



**AUTOMATIZACIÓN DE RESTAURACIÓN DE ARCHIVOS  
PARA DISPOSITIVOS DE ALMACENAMIENTO NAS  
CON MÓDULO DE RED NEURONAL**

**TESIS**

PARA OBTENER EL GRADO DE

**MAESTRO EN  
SISTEMAS INTELIGENTES MULTIMEDIA**

PRESENTA

**ING. SAÚL ESQUIVEL GARCÍA  
ASESOR: DR. ÓSCAR HÉRNANDEZ URIBE**

ZAPOPAN, JALISCO, NOVIEMBRE 2020.

# CARTA LIBERACIÓN DEL ASESOR



Santiago de Queretaro, 9 de NOVIEMBRE del 2020.

Mtro. Geovany González Carlos  
Coordinador Académico de Posgrado  
CIATEQ, A.C.

El abajo firmante, miembro del Comité Tutorial del Ingeniero Saúl Esquivel García, una vez revisado su Proyecto Terminal de tesis, titulado **"AUTOMATIZACIÓN DE RESTAURACIÓN DE ARCHIVOS PARA DISPOSITIVOS DE ALMACENAMIENTO NAS CON MÓDULO DE RED NEURONAL"** autorizo que el citado trabajo sea presentado por el alumno para su revisión, con el fin de alcanzar el grado de **Maestría**.

Sin otro particular por el momento, agradezco la atención prestada.

---

Dr. Óscar Hernández Uribe  
HEUO720115HQTRRS01  
Asesor Académico

# CARTA LIBERACIÓN DEL REVISOR

Querétaro, Qro., 13 de Noviembre del 2020.

Dra. María Guadalupe Navarro Rojero  
Directora de Posgrado  
CIATEQ, A.C.

Por medio de la presente me dirijo a usted en calidad de Revisor del proyecto terminal del alumno **SAÚL ESQUIVEL GARCÍA**, cuyo título es:

**"AUTOMATIZACIÓN DE RESTAURACIÓN DE ARCHIVOS PARA DISPOSITIVOS DE ALMACENAMIENTO NAS CON MÓDULO DE RED NEURONAL"**

Después de haberlo leído, corregido e intercambiado información con el alumno, y realizado los cambios que le fueron sugeridos, puede ser autorizada su impresión, a fin de que se inicien los trámites correspondientes para su defensa.

Sin otro particular por el momento, y en espera de que mis sugerencias sean tomadas en cuenta en beneficio del estudiante y la Institución, agradezco la atención prestada.

Atentamente,

M.I. Zaida Antonieta Mora Alvarez



---

Grado y nombre del Revisor

## **AGRADECIMIENTOS**

Gracias a Dios. Gracias a los que estuvieron, están y estarán.

## RESUMEN

La administración y el soporte de la infraestructura empresarial de almacenamiento (storage en inglés) de tecnologías de la información (TI) se está volviendo cada vez más complicado. Los ingenieros de storage tienen que enfrentarse todos los días a desafíos técnicos para garantizar la disponibilidad y el rendimiento de los datos para usuarios e instancias virtuales. Además, los requisitos de almacenamiento son diferentes para cada unidad de negocio y los equipos de ingeniería de storage deben lidiar con sistemas de almacenamiento de múltiples proveedores. Cierta grupo de soporte suele recibir en sus órdenes de trabajo numerosos tickets de tareas de restauración por parte de usuarios finales que eliminaron por error archivos o carpetas importantes. La alta repetitividad de las tareas de restauración puede ser peligrosa para el ingeniero de storage porque varias restauraciones involucran archivos y carpetas grandes con nombres similares (debido a la convención de nomenclatura empresarial), y el tedio puede llevar al ingeniero a reducir el enfoque y aumentar el error humano. Se ha desarrollado una automatización inteligente, capaz de analizar el texto del ticket y realizar las tareas de restauración repetitivas, las cuales consumen gran tiempo del horario laboral del ingeniero y además alivia la carga de trabajo del equipo de soporte.

**Palabras clave:** Network attached storage, Distributed file system, Automation.

## **ABSTRACT**

IT storage enterprise infrastructure management and support is becoming more and more complicated. Engineers have to face every day with technical challenges to ensure the availability and performance of the data for users and virtual instances. In addition, storage requirements are different for every single business unit and storage support teams have to deal with multivendor storage systems. Certain storage support group used to receive in their ticket queue numerous restore tasks from end users which wrongly deleted important files or folders. The high repetitiveness of restore tasks can be dangerous for the storage engineer because several restores involves larges files and folders with similar names (because of the business naming convention), and the tediousness may lead the engineer to lower the focus and increase human error. An intelligent automation, capable to analyze text and perform repetitive large time consuming restore tasks has been developed to alleviate the workload of the storage support group.

**Keywords:** Network attached storage, Distributed file system, Automation.

# ÍNDICE DE CONTENIDOS

|   |      |
|---|------|
| RESUMEN.....  | ii   |
| ABSTRACT .....  | iii  |
| ÍNDICE DE CONTENIDOS.....                                 | iv   |
| ÍNDICE DE FIGURAS .....                                   | VI   |
| ÍNDICE DE TABLAS.....                                     | VII  |
| GLOSARIO .....  | VIII |
| CAPÍTULO 1. INTRODUCCIÓN .....                            | 1    |
| 1.1 ANTECEDENTES .....                                    | 1    |
| 1.2 DEFINICIÓN DEL PROBLEMA .....                         | 2    |
| 1.3 JUSTIFICACIÓN .....                                   | 3    |
| 1.4 OBJETIVOS.....  | 4    |
| 1.4.1 Objetivo general.....                               | 4    |
| 1.4.2 Objetivos específicos.....                          | 4    |
| 1.5 HIPÓTESIS .....                                       | 5    |
| Capítulo 2. MARCO TEÓRICO .....                           | 6    |
| 2.1 STORAGE .....   | 6    |
| 2.1.1 Fundamentos de almacenamiento .....                 | 6    |
| 2.1.2 NAS .....   | 9    |
| 2.1.3 Resaldos y restauraciones basados en snapshots..... | 11   |
| 2.1.4 Sistema de archivos distribuidos (DFS) .....        | 14   |
| 2.2 SISTEMA DE TICKETS BASADO EN ITSM .....               | 16   |
| 2.3 AUTOMATIZACIÓN .....                                  | 18   |
| 2.3.1 REST API.....                                       | 18   |
| 2.3.2 PowerShell.....                                     | 22   |
| 2.3.3 Red Neuronal.....                                   | 24   |
| 2.4 MONOTONÍA Y ACTIVIDADES REPETITIVAS .....             | 28   |
| 2.5 TRABAJOS RELACIONADOS .....                           | 28   |
| Capítulo 3. PROCEDIMIENTO .....                           | 30   |
| 3.1 CONFIGURACIÓN DE ALMACENAMIENTO .....                 | 30   |

|   |    |
|---|----|
| 3.2 SERVIDOR, PYTHON .....  | 30 |
| 3.3 EXTRACCIÓN DE INFORMACIÓN DE LOS TICKETS DE LA HERRAMIENTA ITSM ..... | 31 |
| 3.4 VALIDACIÓN DEL PATH DFS .....   | 32 |
| 3.5 COMANDOS NAS .....  | 34 |
| 3.5.1 Snapshots disponibles .....   | 34 |
| 3.5.2 Búsqueda del archivo dentro del snapshot.....                       | 34 |
| 3.5.3 Creación de carpeta alterna y comando de restauración .....         | 35 |
| 3.6 NOTIFICACIÓN DE CORREO ELECTRÓNICO Y RESOLUCION DE TICKET.....        | 36 |
| 3.7 MÓDULO DE RED NEURONAL .....  | 37 |
| Capítulo 4. RESULTADOS.....   | 40 |
| 4.1 RED NEURONAL .....  | 40 |
| 4.2 AHORRO DE TIEMPOS DE LA AUTOMATIZACIÓN .....                          | 41 |
| CONCLUSIONES .....  | 43 |
| APORTACIÓN DE LA TESIS.....   | 44 |
| APORTACIÓN SOCIAL DE LA TESIS .....                                       | 45 |
| RECOMENDACIONES.....  | 46 |
| REFERENCIAS .....   | 47 |
| ANEXOS.....   | 50 |



## ÍNDICE DE FIGURAS

|  |    |
|--|----|
| Figura 1. Ciclo virtuoso de la información [2] .....                                   | 7  |
| Figura 2. Tipos de datos [2] .....   | 8  |
| Figura 3. Servidor de propósito general vs dispositivo NAS .....                       | 10 |
| Figura 4. Arquitectura de conectividad NAS.....  | 11 |
| Figura 5. Arquitectura de funcionamiento del snapshot [5] .....                        | 12 |
| Figura 6. Estructura del namespace de DFS [8].....                                     | 15 |
| Figura 7. Procesos operacionales de infraestructura de TI del framework ITIL [9] ..... | 17 |
| Figura 8. Búsquedas en Google XML vs JSON [18] .....                                   | 22 |
| Figura 9. PowerShell ISE.....  | 24 |
| Figura 10. Modelo de neurona artificial de McCulloch & Pitts.....                      | 25 |
| Figura 11. Red unicapa feedforward.....  | 27 |
| Figura 12. Chrome controlado por Selenium [33] .....                                   | 31 |
| Figura 13. CLI Información del ticket.....   | 32 |
| Figura 14. Cmdlet 'Get-DfsnFolderTarget' [34] .....                                    | 32 |
| Figura 15. Menú de selección del snapshot .....  | 34 |
| Figura 16. Menú de selección del archivo .....   | 35 |
| Figura 17. DFS Path de la carpeta destino de la restauración.....                      | 36 |
| Figura 18. Notificación de correo electrónico.....                                     | 36 |
| Figura 19. Datos de entrenamiento de red neuronal.....                                 | 38 |
| Figura 20. Arquitectura de la red neuronal.....  | 39 |
| Figura 21. Iteraciones de entrenamiento de la red neuronal .....                       | 40 |
| Figura 22. Clasificación con DFS path valido.....                                      | 41 |
| Figura 23. Clasificación con DFS path erróneo .....                                    | 41 |
| Figura 24. Gráfica de tiempos manual restore vs automation restore .....               | 42 |

## ÍNDICE DE TABLAS

|  |    |
|--|----|
| Tabla 1. Verbos HTTP REST .....  | 20 |
| Tabla 2. Path DFS segmentado.....  | 37 |
| Tabla 3. Registro de tiempos de restore manual en ambiente controlado..... | 42 |

## GLOSARIO

*ACL*: Access Control List – Lista de Control de Acceso. Lista de control que proporciona permisos (control, lectura, escritura, borrado) a un archivo o directorio. El ACL establece estos permisos a un único usuario o grupo de usuarios.

*ANN*: Artificial Neural Network – Red Neuronal Artificial: Función flexible que adapta de forma autónoma su comportamiento para satisfacer en mayor medida la relación entre las entradas y los resultados esperados [22].

*API*: Application Programming Interface – Interfaz de Programación de Aplicación. Es una serie de rutinas usadas por una aplicación para gestionar generalmente servicios de bajo nivel.

*BPNN*: Back Propagation Neural Network – Red Neuronal de Propagación Reversa. Red neuronal que sintoniza sus pesos sinápticos basado en el rango de error de la época previa.

*CIFS*: Common Internet File System – Sistema de Archivos de Internet Común. Es un estándar de sistema de archivos diseñado para compartir archivos a través de Internet. Es parte del protocolo SMB, que permite que múltiples tipos de computadoras compartan datos y periféricos en más de una red.

*CLI*: Command Line Interface – Interfaz de Línea de Comandos. Programa que permite a los usuarios escribir comandos de texto instruyendo a la computadora o sistema para que realice tareas específicas.

*CmdLet*: Nombre de los comandos utilizados en PowerShell basados en la combinación verbo-nombre.

*CNN*: Convolutional Neural Network – Red Neuronal Convolutiva. Red Neuronal comúnmente utilizada para el reconocimiento de imágenes debido a su estructura de muestreo segmentado.

*COCOMO*: Cost Constructive Model – Modelo de Contrucción de Costos. Herramienta de estimación de esfuerzos, tiempos y finanzas implementado en el desarrollo de proyectos de software.

*DBMS*: Database Management System –Sistema de Administración de Base de Datos. Es un conjunto de programas que permiten la completa administración de base de datos.

*DFS*: Distributed File System – Sistema de Archivos Distribuidos. Tecnología de Windows, organización lógica de drives compartidos, el cual basa su estructura jerárquica en un espacio de nombres (namespace). Al ser una instancia lógica proporciona redundancia y mejor rendimiento.

*DR*: Disaster Recovery – Recuperación de Desastre. Parte de BCP (Business Continuity Planning), DR proporciona planes y procedimientos en caso de desastre para la recuperación de la continuidad del servicio TI.

*DNS*: Domain Name System – Sistema de Nombre de Dominio. Sistema de bases de datos distribuidas en la red con la función principal de traducir la solicitud de ciertos nombres de host a números de IP específicos.

*DPS*: Data Protection Service – Servicio de Protección de Datos. Servicio de respaldo y restauración de archivos, cuyo manejo involucra políticas de retención, horarios de ejecución de los dispositivos de respaldo, entre otros.

*FC*: Fibre Channel – Canal de Fibra. Es una tecnología basada en fibra óptica desarrollada para transmisión de datos a alta velocidad entre superordenadores, servidores de altas prestaciones y dispositivos de almacenamiento.

*File System*: Sistema de Archivos. Sistema de gestión de almacenamiento, que estructura y organiza la escritura, búsqueda, lectura, edición y eliminación de archivos.

*Firmware*: Tipo de software que es capaz de proporcionar un control de bajo nivel específico para circuitos electrónicos.

*GUI*: Graphical User Interface – Interfaz Gráfica de Usuario. Entorno visual de imágenes y objetos mediante el cual una máquina y un usuario interactúan.

*ITIL*: Information Technologies Infrastructure Libraries - Biblioteca de Infraestructura de Tecnologías de Información. Metodología de gestión que propone una serie de prácticas estandarizadas para a mejorar la prestación de los servicios de tecnologías de la información.

*ITSM*: Information Technologies Service Management – Gestión de Servicios de Tecnologías de Infomación. Enfoque estratégico para el diseño, entrega, gestión y mejora de la forma en que la tecnología de la información (TI) se utiliza dentro de una organización.

*JSON*: JavaScript Object Notation – Notación de Objetos de JavaScript. Formato de texto ligero diseñado para el intercambio de información.

*LDAP*: Lightweight Directory Access Protocol – Protocolo Ligero de Acceso a Directorios. Define una serie de operaciones para la consulta de protocolo y, el método de crear y obtener datos a través de estas consultas. Soporta autenticación con la librería SASL y el protocolo SSL.

*LVM*: Logical Volume Manager – Gestor de Volúmenes Lógicos. Método de localización del espacio disco duro en volúmenes lógicos que pueden ser fácilmente redimensionados en vez de particiones.

*NAS*: Network Attached Storage – Almacenamiento Conectado en Red. Sistema empresarial de almacenamiento digital basado en archivos.

*NDMP*: Network Data Management Protocol – Protocolo de Gestión de Datos de Red. Protocolo que permite el tráfico de datos para respaldos como recuperación de archivos enfocado a dispositivos de backup empresariales. Fue desarrollado por las compañías EMC y NetApp.

*NIS*: Network Information Service – Servicio de Información de Red. Servicio utilizado por administradores UNIX con el objetivo de gestionar bases de datos distribuidas en una red.

*Perceptron*: Red neuronal usada para la clasificación de un tipo especial de patrones, los linealmente separables.

*SAN*: Storage Area Network – Red de Área de Almacenamiento. Sistema empresarial de almacenamiento digital basado en bloques.

*SLA*: Service Level Agreement – Acuerdo de Nivel de Servicio. Acuerdo de trabajo y tiempos de respuesta dado en la relación entre un proveedor de servicios y el cliente.

*NLP*: Natural Language Processing – Procesamiento de Lenguaje Natural. Rama de la inteligencia artificial que ayuda a las computadoras a entender, interpretar y manipular el lenguaje humano.

*Shell*: Programa a través del cual un usuario se comunica con el sistema operativo.

*SMTP*: Simple Mail Transfer Protocol – Protocolo de Transferencia Simple de Correo. Protocolo usado para la transferencia de correo electrónico entre computadoras.

# **CAPÍTULO 1. INTRODUCCIÓN**

## **1.1 ANTECEDENTES**

El soporte de la infraestructura de las tecnologías de la información (TI) a nivel empresarial, compuesta por servidores, dispositivos de red, aplicativos de software, etc., es la clave principal del éxito para cualquier empresa. El soporte de la tecnología de almacenamiento (storage en inglés) es uno de los servicios más importantes en la industria de TI, la información debe estar disponible para los usuarios todo el tiempo, manteniendo siempre su integridad y confidencialidad. Además, la complejidad técnica se eleva al grupo de soporte de storage, ya que las arquitecturas de TI suelen desplegar dispositivos de almacenamiento de múltiples proveedores para diferentes propósitos, como el almacenamiento en bloque (SAN), almacenamiento de archivos (NAS) y respaldo (DPS), de acuerdo a las necesidades de cada unidad de negocio.

Los equipos de soporte de storage se comprometen a mantener estables los sistemas de almacenamiento de TI y brindan el mejor servicio profesional a la empresa. Los ingenieros están involucrados en las actividades clave de infraestructura como la planificación de la capacidad de los dispositivos storage, coordinación con otros equipos de trabajo para la actualización de software y firmware de los dispositivos storage, migración de datos y planificación de la continuidad del negocio para la recuperación de desastres (Disaster Recovery), entre otros.

Con el fin de realizar un seguimiento de las actividades diarias, incidentes, cambios y problemas en la infraestructura de TI, las empresas implementan sistemas de tickets los cuales además del seguimiento y organización de soporte en diferentes grupos de trabajo especializados, recuperan el nivel habitual de funcionamiento y minimizan la disrupción del elemento afectado, facilitando un canal de comunicación y por lo tanto habilitando soluciones (gestión de incidentes). También alinean los procesos de negocio y la infraestructura de TI, aumentan la eficiencia y la productividad, brindan información y estadísticas del servicio, tiempos de respuesta y cumplimiento de los acuerdos establecidos a nivel directivo (Service Level Agreement).

## 1.2 DEFINICIÓN DEL PROBLEMA

Cierto grupo de soporte de storage recibe diariamente, en su sistema de tickets, órdenes de trabajo para recuperar archivos y carpetas borrados accidentalmente por usuarios finales tanto en unidades de almacenamiento personal como compartidos (shared drive). Este storage se proporciona a los usuarios finales a través de dispositivos storage NAS. Los snapshots son copias instantáneas de respaldo tomadas en un horario programado y es uno de los métodos de protección de respaldo preferidos en los dispositivos NAS porque utilizan una capacidad mínima en el volumen, no hay costos de rendimiento y los datos se pueden restaurar rápidamente. La restauración de archivos y carpetas es una tarea clave, el equipo de soporte de storage debe proceder con rapidez y cautela en las restauraciones durante las ventanas de mantenimiento establecidas, ya que dicha actividad tiene un gran impacto en la continuidad del negocio de los usuarios finales. Este ha sido el proceso de restauración ejecutado por el ingeniero de storage:

1. Desde el sistema de tickets, validar si la ruta (path) del recurso a restaurar (carpeta o archivo) proporcionada por el usuario final es correcta (sin errores sintácticos).
2. Extraer del path el shared drive destino del sistema de archivos distribuido (DFS) y la ubicación del volumen físico.
3. Los volúmenes físicos residen en el dispositivo NAS, iniciar sesión en el dispositivo correspondiente y localizar el volumen.
4. Seleccionar el volumen y el snapshot de acuerdo a la fecha de restauración solicitada por el usuario final en el ticket.
5. Dentro del snapshot se supone que debe estar localizado el archivo antes de que ocurriera la eliminación. Navegar dentro de las carpetas del snapshot y localizar el archivo o carpeta solicitada, ejecutar el comando "restaurar".
6. Validar la existencia del archivo después de que se complete la tarea de restauración. Notificar al usuario final por correo electrónico y cerrar el ticket.

Si el usuario final ha enviado correctamente el path y la fecha de restauración en el formulario del ticket, el proceso anterior le toma al ingeniero de storage alrededor de 13



minutos en completarse. Si se detectan inconsistencias en el ticket, el tiempo de ejecución de la restauración será mayor.

Los tickets del equipo de soporte de storage suele recibir al menos 4 tareas de restauración por día. En un día de alta carga de trabajo se reciben hasta 12 tareas. Esto significa que, durante un día normal, el ingeniero de almacenamiento se centra en las actividades de restauración al menos 45 minutos. Este tiempo puede ser utilizado por el ingeniero en diferentes actividades de infraestructura de almacenamiento de alto impacto o en capacitación y entrenamiento constante.

La alta repetitividad de las tareas de restauración puede ser peligrosa para el ingeniero, ya que varias tareas de restauración involucran numerosos archivos y carpetas con nombres similares, esto se debe a la convención de nomenclatura empresarial, y el tedio puede llevar al ingeniero a reducir el enfoque y aumentar el error humano.

Este trabajo presenta una automatización inteligente programada en lenguaje Python que se conecta al sistema de tickets, extrae solo los tickets de restauración activos, el texto del ticket es procesado y clasificado por una Red Neural Artificial (ANN), un comando de restauración es generado y ejecutado y finalmente un algoritmo de correo electrónico notifica al usuario final que su archivo o carpeta fue restaurado exitosamente.

### **1.3 JUSTIFICACIÓN**

El impacto de la revolución informática está en nuestro día a día, la gente genera información tomando fotografías, enviando mensajes instantáneos, compartiendo documentos vía correo electrónico, realizando movimientos bancarios etcétera. La información generada cada segundo se ha convertido en el activo más importante para las empresas, por lo que en la última década se han realizado grandes inversiones en tecnología de almacenamiento. Administrar, almacenar y archivar todos los datos es un desafío continuo para los ingenieros de storage porque deben garantizar la disponibilidad, seguridad, escalabilidad, integridad, capacidad y capacidad de administración de los datos.

Los equipos de almacenamiento de TI suelen respaldar una infraestructura de TI grande y compleja, esto a menudo involucra dispositivos de múltiples proveedores:

- Almacenamiento a nivel bloque (SAN)
- Almacenamiento a nivel de archivo (NAS)
- Directores y conmutadores de canal de fibra (FC)

Una automatización inteligente capaz de realizar las tareas repetitivas de restauración de archivos es necesaria para que los equipos de soporte de storage les permita centrarse en actividades de infraestructura clave como:

- Planificación e implementación, migración, gestión y soporte de nuevos dispositivos de storage.
- Planificación de migración y continuidad empresarial. Informes ejecutivos.
- Controles de calidad, gestión de capacidad y rendimiento para mantener la estabilidad del ambiente de TI.
- Garantizar la disponibilidad, confiabilidad, escalabilidad y capacidad de administración del entorno del storage.
- Responder a los incidentes con la prioridad adecuada, proporcionando una resolución rápida e investigación de la causa raíz de la falla.

## **1.4 OBJETIVOS**

### **1.4.1 Objetivo general**

Crear una automatización inteligente capaz de restaurar archivos y carpetas NAS según el texto ingresado por el usuario en el ticket.

### **1.4.2 Objetivos específicos**

- Implementar una conexión a la base de datos del sistema de tickets, filtrar los tickets del equipo de storage y extraer solo los tickets de restauración.
- Crear una red neuronal para procesar el texto insertado por el usuario en el campo path, validar la existencia del path.

- Diseñar una interface de línea de comandos (CLI) para el ingeniero de storage que supervise la ejecución automatización.
- Realizar un algoritmo que genere un script para el dispositivo NAS, el cual encontrará el archivo a restaurar y pedirá a través de la CLI al ingeniero de storage una aprobación final del comando de restauración mostrando los datos obtenidos del ticket, así como alertas de capacidad para archivos mayores a 1 GB.
- Elaborar una función de envío de correo electrónico que notificará al usuario final si la restauración fue exitosa o si algún error fue detectado en el ticket según sea el caso.
- Desplegar una función dentro del código que permita documentar y cerrar el ticket.

## **1.5 HIPÓTESIS**

Es posible proporcionar al equipo de ingenieros de storage una automatización inteligente que, además de reducir el tiempo de operación por ticket de restauración de archivos hasta 70%, permita optimizar el flujo de trabajo y comunicación al notificar al usuario final cuando un error sintáctico es identificado en su ticket.

## **CAPÍTULO 2. MARCO TEÓRICO**

### **2.1 STORAGE**

#### **2.1.1 Fundamentos de almacenamiento**

El almacenamiento de datos es un pilar central de las tecnologías de la información. Los consumidores individuales y corporativos de TI crean en todo momento una gran cantidad de información digital, la cual debe almacenarse en entornos estables que provean la protección necesaria. No hace mucho tiempo, el almacenamiento de la información se consideraba como solo un montón de discos duros o cintas conectados a los dispositivos destinados a almacenar datos sin ninguna optimización u opción de escalabilidad.

Nos hemos convertido en dependientes de la información, accedemos a internet todos los días para realizar búsquedas, participar en redes sociales, enviar y recibir correos electrónicos y decenas de otras aplicaciones [1]. Con un número creciente de dispositivos, más información está siendo creada por los individuos que por empresas. Cuando se crea, la información reside localmente en dispositivos como teléfonos inteligentes, cámaras y laptops [2].

La información creada por individuos gana mayor valor cuando se comparte con otros, para compartir esta información, se necesita cargar a través de la red hacia los centros de datos, la figura 1 describe este círculo virtuoso de la información.

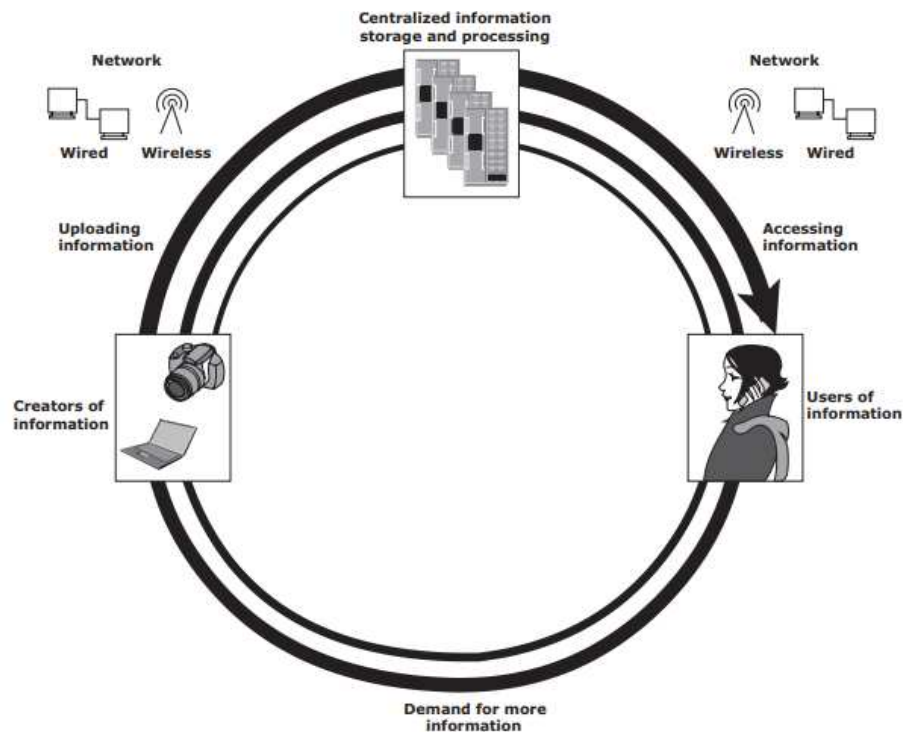


Figura 1. Ciclo virtuoso de la información [2]

Los datos creados por personas o empresas deben almacenarse de modo que sean fácilmente accesibles para su posterior utilización. Los datos se pueden clasificar como estructurados o no estructurados (figura 2). Los datos estructurados se organizan en filas y columnas en un formato rígidamente definido para que las aplicaciones puedan recuperar y procesar la información de manera eficiente. Los datos estructurados generalmente se almacenan mediante la gestión de base de datos sistema (DBMS).

Los datos no son estructurados si sus elementos no se pueden almacenar en filas y columnas, y, por lo tanto, es difícil de consultar y recuperar mediante aplicaciones comerciales. Por ejemplo, la información de contacto de una persona puede almacenarse de varias formas, como mensajes de correo electrónico, archivos de formato .doc, .txt, .pdf, etcétera [2]. Debido a su naturaleza no estructurada, es posible que los datos no estructurados no tengan los componentes necesarios para identificarse de forma única para cualquier tipo de interpretación o procesamiento.

Las empresas se preocupan principalmente por la gestión de la información porque más del 80 por ciento de los datos empresariales no están estructurados y requieren espacio de almacenamiento significativo y mayor esfuerzo de gestión.

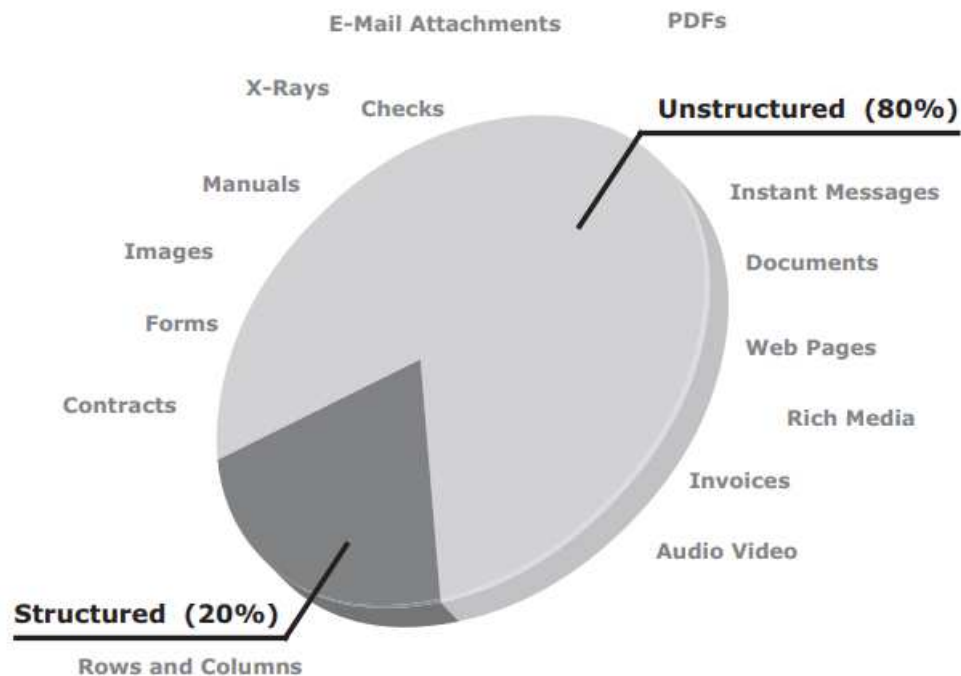


Figura 2. Tipos de datos [2]

En un entorno informático, los dispositivos diseñados para el almacenamiento de datos se denominan comúnmente storage. El tipo de almacenamiento utilizado varía según el tipo de dato y la velocidad de procesamiento requerido. Dispositivos como la memoria de un teléfono celular o una cámara digital, DVD, CD-ROM y los discos duros de las computadoras personales son ejemplos de dispositivos de storage personal.

Como activo de las empresas, el storage se reconoce como uno de los elementos empresariales clave de un centro de datos, un recurso único que necesita su propio enfoque y especialización para su implementación y administración [3].

Las empresas tienen varias opciones disponibles para almacenar datos, incluidos los discos duros internos, arreglos de discos externos (RAID) y cintas magnéticas.

Los datos fluyen desde una aplicación hasta el storage a través de varios componentes, los tres componentes esenciales de este entorno son el host, la red y el storage. Estas entidades, junto con sus componentes de hardware e interfaces lógicas, facilitan el acceso a los datos.

Similar a la forma en que los datos han evolucionado, los propios dispositivos de almacenamiento han experimentado un cambio revolucionario que ha dado como resultado diferentes tipos de almacenamiento cada uno con un propósito específico.

### **2.1.2 NAS**

El NAS (Network Attached Storage) es un dispositivo de almacenamiento que se conecta a una red y proporciona servicios de lectura y escritura de archivos. Tan pronto como un dispositivo NAS es conectado a la red, todas las demás computadoras de la red pueden acceder a él muy fácilmente.

Se accede al NAS a través de una red IP y ofrece acceso simple, pero de alto nivel para compartir archivos. Un dispositivo NAS se compone básicamente de unidades de disco que almacenan archivos y controladores que exportan servicios de acceso a los archivos.

Los dispositivos de almacenamiento NAS están optimizados para funciones de servicio de archivos, como almacenar, recuperar y acceder a archivos. La figura número 3 muestra cómo se puede utilizar un servidor de propósito general para alojar cualquier archivo, debido a que utiliza un sistema operativo genérico (no dedicado exclusivamente al servicio de almacenamiento). A diferencia de un servidor de propósito general, los dispositivos de almacenamiento NAS son diseñados para la optimización del servicio de archivos, además cuenta con un OS en tiempo real cuyo objetivo está enfocado al servicio de directorios y archivos mediante protocolos de estándar abierto [3]. Algunos proveedores de NAS admiten funciones como clústeres nativos para alta disponibilidad y rendimiento.

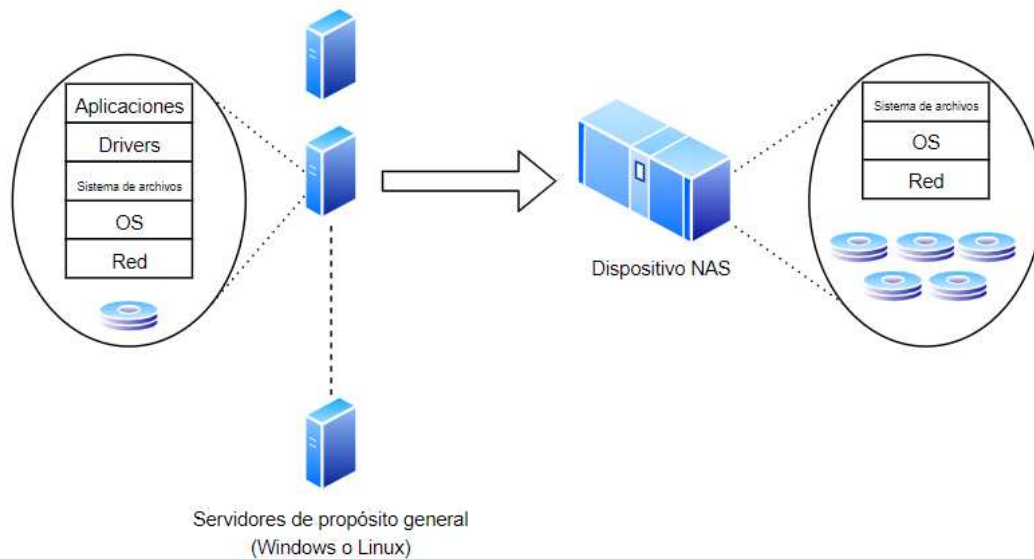


Figura 3. Servidor de propósito general vs dispositivo NAS

Los dispositivos NAS utilizan un sistema de archivos (File System) como forma estructurada de almacenar y organizar archivos de datos [4]. Muchos sistemas de archivos mantienen una tabla de acceso a archivos para simplificar el proceso de búsqueda y acceso a archivos.

En un entorno de intercambio de archivos, el usuario que crea el archivo (el creador o propietario del archivo) determina el tipo de acceso que se dará a otros usuarios (leer, escribir, ejecutar, agregar, eliminar y listar) y controla los cambios en el archivo. Cuando varios usuarios intentan acceder a un archivo compartido al mismo tiempo, se requiere un esquema de protección para mantener la integridad de los datos y, al mismo tiempo, hacer posible este intercambio.

El Protocolo de transferencia de archivos (FTP), los sistemas de archivos distribuidos y un modelo cliente/servidor que utiliza un protocolo de intercambio de archivos son algunos ejemplos de implementaciones de entornos de intercambio de archivos.

En el modelo tradicional cliente/servidor, que se implementa con protocolos de intercambio de archivos, los clientes montan sistemas de archivos remotos que están disponibles en dispositivos NAS o servidores de archivos dedicados, comúnmente derivados de servidores de propósito general.



Los protocolos estándar de intercambio de archivos cliente/servidor son NFS para UNIX y CIFS para Windows (figura 4). NFS y CIFS permiten al propietario de un archivo establecer el tipo de acceso requerido, como solo lectura o lectura-escritura para un usuario o grupo de usuarios en particular.

En ambas implementaciones, los usuarios desconocen la ubicación del sistema de archivos. Además, un servicio de nombres, como el Sistema de Nombres de Dominio (DNS), el Protocolo Ligero de Acceso a Directorios (LDAP) y los Servicios de Información de Red (NIS), ayuda a los usuarios a identificar y acceder a un recurso único de almacenamiento en la red [4].

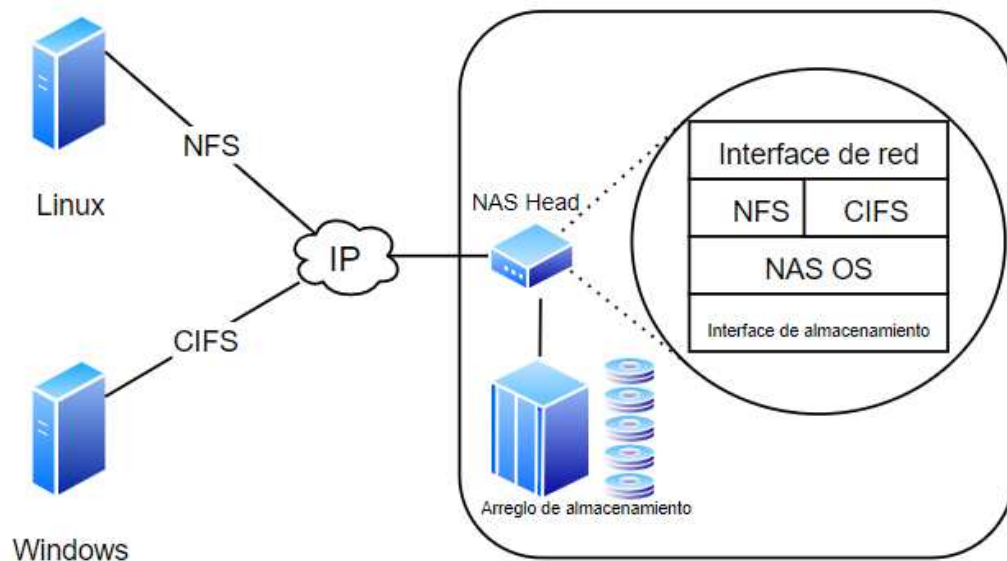


Figura 4. Arquitectura de conectividad NAS

### 2.1.3 Respaldos y restauraciones basados en snapshots

El snapshot es una réplica de datos basada en punteros que requiere una fracción del espacio utilizado por el sistema de archivos original. Este snapshot puede ser gestionado e implementado por el mismo sistema de archivos o por LVM (Logical Volume Manager).

Cuando se crea el snapshot, se crean un mapa de bits y un mapa de bloques en la metadata del sistema de archivos. El mapa de bits se utiliza para realizar un seguimiento

de los bloques que han cambiado en el archivo después de la creación del snapshot. El mapa de bloques se utiliza para indicar la dirección exacta desde la cual se leerán los datos cuando se acceda al snapshot de respaldo. Inmediatamente después de la creación del snapshot, todas las lecturas del snapshot se desplegarán a través del sistema de archivos activo [5].

Un conjunto de datos sin cambios existe físicamente solo una vez, pero se accede tanto por el sistema de archivos como por el snapshot. Si el usuario desea modificar la información, el snapshot debe almacenar la versión original de los datos antes de que el nuevo conjunto de datos se pueda escribir en el sistema de archivos (esto se denomina CoFW "copia en la primera escritura").

Por lo tanto, un snapshot sirve mucho mejor para tareas temporalmente limitadas, como copias de seguridad de respaldo o transferencias de datos [5]. Además, el despliegue de los punteros direccionados hacia el conjunto de datos original garantiza que el snapshot esté disponible de inmediato, en la figura 5 se despliega gráficamente la arquitectura del snapshot.

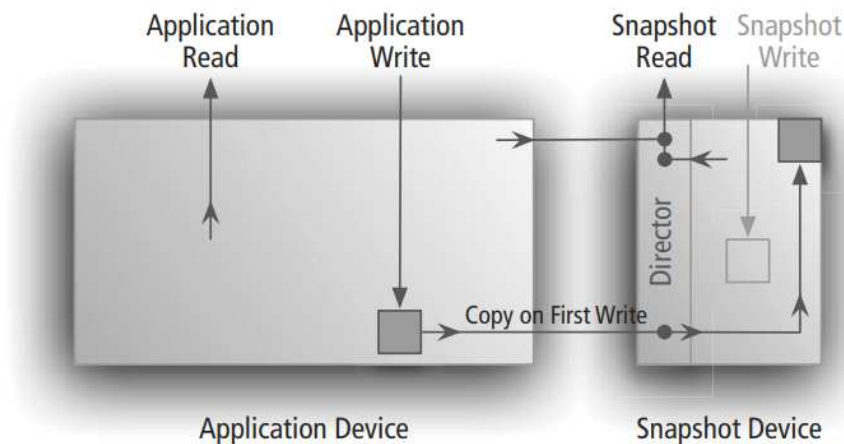


Figura 5. Arquitectura de funcionamiento del snapshot [5]

Algunas de las principales características de los snapshots son:

1. Se puede acceder al snapshot en modo de solo lectura.
2. El snapshot se puede utilizar para la recuperación inmediata de datos.

3. Los snapshots se pueden eliminar fácilmente de forma manual o automatizada de acuerdo a los requerimientos de su implementación.
4. Los snapshots solo requieren, en comparación con el sistema de archivos, solo una pequeña parte de almacenamiento.
5. No es necesaria ninguna sincronización de datos preparatoria, el snapshot está disponible de inmediato.

Tradicionalmente, la recuperación de archivos mediante un snapshot se puede lograr de dos maneras:

1. Monte el snapshot en un punto de montaje (mount point) temporal y copie solo los archivos requeridos en el sistema de archivos de la aplicación que desea recuperar. Una recuperación completa de todos los archivos mediante este procedimiento consumirá un espacio considerable, debido a que los archivos actuales del mount point se guardan dentro del dispositivo del sistema de archivos activo.
2. Desmonte la instancia de almacenamiento del sistema de archivos de la aplicación y monte el snapshot deseado en el mount point de la aplicación. Luego reinicie (y recupere) sus archivos y aplicativos requeridos. Todos los snapshots creados después del montado actualmente perderán su funcionalidad.

Adicionalmente, diversos proveedores de NAS proporcionan la opción de restauración dentro de su interfaz de administración. Con esta característica se puede seleccionar granularmente el archivo a restaurar navegando dentro de las carpetas del snapshot.

Una de las ventajas de este método es la opción de elegir la carpeta destino de restauración, de esta forma se recupera el archivo del snapshot sin sobrescribir el archivo original. Además, se evita el montaje del snapshot como sistema de archivos activo.

## 2.1.4 Sistema de archivos distribuidos (DFS)

DFS crea una perspectiva virtual de directorios compartidos en una red. Funciona creando un espacio de nombres (namespace) con referencias hacia recursos compartidos de red y rastrea el acceso a los archivos [6]. Cuando un usuario solicita un recurso en particular, DFS compara la solicitud con el servidor o dispositivo NAS correcto y dirige al cliente al recurso.

Para el usuario final, las carpetas aparentan estar almacenadas en un solo disco duro. Como administrador del DFS, se tiene la capacidad de decidir qué carpetas compartidas aparecerán en el namespace del DFS. Además, también se tiene la capacidad de diseñar la jerarquía de los directorios dentro del namespace. Convenientemente, la seguridad es reforzada por el sistema de archivos existente y los permisos de carpetas compartidas [7].

En el pasado, la mayor parte de la información se almacenaba localmente en una computadora o centralizadamente en un servidor, pero ahora la tendencia de las compañías a distribuir sus sistemas de archivos a través de su red interna ofrece muchas ventajas:

- El sistema de archivos distribuido refleja la estructura corporativa. Por ejemplo, el Departamento de Ventas puede estar distribuido por todo el mundo, pero la red a la que se conecta es idéntica a la forma en que está organizado el Departamento de Ventas.
- Más fácil de proteger los derechos de acceso en los sistemas de archivos. En un sistema de archivos distribuido, es típico tener una política de seguridad sólida en el sistema de archivos, y cada archivo tendrá un propietario que puede definir los privilegios en este archivo.
- Mayor acceso a fuentes únicas de información. Muchos usuarios pueden tener acceso a una única fuente de información. Tener varias versiones de un archivo puede causar una gran cantidad de problemas, especialmente si no se sabe cuál es la más actualizada.

- Instalaciones de respaldo mejoradas. La computadora de un usuario se puede apagar, pero sus archivos aún se pueden respaldar desde el sistema de archivos distribuido a través de los snapshots.
- Mayor confiabilidad. El sistema de archivos distribuido puede tener una red troncal construida a partir de hardware confiable y robusto, que es virtualmente 100% confiable, incluso cuando hay una falla de energía o cuando hay una falla de hardware que garantiza la alta disponibilidad de la información.
- Sistemas de archivos más grandes. En algunos tipos de sistemas de archivos distribuidos, es posible crear grandes sistemas de archivos a partir de una red de dispositivos de almacenamiento interconectados.

El link de acceso de una carpeta compartida o archivo en un DFS se compone de los elementos de la figura 6 separados por una diagonal invertida '\'.

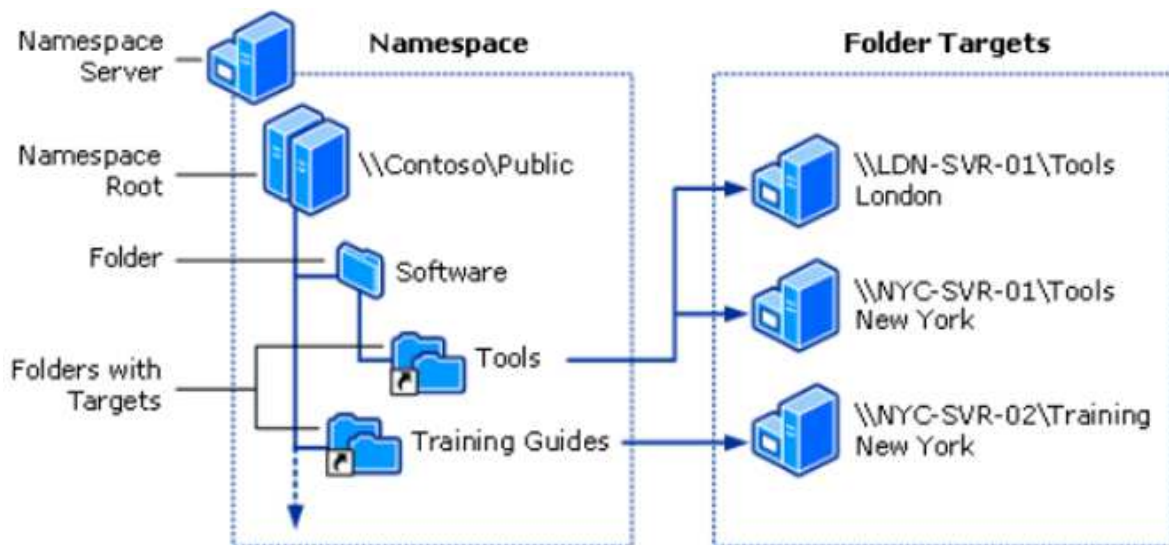


Figura 6. Estructura del namespace de DFS [8]

- Namespace server: un servidor de namespace aloja un espacio de nombres, puede ser un servidor miembro de un clúster o un controlador de dominio.
- Namespace root: la raíz del namespace es el punto de partida del espacio de nombres. En la figura 6, el nombre de la raíz es Public y la ruta del espacio de nombres es \\Contoso\Public
- Carpeta: cuando los usuarios exploran una carpeta que tiene destinos de carpeta en el namespace, el cliente recibe una referencia que lo redirige de forma transparente a la computadora del cliente a uno de los destinos de carpeta.

- Destinos de carpeta: un destino de carpeta es la ruta de la Convención de Nomenclatura Universal (UNC) de una carpeta compartida. El destino de la carpeta es donde se almacenan los datos y el contenido. Un usuario que navega a \\Contoso\Public\Software\Tools se redirige de forma transparente a la carpeta compartida \\LDNSVR-01\Tools o \\NYC-SVR-01\Tools, según el sitio en el que se encuentre actualmente el usuario.

## **2.2 SISTEMA DE TICKETS BASADO EN ITSM**

A lo largo de los años, las TI se han convertido en la columna vertebral de las empresas hasta el punto donde sería imposible para muchas funcionar (y mucho menos tener éxito) sin ellas [9]. Como resultado de su papel cada vez más importante en las corporaciones, la función de las TI está cambiando, transformándose de proveedor de tecnología a un compañero estratégico.

A medida que aumentaban los dispositivos de TI en las infraestructuras empresariales, además del reto que supone la adopción de nuevas tecnologías como virtualización, computo en la nube, etc., la complejidad de la administración de los equipos como los servicios ofrecidos creció hasta convertirse en una necesidad cuya gestión es clave en el cumplimiento de objetivos de las empresas.

La evolución de las organizaciones de TI de proveedores de tecnología a proveedores de servicios, requiere adoptar una perspectiva diferente sobre su gestión teniendo como prioridad el conjunto de procesos que cooperan para garantizar la calidad de los servicios de acuerdo con los niveles de servicio acordados con el cliente [10].

La adopción y uso de mejores prácticas de TI se está volviendo cada vez más común, varios estudios se han centrado en la adopción de la gestión de servicios de TI (ITSM) como un servicio específico orientado a la implementación de las mejores prácticas.

En los últimos años, el uso de estas mejores prácticas encapsuladas en frameworks operativos de TI basados en principios como ITIL (Information Technology Infrastructure Libraries) ha ido en aumento [11].

La Biblioteca de Infraestructura de Tecnología de la Información (ITIL) fue establecida en 1989 por la antigua Agencia Central de Computación y Telecomunicaciones (CCTA) del Reino Unido para mejorar su organización de TI. ITIL consiste en un conjunto interrelacionado de mejores prácticas para reducir el costo, mientras se mejora la calidad de los servicios de TI entregados a los usuarios.

ITIL se organiza en torno a cinco áreas clave: perspectiva empresarial, gestión de aplicaciones, prestación de servicios, soporte de servicios y gestión de infraestructura (figura 7).

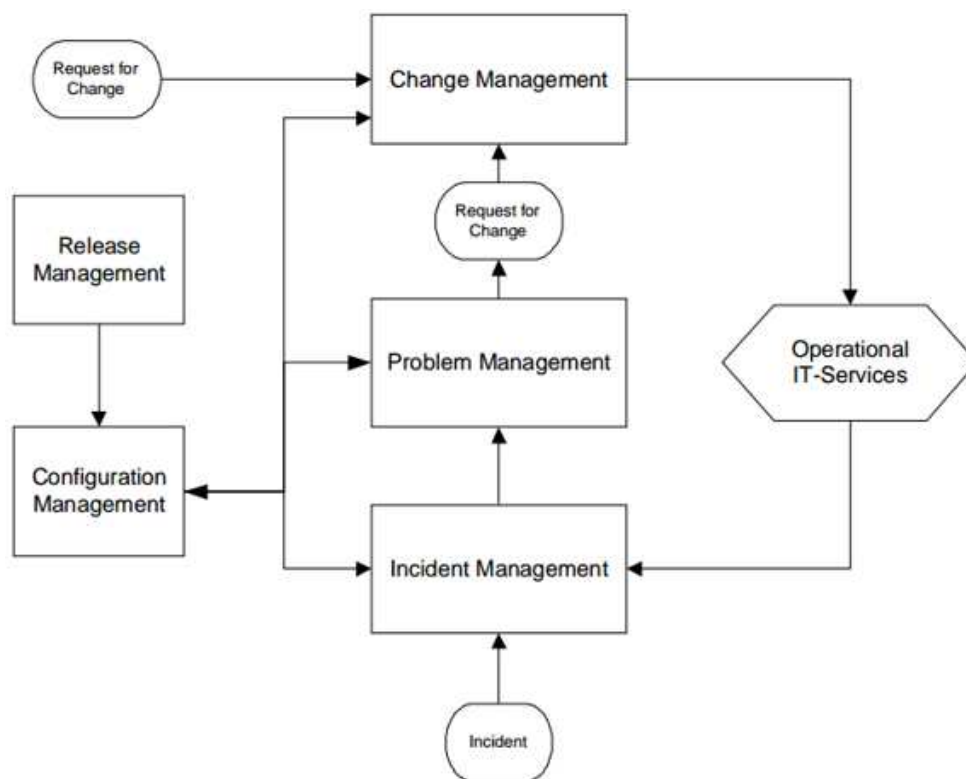


Figura 7. Procesos operacionales de infraestructura de TI del framework ITIL [9]

El valor de los procesos ITIL viene a hacer la vida mucho más fácil para las personas a las que apoya, tratándolas como elementos valiosos que no se preocupan por el agente de soporte o equipo de trabajo participará en el cumplimiento de su solicitud [12].

Estas solicitudes, conocidas comúnmente como tickets son registradas en una base de datos a través de un software dedicado. Estos tickets tienen diferentes categorías de

acuerdo al servicio requerido como ordenes de trabajo, cambios en la infraestructura, reportes y seguimiento de incidencias entre otros [13].

Los tickets correctamente documentados deben permitir que el personal de soporte de un seguimiento fácilmente del requerimiento solicitado por el cliente sin tener que explicarlo nuevamente.

## **2.3 AUTOMATIZACIÓN**

Un atributo común en el desarrollo de proyectos es el deseo de hacer las cosas de forma rápida, coherente y de una sola vez. Para abordar este deseo, a menudo se utilizan secuencias de comandos y automatización, ya que proporcionan un método consistente para completar tareas que ocurren regularmente.

La automatización se define como la operación controlada de un aparato, proceso, software o sistema que reemplazan el trabajo humano [14]. La automatización es deseable porque cuando intentamos implementar manualmente procesos complejos, se pueden cometer errores.

El propósito de la automatización es aliviar el trabajo manual, lento y repetitivo para que el equipo de soporte pueda dedicar su tiempo a actividades más valiosas. La automatización también aumenta la repetibilidad. Cuando un software completa un proceso, elimina el elemento humano y por lo tanto produce resultados más predecibles [15].

La automatización puede parecer un gasto al inicio de un proyecto, pero invertir en automatización pagará dividendos a medida que los sistemas crezcan y se vuelvan más sofisticados.

### **2.3.1 REST API**

Creado por Roy Fielding, uno de los autores principales de la especificación HTTP, REST está destinado a aprovechar los estándares y tecnologías dentro de HTTP [16]. REST



(abreviatura de REpresentational State Transfer) es una arquitectura de estilo definido para ayudar a crear y organizar sistemas distribuidos.

Las API de REST usan verbos HTTP para buscar, obtener, o agregar información sobre los recursos. Por lo tanto, se realizan acciones contra los recursos o, dicho de otra forma, los verbos actúan sobre sustantivos.

La idea principal detrás de la implementación de un sistema distribuido REST, es mejorar en las siguientes áreas:

- Escalabilidad de la interacción de los componentes: cualquier sistema distribuido debe ser capaz de manejar este aspecto, y la interacción simple propuesta por REST lo permite en gran medida.
- Modificabilidad de componentes: la naturaleza distribuida del sistema, y la separación de elementos propuesta por REST, permite que los componentes se modifiquen independientemente uno de otro a un costo y riesgo mínimos.
- Portabilidad: REST es independiente de la tecnología y el lenguaje, lo que significa que puede ser implementado y consumido por cualquier tipo de tecnología. El sistema puede ser utilizado por una gama amplia de desarrolladores.
- Fiabilidad: permite la recuperación más fácil de un sistema después de una falla.

De esta lista, se pueden extrapolar algunos beneficios directos.

- Un diseño centrado en componentes permite crear sistemas tolerantes a fallas. Que la falla de un componente no afecte a la estabilidad completa del sistema es un gran beneficio para cualquier sistema.
- La interconexión de componentes es bastante sencilla, minimizando los riesgos al agregar nuevas funciones o escalar hacia arriba o hacia a los lados.

## Restricciones REST

Para comenzar con el conjunto completo de necesidades del sistema, las restricciones se agregan a los componentes individuales hasta que las fuerzas que influyen en el sistema puedan interactuar en armonía entre sí [17].

Para definir una arquitectura REST, un estado nulo es inicialmente definido —un sistema que no tiene restricciones de ningún tipo y donde la diferenciación de componentes no es existente— y las restricciones se agregan una por una.

## Recursos REST

Los principales componentes básicos de una arquitectura REST son los recursos. Todo lo que se pueda nombrar (sustantivo) puede ser un recurso [17]. Los recursos definen de qué van a tratar los servicios, el tipo de información que se va a transferir, y sus acciones relacionadas. El recurso es la entidad principal de la que nace todo lo demás.

## Access Points REST

Los puntos de contacto entre todas las aplicaciones finales y la API se denominan Access Points [17]. La API debe proporcionarlos para permitir que los clientes accedan a sus funcionalidades. Esto se puede hacer a través del protocolo de comunicaciones que se elija. Estos puntos de acceso deben tener nombres mnemotécnicos fáciles de recordar para ayudar al desarrollador a comprender su propósito con solo leerlos.

## Acciones REST

Tabla 1. Verbos HTTP REST

| Verbo HTTP | Acción   |
|------------|--|
| GET        | Accede a un recurso en modo de solo lectura  |
| POST       | Se utiliza regularmente para enviar un nuevo recurso al servidor (acción de creación)    |
| PUT        | Se utiliza regularmente para actualizar un recurso determinado (acción de actualización) |
| DELETE     | Usado para eliminar un recurso   |

La tabla 1 detalla los verbos HTTP más usados. Se dice que los verbos GET, PUT y DELETE son idempotentes; es decir, llamarlos una y otra vez producirá el mismo resultado sin efectos secundarios adicionales [16]. Por ejemplo, para la acción PUT a un recurso único dado, enviar una solicitud PUT debería actualizar el recurso si ya existe, o si no existe, el sistema debe crear el recurso tal como se envió. En otras palabras, llamar a PUT una y otra vez produce el mismo resultado sin efectos secundarios adicionales

Se dice que la acción GET es segura, esto significa que no se cambia nada en el sistema, lo cual es apropiado para las llamadas HTTP que están destinadas a consultar el sistema por una colección de recursos o por un recurso específico [16].

Es importante que las operaciones GET, PUT y DELETE del servicio sean coherentes con los estándares del protocolo HTTP. Por lo tanto, se debe hacer todo lo posible para garantizar que esas tres acciones se puedan llamar una y otra vez sin errores.

A diferencia de las otras tres acciones, POST no se considera idempotente [16]. Esto se debe a que POST se utiliza para crear una nueva instancia del recurso identificado para cada invocación del método. La llamada a POST dará como resultado nuevas instancias de recursos, una para cada llamada. Esto es apropiado para los casos en los que el sistema debe generar el identificador del nuevo recurso y devolverlo en la respuesta.

### **Idioma de transporte**

JSON ha ido ganando terreno durante los últimos años a XML (figura 8) como formato de transferencia de datos estándar [17]. Esto se debe principalmente a las ventajas que ofrece.

- Es ligero. Hay muy pocos datos en un archivo JSON que no estén directamente relacionados con la información que se transfiere. Esto es una ventaja importante sobre formatos más detallados como XML.
- Es legible por humanos. El formato en sí es tan simple que puede ser leído y escrito por un humano. Esto es particularmente importante considerando el factor humano como punto de enfoque.

- Soporta diferentes tipos de datos. Dado que no todo lo que se transfiere es una cadena, esta función permite al desarrollador proporcionar significado adicional a la información transferida.

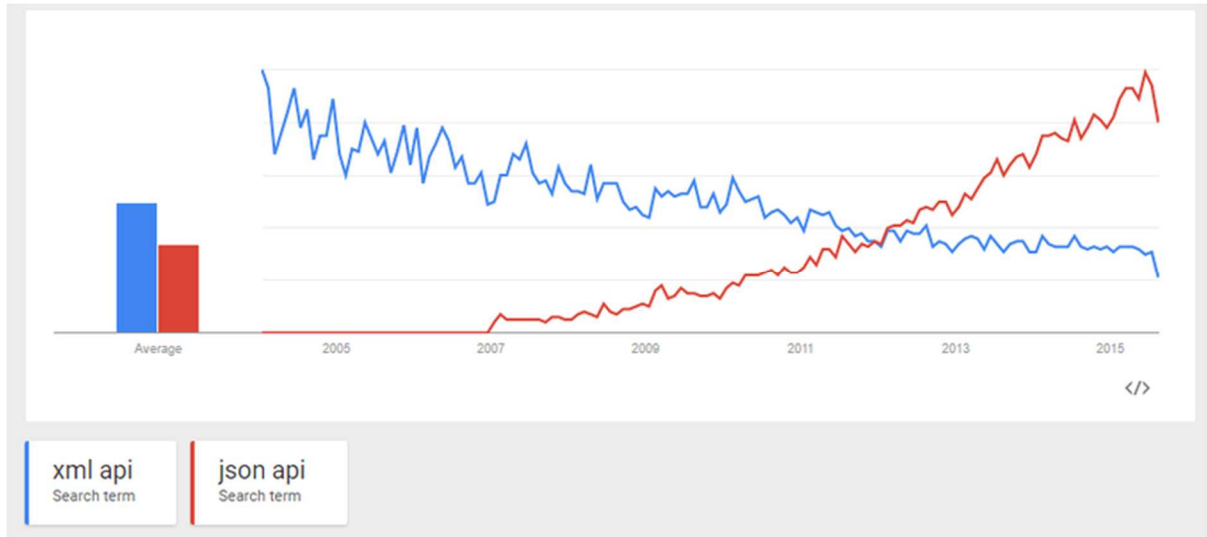


Figura 8. Búsquedas en Google XML vs JSON [18]

### 2.3.2 PowerShell

PowerShell es un framework de Microsoft que incluye un shell de comandos y un lenguaje de secuencias de comandos. PowerShell ha sido utilizado tradicionalmente por administradores de sistemas, equipos de TI e incluso investigadores forenses para obtener acceso a la información operativa sobre las infraestructuras que administran [19].

PowerShell proporciona un gran motor de adquisición para obtener una amplia gama de información de sistemas activos, servidores, periféricos, dispositivos móviles y aplicaciones basadas en datos como Active Directory.

Un CmdLet suele ser un script ligero de PowerShell que realiza una función específica, la cual es ejecutada por un usuario al sistema operativo o a una aplicación para realizar un servicio.

Todos los CmdLets se expresan como un par de verbo-sustantivo y tienen un archivo de ayuda al que se puede acceder usando el par de verbo-sustantivo Get-Help <CmdLet>. Incluso la ayuda es otro CmdLet [19].

Normalmente, PowerShell ya está instalado en las plataformas modernas de escritorio y servidor de Windows. El ISE, Entorno de Scripting Integrado (figura 9) proporciona:

1. Un entorno integrado con ayudas y experimentación de los CmdLets .
2. La capacidad de escribir, probar y depurar scripts.
3. Finalización automática de comandos que aceleran tanto el desarrollo como el aprendizaje.

Debido a la decisión de Microsoft de abrir PowerShell y brindar la capacidad de adquirir información de otras plataformas que no son propios de Microsoft, diversos proveedores de hardware, como storage, han desarrollado sus propios módulos PowerShell para la administración y soporte de sus equipos.

Esto es un gran estímulo para los administradores de storage debido a la facilidad de crear scripts para automatizar procesos manuales aliviando la carga de trabajo. Obtener acceso a estos módulos es un proceso simple al solo descargar el módulo requerido y luego importarlo en el desktop de administración.

A medida que las organizaciones de TI se esfuerzan por lograr una mayor automatización, vale la pena validar las posibilidades de automatización que brindan estos módulos.

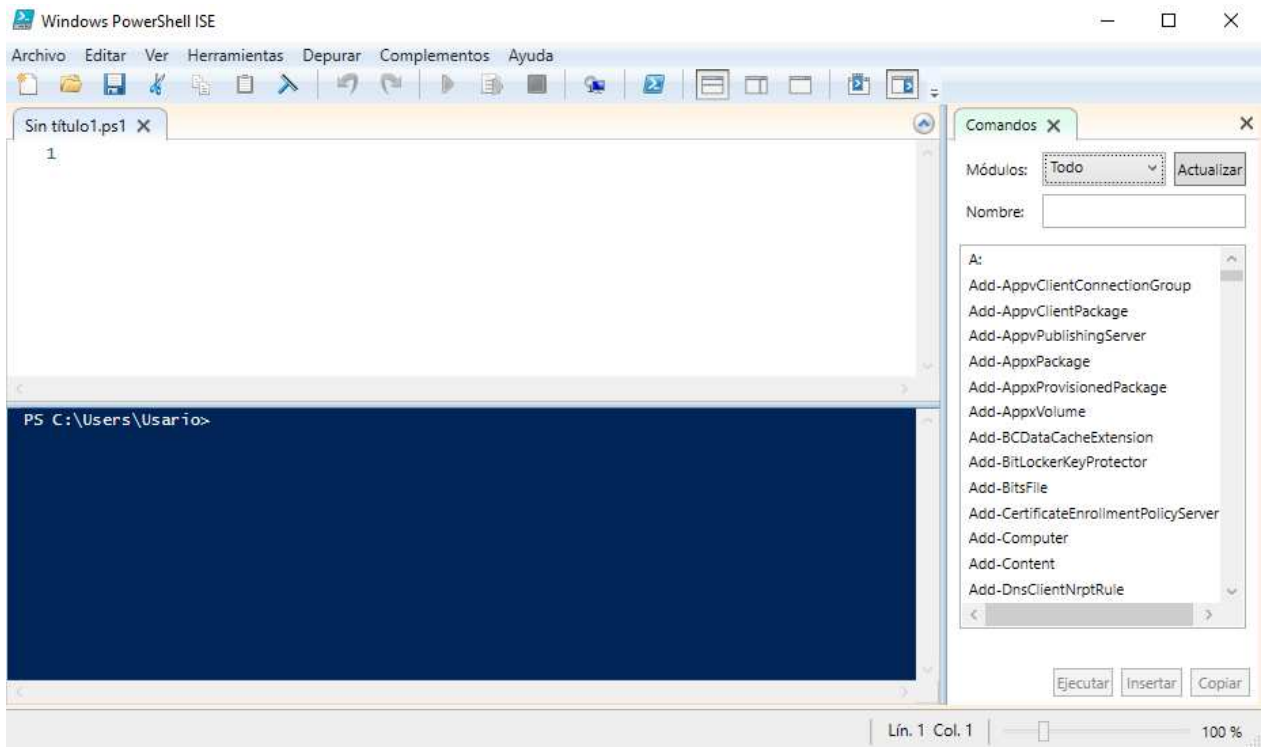


Figura 9. PowerShell ISE

### 2.3.3 Red Neuronal

Las redes neuronales son un paradigma que imita el funcionamiento del cerebro de los mamíferos, permite a una computadora aprender las facultades humanas a partir de datos de observación. Actualmente brindan soluciones a muchos problemas: clasificación de elementos, reconocimiento de imágenes, reconocimiento de escritura a mano, reconocimiento de voz, análisis de voz y PNL, entre otros.

Una red neuronal es una función flexible que adapta de forma autónoma su comportamiento para satisfacer en mayor medida la relación entre las entradas y los resultados esperados donde en una primera fase la red neural es entrenada con un conjunto de datos, y, en una segunda fase se validan resultados datos ajenos al conjunto de entrenamiento. Esto le da la capacidad de adquirir y mantener conocimientos, y definirla como un conjunto de unidades de procesamiento que se reduce a neuronas artificiales interconectadas con una cantidad considerable de conexiones -sinapsis-artificiales que son implementadas por vectores y matrices de pesos sinápticos [40].

## Neurona

Las estructuras de la red neuronal se desarrollaron a partir de modelos conocidos de sistemas nerviosos biológicos y del cerebro humano [20]. Los componentes computacionales o unidades de procesamiento, llamadas neuronas artificiales, son modelos simplificados de neuronas biológicas.

Las neuronas utilizadas en las redes neuronales son no lineales, por lo general proporcionan salidas continuas y realizan funciones simples, como recopilar señales disponibles en sus entradas, ensamblarlas de acuerdo con sus funciones operativas y producir una respuesta considerando sus funciones de activación innatas [21].

McCulloch y Pitts en 1943 en su capítulo "A logical calculus of the ideas immanent in nervous activity" [36] propusieron un modelo de neuronal artificial simple que incluye el paralelismo y la alta conectividad, propiedades principales de una neurona en una red neuronal biológica [22]. En ese modelo, cada neurona de la red se puede implementar como se muestra en la figura 10.

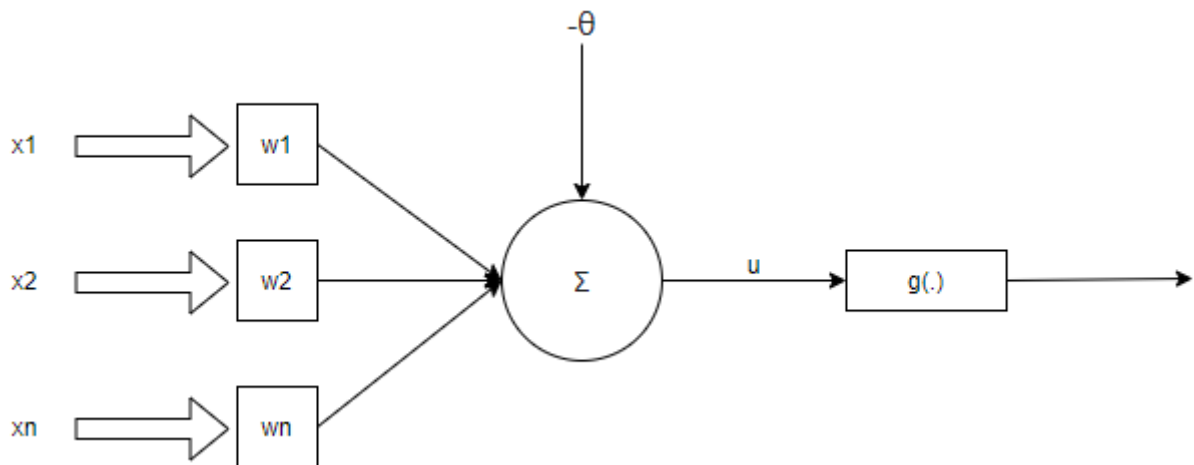


Figura 10. Modelo de neurona artificial de McCulloch & Pitts

Las múltiples señales de entrada provenientes del entorno externo están representadas por el conjunto  $\{x_1, x_2, x_3, \dots, x_n\}$ . El pesaje realizado por las uniones sinápticas de la red se implementa en la neurona artificial como un conjunto de pesos sinápticos  $\{w_1, w_2, \dots, w_n\}$ .

De forma análoga, la relevancia de cada una de las entradas de neuronas  $\{x_i\}$  se calcula multiplicándolas por su peso sináptico correspondiente  $\{w_i\}$ , ponderando así toda la información externa que llega a la neurona.

El objetivo de la función de activación (figura 10, "g(.)") es limitar la salida de la neurona dentro de un rango razonable de valores, asumido por su propia imagen funcional.

## **Componentes de una red neuronal**

En general, una red neuronal artificial se puede dividir en tres partes, denominadas capas, que se conocen como:

### Capa de entrada

Esta capa es responsable de recibir información del entorno externo. Estas entradas generalmente se normalizan dentro de valores límite producidos por las funciones de activación [22]. Esta normalización da como resultado una mejor precisión numérica para las operaciones matemáticas realizadas por la red.

### Capas ocultas, intermedias o invisibles

Estas capas están compuestas por neuronas que son responsables de extraer patrones asociados con el proceso o sistema que se analiza [22]. Estas capas realizan la mayor parte del procesamiento interno de una red.

### Capa de salida

Esta capa también está compuesta por neuronas y, por tanto, es responsable de producir y presentar las salidas finales de la red, que resultan del procesamiento realizado por las neuronas en las capas anteriores [22].

Las principales arquitecturas de las redes neuronales artificiales, considerando la disposición de las neuronas, así como también cómo se interconectan y cómo se componen sus capas, se pueden dividir de la siguiente manera: unicapa feedforward, multicapa feedforward, redes recurrentes y redes malladas.



## Arquitectura unicapa feedforward

Esta red neuronal artificial tiene solo una capa de entrada y una sola capa neuronal, que es también la capa de salida [20]. La figura 11 ilustra un feedforward de capa simple compuesta por  $n$  entradas y  $m$  salidas.

La información siempre fluye en una sola dirección (unidireccional), que es de la capa de entrada a la capa de salida. En la figura 11, es posible ver que, en esta arquitectura, el número de salidas de red siempre coinciden con su cantidad de neuronas. Estas redes suelen emplearse en problemas de clasificación de patrones y filtrado lineal.

Para entrenar esta arquitectura particular se efectúan un conjunto de pasos ordenados para ajustar los pesos y umbrales de sus neuronas [22]. Por lo tanto, dicho proceso de ajuste, también conocido como algoritmo de aprendizaje, tiene como objetivo sintonizar la red para que sus salidas estén cerca de los valores deseados.

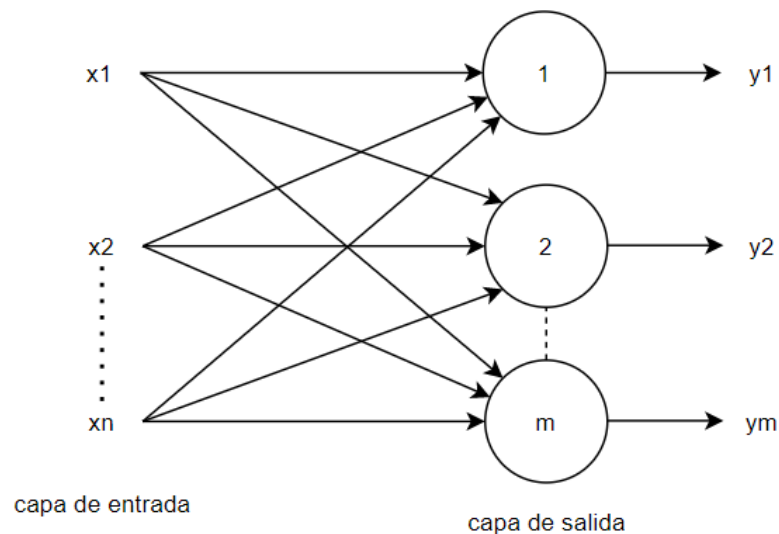


Figura 11. Red unicapa feedforward

## Aprendizaje supervisado

La estrategia de aprendizaje supervisado consiste en tener disponibles las salidas deseadas para un conjunto dado de señales de entrada [20]. De ahora en adelante, se requiere una tabla con datos de entrada/salida o atributo/valor, que representa el proceso y su comportamiento. Es a partir de esta información que las estructuras neuronales formularán "hipótesis" sobre el sistema que se está aprendiendo.

En este caso, la aplicación del aprendizaje supervisado solo depende de la disponibilidad de esa tabla de atributo/valor, y se comporta como si un "coach" estuviera enseñando a la red cuál es la respuesta correcta para cada muestra presentada por entrada.

Los pesos y umbrales sinápticos de la red se ajustan continuamente mediante la aplicación de acciones comparativas, ejecutadas por el propio algoritmo de aprendizaje, que supervisa la discrepancia entre las salidas producidas con respecto a las salidas deseadas, utilizando esta diferencia en el procedimiento de ajuste.

La red se considera "entrenada" cuando esta discrepancia se encuentra dentro de un rango de valor aceptable, teniendo en cuenta los propósitos de generalizar soluciones.

## **2.4 MONOTONÍA Y ACTIVIDADES REPETITIVAS**

Las actividades monótonas y prolongadas pueden causar a las personas problemas de atención y pueden conducir a una baja excitación, falta de alerta y más accidentes [37].

Investigaciones anteriores han otorgado un lugar central a las tareas de trabajo repetitivas sobre la inhibición de la actividad cortical a través de la habituación a las tareas monótonas [38].

Además, la monotonía de la tarea afecta la atención, el supuesto es que cuando existe la necesidad de realizar una tarea repetitiva, se percibirá como aburrida o poco interesante, la atención se deteriorará y esto se expresará como un menor rendimiento [39].

## **2.5 TRABAJOS RELACIONADOS**

Según Fernandez de Mello [23], el creciente volumen de documentos disponibles en la World Wide Web ha hecho que la indexación y la búsqueda de documentos sean cada vez más complejas. Este tema ha motivado el desarrollo de varias investigaciones en el

área de clasificación de textos. El modelo propuesto en este trabajo aborda las etapas de extracción de características, clasificación, etiquetado e indexación de documentos para fines de búsqueda.

En Chen [24], se presentan los desafíos en la clasificación de texto de redes neuronales, en este trabajo hay una gran cantidad de funciones en espera cuando se realiza la extracción de funciones. Un grupo de 100 textos puede generar 10 veces los candidatos a clasificar. La clasificación se basa en la colección de textos Reuter-21578 enfocada en el lenguaje natural.

Quinteiro-González [25] separa cada palabra (tokenize) y luego las palabras se derivan de su palabra raíz para generar una bolsa de palabras que funcionará como "corpus" en su clasificación de texto de lenguaje natural.

Zhang [26], lleva a cabo algunos experimentos para la clasificación de texto de múltiples clases mediante el uso de la red neuronal de backward propagation (BPNN). La adaptación del método combinado logró la mejora de la precisión para la tarea de clasificación de texto de clases múltiples.

Han [27], se propone una red neuronal convolucional (CNN o ConvNet) para la clasificación de tickets en un sistema ITSM. La comparación entre los enfoques de aprendizaje tradicionales contra deep learning para la categorización automática de tickets se pone en discusión por parte del autor.

Kandakumar [28], la clasificación de texto con redes neuronales ahora aborda un desafío empresarial en la gestión de servicios de TI. En el departamento de Helpdesk, casi el 40% de los tickets de incidencias no se enrutan al equipo de soporte correcto y los tickets pierden tiempo valioso de SLA dando vueltas entre diferentes equipos.

## **CAPÍTULO 3. PROCEDIMIENTO**

### **3.1 CONFIGURACIÓN DE ALMACENAMIENTO**

Para no comprometer la información de producción durante el desarrollo y pruebas de la automatización, se decidió utilizar un dispositivo NAS no productivo el cual funciona como laboratorio para validar diversas pruebas y proyectos antes de llevarlos a producción.

Desde este dispositivo se creó un volumen CIFS de 10 GB con el nombre 'Automation\_Test\_vol', se le asignó una política de retención de snapshot de 15 días y se configuró como un share drive con el nombre 'Automation\_Test'. Se asignaron los permisos correspondientes para que solo los miembros del equipo de storage tengan acceso.

Este nuevo share se asignó dentro del DFS subfolder "test", el path lógico final fue:  
`\\company.domain.com\test_environment\test\Automation_Test`

Dentro de este share se crearon carpetas y archivos "dummy" para la etapa posterior de pruebas de la automatización.

### **3.2 SERVIDOR, PYTHON**

Con el fin de centralizar el acceso a la automatización para los integrantes del equipo de storage, el código se desarrolló en un servidor Windows Server 2012, el cual cuenta con los dos módulos de PowerShell requeridos para la extracción del path físico y la administración de storage:

1. DataONTAP: NetApp PowerShell Toolkit 9.7.1 proporciona automatización de extremo a extremo y administración de almacenamiento en todas las plataformas de almacenamiento de NetApp. El módulo contiene 2340 cmdlets, lo que permite la administración de los sistemas de NetApp. También contiene varios cmdlets destinados a la administración de almacenamiento en hosts Windows [29].

2. DFSN: Los cmdlets de DFS Namespace de PowerShell cubren la mayor parte de la funcionalidad de administración del lado del servidor DFSN [30]. DFSN proporciona una vista simplificada de todos los recursos del sistema de archivos en varios servidores y sitios, además de la administración centralizada de los recursos de almacenamiento.

La automatización se codificó en lenguaje Python, basado en secuencias de comandos, una de sus principales ventajas es que se pueden escribir programas en menos líneas de código que un programa equivalente en C/C ++ o Java. [31].

### 3.3 EXTRACCIÓN DE INFORMACIÓN DE LOS TICKETS DE LA HERRAMIENTA ITSM

La ITSM cuenta con diversos REST API access points para la extracción, adición y edición de tickets de la base de datos. Por políticas de seguridad, las credenciales de conexión REST API solo aceptan miembros de la compañía con un rol de software developer.

Sin embargo, los gerentes de storage han lanzado una petición de excepción para el equipo de 'Security'. Para continuar con el trabajo de investigación se procedió a extraer la información del ITSM con Selenium. Selenium proporciona extensiones que permiten emular las interacciones que realizan los usuarios con los navegadores y ejecutar pruebas. Esto es como si un usuario real estuviera manipulando el navegador [32]. La figura 12 muestra el navegador Chrome siendo controlado por Selenium.

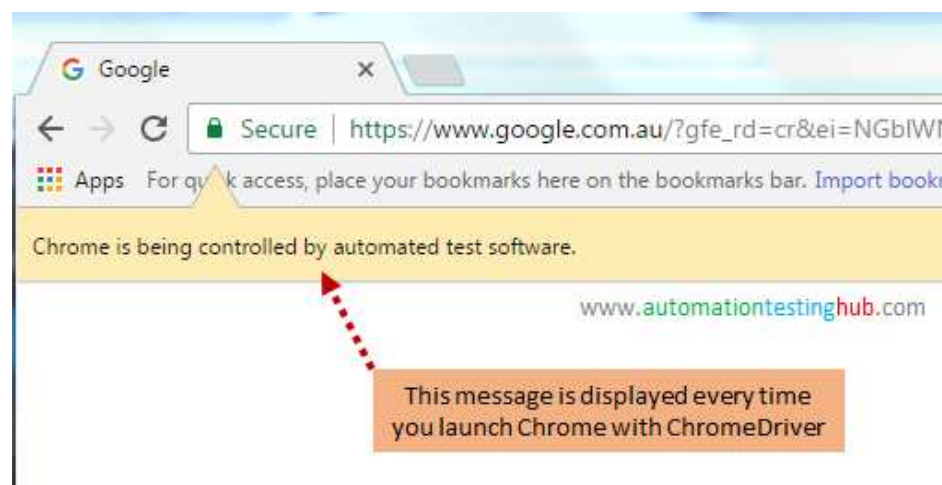


Figura 12. Chrome controlado por Selenium [33]

Selenium carga la página del ITSM dentro de la sección de 'Storage tasks >> File Restore'. Enseguida, utilizando la localización de elementos por CCS selector, se identifica la metadata del ticket y la información cruda es procesada por la automatización.

Cuando el ingeniero de storage ejecuta la automatización, se despliega a modo de CLI en el shell de Python la información depurada del ticket capturado (figura 13). Esta primera vista de la CLI muestra los siguientes campos:

- Numero de ticket
- Nombre del usuario
- Descripción de la actividad
- Instrucciones especiales
- UNC DFS path del archivo a restaurar
- Fecha solicitada a restaurar

```
Ticket: TI00001
User: FirstName LastName
Description: Restore excel file
Special instructions: Please restore this file from 11 am or earlier
Path: \\company.domain.com\test_environment\test\Automation_Test\excel_dummy.xlsx
Restore as of date: 01-11-2020
```

Figura 13. CLI Información del ticket

### 3.4 VALIDACIÓN DEL PATH DFS

El path lógico proporcionado por el usuario es segmentado y enviado a una función que extrae el path físico del path lógico. Esto es logrado con un subprocesso de PowerShell que corre el cmdlet 'Get-DfsnFolderTarget' el cual es parte del módulo de PowerShell DFSN. En la figura 14 se muestra un ejemplo de este cmdlet.

```
Administrator: Windows PowerShell
PS C:\> Get-DfsnFolderTarget -Path \\lab.ad\DFS\IT

Path          TargetPath          State  ReferralPriorityClass ReferralPriorityRank
-----
\\lab.ad\DFS\IT \\WINSRV2016\IT Online sitecost-normal 0

PS C:\> _
```

Figura 14. Cmdlet 'Get-DfsnFolderTarget' [34]

En caso de que el path que insertó el usuario no sea existente, se manda el path al módulo de red neuronal.

Cuando el path si ha sido introducido correctamente por el usuario en el ticket, la salida del cmdlet nos arrojará el path físico (TargetPath), cuyo primer segmento nos indica el nombre identificativo del servidor o dispositivo NAS y el segundo segmento nos indica el nombre del share. En el caso del share de prueba path lógico:

```
'\\company.domain.com\test_environment\test\Automation_Test'
```

El path físico corresponde a:

```
\\laboratorionas01\Automation_Test
```

Donde 'laboratorionas01' identifica el nombre del dispositivo y 'Automation\_Test' es el nombre del share.

A continuación, la automatización valida que el ultimo folder del path sea accesible, esto con el comando de Python `os.isdir()`. En caso de que la validación no sea exitosa, se puede deber a uno de los siguientes factores:

- Que la carpeta contenedora final tenga un ACL diferente para proteger datos sensibles.
- Que alguna de las carpetas del DFS path haya cambiado de nombre desde la creación del ticket hasta el momento de la ejecución de la automatización.

Después se verifican los contenidos de la última carpeta del path. En caso de que el archivo solicitado a restaurar se encuentre ahí mismo, se mostrara en la CLI el siguiente mensaje:

- "Archivo a restaurar existente, una versión anterior del archivo es requerida".

En caso de que el archivo no se encuentre, se mostrará en la CLI:

- "Archivo no existente. Este es un restore de recuperación".

El flujo del programa continua con la sección de comandos NAS, cuya función es orquestar todos los comandos hacia el dispositivo de almacenamiento con subprocessos de PowerShell basados en el módulo DataONTAP.

## 3.5 COMANDOS NAS

### 3.5.1 Snapshots disponibles

Se inicia el flujo de comandos estableciendo la sesión con el dispositivo NAS usando el cmdlet:

```
'Connect-NcController'
```

Una vez iniciada la sesión se obtiene el nombre del volumen donde reside el share y se ejecuta el siguiente cmdlet que retorna los snapshots disponibles:

```
'Read-NcDirectory -Path '
```

Todos los snapshots son depurados y se muestra solo en CLI, en forma de menú seleccionable, solo los de la fecha solicitado por el usuario final. Adicionalmente, con la librería de Python 'datetime' se muestra en la CLI un snapshot del día anterior al solicitado por el usuario como se muestra en la figura 15.

```
Snapshots available:
[1] daily_snap_2020_11_01
[2] daily_snap_2020_10_31
Select an snapshot(0 to cancel):_
```

Figura 15. Menú de selección del snapshot

### 3.5.2 Búsqueda del archivo dentro del snapshot

El snapshot seleccionado es ahora explorado usando como referencia el path previamente validado y se agregan los siguientes parámetros al cmdlet:

```
'Read-NcDirectory | Select-Object Name, Type, Size, Modified'
```



Estos parámetros nos retornan una salida a detalle de los archivos y carpetas existentes en ese punto del tiempo.

Nuevamente la salida del cmdlet es procesada y se muestra en la CLI las opciones que coinciden con el nombre del archivo solicitado (figura 16).

```
Matches found:

[1] Name: excel_dummy.xlsx
    Type: File
    Size: 38.92 MB
    Last modified date: 11/1/2020 08:48:51 AM

Select an snapshot(0 to cancel):_
```

Figura 16. Menú de selección del archivo

### 3.5.3 Creación de carpeta alterna y comando de restauración

Cuando el archivo a restaurar es seleccionado, el flujo de programa entra a una sección de código que crea una subcarpeta dentro de la carpeta final del path con el comando 'os.makedirs()'. La finalidad de esta nueva carpeta es la de no sobrescribir información en la carpeta original y tener mayor facilidad de seguimiento de la actividad de restauración, ya que el nombre de la nueva carpeta tiene el formato 'Restore\_Ticket001'.

Al momento de ejecutarse la restauración, el cmdlet incluye el folder alterno como destino:

```
'Restore-NcSnapshotFile -Path "p" -RestorePath "pd" -SnapName "snpashot_selected"'
```

Donde "p" es el path origen y "pd" es el path destino.

Por último, el comando os.count() verifica la existencia del archivo restaurado en la carpeta alterna, quedando esta última accesible, como se ilustra en la figura 17, con el path:

```
\\company.domain.com\test_environment\test\Automation_Test\Restore_TI00001
```

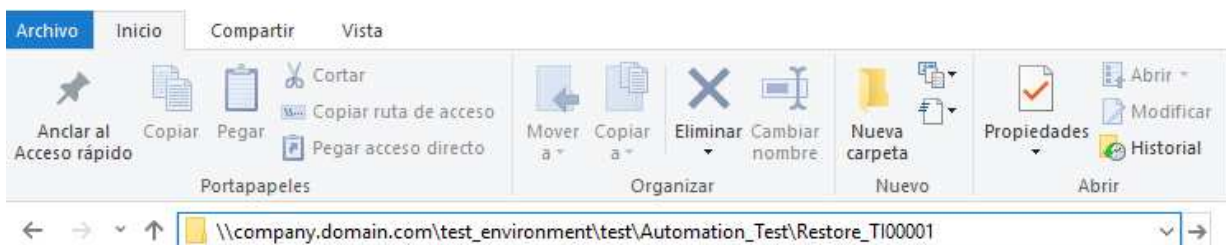


Figura 17. DFS Path de la carpeta destino de la restauración

### 3.6 NOTIFICACIÓN DE CORREO ELECTRÓNICO Y RESOLUCION DE TICKET

La última etapa de la automatización envía una notificación por correo electrónico al usuario final notificando que la restauración del archivo ha sido completada exitosamente. El mensaje de correo electrónico contiene el nombre del archivo restaurado y su ruta DFS, la plantilla de correo también incluye la información de contacto del equipo de storage (figura 18).

```
From: automation@company.com
To: saul@company.com
Subject: TI00001 File restore

Hello Saul,

Your file "excel_dummy.xlsx" has been restored as requested. Location:
\\company.domain.com\test_environment\test\Automation_Test\Restore_TI00001

Feel free to contact storage team for any comment or question,
storage_team@company.com
```

Figura 18. Notificación de correo electrónico

La dependencia del correo electrónico se ha configurado en la automatización mediante un servidor SMTP local (Protocolo Simple de Transferencia de Correo). Las instancias SMTP transfieren correo entre dominios y también se conectan con los usuarios de correo electrónico en una relación cliente/servidor [35].

Finalmente, Selenium webdriver asigna el ticket al ingeniero de storage que corre la automatización, agrega como notas de cierre el cuerpo del mensaje de correo electrónico enviado y cambia el estado del ticket de 'Open' a 'Closed successfull'. Luego analiza el siguiente ticket.

### 3.7 MÓDULO DE RED NEURONAL

Cuando el path que insertó el usuario no es identificado por el cmdlet 'Get-DfsnFolderTarget', el módulo de red neuronal clasifica los elementos del path para identificar en que segmento el usuario introdujo algún error sintáctico.

La automatización extrae la estructura inicial DFS y los primeros cuatro elementos del path ruta son almacenados en variables locales, basados en la estructura de Microsoft DFS, como se muestra en la tabla 2.

Tabla 2. Path DFS segmentado

| <b>Categoría DFS</b> | <b>Nombre variable</b> | <b>Path segmentado</b> |
|----------------------|------------------------|------------------------|
| Namespace            | namespace              | company.domain.com     |
| DFS Folder           | dfs_folder             | test_environment       |
| DFS Subfolder        | dfs_subfolder          | test                   |
| Share                | Share                  | Automation_Test        |

Estas variables son clasificadas consecutivamente por la red neuronal. El éxito de la clasificación se basa en el conjunto de datos de entrenamiento de la red neuronal.

Como ocurre con todos los algoritmos de aprendizaje supervisado, se requiere un conjunto de datos debidamente etiquetado para diseñar y entrenar un algoritmo capaz de clasificar la información de manera apropiada. La información del servidor de administración de DFS del ambiente de pruebas del cliente ha sido extraído parcialmente hacia nuestros datos de entrenamiento, con los que se han creado cuatro clases basadas en la estructura del espacio de nombres: namespace\_c, dfs\_folder\_c, dfs\_subfolder\_c, share\_c (figura 19).

Cada clase se alimenta con los nombres de las carpetas contenidas en cada nivel de la jerarquía DFS. Para la clase interna "share\_c", el cmdlet de PowerShell 'GetChildItem' resultó muy útil ya que la salida de texto sin formato se exportó fácilmente al código de Python. La figura 19 muestra los nombres de las carpetas como parámetros de codificación agregados a las clases de entrenamiento de datos.

```

training_data.append({"class": "namespace_c", "sentence": "company.domain.com"})
training_data.append({"class": "namespace_c", "sentence": "company"})

training_data.append({"class": "dfs_folder_c", "sentence": "production"})
training_data.append({"class": "dfs_folder_c", "sentence": "test_environment"})
training_data.append({"class": "dfs_folder_c", "sentence": "read-only"})
training_data.append({"class": "dfs_folder_c", "sentence": "qos"})

training_data.append({"class": "dfs_subfolder_c", "sentence": "marketing"})
training_data.append({"class": "dfs_subfolder_c", "sentence": "human_resources"})
training_data.append({"class": "dfs_subfolder_c", "sentence": "accounting"})
training_data.append({"class": "dfs_subfolder_c", "sentence": "logistics"})
training_data.append({"class": "dfs_subfolder_c", "sentence": "invoices"})
training_data.append({"class": "dfs_subfolder_c", "sentence": "communications"})
training_data.append({"class": "dfs_subfolder_c", "sentence": "test"})

training_data.append({"class": "share_c", "sentence": "plans_2019"})
training_data.append({"class": "share_c", "sentence": "project01"})
training_data.append({"class": "share_c", "sentence": "FY2020"})
training_data.append({"class": "share_c", "sentence": "Automation_Test"})

```

Figura 19. Datos de entrenamiento de red neuronal

Los datos de entrenamiento se transforman en una matriz, la librería de Python 'NumPy' proporciona las funciones para crear y manipular matrices. Esta matriz de entrenamiento es procesada por un perceptrón de una sola capa (figura 20). La combinación lineal ponderada de valores de entrada ingresa al nodo activo, donde será transformada por una función de activación esta función de activación se utiliza para hacer que la salida caiga dentro de un rango deseado, evitando así la parálisis de la red.

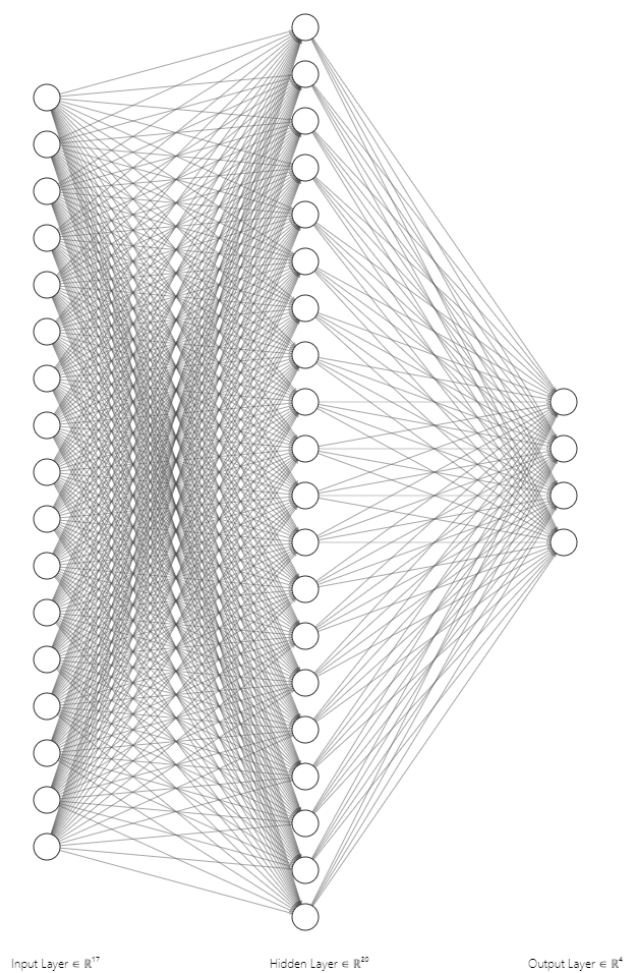


Figura 20. Arquitectura de la red neuronal

Estructura de la red neuronal:

- Capa de entrada: 17 neuronas
- Capa oculta: 20 neuronas
- Capa de salida: 4 neuronas

## CAPÍTULO 4. RESULTADOS

### 4.1 RED NEURONAL

Una vez que los datos de entrenamiento fueron introducidos al código, la red neuronal fue entrenada con una época de 10,000 iteraciones reduciendo el error delta de 0.542089 a 0.007707 (figura 21).

```
17 sentences in training data
Training with 20 neurons
Input matrix: 17x17   Output matrix: 1x4
delta after 0 iterations:0.5420895905494633
delta after 100 iterations:0.2382688479544547
delta after 200 iterations:0.144468569242975
delta after 300 iterations:0.0928095904576039
delta after 400 iterations:0.0689422790522891
delta after 500 iterations:0.05601156593698377
delta after 600 iterations:0.04787363996016239
delta after 700 iterations:0.042232564950937725
delta after 800 iterations:0.03806090691965077
delta after 900 iterations:0.034831145367980876
delta after 1000 iterations:0.032244012502477756
delta after 2000 iterations:0.02014911772129237
delta after 3000 iterations:0.015626632945500925
delta after 4000 iterations:0.013128065667839002
delta after 5000 iterations:0.011499802521268648
delta after 6000 iterations:0.010335683079261845
delta after 7000 iterations:0.00945232818999067
delta after 8000 iterations:0.008753571644000003
delta after 9000 iterations:0.008183615903329135
delta after 10000 iterations:0.007707625564381474
```

Figura 21. Iteraciones de entrenamiento de la red neuronal

Enseguida, los pesos sinápticos se almacenaron en un archivo .json local, cuyo objetivo es ser utilizados por la red neuronal como referencia para posteriores clasificaciones.

Para la validación de la clasificación de texto se generaron algunos tickets de prueba con un DFS path correcto y otros con información incorrecta respectivamente. La automatización extrajo los metadatos de los tickets y segmentó el DFS path. Para el primer caso, el escenario de la información correcta, se mostró una tasa de certeza de clasificación superior a 0.99 o casi un 100% (figura 22).

```
company.domain.com
classification: [['namespace_c', 0.9957933042434507]]
test_environment
classification: [['dfs_folder_c', 0.9964677963593462]]
test
classification: [['dfs_subfolder_c', 0.9975515858362045]]
Automation_Test
classification: [['share_c', 0.9967755478884792]]
```

Figura 22. Clasificación con DFS path válido

Por otro lado, en el caso del ticket de prueba con información errónea, las tasas de clasificación dieron como resultado una puntuación de alrededor de 0.12. Este resultado no cumple con los criterios de clasificación. Algunos segmentos del path ni siquiera pudieron ser clasificados (figura 23). En este caso la automatización pide autorización, a través de la CLI, de mandar un correo electrónico al usuario final para que verifique la información proporcionada en el ticket, especificando el o los segmentos del path donde la red neuronal no pudo clasificar el texto.

```
wrong.domain.com
classification: []
unexistent_folder
classification: []
badsubfolder
classification: []
nothing
classification: []
```

Figura 23. Clasificación con DFS path erróneo

## 4.2 AHORRO DE TIEMPOS DE LA AUTOMATIZACIÓN

Para validar la eficacia del ahorro de tiempos de la automatización se registró en la tabla 3 el tiempo consumido por actividad en segundos. Es importante mencionar que esta prueba cronometrada se realizó en un ambiente controlado, con el explorador web abierto en la página de tickets de storage, así como las sesiones de servidor habilitadas listas para insertar el path DFS. El archivo a restaurar tenía una capacidad de 46 KB.

Tabla 3. Registro de tiempos de restore manual en ambiente controlado

| Actividad                                    | Tiempo (s) | Σ Tiempos (s) |
|--|------------|---------------|
| Localización del ticket de restauración      | 39         | 39            |
| Validación del DFS path                      | 82         | 121           |
| Extracción del path físico destino           | 151        | 272           |
| Login al dispositivo NAS                     | 42         | 314           |
| Encontrar el snapshot                        | 64         | 378           |
| Encontrar el archivo, ejecutar el comando    | 67         | 445           |
| Enviar notificación de correo, cerrar ticket | 158        | 603           |

La ejecución de la restauración manual tomó al ingeniero de almacenamiento 603 segundos (10.1 min). La figura 24 muestra de manera gráfica este tiempo contra los 38 segundos totales de la restauración automatizada.

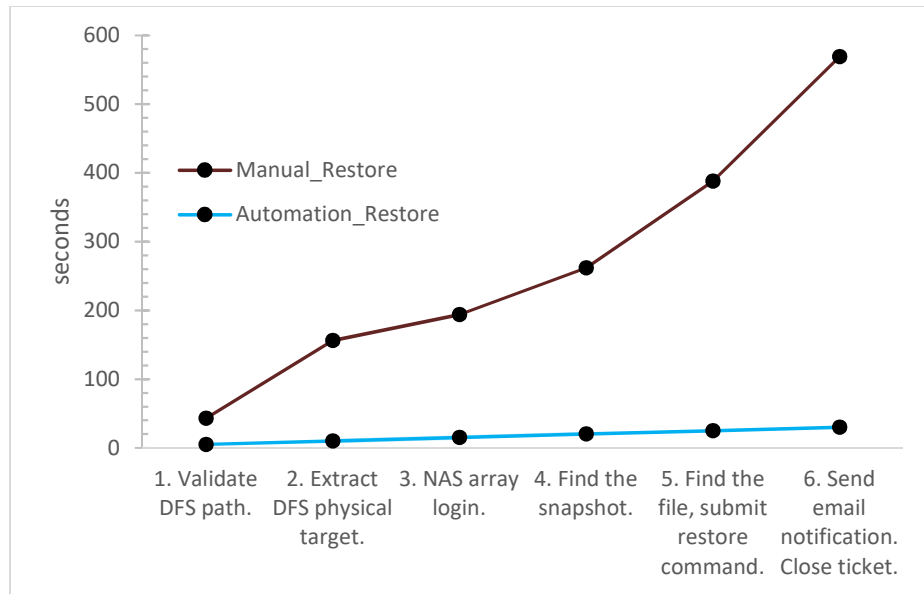


Figura 24. Gráfica de tiempos manual restore vs automation restore



## CONCLUSIONES

La implementación de esta automatización involucra mínimamente al equipo de storage en las actividades de restauración. Los expertos en storage solo participan cuando la red neuronal no puede clasificar correctamente el path debido a errores de sintácticos en el texto ingresado por el usuario final. Así mismo, las actividades de restauración ahora ya no son una actividad monótona y repetitiva, ahora se ha convertido en soporte básico para el usuario final donde el ingeniero de storage brinda orientación y retroalimentación al usuario.

Durante la etapa de prueba, la automatización demostró su rápida ejecución y rendimiento, lo que ahorra hasta 10 minutos por tarea ticket en comparación al proceso de restauración manual ejecutado por el ingeniero.

La automatización ayuda al equipo de soporte de storage a aliviar la carga de trabajo y les permite concentrarse en tickets de alto impacto, respondiendo a interrupciones del servicio, problemas e incidentes con el enfoque adecuado, investigando para determinar la causa raíz del problema y proporcionando una resolución rápida.

Además, el ahorro de tiempos permite que los expertos en storage se centren en actividades de infraestructura como:

- Planificación de la capacidad del sistema de storage, migración de datos, gestión y soporte.
- Análisis y planificación de la continuidad del negocio. Informes.
- Comprobaciones de estado proactivas, gestión de capacidad y rendimiento.
- Garantizar la disponibilidad, confiabilidad, escalabilidad y capacidad de administración del entorno de almacenamiento.

## APORTACIÓN DE LA TESIS

La SNIA (Storage Networking Industry Association) es un organismo neutral vendor cuyo objetivo es el desarrollo de la tecnología de storage. Esta organización ha centralizado grandes avances de diferentes en la industria del almacenamiento digital enfocados tanto al avance técnico y rendimiento, entre otros tópicos. Sin embargo, el soporte al usuario final aún sigue siendo un área de gran oportunidad donde este organismo, no ha penetrado aún. Este trabajo de tesis contribuye a la atención del cliente final aliviando la carga de trabajo de los ingenieros de storage y mejorando el flujo de comunicación con el usuario final habilitando la solución de restauración adecuada.

Finalmente, fruto de este trabajo de investigación se logró la siguiente publicación: Esquivel-García, S., & Hernández-Uribe, Ó. (2020). File Restore Automation with Machine Learning. In M. F. Mata-Rivera, R. Zagal-Flores, & C. Barria-Huidobro (Eds.), *Telematics and Computing* (Vol. 1280, pp. 61–72). Springer International Publishing. [https://doi.org/10.1007/978-3-030-62554-2\\_5](https://doi.org/10.1007/978-3-030-62554-2_5)

## **APORTACIÓN SOCIAL DE LA TESIS**

Este trabajo de tesis proporciona a la comunidad de profesionales de almacenamiento digital una herramienta que permite aliviar la carga de trabajo, reducir tiempos de operación y mejorar el flujo de comunicación con el usuario final. Así mismo, la reducida interacción directa humano-máquina en el proceso manual de restauración minimiza las posibilidades de cometer un error humano.

## RECOMENDACIONES

Para la implementación de este trabajo se recomienda altamente la extracción de tickets a través de REST API en lugar de un webdriver, debido a los siguientes motivos:

1. Los elementos web suelen cambiar mínimamente de posición con los releases de los desarrolladores de la herramienta. Esto significa actualizar constantemente la sección de código donde se ejecutan los CSS locators, haciendo mantenibilidad del software más complicada.
2. Se tiene que descargar el webdriver correspondiente a cada versión actualizada del navegador, no hay retrocompatibilidad. Durante el desarrollo y pruebas de la automatización, el navegador se actualizó a las siguientes versiones:
  - a. 81.0.4044.138
  - b. 83.0.4103.116
  - c. 85.0.4183.102
  - d. 86.0.4240.111

El uso estandarizado de consultas REST API evita los inconvenientes antes descritos.

Se recomienda como trabajo futuro migrar la CLI por una GUI de acceso web, se recomienda Angular.js para su desarrollo.

Para mayor alcance e impacto de la automatización se recomienda el uso del protocolo NDMP para restaurar carpetas, donde la interfaz de administración muestre la capacidad de la carpeta a restaurar, esta función de lectura no debe impactar en el rendimiento del dispositivo NAS.

Por último, para la implementación del módulo de red neuronal en un ambiente productivo, por la naturaleza de la estructura jerárquica de árbol del DFS se recomienda implementar dos redes neuronales, una para la clasificación de los campos de namespace y DFS folder y la segunda para la clasificación de DFS subfolder y share. Con esta estrategia se pretende no desbalancear una única red y sus pesos sinápticos.

## REFERENCIAS

- [1] Nagy P.G., Schultz T.J. (2006) Storage and Enterprise Archiving. In: Dreyer K.J., Thrall J.H., Hirschorn D.S., Mehta A. (eds) PACS. Springer, New York, NY
- [2] Services, E. E. (2009). Information Storage and Management. Indianapolis, Indiana: Wiley Publishing.
- [3] Surianarayanan C., Chelliah P.R. (2019) Storage Fundamentals and Cloud Storage. In: Essentials of Cloud Computing. Texts in Computer Science. Springer, Cham.
- [4] (2009) NAS. In: LIU L., ÖZSU M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA.
- [5] Herminghaus, V. (2009). Storage Management in Data Centers. Berlin Heidelberg: Springer-Verlag.
- [6] Buchanan W.J. (2002) Distributed file systems. In: The Complete Handbook of the Internet. Springer, Boston, MA.
- [7] (2005) Using Virtual File Systems. In: Virtualization from the Desktop to the Enterprise. Apress.
- [8] Docs, Overview of DFS Namespaces: [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc730736\(v=ws.11\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc730736(v=ws.11)?redirectedfrom=MSDN), last accessed on 2020/4/3.
- [9] Sallé, M. (2004) IT Service Management and IT Governance: Review, Comparative Analysis and their Impact on Utility Computing. HP Research Labs.
- [10] Winniford M, Conger S, Erickson-Harris L (2009) Confusion in the ranks: IT service management practice and terminology. Information Systems Management, 26:2, 153-163.
- [11] Marrone, M., Kolbe, L.M. (2011) Uncovering ITIL claims: IT executives' perception on benefits and Business-IT alignment. Inf Syst E-Bus Manage 9, 363–380
- [12] Potgieter B.C., Botha J.H., Lew C. (2005) Evidence that use of the ITIL framework is effective.
- [13] Kahlout G. (2017) Pragmatic or Tragic ITSM. In: Spinning Up ServiceNow. Apress, Berkeley, CA.
- [14] Familiar B. (2015) Automation. In: Microservices, IoT, and Azure. Apress, Berkeley, CA.
- [15] Ritchie S.D. (2011) Build Automation. In: Pro .NET Best Practices. Apress.

- [16] Doglio F. (2018) REST 101. In: REST API Development with Node.js. Apress, Berkeley, CA.
- [17] Kurtz J., Wortman B. (2014) What Is RESTful?. In: ASP.NET Web API 2: Building a REST Service from Start to Finish. Apress, Berkeley, CA.
- [18] Composition of Semantically Enabled Geospatial Web Services - Scientific Figure on ResearchGate. Disponible de: [https://www.researchgate.net/figure/Trends-of-XML-and-JSON-API-in-Google-searches\\_fig2\\_296702598](https://www.researchgate.net/figure/Trends-of-XML-and-JSON-API-in-Google-searches_fig2_296702598) [accessed 24 Oct, 2020]
- [19] Hosmer C. (2019) An Introduction to PowerShell for Investigators. In: PowerShell and Python Together. Apress, Berkeley, CA.
- [20] Goyal, P. (2018). Deep Learning for Natural Language Processing.
- [21] Awange, J. (2020). Hybrid Imaging and Visualization. Switzerland: Springer Nature Switzerland.
- [22] Da Silva I.N., Hernane Spatti D., Andrade Flauzino R., Liboni L.H.B., dos Reis Alves S.F. (2017) Artificial Neural Networks. Springer, Cham.
- [23] Fernandes de Mello, R. (n.d.). Automatic text classification using an artificial neural network. Sao Paulo: Universidade de Sao Paulo.
- [24] Chen, J. (2011). Study a Text Classification Method Based on Neural Network Model. Advances in MSEC Vol. 1, 471-465.
- [25] Quinteiro-González, J. M. (2011). Clasificación de textos en lenguaje natural usando la Wikipedia. Revista Ibérica de Sistemas e Tecnologias de Informação.
- [26] Zhang, W. (2007). Text Classification with Support Vector Machine and Back Propagation Neural Network. ICCS, 150-157.
- [27] Han, J. (2018). Vertical Domain Text Classification: Towards Understanding IT Tickets Using Deep Neural Networks. The Thirty-Second AAAI Conference on Artificial Intelligence, 8202-8203.
- [28] Kandakumar, K. (2020, March 3). medium.com. Retrieved from [https://medium.com/@karthikkumar\\_57917/it-support-ticket-classification-using-machine-learning-and-ml-model-deployment-ba694c01e416](https://medium.com/@karthikkumar_57917/it-support-ticket-classification-using-machine-learning-and-ml-model-deployment-ba694c01e416)
- [29] NetApp. (2020, October 29). NetApp PowerShell Toolkit. Retrieved from <https://mysupport.netapp.com/site/tools/tool-eula/powershell-toolkit>
- [30] Chadalapaka, M. (2020, October 29). Introducing DFS Namespaces Windows PowerShell Cmdlets. Retrieved from

- <https://techcommunity.microsoft.com/t5/storage-at-microsoft/introducing-dfs-namespaces-windows-powershell-cmdlets/ba-p/424612>
- [31] Kelly S. (2016) What Is Python?. In: Python, PyGame and Raspberry Pi Game Development. Apress, Berkeley, CA.
- [32] Selenium. (2020, October 29). Selenium, el proyecto para automatizar navegadores. Retrieved from <https://www.selenium.dev/documentation/es/>
- [33] anish. (2020, October 30). Selenium ChromeDriver – 2 ways to launch Chrome. Retrieved from <http://www.automationtestinghub.com/selenium-chromedriver/>
- [34] Microsoft TechNet. (2020, November 1). How to find from where DFS Namespaces are coming from? Retrieved from <https://social.technet.microsoft.com/Forums/en-US/daafcb1b-7ae0-41c7-9828-2eb11219b451/how-to-find-from-where-dfs-namespaces-are-coming-from?forum=winserverfiles>
- [35] Howser, G.: Simple mail transfer protocol: email. In: Howser, G. (ed.) Computer Networks and the Internet, pp. 385–417. Springer, Cham (2020)
- [36] McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115–133
- [37] McBain in Loukidou, E. (2009). Boredom in the workplace: More than monotonous tasks. *International Journal of Management Reviews*, 381-405.
- [38] O'Hanlon in Loukidou, E. (2009). Boredom in the workplace: More than monotonous tasks. *International Journal of Management Reviews*, 381-405.
- [39] Loukidou, E. (2009). Boredom in the workplace: More than monotonous tasks. *International Journal of Management Reviews*, 381-405.
- [40] Barrios-Córdova, A. et al (2019). Estudio comparativo de técnicas de optimización multirespuesta en diseños experimentales. *Ingeniería Investigación y Tecnología vol XXI (número 2)*, 1-12. <https://doi.org/10.22201/ifi.25940732e.2020.21n2.016>

# ANEXOS

## A1. IEEE 830 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

### 1 Introducción

#### 1.1 Propósito

Este documento es la especificación de requisitos de software para la aplicación "Automatización de restauración de archivos". El propósito de este documento es presentar al lector las funcionalidades y requisitos del producto para su correcto despliegue e implementación. Esta especificación está basada en ANSI/IEEE 830, 1998.

#### 1.2 Alcance

El objetivo de la "Automatización de Restauración de Archivos" es ayudar a los administradores de storage en sus actividades diarias. La automatización extrae los tickets de restauración de la herramienta ITSM y procede con la ejecución automática de la restauración del archivo. En caso de que se identifique información incorrecta en el ticket, un módulo NN identifica el error sintáctico y la automatización envía un correo electrónico de notificación al usuario final.

#### 1.3 Tabla de acrónimos

| Acrónimo | Significado                               |
|----------|---|
| CIFS     | Common Internet File System               |
| CLI      | Command Line Interface                    |
| DC       | Data Center                               |
| DFS      | Distributed File System                   |
| HA       | High Availability                         |
| HTTP     | Hypertext Transfer Protocol               |
| IP       | Internet Protocol                         |
| ITSM     | Information Technology Service Management |
| NAS      | Network Attached Storage                  |
| NFS      | Network File System                       |
| NIC      | Network Interface Card                    |
| NN       | Neural Network                            |
| RAM      | Random Access Memory                      |
| SMTP     | Simple Mail Transfer Protocol             |
| TCP      | Transmission Control Protocol             |



## **1.4 Referencias**

IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.  
IEEE Computer Society, 1998.

## **1.5 Visión general del documento**

Este documento proporciona al lector una descripción general de la funcionalidad del producto de software. Describe los requisitos y contextos necesarios para un desempeño óptimo.

La primera sección contiene los propósitos del documento, alcance, acrónimos y referencias. La segunda sección incluye una descripción general de la perspectiva y las funciones del producto, las características del usuario, las restricciones del sistema y los supuestos y dependencias. La tercera sección explica las interfaces de usuario, hardware, software y comunicación, así como los requisitos funcionales y no funcionales.

## **2 Descripción general**

### **2.1 Perspectiva del producto**

Esta aplicación es una nueva solución para la industria del almacenamiento. El acceso a este producto a través de un servidor centralizado permite controlar el acceso debido a la restricción de la ventana de mantenimiento permitida por el cliente. Su conexión rápida y sin interrupciones a la base de datos de tickets ITSM le brinda al administrador de storage una experiencia de trabajo fluida.

### **2.2 Funciones del producto**

- Conexión a la base de datos de la herramienta ITSM de ticket. Filtra los tickets para el equipo de almacenamiento y extrae solo los tickets de restauración de archivos. Cierre y resolución de ticket cuando la automatización restaura el archivo.
- CLI muestra el progreso de las etapas de la automatización al administrador de storage para supervisar y controlar la ejecución de la automatización.

- Algoritmo genera una secuencia de comandos para el dispositivo NAS, que encuentra el archivo para restaurar y solicita al administrador de storage, a través de la CLI, una aprobación final para la ejecución del comando.
- NN procesa el texto incorrecto insertado por el usuario final en el campo del path. Identifica el segmento incorrecto y confirma la inexistencia del path.
- La función de correo electrónico notifica al usuario final si la restauración del archivo se realizó correctamente o si el NN detectó algún error.

### **2.3 Características del usuario**

Tipo de usuario: Operador.

Educación: Fundamentos en de almacenamiento NAS.

Actividades: Ejecutar la automatización y monitorear el progreso de la restauración.

### **2.4 Restricciones**

- El acceso a la Automatización está restringido. Solo los administradores de storage pueden acceder a él a través de la intranet de la empresa.
- La automatización no restaura carpetas, solo archivos individuales.
- Solo se restaurarán los archivos CIFS, no se admite NFS.

### **2.5 Supuestos y dependencias**

- El sistema se asume en un entorno HA de 5 nueves (99,999%).
- Se supone que la Automatización no funciona cuando la instancia de tickets ITSM no está disponible.
- La Automatización de restauración de archivos se ejecuta en la infraestructura de DC del cliente, cuyo storage está basado a nivel archivo.
- Los dispositivos de almacenamiento están configurados apropiadamente para enviar alertas de capacidad cuando está a punto de alcanzarse la cuota de capacidad máxima establecida.
- La Automatización se despliega en un servidor centralizado que se puede alojar tanto en un host virtual como físico dentro de las instalaciones del cliente.

### **3 Requerimientos específicos**

#### **3.1 Interfaz de usuario**

La interfaz de usuario es una CLI que muestra la información general del ticket, número de ticket, nombre del usuario final, DFS path, fecha de restauración requerida e instrucciones especiales requeridas por el usuario. La información fluye desde la automatización hacia el administrador de storage mostrando un menú seleccionable para los snapshots históricos y otro para las coincidencias de archivos encontrados. Además, muestra mensajes de advertencia de capacidad y mensajes de error.

#### **3.2 Interfaz de hardware**

La Automatización se despliega en un servidor centralizado que se puede alojar tanto en un host virtual como físico el cual debe cumplir al menos las siguientes características:

- 4 GB RAM.
- 1.66 GHz CPU de un núcleo.
- NIC para acceso de red.

#### **3.3 Interfaz de software**

Servidor con sistema operativo Windows Server 2012 o superior sin restricciones de firewall para la comunicación con el sistema de tickets y el dispositivo de almacenamiento NAS. Shell disponible para la CLI de la Automatización.

#### **3.4 Interfaz de comunicación**

- HTTP para acceso al ITSM.
- TCP/IP para conexión y comunicación con el dispositivo NAS.
- SMTP para la notificación de correo electrónico.

### 3.5 Requerimientos funcionales

| <b>Id</b>   | <b>Nombre</b>               | <b>Descripción</b>  |
|-------------|-----------------------------|---|
| <b>RF01</b> | Ticket ITSM tool connection | Conexión con la ITSM y extracción de tickets de restauración.                                       |
| <b>RF02</b> | Text depuration             | Procesamiento de texto de la metadata del ticket para ser mostrado en la CLI.                       |
| <b>RF03</b> | Neural Network              | Análisis del path incorrecto para la identificación del error sintáctico.                           |
| <b>RF04</b> | CLI                         | Despliegue de mensajes, menús, advertencias de capacidad y progreso de la restauración.             |
| <b>RF05</b> | NAS device commands         | Conexión con el dispositivo NAS identificado, gestión de comandos a ejecutar.                       |
| <b>RF06</b> | Email notification          | Mensajes de correo electrónico a los usuarios, notificación del progreso y finalización del ticket. |

### 3.6 Non-Functional Requirements

| <b>Id</b>   | <b>Nombre</b>                                  | <b>Descripción</b>  |
|-------------|--|---|
| <b>RN01</b> | Ticket metadata structure                      | La estructura del texto de los metadatos no afecta al software. La automatización extrae solo frases clave.   |
| <b>RN02</b> | Security                                       | El acceso a la automatización está disponible para el equipo de storage. El servidor debe cumplir con las políticas de seguridad para su uso en ambiente de producción. |
| <b>RN03</b> | Ticket ITSM latency                            | La latencia de las consultas a la herramienta de tickets ITSM no afecta la funcionalidad de la automatización.  |
| <b>RN04</b> | Storage admin response to Automation CLI menus | El administrador de storage no tiene tiempo límite en elegir una opción del menú de la CLI.   |
| <b>RN05</b> | Type of files to restore                       | Todo tipo de archivos son permitidos a ser restaurados (.docx, .xlsx, .pdf, .jpeg, .accdb, .sql, etcétera)  |

## A2. ESTUDIO DE FACTIBILIDAD

[ESP] Con base en el análisis tecnológico, financiero, de tiempos, esfuerzo y recursos humanos, el proyecto "Automatización de Restauración de Archivos" se considera factible desde la perspectiva de la viabilidad de los elementos en el entorno de desarrollo. El análisis de cada elemento se muestra a continuación.

### Feasibility Study

[ENG] Based on the technological, financial, time, effort and human resource analysis, File Restore Automation project is considered to be feasible since the viability of the elements in the development environment. Each element analysis is shown below.

### Technology

- Hardware. This Automation can live in either virtual or physical server. Host should have at least:
  - ❖ 4 GB RAM.
  - ❖ 1.66 GHz CPU one core.
  - ❖ NIC for network access.
- Development instances involved:
  - ❖ Python
  - ❖ PowerShell Scripting
  - ❖ SQL
  - ❖ CSS
  - ❖ HTTP REST Verbs

### Kilo Lines of Code (KLoC) estimation

| KLoC estimation           |                         |
|---------------------------|-------------------------|
| Language complexity value | SQL(13)+Python(20) = 33 |
| LoC                       | 1475.76                 |
| KLoC                      | <b>1.48</b>             |

## Effort, development time & human resources

| <b>Resolution effort-time-people</b> |             |
|--------------------------------------|-------------|
| Effort People/Month                  | <b>3.61</b> |
| Development time (months)            | <b>4.07</b> |
| Number of persons                    | <b>0.89</b> |

## Finance

| <b>Resolution person-month payment</b> |                       |
|--|-----------------------|
| Payment person/month (30,000 MXN)      | 30,150.82 MXN         |
| Payment during months/persons          | <b>122,788.21 MXN</b> |

| <b>Various</b>  |                      |
|-----------------|----------------------|
| Office rent     | 5,000 MXN            |
| Electricity     | 300 MXN              |
| Internet        | 600 MXN              |
| Travel expenses | 0                    |
| Monthly         | 5,900 MXN            |
| Monthly total   | <b>24,027.55 MXN</b> |

| <b>Project results</b>    |                       |
|---------------------------|-----------------------|
| Months                    | 4                     |
| Persons                   | 1                     |
| <b>Project total cost</b> | <b>146,815.77 MXN</b> |

## COCOMO support values

### Non-Adjustable Function Point values

| Values                   |    |
|--------------------------|----|
| Inputs                   | 6  |
| Outputs                  | 12 |
| Petitions                | 4  |
| Logical Internal Files   | 14 |
| External Interface Files | 7  |

#### Inputs

| Number | Input           | Files | Data elements | Level | Value |
|--------|-----------------|-------|---------------|-------|-------|
| 1.     | Ticket metadata | 1     | 15            | Low   | 3     |
| 2.     | CLI menu input  | 1     | 3             | Low   | 3     |

Total value inputs = **6**

#### Outputs

| Number | Output               | Files | Data elements | Level | Value |
|--------|----------------------|-------|---------------|-------|-------|
| 1.     | CLI ticket info      | 1     | 6             | Low   | 4     |
| 2.     | CLI selectable menus | 1     | 3             | Low   | 4     |
| 3.     | Notification email   | 1     | 2             | Low   | 4     |

Total value outputs = **12**

#### Petitions

| Number | Petition                 | Files | Data elements | Level  | Value |
|--------|--------------------------|-------|---------------|--------|-------|
| 1.     | NAS device communication | 1     | 60            | Medium | 4     |

Total value petitions = **4**



**Logical Internal Files**

| <b>Number</b> | <b>File</b>           | <b>Register item<br/>(table)</b> | <b>Data elements<br/>(attributes)</b> | <b>Level</b> | <b>Value</b> |
|---------------|-----------------------|----------------------------------|---------------------------------------|--------------|--------------|
| 1.            | Storage Identifier DB | 1                                | 4                                     | Low          | 7            |
| 2.            | NAS Login Credential  | 1                                | 8                                     | Low          | 7            |

Total value internal files = **14**

**External Interface Files**

| <b>Number</b> | <b>File</b>      | <b>Register item</b> | <b>Data elements</b> | <b>Level</b> | <b>Value</b> |
|---------------|------------------|----------------------|----------------------|--------------|--------------|
| 1.            | NN Training data | 1                    | 100                  | Medium       | 7            |

Total value external files = **7**

## Adjustable Function Point values

| <b>Number</b> | <b>Item</b>                  | <b>Value</b> |
|---------------|------------------------------|--------------|
| 1.            | Data communications          | 5            |
| 2.            | Data Processing Distribution | 4            |
| 3.            | Performance                  | 3            |
| 4.            | Configuration level to use   | 3            |
| 5.            | Transactions                 | 4            |
| 6.            | Online data entry            | 5            |
| 7.            | Terminal efficiency          | 4            |
| 8.            | Online registers             | 2            |
| 9.            | Process complexity           | 3            |
| 10.           | Reusability                  | 1            |
| 11.           | Installation                 | 1            |
| 12.           | Operations                   | 3            |
| 13.           | Multiple sites               | 0            |
| 14.           | Ease of change               | 1            |
|               | <b>Total</b>                 | <b>39</b>    |