66
Figura 36. Ejemplo de datos de salida para la red.....	67
Figura 37. Red generada para el experimento, 30 x 50 x 8.....	68
Figura 38. Comparación 1 valores esperados vs obtenidos.....	69
Figura 39. Comparación 2 valores esperados vs obtenidos.....	69
Figura 40. Comparación 3 valores esperados vs obtenidos.....	70

ÍNDICE DE TABLAS

Tabla 1. Análisis FODA.	15
Tabla 2. Pregunta de Investigación.	30
Tabla 3. Preguntas de investigación específicas.	31
Tabla 4. Hipótesis.	32
Tabla 5. Variables experimento 1.	34
Tabla 6. Variables experimento 2.	35
Tabla 7. Datos de muestra.	49
Tabla 8. Normalización de datos.	50
Tabla 9. Datos de salida.	55
Tabla 10. Priorización de resultados.	56

CAPITULO I - PLANTEAMIENTO DE LA INVESTIGACIÓN

1.1 Introducción

El constante crecimiento económico y su consecuente necesidad de agilización de los procesos y mejoras en la calidad y eficiencia han sido algunos de los principales impulsores del avance tecnológico y su propia aceptación, tanto a nivel empresarial y gubernamental, así como parte fundamental del estilo de vida de la población.

Las tecnologías de información y comunicaciones (TIC) han evolucionado en respuesta a las crecientes necesidades institucionales de proveer mejores servicios en un entorno sumamente competitivo, con un enfoque orientado al cliente donde fallar se vuelve cada vez más crítico y con un impacto potencial cada vez mayor.

Las soluciones tecnológicas han venido a cambiar radicalmente el cómo las empresas hacen negocios, permitiendo pasar a escalas que, de otro modo, serían extremadamente difíciles de alcanzar. Sin embargo, esto ha generado una dependencia e interrelación de la tecnología con el negocio al grado que se ha convertido en un pilar base del mismo, y como tal la necesidad de su funcionamiento de clase mundial; es decir óptima e impecable.

Es evidente que las soluciones tecnológicas no son capaces, al menos por ahora, de proporcionar un servicio perfecto e ininterrumpido. Conforme el entorno cambia, incluso sea solamente en el factor tiempo, la tecnología puede y seguramente va a fallar. El impacto varía, pero algunos fallos pueden hacer tambalear los cimientos de la organización o incluso llevarla al fracaso.

Este es el enfoque que el presente trabajo aborda. Desde las recomendaciones estandarizadas para la gestión del riesgo hasta la propuesta de un sistema de soporte funcional para la toma de decisiones basado en inteligencia computacional; con el propósito de apoyar el proceso de gestión y optimización del riesgo en TI.

1.2 Antecedentes del Problema

Son muchas las empresas donde las Tecnologías de Información no son solamente una ayuda o soporte del negocio, sino la base donde se construye el mismo. No indicando que esto sea algo malo, al contrario; el grado de madurez tecnológico que estas empresas han alcanzado les permite ofrecer una calidad de servicio que en otros tiempos sería inimaginable.

Por otro lado, es a través de la historia que podemos observar múltiples ejemplos donde las empresas no han podido asegurar sus servicios tecnológicos al grado necesario. Muchas veces esto ha sido aprendido por las malas experiencias.

Un ejemplo es el caso de Delta Air Lines, agosto 2016. Donde un fallo computacional atribuido a una pieza de hardware en su centro de datos causo la cancelación de 2,300 vuelos, generando 3 días de caos y un costo estimado en 100 millones de dólares americanos ese mismo mes. (Financial Times, 2017).

En junio de 2015, alrededor de 600,000 pagos y débitos de clientes del Banco Real de Escocia (Royal Bank of Scotland), se perdieron resultado de la tecnología arcaica con la que la institución difícilmente operaba. El año siguiente la misma es multada por la regulación del Reino Unido por 56 millones de libras (aproximadamente 82 millones de dólares americanos). (Financial Times, 2017).

En abril de 2018 millones de clientes del banco británico TSB se vieron privados de acceder a los servicios de banca en línea, donde una actualización planificada para un fin de semana terminó alargándose causando meses de interrupción. Cuando el servicio fue finalmente reestablecido, múltiples clientes presentaban quejas ya que veían información incorrecta o incluso de otras personas. Algunos reportes de The Financial Times, apuntan a que el grupo bancario considera vender TSB. (Computer World UK, 2018)

En abril de 2018, un fallo de hardware en el centro de datos de Tigo Honduras, causo una interrupción de los servicios de información e inteligencia desde el Data Warehouse¹, causando una afectación de casi un mes, con un costo estimado de más de 800,000 dólares americanos solo en costo de oportunidad. Se desconoce el alcance total de las pérdidas.

Las causas de los fallos son muchas y muy variadas, sin embargo, algo es común. Un fallo en TI puede ser catastrófico para las empresas, sea esto en temas financieros o de reputación por lo que surge la necesidad de gestionar de alguna forma estos riesgos en tecnología.

¹ Almacén de Datos. Departamento dedicado a la gestión de datos e información empresarial.

1.3 Definición del problema

¿Cómo puede la tecnología y aplicación de la inteligencia artificial, apoyar en la gestión, optimización, mejora continua y aseguramiento de los servicios de Tecnologías de Información dentro de un entorno competitivo y cambiante?

1.4 Objetivos del proyecto

1.4.1 Objetivo General

Demostrar la aplicación de tecnologías de inteligencia artificial como apoyo a la gestión, toma de decisiones y optimización en los procesos de Tecnologías de Información.

1.4.2 Objetivos Específicos

- a. Recomendar mediante estándares aceptados internacionalmente, las prácticas o metodologías de utilidad para la optimización del riesgo y mejora continua, como parte de la entrega de valor en TI.
- b. Identificar y definir la situación actual de una empresa piloto, considerando las necesidades específicas, problemática, riesgos y la resolución presente en caso de existir.
- c. Desarrollar un modelo sistemático aplicando Inteligencia Artificial en la detección, pronóstico, valoración, medición y recomendación ante eventos de fallo y posibles mejoras en la prestación de servicios de Tecnologías de Información; demostrando su utilidad práctica en la toma de decisiones.

1.5 Justificación

El acelerado crecimiento tecnológico en las últimas décadas producto de la relación simbiótica entre las constantes demandas y necesidades de las empresas en conjunto con los avances en tecnologías y lo que estas pueden ofrecer; ha traído consigo una fuerte interacción en donde las estrategias de muchas empresas no solo se sirven de TI como apoyo, sino como base o pilar de dicha estrategia.

Esta dependencia genera a su vez un posible riesgo ya que, como todo servicio, TI puede fallar.

Lo que da peso a esta investigación es el hecho que, a diferencia de otros rubros, un fallo en tecnología puede ser fatal para una empresa, tal como se muestra en la sección de antecedentes del problema, donde errores en TI han traído grandes repercusiones negativas, tanto financiera como en su credibilidad de las que incluso algunas empresas no logran recuperarse.

Lo que es más, algunas empresas aun hoy, desconocen el costo que implicaría un posible fallo en sus servicios de TI.

De esto surge la criticidad en la detección de fallos y en la planeación de alternativas o soluciones que permitan mantener los servicios a flote dentro de lo posible, basados en una metodología de priorización y valoración; las recomendaciones internacionalmente aceptadas, así como el uso de herramientas que apoyen esta tarea.

CAPITULO II - MARCO TEÓRICO

El presente capítulo expone las bases teóricas de la investigación, considerando los estándares internacionalmente aceptados; así como la situación presente en la empresa piloto donde se desarrollará el sistema.

2.1 Generación de Valor

La existencia de Tecnologías de Información (TI) en las empresas se resume a un propósito genérico, la creación de Valor.

Dicho de otra forma y ligado a la estrategia que esta implementa, TI se convierte en un pilar o columna que soporta las operaciones de la empresa, de la cual la organización espera obtener un beneficio. Este “Valor” no necesariamente es un recurso monetario, en muchas ocasiones el valor que se espera obtener de TI puede ser distinto, por ejemplo: El satisfacer una necesidad de comunicación, obtener visibilidad que de soporte a las decisiones, agilizar procesos, reducir costos, optimizar operaciones; adicionalmente existen empresas sin ánimos de lucro que, sin embargo, siempre esperan obtener un retorno de TI, como ser la propia capacidad de operar, disponibilidad de información, transparencia de operaciones en un gobierno, recursos de apoyo al aprendizaje etcétera..

Como se observa, son muchas las formas en que las tecnologías de la información generan valor para el negocio, algunas de forma económica, directa o indirectamente; y otras con otros tipos de servicios de los que la empresa se sirve. Ahora bien, el valor que TI genera y su percepción dentro de la organización son términos ambiguos que dependen de muchos factores externos al propio TI.

Como tal, existen múltiples recomendaciones y fuentes que pueden apoyar la tarea de dirigir TI hacia un enfoque de alineación – e Integración – con la empresa, sin embargo en el presente trabajo se hará enfoque en uno de los modelos estandarizados más comúnmente aceptados y que ha demostrado su valor a lo largo del tiempo, esto es COBIT.

2.2 COBIT

COBIT (Control Objectives for Information and related Technology) es un marco de trabajo que permite que las tecnologías de la información y relacionadas se gobiernen y administren de una manera holística a nivel de toda la Organización, incluyendo el alcance completo de todas las áreas de responsabilidad funcionales y de negocios, considerando los intereses relacionados con la TI de las partes interesadas internas y externas.

Dicho de otra forma, COBIT apoya a las organizaciones en la creación de un valor óptimo a partir de la TI, manteniendo un equilibrio entre la realización de beneficios y la optimización de los niveles de riesgo y utilización de los recursos. (ISACA, 2018)

A fin de que exista valor perceptible dentro de TI, este debe darse en respuesta a un interés de la o las partes interesadas de la empresa, las cuales transmiten la necesidad a satisfacer en un ámbito descendente.

Como es evidente, difícilmente una empresa tendrá una única necesidad o interés dentro de su existencia; los intereses o impulsores son muchos y muy distintos, algunos más fáciles que otros, algunos más urgentes que otros, y algunos más importantes que otros.

Esto suma a la ya compleja tarea de dirigir TI; evaluar y determinar cómo se deben establecer y satisfacer las múltiples necesidades de una empresa a fin de, tal como se indicó anteriormente, lograr un equilibrio óptimo en la generación de valor en completa alineación con la estrategia del negocio. Esto es Gobierno de TI.

Para este fin, COBIT sugiere un enfoque de tres puntos que determinan el objetivo del Gobierno de TI, la generación de Valor. Los enfoques consisten en la realización de los Beneficios, La optimización de los Riesgos y la optimización de los Recursos, tal como se muestran en la Ilustración 1.

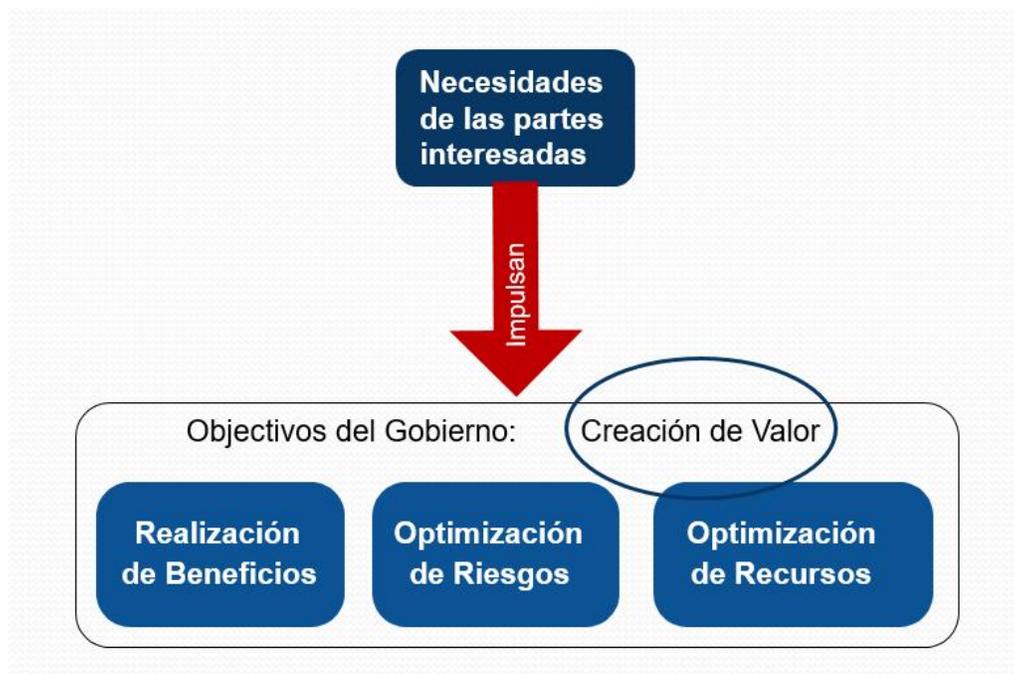


Figura 1. Componentes de la creación de Valor.

(Fuente COBIT 5)

Los tres componentes son necesarios para generar un equilibrio entre el costo, garantía y prestación del valor otorgado; existe abundante documentación orientada al Gobierno de TI mediante COBIT, sin embargo, para fines del presente trabajo se reduce el enfoque al componente de optimización del riesgo.

2.3 Optimización del Riesgo

La palabra Riesgo tiene origen en el árabe clásico *rizq* que significa “lo que depara la providencia”. El término en nuestro lenguaje hace referencia a la “proximidad o contingencia de un posible daño.” (Real Academia Española, 2018).

En términos del riesgo tecnológico, existe consenso generalizado en definirlo como la posibilidad de pérdidas derivadas de un evento relacionado con el acceso o uso de la tecnología, que afecta el desarrollo de los procesos del negocio y la gestión de riesgos de la organización, al comprometer o degradar las dimensiones críticas de la información, esto es Disponibilidad, Integridad y Confidencialidad.

COBIT 5 for Risk define el riesgo de TI como un riesgo para el negocio, específicamente el riesgo para el negocio asociado con el uso, propiedad, operación, involucramiento, influencia y adopción de TI dentro de una empresa. (Rigante, 2014)

Existe un punto interesante en el motivo por el que COBIT decide llamar su apartado como Optimización en lugar de Gestión o Mitigación del riesgo. Esto demuestra una evolución en el enfoque relacionado a los riesgos, donde anteriormente presentaba un enfoque hacia auditoria más que al dinamismo del negocio.

Considerando esto, COBIT 5 interpreta la Optimización del Riesgo como la “Garantía que los riesgos para el negocio relacionados con TI no exceden el nivel aceptable”.

En el manejo del riesgo en TI de forma generalizada, existe una dualidad de resultados:

- Positivo, donde se genera o preserva el valor.
- Negativo, donde se destruye o no se entrega el valor esperado.

En el primer escenario, la empresa puede valerse de TI para obtener nuevas oportunidades de negocio, ampliar las ya existentes o bien generar ventajas competitivas sostenibles. En cambio, el segundo escenario contiene la posibilidad de pérdida de oportunidades, la no entrega de valor o incluso la destrucción de este.

La optimización del riesgo en Tecnologías de Información supone un factor crítico en la obtención y generación de valor.

Esto resulta aún más evidente cuando se aborda un punto de vista más específico, considerando TI como el proveedor de múltiples servicios. Servicios cuya entrega y disponibilidad crece a medida que la aceptación de TI aumenta dentro de la empresa.

Es por eso que COBIT sugiere un elemento que garantice la entrega del servicio a fin de reducir los riesgos incurridos ante la posible incertidumbre. Función que es cubierta ampliamente por un estándar perfectamente compatible con COBIT, esto es ITIL.

2.4 ITIL

ITIL (Information Technology Infrastructure Library) es, como su nombre lo indica, una Biblioteca de Infraestructura TI, más concretamente un marco de referencia que describe un conjunto de mejores prácticas y recomendaciones para los servicios de Tecnologías de Información.

El enfoque de ITIL es el aseguramiento del ciclo de vida de los servicios de TI, de modo que estos sean provistos cumpliendo los fines para los que fueron creados, según la operativa aceptable, tolerantes o con respuesta a los cambios y en un proceso de mejoramiento de su calidad. Claro está, en alineación con la estrategia del negocio.

Esto resume los cinco libros que componen la biblioteca ITIL:

- Estrategia de servicios.
- Diseño de servicios
- Transición de servicios.
- Operación de servicios.
- Mejora continua de servicios.



Figura 2. Ciclo de Vida de los Servicios en TI.

(Fuente ITIL v3)

Enfocando en la sección de Operativa del servicio, cuyo principal objetivo es garantizar la entrega de los servicios pactados según el funcionamiento normal; es provechoso mencionar que ITIL sugiere en esta etapa la existencia de procesos de Gestión de Incidentes y Gestión de problemas. Resumiendo, el primero se especializa en el restablecimiento de los servicios frente a fallos en el menor tiempo posible, asignando urgencia, impacto y prioridades.

Para el caso del segundo, este toma como entradas los distintos eventos generados, esto es incidentes, logs, registros de sistemas, alertas, etcétera; a fin de determinar una correlación e identificar lo que en ITIL se llama un problema, reuniendo a los expertos para identificar posibles soluciones (sean estas definitivas o no, aplicables en su totalidad o no), y así prevenir futuros incidentes o fallos.

De esto es clara la importancia que ITIL otorga a la identificación y resolución ante fallos e interrupciones del servicio ante lo fácil que la confianza en TI puede decaer frente a inconvenientes no gestionados de la mejor manera.

2.5 Actualidad en empresa piloto

Si bien existen muchos puntos de valor que se pueden obtener de ITIL, COBIT y otros marcos de referencia; para fines de la presente investigación se toma la importancia que estos estándares sugieren de forma unísona ante su criticidad, la garantía en la entrega del servicio. Considerando el valor estratégico de TI, su aporte al negocio, la percepción del servicio provisto a sus clientes (internos o externos) y la generación de valor en alineación con la estrategia empresarial; la consideración del riesgo de fallar en TI se vuelve una necesidad cada vez de mayor peso.

Para fines demostrativos, el presente estudio realizará la aplicación del sistema de soporte a la detección, evaluación, y gestión de riesgo de fallo en la operación de TI dentro de la empresa TIGO, con un alcance a facilitar la entrega del servicio de información a nivel del departamento de Data Warehouse.

En la actualidad la empresa cuenta con múltiples alertas, logs, eventos y un sistema estructurado que permite obtener visibilidad de fallos en procesos analíticos de información. En su mayoría estas alertas son posteriores a que el evento de fallo se ha generado, de modo que el personal puede tomar acciones en un lapso de tiempo menor.

En la ilustración 3 se muestra un ejemplo de una alerta generada ante el fallo de un proceso lógico que, en cadena, generaría un impacto de retraso en una serie de procesos, reportes y cubos que se determina allí mismo.

domingo, 07:44

Proceso fallido, posible retraso:

Nombre de proceso X

92 Procesos

4 Cubos OLAP

23 Cubos Excel

Figura 3. Alerta enviada vía SMS por evento de fallo en un proceso determinado.

(Fuente propia).

Solamente en los sistemas de entrega de información del departamento, convergen más de 4000 procesos de datos distintos de los que se sirve de forma transversal la organización casi en su totalidad.

Sumado a esto y en comparación con el mismo departamento en otros países de la zona (El Salvador, Guatemala, Colombia, etc.). Honduras cuenta con los tiempos de entrega más agresivos, situación que ha requerido la constante mejora y optimización de recursos hasta llegar a satisfacer la necesidad presente en un nivel aceptable.

El departamento está en proceso de la aplicación de ciertos estándares o recomendaciones que ayudaran a garantizar la entrega del servicio en tiempo y calidad, por lo que se encuentra en un rediseño de algunas políticas, procesos y roles.

Se cuenta también con la interacción con otros departamentos especializados en escalamiento (NOC²), gestión de incidentes (Mesa de Servicio³), gestión de problemas entre otros; en algunos casos aun en etapas tempranas.

En general, se cuenta con múltiples variables y registros de eventos que permiten determinar los fallos. Hoy día el aprovechamiento de estos se da desde un punto de vista post evento en su gran mayoría, no se descarta que se ha mejorado notablemente el tiempo de respuesta y resolución de fallos e interrupciones, sumado a un proceso de gestión de problemas aun aprendizaje.

Por otro lado, se ha detectado cierta carencia en herramientas de análisis predictivo, prescriptivo y de inteligencia en general; es decir que permitan predecir un posible escenario de fallo y presentar

² Centro de control de Red. Sitio que da seguimiento y control a las redes de computación y telecomunicaciones.

³ Conjunto de recursos tecnológicos y humanos dedicados a la ayuda y soporte a los usuarios.

recomendaciones para prevenirlo o reducir su impacto. Situación que pretende mejorar la aplicación del presente trabajo.

Tabla 1. Análisis FODA de empresa Piloto.

<p>Fortalezas:</p> <ul style="list-style-type: none"> • Se cuenta con un sistema estructurado y automatizado de control de procesos y flujos de información. • Autonomía y fiabilidad del sistema, poca interacción humana y facilita la misma en caso de requerirse. • Acceso a logs e información histórica de eventos e incidentes. • Credibilidad y confianza en los entregables del sistema y del departamento. • Centralización de datos y aseguramiento de la calidad mediante métricas e indicadores. 	<p>Oportunidades:</p> <ul style="list-style-type: none"> • Mejora en detección de fallos o eventos críticos predecibles, pero hoy no explotado. • Gran cantidad de procesos, algunos de ellos candidatos para mejoras (ya sea utilizando nuevas técnicas, metodologías u otros). • Mejor aprovechamiento del espacio usando técnicas de compresión, mejora en tiempos de respuesta utilizando técnicas optimizadas para la consulta.
<p>Debilidades:</p> <ul style="list-style-type: none"> • Poca documentación disponible y de acceso limitado al personal crítico. • Alta complejidad del sistema y necesidad de conocimientos avanzados. • Poca visibilidad del nivel de eficiencia en desarrollos y procesos, tanto nuevos como antiguos. • Tiempo de respuesta ante incidentes variable, dependiendo en gran medida del juicio experto. 	<p>Amenazas:</p> <ul style="list-style-type: none"> • Vulnerabilidades y afectaciones derivadas de un fallo en la infraestructura (Red, Servidores, Bases de Datos, etc.). • Fallo en uno o múltiples procesos independientes, causados por cambios no gestionados, errores o ineficiencias. • Afectación en el rendimiento con repercusión en tiempos, con causas difíciles de identificar. • Alta posibilidad de afectación general en caso de fallo, derivado de una elevada interdependencia de múltiples orígenes de información.

2.6 Técnicas de Análisis de Datos

Son muchas las técnicas que permiten agrupar, categorizar, relacionar y en general hacer uso inteligente de los datos. No todas ellas requieren de una computadora, existen múltiples técnicas puramente matemáticas que permitirán obtener un punto de partida para la toma de decisiones. Ahora bien, esta situación se complica cuando hablamos de cantidades abrumadoras de información, rondando los miles o millones o incluso más muestras; es allí donde los sistemas computacionales tienen un papel clave en dar soporte a esta tarea.

Para fines informativos, se exponen a continuación algunas de las técnicas de análisis de datos más comunes, pudiendo existir más ya sea en el presente o futuro.

2.6.1 Análisis de Correlaciones

Este tipo de análisis estadístico se utiliza para intentar establecer una correlación entre dos o más variables, indicando el grado de dicha correlación.

2.6.2 Análisis de Regresión

Otro tipo de análisis estadístico que permite investigar la relación entre múltiples variables ya sea esto mediante técnicas lineales o polinomiales a fin de identificar el tipo de comportamiento que guardan las variables dependientes e independientes.

2.6.3 Minería de Datos

Actualmente muy relacionado con Big Data, este tipo de análisis se realiza sobre grandes volúmenes de información para detectar patrones o relaciones sobre los datos que permitan explicar un escenario determinado.

2.6.4 Clusterización de Datos

Con esta técnica obtenida de la minería de datos, se intenta realizar una segregación de los elementos, basados en cualidades comunes entre ellos; a fin de generar grupos o clusters que los agrupen.

2.6.5 Algoritmos Matemáticos

Existen múltiples algoritmos matemáticos especializados que han sido diseñados para tareas específicas, por ejemplo, la Simulación de Monte Carlo, Método de Newton Raphson, Método de Bairstow y en general una serie de trabajos matemáticos que son aplicables dentro de los sistemas de cómputo.

Para fines del presente trabajo, se seleccionaron los métodos de Redes Neuronales (Parte de la Inteligencia Artificial) ya que a pesar de ser uno de los métodos más complejos, supone un gran avance al proporcionar a las máquinas la inteligencia humana de la que carecían. Es mediante estas técnicas que ha sido posible realizar análisis que por métodos tradicionales serían sumamente complejos para una máquina. Por mencionar algunos, el reconocimiento de voz, de imágenes, de sentimientos, de manuscritos y demás.

Un factor adicional en la selección de este método es la capacidad de incorporar en el análisis el propio aprendizaje. A diferencia de otros métodos, una red neuronal puede ser capaz de recopilar información nueva y agregarla a su propio conocimiento para mejorar así los resultados de sus análisis. Estos temas se profundizarán más adelante.

Además del aprendizaje mencionado, una de las razones que motiva a la selección de esta metodología (Red Neuronal) es su retorno en precisión y rendimiento conforme aumentan los volúmenes de datos ya que con el poder de computo actual, se supera el punto de quiebre entre las metodologías tradicionales antes mencionadas y esta. Tal como se muestra en la siguiente imagen.

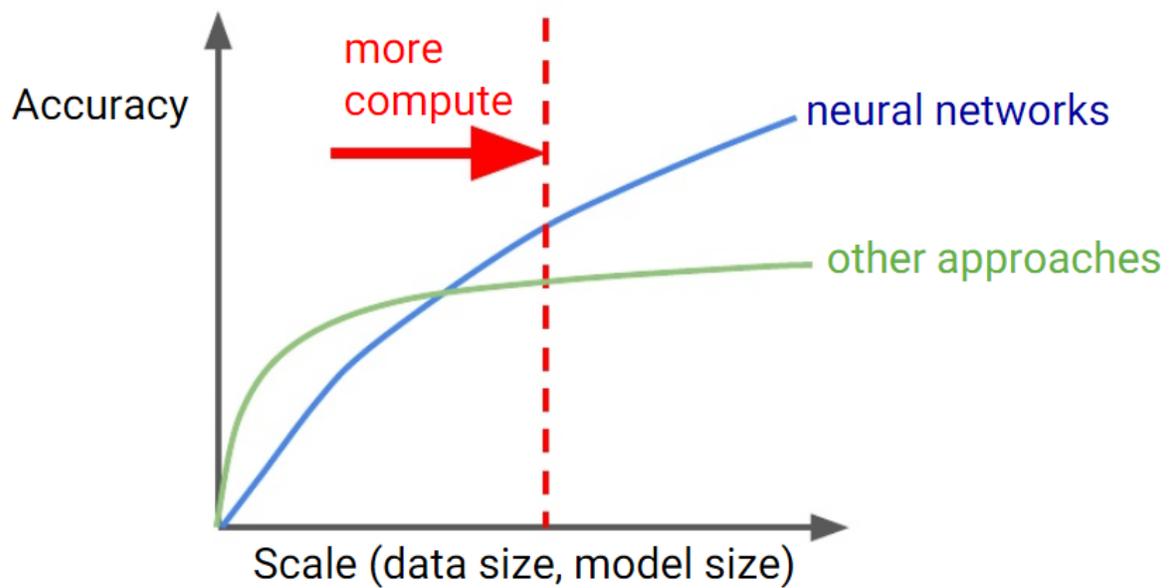


Figura 4. Comparativo de precisión entre Redes Neuronales a partir de los años 90.

(Dean, 2019)

2.7 Inteligencia Artificial

El cerebro humano es, en pocas palabras, una maravilla.

Mediante el, somos capaces de realizar tareas sumamente complejas de forma intuitiva y en fracciones de segundo sin siquiera percatarnos de la complejidad que esta conlleva. No esta demás decir que tenemos una supercomputadora entrenada que se adapta perfectamente al entorno y es capaz de interpretar lo que observa y hacer sentido de ello.

Una tarea tan sencilla como leer un texto manuscrito, si es estudiada, nos damos cuenta de la enorme dificultad que llevaría entrenar un sistema para realizar esta función. Bajo los métodos tradicionales de programación nos encontraríamos con una cantidad abrumadora de variables y excepciones que harían prácticamente imposible resolver algo que para nuestro cerebro es tan simple. Es por esto que surge la necesidad de buscar un acercamiento diferente.

El termino Inteligencia Artificial fue acuñado primeramente por John McCarthy en 1956 para referirse a “La ciencia e ingeniería de hacer maquinas inteligentes” (Science Daily, 2019).

Este concepto actualmente se entiende como una rama de la ciencia computacional que enfoca sus esfuerzos en la creación de máquinas capaces de accionar de forma similar a lo que lo haría la mente humana.

Entre algunos de los puntos de estudio dentro de la Inteligencia Artificial se encuentran el razonamiento, aprendizaje, planeación, percepción, resolución de problemas, administración del conocimiento, y habilidad de interactuar entre otros.

Como se indicó, parte de la inteligencia artificial consiste en el aprendizaje, término que nos lleva al Machine Learning o aprendizaje de máquina. Este término consiste en una parte de la inteligencia artificial enfocada en satisfacer la necesidad de la máquina de aprender a hacer las

cosas o cómo responder a distintas situaciones sin ser programada previa y específicamente para ello.

Uno de los avances que más ha impulsado el aprendizaje de máquinas es la aplicación de las llamadas redes neuronales; las cuales buscan emular la forma en que el cerebro humano trabaja y procesa la información.

2.8 Redes neuronales

El componente principal de nuestro sistema nervioso y el cerebro en particular es llamado neurona. Consiste en una célula estimulable de forma eléctrica que recibe, procesa y transmite información mediante una comunicación electroquímica a través de lo que se denomina Sinapsis. A diferencia del resto de las células de nuestro cuerpo, las neuronas no están hechas para dividirse, morir o ser reemplazadas.

Un cerebro humano tiene en promedio 100 billones de neuronas interconectadas en una compleja red donde cada una de forma individual puede conectarse con incluso 10,000 neuronas distintas, utilizando un número tan grande como 1,000 trillones de conexiones sinápticas. (Human Memory, 2019). Si intentáramos hacer un comparativo en forma de un procesador de computadora, necesitaríamos uno de aproximadamente uno de un trillón de bits por segundo.

La tecnología actual aun no es capaz de replicar semejante complejidad que corresponde la estructura de un cerebro humano; sin embargo, el aprendizaje de cómo funciona este nos ha permitido experimentar con una red artificial que se asemeja a las redes neuronales reales, en una escala reducida pero que funciona bajo los mismos principios.

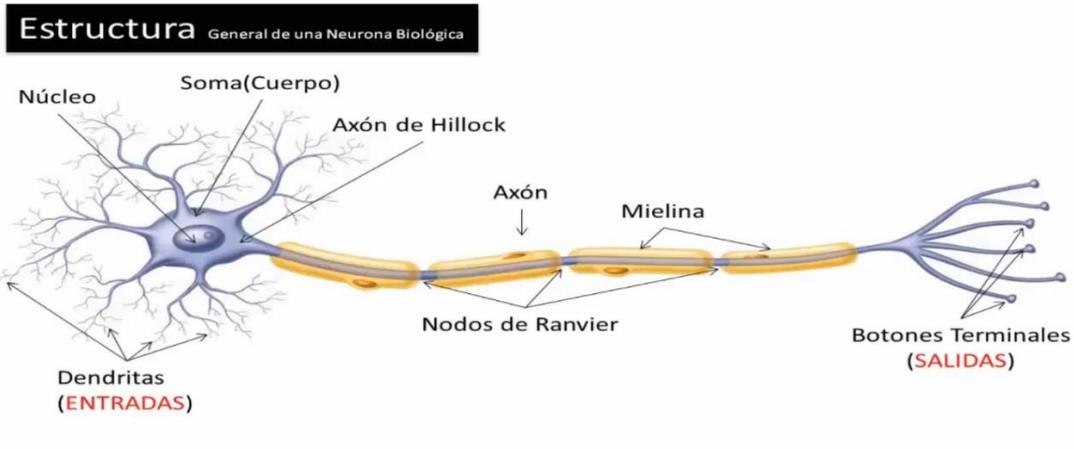


Figura 5. Estructura general de una Neurona Biológica.
(Gómez, 2015)

Una neurona recibe (a través de su terminal llamada Dendrita), una señal eléctrica también llamada Potencial de Acción, la salida de la misma se envía a través del Axón utilizando para ello el recubrimiento de mielina para acelerar la transferencia y los denominados nodos de Ranvier como impulsores de dicha señal.

Si intentáramos explicar esto de forma matemática, tendríamos algo como lo que se muestra en la figura siguiente:

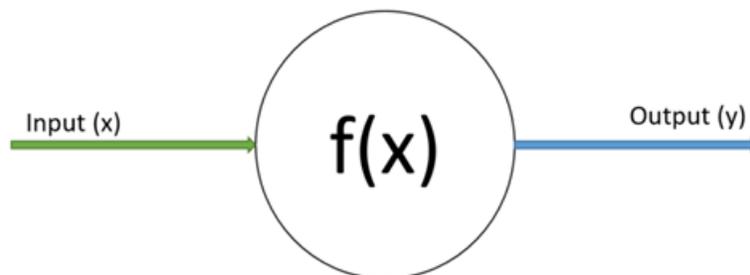


Figura 6. Función matemática de procesamiento.
(Battini, 2018)

Las dendritas de la neurona biológica se traducen a nuestra entrada X , el núcleo a la función $f(x)$ y por ende el axón resulta en la salida Y .

Como se mencionó antes, una neurona no está aislada ni recibe únicamente una entrada, sino muchas, posteriormente aplica una operación matemática que genera una salida. La figura abajo mostrada representa este funcionamiento.

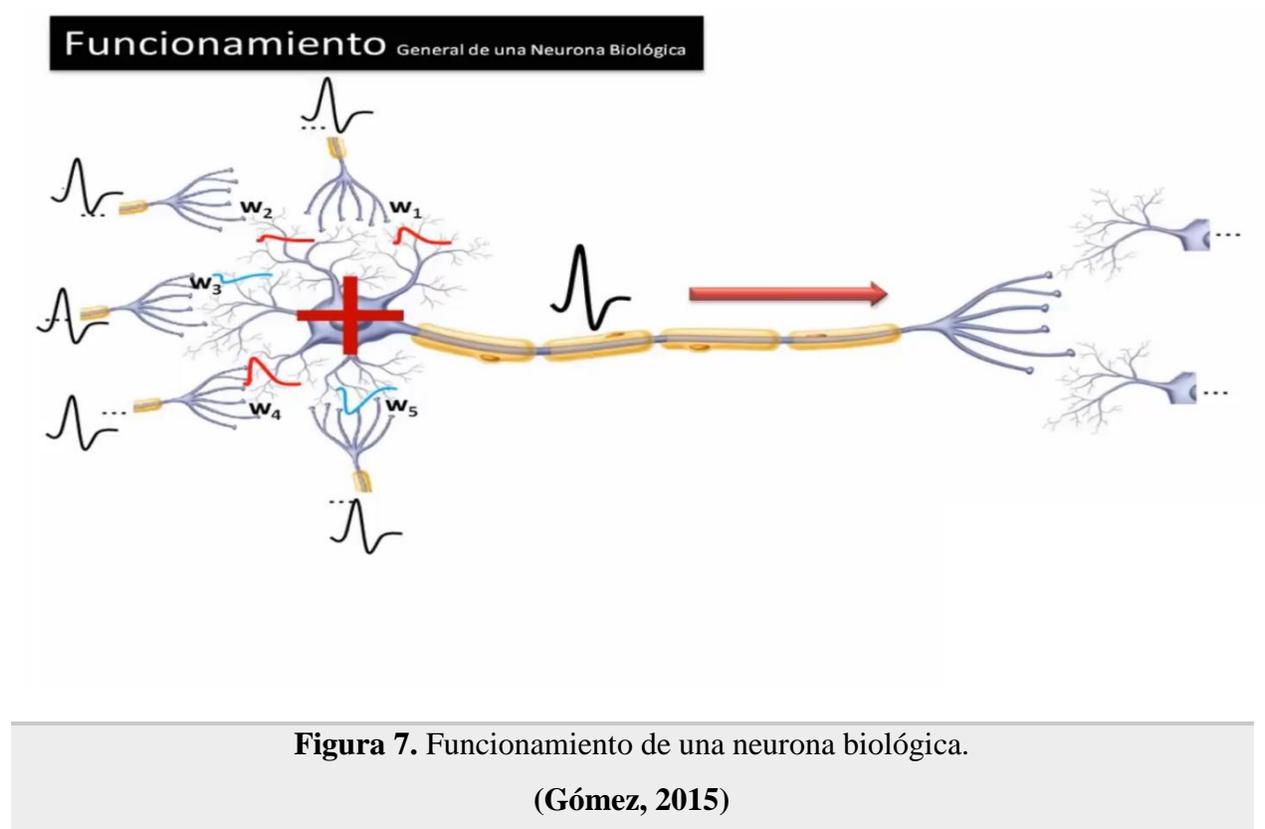


Figura 7. Funcionamiento de una neurona biológica.

(Gómez, 2015)

Los diferentes pulsos de activación que se reciben en la entrada llegan a ella a través de la conexión sináptica con una fuerza controlada directamente por la intensidad en que la sinapsis se conecta, es decir que este impulso (p) es regulado mediante el peso sináptico (w) más una polarización propia de la neurona (b).

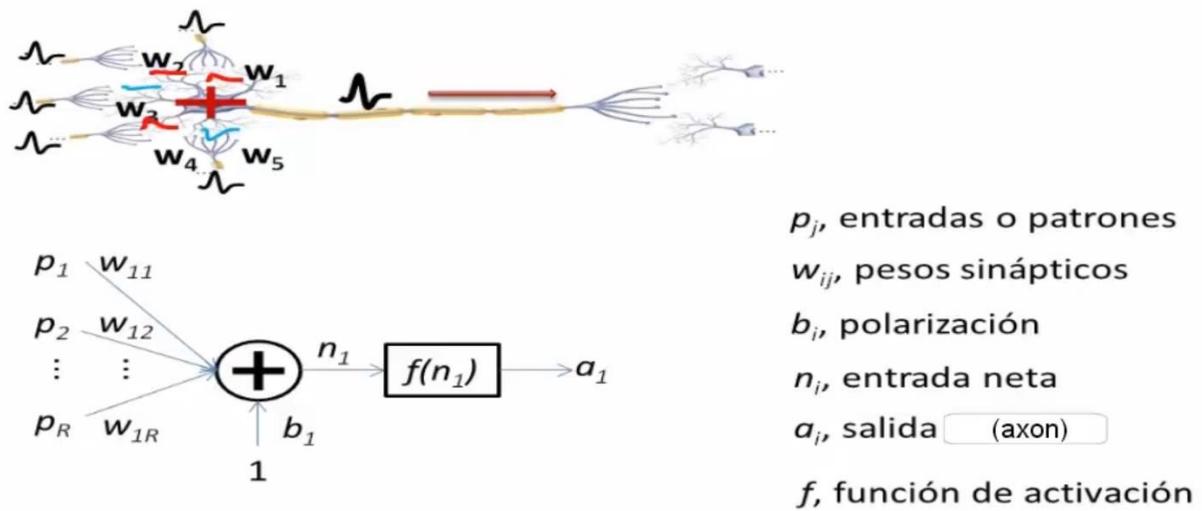


Figura 8. Representación matemática del funcionamiento de una neurona.
 (Gómez, 2015)

De esta forma se tiene que, las múltiples entradas p son multiplicadas por los múltiples pesos w más una desviación b para dar como resultado un valor n , que es el que se envía a la función de cómputo dentro de la neurona.

Si intentamos representar esto de forma matemática tendríamos que:

$$n_1 = p_1 \cdot w_{11} + p_2 \cdot w_{12} + \dots + p_R \cdot w_{1R} + b$$

Expresado de otra forma:

$$n_1 = \sum_1^R (p \cdot w) + b$$

Por lo que el resultado (a) sería la función matemática de dicho resultado:

$$a_1 = f(n_1)$$

Si se apilan múltiples neuronas artificiales se tendrá una red más compleja como la que se muestra a continuación, conformada por la capa de entrada, capa de salida y una o más capas ocultas.

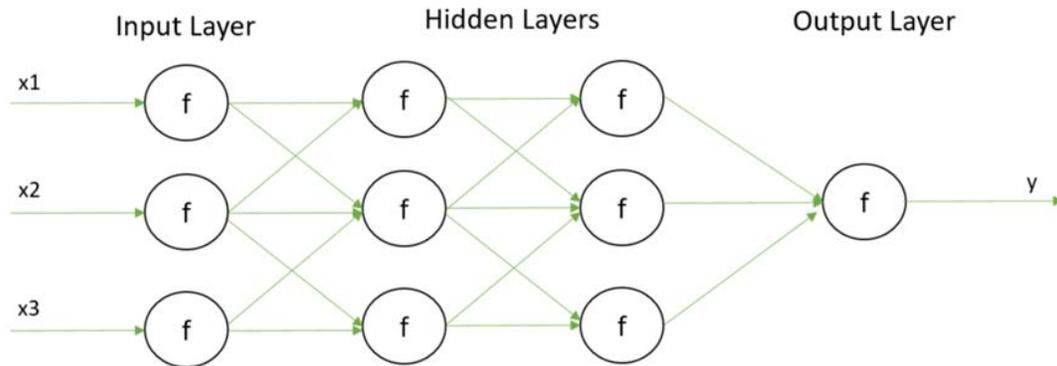


Figura 9. Red de neuronas interconectadas.

(Battini, 2018)

2.9 Aprendizaje y entrenamiento

Podría decirse en pocas palabras que las redes neuronales intentan establecer relaciones arbitrarias entre los datos de entrada y los datos de salida. Esto nos lleva a la necesidad que tiene la red de encontrar dicha relación y para ello se usa lo que se denomina entrenamiento o aprendizaje.

Básicamente hay tres formas en que se puede entrenar una red:

- Entrenamiento Supervisado.
- Entrenamiento No Supervisado.
- Entrenamiento por Reforzamiento.

El entrenamiento asistido o supervisado, que es el que servirá de base en este trabajo, proporciona a la red los datos de entrada y de salida esperados, de forma que ella pueda generalizar la información y obtener la relación matemática entre las variables.

En el entrenamiento no supervisado, no se proporcionan los datos de salida sino únicamente los datos de entrada; en cuyo caso la red intentara establecer patrones que permitan clasificar estos datos.

El aprendizaje por reforzamiento utiliza una interacción de recompensa y castigo sobre resultados obtenidos en la red. De forma similar que un bebe aprende ya sea por premios o castigos. La red funciona bajo una serie de criterios que determinan cuando se debe reforzar o penalizar ciertas acciones.

2.10 Aprendizaje profundo (Deep Learning)

El entrenamiento de la red de forma que se encuentren esas incógnitas que corresponden a los pesos sinápticos (w_{ij}) y la desviación o bias (b_j) dentro de múltiples capas abstraídas dentro del modelo neural; es lo que en esencia representa lo llamado aprendizaje profundo o Deep Learning.

Uno de los algoritmos más utilizados en el entrenamiento supervisado es el de propagación hacia atrás (*Backpropagation*).

Podemos ver este tipo de entrenamiento como un proceso iterativo de “ida y regreso” entre las distintas capas en la red neuronal. (Torres, 2018)

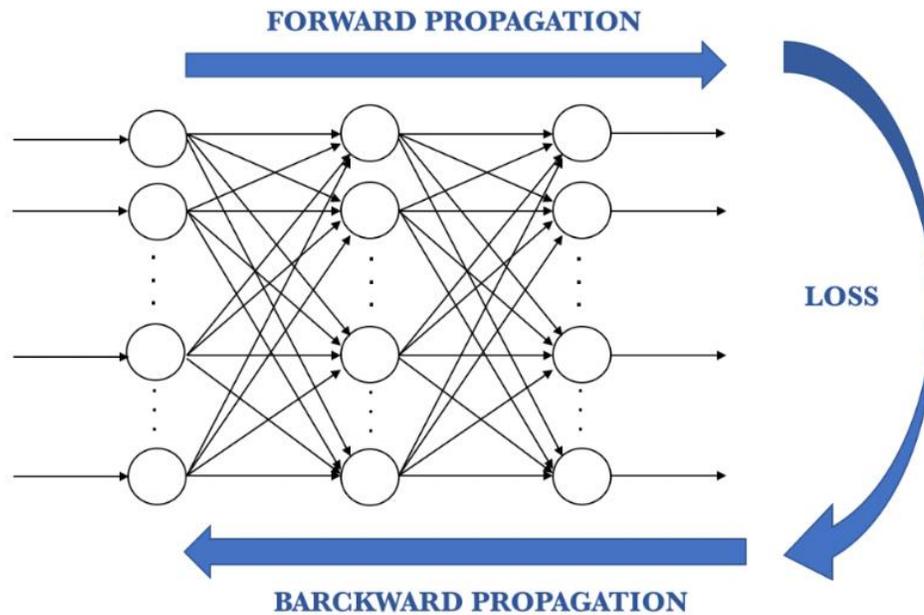


Figura 10. Representación de Propagación hacia atrás.
(Torres, 2018)

La primera fase del algoritmo se denomina propagación hacia adelante (Forward Propagation). Se estimula la red mediante una entrada inicial, la cual se propaga de izquierda a derecha (desde las capas cercanas a la entrada, hasta las capas cercanas a la salida) a través de cada una de las capas internas y por ende las neuronas contenidas en dicha capa; a fin de obtener una respuesta.

En esta fase las neuronas de la red contienen pesos y biasses aleatorios, por lo que muy seguramente los resultados contendrán un *Error*.

Seguidamente se utiliza una función de perdida a fin de determinar qué tan cerca o lejos de la respuesta real está la respuesta generada por la red, recordando que en el aprendizaje asistido suministramos tanto la entrada como la salida real.

Una vez que se ha determinado el nivel de Error, este necesita ser enviado a las capas posteriores de modo que estas puedan hacer los ajustes necesarios. Este tipo de transferencia se realiza

partiendo de la capa final hacia las iniciales, esto es de derecha o izquierda o en otras palabras, hacia atrás. De allí su nombre *backpropagation*.

Cada una de las capas ocultas recibe la retroalimentación del error, sin embargo, solo de forma parcial, estando ésta ligada a la propia contribución de cada neurona a la respuesta generada.

Una vez transmitida la información del error hacia las capas iniciales, es posible hacer la actualización de los pesos sinápticos a fin de acercar el error a cero. Existen varias técnicas, siendo una de las más populares el Descenso de Gradiente, llamado también de primer orden.

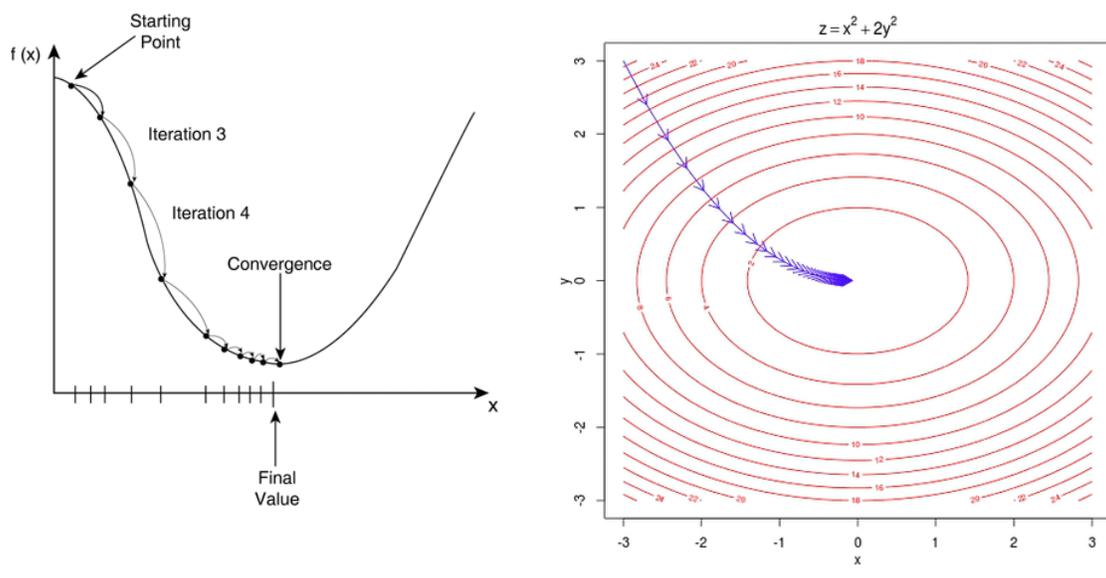


Figura 11. Ejemplo de Descenso de Gradiente
(Caparrini, 2018)

Si bien este algoritmo es recomendado en redes neuronales de gran tamaño, tiene deficiencias.

Aunque es muy sencillo, este algoritmo tiene el gran inconveniente de que, para funciones de error con estructuras con valles largos y estrechos, requiere muchas iteraciones. Se debe a que, aunque la dirección elegida es en la que la función de error disminuye más rápidamente, esto no significa que necesariamente produzca la convergencia más rápida. (Caparrini, 2018)

2.11 Aplicación de Redes Neuronales

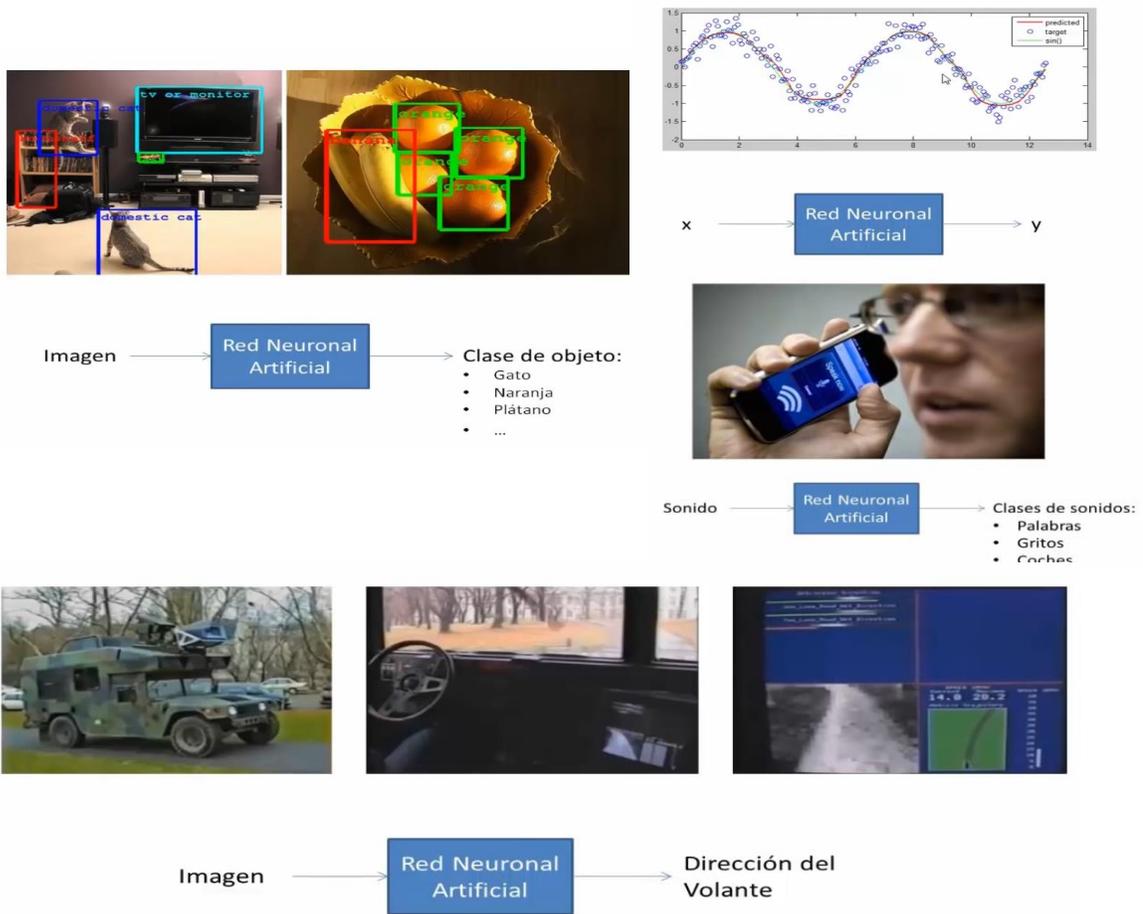


Figura 12. Aplicaciones de Redes Neuronales.

(Gómez, 2015)

Detección de imágenes (Arriba izquierda), Aproximación de funciones y reconocimiento del habla (Arriba derecha), Control automático de vehículos (Abajo).

Para propósitos del presente trabajo este se apunta a la detección de entornos de fallo para reducir la probabilidad del riesgo o bien minimizar el impacto dentro de la cadena de entrega de información.

Adicionalmente se entrenará de forma simultánea un modelo para la detección de ineficiencias en los procesos. De modo que permita la optimización de los recursos y el esfuerzo necesario para llevar a cabo la mejora continua.

CAPITULO III - METODOLOGÍA DE LA INVESTIGACIÓN

En este capítulo se detalla la metodología de investigación a aplicar en el estudio; en conjunto con las herramientas, técnicas, enfoque y fuentes de información tomadas como base para el mismo.

3.1 Enfoque de la investigación

Considerando que el presente trabajo tiene como fin la aplicación práctica de un sistema de apoyo a la toma de decisiones basado en análisis estadísticos y matemáticos de múltiples variables; el método de investigación cualitativo en conjunto con las distintas herramientas relacionadas será de utilidad para demostrar las múltiples hipótesis de correlación de variables, peso y relevancia en los resultados finales.

Como alimentación del modelo se cuenta con una serie de variables con supuesta relación en la calidad, tiempo y cumplimiento de los servicios de Tecnologías de Información. Gran parte de la demostración o descarte de estas hipótesis se hará a través de análisis numéricos y estadísticos mediante los cuales se pretende afinar el modelo de forma que se reduzca el margen de error y se provea una herramienta con un nivel de precisión aceptable.

Esto hace el método en cuestión apropiado para el presente estudio, pues tal como expresa Sampieri (2014) en su libro: “El Enfoque cuantitativo usa la recolección de datos para probar hipótesis, con base en la medición numérica y el análisis estadístico, para establecer patrones de comportamiento y probar teorías.” (p. 46)

3.2 Alcance de la investigación

El estudio demostrará y dará respuesta a la pregunta de como la inteligencia computacional puede dar soporte a la toma de decisiones que conlleva la gestión y optimización del riesgo en TI; considerando la gran cantidad de datos que son generados en un sistema determinado de la empresa, permitiendo obtener a partir de ellos la información y posteriormente el conocimiento de utilidad en la garantía de prestación en los servicios de TI.

Para delimitar y enfocar el estudio a los objetivos de la investigación, resulta apropiado recalcar las preguntas de investigación.

Las preguntas harán énfasis en los objetivos que el estudio intentara dar respuesta.

Tabla 2. Pregunta de Investigación.

Pregunta de Investigación	Objetivo General
¿Cómo pueden las tecnologías de inteligencia artificial apoyar en la gestión, optimización, mejora continua y aseguramiento de los servicios de Tecnologías de Información dentro de un entorno competitivo y cambiante?	Demostrar la aplicación de tecnologías de inteligencia artificial como apoyo a la gestión, toma de decisiones y optimización en los procesos de Tecnologías de Información.

Adicionalmente existen otras interrogaciones que permitirán encaminar la investigación hacia los objetivos planteados, como ser:

Tabla 3. Preguntas de investigación específicas.

Preguntas de Investigación	Objetivos Específicos
¿Cómo la gestión y optimización del riesgo en TI ha apoyado la estrategia empresarial en empresas alrededor del mundo?	Recomendar mediante estándares aceptados internacionalmente, las prácticas o metodologías de utilidad para la optimización del riesgo, como parte de la entrega de valor en TI.
¿Qué variables pueden encontrarse en un ambiente real de experimentación, cuál es su relación y como pueden convertirse en conocimiento útil para la empresa?	Identificar y definir la situación actual de una empresa piloto, considerando las necesidades específicas y la resolución presente a las mismas en caso de existir.
¿Cómo pueden las metodologías de inteligencia computacional apoyar la captura, evaluación, ponderación, interpretación y prescripción de información?	Desarrollar un modelo sistemático de detección, pronóstico, valoración, medición y recomendación ante eventos de fallo o posibles mejoras en la prestación de servicios de Tecnologías de Información; demostrando su utilidad práctica en la toma de decisiones.

Hipótesis

A continuación, se listan las hipótesis iniciales que el estudio intentara probar.

Tabla 4. Hipótesis.

	Descripción
Hipótesis	H1 Los niveles de detección, evaluación y control de riesgos tempranos tienen una relación directa con la calidad de los servicios de TI entregados.
	H2 Existen múltiples variables que permiten determinar y/o predecir un entorno de fallo, retraso o afectación en los servicios de TI.
	H3 Es posible determinar los niveles de correlación e inferencia entre las variables, por ende, definir un modelo automatizado de aprendizaje que automatice esta correlación y así permita ejercer su procesamiento.
	H4 La aplicación de inteligencia computacional sobre los volúmenes de datos generados en los sistemas tecnológicos apuntados a la detección, predicción y prescripción proporcionara una mejora significativa en el nivel de servicio entregado por TI.

3.3 Técnicas e instrumentos a utilizar

Se listan a continuación los principales instrumentos y herramientas a utilizar en la investigación:

- Análisis numérico y Estadístico.
- Redes neuronales con aprendizaje asistido.
- Hojas de cálculo.

3.4 Fuentes de Información

- Datos y registros de eventos obtenidos de sistemas críticos.
- Registros de eventos de ejecuciones de procesos y resultados.
- Estadísticas de almacenamiento, uso de recursos, código fuente y similares.
- Libros, documentos, investigaciones y trabajos relacionados.

3.5 Definición de Variables

Tabla 5. Variables experimento 1.

Conjunto de Variables #1	Descripción	Indicador	Tipo de Variable
Cantidad de procesos completados	Total de procesos completados con éxito a una hora determinada.	Métrica de Sistema de Calendarización	Numérica
Cantidad de errores	Total de errores detectados (resueltos o no) a una hora determinada.		
Hora actual	Hora de evaluación.		
Hito #1	Flag que determina se ha superado un punto relevante en la cadena de ejecución.		
Hito #2			
Hito #...			
Hito #10			
% completado cadena 1	Evaluación por flujos, determina el % de procesos completados a una hora determinada para un flujo específico de datos.		
% completado cadena 2			
% completado cadena ...			
% completado cadena N			
Flag de proceso critico	Flag que determina si existe un proceso critico en estado fallido.		
Cantidad de sesiones activas	Cantidad de sesiones activas en una hora determinada.	Métrica de Base de Datos	
Cantidad de procesos paralelos en base de datos	Cantidad de Parallel usados a nivel de base de datos.		
Sesiones en bloqueo	Total de sesiones generando bloqueos de tablas.		
Uso de procesador	Utilización de procesadores.	Métricas de Sistema Operativo	
Uso de memoria	Utilización de Memoria.		

Tabla 6. Variables experimento 2.

Conjunto de Variables #2	Descripción	Indicador	Tipo de Variable
Cantidad de Tablas utilizadas	Total de tablas que se utilizan	Métrica de Sistema de Calendarización	Numérica
Cantidad de registros escritos	Total de registros insertados		
Tiempo de ejecución	Total de minutos que tarda en completarse		
Cantidad de líneas de código	Total de líneas de código		
Total aproximado de producto cartesiano	Total aproximado de registros unidos de las tablas fuente		

CAPITULO IV – ANÁLISIS Y RESULTADOS

4.1. Preparación de modelo de pruebas

Como punto de inicio a los análisis, se realiza la preparación de un modelo analítico de redes neuronales utilizando para ello la información recopilada y mostrada en el marco teórico contenido en el Capítulo II.

Se pretende aplicar los distintos algoritmos en forma programable, utilizando para ello las librerías .NET y el lenguaje de C#. Cabe destacar que durante la investigación se consideraron múltiples opciones para aplicar análisis a los datos.

Utilizando las técnicas de programación orientada a objetos, primeramente se define una clase Neurona, la cual encapsula:

- Vector de Entradas
- Vector de Pesos sinápticos
- Variable de Error
- Bias o Desviación
- Cálculo de Salida
- Cálculo de Ajuste de error

Como función de activación para cada neurona, se define la función matemática Sigmoide, la cual se detalla a continuación.

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

Se decide utilizar esta función ya que la misma generara para cada neurona un resultado comprendido entre 0 y 1.

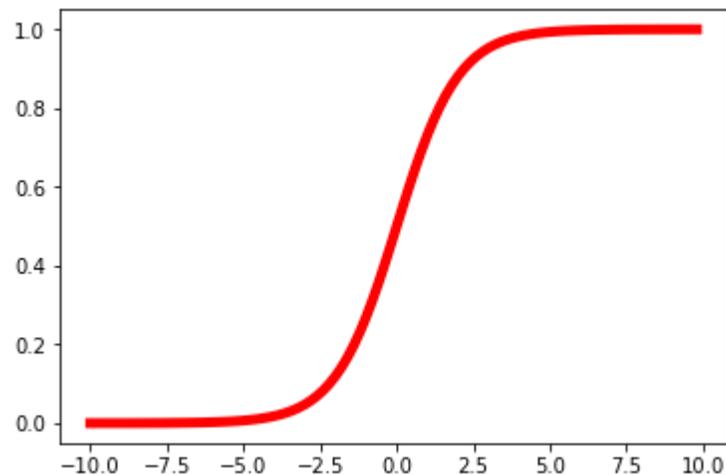


Figura 13. Función Sigmoide
(Kapoor, 2016)

Como se observa, la función tiene una característica forma de S, con dos límites; 0 cuando la variable x se acerca a $-\infty$ y 1 cuando la variable x se acerca a $+\infty$.

Teniendo como punto medio el valor 0.5, esta función resulta especialmente útil para facilitar la interpretación de resultados en un formato binario.

Se realizará la aplicación del algoritmo de aprendizaje asistido de Retro propagación. Por lo que se definen también los procedimientos necesarios para realizar el entrenamiento y aprendizaje.

En pocas palabras, el aprendizaje por retro propagación se realiza generando inicialmente una propagación hacia adelante dentro de la red neuronal no entrenada (coeficientes de pesos, bias y

demás aleatorios), posteriormente se compara el resultado obtenido con el resultado esperado (recordando que el aprendizaje es asistido) y a partir de esto se realiza un cálculo del error.

Para el cálculo del error, se realiza la operación de cálculo de gradiente, descrita previamente en el Capítulo III y posteriormente se genera una propagación en retroceso del error, considerando la participación (peso sináptico) de cada dendrita de las neuronas y por ende realizando los ajustes positivos o negativos a fin de preparar para una ejecución más certera en la próxima iteración.

Este proceso se repite múltiples veces (Epoch), lo que da lugar a que la red gradualmente aproxime sus valores internos a una función pseudo aleatoria que permita describir con cierto grado de exactitud la salida esperada a las entradas provistas.

Se genera para esto un proyecto en Visual Studio 2017 Community (edición para estudiantes e investigadores) aplicando lo antes mencionado.

4.2. Compuertas Lógicas

Como prueba de concepto para determinar la funcionalidad de los algoritmos dentro de la programación, antes de proceder con los análisis más complejos, se decide experimentar con una red neuronal de dos entradas que, bajo la misma estructura pero con aprendizajes distintos, deberá dar respuesta en similitud a las compuertas lógicas. Se determinan las compuertas AND, OR, NAND, NOR, XOR y XNOR para hacer las pruebas,

Como tales, estas compuertas lógicas poseerán dos entradas, A y B. Estas entradas serán los inputs de las dendritas de las neuronas; las cuales se transmitirán a lo interno de la red a fin de generar una única salida.

La siguiente figura muestra la representación y las tablas de verdad para cada una de las compuertas.

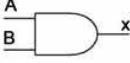
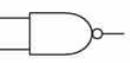
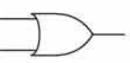
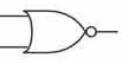
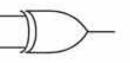
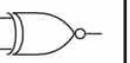
Name	AND	NAND	OR	NOR	XOR	XNOR																																																																																										
Alg. Expr.	AB	\overline{AB}	$A+B$	$\overline{A+B}$	$A\oplus B$	$\overline{A\oplus B}$																																																																																										
Symbol																																																																																																
Truth Table	<table border="1"> <thead> <tr><th>B</th><th>A</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	B	A	X	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <thead> <tr><th>B</th><th>A</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	B	A	X	0	0	1	0	1	1	1	0	1	1	1	0	<table border="1"> <thead> <tr><th>B</th><th>A</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	B	A	X	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <thead> <tr><th>B</th><th>A</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	B	A	X	0	0	1	0	1	0	1	0	0	1	1	0	<table border="1"> <thead> <tr><th>B</th><th>A</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	B	A	X	0	0	0	0	1	1	1	0	1	1	1	0	<table border="1"> <thead> <tr><th>B</th><th>A</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	B	A	X	0	0	1	0	1	0	1	0	0	1	1	1
B	A	X																																																																																														
0	0	0																																																																																														
0	1	0																																																																																														
1	0	0																																																																																														
1	1	1																																																																																														
B	A	X																																																																																														
0	0	1																																																																																														
0	1	1																																																																																														
1	0	1																																																																																														
1	1	0																																																																																														
B	A	X																																																																																														
0	0	0																																																																																														
0	1	1																																																																																														
1	0	1																																																																																														
1	1	1																																																																																														
B	A	X																																																																																														
0	0	1																																																																																														
0	1	0																																																																																														
1	0	0																																																																																														
1	1	0																																																																																														
B	A	X																																																																																														
0	0	0																																																																																														
0	1	1																																																																																														
1	0	1																																																																																														
1	1	0																																																																																														
B	A	X																																																																																														
0	0	1																																																																																														
0	1	0																																																																																														
1	0	0																																																																																														
1	1	1																																																																																														

Figura 14. Compuertas Lógicas.

(Fulton, 2018)

A fin de dar respuesta a este análisis, se estima una red de 5 neuronas que constan de una capa de entrada (Input) una capa oculta (Hidden) y una capa de salida (Output). Tal como se indica en la figura siguiente.

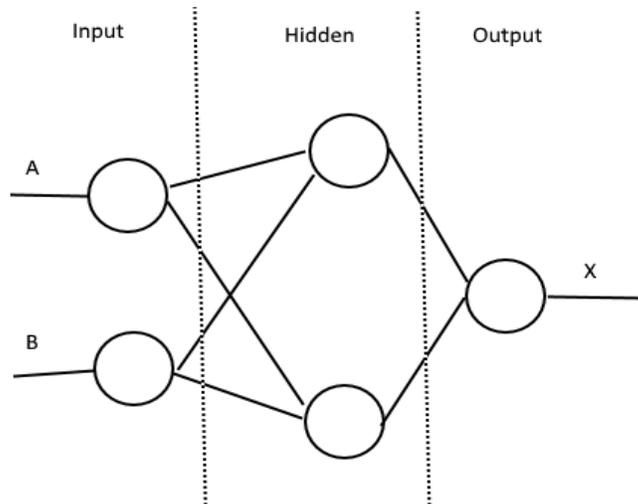


Figura 15. Red propuesta.

(Fuente propia)

Se diseña una aplicación que permitirá posicionar la red tanto en entrenamiento / aprendizaje, como en prueba. Se resumen a continuación los resultados.

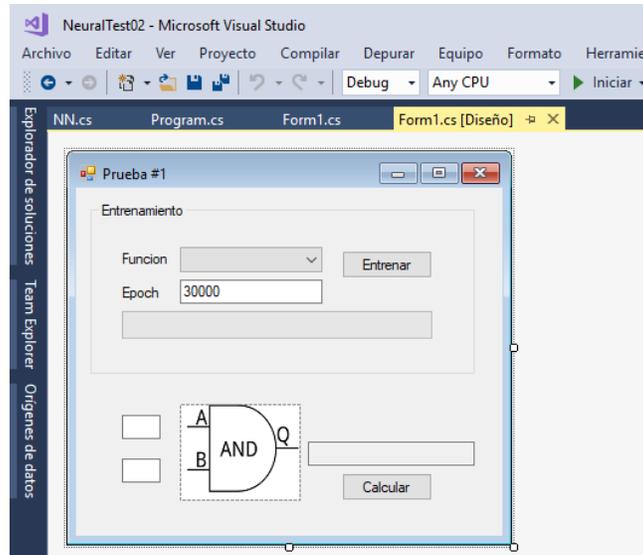


Figura 16. Proyecto de solución a Compuertas Lógicas.

(Fuente propia)

AND. Aproximadamente un mínimo de 25,000 repeticiones.

Input A	Input B	Output
0	0	0.000153
0	1	0.003962
1	0	0.003962
1	1	0.992764

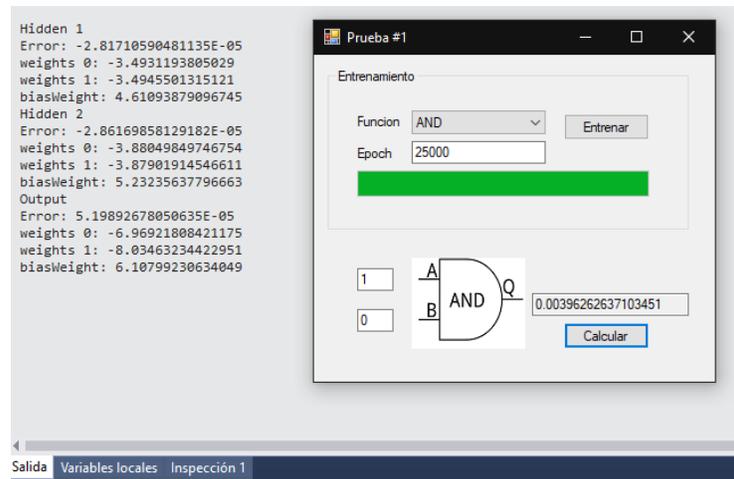


Figura 17. Prueba AND.

(Fuente propia)

OR. Aproximadamente un mínimo de 700 repeticiones.

Input A	Input B	Output
0	0	0.054297
0	1	0.970479
1	0	0.970181
1	1	0.994123

Hidden 1
Error: 1.24257165920545E-06
weights 0: 3.34903000515752
weights 1: 3.35651495395854
biasWeight: -1.85158696876859
Hidden 2
Error: 1.30039879889414E-06
weights 0: 3.30679245065818
weights 1: 3.31407395815976
biasWeight: -1.83160187279488
Output
Error: 3.43403649207334E-05
weights 0: 4.71312142359553
weights 1: 4.62824863471163
biasWeight: -4.1358665588508

Prueba #1
Entrenamiento
Funcion: OR
Epoch: 700
0.994123234751684
Calcular

Figura 18. Prueba OR.
(Fuente propia)

NAND. Aproximadamente un mínimo de 25,000 repeticiones.

Input A	Input B	Output
0	0	0.999847
0	1	0.996045
1	0	0.996045
1	1	0.007225

Hidden 1
Error: -2.84411900525734E-05
weights 0: -3.77067988376767
weights 1: -3.77029811136099
biasWeight: 5.05686347034615
Hidden 2
Error: -2.82645886070672E-05
weights 0: -3.61409727729433
weights 1: -3.61438397816027
biasWeight: 4.80538200732202
Output
Error: -5.18399874007405E-05
weights 0: 7.72086421493388
weights 1: 7.28885499594365
biasWeight: -6.11808625402752

Prueba #1
Entrenamiento
Funcion: NAND
Epoch: 25000
0.0072251116595246
Calcular

Figura 19. Prueba NAND.
(Fuente propia)

NOR. Aproximadamente un mínimo de 1000 repeticiones.

Input A	Input B	Output
0	0	0.957825
0	1	0.022931
1	0	0.023093
1	1	0.004487

Hidden 1
Error: 6.33544050363206E-07
weights 0: 3.477809856191
weights 1: 3.48292451731556
biasWeight: -1.92084837056825
Hidden 2
Error: 6.45038295622818E-07
weights 0: 3.46138253123664
weights 1: 3.46644269128296
biasWeight: -1.91309493418148
Output
Error: -2.00540633981264E-05
weights 0: -4.9428091199563
weights 1: -4.90907704042946
biasWeight: 4.38585981164989

Prueba #1
Entrenamiento
Funcion: NOR
Epoch: 1000
Entrenar
1 1 NOR Q 0.00448799289013598
Calcular

Salida Variables locales Inspección 1

Figura 20. Prueba NOR.

(Fuente propia)

XOR. Aproximadamente 50,000 repeticiones.

Input A	Input B	Output
0	0	0.004928
0	1	0.994313
1	0	0.995306
1	1	0.004439

Hidden 1
Error: 4.68874803071067E-06
weights 0: 7.48176398108522
weights 1: -7.38188742366162
biasWeight: 3.72680850910062
Hidden 2
Error: -6.54627387847262E-06
weights 0: 6.19686556525941
weights 1: -6.41308820306213
biasWeight: -3.30748629990516
Output
Error: -1.96239215596441E-05
weights 0: -11.4522203805072
weights 1: 11.9886925348496
biasWeight: 5.45191713922932

Prueba #1
Entrenamiento
Funcion: XOR
Epoch: 50000
Entrenar
1 1 XOR C 0.00443954255043065
Calcular

Salida Variables locales Inspección 1

Figura 21. Prueba XOR.

(Fuente propia)

XNOR. Aproximadamente 50,000 repeticiones.

Input A	Input B	Output
0	0	0.995080
0	1	0.005683
1	0	0.004695
1	1	0.995545

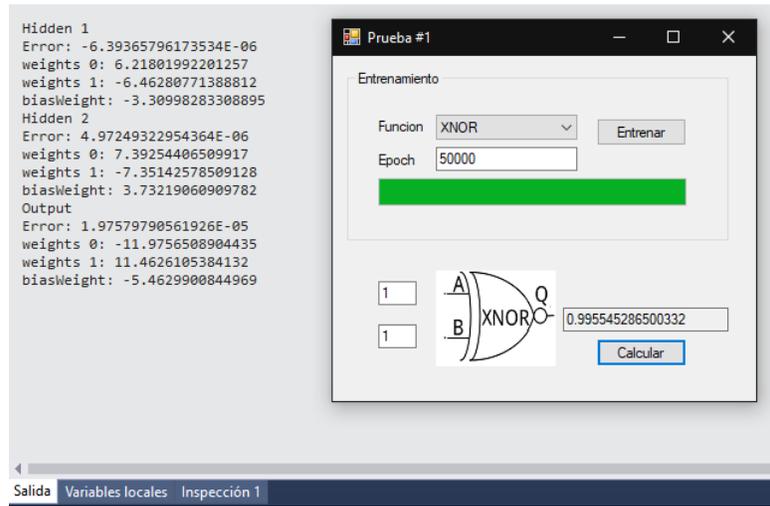


Figura 22. Prueba XNOR.

(Fuente propia)

Para todas las pruebas, se realizaron las iteraciones necesarias para aproximar el error en la neurona de salida a un valor en exponencial -5 (5 ceros después del punto).

Como se observa, el nivel de entrenamiento varía, en algunos como la compuerta OR NOR, apenas requerían menos de 1,000 iteraciones para generar resultados con un error menor al definido.

Para el caso de compuertas XOR y XNOR, fueron necesarias muchas más iteraciones (~50,000) para llegar a un error aceptable.

Para fines de demostrar la aplicación de la retro propagación del error a fin de que cada iteración genera un resultado más aproximado al real, se realizó la captura del error en un proceso de 3 millones de iteraciones.

Se muestra el resultado de su aproximación en forma de gráfica, en la ilustración siguiente.

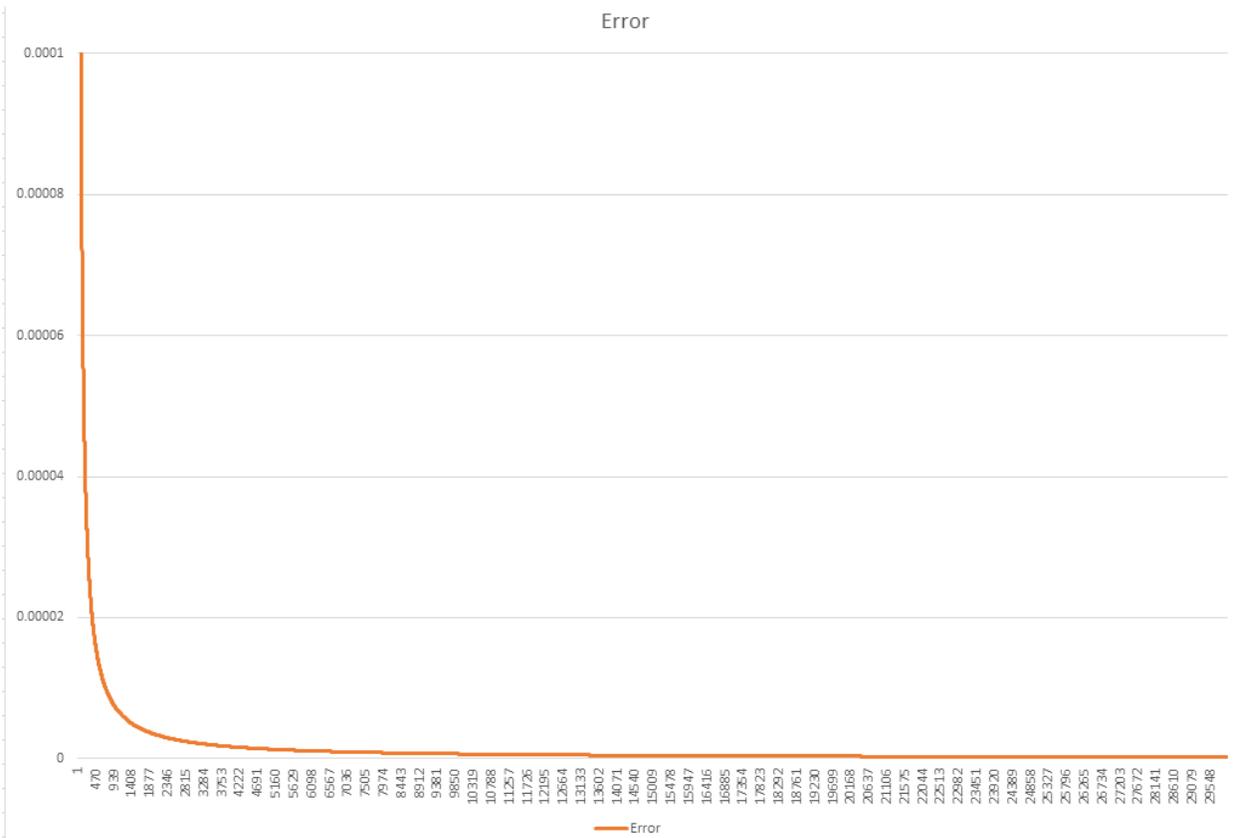


Figura 23. Gráfica del Error durante 3 millones de iteraciones.

(Fuente propia)

Tal como se definió en el marco teórico, el algoritmo de descenso de gradiente está diseñado para converger rápidamente a valores cercanos al real. Prueba de esto es la drástica reducción del error conforme se itera sobre la red y se propagan los ajustes sinápticos entre ella. Si bien tiene posibles fallos como ser el estancamiento, se presume que este algoritmo responde lo suficientemente bien para los análisis propuestos.

4.2.1. Saturación de la red

Entre las observaciones del experimento inicial se detectó un fallo en los cálculos con ocurrencia aparentemente aleatoria. Esto significa que a pesar de que se realizaban múltiples iteraciones, la red no siempre aprendía y progresaba, sino que se estancaba en un punto.

Un análisis más a profundidad reveló que bajo algunas situaciones, en ese momento desconocidas, las diferentes neuronas comenzaban a generar valores de salida cercanos a uno sin importar la entrada. En vista del diseño, el principal punto para realizar la búsqueda fue sobre la función de activación, en este caso Sigmoide y se encontró que, en efecto, la misma estaba recibiendo valores demasiado altos y por ende el resultado siempre tendía a cero.

La situación se presentaba esporádicamente, lo que llevó a considerar los valores iniciales de los diferentes pesos sinápticos en las dendritas, y se comprobó que la función Random⁴ era el origen. En ocasiones los números aleatorios generados eran lo suficientemente altos como para generar una saturación en la neurona, impidiendo que esta genere valores útiles.

Una reducción significativa (antes 0 – 1, ahora 0 – 0.1) bastó para eliminar la incidencia.

Adicionalmente otro punto agregado consiste en la velocidad con que la red aprende. Existe un parámetro llamado Ratio de Aprendizaje, el cual determina que tanta relevancia se generara en cada iteración. Sin embargo, un valor muy elevado podría generar un estancamiento entre los valores positivos y negativos durante el descenso de gradiente. Por este hecho se consideró reducir la velocidad de aprendizaje a un 30%. Si bien esto requerirá más iteraciones para reducir el error, también reducirá la posibilidad de incidir en algún valle y obstruir el aprendizaje.

⁴ Función random o aleatoria.

4.3. Experimento 1. Detección de procesos ineficientes

El primer experimento consistirá en obtener una medida indicativa del grado de optimización de un proceso. Dentro del departamento existen alrededor de 4,000 procesos distintos, algunos más eficientes que otros.

En ocasiones, las ineficiencias de algunos procesos han resultado relevantes y han contribuido al fallo en la entrega de información a tiempo; es por esta razón que, para reducir la posibilidad de incurrir en este escenario, se ha hecho un esfuerzo por mejorar algunos procesos detectados, ya que varios de ellos han quedado desfasados o bien obsoletos, utilizan técnicas poco optimizadas o contienen errores que reducen su rendimiento (tanto en tiempo como en uso de recursos).

A modo de ejemplo, se han encontrado procesos con duraciones de hasta 2 o 3 horas y que después de un análisis y reingeniería, se ha reducido su tiempo a menos de 3 minutos.

La solución ideal sería examinar los 4,000 procesos y realizar las mejoras, sin embargo es claro que esta tarea no es factible considerando que los recursos (tiempo y personal) no son ilimitados. Es allí donde entra la primera aplicación de la red neuronal, con el objetivo de reemplazar el criterio humano que detecta los procesos ineficientes y de forma automática indicar aquellos de mayor impacto, tanto en posible optimización como en criticidad y por ende enfocar los esfuerzos en esos puntos que mejor resultado generaran.

4.3.1. Recopilación de datos

Tal como se indica en el Capítulo III, Metodología de la Investigación, algunas de las variables más relevantes que serán usadas como entradas de la red, serán:

- Tiempo de ejecución
- Cantidad de registros insertados

- Cantidad de tablas consultadas
- Total de líneas de código
- Producto cartesiano de registros por tablas
- Total de registros escritos

Se realizará una extracción en primera instancia de todos aquellos procesos que tenían ineficiencias y que lograron optimizarse considerablemente. Para ello se creó una consulta SQL que detecta cuando un proceso paso de tomar mucho tiempo a poco tiempo, para luego evaluar manualmente si el motivo de esto fue una interacción externa. Se descartan asimismo aquellos procesos que por su naturaleza tienen un comportamiento acoplado a este criterio, por ejemplo, los que son mensuales (un día tardaran mucho, los siguientes del mes no), entre otras excepciones.

Una vez identificados estos procesos, se considerarán muchas de sus ejecuciones previas (ineficientes) asignando a estas un valor representativo a posible optimización (por ejemplo, salida igual a 1 indicando que se puede optimizar) o bien una salida de punto decimal comprendida entre 0 y 1 que indicara el grado de optimización que se logró, por ejemplo, un 80% de mejora en el tiempo equivaldrá a un 0.8 de salida. Se realizarán las pruebas para determinar el escenario que mejor resultado presente.

Normalización de datos

Existen una gran fluctuación en las variables de entrada, en algunas situaciones algunos procesos pueden leer miles de millones de registros, otros solamente unos pocos cientos. Similar a los tiempos, algunos pueden tardar 70 minutos en estado ineficiente, y otros 500 minutos en estado ineficiente.

La complejidad de tener valores demasiado dispersos podría dar lugar a falsos positivos, por ejemplo, un proceso que podría tardar 5 minutos de forma optimizada tarda 300 de forma no optimizada, pero un día x toma 50. Esto podría generar una decisión incorrecta ya que los 50 minutos podrían considerarse un “buen” tiempo ya que es mucho menor que los 300, sin embargo, esto está lejos de la realidad pues es diez veces el tiempo óptimo.

Si lo viéramos en una distribución normal, tal como es un principio en estadística, los valores extremos pueden degradar el análisis.

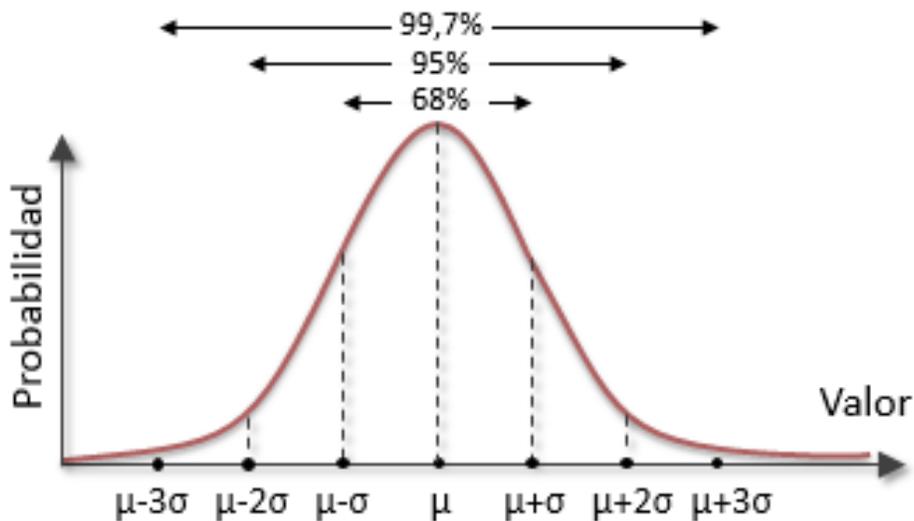


Figura 24. Curva normal con división de cuartiles.
(Matemáticas 10, 2017)

Este tipo de análisis lleva a considerar la necesidad de estandarizar los valores de entrada según algún tipo de criterio. Ya sea este la eliminación de extremos o bien su reemplazo por valores pseudo máximos. Esto último podría ser la utilización de los cuartiles Q1 y Q3 o incluso la mediana en las posiciones 25 y 75.

4.3.2. Muestra de datos

En primer lugar, se recopila una serie de datos sobre procesos que han sido optimizados considerablemente (mayor que 60% de su rendimiento), esta información incluye todos los datos previos a su optimización más un resultado conocido, en este caso un valor porcentual que representa el grado de mejora logrado. La tabla a continuación muestra una fracción de los datos recopilados.

Tabla 7. Datos de muestra.

Índice	Duración (Minutos)	Complejidad de código	Orígenes de datos	Estimado de datos entrada	Destino de datos	Estimado de datos de salida	Ponderación de mejoras
1	45.82	226	8	44763427.71	1	732140.57	0.7533
2	60.1	992	22	8403109.68	1	627719.04	0.9421
3	60.51	1391	35	16908924.93	2	150996.5	0.8312
4	86.5	280	10	6647304.31	1	7626503.2	0.9312
5	218.43	648	6	736485.31	4	5232500.81	0.8715
6	260.26	488	17	10608221.36	1	165309.8	0.958
7	42.38	398	4	2285412.78	1	587617.45	0.964
8	206.33	166	5	33718810.09	1	2295.11	0.814
9	213.05	428	9	38911192.49	2	4960979.84	0.8799
10	78.1	740	2	83856.53	1	4839.7	0.9524
11	84.65	416	5	45730150.59	1	441.54	0.9555
12	412.6	536	27	71752704.37	1	754456.22	0.9828
13	83.52	242	8	7021741.34	1	6410210.96	0.9327
14	88.3	837	8	9905698.24	5	2875553.56	0.7687
15	196.66	1898	16	4094740.93	1	15766.31	0.7774
16	30.38	676	10	5183296.72	1	194401.76	0.9193
17	143.89	4325	31	11595129.63	2	0	0.9871
18	115.41	1747	12	5287824.37	8	2582286.42	0.9661
19	77.22	1538	28	11956048.35	2	283344.41	0.9065
20	362.43	391	24	37520196.52	1	6190181.4	0.9586
21	285.73	590	22	44672129.25	1	757.05	0.933
22	131.54	472	7	3520480.05	1	933.38	0.9962

Como se observa, se incluyen datos puntuales que indican el tiempo de duración (en minutos), una variable compuesta que refleja la complejidad de código en ese momento, cantidad de fuentes consultadas, total de registros estimados en las consultas, total de tablas escritas, total de registros escritos y, por último, el valor que representa el porcentaje de mejora logrado.

El siguiente paso en la muestra de los datos, es una estandarización y conversión a valores comprendidos entre 0 y 1. Esto significa en primer lugar y tal como se indicó en la sección 4.3.1 reemplazar los valores máximos o extremos por el tercer o primer cuartil; posteriormente ponderar estos datos para que sean comprendidos entre un rango de 0 y 1.

Tabla 8. Normalización de datos.

Índice	Duración	Complejidad	Orígenes de datos	Volumen de entrada	Destinos de datos	Volumen de salida	Ponderación
1	0.458200	0.226000	0.533333	1.000000	0.100000	0.146428	0.753300
2	0.601000	0.992000	1.000000	0.280104	0.100000	0.125544	0.942100
3	0.605100	1.000000	1.000000	0.563631	0.200000	0.030199	0.831200
4	0.865000	0.280000	0.666667	0.221577	0.100000	1.000000	0.931200
5	1.000000	0.648000	0.400000	0.024550	0.400000	1.000000	0.871500
6	1.000000	0.488000	1.000000	0.353607	0.100000	0.033062	0.958000
7	0.423800	0.398000	0.266667	0.076180	0.100000	0.117523	0.964000
8	1.000000	0.166000	0.333333	1.000000	0.100000	0.000459	0.814000
9	1.000000	0.428000	0.600000	1.000000	0.200000	0.992196	0.879900
10	0.781000	0.740000	0.133333	0.002795	0.100000	0.000968	0.952400
11	0.846500	0.416000	0.333333	1.000000	0.100000	0.000088	0.955500
12	1.000000	0.536000	1.000000	1.000000	0.100000	0.150891	0.982800
13	0.835200	0.242000	0.533333	0.234058	0.100000	1.000000	0.932700
14	0.883000	0.837000	0.533333	0.330190	0.500000	0.575111	0.768700
15	1.000000	1.000000	1.000000	0.136491	0.100000	0.003153	0.777400
16	0.303800	0.676000	0.666667	0.172777	0.100000	0.038880	0.919300
17	1.000000	1.000000	1.000000	0.386504	0.200000	0.000000	0.987100
18	1.000000	1.000000	0.800000	0.176261	0.800000	0.516457	0.966100
19	0.772200	1.000000	1.000000	0.398535	0.200000	0.056669	0.906500
20	1.000000	0.391000	1.000000	1.000000	0.100000	1.000000	0.958600
21	1.000000	0.590000	1.000000	1.000000	0.100000	0.000151	0.933000
22	1.000000	0.472000	0.466667	0.117349	0.100000	0.000187	0.996200

El patrón de colores se utiliza únicamente para fines visuales y determina en verde cuando un valor aporta positivamente a una optimización, y en rojo cuando se aleja de ella. La columna final muestra en azul el rango de optimización obtenido.

Los valores de entrada se separan de los de salida en dos archivos individuales, los cuales servirán de insumo para la red construida a modo de entrenamiento.

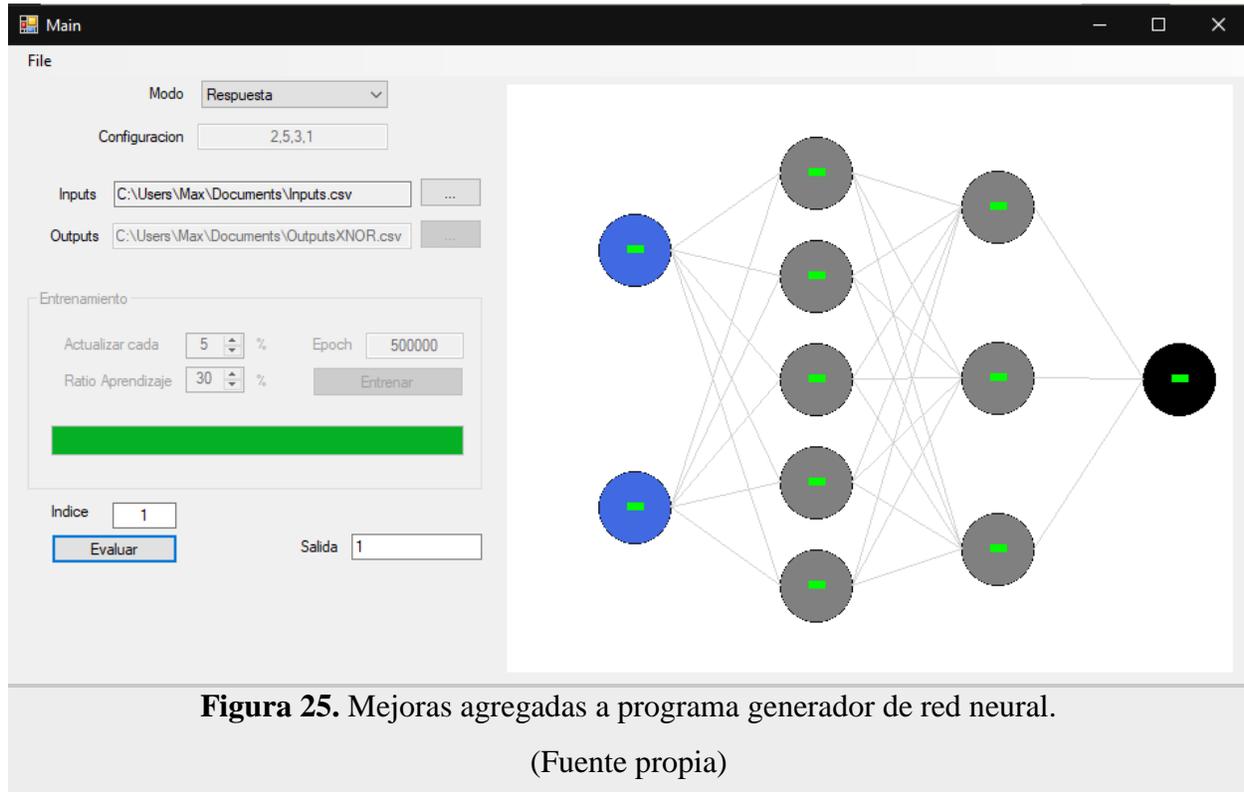
4.3.3. Preparación de sistema neural

Se agregaron algunas mejoras al programa desarrollado con el fin de soportar las siguientes acciones:

- Serialización ⁵de objetos: Permitirá guardar y cargar una red completa, incluyendo su entrenamiento y aprendizaje.
- Lectura de entradas y salidas: Permitirá indicar un archivo en formato separado por comas (*.csv) tanto las entradas como salidas para operar en modo aprendizaje, de este modo es posible alimentar el modelo con un set de datos externo.
- Configuración de estructura de red: Permite crear fácilmente una red de tamaño variable, especificando la cantidad de capas y la cantidad de neuronas por cada capa.
- Modo de respuesta: Permite indicar el índice de entradas y generar una respuesta utilizando dichas entradas.
- Modo aprendizaje: Habilita las funciones de aprendizaje de la red, permitiendo determinar la velocidad (ratio) y la frecuencia de actualización para operaciones externas.
- Representación visual de la red: Dibuja una representación gráfica de la estructura de la red, mostrando cada capa, neurona y el error contenido en dicha neurona, sumado a sus conexiones.
- Modo Batch Respuesta: Permite, mediante un archivo de entradas, generar un lote de respuestas para cada una de dichas entradas, útil para analizar grandes volúmenes de datos.

⁵ Función para enlazar los datos en forma de una serie o sucesión lógica.

La figura a continuación muestra un ejemplo del diseño del programa, en este caso alimentado con archivos de entradas y salidas correspondientes a una compuerta lógica XNOR.



Los cuadros dentro de cada circulo (neurona) representan el error contenido en dicha neurona, rojo es igual a un error alto, verde igual a un error bajo.

En la figura se muestra también la evaluación del primer elemento de la serie de entradas (Inputs.csv), que en este caso representa 0,0 y el resultado generado por la red es igual a 1.

4.3.4. Pruebas funcionales

Para alimentar el modelo, habiendo normalizado y estandarizado los datos, se determina eliminar la variable cantidad de tablas de respuesta (CHILD_COUNT), ya que en la gran mayoría de los casos los procesos escribían a una única tabla, se considera podría entorpecer el análisis al estar correlacionada directamente con la variable de total de datos escritos, es decir no es una variable independiente sino dependiente.

Se consideran un mínimo de 5 variables independientes que tendrán un impacto directo en los resultados, pudiendo existir otras variables que se considerara agregar dependiendo de los resultados obtenidos y su precisión.

Se determina una estructura de red de 5 x 9 x 4 x 1 para generar el aprendizaje. El diseño de la misma se adjunta a continuación.

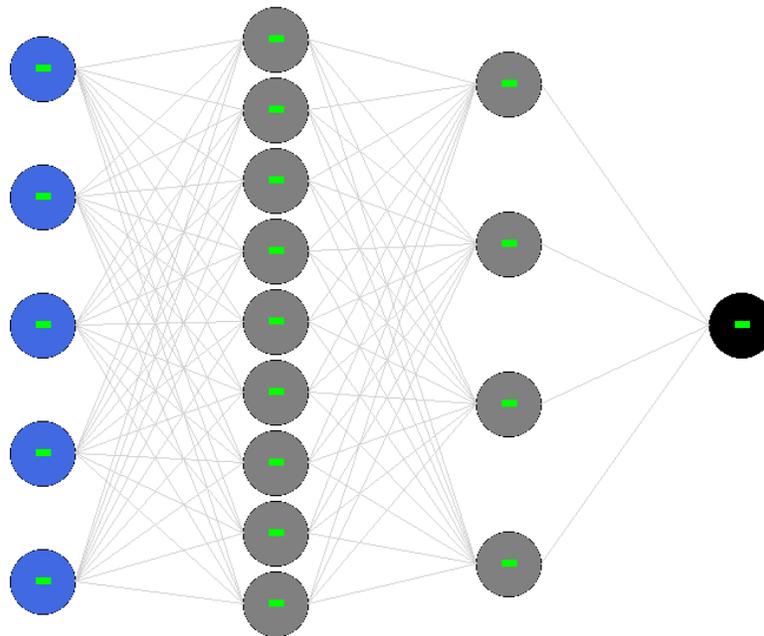


Figura 26. Diseño de red 5x9x4x1 propuesto para experimento 1.

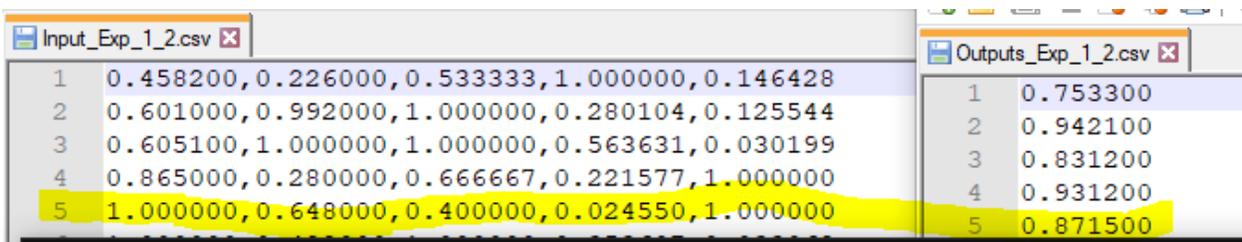
(Fuente propia)

Posterior a la carga de entradas y salidas de entrenamiento, se determina un porcentaje de aprendizaje de 30%, y 5 millones de iteraciones sobre los datos.

Esto conlleva aproximadamente 1 hora de entrenamiento con un procesador i7 6700K a 4GHz.

Tal como se visualiza en la figura anterior, el error contenido dentro de cada neurona reduce rápidamente hasta rondar valores menores a un exponencial -16 (un punto decimal seguido de 15 ceros seguido de otros números), es decir muy pequeño.

Una vez completado el entrenamiento, se procede inicialmente a validar el mismo utilizando para ello los propios datos de entrenamiento y datos de un grupo de control.



Index	Input 1	Input 2	Input 3	Input 4	Input 5	Output
1	0.458200	0.226000	0.533333	1.000000	0.146428	0.753300
2	0.601000	0.992000	1.000000	0.280104	0.125544	0.942100
3	0.605100	1.000000	1.000000	0.563631	0.030199	0.831200
4	0.865000	0.280000	0.666667	0.221577	1.000000	0.931200
5	1.000000	0.648000	0.400000	0.024550	1.000000	0.871500

Figura 27. Datos de entrenamiento.

(Fuente propia)



Indice

Salida

Figura 28. Salida del programa.

(Fuente propia)

Se realizaron comparaciones similares sobre datos tanto de entrenamiento como de control y se determina que la red inicial aprendió correctamente.

A continuación se procede a realizar una generación de resultados en grupo (batch) sobre los aproximadamente 4 mil procesos, a fin de determinar el resultado de mejora pronosticado por la red.

Tabla 9. Datos de salida.

Índice	Duración	Complejidad	Orígenes de datos	Volumen de entrada	Destinos de datos	Volumen de salida	Ponderación
733	0.390200	0.903000	1.000000	1.000000	0.100000	1.000000	0.000000
736	0.000900	0.173000	0.066667	0.000000	0.100000	0.000000	0.009079
741	0.050500	0.373000	0.133333	0.199367	0.100000	1.000000	0.000000
749	0.059300	1.000000	1.000000	0.151201	0.200000	0.025902	0.000000
755	0.001400	0.328000	0.533333	0.042216	0.200000	0.002371	0.026845
758	0.537700	0.438000	0.666667	1.000000	0.100000	1.000000	0.885610
761	0.000900	0.178000	0.000000	0.000000	0.100000	0.000006	0.000265
801	0.000800	0.165000	0.000000	0.000000	0.100000	0.000001	0.000507
803	0.003900	0.515000	0.333333	0.003117	0.200000	0.212993	0.000000
811	0.000800	0.145000	0.000000	0.000000	0.100000	0.000049	0.001450
812	0.000400	0.195000	0.000000	0.000000	0.100000	0.000000	0.000109
813	0.365100	1.000000	0.733333	1.000000	0.100000	0.001649	0.000000
814	0.000700	0.325000	0.066667	0.000000	0.100000	0.000000	0.000002
815	1.000000	1.000000	0.400000	0.005148	0.300000	0.031155	0.929919

La tabla anterior muestra una fracción de los resultados generados.

Sobre los resultados generados, se realizará un análisis para identificar procesos ineficientes sobre los que una mejora tendrá un impacto considerable, utilizando para ello una tabla de priorización que determina cual flujo es más importante.

Esta tabla se genera a partir de las relaciones y dependencias entre los datos, tal como se indicó el flujo de información genera una cadena por lo que es posible determinar el impacto que tiene cada punto en dicha cadena. (Véase sección 2.5).

4.3.5. Aplicación de resultados Experimento 1

Se realiza un cruce de datos entre los resultados generados por la red neuronal en conjunto con el análisis de procesos críticos a fin de definir los procesos de mayor impacto posible, considerando su importancia en los flujos (criticidad) y la posibilidad de optimización según la red (mayor que 80%); tal como se muestra en la tabla a continuación.

Tabla 10. Priorización de resultados.

ID Proceso	Índice	Nombre	Minutos	Posibilidad de optimización	Criticidad	Olap	Cubos
275066	1421	Esquema1.Proceso1	78.63	93.54%	222	4	32
31223278	1397	Esquema2.Proceso1	0.17	85.45%	157	3	11
31447607	1602	Esquema3.Proceso1	11.00	88.12%	156	3	11
371698	346	Esquema2.Proceso2	124.32	92.99%	143	3	11
414617	1536	Esquema3.Proceso2	62.57	90.48%	129	3	11
34264407	758	Esquema3.Proceso3	82.59	88.56%	92	2	9

Los tres procesos marcados en verde son los procesos seleccionados para intentar realizar una evaluación de posible optimización.

Los procesos en amarillo se omiten ya que, a pesar de que la red genera un buen pronóstico, los mismos ya tienen un tiempo de ejecución reducido, por lo que aun si se puede mejorar, el tiempo ganado será mínimo.

El proceso marcado en naranja se excluye ya que incluye una variable que no se incorporó al modelo ni a su aprendizaje (dependencia de base de datos externa) por lo que de momento no se estudiara.

Una criticidad mayor que 200 representa un impacto sobre un 80% de los flujos, mientras que una criticidad de 100 representa un 40% de los flujos. Se considera que una criticidad mayor a 100

representa un punto vital para la entrega de datos pues el valor para el negocio se genera en el final de la cadena, por lo que un 95% a los ojos del usuario final es casi tan malo como un 0%.

Los campos OLAP (Cubo de procesamiento analítico en línea) y Cubos (tradicionales) representan el total de reportes, informes y dashboard⁶ que se generan usando esta información, el que tenga 1 o más OLAP significa que a partir de estos datos se entrega un servicio crítico al usuario, por lo que aun siendo solo 1, el impacto de no cumplir será significativo.

La tarea de optimización sobre cada proceso se realiza a modo de un estudio y comprensión del funcionamiento y objetivo del mismo. Aplicando diferentes técnicas de programación y optimización de código sumado a una posible reestructuración, en conjunto con herramientas como analizador de SQL⁷ y demás, se determina si es o no posible realizar una mejora y en base a pruebas, su aplicación.

A continuación se muestran los resultados sobre los 3 procesos en el top de Criticidad x Posibilidad. El rectángulo mostrara los tiempos en minutos de los mismos, mientras que la línea verde determina el momento en que se intentó aplicar la mejora.

Cabe notar que en los tres casos, el resultado generado por la red es apropiado pues en 2 de ellos, al momento de estudiar el código, se encontraron fallos de rendimiento significativos y en algunos casos bastante obvios, por lo que en estos dos primeros casos el esfuerzo se resume a poco menos de 30 minutos cada uno.

⁶ Tablero de mando y visualización de la información.

⁷ SQL: Lenguaje Estructurado de Consulta.

Para el tercer caso en principio se creía que no era posible una mejora ya que el código se veía bastante bien, sin embargo, un análisis profundo sobre el Plan de Ejecución de SQL⁸ demostró un fallo en la aplicación de paralelismo en un query⁹ que lo requería. La optimización tomo alrededor de 1 hora.

Dentro del departamento, la autorización para realizar las mejoras cae dentro de una solicitud de cambio derivada de la mejora continua y su posterior paso a producción, a lo interno del departamento de TI. En otras empresas esta implementación podría variar ya sea en sus autorizaciones y/o el proceso a seguir.

Con respecto al proceso de optimización; este conlleva la asignación de personal con conocimientos avanzados en el análisis, funcionamiento, metodologías y tecnologías asociadas resolución de problemas para un código eficiente y con una apropiada gestión de recursos. Se hacen uso de múltiples herramientas y estudio de plan de ejecución, así como el juicio experto y experiencia. Enfocados en comprender los motivos por los que el proceso no trabaja según lo esperado o bien los puntos que podrían hacer de cuello de botella y demás.

Dentro de las posibilidades se validan los factores encontrados y se somete a un rediseño o posible reestructuración con el fin de generar los mismos resultados de una manera más eficiente.

⁸ Ruta o plan que dicta el cómo se responderá una solicitud SQL.

⁹ Consulta, usualmente de Base de Datos.

En el primer proceso los resultados fueron sorprendentes. Se detectó una grave ineficiencia donde el proceso hacía un recorrido innecesario sobre millones de registros en busca de casos particulares, la mejora consistió en limitar los resultados a aquellos útiles antes de que pase a la segunda etapa de la ejecución, por lo que al llegar a esta el trabajo era realmente pequeño en cuanto a tiempo y procesamiento. Cabe destacar que el resultado final después de la mejora es exactamente el mismo, pero de forma mucho más eficiente.

Pasa de ~80 minutos a poco más de 10 segundos.

Min.Ejecución	Resultado Ejecución	Estado	Tipo de Ejecución
.17	ORA-0000: normal, successful completion	Completada	Calendarizada
.15	ORA-0000: normal, successful completion	Completada	Calendarizada
.15	ORA-0000: normal, successful completion	Completada	Calendarizada
74.02	ORA-0000: normal, successful completion	Completada	Calendarizada
80	ORA-0000: normal, successful completion	Completada	Calendarizada
82.68	ORA-0000: normal, successful completion	Completada	Calendarizada
75.18	ORA-0000: normal, successful completion	Completada	Calendarizada
77.03	ORA-0000: normal, successful completion	Completada	Calendarizada
78.85	ORA-0000: normal, successful completion	Completada	Calendarizada
75.63	ORA-0000: normal, successful completion	Completada	Calendarizada
78.52	ORA-0000: normal, successful completion	Completada	Calendarizada
84.25	ORA-0000: normal, successful completion	Completada	Calendarizada

Figura 29. Optimización proceso 1.

(Fuente propia)

En el segundo proceso se pasa de un tiempo de ~120 minutos a un tiempo de ~3 minutos, que representa, al igual que el anterior caso, una mejora muy significativa. (Mayor que el 90%)

Min.Ejecución	Resultado Ejecución	Estado	Tipo de Ejecución
2.23	ORA-0000: normal, successful completion	Completada	Calendarizada
2.78	ORA-0000: normal, successful completion	Completada	Calendarizada
2.75	ORA-0000: normal, successful completion	Completada	Calendarizada
2.22	ORA-0000: normal, successful completion	Completada	Calendarizada
3.38	ORA-0000: normal, successful completion	Completada	Calendarizada
2.87	ORA-0000: normal, successful completion	Completada	Calendarizada
2.72	ORA-0000: normal, successful completion	Completada	Calendarizada
2.07	ORA-0000: normal, successful completion	Completada	Calendarizada
2.68	ORA-0000: normal, successful completion	Completada	Calendarizada
89.35	ORA-0000: normal, successful completion	Completada	Calendarizada
130.27	ORA-0000: normal, successful completion	Completada	Calendarizada
125.63	ORA-0000: normal, successful completion	Completada	Calendarizada
126.9	ORA-0000: normal, successful completion	Completada	Calendarizada
122.35	ORA-0000: normal, successful completion	Completada	Calendarizada
88.62	ORA-0000: normal, successful completion	Completada	Calendarizada
126.42	ORA-0000: normal, successful completion	Completada	Calendarizada

Figura 30. Optimización proceso 2.

(Fuente propia)

Para el tercer proceso, el cual en inicio se consideró sin margen de mejora, se encontró que no hacía uso de los recursos que explícitamente se indicaban debido a un fallo del optimizador en la base de datos, la solución fue una ligera modificación a las instrucciones enviadas al optimizador de código donde se soluciona el problema y se logra con ello una mejora significativa en tiempo.

Antes ~50 minutos, después ~26 minutos.

Min.Ejecución	Resultado Ejecución	Estado	Tipo de Ejecución
28.87	ORA-0000: normal, successful completion	Completada	Calendarizada
25.05	ORA-0000: normal, successful completion	Completada	Calendarizada
24.85	ORA-0000: normal, successful completion	Completada	Calendarizada
27.67	ORA-0000: normal, successful completion	Completada	Calendarizada
50.05	ORA-0000: normal, successful completion	Completada	Calendarizada
48.5	ORA-0000: normal, successful completion	Completada	Calendarizada
45.6	ORA-0000: normal, successful completion	Completada	Calendarizada
48.58	ORA-0000: normal, successful completion	Completada	Calendarizada
40.78	ORA-0000: normal, successful completion	Completada	Calendarizada
46.78	ORA-0000: normal, successful completion	Completada	Calendarizada
69	ORA-0000: normal, successful completion	Completada	Calendarizada
50.32	ORA-0000: normal, successful completion	Completada	Calendarizada

Figura 31. Optimización proceso 3.

(Fuente propia)

En conjunto, estas mejoras representan una optimización de entre 1.5 a 2.5 horas en el total de los flujos, lo cual es sumamente significativo. Demostrando así que la red genero resultados acertados al resaltar los procesos más ineficientes que, en conjunto con la criticidad, reducen una tarea de estudio de 4,000 procesos, en apenas unos pocos.

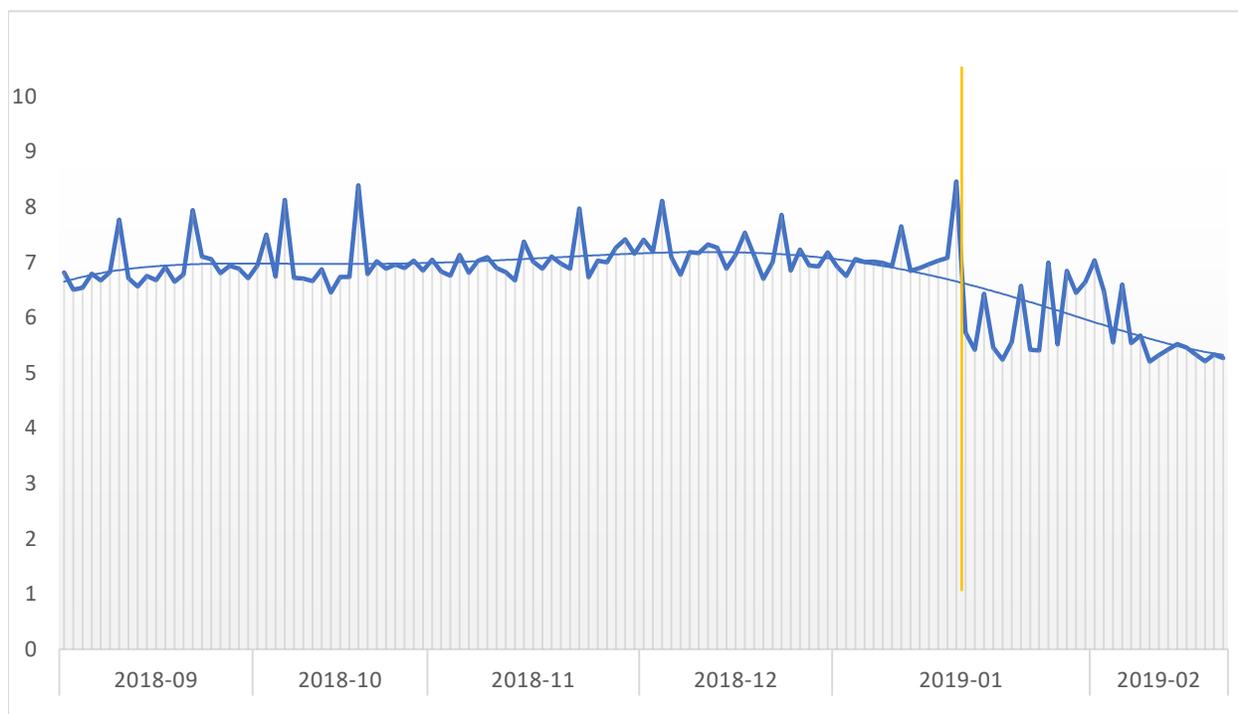


Figura 32. Impacto de los cambios en el tiempo de entrega de información.

(Fuente propia)

El grafico mostrado expone en el eje y (vertical) la hora en que se completa la carga de información, el eje x (horizontal) muestra una foto de cada día durante 6 meses. La línea naranja muestra el inicio de las optimizaciones realizadas; cabe notar que no todas se realizaron el mismo día, siendo las ultimas a mitad de febrero, donde es visible la reducción de tiempo en conjunto, así como la estabilización de los mismos.

4.4. Experimento 2. Detección de fallos.

El segundo experimento consiste en el entrenamiento de la red neuronal para detectar escenarios de fallo dentro de un sistema de flujo de información.

Un sistema Data Warehouse se conecta a muchos otros, por lo que en muchos casos no está dentro de su alcance directo la optimización de procesos externos; sin embargo, un fallo (ya sea interno o externo) sí que afecta el mismo y con ello los tiempos de entrega causando un retraso general en la generación de la información.

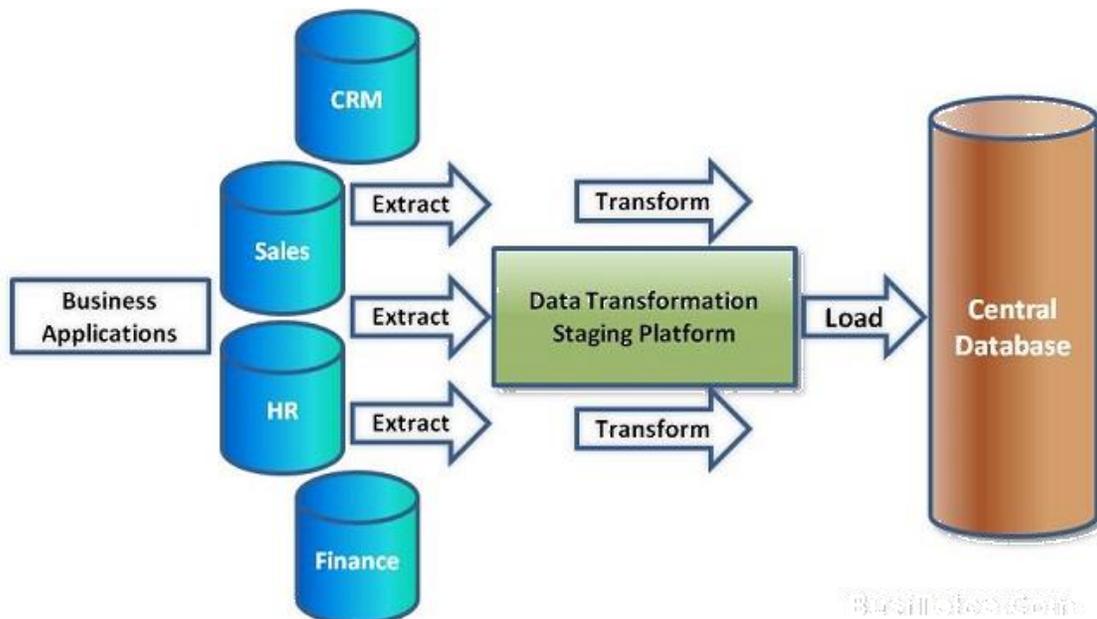


Figura 33. Interconexión de un Data Warehouse con múltiples sistemas.

(BusiTelCe, 2015)

El propósito de este segundo ejercicio es el de entrenar la red para que esta sea capaz de detectar con la mayor anticipación posible cuando se está o estará presentando un retraso en la entrega de los datos. Considerando para ello las variables pertinentes.

4.4.1. Muestra de datos

Se inicia la recopilación de datos generando información basada en el histórico de ejecuciones de 6 meses. Se genera a su vez una captura del estado del sistema por hora, es decir que el mismo tendrá 24 capturas por día, para los seis meses que se estarán evaluando.

Al mismo tiempo se realiza un análisis de las dependencias y estatus de las mismas para cada una de estas capturas, de modo que es posible conocer el estado de todos los flujos en cada momento del tiempo, a la vez que se puede determinar o clasificar si dicho estado llevo a un fallo o no.

Se almacenan variables de tiempo, cantidad de dependencias y cantidad de dependencias completadas. Se clasifican 6 flujos principales a los que se les desea detectar posibles inconvenientes. Estos flujos representan la información comercial del negocio, en este caso: Tráfico de Voz, Datos, Servicios de Valor agregado, Inter operador, Facturación, Roaming y Servicios Financieros.

Se estiman 8 salidas de la red, una de alerta general y 7 para representar cada uno de los flujos que se desean medir.

El cálculo inicial de los datos de entrada con que se alimentara el modelo se estima supere los 5 millones de registros, por lo que se realizaran pruebas para determinar si es conveniente reducir o no el mismo.

Como primer paso se recopila información correspondiente a 6 meses de histórico, es decir alrededor de 180 días. Cada día se compone por 24 muestras que corresponden a cada hora.

Como variables de entrada se consideran, entre otras, un valor asignado que corresponde a la cantidad de procesos predecesores completados para cada flujo, sobre el total de sí mismo. Es decir similar a un porcentaje, pero los valores se contemplan entre 0 y 1, estando 0.7 como valor mínimo para la función de activación en las neuronas.

Row #	FCT_DT	HOUR_READY	HOUR	29574314	29574315	29574316	29574317	29574318	29574319	29677329
1	01/08/2018	01/08/2018 11:32:36 a.m.	1	0.88	0.88	0.88	0.88	0.88	0.88	0.88
2	01/08/2018	01/08/2018 11:32:36 a.m.	2	0.86	0.86	0.86	0.86	0.86	0.86	0.86
3	01/08/2018	01/08/2018 11:32:36 a.m.	3	0.84	0.84	0.84	0.84	0.84	0.84	0.84
4	01/08/2018	01/08/2018 11:32:36 a.m.	4	0.8	0.8	0.8	0.8	0.82	0.8	0.82
5	01/08/2018	01/08/2018 11:32:36 a.m.	5	0.8	0.8	0.8	0.8	0.8	0.8	0.8
6	01/08/2018	01/08/2018 11:32:36 a.m.	6	0.8	0.8	0.8	0.8	0.8	0.8	0
7	01/08/2018	01/08/2018 11:32:36 a.m.	7	0	0.8	0.8	0.8	0.8	0.8	0
8	01/08/2018	01/08/2018 11:32:36 a.m.	8	0	0	0.8	0.8	0.8	0.8	0
9	01/08/2018	01/08/2018 11:32:36 a.m.	9	0	0	0.8	0	0	0.8	0
10	01/08/2018	01/08/2018 11:32:36 a.m.	10	0	0	0.8	0	0	0	0
11	01/08/2018	01/08/2018 11:32:36 a.m.	11	0	0	0.8	0	0	0	0
12	01/08/2018	01/08/2018 11:32:36 a.m.	12	0	0	0	0	0	0	0
13	01/08/2018	01/08/2018 11:32:36 a.m.	13	0	0	0	0	0	0	0
14	01/08/2018	01/08/2018 11:32:36 a.m.	14	0	0	0	0	0	0	0
15	01/08/2018	01/08/2018 11:32:36 a.m.	15	0	0	0	0	0	0	0

Figura 34. Ejemplo inicial de muestra de datos de entrada.

(Fuente propia)

En la figura arriba mostrada, se puede observar los valores de fecha, momento en que la información se completó, hora y las columnas siguientes representan el valor de entrada para cada flujo, considerando su avance en el momento de la captura de datos en la hora especificada.

410	1, 1, 1, 1, 1, 1, 1, 1
411	1, 1, 1, 1, 1, 1, 1, 1
412	0, 0, 0, 0, 0, 0, 0, 0
413	0, 0, 0, 0, 0, 0, 0, 0
414	0, 0, 0, 0, 0, 0, 0, 0
415	0, 0, 0, 0, 0, 0, 0, 0
416	0, 0, 0, 0, 0, 0, 0, 0
417	0, 0, 0, 0, 0, 0, 0, 0
418	1, 0, 0, 0, 0, 0, 1, 0
419	0, 0, 0, 0, 0, 0, 0, 0
420	1, 0, 0, 0, 0, 0, 1, 0
421	0, 0, 0, 0, 0, 0, 0, 0
422	0, 0, 0, 0, 0, 0, 0, 0
423	1, 0, 0, 0, 0, 0, 1, 0
424	0, 0, 0, 0, 0, 0, 0, 0
425	0, 0, 0, 0, 0, 0, 0, 0

Figura 36. Ejemplo de datos de salida para la red.

(Fuente propia)

Tal como se indicó, la red constara de 8 salidas. Se determina también utilizar 30 entradas para alimentar la misma.

4.4.2. Preparación del sistema neural

A continuación se procede a hacer el diseño de la red. Se consideran varios escenarios para el entrenamiento, sin embargo, debido a la cantidad de entradas (30) y salidas (8) se opta por un modelo de una sola capa, con 50 neuronas en ella.

Se programa un ratio de aprendizaje de 45% y 10,000 iteraciones sobre los casi 5,000 datos de entrada generados. Estos datos de entrada son los sumarizados correspondientes a los 6 meses de histórico, reduciendo mediante técnicas de agrupamiento y lógica los más de 5 millones de registros de data cruda.

A pesar del diseño relativamente simple, el aprendizaje de la red demanda considerable tiempo y recursos de procesamiento.

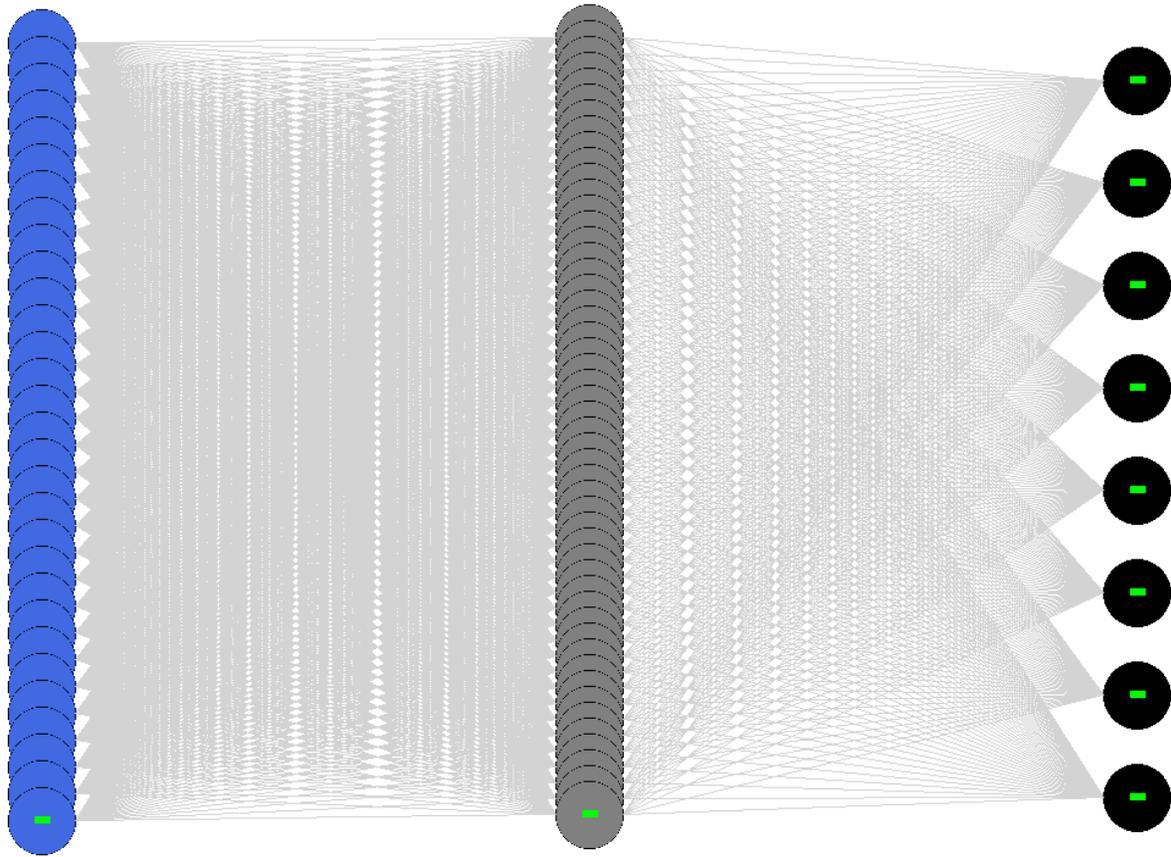


Figura 37. Red generada para el experimento, 30 x 50 x 8.

(Fuente propia)

El aprendizaje de la red toma más de 11 horas continuas, pero finalmente se completa con un margen de error poco significativo durante las pruebas.

4.4.3. Pruebas funcionales

Una vez completado el entrenamiento, se procede a comparar los resultados de la red contra un grupo de datos de control, es decir con datos que fueron previamente reservados para verificar la precisión con que la red detectaría este tipo de fallos.

Los ejemplos se muestran a continuación.

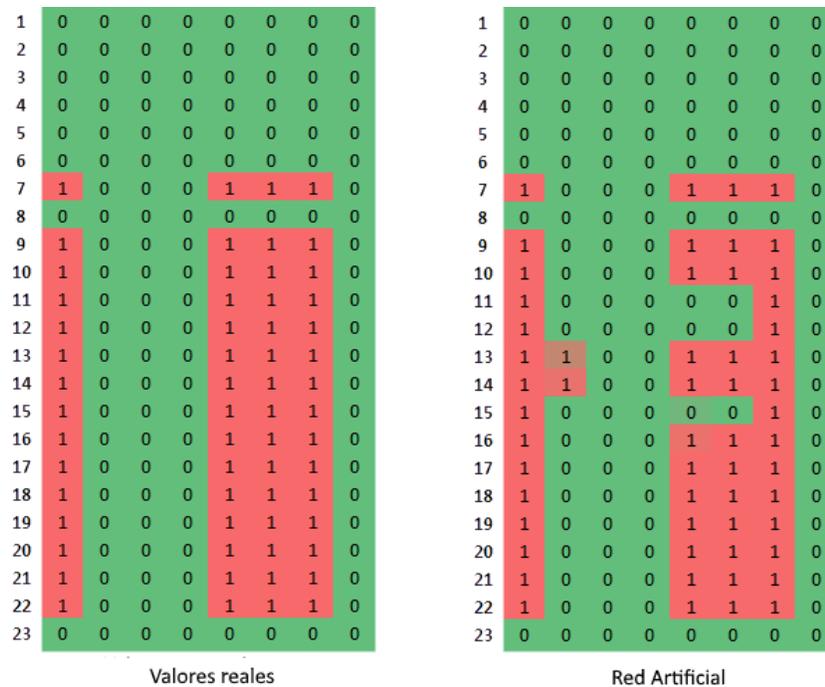


Figura 40. Comparación 3 valores esperados vs obtenidos.

(Fuente propia)

En los tres ejemplos mostrados, la primera columna representa la hora de la captura de datos; la segunda columna (la primera con colores) representa una alarma global y las siguientes 7 representan cada uno de los flujos que se monitorean.

Para cada ejemplo, la imagen de la izquierda muestra el resultado real de los datos, es decir cuál fue el resultado real de los datos de entrada. La imagen de la derecha representa los datos generados por la red neuronal artificial.

La mayoría de los datos de prueba se asemejan a la primera imagen en el sentido que la red provee con mucha exactitud el resultado que se generara.

Hay casos como los mostrados en las figuras de ejemplos 2 y3, donde lo previsto por la red no es 100% exacto a lo que realmente sucedió; sin embargo, se observa que, en efecto, la red genera una

alerta e identifica previamente cuales son los flujos más posibles para alarmarse con un nivel de precisión considerable, resultado bastante satisfactorio.

Se considerara un entrenamiento mayor ya que el actual fue de 10,000 iteraciones sobre 5,000 entradas (alrededor de 50 millones de evaluaciones, sumando a ello retro propagación recursiva) con tiempo de 11 horas. Se prevé que un mayor entrenamiento podría mejorar aún más la exactitud de la red, sin embargo, con el estado actual se realizaran evaluaciones para su integración dentro del sistema de alarmas y monitoreo 24/7 del departamento ya que se demuestra que prevé con una exactitud razonable un escenario de fallo, antes que suceda.

4.5. Referente a metodología y resultados.

La propuesta de implementación de técnicas de Inteligencia Artificial así como el programa diseñado, son propiedad intelectual generada por mi persona en la realización de este trabajo. Se facilita a Tigo Honduras la prestación del servicio derivado de este, quedando limitado únicamente a las ventajas obtenidas de la aplicación de este servicio y no a la totalidad de esta investigación.

En ambos experimentos se utiliza una metodología cuantitativa para medir los resultados generados de esta implementación, para ello se utilizan indicadores de rendimiento, también llamados KPI, los cuales sirven como base para determinar la mejora en factores que el negocio ha determinado previamente como valiosos y críticos; que funcionan a su vez como punto de evaluación para el departamento. En su mayoría estos son factores de tiempo y disponibilidad, pero también se suman términos de mejor utilización de recursos y generación de valor en TI.

CAPITULO V – CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- 1) Según se mostró en los apartados referentes a ITIL y COBIT; las tecnologías de información y concretamente los servicios prestados mediante ellas son de crucial importancia para las empresas modernas, por tanto estos estándares no solo se mantienen relevantes, sino que son cada vez más importantes en el mejoramiento de TI. Ambos se complementan mutuamente y resaltan la importancia tanto de la garantía del servicio como de la mejora continua, puntos que sin problema pueden apoyarse de la inteligencia artificial.
- 2) A fin de aplicar ciencia de datos sobre una empresa con un fin específico, es de vital importancia analizar, estudiar, interpretar y comprender cuales son las circunstancias en las que se encuentra. Permitiendo así definir las variables de importancia, descartando aquellas irrelevantes y aquellas interdependientes; así como estandarizar y normalizar la información. Esto permanece verdadero para prácticamente cualquier análisis de datos ya que al final serán estos datos de entrada los que alimenten el modelo y, claro está, la calidad de los resultados generados estará directamente ligado a los mismos.
- 3) Tal como se demostró en el experimento 1, no solo es posible sino que es muy favorable la aplicación de técnicas de inteligencia artificial sobre análisis extensivos, pero con comportamientos similares; en este caso se demostró que una red artificial puede y realizó una identificación de ineficiencias sumamente certero; sobre una base detectada menor a 1% de los datos (6 procesos de 4000) se logró un cambio de casi un 10% de mejora global (>2 horas al día).

- 4) El experimento 2 demuestra que es posible predecir con anticipación un evento de fallo, si bien por la gran cantidad de variables de entrada y salida el análisis es intensivo y requiere más tiempo; los resultados son sumamente útiles para atacar y solventar incidentes antes que estos crezcan y se conviertan en una afectación a un servicio de TI donde cada minuto se vuelve cada vez más crítico.
- 5) Tal como se muestra en los dos experimentos realizados; el uso de la inteligencia artificial, en este caso redes neuronales, supone un punto de ventaja no solo en la detección de posibles riesgos o fallos (ya sean estos ineficiencias internas o afectaciones externas) sino también en el tratamiento de forma óptima ya que al apoyarse en estos sistemas, un gestor de TI tendrá a su disposición un soporte tecnológico que encamina o resalta los factores más relevantes y de mayor impacto de forma eficaz y eficiente.

5.2. Recomendaciones

- 1) Los diferentes estándares internacionales como COBIT, ITIL, ISO27001 y demás son sumamente útiles como guías para una integración TI – Negocio y por ende la generación de valor para la misma, así como el énfasis en la seguridad y aseguramiento de los servicios; por lo que su conocimiento es algo recomendado fuertemente para un gestor de tecnologías de información.
- 2) Uno de los factores más importantes y que servirá de base para cualquier análisis futuro, es la información que servirá como entrada. Es de vital importancia que nuestros sistemas cuenten con información detallada y verídica de los eventos presentes y pasados ya que la exactitud o incluso la propia viabilidad de un análisis será dependiente de esto.
- 3) Un análisis mediante Inteligencia Artificial, en este caso Redes Neuronales; no es el único método para resolver un problema. Tal como se muestra en el apartado 2.6 del Marco Teórico, existen muchas metodologías que pueden dar respuesta, entre estas técnicas están los métodos matemáticos, numéricos, estadísticos, lógicos, computacionales por mencionar algunos. Para este trabajo se determinó que una Red Neuronal podría satisfacer la necesidad, sin embargo, queda a criterio del analista la técnica a utilizar.
- 4) La optimización en TI y su aseguramiento del servicio puede darse de muchas formas. En este trabajo se atacaron desde perspectivas internas (eficiencia) y externas (fuera del área), sin embargo, las técnicas de Inteligencia Artificial pueden aplicarse sobre muchos análisis distintos en muchas empresas distintas ya que en toda empresa existen puntos de mejora y algunos de ellos pueden ser abordados utilizando estas técnicas por lo que gran parte queda a la creatividad del investigador.

BIBLIOGRAFÍA

- Battini, D. (04 de 07 de 2018). *Tech Quantum*. Recuperado en Diciembre de 2018, de Tech Quantum: <https://www.tech-quantum.com/need-for-an-artificial-brain/>
- BusiTelCe. (17 de 07 de 2015). *BusiTelCe*. Recuperado en Febrero de 2019, de <http://www.busitelce.com/data-warehousing/13-etl-process-in-data-warehouse>
- Caparrini, F. (23 de 04 de 2018). *Universidad de Sevilla*. Recuperado en Enero de 2019, de Departamento de Ciencias de la Computación e Inteligencia Artificial: <http://www.cs.us.es/~fsancho/?e=165>
- Computer World UK. (10 de July de 2018). *Computer World UK*. Recuperado en Enero de 2019, de <https://www.computerworlduk.com/galleries/infrastructure/top-software-failures-recent-history-3599618/>
- Dean, J. (02 de 05 de 2019). *Google Brain Team*. Recuperado en Abril de 2019, de <https://es.scribd.com/document/355752799/Jeff-Dean-s-Lecture-for-YC-AI>
- Financial Times. (28 de May de 2017). *Financial Times*. Recuperado en Octubre de 2018, de <https://www.ft.com/content/270563ee-43b9-11e7-8d27-59b4dd6296b8>
- Fulton, J. (16 de 08 de 2018). *Towards Data Science*. Recuperado en Enero de 2019, de <https://towardsdatascience.com/emulating-logical-gates-with-a-neural-network-75c229ec4cc9>
- Gómez, E. Z. (07 de 07 de 2015). *Aplicaciones de las Redes Neuronales Artificiales*. Recuperado en Febrero de 2019, de <https://www.youtube.com/channel/UCCFT304Nz5pSWktsiidUy0Q>
- Han, J. (2012). *Science Direct*. Recuperado en Diciembre de 2018, de <https://www.sciencedirect.com/topics/computer-science/backpropagation-algorithm>
- Hof, R. D. (2013). *MIT Technology Review*. Recuperado en Diciembre de 2018, de <https://www.technologyreview.com/s/513696/deep-learning/>

Human Memory. (24 de 01 de 2019). *Human Memory*. Recuperado en Febrero de 2019, de Human Memory: http://www.human-memory.net/brain_neurons.html

ISACA. (20 de 11 de 2018). *ISACA.org*. Recuperado en Noviembre de 2018, de ISACA.org: <https://m.isaca.org/COBIT/Documents/COBIT5-Introduction-Spanish.ppt>

Kapoor, K. (10 de 12 de 2016). *Quora*. Recuperado en Diciembre de 2018, de <https://www.quora.com/What-is-the-sigmoid-function-and-what-is-its-use-in-machine-learning-neural-networks-How-about-the-sigmoid-derivative-function>

Lorenz, L. (07 de 03 de 2019). *Distric Data Labs*. Recuperado en Marzo de 2019, de <https://www.districtdatalabs.com/nlp-research-lab-part-3-forward-propagation-1>

Matematicas 10. (09 de 02 de 2017). *Matematicas 10*. Recuperado en Octubre de 2018, de <https://www.matematicas10.net/2017/02/ejemplos-de-distribucion-normal.html>

Real Academia Española. (20 de 11 de 2018). *RAE*. Recuperado en Noviembre de 2018, de RAE: <http://dle.rae.es/srv/fetch?id=WT8tAMI>

Rigante, F. (10 de 10 de 2014). *Magazciturum*. Recuperado en Febrero de 2019, de Magazciturum: http://www.magazciturum.com.mx/?p=2722#.W_TJF1xsmUk

Sampieri, R. H. (2014). *Metodología de la investigación*. MCGRAW-HILL.

Science Daily. (24 de 01 de 2019). *Science Daily*. Recuperado en Enero de 2019, de Science Daily: https://www.sciencedaily.com/terms/artificial_intelligence.htm

Stergiou, C., & Siganos, D. (28 de 03 de 2019). *Imperial College London*. Recuperado en Marzo de 2019, de Department of Computing: https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html

Torres, J. (21 de 09 de 2018). *Towards Data Science*. Recuperado en Octubre de 2018, de Towards Data Science: <https://towardsdatascience.com/how-do-artificial-neural-networks-learn-773e46399fc7>