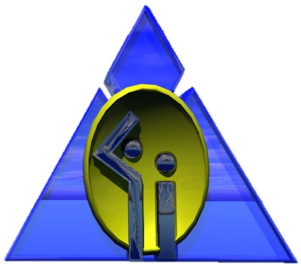

IV MEMORIA

CIENCIA Y TECNOLOGÍA INFORMÁTICA

ACTAS DEL "CONGRESO DE SOFTWARE LIBRE DEL XXI FLISOL LLALLAGUA– BOLIVIA"

Universidad Nacional "Siglo XX"

Ingeniería Informática

Instituto de Investigación y Desarrollo de

Aplicaciones Informáticas

Llallagua - Bolivia

Abril, 2025

ISBN: 979-8-2677-3565-0

Depósito Legal: 7-1-2565-2025

AUTORIDADES UNIVERSITARIAS

M.Sc. Odt. Boris Patiño Martínez
RECTOR

Ing. Ramiro Daga Canaviri
VICERRECTOR

M.Sc. Lic. Heriberto Richard Illanes Zeballos
DIRECTOR GENERAL ACADÉMICO

M.Sc. Ing. Miguel Ángel Terán Luna
DIRECTOR GENERAL DE INVESTIGACIÓN

Lic. Rocío Dueñas Plaza
DIRECTOR GENERAL DE EXTENSIÓN

Cc. Felix Tito Canaza Ramos
DIRECTOR FORMACIÓN POLÍTICO SINDICAL

Santos Ireneo Juchasara Colque, Ph.D.
DIRECTOR INGENIERÍA INFORMÁTICA

DIRECCIÓN Y EDICIÓN
Juan Pablo Luna Felipez, Ph.D.

EQUIPO TÉCNICO
Lea Fernanda Choque Martínez
Ilsen Arlette Corpa Limachi
Grace Salinas Mamani

PUBLICACIÓN

Instituto de Investigación y Desarrollo de Aplicaciones Informáticas (IIDAI) Ingeniería Informática
Universidad Nacional “Siglo XX”

AUTORES

Juan Pablo Luna Felipez, Ph.D.

Santos Juchasara Colque, Ph.D.

Leyna Salinas Veyzaga, Ph.D.

Valente Torija Pérez, M.Sc.

Ricardo Naranjo Faccini, M.Sc.

Jorge Ayala Niño de Guzmán, M.Sc.

Franz Villca Aro

Alexander Lino Fernandez Callapa

FLISOL

El Festival Latinoamericano de Instalación de Software Libre es el evento de mayor relevancia en América Latina dedicado a la promoción, difusión e instalación de software libre. En su vigésima primera edición, se celebró de manera simultánea en 17 países y 68 ciudades, consolidándose como un espacio de encuentro para entusiastas, profesionales, estudiantes y público en general interesados en la cultura del software libre y las tecnologías abiertas.

En la ciudad de Llallagua, Bolivia, la sede oficial fue la Universidad Nacional "Siglo XX", a través de la carrera de Ingeniería Informática y el Instituto de Investigación y Desarrollo de Aplicaciones Informáticas (IIDAI). El evento se llevó a cabo los días 25 y 26 de abril de 2025, y estuvo compuesto por dos actividades principales: el Congreso de Software Libre, donde se presentaron ponencias, talleres y experiencias sobre el uso, desarrollo e impacto del software libre, y el Install Fest, una jornada práctica de instalación y configuración de sistemas y aplicaciones libres.

El objetivo central de FLISOL es fomentar la adopción del software libre, promover la colaboración y el intercambio de conocimientos, y fortalecer la comunidad tecnológica local y regional.

Bajo el lema "El software libre es libertad para aprender, crear y compartir sin límites", FLISOL 2025 en Llallagua reafirmó el compromiso de la Universidad Nacional "Siglo XX" y su comunidad académica con la innovación, la educación tecnológica y la construcción colectiva del conocimiento abierto.

PRESENTACIÓN

Nos complace presentar la IV Memoria de Ciencia y Tecnología Informática, que reúne las actas del Congreso de Software Libre en el marco de la vigésima primera edición del Festival Latinoamericano de Instalación de Software Libre - FLISOL 2025 de la sede oficial en la Universidad Nacional "Siglo XX", carrera de Ingeniería Informática, en Llallagua, Bolivia.

FLISOL es el evento de mayor alcance en Latinoamérica dedicado a la promoción y difusión de la cultura del Software Libre. En su versión número 21, se llevó a cabo de manera simultánea en 17 países y 68 ciudades, consolidándose como un espacio de encuentro, aprendizaje y colaboración para entusiastas, profesionales y estudiantes de la informática y la tecnología.

En nuestra sede, el evento fue organizado por el Instituto de Investigación y Desarrollo de Aplicaciones Informáticas (IIDAI), y contó con una agenda diversa que incluyó el Congreso de Software Libre y el tradicional Install Fest, realizados el viernes 25 y sábado 26 de abril de 2025, respectivamente. Estas actividades permitieron el intercambio de conocimientos, la instalación y configuración de software libre, así como la reflexión sobre los desafíos y oportunidades que ofrece la adopción de tecnologías abiertas en nuestra región.

La presente memoria recopila las ponencias, experiencias y resultados compartidos durante el congreso, reflejando el compromiso de nuestra comunidad universitaria con la innovación, la libertad tecnológica y la construcción colectiva del conocimiento. Agradecemos la participación activa de todos los asistentes, ponentes y organizadores, quienes hicieron posible este evento.

Juan Pablo Luna Felipez, Ph.D.
ORGANIZADOR FLISOL LLALLAGUA - BOLIVIA
UNIVERSIDAD NACIONAL "SIGLO XX"

ÍNDICE

TALLER DE INSTALACIÓN DE SERVIDOR WEB CON NGINX, MARIADB Y PHP EN SISTEMA ROCKY LINUX 6

HERRAMIENTAS DE SOFTWARE LIBRE PARA CIBERSEGURIDAD 11

APLICACIONES DE SOFTWARE LIBRE EN LA INGENIERÍA DEL AGUA Y CIENCIAS DE LA GEO-INFORMACIÓN 31

INTELIGENCIA ARTIFICIAL: PROCESAMIENTO DEL LENGUAJE NATURAL CON SOFTWARE LIBRE 36

ANÁLISIS DE SENTIMIENTOS CON PYTHON 48

ADMINISTRA SERVIDORES LINUX COMO UN PRO (¡SIN MORIR EN EL INTENTO!) 52

AUTOMATIZACIÓN CONVENCIONAL MULTILINGÜE EN TELEGRAM USANDO HUGGING CHAT UN ENFOQUE CON CONTEXTO PERSONALIZADO 60

GPT4ALL: AUTOMATIZACIÓN, ANÁLISIS Y PRIVACIDAD CON LA IA LOCAL 65

TALLER DE INSTALACIÓN DE SERVIDOR WEB CON NGINX, MARIADB Y PHP EN SISTEMA ROCKY LINUX

M.Sc. ING. VALENTE TORIJA PÉREZ

torijavalente@itat.edu.mx

Ingeniería de Tecnologías de la Información y Comunicaciones

Instituto Tecnológico del Altiplano de Tlaxcala

México

RESUMEN

"Es importante que estudiantes, profesionistas y público interesado conozca una alternativa adicional, como es el caso del servidor Web NginX ya que en los últimos años se ha consolidado como uno de los servidores web más utilizados en el mundo, superando en muchos casos al servidor Apache, especialmente en sitios web de alto tráfico y entornos de alto rendimiento."

1. INTRODUCCIÓN

Contexto y relevancia.

Los servidores web son fundamentales para alojar aplicaciones y páginas web.

Nginx ha surgido como una alternativa eficiente a Apache, especialmente en entornos de alto tráfico.

Rocky Linux es una distribución estable y compatible con RHEL, ideal para servidores empresariales.

La combinación LEMP (Linux, Nginx, MariaDB, PHP) permite desplegar

aplicaciones web dinámicas con alto rendimiento.

2. DESARROLLO

Objetivos del Taller.

- Instalar y configurar Nginx como servidor web.
- Implementar MariaDB para gestión de bases de datos.
- Integrar PHP para contenido dinámico.
- Verificar el funcionamiento del Servidor LEMP.

Justificación

- Nginx en la actualidad es el Servidor Web más usado a nivel global (W3Techs, 2025) por su eficiencia en manejo de conexiones concurrentes.
- Rocky Linux ofrece soporte a largo plazo (LTS), relevante para entornos

Métodos

Para ésta implementación se requieren conocimientos previos, pero no será indispensable.

- Manejo básico del sistema operativo GNU/Linux.
- Conocimiento básico de redes.
- Conocimientos básicos en el uso de servidor.

Proceso para su implementación.

Requisitos previos mínimos

- Hardware: Máquina virtual con 4GB RAM, 40GB disco, procesador Intel i3/AMD equivalente o superior.
- Software Oracle VirtualBox como hipervisor.
- Imágen ISO de Rocky Linux.

Instalación y configuración

1. Preparación del sistema:

- Actualización de paquetes: dnf update

2. Instalación de Nginx:

- Instalación y activación del servicio:

  dnf install nginx

  systemctl start nginx

  systemctl status nginx

  systemctl enable nginx

- Configuración del firewall:

  firewall-cmd --permanent –add-service=http --zone=public

  firewall-cmd --reload

3. Instalación de MariaDB

- Instalación y secuenciado:

  dnf install mariadb-server

  systemctl start mariadb

  systemctl status mariadb

  systemctl enable mariadb

  mysql_secure_installation # Configuración de seguridad (contraseña root, eliminar usuarios anónimos, entre otros).

4. Integración de PHP

- Instalación de PHP y módulo MySQL:

  dnf install php php-mysqlnd

- Creación de archivo de prueba en la siguiente ruta /usr/share/nginx/html/test.php

Captura el siguiente código de php y guardarlo con el nombre test.php:

<?php

phpinfo(); ?>

- **Reinicio de Nginx:**

  systemctl restart nginx

Evidencias del éxito en la implementación del Servidor LEMP.

- ➢ Acceso a la página predeterminada de Nginx vía navegador Web
- ➢ (http://<Dirección IP-del-servidor>)
- ➢ Visualización correcta de la ejecución de PHP mediante la llamada al archivo phpinfo() en la dirección web http://<Dirección IP-del-servidor>/test.php.
- ➢ Estado activo de servicios verificado con: systemctl status nginx mariadb php-fpm

Posibles errores y soluciones:

- PHP no se ejecuta: Verificar que el servicio php-fpm esté activo y Nginx configurado para procesar archivos .php
- Firewall bloquea conexiones: Asegurarse que los puertos 80 (HTTP) y 443 (HTTPS) estén abiertos.

3. CONCLUSIÓN

Se tiene planeado concluir el Taller LEMP funcionando en Rocky Linux con la implementación exitosa junto con participantes y asistentes.

Al concluir el Taller se llevarán los siguientes conocimientos:

- Gestión de servicios con systemd.
- Configuración de reglas del firewall con firewalld.
- ntegrar componentes clave para aplicaciones web dinámicas.

Recomendaciones y aplicaciones futuras para el siguiente Taller en marco del siguiente FliSol 2026.

- Implementar SSL/TLS con Let’s Encrypt para seguridad.
- Explorar configuraciones avanzadas de Nginx (balanceo de carga, caché).
- Usar herramientas como phpMyAdmin para gestión gráfica de MariaDB.

A manera de conclusión. El taller demostró la viabilidad de utilizar el Sistema Rocky Linux como plataforma para servidores web modernos, combinando rendimiento (Nginx) y flexibilidad (PHP + MariaDB).

REFERENCIAS

Documentación oficial de Nginx, Rocky Linux, y MariaDB.

https://blog.vtorija.com

SOBRE EL AUTOR

Valente Torija Pérez es informático por el Instituto Tecnológico de Apizaco y Maestro en Nuevas Tecnologías por la Universidad Iberoamericana Campus Puebla. Además, cuenta con una formación de Linux Professional Institute como Linux Administrator LPIC-1.

Con más de 20 años de experiencia, ha trabajado con aplicaciones y soluciones basadas en software libre, así como en servicios de Tecnologías de Información. Desde 2008, se ha desempeñado en el Estado de Tlaxcala como docente y directivo del programa educativo de Técnico Superior Universitario en TIC e Ingeniería en Tecnologías de la Información y Comunicaciones en la Universidad Tecnológica de Tlaxcala. También ha sido docente en la Universidad Metropolitana de Tlaxcala y en la Universidad Politécnica de Tlaxcala Región Poniente.

Actualmente, es docente de tiempo completo en el Instituto Tecnológico del Altiplano de Tlaxcala (ITAT), donde ha desempeñado roles como Webmaster, Jefe del Departamento de Gestión Tecnológica y Vinculación, y Jefe de Centro de Cómputo, recientemente dejó la presidencia de academia de la Ingeniería en Tecnologías de la Información y Comunicaciones. En la actualidad, continúa desarrollándose como docente, impartiendo clases en las Ingenierías de Agronomía y Tecnologías de la Información y Comunicaciones.

Sus principales áreas de interés incluyen el impacto del software libre en las organizaciones, el diseño e impartición de cursos, tecnología educativa, desarrollo web, implementación y administración de servidores, tecnologías de virtualización, contenedores LXC, cómputo de alto desempeño, desarrollo de objetos virtuales de aprendizaje, implementación y administración de servidores Moodle y servicios de correo con Google Workspace, usuario de sistemas GNU/Linux Debian, Rocky Linux y Ubuntu.

Figura 1: Fotografía de presentación de la ponencia

HERRAMIENTAS DE SOFTWARE LIBRE PARA CIBERSEGURIDAD

M.Sc. ING. RICARDO NARANJO FACCINI

gerencia@skinait.com

Ingeniería Civil

Universidad de los Andes

Bogotá, Colombia

RESUMEN

Se proporciona una visión general de diversas herramientas de software libre aplicables a la ciberseguridad, organizadas en cinco categorías funcionales: ***Preparativas:*** *Estas herramientas están diseñadas para facilitar la planificación y la preparación de medidas de seguridad. Incluyen herramientas para llevar el inventario de activos o para redactar las políticas y el SGSI;* ***Preventivas:*** *Estas herramientas se utilizan para evitar que ocurran incidentes de seguridad. Incluyen software para el análisis de vulnerabilidades, gestión de configuraciones y control de accesos, que ayudan a prevenir ataques y errores antes de que ocurran;* ***Detectivas:*** *Las herramientas en esta categoría están enfocadas en la identificación de problemas y anomalías en tiempo real. Incluyen sistemas de monitoreo y detección de intrusiones que permiten a los equipos de seguridad identificar amenazas y actividades sospechosas de manera proactiva;* ***Reactivas:*** *Estas herramientas se emplean para responder a incidentes de seguridad una vez que han ocurrido. Incluyen plataformas para el*

análisis forense y la gestión de incidentes, que permiten investigar y mitigar los efectos de los ataques; ***Retrospectivas:*** *Aunque no se identificaron herramientas específicas de software libre para actividades retrospectivas, las funciones correspondientes se encuentran integradas en las herramientas de las categorías anteriores. Estas herramientas permiten realizar evaluaciones y análisis post-incidente para mejorar la seguridad a largo plazo.*

El artículo destaca cómo estas herramientas, al ser utilizadas en conjunto, proporcionan una cobertura integral para la gestión de la seguridad en diferentes fases del ciclo de vida de los incidentes.

Es importante destacar que, si bien el hecho de que un software esté protegido por una licencia de software libre no garantiza su calidad o seguridad, es necesario revisar la actividad de la comunidad que rodea a la herramienta, como la generación de tutoriales en diversos idiomas y la frecuente generación de nuevas versiones con correcciones de errores y nuevas funcionalidades, como una medida de garantía de calidad y seguridad.

1. INTRODUCCIÓN

La popularidad del software libre ha crecido significativamente en los últimos años, ya que cada vez más personas y organizaciones reconocen las ventajas de utilizar software de código abierto en lugar de software privativo. Una de las áreas donde el software libre ha ganado terreno es en la ciberseguridad, ya que existen numerosas herramientas de seguridad de código abierto disponibles que son utilizadas por profesionales de la seguridad en todo el mundo.

Sin embargo, es importante destacar que el hecho de que un software esté protegido por una licencia de código abierto no es garantía de calidad ni de seguridad. Lo mismo sucede con el software privativo. Aunque el código abierto permite que cualquiera pueda ver el código fuente y modificarlo, esto no significa que el software sea seguro o esté libre de errores. De hecho, algunos proyectos de software libre y privativo pueden contener errores o vulnerabilidades que pueden ser explotados por atacantes malintencionados.

Entonces, ¿cómo podemos estar seguros de que el software de seguridad que estamos utilizando es seguro y confiable? La respuesta radica en la comunidad que rodea al proyecto. Cuando se trata de software, la comunidad de desarrolladores y usuarios es crucial para garantizar la calidad y seguridad del software.

Una comunidad activa y comprometida en torno a un proyecto de software es una buena señal de que el software es seguro y confiable. Una comunidad activa significa que hay muchos desarrolladores y usuarios trabajando en el proyecto, discutiendo los problemas y soluciones, y compartiendo información sobre el software.

Además, una comunidad activa también significa que hay muchas conversaciones en línea, tutoriales y documentación disponible en varios idiomas, lo que facilita el aprendizaje y la utilización del software. Otro indicador de calidad y seguridad del software es la frecuencia con la que se publican nuevas versiones y actualizaciones. Los proyectos de software con una comunidad activa suelen publicar nuevas versiones con frecuencia, lo que indica que están corrigiendo errores y añadiendo nuevas funcionalidades de forma constante. La publicación frecuente de nuevas versiones es un buen indicador de que el software está siendo mantenido y actualizado activamente.

2. DESARROLLO

Clasificación del software libre para ciberseguridad

El software libre para ciberseguridad tiene diversas aplicaciones y se clasifica de acuerdo con el tipo de actividad de ciberseguridad que atiende, las cuales pueden ser: Preparativas, Preventivas, Detectivas, Reactivas o Retrospectivas. Cabe aclarar que algunas herramientas de software libre pueden atender a varias de éstas actividades.

Para actividades Preparativas

a) Acceso remoto

- **ssh:** secure shell, es un comando utilizado para conectarse a un servidor remoto de forma segura utilizando el protocolo SSH. Al ejecutar el comando ssh, se puede ingresar la dirección IP o el nombre de dominio del servidor y establecer una sesión de terminal remota segura.

- **gsh:** el group shell es similar al ssh, con la diferencia que los comandos que se ingresan en el computador maestro queda inmediatamente replicado en los otros computadores.

- **Xwindows:** El entorno gráfico de UNIX/Linux es de resaltar, dado que desde su creación en los años 60 fue pensado como un entorno de red y el despliegue remoto de aplicativos siempre ha estado presente en éste tipo de entornos, tanto para desplegar aplicativos en ventanas individuales como para todo el escritorio. Se puede utilizar para entornos de programación y despliegue remoto, facilitando la administración y ejecución de herramientas de ciberseguridad en máquinas remotas a través de interfaces gráficas, sin necesidad de acceso físico.

b) Documentación del SGSI

- **LibreOffice y OpenOffice:** La suite de oficina de UNIX/Linux es tan poderosa como las diferentes versiones de ellas en el mundo privativo, se encontrarán algunas funcionalidades más poderosas o menos que en el mundo privado, al igual que se destacarán funcionalidades novedosas como el graficador de ecuaciones matemáticas y algunas funcionalidades que se consiguen en el mundo

privado harán falta en éstas dos suites. Sin embargo, a nivel general pueden equipararse a las más famosas.

c) Automatización

- **crontab:** es un comando utilizado para programar tareas y comandos para que se ejecuten en momentos específicos, utilizando el cron daemon. Al ejecutar el comando crontab, se puede editar el archivo de configuración cron del usuario y programar tareas para que se ejecuten en diferentes momentos y frecuencias.

d) Protección de contraseñas

La protección de contraseñas es una herramienta que puede proteger los datos personales y de inicio de sesión almacenados en un ordenador. Puede almacenar contraseñas de forma segura y generar contraseñas fuertes para una mayor seguridad.

- **KeePass:** Es un administrador de contraseñas de código abierto y gratuito que se utiliza para almacenar y gestionar contraseñas encriptadas en una base de datos segura. KeePass ofrece una forma fácil y segura de almacenar todas las contraseñas y credenciales importantes en un solo lugar. La base de datos de KeePass está protegida por una contraseña maestra y puede ser encriptada con varios algoritmos de cifrado para proporcionar un nivel adicional de seguridad.

e) Gestión de activos

- **GLPI:** es una herramienta de gestión de activos de TI y de mesa de ayuda de código abierto. Se utiliza para gestionar la información y los problemas relacionados con los recursos de TI, incluyendo la gestión de inventario, la gestión de incidentes y la gestión de cambios. GLPI también puede utilizarse para realizar seguimiento y resolver problemas de seguridad.

Para actividades Preventivas

a) Inteligencia

- **MISP (Malware Information Sharing Platform):** es una plataforma de compartición de información sobre amenazas. Permite compartir información sobre amenazas de forma segura entre organizaciones para mejorar la detección y respuesta a amenazas.

  Es una plataforma de software libre para compartir, almacenar y correlacionar indicadores de compromiso (IoC) y otros datos de inteligencia de amenazas cibernéticas. MISP permite a las organizaciones colaborar y mejorar su defensa mediante la inteligencia de amenazas compartida.

- **Maltego:** Una poderosa herramienta de OSINT (Open Source Intelligence) que permite la visualización de relaciones y conexiones entre entidades. En ciberseguridad, se utiliza para investigaciones de inteligencia, mapeando redes sociales, dominios, correos electrónicos, y más, para descubrir vínculos que puedan revelar amenazas o actores maliciosos.

- **T-Pot:** Es un honeypot de código abierto listo para usar que está diseñado para atraer y analizar las actividades de ciberdelincuentes. T-Pot permite a los equipos de ciberseguridad monitorear ataques en tiempo real y recopilar información valiosa sobre técnicas y tácticas utilizadas por los atacantes.

b) Criptografía

- **Criptografía en una dirección:** El comando md5sum se utiliza para calcular el hash MD5 de un archivo, lo que permite verificar la integridad de su contenido. Por ejemplo, "md5sum archivo.txt" generaría el hash MD5 del archivo.txt. sha256sum es un comando utilizado para calcular el valor hash de un archivo utilizando el algoritmo SHA-256. Al ejecutar el comando sha256sum, se puede calcular el hash de un archivo y verificar su integridad, ya que cualquier cambio en el archivo cambiará su hash.

- **Criptografía simétrica:** El comando openssl permite cifrar y descifrar archivos utilizando algoritmos de cifrado simétrico como AES o DES. Por ejemplo, "openssl enc -aes256 -in archivo.txt -out archivo.cifrado" cifrar el archivo.txt con AES-256 y generar el archivo cifrado archivo.cifrado.

- **Criptografía asimétrica:** El comando openssl también se puede utilizar para generar claves y certificados digitales para la criptografía asimétrica. Por ejemplo, "openssl genpkey -algorithm RSA -out clave.privada" generaría una clave privada RSA y "openssl req -new -key clave.privada -out certificado.csr" generaría una solicitud de certificado digital que se puede enviar a una autoridad de certificación para obtener un certificado digital que se puede utilizar para cifrar y descifrar datos utilizando la clave pública correspondiente a la clave privada generada.

c) Backups

- **rsync:** Es un comando utilizado para sincronizar archivos y directorios entre sistemas, de forma segura y eficiente. Al ejecutar el comando rsync, se pueden sincronizar archivos y directorios entre sistemas utilizando diferentes protocolos de transferencia de archivos, como SSH. Es valioso para replicar datos de manera segura y eficiente entre diferentes sistemas o para crear respaldos en tiempo real.

- **grsync:** Es una interfaz gráfica para rsync, que facilita la configuración y el uso de rsync para usuarios que prefieren trabajar en un entorno visual. Es útil en la gestión de sincronización y backups seguros en sistemas de ciberseguridad.

- **borg:** Es una herramienta de respaldo (backup) que se especializa en crear copias de seguridad eficientes, seguras y comprimidas. En ciberseguridad, Borg es útil para asegurar datos críticos y garantizar la continuidad de la operación en caso de incidentes de seguridad.

- **pikabackup:** Es una interfaz gráfica para Borg, facilitando a los usuarios menos experimentados la creación y gestión de copias de seguridad sin necesidad de utilizar la línea de comandos. Es útil para realizar backups en entornos donde la ciberseguridad es prioritaria.

d) Antimalware

Un antivirus es una herramienta que protege un sistema contra malware y virus. Puede escanear archivos y programas para detectar y eliminar cualquier amenaza.

- **ClamAV:** Es una herramienta de seguridad de código abierto utilizada para detectar y eliminar virus, troyanos, malware y otras amenazas en sistemas Unix y Windows. ClamAV es una herramienta gratuita y multiplataforma que utiliza una base de datos actualizada de firmas de virus para escanear archivos y directorios en busca de amenazas. ClamAV también se puede integrar en sistemas de correo electrónico y escanear el tráfico entrante y saliente para detectar correos electrónicos con archivos adjuntos maliciosos previniendo que lleguen a los sistemas Windows que puedan ser afectados por el malware entrante.

- **Pdf-parser.py:** es un script de Python que se utiliza para analizar archivos PDF y extraer información de ellos. Este script es de código abierto y se puede descargar de forma gratuita desde su sitio web. Pdf-parser.py puede ser utilizado para extraer objetos PDF como fuentes, imágenes, scripts, y para analizar la estructura interna de un archivo PDF. También puede utilizarse para decodificar objetos codificados en Base64 y FlateDecode, lo que puede ser útil para analizar archivos maliciosos.

- **Pdftools:** es otra herramienta de software libre para analizar archivos PDF. Esta herramienta está diseñada para escanear archivos PDF en busca de virus y malware. Además, pdftools puede ser utilizado para extraer información de los archivos PDF, incluyendo metadatos, fuentes, scripts, y objetos embebidos.

- **RKhunter:** Es una herramienta de seguridad que busca detectar malware, rootkits y otras posibles amenazas en sistemas Linux y Unix. La herramienta realiza comprobaciones en los archivos del sistema, las bibliotecas compartidas y los binarios, para detectar cualquier alteración inesperada. También verifica los servicios activos y los puertos abiertos en busca de posibles vulnerabilidades.

e) Capacitación y sensibilización

- **Moodle:** Es una plataforma de aprendizaje en línea de código abierto. En ciberseguridad, Moodle se utiliza para capacitar y sensibilizar al personal en temas de seguridad, proporcionando un entorno estructurado para cursos, talleres y materiales educativos sobre ciberseguridad.

- **Jitsi Meet:** Es una solución de videoconferencia de código abierto que garantiza la privacidad y seguridad en las comunicaciones. Es ideal para reuniones internas, capacitaciones y cualquier otra interacción remota en entornos donde la seguridad es crítica.

- **OBS (Open Broadcaster Software):** Es una herramienta de código abierto para la grabación de video y transmisión en vivo. En ciberseguridad, OBS puede utilizarse para crear videotutoriales, webinars, y otras formas de contenido educativo o de sensibilización en seguridad informática.

- **Pitivi y OpenShot:** Son editores de vídeo de código abierto. En ciberseguridad, estas herramientas son útiles para editar videos educativos, tutoriales, o registros visuales de eventos de capacitación, brindando una forma efectiva de compartir conocimiento en seguridad.

f) Detección de vulnerabilidades

- **Nessus:** identificador de vulnerabilidades que permite escanear sistemas y aplicaciones en busca de vulnerabilidades conocidas y desconocidas.

- **Nikto:** Escáner de vulnerabilidades de servidores web por medio de línea de comandos, busca en particular archivos/CGI peligrosos, software de servidor obsoleto y otros problemas.
- **Lynis:** es una herramienta de auditoría de seguridad que escanea sistemas UNIX y Linux en busca de vulnerabilidades. Proporciona informes detallados sobre la configuración del sistema, la seguridad y las posibles vulnerabilidades.

- **OpenVAS (Open Vulnerability Assessment System):** Es un marco de software libre para escaneo y gestión de vulnerabilidades. En ciberseguridad, OpenVAS permite identificar y evaluar posibles vulnerabilidades en los sistemas, ayudando a las organizaciones a fortalecer su postura de seguridad.

- **AlienVault OSSIM:** Es una plataforma unificada de gestión de eventos e información de seguridad (SIEM) de código abierto. En ciberseguridad, AlienVault permite la correlación de eventos de seguridad, detección de amenazas y gestión de incidentes, integrando múltiples herramientas de seguridad en un solo panel.

g) Pruebas de penetración

- **Metasploit:** Es una plataforma de pruebas de penetración (pentesting) ampliamente utilizada que permite a los profesionales de la seguridad realizar ataques simulados para identificar vulnerabilidades en sistemas informáticos. Metasploit ofrece una vasta base de datos de exploits, payloads y herramientas auxiliares que permiten realizar pruebas de penetración completas, desde la explotación de vulnerabilidades hasta la post-explotación y el análisis de la seguridad de los sistemas.

- **Armitage:** Es una interfaz gráfica para Metasploit que facilita la ejecución de pruebas de penetración y ciberataques simulados. Armitage permite la automatización de muchas tareas complejas, como la selección de exploits y payloads, y proporciona una vista gráfica de los objetivos comprometidos. Es ideal para quienes prefieren un entorno visual para gestionar sus pruebas de seguridad, especialmente en entornos colaborativos.

- **Kali Linux:** Es una distribución de Linux basada en Debian diseñada específicamente para pruebas de penetración y auditorías de seguridad. Incluye una vasta colección de herramientas preinstaladas para realizar pruebas de seguridad en redes, aplicaciones web, sistemas operativos y más. Kali Linux es la herramienta predilecta de los profesionales de la ciberseguridad para realizar análisis de vulnerabilidades, pruebas de penetración, y forense digital.

- **Cyborg Hawk Linux**: Es otra distribución de Linux orientada a la ciberseguridad, que al igual que Kali Linux, ofrece una amplia gama de herramientas para pruebas de penetración, análisis forense, y auditorías de seguridad. Cyborg Hawk se destaca por su enfoque en la seguridad ofensiva y defensiva, y su entorno personalizable, lo que la convierte en una opción potente para investigadores de seguridad y profesionales del pentesting.

- **Paladin Linux:** Es una distribución basada en Ubuntu diseñada para investigaciones forenses digitales. Incluye una colección de herramientas forenses para la recuperación de datos, análisis de discos, y la creación de imágenes forenses. Paladin Linux es ideal para investigadores que necesitan una solución portátil y completa para realizar análisis forense de dispositivos digitales y redes.

h) Control de red

- **Bind:** también conocido como Berkeley Internet Name Domain, es un servidor de nombres de dominio (DNS) de código abierto. Se utiliza para traducir nombres de dominio en direcciones IP y viceversa. Bind es compatible con muchos sistemas operativos, incluyendo Linux, Unix y Windows. Es una herramienta esencial para cualquier red que utilice DNS.

- **E2guardian:** Es un proxy web de código abierto que se utiliza para filtrar contenido en la red. Proporciona filtros de contenido para bloquear sitios web inapropiados o peligrosos, y es capaz de bloquear el acceso a contenido basado en palabras clave y categorías. E2guardian también puede utilizarse para monitorizar y registrar el tráfico de la red.

- **WireGuard:** es un protocolo de VPN de código abierto. Se utiliza para establecer conexiones VPN seguras entre dispositivos y redes. WireGuard utiliza criptografía de última generación para proteger las conexiones y es muy rápido y eficiente en términos de recursos. Además, WireGuard es fácil de configurar y utilizar.

- **Squid:** es un servidor proxy de código abierto que se utiliza para acelerar y optimizar el tráfico web. Squid se utiliza comúnmente para cachear contenido web y reducir la carga de los servidores web.

También puede utilizarse para filtrar y bloquear el acceso a sitios web no deseados. Squid es compatible con muchos sistemas operativos y es muy configurable.

- **Cortafuegos de escritorio:** Un cortafuegos de escritorio es una herramienta que controla el tráfico de red entrante y saliente en un ordenador individual. Puede bloquear el acceso no autorizado a un sistema y prevenir ataques externos.

- **iptables:** es un comando utilizado para configurar y administrar el firewall del sistema, incluyendo reglas de filtrado de paquetes y configuraciones de NAT. Al ejecutar el comando iptables, se pueden agregar, eliminar y modificar reglas del firewall y configuraciones de red.

- **Endian Firewall:** Es una distribución de seguridad basada en Linux que combina múltiples funciones de ciberseguridad en una sola solución, conocida como UTM (Unified Threat Management). Endian Firewall incluye firewall, VPN, antivirus, filtrado de contenido web, detección y prevención de intrusiones (IDS/IPS), y más. Es especialmente útil en pequeñas y medianas empresas que necesitan una solución todo en uno para proteger sus redes contra una variedad de amenazas sin necesidad de configurar y mantener múltiples herramientas separadas.

- **ModSecurity:** Es un firewall de aplicaciones web (WAF) de código abierto que proporciona protección contra una amplia gama de ataques web. Se integra con servidores web como Apache y Nginx, siendo esencial para proteger aplicaciones web contra inyecciones SQL, XSS, y otros ataques.

- **pfSense:** Es una distribución de firewall y enrutador basada en FreeBSD, ampliamente utilizada en entornos de ciberseguridad. pfSense ofrece características avanzadas de firewall, VPN (Red Privada Virtual), detección de intrusiones (a través de plugins como Snort), y filtrado de contenido, todo configurable a través de una interfaz web intuitiva. Es ideal para proteger redes corporativas, proporcionando una capa adicional de seguridad en el control del tráfico de red, segmentación de redes, y acceso remoto seguro.

Para actividades Detectivas

a) Auditoría de contraseñas

- **John the Ripper:** herramienta de cracking de contraseñas que permite probar diferentes combinaciones de contraseñas para romper la seguridad de cuentas de usuario. Se puede utilizar para auditar las contraseñas de los usuarios de una organización identificando quienes utilizan contraseñas débiles. Descifra contraseñas probando todas las combinaciones posibles o utilizando listas de palabras comunes mediante el manejo de una amplia gama de algoritmos de hash, incluidos MD5, SHA-1, SHA-256, entre otros.

- **Hydra:** Es una herramienta de prueba de penetración de redes que se utiliza para adivinar contraseñas. Permite probar miles de posibles contraseñas en un corto periodo de tiempo,

utilizando diferentes métodos, como el diccionario, la fuerza bruta y el ataque híbrido. Al igual que John the Ripper puede ser utilizada para auditar las contraseñas débiles que usen los usuarios de un sistema de información.

- **Hashcat:** Es una herramienta de cracking de contraseñas que utiliza la potencia de cálculo de la GPU para realizar ataques de fuerza bruta y ataques de diccionario contra contraseñas cifradas con algoritmos de hash. Está optimizado para aprovechar la potencia de cálculo de la GPU, lo que le permite realizar ataques de cracking de contraseñas a alta velocidad. Puede manejar una gran variedad de algoritmos de hash, incluidos MD5, SHA-1, SHA-256, bcrypt, entre otros

- **Aircrack-NG:** Es una suite de herramientas de seguridad inalámbrica que se utiliza para auditorías de redes Wi-Fi. Permite el monitoreo de redes, captura de paquetes, inyección de paquetes, y análisis de tráfico para descifrar claves WEP y WPA/WPA2. Se utiliza para auditar y evaluar la seguridad de redes Wi-Fi. Permite realizar ataques de descifrado de claves WEP y WPA/WPA2 utilizando técnicas como fuerza bruta, diccionario y ataques basados en la inyección de paquetes. Proporciona herramientas para monitoreo y captura de paquetes de red, así como para el análisis y descifrado de tráfico cifrado.

b) Monitoreo de recursos del equipo

Un monitor de procesos es una herramienta que rastrea todos los procesos que se ejecutan en un sistema y puede identificar cualquier proceso malicioso o sospechoso. Puede valorar la cantidad porcentual de CPU o de memoria RAM que están consumiendo.

Procesos y recursos

- **ps:** es un comando utilizado para mostrar información sobre los procesos en ejecución en el sistema. Al ejecutar el comando ps, se muestra una lista de procesos en ejecución, cada uno con su identificador de proceso (PID), estado, uso de recursos y otros detalles. Es útil para identificar procesos específicos y detener o matar procesos si es necesario.

- **Kill:** se utiliza para detener un proceso en ejecución enviando una señal a su identificador de proceso (PID). El usuario debe especificar el PID del proceso que desea detener y la señal que se enviará al proceso. La señal predeterminada es SIGTERM, que indica al proceso que se detenga de manera ordenada, pero también se pueden enviar otras señales, como SIGKILL, que fuerza al proceso a detenerse inmediatamente.

- **Killall:** se utiliza para detener todos los procesos que tengan el mismo nombre. En lugar de especificar el PID de un proceso, el usuario especifica el nombre del proceso. Killall envía la señal SIGTERM a todos los procesos con el nombre especificado.

- **nice:** Es un comando de Unix/Linux que permite ajustar la "prioridad de ejecución" de un proceso antes de que se inicie. En ciberseguridad, nice puede ser útil para gestionar los recursos del sistema durante pruebas intensivas, como escaneos de vulnerabilidades o análisis forense, asegurando que

  estas tareas no ralentice otros servicios críticos en el sistema. Al ajustar la prioridad con nice, los administradores pueden optimizar el rendimiento del sistema, dándole más o menos recursos a ciertos procesos según lo requiera la situación.

- **renice**: Es un comando similar a nice, pero se utiliza para cambiar la prioridad de un proceso que ya está en ejecución. En ciberseguridad, renice es útil cuando se necesita ajustar dinámicamente el uso de recursos del sistema, por ejemplo, si un escaneo de seguridad está consumiendo demasiados recursos y se requiere reducir su prioridad para mantener la estabilidad del sistema. renice permite a los administradores intervenir en tiempo real para gestionar la carga del sistema y asegurar que las operaciones críticas continúen funcionando sin interrupciones.

- **top:** Es una herramienta de línea de comandos en Unix/Linux que muestra en tiempo real los procesos que se están ejecutando en el sistema, ordenados por el uso de recursos como la CPU, la memoria, y el tiempo de ejecución. En ciberseguridad, top es útil para monitorear el sistema en busca de procesos sospechosos o maliciosos que puedan estar consumiendo recursos inusuales, lo que podría indicar un compromiso de seguridad. También permite a los administradores de sistemas identificar rápidamente cuellos de botella y ajustar prioridades con herramientas como nice y renice.

- **htop:** Es una versión mejorada y más interactiva de top que ofrece una interfaz más amigable y visual. htop permite a los usuarios visualizar los procesos en una estructura en árbol, facilitando la identificación de procesos secundarios y la relación entre procesos. En ciberseguridad, htop es útil para monitorear de forma más intuitiva el uso de recursos del sistema, permitiendo a los administradores detectar patrones anómalos o procesos que podrían estar vinculados a actividades maliciosas, como malware o minería de criptomonedas no autorizadas.

- **inxi:** Es una herramienta de línea de comandos que proporciona una detallada información sobre el sistema, incluyendo el hardware, la configuración de red, y los dispositivos conectados. En ciberseguridad, inxi es especialmente útil para realizar auditorías de seguridad y obtener un rápido resumen del estado del sistema. Puede ayudar a los administradores a verificar la integridad del hardware y las configuraciones, identificar dispositivos desconocidos o no autorizados, y asegurarse de que la infraestructura cumple con las políticas de seguridad establecidas.

- **Acct:** Es un conjunto de herramientas que registra la actividad del sistema y de los usuarios en sistemas Linux y Unix. Permite a los administradores de sistemas supervisar el uso de los recursos del sistema y detectar posibles problemas de rendimiento. Además, la herramienta puede utilizarse para generar informes de uso de recursos para facturación o fines de contabilidad.

- **which:** es un comando que permite encontrar la ubicación de un archivo ejecutable en el sistema. Esta herramienta puede ser útil para verificar la integridad de los archivos del sistema y detectar archivos maliciosos que se hayan instalado en el sistema.

- **Auditd:** es una herramienta de auditoría que registra eventos del sistema en sistemas Linux y Unix. Permite a los administradores de sistemas supervisar la actividad de los usuarios y detectar posibles intrusiones. La herramienta registra eventos de inicio de sesión, cambios en los archivos del sistema y otros eventos importantes.

- **Monitores de recursos:** Un monitor de recursos es una herramienta que se utiliza para medir y visualizar la utilización del hardware de un sistema informático, como la CPU, la memoria, el disco y la red. Proporciona información sobre el uso actual y el historial de uso, lo que permite al usuario realizar un seguimiento del rendimiento del sistema y detectar cualquier problema.

- **El monitor del sistema de mate y ksysguard:** son herramientas gráficas utilizadas para monitorear el sistema y visualizar información sobre el uso de recursos, los procesos en ejecución y otros detalles del sistema. Estas herramientas pueden ser útiles para identificar procesos que están consumiendo recursos excesivos o para monitorear el rendimiento del sistema en general.

- **Baobab:** Es una herramienta gráfica para analizar el espacio en disco. Proporciona información detallada sobre el tamaño de los archivos y carpetas, permitiendo identificar fácilmente los archivos más grandes y liberar espacio en disco.

- **Glances:** Es un monitor de sistema en línea de comandos que proporciona una visión general del uso del sistema en tiempo real. Permite supervisar la CPU, memoria, carga del sistema, uso de la red, entre otros.

Usuarios

- **who:** Es un comando que muestra información sobre los usuarios que están actualmente conectados al sistema, incluyendo su nombre de usuario, terminal y hora de inicio de sesión. Esta información puede ser útil para monitorear la actividad del sistema y detectar intentos de acceso no autorizado.

- **lastcomm:** Es una herramienta de software libre para sistemas Unix y Unix-like que proporciona información sobre los comandos que se han ejecutado recientemente en el sistema. Este programa lee los registros de contabilidad del sistema (también conocidos como registros de procesos), que registran información sobre los procesos que se ejecutan en el sistema, y muestra una lista de los comandos que se han ejecutado, junto con detalles como el usuario que los ejecutó, la hora y la duración de la ejecución.

Red

- **ip:** es un comando utilizado para configurar y mostrar información sobre la red y las interfaces de red en el sistema. Al ejecutar el comando ip, se pueden realizar tareas como configurar direcciones IP, agregar rutas y mostrar información detallada sobre las interfaces de red.

- **Tcpdump:** Tcpdump es una herramienta de línea de comandos que permite capturar y analizar el tráfico de red en tiempo real. Es una herramienta muy útil para la detección de problemas de red y para la investigación de ataques en la red. Tcpdump puede ser utilizado para capturar y analizar paquetes en la red en diferentes formatos,incluyendo el popular formato pcap utilizado por Wireshark y otras herramientas de análisis de tráfico de red.

  Tiene muchas opciones y filtros para personalizar la captura de paquetes, lo que permite a los usuarios capturar únicamente el tráfico que necesitan y evitar la sobrecarga de información innecesaria. Tcpdump puede filtrar el tráfico por dirección IP, puerto, protocolo, y muchas otras características.

  Además, Tcpdump tiene una capacidad limitada de decodificación de protocolos de red, lo que permite a los usuarios identificar los protocolos utilizados por los paquetes capturados. Tcpdump es una herramienta muy útil para la solución de problemas de red y la investigación de incidentes de seguridad.

- **netstat:** es un comando utilizado para mostrar información sobre la red y las conexiones de red activas. Al ejecutar el comando netstat, se muestra una lista de conexiones activas, incluyendo la dirección IP, el puerto y el estado de cada conexión. Es útil para identificar conexiones de red no deseadas o sospechosas y para monitorear la actividad de la red en general.

- **nslookup:** es un comando utilizado para realizar consultas de resolución de nombres de dominio (DNS) y obtener información sobre registros de recursos de DNS, como las direcciones IP asociadas a un nombre de dominio. Al ejecutar el comando nslookup, se puede ingresar un nombre de dominio y obtener la dirección IP correspondiente, o viceversa.

- **dig:** es similar a nslookup, pero proporciona información más detallada y opciones de configuración adicionales. Al ejecutar el comando dig, se pueden realizar consultas de DNS y obtener información detallada sobre los registros de recursos, incluyendo la dirección IP, la TTL y la autoridad del servidor.

- **whois:** es un comando que permite buscar información sobre un dominio o una dirección IP en bases de datos públicas. Esta herramienta puede ser útil para obtener información sobre un sitio web o identificar el propietario de una dirección IP que se está utilizando para realizar actividades maliciosas.

- **traceroute:** es una herramienta que permite seguir la ruta que sigue un paquete de datos desde un origen hasta un destino. Esto puede ser útil para diagnosticar problemas de red y detectar posibles puntos de fallo o de ataque en la comunicación.

c) Monitoreo de equipos en una red

- **Nmap:** Es una herramienta de escaneo de red de código abierto que se utiliza para descubrir hosts y servicios en una red. Nmap puede utilizarse para encontrar vulnerabilidades y puertos abiertos en una red, y también se puede utilizar para identificar sistemas operativos y servicios en la red. Nmap es una herramienta esencial para cualquier profesional de la seguridad de la red.

- **Wireshark:** Es un analizador de protocolos de red de código abierto. Se utiliza para capturar y analizar el tráfico de red en tiempo real. Wireshark puede utilizarse para detectar problemas de seguridad y de red, y también puede utilizarse para examinar la comunicación entre aplicaciones. Wireshark es una herramienta muy versátil y es una de las más utilizadas en la industria.

- **OSQuery:** sistema de monitorización y gestión de sistemas que permite obtener información detallada sobre los sistemas en tiempo real, como el estado de los procesos, el uso de la memoria y la red, y otros aspectos de la configuración.

- **aide:** Es una herramienta de detección de intrusiones similar a tripwire que permite verificar la integridad de los archivos del sistema en busca de cambios no autorizados. AIDE puede generar una base de datos de los archivos del sistema y luego compararla con el estado actual del sistema para detectar cualquier cambio.

Aplicativos Web

- **Tripwire:** es una herramienta de integridad de archivos de código abierto. Se utiliza para monitorear los cambios en los archivos y directorios en el sistema. Tripwire puede utilizarse para detectar cambios malintencionados en los archivos y para alertar al usuario en caso de cambios no autorizados. Tripwire es una herramienta esencial para la detección de intrusiones.

- **Monitorix:** Es una herramienta de monitorización de sistemas y redes. Proporciona información en tiempo real sobre el uso de CPU, memoria, disco y red. También incluye gráficos y alertas para ayudar a identificar problemas.

- **Nagios:** Es una herramienta de monitoreo de red de código abierto que se utiliza para monitorear la disponibilidad y el rendimiento de los equipos y servicios en la red. Nagios puede utilizarse para enviar alertas cuando se detectan problemas y para llevar un registro del tiempo de actividad y el rendimiento de los equipos y servicios. Nagios es muy configurable y puede utilizarse para monitorear cualquier tipo de equipo o servicio.

- **Elastic Stack:** es una suite de herramientas que incluye Elasticsearch, Logstash y Kibana, que permiten recopilar, almacenar y visualizar registros de manera efectiva. Elasticsearch es un motor de búsqueda y análisis de datos en tiempo real, Logstash es una herramienta de procesamiento de registros y Kibana es una plataforma de visualización de datos.

- **Graylog:** Es una plataforma de gestión de registros que permite recopilar, indexar y analizar grandes volúmenes de registros de varias fuentes. Permite realizar búsquedas y análisis avanzados.

- **Zabbix:** Es una solución de monitoreo de código abierto que permite la supervisión en tiempo real de servidores, aplicaciones, redes y dispositivos. En el ámbito de la ciberseguridad, Zabbix es crucial para detectar anomalías en el rendimiento del sistema y la red, que podrían indicar intentos de intrusión, ataques DDoS, o actividades maliciosas. Zabbix también permite configurar alertas automáticas basadas en umbrales específicos, lo que ayuda a los administradores de seguridad a responder rápidamente ante posibles incidentes. Además, su capacidad de integración con otras herramientas de seguridad amplía su utilidad para la gestión proactiva de la infraestructura.

- **Cacti:** Es una herramienta de código abierto diseñada para la visualización y almacenamiento de datos de rendimiento de la red, utilizando gráficos generados por RRDTool. En ciberseguridad, Cacti es útil para la monitorización continua de la infraestructura de red, permitiendo a los administradores visualizar patrones de tráfico que podrían indicar actividades sospechosas, como el escaneo de puertos, intentos de exfiltración de datos, o tráfico no autorizado. Al analizar estas tendencias a lo largo del tiempo, Cacti ayuda a identificar comportamientos anómalos y facilita la implementación de medidas preventivas o correctivas para asegurar la red.

d) Análisis de bitácora

Un analizador de bitácoras del sistema es una herramienta que revisa los archivos de registro del sistema para identificar actividades maliciosas o sospechosas.

- **Rsyslog:** Es un sistema de registro de eventos de alta rendimiento y confiable para sistemas Linux. Permite la recolección, procesamiento y envío de registros de diferentes fuentes a diferentes destinos. Rsyslog puede ser configurado para enviar registros a un servidor centralizado o a una base de datos para su posterior análisis y monitoreo. Además, ofrece capacidades de filtrado y enriquecimiento de registros para facilitar la búsqueda y el análisis de datos.

- **Syslog-ng:** Es una herramienta de software libre que se utiliza para recolectar, procesar y almacenar registros de eventos (logs) generados por los sistemas y aplicaciones en un sistema informático. Syslog-ng se ejecuta en sistemas operativos tipo Unix y puede ser configurado para enviar los registros a diferentes destinos, como bases de datos, archivos de texto plano, servidores remotos de syslog, etc.

- **Logwatch:** es una herramienta que revisa diariamente los registros de actividad del sistema y los envía por correo electrónico al administrador. Puede personalizarse para mostrar los registros que se deseen.

- **Logrotate:** es una herramienta que administra los registros del sistema, eliminando los archivos de registro antiguos y archivando los nuevos. Puede ser configurado para comprimir y rotar registros de

  manera automática.

- **grep:** es una herramienta de búsqueda que permite encontrar líneas que contengan un patrón específico en un archivo o en una secuencia de archivos. Se utiliza con frecuencia para buscar errores en los archivos de bitácora. Por ejemplo, grep "error" /var/log/syslog buscará todas las líneas en el archivo /var/log/syslog que contengan la palabra "error".

- **head:** muestra las primeras líneas de un archivo. Es útil para obtener una vista previa rápida de los contenidos de un archivo de registro sin tener que abrir todo el archivo. Por ejemplo, head /var/log/syslog mostrará las primeras diez líneas del archivo /var/log/syslog.

- **tail:** muestra las últimas líneas de un archivo. Se utiliza para obtener las últimas entradas en un archivo de bitácora en tiempo real. Por ejemplo, tail -f /var/log/syslog mostrará las últimas líneas del archivo /var/log/syslog en tiempo real a medida que se agregan.

- **cut:** permite recortar una sección de un archivo de bitácora. Se utiliza con frecuencia para extraer información específica de un archivo de bitácora. Por ejemplo, cut -d " " -f 1,4 /var/log/syslog extraerá el primer y cuarto campo del archivo /var/log/syslog, utilizando un espacio como delimitador.

- **sort:** ordena las líneas de un archivo de bitácora en orden alfabético o numérico. Es útil para organizar los registros de bitácora en un formato más legible. Por ejemplo, sort /var/log/syslog ordenará el archivo /var/log/syslog en orden alfabético.

- **Fluentd:** Es un recolector y procesador de registros que permite la recopilación de registros de varias fuentes, la transformación y enrutamiento de los mismos y su almacenamiento en diferentes destinos.

Para actividades Reactivas

a) Gestión de intrusiones

- **Fail2ban:** Es una herramienta de prevención de intrusiones basada en host (HIPS) que monitorea los registros de autenticación y otros archivos de registro del sistema para detectar intentos fallidos repetidos de acceso o actividades sospechosas. Si Fail2ban detecta un patrón de actividad maliciosa

(como múltiples intentos fallidos de inicio de sesión), bloquea la dirección IP ofensiva mediante la modificación de las reglas del firewall, evitando así ataques de fuerza bruta y otras amenazas. Es especialmente útil en servidores y sistemas expuestos a internet, brindando una capa adicional de seguridad de forma automatizada.

- **Snort:** Es un sistema de detección y prevención de intrusiones en red (NIDS/NIPS) de código abierto que analiza el tráfico de red en tiempo real para identificar patrones y firmas de actividades maliciosas, como intentos de explotación de vulnerabilidades, escaneos de puertos, ataques DDoS, y más. Snort utiliza una combinación de análisis basado en firmas, análisis de protocolos y detección de anomalías para identificar y alertar sobre posibles amenazas, permitiendo a los administradores de red tomar acciones para mitigar los riesgos.

- **Snortsam:** Es una extensión de Snort que permite la integración de Snort con diferentes sistemas de firewall para bloquear automáticamente el tráfico malicioso identificado por Snort. Snortsam funciona interceptando las alertas de Snort y traduciéndose en reglas de firewall que bloquea dinámicamente las direcciones IP involucradas en actividades sospechosas. Esta capacidad de respuesta activa ayuda a contener y mitigar amenazas de manera rápida y efectiva, protegiendo la red en tiempo real.

- **OSSEC:** Es una plataforma de seguridad de código abierto que funciona como un sistema de detección de intrusiones basado en host (HIDS). OSSEC realiza monitoreo de archivos, análisis de registros, y detección de rootkits, además de la correlación de eventos y la respuesta automática ante incidentes. En ciberseguridad, OSSEC es fundamental para la protección proactiva de los sistemas al detectar y alertar sobre cambios no autorizados en archivos críticos, intentos de escalada de privilegios, y otras actividades sospechosas que podrían indicar una intrusión o ataque.

- **Suricata:** Es un motor de inspección de red de código abierto que actúa como un sistema de detección y prevención de intrusiones (IDS/IPS), con la capacidad de realizar monitoreo de red, captura de paquetes, y análisis en profundidad de protocolos. Suricata se destaca por su capacidad de manejar grandes volúmenes de tráfico y por su soporte para la inspección multicapa, permitiendo identificar amenazas tanto en la capa de red como en la capa de aplicación. Además, Suricata es compatible con las reglas de Snort, lo que facilita la transición para los usuarios de Snort y permite la implementación de un enfoque de defensa en profundidad en la infraestructura de red.

- **Wazuh:** Es una plataforma de seguridad de código abierto que proporciona monitoreo en tiempo real, detección de amenazas, y respuesta ante incidentes. Wazuh combina las funcionalidades de un SIEM (Security Information and Event Management) con un HIDS (Host-based Intrusion Detection System), ofreciendo una solución integral para la gestión de la seguridad en sistemas de TI. Es utilizado para la supervisión de la integridad de archivos, análisis de registros, y detección de comportamientos anómalos en los sistemas.

- **Security Onion:** Es una plataforma de código abierto para la monitorización, detección y respuesta ante amenazas (MDR) que integra una variedad de herramientas de ciberseguridad en un entorno centralizado. Security Onion está diseñado para facilitar la implementación de capacidades avanzadas de defensa en redes, como la detección de intrusiones, análisis de tráfico de red y respuesta ante incidentes.

- **Detección de intrusiones:** Integra herramientas como Suricata y Zeek (anteriormente conocido como Bro) para la detección de intrusiones en la red. Estas herramientas analizan el tráfico de red en tiempo real para identificar patrones de comportamiento malicioso o anómalo.

- **Análisis de Logs:** Utiliza Elastic Stack (Elasticsearch, Logstash, Kibana) para la recolección, indexación y visualización de logs y datos de eventos. Esto permite a los analistas correlacionar eventos de diferentes fuentes y obtener una visión completa de la actividad en la red.

- **Análisis Forense:** Incluye herramientas para la captura y análisis de paquetes de red, permitiendo a los equipos de respuesta a incidentes realizar un análisis forense detallado después de un incidente de seguridad. Moloch (Arkime) es una de las herramientas que pueden integrarse para realizar este tipo de análisis.

- **Monitoreo de Host:** A través de la integración con OSSEC, Security Onion también ofrece capacidades de detección de intrusiones en host (HIDS), lo que permite monitorear la integridad de los sistemas y detectar comportamientos sospechosos a nivel de host.

- **Respuesta a Incidentes:** Security Onion no solo se enfoca en la detección, sino que también incluye herramientas para la respuesta a incidentes, ayudando a los equipos de seguridad a tomar decisiones informadas y ejecutar acciones para contener y remediar las amenazas.

b) Antimalware

- **Arkime (antes llamado Moloch):** Es una plataforma de captura y análisis de paquetes de red de código abierto diseñada para la supervisión y análisis forense del tráfico de red a gran escala. En el contexto de ciberseguridad, Arkime es invaluable para realizar un análisis detallado de las comunicaciones de red, permitiendo a los analistas investigar incidentes de seguridad, como brechas de datos, intrusiones, o actividades sospechosas.

  Arkime almacena los paquetes capturados en un formato optimizado y los indexa, facilitando búsquedas rápidas y el análisis de eventos específicos en la red. Además, su capacidad para integrarse con otras herramientas de seguridad y su soporte para consultas avanzadas lo convierten en una herramienta potente para los equipos de respuesta a incidentes y análisis forense.

- **Cuckoo:** Es un sistema de análisis de malware de código abierto. Utiliza máquinas virtuales para analizar archivos y detectar posibles amenazas. Kuckoo es capaz de ejecutar malware en un entorno controlado y analizar su comportamiento para detectar posibles amenazas.

- **Malware Analysis Lab:** es un entorno controlado y seguro diseñado específicamente para analizar y estudiar el comportamiento de malware. En este laboratorio, los analistas de seguridad pueden ejecutar, observar y descomponer muestras de software malicioso para comprender sus mecanismos de funcionamiento, vectores de ataque y objetivos. El laboratorio está completamente aislado del resto de la red para evitar cualquier propagación accidental del malware. Incluye una variedad de herramientas para análisis estático (inspección del código sin ejecutarlo) y dinámico (ejecución del malware para observar su comportamiento). Ejemplos de estas herramientas son IDA Pro, Ghidra (análisis estático), Cuckoo Sandbox, y Remnux (análisis dinámico).

  El laboratorio puede registrar todas las acciones del malware, como las conexiones de red que intenta establecer, los archivos que crea o modifica, y las claves de registro que altera. Proporciona herramientas para desempaquetar y desofuscar el código, ya que los autores de malware a menudo utilizan técnicas para dificultar su análisis. Permite el uso de técnicas de ingeniería inversa para descompilar o desensamblar el malware y estudiar su código fuente.

c) Informática forense

- **SleuthKit Autopsy:** Es una herramienta de análisis forense que permite examinar sistemas de archivos y recuperar datos de discos duros, particiones y sistemas de archivos.

- **Santoku Linux:** Es una distribución de Linux especializada en análisis forense y pruebas de penetración. Incluye herramientas como Nmap, Wireshark, Metasploit,entre otras.

- **Android Debug Bridge (ADB):** Es una herramienta de línea de comandos que permite a los desarrolladores de Android comunicarse con dispositivos Android y emuladores desde un sistema operativo host. Se utiliza para depurar aplicaciones de Android, instalar aplicaciones y controlar dispositivos de forma remota.

Para actividades Retrospectivas

No hemos encontrado herramientas de software libre específicamente diseñadas para actividades retrospectivas. Sin embargo, estas funcionalidades se encuentran distribuidas entre las herramientas mencionadas en los capítulos anteriores. Las herramientas descritas abarcan una gama de funciones preparativas, preventivas, detectivas y reactivas que, en conjunto, pueden ser utilizadas para realizar evaluaciones y análisis retrospectivos efectivos.

3. CONCLUSIONES

La ciberseguridad es un tema crucial en la era digital en la que vivimos, y el software libre ha demostrado ser una herramienta valiosa en este campo. La comunidad de software libre ha creado y mantenido herramientas para preparación, prevención, detección y reacción que son esenciales para la protección de sistemas y redes.

El software libre ha demostrado ser una alternativa viable y poderosa para la ciberseguridad. Sin embargo, es importante recordar que el hecho de que un software esté protegido por una licencia de código abierto no es garantía de calidad o seguridad. Es fundamental revisar la actividad de la comunidad, como las conversaciones y la generación de nuevas versiones con corrección de errores y nuevas funcionalidades, para asegurarnos de la calidad y seguridad del software que utilizamos.

SOBRE EL AUTOR

Ingeniero, empresario y docente enfocado en el sector de la tecnología de información y comunicaciones, con más de 30 años de experiencia en desarrollo de software seguro, seguridad de la información, el uso de estándares de comunicación y la óptima integración del software libre en las organizaciones.

Actualmente se desempeña como gerente de Skina IT Solutions y complementa su vida profesional como profesor de cátedra en la Universidad Javeriana, representando a Colombia en el comité de ciberseguridad de la UPADI.

- Magíster en Ingeniería de Sistemas y Computación Universidad de los Andes, Bogotá, 1998.
- Ingeniero Civil Universidad de los Andes, Bogotá, 1995.
- Diplomado Docencia en Ingeniería Pontificia Universidad Javeriana, Bogotá, 2008

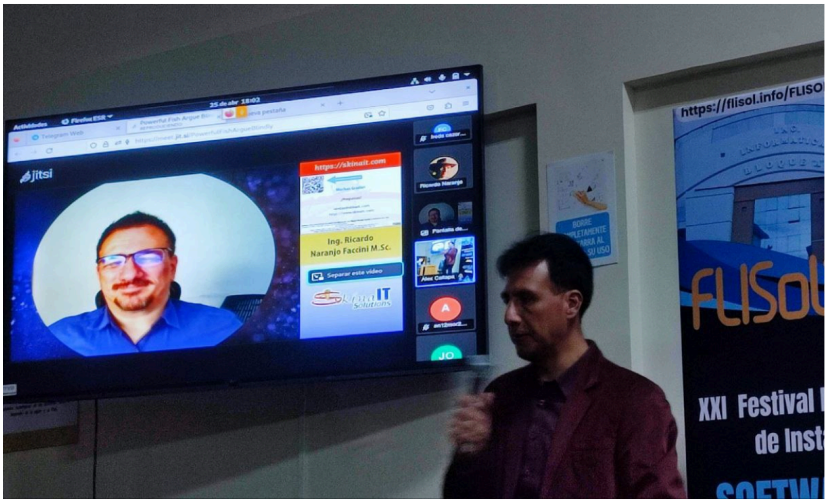

Figura 1: Fotografía de presentación de la ponencia

APLICACIONES DE SOFTWARE LIBRE EN LA INGENIERÍA DEL AGUA Y CIENCIAS DE LA GEO-INFORMACIÓN

M.Sc. ING. JORGE AYALA NIÑO DE GUZMÁN

comunidad@qgisbolivia.org

Ingeniería Civil

Universidad Mayor de San Simón

Bolivia

RESUMEN

"Promover el uso de tecnologías libres más allá del área de sistemas, destacando su aplicabilidad en campos técnicos como la Ingeniería del Agua y las Ciencias de la Geo-información, un llamado a la acción para que, en el marco del FLISoL dentro de las universidades, se invite autoridades, docentes y estudiantes de otras facultades y carreras a sumarse al uso de tecnologías libres y a compartir sus experiencias.

1. INTRODUCCIÓN

La disponibilidad de Información Geoespacial y el manejo eficiente de los recursos hídricos son pilares fundamentales para el desarrollo sostenible. La Ingeniería del Agua aborda la planificación, diseño y monitoreo de sistemas de agua potable, alcantarillado y gestión de riesgos hídricos. Paralelamente, la Ciencia de la Geo-Información facilita la captura, análisis y representación espacial de estos recursos a través de tecnologías avanzadas como los Sistemas de Información Geográfica (SIG), percepción remota (teledetección), y bases de datos espaciales.

Hasta principios del siglo XXI, las herramientas de software para estas disciplinas eran principalmente privativas, con altos costos de licenciamiento que limitaban su accesibilidad. Sin embargo, el auge del Software Libre ha democratizado estas tecnologías, permitiendo su adopción masiva incluso en contextos con limitaciones presupuestarias.

El Festival Latinoamericano de Instalación de Software Libre (FLISoL) ha jugado un rol fundamental en la promoción de estas alternativas. A través de este espacio, se han generado redes de colaboración y transferencia de conocimiento que han impactado positivamente en la práctica profesional y académica.

Este artículo analiza desde una perspectiva práctica y retrospectiva la evolución del Software Libre aplicado a la Ingeniería del Agua y la Geo-Información, a partir de experiencias directas de participación y organización en el FLISoL desde 2005.

2. DESARROLLO

Evolución de herramientas de Software Libre para SIG e Ingeniería del Agua:

- **Antes de 2005**: la falta de opciones libres sólidas, proyectos libres en SIG en etapas aún de desarrollo obligaba al uso de software privativo o a recurrir a versiones ilegales en entornos académicos y profesionales.

- **2013:** lanzamiento de QGIS 2.0, que introduce una interfaz gráfica mejorada, extensibilidad mediante plugins y compatibilidad avanzada con bases de datos espaciales, marcando un punto de inflexión en el uso de SIG libres.

Herramientas complementarias:

- **PostGIS** para bases de datos espaciales en PostgreSQL
- **GeoServer** para servicios de mapas en línea
- **GRASS GIS y SAGA GIS** para análisis avanzados de terreno y modelado hidrológico.

Adopción de hardware de bajo costo y plataformas de instalación

- **Raspberry Pi 4 con QGIS:** demuestra la viabilidad de operar SIG en dispositivos económicos.

- **Uso de subsistemas Linux en Windows**, Live USB persistentes y adquisición de planes de VPS accesibles, facilita la instalación de servidores y ambientes de aprendizaje y desarrollo geoespacial sin necesidad de infraestructura costosa.

Innovación en ingeniería del agua mediante Software Libre:

- **EPANET**: Toolkit y su integración con Python permiten la modelación hidráulica avanzada y la creación de plugins para un trabajo en entorno de QGIS.

- **QCAD:** alternativa libre para el dibujo asistido por computadora para representación en 2D de planos de diseño de infraestructuras hidráulicas.

- **OpenDroneMap**: como alternativa libre para el procesamiento aerofotogramétrico de imágenes aéreas capturadas por drones, favoreciendo la generación de modelos digitales de superficie aplicables en modelaciones hidráulicas e hidrológicas, así como en la producción de ortomosaicos destinados al monitoreo de recursos hídricos mediante imágenes de alta resolución, sin requerir unidades de procesamiento gráfico (GPU), a diferencia del software privativo.

- **CloudCompare:** herramienta para el análisis de nubes de puntos 3D, obtenidas mediante tecnologías LiDAR o fotogrametría, aplicada en diversos campos de la Ingeniería del Agua.

Creación de proyectos Nacionales y comunidades:

- **La Infraestructura de Datos Espaciales del Estado Plurinacional Bolivia (GeoBolivia)**
  Es un ejemplo de implementación de servicios geoespaciales basados en Software Libre a nivel gubernamental desde el año 2012.

- **Crecimiento de geoportales web de ministerios e instituciones gubernamentales**
  Como el MMAyA, IGM, INRA e INE, con el objetivo de optimizar el acceso y análisis de datos geoespaciales clave para la toma de decisiones y la gestión pública.

- **Consolidación de la Comunidad de Usuarios de QGIS Bolivia**
  Para promover el aprendizaje, colaboración y resolución de problemáticas locales mediante SIG libres.

Discusión

El progreso tecnológico evidenciado en los últimos 20 años ha permitido que herramientas de Software Libre se posicionen como alternativas competitivas frente a sus contrapartes privativas. Este fenómeno ha sido especialmente significativo en disciplinas como la Ingeniería del Agua y la Geo-Información, donde el acceso a software especializado era tradicionalmente restrictivo.

Sin embargo, se observa una resistencia persistente en algunas universidades bolivianas, que continúan formando profesionales en pregrado bajo esquemas obsoletos basados en software privativo y, en muchos

casos, en versiones piratas. Esta brecha formativa no solo implica riesgos legales y éticos, sino que limita las oportunidades de innovación y competitividad de los futuros profesionales.

La adopción del Software Libre no debería entenderse solamente como una alternativa económica, sino como una estrategia de independencia tecnológica, soberanía de datos, y estímulo a la investigación y desarrollo. La actualización curricular es urgente y debe acompañarse de iniciativas de sensibilización, capacitación y acompañamiento institucional.

Además, las herramientas de Software Libre promueven un ecosistema colaborativo que permite la adaptación y personalización de soluciones a problemas locales, como la gestión de cuencas hidrográficas, el diseño de sistemas de agua potable, y el monitoreo de variables ambientales en escenarios de cambio climático.

Finalmente, el fortalecimiento de comunidades de usuarios, como la Comunidad de Usuarios de QGisBolivia, constituye un pilar estratégico para sostener y expandir el uso de estas tecnologías, facilitando el intercambio de conocimientos y el desarrollo de capacidades técnicas en el ámbito nacional y regional.

3. CONCLUSIONES

El Software Libre ha evolucionado de manera significativa, proporcionando herramientas robustas para la Ingeniería del Agua y la Ciencia de la Geo-Información. Su adopción ha permitido democratizar el acceso a tecnologías de punta, impulsar proyectos de gran escala y fortalecer la formación profesional.

Es necesario que las universidades bolivianas adapten sus currículas a estas realidades tecnológicas, abandonando paradigmas obsoletos y formando profesionales preparados para liderar los desafíos de la transformación digital en la gestión del agua y el territorio. Eventos como el FLISoL deben continuar su labor de difusión de nuevas tecnologías y capacitación, expandiendo su alcance hacia nuevas generaciones de estudiantes e investigadores.

SOBRE EL AUTOR

Ingeniero Civil con Maestría en Ciencias de la Geo-información y Observación de la Tierra Mención Evaluación de Recursos Hídricos (UMSS), con más de 17 años de experiencia en el sector de la Ingeniería del Agua, a lo largo de su trayectoria, ha sido consultor para diversas instituciones, entre las más importantes BID, MMAyA, UCEP-MMAyA, SENASBA, EMAGUA, FPS, SeLA, etc.

Así mismo es docente de posgrado de la Universidad Autónoma Gabriel René Moreno (UAGRM), de la “Maestría en Catastro Territorial y Geodesia”

Organizó, con el apoyo del colectivo PIMI (Pingüinos del Mismo Iceberg), la primera edición del FLISOL 2005 en la ciudad de Cochabamba.

Actualmente es CEO de UmaYakuY Consultores SRL, una startup dedicada a impulsar la “Transformación Digital de la Gestión del Agua y Saneamiento con SIG y Tecnologías Libres." , así como a promover la actualización profesional a través de cursos especializados en estas tecnologías libres, habiendo alcanzado hasta la fecha más de 500 participantes.

Figura 1: Fotografía de presentación de la ponencia.

INTELIGENCIA ARTIFICIAL: PROCESAMIENTO DEL LENGUAJE NATURAL CON SOFTWARE LIBRE

JUAN PABLO LUNA FELIPEZ, Ph.D.

jplunaf@gmail.com

Ingeniería Informática

Universidad Nacional "Siglo XX"

Llallagua, Bolivia

RESUMEN

El Procesamiento del Lenguaje Natural (PLN) es una rama de la inteligencia artificial que permite a las máquinas entender, interpretar y generar lenguaje humano. Entre sus aplicaciones destacan: análisis de sentimientos, clasificación de textos, traducción automática, resumen de textos, reconocimiento de voz y texto, y detección de entidades como nombres o lugares. El PLN se apoya en herramientas y bibliotecas de software libre, especialmente en lenguajes como Python, R, Julia, Java y C++, y librerías como spaCy, NLTK, Transformers, Gensim y TextBlob. El proceso incluye técnicas como tokenización, eliminación de palabras vacías (stopwords), stemming, lematización, POS tagging y word embeddings.

1. INTRODUCCIÓN

El lenguaje es una de las herramientas más poderosas del ser humano para comunicar pensamientos, emociones e información. En la era digital, la necesidad de que las máquinas comprendan y generen lenguaje humano dieron origen a una disciplina clave dentro de la inteligencia artificial: el Procesamiento del Lenguaje Natural (PLN). Esta área busca desarrollar sistemas capaces de interpretar, analizar y producir lenguaje de manera automática, facilitando la interacción entre humanos y computadoras.

Gracias al uso de software libre, hoy en día es posible acceder a poderosas herramientas y bibliotecas que permiten implementar soluciones de PLN de forma accesible y eficiente. Tecnologías como el análisis de sentimientos, la clasificación automática de textos o los asistentes virtuales son solo algunos ejemplos del impacto real que tiene esta disciplina en campos como el marketing, la salud, la educación y el servicio al cliente. Este trabajo explora los fundamentos del PLN, sus principales tareas y aplicaciones, así como las herramientas de código abierto más utilizadas para su desarrollo.

2. DESARROLLO

Procesamiento del lenguaje natural (PLN)

Disciplina de la IA, busca que las computadoras comprendan y procesen el lenguaje humano de forma natural y eficiente. Puede o no, usar aprendizaje automático

Tareas

- **Comprensión del lenguaje natural (NLU):** implica comprender e interpretar el lenguaje humano,
- como el reconocimiento de voz, la clasificación de texto, el análisis de sentimientos y la extracción de información.
- **Generación de lenguaje natural (NLG):** implica generar texto legible por humanos a partir de datos estructurados, como resúmenes de texto, sistemas de diálogo y traducción de idiomas.

Aplicaciones de NLP:

- **Análisis de Sentimientos:** Detectar emociones o intenciones en textos. Por ejemplo, opiniones en redes sociales, encuestas de satisfacción.
- **Clasificación de Texto:** Categorizar documentos automáticamente. Por ejemplo, filtros de spam, categorización de correos o noticias.
- **Reconocimiento de Entidades (NER):** Identificar nombres de personas, lugares, fechas, organizaciones. Por ejemplo, chatbots, sistemas de preguntas y respuestas.
- **Traducción automática:** Convertir texto de un idioma a otro. Por ejemplo, Google Translate, DeepL.

- **Resumen Automático:** Reducir textos largos conservando ideas principales. Por ejemplo, noticias, documentos legales, artículos científicos.
- **Generación de Texto:** Crear texto coherente de manera automática. Por ejemplo, chatGPT, generación de noticias automáticas.
- **Análisis Morfosintáctico (POS tagging):** Identificar la función gramatical de cada palabra. Por ejemplo, preprocesamiento para traductores y chatbots.
- **Extracción de Información:** Detectar datos clave en grandes volúmenes de texto. Por ejemplo, minería de datos, informes automáticos.
- **Reconocimiento de Voz a Texto (ASR):** Convertir voz en texto escrito. Por ejemplo, asistentes virtuales como Siri, Alexa, Google Assistant.
- **Conversión de Texto a Voz (TTS):** Convertir texto escrito en audio. Por ejemplo, lectores de pantalla, asistentes virtuales.
- **Detección de Idioma:** Identificar en qué idioma está escrito un texto. Por ejemplo, plataformas multilingües, filtros automáticos.
- **Corrección Gramatical:** Detectar y corregir errores en el texto. Por ejemplo, Grammarly, correctores de texto automáticos.

Lenguajes de software libre:

- **Python:** Python Software Foundation License (Libre). Razones para usarse en Enorme ecosistema: NLTK, spaCy, Hugging Face Transformers, **Gensim.** Muy usado en IA y Deep Learning.
- **R:** GPL (Libre). Enfoque estadístico, ideal para análisis de texto académico. Paquetes como tm, quanteda, text.
- **Julia:** MIT (Libre). Alta velocidad, sintaxis moderna, librería TextAnalysis.jl para NLP. Excelente para modelos grandes.
- **Java:** GPL (Libre). Librerías como Apache OpenNLP y Stanford CoreNLP. Muy usado en entornos empresariales y producción.
- **C++ :** GPL / BSD / MIT (según proyecto). Máximo rendimiento. Usado en motores de NLP como spaCy (núcleo) y software que exige velocidad extrema.

Bibliotecas libres Python:

- **NLTK:** Apache 2.0 (Libre). Herramientas clásicas para análisis léxico, tokenización, stemming, POS tagging.
- **spaCy:** MIT (Libre) Procesamiento eficiente de texto, modelos preentrenados, entidades, dependencias.
- **Transformers (Hugging Face):** Apache 2.0 (Libre). Modelos preentrenados: BERT, GPT, RoBERTa, T5 para tareas modernas de NLP.
- **Gensim:** LGPL (Libre). Modelado de tópicos y vectores: Word2Vec, Doc2Vec, FastText.
- **TextBlob:** MIT (Libre). Simplificación de tareas NLP: traducción, análisis de sentimientos, POS tagging.
- **Flair:** MIT (Libre). NLP de última generación con embeddings contextuales, desarrollado por Zalando.
- **Stanza:** Apache 2.0 (Libre). Toolkit de Stanford para NLP: análisis sintáctico, NER, modelos multilingües.
- **Polyglot:** GPLv3 (Libre). Procesamiento multilingüe: detección de idioma, NER, embeddings.

Tokenización: Dividir un texto en partes pequeñas, como palabras o frases.

```python
from nltk.tokenize import word_tokenize
texto = "Hola mundo"
tokens = word_tokenize(texto)
print(tokens)  # ['Hola', 'mundo']
```

Figura 1: Ejemplo práctico, separar la frase "Hola mundo" en las palabras "Hola" y "mundo".

Eliminación de stopwords: Quitar palabras comunes que no aportan mucho significado, como "el", "y", "de".

```python
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import nltk
nltk.download('stopwords')

texto = "Este es un ejemplo simple"
tokens = word_tokenize(texto.lower())
stop_words = set(stopwords.words('spanish'))
filtrado = [t for t in tokens if t not in stop_words]
print(filtrado)  # ['ejemplo', 'simple']
```

Figura 2: Ejemplo práctico, en la frase "Este es un ejemplo simple", eliminar palabras como "es", "un".

Stemming: Reducir palabras a su raíz para agrupar variantes.

```python
from nltk.stem import PorterStemmer
stemmer = PorterStemmer()
print(stemmer.stem("running"))  # run
```

Figura 3: Ejemplo práctico "corriendo", "corrió" y "correr" se reducen a "corr".

Lematización: Convertir palabras a su forma base real según el contexto.

```python
from nltk.stem import WordNetLemmatizer
import nltk
nltk.download('wordnet')

lemmatizer = WordNetLemmatizer()
print(lemmatizer.lemmatize("running", pos='v'))  # run
```

Figura 4: Ejemplo práctico, "corriendo" se convierte en "correr".

Etiquetado gramatical (POS Tagging): Identificar la función gramatical de cada palabra (sustantivo, verbo, adjetivo, etc.).

```
from nltk import pos_tag, word_tokenize
tokens = word_tokenize("El gato corre")
tags = pos_tag(tokens)
print(tags)  # [('El', 'DT'), ('gato', 'NN'), ('corre', 'VBZ')]
```

Figura 5: Ejemplo práctico: En "El gato corre", "gato" es sustantivo y "corre" es verbo.

Reconocimiento de entidades nombradas (NER): Detectar nombres propios como personas, lugares u organizaciones en un texto.

```
from nltk import ne_chunk, pos_tag, word_tokenize
import nltk
nltk.download('maxent_ne_chunker')
nltk.download('words')

texto = "Barack Obama nació en Hawaii"
tokens = word_tokenize(texto)
tags = pos_tag(tokens)
tree = ne_chunk(tags)
print(tree)
```

Figura 6: Ejemplo práctico, en "Barack Obama nació en Hawaii", reconocer "Barack Obama" como persona y "Hawaii" como lugar.

N-gramas: Secuencias de “n” palabras juntas para analizar contexto.

```
from nltk.util import ngrams
tokens = ["software", "libre"]
bigrams = list(ngrams(tokens, 2))
print(bigrams)  # [('software', 'libre')]
```

Figura 7: Ejemplo práctico, en "software libre", el bigrama es ("software", "libre").

Conteo de frecuencia: Contar cuántas veces aparece cada palabra en un texto.

```
from collections import Counter
tokens = ["libre", "libre", "software"]
freq = Counter(tokens)
print(freq)  # {'libre': 2, 'software': 1}
```

Figura 8: Ejemplo práctico: En "libre libre software", "libre" aparece 2 veces.

Clasificación de texto: Asignar una categoría a un texto, como spam o no spam.

```
texto = "Gana dinero rápido"
print("spam" if "dinero" in texto else "ham")  # spam
```

Figura 9: Ejemplo práctico, detectar si un correo es spam según palabras clave.

Word Embeddings: Representar palabras con vectores numéricos que capturan su significado.

```
from gensim.models import Word2Vec

sentences = [["software", "libre"], ["libertad", "comunidad"]]
model = Word2Vec(sentences, vector_size=10, min_count=1)
print(model.wv['software'])  # vector numérico
```

Figura 10: Ejemplo práctico: Palabras similares tienen vectores cercanos, como "rey" y "reina".

Flujo típico de NLP:

Preprocesamiento de texto: Preparar el texto para su análisis: limpieza, normalización y tokenización.

```
# Limpieza y normalización
texto = "¡Hola Mundo! ¿Cómo estás?"
texto_limpio = texto.lower().replace("¡", "").replace("!", "").replace("¿",
"").replace("?", "")
print(texto_limpio)
# Salida: hola mundo cómo estás

# Tokenización
from nltk.tokenize import word_tokenize
import nltk
nltk.download('punkt')

tokens = word_tokenize(texto_limpio)
print(tokens)
# Salida: ['hola', 'mundo', 'cómo', 'estás']
```

Figura 11: Preparación del texto para analizarlo

Eliminación de stopwords: Quitar palabras sin significado relevante.

```
from nltk.corpus import stopwords
nltk.download('stopwords')

stop_words = set(stopwords.words('spanish'))
tokens_filtrados = [t for t in tokens if t not in stop_words]
print(tokens_filtrados)
# Ejemplo salida: ['hola', 'mundo']
```

Figura 12: Eliminación de stopwords

Stemming y Lematización: Reducir palabras a su raíz o forma base para agrupar variantes.

```
from nltk.stem import PorterStemmer, WordNetLemmatizer
import nltk
nltk.download('wordnet')

stemmer = PorterStemmer()
lemmatizer = WordNetLemmatizer()

print(stemmer.stem("running"))        # run
print(lemmatizer.lemmatize("running", pos='v'))  # run
```

Figura 13: Stemming y lematización

- **Etiquetado gramatical (POS Tagging):** Asignar categorías gramaticales a cada palabra.

```
from nltk import pos_tag

nltk.download('averaged_perceptron_tagger')

tags = pos_tag(tokens_filtrados)
print(tags)
# Ejemplo salida: [('hola', 'NN'), ('mundo', 'NN')]
```

Figura 14: Etiquetado gramatical

Reconocimiento de entidades nombradas (NER): Detectar nombres propios como personas, lugares u organizaciones.

```
from nltk import ne_chunk

nltk.download('maxent_ne_chunker')
nltk.download('words')

arbol = ne_chunk(tags)
print(arbol)
```

Figura 15: Reconocimiento de entidades nombradas

Extracción de características: Convertir texto en datos numéricos para modelos (Bag of Words, TF-IDF, embeddings).

```
from collections import Counter

frecuencia = Counter(tokens_filtrados)
print(frecuencia)
# Ejemplo salida: Counter({'hola': 1, 'mundo': 1})
```

Figura 16: Extracción de características

Modelado y aprendizaje automático: Entrenar modelos para tareas como clasificación o análisis de sentimientos.

```
# Clasificación simple basada en palabras clave
texto = "Gana dinero rápido"
print("spam" if "dinero" in texto.lower() else "ham")
# Salida: spam
```

Figura 17: Clasificación simple basada en palabra clave

Análisis de Sentimientos: Determinar si un texto expresa opinión positiva, negativa o neutral.

```python
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import nltk
nltk.download('vader_lexicon')

sid = SentimentIntensityAnalyzer()
scores = sid.polarity_scores("I love this product!")
print(scores)
# Ejemplo salida: {'neg': 0.0, 'neu': 0.294, 'pos': 0.706, 'compound': 0.6696}
```

Figura 18: Análisis de sentimientos

Modelado de temas: Identificar temas principales en textos largos o colecciones.

```python
from gensim import corpora, models

documentos = [
    "El fútbol es un deporte popular",
    "La política afecta la economía",
    "El baloncesto es divertido"
]

texts = [doc.lower().split() for doc in documentos]
diccionario = corpora.Dictionary(texts)
corpus = [diccionario.doc2bow(text) for text in texts]

lda = models.LdaModel(corpus, num_topics=2, id2word=diccionario, passes=10)
for idx, topic in lda.print_topics(-1):
    print(f"Tema {idx}: {topic}")
```

Figura 19: Modelado de temas

Comprensión del lenguaje natural (NLU) básica: Interpretar la intención o significado detrás del texto.

```
nltk.download('wordnet')

texto = "¿Puedes reservarme un taxi para mañana?"

# Tokenizar el texto
tokens = word_tokenize(texto.lower())

# Lematizar cada palabra
lemmatizer = WordNetLemmatizer()
lemmas = [lemmatizer.lemmatize(token) for token in tokens]

# Definir palabras clave en su forma base
intencion = ("reservar" in lemmas or "reservar" in tokens) and "taxi" in lemmas

if intencion:
    print("Intento detectado: Reservar un taxi")
else:
    print("Intento desconocido")
```

Figura 20: Interpretación del significado del texto

3. CONCLUSIÓN

El Procesamiento del Lenguaje Natural permite a las máquinas comprender y generar lenguaje humano, facilitando numerosas aplicaciones como asistentes virtuales, análisis de sentimientos y traducción automática. Gracias al software libre, estas tecnologías están al alcance de todos, impulsando la innovación y el aprendizaje. En un entorno cada vez más digital, el PLN se vuelve esencial para aprovechar al máximo la información escrita y mejorar la interacción entre personas y sistemas inteligentes.

SOBRE EL AUTOR

Doctor en Ciencias de la Computación, Máster en Ciencias de la Computación con mención en Seguridad Informática y Software Libre, Máster en Educación Superior, cuenta con Diplomados en Preparación Evaluación y Gestión de Proyectos, Formación Basada por Competencias y Metodología de la Investigación Científica.

Ha sido Director de la Carrera Ingeniería Informática y actualmente es Presidente de la Sociedad Científica de Docentes (SOCID) de la Universidad Nacional “Siglo XX”, Coordinador del Instituto de Investigación y Desarrollo de Aplicaciones Informáticas IIDAI, Director de la Revista Científica “Ciencia y Tecnología Informática”, Director de las publicaciones Memorias SOCID, Memorias Ciencia y Tecnología Informática y Docente Universitario Titular en la carrera Ingeniería Informática de la Universidad Nacional “Siglo XX”, Docente de pre y postgrado, organiza varios eventos académicos.

Expositor y organizador en varios eventos académicos.

Publicó varias Revistas Científicas, Artículos y Libros, recibió varios reconocimientos y premios.

Figura 21: Fotografía de presentación de la ponencia.

ANÁLISIS DE SENTIMIENTOS CON PYTHON

LEYNA ROXANA SALINAS VEYZAGA, Ph.D.

leynasud@gmail.com

Ingeniería Informática

Universidad Nacional "Siglo XX"

Llallagua, Bolivia

RESUMEN

El análisis de sentimientos permite identificar si un texto expresa emociones positivas, negativas o neutras, aunque es una tarea sencilla para los humanos, las computadoras requieren modelos estadísticos para lograrlo. Este proceso se basa en analizar palabras que por naturaleza, evocan ciertas emociones. Por ejemplo: "hermoso" sugiere positividad, mientras que "terrible" refleja negatividad. Python ofrece herramientas para automatizar este análisis usando librerías que asignan un puntaje de polaridad.

Dicho puntaje va de 0 (negativo) a 1 (positivo), siendo 0.5 el punto neutral. El análisis de sentimientos se aplica en redes sociales, encuestas, marketing y ventas. Tiene como objetivo extraer valor de las opiniones y emociones expresadas en textos. Esta técnica es clave para entender mejor a los usuarios y tomar decisiones basadas en datos.

1. INTRODUCCIÓN

El análisis de sentimientos es una técnica de procesamiento de lenguaje natural (PLN) que permite evaluar y clasificar automáticamente las emociones expresadas en un texto. Este tipo de análisis ha cobrado gran relevancia en áreas como el marketing digital, la atención al cliente y el monitoreo de redes sociales, ya que ofrece una visión clara de las percepciones y actitudes de los usuarios frente a productos, servicios o temas específicos. Herramientas como la librería *SentimentClassifier* facilitan este proceso al asignar un puntaje de polaridad que varía entre 0 (sentimiento negativo) y 1 (sentimiento positivo), con 0.5 como punto neutral.

2. DESARROLLO

¿Qué es el análisis de sentimientos?

Está definida como: “Extraer información de valor luego de haber evaluado las emociones, actitudes y opiniones de los usuarios detrás de una serie de palabras”.

Principales usos:

- Análisis de datos en redes sociales
- Marketing personalizado
- Encuestas
- Posicionamiento de marca
- Previsión de ventas

Ejemplo 1:

X= Esta muy buena la comida
Y= Que fea película
Z= hoy tuve una experiencia neutral

```
from classifier
importSentimentClassifier
clf= SentimentClassifier()
x= "Esta comida esta rica"
y= "que horrible esa pelicula"
z= "Tuve una experiencia neutral"
sentimiento=clf.predict(x)
print(sentimiento)
_____
0,93155720156
```

Figura 1: Resultado de la variable “x”

```
from classifier
importSentimentClassifier
clf= SentimentClassifier()
x= "Esta comida esta rica"
y= "que horrible esa pelicula"
z= "Tuve una experiencia neutral"
sentimiento=clf.predict(y)
print(sentimiento)

_____
0,18977012365
```

Figura 2: Resultado de la variable "y"

```
from classifier
importSentimentClassifier
clf= SentimentClassifier()
x= "Esta comida esta rica"
y= "que horrible esa pelicula"
z= "Tuve una experiencia neutral"
sentimiento=clf.predict(z)
print(sentimiento)

_____
0,533548978
```

Figura 3: Resultado de la variable "z"

Librería "SentimentClassifier": Esta librería especifica el score de polaridad que va ir desde 0 hasta 1. siendo 0 el extremo negativo. 1 el extremo positivo. Y 0,5 el valor neutral

Ejemplo 2:

son muy mala onda esa vecina
esta muy mala esa película
me gusto mucho la película

	son	mala	onda	esa	vecina	esta	muy	pelicula	Me	gusto	Mucho	la
M1	1	2	1	1	1	0	0	0	0	0	0	0
M2	0	1	0	1	0	1	1	1	0	0	0	0
M3	0	0	0	0	0	0	0	1	1	1	1	1

Figura 4: Conversión del texto a numérico

	son	mala	onda	esa	vecina	esta	muy	pelicula	Me	gusto	Mucho	la	Salida
M1	1	2	1	1	1	0	0	0	0	0	0	0	0 negativo
M2	0	1	0	1	0	1	1	1	0	0	0	0	0 negativo
M3	0	0	0	0	0	0	0	1	1	1	1	1	1 positivo

Figura 5: Resultado

3. CONCLUSIÓN

El análisis de sentimientos se ha consolidado como una herramienta poderosa para comprender mejor a los usuarios y sus emociones. Gracias a librerías como *SentimentClassifier*, es posible automatizar este proceso de forma precisa, permitiendo una rápida clasificación de sentimientos en textos escritos. Su utilidad abarca desde mejorar estrategias de marketing hasta anticipar tendencias o medir la reputación de una marca en tiempo real. En una era donde la opinión del usuario es clave, integrar el análisis de sentimientos en los sistemas de información representa una ventaja competitiva y una fuente confiable de inteligencia emocional digital.

SOBRE LA AUTORA

Ingeniero informático, Docente de la carrera Ing. Informática desde la gestión 2008. Doctorado en Ciencias de la Computación y Maestría en Ciencias de la Computación(Mención Seguridad Informática), cuenta con Diplomados en: Diplomado en Educación superior, Diplomado en formación basada en competencias, Diplomado en Metodología de la investigación, Diplomado en gestión de Seguridad y Auditoría Informática, Diplomado en Herramientas Tecnológicas para educación Superior Virtual, Diplomado en Internet de las cosas y la industria 4.0. tiene Certificación Oficial MikroTik Network Associate. Fue secretaria Ejecutiva de la FUD gestión 2015-2017 y Juez en la competencia de programación ICPC de la carrera Ing. informática.

Figura 6: Fotografía de presentación de la ponencia

ADMINISTRA SERVIDORES LINUX COMO UN PRO (¡SIN MORIR EN EL INTENTO!)

SANTOS I. JUCHASARA COLQUE, Ph.D.

sijucol@gmail.com

Ingeniería Informática

Universidad Nacional "Siglo XX"

Llallagua, Bolivia

RESUMEN

La administración de sistemas Linux ha evolucionado desde un enfoque exclusivo en la línea de comandos hacia soluciones más visuales como Cockpit, una interfaz web desarrollada por Red Hat. Esta herramienta facilita la gestión de servidores al ofrecer un entorno gráfico que complementa la terminal, sin sacrificar control ni seguridad. Cockpit permite monitorear servicios, redes, almacenamiento y procesos del sistema en tiempo real, con soporte para múltiples usuarios gracias a su arquitectura basada en WebSockets, Polkit y systemd. Su instalación y uso son sencillos, y es compatible con distribuciones como Fedora y RHEL. Además, ofrece la posibilidad de extender su funcionalidad mediante módulos personalizados, como se demuestra en la creación de un módulo "FLISOL 2025". Cockpit representa una solución eficaz y moderna para administrar servidores de forma segura y accesible.

1. INTRODUCCIÓN

La creciente complejidad de los entornos tecnológicos ha impulsado la necesidad de herramientas que simplifiquen la administración de sistemas sin comprometer su potencia y seguridad. En este contexto, Cockpit surge como una solución moderna que combina la robustez de Linux con una interfaz gráfica amigable. Diseñada por Red Hat, esta plataforma permite gestionar servicios, redes, almacenamiento y otros componentes del sistema desde un navegador web, facilitando el trabajo de los administradores sin reemplazar la línea de comandos. Este documento analiza los fundamentos, beneficios y arquitectura de Cockpit, así como su aplicación práctica en entornos reales.

2. DESARROLLO

Evolución en la administración de sistemas Linux

Linux históricamente es administrado mediante la línea de comandos (CLI), lo que exige un conocimiento profundo de herramientas como systemd, journalctl, iptables/nftables, entre otras. Sin embargo, con la creciente complejidad de los entornos IT, han surgido interfaces web (WebUI) que facilitan la gestión sin sacrificar el poder del sistema operativo. Cockpit es una solución desarrollada por Red Hat que ofrece una interfaz visual sin reemplazar la terminal, sino la complementa.

Problemas clásicos del sysadmin tradicional

- **Errores humanos:** En CLI, un solo carácter incorrecto puede causar consecuencias graves. Por ejemplo (rm -rf /) puede ser catastrófico.

- **Curva de aprendizaje:** El dominio de comandos, scripts Bash, permisos, servicios (systemd), redes, firewalld, etc., requiere años de experiencia. Muchos comandos no son intuitivos, y su uso correcto requiere comprensión del sistema subyacente. Ejemplo: Para configurar una interfaz de red manualmente:

  nmcli con mod enp0s3 ipv4.addresses 192.168.1.10/24

- **Falta de Escalabilidad:** La administración de múltiples servidores con CLI exige uso de herramientas externas (como SSH en scripts, Ansible, etc.).

- **Interfaz no intuitiva para tareas complejas:** Tareas como gestionar LVM, configurar RAID, o ver logs en tiempo real son mucho más complicadas en terminal que mediante una interfaz gráfica estructurada como la de Cockpit.

¿Por qué un enfoque visual sin perder control?

Cockpit no es un reemplazo de la CLI, sino una capa de abstracción que:

- Reduce errores en configuraciones críticas.
- Permite monitoreo en tiempo real.
- Facilita la administración remota.
- Usa TLS/SSL para cifrar la comunicación.
- Integración con Polkit (Polkit es un marco de autenticación utilizado en entornos de escritorio gráficos de Linux) para autorización granular.

Arquitectura de Cockpit:

- Modelo cliente-servidor:

Cockpit-ws (Web Service): El componente cockpit-ws funciona como el servidor web que:

- Sirve la interfaz de usuario basada en web
- Maneja las conexiones HTTPS/WebSocket de los navegadores
- Escucha por defecto en el puerto 9090.
- Gestiona la autenticación de usuarios
- Enruta las comunicaciones entre el navegador y el cockpit-bridge

Este componente se ejecuta con privilegios limitados y actúa como la frontera segura entre los clientes web externos y el sistema.

Cockpit-bridge: El cockpit-bridge funciona como un puente entre la interfaz web y el sistema operativo:

- Se ejecuta con los privilegios del usuario autenticado
- Proporciona acceso a los servicios del sistema (systemd, storage, networking, etc.)
- Traduce las solicitudes de la interfaz web en comandos del sistema
- Ejecuta procesos y recopila resultados

• Proporciona acceso a APIs del sistema (D-Bus, etc.)

Cada sesión de usuario tiene su propio proceso bridge, lo que garantiza el aislamiento de sesiones y la correcta aplicación de permisos.

Tecnologías subyacentes: Cockpit se construye sobre:

• **JavaScript/HTML5**: Para la interfaz de usuario

• **WebSockets:** Para comunicación en tiempo real con el sistema

• **Polkit:** Para el manejo de privilegios

• **SystemD:** Integración profunda para gestión de servicios

• **SSH:** Para conexiones remotas seguras

Esta combinación permite que Cockpit ofrezca una experiencia interactiva sin sacrificar la seguridad o la potencia que caracteriza a Linux.

Compatibilidad

	Tested	Available	
fedora	✓	✓	View instructions
Red Hat Enterprise Linux	✓	✓	View instructions
fedora COREOS	✓	✓	View instructions
CentOS	✓	✓	View instructions
debian	✓	✓	View instructions
archlinux	✓	✓	View instructions
ubuntu	✓	✓	View instructions
Clear Linux		✓	View instructions
Tumbleweed		✓	View instructions
SUSE Linux Enterprise Micro		✓	View instructions

Figura 1: Compatibilidad

Demostración práctica (Fedora):

- **Instalación básica:** Pasos iniciales para comenzar a utilizar cockpit desde Fedora siga las instrucciones. Para otras distribuciones visitar: https://cockpit-project.org/running.html

```
# Instalar cockpit
sudo dnf install cockpit
# Habilitar el servicio
sudo systemctl enable --now  cockpit.socket
# Inicia el seervicio
sudo systemctl start cockpit.socket
```

Figura 2: Instalación básica

Este conjunto de comandos instala el paquete base, habilita el socket para que se inicie automáticamente durante el arranque del sistema y finalmente inicia el servicio.

- **Configuración del firewall:** Para acceder remotamente a Cockpit, es necesario abrir el puerto correspondiente (por defecto, 9090):

```
sudo firewall-cmd --add-service=cockpit
sudo firewall-cmd --add-service=cockpit --permanent
sudo firewall-cmd --reload
```

Figura 3: configuración del firewall

En el navegador web, introduzca la siguiente direcci´on para acceder a la consola web: https://localhost:9090

- **Instalación de módulos adicionales:** Fedora ofrece diversos módulos que extienden la funcionalidad de Cockpit:

```
# Para gestion de maquinas virtuales
sudo dnf install cockpit-machines
# Para gestion de almacenamiento avanzado
sudo dnf install cockpit-storaged
# Para gestion de contenedores
sudo dnf install cockpit-podman
# Para monitorizacion de rendimiento
sudo dnf install cockpit-pcp
```

Figura 3: instalación de módulos adicionales

Cada módulo se integra perfectamente en la interfaz principal, ampliando las capacidades de administración.

Demo añadiendo mi Propio Módulo a Cockpit

Desarrollar un modelo básico para Cockpit que muestre ”FLISOL 2025” proyecto para entender su arquitectura basada en JavaScript, HTML y DBus.

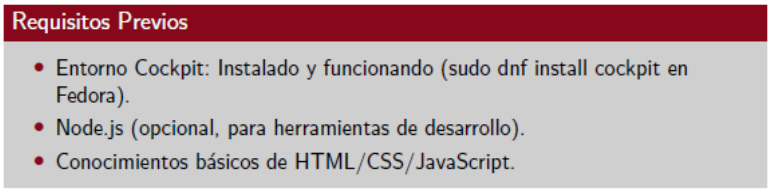

Figura 4: Pasos para crear e integrar un módulo en Fedora/RHEL

- **Paso 1: Estructura del Módulo**

  Cockpit sigue una estructura específica. Crea estos archivos:

```
mkdir -p ~/cockpit-flisol/flisol
cd ~/cockpit-flisol
```

Figura 5: Estructura del módulo.

```
{
  "version": 1,
  "tools": {
    "flisol": {
      "label": "FLISOL 2025",
      "path": "flisol"
    }
  }
}
```

Figura 6: manifest.json: Metadata del módulo.

```
<!DOCTYPE html>
<html>
  <head>
    <title>FLISOL 2025</title>
     <meta charset="utf-8">
     <link href="../base1/cockpit.css" rel="stylesheet">
  </head>
<body>
 <div class="container-fluid" style="padding: 20px;">
  <h1>Bienvenido al FLISOL 2025</h1>
  <p>Este es mi primer modulo para Cockpit.</p>
 </div>
<script src="../base1/cockpit.js"></script>
</body>
</html>
```

Figura 7: flisol/index.html: Interfaz del módulo.

- **Paso 2: Integración con Cockpit**

```
sudo cp -r ~/cockpit-flisol /usr/share/cockpit/
sudo systemctl restart cockpit
sudo rm -rf /var/cache/cockpit/*
```

Figura 8: Copiando el módulo a la ruta oficial

- **Paso 3: Acceder al Módulo**

  Verificar el módulo desarrollado:

  1. Abre Cockpit en tu navegador: https://tuserver:9090 (usa tu IP o localhost).
  2. Verás la opción ˝FLISOL 2025˝ en el menú lateral

Consideraciones finales:

- Cockpit no es una herramienta básica, sino una poderosa interfaz para administración avanzada, segura y moderna.
- Permite disminuir la carga cognitiva sin sacrificar control ni robustez.
- Ideal para entornos educativos, productivos y gubernamentales.
- No se trata de reemplazar la terminal, sino de complementar con herramientas que hagan la administración más accesible, visual e intuitiva.

3. CONCLUSIÓN

Cockpit representa un avance significativo en la administración de sistemas Linux, al ofrecer una interfaz gráfica moderna, intuitiva y segura que no sustituye, sino complementa, el uso de la terminal. Su arquitectura basada en cockpit-ws y cockpit-bridge, junto con tecnologías como WebSockets, Polkit y systemd, le permite ofrecer una experiencia de administración remota eficaz y escalable. A través de esta herramienta, tareas tradicionalmente complejas como la gestión de redes, servicios, almacenamiento y usuarios pueden ejecutarse de forma más sencilla y con menor margen de error, lo que resulta especialmente útil para administradores menos experimentados o entornos donde se busca mayor eficiencia y seguridad.

Además, su capacidad de expansión mediante módulos personalizados, como el ejemplo práctico del módulo “FLISOL 2025”, demuestra la flexibilidad y adaptabilidad de la plataforma para diferentes contextos, incluyendo entornos educativos, institucionales y empresariales. En un mundo donde la complejidad de los sistemas IT sigue creciendo, herramientas como Cockpit permiten a los administradores enfocarse en la solución de problemas y la toma de decisiones estratégicas, sin verse limitados por la dificultad técnica de

las tareas básicas. Sin duda, Cockpit se posiciona como un componente clave en la administración moderna de servidores Linux.

SOBRE EL AUTOR

Doctor en Ciencias de la Computación
Ingenieria Informático, Director de Carrera Ingeniería Informática de la Universidad Nacional “Siglo XX”
Maestría en Ciencias de la Computación Mención Seguridad informática y Software Libre
Diplomado en Diseño Curricular

Diplomado en Herramientas Tecnológicas para Educación Superior Virtual

Diplomado en Nuevas Tecnologías de Información y Comunicación

Diplomado en Auditoria de Sistemas

Diplomado en Educación Superior

Diplomado en Formación Basada en Competencias en la Universidad Boliviana.

Figura 9: Fotografía de presentación de la ponencia

AUTOMATIZACIÓN CONVENCIONAL MULTILINGÜE EN TELEGRAM USANDO HUGGING CHAT UN ENFOQUE CON CONTEXTO PERSONALIZADO

FRANZ VILLCA ARO

franzvillcaaro12345@gmail.com

Ingeniería Informática

Universidad Nacional "Siglo XX"

Llallagua, Bolivia

RESUMEN

Este proyecto consiste en el desarrollo de un bot conversacional multilingüe para Telegram utilizando inteligencia artificial a través del modelo Mistral-7B-Instruct, alojado en la plataforma HuggingFace. El bot está programado en Python e incorpora la capacidad de mantener el contexto de las conversaciones, lo que mejora la coherencia de sus respuestas. Además, permite la integración de comandos personalizados y es compatible con múltiples sistemas operativos. Su implementación se basa en la arquitectura Transformer, reconocida por su eficiencia en tareas de procesamiento de lenguaje natural. La solución es ligera en recursos, accesible para usuarios con conocimientos básicos en programación y cuenta con una licencia MIT, lo que permite su libre uso y modificación para todo tipo de proyectos.

1. INTRODUCCIÓN

En la era digital, la inteligencia artificial (IA) ha revolucionado la manera en que nos comunicamos, permitiendo la creación de asistentes virtuales inteligentes y personalizados. Este proyecto se enfoca en la implementación de un bot conversacional multilingüe para Telegram, utilizando modelos de lenguaje avanzados proporcionados por HuggingFace, específicamente Mistral-7B-Instruct. A diferencia de otros bots, esta propuesta destaca por su capacidad de mantener el contexto de la conversación, lo que le permite generar respuestas coherentes y adaptadas a cada usuario. Además, se permite la integración de comandos personalizados, ampliando sus posibilidades de aplicación. Este enfoque es accesible y flexible gracias a su licencia MIT, y está sustentado en la arquitectura Transformer, pilar de la inteligencia artificial moderna.

2. DESARROLLO

Requisitos de Hardware:

- PC o laptop con Windows, Linux o macOS.
- Al menos 4 GB de RAM (recomendado: 8 GB).
- Conexión a Internet estable.
- Micrófono y cámara (opcional).

Requisitos de Software:

- Python 3.10 o superior.
- Editor de código (como VS Code).
- Cuenta en HuggingFace con token de acceso.
- Bibliotecas: python-telegram-bot, huggingface_hub, requests, etc.
- Telegram instalado para pruebas.

Características:

- Usa inteligencia artificial de HuggingFace (como Mistral).
- Guarda el contexto de la conversación para respuestas coherentes.
- Puedes incluir comandos personalizados /STAR

Licencia:

- Licencia: MIT (libre, abierta y flexible).

- Permite su uso tanto personal como comercial.

- Puedes usar, copiar, modificar y compartir el proyecto.

Sustento teórico-científico:

¿Qué es un modelo de lenguaje?

Un modelo de lenguaje es un sistema de inteligencia artificial capaz de comprender y generar texto similar al que produciría un ser humano.

Arquitectura transformer

El corazón del bot es la arquitectura Transformer. Esta arquitectura fue propuesta por Vaswanien en 2017, y hoy es la base de modelos como GPT, BERT, y Mistral (que usamos aquí).

Fórmula clave

El "atención" (attention) es el componente clave de los Transformers. Le permite al modelo "decidir" a qué palabras del texto debe prestar más atención para generar una respuesta adecuada.

$$Attention(Q,K,V)=softmax(Q\times KT/\sqrt{dk})\times V$$

Q (queries), K (keys), V (values): representaciones matemáticas del texto. Softmax() convierte los resultados en probabilidades.Este proceso guía la generación de texto.

Modelo usado: Mistral-7B-Instruct

Este bot se conecta al modelo Mistral 7B-Instruct, optimizado para seguir instrucciones dadas por los usuarios.

Diagrama conceptual del funcionamiento del BOT

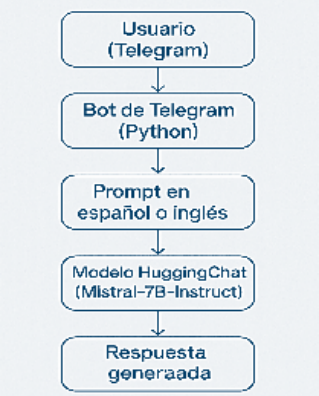

Figura 1: Diagrama

3. CONCLUSIÓN

El desarrollo de un bot conversacional multilingüe con HuggingChat y Telegram demuestra cómo la inteligencia artificial puede integrarse fácilmente en aplicaciones cotidianas. Gracias a modelos avanzados como Mistral y herramientas accesibles, es posible crear soluciones personalizadas, eficientes y escalables para la interacción automática en tiempo real.

SOBRE EL AUTOR

Franz Villca Aro es estudiante de la carrera de Ingeniería Informática en la Universidad Nacional Siglo XX, donde fue auxiliar de Taller de Programación. Además, cuenta con formación técnica como Técnico Medio en Mecánica Automotriz.

A lo largo de su trayectoria universitaria, ha demostrado un alto desempeño en el área de la programación y la robótica, logrando reconocimientos importantes como el 1er lugar en la competencia de programación UNSXX-ICPC en sus fases nacionales 2023 y 2024, así como el 3er lugar en el I Concurso de Fútbol de Robots UNSXX 2024. También ha representado a su universidad y carrera en las Finales Sudamérica/Sur ICPC 2023 y 2024, mostrando un fuerte compromiso con la excelencia académica y técnica.

Actualmente, continúa su formación con el objetivo de aportar soluciones innovadoras en el campo de la informática y la tecnología.

Figura 2: Fotografía de la presentación de la ponencia

GPT4ALL: AUTOMATIZACIÓN, ANÁLISIS Y PRIVACIDAD CON LA IA LOCAL

ALEXANDER LINO FERNANDEZ CALLAPA

alexfer67@gmail.com

Ingeniería Informática

Universidad Nacional "Siglo XX"

Llallagua, Bolivia

RESUMEN

GPT4All es una plataforma de inteligencia artificial local enfocada en la privacidad del usuario y el control total de los datos. Permite ejecutar modelos de lenguaje sin conexión a la nube, aprovechando tecnologías como llama.cpp y ggml para optimizar el rendimiento en CPU. Con requisitos accesibles (Windows, Linux o macOS y al menos 4 GB de RAM), su arquitectura simplificada facilita la implementación de modelos como Mistral o DeepSeek sin necesidad de frameworks pesados. Bajo licencia MIT, ofrece flexibilidad para usos personales y comerciales, representando una alternativa ética, eficiente y libre para el uso de la IA.

1. INTRODUCCIÓN

La creciente dependencia de plataformas de inteligencia artificial basadas en la nube ha generado preocupaciones legítimas sobre la privacidad y el control de los datos. En este contexto, surge GPT4All, una alternativa local que permite ejecutar modelos de lenguaje natural directamente en el dispositivo del usuario. Esta solución no solo protege la información personal, sino que también promueve el uso del software libre, brindando mayor autonomía tecnológica. Su implementación es accesible, eficiente y no requiere GPU, lo que la convierte en una herramienta poderosa para quienes buscan aplicar IA de forma ética y segura.

2. DESARROLLO

GPT4All: Conceptos Clave:

Definición de GPT4All

GPT4All es una plataforma de inteligencia artificial diseñada para funcionar de manera local, que permite a los usuarios ejecutar modelos de procesamiento del lenguaje natural sin depender de recursos externos. Se destaca por su enfoque en la privacidad y el control total sobre los datos del usuario, promoviendo así software libre y accesible para todos.

Problemas que aborda: privacidad y control

En un mundo donde la dependencia de la nube es cada vez más común, GPT4All se presenta como una solución que resuelve problemas críticos como la privacidad. La incapacidad de controlar los datos en plataformas de IA tradicionales puede llevar a brechas de seguridad y falta de confianza. GPT4All permite a los usuarios tener control total sobre su información, asegurando un uso más responsable y ético de la inteligencia artificial.

Requerimientos del sistema

Para ejecutar GPT4All, se requiere un sistema operativo Windows, Linux o macOS. El mínimo de RAM necesario es de 4 GB, pero se recomienda contar con al menos 8 GB para un rendimiento óptimo. No se necesita una GPU, aunque su uso puede mejorar el rendimiento.

Licencia MIT

GPT4All está licenciado bajo la Licencia MIT, lo que significa que es un software libre. Esta licencia permite a los usuarios utilizar, modificar y distribuir el software en sus proyectos, tanto de forma personal como comercial, brindando flexibilidad y libertad.

Arquitectura y Fundamentos:

Diagrama de flujo de implementación

La arquitectura de GPT4All se representa mediante un diagrama de flujo que ilustra su interfaz de usuario conectada al backend. El flujo comienza con la interfaz, que interactúa con el backend, utilizando componentes como llama.cpp o ggml para procesar las solicitudes y acceder a modelos de lenguaje como DeepSeek o Mistral. Este diseño sencillo permite la ejecución local sin la necesidad de frameworks complejos.

Llama.cpp: Es una biblioteca optimizada en C++ que permite ejecutar modelos LLM como LLaMA o Mistral en CPU, de forma rápida y eficiente, sin necesidad de GPU ni frameworks pesados como PyTorch o TensorFlow.

Ggml: Es una librería de bajo nivel en C que permite ejecutar y optimizar modelos de IA en CPU mediante técnicas como cuantización, reduciendo el tamaño del modelo sin perder mucha precisión.

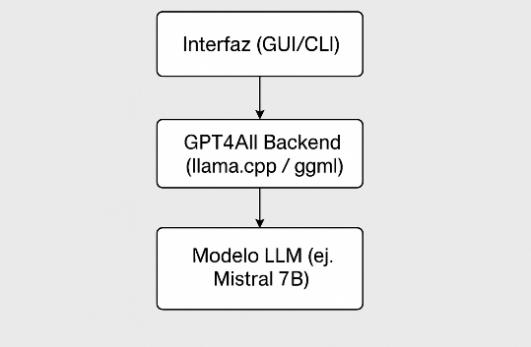

Figura 1: Diagrama de flujo

3. CONCLUSIÓN

GPT4All representa un avance significativo en la democratización de la inteligencia artificial, al ofrecer una solución accesible, potente y centrada en la privacidad. Su capacidad de funcionar localmente sin depender de servicios en la nube permite a los usuarios mantener un control completo sobre sus datos, reduciendo los riesgos asociados al almacenamiento externo y fortaleciendo la seguridad informática. Además, al basarse en herramientas optimizadas como llama.cpp y ggml, hace posible ejecutar modelos avanzados en hardware común, eliminando barreras técnicas y económicas para el desarrollo de soluciones personalizadas.

El enfoque de código abierto y la licencia MIT fomentan la innovación, permitiendo que tanto desarrolladores individuales como organizaciones adopten y adapten la tecnología a sus necesidades

específicas. GPT4All no solo promueve la eficiencia y el rendimiento en la automatización y análisis de datos, sino que también impulsa prácticas más éticas y sostenibles en el uso de la inteligencia artificial. En un entorno cada vez más consciente del valor de la privacidad, esta herramienta se posiciona como una alternativa estratégica para quienes buscan implementar IA responsable, transparente y al alcance de todos.

SOBRE EL AUTOR

Estudiante de seguridad de cuarto año de la Carrera Ingeniería Informática Universidad Nacional “Siglo XX”

Figura 2: Fotografía de presentación de la ponencia.

GALERÍA DE IMÁGENES
CONGRESO DE SOFTWARE LIBRE – XXI
FLISOL LLALLAGUA – BOLIVIA

RECONOCIMIENTOS Y RECUERDOS

ENTREGA DE RECUERDO Y CERTIFICADO A LOS PONENTES

FOTOGRAFÍAS DEL EVENTO

FLISOL LLALLAGUA – BOLIVIA

FOTOGRAFÍAS DEL INSTALL FEST (CONFIGURACIÓN DE SISTEMAS Y APLICACIONES LIBRES EN LOS EQUIPOS DE LOS ASISTENTES)