



Cuadernos Técnicos Universitarios  
de la **DGTIC**

ISSN-e: 3061-8096



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

DIRECCIÓN GENERAL DE CÓMPUTO Y DE  
TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN



**DGTIC UNAM**

DIRECCIÓN GENERAL DE CÓMPUTO Y  
DE TECNOLOGÍAS DE INFORMACIÓN  
Y COMUNICACIÓN



## CUADERNOS TÉCNICOS UNIVERSITARIOS DE LA DGTIC

### Consejo Editorial

Instituto Politécnico Nacional, Dr. Marco Antonio Moreno Ibarra • Instituto Tecnológico de Morelia, Dr. Juan Carlos Olivares Rojas • Universidad Nacional Autónoma de México • Dra. Laurette Godinas (Instituto de Investigaciones Bibliográficas) • Dr. Paul Rolando Maya Ortiz (Facultad de Ingeniería) • Dra. María Cristina Verde Rodarte (Instituto de Ingeniería).

### Equipo Editorial

Editor Responsable Héctor Benítez Pérez • Editora Académica Marcela J. Peñaloza Báez • Editora Técnica Pamela Valdés Reséndiz • Asistente Editorial Eric Villar Juárez • Corrector de estilo Pablo Vázquez Castellanos • Normalización de citas y referencias Jorge Alberto Colín Rojas • Diseño Gráfico y editorial Miguel Ángel Islas Flores • Maquetación Jonathan Cedillo Castro • Servicio Social Juan Manuel Campos Coto

Para citar un reporte técnico de la obra: Apellidos 1 Apellidos 2, Iniciales nombres. (2026). Título del reporte técnico. *Cuadernos Técnicos Universitarios de la DGTIC*, 4 (2), páginas (N1-N2). <https://doi.org/10.22201/dgtic.30618096e.2026.4.2>

CUADERNOS TÉCNICOS UNIVERSITARIOS DE LA DGTIC, Año 4, No. 2, abril-junio 2026, es una publicación trimestral editada por la Universidad Nacional Autónoma de México, Ciudad Universitaria, Alcaldía Coyoacán, C.P. 04510, Ciudad de México, a través de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación, Circuito Exterior s/n, frente a la Facultad de Contaduría y Administración, Ciudad Universitaria, Alcaldía Coyoacán, C.P. 04510, Tel. (55) 5622 8502 y 5622 8354, URL: <https://cuadernos.tic.unam.mx>, correo electrónico [cuadernostecnicos-dgtic@unam.mx](mailto:cuadernostecnicos-dgtic@unam.mx), Editor responsable: Dr. Héctor Benítez Pérez. Certificado de Reserva de Derechos al Uso Exclusivo de Título: 04-2023-100610042700-102, ISSN-e: 3061-8096, ambos otorgados por el Instituto Nacional del Derecho de Autor. Dra. Marcela J. Peñaloza Báez, responsable de la última actualización de este número, Circuito Exterior s/n, frente a la Facultad de Contaduría y Administración, Ciudad Universitaria, Alcaldía Coyoacán, C.P. 04510, Ciudad de México. Fecha de la última modificación, 13 de mayo de 2026.

El contenido de los textos es responsabilidad de los autores y no refleja forzosamente el punto de vista de los dictaminadores, o de los miembros del Consejo Editorial, o del Comité Editorial de la DGTIC, o la postura de las personas editoras y la editorial de la publicación.

Se autoriza la reproducción total o parcial de los textos aquí publicados siempre y cuando se cite la fuente completa y la dirección electrónica de la publicación.

AVISO DE PRIVACIDAD

<https://www.tic.unam.mx/avisosprivacidad/>

## COLABORADORES

Dirección General de Publicaciones y Fomento Editorial a través de Socorro Venegas Pérez, Directora General • Guillermo Chávez Sánchez, Subdirector de Revistas Académicas y Publicaciones Digitales • Jaqueline Segura Bautista, Gestión de Revistas Académicas • Jorge Pérez García, Jefe del Departamento de Soporte Técnico de Sistemas Editoriales • Juan Manuel Rodríguez Martínez, Jefe de Desarrollo.

Dirección General de Cómputo y de Tecnologías de Información y Comunicación a través de Ana Yuri Ramírez Molina, Directora de Colaboración y Vinculación • Juan Manuel Castillejos Reyes, Líder de proyecto de Soporte Técnico • Alberto González Guízar, Infraestructura y software • José Othoniel Chamú Arias, Servidores y bases de datos.

## UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Dr. Leonardo Lomelí Vanegas, Rector • Dra. Patricia Dolores Dávila Aranda, Secretaria General • Mtro. Hugo Alejandro Concha Cantú, Abogado General • Mtro. Tomás Humberto Rubio Pérez, Secretario Administrativo • Dra. Diana Tamara Martínez Ruíz, Secretaria de Desarrollo Institucional • Dr. Héctor Benítez Pérez, Director General de Cómputo y de Tecnologías de Información y Comunicación.

## CONTENIDO

Despliegue de n8n en producción para automatización de flujos de trabajo Ernesto Pathros Ibarra García, Julio Cesar Cruz Estrada .....	8
Modelo lógico para la transferencia de líquidos en entornos tridimensionales web e inmersivos Tayde Martín Cruz Lovera .....	29
Análisis documental para la toma de decisiones en el ámbito de las tecnologías de la información y comunicación Rebeca Valenzuela Argüelles .....	49
Modernización y estandarización en el desarrollo de complementos para Moodle Pablo Enrique Zenil Rivas .....	65
Implementación de un servidor de impresión bajo un enfoque de seguridad y sostenibilidad tecnológica Alfonso Gutiérrez Molina.....	79
Conector de gestión de pagos digitales en línea Nidia Cendejas Cervantes .....	99
Implementación de una infraestructura de correo electrónico para pruebas con Exchange Server Laura Martínez Sierra .....	108
Propuesta de un sistema de adquisición y generación de señales en plataformas Raspberry Pi Signal acquisition and generation system on Raspberry Pi platforms Benjamín Valera Orozco, Gerardo Antonio Ruiz Botello, Karen Lucero Roldán Serrato.....	124
Sistema para la evaluación de resúmenes de trabajos libres y carteles en la Escuela Nacional de Ciencias Forenses Luis Floriberto Flores Aguilar .....	148
Manejo distribuido de matrices a gran escala en un equipo HPC Adrián Durán Chavesti.....	165
Personas revisoras julio 2025 a junio 2026 .....	185

# Despliegue de n8n en producción para automatización de flujos de trabajo

*Deployment of n8n in production for workflow automation*

## Información del reporte:

Licencia Creative Commons



El contenido de los textos es responsabilidad de los autores y no refleja forzosamente el punto de vista de los dictaminadores, o de los miembros del Comité Editorial, o la postura del editor y la editorial de la publicación.

Para citar este reporte técnico:

Ibarra García, E.P. y Cruz Estrada, J.C. (2026). Despliegue de n8n en producción para automatización de flujos de trabajo. *Cuadernos Técnicos Universitarios de la DGTIC*, 4(2), páginas (8 - 28). <https://doi.org/10.22201/dgtic.30618096e.2026.4.2.171>

## Ernesto Pathros Ibarra García

Instituto de Investigaciones Sociales  
Universidad Nacional Autónoma de México  
[pathros@sociales.unam.mx](mailto:pathros@sociales.unam.mx)  
ORCID: 0009-0007-5972-5558

## Julio Cesar Cruz Estrada

Instituto de Investigaciones Sociales  
Universidad Nacional Autónoma de México  
[juliocruz@sociales.unam.mx](mailto:juliocruz@sociales.unam.mx)  
ORCID: 0000-0003-0107-6616

## Resumen

La automatización de flujos de trabajo mediante plataformas de bajo código representa una oportunidad estratégica para optimizar procesos en instituciones de educación superior. En este contexto, se llevó a cabo la instalación, configuración y despliegue en producción de la plataforma n8n en el Instituto de Investigaciones Sociales de la Universidad Nacional Autónoma de México, como infraestructura de soporte para el “Taller Desarrollo de Aplicaciones con IA” de Educación Continua. La metodología siguió cuatro etapas: diseño de arquitectura mediante contenedores con Docker; configuración del entorno con variables seguras y *proxy* inverso Traefik; instalación verificada con certificados HTTPS automatizados mediante Let’s Encrypt; así como pruebas funcionales de integración con servicios externos. Se implementó adicionalmente un *script* de actualización semiautomática que respalda la base de datos PostgreSQL y los volúmenes de datos antes de aplicar nuevas versiones, lo cual permitió un mantenimiento controlado y reversible. Los resultados mostraron una plataforma estable con disponibilidad superior al 98% durante

el período de operación, capaz de ejecutar flujos de trabajo que integran modelos de lenguaje con servicios institucionales. Se concluye que el despliegue auto alojado de n8n constituye una alternativa viable y económica para instituciones académicas que requieren automatizar procesos de investigación e incorporar capacidades de inteligencia artificial en sus flujos de trabajo, al tiempo que se mantiene el control total sobre los datos institucionales.

**Palabras clave:** Automatización de bajo código, contenedores Docker, infraestructura como código, inteligencia artificial en educación, plataformas de bajo código.

### Abstract

*Automating workflows using low-code platforms presents a strategic opportunity for process optimization in higher education institutions. The installation, configuration and production deployment of the n8n platform was carried out at the Instituto de Investigaciones Sociales of the Universidad Nacional Autónoma de México, as the support infrastructure for the "Taller Desarrollo de Aplicaciones con IA" in Educación Continua. The implementation process consisted of four phases: designing a containerized architecture with Docker; configuring the environment with secure variables and a reverse proxy using Traefik; verifying the installation with automatic HTTPS certificates from Let's Encrypt; and testing functional integration with external services. In this context, a semi-automatic update script was also developed to back up the PostgreSQL data volume before applying new versions, thus facilitating reversible and controlled maintenance. The results demonstrated that the platform is stable, achieving over 98% availability during operation, and is capable of executing workflows that integrate language models with institutional services. It can be concluded that the self-deployment of n8n is a feasible and cost-effective alternative for academic institutions that require automation of research processes and seek to integrate artificial intelligence capabilities into their workflows, while maintaining full control over their own institutional data.*

**Keywords:** Artificial intelligence in education, Docker containers, infrastructure as code, low-code platforms, low-code automation.

## 1. INTRODUCCIÓN

En los últimos años, se ha producido un creciente interés en las plataformas de automatización de flujos de trabajo de bajo código, tanto en el entorno empresarial como en el académico (Venkiteela, 2025; Joshi, 2025). La capacidad de integrar sistemas heterogéneos y servicios web mediante flujos visuales ha demostrado ser capaz de agilizar procesos repetitivos y reducir errores humanos (Hartz, 2025). De manera particular, en el ámbito de las operaciones de desarrollo, se ha encontrado que la configuración manual de *pipelines*, contenedores y despliegues incluye tareas repetitivas y propensas a error, por lo que la automatización resulta esencial para mejorar la fiabilidad y rapidez de entrega de *software* (Joshi, 2025). Diversos estudios recientes destacan la incorporación de plataformas nativas de la nube, como contenedores Docker y orquestadores tipo *Kubernetes*, junto con inteligencia artificial generativa en estas automatizaciones, lo que incrementa la eficiencia sin sacrificar seguridad ni gobernanza (Jennings, 2025; Traefik Labs, 2022).

La plataforma n8n se creó como una alternativa de código abierto de propósito general para orquestar flujos de trabajo complejos (Venkiteela, 2025). A diferencia de servicios propietarios de integración tipo

"If This Then That", n8n cuenta con una interfaz gráfica donde se integran y conectan fácilmente distintos servicios: APIs, bases de datos, aplicaciones en la nube e incluso lógica personalizada mediante código JavaScript. Esta plataforma permite el auto alojamiento con control total de los datos y una extensibilidad mediante nodos personalizados y API (Venkiteela, 2025). Según investigaciones especializadas, n8n es capaz de conectar aplicaciones empresariales con servicios de inteligencia artificial en un mismo flujo, lo que posibilita la creación de automatizaciones híbridas que combinan reglas determinísticas con razonamiento cognitivo. En pruebas comparativas, n8n ha demostrado un desempeño escalable y confiable, similar o superior al de otras herramientas similares como Node-RED, Apache Airflow o Zapier, especialmente debido a su modelo abierto y auto alojado (Venkiteela, 2025).

Es importante destacar que hay varias alternativas para implementar n8n. Por un lado, el servicio n8n Cloud permite trabajar de inmediato sin necesidad de realizar instalaciones, aunque impone algunas limitaciones según el plan contratado (como restricciones en la cantidad de ejecuciones y en la cantidad de flujos de trabajo activos). El plan Starter, por ejemplo, permite 2,500 ejecuciones mensuales y hasta cinco flujos de trabajo activos (n8n-io, 2024). Por otro lado, n8n puede alojarse en infraestructura propia o en un servicio de servidor virtual privado (VPS), lo que desbloquea flexibilidad, privacidad y escalabilidad. En este último caso, se tiene la libertad de ejecutar y activar cualquier número de flujos de trabajo, en donde el único límite es la potencia del equipo de cómputo elegido. El servicio de *hosting* VPS ofrece control y buen rendimiento, además de ser más asequible y escalable que un servidor dedicado; diversas compañías ofrecen estos servicios con diferentes planes que se adaptan a los requerimientos específicos de cada implementación.

Esta apertura y neutralidad por parte del proveedor hacen que n8n sea una plataforma atractiva en ambientes académicos, donde a menudo se requiere la integración de múltiples fuentes de datos y APIs de manera personalizada y con restricciones presupuestarias. De hecho, plataformas visuales como Node-RED ya han sido utilizadas en proyectos de investigación por su baja curva de aprendizaje y capacidad de combinar servicios diversos; por ejemplo, en bioinformática se ha utilizado Node-RED para flujos de análisis genómica, donde se obtuvo un resultado útil no sólo para especialistas en el campo sino también como herramienta educativa para estudiantes gracias a su interfaz visual intuitiva (Fàbregas, 2022). Estas características muestran que herramientas de flujo de trabajo de bajo código pueden trasladar la automatización a ambientes académicos, las cuales permiten a investigadores y personal técnico-académico implementar integraciones complejas sin necesidad de programación tradicional.

Por otro lado, la convergencia reciente entre la automatización de flujos de trabajo y la inteligencia artificial reafirma su relevancia en el ámbito académico. En ciencias sociales computacionales, se explora cómo los modelos de lenguaje avanzados pueden ser integrados en los flujos de trabajo para ayudar en tareas de análisis de datos, redacción de informes o formulación de hipótesis (Ziems *et al.*, 2024; Manning *et al.*, 2024). Se ha sugerido que los investigadores que aprovechen la automatización cognitiva, como agentes de IA integrados en plataformas de flujo, podrían aumentar significativamente su productividad (Korinek, 2023). Estas tendencias indican que la adopción de plataformas como n8n en el ámbito académico no sólo optimiza procesos técnicos, sino que también actúa como un puente formativo, ya que, al exponer al personal académico a herramientas de flujo visual con IA integrada, se facilita la comprensión práctica del funcionamiento de la inteligencia artificial y la mejora de procesos mediante automatización.

En el Instituto de Investigaciones Sociales de la Universidad Nacional Autónoma de México (IIS-UNAM) se identificó la necesidad de disponer de una plataforma de automatización que cumpliera con el doble propósito de ser infraestructura para proyectos de investigación y material didáctico para la capacitación

en tecnologías emergentes. En este sentido, el Departamento de Cómputo del IIS-UNAM desarrolló e impartió el “Taller Desarrollo de Aplicaciones con IA” a través de Educación Continua durante 2025, en el que se utilizó n8n como herramienta central para enseñar automatización de flujos de trabajo con integración de modelos de lenguaje. Sin embargo, la documentación disponible sobre la implementación de n8n en ambientes de producción académica se encontraba dispersa y enfocada fundamentalmente en entornos empresariales. Los asistentes al taller, investigadores y técnicos de ciencias sociales necesitaban una guía sistemática que abordara desde la implementación hasta el mantenimiento continuo, enfocada en seguridad y mejores prácticas para instituciones de educación superior. Por lo tanto, el objetivo fue implementar, configurar y desplegar en producción la plataforma n8n en el IIS-UNAM como infraestructura de automatización de flujos de trabajo con integración de inteligencia artificial, así como documentar un procedimiento sistemático y reproducible que otras instituciones académicas pudieran adaptar para implementar soluciones similares.

## 2. DESARROLLO TÉCNICO

### 2.1 METODOLOGÍA

Para la instalación y despliegue en producción de n8n, se siguió una metodología estructurada en cuatro etapas: diseño de la arquitectura, configuración del entorno, instalación de la solución y pruebas, junto con procedimientos de mantenimiento. El procedimiento se realizó sobre un servidor Linux (Ubuntu 22.04 LTS) configurado como servidor virtual privado con recursos de nivel producción (8 vCPU, 16 GB de RAM), a fin de asegurar el rendimiento y la estabilidad de la plataforma. Cabe mencionar que este procedimiento también es aplicable a equipos de cómputo local que cumplan con los requisitos mínimos de *hardware*: al menos, 4 vCPU y 8 GB de memoria RAM para una configuración básica, o bien, 8 vCPU y 16 GB de memoria RAM para ambientes de producción con mayor demanda.

Los requisitos adicionales considerados incluyeron: sistema operativo Ubuntu 22.04 o más reciente, un nombre de dominio registrado para apuntar al servicio (recomendado para certificados SSL), acceso SSH al servicio VPS, Docker y Docker Compose instalados y activos, así como familiaridad con comandos básicos de Linux y conocimiento de Docker. Estos prerrequisitos resultaron fundamentales para garantizar un despliegue exitoso y un mantenimiento adecuado de la plataforma a lo largo del tiempo.

#### 2.1.1 DISEÑO

En la etapa de diseño se planeó la arquitectura del sistema, en la que se priorizó la seguridad, la persistencia de datos y la facilidad de mantenimiento. Se decidió utilizar un despliegue mediante contenedores utilizando Docker para envolver el n8n y sus dependencias, ya que la virtualización mediante contenedores permite tener entornos homogéneos y portátiles entre desarrollo y producción (Joshi, 2025; Jennings, 2025). Esta decisión se fundamentó en la necesidad de facilitar la creación, administración y ejecución del servicio de bases de datos y del servicio n8n en sí. Se definen tres componentes principales en un conjunto de contenedores Docker orquestados con Docker Compose: (1) un contenedor para la aplicación n8n; (2) un contenedor de base de datos PostgreSQL para almacenar de forma persistente la información de la plataforma —flujos de trabajo, credenciales cifradas e historiales—; y (3) un contenedor *proxy* inverso Traefik, encargado de gestionar el tráfico HTTP/HTTPS hacia n8n. Esta arquitectura modular asegura que

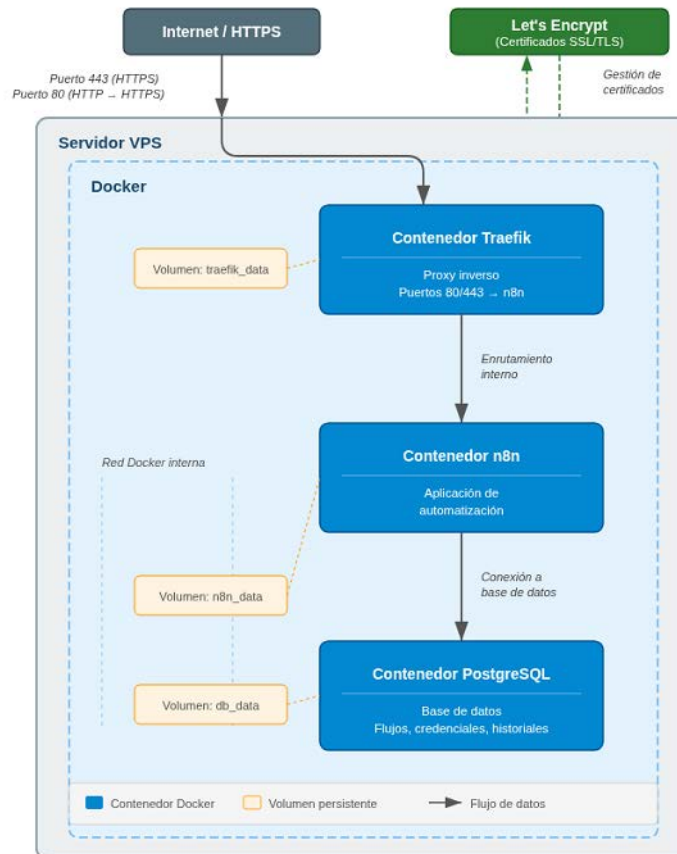
cada componente pueda actualizarse o reiniciarse de forma independiente y promueve la segregación de responsabilidades.

La elección de PostgreSQL como base de datos se basó en las recomendaciones oficiales de n8n para entornos productivos, donde una base SQL robusta ofrece mejor escalabilidad que el archivo SQLite predeterminado (n8n-io, 2024).

La arquitectura implementada se ilustra en la Figura 1, donde se observa la disposición de los contenedores Docker y el flujo de comunicación entre componentes. El servidor VPS aloja tres contenedores principales: Traefik como *proxy* inverso y gestor de certificados SSL/TLS, n8n como aplicación de automatización y PostgreSQL como sistema de gestión de base de datos. La red Docker interna permite la comunicación segura entre contenedores, mientras que Traefik gestiona el tráfico HTTPS externo y la renovación automática de certificados mediante Let's Encrypt.

**Figura 1**

*Arquitectura de despliegue de n8n en producción mediante contenedores Docker*



Se diseñó la persistencia de datos junto con el almacenamiento de la base de datos en un volumen Docker dedicado y un volumen adicional para los datos propios de n8n. Esto asegura que, en caso de reinicios o actualizaciones de contenedores, no se pierda la información de los flujos

ni las configuraciones del sistema. Esta consideración resultó especialmente importante dado el contexto académico, en el que la pérdida de flujos de trabajo desarrollados durante el taller hubiera tenido un impacto significativo en la experiencia de aprendizaje de los participantes. La decisión de utilizar contenedores Docker para el despliegue se fundamentó en las mejores prácticas documentadas para investigación reproducible, donde la virtualización mediante contenedores garantiza que el entorno de ejecución sea consistente y portable entre diferentes sistemas (Nüst *et al.*, 2020). Adicionalmente, se implementaron medidas de seguridad específicas para contenedores, considerando que, aunque la virtualización mediante contenedores proporciona aislamiento, el *software* malicioso puede explotar vulnerabilidades en imágenes de contenedores para atacar el sistema anfitrión (Devi Priya *et al.*, 2023)

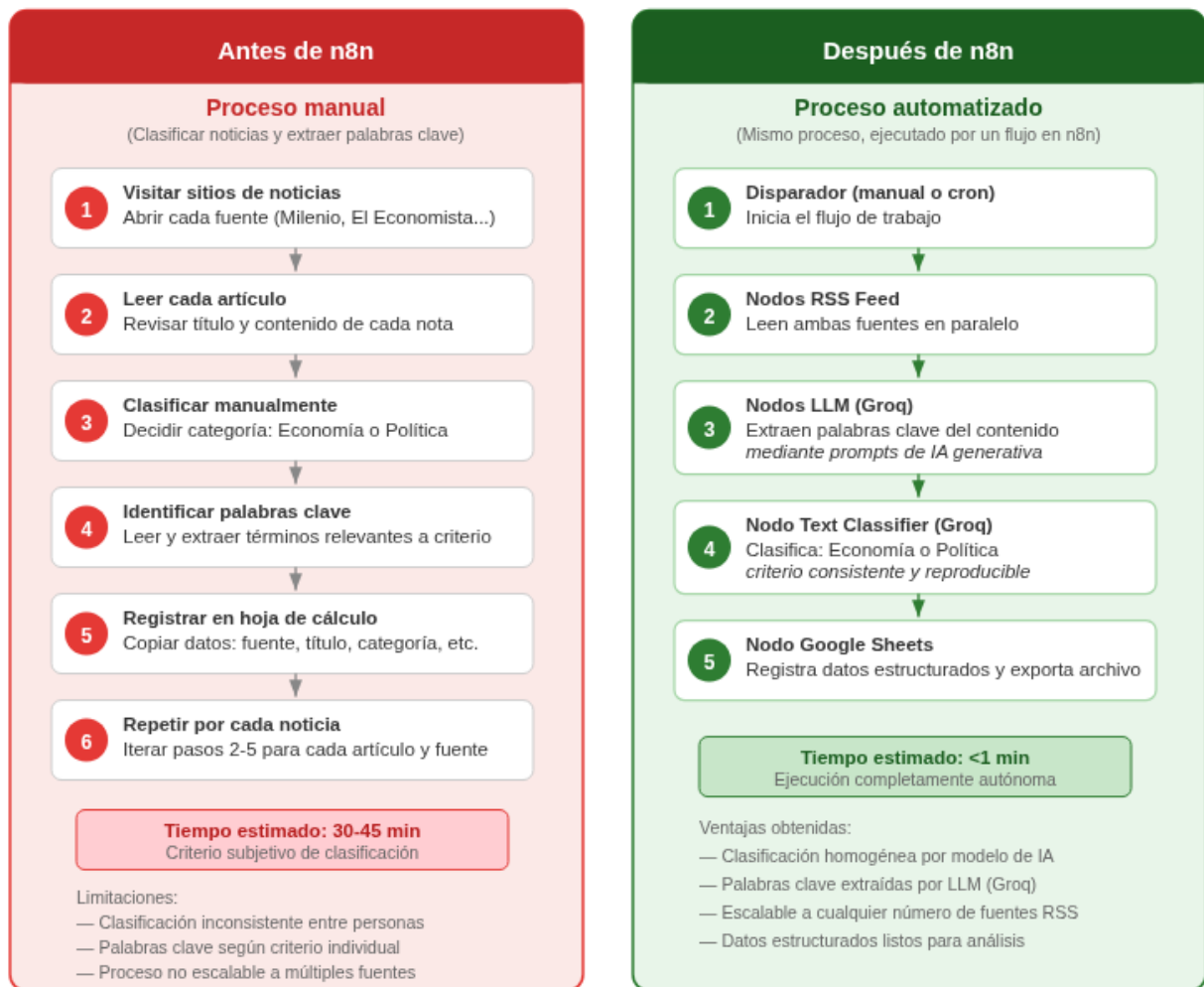
El *proxy* Traefik fue elegido en lugar de otros como Nginx por su integración nativa con Docker y soporte automático de HTTPS con Let's Encrypt (Traefik Labs, 2022). Se configuró Traefik para que monitoreara automáticamente los contenedores expuestos y gestionara los certificados TLS de manera automática, lo que facilitó la habilitación de la conexión segura, necesaria ya que algunas funcionalidades de n8n, como la autenticación OAuth2 con servicios de Google, requieren un *endpoint* HTTPS válido. Además, se aprovechó Traefik para centralizar la entrada a través de los puertos 80/443 del servidor, manteniendo al contenedor de n8n sin exposición directa a Internet, lo que permitió añadir una capa de seguridad adicional al reducir la superficie de ataque potencial.

En lo que respecta a la seguridad y control de acceso, se planificaron varias medidas desde la etapa de diseño. En primer lugar, se contempló la habilitación de la autenticación básica HTTP en n8n para proteger la interfaz web con credenciales durante el primer arranque del sistema. Cabe señalar que, en versiones recientes de n8n, estas variables de autenticación son reconocidas únicamente durante la configuración inicial, ya que, una vez creada la cuenta de propietario a través de la interfaz web, el sistema nativo de gestión de usuarios (*User Management*), basado en base de datos, toma precedencia y la autenticación básica deja de utilizarse. Asimismo, se diseñó la utilización de variables de entorno para gestionar la información sensible (contraseñas de base de datos, claves de API, usuario/contraseña de autenticación), manteniéndolas fuera de los archivos de configuración en texto plano. Finalmente, se diseñó un esquema de actualización y mantenimiento mediante un *script* de actualización semiautomático capaz de respaldar datos y renovar la imagen de n8n. De esta manera, se aseguró que la plataforma pudiera mantenerse actualizada con un mínimo de intervención y riesgo.

La Figura 2 ilustra la comparación entre el proceso manual y el proceso automatizado para una tarea representativa del taller: la clasificación de noticias y la extracción de palabras clave a partir de fuentes RSS. En el esquema manual, cada paso requería intervención humana directa, desde la visita a los sitios de noticias hasta el registro en hoja de cálculo, con un tiempo estimado de 30 a 45 minutos y criterios de clasificación subjetivos que variaban entre personas. Con la implementación de n8n, el mismo proceso se ejecutó de manera autónoma mediante un flujo de trabajo que integra nodos de lectura de fuentes RSS, extracción de palabras clave a través de un modelo de lenguaje (Groq) y clasificación automática de las noticias en categorías (Economía o Política) mediante un nodo de clasificación de texto con IA. Los resultados fueron consistentes y reproducibles, registrados automáticamente en Google Sheets.

**Figura 2**

*Comparación del proceso de clasificación de noticias y extracción de palabras clave antes y después de la implementación de n8n*



Este contraste evidencia el potencial de la plataforma para transformar tareas manuales repetitivas en procesos automatizados con criterios homogéneos.

## 2.1.2 CONFIGURACIÓN

Después de la planificación, se procedió a establecer el entorno de acuerdo con lo diseñado. En primer lugar, se configuró el servidor VPS con los prerequisites básicos. Para instalaciones en equipos de cómputo locales, se ejecutaron los siguientes comandos de instalación de Docker Engine, containerd y Docker Compose, en donde se agregó primero el repositorio que permite la descarga del *software*; en el Anexo A del reporte, se presentan detalles adicionales sobre este tema.

Para servicios VPS, Docker generalmente viene preinstalado, por lo que únicamente se verificó que estuviera activo mediante el comando `docker --version` —si fuera necesario, se inicia el servicio con `sudo systemctl start docker`—. Después, se configuró un nombre de dominio (subdominio institucional), el cual apunta a la dirección IP del servidor, dado que la emisión de certificados TLS de Let's Encrypt requiere validar el dominio.

Con el dominio listo, se creó el archivo de entorno (.env) para almacenar las variables de configuración sensibles y específicas del despliegue. Entre ellas, se incluyeron: el subdominio (como parte de .unam.mx), las credenciales de base de datos (usuario, contraseña y nombre de base de datos PostgreSQL), la zona horaria, así como las credenciales para la autenticación básica de n8n, las cuales aplican únicamente durante el primer inicio del sistema antes de la creación de la cuenta de propietario. En este archivo, se incluyeron variables propias de n8n como la activación del modo de autenticación básica, el protocolo HTTPS y la URL base para *webhooks*. Estas configuraciones garantizaron que n8n reconociera su entorno de despliegue en producción y generara enlaces correctos para funciones especiales como URLs de *webhook* o autenticaciones externas.

Por último, se creó el directorio de proyecto y el archivo `docker-compose.yml` con la configuración de los tres servicios. Primero, se creó la estructura de directorios (en el Anexo B del reporte se detalla este punto), y luego se procedió con la configuración del archivo `docker-compose.yml`, en donde se incluyeron los tres servicios principales mencionados (en el Anexo C del reporte se presentan detalles adicionales acerca de este tema).

### 2.1.3 INSTALACIÓN

Una vez finalizado el entorno, se pasó a la instalación y despliegue propiamente dicho. Todos los servicios se iniciaron mediante Docker Compose con el siguiente comando:

```
docker compose up -d
```

Docker Compose descargó automáticamente las imágenes de n8n, PostgreSQL y Traefik desde el registro oficial de Docker Hub y se procedió a crear las instancias de los contenedores según la configuración definida. Gracias a la directiva *healthcheck*, el inicio fue secuencial y controlado: primero, PostgreSQL se puso en estado saludable después de comprobar repetidamente que el servicio estuviera listo para aceptar conexiones, lo que permitió al servicio n8n conectarse con la base de datos en funcionamiento sin problemas de conexión.

Por su parte, Traefik comprobó, mediante las etiquetas definidas en el servicio n8n, que debía enrutar el subdominio configurado al contenedor y, en ese momento, generó automáticamente una solicitud a Let's Encrypt para obtener el certificado SSL del dominio. Este proceso de emisión de certificado se completó satisfactoriamente en cuestión de segundos. Dichos certificados fueron almacenados en el volumen correspondiente, lo que permitió su reutilización y renovación automática antes del vencimiento, típicamente cada 90 días según las políticas de Let's Encrypt.

Una vez iniciados los tres contenedores, se comprobó el estado mediante comandos de inspección como `docker compose ps` y `docker compose logs`. Las bitácoras confirmaron que n8n se inició en modo producción, escuchando en el puerto interno configurado, y que Traefik expuso correctamente el enrutador con certificado válido. No se encontraron problemas en el inicio, mientras que PostgreSQL

confirmó la creación de la base de datos indicada y la aceptación de conexiones desde el contenedor de n8n.

### 2.1.4 PRUEBAS

Después de la instalación, se realizaron pruebas de funcionamiento y seguridad para comprobar que el despliegue se ajustaba a los requisitos establecidos. En primer lugar, se accedió a través de un navegador web al dominio configurado para comprobar la carga de la interfaz de n8n. El navegador reflejó el certificado TLS emitido por Let's Encrypt como válido, lo que indicaba que la comunicación se realizaba cifrada de forma correcta a través de HTTPS. Esta comprobación fue especialmente relevante dado que varios participantes en el taller accedieron a la plataforma desde redes externas a la universidad.

Al acceder por primera vez, n8n mostró una secuencia de formularios de configuración inicial que guiaron al administrador en el proceso de inicio. El primer formulario solicitó la creación de la cuenta de propietario, donde se requería dirección de correo electrónico, nombre y apellido, así como contraseña segura con un mínimo de ocho caracteres y, al menos, un número y una letra mayúscula. Este paso era necesario para crear al usuario administrador de la instancia.

El segundo formulario era una secuencia de preguntas de personalización de la instancia, diseñadas para adaptar la experiencia de usuario: tipo de organización, rol del usuario, finalidad principal de las automatizaciones, tamaño de la organización y cómo se conoció n8n. Aunque estas preguntas no eran necesarias para la funcionalidad, facilitaron la personalización de la interfaz.

El tercer paso proporcionó la posibilidad de desbloquear características gratuitas adicionales a través del registro de la instancia, como historial de ejecución extendido, herramientas de depuración avanzadas y estructura de carpetas para la organización de flujos de trabajo. Se introdujo una cuenta de correo institucional para obtener una licencia gratuita. Posteriormente, se verificó la bandeja de entrada del correo proporcionado para obtener el código de activación, denominado *Activation Key*, que se ingresó en la ruta *Settings* → *Usage and Plan* → *Enter Activation Key*; de esta manera, la instancia pasó a ser considerada registrada, la cual desbloqueó características avanzadas por tiempo ilimitado.

Una vez verificada la accesibilidad de la interfaz web, se procedió a probar la ejecución de flujos de trabajo básicos. Se creó un flujo sencillo con nodos de ejemplo —un disparador manual y un nodo que retornaba un mensaje de texto— para probar que el motor de ejecución funcionaba correctamente. El flujo se ejecutó sin problemas y los registros de n8n mostraron la secuencia de pasos completados con sus tiempos de ejecución.

También, se probó la integración con servicios externos a través de un nodo de solicitud HTTP que apunta a una API pública de prueba. La ejecución obtuvo la respuesta esperada, lo que indicó que n8n podía realizar solicitudes salientes a Internet sin restricciones del *firewall* o la configuración de red. Además, se probó una autenticación OAuth2 contra los servicios de Google para corroborar que n8n podía manejar el ciclo completo de autorización; la redirección de Google reconoció el dominio HTTPS como válido y permitió completar la conexión exitosamente.

Finalmente, se llevó a cabo una prueba de resistencia y estabilidad a largo plazo: se dejó correr el contenedor de n8n durante varias semanas con flujos programados mediante disparadores de tipo cron, incluyendo flujos que enviaban recordatorios por correo y flujos que consultaban APIs externas periódicamente. Se monitoreó el consumo de recursos mediante herramientas como *docker stats* y se

verificó la estabilidad del sistema. Los resultados fueron satisfactorios: n8n mantuvo un uso moderado de CPU y memoria acorde a lo esperado (menos del 10% de CPU en reposo y picos controlados durante ejecuciones); no se observaron caídas del servicio y la base de datos PostgreSQL mostró estabilidad en la inserción y consulta periódica de datos sin degradación del rendimiento.

## 2.2 MANTENIMIENTO Y ACTUALIZACIÓN

Como parte integral de la metodología, se implementó un procedimiento de mantenimiento continuo con énfasis en la actualización segura de la plataforma n8n a futuras versiones. El equipo de n8n, junto con la comunidad de desarrollo, publica actualizaciones frecuentes que incluyen correcciones de seguridad, nuevas funcionalidades e integraciones con servicios externos y mejoras de rendimiento. Debido a esto, mantener la plataforma actualizada resulta importante para aprovechar estas mejoras y proteger la infraestructura.

Con el fin de automatizar gran parte del proceso de actualización, tanto en ambiente local de pruebas como en producción, se desarrolló un *script* Bash denominado *update\_core.sh* con ayuda de IA generativa. Este *script* fue diseñado con un enfoque preventivo que prioriza la preservación de datos sobre la velocidad de actualización.

Entre las funcionalidades implementadas en el *script*, se encuentran las siguientes: (1) respaldo de base de datos, en donde se utiliza el volcado de PostgreSQL con el comando *pg\_dump* ejecutado dentro del contenedor que almacena el respaldo SQL compactado con fecha en un directorio específico para respaldos; (2) respaldo de volúmenes, en donde se almacena el contenido del volumen de datos de n8n en un archivo compactado; (3) detección automática de servicios definidos en el archivo *docker-compose.yml*, donde se incluyen servicios opcionales tales como bases de datos vectoriales (Qdrant) o servidores de modelos locales (Ollama) que podrían agregarse en un futuro; (4) actualización de imágenes Docker, sólo después de que se hayan realizado de manera exitosa los respaldos, donde se obtienen las nuevas versiones con comandos de Docker Compose; (5) verificación de salud post-actualización, donde se conecta internamente a los *endpoints* de cada servicio hasta obtener una respuesta satisfactoria, con reintentos y tiempos de espera configurables; y (6) registro detallado de operaciones en un archivo de log con sello de tiempo, que permite a los administradores revisar cualquier incidencia ocurrida durante la actualización.

Es importante destacar que, aunque el *script* automatiza la secuencia de respaldo, descarga y actualización, la toma de decisión para llevar a cabo la actualización es manual e informada. A través de las mejores prácticas de DevOps, documentadas en la literatura (Saini, 2025), primero se probó la nueva versión de n8n en un entorno de pruebas local o en una instancia secundaria para asegurarse de su estabilidad y compatibilidad con los flujos existentes. Sólo después de validar en este entorno de laboratorio que la versión funcionaba correctamente con los nodos y credenciales utilizados en producción, se procedió a ejecutar el *script* en el servidor de producción.

Esta precaución se originó después de que una experiencia durante el período de funcionamiento demostrara su necesidad: una versión particular de n8n presentó un error inesperado en la interfaz de credenciales, impidiendo llenar algunos campos necesarios para conectarse con APIs externas (Cristiangrando, 2025). Este *bug* afectó específicamente a la creación de nuevas credenciales, debido a que las existentes funcionaban con normalidad. Ese problema hubiera afectado gravemente a los flujos de trabajo del taller —donde los participantes necesitaban configurar nuevas conexiones

con servicios de Google Gemini y OpenRouter—, pero la política de prueba previa evitó una interrupción significativa. A raíz de esta experiencia, se consolidó la política de “actualización semiautomática supervisada”: el *script* asegura rapidez y consistencia en la actualización (junto con la posibilidad de reversión mediante los respaldos en caso de ser necesario), pero la activación de la actualización requiere ejecución humana tras pruebas satisfactorias en un entorno controlado. La adopción de estas prácticas DevOps responde a la creciente importancia de esta metodología, tanto en la industria como en contextos académicos donde se ha documentado que las instituciones de educación superior están incorporando gradualmente principios DevOps en sus infraestructuras tecnológicas. Persisten, sin embargo, brechas tecnológicas entre la formación académica y las demandas de la industria (Sánchez-Cifo *et al.*, 2023).

### 3. RESULTADOS

El despliegue de n8n en el servidor del IIS-UNAM se llevó a cabo de manera exitosa y la plataforma se puso en marcha como infraestructura de soporte para el “Taller Desarrollo de Aplicaciones con IA” de Educación Continua durante 2025. A continuación, se presentan los resultados obtenidos durante el período de funcionamiento.

#### 3.1 DISPONIBILIDAD Y ESTABILIDAD

Durante el período de funcionamiento (enero a diciembre de 2025), la plataforma logró una disponibilidad superior al 98%. Las interrupciones registradas fueron exclusivas a actualizaciones programadas del sistema operativo y de la plataforma n8n en sí misma, así como a un reinicio del servidor por mantenimiento de seguridad programado por la administración de la infraestructura universitaria. El *script* de actualización fue ejecutado en varias ocasiones a lo largo del año para aplicar actualizaciones de seguridad y funcionalidades nuevas de n8n, sin pérdida de datos ni flujos de trabajo a través del sistema de respaldos implementado. En todos los casos, el tiempo de inactividad no superó los 15 minutos por evento de mantenimiento.

#### 3.2 USO EN EL TALLER DE CAPACITACIÓN

La plataforma se utilizó como espacio de práctica para las 16 sesiones del taller, en los cuales los participantes (investigadores y técnicos académicos del área de ciencias sociales) aprendieron a configurar credenciales de API para servicios de inteligencia artificial (Google Gemini, OpenRouter), a diseñar flujos de trabajo que integren modelos de lenguaje con hojas de cálculo de Google, a implementar memoria en agentes conversacionales para mantener contexto entre interacciones y a automatizar tareas repetitivas de recolección y procesamiento de datos relevantes para sus proyectos de investigación.

Los participantes lograron diseñar y ejecutar flujos de trabajo funcionales sin necesitar conocimientos previos de programación, lo que validó la hipótesis de que las plataformas de bajo código pueden democratizar la automatización en ambientes académicos (Venkiteela, 2025). El taller se dividió en tres módulos temáticos complementarios: el primer módulo, enfocado en la automatización con n8n, se dividió en ocho sesiones, en las que los participantes aprendieron los conceptos básicos de flujos de trabajo (nodos, conexiones, disparadores y acciones) y se trató ampliamente con el concepto de JSON como formato de intercambio de datos fundamental para la comunicación entre servicios; el segundo

módulo se enfocó en herramientas de prototipado rápido que complementaron las capacidades de automatización; y el tercer módulo se enfocó en el uso de entornos de desarrollo asistidos por IA para casos que requirieron personalización más allá de las capacidades de bajo código.

Un hallazgo relevante fue la curva de aprendizaje diferenciada observada entre los participantes: aquellos sin conocimientos previos en programación lograron diseñar flujos funcionales a lo largo de las sesiones mediante la interfaz visual, mientras que los participantes con conocimientos básicos en programación pudieron diseñar automatizaciones más complejas en donde se aprovechó los nodos de código JavaScript disponibles en n8n. Este hallazgo sugiere que, efectivamente, n8n reduce la barrera de entrada para la automatización de procesos, aunque no la elimina completamente —los participantes aún tuvieron que comprender conceptos fundamentales como APIs, métodos de autenticación (claves de API vs. OAuth2) y estructuras de datos en formato JSON—. Este patrón de aprendizaje es consistente con investigaciones recientes sobre el desarrollo de habilidades metacognitivas en plataformas de bajo código en contextos educativos, donde se ha documentado que la interfaz visual facilita la comprensión de conceptos abstractos de programación sin eliminar la necesidad de comprender la lógica subyacente (Matook *et al.*, 2024).

### 3.3 INTEGRACIÓN CON SERVICIOS DE IA

Se verificó la correcta integración de n8n con los siguientes servicios de inteligencia artificial y productividad, todos ellos utilizados activamente durante las sesiones del taller (el Anexo A del reporte presenta detalles adicionales acerca de este punto).

La implementación de HTTPS probó ser fundamental para las integraciones OAuth2, lo que ratificó la estrategia de diseño de implementar Traefik con certificados SSL automáticos desde el comienzo del proyecto. De no tener el certificado SSL válido, las integraciones con servicios de Google no habrían sido posibles, lo que habría limitado severamente la utilidad de la plataforma para el taller.

### 3.4 COMPARACIÓN CON ALTERNATIVAS

El auto alojamiento de n8n probó tener ventajas sustanciales frente a las alternativas evaluadas inicialmente, las cuales se resumen en la tabla comparativa que se encuentra en el Anexo E.

Para el contexto académico del IIS-UNAM, el modelo auto alojado en servidor local resultó el más adecuado tanto por consideraciones presupuestarias —dado que el costo del VPS es significativamente menor que las suscripciones a servicios en la nube para el volumen de ejecuciones requerido— como por el requisito de mantener control sobre datos de investigación potencialmente sensibles, lo cual es consistente con las recomendaciones de la literatura sobre autonomía de datos en entornos académicos (Venkiteela, 2025).

### 3.5 ANÁLISIS DE RESULTADOS

Los resultados obtenidos confirman lo reportado en la literatura respecto a la viabilidad de n8n en entornos productivos con cargas de trabajo moderadas a intensas. La tasa de disponibilidad superior al 98% es consistente con los *benchmarks* publicados para despliegues similares en infraestructura de nivel producción. Asimismo, la experiencia del taller demostró empíricamente que investigadores sin formación técnica formal en programación pueden diseñar y ejecutar automatizaciones complejas en períodos de

capacitación relativamente cortos (16 sesiones de dos horas cada una), en relación con los hallazgos teóricos sobre el potencial de las plataformas de bajo código para democratizar la automatización y la integración de inteligencia artificial en flujos de trabajo de investigación (Venkiteela, 2025).

El caso del *bug* en la versión de *n8n* que afectaba la interfaz de credenciales (Cristiangrando, 2025) ratificó posteriormente la política de actualización supervisada que se implementó desde el inicio del proyecto. Si se hubiera actualizado de manera automática sin comprobar previamente en un entorno de pruebas, los asistentes al taller no hubieran podido establecer nuevas conexiones con APIs externas durante varias sesiones, lo que hubiera tenido un impacto significativo en la experiencia de aprendizaje y el cumplimiento de los objetivos pedagógicos del taller.

## 4. CONCLUSIONES

Con el uso del *n8n* en el Instituto de Investigaciones Sociales de la UNAM se pudo proveer una herramienta de automatización de flujos de trabajo con inteligencia artificial integrada para uso académico y formativo. Los resultados obtenidos durante el tiempo de funcionamiento demostraron la factibilidad técnica y pedagógica del enfoque utilizado.

La arquitectura, contenerizada con Docker, PostgreSQL como base de datos y Traefik como *proxy* inverso, resultó ser una combinación sólida que facilitó tanto el proceso de implementación inicial como el mantenimiento continuo. La implementación de certificados HTTPS automatizada con Let's Encrypt eliminó un obstáculo técnico significativo para la integración con servicios externos que requieren conexiones seguras, especialmente aquellos que utilizan OAuth2 como mecanismo de autenticación.

El *script* de actualización semiautomática, desarrollado con asistencia de IA generativa, es una herramienta útil para administradores de sistemas que manejan instancias de *n8n* en producción. El enfoque de respaldar antes de actualizar con verificación manual previa en ambiente de pruebas ofrece un equilibrio apropiado entre mantenerse actualizado con las últimas actualizaciones de correcciones de seguridad y las nuevas funcionalidades, al tiempo que se minimizan los riesgos de interrupción del servicio por incompatibilidades o errores introducidos en nuevas versiones.

La experiencia del taller ratificó que las plataformas de automatización de bajo código pueden funcionar como puentes formativos para acercar a investigadores de ciencias sociales a las tecnologías de inteligencia artificial que los potencia para diseñar soluciones a problemas específicos de su disciplina sin requerir conocimientos avanzados de programación. Esta democratización de la automatización tiene implicaciones importantes para la productividad de la investigación académica en el contexto actual de transformación digital.

Se recomienda a otras instituciones de educación superior interesadas en implementar soluciones similares considerar los siguientes aspectos derivados de la experiencia documentada: (a) planificar la arquitectura con énfasis en persistencia de datos y seguridad desde el diseño inicial, junto con el uso de volúmenes Docker dedicados y variables de entorno para información sensible; (b) implementar procedimientos de respaldo automatizado antes de cualquier actualización, se incluye tanto la base de datos como los volúmenes de configuración; (c) mantener un entorno de pruebas paralelo para validar nuevas versiones antes de aplicarlas en producción, especialmente para actualizaciones mayores; y (d)

documentar los procedimientos de manera que sean replicables por personal técnico con conocimientos básicos de administración de sistemas Linux y Docker.

Durante las pruebas, se identificó una limitación importante: el servidor VPS donde se alojó n8n no contaba con recursos suficientes para ejecutar modelos de lenguaje locales mediante Ollama. Incluso modelos relativamente pequeños, de 8 mil millones de parámetros, presentaron un rendimiento insuficiente para atender consultas simples o conversaciones básicas, lo cual descartó inicialmente la posibilidad de independizarse de APIs externas para ciertos casos de uso.

Como alternativa, se efectuaron pruebas exitosas con LM Studio en una Mac mini M4 Pro con 24 GB de memoria RAM unificada, se logró un desempeño aceptable de 68 *tokens* por segundo y 3.77 segundos para el primer *token* en consultas simples. Los proyectos futuros incluirán la implementación de una arquitectura que permitirá a n8n acceder a modelos locales ejecutados en este equipo como si fueran un servicio interno. La arquitectura permitirá mantener el principio de control institucional sobre los datos y las automatizaciones, lo que permitirá ejecutar modelos de lenguaje sin depender de servicios externos ni exponer la infraestructura de inferencia a Internet.

Desde una perspectiva más amplia, la experiencia documentada en este reporte contribuye al creciente cuerpo de conocimiento sobre la adopción de tecnologías de inteligencia artificial en instituciones de educación superior mexicanas. La combinación de infraestructura auto alojada con capacitación práctica representa un modelo que otras áreas universitarias podrían replicar y adaptar a sus necesidades específicas de investigación y docencia. La documentación detallada de los procedimientos técnicos, junto con el *script* de mantenimiento, busca facilitar esta transferencia de conocimiento y reducir la curva de aprendizaje para equipos técnicos que deseen implementar soluciones similares en sus propios contextos institucionales.

## AGRADECIMIENTOS

Se agradece al Instituto de Investigaciones Sociales de la UNAM por el apoyo institucional para el desarrollo del “Taller Desarrollo de Aplicaciones con IA” a través de Educación Continua, y a la Mtra. Sofía Aké Farfán, jefa del Departamento de Cómputo, por las facilidades otorgadas en infraestructura de cómputo. Asimismo, se reconoce la participación entusiasta de los asistentes al taller, cuyas preguntas y casos de uso enriquecieron la implementación documentada en este reporte.

### Declaración de contribución de autoría

**Ernesto Pathros Ibarra García:** Docente principal del taller; diseñador del currículo académico; especialista técnico en integración de servicios de IA. Realizó la automatización de procesos con inteligencia artificial; desarrollo de aplicaciones inteligentes; educación en tecnologías de información y comunicación. Lideró el diseño del módulo de automatización del taller “Desarrollo de Aplicaciones con IA” y definió los objetivos de aprendizaje y la secuencia didáctica de las 16 sesiones impartidas. Desarrolló con IA generativa el *script* de actualización semiautomática *update\_core.sh* documentado en este reporte, así como la implementación la lógica de respaldo de base de datos, respaldo de volúmenes y verificación de salud post-actualización. Contribuyó a las secciones del artículo sobre configuración de APIs de inteligencia artificial (Google Gemini, OpenRouter), mejores prácticas de seguridad en la plataforma n8n y análisis de la experiencia docente que motivó el reporte técnico.

**Julio César Cruz Estrada:** Docente del taller; especialista en infraestructura y despliegue; responsable técnico de instalación del servidor. Aportó en la infraestructura de tecnologías de información y comunicación; DevOps; administración de sistemas; despliegue en producción. Lideró la implementación de n8n en la infraestructura del servidor del IIS-UNAM y la ejecución del aprovisionamiento del VPS, la configuración de Docker y Docker Compose, así como el despliegue inicial de los contenedores. Elaboró la documentación detallada de los procedimientos de instalación tanto en modo local como en servidor de producción, que sirvió como base para este reporte técnico. Participó como docente en las sesiones del taller enfocadas a configuración de credenciales y resolución de problemas técnicos identificados por los participantes durante los ejercicios prácticos.

## REFERENCIAS

- Cristiangrando. (2025). *Visual bug when registering credentials or auth* (Issue #20279). GitHub. <https://github.com/n8n-io/n8n/issues/20279>
- Devi Priya, V. S., Sethuraman, S. C., & Khan, M. K. (2023). Container security: Precaution levels, mitigation strategies, and research perspectives. *Computers & Security*, 135, 103490. <https://doi.org/10.1016/j.cose.2023.103490>
- Fàbregas, F. (2022). *Development of Node-RED web-based, open-source visual workflow tool for bioinformatics* [Trabajo de fin de máster, Universitat Oberta de Catalunya]. Repositorio Institucional UOC. <https://openaccess.uoc.edu/handle/10609/146604>
- Hartz, E. L. (2025). *Development of an AI-driven DevOps Engineer: Automating workflows with an LLM based multi-agent system* [Tesis de maestría, Universidad Técnica de Múnich]. DiVA Portal. <https://www.diva-portal.org/smash/record.jsf?pid=diva2:1987584>
- Jennings, T. (2025). *Building strong container security for modern applications*. Mend.io. <https://www.mend.io/blog/building-strong-container-security/>
- Joshi, S. (2025). A review of generative AI and DevOps pipelines: CI/CD, agentic automation, MLOps integration, and LLMs. *International Journal of Innovative Research in Computer Science and Technology*, 13(4), 1-14. <https://doi.org/10.55524/ijrcst.2025.13.4.1>
- Korinek, A. (2023). Generative AI for economic research: Use cases and implications for economists. *Journal of Economic Literature*, 61(4), 1281-1317. <https://doi.org/10.1257/jel.20231736>
- Manning, B. S., Zhu, K., & Horton, J. J. (2024). *Automated social science: Language models as scientist and subjects* [Documento de trabajo]. National Bureau of Economic Research. <https://www.nber.org/papers/w32381>
- Matook, S., Maggie Wang, Y., Koeppel, N., & Guerin, S. (2024). Metacognitive skills in low-code app development: Work-integrated learning in information systems development. *Journal of Information Technology*, 39(1), 41-70. <https://doi.org/10.1177/02683962231170238>
- n8n-io. (2024). *Self-hosted AI starter kit*. GitHub. <https://github.com/n8n-io/self-hosted-ai-starter-kit>
- Nüst, D., Sochat, V., Marwick, B., Eglén, S. J., Head, T., Hirst, T., & Evans, B. D. (2020). Ten simple rules for writing Dockerfiles for reproducible data science. *PLOS Computational Biology*, 16(11), e1008316.

<https://doi.org/10.1371/journal.pcbi.1008316>

- Saini, A. (2025). Continuous Integration & Continuous Deployment (CI/CD) best practices to follow in 2026. *Kellton Tech Blog*. <https://www.kellton.com/kellton-tech-blog/continuous-integration-deployment-best-practices-2025>
- Sánchez-Cifo, M. Á., Bermejo, P., & Navarro, E. (2023). DevOps: Is there a gap between education and industry? *Journal of Software: Evolution and Process*, 35(12), e2534. <https://doi.org/10.1002/smr.2534>
- Traefik Labs. (2022). *HTTPS encryption: TLS, SSL, and Let's Encrypt*. Traefik Labs Glossary. <https://traefik.io/glossary/https-encryption-tls-ssl-letsencrypt>
- Venkiteela, P. (2025). n8n: An open-source workflow automation platform for enterprise integration and AI-driven orchestration. *International Journal of Computer Applications*, 187(63). <https://doi.org/10.5120/ijca2025926031>
- Ziems, C., Held, W., Shaikh, O., Chen, J., Zhang, Z., & Yang, D. (2024). Can large language models transform computational social science? *Computational Linguistics*, 50(1), 237-291. [https://doi.org/10.1162/coli\\_a\\_00502](https://doi.org/10.1162/coli_a_00502)

## ANEXO A. LISTADO DE COMANDOS BÁSICOS PARA LA INSTALACIÓN DE DOCKER EN EL SERVIDOR

# Agregar la llave GPG oficial de Docker:

```
sudo apt-get update
```

```
sudo apt-get install ca-certificates curl
```

```
sudo install -m 0755 -d /etc/apt/keyrings
```

```
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg \  
-o /etc/apt/keyrings/docker.asc
```

```
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

# Agregar el repositorio a las fuentes de Apt:

```
echo "deb [arch=$(dpkg --print-architecture) \  
signed-by=/etc/apt/keyrings/docker.asc] \  
https://download.docker.com/linux/ubuntu \  
$(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update
```

```
sudo apt-get update
```

```
sudo apt-get update
```

```
sudo apt-get update
```

# Instalar Docker:

```
sudo apt-get install docker-ce docker-ce-cli containerd.io \  
docker-buildx-plugin docker compose-plugin
```

```
docker-buildx-plugin docker compose-plugin
```

# Iniciar y habilitar el servicio:

```
sudo systemctl start docker
```

```
sudo systemctl enable docker
```

# Agregar usuario al grupo docker:

```
sudo groupadd docker
```

```
sudo usermod -aG docker usuario_admin
```

```
newgrp docker
```

## ANEXO B. LISTADO DE COMANDOS PARA LA CREACIÓN DE DIRECTORIOS

```
mkdir ~/n8n && cd ~/n8n  
touch docker-compose.yml
```

## ANEXO C. CONFIGURACIÓN DEL ARCHIVO DOCKER-COMPOSE.YML

```
version: '3.8'  
services:  
  traefik:  
    image: traefik:v2.9  
    restart: unless-stopped  
    command:  
      --api=true  
      --api.insecure=true  
      --providers.docker=true  
      --providers.docker.exposedbydefault=false  
      --entrypoints.websecure.address=:443  
      --certificatesresolvers.mytlschallenge.acme.tlschallenge=true  
      --certificatesresolvers.mytlschallenge.acme.email=${EMAIL}  
      --certificatesresolvers.mytlschallenge.acme.storage=/letsencrypt/acme.json  
    ports:  
      - 443:443  
      - 80:80  
    volumes:  
      - traefik_data:/letsencrypt  
      - /var/run/docker.sock:/var/run/docker.sock:ro  
n8n:  
  image: n8nio/n8n:latest
```

restart: unless-stopped

environment:

- GENERIC\_TIMEZONE=\${GENERIC\_TIMEZONE}
- TZ=\${GENERIC\_TIMEZONE}
- N8N\_HOST=\${SUBDOMAIN}.\${DOMAIN\_NAME}
- N8N\_PORT=443
- N8N\_PROTOCOL=https
- NODE\_ENV=production
- WEBHOOK\_URL=https://\${SUBDOMAIN}.\${DOMAIN\_NAME}/
- N8N\_BASIC\_AUTH\_ACTIVE=\${N8N\_BASIC\_AUTH\_ACTIVE}
- N8N\_BASIC\_AUTH\_USER=\${N8N\_BASIC\_AUTH\_USER}
- N8N\_BASIC\_AUTH\_PASSWORD=\${N8N\_BASIC\_AUTH\_PASSWORD}
- N8N\_DATA\_TABLES\_MAX\_SIZE\_BYTES=\${N8N\_DATA\_TABLES\_MAX\_SIZE\_BYTES}
- DB\_TYPE=\${DB\_TYPE}
- DB\_POSTGRESDB\_DATABASE=\${DB\_POSTGRESDB\_DATABASE}
- DB\_POSTGRESDB\_HOST=\${DB\_POSTGRESDB\_HOST}
- DB\_POSTGRESDB\_PORT=\${DB\_POSTGRESDB\_PORT}
- DB\_POSTGRESDB\_USER=\${DB\_POSTGRESDB\_USER}
- DB\_POSTGRESDB\_PASSWORD=\${DB\_POSTGRESDB\_PASSWORD}

volumes:

- n8n\_data:/home/node/n8n
- ./local-files:/files

labels:

- traefik.enable=true
- traefik.http.routers.n8n.rule=Host(`\${SUBDOMAIN}.\${DOMAIN\_NAME}`)
- traefik.http.routers.n8n.tls=true
- traefik.http.routers.n8n.entrypoints=websecure
- traefik.http.routers.n8n.tls.certresolver=mytlschallenge

depends\_on:

- db

- traefik

db:

image: postgres:14

restart: unless-stopped

environment:

- POSTGRES\_USER=\${DB\_POSTGRESDB\_USER}

- POSTGRES\_PASSWORD=\${DB\_POSTGRESDB\_PASSWORD}

- POSTGRES\_DB=\${DB\_POSTGRESDB\_DATABASE}

volumes:

- db\_data:/var/lib/postgresql/data

healthcheck:

test: ["CMD-SHELL", "pg\_isready -U \${DB\_POSTGRESDB\_USER} -d \${DB\_POSTGRESDB\_DATABASE}"]

interval: 10s

timeout: 5s

retries: 5

volumes:

traefik\_data:

n8n\_data:

db\_data:

## ANEXO D. INTEGRACIÓN CON SERVICIOS DE IA

**Tabla 1**

*Integración con servicios de IA*

Servicio	Tipo de integración	Funcionalidad verificada
Google Gemini	API REST con OAuth2 y clave API	Generación de texto, análisis de documentos, transcripción de audios, respuestas a preguntas.
OpenRouter	API REST con clave de API	Acceso a múltiples modelos de lenguaje de diferentes proveedores.

Servicio	Tipo de integración	Funcionalidad verificada
Google Sheets	OAuth2 y clave API	Lectura y escritura de datos tabulares, actualización de celdas.
Webhooks	HTTP/HTTPS	Disparadores automáticos de flujos desde aplicaciones externas
Gmail	OAuth2 y clave API	Envío automatizado de correos electrónicos con contenido dinámico

## ANEXO E. COMPARATIVA DE ALTERNATIVAS

**Tabla 2**

*Comparativa de alternativas*

Criterio	n8n en VPS	n8n Cloud	Zapier/Make
Costo mensual	~\$20 USD	\$50+ USD	\$20-100+ USD
Control de datos	Total	Parcial	Limitado
Personalización	Alta	Media	Baja
Integración con IA local	Posible	No disponible	No disponible
Ejecuciones al mes	Ilimitadas	Limitadas por plan	Limitadas por plan
Soporte técnico	Comunidad	Incluido	Incluido

# Modelo lógico para la transferencia de líquidos en entornos tridimensionales web e inmersivos

## *Logical Model for Liquid Transfer in Web and Immersive Three-Dimensional Environments*

### Información del reporte:

Licencia Creative Commons



El contenido de los textos es responsabilidad de los autores y no refleja forzosamente el punto de vista de los dictaminadores, o de los miembros del Comité Editorial, o la postura del editor y la editorial de la publicación.

Para citar este reporte técnico:

Cruz Lovera, T. M. (2026). Modelo lógico para la transferencia de líquidos en entornos tridimensionales web e inmersivos. *Cuadernos Técnicos Universitarios de la DGTIC*, 4(2), páginas (29 - 48). <https://doi.org/10.22201/dgtic.30618096e.2026.4.2.169>

### Tayde Martín Cruz Lovera

Dirección General de Cómputo y de  
Tecnologías de Información y Comunicación  
Universidad Nacional Autónoma de México

[taydevr@comunidad.unam.mx](mailto:taydevr@comunidad.unam.mx)

ORCID:0009-0003-9519-8805

### Resumen

Este trabajo presenta el diseño y la implementación de un modelo lógico para la transferencia de líquidos entre objetos virtuales dentro de una aplicación interactiva tridimensional. En lugar de utilizar simulación continua de fluidos físicos, el enfoque propuesto representa el comportamiento del líquido mediante estados lógicos y reglas de transferencia. Esto permite controlar tanto el volumen total como su composición, reduciendo además la complejidad computacional del sistema.

El modelo se basa en la identificación de las entidades clave involucradas en el proceso (recipientes, instrumentos de laboratorio y líquido) y está estructurado siguiendo una arquitectura modular que separa la lógica del sistema, la representación visual y los mecanismos de interacción. Esta separación permitió trabajar los componentes de forma independiente y reutilizar el modelo en distintos contextos de interacción. La representación visual del líquido refleja el volumen y la composición mediante cambios en el nivel y apariencia dentro de los recipientes virtuales usando *shaders*.

El modelo fue diseñado, implementado y evaluado mediante pruebas funcionales en una aplicación desarrollada en Unity y se aplicó a un instrumento de laboratorio virtual tridimensional, concretamente una micropipeta. La solución se logró implementar tanto en una aplicación web como en una aplicación de realidad

virtual, lo que demostró su adaptabilidad a diferentes plataformas y manteniendo un comportamiento lógico consistente.

**Palabras clave:** Abstracción funcional, aplicación interactiva, laboratorios virtuales, realidad virtual, Unity.

### Abstract

*This work presents the design and implementation of a logical interaction model for liquid transfer between virtual objects within a three-dimensional interactive application. Instead of relying on continuous physical fluid simulation, the proposed approach represents liquid behavior through logical states and transfer rules. This enables control over both total volume and composition while reducing the computational complexity of the system.*

*The model is based on the identification of the key entities involved in the process (containers, instruments, and liquid) and is structured following a layered architecture that separates system logic, visual representation, and interaction mechanisms. This separation allowed components to be developed independently and enabled the reuse of the model in different interaction contexts. The visual representation of the liquid is derived from its logical state, showing volume and composition through changes in level and appearance within the virtual containers using shaders.*

*The model was implemented in an application developed with Unity and applied to a three-dimensional virtual instrument, specifically a micropipette, allowing its behavior to be evaluated in scenarios requiring precise and repeated liquid transfers. The solution was deployed both as a web application and as a virtual reality application, demonstrating its adaptability across platforms while maintaining consistent logical behavior*

**Keywords:** Functional abstraction, interactive application, Unity, virtual laboratories, virtual reality.

## 1. INTRODUCCIÓN

A finales de 2021, la Facultad de Odontología de la UNAM se acercó al Observatorio Ixtli (ahora LAD UNAM) en busca de una alternativa basada en realidad virtual para llevar a cabo prácticas de laboratorio. Esta iniciativa surgió debido a que, durante la pandemia, las clases se impartían mediante videos, lo que impedía replicar los procesos en casa.

A partir de ese acercamiento inicial, se realizó una colaboración donde se desarrollaron interactivos tridimensionales correspondientes a dos prácticas de laboratorio, cuya primera implementación en versión web se realizó entre 2023 y 2024. Posteriormente, durante 2025 y 2026, se llevó a cabo la adaptación e implementación de las versiones en realidad virtual, etapa en la que se centra el presente reporte técnico.

En los últimos años, las aplicaciones interactivas tridimensionales se han consolidado como una herramienta relevante en contextos educativos, particularmente a partir del uso de entornos de realidad virtual aplicados a la enseñanza y el entrenamiento, los cuales han mostrado un amplio potencial en múltiples dominios educativos (Radianti *et al.*, 2020). Los laboratorios virtuales representan un caso particular de estas aplicaciones, ya que permiten la simulación de procedimientos experimentales sin los

riesgos y costos asociados al entorno físico real. En estas aplicaciones, la interacción debe ser coherente y comprensible, reflejando el comportamiento esperado de los instrumentos y materiales representados.

Uno de los procesos más comunes en entornos de laboratorio es la transferencia de líquidos entre recipientes, ya sea mediante instrumentos especializados o de forma directa. La representación de la transferencia de líquidos en entornos interactivos no requiere necesariamente una simulación física completa del comportamiento del fluido. A pesar de que existen técnicas avanzadas para simular dinámicas complejas de fluidos, estas implican un alto costo computacional derivado del cálculo numérico continuo de variables físicas, innecesario para la interacción planteada en este trabajo.

El objetivo del presente reporte técnico, en el contexto del desarrollo de una aplicación interactiva tridimensional, es proponer un modelo lógico de interacción desacoplado de los mecanismos de entrada que permita la transferencia de líquidos entre objetos virtuales en entornos web y de realidad virtual, de modo que pueda integrarse con distintos sistemas sin modificar su lógica interna.

## 2. DESARROLLO TÉCNICO

El desarrollo técnico de este trabajo describe la construcción y el uso de un modelo lógico para la transferencia de líquidos entre objetos virtuales dentro de una aplicación interactiva tridimensional. A partir de la identificación de los elementos involucrados en el proceso, recipientes, instrumentos y líquido, se propuso una solución basada en una abstracción lógica que sustituye el uso de simulación física.

En entornos interactivos en tiempo real, como aplicaciones web y de realidad virtual autónomas, la simulación física de fluidos presenta desafíos significativos de rendimiento y complejidad. Estudios recientes sobre modelado de fluidos en contextos inmersivos muestran que incluso las soluciones optimizadas requieren técnicas especializadas para mantener tasas de actualización estables, lo que limita su viabilidad en aplicaciones interactivas generales (Cen *et al.*, 2024).

Con base en estas consideraciones, el modelo se definió como una alternativa para regular la transferencia de líquidos entre objetos virtuales, evitando el uso de simulaciones físicas, mediante una abstracción lógica que permite controlar el proceso, así como la cantidad y composición de líquido en cada objeto virtual.

La literatura reciente en ingeniería de *software* subraya la importancia de la abstracción como estrategia para abordar la complejidad propia de los sistemas, lo que permite centrar el diseño en el comportamiento relevante del sistema sin incorporar detalles innecesarios (Bencomo *et al.*, 2024).

### 2.1 METODOLOGÍA

Como parte del proceso de desarrollo, se identificaron las siguientes necesidades y requerimientos del sistema:

- Representar la transferencia de líquidos sin recurrir a simulaciones físicas, dado que no se requiere un comportamiento realista de fluidos.
- Controlar el volumen total y la composición del líquido durante la transferencia.

- Permitir transferencias consecutivas.
- Poder configurar o variar el volumen a transferir.
- Mantener consistencia lógica entre plataformas (web y realidad virtual).

A partir de estos requerimientos, se definió el modelo lógico propuesto, desarrollado mediante el siguiente proceso metodológico. Este se dividió en cuatro etapas principales:

1. Análisis conceptual: identificación de las entidades involucradas (recipientes, instrumentos y líquido), así como de sus relaciones dentro del proceso de transferencia.
2. Diseño del modelo: definición de los estados, reglas de transferencia y estructura de datos que conforman el modelo lógico.
3. Implementación: integración del modelo en una aplicación desarrollada en Unity.
4. Pruebas: ejecución de escenarios controlados para validar su comportamiento mediante pruebas funcionales.

## 2.2 MODELO LÓGICO PARA TRANSFERENCIA DE LÍQUIDOS

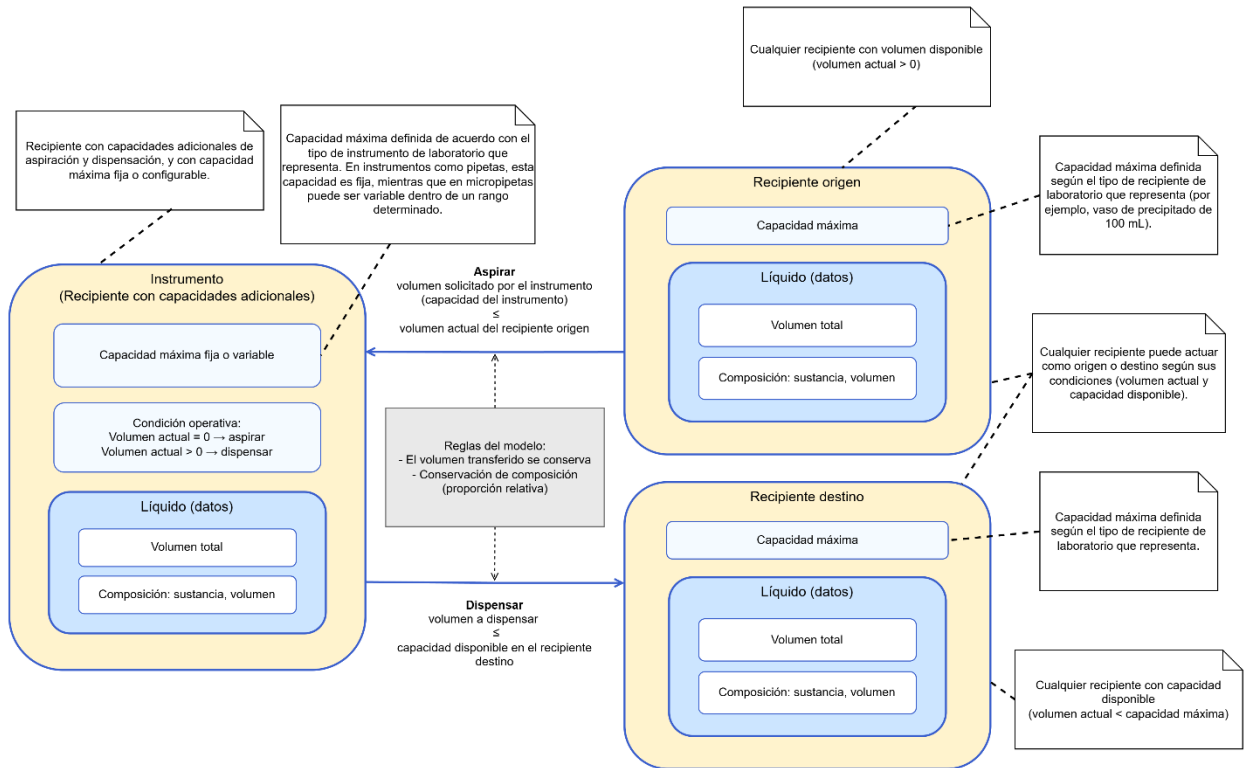
Para el diseño del modelo, se llevó a cabo un análisis conceptual de los elementos involucrados en el proceso de transferencia de líquidos, identificando como componentes principales a los recipientes, los instrumentos y el líquido contenido en ambos. Desde estos elementos, se definieron las relaciones y reglas que estructuran el modelo:

- Recipientes: elementos que actúan como contenedores capaces de almacenar y proporcionar líquido; cada uno tiene parámetros que indican su capacidad máxima y la cantidad actual de líquido, lo que determina el volumen que puede recibir o transferir.
- Instrumentos: elementos que funcionan como intermediarios en la transferencia de líquido entre recipientes, al trasladarlo de un recipiente a otro; en el modelo, se consideran como un tipo de recipiente con capacidades adicionales que permiten la aspiración y dispensación controlada del líquido.
- Líquido: estructura de datos que describe la composición del contenido en términos de volumen total y proporción relativa de sus componentes, la cual es manipulada durante el proceso de transferencia.

La relación lógica entre recipientes, instrumentos y líquido se representa en el diagrama de la Figura 1.

**Figura 1**

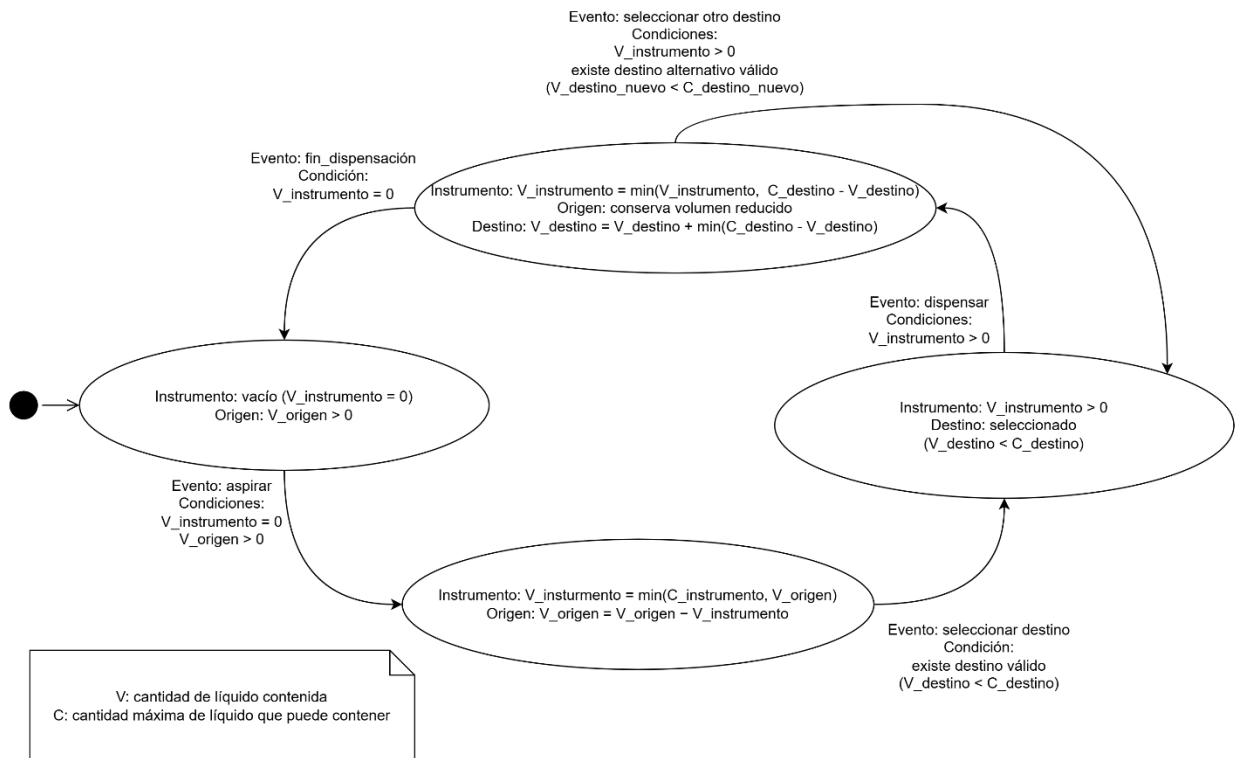
*Modelo lógico de interacción para la transferencia de líquidos*



El proceso de transferencia puede resumirse esquemáticamente en la Figura 2.

**Figura 2**

*Secuencia típica de estados del modelo*



Se muestra cómo cambia el estado del sistema a lo largo de una secuencia típica.

## 2.3 INTEGRACIÓN DEL MODELO EN UNA APLICACIÓN INTERACTIVA TRIDIMENSIONAL

El modelo fue implementado en una aplicación desarrollada en Unity, siguiendo un principio de separación en módulos con el objetivo de facilitar su adaptación a distintas plataformas. Diversos trabajos han señalado que el diseño modular es una estrategia eficaz para gestionar la complejidad de sistemas de *software*, permitiendo que distintos componentes evolucionen de manera independiente (Guntakandla, 2025).

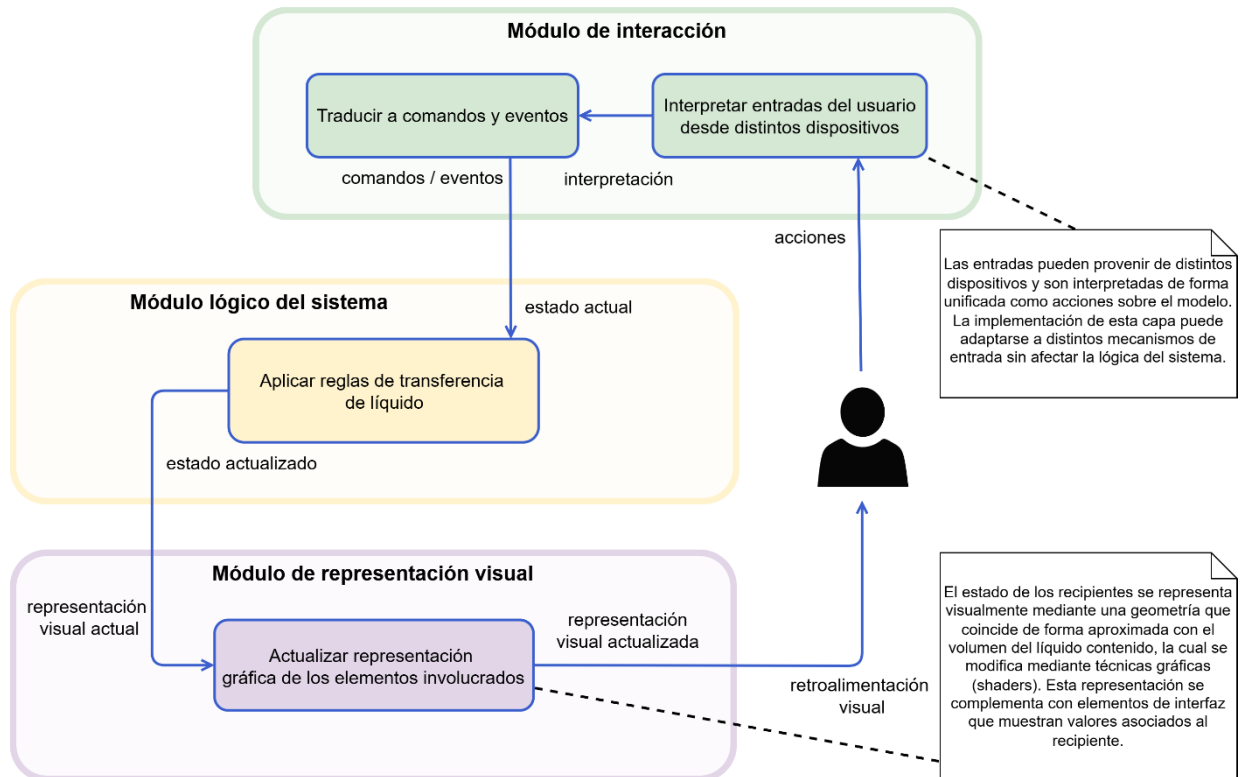
Este diseño distingue tres módulos principales: la lógica del sistema, la representación visual y el mecanismo de interacción. La Figura 3 muestra estos módulos y su relación:

- Módulo lógico del sistema: responsable de gestionar el estado y las reglas que rigen la transferencia de líquidos.
- Módulo de representación visual: encargado de mostrar visualmente el estado actual de los elementos del sistema.

- Módulo de interacción: interpreta las acciones del usuario y las traduce en operaciones sobre el modelo lógico.

**Figura 3**

*Diseño modular de la aplicación interactiva tridimensional*



Las acciones del usuario activan el instrumento virtual por medio del módulo de interacción, el cual se comunica con el módulo lógico para controlar el estado de los recipientes involucrados y el proceso de transferencia de líquido. Finalmente, el módulo lógico se comunica con el módulo de representación visual para actualizar gráficamente el estado de los recipientes.

En el módulo lógico, el comportamiento se define a partir de la representación y manipulación del estado del líquido en cada recipiente, el cual se describe en términos de volumen total y proporción relativa de sus componentes. Esta representación se implementa en los *scripts* LiquidDefinitions.cs y Liquid.cs, mostrados en las Figuras 4 y 5. Los fragmentos de código presentados corresponden a una versión simplificada y funcional de la implementación, utilizada con fines ilustrativos.

## Figura 4

*Definición de la estructura de datos para la composición del líquido (LiquidDefinitions.cs)*

```

01. // Archivo: LiquidDefinitions.cs
02.
03. using System;
04.
05. // Define la estructura que representa un componente dentro de un líquido
06. [Serializable]
07. public struct LiquidData
08. {
09.     public LiquidType type; // Tipo de sustancia
10.     public float amount; // Cantidad en mL
11.
12.     // Constructor del componente
13.     public LiquidData(LiquidType type, float amount)
14.     {
15.         this.type = type;
16.         this.amount = amount;
17.     }
18. }
19.
20. // Enum que define los tipos posibles de sustancias
21. public enum LiquidType
22. {
23.     Water,
24.     Alcohol,
25.     ReagentA,
26.     ReagentB
27. }

```

## Figura 5

*Implementación del recipiente y estado del líquido (Liquid.cs)*

```

01. // Archivo: Liquid.cs
02.
03. using System.Collections.Generic;
04. using UnityEngine;
05.
06. // Representa un recipiente o volumen de líquido
07. public class Liquid : MonoBehaviour
08. {
09.     [Header("Capacidad")]
10.
11.     [SerializeField] protected float capacity = 10f; // Capacidad máxima del recipiente
12.     [SerializeField] protected float currentVolume = 0f; // Volumen actual contenido
13.
14.     // Getter de la capacidad máxima
15.     public float GetCapacity() { return capacity; }
16.
17.     // Getter del volumen actual
18.     public float GetCurrentVolume() { return currentVolume; }
19.
20.     // Modifica volumen (uso interno controlado)
21.     public void ModifyVolume(float delta)
22.     {
23.         currentVolume = Mathf.Clamp(currentVolume + delta, 0f, capacity);
24.     }
25.
26.     // Setter protegido del volumen (limitado entre 0 y la capacidad)
27.     protected void SetCurrentVolume(float value)
28.     {
29.         currentVolume = Mathf.Clamp(value, 0f, capacity);
30.     }
31.
32.     [Header("Composición")]
33.
34.     [SerializeField] protected List<LiquidData> composition = new List<LiquidData>(); // Componentes del líquido
35.
36.     // Getter de la composición (uso para visualización o depuración)
37.     public List<LiquidData> GetComposition() { return composition; }
38. }

```

La transferencia de líquido entre recipientes se realiza mediante un cálculo proporcional que preserva la composición del líquido durante el proceso. Esta se implementa en el *script* Instrument.cs, mostrado en la Figura 6.

**Figura 6**

*Implementación de la transferencia de líquido mediante el instrumento (Instrument.cs)*

```

91. // Archivo: Instrument.cs
92.
93. using UnityEngine;
94.
95. // Instrumento que transfiere líquido entre recipientes
96. public class Instrument : ISquad
97. {
98.     [Header("Transferencia")]
99.
100.    // Indica si el instrumento permite configurar un volumen de transferencia menor a su capacidad.
101.    [SerializeField] private bool allowVariableTransferVolume = true;
102.
103.    // Volumen configurado para elegir o suspender cuando la transferencia variable está habilitada.
104.    [SerializeField] private float configuredVolume = 5f; // Volumen configurado
105.
106.    // Setter de la bandera que indica si el instrumento permite volúmenes de transferencia variable.
107.    public bool AllowVariableTransferVolume() { return allowVariableTransferVolume; }
108.
109.    // Setter de la bandera que habilita o deshabilita el uso de un volumen de transferencia configurable.
110.    public void SetAllowVariableTransferVolume(bool value)
111.    {
112.        allowVariableTransferVolume = value;
113.    }
114.
115.    // Setter del volumen configurado
116.    public float GetConfigureVolume() { return configuredVolume; }
117.
118.    // Setter del volumen configurado (limitado por el máximo)
119.    public void SetConfigureVolume(float value)
120.    {
121.        configuredVolume = Mathf.Clamp(value, 0f, capacity);
122.    }
123.
124.    // Setter del volumen efectivo de transferencia según la configuración del instrumento.
125.    public float GetTransferVolume()
126.    {
127.        return allowVariableTransferVolume ? Mathf.Clamp(configuredVolume, 0f, capacity) : capacity;
128.    }
129.
130.    // Setter del volumen contenido en el instrumento
131.    public float GetInstrumentVolume() { return currentVolume; }
132.
133.    // Setter del volumen interno del instrumento (limitado por su capacidad)
134.    public void SetInstrumentVolume(float value)
135.    {
136.        currentVolume = Mathf.Clamp(value, 0f, capacity);
137.    }
138.
139.
140.    // Transfiere volumen entre source y target de forma proporcional
141.    public void Transfer(float volume, Liquid source, Liquid target)
142.    {
143.        if (volume == 0f || source.GetCurrentVolume() == 0f) return;
144.
145.        volume = Mathf.Min(volume, source.GetCurrentVolume());
146.        volume = Mathf.Min(volume, target.GetCapacity() - target.GetCurrentVolume());
147.
148.        var sourceComp = source.GetComposition();
149.        var targetComp = target.GetComposition();
150.
151.        float sourceVolume = source.GetCurrentVolume();
152.
153.        for (int i = 0; i < sourceComp.Count; i++)
154.        {
155.            var data = sourceComp[i];
156.
157.            float fraction = data.amount / sourceVolume;
158.            float amount = volume * fraction;
159.
160.            // Restar del source
161.            data.amount -= amount;
162.            sourceComp[i] = data;
163.
164.            // Agregar al target
165.            bool found = false;
166.            for (int j = 0; j < targetComp.Count; j++)
167.            {
168.                if (targetComp[j].type == data.type)
169.                {
170.                    var t = targetComp[j];
171.                    t.amount += amount;
172.                    targetComp[j] = t;
173.                    found = true;
174.                    break;
175.                }
176.            }
177.
178.            if (!found)
179.                targetComp.Add(new LiquidData(data.type, amount));
180.        }
181.
182.        // Actualizar volúmenes usando método controlado
183.        source.ModifyVolume(-volume);
184.        target.ModifyVolume(volume);
185.    }
186.
187.    // Aspira líquido desde un recipiente al instrumento
188.    public void Aspirate(Liquid source)
189.    {
190.        Transfer(GetTransferVolume(), source, this);
191.    }
192.
193.    // Dispensa líquido desde el instrumento a un recipiente
194.    public void Dispense(Liquid target)
195.    {
196.        Transfer(GetTransferVolume(), this, target);
197.    }
198. }

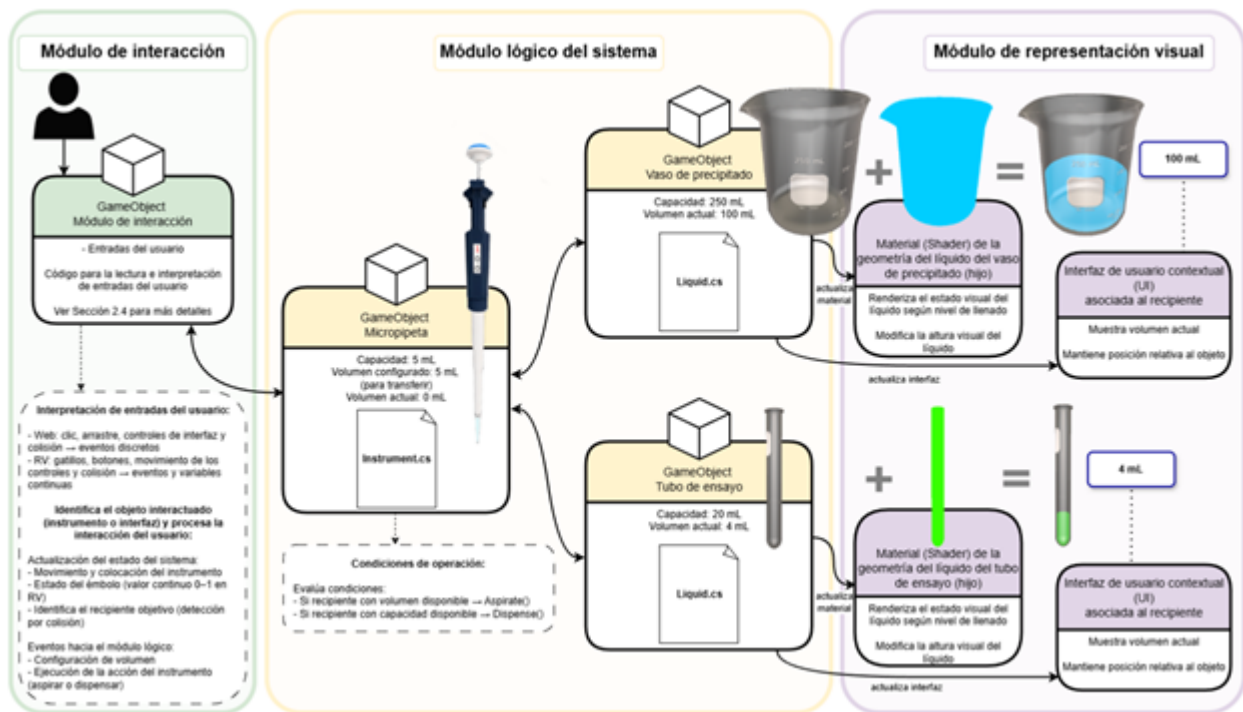
```

El modelo se implementó sobre objetos virtuales tridimensionales, en los que los instrumentos actúan como recipientes con capacidades adicionales para la transferencia de líquido, mientras que los recipientes actúan como origen o destino según la operación.

En la aplicación implementada, el instrumento usado es una micropipeta, la cual permite controlar con precisión el volumen transferido en escenarios que requieren múltiples transferencias entre distintos recipientes, con parámetros específicos de volumen inicial y capacidad máxima. La Figura 7 resume la implementación del modelo en Unity.

**Figura 7**

*Implementación del modelo en Unity: relación entre módulos, objetos de la escena y scripts*



*Nota.* Modelos 3D de micropipeta, vaso de precipitado y tubo de ensayo elaborados por Víctor Hugo Franco Serrano.

La representación visual del líquido se realiza mediante un *shader* que ajusta dinámicamente el nivel y color en función de la composición y volumen del líquido en el recipiente (véase Anexo A).

## 2.4 ADAPTACIÓN DEL MODELO A ENTORNOS WEB Y DE REALIDAD VIRTUAL

El modelo lógico de transferencia de líquidos se diseñó de forma independiente de la plataforma, permitiendo su integración con distintos dispositivos de entrada mediante una abstracción de las interacciones. Esta separación permite mapear diversas formas de interacción física a un mismo conjunto de operaciones lógicas.

En ambas plataformas, los elementos interactivos se representan mediante objetos 3D (instrumentos y recipientes), complementados con interfaces de usuario contextuales que muestran estados como volumen configurado y volumen actual.

### 2.4.1 WEB

En el entorno web, por su mayor alcance y accesibilidad, la aplicación se orienta al uso de dispositivos de entrada convencionales (ratón o pantalla táctil). En este contexto, las interacciones se basan en la interpretación de eventos de interfaz de usuario (clic, arrastre y controles de interfaz, como botones y controles deslizantes), que se traducen en eventos discretos del sistema.

### 2.4.2 REALIDAD VIRTUAL

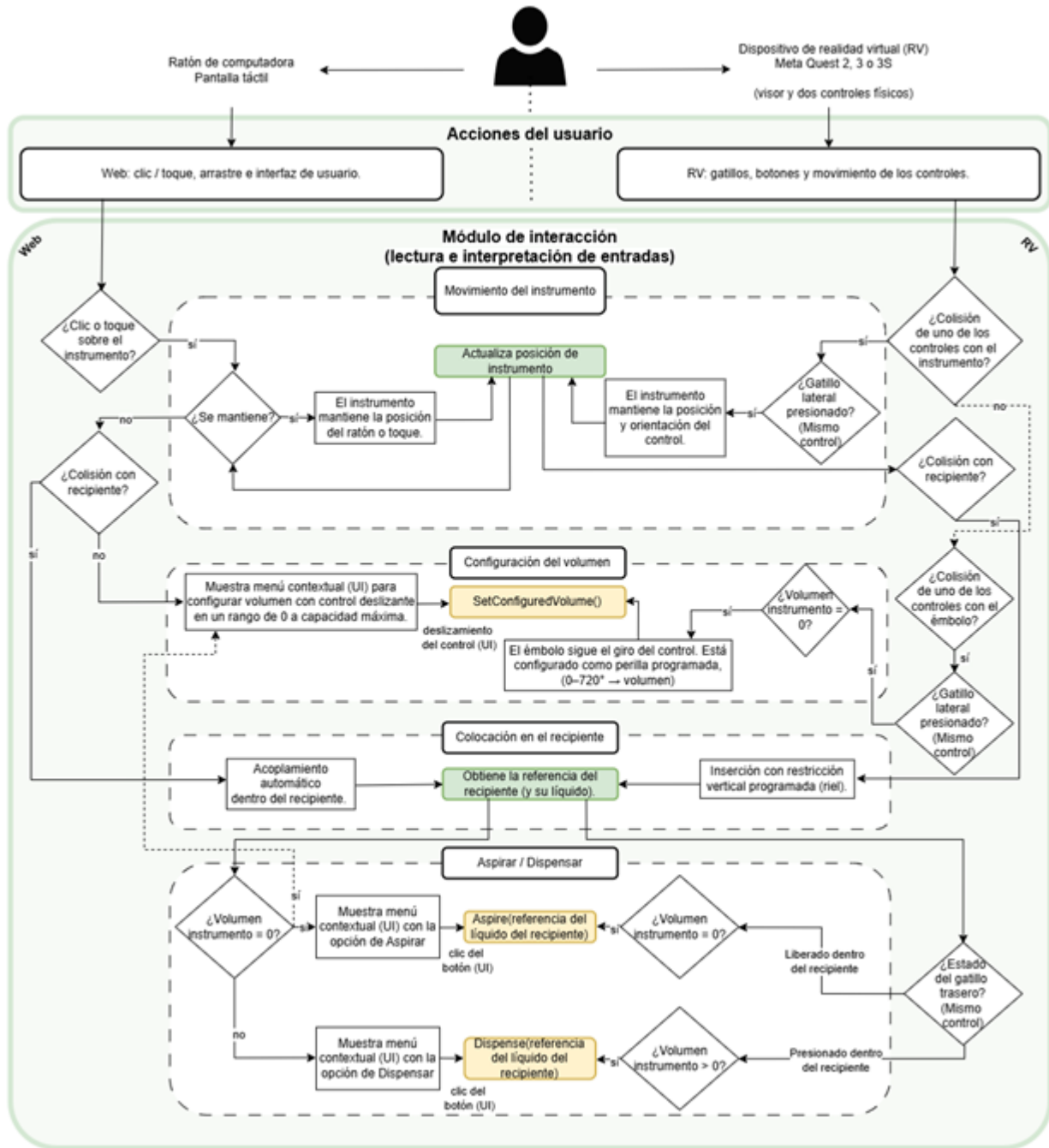
En realidad virtual, se implementaron interacciones mediante los controles físicos del dispositivo (Meta Quest 2, 3 o 3S), de forma que la manipulación del instrumento se asemeje a su uso en la vida real. Este tipo de interacción en realidad virtual, basada en acciones funcionales, permite que el usuario comprenda y anticipe el comportamiento del objeto virtual, favoreciendo una interacción coherente y significativa, aun cuando ésta se realice mediante dispositivos intermediarios como controles de realidad virtual (Fusaro *et al.*, 2025).

### 2.4.3 MODELO GENERAL DE INTERACCIÓN

Las acciones del usuario se traducen en llamadas a la misma lógica de actualización del estado del sistema sin importar la plataforma usada, como se muestra en la Figura 8.

**Figura 8**

*Traducción de interacciones del usuario al modelo lógico de la micropipeta*



### 3. RESULTADOS

El comportamiento del modelo lógico de interacción para la transferencia de líquidos fue evaluado mediante pruebas funcionales basadas en escenarios controlados dentro de la aplicación. Las condiciones de validación consistieron en la ejecución de secuencias de transferencia entre recipientes con diferentes configuraciones de volumen inicial, capacidad máxima y volumen de transferencia, incluyendo transferencias consecutivas. También se evaluaron casos límite como aspiración de volúmenes mayores al disponible, dispensación en recipientes con capacidad insuficiente, aspiración desde recipientes vacíos y dispensación en recipientes llenos.

Los criterios de validación considerados fueron:

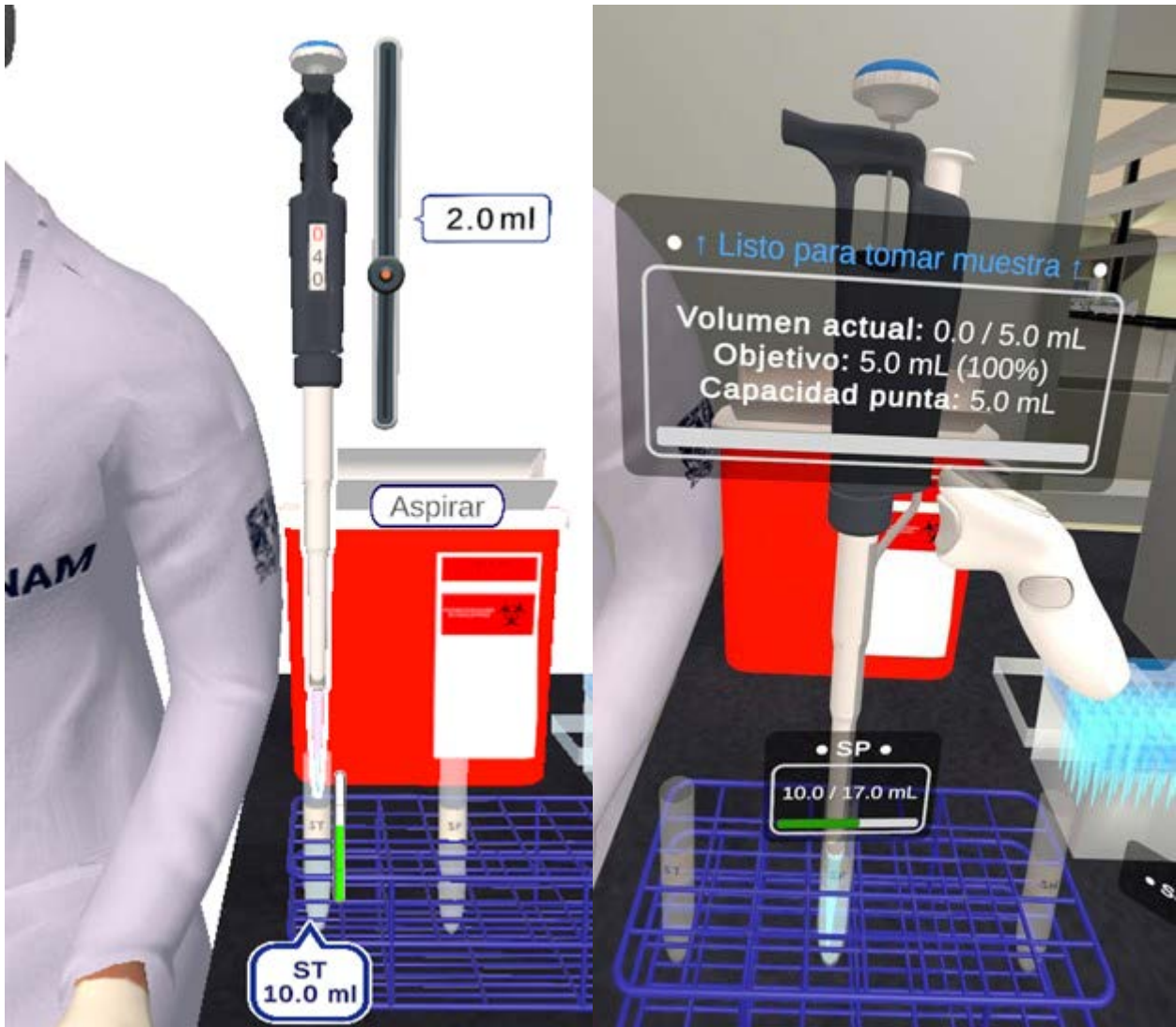
- Conservación del volumen total tras cada transferencia.
- Preservación de la proporción de componentes.
- Manejo correcto de casos límite (desbordamiento y volumen insuficiente).
- Consistencia del comportamiento en transferencias consecutivas.
- Consistencia del comportamiento entre plataformas.

El modelo cumplió con estos criterios en todos los escenarios evaluados, manteniendo un comportamiento consistente.

En cuanto a la adaptación a distintas plataformas, el modelo mantuvo un comportamiento lógico consistente tanto en entornos web como en aplicaciones de realidad virtual. La Figura 9 presenta una comparación visual entre ambas versiones.

### Figura 9

*Micropipeta en versiones web y de realidad virtual*



*Nota.* La versión web se muestra a la izquierda y la de realidad virtual a la derecha.

El modelo presenta ciertas limitaciones derivadas de su enfoque abstracto; en particular, sólo soporta mezclas homogéneas, por lo que no considera fenómenos asociados a sistemas heterogéneos como la separación de fases, las reacciones químicas o los cambios derivados de las interacciones entre líquidos.

## 4. CONCLUSIONES

El modelo lógico de interacción presentado en este trabajo, basado en estados y reglas lógicas, permitió regular de forma consistente el volumen y la composición del líquido transferido, sin recurrir a simulaciones físicas de fluidos. El modelo es capaz de realizar los procesos de aspiración y dispensación mediante una micropipeta, preservando la proporción relativa de cada sustancia en la composición del líquido.

La separación del modelo en módulos facilitó la adaptación del modelo a aplicaciones web y de realidad virtual, lo que le da independencia de la plataforma destino.

La decisión de emplear una abstracción lógica basada en datos en lugar de simulación física representa una alternativa viable para la transferencia de líquidos en entornos interactivos, especialmente donde no se requiere simulación física realista de fluidos.

## AGRADECIMIENTOS

Se agradece a la Dra. Silvia Maldonado Frías de la Facultad de Odontología de la UNAM por su colaboración para poder desarrollar las prácticas de laboratorio. Se reconoce también la colaboración del Mtro. Víctor Hugo Franco Serrano por el material elaborado y presentado en la Figura 7 de este reporte técnico.

## REFERENCIAS

- Bencomo, N., Cabot, J., Chechik, M., Cheng, B. H. C., Combemale, B., Wąsowski, A., & Zschaler, S. (2024). Abstraction Engineering. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2408.14074>
- Cen, Y., Deng, H., Ma, Y., & Liang, X. (2024). A real-time and interactive fluid modeling system for mixed reality. *IEEE Transactions on Visualization and Computer Graphics*, 30(11), 7310–7320. <https://doi.org/10.1109/TVCG.2024.3456140>
- Fusaro, M., Lisi, M. P., Era, V., Porciello, G., Candidi, M., Aglioti, S. M., & Boukarras, S. (2025). The transformative power of virtual reality: redefining interactions in virtual platforms, education, healthcare, and workplaces. *Topoi*, 44(4), 1071–1086. <https://doi.org/10.1007/s11245-025-10216-1>
- Guntakandla, A. R. (2025). Modular architecture: A scalable and efficient system design approach for enterprise applications. *World Journal of Advanced Research and Reviews*, 26(1), 3114–3126. <https://doi.org/10.30574/wjarr.2025.26.1.1340>
- Radianti, J., Majchrzak, T. A., Fromm, J., & Wohlgenannt, I. (2020). A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education*, 147, 103778. <https://doi.org/10.1016/j.compedu.2019.103778>

## ANEXO A. REPRESENTACIÓN VISUAL DEL LÍQUIDO MEDIANTE SHADER

### A.1 Instrucciones

1. Adjuntar el *script* LiquidShaderController al objeto que contiene la malla del líquido.
2. Asignar un material con el *shader* DRDE/Liquid/SimpleFill a la geometría interior que representa el líquido con el recipiente lleno. La malla debe corresponder únicamente al volumen del líquido. Para una prueba rápida se puede usar un cilindro.
3. Desde el componente responsable de la transferencia, al finalizar la operación, solicitar al recipiente (Liquids.cs) la actualización de los valores de fillLevel (en el rango de 0 y 1) y liquidColor, según el volumen actual del líquido y su composición.

Los fragmentos de código correspondientes al *shader* y al *script* de control se presentan en las Secciones A.5 y A.6, respectivamente.

### A.2 Funcionamiento

El *script* LiquidShaderController calcula automáticamente la altura mínima y máxima de la malla del líquido. El *shader* trabaja con un valor de llenado normalizado entre 0 y 1, pero la malla puede tener cualquier altura o escala local. Para ello, el *script* toma los límites reales verticales de la malla y los envía al *shader* para normalizar la altura internamente.

### A.3 Consideraciones

- Este ejemplo es ilustrativo. Los fragmentos han sido simplificados con fines de claridad.
- La malla del líquido debe corresponder a la malla interior del recipiente.
- La altura visible no representa necesariamente el volumen real del líquido de forma exacta.
- La relación entre volumen y altura depende de la forma del recipiente y, en caso necesario, debe aproximarse mediante un cálculo adicional.
- Si la malla incluye partes que no pertenecen al volumen del líquido, el *script* no calculará adecuadamente los valores de MinY y MaxY.
- Si MinY y MaxY no describen adecuadamente la geometría útil del líquido, el nivel visual resultante será incorrecto.

### A.4 Versiones de implementación

Unity 6000.4.0f1

URP 17.4.0

XR Interaction Toolkit 3.3.1 (para la versión de RV)

## A.5 Código del *shader* SimpleFill

```

01. // Archivo SimpleFill.shader
02.
03. // Shader simple para representar el nivel visible de un liquido dentro de una malla.
04. Shader "DRDE/Liquid/SimpleFill"
05. {
06.     Properties
07.     {
08.         // Color uniforme del liquido, incluyendo su transparencia.
09.         [MainColor] _LiquidColor ("Liquid Color", Color) = (0.2, 0.5, 0.9, 0.75)
10.         // Nivel de llenado normalizado del 0 al 1.
11.         _Filllevel ("Fill Level", Range(0, 1)) = 0.5
12.         // Alturas locales del volumen del liquido, usadas para normalizar la Y.
13.         _MinY ("Min Y", Float) = 0
14.         _MaxY ("Max Y", Float) = 1
15.         // Control explicito de caras visibles sin requerir geometria duplicada.
16.         [Enum(Caras delanteras, 0, Caras traseras, 1, Ambas, 2)] _RenderFaces ("Render Faces", Float) = 2
17.         // Superficie superior aproximada en el plano del nivel de llenado.
18.         [Toggle] _TopSurface ("Top Surface", Float) = 0
19.     }
20.
21.     SubShader
22.     {
23.         Tags
24.         {
25.             "RenderPipeline" = "UniversalPipeline"
26.             "Queue" = "Transparent"
27.             "RenderType" = "Transparent"
28.         }
29.
30.         HLSLINCLUDE
31.         #include "Packages/com.unity.render-pipelines.universal/ShaderLibrary/Core.hlsl"
32.
33.         CBUFFER_START(UnityPerMaterial)
34.         half4 _LiquidColor;
35.         float _Filllevel;
36.         float _MinY;
37.         float _MaxY;
38.         float _RenderFaces;
39.         float _TopSurface;
40.         CBUFFER_END
41.
42.         struct Attributes
43.         {
44.             float4 positionOS : POSITION;
45.         };
46.
47.         struct Varyings
48.         {
49.             float4 positionHCS : SV_POSITION;
50.             float localY : TEXCOORD0;
51.         };
52.
53.         Varyings vert(Attributes input)
54.         {
55.             Varyings output;
56.             // Se proyecta la geometria normalmente y se conserva la Y local
57.             // para decidir luego que parte del volumen se dibuja.
58.             output.positionHCS = TransformObjectToHClip(input.positionOS.xyz);
59.             output.localY = input.positionOS.y;
60.             return output;
61.         }
62.
63.         Varyings vertTopSurface(Attributes input)
64.         {
65.             Varyings output;
66.             float fillY = lerp(_MinY, _MaxY, saturate(_Filllevel));
67.             float3 topPositionOS = float3(input.positionOS.x, fillY, input.positionOS.z);
68.             output.positionHCS = TransformObjectToHClip(topPositionOS);
69.             output.localY = fillY;
70.             return output;
71.         }
72.
73.         half4 fragBase(Varyings input) : SV_Target
74.         {

```

```

75.         // Se normaliza la altura local de la malla usando el rango vertical
76.         // Útil del volumen del líquido. Esto evita exigir una malla 0..1.
77.         float heightRange = max(_MaxY - _MinY, 0.0001);
78.         float normalizedY = saturate((input.localY - _MinY) / heightRange);
79.         // Se recorta todo lo que quede por encima del nivel de llenado.
80.         float cutoff = (_Filllevel * 1.0001) - 0.0001;
81.         clip(cutoff - normalizedY);
82.         // El color es uniforme.
83.         return _LiquidColor;
84.     }
85.
86.     half4 fragFront(Varyings input) : SV_Target
87.     {
88.         if (_RenderFaces > 0.5 && _RenderFaces < 1.5)
89.         {
90.             clip(-1);
91.         }
92.         return fragBase(input);
93.     }
94.
95.     half4 fragBack(Varyings input) : SV_Target
96.     {
97.         if (_RenderFaces < 0.5)
98.         {
99.             clip(-1);
100.        }
101.        return fragBase(input);
102.    }
103.
104.    half4 fragTopSurface(Varyings input) : SV_Target
105.    {
106.        if (_TopSurface < 0.5)
107.        {
108.            clip(-1);
109.        }
110.        return fragBase(input);
111.    }
112.    ENDHLSL
113.
114.    Pass
115.    {
116.        Name "BackFaces"
117.        Tags { "LightMode" = "UniversalForward" }
118.        // Mezcla transparente simple.
119.        Blend SrcAlpha OneMinusSrcAlpha
120.        ZWrite Off
121.        Cull Front
122.        HLSLPROGRAM
123.        #pragma vertex vert
124.        #pragma fragment fragBack
125.        ENDHLSL
126.    }
127.
128.    Pass
129.    {
130.        Name "FrontFaces"
131.        Tags { "LightMode" = "SRPDefaultUnlit" }
132.        Blend SrcAlpha OneMinusSrcAlpha
133.        ZWrite Off
134.        Cull Back
135.        HLSLPROGRAM
136.        #pragma vertex vert
137.        #pragma fragment fragFront
138.        ENDHLSL
139.    }
140.
141.    Pass
142.    {
143.        Name "TopSurface"
144.        Tags { "LightMode" = "SRPDefaultUnlit" }
145.        Blend SrcAlpha OneMinusSrcAlpha
146.        ZWrite On
147.        ZTest Less
148.        Cull Off
149.        HLSLPROGRAM
150.        #pragma vertex vertTopSurface
151.        #pragma fragment fragTopSurface
152.        ENDHLSL
153.    }
154. }
155. }

```

## A.6 Código del script de control del *shader*

```

01. // Archivo LiquidShaderController.cs
02.
03. using UnityEngine;
04.
05. // Controlador simple para el shader del liquido.
06. [ExecuteAlways]
07. [RequireComponent(typeof(MeshFilter), typeof(Renderer))]
08. public class LiquidShaderController : MonoBehaviour
09. {
10.     public enum FaceRenderMode
11.     {
12.         FrontFaces = 0,
13.         BackFaces = 1,
14.         Both = 2
15.     }
16.
17.     [Header("Visual")]
18.     [Range(0f, 1f)]
19.     public float fillLevel = 0.5f;
20.     public Color liquidColor = new Color(0.2f, 0.5f, 0.9f, 0.75f);
21.     public FaceRenderMode renderFaces = FaceRenderMode.Both;
22.     public bool topSurface = false;
23.
24.     [Header("Bounds")]
25.     public bool autoCalculateBounds = true;
26.     public float manualMinY = 0f;
27.     public float manualMaxY = 1f;
28.
29.     [Header("Update")]
30.     public bool updateContinuously = false;
31.
32.     static readonly int FillLevelId = Shader.PropertyToID("_FillLevel");
33.     static readonly int LiquidColorId = Shader.PropertyToID("_LiquidColor");
34.     static readonly int MinYId = Shader.PropertyToID("_MinY");
35.     static readonly int MaxYId = Shader.PropertyToID("_MaxY");
36.     static readonly int RenderFacesId = Shader.PropertyToID("_RenderFaces");
37.     static readonly int TopSurfaceId = Shader.PropertyToID("_TopSurface");
38.
39.     Renderer m_renderer;
40.     MeshFilter m_meshFilter;
41.     MaterialPropertyBlock m_propertyBlock;
42.
43.     void OnEnable()
44.     {
45.         CacheComponents();
46.         ApplyToShader();
47.     }
48.
49.     void OnValidate()
50.     {
51.         CacheComponents();
52.         ApplyToShader();
53.     }
54.
55.     void Update()
56.     {
57.         if (!updateContinuously)
58.         {
59.             return;
60.         }
61.         ApplyToShader();
62.     }
63.
64.     // Método público para cambiar el nivel de llenado desde otros scripts.
65.     public void SetFillLevel(float newFillLevel)
66.     {
67.         FillLevel = Mathf.Clamp01(newFillLevel);
68.         ApplyToShader();
69.     }
70.

```

```
71. // Método público para cambiar el color del líquido desde otros scripts.
72. public void SetLiquidColor(Color newLiquidColor)
73. {
74.     liquidColor = newLiquidColor;
75.     ApplyToShader();
76. }
77.
78. // Método público de conveniencia para actualizar nivel y color al mismo tiempo.
79. public void SetFillLevelAndColor(float newFillLevel, Color newLiquidColor)
80. {
81.     fillLevel = Mathf.Clamp01(newFillLevel);
82.     liquidColor = newLiquidColor;
83.     ApplyToShader();
84. }
85.
86. // Método público para recalcular los límites de la malla si esta cambia.
87. public void RefreshShaderBounds()
88. {
89.     ApplyToShader();
90. }
91.
92. void CacheComponents()
93. {
94.     if (m_renderer == null)
95.     {
96.         m_renderer = GetComponent<Renderer>();
97.     }
98.     if (m_meshFilter == null)
99.     {
100.         m_meshFilter = GetComponent<MeshFilter>();
101.     }
102.     if (m_propertyBlock == null)
103.     {
104.         m_propertyBlock = new MaterialPropertyBlock();
105.     }
106. }
107.
108. void ApplyToShader()
109. {
110.     if (m_renderer == null || m_meshFilter == null)
111.     {
112.         return;
113.     }
114.     Mesh mesh = m_meshFilter.sharedMesh;
115.     if (mesh == null)
116.     {
117.         return;
118.     }
119.     float minY;
120.     float maxY;
121.     if (autoCalculateBounds)
122.     {
123.         // Mesh.bounds esta en espacio local del mesh.
124.         // El objeto puede cambiar de escala en la escena.
125.         Bounds bounds = mesh.bounds;
126.         minY = bounds.min.y;
127.         maxY = bounds.max.y;
128.     }
129.     else
130.     {
131.         minY = manualMinY;
132.         maxY = manualMaxY;
133.     }
134.     m_renderer.GetPropertyBlock(m_propertyBlock);
135.     m_propertyBlock.SetFloat(FillLevelId, fillLevel);
136.     m_propertyBlock.SetColor(LiquidColorId, liquidColor);
137.     m_propertyBlock.SetFloat(MinYId, minY);
138.     m_propertyBlock.SetFloat(MaxYId, maxY);
139.     m_propertyBlock.SetFloat(RenderFacesId, (float)renderFaces);
140.     m_propertyBlock.SetFloat(TopSurfaceId, topSurface ? 1f : 0f);
141.     m_renderer.SetPropertyBlock(m_propertyBlock);
142. }
143. }
```

## Análisis documental para la toma de decisiones en el ámbito de las tecnologías de la información y comunicación

*Documentary analysis for decision-making in the field of information and communication  
technologies*

### Información del reporte:

Licencia Creative Commons



El contenido de los textos es responsabilidad de los autores y no refleja forzosamente el punto de vista de los dictaminadores, o de los miembros del Comité Editorial, o la postura del editor y la editorial de la publicación.

Para citar este reporte técnico:

Valenzuela Argüelles, R. (2026). Análisis documental para la toma de decisiones en el ámbito de las tecnologías de la información. *Cuadernos Técnicos Universitarios de la DGTIC*, 4 (2), páginas (49 - 64). <https://doi.org/10.22201/dgtic.30618096e.2026.4.2.175>

### Rebeca Valenzuela Argüelles

Dirección General de Cómputo y de Tecnologías de  
Información y Comunicación

Universidad Nacional Autónoma de México

[rebecav@unam.mx](mailto:rebecav@unam.mx)

ORCID: 0009-0002-8243-3258

### Resumen

Con el fin de orientar los procesos universitarios mediante planes, políticas y soluciones en el marco de un plan de gobierno digital, se analizaron 81 planes de desarrollo de instancias de la Universidad Nacional Autónoma de México. Para ello, se identificaron los programas y proyectos con el uso de Tecnologías de Información y Comunicación, en los cuales las autoridades de las escuelas, facultades, institutos y centros de investigación consideran enfocar sus esfuerzos durante sus respectivas administraciones. El análisis se centró en dos grandes dimensiones: las Tecnologías de Información y Comunicación (TIC) en la administración (sistemas, equipamiento y conectividad) y las TIC en la academia (docencia, investigación y difusión). De este análisis, destacaron los temas que más ocupan a las instancias universitarias en dichas dimensiones, lo cual permitió correlacionar esta información con otras fuentes, a efectos de considerar aquellos procesos universitarios que pueden incluirse en dicho plan de gobierno digital.

**Palabras clave:** Metodología cualitativa, transformación digital universitaria, plan de gobierno digital en universidades públicas, TIC en la administración universitaria, TIC para la academia.

## Abstract

*In order to guide university processes through plans, policies, and solutions within the framework of a digital governance plan, 81 development plans from units of the Universidad Nacional Autónoma de México were analyzed. This involved identifying programs and projects utilizing Information and Communication Technologies, in which the authorities of schools, faculties, institutes, and research centers intend to focus their efforts during their respective administrations. The analysis centered on two main dimensions: Information and Communication Technologies (ICT) in administration (systems, equipment, and connectivity) and ICT in academia (teaching, research, and dissemination). The analysis highlighted the issues that most concern university units in these dimensions, allowing for correlation of this information with other sources to identify university processes that can be included in the digital governance plan.*

**Keywords:** *Qualitative methodology, university digital transformation, digital government plan in public universities, ICT in university administration, ICT for academia.*

## 1. INTRODUCCIÓN

La Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) se encuentra trabajando en el *Diseño de un plan de gobierno digital para la UNAM*, el cual es un proyecto del Plan de Desarrollo Institucional 2023-2027 (UNAM, 2024). En este sentido, el gobierno digital, desde la perspectiva universitaria, está entendido como el “conjunto de estrategias institucionales coordinadas para orientar la digitalización, simplificación, sistematización y/o automatización de los procesos universitarios, a través de planes, políticas y soluciones tecnológicas que favorezcan la eficiencia de los procesos sustantivos y administrativos de la institución” (Ventura Miranda *et al.*, comunicación personal, 1 de septiembre de 2024).

De acuerdo con el planteamiento de la necesidad de la Universidad Nacional Autónoma de México (UNAM), se llevó a cabo el análisis de los planes de desarrollo de la institución vigentes al año 2025 (UNAM, 2025). Las dos dimensiones que se plantearon para este análisis fueron las TIC para la administración y las TIC para la academia.

El objetivo del proyecto fue analizar 81 planes de desarrollo publicados por la UNAM, con el fin de identificar y acotar las líneas de trabajo en el ámbito de TIC y, de este modo, aportar información al plan de gobierno digital de la UNAM.

## 2. DESARROLLO TÉCNICO

### 2.1 METODOLOGÍA

Como se ha indicado, se identificó la utilidad de realizar el análisis de 81 planes de desarrollo publicados por las instancias universitarias (los cuales se pueden consultar en el Anexo A), con el fin de distinguir los programas, proyectos y actividades que se tienen previstos en el ámbito de TIC. Esto buscó enriquecer a las demás iniciativas en curso para la definición de los temas prioritarios en este ámbito dentro del plan de gobierno digital de la UNAM.

El análisis de documentos es una metodología cualitativa que involucra la revisión sistemática y la interpretación, con el fin de obtener información sobre un cierto fenómeno (Chand, 2025). Para la definición del proceso de análisis de los planes de desarrollo, se construyó un planteamiento propio que obedece a las características particulares de los documentos y del estudio a realizar, el cual se basó en diferentes autores.

### 2.1.1 PLANTEAMIENTO 1

Martínez-Corona *et al.* (2023) proponen un proceso de 11 fases que se listan a continuación:

Fase 1. Determinar el marco del análisis documental.

Fase 2. Plantear el problema de investigación:

- a) Sujeto: a quién se estudiará,
- b) Objeto: qué se estudiará del sujeto,
- c) Método: cómo se observará.

Fase 3. Establecer categorías de análisis.

Fase 4. Establecer el tipo de documentos.

Fase 5. Seleccionar el criterio de la búsqueda de información. Mercado (2008) recomienda representar a todos los puntos de vista que ofrecen los autores del problema de estudio bajo la selección de la mejor literatura disponible.

Fase 6. Establecer criterios de búsqueda adecuados, como idioma, periodo o tipo de documento y palabras esenciales.

Fase 7. Establecer criterios de revisión:

- a) normas,
- b) criterios de inclusión,
- c) criterios de exclusión,
- d) criterios de extracción de datos (citas textuales o no textuales).

Fase 8. Diseñar o adaptar un instrumento de registro documental.

Fase 9. Localizar documentos y extraer los datos, categorizarlos y registrarlos en el instrumento y su respectiva referencia.

Fase 10. Verificar la categorización subyacente.

Fase 11. Elaborar el informe.

## 2.1.2 PLANTEAMIENTO 2

Con base en Chand (2025), se recomiendan cuatro pasos que implican: identificar documentos relevantes, evaluar tanto su autenticidad como su credibilidad y codificar los datos e interpretar los hallazgos. En la metodología, se recuperó el tercer y cuarto paso, pues los pasos uno y dos están atendidos *a priori*.

El primer paso es identificar y seleccionar los documentos relevantes para las preguntas a resolver. Es necesario definir criterios claros para su selección, a fin de garantizar que los materiales se ajusten a los objetivos del estudio. Esto puede implicar la búsqueda en fuentes diversas y es fundamental verificar la accesibilidad, así como las consideraciones éticas del uso de documentos, especialmente cuando se trata de materiales confidenciales o sensibles.

El segundo paso es evaluar la autenticidad y credibilidad de los documentos. La autenticidad se refiere a si un documento es genuino, mientras que la credibilidad evalúa la fiabilidad y la exactitud del contenido. Se debe considerar el contexto en el que se genera el documento, el cual puede generar sesgos.

El tercer paso implica la codificación y categorización de los datos. En este paso, se sugiere que se analice sistemáticamente el contenido de los documentos, identificando temas, patrones y conceptos clave. Esto puede implicar la codificación manual, con el fin de resaltar segmentos de texto significativos, o el empleo de *software* de análisis de datos cualitativos para organizar y analizar grandes volúmenes de documentos de forma eficiente. La codificación ayuda a estructurar los datos, lo que facilita la elaboración de interpretaciones significativas y la vinculación de los hallazgos con los marcos teóricos.

El cuarto paso plantea la interpretación de los hallazgos, lo cual implica sintetizar la información extraída para responder a las preguntas planteadas. Este proceso requiere una comprensión profunda del contexto, el lenguaje y los significados de los documentos. El análisis cualitativo de documentos busca descubrir las narrativas, los valores y las ideologías presentes en un texto.

## 2.1.3 PLANTEAMIENTO 3

Las etapas de una revisión sistemática (Arias-Odón, 2023) son cuatro:

1. Formulación de una pregunta.
2. Búsqueda y selección de los estudios o artículos en función de los criterios de inclusión y exclusión.
3. Extracción de datos relevantes.
4. Análisis e interpretación de resultados.

## 2.1.4 METODOLOGÍA PROPIA BASADA EN LOS AUTORES DESCRITOS

De acuerdo con los autores consultados, a continuación, se describen los pasos definidos para el caso del análisis de los documentos de los planes de desarrollo de las instancias universitarias de la UNAM:

Paso 1. Planteamiento del problema

- a) Definición del sujeto de estudio,
- b) definición del objeto de estudio,
- c) definición de categorías generales,
- d) método de revisión.

Paso 2. Establecer el criterio de búsqueda y revisión

- a) Criterios de inclusión,
- b) criterios de exclusión,
- c) criterios de extracción.

Paso 3. Registro documental

- a) Definición de los datos a obtener,
- b) extracción de los datos relevantes.

Paso 4. Análisis de la información

- a) Revisión de las citas textuales y definición de líneas generales de información y primer conteo de menciones,
- b) análisis de líneas redundantes.

## 2.2 APLICACIÓN DE LA METODOLOGÍA

Paso 1. Planteamiento del problema de investigación

Sujeto de estudio: Planes de desarrollo publicados en la página de la Coordinación de Planeación, Evaluación y Simplificación de la Gestión Institucional (UNAM, 2025).

Objeto de estudio: Identificar proyectos y acciones relacionadas con el uso de TIC, para detectar las prioridades en el tema para las instancias universitarias, con el fin de enriquecer la definición de un plan institucional de gobierno digital.

Definición de categorías generales: Las categorías obedecen a la estructura organizacional de la UNAM en relación con el desarrollo de las TIC; de esta manera, se llevó a cabo el análisis documental, considerando dos dimensiones de temas a analizar:

TIC en la administración:

- Sistemas académico-administrativos
- Equipamiento
- Conectividad

TIC en la academia:

- Docencia
- Investigación
- Difusión

Método de revisión: definición de palabras clave, búsqueda y lectura rápida de documentos.

Paso 2. Establecer el criterio de búsqueda y revisión

- Criterios de inclusión: temas administrativos, educativos, investigación y difusión con TIC.
- Criterios de exclusión: innovación en TIC, incubadoras, vinculación, todos ellos fuera del alcance de este primer proyecto de gobierno digital.

- Criterios de extracción: citas textuales de los documentos originales con la intención de preservar la información para posteriores estudios.

### Paso 3. Registro documental

- Definición de los datos: Instancia académica, periodo del plan, eje, programa, proyecto o acción que dan origen a la cita dentro del registro.
- Extracción de los datos relevantes: archivo de hoja de cálculo con dos hojas, una para cada categoría (dimensión), con las columnas definidas en el punto anterior. La primera lista de citas textuales incluyó 212 proyectos en el área de sistemas académico-administrativos, equipamiento y conectividad, así como 351 en el área de docencia, investigación y difusión con TIC.

### Paso 4. Análisis de la información

Los parámetros de análisis considerados fueron: la revisión de las citas textuales, la definición de las líneas de trabajo por categoría y un primer conteo.

Esta fase requiere un trabajo paralelo en tres sentidos: el análisis de la información original identificada, la síntesis en líneas de trabajo y el conteo de ocurrencias para cubrir todas las citas textuales.

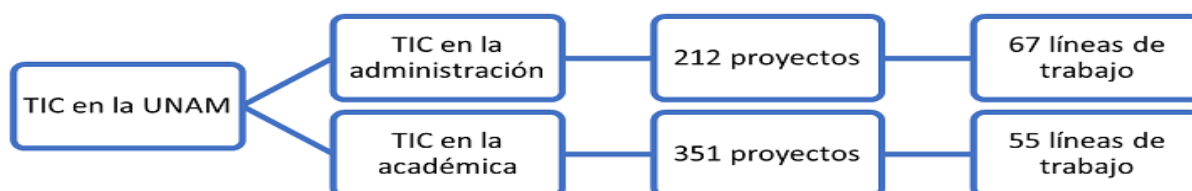
Este punto demanda un alto nivel de experiencia, a fin de analizar los textos recuperados y determinar la primera lista de líneas de trabajo, en la que se mantenga, lo más posible, la idea original planteada por la instancia universitaria.

Requiere un trabajo minucioso para identificar las relaciones entre líneas definidas y las citas textuales originales, de modo que cada una se encuentre reflejada en una línea de trabajo.

En la primera aproximación, se obtuvieron los resultados que se muestran en la Figura 1.

#### Figura 1

*Definición inicial de líneas de trabajo por dimensión*



## 2.2.1 ANÁLISIS DE LÍNEAS REDUNDANTES

En la Tabla 1 se muestra el resumen del proceso iterativo de simplificación de líneas de trabajo que se aplicó a la dimensión de TIC para la administración; en la Tabla 2 se presenta la dimensión de TIC para la academia. En la primera iteración, se hizo énfasis en condensar las líneas con una sola mención; y en la segunda, el énfasis fue en el fortalecimiento de líneas con la mayoría de menciones.

**Tabla 1**

*Proceso de simplificación de líneas de trabajo de la dimensión de TIC para la administración*

Líneas de trabajo	Resultados originales	Primera iteración	Segunda iteración
Con más de una mención	33	27	20
Con una mención	34	15	15
Total	67	42	35

**Tabla 2**

*Proceso de simplificación de líneas de trabajo de la dimensión de TIC para la academia*

Líneas de trabajo	Resultados originales	Primera iteración	Segunda iteración
Con más de una mención	41	36	36
Con una mención	14	7	7
Total	55	43	43

En las Figuras 2 y 3, se muestran las gráficas con las distribuciones obtenidas para cada una de las líneas de trabajo identificadas en las dos dimensiones definidas.

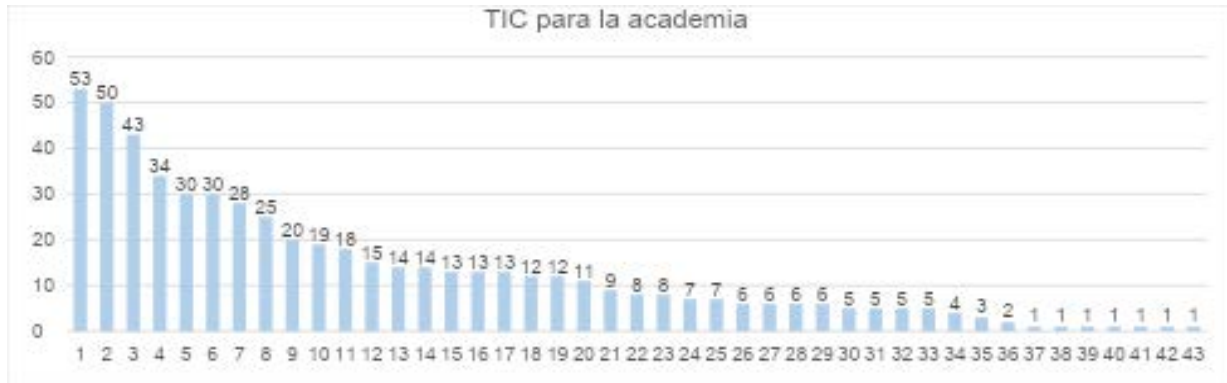
**Figura 2**

*Distribución de frecuencias de la dimensión de TIC para la administración*



**Figura 3**

*Distribución de frecuencias de la dimensión de TIC para la academia*



Aunque se trataba de una lista amplia de líneas de trabajo para ambas dimensiones, también era evidente que no todas representaban el mismo impacto. Por ello, se determinó ubicar los bloques con la mayoría de menciones y, con ese fin, se utilizó el análisis de 80/20 del principio de Pareto, el cual plantea que pocas causas producen la mayoría de los efectos observados (Chávez *et al.*, 2024).

### 2.2.2 ANÁLISIS DEL 80/20 DE PARETO

Dimensión de TIC para la administración: 317 menciones totales x 0.80 = 254. En la línea de trabajo 8, quedan exactamente 254 menciones, como se observa en la Figura 4, al sumar el número de ocurrencias para las barras en color anaranjado.

**Figura 4**

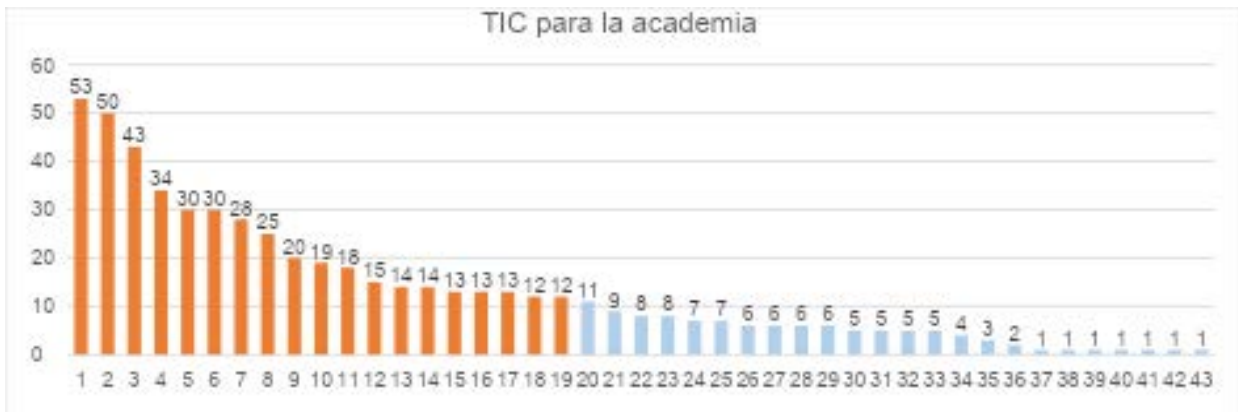
*Líneas de trabajo prioritarias con análisis de Pareto – TIC para la administración*



Dimensión de TIC para la academia: 566 menciones totales x 0.80 = 453. Considerando hasta la línea de trabajo 19, quedan 456 menciones, lo cual se puede identificar en la Figura 5.

**Figura 5**

*Líneas de trabajo prioritarias con análisis de Pareto - TIC para la academia*



Majka (2024) indica la relevancia de emplear la experiencia en el tema para no generalizar el análisis de Pareto y, de ese modo, aplicar el porcentaje adecuado considerando el que podría ser el más adecuado: 70/30 o 90/10.

Para este caso, se identificó que era conveniente no dejar fuera las líneas 9 a 11 por su relevancia para la dimensión de TIC para la administración, por lo que se decidió aplicar la regla 90/10, lo que se representa en la Figura 6.

**Figura 6**

*Líneas de trabajo prioritarias con análisis de Pareto 90/10 - TIC para la administración*

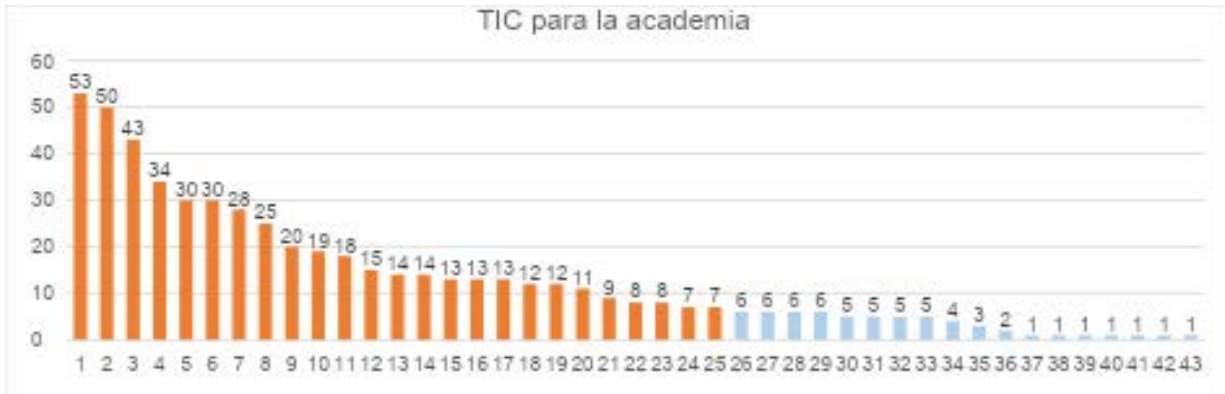


Nota. 317 menciones totales x 0.90 = 285

En el caso de la dimensión de TIC para la academia, sucede algo semejante, por lo que se aplicó también el análisis con 90%, lo que se representa en la Figura 7.

**Figura 7**

*Líneas de trabajo prioritarias con análisis de Pareto 90/10 - TIC para la academia*



Nota. 566 menciones totales x 0.90 = 509

### 3. RESULTADOS

Con el objeto de profundizar en la información que se ha recabado por diferentes medios para el proceso de construcción del plan de gobierno digital de la UNAM, se llevó a cabo el análisis documental de 81 planes de desarrollo de instancias universitarias (UNAM, 2025).

Después del análisis descrito, se identificaron 12 líneas de trabajo principales de la dimensión de TIC para la administración y 25 líneas de trabajo de la dimensión de TIC para la academia, las cuales provienen de 212 y 351 proyectos, respectivamente, definidos en dichos planes de desarrollo.

En la Tabla 3 se presentan las siguientes líneas prioritarias, obtenidas dentro de 81 planes de desarrollo de la dimensión de TIC para la administración:

**Tabla 3**

*Líneas de trabajo prioritarias para la dimensión de TIC para la administración*

No.	Línea de trabajo	Número de menciones
1	Equipamiento de cómputo y servidores	83
2	Sistematización de servicios administrativos	57
3	Conectividad	37
4	Sistematización de servicios escolares	24
5	Sistematización de información de personal académico	19
6	Sistematización de la comunicación interna y externa	13
7	Actualización y mantenimiento de <i>software</i> propietario y libre	11
8	Sistematización de la gestión documental y archivística	10
9	Sistematización de los trabajos de investigación	9
10	Mantenimiento preventivo y correctivo de equipo	8
11	Seguridad informática	8
12	Administración sustentable	6

De la dimensión de TIC para la academia, se presentan los temas identificados en la Tabla 4.

**Tabla 4**

*Líneas de trabajo prioritarias de la dimensión de TIC para la academia*

No.	Línea de trabajo	Número de menciones
1	Contenidos digitales: culturales, de divulgación y de difusión	53
2	Sitios web institucionales	50
3	Bases de datos, colecciones y repositorios universitarios	43
4	Redes sociales institucionales	34
5	Formación en de docentes e investigadores	30
6	Recursos educativos digitales	30
7	Plataformas y herramientas virtuales	28
8	Inteligencia artificial	25
9	Bibliotecas digitales	20
10	Enfoques pedagógicos diversos e innovadores	19
11	Actualización de espacios bibliotecarios	18
12	Asignaturas, licenciaturas y posgrados en línea	15
13	Acceso abierto	14

No.	Línea de trabajo	Número de menciones
14	Alto índice de reprobación	14
15	Educación continua	13
16	Proceso editorial	13
17	Simuladores, realidad virtual y realidad aumentada	13
18	Aprendizaje digital	12
19	Evaluación en línea	12
20	Idiomas	11
21	Tutoría y asesoría académica en línea y personalizada	9
22	Temas transversales para estudiantes: matemáticas, robótica, ciencia de datos e inteligencia artificial	8
23	Investigación difundida en publicaciones electrónicas	8
24	Servicios y contenidos para personas con discapacidad	7
25	Publicaciones digitales institucionales (gacetas, boletines, etc.)	7

Lo anterior refleja aquellos temas en los cuales las escuelas, facultades, institutos y centros de la UNAM podrían requerir apoyo para avanzar en su digitalización y transformación digital. Este trabajo abonará información al proyecto de gobierno digital a cargo de la DGTIC, con bases mucho más claras de lo que resulta relevante en la actualidad para las instancias universitarias.

## 4. CONCLUSIONES

La referencia de las obras consultadas permitió la adaptación y definición de una metodología de análisis documental para el caso específico que se requería resolver en este proyecto, de modo que fuera posible aprovechar la gran cantidad de información disponible.

Como ya se ha mencionado, se requerirá la correlación de la información recabada por otros medios, con la finalidad de terminar de acotar las líneas principales en las cuales la DGTIC se centrará para construir el plan de gobierno digital de la UNAM y su futura transformación digital, ya que, como indica Chand (2025), aunque la metodología de análisis documental es muy recomendable por no ser intrusiva, gracias a la disponibilidad de los documentos y el costo y tiempo implicados, también representa retos como la parcialidad y la limitación de la información que es posible encontrar en ellos (Bowen, 2009).

La metodología puede ser de gran utilidad para el análisis y sistematización de información. Por ejemplo, para las preguntas de respuesta abierta, que suelen requerirse en encuestas y que requieren representarse de algún modo para facilitar su aprovechamiento y la toma de decisiones con base en el panorama que proporcionan.

## AGRADECIMIENTOS

A María Teresa Ventura Miranda y a Erika San Juan Hernández por sus valiosas aportaciones en la elaboración de este trabajo.

## REFERENCIAS

- Arias-Odón, F. (2023). Investigación documental, investigación bibliométrica y revisiones sistemáticas. *Revista electrónica de Humanidades, Educación y Comunicación Social*, 22, 9-28. <https://dialnet.unirioja.es/servlet/articulo?codigo=9489470>
- Bowen, G.A. (2009). Document Analysis as a Qualitative Research Method. *Qualitative Research Journal*, 9, 27-40. <https://doi.org/10.3316/QRJ0902027>
- Chand, S. P. (2025). Methods of Data Collection in Qualitative Research: Interviews, Focus Groups, Observations, and Document Analysis. *Advances in Educational Research and Evaluation*, 6(1), 303-317. <https://doi.org/10.25082/AERE.2025.01.001>
- Chávez, F. L., De la Rosa, S. E., Manjarres, J. C., Valbuena, S., G., & Becerra-Torres, M. (2024). Diagrama de Pareto. Perspectiva de la Asignatura de Control de la Calidad. *Boletín de Innovación, Logística y Operaciones*, 6(1), 51-56. <https://revistascientificas.cuc.edu.co/bilo/article/view/4920>
- Majka, M. (2024). *Pareto Analysis: The 80/20 Rule for Effective Decision-Making*. [https://www.researchgate.net/publication/385085105\\_Pareto\\_Analysis\\_The\\_8020\\_Rule\\_for\\_Effective\\_Decision-Making](https://www.researchgate.net/publication/385085105_Pareto_Analysis_The_8020_Rule_for_Effective_Decision-Making)
- Martínez-Corona, J. I., Palacios-Almón, G. E., & Oliva-Garza, D. B. (2023). Guía para la revisión y el análisis documental: propuesta desde el enfoque investigativo. *Ra Ximhai*, 19(1), 67-83. <https://doi.org/10.35197/rx.19.01.2023.03.jm>
- Mercado, S. (2008). *¿Cómo hacer una tesis?: Licenciatura, Maestría y Doctorado*. (4a ed.). LIMUSA.
- Universidad Nacional Autónoma de México (2024). *Plan de Desarrollo Institucional de la Universidad Nacional Autónoma de México 2023-2027*. Página del Rector. <https://www.rector.unam.mx/docs/PDI-2023-2027.pdf>
- Universidad Nacional Autónoma de México (2025). *Planes de desarrollo*. Coordinación de Planeación, Evaluación y Simplificación de la Gestión Institucional. <https://www.planeacion.unam.mx/>

## ANEXO A. LISTA DE PLANES DE DESARROLLO

A continuación, se incluye el listado de planes de desarrollo las escuelas, facultades, institutos y centros con sus periodos de administración correspondientes, disponibles en el sitio web de la Coordinación de Planeación, Evaluación y Simplificación de la Gestión Institucional (UNAM, 2025), en los cuales se basó el análisis presentado.

Escuelas y Facultades	
1	Facultad de Arquitectura (2021-2025)
2	Facultad de Artes y diseño (2022-2026)
3	Facultad de Ciencias (2021-2025)
4	Facultad de Ciencias Políticas y Sociales (2024-2028)
5	Facultad de Contaduría y Administración (2024-2028)
6	Facultad de Derecho (2024-2028)
7	Facultad de Economía (2024-2028)
8	Facultad de Enfermería y Obstetricia (2023-2027)
9	Facultad de Filosofía y Letras (2021-2025)
10	Facultad de Ingeniería (2023-2027)
11	Facultad de Medicina (2024-2028)
12	Facultad de Medicina Veterinaria y Zootecnia (2024-2028)
13	Facultad de Música (2024-2028)
14	Facultad de Odontología (2022-2026)
15	Facultad de Psicología (2024-2028)
16	Facultad de Química (2023-2027)
17	Facultad de Estudios Superiores Acatlán (2024-2028)
18	Facultad de Estudios Superiores Aragón (2024-2028)
19	Facultad de Estudios Superiores Cuautitlán (2021-2025)
20	Facultad de Estudios Superiores Iztacala (2024-2028)
21	Facultad de Estudios Superiores Zaragoza (2022-2026)
22	Escuela Nacional de Artes Cinematográficas (2023-2027)
23	Escuela Nacional Colegio de Ciencias y Humanidades (2022-2026)
24	Escuela Nacional de Ciencias Forenses (2023-2027)
25	Escuela Nacional de Ciencias de la Tierra (2022-2026)
26	Escuela Nacional de Estudios Superiores, Unidad Juriquilla (2022-2026)
27	Escuela Nacional de Estudios Superiores, Unidad León (2021-2025)
28	Escuela Nacional de Estudios Superiores, Unidad Mérida (2022-2026)

<b>Escuelas y Facultades</b>	
29	Escuela Nacional de Estudios Superiores, Unidad Morelia (2024-2028)
30	Escuela Nacional de Lenguas, Lingüística y Traducción (2021-2025)
31	Escuela Nacional de Trabajo Social (2024-2028)
32	Escuela Nacional Preparatoria (2022-2026)
<b>Institutos</b>	
33	Astronomía (2024-2028)
34	Biología (2023-2027)
35	Biotecnología (2021-2025)
36	Ciencias Aplicadas y Tecnología (2022-2026)
37	Ciencias de la Atmósfera y Cambio Climático (2021-2025)
38	Ciencias del Mar y Limnología (2023-2027)
39	Ciencias Nucleares (2024-2028)
40	Ciencias Físicas (2022-2026)
41	Ecología (2020-2024)
42	Energías Renovables (2021-2025)
43	Física (2023-2027)
44	Fisiología Celular (2024-2028)
45	Geociencias (2024-2028)
46	Geofísica (2021-2025)
47	Geografía (2024-2028)
48	Geología (2022-2026)
49	Ingeniería (2024-2028)
50	Investigaciones Antropológicas (2024-2028)
51	Investigaciones Bibliográficas (2024-2028)
52	Investigaciones Bibliotecológicas y de la Información (2024-2028)
53	Investigaciones Biomédicas (2023-2027)
54	Investigaciones Económicas (2022-2026)
55	Investigaciones en Ecosistemas y Sustentabilidad (2023-2027)
56	Investigaciones en Materiales (2024-2028)
57	Investigaciones en Matemáticas Aplicadas y en Sistemas (2024-2028)
58	Investigaciones Estéticas (2021-2025)
59	Investigaciones Filológicas (2024-2028)
60	Investigaciones Filosóficas (2020-2024)
61	Investigaciones Históricas (2021-2025)

<b>Institutos</b>	
62	Investigaciones Jurídicas (2022-2026)
63	Investigaciones Sociales (2024-2028)
64	Investigaciones sobre la Universidad y la Educación (2022-2026)
65	Matemáticas (2024-2028)
66	Neurobiología (2024-2028)
67	Química (2022-2026)
68	Radioastronomía y Astrofísica (2023-2027)
<b>Centros</b>	
69	Ciencias Genómicas (2021-2025)
70	Ciencias Matemáticas (2023-2027)
71	Física Aplicada y Tecnología Avanzada (2022-2026)
72	Investigaciones en Geografía Ambiental (2023-2027)
73	Investigaciones en Materiales 2016-2020)
74	Investigaciones Interdisciplinarias en Ciencias y Humanidades (2024-2028)
75	Investigaciones Multidisciplinarias sobre Chiapas y la Frontera Sur (2023-2027)
76	Investigaciones sobre América del Norte (2021-2025)
77	Investigaciones sobre América Latina y el Caribe (2024-2028)
78	Investigaciones y Estudios de Género (2025-2029)
79	Nanociencias y Nanotecnología (2022-2026)
80	Peninsular en Humanidades y en Ciencias Sociales (2023-2027)
81	Regional de Investigaciones Multidisciplinarias (2023-2027)

# Modernización y estandarización en el desarrollo de complementos para Moodle

*Modernization and Standardization in Moodle Plugin Development*

## Información del reporte:

Licencia Creative Commons



El contenido de los textos es responsabilidad de los autores y no refleja forzosamente el punto de vista de los dictaminadores, o de los miembros del Comité Editorial, o la postura del editor y la editorial de la publicación.

Para citar este reporte técnico:

Zenil Rivas, P.E. (2026). Modernización y estandarización en el desarrollo de complementos para Moodle. *Cuadernos Técnicos Universitarios de la DGTIC*, 4 (2), páginas (65 - 78). <https://doi.org/10.22201/dgtic.30618096e.2026.4.2.169>

**Pablo Enrique Zenil Rivas**

Dirección General de Cómputo y de  
Tecnologías de Información y Comunicación  
Universidad Nacional Autónoma de México

[pablozr@unam.mx](mailto:pablozr@unam.mx)

ORCID: 0009-0007-3956-0194

## Resumen

El TICómetro de la Universidad Nacional Autónoma de México enfrentaba un doble desafío: la obsolescencia técnica de sus simuladores, la cual impidió su operación en versiones actualizadas de la plataforma, y la falta de un entorno de desarrollo estandarizado a nivel institucional. Para resolver esta problemática, se conformó un grupo multidisciplinario que definió una metodología ágil de desarrollo. El trabajo incluyó la comparación de diversas infraestructuras, la cual respaldó la decisión de elegir una solución basada en contenedores debido a su flexibilidad y reproducibilidad. Paralelamente, se diseñó una plantilla de complemento reutilizable que integró tecnologías web modernas dentro de la arquitectura de Moodle, lo que superó las limitaciones de su sistema de módulos nativo.

Como resultado, se implementó una versión moderna del simulador de hoja de cálculo que incorporó un sistema de evaluación granular, enfocado en analizar las características de la respuesta del estudiante, resolviendo así la ambigüedad de versiones previas. Se concluyó que el proyecto logró establecer un ambiente de desarrollo estandarizado, el cual ya ha sido utilizado en la creación e integración de nuevos simuladores.

**Palabras clave:** Contenedores en Docker, Node.js, interfaces con React, simuladores para TICómetro.

## Abstract

*The TICómetro project of the Universidad Nacional Autónoma de México faced a dual challenge: the technical obsolescence of its simulators, which prevented their operation on updated platform versions, and the lack of a standardized institutional development environment. To solve this problem, a multidisciplinary group was formed to define an agile development methodology. The work included comparing various infrastructures, ultimately opting for a container-based solution due to its flexibility and reproducibility. In parallel, a reusable plugin template was designed that integrated modern web technologies into the Moodle architecture, overcoming the limitations of the native module system.*

As a result, a modern version of the spreadsheet simulator was implemented, incorporating a granular evaluation system focused on analyzing the characteristics of the student's response, thus resolving the ambiguity of previous versions. It was concluded that the project successfully established a standardized development environment, which has already been used in the creation and integration of new simulators.

**Keywords:** *Docker containers, Node.js, React interface, simulators for TICómetro.*

## 1. INTRODUCCIÓN

La evaluación de habilidades en tecnologías de información y comunicación (TIC) es fundamental en la educación (Juca-Maldonado *et al.*, 2025). En la Universidad Nacional Autónoma de México (UNAM), el TICómetro (s. f.) es un instrumento de evaluación diagnóstica diseñado con el propósito de contar con información sobre el nivel de habilidades en el uso de las TIC de los estudiantes de bachillerato y licenciatura de nuevo ingreso a la UNAM. El instrumento, diseñado por la Dirección General de Cómputo y de Tecnologías de Información y Comunicación, proporciona datos empíricos a las entidades académicas. Estos datos permiten caracterizar el perfil de los estudiantes de primer ingreso en relación con sus habilidades en el manejo de las TIC; además, ofrece información relevante para la toma de decisiones en cuanto a la enseñanza y el uso de las TIC en los programas, actividades y necesidades de infraestructura. El instrumento se valida estadísticamente cada año. Su implementación inicial se realizó sobre la plataforma Moodle, de la cual se aprovechó el módulo de cuestionario para automatizar la calificación e integrar simuladores que evalúan habilidades prácticas.

El avance tecnológico y la migración de la plataforma a la versión 4.3 requirieron una actualización integral del *software*. Esto propició la actualización tecnológica de tres simuladores clave del TICómetro (hoja de cálculo, procesador de texto y búsqueda en Internet) y evidenció la necesidad de un entorno de desarrollo unificado en el área responsable del servicio, para facilitar el mantenimiento y el desarrollo futuro.

Para solventar esto y seguir las mejores prácticas en metodologías ágiles (López-Mora *et al.*, 2019), se conformó un grupo multidisciplinario. El equipo definió como premisas principales evitar complementos de paga, garantizar la compatibilidad con el núcleo de Moodle, facilitar el mantenimiento, fomentar la colaboración y publicar los resultados en la comunidad oficial de Moodle (Moodle, s.f.).

El objetivo general fue modernizar el desarrollo de complementos de simulación para el TICómetro mediante la creación de una plantilla que integra tecnologías web modernas (Lucas *et al.*, 2025) como React y Vite, superando las limitaciones del sistema de módulos nativo. Para asegurar la reproducibilidad del entorno, se optó por una arquitectura basada en contenedores (Wang, 2022). Esta metodología,

desarrollada entre el segundo semestre de 2024 y el primero de 2025, se aplicó en la actualización del simulador de hoja de cálculo y facilita la creación de nuevas herramientas que evalúen habilidades digitales emergentes.

## 2. DESARROLLO TÉCNICO

Para alcanzar el objetivo propuesto, el desarrollo se dividió en dos áreas principales: primero, se definió la metodología de trabajo y se implementaron los entornos de desarrollo; y segundo, se avanzó con el desarrollo del simulador de hoja de cálculo y, de forma paralela, con el de la plantilla de complemento tipo pregunta reutilizable, que sirvió como base para los nuevos simuladores.

### 2.1 METODOLOGÍA

El trabajo se organizó bajo un marco ágil (López-Mora *et al.*, 2019), mediante sesiones quincenales para la revisión de avances, discusión de soluciones y redefinición de prioridades. Se utilizó el GitLab institucional para el control de versiones y la gestión de la comunicación entre los académicos.

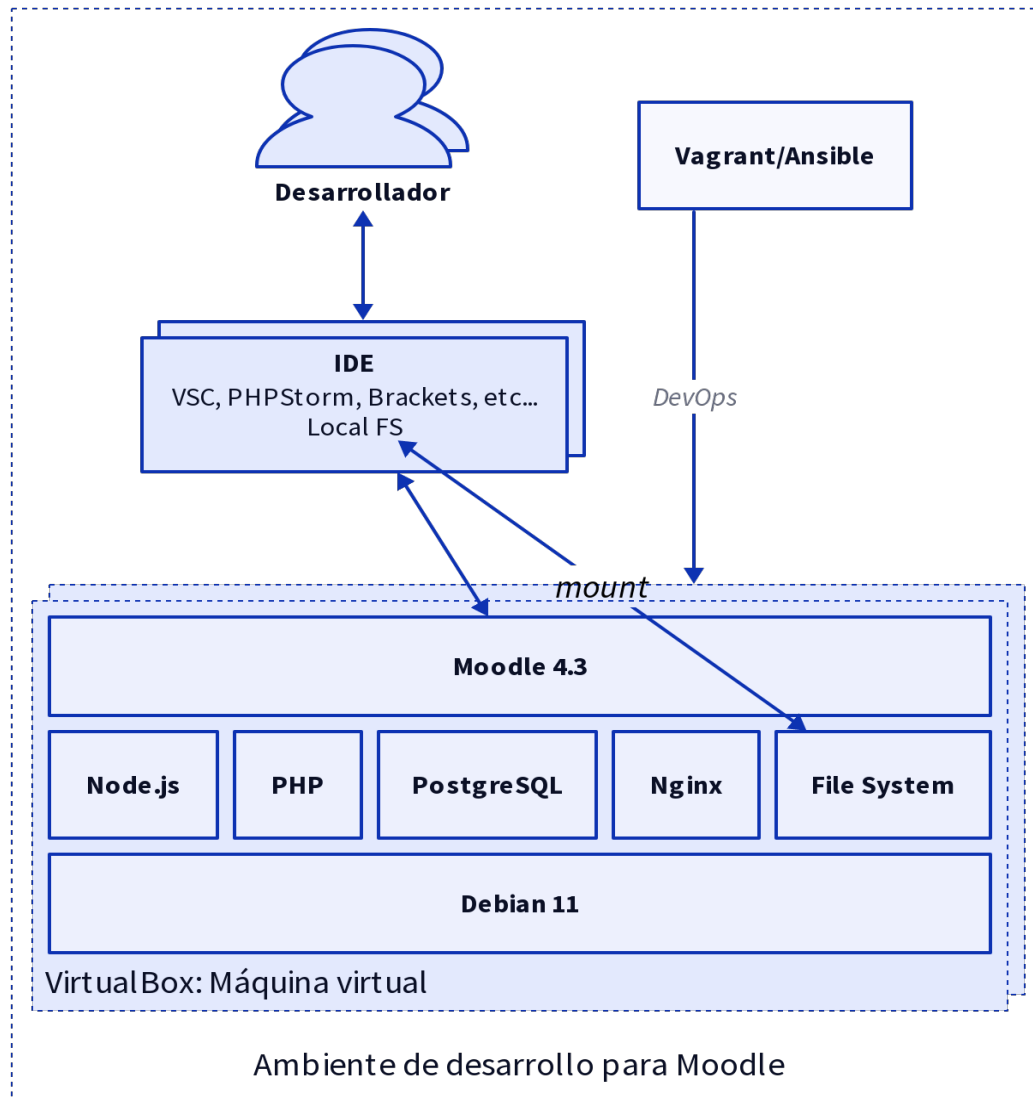
#### 2.1.1 ANÁLISIS Y SELECCIÓN DEL ENTORNO DE DESARROLLO

El proyecto requería un entorno de desarrollo reproducible con dependencias específicas para Moodle 4.3, incluyendo tecnologías de servidor y bases de datos como PHP, PostgreSQL y Nginx, así como las herramientas necesarias para la nueva plantilla web, tales como Node.js, React y Vite.

La primera solución utilizó máquinas virtuales en VirtualBox, gestionadas con Vagrant, para la reproducibilidad, y Ansible, para la automatización de la instalación sobre el sistema operativo Debian. Este enfoque empaquetaba el entorno en archivos de configuración, asegurando la replicabilidad. En la Figura 1 se presenta el ambiente de desarrollo utilizado.

**Figura 1**

*Ambiente de desarrollo para Moodle con Vagrant*



Sin embargo, esta solución presentó dos deficiencias:

- Manejo del sistema de archivos: obligaba al desarrollador a instalar sus herramientas de desarrollo (IDE) dentro de la máquina virtual, generando un cuello de botella en el flujo de trabajo.
- Gestión de versiones: para soportar otra versión de Moodle, se debía crear una máquina virtual completamente nueva, creando redundancias innecesarias.

En una segunda instancia, se adoptó el uso de contenedores con Docker y Docker Compose. Esta alternativa conservó las ventajas de las máquinas virtuales (replicando la arquitectura de la Figura 1) y

solucionó sus deficiencias, lo cual permitió trabajar sobre el sistema de archivos local y agilizó el flujo de trabajo (Wang, 2022).

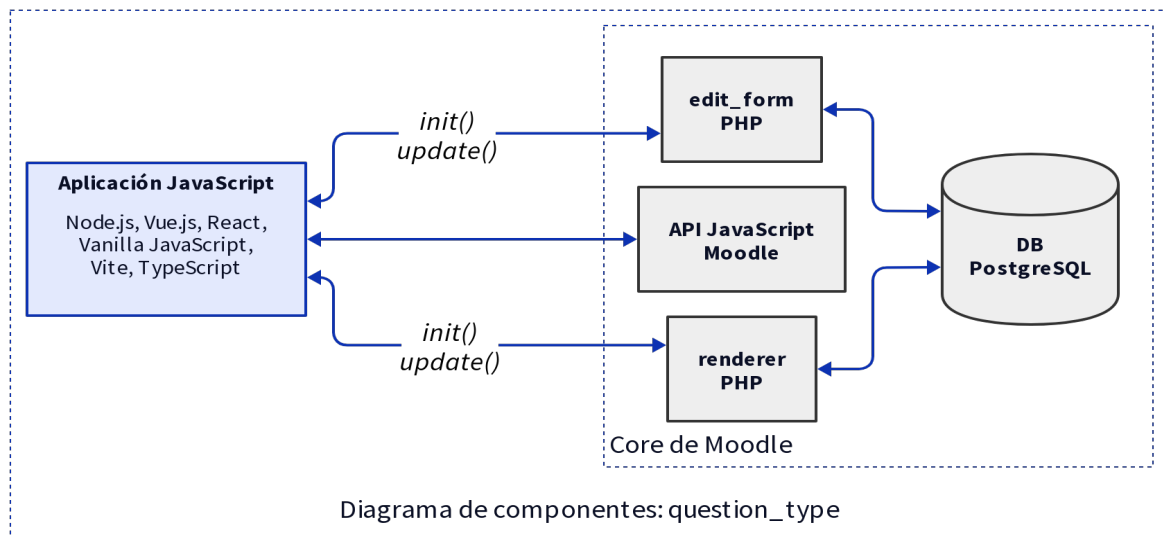
### 2.1.2 DISEÑO DE LA PLANTILLA DE COMPLEMENTO

Un reto clave fue diseñar una plantilla que permitiera el desarrollo sobre un entorno moderno de JavaScript (Lucas *et al.*, 2025), ya que Moodle utiliza nativamente un *stack* obsoleto basado en AMD (Wikipedia contributors, 2025), RequireJS y Grunt. Dicho *stack* presenta compilaciones lentas, requiere recarga manual y dependencias antiguas, tales como Node.js 14 o 16, (la versión actual, al momento del desarrollo, era la 22).

Por lo tanto, el diseño de la nueva plantilla se enfocó en el módulo App, el cual se presenta en la Figura 2 como una aplicación JavaScript general que se comunica con los componentes de Moodle.

**Figura 2**

*Diagrama de componentes plantilla de complemento tipo pregunta*



Dicha plantilla puede también desarrollarse en un entorno moderno de Node.js.

### 2.1.3 DISEÑO DEL SIMULADOR DE HOJA DE CÁLCULO

Para el simulador de hoja de cálculo, se analizaron varios paquetes de código abierto y se eligió FortuneSheet (Ruilisi, s. f.). La elección se fundamentó en su compatibilidad nativa con React, su facilidad de configuración y una interfaz de programación de aplicaciones (API) que provee información relevante sobre el contenido de cada celda (fórmulas y referencias, entre otros).

Además, se rediseñó el sistema de evaluación. La versión anterior obligaba a configurar múltiples "respuestas esperadas" por reactivo, ya que era imposible analizar las características de la celda. Esto generaba un problema crítico en la evaluación (Pordanjani & Salehi, 2025): si una respuesta correcta no estaba preconfigurada, el sistema la calificaba como incorrecta, invalidando la automatización. La nueva

versión subsanó esto al permitir la especificación de las características de cada celda (por ejemplo: uso de fórmula, función o rango) en una única respuesta esperada, lo que garantizó una evaluación objetiva.

### 3. RESULTADOS

A continuación, se presentan los resultados obtenidos tras aplicar la metodología propuesta. Primero, explicaremos cómo armamos un entorno de desarrollo unificado que el equipo puede replicar fácilmente mediante contenedores, y después presentaremos el desarrollo técnico del nuevo complemento y cómo funciona en la práctica el simulador de hoja de cálculo ya integrado en Moodle.

#### 3.1 ENTORNO DE DESARROLLO ESTANDARIZADO

Se adaptó el repositorio público *moodle-docker* (Moodle HQ, s. f.), el cual replica la arquitectura mostrada en la Figura 1 sin las desventajas de las máquinas virtuales. La adaptación se centró en la carga de parámetros mediante variables de entorno en un archivo *.env*, siguiendo un enfoque de cero configuración (*zero-configuration*). Esta solución facilita probar los complementos contra múltiples versiones de bases de datos, PHP y Moodle, proveyendo al equipo de un entorno ágil (Wang, 2022).

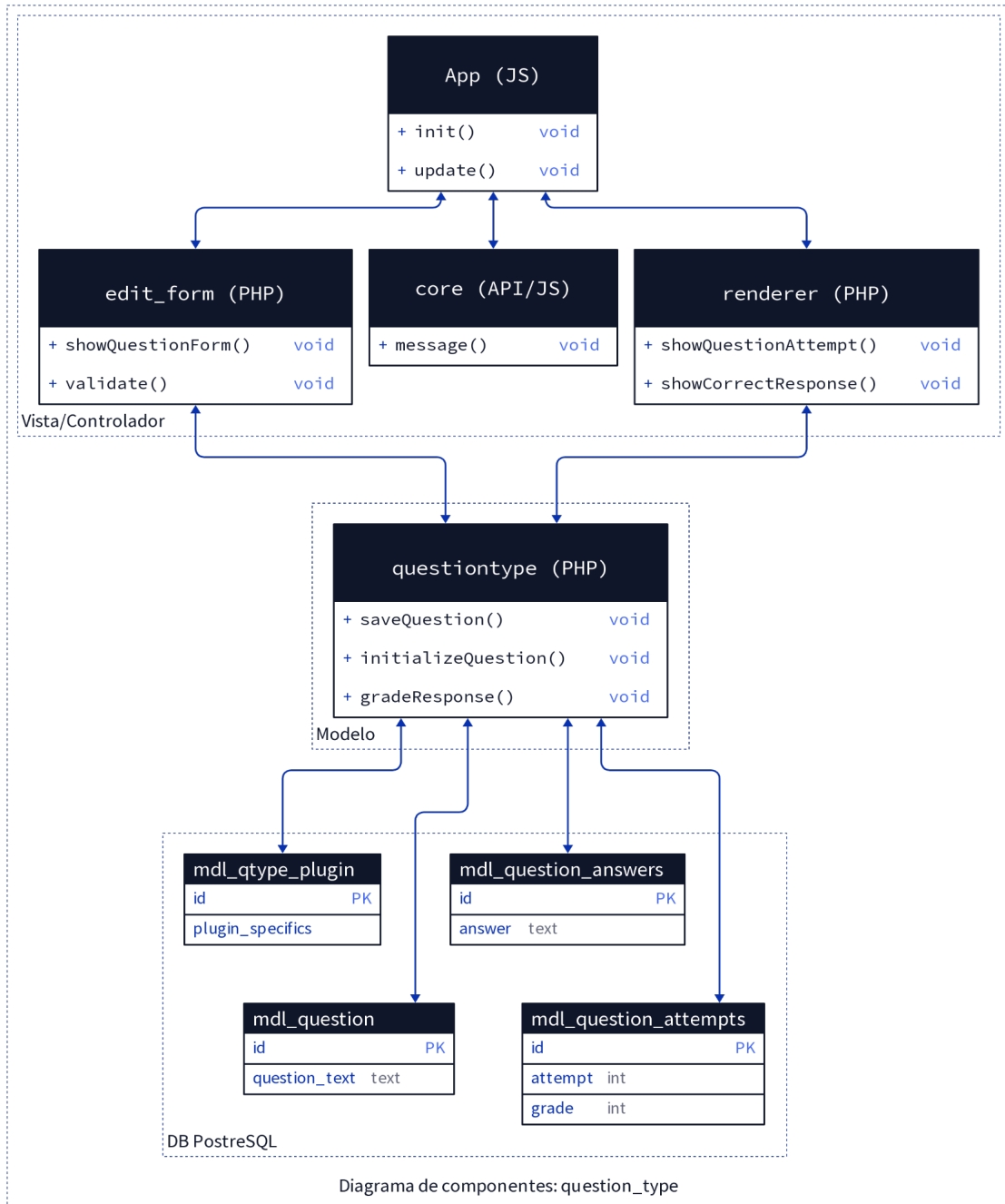
#### 3.2 PLANTILLA DE COMPLEMENTO MODERNO

Se desarrolló la plantilla de complemento de tipo “pregunta”, ilustrada en la Figura 3, la cual integra aplicaciones JavaScript de propósito general. Sus componentes principales son:

- *App*: encapsula la aplicación JavaScript de propósito general. Se comunica con la API de JavaScript de Moodle y los componentes de vista, manteniendo la coherencia de la información.
- *edit\_form*: componente PHP (vista/controlador) para crear, guardar y editar las preguntas en el banco de Moodle.
- *renderer*: componente PHP (vista/controlador) que presenta la pregunta al estudiante, gestiona el intento de respuesta y muestra la retroalimentación.
- *questiontype*: componente PHP del modelo que sirve como interfaz con la capa de persistencia.
- *DB*: Capa de persistencia donde se almacena la información de preguntas, respuestas e intentos.

**Figura 3**

*Componentes principales de la plantilla del complemento de tipo pregunta*



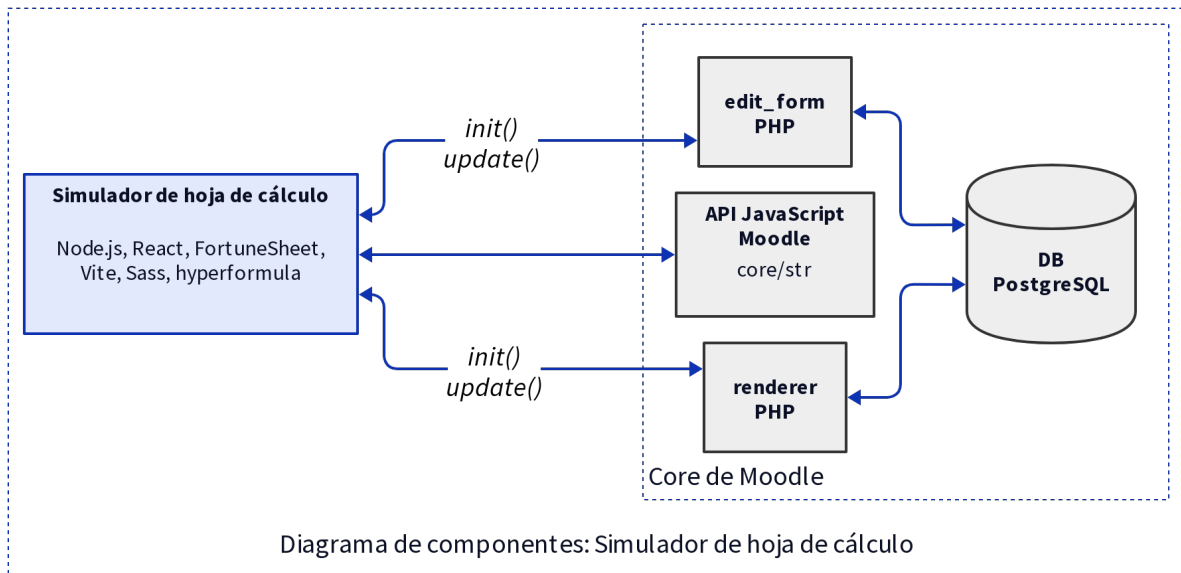
La relevancia de esta plantilla radica en que su componente App admite módulos de Node.js y se compila con Vite, lo que optimiza el flujo de desarrollo (Lucas *et al.*, 2025). Esta plantilla, compatible con React, se utilizó como base para los simuladores de hoja de cálculo y búsqueda en Internet.

### 3.3 SIMULADOR DE HOJA DE CÁLCULO

Con la plantilla descrita, se implementó la nueva versión del simulador de hoja de cálculo, cuya arquitectura se presenta en la Figura 4. Este complemento integra React para la interfaz, utiliza FortuneSheet para la manipulación interactiva de la hoja y accede a la API de Moodle para gestionar la internacionalización.

**Figura 4**

*Arquitectura del simulador de hoja de cálculo*



Los componentes *edit\_form* y *renderer* de Moodle gestionan las instancias de la hoja de cálculo y almacenan los datos en la base de datos. Se definieron cuatro vistas (instancias) específicas:

1. Vista de valores iniciales (*edit\_form*): ilustrada en la Figura 5, permite configurar el estado inicial de la hoja que verá el estudiante.

### Figura 5

Vista de valores iniciales en la configuración del componente



2. Vista de respuesta esperada (*edit\_form*): como se detalla en la Figura 6, configura la respuesta correcta e incluye opciones para definir qué elementos se evaluarán específicamente por cada celda (referencias, rangos o funciones).

**Figura 6**

*Vista de respuesta esperada en la configuración del componente*

## Edit form: respuesta esperada

fx

	A	B	C	D	E	F
1	3					
2	4					
3	5					
4	6					
5		7				
6		7				
7		8				
8		23				
9						
10						

Añadir
50 (más filas al final)
Volver arriba

+ Hoja1
- 100% +

Carga vista inicial

Hoja1			Re	Fu	Ra
Celda	Fórmula	Valor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A2		4			
A3		5			
A4		6			
B5	=sum(A1,A2)	7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B6	=a1+a2	7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
B7	=5+a1	8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
B8	=sum(A1:A4) + a1 + 2	23	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Re: Usa referencias a celda, Fu: Usa funciones, Ra: Usa rangos

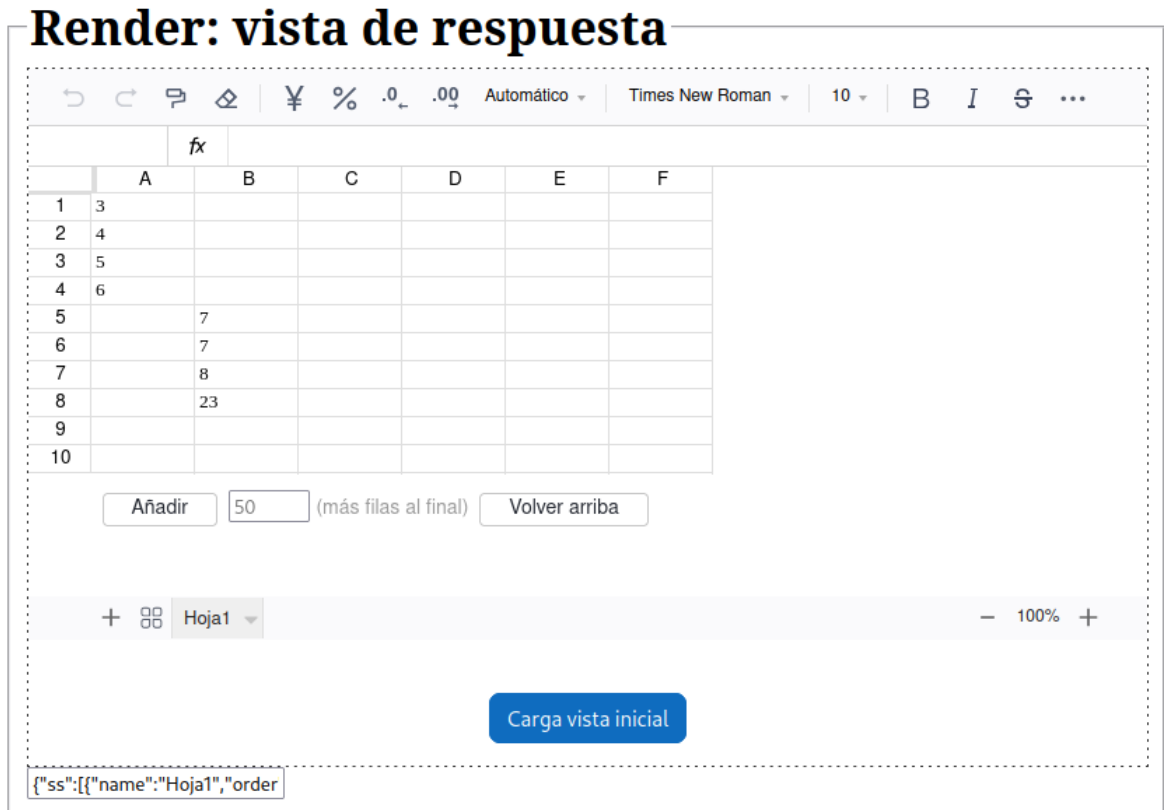
{"ss":{"name":"Hoja1","order

3. Vista de respuesta (*renderer*): presentada en la Figura 7, es la interfaz que ve el estudiante al responder el examen, la cual carga automáticamente el estado inicial configurado.

### Figura 7

Vista de respuesta del estudiante durante la evaluación

## Render: vista de respuesta



The screenshot displays the 'Render: vista de respuesta' interface, which includes a spreadsheet editor. The spreadsheet has 10 rows and 6 columns (A-F). The data in the spreadsheet is as follows:

	A	B	C	D	E	F
1	3					
2	4					
3	5					
4	6					
5		7				
6		7				
7		8				
8		23				
9						
10						

Below the spreadsheet, there are buttons for 'Añadir' (Add), a text input field containing '50', '(más filas al final)' (more rows at the end), and 'Volver arriba' (Go up). At the bottom of the interface, there is a 'Carga vista inicial' (Load initial view) button and a status bar showing 'Hoja1' and '100%' zoom.

4.Vista de sólo lectura (*renderer*): mostrada en la Figura 8.

**Figura 8**

Vista de sólo lectura con retroalimentación de la evaluación

## Render: solo lectura

↶ ↷ 🔍 🗑️ ¥ % .0 .00 Automático Times New Roman 10 B I 🔗 ⋮

	A	B	C	D	E	F
1	3					
2	4					
3	5					
4	6					
5		7				
6		7				
7		8				
8		23				
9						
10						

Añadir

(más filas al final)
Volver arriba

+ 🗄️
Hoja1
- 100% +

Hoja1			Re	Fu	Ra
Celda	Fórmula	Valor			
× A2		4			
× A3		5			
× A4		6			
✓ B5	=sum(A1,A2)	7	✓	✓	☐
× B6	=a1+a2	7	✓	☐	☐
× B7	=5+a1	8	✓	☐	☐
× B8	=sum(A1:A4) + a1 + 2	23	✓	✓	✓

Re: Usa referencias a celda, Fu: Usa funciones, Ra: Usa rangos

```

{"ss":{"name":"Hoja1","order":1}}
[{"B5":{"grade":1},"C5":{"grad
          
```

Esta vista se presenta al estudiante a modo de retroalimentación, indicando cuáles celdas fueron correctas y cuáles incorrectas.

### 3.3.1 MECANISMO DE EVALUACIÓN

Una parte clave fue la implementación de un mecanismo de evaluación granular (Pordanjani & Salehi, 2025). La respuesta dada (RD) se califica de 0 a 1, comparándola con la respuesta esperada (RE). Para cada celda modificada, la calificación es correcta (1) sólo si cumple todas las condiciones preconfiguradas, es decir:

- El valor de la celda en RD es igual al valor en RE.
- Si en RE se usa fórmula, en RD también.
- Si en RE se usan referencias a celdas, en RD también.
- Si en RE se usan funciones, en RD también.
- Si en RE se usan rangos, en RD también.

Cada celda puede obtener un máximo de 5/5; si el valor es distinto, la calificación es 0.

## 4. CONCLUSIONES

El proyecto surgió de una doble necesidad crítica: por un lado, la obsolescencia técnica de los simuladores del TICómetro, que impedía su funcionamiento en Moodle 4.3 y la correcta evaluación de habilidades TIC; y por otro, la falta de un proceso de desarrollo estandarizado en el área responsable del servicio, donde la diversidad de entornos dificultaba la colaboración y el mantenimiento.

Los objetivos se lograron mediante el trabajo colaborativo de un grupo interdisciplinario de personas académicas. La implementación de un entorno estándar de desarrollo ágil y fácilmente replicable, así como el diseño de una plantilla de complemento reutilizable, resolvieron la problemática. Sobre esta base, se actualizó tecnológicamente el simulador de hoja de cálculo, dotándolo de un sistema de evaluación granular. Esto valida la nueva arquitectura como una solución robusta y mantenible, disponible para futuros desarrollos.

Actualmente, se ha adoptado esta plantilla y el entorno *moodle-docker* (Moodle HQ, s. f.) como el estándar para nuevos simuladores (edición de video y de audio). Asimismo, esta infraestructura se utiliza para concluir la actualización del simulador de búsqueda en Internet y se continúa con el trabajo para la publicación de estas herramientas en el repositorio oficial de la comunidad (Moodle, s.f.).

Aunque la adopción de la arquitectura basada en contenedores ha demostrado una mejora cualitativa significativa en la agilidad del trabajo diario, una futura línea de análisis podría definir y cuantificar estas métricas de productividad. Por otro lado, la nueva versión del simulador de hoja de cálculo fue aplicada a alumnos de nuevo ingreso a finales de 2025. En 2026 el equipo pedagógico se encuentra analizando los datos recopilados. Los resultados cuantitativos de esta aplicación y su comparativa con versiones anteriores del simulador serán objeto de un trabajo futuro para validar estadísticamente las mejoras en la evaluación. Este proyecto va más allá de sólo actualizar una herramienta de evaluación. A nivel institucional, sienta las bases para crear de forma ágil y colaborativa futuros recursos educativos dentro de la Universidad. Además, a un nivel más global, la meta de publicar estos complementos en la comunidad de Moodle

abre la puerta para que otras instituciones superen las limitaciones de la plataforma y puedan evaluar habilidades digitales con mayor precisión.

## AGRADECIMIENTOS

Agradezco a todo el equipo de trabajo de la Dirección de Innovación en Tecnologías para la Educación de la DGTIC-UNAM, cuyas discusiones dieron pie a una solución robusta para los retos que presentó este proyecto: María Ramírez Bedolla, Mónica Avila Quintana, Adriana Areli Bravo Lozano y María Elizabeth Martínez Sánchez, Miguel Zuñiga González, Francisco Isaac Moguel Pedraza y Javier Rodrigo Díaz Espinosa.

## REFERENCIAS

- Juca-Maldonado, A. X., García-Vera, J. S., Juca Maldonado, F., & Carrión González, J. T. (2025). Competencias tecnológicas de la generación Z: Una evaluación cuantitativa de las habilidades frente a la percepción de los nativos digitales. *Revista Transdisciplinaria de Estudios Sociales y Tecnológicos (RTEST)*, 5(2), 16–24. <https://doi.org/10.58594/rtest.v5i2.160>
- López-Mora, D., Villamar Coloma, M., Bravo-Pino, Á., & Lozano-Rodríguez, E. (2019). El uso de las metodologías ágiles y su importancia para el desarrollo de software. *Killkana Técnica*, 3(1), 23–28. [https://doi.org/10.26871/killkana\\_tecnica.v3i1.473](https://doi.org/10.26871/killkana_tecnica.v3i1.473)
- Lucas, W., Nunes, R., Bonifácio, R., Carvalho, F., Lima, R., Silva, M., Torres, A., Accioly, P., Monteiro, E., & Saraiva, J. (2025). Understanding the adoption of modern JavaScript features: An empirical study on open-source systems. *Empirical Software Engineering*, 30, Article 107. <https://doi.org/10.1007/s10664-025-10663-9>
- Moodle. (s.f.). Moodle. <https://moodle.org/>
- Moodle HQ. (s. f.). *moodle docker*. GitHub. <https://github.com/moodlehq/moodle-docker>
- Pordanjani, Z., & Salehi, K. (2025). Limitations of electronic assessment: A systematic review. *Qualitative Research*, 3(1), 111–130. <https://doi.org/10.15157/qr.2025.3.1.111-130>
- Ruilisi. (s. f.). *fortune-sheet*. GitHub. <https://github.com/ruilisi/fortune-sheet>
- Universidad Nacional Autónoma de México. (s. f.). *TICómetro*. Educatic. <https://ticometro.educatic.unam.mx>
- Wang, W. (2022). Research on using Docker container technology to realize rapid deployment environment on virtual machine. En *2022 8th Annual International Conference on Network and Information Systems for Computers (ICNISC)*, (pp. 541–544). IEEE. <https://doi.org/10.1109/ICNISC57059.2022.00112>
- Wikipedia contributors. (2025). *Asynchronous module definition*. Wikipedia, The Free Encyclopedia. [https://en.wikipedia.org/wiki/Asynchronous\\_module\\_definition](https://en.wikipedia.org/wiki/Asynchronous_module_definition)

# Implementación de un servidor de impresión bajo un enfoque de seguridad y sostenibilidad tecnológica

## *Implementation of a print server under a security and technological sustainability approach*

### Información del reporte:

Licencia Creative Commons



El contenido de los textos es responsabilidad de los autores y no refleja forzosamente el punto de vista de los dictaminadores, o de los miembros del Comité Editorial, o la postura del editor y la editorial de la publicación.

Para citar este reporte técnico:

Gutiérrez Molina, A. (2026). Implementación de un servidor de impresión bajo un enfoque de seguridad y sostenibilidad tecnológica. *Cuadernos Técnicos Universitarios de la DGTIC*, 4 (2), páginas (79 - 98). <https://doi.org/10.22201/dgtic.30618096e.2026.4.2.174>

### Alfonso Gutiérrez Molina

Dirección General de Cómputo y de Tecnologías de  
Información y Comunicación

Universidad Nacional Autónoma de México

[amolina@unam.mx](mailto:amolina@unam.mx)

ORCID: 0000-0002-7467-0651

### Resumen

Debido a la necesidad de contar con documentos físicos en el ambiente académico y laboral, el servicio de impresión es de gran importancia en el día a día; sin embargo, éste puede presentar problemas de desconexión y de acceso no autorizado, por lo que se decidió implementar un servidor de impresión regulado para solucionar dichas situaciones.

Esta implementación incluyó una actualización de *hardware*, así como el mantenimiento preventivo de un equipo reutilizado con la finalidad de fungir como servidor de impresión. Asimismo, se realizó la instalación del sistema operativo, una *suite* ofimática y herramientas de seguridad, que incluyen la instalación y configuración de un *firewall* y listas de control de acceso para restringir el uso de equipos no autorizados. La conexión de los dispositivos cliente al servidor se realizó mediante el uso de rutas *Universal Naming Convention* bajo el esquema cliente-servidor. De esta manera, se incorporaron mecanismos de autenticación y segmentación de red para fortalecer la seguridad.

Los resultados evidenciaron una reducción en los reportes o solicitudes relacionadas con fallas de impresión y reconfiguración de dispositivos, lo que permitió optimizar la carga operativa del área de Soporte Técnico. De igual manera, la reutilización de equipos de cómputo programados para su desincorporación contribuyó a la

disminución de residuos electrónicos, promoviendo así la sostenibilidad tecnológica. Esta estrategia demostró que la centralización de servicios, junto con el uso de *software* libre y aplicaciones de controles de seguridad, permite incrementar la eficiencia operativa, además de fortalecer la seguridad de la información y fomentar prácticas sostenibles dentro de la dependencia.

**Palabras clave:** Gestión de TI, seguridad de la información, servidor de impresión, *software* libre, sostenibilidad tecnológica.

### Abstract

*Due to the need for physical documents in academic and professional environments, printing services are of great importance on a daily basis; however, they often present problems with disconnections and unauthorized access. Therefore, it was decided to implement a regulated print server to address these issues.*

*This implementation included a hardware upgrade, as well as preventive maintenance of a repurposed computer to serve as the print server. The installation of the operating system, an office suite, and security tools was also carried out, including the installation and configuration of a firewall and access control lists to restrict the use of unauthorized equipment. Client devices were connected to the server using Universal Naming Convention paths under a client-server architecture. This incorporated authentication mechanisms and network segmentation to strengthen security.*

*The results showed a reduction in reports or requests related to printing failures and device reconfiguration, which optimized the workload of the Technical Support area. Similarly, the reuse of computers that were going to be discarded contributed to a decrease in electronic waste, thus promoting technological sustainability. This strategy demonstrated that centralizing services, using open-source software, and implementing security controls increases operational efficiency, strengthens information security, and fosters sustainable practices within the department.*

**Keywords:** IT management, information security, print server, open-source software, technological sustainability.

## 1. INTRODUCCIÓN

En la actualidad, el uso de documentos digitales en ambientes laborales, académicos y personales se ha convertido en una práctica fundamental, ya que permite optimizar el manejo de la información, mejorar la eficiencia en procesos administrativos, colaborar entre varias personas a la vez y almacenar los datos de forma sencilla y segura. Por consiguiente, logra reducir costos, ahorrar tiempo y promover formas de trabajo ágil y sostenible, además de que facilita el trabajo remoto. A pesar de lo anterior, hoy en día existen situaciones específicas en las que el uso de un documento físico sigue siendo relevante, ya sea por razones legales, de seguridad, de accesibilidad, etc.

En algunas áreas de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC), es imprescindible imprimir documentos; por ello, es primordial que el servicio de impresión siga operando de manera constante, funcional y eficaz al momento de ser necesario. En ocasiones, dicho servicio presentaba problemas como la pérdida de conexión entre el equipo de cómputo y la impresora,

el acceso de usuarios no autorizados y la dificultad para configurar una impresora predeterminada; esto podía derivar en problemas de seguridad, vulnerabilidades y retrasos en el servicio.

Debido a las situaciones expuestas con anterioridad, en el año 2025 se decidió implementar un servidor de impresión con el objetivo de brindar un servicio continuo, estable, seguro y disponible sólo para el personal autorizado.

## 2. DESARROLLO TÉCNICO

Un servidor de impresión es un equipo diseñado para que los usuarios puedan tener acceso a servicios de impresión. Para ello, los usuarios se conectan de manera remota por medio de una cuenta y contraseña establecidos, lo que permite que tengan acceso al servicio de impresión. Además, la utilización de credenciales de acceso permite gestionar el uso por medio de listas administradas a través de un *firewall* o a través de las herramientas del mismo sistema operativo. Lo anterior trae consigo beneficios significativos, tales como la disminución en el consumo de tóner, papel y energía eléctrica. Esto propicia la optimización de recursos materiales y la reducción de costos; la eliminación de equipos de impresión personales, ahorro de espacio y la disminución de solicitudes de servicio de soporte correctivo, lo que permite que el trabajo se centre en el mantenimiento preventivo de otras áreas (Albarracín *et al.*, 2021).

El servidor de impresión que se propuso como solución está compuesto por una computadora de escritorio, una impresora láser monocromática, un *firewall*, *software* ofimático y un antivirus. Las características específicas de los componentes que conforman el servidor se presentan en el Anexo A del reporte.

Aplicaciones de uso para el servidor de impresión:

Se consideró la posibilidad de que fuera necesaria la modificación ocasional de archivos o documentos directamente en el servidor de impresión, particularmente en situaciones que requieran atención inmediata o por necesidades operativas del área. En este sentido, se contempló la instalación de un *software* libre de ofimática básico que permita la creación, modificación y revisión de documentos sin comprometer la función principal del equipo. Adicionalmente, el equipo debe contar con mecanismos de seguridad que garanticen la integridad, disponibilidad y confidencialidad del servicio de impresión, por lo cual, se consideró necesaria la implementación de un *firewall* debidamente configurado, así como la instalación de un antivirus. Las herramientas elegidas para este fin fueron LibreOffice (ofimática), ZoneAlarm (*firewall*) y Avira (antivirus). El proceso de selección basado en pruebas de uso se explica a detalle en el Anexo B del reporte.

### 2.1 METODOLOGÍA

Debido a las características técnicas del equipo de cómputo de escritorio y con el propósito de optimizar su rendimiento, primero se llevó a cabo un proceso integral de mantenimiento preventivo y actualización de *hardware* y *software*. El mantenimiento preventivo en sistemas informáticos constituye una práctica fundamental para prolongar la vida útil de los equipos, reducir fallas y mejorar la eficiencia operativa (Vermaat *et al.*, 2018).

En primera instancia, se realizó una limpieza interna y externa del equipo, lo que incluyó la remoción de polvo en componentes críticos, limpieza de ventiladores y sustitución de la pasta térmica en el disipador

de calor del CPU, con el fin de mejorar la disipación de calor y garantizar un funcionamiento térmico adecuado. Esto evitó el sobrecalentamiento, uno de los principales factores que afectan el desempeño y la estabilidad de los sistemas de cómputo (Andrews, 2019).

Como siguiente paso, se amplió la memoria RAM de 8 GB a 16 GB, lo que potenció la capacidad de procesamiento y el desempeño en multitareas y mejoró la eficacia del sistema operativo (Patterson & Hennessy, 2017). Posteriormente, se reemplazó el disco duro mecánico (HDD) de 1 TB por una unidad de estado sólido (SSD) de 500 GB para optimizar la velocidad de lectura y escritura del sistema.

Finalmente, se instaló el sistema operativo Windows 10 Pro de 64 bits, junto con el paquete de herramientas ofimáticas y *software* de seguridad, específicamente ZoneAlarm y Avira antivirus, con el objetivo de garantizar un entorno de trabajo seguro y eficiente. La implementación de soluciones de seguridad informática es esencial para la protección de datos y la prevención de amenazas cibernéticas (Stalling & Brown, 2018).

### **Instalación de LibreOffice**

Después de haber descargado el instalador de LibreOffice, se procedió con la instalación correspondiente mediante el asistente. Durante el proceso, se recomienda seleccionar la configuración predeterminada; de esta manera, se asegura la instalación completa de los componentes principales de la *suite* ofimática, tales como procesador de texto, hoja de cálculo, presentaciones y manejador de bases de datos. Como siguiente paso, se verificó la correcta integración con el sistema operativo, comprobando la creación de accesos directos y la asociación de formatos de archivos compatibles.

### **Instalación de ZoneAlarm**

Una vez descargado el *software* ZoneAlarm, se procedió a la instalación mediante el asistente correspondiente, verificando que la integración con el sistema operativo y la activación de los módulos de protección en tiempo real fuera correcta.

Posteriormente, se realizó la configuración inicial del *firewall*, ajustando los niveles de seguridad y definiendo reglas específicas para el filtrado de tráfico. La configuración correcta de las reglas en un *firewall* permite reducir vulnerabilidades y ataques en entornos organizacionales (Whitman & Mattord, 2021).

Como parte del proceso, se estableció la lista de control de acceso (ACL), especificando los equipos autorizados para hacer uso del servicio de impresión dentro de la red local. Las ACL permiten aplicar políticas de seguridad basadas en direccionamiento IP, puertos y protocolos, garantizando que únicamente los dispositivos previamente enlistados puedan acceder a los recursos compartidos dentro de la red local, fortaleciendo así la confidencialidad, integridad y disponibilidad de los recursos tecnológicos (Stalling & Brown, 2022).

Estas acciones se alinean con los controles establecidos en la ISO/IEC 27001:2022, norma que establece la necesidad de implementar controles técnicos apropiados para gestionar riesgos de seguridad de la información dentro del Sistema de Gestión de Seguridad de la Información (SGSI) (International Organization for Standardization, 2022).

Por consiguiente, la instalación y correcta configuración del *firewall* protegerá al sistema de accesos no autorizados, además de que garantizará un uso controlado y seguro de los servicios de la red local.

### **Instalación de Avira**

Se procedió con la descarga del *software* Avira antivirus desde su sitio oficial, verificando la compatibilidad con el sistema operativo del servidor de impresión; después, se ejecutó el archivo de instalación mediante el asistente correspondiente, seleccionando la configuración predeterminada para garantizar la activación de los módulos de protección.

Al terminar la instalación, se verificó la correcta actualización de las bases de datos de firmas de virus y se comprobó la activación de la protección del sistema en tiempo real para asegurar la detección y prevención de amenazas informáticas. Este *software* permite fortalecer la seguridad del equipo, reducir riesgos de infección por *malware* y garantizar la integridad de la información que se encuentra en el servidor.

### **Configuración de la impresora láser B/N**

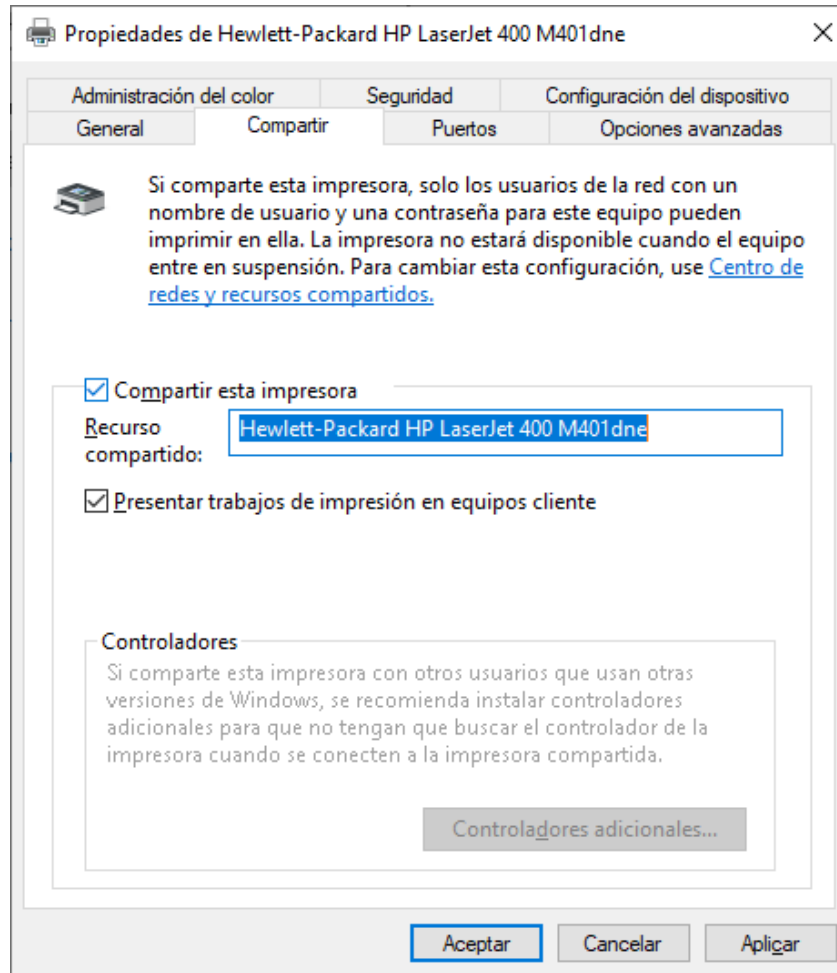
Como primer paso, se realizó la configuración y conexión directa de la impresora al servidor de impresión, utilizando la interfaz USB. Para garantizar el correcto funcionamiento, es indispensable contar con los controladores (*drivers*) actualizados del fabricante, ya que los controladores actúan como intermediarios entre el sistema operativo y la impresora, asegurando la compatibilidad y el acceso a todas las funciones de la impresora (Silberschatz *et al.*, 2022).

A pesar de que el sistema operativo Windows 10 Pro cuenta con mecanismos de detección automática de *hardware* (*Plug and Play*) que permiten reconocer e instalar dispositivos de manera básica, es recomendable asegurarse que la instalación de los controladores sean los específicos de la impresora para optimizar el rendimiento y evitar conflictos de compatibilidad. Como siguiente paso, se realizaron las configuraciones necesarias para compartir la impresora en red, habilitando las opciones de uso compartido y definiendo los permisos de acceso para los usuarios autorizados. Compartir recursos dentro de una red local es una práctica común en entornos organizacionales, pues permite optimizar la infraestructura, reducir costos y mejorar la eficiencia operativa (Kurose & Ross, 2021).

En la Figura 1 se puede observar la configuración de la impresora como un recurso compartido dentro de la red local.

## Figura 1

### Configuración de la impresora como recurso compartido



### Configuración de las listas de acceso en ZoneAlarm

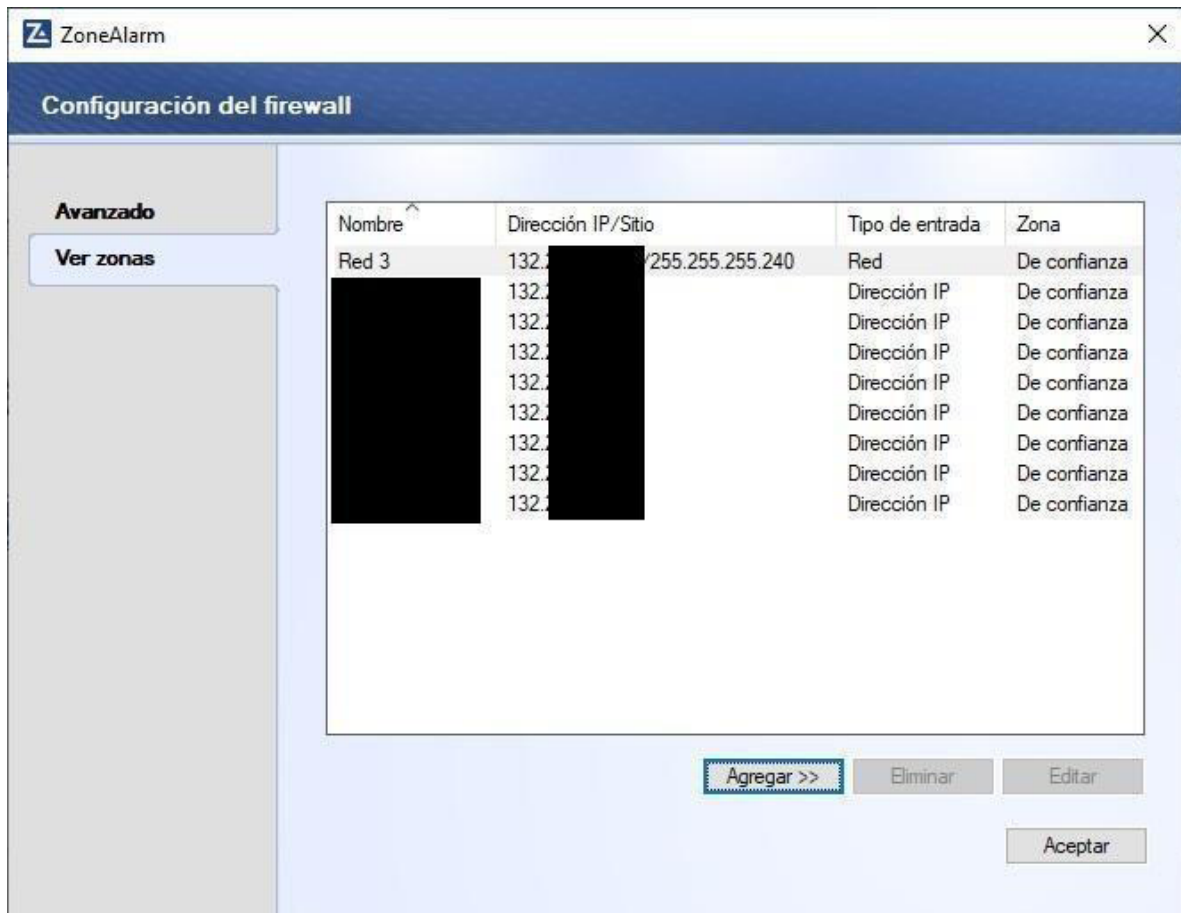
Después de haber instalado ZoneAlarm, se procedió a la configuración de las listas de control de acceso (ACL), con el objetivo de regular el uso del servicio de impresión compartido en la red local. Para ello, se creó una lista conformada por las direcciones IP de los equipos autorizados. Las ACL son un mecanismo básico en la administración de redes, pues permiten filtrar el tráfico con base en parámetros como dirección IP, protocolo y puerto, proporcionando un control granular sobre servicios expuestos (Kurose & Ross, 2021).

De esta manera, un *firewall* personal como Zone Alarm permite implementar políticas a nivel *host*, reforzando la protección perimetral y complementando otros controles de seguridad en la infraestructura de la red local (Stalling & Brown, 2022).

En la Figura 2 se puede observar la Lista de Control de Acceso (ACL).

**Figura 2**

*Lista de Control de Acceso (ACL) de equipos autorizados para imprimir*



Esta lista está conformada por la dirección IP de los equipos autorizados a imprimir.

### Conexión y configuración de los equipos autorizados al servidor de impresión

Una vez configurada y compartida la impresora, y tras haber definido previamente la lista de acceso a la zona de confianza en el *firewall*, se procedió a la conexión de los equipos autorizados al servidor de impresión. La vinculación se realizó mediante el uso de rutas UNC (Universal Naming Convention), empleando la sintaxis `\\NombreDelHost\NombreImpresora`, lo que permite localizar y acceder a recursos compartidos dentro de una red local basada en arquitectura cliente-servidor, ya que este esquema facilita la administración centralizada de dispositivos y optimiza la adecuada gestión de recursos.

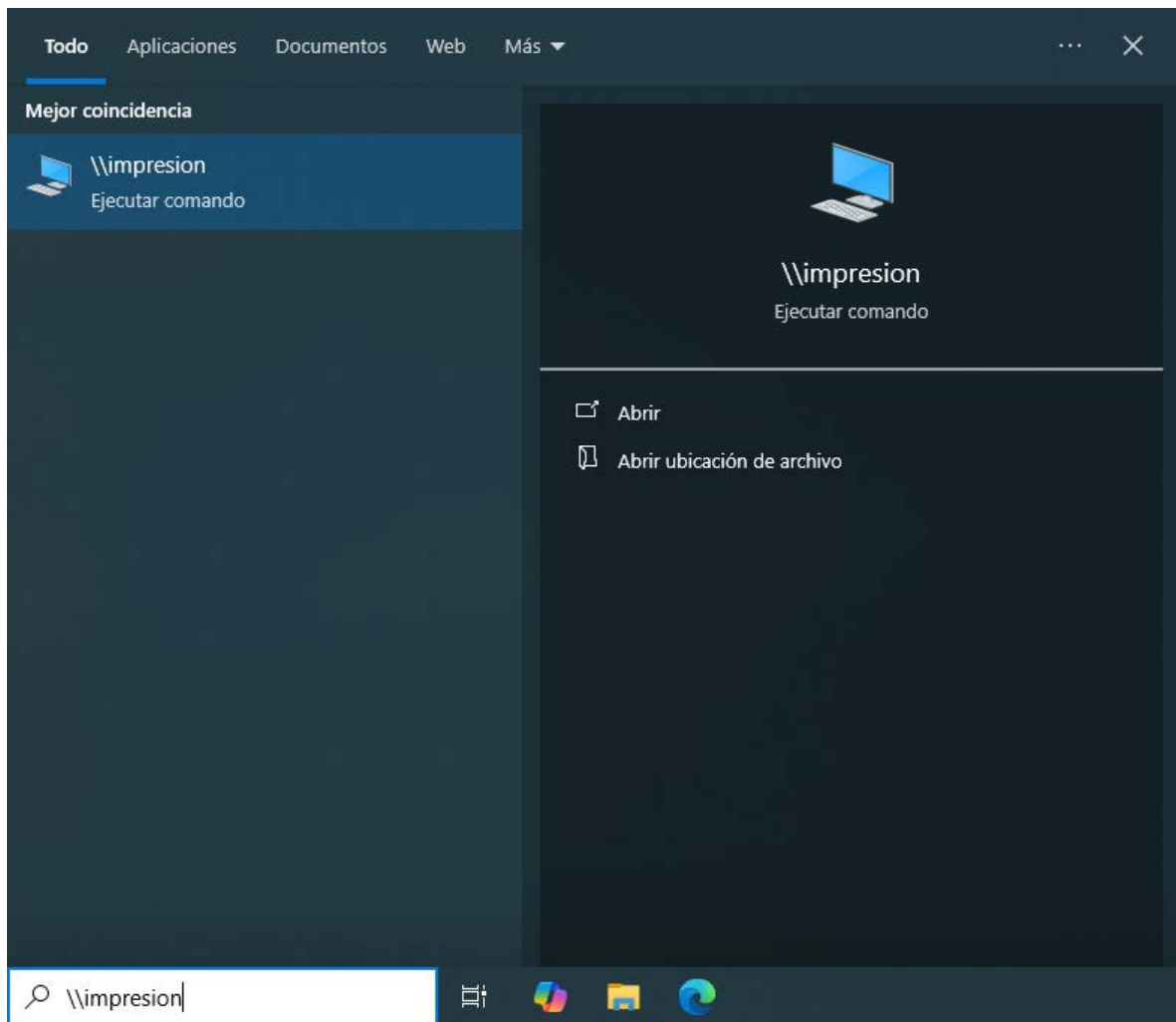
Es recomendable que el servidor de impresión, como los equipos cliente, cuenten con credenciales de usuario compatibles o una cuenta en común para garantizar un proceso de autenticación adecuado y evitar conflictos de permisos. Es importante mencionar que la autenticación es un mecanismo esencial de control de acceso lógico, pues permite verificar la identidad de los usuarios antes de otorgar acceso a los recursos compartidos en la red local (Silberschatz *et al.*, 2022).

Por consiguiente, la conexión y configuración de los equipos autorizados responde a una necesidad operativa fundamentada en las buenas prácticas técnicas y normativas de la administración de redes y seguridad de la información.

En la Figura 3, se muestra la conexión del equipo cliente con el servidor.

### Figura 3

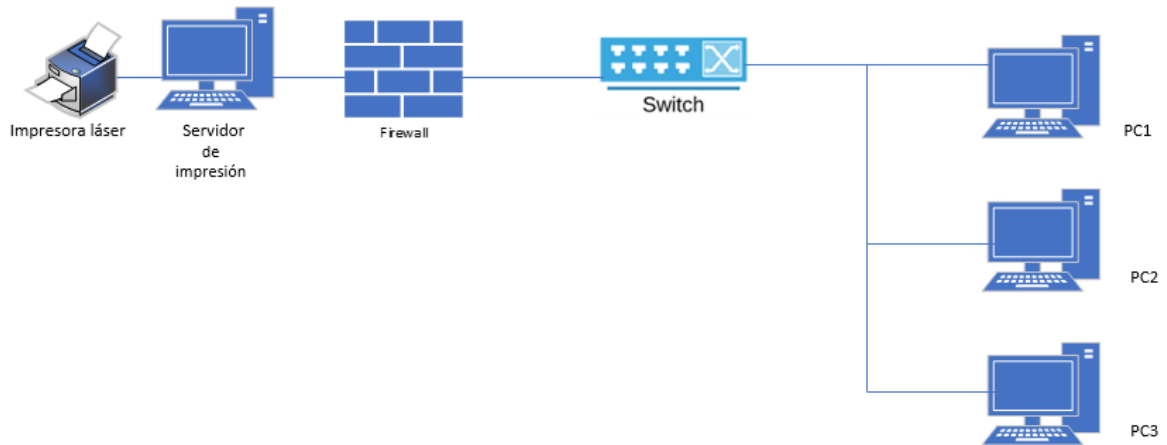
*Conexión cliente-servidor por medio de rutas UNC*



En la Figura 4, se puede apreciar el esquema básico del servidor de impresión.

**Figura 4**

*Diagrama básico de cliente-servidor*



### 3. RESULTADOS

En el año 2023, se atendieron 74 solicitudes; en el año 2024, se atendieron 77 solicitudes; y en el año 2025, se atendieron 44 solicitudes. Los rubros de estos reportes fueron: configurar impresora, impresora no responde/lenta, usuario no puede imprimir, cambio de tóner, atasco de papel, mantenimiento preventivo/correctivo. A continuación, en las Tablas 1, 2, 3 y 4 y en las Figuras 5, 6, 7 y 8, se presentan los datos referentes a dichos servicios:

**Tabla 1**

*Servicios atendidos en el año 2023*

Tipo de servicios	No. de servicios
Configurar impresora	15
Impresora no responde/lenta	15
Usuario no puede imprimir	23
Cambio de tóner	5
Atasco de papel	11
Mantenimiento preventivo/correctivo	5
<b>Total</b>	<b>74</b>

**Figura 5**

*Gráfico por rubro atendido en el año 2023*



**Tabla 2**

*Servicios atendidos en el año 2024*

Tipo de servicios	No. de servicios
Configurar impresora	19
Impresora no responde/lenta	18
Usuario no puede imprimir	25
Cambio de tóner	4
Atasco de papel	8
Mantenimiento preventivo/correctivo	3
<b>Total</b>	<b>77</b>

**Figura 6**

*Gráfico por rubro atendido en el año 2024*



**Tabla 3**

*Servicios atendidos en el año 2025*

Tipo de servicios	No. de servicios
Configurar impresora	8
Impresora no responde/lenta	9
Usuario no puede imprimir	10
Cambio de tóner	5
Atasco de papel	6
Mantenimiento preventivo/correctivo	6
<b>Total</b>	<b>44</b>

**Figura 7**

*Gráfico por rubro atendido en el año 2025*



**Tabla 4**

*Servicios atendidos del 2023 al 2025*

Año	No. de servicios
2023	74
2024	77
2025	44

## Figura 8

Gráfico de los servicios totales del 2023 al 2025



Con base en la información presentada en las gráficas y tablas que se mostraron, se puede observar la distribución y número de reportes atendidos correspondientes a los servicios de impresión. Este análisis cuantitativo permite identificar cuáles son las incidencias más recurrentes referentes a los servicios de soporte técnico solicitados para impresoras.

Los datos reflejan que la categoría o tipo de servicio solicitado con mayor frecuencia corresponde a la falla reportada por el usuario como "Usuario no puede imprimir", lo que sugiere posibles problemas de pérdida de conexión entre la computadora y la impresora conectada a la red local, deficiencias en la configuración de la red, errores relacionados a los controladores (*drivers*), permisos, etc. Tras la implementación del servidor de impresión, se logró reducir en gran medida los reportes referentes a impresoras.

Por otro lado, la configuración de direcciones IP en impresoras evidencia la necesidad de una administración adecuada de la red, especialmente en entornos donde se utiliza el direccionamiento IP estático, pues una planificación eficiente de este tipo de esquema puede contribuir a disminuir conflictos y desconexiones.

Por último, los servicios de soporte técnico referente al mantenimiento preventivo demuestran la importancia de estas acciones, ya que están orientadas a prolongar la vida útil de las impresoras y, por consiguiente, a la reducción de las fallas referentes a los mantenimientos correctivos.

## 4. CONCLUSIONES

La implementación de un servidor de impresión permitió garantizar la continuidad del servicio, evitando interrupciones en los procesos administrativos y operativos. Como resultado, se redujo de manera significativa el número de solicitudes de soporte técnico relacionados a la reconfiguración de impresoras,

lo que ha permitido que los tiempos de atención sean más breves y también que se liberen recursos del área de Infraestructura y Soporte Técnico para atender otras incidencias prioritarias.

Esta estrategia ha fortalecido la seguridad de las impresoras al centralizar la administración y aplicar listas de control de acceso (ACL) adecuados, de manera conjunta. Se ha promovido el uso de *software* libre para aprovechar las ventajas en términos de reducción de costos de licenciamiento, independencia tecnológica y flexibilidad operativa.

De igual manera, el reutilizar equipos de cómputo que estaban destinados a baja o desecho otorgó una segunda oportunidad a la vida útil de los equipos, fomentando de esta manera la reducción de residuos electrónicos y alinear prácticas referentes tanto a la sostenibilidad tecnológica como al aprovechamiento responsable de los recursos institucionales.

Como parte de la mejora continua, se contempla proponer la implementación de servidores de impresión por departamento, gestionados en su totalidad mediante *software* libre, incluyendo sistema operativo, *suite* ofimática y *firewall*. Esta propuesta permitirá una administración descentralizada, con mayor control, mejor eficiencia operativa y adaptable a las necesidades específicas de cada área.

Por último, se plantea trabajar con el área responsable en la DGTIC para la asignación de direccionamiento IP segmentado o no homologado, con el fin de fortalecer la organización del direccionamiento de red y mejorar la seguridad interna.

## REFERENCIAS

- Andrews, J. (2019). *CompTIA A+ guide to IT technical support (10th ed.)*. Cengage Learning.
- International Organization for Standardization. (2022). *Information security, cybersecurity and privacy protection — Information security management systems — Requirements (ISO/IEC 27001:2022)*. ISO.
- Kurose, J. F., & Ross, K. W. (2021). *Computer networking: A top-down approach (8th ed.)*. Pearson.
- Patterson, D. A., & Hennessy, J. L. (2017). *Computer organization and design: The hardware/software interface (5th ed.)*. Morgan Kaufmann.
- Vermaat, M.E., Sebok, S., Freund, S.M., Campbell, J., & Frydenberg, M. (2018). *Discovering computers (16<sup>a</sup> ed.)*. Cengage.
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2022). *Operating system concepts (10th ed.)*. Wiley.
- Stallings, W., & Brown, L. (2018). *Computer security: Principles and practice (4th ed.)*. Pearson.
- Stallings, W., & Brown, L. (2022). *Computer security: Principles and practice (5th ed.)*. Pearson.
- Whitman, M. E., & Mattord, H. J. (2021). *Principles of information security (7th ed.)*. Cengage Learning.

## ANEXO A. CARACTERÍSTICAS DEL *HARDWARE* DEL SERVIDOR

### Equipo de escritorio

El equipo corresponde a una computadora personal de escritorio, que tiene la función de administrar y gestionar el servicio de impresión, por consiguiente, no requiere de un alto poder de procesamiento ni recursos avanzados de *hardware*. Es importante señalar que el equipo ya había cumplido su ciclo de vida útil conforme a los criterios institucionales de renovación tecnológica. No obstante, con el objetivo de optimizar el aprovechamiento de los recursos, reducir la generación de residuos electrónicos y mitigar los efectos de la obsolescencia programada, se determinó su reacondicionamiento y reutilización para implementarlo al servidor de impresión. A continuación, se presentan las características del equipo en la Tabla 5:

**Tabla 5**

*Especificaciones físicas del equipo de escritorio*

Especificaciones del equipo	
Procesador	Intel® Core (™) i5-4460 CPU @ 3.20 GHz 3.20 GHz
RAM Instalada	16.0 GB (15.9 GB usable)
Tipo de sistema	Sistema operativo de 64 bits (procesador basado en x64)

Impresora HP LaserJet Pro 400 Serie M401

Es una impresora láser monocromática (blanco/negro), diseñada para trabajo de oficina moderado. Dado que la mayoría de los documentos generados corresponden a trámites administrativos e información interna, no se requiere salida a color. En consecuencia, utilizar una impresora monocromática para el servidor de impresión constituye una decisión técnicamente viable, económicamente eficiente y congruente con las políticas institucionales de uso responsable de los recursos.

A continuación, en la Tabla 6, se presentan las características de la impresora:

**Tabla 6**

*Características físicas de la impresora láser B/N*

Impresora HP Laser Jet Pro 400 serie M401	
Tecnología de impresión	Láser
Calidad de impresión en negro (óptima)	Hasta 1200 x 1200 ppp
Pantalla	LCD de 2 líneas (texto y gráficos)
Capacidad inalámbrica	No
Conectividad, estándar	1 USB 2.0 de alta velocidad 1 red Ethernet 10/100/1000T
Preparado para red	Estándar (Gigabit Ethernet incorporado)

## ANEXO B. SELECCIÓN DE *SOFTWARE* BASADO EN PRUEBAS DE USO

### Software de ofimática

Se determinó la necesidad de instalar en el servidor de impresión una *suite* ofimática para atender requerimientos operativos específicos. Para tal efecto, se evaluaron tres soluciones gratuitas de ofimática: LibreOffice, FreeOffice y Documentos de Google (incluidas las herramientas Hojas de cálculo y Presentaciones). La evaluación consideró criterios de funcionalidad, compatibilidad con formatos estándar, requerimientos técnicos y viabilidad de implementación en el entorno institucional. A continuación, se muestran las características más significativas de los aplicativos ofimáticos considerados a usarse en el servidor de impresión en la Tabla 7:

**Tabla 7**

*Comparativa de aplicaciones de ofimática*

Característica	LibreOffice <a href="https://es.libreoffice.org/">https://es.libreoffice.org/</a>	Free Office <a href="https://www.freeoffice.com/es/">https://www.freeoffice.com/es/</a>	Google Docs/Sheets/Slides <a href="https://gmail.com">https://gmail.com</a>
Modelo	Software de escritorio	Software de escritorio	Software basado en la nube
Costo	Gratis (código abierto)	Gratis (propietario)	Gratis (con opción de paga para Workspace)
Colaboración en tiempo real	No	No	Sí
¿Se puede usar sin Internet?	Sí	Sí	Limitado en modo <i>offline</i>
Compatible con MS Office	Sí, alta compatibilidad	Sí, alta compatibilidad	Sí, pero limitado
Automatización/Macros	Sí, alto	Sí, medio	Sí, medio
Extensiones/Plugins	Sí, muchos	Sí, pocos	Integrado con <i>apps</i> web
Privacidad de datos	Sí, local	Sí, local	En la nube

### Pruebas realizadas en aplicaciones de ofimática

Para poder determinar qué opción resultaba más conveniente, se evaluaron diversos factores y se llevaron a cabo pruebas básicas de funcionamiento. En una primera etapa, se consideraron las características físicas del servidor de impresión (Tabla 1), dado que estas pueden influir directamente al procesar y gestionar archivos ofimáticos. Las pruebas realizadas fueron las siguientes:

- Abrir un documento previamente creado en Microsoft Office y observar si el formato cambiaba: En este caso, LibreOffice presentó mejor compatibilidad con archivos .docx sin alterar el contenido de los mismos.

- Abrir un trabajo o archivo de más de 50 páginas: En este punto se analizó cuál fue la opción a la que le tomó menos tiempo abrir y mostrar el contenido completo del documento. FreeOffice dio mejores resultados, LibreOffice tardó unos segundos más y Documentos de Google presentó una gran variación de tiempos según el navegador y la velocidad de la conexión a Internet.
- Pruebas de trabajo colaborativo: En este rubro, Documentos de Google fue, sin lugar a duda, la mejor opción, ya que el *software* está enfocado al trabajo en línea mientras que las otras aplicaciones están diseñadas para trabajar de manera local.
- Prueba de impresión: Aquí, LibreOffice presentó un mejor control y compatibilidad de impresión con Windows.
- Prueba de usabilidad: Para esta última prueba, se tomó en cuenta que el entorno gráfico y el comportamiento del aplicativo fueran similares al de Microsoft Office, pues dicho *software* representa la principal opción de los usuarios en cuanto a tareas de ofimática se refiere, por lo que era necesario buscar una herramienta que no representara un gran cambio frente a la experiencia de uso del aplicativo de Microsoft. LibreOffice resultó ser la opción más cómoda y familiar en este apartado, gracias a sus opciones de personalización de interfaz que permiten que esta sea prácticamente igual a la de Word.

En la Tabla 8 se presentan los resultados de las pruebas de manera concreta:

**Tabla 8**

*Comparativo de pruebas*

Prueba	LibreOffice	FreeOffice	Google Docs/Sheets/Slides
Compatibilidad	✓	-	-
Velocidad de apertura de archivo	-	✓	-
Trabajo colaborativo	-	-	✓
Prueba de impresión	✓	-	-
Usabilidad	✓	-	-

Derivado de las pruebas realizadas y del análisis técnico y administrativo, se concluyó que la mejor opción es LibreOffice, pues representa la alternativa más viable para su uso e implementación en el servidor de impresión. La decisión se fundamenta en su carácter de *software* libre, ausencia de costos de licenciamiento, compatibilidad adecuada con formatos estándar utilizados en la dependencia y su funcionamiento sin necesidad de conexión a Internet. Por consiguiente, su bajo requerimiento de recursos técnicos resulta congruente con las capacidades del equipo reutilizado, garantizando estabilidad operativa sin comprometer el rendimiento del servidor de impresión.

## Firewall

Con el fin de fortalecer la seguridad del servidor de impresión, se decidió instalar un *firewall* adicional. Cabe mencionar que este no reemplazará al *firewall* integrado de Windows 10 Pro, sino que trabajará en conjunto para robustecer la protección del sistema.

El *firewall* puede implementarse mediante *hardware*, *software* o una combinación de ambos. Su función principal consiste en monitorear el tráfico de red y controlar el acceso al servidor, actuando como un sistema que aplica la política de seguridad de la red privada y la Internet (Cuenca, 2016). A través del *firewall*, es posible generar listas de control de acceso que definan qué usuarios pueden utilizar el servicio de impresión.

Para la implementación del *firewall* se consideraron tres opciones: Zone Alarm, TinyWall y Comodo Firewall. Con el objetivo de seleccionar la solución más adecuada, se llevaron a cabo pruebas de funcionalidad, compatibilidad y desempeño, así como comparativos técnicos entre las alternativas seleccionadas. En la Tabla 9 se muestra la comparativa de las tres opciones consideradas para instalación en el servidor de impresión.

**Tabla 9**

*Comparativa del firewall (cortafuegos)*

Características	ZoneAlarm <a href="https://www.ZoneAlarm.com/">https://www.ZoneAlarm.com/</a>	TinyWall <a href="https://tinywall.pados.hu/">https://tinywall.pados.hu/</a>	Comodo Firewall <a href="https://www.comodo.com/?af=7639">https://www.comodo.com/?af=7639</a>
Modelo	Software de escritorio	Software de escritorio	Software de escritorio
Costo	Versión gratuita/ Versión paga	Versión 100% gratuita	Versión gratuita/ Versión paga
Requisitos de <i>hardware</i>	Uso medio de RAM y CPU	Uso mínimo de RAM y CPU	Uso medio de RAM y CPU
Nivel de usabilidad	Fácil	Medio	Avanzado
Perfil de usuario	Intermedio	Principiante	Avanzado
Nivel de protección	Seguro	Seguro	Muy seguro

## Pruebas realizadas para la elección del *firewall*

Para seleccionar el mejor *firewall* a implementarse en el servidor de impresión, se siguieron la siguiente metodología y criterios: mismo equipo y sistema operativo, un *firewall* a la vez y uso de la configuración por defecto. Las pruebas realizadas consistieron en pruebas de control de tráfico entrante/saliente, rendimiento de los recursos de *hardware* del equipo, desactivación del *firewall* sin autorización y usabilidad, así como experiencia previa del usuario. En la Tabla 10 se observa el comparativo de las pruebas realizadas a los tres *firewalls* considerados para instalar en el servidor de impresión.

**Tabla 10**

*Comparativo de pruebas*

Prueba	Zone Alarm	TinyWall	Comodo Firewall
Pruebas de tráfico entrante/saliente	Alto	Alto	Muy alto
Rendimiento	Medio	Excelente	Bajo
Usabilidad	Alto	Medio	Alto
Desactivación	Alto	Medio	Muy alto

### **Elección del *firewall***

Se determinó que ZoneAlarm es la solución más adecuada para su implementación en el servidor de impresión con base en los siguientes criterios: presenta un buen equilibrio con respecto a la seguridad, rendimiento y usabilidad; es totalmente compatible con Windows 10 Pro, garantizando una integración estable con el sistema operativo y con el *firewall* nativo, permitiendo un esquema de seguridad por capas, sin conflictos de *software*.

La interfaz de ZoneAlarm es intuitiva y permite configurar listas de control de acceso y reglas de seguridad de manera sencilla, facilitando la administración del servidor y reduciendo el riesgo de errores de configuración por parte del personal técnico. El nivel de seguridad incluye monitoreo del tráfico de la red, realización de bloqueos de conexiones no autorizadas y emisión de alertas en tiempo real, permitiendo que sólo los usuarios autorizados puedan acceder al servicio de impresión. Además, mantiene un impacto moderado en el rendimiento del servidor, asegurando que el servicio de impresión no se vea afectado. Al ser una instalación ligera y de fácil gestión, se ajusta a los recursos del servidor reutilizado, evitando la necesidad de actualizar *hardware* o generar costos adicionales de licenciamiento. Por estas razones, la elección de ZoneAlarm proporciona una protección robusta y confiable al servidor de impresión, cumpliendo con los principios de seguridad y eficiencia operativa.

### **Antivirus**

El objetivo de instalar un antivirus en el servidor de impresión es añadir una capa adicional de seguridad para proteger los archivos y el sistema contra amenazas que puedan comprometer su funcionamiento. Los antivirus emplean bases de datos de firmas de *malware* conocidas para identificar *software* malicioso y también son capaces de analizar el comportamiento del sistema en busca de actividades sospechosas (Arango, 2023).

Para la implementación en el servidor, se consideraron tres opciones gratuitas: Avira, AVG y Avast. Para la implementación del antivirus, la evaluación se basó en los siguientes criterios: compatibilidad con el sistema operativo Windows 10 Pro, consumo bajo de recursos para garantizar que el antivirus no afecte el rendimiento del servidor, facilidad tanto de configuración como de actualización y, por último, la eficiencia en la detección de amenazas y cobertura frente a *malware* conocido y emergente. En la Tabla 11 se muestra una comparativa de los tres antivirus considerados a instalar en el servidor de impresión:

**Tabla 11**

*Comparativa de antivirus*

Característica	AVIRA <a href="https://www.avira.com/es">https://www.avira.com/es</a>	AVG <a href="https://www.avg.com/es-mx/homepage#pc">https://www.avg.com/es-mx/homepage#pc</a>	AVAST <a href="https://www.avast.com/es-mx/index#pc">https://www.avast.com/es-mx/index#pc</a>
Protección contra <i>malware</i>	Bueno	Excelente	Excelente
Rendimiento	Excelente	Bueno	Bueno
Funciones extras	Excelente	Bueno	Bueno
Versión gratuita	Excelente	Excelente	Excelente
Privacidad	Excelente	Bueno	Bueno
Usabilidad	Excelente	Excelente	Bueno

### Elección del antivirus

Finalmente, se seleccionó Avira para proteger el servidor de impresión debido a su bajo consumo de recursos, compatibilidad con Windows 10 Pro y facilidad de gestión, además de que ofrece actualizaciones automáticas de su base de datos de *malware*, lo que garantiza la protección frente a nuevas amenazas de manera continua y confiable.

# Conector de gestión de pagos digitales en línea

*Online digital payment management connector*

## Información del reporte:

Licencia Creative Commons



El contenido de los textos es responsabilidad de los autores y no refleja forzosamente el punto de vista de los dictaminadores, o de los miembros del Comité Editorial, o la postura del editor y la editorial de la publicación.

Para citar este reporte técnico:

Cendejas Cervantes, N. (2026). Conector de gestión de pagos digitales en línea. *Cuadernos Técnicos Universitarios de la DGTIC*, 4(2), páginas (99 - 107). <https://doi.org/10.22201/dgtic.30618096e.2026.4.2.177>

## Nidia Cendejas Cervantes

Dirección General de Cómputo y de  
Tecnologías de Información y Comunicación  
Universidad Nacional Autónoma de México

[nidia@unam.mx](mailto:nidia@unam.mx)

ORCID: 0009-0001-2929-2575

## Resumen

La digitalización de servicios institucionales ha incrementado la necesidad de integrar mecanismos seguros y eficientes para el procesamiento de algunos de los servicios que se pueden ofrecer, entre ellos, el pago en línea. En este contexto, se implementó un punto de acceso para la gestión de transacciones electrónicas, con el propósito de optimizar los procesos administrativos.

Para implementar el punto de acceso, la metodología se estructuró en cuatro etapas: diseño de la arquitectura basada en servicios web, configuración del entorno de comunicación, instalación de componentes en el servidor institucional, así como ejecución de pruebas funcionales y de seguridad. Asimismo, se estableció la conexión estructurada al *webservice* proporcionado por la Tesorería del Patronato Universitario mediante protocolos compatibles con sus especificaciones técnicas.

Los resultados mostraron una reducción en los tiempos de procesamiento de pagos, así como en el incremento de inscripciones con pago en línea y una disminución significativa en errores manuales de conciliación. Como principal aportación, se obtuvo un punto de acceso que permite la integración estandarizada con el *webservice*. Una limitación del trabajo es que la solución depende de la disponibilidad y especificaciones técnicas del servicio externo, por lo que los cambios en éste podrían requerir actualizaciones en el conector.

**Palabras clave:** Cifrado de datos, servicios web estandarizados, automatización de pago en línea, validación transaccional.

### Abstract

*The digitalization of institutional services has increased the need to integrate secure and efficient mechanisms for processing some of the services that can be offered, including online payment. In this context, an endpoint was implemented for the management of electronic transactions, with the purpose of optimizing administrative processes.*

*For the implementation of the endpoint, the methodology was structured into four main stages: design of a web service-based architecture, configuration of the communication environment, installation of components on the institutional server, and execution of functional and security testing. Also, a structured connection was established to the webservice provided by the Tesorería del Patronato Universitario, using protocols compatible with its technical specifications.*

*The results showed a reduction in payment processing times, as well as an increase in registrations with online payment and a significant decrease in manual reconciliation errors. The main contribution of this work is an access point that enables standardized integration with the web service. A limitation of the work is that the solution depends on the availability and technical specifications of the external service, so changes to it could require updates to the connector.*

**Keywords:** Data encryption, standarized webservice, online payment automation, transactional validation.

## 1. INTRODUCCIÓN

La transformación digital ha exigido la modernización de procedimientos para la gestión administrativa y financiera de las instituciones, particularmente en el procesamiento de pagos electrónicos. En los últimos años, la adopción de arquitecturas basadas en servicios y el uso de interfaces de programación de aplicaciones (API) han mejorado la interoperabilidad entre sistemas con características diferentes, fortaleciendo la seguridad en las transacciones digitales.

En el ámbito de la seguridad de APIs, el *OWASP API Security Top 10* (OWASP Foundation, 2023) ha identificado riesgos críticos tales como la exposición excesiva de datos, la asignación masiva y el consumo inseguro de APIs, los cuales son especialmente relevantes en sistemas de pago en línea. Adicionalmente, la integración con sistemas de pago externos requiere mecanismos robustos de autenticación y autorización; estándares como OAuth 2.0 permiten gestionar el acceso de manera controlada.

Investigaciones previas en materia de seguridad de interfaces de aplicaciones señalan que los puntos de acceso seguros contribuyen a la reducción de vulnerabilidades y a la optimización de tiempos de respuesta, al hacer más ágil el proceso de validación de transacciones (Behl & Behl, 2020; Stuttard & Pinto, 2021).

En la Dirección General de Cómputo y de Tecnologías de la Información y Comunicación (DGTIC), se identificó la necesidad de automatizar el pago en línea de eventos (cursos y diplomados), ya que el

proceso previo dependía de validaciones manuales y conciliaciones administrativas que generaban retrasos e inconsistencias.

El objetivo principal del proyecto presentado en este reporte técnico fue integrar un mecanismo automatizado y confiable para el pago en línea que operara dentro de la infraestructura institucional, sin depender de procesos manuales externos.

La integración de sistemas internos con servicios externos requirió establecer una comunicación segura, controlada y estructurada. La implementación de un punto de acceso como capa intermedia ha permitido desacoplar la lógica institucional y abstraer el *webservice* externo, facilitando el flujo de información entre ambos sistemas. De acuerdo con el *Open Web Application Security Project* (OWASP Foundation, 2021), las interfaces de programación de sistemas representan uno de los principales riesgos en aplicaciones actuales, por lo que su adecuada implementación ayuda significativamente a la protección de la información.

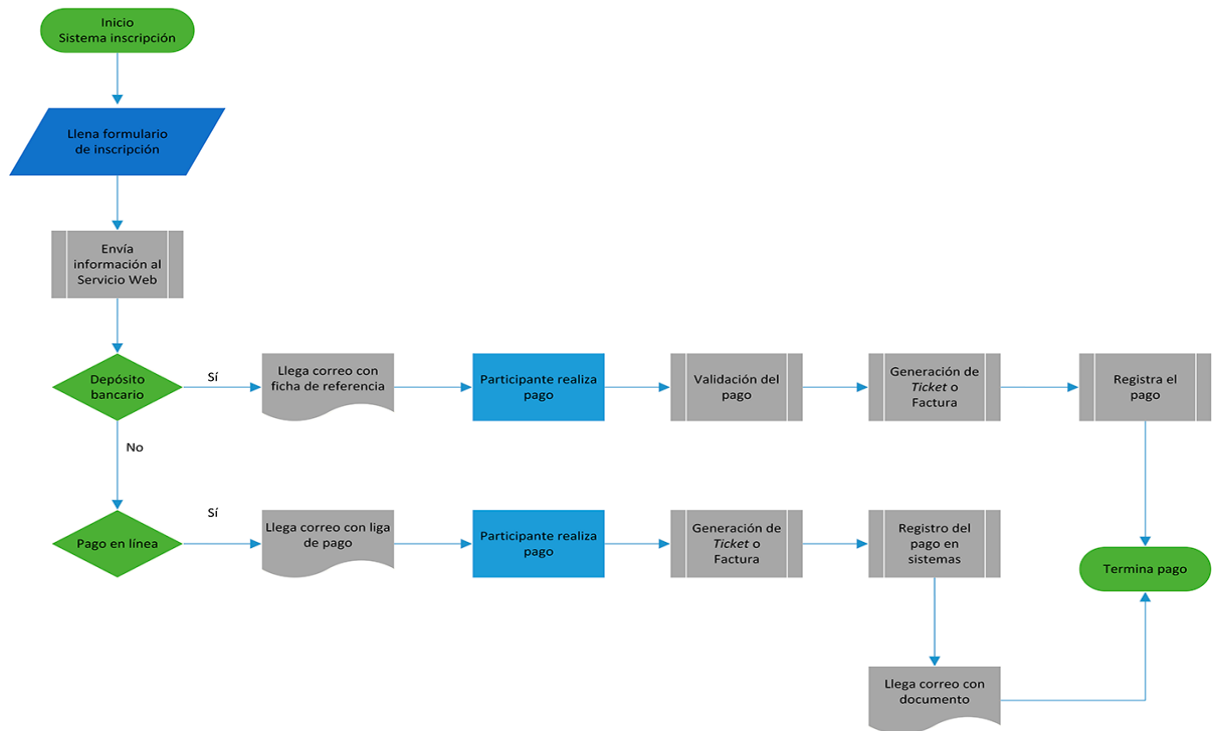
## 2. DESARROLLO TÉCNICO

El punto de acceso se estableció como una capa intermedia entre el sistema institucional y el *webservice* de pago en línea de la Tesorería. Su función principal es desacoplar la lógica del sistema interno y abstraer la complejidad del servicio externo, centralizando procesos como la validación de datos, la autenticación y el control de la transacción. Esta arquitectura de tres capas (presentación, lógica de negocio e integración) permite separar responsabilidades, facilita el mantenimiento y aísla la lógica de negocio de posibles cambios en el servicio externo (Martin, 2020).

El proyecto se organizó en dos etapas principales: la definición de la arquitectura y el desarrollo funcional del flujo transaccional. El flujo general del proceso se muestra en la Figura 1.

**Figura 1**

*Proceso para realizar un pago en línea*



Como se muestra en la Figura 1, el proceso de pago inicia cuando el usuario se registra en el sistema de inscripciones. Durante el registro, el sistema captura los datos necesarios (nombre, correo, monto) y genera una orden de pago. Posteriormente, estos datos son enviados al *webservice* de la Tesorería a través del punto de acceso, el cual actúa como intermediario. Este punto valida la integridad y el formato de la información, es decir, la estructura según los parámetros requeridos, y la transmite al servicio externo. Una vez recibida la confirmación, el sistema actualiza el estatus de la inscripción y notifica al usuario.

El proceso se divide en dos líneas según el método de pago: en el depósito bancario, el sistema genera una referencia de pago que es enviada por correo electrónico, el usuario realiza la transacción y, posteriormente, se ejecuta la validación mediante el punto de acceso. Mientras que en el pago en línea, el punto de acceso lleva a cabo la comunicación directa con el servicio de pago, valida la transacción en tiempo real y garantiza la integridad de la información.

En ambos procesos, el punto de acceso centraliza la lógica de validación y el control de la transacción, permitiendo el registro automático del pago y la generación de comprobantes (*ticket* o *factura*).

## 2.1 METODOLOGÍA

La metodología se estructuró en cinco etapas: análisis, diseño, configuración, instalación y pruebas.

## 2.1.1 ANÁLISIS

Se analizaron las especificaciones técnicas del *webservice* de la Tesorería para comprender su funcionamiento, restricciones y mecanismos de interacción. Se identificaron los métodos disponibles para la gestión de transacciones, así como los parámetros obligatorios: referencia de pago, monto, concepto del pago y datos del usuario.

Se examinó la estructura de los mensajes de intercambio (formato JSON), y los códigos de respuesta emitidos por el servicio, lo que permitió clasificar las salidas del sistema en transacciones exitosas, rechazos operativos y errores técnicos. La correcta validación de los datos de entrada es esencial para la seguridad de las API, ya que un manejo incorrecto puede generar vulnerabilidades (Stuttard & Pinto, 2021; Acar *et al.*, 2018). Adicionalmente, el *webservice* requiere el envío de datos mediante el protocolo XML-RPC (UTICT, 2024).

Derivado del análisis, se observó la necesidad de incorporar mecanismos de validación que garantizaran integridad y consistencia, dado que la información incorrecta en los parámetros establecidos ocasionaría la interrupción del proceso.

## 2.1.2 DISEÑO

Con base en el análisis, se definieron los siguientes criterios de validación que todo pago debe cumplir antes de ser enviado al servicio. La Tabla 1 muestra estos criterios, definidos según las especificaciones de la Tesorería, que establece formatos obligatorios para campos de Referencia CHAR (20), Importe (DECIMAL (22,6) y la exigencia de que todos los parámetros de entrada deben estar en mayúsculas (UTICT, 2024).

**Tabla 1**

*Criterios de validación del punto de acceso*

Campo	Criterio de validación	Acción en caso de incumplimiento
Referencia de pago	Formato alfanumérico de 20 caracteres, no vacío	Rechazar solicitud, código 400
Monto	Número decimal positivo, mínimo \$10 MXN, Máximo \$50,000 MXN	Rechazar solicitud, código 400
Concepto	Texto de hasta 100 caracteres, sin caracteres especiales peligrosos ( < > & " ' )	Sanitizar la cadena y validar, rechazar si persiste el error
Correo del usuario	Formato de correo electrónico válido (usuario@dominio)	Rechazar solicitud, código 400
Token de autenticación	Debe coincidir con el valor almacenado en configuración	Rechazar solicitud, código 400

Estos criterios se implementaron antes de enviar la solicitud al *webservice* externo, reduciendo así transacciones fallidas por errores de formato.

Se elaboró el flujo transaccional considerando tres escenarios: éxito, en el que la operación es válida y confirmada por el servicio; rechazo, cuando la transacción no cumple con los criterios mínimos establecidos

por el *webservice*; y error técnico, cuando existe alguna falla en la comunicación y/o disponibilidad en el servicio. Ese modelado permitió definir la lógica de control del punto de acceso.

El punto de acceso contempla un esquema de manejo de errores que distingue tres categorías:

- Errores de validación (códigos 400): ocurren cuando los parámetros enviados no cumplen con los formatos esperados.
- Errores de autenticación (códigos 401): se visualizan cuando el *token* de acceso no es válido.
- Errores de comunicación (códigos 500 o de tiempo de espera): ocurren cuando el *webservice* no responde en un tiempo máximo de 30 segundos o hay fallas de red.

### 2.1.3 CONFIGURACIÓN

Se desarrolló un módulo en PHP bajo el principio de separación de responsabilidades, el cual establece que cada componente del sistema debe encargarse de una función específica, reduciendo la complejidad, facilitando el mantenimiento y mejorando la seguridad del software (Martin, 2020). La elección de PHP se basó en su amplia adopción en entornos institucionales, su madurez para integraciones de tipo transaccional y la disponibilidad de extensiones como cURL para consumo de servicios web. En este sentido, se diseñó el punto de acceso como un componente independiente.

El punto de acceso implementa un esquema de autenticación basado en *token* fijo, compartido entre el sistema institucional y el *webservice*. Cada solicitud debe incluir en el encabezado HTTP un *token* de acceso previamente acordado, el cual es validado por el servicio antes de procesar la transacción.

Según la documentación técnica de la Tesorería, cada dependencia debe tramitar un usuario y una contraseña que son únicos para acceder a los servicios web, los cuales se envían como parte de los parámetros de inicialización junto con la URL del servicio. También es necesario registrar las direcciones IP desde las cuales se consumirán los servicios (UTICT, 2024).

En este aspecto, se incorporó la validación de entrada de la información para asegurar que los datos recibidos cumplieran con los tipos de datos y restricciones definidos, evitando algún tipo de inconsistencia en la comunicación con el *webservice*. Esta validación previene el envío de datos malformados que podrían ser rechazados por el servicio externo o explotados como vectores de ataque (Stuttard & Pinto, 2021). Se emplearon consultas parametrizadas para prevenir ataques de inyección SQL (OWASP Foundation, 2021) y se implementó un registro estructurado de eventos en bitácoras para monitorear y detectar algún tipo de incidente de seguridad de manera temprana (Behl & Behl, 2020).

Para la integración con el *webservice*, se empleó la extensión cURL bajo protocolo seguro, garantizando así que todos los datos sensibles viajen cifrados durante la transmisión, además de utilizar JSON como formato de seriación de datos. Se configuraron tiempos de espera controlados y mecanismos de manejo de excepciones para garantizar la estabilidad transaccional.

### 2.1.4 INSTALACIÓN

El sistema fue desplegado en una máquina virtual proporcionada por el Centro de Datos de la DGTIC, habilitando únicamente el tráfico por HTTPS, con el cifrado TLS para proteger los datos durante la transmisión (Behl & Behl, 2020). Se configuró un certificado digital válido, logrando que todas las

solicitudes al punto de acceso se realizaran a través de conexiones seguras. El uso de infraestructura virtualizada garantiza disponibilidad, respaldo y capacidades de recuperación ante contingencias, sin requerir inversión adicional en hardware dedicado.

Se estableció un entorno basado en software libre: Linux, Apache, PHP, PostgreSQL y Laravel. Estas tecnologías se seleccionaron porque combinan la madurez, estabilidad y amplio soporte, lo que reduce costos de licenciamiento y facilita el mantenimiento interno. Se pueden observar los componentes utilizados en la Tabla 2.

**Tabla 2**

*Componentes tecnológicos utilizados para la implementación del punto de acceso*

Componente	Descripción
Sistema operativo	Linux Debian
Servidor web	Apache HTTP Server 2.4
Lenguaje	PHP 8.1
Gestor de bases de datos	PostgreSQL 17.9
Framework de programación	Laravel
Extensión para consumo de servicios	cURL 7.68

Adicionalmente, se implementaron medidas de seguridad a nivel infraestructura, mediante la configuración de reglas de *firewall*, en este caso, con *IPTables*. Gracias a esto, se restringió el acceso a direcciones IP y puertos autorizados, permitiendo limitar la exposición de servicios y así reducir la posibilidad de ataques.

### 2.1.5 PRUEBAS

Se realizaron pruebas funcionales y de seguridad para validar el funcionamiento del punto de acceso (García-Peñalvo *et al.*, 2021). Se verificó el correcto envío, procesamiento y recepción de datos, así como la consistencia de las respuestas. También se evaluaron riesgos tales como la inyección de código SQL, la manipulación de parámetros y el reenvío de solicitudes.

Durante la integración del *webservice*, se observó que la arquitectura requería del cumplimiento estricto de parámetros y tipos de datos. Al enviar información incorrecta, en automático, se cancelaba la transacción, fortaleciendo los mecanismos de validación y reducción de algún tipo de riesgo.

Finalmente, se efectuaron pruebas de conciliación contable para validar la correspondencia tanto en registros internos como en confirmación del servicio externo, proporcionado por la Tesorería.

## 3. RESULTADOS

La implementación logró establecer una comunicación estable y segura con el servicio, el cual permitió el registro automatizado de transacciones y la confirmación inmediata del estatus de pago.

Para evaluar la efectividad de la solución, se analizaron los datos operativos de los años 2024 y 2025:

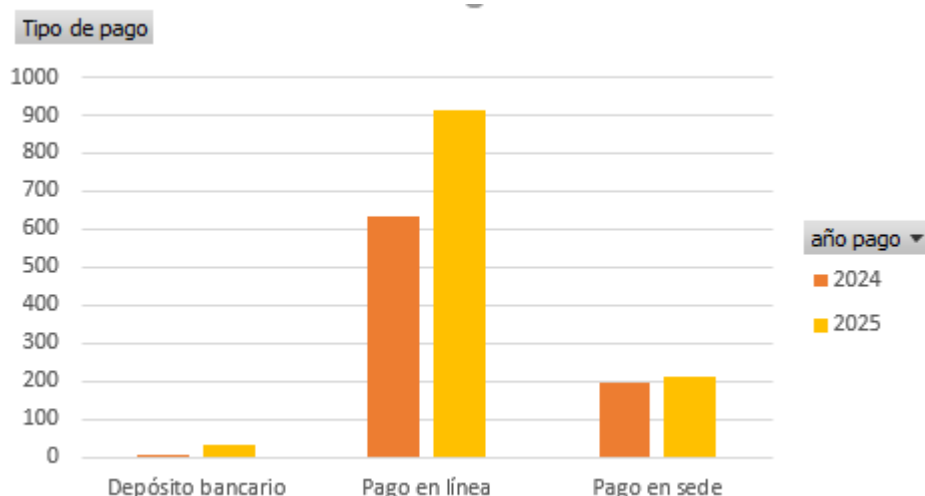
- Tiempo de procesamiento para pagos en línea: la mediana del tiempo de espera para la confirmación de pagos se redujo de 2.5 días (60 horas) en 2024 a 0.8 (19 horas) en 2025, lo que representa una disminución del 68%.
- Pagos sin conciliar: el porcentaje de transacciones que no lograron conciliarse automáticamente disminuyó del 26.5% en 2024 al 10.2% en 2025, una reducción del 61%.
- Generación automática de comprobantes (*ticket* o factura): se incrementó en un aproximado de 860 comprobantes en 2024 a 1,161 en 2025, lo que equivale a un aumento del 35%. Esto refleja una mayor adopción del sistema para la emisión de comprobantes fiscales.

También se observó una disminución significativa en errores de conciliación, debido a la generación automática de referencias únicas y a la sincronización directa con el sistema de pagos institucional.

Como resultado, se ha notado un incremento en la selección de tipo de pago por parte de los participantes que ingresan a los eventos, tal como se observa en la Figura 2:

**Figura 2**

*Tipo de pago para inscripción a cursos y diplomados*



Como se observa en la Figura 2, el pago en línea muestra una tendencia creciente en el periodo analizado, mientras que los métodos tradicionales (pago en sede y depósito bancario) presentan una disminución. En el último año registrado, el pago en línea representó el 54% del total de las transacciones, frente al 28% del pago en sede y el 18% del pago por depósito bancario. Esta distribución refleja que los usuarios prefieren la inmediatez y comodidad del pago electrónico sobre los métodos presenciales o diferidos. La disponibilidad de un punto de acceso seguro y confiable ha sido un factor clave para habilitar esta opción y diversificar los mecanismos de pago ofertados por la DGTIC.

El registro inmediato de los pagos en los ingresos facilitó la consulta en tiempo real de los ingresos percibidos por concepto de servicios y oferta académica.

## 4. CONCLUSIONES

La principal aportación de este trabajo es la implementación de un punto de acceso seguro y funcional que actúa como capa intermedia entre el sistema institucional de la DGTIC y el *webservice* de pagos de la Tesorería. Este conector ha permitido automatizar el ciclo de pago en línea, desde la generación de la orden hasta la validación y registro de la transacción.

Como resultados concretos, se logró reducir los tiempos de procesamiento, minimizar errores operativos asociados a la conciliación manual y aumentar la adopción del pago en línea entre los usuarios de los cursos y diplomados ofertados. La arquitectura basada en servicios facilita el mantenimiento y la evolución independiente de cada capa.

El alcance del trabajo se limita a la integración con el *webservice* de la Tesorería en su versión actual. Cambios futuros en las especificaciones del servicio externo podrían requerir ajustes en el conector. No obstante, el diseño modular del punto de acceso permite adaptarse a dichos cambios con un menor impacto.

## REFERENCIAS

- Acar, Y., Backes, M., Fahl, S., Kim, D., Mazurek, M. L., & Stransky, C. (2018). You get where you're looking for: The impact of information sources on code security. *Proceedings of the IEEE Symposium on Security and Privacy*, 289–305.
- Behl, A., Behl, K. (2020). *Cybersecurity and cyberwar: What everyone needs to know*. Oxford University Press.
- García-Peñalvo, F. J., Cruz-Benito, J., & Therón, R. (2021). *Seguridad en sistemas de información: tendencias y desafíos actuales*. Ediciones Universidad de Salamanca.
- Martin, R. C. (2020). *Arquitectura limpia: guía para especialistas en la estructura y el diseño del software*. Anaya Multimedia.
- OWASP Foundation. (2023). *OWASP API Security Top 10 2023*. OWASP. <https://owasp.org/API-Security>
- OWASP Foundation. (2021). *OWASP Top Ten 2021: The ten most critical web application security risk*. OWASP. <https://owasp.org>
- Stuttard, D., & Pinto, M. (2021). *El libro del hacker de aplicaciones web: Cómo descubrir y explotar fallos de seguridad* (2a ed.). Anaya Multimedia.
- UTICT. (2024). *Documentación de servicios web: Servicios Web y Servlet para la emisión de Facturas Digitales*. Unidad de Tecnologías de Información y Comunicación de la Tesorería, Universidad Nacional Autónoma de México.

# Implementación de una infraestructura de correo electrónico para pruebas con Exchange Server

## *Implementing an email infrastructure for testing with Exchange Server*

### Información del reporte:

Licencia Creative Commons



El contenido de los textos es responsabilidad de los autores y no refleja forzosamente el punto de vista de los dictaminadores, o de los miembros del Comité Editorial, o la postura del editor y la editorial de la publicación.

Para citar este reporte técnico:

Martínez Sierra, L. (2026). Implementación de una infraestructura de correo electrónico para pruebas con Exchange Server. *Cuadernos Técnicos Universitarios de la DGTIC*, 4 (2), páginas (108 - 123). <https://doi.org/10.22201/dgtic.30618096e.2026.4.2.172>

**Laura Martínez Sierra**

Dirección General de Cómputo y de Tecnologías de Información y Comunicación  
Universidad Nacional Autónoma de México

[laura.martinez@unam.mx](mailto:laura.martinez@unam.mx)

ORCID: 0009-0004-2850-2384

### Resumen

El correo electrónico es una herramienta esencial dentro de las dependencias educativas, ya que permite una comunicación rápida y eficaz entre los integrantes de dichas comunidades. Por lo anterior, se requiere que el servicio de correo se mantenga operativo durante el mayor tiempo posible; sin embargo, como en toda infraestructura, es necesario realizar actividades de mantenimiento periódicamente, lo que, en algunas ocasiones, obliga a realizar pausas en el servicio. Ante esta situación, se implementó un ambiente de pruebas que permitiera validar configuraciones antes de aplicarlas a producción para disminuir los tiempos de inactividad y prevenir los errores. La implementación se realizó a través de máquinas virtuales que simulaban el entorno productivo. Cada máquina virtual fue asignada a una función dentro de la infraestructura de correo, y cada una se instaló y configuró según sus requisitos. Al completar la instalación, se realizó la validación y verificación de configuraciones en el entorno de pruebas, las cuales, al ser exitosas, permitieron que se aplicaran al entorno productivo. El proyecto permitió la disminución de los tiempos de las actividades de mantenimiento, la detección temprana de problemas de *software* y, en consecuencia, la disminución de los tiempos de inactividad del entorno productivo para evitar afectar a los usuarios que hacen el uso del servicio de correo electrónico.

**Palabras clave:** Ambiente de pruebas, máquinas virtuales, servidor de correo, entorno productivo.

### Abstract

*Email is an essential tool within educational institutions, enabling fast and efficient communication among members of these communities. Therefore, it is crucial that the email service remains operational for as long as possible; however, as with any infrastructure, periodic maintenance is necessary, which sometimes causes service interruptions. To address this, a test environment was implemented to validate configurations before deploying them to production, thereby minimizing downtime and preventing errors. Implementation was carried out using virtual machines that simulated the production environment. Each virtual machine was assigned a role within the email infrastructure, and each one was installed and configured according to its specific requirements. Upon completion of the installation, the configurations were validated and verified in the test environment. Successful validation allowed for their deployment to the production environment. The project resulted in reduced maintenance time, early detection of software issues, and consequently, decreased downtime in the production environment.*

**Keywords:** Testing environment, virtual machines, mail server, production environment.

## 1. INTRODUCCIÓN

La comunicación dentro de una dependencia educativa requiere del intercambio de información a través de medios eficientes y rápidos para que los integrantes de la comunidad, conformada por alumnos, académicos, investigadores y trabajadores, puedan enviar y recibir información, intercambiar documentos, organizar mensajes, entre otras tareas. Ante esta necesidad, el servicio de correo electrónico institucional surge como un medio accesible y eficaz que permite la comunicación digital de manera sencilla y confiable.

Dicho servicio depende de una infraestructura conformada por un conjunto de elementos tecnológicos que permiten el envío y recepción de mensajes. Algunos de estos elementos son: los servidores de correo, dominios, DNS (Domain Name System), direcciones IP, protocolos de comunicación y autenticación, etc. La importancia del servicio de correo electrónico para una entidad ha llegado a ser tal, que la infraestructura que lo sostiene debe mantenerse disponible y funcionar correctamente.

Con el fin de que la infraestructura de correo se mantenga operativa, se requieren actividades de actualización, pruebas y mantenimiento; sin embargo, cuando estas tareas se llevan a cabo, es posible que los servicios sean interrumpidos por periodos variables de tiempo. Para aminorar las fallas y pausas en el servicio, así como su impacto, se puede usar una infraestructura de pruebas.

Los ambientes de prueba son usados para diversos propósitos, pues permiten simular las condiciones de producción para verificar configuraciones y evaluar si estas son compatibles y funcionales de manera controlada antes de aplicarlas a producción. Además, hacen posible evaluar cambios en la infraestructura, así como la integración con otras aplicaciones o servicios. Esto permite detectar fallas en las configuraciones sin afectar a los usuarios. Dos ejemplos de cómo se ha utilizado el ambiente de pruebas son: el caso de Rochina Rochina (2021), donde la finalidad del entorno de pruebas es detectar correos electrónicos maliciosos; y el caso de Hurtado Chichande (2018), que propone tecnologías de

contenedores de *software* en ambientes de pruebas. Por nuestra parte, realizamos la implementación de una infraestructura de pruebas durante el periodo 2024-2025.

El objetivo del proyecto consistió en instalar y configurar una infraestructura de correo electrónico para pruebas utilizando Exchange Server de Microsoft, con el fin de validar configuraciones antes de aplicarlas en un entorno productivo. Esto logró reducir el tiempo de ventanas de mantenimiento y aminorar las fallas.

## 2. DESARROLLO TÉCNICO

### 2.1 ANTECEDENTES

La infraestructura de correo electrónico institucional de la Universidad Nacional Autónoma de México está integrada por Exchange local (servidor de Exchange) y Exchange Online (Microsoft 365), es decir, es una implementación híbrida. Una infraestructura híbrida es “un modelo de gestión de Tecnologías de la Información que combina entornos de TI *on-premises*<sup>1</sup> con servicios en la nube, para maximizar la eficiencia operativa y optimizar costos en las empresas” (Goyes López, 2025). Como menciona Benavides Narváez (2019), este tipo de implementaciones también se pueden encontrar en instituciones de educación superior.

La infraestructura de Exchange local se encuentra implementada en un ambiente virtual; cuenta con la versión Exchange Server 2019 y aloja aproximadamente 2,000 cuentas. Anteriormente, se tuvieron experiencias con versiones de Exchange Server 2010 y 2013. Estas versiones actualmente ya no cuentan con soporte, por lo que, ante esta situación, se buscó la manera de mantener una infraestructura actualizada con las versiones disponibles del proveedor.

La infraestructura local se mantenía operando de manera estable y se realizaban actividades de mantenimiento de manera periódica, como la instalación de actualizaciones, copias de seguridad, verificación del estado de los servidores, revisión de bitácoras y de unidades de almacenamiento, etc.; además, se llevaban a cabo actividades de configuración y de modificación de elementos de la infraestructura. Una de las actividades con más impacto eran las actualizaciones, ya que son fundamentales para que los servidores de Exchange sean estables y cuenten con las actualizaciones de seguridad necesarias.

Estas actividades se realizaban de manera controlada; sin embargo, podían variar en la duración y aumentar los tiempos de las actividades de mantenimiento. Para aminorar las posibles fallas y pausas en el servicio, se preparó e instaló una infraestructura para un ambiente de pruebas.

### 2.2 METODOLOGÍA

El entorno de pruebas que se implementó es un modelo que simula un escenario con las características suficientes para validar las configuraciones requeridas. En el caso del entorno de pruebas construido,

---

1 On-premises, se refiere a un entorno de TI en el que los datos, el software y la infraestructura de una empresa se alojan localmente dentro de sus propias instalaciones físicas (Supermicro, 2026).

este se realizó con el modelo de desarrollo en cascada. Dicho modelo es secuencial y se conforma por distintas fases que deben completarse antes de iniciar la siguiente.

Como mencionan Delgado Olivera *et. al* (2021), una de las principales ventajas del modelo es que, con este, “es fácil de manejar los planes de proyectos y permite estimar calendarios y presupuestos con mayor precisión”, lo que asegura que la implementación sea exitosa y funcional.

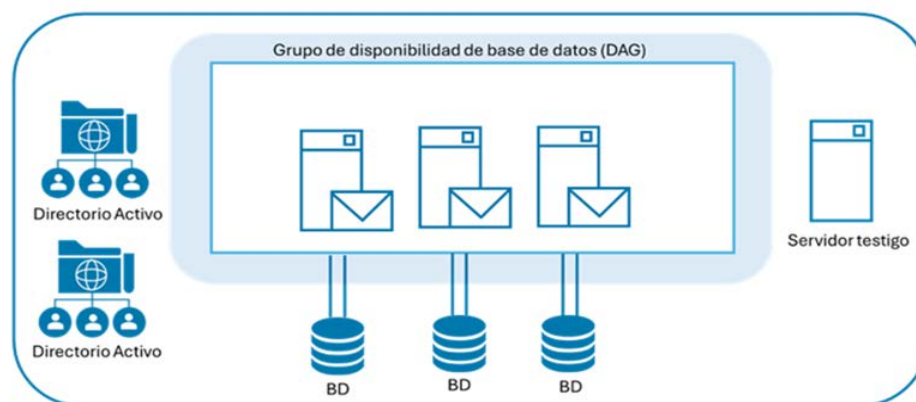
La característica de fases sucesivas en la metodología de cascada la convierte en la indicada para la instalación del entorno de pruebas, ya que permite el ejercicio de prueba y error en procesos que requieren estructura y orden para llevarse a cabo. Estos procesos cumplen funciones cruciales, desde la instalación de las máquinas virtuales para cada función requerida, la generación y configuración del Directorio Activo, la unión al dominio de las máquinas virtuales generadas, la instalación y configuración de los servidores de correo, hasta la validación de entrada y salida de correos. Cada actividad mencionada requiere de completarse para continuar con el siguiente proceso de implementación; si un paso se omite, suceden errores que impiden la instalación de Exchange Server.

Para simular el escenario real, se recopilaron los requerimientos para realizar la implementación sin problemas. Como punto clave, se decidió generar el ambiente de pruebas en máquinas virtuales para aprovechar los recursos que se tienen en el centro de datos. Lo anterior se ve reflejado al utilizar máquinas virtuales y aprovisionarlas con memoria RAM, procesadores y almacenamiento necesarios. Cabe resaltar que los recursos abastecidos de *hardware*, requeridos en cada una de las máquinas virtuales, son elevados, es decir, los recursos requeridos por Exchange no son tan fácilmente aprovisionados a menos que se cuente con la infraestructura necesaria para cubrir estos requisitos; sin embargo, al ser una infraestructura de pruebas, se utilizaron recursos menores comparados a los de un entorno productivo.

De acuerdo con el diseño evaluado, se decidió generar una infraestructura con seis máquinas virtuales en total: tres como servidores de correo electrónico, cada uno con 64 GB de RAM; dos como servidores con el rol de Directorio Activo (Directorio Activo primario y secundario), donde a cada uno se le asignaron 10 GB de RAM; y una como servidor testigo con 3 GB de RAM. Finalmente, se estableció el direccionamiento de red, sistema operativo a usar y los dominios. Estos elementos se muestran en la Figura 1.

**Figura 1**

*Componentes de la infraestructura de correo electrónico de pruebas*



Cada uno de los elementos de la arquitectura de correo electrónico cumple con una función específica que permite el correcto funcionamiento de Exchange Server. Este usa Active Directory para almacenar la información en los buzones y la configuración de los servidores de Exchange en la organización.

En el caso de los servidores de buzones de correo, estos contienen las bases de datos de buzones, incorporan los servicios de transporte que se usan para enrutar<sup>2</sup> el correo y almacenan datos y los servicios de acceso que aceptan conexiones de cliente para todos los protocolos (Microsoft, 2025).

El servidor testigo es requerido para mantener el *quorum* cuando existe un número par de miembros en un grupo de disponibilidad de base de datos (DAG). Un DAG es un grupo de servidores de buzones de correo que hospeda un conjunto de bases de datos y que proporciona recuperación automática de nivel de base de datos en errores de bases de datos, redes y servidores.

Cabe mencionar que, al ser un ambiente de pruebas, es necesario replicar los elementos de la infraestructura de correo electrónico local productiva. Para lograr este propósito, se requiere cumplir con una serie de requisitos que permiten llevar a cabo la instalación de Exchange Server. Con este fin, se realizó la generación de seis máquinas virtuales con sistema operativo Windows Server 2022 Datacenter para cada rol requerido. En cada máquina virtual, se descargaron las actualizaciones propias del sistema operativo, se realizó el ajuste de zona horaria, el cambio de nombre y se asignaron los parámetros de red.

En primer lugar, dentro de cada una de las máquinas virtuales, se realizó la instalación y configuración del Directorio Activo primario y secundario para Exchange Server, a través del uso del asistente de instalación. El Directorio Activo en Exchange Server es el componente principal, ya que permite a los servicios de Exchange la creación, modificación y consulta de datos de los servicios de los Servicios de Dominio de Directorio Activo.

A continuación, se procedió con la instalación de los requisitos previos de Exchange Server. Éstos consistieron en la instalación de *software* y en la ejecución de comandos en los servidores de buzones. Dicho proceso se explica a detalle en el Anexo A. Una vez finalizada la instalación de los prerrequisitos, se procedió a unir los servidores de buzones al dominio, lo cual es necesario para que funcione la infraestructura de correo, pues Exchange depende del Directorio Activo. Posteriormente, se instalaron y configuraron los servidores de buzones. Para este procedimiento, se ejecutaron los comandos que indica la documentación disponible en la página oficial de Microsoft y se instaló la función de servidor de buzón de Exchange por medio del asistente de instalación del *software*. Este proceso se explica a detalle en el Anexo B. Después, se validó el ingreso al Centro de administración de Exchange (EAC).

Una vez dentro del EAC, se realizó la configuración de un conector de salida. Por último, se configuró el ruteo para la salida a red, así como los registros TXT, PTR, A en los DNS.

Tras finalizar la implementación y para poder validar el correcto funcionamiento de la infraestructura de correo electrónico, se realizó la comprobación del envío y recepción de correos a través de la generación de cuentas de dicho servicio. Se generaron 4 cuentas para simular el envío de correos a cuentas institucionales y a cuentas de correo de Gmail. La primera prueba consistió en validar el envío y recepción

---

2 Enrutar es la acción de seleccionar rutas en cualquier red. Una red de computación está formada por muchas máquinas, llamadas nodos, y rutas o enlaces, que conectan dichos nodos. La comunicación entre dos nodos en una red interconectada se puede producir a través de muchas rutas diferentes. El enrutamiento es el proceso de selección de la mejor ruta mediante algunas reglas predeterminadas (AWS, 2026).

de 10 correos electrónicos con dos cuentas creadas en el entorno de pruebas. Con esta prueba, se validó que el envío y recepción de correos interno fue correcto.

Como segunda prueba, se realizó el envío y recepción de 10 correos electrónicos con dos cuentas: una generada en el entorno de pruebas y otra generada en la infraestructura productiva, es decir, una cuenta institucional. Esta prueba comprobó el funcionamiento del conector de salida generado y del flujo de correo entrante y saliente. Como última validación, se realizó el envío y recepción de correo por medio de dos cuentas: una generada en el entorno de pruebas y la segunda en Gmail. Con esto, se verificó que la infraestructura de pruebas enviara y recibiera mensajes de remitentes externos, confirmando la interoperabilidad con otro proveedor de servicio de correo electrónico. El resultado de estas pruebas fue favorable, ya que se confirmó el envío y recepción de correos sin ningún tipo de falla, por lo que los correos que se enviaron llegaron a su destino.

En la etapa de mantenimiento, se realizaron mejoras a la infraestructura, como la creación de un nuevo dominio para tener la posibilidad de generar una mejor clasificación de las cuentas de correo y se generó un grupo de disponibilidad de base de datos (DAG) integrado por los servidores generados.

### 3. RESULTADOS

Se realizaron diversas validaciones en el ambiente de pruebas con resultados exitosos. A continuación, se presentan las pruebas y sus resultados:

- Se limitó la cantidad de correos enviados por día: la realización de este procedimiento en el entorno de pruebas permitió que se ejecutaran las configuraciones antes de aplicarlas en el entorno productivo, lo que permitió disminuir considerablemente el tiempo de duración de la ventana de mantenimiento, la cual fue de 90 minutos. De esta manera, se redujo el tiempo pues el promedio era de aproximadamente 150 minutos.
- Se ejecutó una actualización acumulativa de Exchange Server: la finalidad de la actualización fue agregar nuevas características y funcionalidades, además de solucionar problemas reportados en Exchange; sin embargo, es una tarea que requiere una gran cantidad de tiempo. En este caso, se simuló el proceso de actualización en el servidor y se registraron los tiempos correspondientes, con el objetivo de tenerlos en cuenta para la planificación de futuras ventanas de mantenimiento en la infraestructura principal.
- Se ejecutó un *script* con el que se obtuvo información de los servidores de correo de manera más rápida: esta actividad permitió obtener información útil y detallada de un servidor de correo sin tener que aplicar varios comandos para encontrar esta información.

### 4. CONCLUSIONES

Con base en las necesidades del servicio de correo electrónico local, se generó un ambiente de pruebas que permitiera simular el entorno productivo para validar y evaluar configuraciones de manera controlada y evitar afectar la infraestructura en producción. Después de la implementación, se lograron verificar de manera segura nuevas funcionalidades, ejecutar mejoras en la infraestructura productiva, evaluar el funcionamiento de *scripts*, validar actualizaciones antes de aplicarlas en producción para evitar fallas

o demoras y se consiguió reducir los tiempos en las ventanas de mantenimiento. Como nota final, es importante tomar en cuenta que, para la correcta implementación y función de Exchange Server, hay que proveer en su totalidad los recursos tecnológicos requeridos que se presentan en la documentación oficial, ya sean físicos o virtuales. De no cumplirse con estos requisitos, pueden generarse problemas de rendimiento, como lentitud en procesos básicos del sistema operativo, dificultades durante la instalación del *software* Exchange Server o al utilizar la consola de administración del mismo, además de degradación del rendimiento del sistema entero.

## REFERENCIAS

- Amazon Web Services. (2026) *¿Qué es el enrutamiento?* <https://aws.amazon.com/es/what-is/routing>
- Benavides Narváez, G. S. (2019). *Creación de un prototipo de nube híbrida para una institución de educación superior* [Tesis de grado. Universidad de las Américas]. <http://dspace.udla.edu.ec/handle/33000/11336>
- Delgado Olivera, L. C. & Díaz Alonso, L. M. (2021). Modelos de Desarrollo de Software. *Revista Cubana de Ciencias Informáticas*, 15, (1), pp. 37-51.
- Goyes López, D. G. (2025). *Estrategia de optimización de infraestructura híbrida: integración de servicios en la nube y entornos On-Premises en pequeñas y medianas empresas (PYMES)* [Tesis de grado, repositorio institucional Universidad Técnica de Babahoyo]. <http://dspace.utb.edu.ec/handle/49000/17904>
- Hurtado Chichande, D. A. (2018). *Tecnología de contenedores de software en entornos de pruebas* [Tesis de grado, Pontificia Universidad Católica del Ecuador Esmeraldas]. <https://repositorio.puce.edu.ec/handle/123456789/37960>
- Microsoft. (2025). *Arquitectura de Exchange*. <https://learn.microsoft.com/es-es/exchange/architecture/architecture>
- Rochina Rochina, C. G. (2021). *Diseño y evaluación de una metodología para reducir los ciberataques originados a través de correo electrónico mediante la aplicación de filtros y reglas sobre un Gateway*. [Tesis de grado, Escuela Superior Politécnica de Chimborazo Riobamba]. <https://dspace.espoch.edu.ec/handle/123456789/14677>
- Supermicro. (2026). *¿Qué es On-Premises?* <https://www.supermicro.com/es/glossary/on-premises>

## ANEXO A. INSTALACIÓN DE REQUISITOS PREVIOS PARA EXCHANGE SERVER

A continuación, se muestra el proceso de instalación de los requisitos previos para llevar a cabo la instalación de Exchange Server de manera exitosa. Estos componentes son requisitos esenciales para el correcto funcionamiento de Exchange Server.

Se requiere de la ejecución de comandos e instalación del siguiente *software* en la máquina virtual que tendrá la función de servidor de correo:

### 1. Preparación para Directorio Activo.

Con el fin de preparar el Directorio Activo para Exchange Server, se debe instalar el paquete de administración de herramientas remotas

1.1 Instalar las Herramientas de administración remota del servidor (RSAT) para Servicios de dominio de Directorio Activo; se debe ejecutar el siguiente comando en la terminal de PowerShell.

- Comando:

```
Install-WindowsFeature RSAT-ADDS
```

### 2. Exchange Server: rol de servidor de buzón de correo.

Instalar *software* requerido para el servidor con rol de buzón de correo.

2.1 El primer *software* a instalar es .NET Framework 4.8. La instalación se realiza a través del asistente de instalación y consta de una serie de pasos indicados por el asistente.

2.2 El siguiente *software* a instalar es Visual C++ Redistributable Package for Visual Studio 2012. La instalación se realiza a través del asistente de instalación.

2.3 El tercer *software* a instalar es Visual C++ Redistributable Package for Visual Studio 2013. La instalación se realiza a través del asistente de instalación.

### 3. Instalar componentes de Skype Empresarial Server necesarios.

3.1 Instalar función de Windows Server Media Foundation; se debe ejecutar el siguiente comando en la terminal de PowerShell.

- Comando:

```
Install-WindowsFeature Server-Media-Foundation
```

3.2 Instalar Unified Communications Managed API 4.0. La instalación se realiza a través del asistente de instalación y consta de una serie de pasos indicados por el asistente.

### 4. Ejecutar comando para características de Windows

4.1 Para experiencia de escritorio, se debe ejecutar el siguiente comando en la terminal de PowerShell.

- Comando:

Install-WindowsFeature Server-Media-Foundation,  
NET-Framework-45-Core, NET-Framework-45-ASPNET,  
NET-WCF-HTTP-Activation45, NET-WCF-Pipe-Activation45,  
NET-WCF-TCP-Activation45, NET-WCF-TCP-PortSharing45,  
RPC-over-HTTP-proxy, RSAT-Clustering,  
RSAT-Clustering-CmdInterface, RSAT-Clustering-Mgmt,  
RSAT-Clustering-PowerShell, WAS-Process-Model, Web-Asp-Net45,  
Web-Basic-Auth, Web-Client-Auth, Web-Digest-Auth,  
Web-Dir-Browsing, Web-Dyn-Compression, Web-Http-Errors,  
Web-Http-Logging, Web-Http-Redirect, Web-Http-Tracing,  
Web-ISAPI-Ext, Web-ISAPI-Filter, Web-Metabase,  
Web-Mgmt-Console, Web-Mgmt-Service, Web-Net-Ext45,  
Web-Request-Monitor, Web-Server, Web-Stat-Compression,  
Web-Static-Content, Web-Windows-Auth, Web-WMI,  
Windows-Identity-Foundation, RSAT-ADDS

4.2 Instalar Módulo de reescritura de direcciones URL de IIS. La instalación se realiza a través del asistente de instalación y consta de una serie de pasos indicados por el asistente.

5. Reiniciar equipo.

## ANEXO B. INSTALACIÓN DE SERVIDOR DE CORREO ELECTRÓNICO

A continuación, se muestran los pasos para la instalación de un servidor de correo electrónico:

1. Preparar Directorio Activo y dominios para Exchange Server.

1.1 Ampliar el esquema de Directorio Activo (la extensión del esquema de Directorio Activo agrega y actualiza clases, atributos y otros elementos).

• Comando:

```
cd <VirtualDVDdrive letter>:\Setup.exe /IAcceptExchangeServerLicenseTerms_DiagnosticDataON /PrepareSchema
```

2. Preparar Directorio Activo.

2.1 Se crean contenedores, objetos y otros elementos de Directorio Activo para almacenar información. Se crea la organización de Exchange.

- Comando:

```
<Virtual DVD drive letter>:\Setup.exe /IAcceptExchangeServerLicenseTerms_DiagnosticDataON /PrepareAD /OrganizationName:"<Organization name>"
```

### 3. Preparar los dominios de Directorio Activo.

3.1 Se crean contenedores y grupos de seguridad adicionales y establece los permisos para que Exchange pueda acceder a ellos.

- Comando:

```
<Virtual DVD drive letter>:\Setup.exe /IAcceptExchangeServerLicenseTerms_DiagnosticDataON /PrepareDomain[:<DomainFQDN>]
```

### 4. Instalar la función de servidor de buzón de Exchange.

4.1 Descargar Exchange Server (archivo ISO).

4.2 Montar ISO.

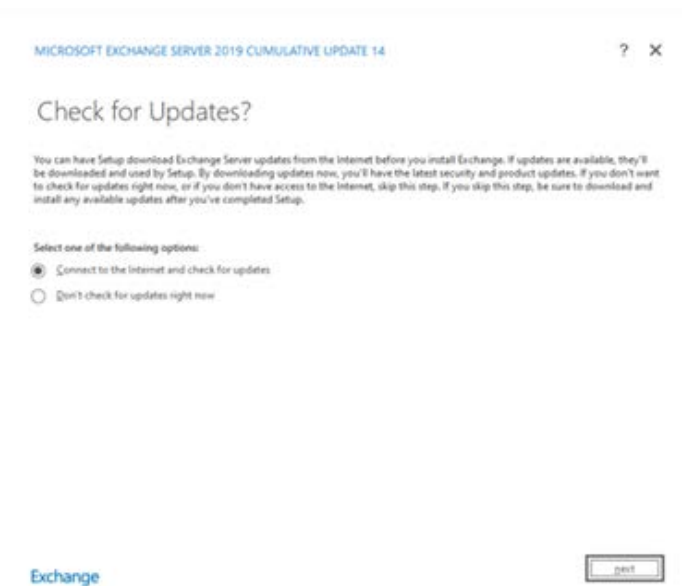
4.3 En la unidad de DVD virtual, iniciar la instalación de Exchange haciendo doble clic en "Setup.exe".

4.4 Se abrirá el asistente de instalación de Exchange Server y se seleccionarán las opciones que solicita el asistente:

4.4.1) Buscar actualizaciones. Se muestra la ventana de esta tarea en la Figura 2.

## Figura 2

### Ventana de actualización de Exchange Server



4.4.2) Copiando archivos. En la Figura 3, se muestra el progreso de esta tarea.

**Figura 3**

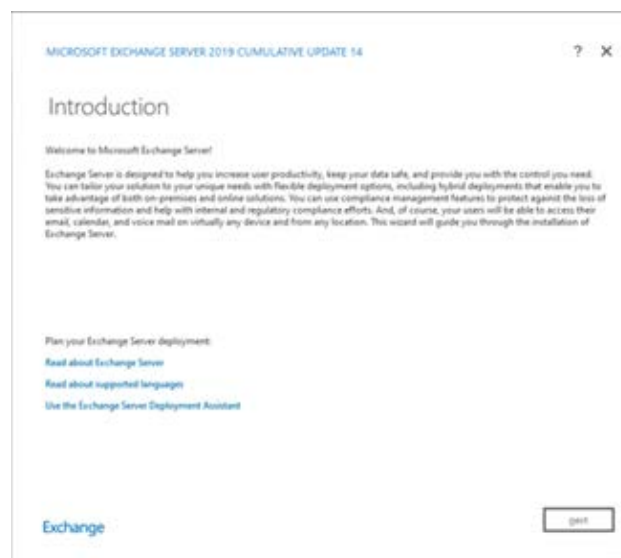
*Ventana de copiado de archivos en Exchange Server*



4.4.3) Introducción. En la Figura 4, se muestra el apartado de bienvenida al *software*.

**Figura 4**

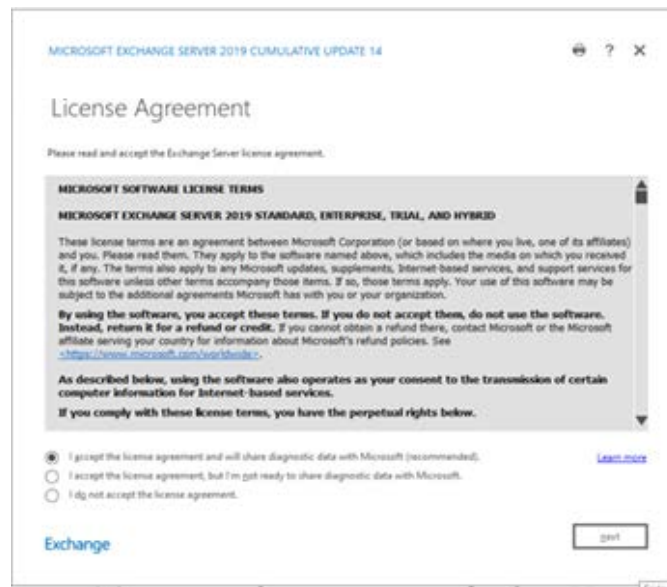
*Ventana de introducción de Exchange Server*



4.4.4) Acuerdo de licencia. En la Figura 5, se muestra el acuerdo de licencia y sus términos.

**Figura 5**

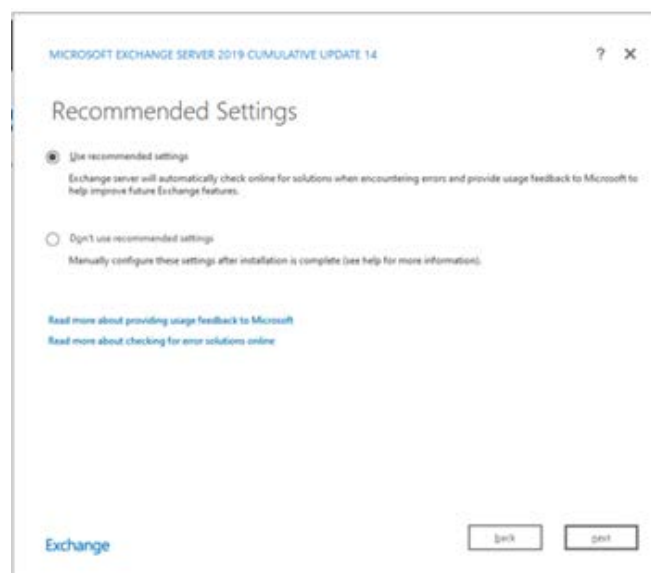
*Ventana de aceptación del acuerdo de licencia de Exchange Server*



4.4.5) Configuración recomendada. En la Figura 6, se muestra la opción para continuar usando la configuración recomendada o con una configuración personalizada.

**Figura 6**

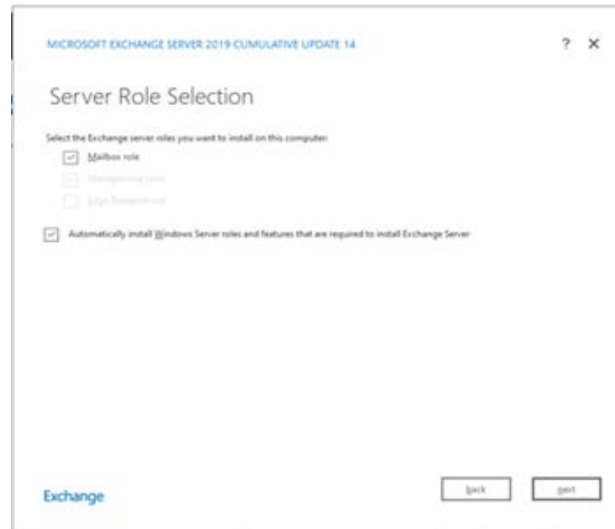
*Ventana de selección de configuración recomendada de Exchange Server*



4.4.6) Selección de rol de servidor. En la Figura 7, se muestra la selección del rol de servidor que se va a instalar en el equipo.

**Figura 7**

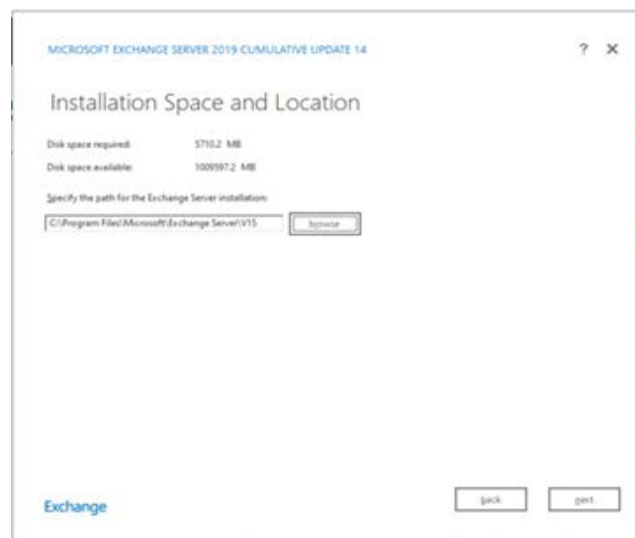
*Ventana de selección de rol de servidor de Exchange Server*



4.4.7) Espacio y ubicación de instalación. En la Figura 8, se muestran las opciones de selección de almacenamiento para la instalación.

**Figura 8**

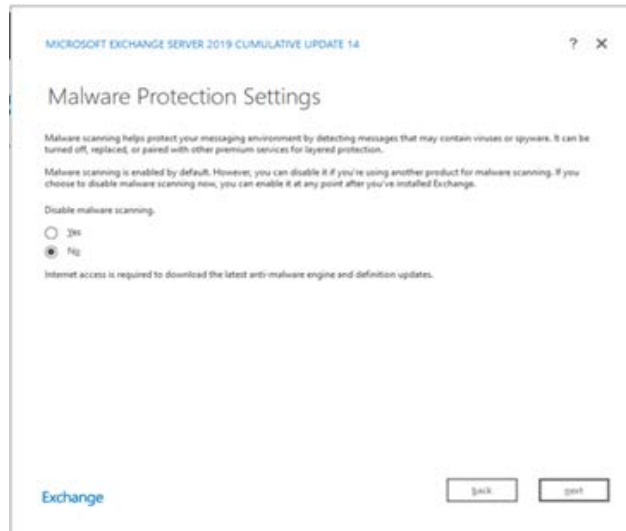
*Ventana de selección de espacio de almacenamiento dedicado y ubicación para la instalación de Exchange Server*



4.4.8) Configuración de protección contra *malware*. En la Figura 9, se presenta el apartado de configuración de seguridad.

**Figura 9**

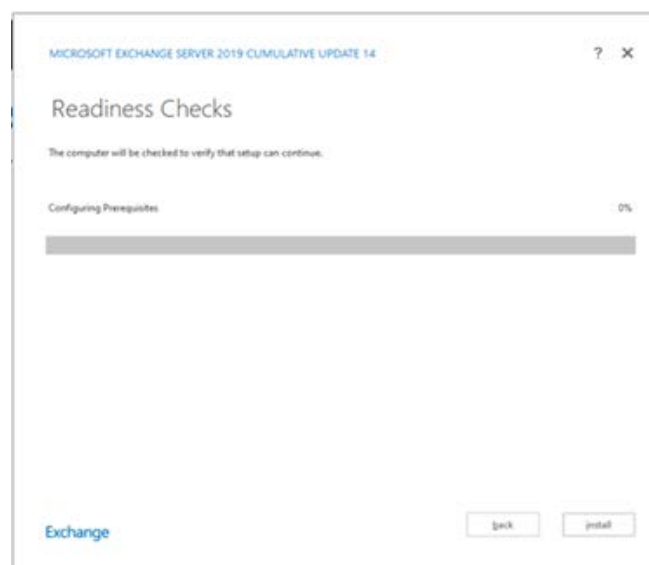
*Ventana de configuración de protección contra malware de Exchange Server*



4.4.9) Comprobaciones de preparación. En la Figura 10, se puede observar el proceso que realiza el *software* para comprobar que la instalación se haya efectuado correctamente.

**Figura 10**

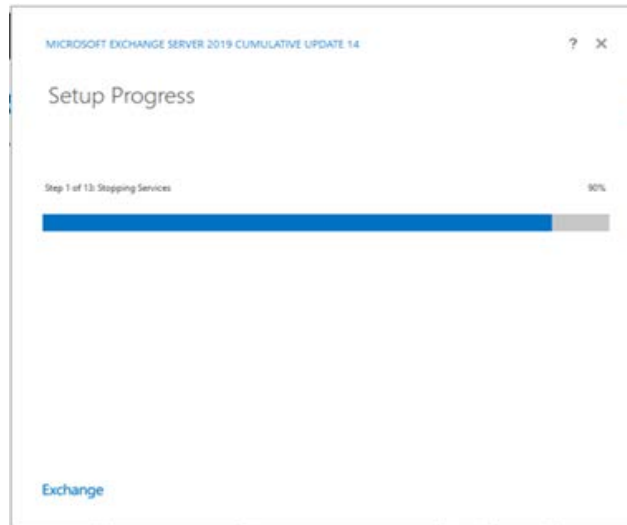
*Ventana de comprobaciones de preparación de Exchange Server*



4.4.10) Progreso de la instalación. En la Figura 11, se ve el progreso de cada uno de los pasos de instalación.

**Figura 11**

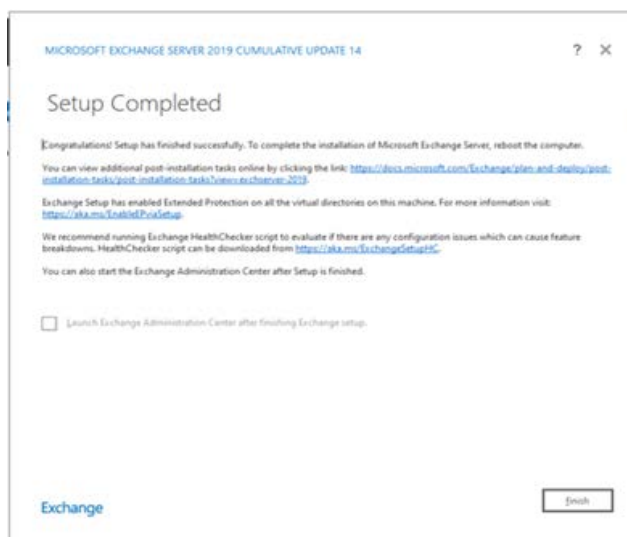
*Ventana de progreso de la instalación de Exchange Server*



4.4.11) Configuración completada. En la Figura 12, se observa la ventana de aviso de instalación correcta.

**Figura 12**

*Ventana de configuración completada de Exchange Server*

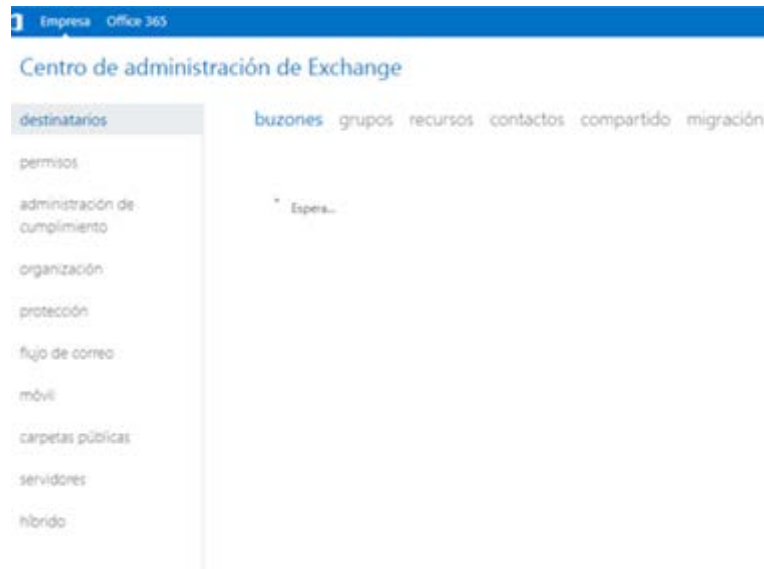


#### 4.4.12) Reinicio de equipo.

5. Validación de ingreso a ECP. En la Figura 13, se presenta la interfaz del inicio del *software* con todas sus opciones y herramientas.

**Figura 13**

*Pantalla de inicio de Exchange Server*



# Propuesta de un sistema de adquisición y generación de señales en plataformas Raspberry Pi

*Signal acquisition and generation system on Raspberry Pi platforms*

## Información del reporte:

Licencia Creative Commons



El contenido de los textos es responsabilidad de los autores y no refleja forzosamente el punto de vista de los dictaminadores, o de los miembros del Comité Editorial, o la postura del editor y la editorial de la publicación.

Para citar este reporte técnico:

Valera Orozco, B., Ruiz Botello, G.A., y Roldán Serrato, K.L. (2026). Propuesta de un sistema de adquisición y generación de señales en plataformas Raspberry Pi. *Cuadernos Técnicos Universitarios de la DGTIC*, 4(2), páginas (124 - 147). <https://doi.org/10.22201/dgtic.30618096e.2026.4.2.146>

## Benjamín Valera Orozco

Instituto de Ciencias Aplicadas y Tecnología  
Universidad Nacional Autónoma de México  
[benjamin.valera@icat.unam.mx](mailto:benjamin.valera@icat.unam.mx)  
ORCID: 0009-0006-7730-8750

## Gerardo Antonio Ruiz Botello

Instituto de Ciencias Aplicadas y Tecnología  
Universidad Nacional Autónoma de México  
[gerardo.ruiz@icat.unam.mx](mailto:gerardo.ruiz@icat.unam.mx)  
ORCID: 0009-0000-3525-4252

## Karen Lucero Roldán Serrato

Instituto de Ciencias Aplicadas y Tecnología  
Universidad Nacional Autónoma de México  
[lucero.roldan@icat.unam.mx](mailto:lucero.roldan@icat.unam.mx)  
ORCID: 0009-0006-9761-5029

## Resumen

En la actualidad, el mercado de los sistemas operativos ha tomado un rumbo hacia el uso de sistemas abiertos, libres y consolidados entre la comunidad de especialistas en cómputo y electrónica. Un caso aplicado es la computadora de placa única Raspberry Pi, a la cual se le pueden instalar diferentes versiones de sistema operativo de plataforma abierta y gratuita. Las aplicaciones enfocadas en la electrónica encuentran un gran potencial en la Raspberry Pi debido a su capacidad de desarrollo de sistemas completos gracias a su conector de propósito general de 40 pines. En este proyecto, se diseñó e implementó un sistema electrónico para la adquisición y generación de señales eléctricas basadas en una microcomputadora

Raspberry Pi y un sistema operativo Raspbian OS. Esta herramienta resultó una implementación adecuada y útil ya que existe un gran número de aplicaciones de instrumentación en las que se requiere este tipo de procesos de manipulación de señales. La aportación de este trabajo fue el desarrollo de un sistema a bajo costo y optimizado para la adquisición y generación de señales, el cual estuvo creado con programación en Python y un ambiente gráfico de visualización. Éste fue utilizado en aplicaciones concretas como laboratorios de electrónica o de experimentación en telecomunicaciones.

**Palabras clave:** Sistema de adquisición de datos, procesamiento de señales, Raspberry Pi 5, instrumentación electrónica.

### Abstract

*Currently, the operating systems market has been shifting toward open, free, and community-established solutions widely adopted by computing and electronics specialists. A practical example is the Raspberry Pi single-board computer, which supports various versions of open and free operating systems. Applications in electronics find significant potential in the Raspberry Pi due to its ability to support full system development through its 40-pin general-purpose connector. In this project, an electronic system for the acquisition and generation of electrical signals was designed and implemented based on a Raspberry Pi microcomputer running Raspbian OS. This tool proved to be an appropriate and useful solution, given the wide range of instrumentation applications that require signal processing capabilities. The main contribution of this work is the development of a low-cost, optimized system for signal acquisition and generation, implemented using Python programming and a graphical visualization environment. It was applied in specific contexts such as electronics laboratories and telecommunications experimentation.*

**Keywords:** Data acquisition system, signal processing, Raspberry Pi 5, electronic instrumentation.

## 1. INTRODUCCIÓN

Hacia el año 2014, comenzaron a desarrollarse diversas aplicaciones con base en Raspberry Pi orientadas no sólo al ámbito educativo, sino también a aplicaciones científicas e industriales (Sreejith *et al.* 2014). Estas capacidades extendidas incluyen dispositivos que compiten con grandes ventajas frente a los microcontroladores Arduino. Por ejemplo, en el ámbito electrónico, además de incluir las características tradicionales en los microcontroladores Arduino, como entradas y salidas de propósito general, la plataforma Raspberry Pi incorpora herramientas para manipular motores, luces, sensores y más. Sin embargo, la principal ventaja de la plataforma Raspberry Pi es que cuenta con un sistema operativo comparable a Windows, con la capacidad de gestionar dispositivos periféricos complejos y bases de datos en ambientes de programación como Python y C.

En este ambiente de crecimiento, se planteó la posibilidad de incursionar en el uso de herramientas modernas para el desarrollo de dispositivos electrónicos capaces de generar y capturar señales analógicas en la plataforma Raspberry Pi. La principal motivación es disminuir la dependencia tecnológica actual de los sistemas de cómputo basados en microprocesadores Intel y en el sistema operativo Microsoft. Adicionalmente, los procesos de conversión analógica a digital (ADC) y digital a analógica (DAC) (ver glosario en el Anexo A) son fundamentales en la elaboración de instrumentación para aplicaciones industriales y de diversos ámbitos de la actividad humana.

Conviene señalar los desarrollos de instrumentación en el área de cómputo aplicado en plataformas de propósito específico. En 2014, algunas plataformas del Internet de las Cosas (IoT) incluyeron el uso de convertidores analógico-digitales externos, tales como el MCP3008, ADS1115 o el de alta precisión ADS1263, los cuales permiten digitalizar señales a través de sensores y sistemas analógicos (Medina, 2017; Rodríguez, 2018). Estos dispositivos son ampliamente utilizados por su fácil adaptación con interfaces como SPI e I2C, que permiten una integración directa con la Raspberry Pi (AG Electrónica, 2024; Meena *et al.*, 2019). En 2015 ya se desarrollaban sistemas del IoT enfocados en la adquisición de señales a bajas frecuencias (VLF), los cuales consistían en un receptor VLF, un preamplificador y un convertidor analógico digital sobre una tarjeta Raspberry Pi portable (Gunawan *et al.*, 2020). Así, a partir del 2018 y para robustecer la instrumentación, se incorporaron diversos módulos de validación y calibración a los sistemas de adquisición y medición basados en tarjetas de procesamiento. Estos desarrollos fueron aplicados a equipos industriales como módulos adicionales (Arif *et al.*, 2018).

En años posteriores, cobraron relevancia los instrumentos de generación de señales con base en radiofrecuencias FM y aplicadas a tarjetas como Raspberry Pi en sus últimas versiones de ese tiempo. A pesar de que siempre se debe contemplar la compatibilidad de controladores y bibliotecas para los últimos modelos de tarjetas, esta plataforma demostró ser una versión estable en un analizador de espectros, donde se observaron buenas evidencias en los componentes armónicos por discriminar (Martínez-Quintero *et al.*, 2019). Por otra parte, hacia 2020, la literatura reportó el uso de plataformas basadas en *software* libre, como GNU Linux, programación en Python y librerías compatibles e incluidas en tarjetas como Raspberry Pi y Arduino en los modelos de 3, 4 y 5 (Cassel Barbosa *et al.*, 2020). En algunas aplicaciones como las del campo biomédico, se han reportado múltiples implementaciones orientadas a la adquisición de señales fisiológicas como EMG (electromiografía) y ECG (electrocardiografía), mediante la combinación de Raspberry Pi con microcontroladores auxiliares que se encargan del preprocesamiento y adquisición de señales a mayor resolución y frecuencia (Chen & Liu, 2025).

Adicionalmente, las tarjetas de propósito específico como las de Adafruit y otras plataformas similares, entre ellas Orange Pi, Banana Pi y Raspberry Pi, permiten implementar diversas aplicaciones de instrumentación, tales como la adquisición y generación de señales (Arif *et al.*, 2018; Gunawan *et al.*, 2020). Además, se cuenta con la programación en lenguajes como Python y C incluidos en el sistema operativo de la Raspberry Pi, así como la integración de sensores y actuadores (bluemoon16, 2023). En términos de costo y flexibilidad, es una opción ideal para sistemas de adquisición de datos (DAQ) y generación de señales analógicas y digitales (Ellison *et al.*, 2024).

De acuerdo con la revisión de los antecedentes, se observa una evolución sostenida en el desarrollo de sistemas de instrumentación basados en plataformas de cómputo de propósito específico, particularmente en el contexto del Internet de las Cosas (IoT). La incorporación de convertidores analógico-digitales externos como MCP3008, ADS1115 y ADS1263 ha permitido la digitalización eficiente de señales provenientes de sensores y sistemas analógicos, facilitando su integración con plataformas como la Raspberry Pi mediante interfaces estándar como I<sup>2</sup>C y SPI, soluciones que han destacado por su facilidad de implementación y adaptabilidad en sistemas embebidos de bajo costo. Además, se han desarrollado sistemas de adquisición de señales de baja frecuencia, así como arquitecturas portables basadas en Raspberry Pi que integran etapas de acondicionamiento, amplificación y conversión analógico-digital; estos sistemas han incorporado módulos de validación y calibración que han contribuido a mejorar su robustez y aplicabilidad en entornos industriales como módulos de instrumentación adicionales. De esta manera, se ha extendido el uso de plataformas de generación de señales, incluyendo aplicaciones en

radiofrecuencia, donde la Raspberry Pi ha demostrado capacidad para la implementación de analizadores de espectro con resultados satisfactorios en la identificación de componentes armónicos.

A partir de este análisis, se justifica el desarrollo de un sistema de adquisición y generación de señales basado en Raspberry Pi y Python como una solución de instrumentación que responde a la necesidad de plataformas flexibles, de bajo costo y fácilmente reconfigurables que integren en un solo entorno las funciones de adquisición, procesamiento y generación de señales.

La principal aportación de este trabajo es la implementación de una solución integral que aprovecha la arquitectura de la Raspberry Pi junto con Python para desarrollar un sistema DAQ (*Data Acquisition System*) (Anexo A) capaz no sólo de adquirir señales mediante interfaces estándar, sino también de generar señales controladas de manera programable. Esto permite un entorno unificado de instrumentación, reduciendo la dependencia de equipos especializados de alto costo y facilitando la replicabilidad del sistema en aplicaciones académicas, de investigación e incluso industriales de prototipos integrando *software* y *hardware* de código abierto. El sistema propuesto contribuye a la consolidación de plataformas abiertas de instrumentación al integrar adquisición y generación de señales en una misma arquitectura, fortaleciendo la tendencia actual hacia sistemas embebidos modulares, escalables y basados en *software* libre, así como adaptados a ambientes institucionales.

El objetivo del proyecto presentado en este reporte técnico es desarrollar un sistema de adquisición y generación de señales en plataformas Raspberry Pi.

## 2. DESARROLLO TÉCNICO

Para este proyecto, se eligió Raspberry Pi versión 5 debido a la experiencia previa con esta plataforma, su facilidad de integración y acoplamiento con los elementos de instrumentación y su compatibilidad con otros módulos o sensores externos. Posteriormente, se realizó la implementación de su sistema operativo, Raspberry Pi OS. En cuanto al lenguaje de programación, se empleó Python, debido a su papel fundamental en el manejo y procesamiento de los datos adquiridos. En seguida, se configuró el entorno de desarrollo mediante la instalación de algunas bibliotecas necesarias para la implementación del sistema, entre las que se incluyeron: NumPy, SciPy y Matplotlib, las cuales se utilizaron para el procesamiento, análisis y visualización de señales. Esto exhibió a la Raspberry Pi como una herramienta poderosa para la investigación científica (Pidora, 2024). En cuanto a los protocolos de comunicación y generación de señales, la Raspberry Pi soporta interfaces como I2C, SPI y UART, lo que permite una gran flexibilidad al momento de integrar convertidores analógicos a digital y sensores de distintos tipos. Además, se han documentado múltiples soluciones que incluyen la transmisión y el almacenamiento de datos en la nube, facilitando el monitoreo remoto y el análisis en tiempo real mediante servicios como ThingSpeak, AWS o plataformas personalizadas (Magalhães, 2024).

El desarrollo e implementación de tecnología adaptada a requerimientos electrónicos tiene la desventaja de altos costos de implementación o fallos ante la falta de calibración a largo plazo. Por lo anterior, se propuso la creación de un prototipo moderno en el desarrollo de dispositivos electrónicos para generar y capturar señales analógicas en la plataforma Raspberry Pi. La principal motivación es reducir la dependencia tecnológica actual de los sistemas de cómputo basados en microprocesadores y sistemas operativos propietarios, así como promover el uso de sistemas de instrumentación portables, de menor costo y apoyados en *software* libre para la programación del sistema. Adicionalmente, los

procesos de conversión analógica a digital y digital a analógica considerados resultan ser fundamentales en la elaboración de instrumentación para aplicaciones industriales y de diversos ámbitos de la actividad humana.

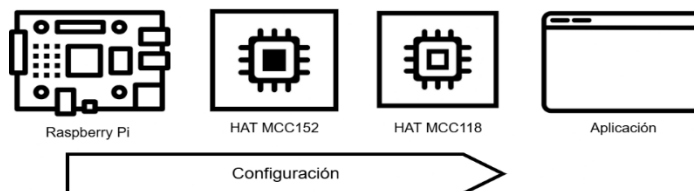
## 2.1 INTEGRACIÓN DE CIRCUITOS DE SEÑALES CON RASPBERRY PI

El proyecto fue propuesto tomando como base el planteamiento de Llamas (2025), que consiste en integrar un sistema DAQ basado en una computadora de bajo costo y diseño compacto, la Raspberry Pi 5. Este sistema integra dos circuitos HAT (ver glosario en el Anexo A) al conector de 40 pines en la Raspberry Pi, y termina con el desarrollo de un *software* con aplicación real en la generación y adquisición de señales eléctricas. Un circuito HAT tiene el propósito específico de generar señales analógicas a partir de datos digitales en la Raspberry Pi. El otro HAT, en cambio, se encarga de generar datos digitales en la Raspberry Pi a partir de señales analógicas del ambiente exterior. Con estos datos, se pueden procesar parámetros a partir de algoritmos específicos de control o procesamiento digital de señales. Con base en el documento informativo de XOREN Ingeniería (2023), el sistema DAQ desarrollado tiene el enfoque de programación en lenguaje Python, ya que es uno de los lenguajes más populares en la industria por su flexibilidad y solución de problemas sencillos con una sintaxis simple y elegante.

El método incluye la instalación física de los tres circuitos, su configuración en el sistema operativo Raspberry OS y el desarrollo de una aplicación para la adquisición y generación de señales analógicas, como se muestra en la Figura 1.

**Figura 1**

*Método por emplear*



La Figura 1 muestra conceptualmente el sistema Raspberry Pi, que hospeda dos subsistemas de tratamiento de señales HAT MCC152 (Measurement Computing, 2019) y HAT MCC118 (Measurement Computing, 2020). El subsistema MCC152 es un circuito electrónico diseñado para la generación de señales de voltaje mediante conversión digital-analógica DAC, a partir de datos digitales, diseñado específicamente para conectarse a la Raspberry Pi. Por otra parte, el subsistema MCC118 es un circuito electrónico para la medición de voltaje ADC, de diseño específico para conectarse también a la tarjeta. El par de circuitos electrónicos MCC152 y MCC118 habilitan a una microcomputadora Raspberry Pi con capacidades extendidas para el procesamiento digital de señales.

En la misma Figura 1, la conexión física entre la Raspberry Pi y los circuitos MCC152 y MCC118 no es suficiente para habilitar las capacidades de procesamiento digital de señales, también se debe configurar el ambiente de programación Python al cargar las utilerías necesarias para este propósito. De esta forma, se puede desarrollar una aplicación funcional que procese datos digitales tanto de entrada como de salida y se pueda interactuar con el entorno real.

El principal objetivo es implementar un sistema DAQ que pueda ser de utilidad para una aplicación futura, ya sea de control o de procesamiento digital de señales, en una plataforma de cómputo abierta y de bajo costo. Este sistema DAQ puede ser de utilidad para la medición de variables físicas como temperatura, presión, desplazamiento, entre otras. Además, se puede actuar sobre transductores como motores, electroválvulas, resistencias, luces, entre otros.

El resultado esperado es un sistema DAQ con las siguientes características obtenidas a partir de las especificaciones de los circuitos HAT:

#### **Generación de señal analógica**

- Excursión analógica: 0 a 5 V.
- Resolución: 12 bits.
- Periodo entre muestras: 12  $\mu$ s.

#### **Adquisición de señales**

- Excursión analógica: 0 a 5 V.
- Tasa de muestreo: 100 kmuestras/s.
- Resolución: 12 bits

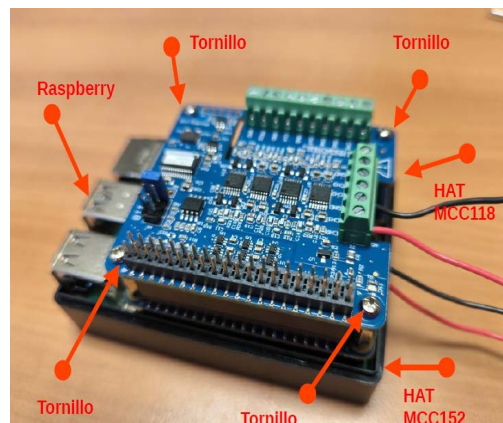
## 2.2 CONFIGURACIÓN DE HARDWARE DEL SISTEMA ELECTRÓNICO DAQ

El sistema electrónico DAQ adquiere y genera señales analógicas, implementando los procesos DAC y ADC respectivamente. Estos procesos son primordiales en diversas aplicaciones de los sistemas electrónicos para la medición, control y procesamiento de información en sistemas digitales.

Para instalar el primer HAT DAC en la Raspberry Pi, primero se deben colocar los postes de sujeción a la carcasa y el extensor para el conector de 40 pines, como se muestra en la Figura 2. El primer HAT, MCC152, se prepara colocando cables de extensión para el canal analógico 0, se especifica su dirección a cero mediante la ausencia de puente en el conector de direcciones y se especifica el voltaje de operación a 5 V con el puente en el conector de polarización. El primer HAT se conecta a la Raspberry Pi añadiendo 4 postes adicionales y un segundo extensor del conector de 40 pines. El segundo HAT ADC, MCC118, se prepara colocando cables de extensión para el canal 0 y especificando la dirección 1 con un puente en el conector de dirección. Posteriormente, el HAT se monta sobre el arreglo y se sujeta con tornillos. Finalmente, al arreglo se le añade teclado y ratón inalámbricos, pantalla HDMI y polarización.

### **Figura 2**

*Sistema DAQ completo*



## 2.3 INTEGRACIÓN DEL SISTEMA

Una vez integrado el *hardware* de la Figura 2, como parte del método empleado, se procede a configurar el *software* para desarrollar una aplicación funcional que efectivamente proporcione tratamiento a las señales de entrada y salida del sistema electrónico. Después, se debe verificar la configuración comprobando el funcionamiento con los programas de prueba que distribuye el fabricante de los HATS. Finalmente, la plataforma estará lista para desarrollar programas propios en aplicaciones específicas.

### 2.3.1 CONFIGURACIÓN Y APLICACIÓN

En la siguiente descripción de la configuración, se asume que el sistema operativo Raspberry Pi OS está instalado y actualizado en la microcomputadora. El primer paso en la configuración consiste en descargar la *daqhats library* (Measurement Computing Corporation, 2024) en la carpeta raíz del usuario mediante las dos siguientes instrucciones en la terminal de comandos:

```
.../cd ~  
.../git clone https://github.com/mccdaq/daqhats.git
```

Posteriormente, se compilan e instalan las herramientas de la librería:

```
.../cd ~/daqhats  
.../sudo ./install.sh
```

Mediante este procedimiento, se instala la librería de los HATS y la documentación junto con los archivos de ejemplo en el directorio `/home/usuario/daqhats/`.

### 2.3.2 VALIDACIÓN Y CALIBRACIÓN

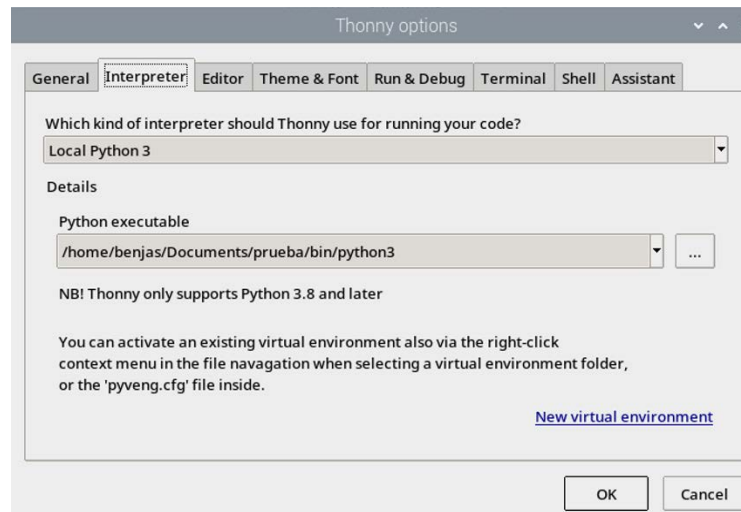
Para verificar la correcta operación de las configuraciones descritas en 2.4.1, se utiliza Thony como ambiente de programación (Jaryd, 2023) en la constitución de un proyecto de aplicación mediante el cual se pueda comprobar la correcta operación de las tarjetas HAT.

En este sentido y con fines de seguridad computacional, se crea un entorno virtual de programación que aisle la operación de las tarjetas HAT del resto del sistema operativo. Inicialmente, este entorno virtual es utilizado para alojar el primer programa básico de prueba que distribuye el fabricante de los HATS. El programa de prueba es un archivo de Python que genera un voltaje analógico en el canal 0 del HAT MCC152.

Para crear el entorno virtual sobre un directorio vacío de prueba, se selecciona la opción del menú de Thony "Run/Configure interpreter...", que despliega el diálogo de la Figura 3.

### Figura 3

Diálogo "Thonny options" para crear un nuevo ambiente virtual

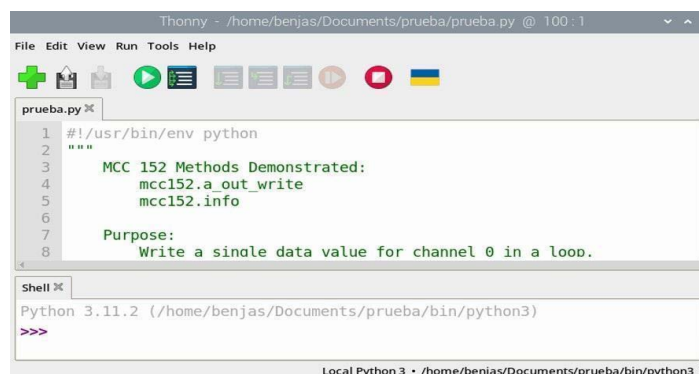


Al presionar el enlace *New virtual environment* de la Figura 3, se despliega una advertencia en donde se hace énfasis que el directorio a seleccionar debe estar vacío para el nuevo ambiente virtual. Al aceptar la advertencia, se selecciona el directorio vacío de prueba y se retorna al diálogo de la Figura 3, en donde se confirma la ubicación del archivo ejecutable.

Al aceptar el diálogo de la Figura 3, se retorna al ambiente de programación Thonny, donde ya es posible probar los ejemplos proporcionados por el fabricante de los HATS con la librería llamada *daqhats*. Con este propósito, se copia el texto del archivo de prueba llamado *analog\_output\_write.py* del directorio /home/usuario/daqhats/ al documento *untitled* y se salva como *prueba.py*, como se muestra en la Figura 4. Por medio de este programa, es posible generar un voltaje analógico en el canal 0 del HAT MCC152.

### Figura 4

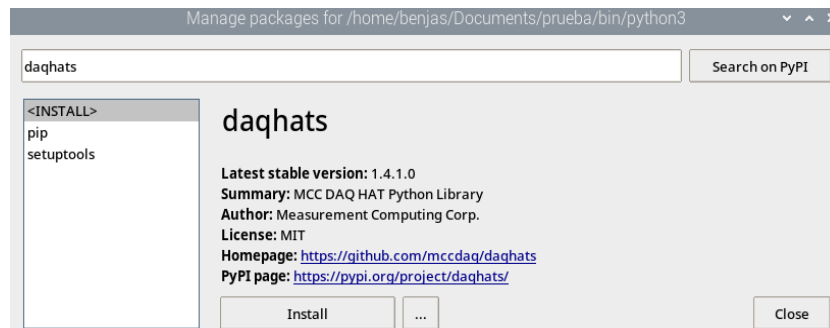
Programa de ejemplo en el archivo *prueba.py*



Para ejecutar el programa en *prueba.py*, se debe agregar al ambiente de programación la librería *daqhats* como un paquete. Para este propósito, mediante la opción *Tools/manage packages...* de Thony, se instala la librería *daqhats* de la Figura 5.

### Figura 5

#### Instalación de la librería *daqhats*



El archivo de ejemplo distribuido por el fabricante contiene algunos detalles que deben ser reparados para que el programa se ejecute sin contratiempos. En este sentido, se hacen los siguientes cambios al archivo *prueba.py* y se ejecuta el programa:

Línea 16 se enmascara la instrucción:

```
#from daqhats_utils import select_hat_device
```

Línea 71 se enmascara la instrucción:

```
#address = select_hat_device(HatIDs.MCC_152)
```

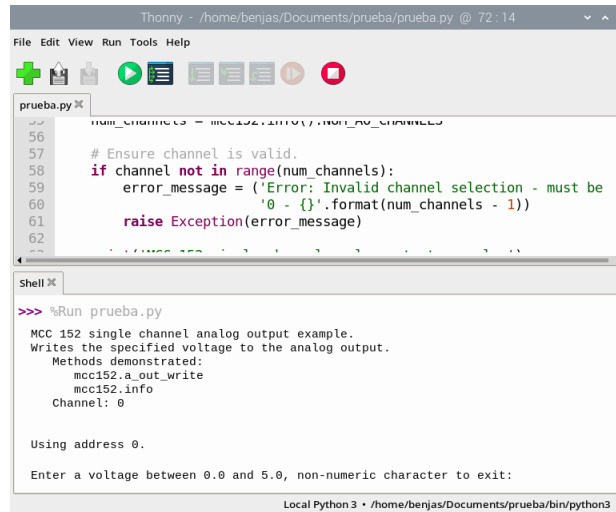
Se agrega la línea 72:

```
address=0
```

Entonces, el programa se ejecuta y éste solicita un valor numérico entre 0 y 5 V para generarlo en el canal 0 del HAT MCC152, como se muestra en la Figura 6. Se ingresa un número y se genera el voltaje como se muestra en la Figura 7. En esta situación, se completa la verificación de las configuraciones y se puede crear una aplicación propietaria en Python.

**Figura 6**

*Programa prueba.py ejecutándose libre de errores*



**Figura 7**

*Comprobación del voltaje generado 1.5 V leído con multímetro como 1.498 V*

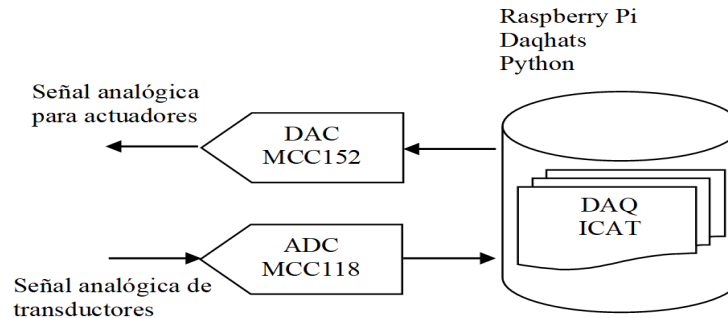


## 2.4 APLICACIÓN PARA LA ADQUISICIÓN Y GENERACIÓN DE SEÑALES

Basado en la configuración de *hardware* y la metodología de *software* descritas en las secciones anteriores, se desarrolló un *software* específico para implementar un sistema DAQ sobre la plataforma Raspberry Pi, como el que se muestra en la Figura 8.

**Figura 8**

*Sistema DAQ utilizando Raspberry Pi y HATS DAC y ADC*



El instrumento de la Figura 8 se puede utilizar como punto de partida simple para elaborar dispositivos más complejos en donde se tengan requerimientos específicos de una aplicación particular.

El esquema de la Figura 8 muestra una Raspberry Pi configurada con la librería de paquetes daqhats y el entorno virtual de programación Python, ejecutando código de diseño específico llamado DAQ ICAT para la generación y adquisición de señales eléctricas mediante HATS DAC y ADC. Las señales analógicas del mundo real pueden ser aplicadas a diversos dispositivos actuadores o digitalizadas provenientes de diversos dispositivos transductores.

### 2.4.1 PROGRAMA DAQ ICAT

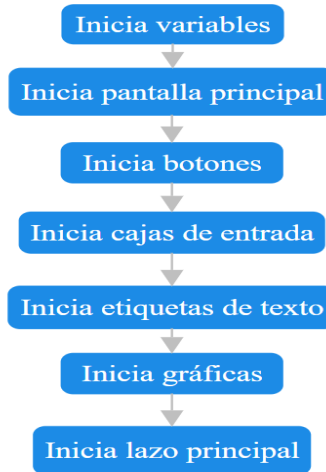
Para plantear el programa DAQ ICAT como un diagrama de flujo, se debe tener en cuenta que un programa que utiliza la interfaz gráfica kinter es un manejador de eventos que sólo refleja los cambios en la pantalla cuando ocurre un evento, ya sea generado por el usuario al interactuar con un control o un evento programado para generar interrupciones. En este sentido, el programa desarrollado se puede esquematizar en un diagrama de flujo como una serie de sentencias que atienden los diversos eventos. En el presente caso, los tres eventos principales de la pantalla de la Figura 9 son la atención a los botones *Inicio CDA*, *Inicio CAD* y *Fin*. De forma adicional, se programó un evento temporizador que se encarga de actualizar la muestra de salida en el generador de la onda senoidal. A estos cuatro eventos, hay que añadir el evento por omisión que inicia el programa.

La resolución del temporizador en Python es de 1 ms y, debido a esto, la tasa a la cual se pueden escribir salidas analógicas en la tarjeta HAT CDA MCC152 es de 1 ms. Comparando esta resolución temporal,  $1\text{ ms} = 1000\ \mu\text{s}$ , con la tasa máxima de actualización en la MCC152 de  $12\ \mu\text{s}$ , se podrían generar en teoría hasta  $1000 / 12 = 83$  muestras cada milisegundo. El programa entonces se puede describir a partir de las 5 siguientes funciones que atienden los eventos asociados:

1. Atención al evento generado por iniciar el programa, *init*. La función *init* se encarga de iniciar la apariencia de la pantalla principal, como se muestra en la Figura 9.

**Figura 9**

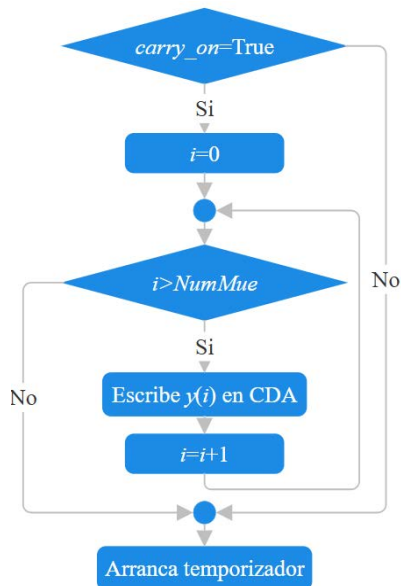
Diagrama de flujo de la función *init*



2. Atención al evento generado por el temporizador, *update*. La función *update* escribe el número de muestras de un periodo de la señal senoidal, almacenado y precalculado en  $y(i)$ , en el DAC de la tarjeta HAT MCC152, *NumMue*, siempre y cuando la bandera *carry\_on* sea verdadera y vuelve a arrancar el temporizador de 1 ms, como se muestra en el diagrama de flujo de la Figura 10.

**Figura 10**

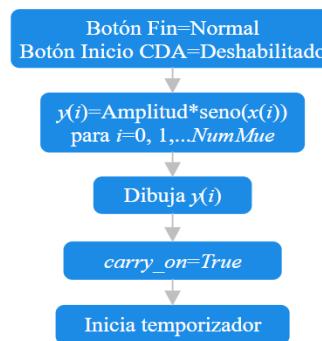
Diagrama de flujo de la función *update*



3. Atención al evento generado por presionar el botón *Inicio CDA*, *startCDA*. La función *startCDA* actualiza el estado de los botones, calcula las muestras de la onda senoidal con la amplitud y la longitud especificadas por el usuario en  $y(i)$ , dibuja el gráfico de la onda senoidal, arranca el temporizador y habilita los eventos del temporizador al establecer la bandera  $carry\_on = True$  como se muestra en la Figura 11.

**Figura 11**

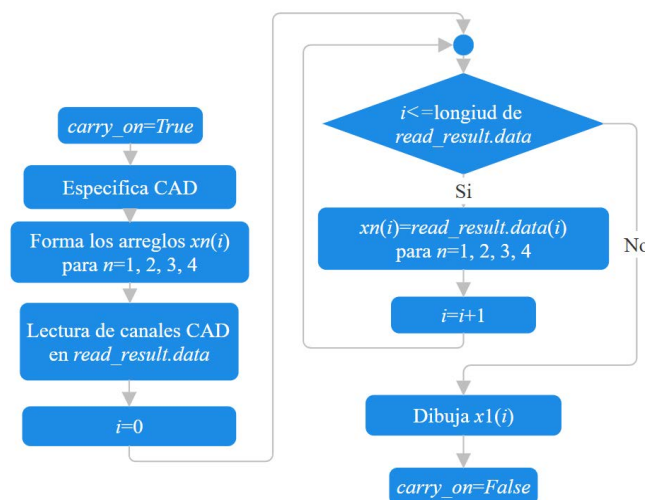
*Diagrama de flujo de la función startCDA*



4. Atención al evento generado por presionar el botón *Inicio CAD*, *startCAD*. La función *startCAD* habilita el temporizador con la variable  $carry\_on = True$ , especifica los parámetros de conversión ADC del HAT MCC118 con el número y tasa de muestras especificadas por el usuario, prepara los arreglos para recibir cuatro canales de conversión ADC del tamaño especificado en el número de muestras, lee los cuatro canales ADC en  $xn(i)$ , obtiene el resultado de la conversión en la variable *read\_result.data*, dibuja el primer canal analógico y deshabilita el temporizador, como se muestra en la Figura 12.

**Figura 12**

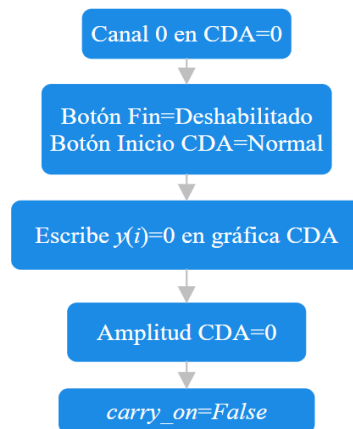
*Diagrama de flujo de la función startCAD*



5. Atención al evento generado por presionar el botón *Fin, stop*. La función *stop* escribe cero volts en el canal cero del HAT MCC152, actualiza el estado de los botones, dibuja la gráfica ADC con ceros, borra el contenido de la caja de entrada para la amplitud ADC y deshabilita el evento del temporizador con la variable *carry\_on = False*, como se muestra en la Figura 13.

**Figura 13**

*Diagrama de flujo de la función stop*



En el Anexo B, se muestra el código fuente Python que implementa lo descrito en esta sección.

## 2.4.2 APLICACIÓN DAQ ICAT

El programa desarrollado, denominado DAQ ICAT, fue implementado en lenguaje Python y estructurado en las siguientes fases (Pidora, 2024):

1. Apariencia gráfica para el usuario (GUI). Se utiliza la librería estándar de Python Tkinter (Nikhil, 2024) para la creación de la interfaz gráfica que contempla botones y etiquetas de entrada.
2. Interfaz con las tarjetas DAQ MCC152 y MCC118. Se utiliza la librería daqhat (GitHub daqhats, 2024) que permite acceder a las funciones de escritura y lectura para la generación de señales analógicas y la adquisición de señales analógicas.
3. Graficación de la información. Se utiliza la librería matplotlib (Matplotlib Development Team. (s.f.)) para el despliegue de gráficos 2D que representen los datos en los ejes coordenados.

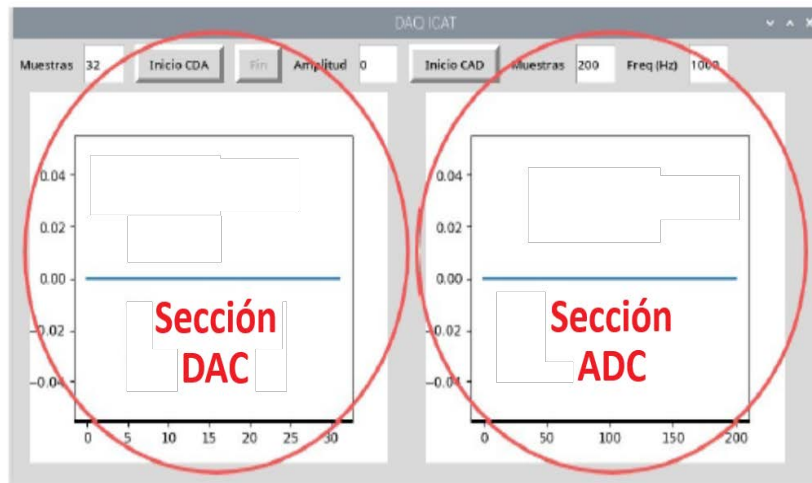
El concepto para desarrollar la pantalla gráfica se pensó que contuviera dos secciones:

1. Generación de una señal senoidal a la máxima tasa de muestreo fija con amplitud y número de muestras variables.
2. Adquisición de cualquier señal con excursión analógica entre 0 y 5V con frecuencia de muestreo y número de muestras variables.

El concepto de la pantalla se muestra gráficamente en la Figura 14.

**Figura 14**

*Concepto para desarrollar el programa de aplicación*



Cada sección contiene un área para el despliegue de datos en una gráfica de dos ejes coordenados, magnitud contra tiempo y controles para especificar los parámetros en los procesos de conversión.

### 3. RESULTADOS

Los resultados obtenidos corresponden a la instrumentación con base en la generación de diferentes números de muestras y a la adquisición de señales analógicas de diferentes frecuencias usando las tarjetas Raspberry Pi, HAT CDA MCC152 y MCC118. A continuación, se presentan los detalles de los resultados.

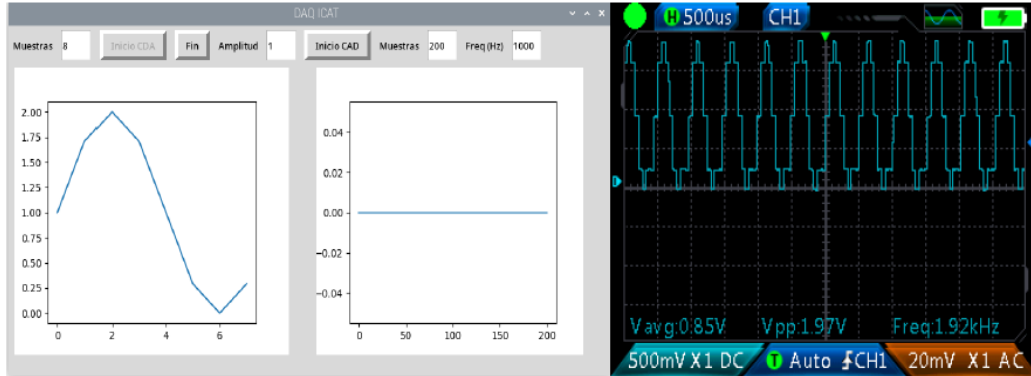
#### 3.1 GENERACIÓN DE SEÑALES ANALÓGICAS

Para comprobar el desempeño del programa DAQ ICAT con respecto a la generación de señales, se especificaron 2 tipos de salidas con número de muestras variables y amplitud constante de 1V. La salida por omisión es del tipo senoidal y el periodo de muestreo es el mínimo que se puede obtener en Python, es decir, 1ms. El programa entonces genera cada 1 ms el número de muestras deseado por el usuario.

La Figura 15 y la Figura 16 muestran las señales analógicas especificadas para generar 8 y 128 muestras de un ciclo de señal senoidal cada 1 ms, tanto en la pantalla del programa DAQ ICAT como en la captura del osciloscopio que se utilizó para comprobar los resultados.

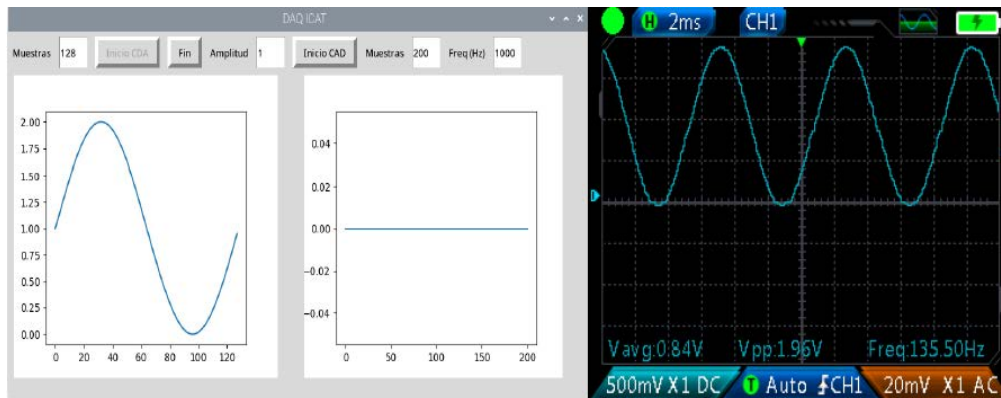
**Figura 15**

*Generación de señal senoidal de 8 muestras por ciclo cada 1 ms*



**Figura 16**

*Generación de señal senoidal de 128 muestras por ciclo cada 1 ms*



Se puede observar que, conforme aumenta el número de muestras, se incrementa la calidad de la forma de onda. Por otra parte, cuando esto sucede, disminuye la frecuencia de la forma de onda senoidal. Estos dos fenómenos deben ser tomados en cuenta en el momento de diseñar un sistema DAQ con aplicación específica.

La Tabla 1 muestra la frecuencia en la forma de onda senoidal que se puede obtener con diferente número de muestras por periodo.

**Tabla 1**

*Frecuencia de la señal senoidal para diferente número de muestras*

Número de muestras	Frecuencia de la señal senoidal (Hz)
8	1920
16	900
32	483
64	242
128	135

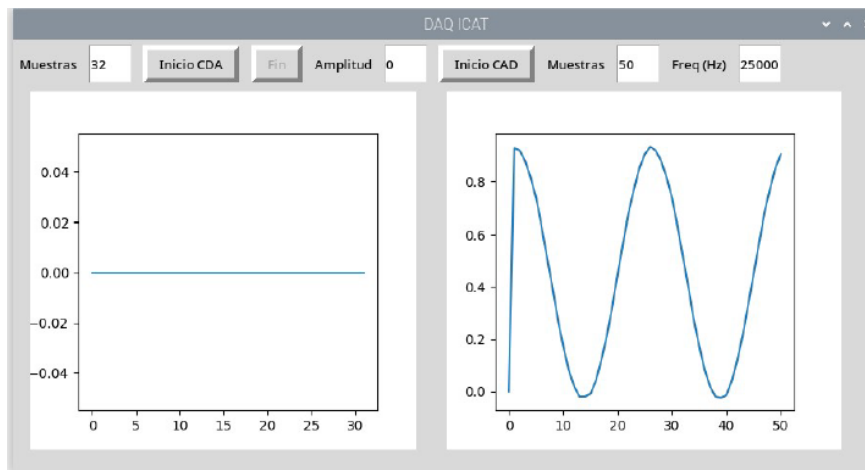
### 3.2 ADQUISICIÓN DE SEÑALES ANALÓGICAS

Para comprobar la sección de adquisición de señales analógicas del programa DAQ ICAT, se generaron diversas formas de onda en un generador de funciones externo y se capturaron los datos con el programa desarrollado.

La Figura 17 y la Figura 18 muestran la pantalla del generador de funciones utilizado en su modo de osciloscopio para obtener los oscilogramas de una señal senoidal y cuadrada.

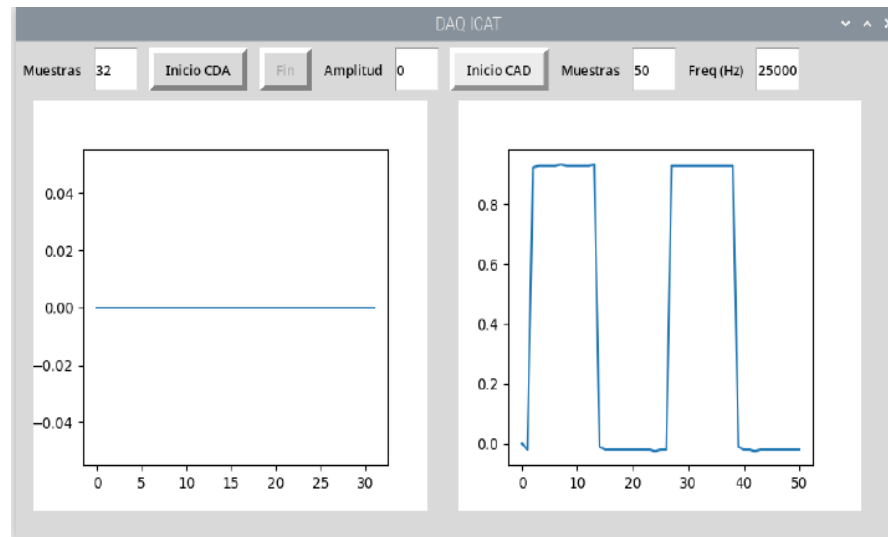
**Figura 17**

*Señal senoidal de 1 kHz*



**Figura 18**

Señal cuadrada de 1 kHz



## 4. CONCLUSIONES

El desarrollo del trabajo permitió demostrar que las plataformas que trabajan con tarjetas de propósito específico, como lo son las Raspberry Pi, constituyen una alternativa viable, eficiente y de bajo costo frente a sistemas tradicionales basados en arquitecturas propietarias. La implementación de un sistema de adquisición y generación de señales (DAQ) sobre esta plataforma mostró una posible integración de soluciones de instrumentación electrónica con base en *software*, adecuadas para aplicaciones académicas, industriales y de investigación.

Entre las principales aportaciones, está la integración de módulos HAT especializados (MCC152 y MCC118) basados en la plataforma Raspberry Pi con un entorno de programación en Python, lo cual permitió no sólo la adquisición de señales analógicas, sino también su procesamiento y visualización en tiempo real mediante una interfaz gráfica. Gracias a esta interfaz, podemos validar la versatilidad de la plataforma y su capacidad de adaptación a distintos requerimientos de instrumentación de medición y control. Además, se evidenció que el uso de *software* libre y bibliotecas especializadas reduce significativamente los costos de implementación y facilita el desarrollo de soluciones personalizadas, promoviendo la independencia tecnológica y el acceso a herramientas avanzadas de instrumentación.

Respecto a los resultados obtenidos, se comprobó que la máxima tasa de muestreo en la tarjeta MCC118 es de 25,000 muestras sobre segundo, a diferencia del dato que proporciona el fabricante de 100,000 muestras sobre segundo. Por lo anterior, se presenta un desempeño adecuado en la generación de señales, donde la calidad de la forma de onda depende directamente del número de muestras utilizadas, así como en la adquisición de señales externas, logrando capturar correctamente diferentes tipos de ondas. Estos resultados validan la funcionalidad del sistema propuesto y su aplicación en entornos reales. Durante el desarrollo del *software* DAQ en un instituto de investigación, se comprobó que las soluciones

de Adafruit constituyen una alternativa a plataformas propietarias. Asimismo, se puede afirmar que las aplicaciones para la Raspberry Pi ofrecen un entorno de confiabilidad y variedad.

El sistema DAQ desarrollado puede ser de utilidad en cursos introductorios al procesamiento digital de señales, en donde se exponen conceptos como el teorema de muestreo, según los conceptos de Nyquist, o la introducción a las señales analógicas y digitales, de gran utilidad en los cursos de sistemas de comunicaciones electrónicos.

Se ha observado que las herramientas utilizadas en el desarrollo de este proyecto también son de utilidad en los cursos de programación Python y de sistemas embebidos, ya que los alumnos pueden experimentar con conceptos como la configuración del ambiente de programación y desarrollo de aplicaciones seguras que no afecten el desempeño de los sistemas operativos. Un ejemplo de esto es la configuración de ambientes virtuales como el expuesto en este reporte técnico.

Una limitación experimentada en el desarrollo de este proyecto fue la tasa de muestreo en el temporizador disponible en el ambiente de programación Python operando en el sistema operativo Raspberry Pi. En este caso, la resolución del temporizador utilizado es de 1ms, que resulta ser el estándar en la mayoría de las aplicaciones. No obstante, en otros ambientes de programación, como por ejemplo Visual C++ programando en ambiente Windows, es posible operar temporizadores de alta resolución más apropiados para aplicaciones más demandantes de recursos, como pueden ser los videojuegos. En este sentido, un trabajo a futuro podría consistir en explorar la posibilidad de utilizar o desarrollar temporizadores de alta resolución para Raspberry Pi.

Así pues, el trabajo no solo cumple con los objetivos planteados, sino que también abre la puerta a futuras líneas de investigación y desarrollo en el área de instrumentación electrónica basada en sistemas abiertos, ya que el *software* descrito en el presente reporte técnico sirve como base para elaborar aplicaciones concretas de procesamiento y control digital de señales. El sistema desarrollado puede considerarse como una plataforma escalable, susceptible de ampliación hacia aplicaciones más complejas, tales como sistemas de monitoreo remoto, procesamiento digital de señales en tiempo real o integración con tecnologías IoT. No se alcanzó el mínimo tiempo de escritura de 12  $\mu$ s especificado por el fabricante de la MCC152, debido a que el tiempo de escritura en lenguaje Python es superior; no obstante las resoluciones y las excursiones analógicas son las esperadas a las que proporciona el fabricante en sus hojas de datos.

#### **Declaración de contribución de autoría**

**Benjamín Valera Orozco:** Instrumentación electrónica; diseño e implementación de proyecto; programación de la interfaz.

**Gerardo Antonio Ruíz Botello:** Metrología y gestión de la calidad; coordinador de diseño y desarrollo. Coordinación y planificación.

**Karen Lucero Roldán Serrato:** Cómputo aplicado; integración y creación de metodología. Integración de las componentes del proyecto

## REFERENCIAS

- AG Electrónica SAPI de CV. (2024, 13 de mayo). *HAT AD de alta precisión para Raspberry Pi: ADC ADS1263 de 10 canales y 32 bits (SKU18983)* [manual técnico]. <https://agelectronica.lat/pdfs/textos/S/SKU18983.PDF>
- Arif, R., Wijaya, S. K., Prawito, & Gani, H. S. (2018, 01-03 de mayo). *Design of EEG data acquisition system based on Raspberry Pi 3 for acute ischemic stroke identification* [ponencia]. 2018 International Conference on Signals and Systems (ICSigSys), Indonesia. <https://doi.org/10.1109/ICSIGSYS.2018.8372771>
- Bengtsson, L. (2024). ADCs and Sampling [capítulo]. En *Electrical Measurement Techniques* (pp. 229-265). Springer. [https://doi.org/10.1007/978-981-99-8187-8\\_11](https://doi.org/10.1007/978-981-99-8187-8_11)
- bluemoon16. (2023, 24 de mayo). *Data Acquisition System with Raspberry Pi* [publicación en foro en línea]. Raspberry Pi Forums. <https://forums.raspberrypi.com/viewtopic.php?t=351826>
- Cassel Barbosa, G. H., Varanis, M., Delgado, K. M. S., & Oliveira, C. de. (2020). An acquisition system framework for mechanical measurements with Python, Raspberry-Pi and MEMS sensors. *Revista Brasileira de Ensino de Física*, 42, e20200167. <https://doi.org/10.1590/1806-9126-RBEF-2020-0167>
- Chen, L. X., & Liu, Y. T. (2025). Implementation of signal acquisition system (SAS) for large voltage waveform using low-cost Raspberry Pi Pico. *Preprints*. <https://doi.org/10.20944/preprints202505.1574.v1>
- Ellison Mathe, S., Kondaveeti, H. K., Vappangi, S., Vanambathina, S. D., & Kumaravelu, N. K. (2024). A comprehensive review on applications of Raspberry Pi. *Computer Science Review*, 52, 100636. <https://doi.org/10.1016/j.cosrev.2024.100636>
- Gunawan, T. S., Rahman, S. N., Kartiwi, M., & Ihsanto, E. (2020, 23-24 de octubre). *Development of Very Low Frequency Data Acquisition System using Raspberry Pi* [ponencia]. 2020 8th International Conference on Cyber and IT Service Management (CITSM), Indonesia. <https://doi.org/10.1109/CITSM50537.2020.9268882>
- Jaryd. (2023, 7 de diciembre). *Using virtual environments in Thonny on a Raspberry Pi*. Core Electronics. <https://core-electronics.com.au/guides/using-virtual-environments-in-thonny-on-a-raspberry-pi/>
- Jolles, J. W. (2021). Broad-scale applications of the Raspberry Pi: A review and guide for biologists. *Methods in Ecology and Evolution*, 12(9), 1562–1579. <https://doi.org/10.1111/2041-210X.13652>
- Kondo, K., Tanno, K., Tamura, H., & Nakatake, S. (2018). Low voltage CMOS current mode reference circuit without operational amplifiers. *IEICE Transactions on Fundamentals*, 101(5), 748–754. <https://doi.org/10.1587/transfun.E101.A.748>
- Llamas, L. (2025). *Raspberry Pi pinout diagram*. <https://www.luisllamas.es/en/raspberry-pi-pinout/>
- Magalhães, R. (2024, 25 de junio). *Cómo leer señales analógicas en Raspberry Pi usando ADS1015/ADS1115*. Compraco. <https://compraco.com.br/es/blogs/tecnologia-e-desenvolvimento/como-ler-sinais-analogicos-no-raspberry-pi-usando-ads1015-ads1115>
- Martínez-Quintero, J. C., Estupiñán-Cuesta, E. P., & Rodríguez-Ortega, V. D. (2019). Raspberry Pi 3 RF signal generation system. *Visión Electrónica*, 13(2). <https://doi.org/10.14483/22484728.15160>

- Matplotlib Development Team. (s.f.). *Matplotlib 3.9.2 documentation*. <https://matplotlib.org/stable/>
- Measurement Computing Corporation. (2019). *MCC 152 voltage output and DIO DAQ HAT for Raspberry Pi* [manual técnico]. <https://files.digilent.com/datasheets/DS-MCC-152.pdf>
- Measurement Computing Corporation. (2020). *MCC 118 voltage measurement DAQ HAT for Raspberry Pi: Datasheet (DS-MCC-118)* [manual técnico]. <https://files.digilent.com/datasheets/DS-MCC-118.pdf>
- Measurement Computing Corporation. (2024). *daqhats: MCC DAQ HAT Library for Raspberry Pi (v1.5.0.0)* [repositorio de software]. <https://github.com/mccdaq/daqhats>
- Medina de Andrés, D. (2017). *Diseño e implementación de un sistema de adquisición y procesamiento digital de señales biomédicas* [trabajo de fin de grado]. Universidad Politécnica de Madrid. [https://oa.upm.es/52911/1/TFG\\_DAVID\\_MEDINA\\_DE\\_ANDRES.pdf](https://oa.upm.es/52911/1/TFG_DAVID_MEDINA_DE_ANDRES.pdf)
- Meena, T., Sharma, V., Jhakar, S., & Sharma, S. K. (2019). Raspberry Pi based electromyography signal acquisition and processing system. *Journal of Emerging Technologies and Innovative Research*, 6(4), 69–75. <https://www.jetir.org/papers/JETIR1904E09.pdf>
- Nikhil (2024). *Python Tkinter tutorial*. <https://www.geeksforgeeks.org/python-tkinter-tutorial/>
- Pidora. (2024, 4 de marzo). *Transforma tu Raspberry Pi en una potente estación de procesamiento de señales*. <https://pidora.ca/transform-your-raspberry-pi-into-a-powerful-signal-processing-workstation/>
- Razavi, B. (2021). *Fundamentals of microelectronics*. Wiley.
- Rodríguez Corbo, F. A., Hernández González, A., & Ramírez Beltrán, J. (2018). Adquisición de datos analógicos con alta precisión usando una BeagleBone Black. *RIELAC: Revista Electrónica de Ingeniería, Electrónica, Automática y Comunicaciones*, 39(3), 68–76. <https://dialnet.unirioja.es/descarga/articulo/6676696.pdf>
- Sreejith, A. G., Mathew, J., Sarpotdar, M., Mohan, R., Nayak, A., Safonova, M., & Murthy, J. (2014, noviembre). A Raspberry Pi-Based Attitude Sensor. *Journal of Astronomical Instrumentation*, 3(2). <https://doi.org/10.1142/S2251171714400066>
- XOREN Ingeniería. (2023, 16 de noviembre). Consejo técnico: adquisición de datos en Raspberry Pi con Universal Library para Linux. <https://xoreningeneria.com/2023/11/16/consejo-tecnico-adquisicion-de-datos-en-raspberry-pi-con-universal-library-para-linux/>

## ANEXO A. GLOSARIO

### DAQ—*Data Acquisition System* (Sistema de Adquisición de Datos)

Es uno de los procesos más importantes del sistema y está compuesto por *hardware* y *software* que permite medir variables físicas, acondicionar señales cuyas fuentes son de sensores y convertirlas a un formato digital para su procesamiento. En el tema de instrumentación, es el medio por el cual en un sistema IoT registra en tiempo real un fenómeno físico (Razavi, 2021; Bengtsson, 2024).

### DAC — *Digital-to-Analog Converter* (Convertidor Digital–Analógico)

Un DAC convierte información digital (generalmente en formato binario) en una señal analógica continua, como voltaje o corriente. En instrumentación, se utiliza para generar señales de prueba, controlar actuadores y crear referencias analógicas. Este principio es fundamental para sistemas IoT, ya que permite que un dispositivo produzca salidas analógicas para control o simulación (Kondo *et al.*, 2018).

### ADC — *Analog-to-Digital Converter* (Convertidor Analógico–Digital)

El convertidor ADC transforma una señal analógica en un valor digital cuantificado que puede ser manejado por un microcontrolador o procesador. En instrumentación, se emplea para adquirir señales de sensores; en IoT, permite que los nodos inteligentes midan variables del entorno (temperatura, luz, presión, etc.) (Razavi, 2021; Bengtsson, 2024).

### HAT — *Hardware Attached on Top* (Tarjeta de Expansión para Raspberry Pi)

Un HAT es una tarjeta de expansión estandarizada que se monta directamente sobre la Raspberry Pi para agregar funcionalidades adicionales, como ADC/DAC de precisión, interfaces industriales o módulos de comunicación. En instrumentación, permite convertir la Raspberry Pi en un módulo DAQ; en IoT, permite ampliar el sistema con conectividad LoRa, NB-IoT, GPS u otros (Jolles, 2021).

## ANEXO B. CÓDIGO FUENTE

```
# GUI
from tkinter import *
# Vectores
import numpy as np
# Graficas
from matplotlib.figure import Figure
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
# DAQ DAC Hat
from sys import version_info

from daqhats import mcc152, OptionFlags, HatIDs, HatError
# DAQ ADC Hat
from daqhats import mcc118, OptionFlags, HatIDs, HatError
class mainwindow(Tk):
def __init__(self, title):
import # variables timer
self.carry_on = True
# variables DAC
```

```

self.i=0
self.optionsCDA=OptionFlags.DEFAULT
self.channelCDA=0
self.hatCDA=mcc152(0)
self.length = np.pi * 2
self.x=np.arange(0, self.length, self.length / 32)
self.y = (np.sin(self.x)*0)+0
self.figCDA = Figure(figsize = (4, 3.7), dpi = 100)
self.plotCDA = self.figCDA.add_subplot(111)
# variables ADC
self.optionsCAD=OptionFlags.DEFAULT
self.hatCAD=mcc118(1)
self.y1 = np.arange(201)
self.x1 = self.y1*0.0
self.x2 = self.y1*0.0
self.x3 = self.y1*0.0
self.x4 = self.y1*0.0
self.figCAD = Figure(figsize = (4, 3.7), dpi = 100)
self.plotCAD = self.figCAD.add_subplot(111)
# GUI
self.master = Tk()
self.master.title(title)
self.master.geometry("900x450")
self.window = Frame(self.master, bd=0, relief=FLAT)
self.window.grid( column = 0, row = 0)
self.NumSen = Label(self.window, text = 'Muestras')
self.NumSen.grid(row=0, column=0, sticky='NSWE', padx=5,
pady=5)
self.NumMue = Entry(self.window, width=5, text = '0')
self.NumMue.grid(row=0, column=1, sticky='NSWE', padx=5,
pady=5)
self.NumMue.insert(0, "32")
self.btnCDA = Button(self.window, text = 'Inicio CDA', bd = '5',
command =
self.startCDA)
self.btnCDA.grid(row=0, column=2, sticky='NSWE', padx=5,
pady=5)
self.btnFinCDA = Button(self.window, text = 'Fin', bd = '5',
command =
self.stop)
self.btnFinCDA.grid(row=0, column=3, sticky='NSWE', padx=5,
pady=5)
self.btnFinCDA.config(state=DISABLED)
self.Amplitud = Label(self.window, text = 'Amplitud')
self.Amplitud.grid(row=0, column=4, sticky='NSWE', padx=5,
pady=5)
Self.Ampl = Entry(self.window, width=5, text = '1')
self.Ampl.grid(row=0, column=5, sticky='NSWE', padx=5,
pady=5)
self.Ampl.insert(0, "0")
self.btnCAD = Button(self.window, text = 'Inicio CAD', bd = '5',
command =
self.startCAD)
self.btnCAD.grid(row=0, column=6, sticky='NSWE', padx=5,
pady=5)
self.Muestras = Label(self.window, text = 'Muestras')
self.Muestras.grid(row=0, column=7, sticky='NSWE', padx=5,
pady=5)
self.Num = Entry(self.window, width=5, text = '2')
self.Num.grid(row=0, column=8, sticky='NSWE', padx=5,
pady=5)
self.Num.insert(0, "200")
self.Frecuencia = Label(self.window, text = 'Freq (Hz)')
self.Frecuencia.grid(row=0, column=9, sticky='NSWE', padx=5,
pady=5)
self.Freq = Entry(self.window, width=5, text = '3')
self.Freq.grid(row=0, column=10, sticky='NSWE', padx=5,
pady=5)
self.Freq.insert(0, "1000")
self.plotCDA.plot(self.y)
canvasCDA = FigureCanvasTkAgg(self.figCDA, master=self.
master)
canvasCDA.draw()
canvasCDA.get_tk_widget().place(relx=0.02, rely=0.12)
self.plotCAD.plot(self.x1)
canvasCAD = FigureCanvasTkAgg(self.figCAD, master=self.
master)
canvasCAD.draw()
canvasCAD.get_tk_widget().place(relx=0.51, rely=0.12)
self.master.mainloop()
def update(self):
if self.carry_on:
for j in range(int(self.NumMue.get())):
value = self.y[j]
self.hatCDA.a_out_write(channel=self.channelCDA,
value=value,
options=self.optionsCDA)
self.master.after(1, self.update)
def startCDA(self):
self.btnFinCDA.config(state=NORMAL)
self.btnCDA.config(state=DISABLED)

```

```

self.x=np.arange(0, self.length, self.length / int(self.NumMue.
get()))
Amplitud=float(self.Ampl.get())
self.y = (np.sin(self.x)*Amplitud)+Amplitud
print(self.y)
self.plotCDA.clear()
self.plotCDA.plot(self.y)
canvasCDA = FigureCanvasTkAgg(self.figCDA, master=self.
master)
canvasCDA.draw()
canvasCDA.get_tk_widget().place(relx=0.01, rely=0.12)
self.carry_on = True
self.master.after(1, self.update)
self.btnFinCDA.config(state=NORMAL)
def startCAD(self):
self.carry_on = True
channel_mask=15 #4 canales
samples_per_channel=int(self.Num.get())
scan_rate=int(self.Freq.get())
num_channels = 4
self.hatCAD.a_in_scan_start(channel_mask, samples_per_
channel, scan_rate,
self.optionsCAD)
self.y1 = np.arange(samples_per_channel+1)
self.x1 = self.y1*0.0
self.x2 = self.y1*0.0
self.x3 = self.y1*0.0
self.x4 = self.y1*0.0
total_samples_read = 0
read_request_size = samples_per_channel
timeout = 5.0
while total_samples_read < samples_per_channel:
read_result = self.hatCAD.a_in_scan_read(read_request_size,
timeout)
# Check for an overrun error
if read_result.hardware_overrun:
print('\n\nHardware overrun\n')
break
elif read_result.buffer_overrun:
print('\n\nBuffer overrun\n')
break
samples_read_per_channel = int(len(read_result.data) / num_
channels)
total_samples_read += samples_read_per_channel
print(total_samples_read)
if samples_read_per_channel > 0:
i=0
ii=1
while i<=len(read_result.data)-4:
self.x1[ii]=read_result.data[i]
i=i+1
self.x2[ii]=read_result.data[i]
i=i+1
self.x3[ii]=read_result.data[i]
i=i+1
self.x4[ii]=read_result.data[i]
i=i+1
ii=ii+1
i=0
while i<samples_per_channel:
i=i+1
print(i,self.x1[i])
self.plotCAD.clear()
self.plotCAD.plot(self.x1)
canvasCAD = FigureCanvasTkAgg(self.figCAD, master=self.
master)
canvasCAD.get_tk_widget().place(relx=0.5, rely=0.12)
self.master.after(1, self.update)
self.hatCAD.a_in_scan_cleanup()
self.carry_on = False
def stop(self):
self.hatCDA.a_out_write(channel=self.channelCDA, value=0,
options=self.optionsCDA)
self.btnFinCDA.config(state=DISABLED)
self.btnCDA.config(state=NORMAL)
Amplitud=0
self.y = (np.sin(self.x)*Amplitud)+Amplitud
print(self.y)
self.plotCDA.clear()
self.plotCDA.plot(self.y)
canvasCDA = FigureCanvasTkAgg(self.figCDA, master=self.
master)
canvasCDA.draw()
canvasCDA.get_tk_widget().place(relx=0.01, rely=0.12)
self.Ampl.delete(0, 'end')
self.Ampl.insert(0, "0")
self.carry_on = False
if __name__ == '__main__':
mainwin = mainwindow("DAQ ICAT")

```

# Sistema para la evaluación de resúmenes de trabajos libres y carteles en la Escuela Nacional de Ciencias Forenses

*System for reviewing abstracts at the Escuela Nacional de Ciencias Forenses*

## Información del reporte:

Licencia Creative Commons



El contenido de los textos es responsabilidad de los autores y no refleja forzosamente el punto de vista de los dictaminadores, o de los miembros del Comité Editorial, o la postura del editor y la editorial de la publicación.

Para citar este reporte técnico:

Flores Aguilar, L.F. (2026). Sistema para la evaluación de resúmenes de trabajos libres y carteles en la Escuela Nacional de Ciencias Forenses. *Cuadernos Técnicos Universitarios de la DGTIC*, 4(2), páginas (148 - 164). <https://doi.org/10.22201/dgtic.30618096e.2026.4.2.158>

## Luis Floriberto Flores Aguilar

Dirección General de Cómputo y de  
Tecnologías de Información y Comunicación  
Universidad Nacional Autónoma de México

[luis.flores@unam.mx](mailto:luis.flores@unam.mx)

ORCID: 0009-0005-1941-8460

## Resumen

La evaluación de trabajos académicos es una tarea importante en la organización del Congreso de Ciencias Forenses ya que permite definir los carteles y presentaciones orales que serán programados en este evento. Para facilitar todas las actividades que implican este proceso, se diseñó, programó, configuró e implementó un sistema capaz de gestionar toda la información de manera centralizada con la infraestructura existente en la Escuela Nacional de Ciencias Forenses. Para el desarrollo de este sistema, se utilizaron herramientas de *software* libre como PHP, Bootstrap y MariaDB en un servidor con sistema operativo Linux. La realización implicó el análisis de extensiones y herramientas para proporcionar seguridad, rendimiento y funcionalidad a nivel de programación, servidor y base de datos. Finalmente, la implementación de este sistema desarrollado a la medida mejoró significativamente la eficiencia y tiempos de respuesta, así como también la reducción en las cargas de trabajo para la evaluación de trabajos académicos y la toma de decisiones por parte del comité académico.

## Palabras clave:

Revisión de trabajos académicos, desarrollo de *software* a la medida, sistema evaluativo, programación PHP, sistema web.

## Abstract

*The evaluation of academic papers is an important task in the organization of the Congreso de Ciencias Forenses. To systematize all the activities involved in this process, a system capable of managing all the information centrally with the existing infrastructure at the Escuela Nacional de Ciencias Forenses was designed, programmed, configured and implemented. For the development of this system, free software tools such as PHP, Bootstrap and MariaDB were used on a server with a Linux operating system. The implementation involved analyzing extensions and tools to provide security, performance, and functionality at the programming, server, and database levels. Finally, the implementation of this custom-planned system significantly improved efficiency and response times, as well as reducing the workload for the evaluation of academic work and decision-making by the academic committee.*

**Keywords:** *Academic work review, custom-planned development, evaluative software, PHP programming, web system.*

## 1. INTRODUCCIÓN

Desde su creación en 2013, la Licenciatura en Ciencia Forense ha tenido como objetivo la formación de profesionales y la divulgación académica en las múltiples áreas científicas y sociales, con la finalidad de contribuir al desarrollo de científicos en el ámbito forense. En 2023, la Licenciatura en Ciencia Forense se convirtió en la Escuela Nacional de Ciencias Forenses (ENaCiF), donde se desarrollan múltiples eventos académicos, como congresos, simposios, talleres, cursos y seminarios, con el objetivo de reunir y promover el intercambio académico de las distintas disciplinas forenses.

A partir de la segunda edición del Congreso de Ciencias Forenses, el comité científico agregó las ponencias libres y presentación de carteles como parte de las actividades en el programa. El proceso de evaluación implica la recepción de la información del trabajo, así como los datos del postulante. Una vez realizada la solicitud, se verifica que cumpla los requisitos y se procede a realizar la evaluación. Por último, con el dictamen final, el comité académico decide los trabajos que serán presentados en el Congreso.

Los temas forenses abordados durante los congresos han alcanzado gran impacto, y la recepción de trabajos aumentó de manera considerable. De acuerdo con Adaile Benítez *et al.* (2025), la participación en eventos académicos contribuye a la motivación, ya que es una oportunidad de interacción por parte de los estudiantes con expertos en la materia (ciencias forenses en este caso).

Anteriormente, la revisión y evaluación de trabajos se realizaba de manera manual a través de correo electrónico y con documentos adjuntos, lo que generaba una gran cantidad de archivos y horas de trabajo para poder procesar y mantener organizada la información. Este proceso se volvía repetitivo, poco eficaz y expuesto al sesgo de la información, además de que tomaba mucho tiempo, retrasando otras actividades de la planeación del congreso.

Al considerar el incremento en las propuestas de trabajos recibidas, la necesidad de reducir tanto las cargas de trabajo como el tiempo de respuesta, se planteó la posibilidad de crear un sistema de información para mantener la gestión de las actividades previas al congreso en relación con la organización, evaluación y dictaminación de los carteles y las presentaciones orales. El objetivo del proyecto fue desarrollar la actualización de un sistema para la evaluación de resúmenes de trabajos libres y carteles en la Escuela Nacional de Ciencias Forenses, el cual permite automatizar procesos que involucran el procesamiento de

información para recibir, enviar, evaluar y generar reportes que apoyen y faciliten la toma de decisiones por parte del comité científico en la selección de trabajos a presentar en el congreso de ciencias forenses.

## 2. DESARROLLO TÉCNICO

### 2.1 ANTECEDENTES

Durante los primeros años de vida académica en la Escuela Nacional de Ciencias Forenses, las evaluaciones de los trabajos postulados se realizaban de manera individual. Se enviaba un documento adjunto del resumen por correo electrónico y otro documento con la rúbrica de la evaluación. Para realizar estas actividades, era necesaria la intervención de varias personas que concentraran la información en un solo documento colaborativo y le dieran un seguimiento puntual a cada solicitud y evaluación. Además, existían múltiples factores que podían provocar errores, como el llenado o captura incorrecta de la rúbrica, fallas en el envío de evaluaciones, correos electrónicos duplicados o retrasos en las respuestas por parte del evaluador. Todos estos aspectos podían afectar de manera significativa este proceso, provocando inconsistencias, sesgos o pérdida de información, ya que no se tenía un control sistematizado ni centralizado.

Este proceso repetitivo se volvía complicado de gestionar a medida que el número de evaluadores y de trabajos aumentaba. El tiempo para esta actividad se extendía notablemente y sobrepasaba las fechas establecidas en el plan de trabajo, lo que afectaba directamente la toma de decisiones por parte del comité científico y propiciaba una comunicación confusa con los postulantes al no tener una respuesta en el tiempo acordado en la convocatoria.

Para la quinta edición del Congreso de Ciencias Forenses en 2021, en un primer intento por organizar la información, se realizó la gestión de la información de los trabajos y el comité evaluador en un servidor de base de datos MySQL, con el fin de tener control en el manejo de los datos y generar vistas a través de consultas SQL. Esta acción mejoró la organización de la información, pero no resolvió el problema de gestionar y cumplir con el proceso de la evaluación de trabajos. En la Figura 1 se muestra el proceso de creación de las tablas dentro de esta base de datos.

## Figura 1

### Creación de tablas en la base de datos

```
CREATE TABLE `trabajosc2021` (
  `folio` int(6) UNSIGNED NOT NULL,
  `nombre` varchar(50) NOT NULL,
  `appaterno` varchar(50) DEFAULT NULL,
  `apmaterno` varchar(50) DEFAULT NULL,
  `correo` varchar(100) NOT NULL,
  `nombrea2` varchar(500) DEFAULT NULL,
  `nombreat` varchar(1000) DEFAULT NULL,
  `correocon` varchar(100) NOT NULL,
  `institucional` varchar(250) DEFAULT NULL,
  `instituciona2` varchar(250) DEFAULT NULL,
  `instituciona3` varchar(1500) DEFAULT NULL,
  `titulot` varchar(1000) DEFAULT NULL,
  `tipot` varchar(100) DEFAULT NULL,
  `palabrac` varchar(3000) DEFAULT NULL,
  `tematica` varchar(2000) DEFAULT NULL,
  `resumen` varchar(10000) DEFAULT NULL,
  `referencias` varchar(1500) DEFAULT NULL,
  `enviado` bit(1) DEFAULT NULL,
  `revisado` bit(1) DEFAULT NULL,
  `aceptado` int(2) NOT NULL DEFAULT 0
)

CREATE TABLE `comite` (
  `idc` int(6) UNSIGNED NOT NULL,
  `nombre` varchar(500) NOT NULL,
  `correo` varchar(200) NOT NULL
);
```

También se evaluó la opción de utilizar plataformas de gestión de eventos que contaran con un módulo para automatizar las tareas de evaluación y creación de reportes para visualización de resultados como Congressus, Fourwaves, EasyChair y Oxford Abstracts. Sin embargo, existe la restricción de agregar un límite de trabajos para las versiones gratuitas o, en su caso, el costo de la licencia por módulo aumenta considerablemente para agregar trabajos o generar reportes, además de que las licencias, en la mayoría de los casos, se gestionan por evento.

Una plataforma de código abierto analizada fue OpenReview, ya que permite el envío de trabajos, la asignación automática de revisores, así como la gestión y generación de reportes. No obstante, es necesario seguir un proceso de registro para cada usuario y revisor, además de mantener actualizado su perfil. Por lo anterior, el seguimiento de estas actividades implicaría mantener una carga de trabajo considerable en la revisión de usuarios y perfiles. Otro aspecto importante a considerar es que todo el contenido de los trabajos y la información de los usuarios quedaría resguardado bajo un tercero. Asimismo, se requiere conocimiento previo del uso de la plataforma por parte del usuario para usarla como evaluador o postulante. En la Tabla 1, se puede ver una comparativa entre las distintas plataformas que fueron analizadas:

**Tabla 1**

*Comparativa de plataformas para evaluación de trabajos académicos*

Plataforma	Licencia	Cantidad de trabajos	Envío de resúmenes	Notificaciones por correo	Reportes
Congresus	Comercial	Hasta 400 con la licencia Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Excel
Fourwaves	Comercial (licencia por año)	25 en versión gratuita 100, 200 o ilimitado con la versión pago	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Datos en plataforma
EasyChair	Comercial (licencia por evento)	20 en versión gratuita 60 o ilimitada con versiones de pago	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Excel y Powerpoint
Oxford Abstracts	Comercial	Ilimitado	Sólo con licencia	Sólo con licencia	<input checked="" type="checkbox"/>
OpenReview	Libre	Ilimitado	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sí (Plataforma)

Al finalizar el análisis de diversas alternativas de *software* para la evaluación de trabajos académicos y tras la consulta de la *Guía para la evaluación y selección de soluciones de software*, publicada por la Dirección General de Cómputo y de Tecnologías de Información y Comunicación en el año 2022, se decidió que la mejor opción era recurrir al desarrollo a la medida, que permite que el *software* esté hecho de acuerdo a las necesidades, recursos y restricciones del área donde se elabore (DGTIC UNAM, 2022).

Tras la revisión de diversas opciones, se decidió planificar un *software* de evaluaciones propio. El objetivo de dicho proyecto fue brindar una herramienta automática, centralizada, confiable y disponible para la evaluación de resúmenes académicos del Congreso de Ciencias Forenses y de otros eventos de la escuela.

## 2.2 DISEÑO DEL PROYECTO

Se definió la necesidad de un sistema web que pudiera almacenar al menos ciento cincuenta trabajos, cada uno con su documento del resumen, así como la información de los evaluadores. Es importante que el sistema asegure alta disponibilidad, así como la capacidad de generar informes para la toma de decisiones del comité organizador.

El sistema se debía integrar a la infraestructura existente en la escuela, para así minimizar los gastos y que la administración fuera centralizada, quedando la información bajo resguardo en la propia entidad.

Se definieron los siguientes puntos principales para el desarrollo de un sistema de evaluación de trabajos:

- Evaluación por pares de doble ciego
- Programación de formularios con la rúbrica definida por parte del comité
- Envío por correo electrónico de invitaciones personalizadas a examinadores
- Lista completa de trabajos
- Informes de evaluaciones
- Una opción para rechazar la invitación por parte del evaluador
- Información del formulario para cada trabajo

Título del trabajo

Tipo: presentación oral o cartel

Palabras clave

Documento con el resumen

Referencias

Rúbrica con la evaluación

### **Roles del sistema**

**Administrador.** Se encarga de cargar la información de los trabajos, la lista de revisores compartida por el comité científico y revisar los formularios para las evaluaciones.

**Editor.** Realiza la asignación de resúmenes, puede ser automático por área de especialidad o de manera manual, involucrando hasta dos revisores por trabajo. Tiene acceso a la vista del trabajo con la información completa, además de los datos de cada revisor.

**Revisor.** Se encarga de realizar la(s) evaluación(es) correspondientes que le fueron asignadas.

El administrador carga los resúmenes desde una hoja de Excel con todos los datos solicitados y listados a la base de datos, el resumen de la evaluación puede cargarse en formato PDF o también desde un documento en línea a través de un enlace con vista web.

Para garantizar el anonimato, la vista del evaluador contempla únicamente los datos del trabajo: título, tipo de trabajo, resumen, palabras clave y referencias.

Para la realización de este sistema, se trabajó la metodología para desarrollo de *software* en cascada ya que permite dividir el proyecto en etapas y mantener una línea de trabajo constante de todos los involucrados.

El proyecto se dividió en los siguientes procesos: planeación del sistema, diseño del modelo entidad-relación, definición de recursos, programación del sistema, implementación y mantenimiento.

Para el sistema, se identificaron las siguientes tareas: carga de trabajos, invitación por correo electrónico, listado de trabajos para cada evaluador, vista completa del trabajo con la rúbrica incluida y generación

de reportes. Para la creación de la base de datos, se utilizó la metodología de diseño, un modelo que proporciona grandes ventajas, como la integridad de datos, la disminución de la redundancia de datos a través de la normalización, las consultas hechas de manera eficiente al aprovechar los recursos con los que se cuentan, la flexibilidad y la escalabilidad, ya que, en el futuro, la demanda de revisión de trabajos irá en aumento.

En este último punto, es fundamental retomar la definición del modelo entidad-relación: “Es una metodología de diseño de bases de datos que consiste en representar a nivel conceptual los datos que soportan el funcionamiento de un sistema. Los componentes básicos de un MER son: Entidades, Atributos y Relaciones” (Araneda, 2022). Para este modelo, se establecieron cuatro entidades principales:

**Estudiante.** Se contempla la información del estudiante o investigador: nombre, apellido paterno, apellido materno, nombre de la institución y correo; además, se agrega un folio que funciona como llave primaria.

**Trabajoscf.** Para esta entidad, se considera la información del trabajo: título, tipo de trabajo (cartel o presentación oral), palabras clave, temática, resumen y referencias. También se tienen atributos de control para el proceso de evaluación: enviado, revisado y aceptado; finalmente, se agrega un folio que funcionará como llave primaria.

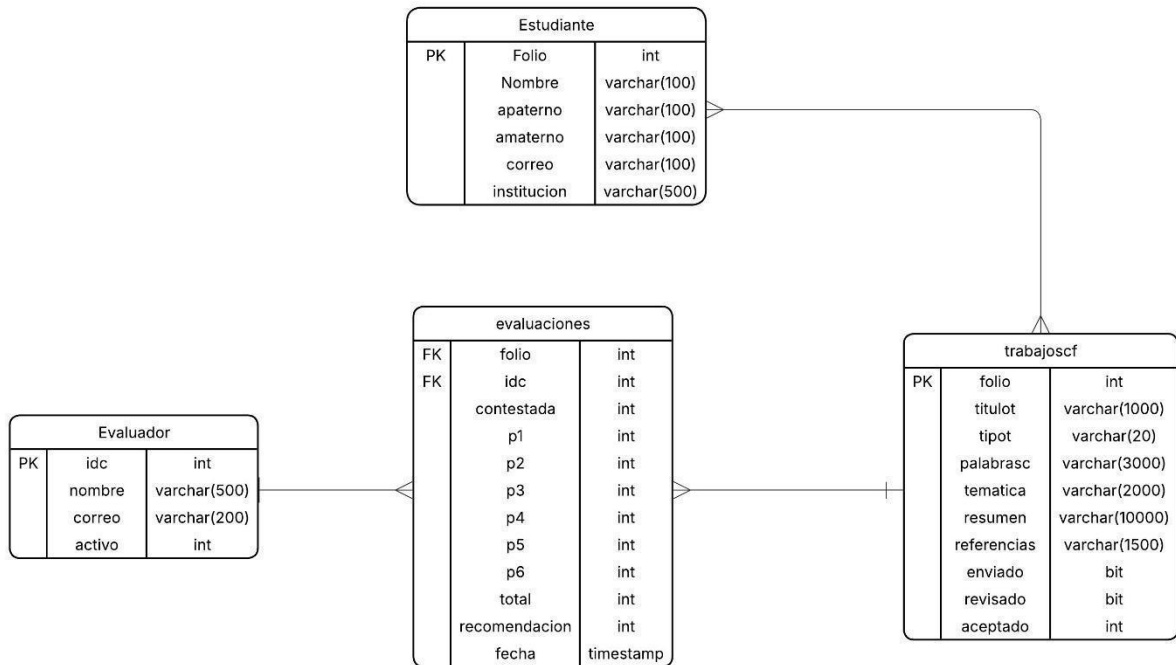
**Evaluador.** Para los evaluadores, se considera el nombre completo, correo electrónico y su área de especialidad. Se agrega idc como folio y llave primaria.

**Evaluaciones.** Esta entidad de asociación se formará a partir de cada evaluación realizada por parte del evaluador. Para esta asociación, se considera la puntuación de cada trabajo, así como la fecha en la que se realizó.

De esta manera, considerando toda la información relevante para el sistema en cuanto estudiantes, investigadores, trabajos, evaluadores y evaluaciones, se generó el diagrama entidad-relación que se muestra en la Figura 2, para definir la estructura del sistema y comenzar a trabajar la base de datos.

**Figura 2**

*Diagrama entidad-relación*



Después de consultar los lineamientos y recomendaciones para la administración de bases de datos publicados por la DGTIC en el año 2022, que establecen que “las áreas universitarias como parte del proceso de creación de la base de datos y sus objetos, deberán realizar un análisis del volumen y crecimiento de los datos, para obtener un dimensionamiento adecuado de la infraestructura, configuración y tamaño de la base de datos a crear” (DGTIC UNAM, 2022), se procedió a la programación del sistema.

Con el objetivo de obtener el máximo rendimiento de la infraestructura con la que cuenta la escuela, así como de reducir gastos de compra e implementación en *hardware*, se decidió utilizar el servidor existente en la ENaCiF para implementar este nuevo sistema: servidor marca Dell modelo PowerEdge R420 con procesador Intel Xeon, 32GB de memoria RAM y una capacidad de almacenamiento de 1TB configurado en modo RAID 1 para garantizar la redundancia y evitar pérdida de información.

El servidor cuenta con un sistema operativo Debian GNU/Linux 11 (bullseye). Al realizar el análisis de las aplicaciones existentes, se encontró que funcionan a través de la plataforma WordPress con PHP versión 8.0 y manejador de base de datos MariaDB versión 10.5. En la parte del servidor web, se utilizó Apache versión 2.4.62.

Un elemento importante para el desarrollo es el sistema manejador de base de datos. Las principales herramientas de *software* libre consideradas fueron MariaDB y PostgreSQL. Tras un exhaustivo estudio, se eligió MariaDB por la influencia del análisis comparativo realizado por Pilicita Garrido *et al.* (2020), quienes afirman que “MariaDB es una versión mejorada de MySQL que ha incorporado nuevos motores

de búsqueda para enfrentar nuevos retos en el procesamiento de grandes volúmenes de datos. La velocidad de estos motores genera que a medida que aumentan los usuarios pueda responder de forma oportuna y mantener un rendimiento aceptable lo que le permite estar a la vanguardia y competir con otros sistemas de gestión de bases de datos”.

Para la elección del lenguaje de programación de este nuevo sistema, se consideraron PHP y Python, ambos lenguajes de código abierto que ofrecen estabilidad, seguridad y compatibilidad con manejadores de base de datos como MySQL, MariaDB y Postgres. En el caso de Python, se puede obtener un mayor rendimiento en áreas específicas, como ciencia de datos o almacenamiento de grandes volúmenes de información. Por otra parte, con PHP es posible enfocarse en el desarrollo web, ya que, a través de este lenguaje y su estructura, se puede realizar mantenimiento, almacenamiento, transferencia y trabajo colaborativo con una alta eficiencia en los tiempos de respuesta, además de su integración con otras herramientas en bases de datos, *frameworks* CSS, servidores web, así como la compatibilidad en diferentes dispositivos. Asimismo, se puede mencionar que importantes plataformas en la gestión de contenidos, como Wordpress, Moodle, Joomla o Drupal, tienen gran porcentaje de desarrollo en PHP. Tras realizar un análisis de estos dos lenguajes, es evidente que son una opción atractiva para el desarrollo de sistemas web, considerando también la comparativa realizada por Domínguez Chávez en el año 2024: “PHP sigue siendo una opción sólida para el desarrollo web debido a su accesibilidad y a la amplia disponibilidad de desarrolladores, mientras que Python está en una trayectoria de crecimiento que ofrece ventajas en áreas específicas”. Finalmente, al tener estas consideraciones con respecto al uso, documentación, escalabilidad, configuración y gestión, el equipo de sistemas optó por utilizar PHP versión 8.0.

Para la integración de esquemas, vistas, y diseño del sistema, se consideraron Bootstrap, Tailwind, Bulma y Foundation como *framework* CSS de código abierto. Estas herramientas proporcionan una amplia biblioteca con componentes predefinidos para presentar el contenido a través de una interfaz adaptable al usuario; asimismo, pueden ajustar estilos para cada una de las vistas del sistema y mantener una estructura estándar del sistema web con una práctica integración. En esta ocasión, se eligió Bootstrap, pues su principal ventaja es que dispone de HTML, CSS y Javascript en este *framework*. Debido a su alto uso, capacidad para crear interfaces y popularidad, cuenta con una amplia biblioteca de componentes, documentación y temas definidos; además, su estructura es flexible y mantiene un diseño responsivo por defecto para adaptarse a la gran mayoría de dispositivos, como lo son computadoras de escritorio, portátiles, tabletas o smartphones (Martínez Ramírez *et al.*, 2024). Otro aspecto importante a considerar es que, con las funciones de Javascript, es posible agregar funciones interactivas o dinámicas para el sistema web.

## 2.3 PRUEBAS DE FUNCIONAMIENTO

Con estas herramientas, se logró realizar la programación del listado de los trabajos para cada evaluador y los formularios con la vista del trabajo en versión web; además, se integró la rúbrica propuesta por el comité científico y se tuvo disponible una vista en formato de listado con todos los trabajos disponibles para una revisión previa para el comité organizador. En la Figura 3, se presenta la lista con los trabajos a revisar con la que cuenta cada evaluador:

### Figura 3

*Vista con el listado de trabajos*



Licenciatura en Ciencia Forense  
Escuela Nacional de Ciencias Forenses  
VII Congreso de Ciencias Forenses:  
Resúmenes

CIENCIA FORENSE UNAM

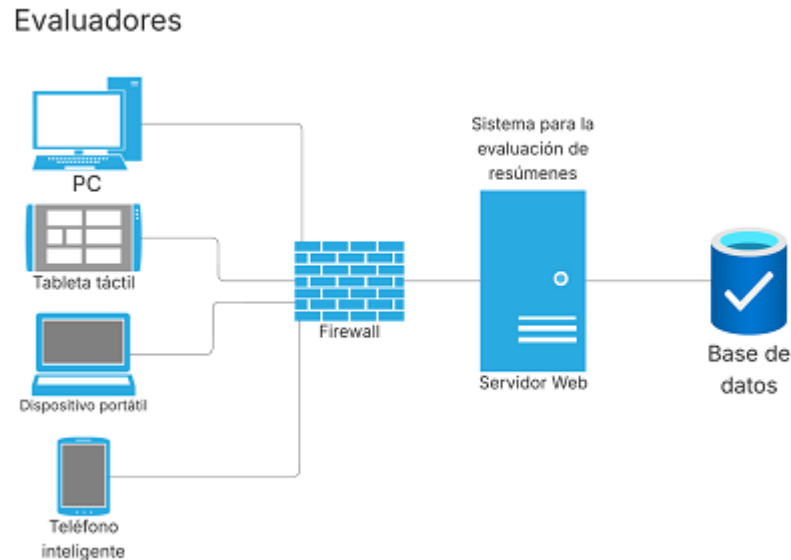
## Resúmenes VII Congreso de Ciencias Forenses UNAM 2025

- 1 Criminología y victimología, Psicología y Psiquiatría Forense [Las sentencias de la Corte](#) cartel
- 2 Criminología y victimología, Derecho probatorio, Psicología y Psiquiatría Forense [Derechos](#) Análisis jurisprudencial oral
- 3 Medicina Forense [Procedimiento](#)
- 4 Criminología y victimología, Psicología y Psiquiatría Forense [Propuesta](#)
- 5 Gestión [Forense](#) oral

La Figura 4 muestra el diagrama del sistema.

#### Figura 4

##### Diagrama del sistema



Para realizar la conexión a la base de datos, se utilizó la clase `mysqli`, la cual, a través de sus funciones y el manejo de PHP, permitió el tratamiento y procesamiento adecuado de los datos de acuerdo con las tareas requeridas: agregar, eliminar, mostrar y modificar registros en las tablas.

## 2.4 ARQUITECTURA Y SEGURIDAD

Un elemento importante para mantener la seguridad y la integridad en los datos es la conexión entre el lenguaje PHP y la base de datos. Para esta tarea, se analizaron dos herramientas: MySQL Improved (MySQLi) y PHP Data Objects (PDO). Estas extensiones de PHP brindan un entorno seguro para realizar la conexión de la base de datos, de tal forma que permite prevenir ataques de inyección SQL y mantiene un rendimiento constante. Además, ambas vienen instaladas de manera predeterminada junto con PHP. A pesar de la compatibilidad que tiene PDO con múltiples manejadores de bases de datos como Postgres, SQLite, Oracle o SQLServer, se optó por MySQLi, ya que ofrece estilo de programación con procedimientos y orientado a objetos, así como también el uso en otras aplicaciones dentro del servidor que se tiene en la escuela.

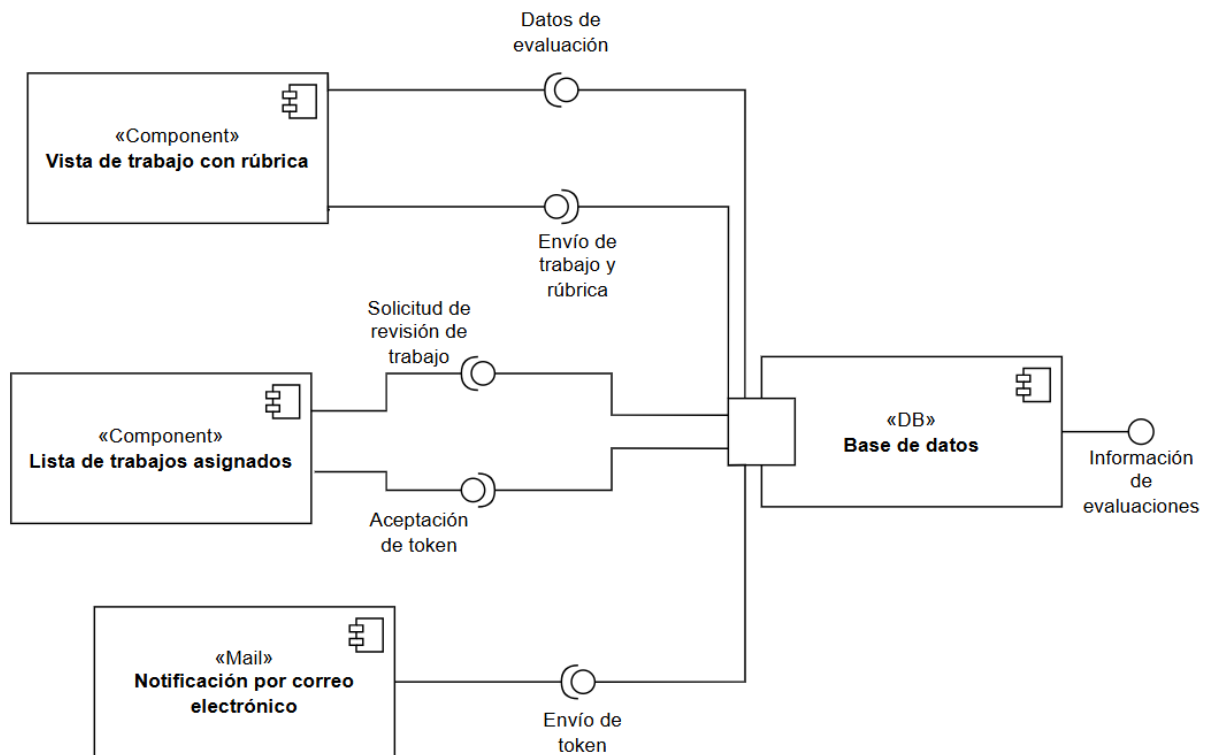
En la interacción del usuario dentro del sistema, se analizó el uso de sesiones PHP a través de usuario y contraseña, así como también una sesión basada en el uso de *tokens*. En la parte de sesiones de PHP, debe existir siempre el intercambio de usuario y contraseña y, para cada sesión, se deben fijar distintos parámetros a nivel del servidor. El uso de recursos incrementa con la conexión simultánea de usuarios y también se pueden presentar errores si el cliente tiene deshabilitadas las *cookies*. Con el uso de *tokens*, se obtiene una mejora en el rendimiento ya que el *token* se procesa una vez y brinda una sesión larga,

disponible para que el usuario realice la evaluación de trabajos; asimismo, se pueden tener múltiples usuarios de manera concurrente sin afectar el rendimiento de las otras aplicaciones dentro del servidor. Finalmente, tras el análisis, se decidió el uso de *tokens* en el sistema.

Es importante mencionar que el servidor cuenta con certificado SSL para mantener una comunicación cifrada entre el servidor y el cliente en todo momento. En la Figura 5, se muestra la interacción del evaluador con el sistema.

**Figura 5**

*Diagrama de componentes*



Un aspecto importante por considerar en la presentación del trabajo al evaluador es el de mostrar íntegramente el documento sin perder el formato ni modificar su contenido. Para esto, se utilizó la característica de integrar a través de la etiqueta *iframe* con la vista web del archivo en la aplicación Documentos de Google. De esta manera, se logró mantener la vista original del documento y el usuario final puede analizarlo dentro del sistema de evaluación sin necesidad de descargarlo o instalar programas o extensiones adicionales en el navegador.

Una vez finalizada la programación del sistema, se agregaron los 76 trabajos a evaluar para la VI edición del Congreso de Ciencias Forenses. Se realizaron pruebas en la carga de los trabajos en la base de datos, así como en la vista del documento final para el evaluador, como se muestra en la Figura 6. En conjunto con el comité organizador, se analizó el seguimiento al proceso para la evaluación por pares: envío de

invitación, aceptación o rechazo, vista de los trabajos, vista del trabajo individual con la rúbrica y reporte final de las evaluaciones.

## Figura 6

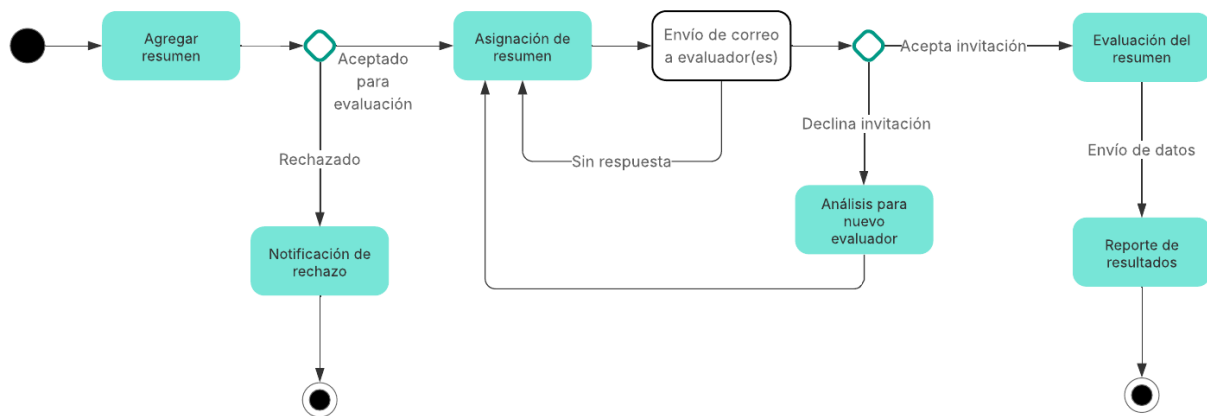
### Vista del documento a evaluar



Con estos elementos, se iniciaron las pruebas al sistema, enviando tres invitaciones para la evaluación de trabajos: envío aceptado, envío rechazado para su reasignación y una vista preliminar de resultados. La Figura 7 muestra el diagrama de estados para este proceso.

**Figura 7**

*Diagrama de estados*



Una vez que se han asignado los trabajos y generado los formularios, el primer paso es enviar un correo electrónico personalizado con un enlace único y un *token* de acceso para ingresar a la vista con los trabajos que van a ser evaluados; de esta manera, se garantiza que el evaluador tendrá el acceso a la evaluación de los trabajos que le fueron asignados. Este esquema de seguridad es funcional, ya que los evaluadores pueden ser expertos invitados de otras instituciones. Además, el proceso de acceso se simplifica y la información se mantiene centralizada, asegurando que los evaluadores sólo tengan acceso al contenido del trabajo propuesto para su análisis.

Para cada trabajo, existirá un formulario al cual se hace referencia con una clave a través del identificador del trabajo y del evaluador. Este proceso permite que las evaluaciones se vayan guardando de manera automática y se pueda consultar la información de todas las evaluaciones en cualquier momento.

Se asignó el puntaje correspondiente para cada criterio de evaluación y, al final, el sistema guarda la calificación asignada y se asigna el resultado sin posibilidad de modificar posteriormente la evaluación emitida.

## 2.5. IMPLEMENTACIÓN

El sistema, con las funciones completas de registro de trabajos, envío de invitaciones, validación de acceso y evaluación por parte del comité, se utilizó para la VI edición del Congreso de Ciencias Forenses en 2023. Se generaron dos formularios para cada trabajo, cumpliendo con la evaluación por pares de doble ciego y recolectando toda la información en la base de datos para presentarlo en un reporte final para su posterior análisis. Desde entonces, se actualizó y mejoró la versión para el VII Congreso de Ciencias Forenses, celebrado en 2025.

## 2.6. MANTENIMIENTO

Es importante mencionar que, al inicio de cada periodo escolar, el administrador del sistema comprueba las versiones de PHP y Bootstrap con la finalidad de realizar las actualizaciones en las versiones y mantener

de esta manera la integridad del sistema para su uso de manera segura. El sistema operativo instalado en el servidor cuenta con actualizaciones periódicas, así como con políticas de seguridad previamente establecidas para mantener su integridad. Asimismo, se renueva anualmente el certificado SSL para mantener habilitada esta importante herramienta de seguridad. Por parte del sistema desarrollado, se analizan las versiones de base de datos y PHP con la finalidad de mantener actualizado el sistema y evitar posibles vulnerabilidades que afecten al funcionamiento del sistema.

Para la evaluación de trabajos propuestos en el VII Congreso de Ciencias Forenses realizado en 2025, se actualizaron a las versiones 8.3 de PHP y la versión 4 de Bootstrap, con la finalidad de mantener la seguridad del sistema y evitar posibles vulnerabilidades. Una vez comprobadas estas versiones, se realizaron pruebas de conectividad a través de *scripts*, utilizando las funciones propias de `mysqli`: `mysqli_connect` y `mysqli_connect_error`. De esta manera, se logra verificar la conexión del sistema web con la base de datos. En la Figura 8 se muestra el proceso de conexión a la base de datos.

### Figura 8

*Prueba de conexión a la base de datos*

```
<?php
/*Parámetros para la base de datos*/
$host = "host";
$user = "usuariobd";
$pass = "*****";
$db = "nombre_base_datos";

/*Conexión a la base de datos*/
$conexion = mysqli_connect($host, $user, $pass, $db);

/*Prueba la conexión*/
if (!$conexion) {
    die("Conexión fallida: " . mysqli_connect_error());
}
echo "Conexión exitosa a la base de datos.";
mysqli_close($conexion);
?>
```

```
luisf@          :condb$ php conexion.php
Conexión exitosa a la base de datos.
```

En la parte del sistema, se realizaron pruebas de funcionalidad para verificar la vista y comprobar el correcto funcionamiento dentro del sistema; se revisaron los trabajos asignados por evaluador, así como la evaluación del trabajo. Estas pruebas se hicieron tanto desde una computadora de escritorio como desde un dispositivo móvil para verificar que el sitio funcionara de manera responsiva.

### 3. RESULTADOS

En un inicio, el proceso de evaluación de trabajos postulados para su presentación en el Congreso de Ciencias Forenses requería un tiempo considerable para completarse. En 2019 y 2021, se debía planear con, al menos, tres meses de anticipación, ya que las actividades se realizaban de manera manual: la recepción de resúmenes, el envío de correos, las notificaciones, los recordatorios, los documentos adjuntos del resumen con la rúbrica, la recepción de correos, así como la organización de la información y trabajos, las cuales eran actividades repetitivas y precisas.

A partir del uso de este sistema web, una vez que cierra la convocatoria y se asignan los trabajos, es posible realizar todo este proceso en un mes. Lo anterior permitió reducir los tiempos de evaluación y, de esta manera, fue posible extender la convocatoria para la recepción de trabajos. Por otra parte, la sistematización de las tareas requeridas disminuyó las cargas de trabajo y las posibilidades de tener errores, al mismo tiempo que los sesgos en la información disminuyeron considerablemente gracias a la centralización de la información en la base de datos con la gestión del sistema.

Para la VI edición del Congreso de Ciencias Forenses, se evaluaron 32 resúmenes para cartel y 44 resúmenes para presentación oral con este sistema, de los cuales, 30 carteles y 29 trabajos libres estuvieron presentes para esta nueva celebración del congreso. Además, se integró la función en la cual el evaluador tiene la opción de declinar su participación.

En 2025, para la VII edición del Congreso de Ciencias Forenses, se evaluaron 43 carteles y 49 trabajos orales con un total de 102 evaluadores; de éstos, se aceptaron 37 carteles y 39 presentaciones orales respectivamente.

La capacidad del sistema puede variar en función de los recursos en *software* y *hardware*, si bien para los Congresos en la ENaCiF, el máximo de trabajos evaluados han sido 103, el número de trabajos puede aumentar de acuerdo con la planificación y diseño, ya que no existe una alta concurrencia que implique uso excesivo de los recursos en el servidor.

### 4. CONCLUSIONES

Se desarrolló e implementó un sistema propio de la Escuela Nacional de Ciencias Forenses para la evaluación de resúmenes. Se logró cumplir con todas las funciones del sistema: carga de trabajos, asignación de trabajos por pares y envío de notificaciones por correo electrónico; asimismo, se logró obtener los reportes de todos los trabajos y verificar su estado (aceptado, rechazado, sin evaluar). Se implementó este sistema para la VI y la VII edición del Congreso de Ciencias Forenses y también, en 2025, para la Reunión Anual de la Asociación Americana de Física, sección México AAPT MX.

Con la aplicación de este sistema sencillo de utilizar, se mejoró la experiencia para el usuario evaluador, así también para los organizadores en la realización de las evaluaciones, reduciendo cargas de trabajo en tareas repetitivas. El sistema proporciona seguridad y resultados visibles con tiempos de respuestas óptimos a comparación del proceso utilizado anteriormente. Adicionalmente, es posible adaptar el sistema a otros eventos académicos que requieran la evaluación de trabajos.

Los próximos retos para el uso de este sistema son la mejora en los procesos para la organización de la información y la optimización de recursos en el proceso de la información y obtención de reportes.

Además de esto, sería viable generar un formato estándar para tener la posibilidad de transferirlo, adaptarlo e implementarlo en otras entidades de acuerdo con los recursos en *software* y *hardware* con los que se cuenten.

## AGRADECIMIENTOS

Agradezco a la Escuela Nacional de Ciencias Forenses por la oportunidad de trabajar en este proyecto, así como también al Comité Académico del Congreso de Ciencias Forenses, especialmente a la Dra. Zoraida García Castillo por todo el apoyo brindado durante esta etapa. De la misma forma, agradezco a todo el equipo editorial de Cuadernos Técnicos Universitarios de la DGTIC por su apoyo y asesoría.

## REFERENCIAS

- Adaile Benítez, N. T., Camacho Vázquez, M. R., Cárdenas Castellanos, J., Pérez Méndez, J., & Martínez García, J. E. (2025). Plataforma web para la gestión eficiente de eventos académicos utilizando la metodología RAD. *Revista Multidisciplinaria Voces de América y el Caribe*, 2(1), 585-616. <https://doi.org/10.69821/REMUUVAC.v2i1.145>
- Araneda, P. (2022). *Base de datos. El camino de los datos a la información*. Opendataclinica. <https://bookdown.org/paranedagarcia/database/index.html>
- Dirección General de Cómputo y de Tecnologías de Información y Comunicación [DGTIC], UNAM. (2022). *Guía para la evaluación y selección de soluciones de software*. REDTIC. <https://www.red-tic.unam.mx/guia-software>
- Dirección General de Cómputo y de Tecnologías de Información y Comunicación [DGTIC], UNAM. (2022). *Lineamientos y recomendaciones para la administración de bases de datos*. REDTIC. <https://www.red-tic.unam.mx/index.php/lineamientos-bases#vii-sobre-la-creacion-y-configuracion-de-bases-de-datos>
- Domínguez Chávez, J. (2024, septiembre). *Análisis de la Popularidad de PHP y Python en el Desarrollo de Software*. Researchgate. [https://www.researchgate.net/publication/383977802\\_Analisis\\_de\\_la\\_Popularidad\\_de\\_PHP\\_y\\_Python\\_en\\_el\\_Desarrollo\\_de\\_Software](https://www.researchgate.net/publication/383977802_Analisis_de_la_Popularidad_de_PHP_y_Python_en_el_Desarrollo_de_Software)
- Martínez Ramírez, V., Luciano Machorro, T., Martínez Rabanes, S., Osorio Ramírez, E. A., & Reyes Ramos, J. J. (2024). Optimización web móvil: El poder de bootstrap en el desarrollo adaptativo. *IPSUMTEC*, 7(2), 284-294. <https://doi.org/10.61117/ipsumtec.v7i2.337>
- Pilicita Garrido, A., Borja López, Y. & Gutiérrez Constante, G. (2020). Rendimiento de MariaDB y PostgreSQL. *Revista Científica y Tecnológica UPSE*, 7(2), 9-16. <https://doi.org/10.26423/rctu.v7i2.538>

# Manejo distribuido de matrices a gran escala en un equipo HPC

*Large-scale distributed array management on an HPC system*

## Información del reporte:

Licencia Creative Commons



El contenido de los textos es responsabilidad de los autores y no refleja forzosamente el punto de vista de los dictaminadores, o de los miembros del Comité Editorial, o la postura del editor y la editorial de la publicación.

Para citar este reporte técnico:

Durán Chavesti, A. (2026). Manejo distribuido de matrices a gran escala en un equipo HPC. *Cuadernos Técnicos Universitarios de la DGTIC*, 4 (2), páginas (165 - 184). <https://doi.org/10.22201/dgtic.30618096e.2026.4.2.150>

## Adrián Durán Chavesti

Instituto de Investigaciones en Matemáticas  
Aplicadas y en Sistemas

Universidad Nacional Autónoma de México

[adrian.chavesti@iimas.unam.mx](mailto:adrian.chavesti@iimas.unam.mx)

ORCID: 0000-0002-0542-1627

## Resumen

La necesidad de computación de alto rendimiento se refiere a la práctica de utilizar recursos de cómputo más potentes que una computadora personal o servidor típico para abordar problemas complejos que requieren grandes cantidades de procesamiento. Una gran parte de las operaciones a resolver son matrices a gran escala. Por esta razón, se implementó un ambiente de clúster de servidores con la herramienta "Servidor Paralelo de MATLAB", que permitió mostrar cómo es que se pueden manejar matrices de gran tamaño de forma paralela y distribuida haciendo uso de un equipo de cómputo de alto rendimiento y el lenguaje de alto nivel MATLAB.

Se logró manejar y distribuir un par de matrices de 49,000 x 49,000 elementos. En contraste, una computadora personal típica con 16 GB de memoria sólo puede manejar un par de matrices de 23,000 x 23,000 elementos.

Este trabajo permitió verificar la facilidad y flexibilidad con la cual se puede escalar un problema usando servidores conectados en red con la herramienta "Servidor Paralelo de MATLAB". Esto se logra usando el mismo algoritmo y añadiendo más servidores a la configuración del clúster. En esta implementación, se usó un sistema de tres nodos conectados en una red local, por lo que, si quiere reproducir, se recomienda, antes de empezar la instalación, tener a la mano la licencia para poder distribuir MATLAB.

El nivel de escalado del clúster va a estar relacionado directamente con el número de servidores que se tengan disponibles. En este caso, han sido beneficiados tres usuarios de la línea de investigación en cómputo de alto rendimiento. El sistema operativo usado fue Linux Ubuntu Server 24.04 LTS y el *software* MATLAB R2024a.

**Palabras clave:** Grupo de servidores, cómputo científico, MATLAB distribuido, escalado de matrices, supercómputo.

### Abstract

*The need for high-performance computing refers to the practice of using computing resources more powerful than a typical personal computer or server to address complex problems that require large amounts of processing power. A large part of the operations to be solved involve large-scale matrices; therefore, a server cluster environment was implemented with the "MATLAB Parallel Server" tool. This allowed them to demonstrate how large matrices can be handled in a parallel and distributed manner using high-performance computing equipment and the high-level language MATLAB.*

*It was possible to handle and distribute a pair of 49,000 x 49,000 matrices, whereas a typical personal computer with 16 GB of memory can only handle a pair of 23,000 x 23,000 matrices.*

*This work demonstrated the ease and flexibility with which a problem can be scaled using networked servers with the MATLAB Parallel Server tool, employing the same algorithm simply by adding more servers to the cluster configuration.*

*This implementation used a three-node system connected to a local network, so if one wishes to replicate it, it is recommended to have your MATLAB license on hand before starting the installation.*

*The cluster's scalability will be directly related to the number of available servers. In this case, three users from the high-performance computing research group benefited from the system. The operating system used was Ubuntu Server 24.04 LTS (Linux), and the software used was MATLAB R2024a.*

**Keywords:** Server cluster, scientific computing, distributed MATLAB, matrix scaling, supercomputing.

## 1. INTRODUCCIÓN

La computación distribuida es una disciplina consolidada en la informática y la ingeniería. Ha evolucionado durante los últimos 40 años hasta convertirse en una de las metodologías más importantes para la implementación de los servicios de procesamiento de datos necesarios para prácticamente todas las actividades de la sociedad (Delfino, 2020).

Ingenieros y científicos utilizan MATLAB para analizar y diseñar sistemas y productos. El lenguaje de MATLAB, basado en matrices, es la forma más natural para expresar las matemáticas computacionales (MathWorks, s.f.).

El cálculo matricial es fundamental en numerosas aplicaciones analíticas con uso intensivo de datos, como es el caso de la minería de datos en redes sociales, los sistemas de recomendación, el procesamiento del lenguaje natural, el análisis numérico a gran escala y la física computacional. Por lo anterior, se considera

una parte importante de la construcción de plataformas computacionales para tales problemas. En el ámbito del *big data*, se ha creado una creciente demanda de implementaciones escalables de cálculo matricial en conjuntos de datos masivos.

La multiplicación de matrices es una operación dominante en muchas aplicaciones analíticas de *big data*, pero consume mucho tiempo. Por lo tanto, la optimización del rendimiento es un problema de investigación importante y fundamental (Gu *et al.*, 2017). Asimismo, HouZhen *et al.* (2020) mencionan que las operaciones de matrices a gran escala son fundamentales y, tomando en cuenta a Lim *et al.* (2018), se deben optimizar los algoritmos clásicos para aprovechar todo el potencial de las nuevas características del *hardware*.

Se han propuesto técnicas de gestión automática del paralelismo para minimizar la sobrecarga relacionada con la creación y sincronización de hilos, manteniendo así altos niveles de rendimiento en cargas de trabajo distribuidas (Westrick *et al.*, 2024). Es por ello que el *Toolbox Parallel Server* es una opción que se considera como gestor automático de distribución de trabajos. Asimismo, se pueden utilizar arreglos distribuidos en *Parallel Computing Toolbox* para ejecutar aplicaciones de *big data* utilizando la memoria combinada del clúster (MathWorks, s. f.).

Puede usarse el *software* Octave como una alternativa libre al *software* comercial MATLAB, éste último cuenta con una interfaz más pulida, amplias herramientas especializadas y soporte comercial, lo que lo hace excelente para resolver necesidades avanzadas de la industria donde su rendimiento optimizado y los kits de herramientas integrales (como procesamiento de señales y sistemas de control) son cruciales. Otra razón por la que se está usando MATLAB en este trabajo es el aprovechamiento de las licencias que ofrece la UNAM, evitando así la necesidad de uso de *software* libre.

El objetivo es instalar y configurar un sistema paralelo y distribuido que ayude a investigadores, académicos y tesis a resolver matrices de grandes proporciones. Para ello, se usa el lenguaje de programación MATLAB y su herramienta *MATLAB Parallel Server*, con la cual pueden escalar a un entorno de decenas de servidores.

## 2. DESARROLLO TÉCNICO

Paralelizar un código, aún en nuestros días, puede llegar a ser una tarea complicada y tardada. Para ello, existen herramientas altamente especializadas que permiten hacer menos difícil la vida del programador, tal como MATLAB. En este lenguaje, se realizaron pruebas de multiplicación de matrices cuadradas comúnmente encontradas en problemas de ingeniería, como son las de tipo diagonal y triangular. Para ubicar temporalmente este trabajo, las pruebas se realizaron en agosto de 2025. MATLAB nos proporciona dos herramientas muy poderosas para escalar un código, éstas son:

*Parallel Computing Toolbox*. Permite resolver problemas de computación y datos intensivos utilizando procesadores multinúcleo, GPU y clústers de computadoras. La herramienta ofrece construcciones de alto nivel (bucles *for* paralelos, tipos de matrices especiales y algoritmos numéricos paralelizados) y también tiene funciones habilitadas para procesamiento en paralelo (MathWorks, s.f.).

*MATLAB Parallel Server*. Permite escalar programas MATLAB y simulaciones Simulink a clústers y nubes. Puede crear prototipos en el escritorio y ejecutarlos en clústers y nubes sin necesidad de recodificar.

MATLAB *Parallel Server* admite trabajos por lotes, cálculos paralelos interactivos y cálculos distribuidos con matrices grandes (MathWorks, s.f.).

## 2.1 METODOLOGÍA

A partir de las herramientas antes mencionadas, se procedió a instalar, configurar y poner en marcha un ambiente paralelo y distribuido de MATLAB. Además, se realizaron pruebas en las que se obtuvieron gráficas de *speedup* y eficiencia al escalar el código base. Cabe señalar que este ambiente no sólo se limita al manejo de matrices, sino que puede usarse para escalar simulaciones con la herramienta Simulink y solucionar problemas de *big data*.

## 2.2 INSTALACIÓN

La PC de comparación es una Dell XPS con las siguientes características:

Procesadores: 1 chip Intel Core i7-4770 3.4 GHz con 4 núcleos.

Memoria RAM: 16 GB.

Disco: 1 TB.

Tarjeta de red: 1 GbE.

Los servidores que se usaron son de la marca Dell PowerEdge R720 con las siguientes características:

Procesadores: 2 chips Intel Xeon E5-2640 2.5 GHz con 6 núcleos cada uno.

Memoria RAM: 64 GB.

Disco: 2 TB.

Tarjeta de red: 1 GbE.

Se requiere un tipo de licencia especial para instalar MATLAB *Parallel Server*; si el usuario no tiene esta licencia, tiene que ponerse en contacto con la Secretaría Técnica de su entidad o dependencia de adscripción.

Enseguida, se describen los pasos iniciales para la instalación:

- Crear un directorio para albergar el *software*.
- Montar la imagen de MATLAB en el sistema de archivos de Linux.
- En el directorio donde se montó el paquete, ejecutar el comando "*install*".

En el Anexo A, se describen con mayor detalle los pasos siguientes de la instalación.

## 2.3 CONFIGURACIÓN

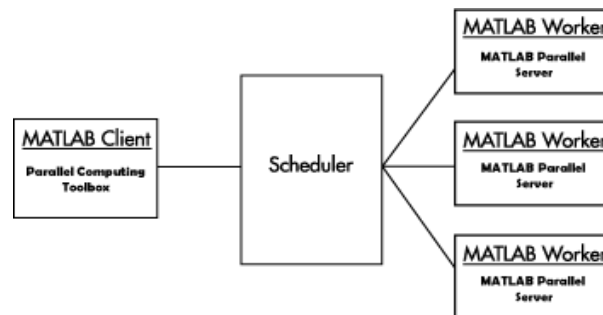
Un *job* es una operación extensa y se divide en segmentos llamados tareas. La sesión de MATLAB donde se definen el *job* y sus tareas se denomina sesión de cliente. El cliente utiliza el *software Parallel Computing Toolbox* para definir trabajos y tareas. MATLAB *Parallel Server* ejecuta el trabajo evaluando cada una de las tareas y devolviendo el resultado a su sesión de cliente.

*Parallel Computing Toolbox* permite ejecutar un clúster de trabajadores de MATLAB en la máquina local. *MATLAB Parallel Server* permite ejecutar tantos trabajadores de MATLAB en un clúster remoto de computadoras como lo permita la licencia. El Planificador de Trabajos de MATLAB es la parte del *software* del servidor que coordina la ejecución de trabajos y la evaluación de las tareas. Distribuye las tareas para su evaluación en las sesiones individuales de MATLAB del servidor, denominadas trabajadores.

En la Figura 1, se observa la configuración de MATLAB en paralelo. En este caso, se implementa en una PC y un servidor con procesadores multinúcleo.

**Figura 1**

*Configuración básica de cómputo paralelo*

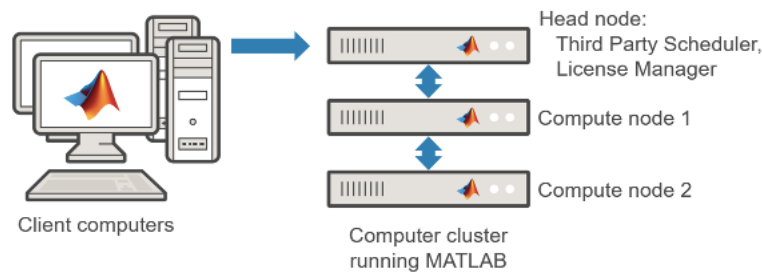


*Nota.* Recuperado de MathWorks. (s. f.).

En la Figura 2, se observa la configuración de MATLAB distribuido. En este caso, se implementa en varios servidores con procesadores multinúcleo a través de una red de comunicaciones.

**Figura 2**

*Configuración básica de cómputo distribuido*



*Nota.* Recuperado de MathWorks. (s. f.).

Para el acceso y configuración a la herramienta *MATLAB Parallel Server*, ejecutar *“admincenter”*:

```

    $> Directorio_Instalacion_MATLAB/toolbox/parallel/bin/admincenter
  
```

Los pasos de configuración detallados se muestran en el Anexo B.

Una vez configurada la herramienta, se procede a crear y probar un *script*, como se muestra a continuación.

## 2.4 PRUEBA DE FUNCIONAMIENTO

Una vez que se ha configurado el clúster, se hace una prueba de funcionamiento. Para ello, se realizó una serie de operaciones usando números aleatorios que se muestra en la Figura 3.

**Figura 3**

*Código usado para la prueba de funcionamiento*

```

1      %% Prueba de funcionamiento de MATLAB parallel server
2
3      % Declaracion de variables.
4      n = 400;
5      A = 500;
6      a = zeros(1,n);
7
8      % Inicia el temporizador
9      tic
10
11     % "parfor" permite paralelizar un ciclo "for"
12     % Las operaciones que se encuentran dentro del ciclo "for" realizan lo
13     % siguiente en paralelo:
14     % Obtiene el maximo de los valores absolutos de un vector columna
15     % que contiene los eigen valores de la matriz cuadrada A de numeros
16     % aleatorios.
17     parfor i = 1:n
18         a(i) = max(abs(eig(rand(A))));
19     end
20
21     % Termina el temporizador
22     toc
23
  
```

En la Figura 4, en la columna "Status", se puede observar que los trabajadores de MATLAB se encuentran ocupados (*busy*), con ello se asegura el buen funcionamiento de nuestro sistema distribuido.

**Figura 4**

*Trabajadores distribuidos ocupados realizando operaciones*

	Worker				MJS		
	Name	Hostname	Status	Up Since	Connection	Name	Hostname
Start...	n49_worker04	n49	● busy	2025-06-16 14:...	● connected	maestro	n49
Stop...	n49_worker03	n49	● busy	2025-06-16 14:...	● connected	maestro	n49
Resume	n49_worker06	n49	● busy	2025-06-16 14:...	● connected	maestro	n49
	n49_worker05	n49	● busy	2025-06-16 14:...	● connected	maestro	n49
	n52_worker03	n52	● busy	2025-06-16 14:...	● connected	maestro	n49
	n52_worker04	n52	● busy	2025-06-16 14:...	● connected	maestro	n49
	n52_worker05	n52	● busy	2025-06-16 14:...	● connected	maestro	n49
	n53_worker02	n53	● busy	2025-06-16 13:...	● connected	maestro	n49
	n53_worker01	n53	● busy	2025-06-16 12:...	● connected	maestro	n49
	n53_worker04	n53	● busy	2025-06-16 14:...	● connected	maestro	n49
	n53_worker03	n53	● busy	2025-06-16 14:...	● connected	maestro	n49
	n53_worker05	n53	● busy	2025-06-16 14:...	● connected	maestro	n49
	n52_worker01	n52	● busy	2025-06-11 17:...	● connected	maestro	n49
	n52_worker02	n52	● busy	2025-06-16 14:...	● connected	maestro	n49
	n49_worker02	n49	● busy	2025-06-16 14:...	● connected	maestro	n49

La descripción de las demás columnas de la Figura 4 se muestran a continuación:

*Worker.* Son los trabajadores o nodos de cálculo con sus respectivas características.

*Hostname.* Es el nombre de cada nodo.

*Up Since.* Muestra la hora en la cual el nodo está disponible dentro del clúster.

*MJS (MATLAB Job Scheduler).* Es el despachador de trabajos de MATLAB *Parallel Server*.

*Connection.* Muestra si el trabajador o *worker* está disponible y puede ser usado.

*Name.* Es el nombre del clúster creado con MATLAB.

## 2.5 MANEJO DE MATRICES

En este punto, se realizó una multiplicación de matrices diagonales y triangulares de 23.000 x 23,000 en una PC con 16 GB de memoria RAM. Antes de la multiplicación, se realizaron transformaciones previas de las matrices (transpuesta, inversa y elevadas al cuadrado), esto con el objetivo de mostrar el beneficio de usar varios servidores para resolver problemas complejos. Posteriormente, se escaló el problema para resolver un sistema de 49,000 x 49,000; esta magnitud de elementos ya no es posible manejarla en la PC, por lo que se seleccionaron todos los núcleos disponibles en tres servidores.

El operador de multiplicación de matrices `**` invoca la biblioteca BLAS (*Basic Linear Algebra Subprograms*), optimizada para realizar la multiplicación. Esta biblioteca es multihilo, lo que es una gran ventaja al usar todos los núcleos de un equipo. En la Figura 5, se muestra el código que se ejecutó en una PC ( $N = 23,000$ ) y en un servidor ( $N = 49,000$ ), donde “N” es cada una de las dimensiones de las matrices.

**Figura 5**

*Multiplicación de matrices en paralelo sólo para un equipo de cómputo*

```

1  %% Multiplicación local de matrices de N x N.
2
3  % Elementos por fila y columna.
4  N = 49000;
5
6  % Total de elementos de las 3 matrices.
7  elementos = 3*N*N;
8  |
9  % Creación de matrices diagonales.
10 A = diag(rand(1,N));
11 B = diag(rand(1,N));
12
13 |
14 % Creación de matrices triangulares.
15 A = triu(rand(N));
16 B = triu(rand(N));
17 %
18
19 |
20 % Multiplicación de matrices con transformaciones previas:
21 % Transpuesta.
22 % Inversa.
23 % Al cuadrado.
24 % Multiplicación.
25 tic
26 C = (inv(A'.^2)*(inv(B^).^2);
27 tec
28 % Imprime las variables y sus detalles.
29 whos
  
```

En la Figura 6, se muestra el código que se ejecutó en un clúster de tres servidores. Con ello, el código se pudo escalar y ejecutar de forma distribuida usando la función `distributed`.

**Figura 6**

*Operación de matrices distribuidas en tres servidores*

```

1  %% Arreglos distribuidos
2
3  % Representan aquellos arreglos que se dividen entre los trabajadores
4  % (workers) en un grupo paralelo.
5  % Una matriz distribuida se parece a una matriz MATLAB
6  % normal en la forma en que indexa y manipula sus elementos, pero ninguno
7  % de sus elementos existe en el cliente.
8
9  % Dimension de filas y columnas.
10 N = 49800;
11
12 % Total de elementos de las 3 matrices.
13 elementos = 3*N*N;
14
15 % Crea dos matrices de N x N.
16 % Los elementos de la matriz se distribuyen en
17 % entre los workers, pertenecientes al cluster creado.
18 disp("Distribución ...")
19
20 % Creación de matrices diagonales.
21 A1 = distributed(diag(rand(1,N)));
22 B1 = distributed(diag(rand(1,N)));
23
24 % {
25 % Creación de matrices triangulares.
26 A1 = distributed(triu(rand(N)));
27 B1 = distributed(triu(rand(N)));
28 % }
29
30 % El calculo se ejecuta sobre los workers.
31 % La matriz resultado tambien es distribuida.
32 % Multiplicación de matrices con transformaciones previas:
33 % Transpuesta.
34 % Inversa.
35 % Al cuadrado.
36 % Multiplicación.
37 disp("Operaciones ...")
38 tic
39 C1 = (inv(A1').^2)*(inv(B1').^2);
40 toc
41
42 whos % Imprime las variables y sus detalles
  
```

Se obtuvo la aceleración (*speedup*) y la eficiencia tanto para la prueba con la PC como para la prueba con el servidor. Estas gráficas son esenciales para verificar que un código sea escalable (Foster, 1995).

En la Figura 6, se muestra la creación de A1 y B1 (líneas 21 y 22, 26 y 27) como arreglos distribuidos, los cuales se han dividido entre los *workers*. Esto se muestra en la Figura 7. Las matrices se dividen por columnas y están listas para realizar la multiplicación.

**Figura 7**

*Ejemplo de una matriz dividida entre el número de workers del clúster MATLAB*

Fila	W1	W2	W3	...	W36
1	1, 2, 3, ..., C	C+1, ..., 2C	2C+1, ..., 3C	...	35C+1, ..., 36C
2	1, 2, 3, ..., C	C+1, ..., 2C	2C+1, ..., 3C	...	35C+1, ..., 36C
3	1, 2, 3, ..., C	C+1, ..., 2C	2C+1, ..., 3C	...	35C+1, ..., 36C
4	1, 2, 3, ..., C	C+1, ..., 2C	2C+1, ..., 3C	...	35C+1, ..., 36C
5	1, 2, 3, ..., C	C+1, ..., 2C	2C+1, ..., 3C	...	35C+1, ..., 36C
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
N	1, 2, 3, ..., C	C+1, ..., 2C	2C+1, ..., 3C	...	35C+1, ..., 36C

Donde:

$$C = N/nw$$

C = Número columnas que le corresponden a cada *worker*.

N: Número de filas o columnas de las matrices.

nw: Número de *workers*.

### 3. RESULTADOS

La primera prueba se realizó en una PC con 16 GB de memoria RAM, por lo que la dimensión máxima de las matrices fue de 23,000 x 23,000. En la Tabla 1, se observa el número de elementos de las matrices, la memoria usada y el tiempo de ejecución.

Tabla 1

*Datos que se obtuvieron al ejecutar una multiplicación de matrices en paralelo en una PC*

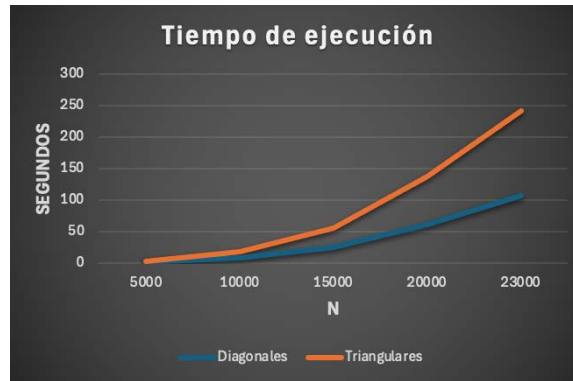
Tamaño de las matrices	Elementos totales de las matrices ( $\times 10^9$ )	Memoria usada por las matrices (GB)	Memoria total usada por el sistema (GB)	Tiempo de ejecución (s). Matrices diagonales	Tiempo de ejecución (s). Matrices triangulares
5,000 x 5,000	0.075	0.60	2.90	1.1	2.5
10,000 x 10,000	0.300	2.40	4.70	8.0	18.1

Tamaño de las matrices	Elementos totales de las matrices ( $\times 10^9$ )	Memoria usada por las matrices (GB)	Memoria total usada por el sistema (GB)	Tiempo de ejecución (s). Matrices diagonales	Tiempo de ejecución (s). Matrices triangulares
15,000 x 15,000	0.675	5.40	7.80	24.9	56.4
20,000 x 20,000	1.200	9.60	12.10	60.9	138.1
23,000 x 23,000	1.587	12.7	15.20	106.9	242.4

En la Figura 8, se observa el tiempo de ejecución de una multiplicación de matrices en una PC.

### Figura 8

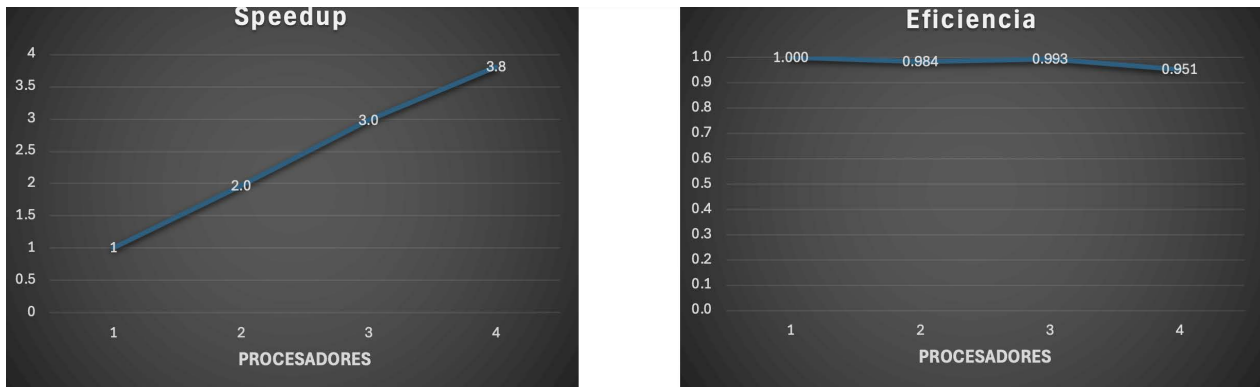
Tiempo de ejecución de multiplicación de matrices en una PC



En la Figura 9, se observa el *speedup* y eficiencia del procesamiento en paralelo usando de uno a cuatro núcleos del procesador Intel Core i7-4770 de la PC.

### Figura 9

Speedup y eficiencia en la PC



La segunda prueba se realizó en un solo servidor, la dimensión máxima de las matrices fue de 49,000 x 49,000, el número de elementos pasó de  $1.587 \times 10^9$  a  $7.2 \times 10^9$ , más de cuatro veces que en una PC. En la Tabla 2, se observa el número de elementos de las matrices, la memoria usada y el tiempo de ejecución.

**Tabla 2**

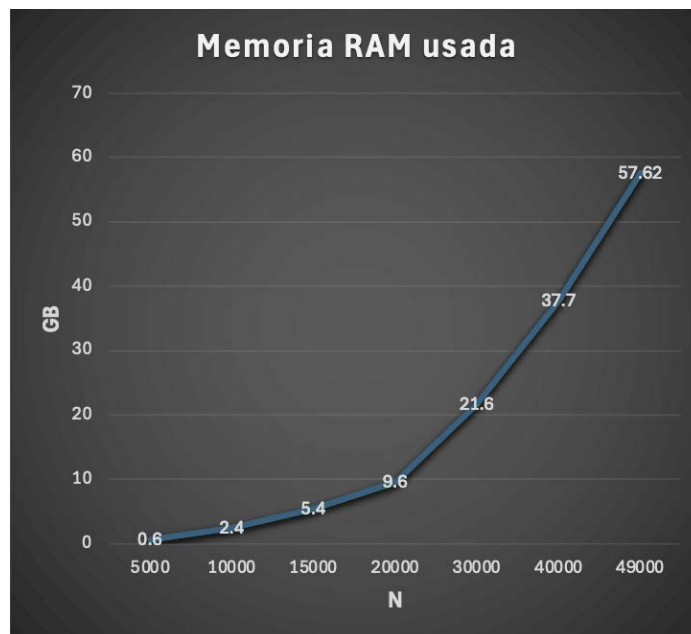
*Datos que se obtuvieron al realizar operaciones con matrices en un solo servidor*

Tamaño de las matrices	Elementos totales de las matrices (x 10 <sup>9</sup> )	Memoria usada por las matrices (GB)	Memoria total usada por el sistema (GB)	Tiempo de ejecución (s). Matrices diagonales	Tiempo de ejecución (s). Matrices triangulares
5,000 x 5,000	0.075	0.60	4.24	0.6	1.3
10,000 x 10,000	0.300	2.40	5.96	4.5	10.1
15,000 x 15,000	0.675	5.40	8.80	14.7	33.2
20,000 x 20,000	1.200	9.60	12.70	34.2	77.6
30,000 x 30,000	2.700	21.60	24.10	116.4	264.0
40,000 x 40,000	4.800	37.70	40.10	341.4	774.2
49,000 x 49,000	7.200	57.62	60.23	977.3	2215.9

Como se observa en la Figura 10, la memoria usada se incrementa exponencialmente con la dimensión de las matrices. En un equipo de cómputo personal ordinario (PC), sólo es posible multiplicar matrices de hasta 23,000 x 23,000; sin embargo, se observa en la Figura 11 cómo es posible realizar un procesamiento de matrices de mayor dimensión. En la Figura 12, se puede observar la comparativa entre los tiempos de ejecución entre la PC y el servidor.

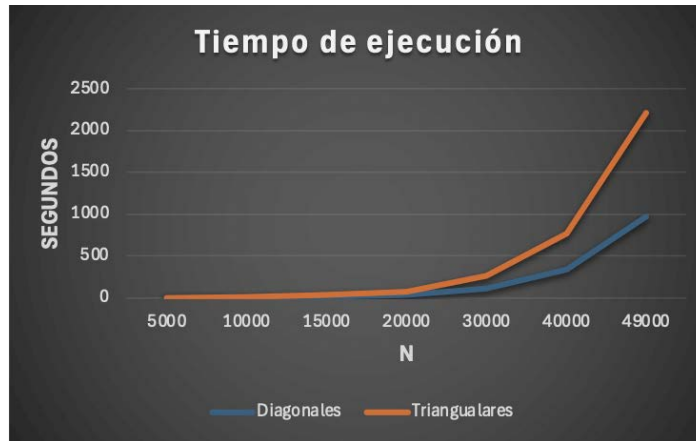
**Figura 10**

*Memoria usada en un servidor en la multiplicación de matrices con dimensión variable*



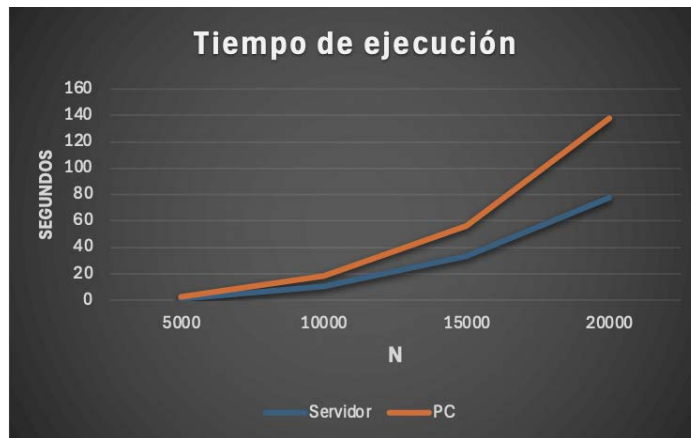
### Figura 11

Tiempo de ejecución de la multiplicación de matrices de una dimensión de 49,000 x 49,000



### Figura 12

Comparativa de los tiempos de ejecución entre la PC y el servidor



En la tercera prueba, se usaron tres servidores con 12 procesadores cada uno. En la Tabla 3, se puede apreciar el tiempo de ejecución de la multiplicación de matrices, con transformaciones previas, de una dimensión de 49,000 x 49,000 con diferente cantidad de procesadores distribuidos, así como su *speedup* y eficiencia.

**Tabla 3**

*Tiempo de ejecución, speedup y eficiencia*

Procesadores	Time de ejecución (s).	Time de ejecución (s).	Speedup	Eficiencia
	Matrices diagonales	Matrices triangulares		
1	11663	26445	1	1
4	2920	6621	3.99	0.999
8	1469	3330	7.93	0.992
12	977	2216	11.93	0.995
16	736	1668	15.84	0.990
20	592	1342	19.70	0.985
24	495	1123	23.56	0.982
28	430	974	27.12	0.969
32	369	838	31.60	0.988
36	332	752	35.13	0.976

En la Figura 13, se observa la gráfica de tiempo de ejecución usando diferente número de procesadores.

**Figura 13**

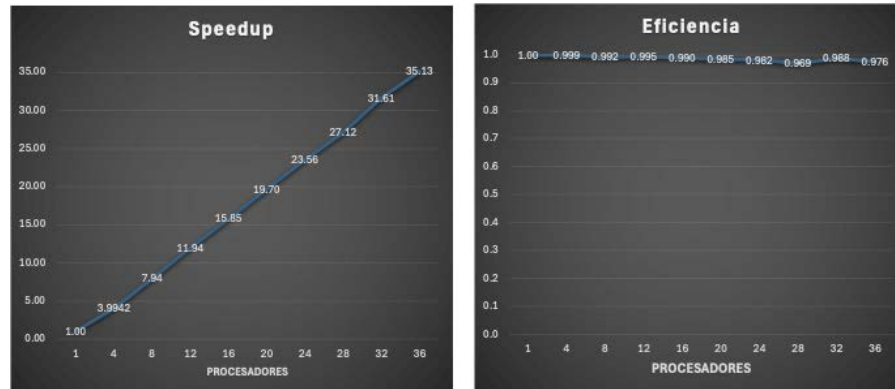
*Tiempo de ejecución para diferente número de procesadores*



En la Figura 14, se observa el *speedup* y la eficiencia del manejo de matrices usando 36 núcleos de 3 servidores del clúster de MATLAB.

**Figura 14**

*Speedup y eficiencia en el clúster MATLAB*



*Nota.* Cabe señalar que esta configuración de clúster ha sido utilizada por tres usuarios, entre ellos un investigador, un tesista y un técnico académico.

En este caso, las gráficas ya no son tan lineales como los resultados de la PC debido al proceso de comunicación de los *workers* a través de la red. También se observa que MATLAB *parallel server* es una herramienta útil para escalar un código de forma distribuida.

## 4. CONCLUSIONES

Los investigadores del IIMAS de la UNAM necesitaban recursos computacionales que una computadora de escritorio no puede ofrecerles. Por lo anterior, se puso en marcha un clúster de servidores con MATLAB distribuido entre ellos a través de una red de comunicaciones que permitió satisfacer las necesidades de cómputo científico. En este caso, han sido beneficiados tres usuarios de la línea de investigación en cómputo de alto rendimiento.

El *software* utilizado, junto con los recursos de *hardware*, permitieron ejecutar procesos para la resolución de problemas basados en operaciones de matrices. Lo anterior permitió utilizar automáticamente los múltiples núcleos presentes de los servidores sin necesidad de configuración adicional por parte de los usuarios. Esta abstracción del código permitió una mayor flexibilidad para los investigadores que no desean emplear tanto tiempo en verificar las comunicaciones entre núcleos en los tres nodos conectados a la red. Si se compara el tiempo y facilidad de configuración y escalamiento de MATLAB con otro *software*, como Octave, la ventaja es del primero. Se usó MATLAB para la distribución de procesos en el clúster aprovechando la facilidad de licencia que está disponible institucionalmente.

La instalación de MATLAB en una sola computadora es muy sencilla; sin embargo, es importante tomar en cuenta que, para la instalación y configuración en un ambiente distribuido en red, se necesita una licencia especial y tener un nodo específico para instalarla. En este trabajo, se usaron tres nodos de cálculo; sin embargo, una vez instalada y configurado el primer nodo de trabajo, se puede escalar el tamaño del clúster con relativa facilidad.

## REFERENCIAS

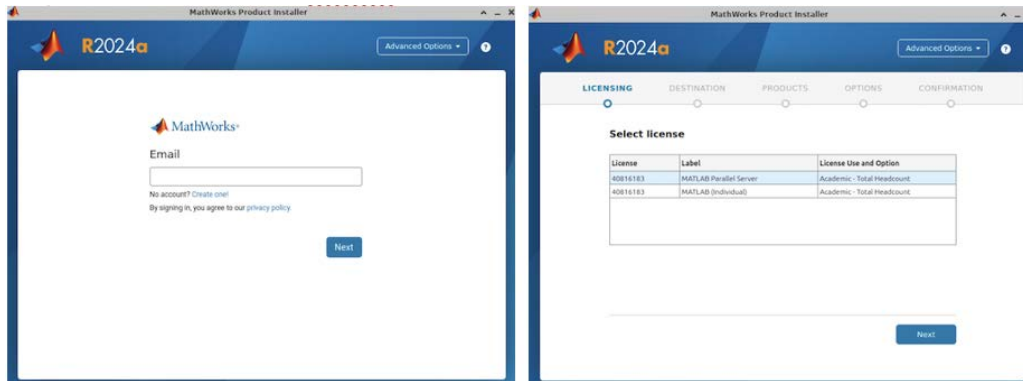
- Delfino, M. (2020). Distributed Computing. En C.W. Fabjan & H. Schopper (Eds.), *Particle Physics Reference Library* (pp. 613-644). Springer. [https://doi.org/10.1007/978-3-030-35318-6\\_14](https://doi.org/10.1007/978-3-030-35318-6_14)
- Foster, I. (1995). *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Addison-Wesley. [https://edoras.sdsu.edu/~mthomas/docs/foster/Foster\\_Designing\\_and\\_Building\\_Parallel\\_Programs.pdf](https://edoras.sdsu.edu/~mthomas/docs/foster/Foster_Designing_and_Building_Parallel_Programs.pdf)
- Gu, R., Tang, Y., Tian, C., Zhou, H., Li, G., Zheng, X., & Huang, Y. (2017). Improving Execution Concurrency of Large-Scale Matrix Multiplication on Distributed Data-Parallel Platforms, *IEEE Transactions on parallel and distributed systems*, 28(9), 2539 – 2552.
- HouZhen, W., Yan, G., & HuanGuo, Z. (2020). A Method of Ultra-Large-Scale Matrix Inversion Using Block Recursion. *Information*, 11(11), 523. <https://doi.org/10.3390/info11110523>
- Lim, R., Lee Y., Kim, R., & Choi, J. (2018). An implementation of matrix–matrix multiplication on the Intel KNL processor with AVX-512. *Cluster Computing*. 21, 1785–1795. <https://doi.org/10.1007/s10586-018-2810-y>
- MathWorks. (s. f.). *Distributed arrays*. Recuperado de <https://la.mathworks.com/help/parallel-computing/distributed-arrays.html>
- MathWorks. (s.f.). *Get started with MATLAB*. Recuperado de <https://la.mathworks.com/help/matlab/getting-started-with-matlab.html>
- MathWorks. (s. f.). *MATLAB Parallel Server* [Documento digital]. Recuperado de [https://la.mathworks.com/help/pdf\\_doc/matlab-parallel-server/matlab-parallel-server.pdf](https://la.mathworks.com/help/pdf_doc/matlab-parallel-server/matlab-parallel-server.pdf)
- MathWorks. (s. f.). *Parallel Computing Toolbox* [Documento digital]. Recuperado de [https://la.mathworks.com/help/pdf\\_doc/parallel-computing/parallel-computing.pdf](https://la.mathworks.com/help/pdf_doc/parallel-computing/parallel-computing.pdf)
- Westrick, S., Fluet, M., Rainey, M., & Acar U. (2024). Automatic Parallelism Management. *Proceedings of the ACM on Programming Languages*, 8, 1118 – 1149. <https://doi.org/10.1145/3632880>

## ANEXO A. PASOS SIGUIENTES DE LA INSTALACIÓN

La Figura 15 y Figura 16 muestran la secuencia de instalación de *MATLAB Parallel Server*.

### Figura 15

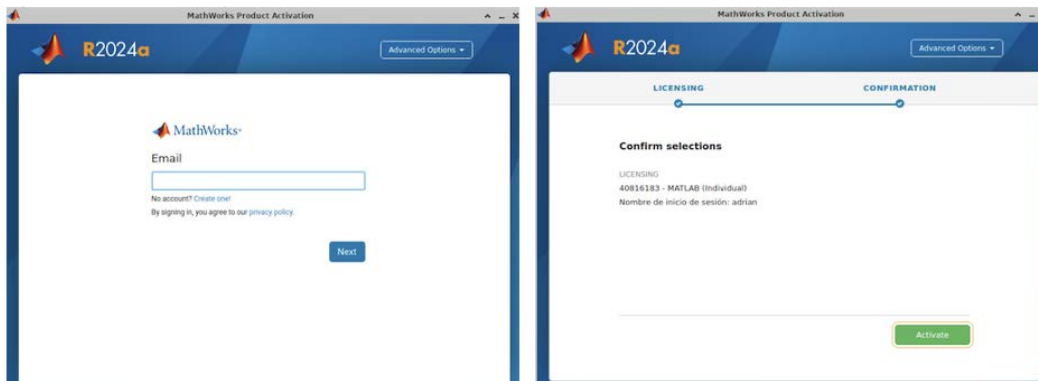
*Solicitud de correo institucional y selección de la licencia MATLAB Parallel Server*



*Nota.* En un paso posterior, se seleccionará el archivo de licencia.

### Figura 16

*Activación de MATLAB*



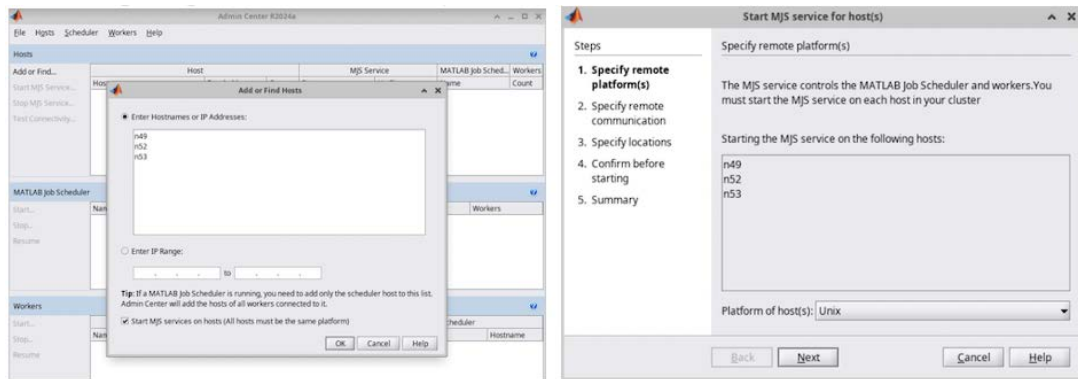
Al ejecutar MATLAB, es importante activarlo con la licencia de red. Para información detallada de la instalación, referirse al manual de *MATLAB parallel server*.

## ANEXO B. PASOS DE CONFIGURACIÓN

De la Figura 17 a la Figura 23, se muestra la secuencia de configuración de MATLAB *Parallel Server*.

**Figura 17**

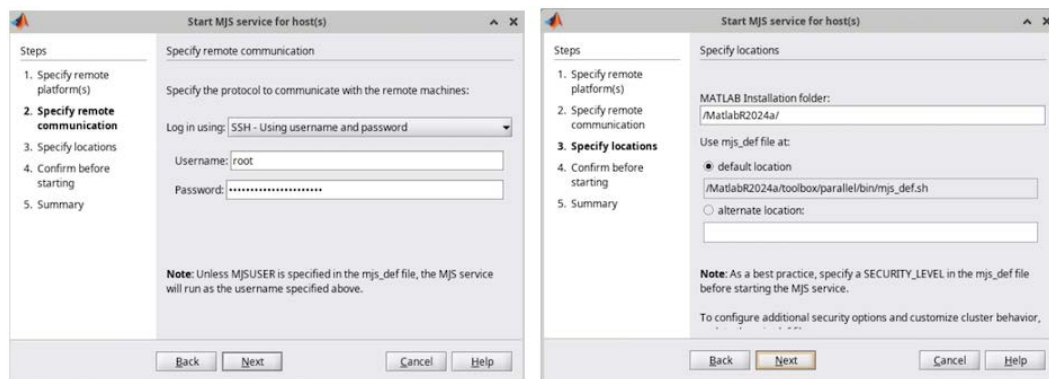
*Nodos y sistema operativo*



*Nota.* En el apartado "Add or Find...", se agregan los equipos que ya tengan instalado MATLAB *Parallel Server* (en este caso, n49, n52 y n53) y se verifica la plataforma.

**Figura 18**

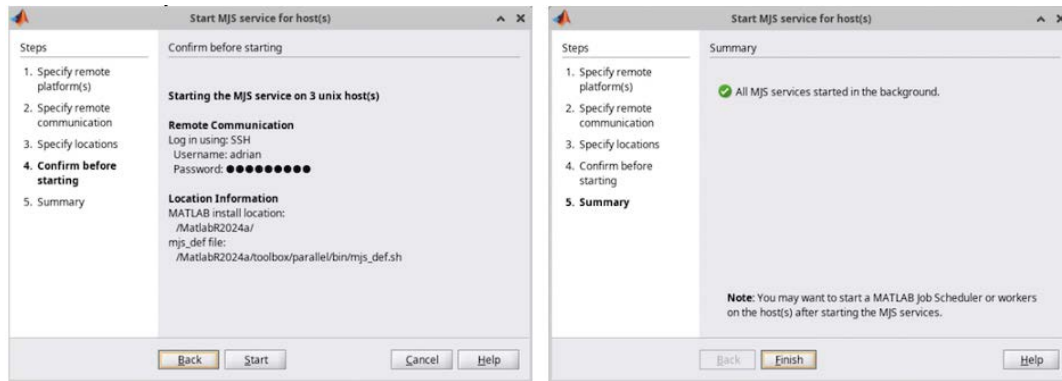
*Usuario y ruta de instalación*



*Nota.* En este paso, se configura el usuario que ejecuta el *software* y el directorio donde está instalado MATLAB.

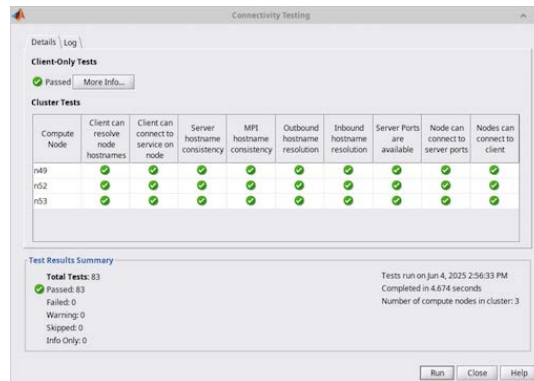
**Figura 19**

*Confirmación de los datos y verificación de inicio de los servicios del MJS (MATLAB Job Scheduler)*



**Figura 20**

*Prueba de conectividad (Test Connectivity)*

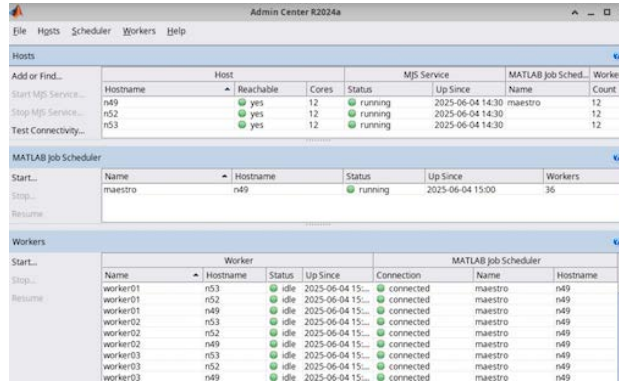


*Nota.* Al finalizar la sección de "Hosts", se prueba la conexión entre los nodos y se observa la verificación.

Se prosigue con la configuración de las secciones "MATLAB Job Scheduler" y "Workers". En la Figura 21, se observa que los servicios están ejecutándose.

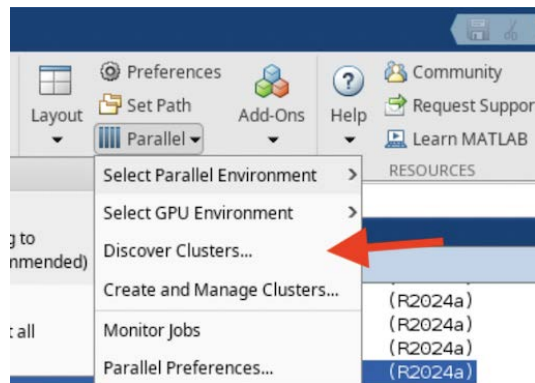
**Figura 21**

Vista de "admincenter" cuando la configuración funciona correctamente



**Figura 22**

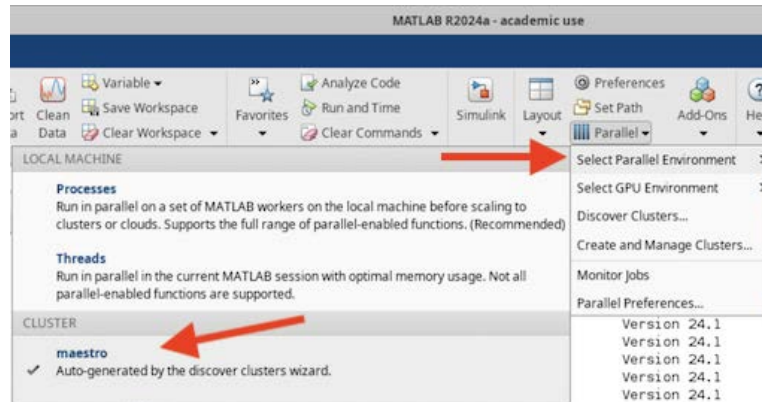
Agregar el perfil del nuevo clúster de MATLAB



Nota. En la pestaña "Parallel" y sección "Discover Clusters...", se agrega la configuración del nuevo clúster.

### Figura 23

*Selección del nuevo clúster "maestro"*



*Nota.* Finalmente se verifica la selección del nuevo clúster. En este caso, llamado "maestro".

## PERSONAS REVISORAS JULIO 2025 A JUNIO 2026

David Eduardo Abad Contreras  
Alejandro Aguilar Sierra  
Rocío Alejandra Aldeco Pérez  
Alfredo Alonso Peña  
Luis Agustín Álvarez Icaza Longoria  
Rubén Aquino Luna  
Lester Erick Arancibia Balderrama  
Ricardo Arroyo Mendoza  
Karla Priscila Ávalos Sandoval  
Luis Daniel Barajas González  
Carlos Félix Campa Arvizu  
Juan Manuel Castillejos Reyes  
Eloísa Chávez Vega  
Angélica Cruz Villamar  
Javier de Andrés Rivero  
Gustavo de la Cruz Martínez  
Max Ulises de Mendizábal Carrillo  
Carmen Humberta de Jesús Díaz Novelo  
Joel Espinosa Longi  
Florasela Fernández Victoria  
Karla Alejandra Fonseca Márquez  
Alma Rosa García Martínez  
Bardo Javier García Martínez  
José Agustín García Reynoso  
Alberto González Guízar  
Pablo Antonio González Peñaloza  
Fernando Israel González Trejo  
María Teresa Hernández Elenes  
Alejandra Herrera Mendoza  
Noel Alejandro Hortiales Corona  
Erick S. Huesca Morales  
Vianey Islas Bastida  
Miguel Ángel Jiménez Bernal  
Alfredo Landa Herrera  
Pilar López Martínez  
Eduardo López Molina  
Ana Laura Márquez Alonso  
Samuel Martínez Espinoza  
María Esther Judith Martínez González  
Esmeralda Martínez Montes  
Elizabeth Martínez Sánchez  
Gerardo Elías Navarrete Terán

Israel Josué Novelo Zel  
Juan Carlos Olivares Rojas  
Israel Ortega Cuevas  
Jorge Palacios Elizalde  
Ana Cecilia Pérez Arteaga  
Marcelo Pérez Medel  
Pedro Temachti Pérez Santillán  
Zaa Ribe Jazmín Ramírez Grajeda  
Hugo Dionisio Ramón  
Hugo Alonso Reyes Herrera  
Sandra Roan Cano  
Juan Manuel Rodríguez Martínez  
Rita Carolina Rodríguez Martínez  
Karen Lucero Roldán Serrato  
Rosalía Rosas Castañeda  
José Antonio Salazar Carmona  
Erika Sánchez Chablé  
Edgar Vargas Zermeño  
Alejandro Velázquez Mena  
María Teresa Ventura Miranda  
María Cristina Verde Rodarte  
Claudia Marina Vicario Solórzano  
Ricardo Federico Villarreal Martínez  
Miguel Zambrano Meza  
Víctor Hugo Zamora Guerrero



Cuadernos Técnicos Universitarios  
de la **DGTIC**

ISSN-e: 3061-8096



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

DIRECCIÓN GENERAL DE CÓMPUTO Y DE  
TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN



**DGTIC UNAM**

DIRECCIÓN GENERAL DE CÓMPUTO Y  
DE TECNOLOGÍAS DE INFORMACIÓN  
Y COMUNICACIÓN