Vol. 4, Núm. 3. julio-septiembre **2026**
DOI:10.22201/dgtic.30618096e.2026.4.3

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

DOI: 10.22201/dgtic.30618096e.2026.4.3

Vol. 4, Núm. 3. julio-septiembre 2026

Cuadernos Técnicos Universitarios de la DGTIC

DOI: 10.22201/dgtic.30618096e.2026.4.3

Vol. 4, Núm. 3. julio-septiembre 2026

CUADERNOS TÉCNICOS UNIVERSITARIOS DE LA DGTIC

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* (3), páginas (N1-N2). https://doi.org/10.22201/dgtic.30618096e.2026.4.3

Cuadernos Técnicos Universitarios de la DGTIC

DOI: 10.22201/dgtic.30618096e.2026.4.3

Vol. 4, Núm. 3. julio-septiembre 2026

CUADERNOS TÉCNICOS UNIVERSITARIOS DE LA DGTIC, Año 4, No. 3, julio-septiembre 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, 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 agosto de 2026.

Cuadernos Técnicos Universitarios de la DGTIC

DOI: 10.22201/dgtic.30618096e.2026.4.3

Vol. 4, Núm. 3. julio-septiembre 2026

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.

Cuadernos Técnicos Universitarios de la DGTIC

DOI: 10.22201/dgtic.30618096e.2026.4.3

Vol. 4, Núm. 3. julio-septiembre 2026

Cuadernos Técnicos Universitarios de la DGTIC

DOI: 10.22201/dgtic.30618096e.2026.4.3 Vol. 4, Núm. 3. julio-septiembre 2026

CONTENIDO

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 8 - 23

Creación de un sistema web de contenido nativo

Development of a native content web system

Información del reporte:

Para citar este reporte técnico:

Chamú Arias, J. O. y Ramírez Muñoz, D. A. (2026). Creación de un sistema web de contenido nativo. *Cuadernos Técnicos Universitarios de la DGTIC, 4*(3), páginas (8 - 23). https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182

José Othoniel Chamú Arias
Dirección General de Cómputo y de Tecnologías de Información y Comunicación
Universidad Nacional Autónoma de México
chamu_as@unam.mx
ORCID: 0009-0001-4949-3024

Diego Alfonso Ramírez Muñoz
Dirección General de Cómputo y de Tecnologías de Información y Comunicación
Universidad Nacional Autónoma de México
diegoramirez.code@gmail.com
ORCID: 0009-0009-1894-9074

Resumen

El desarrollo de un sistema a la medida que trabaja en dos servidores necesita de un alto grado de personalización para actualizar distintos servicios. Derivado de ello, surge el reto de actualizar ambos servidores: el principal, el cual sólo es accesible en la red interna y que, para funcionar, requiere únicamente que los archivos multimedia y tablas específicas sean actualizados; y el secundario, que aloja el sistema web y presenta la interfaz al público usuario. Para implementarlo, se consideraron dos opciones: el gestor de contenidos WordPress o código PHP puro. Para efectos de este proyecto, se optó por utilizar la segunda opción. En consecuencia de esta decisión, se absorbieron muchas actividades (como la seguridad, la cual estaba a cargo del Gestor de Contenido) y quedaron bajo responsabilidad del desarrollador. Para cumplir con esta tarea, se decidió usar Apache Firewall, que ofrece protección contra ataques comunes.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 9 - 23

Entre los retos a resolver, se encontraba la necesidad de sincronizar la comunicación entre ambos servidores; para ello, se utilizó la librería libssh2, que cuenta con funciones de ejecución de comandos remotos, transferencia de archivos seguros, autenticación y reenvío de puertos para comunicaciones seguras. Con esta herramienta, que permite ejecutar comandos remotos, se sorteó la labor de importar y exportar la base de datos completa de un servidor a otro y se logró operar consultas directas al servidor secundario para actualizar, insertar o borrar registros específicos; también se consiguió transferir archivos multimedia sin necesidad de saturar los servidores con procesos repetidos e innecesarios. Como resultado, esta técnica de comunicación permitió una conexión ágil entre los servidores, además de que coadyuvó en la protección de la información sensible del servidor principal, debido a que no comprometió en ningún momento el acceso, siendo una alternativa confiable para proyectos que requieren una sincronización selectiva y segura.

Palabras clave: Base de datos MySQL, firewall web Apache ModSecurity, libssh2, protocolo SSH, sincronización selectiva.

Abstract

The development of a custom system running on two servers requires a high degree of customization to update its various services. Consequently, a challenge arises when updating both servers: the main one, which can only be accessed via a local network and requires only updates to multimedia files and specific tables for its operation; and the secondary, which hosts the system that functions as the website and presents the user interface to the public. To implement this, two options were considered: the WordPress content management system or native PHP code. This project adopted the latter approach. In consequence of this decision, numerous activities (such as security, which fell under the responsibility of the Content Manager) were absorbed and placed under the developer's purview. To ensure compliance with security standards, it was decided to use Apache Firewall, which offers protection against common attacks.

Among the challenges to be resolved was the need to synchronize communication between both servers. To achieve this, the libssh2 library was utilized, which features functions for remote command execution, secure file transfer, authentication, and port forwarding for safe communications. This tool, which allows the execution of remote commands, bypassed the need to import and export the entire database from one server to another. It successfully enabled direct queries to the secondary server to update, insert, or delete specific records, as well as transfer multimedia files without overloading the servers with repeated and unnecessary processes. As a result, this communication technique allowed a fast connection between the servers, and also helped to protect the sensitive information on the main server, since it did not compromise access at any time. This proved to be a safe alternative for projects that require selective and secure synchronization.

Keywords: *MySQL database, Apache ModSecurity web firewall, libssh2, SSH protocol, selective synchronization.*

1. INTRODUCCIÓN

En el ámbito académico universitario, la disponibilidad y el flujo constante de datos son pilares fundamentales para la operación de servicios educativos y administrativos. Los sistemas de almacenamiento y transferencia de información se utilizan diariamente para gestionar desde contenidos didácticos y repositorios de investigación hasta trámites escolares y registros de personal.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 10 - 23

El almacenamiento y transferencia de información en los procesos académicos son elementos fundamentales que están presentes en todas las áreas de la universidad. Este proceso debe llevarse a cabo priorizando la seguridad de los datos durante la transferencia, almacenamiento y presentación en la página, dependiendo del tipo de información que se maneje (como un número de contacto de atención al público en general o los nombres, correos, fechas de nacimiento, CURP, direcciones de personal y alumnos, por mencionar algunos ejemplos).

El tratamiento de dicha información, siguiendo las buenas prácticas de seguridad, se separa en componentes públicos y privados, con la finalidad de establecer políticas diferenciadas de acceso y protección, además de facilitar las labores de mantenimiento. Para este caso, el sistema se dividió en dos grandes apartados: una parte que almacena la información sensible y otra donde se almacena información de consulta general. El sistema, dependiendo del usuario, se encarga de dividir el acceso a la información, lo cual garantiza la separación y protección de los datos sensibles y aseguran que los datos privados permanezcan aislados de cualquier perfil de acceso público.

En el área de trabajo se tenían que considerar las siguientes restricciones: el uso de un portal de gestión de contenidos no permitía crear roles para acceder a la información de manera jerárquica y su modificación era compleja. Por lo anterior, se evaluó usar herramientas que permitieran personalización total y que disminuyeran los archivos a vigilar y mantener.

El problema base que dio origen a este proyecto (implementado en 2023 – 2024) fue la necesidad de alimentar un sitio web alojado en un servidor secundario (público) sin comprometer la seguridad del sistema principal ni recurrir a procesos manuales lentos. Tras analizar las opciones, se identificó que el uso de gestores de contenido (CMS) presentaba restricciones importantes, ya que no permitían una integración directa con el sistema de gestión interna ni permitía la creación de roles jerárquicos personalizados para el acceso a la información. Debido a ello, el trabajo se dividió en dos partes: la gestión de seguridad del sistema y el desarrollo de la aplicación sin recurrir al uso de gestores de contenido.

El objetivo del proyecto fue desarrollar un módulo para el sistema de gestión interna que permitiera la transferencia segura y selectiva de archivos multimedia y tablas de bases de datos mediante el uso de la librería libssh2. A diferencia de una réplica total, este desarrollo buscó implementar una sincronización específica a través del protocolo SSH, permitiendo actualizar únicamente los componentes necesarios en el servidor secundario que es de acceso público. Con esto, se garantiza una comunicación ágil entre servidores, manteniendo el control total sobre la seguridad y optimizando los recursos de procesamiento.

2. DESARROLLO TÉCNICO

Se desarrolló un módulo que gestiona la carga de una imagen desde un formulario HTML, la guarda en el servidor primario, actualiza la base de datos con el nombre del archivo en la columna correspondiente, realiza una copia del mismo en el servidor secundario y ejecuta una consulta remota que actualiza la base de datos en el servidor secundario, que tiene únicamente datos de lectura y la información necesaria para la consulta. Para gestionar de manera eficiente este método, fue primordial configurar correctamente el entorno de desarrollo. Con este fin, se instalaron las extensiones mysqli (MySQL Improved), que permiten a los *scripts* gestionar la base de datos, y libssh2, para la ejecución remota de comandos y transferencia de archivos.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 11 - 23

Este proceso cubre el ciclo completo de carga de archivos necesarios para visualizar información en el sitio web, desde el guardado del archivo y la actualización de la base de datos en el servidor principal hasta la actualización remota del servidor secundario mediante funciones como Secure Copy (SCP) y consultas remotas.

2.1 METODOLOGÍA

La metodología que mejor se adaptó para el desarrollo de este proceso fue el modelo en cascada. Este enfoque, de acuerdo con González *et al*. (2021), "consiste en la ejecución de etapas unas tras otras, de arriba hacia abajo; esto requiere que antes de pasar a la siguiente fase se necesita que la primera fase haya concluido para proseguir con la siguiente etapa" (p. 188). Por ello, este acercamiento metodológico fue el apropiado para sistemas basados en transferencias y resultó fundamental para delimitar los momentos exactos en los que el método cambiaba de servidor. Debido a que ésta no permite avanzar a la siguiente fase si no se ha verificado la anterior, se garantizó una implementación segura y estructurada, a través de las siguientes etapas:

Diseño del proyecto

Se evaluaron las herramientas de terceros (*plugins*) que realizaran esta tarea de forma segura, y, debido a que no se encontró una opción que se integrara a las necesidades específicas de la institución, se decidió programar un módulo desde cero.

Diseño de flujo de datos

Se determinó de manera estricta qué archivos y datos debían permanecer aislados en el servidor primario y cuáles debían transferirse al servidor secundario para estructurar el recorrido de la información.

Selección de tecnologías

Con el flujo establecido, se definió el entorno de desarrollo y se seleccionó el lenguaje de programación, el motor de base de datos y la configuración del sistema. Además, se integraron las extensiones mysqli para la gestión de datos y libssh2 para la ejecución remota de comandos y transferencia de archivos (SCP).

Programación

Se desarrolló la estructura arquitectónica necesaria para integrar este nuevo módulo dentro del sistema de gestión interna ya existente. Se programó, en primera instancia, la lógica del servidor principal (carga de imágenes, almacenamiento local y actualización de la base de datos interna). Posteriormente, se estructuraron las versiones de la base de datos en el servidor secundario, creando únicamente las columnas estrictamente necesarias para el entorno público. Finalmente, se desarrolló el *front-end*, el cual, de acuerdo con Pressman y Maxim (2021), abarca la arquitectura de la interfaz de usuario y la lógica de presentación con la que interactúa de manera directa el público; en este proyecto, dicho entorno quedó encargado de consumir y mostrar esta información ya filtrada de forma segura.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 12 - 23

Instalación

Se configuró el entorno de producción en ambos servidores y se certificó que las librerías necesarias estuvieran habilitadas, así como que los permisos de red permitieran la comunicación por el puerto correspondiente para la ejecución de comandos de forma remota.

Pruebas

Para la validación técnica de cada etapa, se implementó un riguroso manejo de excepciones y verificación de resultados en el código con pruebas de funcionalidad. Según Rangel Cano (2021), las pruebas funcionales son la "actividad dedicada a la aplicación de las pruebas que analizarán el funcionamiento del *software* bajo un enfoque de usuario final... identificando interrupciones derivados de defectos funcionales" (p. 5). Además, estas pruebas se alinean con el modelo en cascada, garantizando que cada fase se complete antes de avanzar a la siguiente.

Mantenimiento

Se establecieron lineamientos para la revisión de los registros (*logs*) de transferencia y errores de las funciones SSH, con el fin de garantizar que la sincronización selectiva se mantenga operativa a lo largo del tiempo ante posibles actualizaciones del servidor.

Seguridad

Se aplicó una estricta segmentación de la información. El uso del protocolo de seguridad SSH garantizó que tanto la copia de archivos como las consultas remotas viajaran de forma encriptada, cumpliendo el objetivo principal de mantener los datos privados completamente aislados del servidor secundario de acceso público. Cabe mencionar que se aplicó un proceso de *hardening* al servicio SSH del servidor mediante el cambio del puerto estándar, tal como lo recomienda Toapanta Rocha (2023) en su metodología de endurecimiento de sistemas operativos para mitigar accesos no autorizados.

2.1.1 ANÁLISIS

Debido a la estructura de un sistema integral de gestión interna, se identificó la necesidad de utilizar la información almacenada en él para actualizar automáticamente diferentes apartados de un sitio web alojado en un servidor independiente. Este proceso eliminó la intervención manual, la cual implicaba una inversión de tiempo innecesaria y el entrenamiento en conocimientos técnicos especializados para los usuarios administradores de esos módulos.

Después de analizar la problemática, se desechó el uso de gestores de contenidos y se concluyó que no resultaban adecuados, ya que no permitía la integración directa con el sistema de gestión interna ni la automatización del intercambio de datos entre servidores, limitándose a un entorno de administración propio y restringido.

Por ello, se decidió implementar la librería libssh2, gracias a que provee, a través de sus funciones, acceso a recursos sobre una máquina remota utilizando una vía de transporte criptográfica segura. Además, esta alternativa ofrece ventajas para optimizar el rendimiento —particularmente relevante dado que ambos servidores cuentan con recursos limitados y se priorizó minimizar el consumo de ancho de banda y procesamiento—, además de otros beneficios orientados a adaptar la seguridad a las necesidades

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 13 - 23

exactas del desarrollo, limitando las operaciones únicamente a los archivos y tablas involucradas en el proceso. Todo lo mencionado se refleja en la Tabla 1.

Tabla 1

Comparación entre libssh2 y migrar el sitio a un gestor de contenidos

Criterio	libssh2	Gestor de Contenidos (CMS)
Control sobre el flujo de trabajo	Control total sobre cada paso del proceso (carga, validación, almacenamiento, etc.).	Control limitado por la configuración predeterminada del CMS y las opciones disponibles en *plugins*.
Flexibilidad	Altamente flexible, permite cualquier personalización según las necesidades del proyecto.	Flexibilidad limitada a la funcionalidad básica y los *plugins* disponibles; personalizaciones avanzadas son difíciles.
Rendimiento	Rendimiento optimizado, ya que sólo se cargan las funcionalidades necesarias.	Generalmente más lento, debido a la sobrecarga causada por módulos y *plugins* innecesarios.
Seguridad	Se pueden implementar medidas de seguridad específicas, como la validación personalizada y control estricto.	Seguridad basada en *plugins* o módulos, que pueden ser vulnerables si no se actualizan adecuadamente.
Transferencia de archivos	Integración directa con SSH o SCP para transferir archivos a servidores remotos de forma segura.	Requiere complementos o configuraciones adicionales para manejar la transferencia entre servidores.
Gestión de bases de datos	Consultas SQL completamente personalizadas, optimizadas para el proyecto específico.	Uso de consultas estándar y limitadas a las funcionalidades del CMS; requiere *plugins* para personalización.
Escalabilidad	Escalable a medida que crece el proyecto, con posibilidad de optimización a nivel de código y base de datos.	Escalabilidad limitada, dependiente de la estructura interna del CMS y de los *plugins* instalados.
Facilidad de implementación	Más complicado de implementar, requiere conocimientos técnicos avanzados de PHP, bases de datos y servidores.	Fácil de implementar para proyectos pequeños o rápidos, con una curva de aprendizaje más baja.
Tiempo de desarrollo	Mayor tiempo de desarrollo debido a la personalización y pruebas necesarias.	Menor tiempo de desarrollo inicial debido a las funciones prediseñadas, pero la personalización puede ser compleja.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 14 - 23

Actualizaciones y mantenimiento	Mantenimiento gestionado por el equipo de desarrollo; control total sobre actualizaciones y cambios.	Actualizaciones dependientes del CMS y los *plugins*, con posibles incompatibilidades entre versiones.

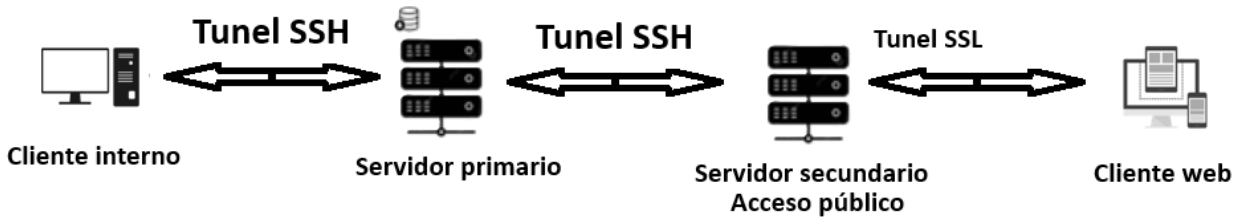
Como se observa en la tabla anterior, se tiene una comparativa de ventajas y desventajas de usar un método u otro, lo cual nos sirve de apoyo visual para seleccionar un estilo de desarrollo.

2.1.2 DISEÑO

En esta fase, se identificó la ruta crítica del proceso, se definió la forma en que los archivos serían transferidos dentro del sistema y se determinó la manera en que serían enviados al servidor secundario. Para ello, el proceso se dividió en dos grandes bloques: el servidor primario, al cual sólo es posible ingresar desde redes locales y se encarga de guardar la información y prohibir su acceso al público; y el servidor secundario de acceso público, donde se coloca la página web. El esquema se muestra en la Figura 1.

Figura 1

Ruta de la información



En la figura se aprecia de manera gráfica cómo el archivo viaja desde la aplicación que se ejecuta en el servidor primario y se transfiere al servidor secundario mediante un túnel cifrado con SSH, para proteger su contenido durante el tránsito entre servidores y clientes.

2.1.3 IMPLEMENTACIÓN

En esta fase de la metodología, se desarrolló el modelo de carga y procesamiento de los archivos. Este proceso se dividió en diferentes etapas: primero, se implementó la captura del archivo cargado por el usuario, el cual se almacena en la ruta previamente definida en el servidor local, utilizando un identificador único y la extensión de cada uno de los archivos. Para ello, se empleó un módulo que permite subirlo mediante la función *move_upload_file()*, lo cual asegura que el archivo se mueva de la ubicación temporal a la ruta definida; después, si la operación resulta exitosa, se realiza la actualización de la base de datos local mediante una consulta SQL (*Structured Query Language*), que registra el nombre del archivo, la extensión y la ruta de su elemento correspondiente.

Las ocasiones en que se validó la actualización de la base de datos y se concretó poder subir el archivo en el servidor local, se desarrolló el procedimiento de transferencia del archivo al servidor secundario mediante la función *ssh_scp_send()*, que permite la transmisión de información mediante el protocolo de copia segura. De acuerdo con Effendi *et al.* (2021), este protocolo utiliza una conexión SSH en la que los datos se envían cifrados. La validación de la carga se realizó mediante la verificación del código de retorno de la función y la comprobación de la existencia y tamaño del archivo en el destino, asegurando

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 15 - 23

que la transferencia se completara exitosamente antes de continuar con el siguiente paso. De igual forma, se implementó la lógica para actualizar la base de datos en el servidor remoto. Cabe aclarar que esta base de datos fue diseñada únicamente con las tablas necesarias para la alimentación del sitio web; la actualización se efectuó mediante la ejecución de una consulta SQL similar a la realizada en el servidor local.

Para asegurar la robustez del módulo, fue necesario integrar estructuras de manejo de errores que permitieran identificar si existió algún problema durante la transferencia de archivos o durante la actualización remota de la base de datos. Posteriormente, se cerraron todas las conexiones abiertas que se comunicaran con el servidor remoto, lo que liberó recursos del servidor y aseguró que no quedaran conexiones activas que pusieran en peligro la seguridad de los servidores. Finalmente, la plataforma mostró mensajes de éxito, indicando que los datos fueron registrados correctamente y, después de un retraso de dos segundos, redirige al usuario a la página donde puede visualizar los archivos subidos.

2.1.3.1 MANEJO DEL ARCHIVO

Aquí se obtiene el nombre del archivo a subir desde el formulario y se extrae el nombre y la extensión que, en este caso, puede ser .jpg o .png. Cabe mencionar que la variable *$archivo* pasó por un proceso de limpieza.

Figura 2

Obtención de extensión del archivo

```
$extArchivo = pathinfo($archivo);
$extArchivo1 = $extArchivo['extension'];
```

La Figura 2 muestra un segmento de código donde se lleva a cabo la obtención de la extensión del archivo para poder clasificarlo y validarlo dentro del servidor. Cabe mencionar que éste es un paso indispensable para evitar que se procesen extensiones no válidas e impedir la inyección de código malicioso, una práctica fundamental en la mitigación de vulnerabilidades de seguridad web tal como recomienda Cajías (2020).

2.1.3.2 DEFINICIÓN DE LA RUTA LOCAL

Se definió la ruta local donde se almacenará el archivo. La ruta incluye el directorio, el identificador único del archivo y su extensión.

Figura 3

Ruta de servidor local

```
$rutaLocal = '/ruta/servidor/local/';
$fichero_subido = $rutaLocal.$idArchivo.'.'.$extArchivo1;
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 16 - 23

El código de la Figura 3 muestra cómo se lleva a cabo del proceso de creación de la ruta donde se va a depositar el archivo. Esto permite personalizar un directorio separado del sistema, de modo que, en caso de ser necesario, pueda redimensionarse su espacio sin afectar al sistema.

2.1.3.3 SUBIR EL ARCHIVO Y ACTUALIZAR LA BASE DE DATOS EN EL SERVIDOR LOCAL

En esta etapa, se realiza el traslado del archivo desde su ubicación temporal, asignada desde la carga, hasta la ubicación definida por *$fichero_subido*, mediante la instrucción mostrada en la Figura 4.

Figura 4

Subir el archivo

```
if (move_uploaded_file($_FILES['archivo']['tmp_name'], $fichero_subido))
```

Como se muestra en el código, se valida la acción de mover el archivo de los directorios temporales al directorio de destino final. Esto es importante para evitar inconsistencias de información, ya que guardar un estado de éxito en la base de datos sin asegurar que el archivo se cargue correctamente puede generar información imprecisa en la base.

Si no existe ningún error, el proceso continúa y se actualiza la base de datos. Se ejecuta una consulta de inserción que actualiza el campo “archivo” de la tabla correspondiente con el nombre y la extensión del fichero subido. La ejecución se muestra en la Figura 5.

Figura 5

Actualizar la base de datos

```
$Update = '
    update tabla
    set archivo = \''.$idArchivo.'.'.$extArchivo1.'\'
    where idArchivo = '.$idArchivo.'';
$mysqli->query($Update) or die ('Error en el query $Update1: '.$mysqli->error);
```

Como se ve en la Figura 5, ya que se validó que el archivo se cargó correctamente, se registra en la base para así tener información consistente en el sistema.

2.1.3.4 TRANSFERENCIA REMOTA

Las ocasiones en que subir el archivo y actualizar la base de datos local se completó de manera exitosa, se incluyó el archivo que contiene los parámetros de conexión para realizar la transferencia remota del archivo y la actualización de la base de datos, para lo cual es necesario primero definir las rutas de los servidores. Posteriormente, se utiliza la función *ssh2_scp_send()* para transferir el archivo del servidor local al remoto, seguido de los permisos para asegurar que esto suceda sin ningún problema.

Una vez que el archivo fue transferido de manera exitosa, se ejecutó una consulta SQL mediante una conexión SSH, que replica la misma actualización realizada en la base local.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 17 - 23

En la Figura 6 se muestra el código con rutas relativas, las cuales son útiles para poder migrar el código de manera más transparente, debido a que no está fijo en una estructura de directorios.

Figura 6

Conexión remota

```
include('../ruta/conexion.php');
$origen = '/ruta/servidor/local/'.$idArchivo.'.'.$extArchivo1;
$destino = "/ruta/servidor/remoto/".$idArchivo.'.'.$extArchivo1;
```

En esta imagen se muestra cómo se construyen las rutas origen y destino para poder transferir la información de manera dinámica dentro del sistema, con lo cual se evita que el proceso se lleve a cabo de forma manual.

La función que se muestra en la Figura 7 es útil para ejecutar el comando de sistema operativo de copiado seguro mediante un canal cifrado.

Figura 7

Transferencia del archivo al servidor remoto

```
ssh2_scp_send($connection, $origen, $destino, 0777);
```

Una vez mencionada la función de la instrucción, el segmento de código de la Figura 7 ejecuta la instrucción para conectar ambos servidores y transferir la información entre los mismos. Cabe mencionar que el último valor corresponde a los permisos otorgados en el servidor remoto e indica que todos los usuarios tienen permiso de lectura, escritura y ejecución dentro del servidor aislado (remoto).

La función de la Figura 8, en este caso *ssh2_exec*, tiene la utilidad de ejecutar comandos en servidores remotos mediante un canal cifrado.

Figura 8

Actualización de la base de datos remota

```
$consulta = '
    update tablaremota
    set archivo = \''.$idArchivo.'.'.$extArchivo1.'\'
    where idArchivo = '.$idArchivo.';';
$stream = ssh2_exec($connection, 'mysql -u user -p "baseremota" -sse "'.$consulta.'";'
```

Como se ve en el segmento de código mostrado arriba, se indica el proceso de cómo se ejecuta un comando en la base de datos de manera remota. De esta manera, se restringe el acceso directo a la base de datos desde exterior, de modo que las consultas únicamente pueden realizarse mediante la función *ssh2_exec* dentro del código PHP. Con esto, se evitan los ataques de fuerza bruta, los cuales se definen

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 18 - 23

como métodos de intrusión basados en el mecanismo de prueba y error. Tras cerrar el acceso en varias capas, se implementa el modelo de seguridad en profundidad. De acuerdo con Rodríguez Gahona (2020), "en esa estrategia de defensa, en lugar de colocar una única línea muy fuerte, se colocan varias líneas consecutivas" (p. 37), lo cual tiene como finalidad lograr que el atacante desista y cambie de objetivo.

2.1.3.5 MANEJO DE ERRORES Y NOTIFICACIONES

Para garantizar que durante el proceso no se produjeran errores inesperados, se implementaron estructuras de manejo de errores basados en la captura y gestión de errores en tiempo real. En primera instancia, se utilizó *or die()* en la ejecución de las consultas SQL y durante el proceso de transferencia de archivos, con el fin de mostrar un mensaje en caso de que ocurriera un problema en la ejecución. Lo anterior permitió obtener el error específico de la causa de la interrupción del proceso.

Para las transferencias remotas, se utilizaron las funciones *ssh_exec()* y *ssh_fetch_stream(),* con el objetivo de capturar los errores de una conexión SSH. Para esto, se configuraron los flujos como bloqueadores utilizando la función *stream_set_blocking(),* lo cual forzó al módulo a esperar una respuesta antes de continuar. Finalmente, los flujos fueron cerrados para liberar los recursos y evitar que se produjeran otro tipo de errores inesperados y, dependiendo del éxito o fallo de la tarea, se notificara al usuario a través de mensajes de error mostrados en pantalla.

Se implementaron las funciones de manejo de flujos (o *streams*) para controlar de manera rigurosa la conexión entre servidores.

Figura 9

Manejo de errores

```
$errorStream = ssh2_fetch_stream($stream, SSH2_STREAM_STDERR);
stream_set_blocking($errorStream, true);
stream_set_blocking($stream, true);
fclose($errorStream);
fclose($stream);
```

El código de la Figura 9 sirve para capturar los errores del servidor remoto de forma controlada y evita que el *script* de PHP se quede congelado.

2.1.4 PRUEBAS

La fase de pruebas consistió en ejecutar diversos escenarios para validar el correcto funcionamiento del módulo. Esto también condujo a definir el tipo de archivo y el tamaño máximo para optimizar la transferencia de información entre servidores, estableciendo sólo la transferencia de imágenes en formato .jpg de máximo 2.5MB y .pdf de máximo 10MB.

A lo largo de cada prueba, se observó el comportamiento del sistema para mover los archivos desde su ubicación temporal hasta el servidor remoto; se probaron también las consultas en ambas bases de datos, verificando que la información sobre los datos cargados fueran los correctos, y que se hubieran registrado en las columnas correspondientes. También se realizaron pruebas para evaluar el proceso

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 19 - 23

de transferencia de archivos vía SSH, que posibilitaron realizar ajustes a las conexiones y personalizar los permisos para reducir el riesgo en la escritura, o dejar abierta alguna conexión que pudiera resultar en el mal funcionamiento del módulo. Cada error fue monitoreado para gestionar adecuadamente las conexiones fallidas o permisos incorrectos.

2.1.5 MANTENIMIENTO

A partir de los resultados obtenidos de las pruebas y después de obtener una versión estable de este módulo, se liberó al usuario la versión que se encargaría del registro de la información, pero se mantuvo una comunicación constante para adaptar el módulo a las necesidades de éste. El intercambio de información permitió realizar ajustes personalizados a la transferencia de archivos, tales como agregar nuevos formatos (.epub y .png) solicitados por el usuario. Estas mejoras permitieron al modelo ser escalable y habilitaron la retroalimentación por parte de los usuarios para conducir los ajustes conforme se avanzara en el desarrollo. Sin embargo, se pudo observar que, en archivos mayores a 25MB, el rendimiento del sistema se veía afectado considerablemente. Por ende, fue necesario cargar los archivos manualmente en el servidor remoto; este hallazgo condujo a empezar a desarrollar nuevas técnicas en el proceso de transferencia que ofrecieran un mejor manejo de archivos de gran volumen.

2.1.6 SEGURIDAD

Una vez finalizados los módulos de carga de archivos y actualizaciones de base de datos, se identificó la necesidad de agregar una capa de seguridad para mitigar ataques, por lo que se llevaron a cabo las siguientes acciones:

Se autorizó el acceso a los servicios solamente a un grupo cerrado de direcciones IP mediante un *firewall* perimetral. Este mecanismo, el cual es descrito por Caiza Chipantaci (2025) como un sistema de seguridad esencial diseñado para segmentar y separar las redes, controla estrictamente el tráfico hacia la red interna y entre los servicios, evitando así la exposición de la infraestructura al público general.

Para mitigar el impacto de los *sniffers*, los cuales se definen como herramientas especializadas en interceptar, escuchar y capturar el tráfico de datos dentro de una red informática, tal como lo señalan Lescano y Quiros (2021), se implementaron certificados digitales en los servidores web. Asimismo, como se mencionó en el desarrollo, se utilizó la librería libssh2 para cifrar la comunicación en la actualización de las bases de datos y en la transferencia de archivos, garantizando la confidencialidad y la secrecía en el tránsito de la información.

Para mitigar algunos ataques que pudieran venir de equipos internos, tales como: SQL Injection, el cual consiste en manipular las instrucciones web para ejecutar consultas maliciosas en la base de datos; *path transversal*, el cual permite navegar entre directorios del sistema operativo manipulando los datos web que se mandan al servidor; *command injection*, que permite ejecutar comandos de sistema operativo en el servidor web manipulando datos que no son debidamente filtrados; *cross site scripting*, que permite ejecutar instrucciones de JavaScript en equipos remotos para secuestrar sesiones a través de robar identificadores de sesión; se delegó la protección a un *firewall* web de Apache, el cual está implementado en el módulo Modsecurity y cuenta con una serie de reglas para mitigar los ataques mencionados y muchos más, ya que basa muchas de sus reglas en el Open Web Application Security Project (OWASP), que es una colección de recomendaciones de seguridad para proteger aplicaciones.

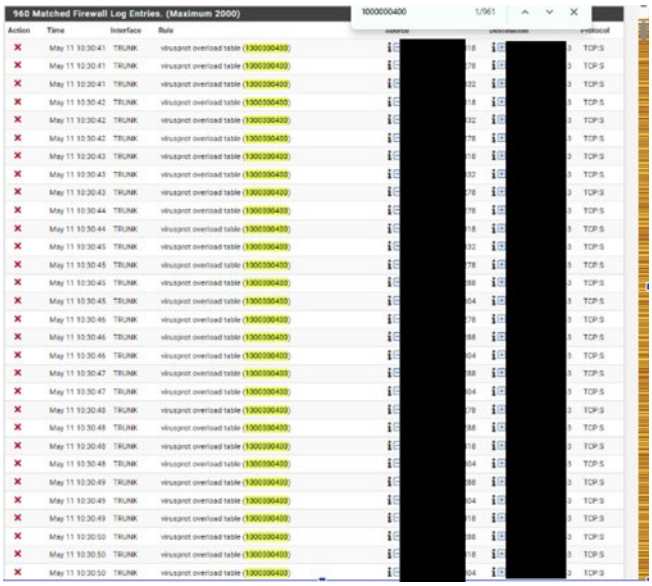
https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 20 - 23

Se utilizó el *firewall* web de Apache porque es gratuito y menos agresivo y complicado de usar comparado con Suricata (Sistema de Detección de Intrusos), de acuerdo con nuestra experiencia. En este contexto, se considera que este último tiene un gran rango de instrucciones de bloqueo de ataques, sin embargo, excede a las necesidades de la aplicación web y eleva la complejidad del despliegue de la infraestructura. Además, en ocasiones previas, se ha observado que puede generar falsos positivos, los cuales expulsan del sistema a usuarios legítimos, así como se debe instalar en una capa externa al servidor web para disminuir la carga a éste.

En la Figura 10 se muestra una salida correspondiente a la implementación del sistema en diversos proyectos. La siguiente imagen muestra el detector de intrusos en entornos de mayor complejidad. En este caso, se aprecia un intento de ataque de virus hacia un servidor, el cual es bloqueado efectivamente por el detector de intrusos al analizar el comportamiento de las conexiones.

Figura 10

Suricata



El uso de esta herramienta no debe ser descartado si se cuenta con los recursos de infraestructura necesarios para implementarla, ya que Suricata optimiza su rendimiento y trabaja mejor en entornos de redes de alta velocidad, tal como lo indican Cabrera Laguapillo y Larco Andrade (2018) en su análisis de sistemas de detección de intrusos periféricos. Para efectos de este proyecto, el *firewall* web de Apache cubrió las necesidades de protección y cumplió con el objetivo de optimizar la carga del servidor web.

3. RESULTADOS

En el desarrollo del módulo, se logró implementar de manera exitosa un mecanismo de sincronización entre el servidor local y el servidor remoto, mediante funciones diseñadas específicamente para la transferencia, registro y validación de archivos. Durante las pruebas realizadas, tanto en entornos

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 21 - 23

controlados como con usuarios finales, se comprobó que los archivos cargados son correctamente almacenados en las rutas definidas y registrados de forma consistente en ambas bases de datos. Este proceso incluye validaciones que garantizan la integridad de la información, evitando inconsistencias entre los servidores. Asimismo, se incorporaron mecanismos de control que detectan errores en la comunicación, tales como fallos en la conexión a bases de datos o interrupciones en la conexión SSH, permitiendo detener el proceso y notificar al usuario con mensajes claros, lo que fortalece la confiabilidad del sistema.

Por otra parte, se consiguió que la seguridad del sistema recayera directamente en la lógica implementada por el desarrollador, por la acción deliberada de no utilizar *frameworks* o sistemas de gestión de contenido. Esta decisión implicó el desarrollo manual de controles de validación, limpieza de datos, manejo de sesiones y verificación de accesos, lo que facilitó el entendimiento profundo de los mecanismos de seguridad involucrados. Entre las ventajas observadas, destacan las siguientes: una mayor flexibilidad en la implementación de políticas específicas de seguridad, una reducción en la dependencia de terceros y un mejor control sobre el comportamiento del sistema. Además, se minimizó la sobrecarga de componentes innecesarios, lo cual optimizó el desempeño general y facilitó la personalización del módulo conforme a los requerimientos del entorno.

Finalmente, a través de las pruebas de desempeño, se logró identificar que ciertas problemáticas relacionadas con el tiempo de carga y la estabilidad en la transferencia de archivos no corresponden directamente a deficiencias del *software*, sino a limitaciones del *hardware* y de la infraestructura de red. En particular, en el proceso de trabajar con archivos de gran tamaño, se observó un incremento considerable en los tiempos de transferencia y, en algunos casos, interrupciones en la conexión. Este comportamiento fue clave para distinguir entre cuellos de botella asociados al sistema y aquellos derivados de recursos físicos, lo cual resulta clave para futuras optimizaciones.

4. CONCLUSIONES

Los resultados obtenidos demuestran que la solución propuesta cumple de manera satisfactoria con los objetivos planteados. Se logró una sincronización efectiva entre servidores, un manejo adecuado de errores y una implementación robusta de medidas de seguridad, además de identificar correctamente las limitaciones asociadas al *hardware*. En conjunto, estos resultados resolvieron el problema inicial gracias a que proporcionaron un sistema funcional, confiable y adaptable, capaz de operar dentro de los parámetros esperados y responder adecuadamente ante escenarios de fallo.

Es importante destacar que el desarrollo prescindió del uso de gestores de contenido y *frameworks* no por una desventaja inherente a estas herramientas, sino por una decisión basada en los requerimientos específicos del usuario. Se reconoce que los gestores de contenido representan soluciones sólidas, ampliamente probadas y con múltiples ventajas en términos de rapidez de desarrollo, mantenimiento y escalabilidad. Sin embargo, para este proyecto en particular, no se requería dicha capa de abstracción, por lo cual se optó por una implementación más ligera y completamente controlada. Esta decisión favoreció la flexibilidad del sistema y la adaptación precisa a las necesidades planteadas; ello no implica una postura en contra del uso de este tipo de herramientas en otros contextos.

El principal aporte de este trabajo es la implementación de mecanismos de optimización en la transferencia de archivos de gran tamaño, como la fragmentación de archivos, el uso de colas de procesamiento o

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182 **Vol. 4, Núm. 3.** julio-septiembre 2026, págs. 22 - 23

la compresión previa a la transferencia. Recomendamos para trabajos futuros la exploración de la integración de herramientas híbridas que mantengan el control del desarrollador, pero aprovechando ciertas ventajas de *frameworks* ligeros en aspectos específicos como seguridad o manejo de sesiones. Este tipo de proyectos puede evolucionar hacia sistemas más complejos de gestión distribuida de archivos o plataformas escalables que integren monitoreo en tiempo real, tolerancia a fallos y balanceo de carga, ampliando así su alcance y aplicabilidad en entornos de mayor exigencia.

AGRADECIMIENTOS

Agradecemos al Equipo editorial de CTUD por acompañar a nuestro texto en todas sus etapas y por sus valiosas recomendaciones que hicieron posible el presente trabajo. En especial valoramos el tiempo dedicado por Eric Villar Juárez y Juan Manuel Campos Coto para asesorarnos sobre la estructuración y redacción de este reporte técnico.

Declaración de contribución de autoría

Diego Alfonso Ramírez Muñoz: Especialista en desarrollo de *software* y análisis de soluciones tecnológicas. Rol dentro del proyecto: desarrollador. Participó en la identificación de las necesidades del cliente/usuario y del sistema, con el propósito de definir una solución tecnológica que respondiera de manera adecuada a los requerimientos funcionales y operativos del proyecto. Su intervención se centró en analizar el flujo de trabajo esperado, las condiciones de uso de la aplicación y los elementos técnicos necesarios para construir una solución clara, funcional y mantenible. A partir de esta revisión, consideró aspectos relacionados con la facilidad de uso, la disminución de la complejidad para el usuario final y la correcta separación entre los componentes de infraestructura y aplicación. Asimismo, colaboró en la definición de medidas orientadas a proteger el tránsito de la información a través de la red, procurando que la solución mantuviera criterios de seguridad, organización y eficiencia durante su operación.

José Othoniel Chamú Arias: Especialista en tecnologías de la información y seguridad informática. Rol dentro del proyecto: *pentester* y administrador de servidores. Participó en la identificación de herramientas, configuraciones y medidas técnicas necesarias para proteger la aplicación una vez desplegada en el servidor web. Su contribución se orientó al fortalecimiento de la seguridad de la solución mediante la revisión de aspectos del código y la detección de oportunidades de mejora para reducir la exposición ante ataques conocidos. Asimismo, colaboró en el despliegue de la aplicación dentro de los servidores correspondientes, realizando actividades de instalación y configuración de los ambientes requeridos para su operación. Estas actividades incluyeron la configuración de la base de datos, servidor web, sistema operativo, creación de cuentas, apertura y restricción de puertos, así como la configuración de módulos necesarios para el funcionamiento de la aplicación. También intervino en la incorporación de herramientas de protección, tales como el *firewall* web, con el fin de mejorar la seguridad y estabilidad del servicio.

REFERENCIAS

Cabrera Laguapillo, M. S. y Larco Andrade, Y. C. (2018). *Análisis de las técnicas de ocultación de código malicioso para evasión de sistemas de protección de usuario final y NIDS* [trabajo de titulación de ingeniería, Escuela Politécnica Nacional]. Repositorio Institucional de la EPN. https://bibdigital.epn.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.182 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 23 - 23

edu.ec/handle/15000/19514

Caiza Chipantaci, E. M. (2025). *Implementación de ciberseguridad a través de pfSense con SIEM* [trabajo de integración curricular, Escuela Politécnica Nacional]. Repositorio Institucional de la EPN. https://bibdigital.epn.edu.ec/handle/15000/26460

Cajías, E. F. (2020). *Evaluación de la eficiencia de ataques de tipo clickjacking y touchjacking en entornos controlados* [trabajo de titulación de maestría, Escuela Politécnica Nacional]. Repositorio Institucional de la EPN. https://bibdigital.epn.edu.ec/handle/15000/21498

Effendi, M. R., Al-Falah, R. S., y Ismail, N. (2021, 19-20 de agosto). *IoT-Based Battery Monitoring System in Solar Power Plants with Secure Copy Protocol (SCP)* [ponencia]. 2021 7th International Conference on Wireless and Telematics (ICWT), Bandung, Indonesia. https://doi.org/10.1109/ICWT52862.2021.9678210

Lescano Andrade, B. L., y Quiros Chulca, C. A. (2021). *Implementación de dos sniffers inalámbricos en un sistema Raspberry Pi para el componente práctico de comunicaciones inalámbricas de la ESFOT* [trabajo de titulación de tecnólogo, Escuela Politécnica Nacional]. Repositorio Institucional EPN. https://bibdigital.epn.edu.ec/handle/15000/21812

González, Y., MC lenan, J., y Abrego, A. (2022). Los proyectos y las metodologías. Modelos de procesos: Ciclos de vida de desarrollo de software. *Revista Semilla Científica*, *3*(3), 185–194. https://revistas.umecit.edu.pa/index.php/sc/article/view/1090

Pressman, R. S., & Maxim, B. R. (2021). *Ingeniería de software: Un enfoque práctico* (9a ed.). McGraw-Hill Interamericana.

Rangel Cano, L. L. (2021). *Metodología para aplicar pruebas funcionales como parte de una auditoría de sistemas de información*. Dirección General de Cómputo y de Tecnologías de Información y Comunicación, UNAM. https://www.red-tic.unam.mx/recursos/2021/metodologia-pruebas-funcionales.pdf

Rodríguez Gahona, G. (2020). *Análisis comparativo de los modelos defensa en profundidad y MSPI, para la implementación de la seguridad informática en el sector privado del país* [trabajo monográfico de especialización, Universidad Nacional Abierta y a Distancia]. https://repository.unad.edu.co/handle/10596/38717

Toapanta Rocha, J. C. (2023). *Implementación de hardening en un sistema operativo de servidor Linux de base Red Hat* [trabajo de integración curricular, Escuela Politécnica Nacional]. Repositorio Institucional de la EPN. https://bibdigital.epn.edu.ec/handle/15000/25203

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 24 - 46

Implementación de un sistema automatizado de monitoreo de red multi-sede en una institución universitaria

Implementation of an automated multi-site network monitoring system in a university institution

Información del reporte:

Para citar este reporte técnico:

López Ibarra, J. G. y Velázquez Portilla, D. (2026). Implementación de un sistema automatizado de monitoreo de red multi-sede en una institución universitaria. *Cuadernos Técnicos Universitarios de la DGTIC, 4*(3), páginas (24 - 46). https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181

Jorge Gerardo López Ibarra
Instituto de Biología
Universidad Nacional Autónoma de México
jlopez@ib.unam.mx
ORCID: 0009-0004-4703-4993

David Velázquez Portilla
Instituto de Biología
Universidad Nacional Autónoma de México
davepo@ib.unam.mx
ORCID: 0009-0009-8467-8225

Resumen

En una entidad universitaria con varias sedes y entornos de operación, el monitoreo de los servicios digitales dependía de una revisión técnica dispersa: consultas separadas en distintas plataformas, registros manuales y validaciones periódicas. Eso hacía difícil tener una visión clara del estado general de la red. Para resolver esta situación, se diseñó e implementó un sistema automatizado que integró fuentes de información heterogéneas, normalizó datos operativos y generó indicadores semanales para cinco entornos institucionales.

La metodología se organizó en las etapas de diseño, configuración, instalación y pruebas, con énfasis en la reutilización de herramientas existentes, la protección de información sensible y la elaboración de una visualización ejecutiva del entorno. Esta implementación permitió disminuir la dependencia de procesos manuales, consolidar información técnica dispersa y apoyar el seguimiento operativo de la red institucional.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 **Vol. 4, Núm. 3.** julio-septiembre 2026, págs. 25 - 46

Palabras clave: Automatización de operaciones, gestión de servicios tecnológicos, indicadores de disponibilidad, redes universitarias, visualización ejecutiva.

Abstract

In a university entity with several sites and operational environments, monitoring digital services relied on a fragmented technical review process involving separate queries across different platforms, manual records, and periodic validations. This made it difficult to obtain a clear view of the overall network status. To address this situation, an automated system was designed and implemented to integrate heterogeneous information sources, normalize operational data, and generate weekly indicators for five institutional environments. The methodology was organized into the stages of design, configuration, installation, and testing, with emphasis on reusing existing tools, protecting sensitive information, and producing an executive visualization of the environment. This implementation reduced dependence on manual processes, consolidated dispersed technical information, and supported operational monitoring of the institutional network.

Keywords: *Operations automation, technology service management, availability indicators, university networks, executive visualization.*

1. INTRODUCCIÓN

La correcta operación de los servicios digitales en las instituciones de educación superior depende, cada vez con mayor intensidad, de redes de datos estables, puntos de acceso inalámbrico disponibles y enlaces de comunicación capaces de atender actividades académicas, administrativas y de investigación. En este contexto, el monitoreo técnico dejó de ser una actividad aislada de revisión de equipos y se convirtió en un proceso necesario para garantizar el óptimo desempeño de los servicios institucionales. Como ha señalado la literatura reciente, contar con información integrada ayuda a tomar decisiones y responder mejor ante fallas en entornos complejos (Farahmand *et al*., 2022; Ta-Armart y Savithi, 2026).

El Instituto de Biología de la Universidad Nacional Autónoma de México (IBUNAM) opera sus servicios de red en espacios con condiciones distintas de conectividad, ubicación geográfica y administración tecnológica. A la infraestructura ubicada en Ciudad Universitaria, se sumaron estaciones foráneas y espacios especializados que utilizaron plataformas de gestión diferentes. Esta diversidad hizo que la supervisión dependiera de consultas en herramientas por separado de revisión de reportes de tráfico, inspección de consolas de administración inalámbrica y captura de información complementaria para sedes con condiciones particulares.

Aunque ya se cuentan con herramientas de administración y monitoreo, la información no se encontraba integrada en una sola vista. Esta situación dificultaba comparar el comportamiento entre sedes, construir históricos semanales y presentar indicadores comprensibles para seguimiento técnico y administrativo. En la práctica, el personal responsable debía dedicar tiempo a recopilar datos de distintas fuentes, validar su consistencia y preparar reportes de forma manual. Este esquema funcionaba, pero no era suficiente para responder con agilidad y rapidez a una infraestructura distribuida que cambiaba de manera constante.

Diversos trabajos recientes propusieron arquitecturas modulares, tableros de visualización y mecanismos automáticos de recolección de datos para mejorar la administración de servicios tecnológicos. Estas

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 26 - 46

propuestas coincidieron en separar la adquisición de datos, el procesamiento y la visualización para facilitar la escalabilidad y reducir errores operativos en redes institucionales y entornos de campus (Boluda-Prieto *et al*., 2026; Espinel-Villalobos *et al*., 2022; Lee *et al*., 2014; Rafiq *et al*., 2025; Witczak *et al*., 2024). En el caso universitario analizado, esa aproximación permitió plantear una solución ajustada a las herramientas disponibles, sin tener que sustituir por completo las plataformas existentes.

En el contexto del IBUNAM, estos desafíos no se expresaron como un problema abstracto de arquitectura tecnológica, sino como una necesidad operativa concreta: consultar entornos multi-marca, reunir datos de sedes con diferentes condiciones de conectividad y conservar evidencia periódica del estado de la red. Por ello, el proyecto se orientó a una analítica semanal estructurada, sin sustituir el diagnóstico operativo inmediato que permanece en las consolas especializadas de cada plataforma.

El objetivo del proyecto fue implementar un sistema automatizado de monitoreo de red multi-sede que integrara información proveniente de plataformas heterogéneas, generara indicadores semanales y presentara una visualización ejecutiva para fortalecer la supervisión operativa de servicios tecnológicos en una institución universitaria.

El proyecto se desarrolló durante el periodo 2025-2026 y se concentró en la construcción de un reporte semanal de estado de red para cinco entornos institucionales, ubicados en Ciudad de México, Jalisco, Colima y Veracruz. El alcance se limitó a indicadores operativos generales de conectividad, disponibilidad, infraestructura inalámbrica y comportamiento de enlaces, sin exponer configuraciones internas, direcciones sensibles, credenciales ni detalles que comprometieran la seguridad de la institución.

2. DESARROLLO TÉCNICO

Para dar mayor orden a la exposición, el desarrollo técnico se presenta en dos niveles. Primero, se explican los términos generales, la base conceptual y el modelo de procesamiento que puede aplicarse a un sistema automatizado de monitoreo multi-sede. Después, se muestra cómo esos principios se aplicaron al caso específico del IBUNAM mediante las etapas de diseño, configuración, instalación y pruebas.

Esta organización permite distinguir entre la teoría que sustenta la solución y la forma en que dicha teoría se convirtió en una herramienta operativa. De esta manera, el reporte no presenta únicamente un producto final, sino el razonamiento técnico que permitió pasar de plataformas separadas a una visualización semanal integrada.

2.1 TÉRMINOS GENERALES Y BASE CONCEPTUAL

Monitoreo de red

Seguimiento periódico de indicadores que permiten conocer el estado general de la conectividad, los dispositivos de red y los servicios asociados. No se limita a observar si un equipo está encendido o apagado; también considera tendencias, disponibilidad, clientes conectados, comportamiento de enlaces y observaciones que ayudan a interpretar la operación de una infraestructura tecnológica.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 27 - 46

Entorno multi-sede

Condición operativa en la que la información proviene de espacios con conectividad, proveedores, equipos, mecanismos de administración y niveles de automatización diferentes. Esta característica dificulta la comparación cuando la revisión se realiza de manera manual o aislada.

Fuente heterogénea de información

Sistema que entrega datos sobre una misma operación institucional, pero con formatos, alcances y niveles de detalle distintos. En una red institucional, puede tratarse de consolas en la nube, plataformas inalámbricas, gráficas históricas, archivos locales o registros validados por personal técnico.

Automatización

Reducción de tareas repetitivas mediante procesos programados que consultan fuentes, recuperan datos y preparan archivos intermedios. En este proyecto, automatizar no significó eliminar el criterio técnico, sino ordenar la información y liberar tiempo de revisión manteniendo validación humana cuando un indicador lo requería.

Normalización de datos

Proceso mediante el cual la información obtenida de distintas fuentes se convierte en una estructura común. Esta etapa permite comparar sedes, preparar indicadores homogéneos y evitar que el tablero dependa directamente del formato particular de cada herramienta.

Visualización ejecutiva

Presentación resumida de indicadores en un formato comprensible para seguimiento operativo y toma de decisiones. Su propósito no es reemplazar las consolas especializadas ni exponer configuraciones internas, sino ofrecer una primera lectura clara del estado general de la red.

2.2 MODELO CONCEPTUAL DE INTEGRACIÓN Y PROCESAMIENTO

A partir de los conceptos desarrollados en el apartado anterior, la solución puede entenderse como una capa de integración colocada sobre herramientas ya existentes. Esta capa no sustituye las plataformas de administración, sino que recupera información de ellas, la transforma y la presenta de manera común. Así, las consolas originales siguen disponibles para diagnósticos detallados, mientras que el tablero semanal funciona como una síntesis institucional.

La Tabla 1 resume las capas funcionales que intervienen en un sistema de este tipo. La separación por capas permite explicar qué componente aporta datos, cuál los procesa, dónde se normalizan y cómo se publican para consulta. Esta organización también ayuda a delimitar responsabilidades y a reducir el riesgo de exponer detalles técnicos innecesarios.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 28 - 46

Tabla 1

Hardware, software y comunicación por capa funcional

Capa funcional	*Hardware* o plataforma	*Software* o componente	Forma de comunicación
Fuentes de información	Puntos de acceso, *switches*, enlaces WAN y plataformas institucionales existentes	Aruba Central, Cisco Meraki Dashboard y estadísticas NOC/Cacti	API REST, consulta de gráficas históricas y captura asistida
Integración automatizada	Servidor Linux institucional	*Scripts* en Python, tareas programadas y archivos de configuración JSON	Consultas HTTPS hacia APIs y lectura de archivos intermedios
Normalización y ensamblado	Almacenamiento local del proyecto	Procesos Python de validación, conversión de unidades y generación de indicadores	Intercambio mediante archivos JSON crudos, normalizados y ensamblados
Visualización y publicación	Servidor web institucional	NGINX, HTML, JavaScript, hojas de estilo y certificado SSL	Publicación del tablero mediante HTTPS para consulta desde navegador

Nota. Se describen componentes generales para explicar la arquitectura sin exponer modelos, rutas, credenciales, direcciones ni parámetros internos.

La Figura 1 representa la arquitectura conceptual de la solución. En ella, se observa el paso desde sedes y fuentes de datos hacia una capa de integración, un proceso de normalización y una visualización ejecutiva.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 29 - 46

Figura 1

Arquitectura conceptual del sistema automatizado de monitoreo multi-sede

Nota. Figura de elaboración propia con apoyo de una herramienta de inteligencia artificial generativa para fines de representación visual. No contiene datos reales, rutas, direcciones ni identificadores sensibles.

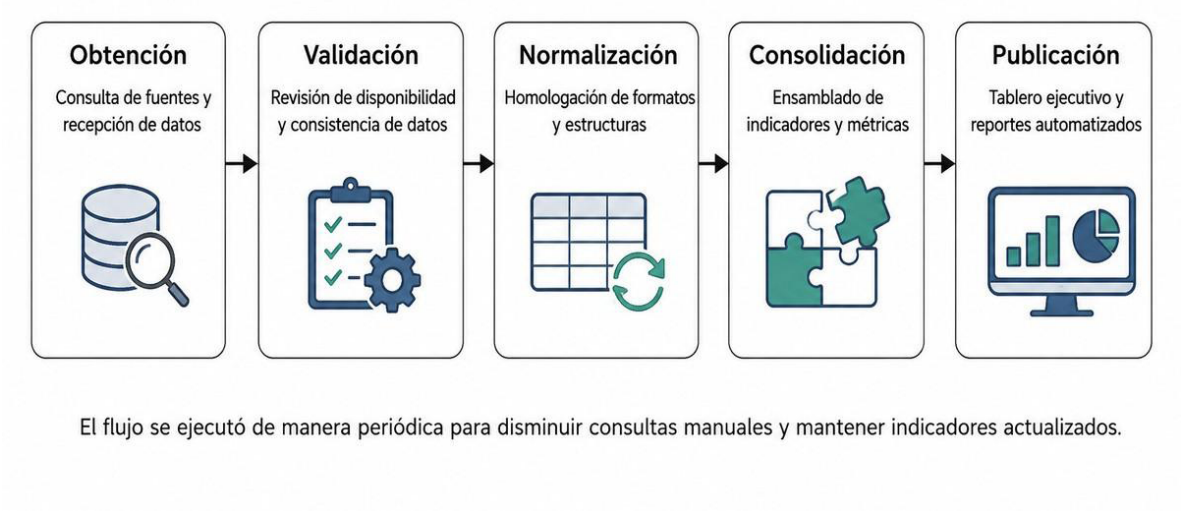
Esta representación muestra que el valor del sistema depende de articular fuentes existentes, procesos de integración y criterios de presentación comunes. Con ello, la arquitectura conceptual establece la base del flujo operativo que transforma una revisión dispersa en una consulta semanal organizada.

La Figura 2 complementa la arquitectura al mostrar el flujo general de procesamiento. La secuencia inicia con la identificación de fuentes y continúa con la extracción, validación, normalización, ensamblado, publicación y revisión operativa. Este flujo sirvió como guía para ordenar la implementación del sistema desarrollado.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 30 - 46

Figura 2

Flujo general de procesamiento de información en un sistema automatizado de monitoreo



Nota. Figura de elaboración propia con apoyo de una herramienta de inteligencia artificial generativa para fines de representación visual. No contiene datos reales ni información sensible.

Este flujo estableció que la confiabilidad del tablero dependía de separar obtención, validación, normalización y publicación, permitiendo revisar cada fuente sin afectar la lógica general del reporte semanal.

2.3 METODOLOGÍA DE DESARROLLO

La metodología se organizó en cuatro etapas: diseño, configuración, instalación y pruebas. Estas fases permitieron pasar de la identificación del problema a la puesta en operación de una aplicación funcional, conservando evidencia básica de las decisiones tomadas. La selección de una arquitectura modular se apoyó en trabajos recientes que destacan la conveniencia de desacoplar componentes de monitoreo para facilitar mantenimiento, crecimiento, integración de datos y generación automatizada de reportes (Khan y Khan, 2018; Quiñones-Cuenca *et al.*, 2025; Rafiq *et al.*, 2025).

La Tabla 2 muestra la relación entre cada etapa metodológica, su propósito general y su aplicación dentro del sistema. Esta tabla funciona como puente entre la explicación conceptual y el caso práctico que se describe en los apartados siguientes.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 31 - 46

Tabla 2

Etapas metodológicas utilizadas para el desarrollo del sistema

Etapa metodológica	Propósito general	Aplicación en el sistema
Diseño	Definir problema, sedes, indicadores y restricciones.	Cinco entornos, fuentes disponibles, métricas comunes y datos sensibles a excluir.
Configuración	Preparar parámetros, archivos y mecanismos de consulta.	JSON por fuente, extractores, datos crudos, datos normalizados y rutas de trabajo.
Instalación	Desplegar componentes y publicar resultados.	Servidor Linux, *scripts* en Python, NGINX, HTTPS y ejecución programada.
Pruebas	Validar datos, carga y correspondencia con fuentes.	Comparación contra consolas, revisión de archivos intermedios y ajustes al tablero.

Nota. Elaboración con base en la metodología seguida durante el diseño, configuración, instalación y pruebas del sistema automatizado de monitoreo.

Las etapas no se aplicaron de forma aislada. Cada una alimentó a la siguiente y, durante las pruebas, fue necesario regresar a etapas previas para ajustar archivos de configuración, validar métricas o corregir la forma en que una fuente entregaba información. Esta dinámica permitió mejorar el sistema sin cambiar su objetivo principal: integrar información de red dispersa y presentarla como un reporte semanal útil para seguimiento institucional.

2.4 APLICACIÓN DE LA METODOLOGÍA EN EL SISTEMA DEL IBUNAM

Una vez establecida la base conceptual y la metodología, el caso de implementación se describe siguiendo las mismas etapas. Esto permite observar cómo los conceptos de monitoreo multi-sede, automatización, normalización y visualización ejecutiva se tradujeron en decisiones concretas dentro del IBUNAM.

2.4.1 ETAPA DE DISEÑO: CONTEXTO, SEDES E INDICADORES

Durante la etapa de diseño, se delimitó el alcance institucional del sistema. Se consideraron cinco entornos de operación: el IBUNAM y el Pabellón Nacional de la Biodiversidad, ubicados en la Ciudad de México, así como la Estación La Posta en Colima, la Estación de Biología Los Tuxtlas en Veracruz y la Estación de Biología Chamela en Jalisco.

Esta distribución geográfica permitió dimensionar mejor el problema operativo. No se trataba únicamente de revisar equipos de red en un mismo edificio, sino de dar seguimiento a espacios con condiciones distintas de conectividad, administración, disponibilidad de datos y mecanismos de consulta. Por ello, desde el diseño se buscó una solución flexible, capaz de integrar información automática y también datos complementarios validados por personal técnico.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 32 - 46

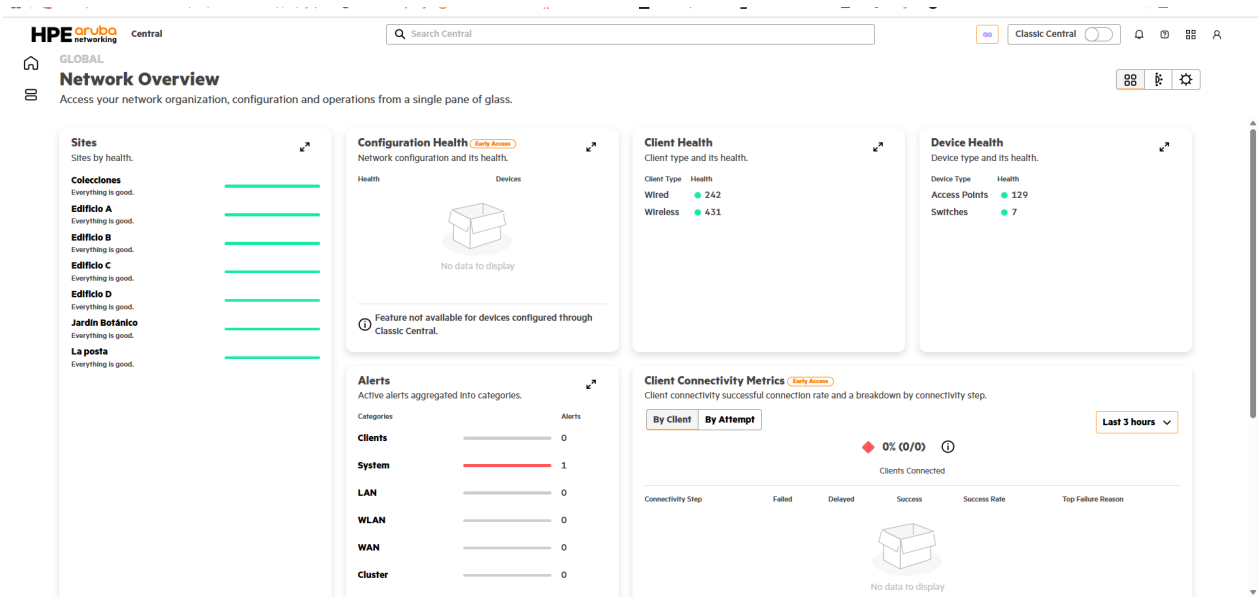
Antes de la implementación, la revisión del estado de la red se realizaba mediante consultas independientes. Para conocer el estado inalámbrico del IBUNAM, era necesario entrar a una plataforma; para revisar el Pabellón Nacional de la Biodiversidad, se consultaba otra consola; y para sedes foráneas, se recurría a gráficas históricas u observaciones operativas. Esta fragmentación dificultaba preparar una lectura semanal común.

También se definió que el reporte no debía mostrar información de configuración interna, direcciones sensibles, rutas, credenciales ni identificadores técnicos críticos. Esta restricción condicionó la forma de presentar resultados, porque el objetivo era ofrecer visibilidad operativa sin comprometer la seguridad de los servicios.

La primera fuente considerada fue Aruba Central, utilizada para la administración y supervisión de la infraestructura inalámbrica del IBUNAM. Desde esta plataforma, se obtuvieron indicadores relacionados con puntos de acceso, clientes conectados y estado operativo de la red inalámbrica. La Figura 3 muestra una vista de esta consola, donde se apreció la revisión por sitios, clientes y estado general de la red inalámbrica.

Figura 3

Vista de Aruba Central como fuente de indicadores inalámbricos institucionales

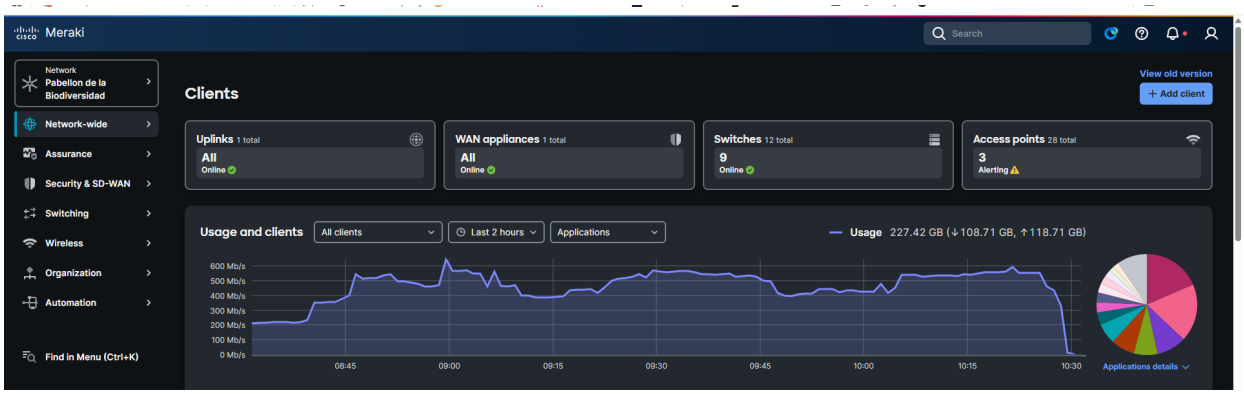


La segunda fuente fue Cisco Meraki Dashboard, empleada para el seguimiento de infraestructura del Pabellón Nacional de la Biodiversidad. Esta consola concentró información sobre equipos administrados desde la nube, clientes, conectividad y operación general de red. La Figura 4 presenta la vista operativa utilizada para consultar clientes, uso de red y estado de los equipos administrados desde la nube.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 33 - 46

Figura 4

Vista de Cisco Meraki Dashboard como fuente de indicadores del Pabellón Nacional de la Biodiversidad

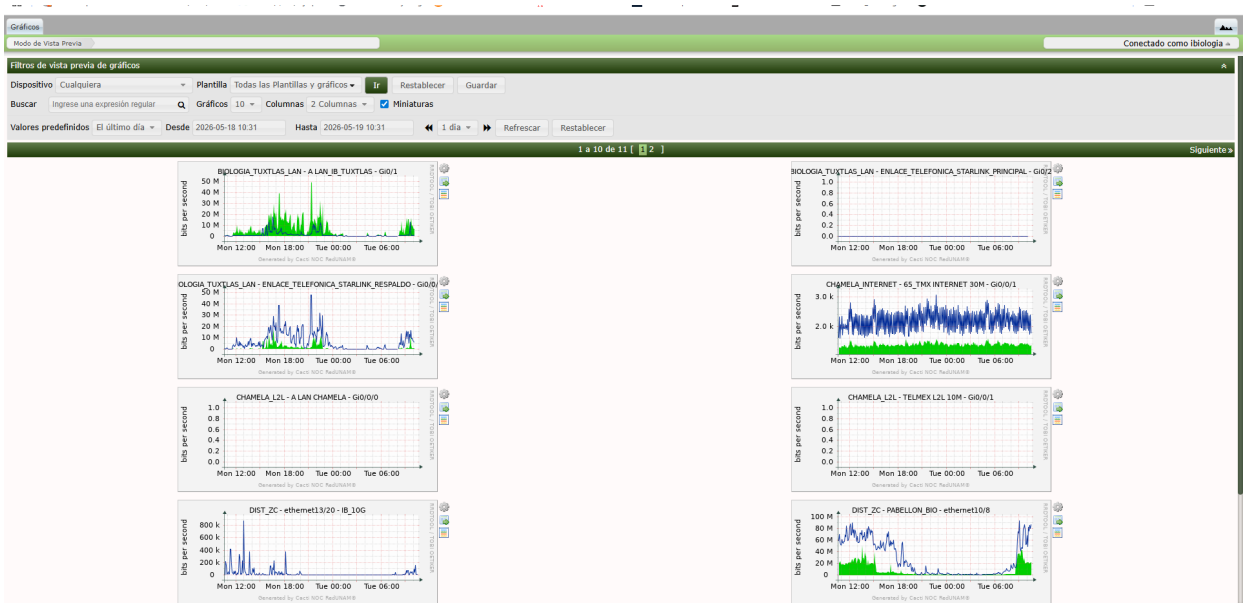


Nota. Se utiliza únicamente con fines descriptivos del flujo de monitoreo.

La tercera fuente correspondió al sistema de estadísticas de red del NOC, basado en gráficas históricas de comportamiento de enlaces. Esta plataforma permitió revisar tendencias de tráfico y disponibilidad observada en enlaces WAN, especialmente para sedes cuya información se evaluaba mediante comportamiento histórico. En la Figura 5 se muestra una representación de las gráficas utilizadas para complementar la captura de indicadores WAN.

Figura 5

Vista de gráficas históricas de enlaces en la plataforma NOC/Cacti



Nota. Se muestra una vista general de gráficas históricas de enlaces sin exponer credenciales.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 34 - 46

En conjunto, estas tres fuentes permitieron cubrir distintos niveles de la operación de red: la infraestructura inalámbrica institucional, los equipos administrados desde plataformas en la nube y el comportamiento histórico de enlaces. Su incorporación confirmó la necesidad de integrar información que originalmente se encontraba distribuida en consolas distintas, con alcances y formatos diferentes. Esta revisión inicial sirvió como base para definir qué datos podían automatizarse, cuáles requerían validación técnica y cómo debían organizarse para alimentar el reporte semanal.

La Tabla 3 resume las fuentes incorporadas y el tipo de datos que se aportaron al reporte semanal. La selección de estas plataformas respondió a su uso real dentro de la operación institucional y a la necesidad de reunir información que antes se revisaba de manera separada.

Tabla 3

Fuentes de información integradas en el reporte semanal

Fuente de información	Uso dentro de la operación	Indicadores aprovechados
Aruba Central	Administración de infraestructura inalámbrica del Instituto de Biología	Puntos de acceso, clientes conectados y estado operativo inalámbrico
Cisco Meraki Dashboard	Supervisión de infraestructura del Pabellón Nacional de la Biodiversidad	Dispositivos administrados, clientes, conectividad y estado general de red
Estadísticas NOC/Cacti	Seguimiento histórico de enlaces y tráfico entre sedes	Tendencias WAN, comportamiento de tráfico y disponibilidad observada
Captura asistida	Complemento para sedes sin integración automática completa	Valores operativos validados por el personal técnico
Aplicación local de reporte	Consolidación y publicación de indicadores semanales	Datos normalizados, históricos y visualización ejecutiva

Nota. Elaborada con base en las fuentes utilizadas durante la implementación del sistema.

La integración de estas fuentes confirmó que el sistema no debía depender de una única plataforma ni de un solo tipo de dato. Fue necesario combinar información automática, registros históricos y captura asistida para representar el estado semanal de la red. Con esta base, la etapa siguiente organizó archivos, parámetros y estructuras internas para transformar estos insumos en datos comparables.

2.4.2 ETAPA DE CONFIGURACIÓN: FUENTES, ARCHIVOS Y ESTRUCTURA INTERNA

En la etapa de configuración, se prepararon los archivos por fuente, los parámetros de consulta, las rutas internas de trabajo y las estructuras de salida. En esta fase, también se ajustaron los mecanismos de autenticación requeridos por las plataformas con API, así como los criterios para almacenar datos crudos y normalizados. La configuración se realizó de forma gradual, primero con una fuente y después con las demás, para reducir errores y facilitar la validación.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 35 - 46

La organización interna del proyecto se definió pensando en operación y mantenimiento, no sólo en desarrollo. Para ello, los archivos de configuración se separaron de los programas de extracción y los datos generados se guardaron en niveles distintos: información original recibida desde las plataformas, datos depurados, archivos normalizados e históricos, así como el archivo consolidado para la visualización.

Además de la descripción por directorio, se incorporó una vista simplificada del árbol del proyecto con la relación entre los componentes principales del sistema. Esta representación se muestra en la Figura 6 de manera general y no incluye rutas absolutas, credenciales ni parámetros internos de operación.

Figura 6

Estructura simplificada del sistema automatizado de monitoreo

```
[raíz del proyecto]/
├── config/
│   ├── sitios.json
│   ├── fuentes.json
│   ├── aruba_ibunam.json
│   ├── meraki_pabellon.json
│   └── pfsense_chamela_wan.json
├── extractors/
│   ├── extract_aruba.py
│   ├── extract_meraki.py
│   ├── extract_pfsense.py
│   └── capturar_wan_sedes.py
├── data/
│   ├── raw/
│   ├── normalized/
│   ├── assembled/
│   └── history/
├── assemblers/
│   └── assemble_reporte_semanal.py
├── utils/
│   ├── refresh_aruba_token.py
│   └── run_weekly_pipeline.sh
├── web/
│   ├── reporte_semanal_dashboard.html
│   ├── css/
│   └── js/
└── logs/
    ├── weekly_pipeline.log
    └── refresh_aruba_token.log
```

Nota. La estructura se muestra de forma simplificada para explicar la organización funcional del sistema; se omitieron rutas absolutas, credenciales, *tokens*, direcciones y parámetros de configuración.

En esta organización, los archivos de configuración definieron sedes y fuentes de datos; los extractores recuperaron información desde las plataformas integradas; los directorios de datos separaron información original, normalizada, ensamblada e histórica; el ensamblador generó el archivo consolidado del reporte; las utilerías apoyaron tareas programadas; el directorio web concentró los elementos publicados; y los registros permitieron revisar ejecuciones, errores y validaciones operativas.

La Tabla 4 resume la estructura general utilizada en el sistema. Por razones de seguridad, se presentan rutas relativas y funciones generales, sin incluir rutas absolutas del servidor, credenciales, identificadores internos ni archivos de configuración sensibles.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 36 - 46

Tabla 4

Organización estructural del sistema automatizado

Directorio o componente	Función principal	Descripción operativa
/config/	Configuración	Parámetros generales, sedes y fuentes; sin exponer credenciales.
/extractors/	Extracción	*Scripts* de consulta a plataformas y recuperación de métricas.
/data/raw/	Datos originales	Salidas preliminares obtenidas desde cada fuente
/data/normalized/	Normalización	Datos homologados para comparación y procesamiento común.
/data/assembled/	Consolidación	Indicadores finales usados por el tablero ejecutivo.
/utils/	Apoyo operativo	Tareas auxiliares, validaciones y ejecución calendarizada.
/web/	Publicación	Archivos HTML, JavaScript y recursos visuales del tablero.
/logs/	Registro	Evidencia básica de ejecuciones, errores y resultados.

Nota. Elaboración con base en la organización funcional del sistema desarrollado. Se omitieron rutas absolutas, credenciales y parámetros internos por criterios de confidencialidad operativa.

Esta organización separó las funciones principales y facilitó el mantenimiento del sistema. Al distinguir entre configuración, extracción, normalización, consolidación, publicación y registros, el proyecto quedó preparado para incorporar nuevas fuentes o ajustar procesos existentes sin modificar toda la estructura. Con esta base, la siguiente etapa consistió en instalar los componentes en el servidor institucional y habilitar la publicación segura del tablero.

2.4.3 ETAPA DE INSTALACIÓN Y PUBLICACIÓN

La etapa de instalación consistió en colocar los extractores, el ensamblador y el tablero web dentro de un servidor con sistema operativo Ubuntu Linux. Este entorno fue seleccionado por su estabilidad, facilidad para ejecutar procesos programados y compatibilidad con herramientas de automatización. Sobre esta base, se organizaron carpetas de configuración, *scripts* de extracción, datos crudos, información normalizada, históricos y archivos ensamblados para publicación.

Los procesos de obtención de información se desarrollaron principalmente mediante *scripts* en Python, apoyados por archivos de configuración en JSON y tareas de ejecución programada en el sistema operativo. Cada extractor tuvo una función específica: consultar una plataforma, recuperar métricas, validar la respuesta, convertir la información a una estructura común y almacenarla en archivos intermedios.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 37 - 46

En las plataformas administradas desde la nube, la consulta automática requirió atender aspectos propios de autenticación, vigencia de *tokens* y formatos de respuesta. Esta condición obligó a validar que las consultas no dependieran de una sesión abierta en navegador, sino de mecanismos controlados por la propia aplicación. Con esto, se redujo la intervención manual y se evitó que el reporte dependiera exclusivamente de capturas visuales realizadas por el personal técnico.

La información obtenida se almacenó en archivos JSON porque este formato facilitó el intercambio entre módulos y permitió revisar datos de manera sencilla durante las pruebas. En una primera etapa, se conservaron archivos crudos para validar qué entregaba cada fuente; después, se generaron archivos normalizados con campos comunes, unidades homogéneas y nombres consistentes. Finalmente, un proceso de ensamblado integró los indicadores por sede y preparó el archivo consumido por el tablero web.

La publicación de resultados fue realizada mediante un tablero desarrollado con HTML, JavaScript y hojas de estilo, servido desde NGINX. El motor de presentación quedó del lado del navegador: JavaScript leyó el archivo JSON consolidado y construyó dinámicamente tarjetas por sede, gráficas, semáforos de estado y observaciones operativas. Para proteger la consulta del sitio, se habilitó un certificado SSL expedido por la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC), de modo que la página pudiera abrirse mediante HTTPS y la comunicación entre el navegador y el servidor quedara cifrada.

2.4.4 ETAPA DE PRUEBAS Y VALIDACIÓN OPERATIVA

La etapa de pruebas se concentró en revisar consistencia de datos, disponibilidad de archivos, carga del tablero, legibilidad de gráficas y congruencia entre lo observado en las plataformas originales y lo mostrado en el reporte consolidado. Cuando algún indicador no coincidía, se revisaba el extractor correspondiente, el archivo normalizado o la etapa de ensamblado. Este proceso permitió ajustar la aplicación antes de usarla como insumo de seguimiento semanal.

Cada extractor verificó que la fuente estuviera disponible y que los datos obtenidos tuvieran la estructura esperada. Cuando alguna fuente no entregaba información o presentaba diferencias con lo observado en su consola original, se revisaba el origen del dato antes de incorporarlo al reporte. Esta validación redujo inconsistencias y evitó publicar indicadores incompletos.

Las pruebas también permitieron detectar diferencias entre la información vista en las consolas y la información disponible para extracción automática. En algunos casos, los paneles gráficos mostraban datos agregados que no se obtenían de forma idéntica por API o que requerían transformaciones adicionales. Esta situación confirmó la necesidad de validar cada métrica antes de incorporarla al tablero y de documentar el origen de los indicadores.

Una vez publicado el reporte, el personal técnico podía comparar la información consolidada con las plataformas originales cuando era necesario. Esta revisión permitió validar que el tablero reflejara adecuadamente el estado semanal de las sedes y, al mismo tiempo, conservó la posibilidad de acudir a cada sistema especializado para diagnósticos más detallados.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 38 - 46

2.4.5 NORMALIZACIÓN DE DATOS

Para que los datos procedentes de plataformas distintas fueran comparables, no se utilizaron directamente los campos originales de cada sistema. Primero, se identificó la dimensión operativa de cada dato —por ejemplo, disponibilidad, conectividad WAN, operación inalámbrica, clientes conectados u observación técnica— y, después, se transformó a un registro común. Esta decisión evitó comparar valores que sólo tenían sentido dentro de la consola de origen y permitió que el tablero trabajara con indicadores homogéneos.

La regla lógica aplicada por los extractores puede resumirse como una función de normalización N (fuente, dato crudo, semana) que produce un registro normalizado con los campos: sede, fuente, indicador, valor, unidad, estado, método de obtención, fecha de consulta y observación operativa. El procesamiento siguió la siguiente secuencia: 1) validar que la fuente respondiera y que el dato tuviera estructura mínima; 2) asociar el dato con una sede institucional mediante un catálogo de configuración; 3) convertir unidades cuando fue necesario, por ejemplo, de bits por segundo, kilobits o megabits a Mbps; 4) homologar nombres y categorías de estado, como disponible, atención, crítico o sin dato; 5) marcar el origen como automático o asistido; y 6) guardar el resultado en archivos JSON normalizados para su ensamblado semanal. Esta lógica coincide con enfoques que ubican la agregación y normalización como una etapa previa necesaria para analizar datos heterogéneos en sistemas de monitoreo (Poltavtseva, 2020).

De esta manera, Aruba Central, Cisco Meraki Dashboard, las estadísticas NOC/Cacti y la captura asistida no se mezclaron como datos equivalentes sin revisión previa. Cada fuente aportó indicadores dentro de su propia dimensión operativa y el sistema sólo comparó métricas previamente transformadas a una estructura común. Cuando una fuente no entregó información completa, el registro se conservó como sin dato o como dato asistido, evitando presentar como automático un valor que requería validación técnica.

Esta formalización permitió resolver las diferencias entre los datos crudos de las plataformas y la información mostrada en sus paneles gráficos. El objetivo no fue replicar cada tablero propietario, sino extraer los elementos necesarios para construir una lectura institucional común, verificable y documentada, adecuada para el reporte semanal de estado de red.

2.5 VISUALIZACIÓN EJECUTIVA, SEGURIDAD Y ALCANCE

El tablero ejecutivo fue diseñado para responder una pregunta básica de operación: ¿cuál era el estado general de la red institucional durante la semana revisada? Para ello, se integraron tarjetas de resumen, gráficas de comportamiento, indicadores por sede y observaciones operativas. La intención no fue reemplazar los diagnósticos especializados, sino ofrecer una primera lectura clara para seguimiento y toma de decisiones.

La visualización separó la información por tipo de indicador: disponibilidad, conectividad, operación inalámbrica, tendencias y observaciones. Esta organización facilitó que el personal técnico identificara rápidamente dónde debía profundizar una revisión. Además, el tablero permitió distinguir entre datos obtenidos automáticamente y datos incorporados mediante captura asistida, lo cual fue importante para mantener transparencia sobre el origen de la información.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 39 - 46

La Figura 7 presenta una vista del tablero desarrollado para el reporte semanal de estado de red. En esta interfaz, se concentraron las fuentes de información, los sitios con datos, la obtención automática o manual de indicadores y el semáforo por sede. La captura fue utilizada para mostrar el resultado operativo del sistema y no como un inventario público de infraestructura.

Figura 7

Tablero ejecutivo desarrollado para el reporte semanal de estado de red

Nota.Se presenta con fines descriptivos y no expone credenciales ni rutas internas de configuración.

Asimismo, el tablero incorporó una herramienta de exportación a PDF, con el propósito de conservar una copia digital del estado del reporte en un momento determinado. Esta función permite generar registros documentales periódicos del tablero ejecutivo, útiles para fines de seguimiento, consulta histórica, respaldo institucional y comparación entre semanas. De esta manera, la información visualizada no quedó limitada a la consulta en pantalla, sino que pudo resguardarse como evidencia documental del comportamiento general de la red durante el periodo revisado.

Desde el inicio, se estableció que el reporte no debía convertirse en un inventario público de infraestructura. Por esa razón, se limitaron los datos visibles a indicadores generales y se evitó documentar configuraciones específicas, direcciones de red, credenciales, rutas internas o detalles que pudieran facilitar acciones indebidas. Esta decisión fue congruente con los lineamientos de confidencialidad aplicables a reportes técnicos sobre servicios TIC.

También se buscó que una falla parcial no impidiera la generación completa del reporte. Si una fuente no entregaba información en una ejecución determinada, el sistema debía permitir revisar el incidente sin

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 40 - 46

detener necesariamente el resto del flujo. Esta característica fue útil porque las plataformas dependían de servicios externos, conectividad y autenticaciones que podían presentar variaciones.

El sistema se concentró en indicadores operativos de red y no en el análisis profundo de seguridad, rendimiento de aplicaciones o monitoreo de servidores. Tampoco se planteó como plataforma de alertamiento en tiempo real. Su propósito fue construir un reporte semanal que permitiera observar tendencias, comparar sedes y disponer de una vista integrada de conectividad e infraestructura inalámbrica.

Esta delimitación permitió mantener el proyecto en un alcance viable. El valor del sistema estuvo en integrar información que ya existía, organizarla bajo un flujo común y presentarla de manera útil para seguimiento institucional. En consecuencia, la aplicación quedó preparada para crecer, pero sin perder el objetivo inicial de apoyar la supervisión operativa multi-sede.

3. RESULTADOS

La implementación permitió integrar en una misma vista información que antes se encontraba distribuida entre plataformas independientes y con diversidad tecnológica. Este resultado fue relevante porque la red institucional no se administra desde una sola herramienta: Aruba Central concentraba parte importante de la operación inalámbrica, Cisco Meraki aportaba información del Pabellón Nacional de la Biodiversidad y las gráficas del NOC permitían revisar comportamiento de enlaces. El tablero semanal reunió esos insumos en una lectura común.

El primer cambio observado fue la reducción de la fragmentación operativa. Antes del sistema, la preparación de un reporte requería abrir varias consolas, revisar gráficas, comparar datos y elaborar una síntesis manual. Después de la implementación, el personal técnico contó con un flujo que recuperó datos, los normalizó y los presentó en un formato homogéneo. Esto no eliminó la necesidad de revisar las plataformas originales cuando se requería diagnóstico detallado, pero sí facilitó la primera evaluación semanal.

El segundo resultado fue la incorporación explícita de cinco entornos institucionales distribuidos geográficamente: IBUNAM y Pabellón Nacional de la Biodiversidad en la Ciudad de México; Estación La Posta en Colima; Estación de Biología Los Tuxtlas en Veracruz; y Estación de Biología Chamela en Jalisco. Esta cobertura permitió observar la operación como un conjunto, no como revisiones aisladas de cada sede.

El tercer resultado fue la generación de una base histórica de indicadores. Al conservar reportes semanales, el sistema permitió revisar tendencias de conectividad, disponibilidad y comportamiento inalámbrico. Esta información será útil para identificar variaciones recurrentes, planear mantenimientos y justificar necesidades de mejora con datos organizados.

La existencia de un histórico también permitió comparar semanas sin depender de capturas aisladas o notas sueltas. Para las sedes foráneas, esta capacidad fue relevante porque los cambios de comportamiento en enlaces o servicios no siempre se observan en una sola revisión. Al acumular reportes semanales, el personal técnico pudo contar con una referencia más clara para distinguir entre una variación temporal y un patrón que requería seguimiento.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 41 - 46

Otro resultado fue la separación entre información técnica detallada e información ejecutiva. El sistema mostró estados, tendencias y observaciones, pero no expuso configuraciones internas ni identificadores críticos. Esta separación permitió comunicar el estado general de la operación sin comprometer la seguridad de la infraestructura.

Además, el reporte favoreció una comunicación más ordenada entre revisión técnica y seguimiento de coordinación. El personal especializado pudo conservar el acceso a las plataformas originales para análisis profundo, mientras que el tablero ofreció una síntesis útil para explicar el estado general de la red. Esta diferencia ayudó a evitar reportes excesivamente técnicos cuando sólo se requería una lectura operativa inicial.

Desde la operación diaria, el cambio también se observó en la forma de iniciar la atención de posibles incidentes. Antes de la implementación, en la red cableada se esperaba generalmente el reporte de una persona usuaria; a partir de esa notificación, se solicitaba la ubicación física, se identificaba el IDF correspondiente y se revisaba el nodo, el *switch* o el puerto asociado. En la red inalámbrica, ocurría algo semejante: después del reporte de baja señal o falta de conectividad, se ingresaba a la plataforma de la marca correspondiente para ubicar el punto de acceso o el *switch* relacionado. Con el tablero, la revisión inicial pasó a apoyarse en una visualización operativa continua y en indicadores consolidados, lo que permitió orientar con mayor rapidez la búsqueda del componente afectado. Esta mejora se considera principalmente una evidencia observacional del equipo técnico, complementada con las métricas de tiempo y ejecución que se presentan más adelante.

La Tabla 5 presenta una comparación entre el proceso previo y el proceso posterior a la implementación. El análisis mostró que el cambio más importante no fue únicamente tecnológico, sino organizativo: la información pasó de revisarse de forma dispersa a integrarse en una lectura semanal común.

Tabla 5

Comparación operativa antes y después de la implementación

Actividad	Antes de la implementación	Después de la implementación
Consulta de plataformas	Revisión independiente en consolas y gráficas separadas	Consulta consolidada desde el tablero semanal
Obtención de datos	Captura manual y validación caso por caso	Extracción automática y captura asistida cuando fue necesario
Seguimiento de sedes	Lecturas aisladas por ubicación o plataforma	Indicadores organizados por sede y tipo de servicio
Preparación de reportes	Integración manual de información dispersa	Ensamblado automatizado de datos normalizados
Comunicación operativa	Dependencia de explicaciones técnicas separadas	Visualización ejecutiva con indicadores resumidos
Protección de información	Riesgo de incluir detalles técnicos innecesarios	Presentación limitada a datos operativos no sensibles

Nota. La tabla fue elaborada con base en la comparación del proceso operativo antes y después de la implementación.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 42 - 46

Esta comparación muestra que la implementación modificó la forma de iniciar la revisión operativa de la red. Antes, el seguimiento dependía de consultas separadas e integración manual; después, el tablero permitió una primera lectura común por sede e indicador. Así, los resultados reflejaron una mejora técnica y una forma más ordenada de documentar, comunicar y dar seguimiento al estado semanal de la infraestructura.

Además de la comparación operativa, en la Tabla 6 se incorporaron métricas de referencia obtenidas durante una ejecución controlada del *pipeline* en el servidor institucional. Estas mediciones permitieron documentar recursos utilizados, duración de procesamiento, volumen de archivos JSON y beneficios estimados en la revisión inicial semanal.

Tabla 6

Métricas de operación y rendimiento incorporadas al análisis de resultados

Indicador	Resultado documentado	Interpretación operativa
Duración y estado del *pipeline*	66.99 segundos; código de salida 0; estado: éxito	La generación completa del flujo semanal se ejecutó correctamente en poco más de un minuto.
Uso de recursos del proceso medido	CPU: 88%; memoria máxima residente: 30,556 KB, aproximadamente 29.8 MB	El proceso utilizó recursos de CPU durante un periodo breve y mantuvo bajo consumo de memoria.
Evidencia JSON e histórico	205 archivos JSON; 1,104,998 bytes; semanas 2026-W22 a 2026-W25	El sistema conserva datos crudos, normalizados, ensamblados e históricos para comparación periódica.
Nivel de automatización de flujos	5 flujos automáticos y 2 asistidos; 71.4% automatizados	La mayor parte de la integración se resolvió mediante extracción automatizada, manteniendo captura asistida cuando la fuente lo requirió.
Tiempo de revisión inicial semanal	Antes: 45 a 60 minutos; después: 5 a 10 minutos	La consulta inicial se redujo aproximadamente entre 78% y 92%, de acuerdo con la estimación operativa del equipo técnico.
Cambio en la atención inicial	De una revisión reactiva por reporte de usuario a una revisión guiada por tablero	La herramienta permitió orientar con mayor rapidez la búsqueda por sede, IDF, *switch*, puerto o punto de acceso, sin sustituir el diagnóstico especializado.

Nota. Las métricas del pipeline corresponden a una ejecución controlada realizada el 21 de junio de 2026 en el servidor institucional. Los tiempos de revisión manual y con tablero son estimaciones operativas del equipo técnico, utilizadas como referencia para documentar la mejora en la consulta inicial semanal.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 43 - 46

Estos datos complementan las observaciones de mejora operativa del personal técnico. En conjunto, muestran que el sistema ordenó información dispersa, redujo el tiempo para iniciar la revisión semanal, mantuvo bajo consumo de recursos y dejó evidencia histórica para seguimiento institucional.

3.1 ANÁLISIS DE LOS RESULTADOS

Los resultados indicaron que la automatización aportó valor principalmente al ordenar el proceso de supervisión. En una red distribuida, el problema no siempre consiste en la ausencia de datos, sino en que esos datos se encuentran en sistemas distintos, con formatos diferentes y niveles variables de actualización. La aplicación desarrollada redujo esa dispersión y permitió que la primera revisión semanal partiera de una base común.

La incorporación de métricas cuantitativas no elimina el valor de la percepción técnica del personal responsable, pero sí permite respaldarla con evidencia mínima de operación. En este caso, las mediciones de ejecución del *pipeline* y la estimación de reducción de tiempo muestran que el tablero funcionó como apoyo para iniciar revisiones con mayor rapidez, mientras que la identificación precisa del incidente continuó dependiendo de las plataformas especializadas y de la experiencia del equipo de TIC.

La integración de plataformas también mostró que no todas las sedes podían tratarse igual. Las fuentes con API permitieron automatizar consultas de forma más directa, mientras que las fuentes basadas en gráficas o revisiones históricas requirieron una combinación de captura y validación. Esta diferencia no se consideró una falla del sistema, sino una condición real de operación que debía incorporarse al diseño.

El tablero ejecutivo mejoró la lectura inicial del estado de red porque tradujo información técnica en indicadores consultables. Esta transformación fue importante para apoyar tanto al personal técnico como a responsables de coordinación que requerían conocer el estado general sin entrar a cada plataforma. En consecuencia, el sistema fortaleció la comunicación operativa y facilitó el seguimiento institucional.

Además, la generación de datos históricos permitió que el reporte dejara de ser una fotografía aislada. Al conservar información semanal, se abrió la posibilidad de analizar tendencias, reconocer cambios en la demanda y anticipar necesidades de mantenimiento. Este componente será especialmente útil para sedes foráneas, donde la detección temprana de degradaciones puede reducir tiempos de atención.

En conjunto, la experiencia mostró que la integración automatizada no sustituyó el criterio del equipo técnico. Más bien, lo apoyó al reducir tareas repetitivas, ordenar la información y hacer visible el comportamiento general de la red. Las decisiones de diagnóstico y corrección siguieron dependiendo de especialistas, pero éstos contaron con información más accesible y mejor organizada.

El proyecto también evidenció que una solución de este tipo requiere mantenimiento operativo. Las APIs pueden cambiar, los permisos de consulta pueden modificarse y las plataformas externas pueden ajustar la forma en que entregan sus datos. Por ello, la aplicación no se concibió como un producto cerrado, sino como un sistema sujeto a revisión periódica, documentación interna y ajustes conforme evolucionen las herramientas de administración utilizadas por la institución.

Esta característica fue relevante porque el entorno de red universitario no permanece estático. La incorporación de nuevos puntos de acceso, cambios en enlaces, crecimiento de usuarios o incorporación de nuevas sedes pueden modificar los indicadores necesarios. La arquitectura modular permitió que

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 44 - 46

esos cambios pudieran atenderse mediante nuevos extractores, ajustes de normalización o ampliación del tablero, sin reconstruir todo el sistema desde el inicio y de forma gradual.

4. CONCLUSIONES

El sistema automatizado de monitoreo de red multi-sede cumplió el objetivo planteado al integrar información proveniente de plataformas heterogéneas, generar indicadores semanales y presentar una visualización ejecutiva para fortalecer la supervisión de servicios de comunicaciones en una institución universitaria. La implementación permitió pasar de una revisión fragmentada a un flujo común de obtención, normalización, ensamblado y publicación de datos.

La aportación principal consistió en construir una capa de integración sobre herramientas existentes. Esta decisión fue adecuada porque aprovechó plataformas ya utilizadas por el personal técnico, evitó duplicar funciones de administración y permitió avanzar sin sustituir la infraestructura de monitoreo disponible. En particular, la combinación de Aruba Central, Cisco Meraki, estadísticas del NOC y captura asistida permitió atender condiciones distintas entre sedes metropolitanas y foráneas.

La experiencia mostró que la distribución geográfica de las sedes no sólo representaba un dato contextual, sino una condición operativa que debía considerarse en el diseño. Supervisar espacios ubicados en Ciudad de México, Colima, Veracruz y Jalisco requería una herramienta capaz de reunir información de conectividad y disponibilidad sin depender de una revisión manual completa en cada plataforma.

Los resultados demostraron que la automatización mejoró la disponibilidad de información para seguimiento semanal, facilitó la comunicación técnica, permitió construir una base histórica útil para planeación y redujo el tiempo de consulta inicial de la red. El tablero ejecutivo no reemplazó las herramientas especializadas, pero sí ofreció una vista inicial integrada para identificar sedes, indicadores o tendencias que requerían mayor revisión.

Como continuidad del proyecto, convendría ampliar la automatización de fuentes que aún dependen de captura asistida, incorporar alertas tempranas para eventos relevantes y evaluar mecanismos de análisis predictivo a partir de históricos. También sería pertinente extender la cobertura hacia otros servicios tecnológicos, manteniendo el mismo criterio de mínima exposición de información sensible.

Finalmente, la implementación confirmó que una institución universitaria con sedes distribuidas puede fortalecer la gestión de su red mediante soluciones modulares, escalables y ajustadas a su operación real. El proyecto dejó una base técnica para continuar mejorando la supervisión, la continuidad y la planeación de los servicios digitales institucionales.

AGRADECIMIENTOS

Se agradece a la Unidad de Tecnologías de la Información y la Comunicación del Instituto de Biología de la UNAM, así como al personal de las sedes consideradas, por el apoyo brindado durante la validación operativa del sistema y la revisión de información necesaria para consolidar el reporte semanal.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 45 - 46

Declaración de contribución de autoría

Jorge Gerardo López Ibarra: Especialista en administración de redes, telecomunicaciones, seguridad informática y automatización de procesos TIC. Rol dentro del proyecto: diseño técnico, implementación operativa, integración de fuentes de información y desarrollo del sistema de reporte. Su contribución consistió en analizar la situación operativa del monitoreo de red institucional, identificar las fuentes de información disponibles y diseñar la arquitectura funcional del sistema automatizado. Participó directamente en la construcción de *scripts* de extracción, normalización y ensamblado de indicadores, así como en la integración de datos provenientes de plataformas de administración inalámbrica, seguridad perimetral, monitoreo remoto y captura asistida para sedes foráneas. También desarrolló y ajustó el tablero ejecutivo utilizado para presentar el reporte semanal, validó la consistencia de los datos obtenidos y documentó los criterios técnicos necesarios para proteger información sensible de infraestructura. Su participación permitió convertir una actividad de revisión dispersa en un proceso ordenado y repetible de supervisión institucional.

David Velázquez Portilla: Especialista en coordinación de servicios TIC, gestión operativa institucional y supervisión de infraestructura tecnológica. Rol dentro del proyecto: coordinación institucional, validación de necesidades operativas, seguimiento estratégico y revisión funcional del servicio implementado. Su contribución se centró en orientar el proyecto desde la perspectiva de las necesidades institucionales de supervisión y continuidad de servicios tecnológicos. Participó en la definición del alcance operativo, en la priorización de indicadores relevantes para el seguimiento de sedes y en la validación de que el reporte semanal fuera útil para actividades de coordinación técnica y comunicación institucional. También aportó criterios para que la solución mantuviera un equilibrio entre profundidad técnica, claridad ejecutiva y confidencialidad de información sensible. Su participación ayudó a vincular el desarrollo técnico con las necesidades reales de operación de la Unidad de Tecnologías de la Información y la Comunicación, fortaleciendo la pertinencia del sistema como herramienta de apoyo para la toma de decisiones.

REFERENCIAS

Boluda-Prieto, M., Sánchez, M., & Fernández, J. (2026). Resilient edge-to-cloud architecture with self-healing and self-correcting mechanisms for industrial data continuity. *Computers & Industrial Engineering*, *213*, 111795. https://doi.org/10.1016/j.cie.2025.111795

Espinel-Villalobos, R. I., Ardila-Triana, E., Zarate-Ceballos, H., & Ortiz-Triviño, J. E. (2022). Design and implementation of network monitoring system for campus infrastructure using software agents. *Ingeniería e Investigación*, *42*(1), e87564. https://doi.org/10.15446/ing.investig.v42n1.87564

Farahmand, H., Young, W. A., & Minsker, B. S. (2022). A network observability framework for sensor placement in infrastructure systems. *Reliability Engineering & System Safety*, *221*, 108363. https://doi.org/10.1016/j.ress.2022.108366

Khan, R., & Khan, S. U. (2018). Design and implementation of an automated network monitoring and reporting back system. *Journal of Industrial Information Integration*, *9*, 24-34. https://doi.org/10.1016/j.jii.2017.11.001

Lee, S., Levanti, K., & Kim, H. S. (2014). Network monitoring: Present and future. *Computer Networks*, *65*, 84-98. https://doi.org/10.1016/j.comnet.2014.03.007

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.181 **Vol. 4, Núm. 3.** julio-septiembre 2026, págs. 46 - 46

Poltavtseva, M. A. (2020). Heterogeneous data aggregation and normalization in information security monitoring and intrusion detection systems of large-scale industrial CPS. *Proceedings of the Institute for System Programming of the RAS*, *32*(5), 131-142. https://doi.org/10.15514/ISPRAS-2020-32(5)-10.

Quiñones-Cuenca, M., González, F., & Pérez, J. (2025). Architecture for Automated Real-Time Bidirectional Data Performance Analysis of LoRaWAN Networks. *Automation*, *6*(3), 38. https://doi.org/10.3390/automation6030038

Rafiq, A., Shakir, M. Z., Gray, D., Inglis, J., & Ferguson, F. (2025). AI and IoT-driven monitoring and visualisation for optimising MSP operations in multi-tenant networks: A modular approach using sensor data integration. *Sensors*, *25*(19), 6248. https://doi.org/10.3390/s25196248

Ta-Armart, W., & Savithi, C. (2026). Operational management of multi-vendor Wi-Fi networks in smart campus environments. *Technologies*, *14*(4), 204. https://doi.org/10.3390/technologies14040204

Witczak, D., Kelm, P., & Rybarczyk, A. (2024). Review of monitoring and control systems based on the Internet of Things. *Applied Sciences*, *14*(19), 8943. https://doi.org/10.3390/app14198943

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.188 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 47 - 59

Diseño de una arquitectura de almacenamiento distribuido basada en Ceph RADOS Block Device para virtualización y nube privada institucional

Design of a distributed storage architecture based on Ceph RADOS Block Device for virtualization and institutional private cloud

Información del reporte:

Para citar este reporte técnico:

Mares Mendoza, E. (2026). Diseño de una arquitectura de almacenamiento distribuido basada en Ceph RADOS Block Device para virtualización y nube privada institucional. *Cuadernos Técnicos Universitarios de la DGTIC, 4*(3), páginas (47 - 59). https://doi.org/10.22201/dgtic.30618096e.2026.4.3.188

Enrique Mares Mendoza

Dirección General de Cómputo y de Tecnologías de Información y Comunicación

Universidad Nacional Autónoma de México

enrique.mares@unam.mx

ORCID: 0009-0008-5529-0080

Resumen

El aumento de los servicios tecnológicos universitarios llevó a plantear una solución de almacenamiento capaz de sostener ambientes de virtualización y servicios de nube privada con mayor estabilidad, crecimiento controlado y administración unificada. Con ese propósito, se diseñó una arquitectura basada en la plataforma Ceph orientada al almacenamiento en bloque, considerando las necesidades operativas de un centro de datos institucional.

La metodología consistió en revisar los requerimientos técnicos, calcular la capacidad disponible, definir la organización física y lógica del clúster, y analizar su conexión con plataformas de virtualización. Se consideraron servidores, unidades de almacenamiento de alto rendimiento, redes separadas para tráfico de acceso y operación interna, mecanismos de réplica, distribución de datos, control de permisos y criterios para mantener margen ante fallas o expansión.

Como resultado, se obtuvo una propuesta integrada por siete nodos de almacenamiento, con una distribución que permitió mejorar la tolerancia a fallas, ordenar el uso de recursos y separar responsabilidades por plataforma. También se identificaron aspectos que debían atenderse antes de una operación plena, como monitoreo continuo, protección de credenciales, políticas de crecimiento y revisión de escenarios de recuperación. Se concluyó que la arquitectura planteada ofreció una base técnica viable

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.188 **Vol. 4, Núm. 3.** julio-septiembre 2026, págs. 48 - 59

para fortalecer los servicios institucionales de virtualización y nube privada, disponibles en el centro de datos, siempre que su adopción estuviera acompañada de documentación, seguimiento operativo y ajustes derivados del comportamiento real del entorno.

Palabras clave: Almacenamiento distribuido, almacenamiento en bloque, sistema de almacenamiento distribuido Ceph, tolerancia a fallas.

Abstract

The increase in university technological services led to the proposal of a storage solution capable of supporting virtualization environments and private cloud services with greater stability, controlled growth, and unified administration. For this purpose, an architecture based on Ceph and oriented toward block storage was designed, considering the operational needs of an institutional data center.

The methodology consisted of reviewing technical requirements, calculating available capacity, defining the physical and logical organization of the cluster, and analyzing its connection with virtualization platforms. Servers, high-performance storage drives, separate networks for access traffic and internal operations, replication mechanisms, data distribution, permission control, and criteria for maintaining margin in the face of failures or expansion were considered.

As a result, a proposal consisting of seven storage nodes was obtained, with a distribution that made it possible to improve failure tolerance, organize resource usage, and separate responsibilities by platform. Aspects that needed to be addressed before full operation were also identified, such as continuous monitoring, credential protection, growth policies, and the review of recovery scenarios. It was concluded that the proposed architecture offered a viable technical foundation for strengthening institutional virtualization and private cloud services, as long as its adoption was accompanied by documentation, operational follow-up, and adjustments derived from the real behavior of the environment.

Keywords: *Distributed storage, block storage, Ceph distributed storage system, fault tolerance.*

1. INTRODUCCIÓN

El crecimiento de los servicios digitales, la virtualización de servidores y las plataformas de nube privada incrementaron la importancia de contar con infraestructuras de almacenamiento capaces de ofrecer capacidad, disponibilidad, continuidad operativa y administración centralizada. En este escenario, el almacenamiento distribuido se consolidó como una alternativa relevante frente a esquemas tradicionales, debido a que permitió distribuir datos entre múltiples nodos, reducir dependencias de dispositivos aislados y acompañar el crecimiento progresivo de las cargas institucionales. Para una dependencia universitaria como la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) de la Universidad Nacional Autónoma de México (UNAM), esta necesidad adquirió especial relevancia, ya que el Centro de Datos soportó servicios tecnológicos que demandaron resiliencia, escalabilidad y operación controlada para atender plataformas de virtualización y nube privada.

Las investigaciones y experiencias técnicas recientes evidencian el uso de tecnologías abiertas en instituciones académicas y centros de investigación. Higuera-Balderrama *et al.* (2023) analizaron el caso del Instituto Tecnológico de La Paz (ITLP), donde la plataforma Ceph fue seleccionada como solución

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.188 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 49 - 59

de almacenamiento distribuido de código abierto por su flexibilidad, disponibilidad e integración con Proxmox VE. De forma complementaria, Zhao *et al.* (2024) propusieron una nube privada educativa basada en OpenStack y Ceph para la gestión unificada y autoservicio de recursos de cómputo, almacenamiento y red. Asimismo, Bocchi *et al.* (2024) documentaron el uso de Ceph en CERN para OpenStack, CephFS y S3, dentro de estrategias de continuidad operativa y recuperación ante desastres. Estas experiencias sustentan su pertinencia en entornos institucionales que requieren escalabilidad, resiliencia y administración centralizada.

A partir de esos antecedentes, en febrero de 2025, se identificó la necesidad de definir una arquitectura de almacenamiento distribuido basada en Ceph RBD para el Centro de Datos de la DGTIC-UNAM, orientada a plataformas consumidoras como Proxmox, OpenStack y Hyper-V. El problema técnico no se limitó únicamente a estimar la capacidad, sino que implicó establecer un diseño coherente entre nodos físicos, unidades NVMe, configuración de red, reglas de distribución de datos, políticas de acceso y criterios operativos. Por ello, el alcance del reporte se delimitó al diseño técnico y dimensionamiento de una arquitectura RBD, sin incluir CephFS, RADOS Gateway (RGW) ni Metadata Server (MDS) como componentes funcionales, con el propósito de concentrar el análisis en el almacenamiento de bloque para discos de máquinas virtuales y volúmenes persistentes.

El objetivo general fue diseñar una arquitectura de almacenamiento distribuido basada en Ceph RBD para plataformas de virtualización y nube privada institucional, considerando capacidad, disponibilidad, red, seguridad, distribución de datos e integración operativa como base para su implementación en producción en el Centro de Datos de la DGTIC-UNAM. Decido al carácter técnico de la propuesta, el análisis se orientó a personal vinculado con la operación de centros de datos, redes, almacenamiento y virtualización

2. DESARROLLO TÉCNICO

El desarrollo técnico se centró en el análisis y dimensionamiento conceptual de una arquitectura de almacenamiento distribuido basada en Ceph RBD en el Centro de Datos de la DGTIC-UNAM. Durante el diseño se examinaron componentes físicos, lógicos, operativos y de integración necesarios para proponer una plataforma de almacenamiento en bloque orientada a virtualización y nube privada institucional. Sin embargo, no se limitó a estimar la capacidad, sino que incorporó criterios de disponibilidad, separación de tráfico, seguridad, distribución de datos, dominios de falla y administración operativa.

Para ello, Ceph se consideró pertinente gracias a que su modelo de distribución mediante CRUSH (*Controlled Replication Under Scalable Hashing*) permite organizar la ubicación de los datos y sus réplicas entre los OSD (*Object Storage Daemon*). Estudios recientes señalan que la distribución equilibrada de datos en Ceph es crítica para evitar puntos calientes (tanto lógicos como físicos); además, reduce cargas desiguales y mejora la utilización de dispositivos en escenarios de gran volumen y alta concurrencia (Miao *et al.*, 2024). Con base en este enfoque, la propuesta se delimitó al uso de RBD como servicio principal de almacenamiento en bloque, debido a que este componente permite provisionar dispositivos de almacenamiento de este tipo que pueden ser utilizados directamente por plataformas de virtualización.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.188 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 50 - 59

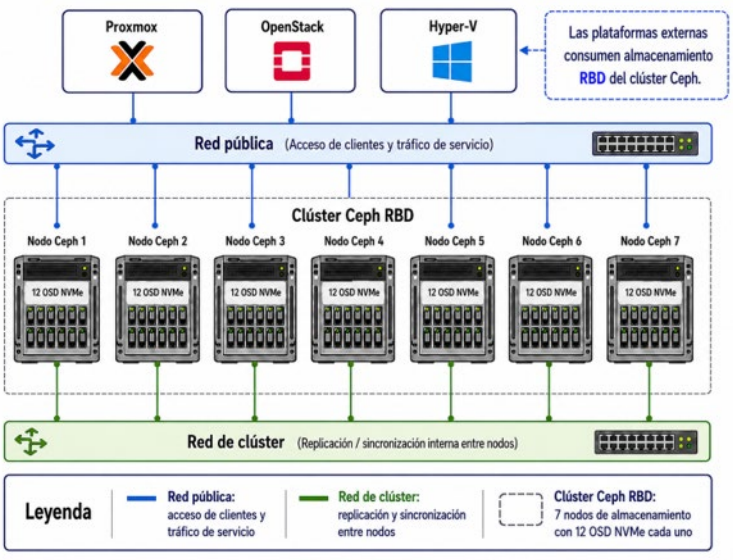
2.1 METODOLOGÍA

La metodología se estructuró en cuatro fases generales: análisis de requerimientos, dimensionamiento de capacidad, definición de arquitectura lógica y revisión de integración con plataformas como OpenStack, Proxmox y Hyper-V. Esta organización permitiría relacionar las necesidades institucionales del Centro de Datos de la DGTIC-UNAM con los componentes técnicos requeridos para una implementación productiva de almacenamiento en bloque, considerando capacidad, disponibilidad, red, seguridad, operación e integración con entornos de virtualización y nube privada.

En primer lugar, se definió una arquitectura física compuesta por siete nodos Ceph, cada uno con doce unidades NVMe de 15.36 TB, dos procesadores Intel Xeon 6747P y 2 TB de memoria RAM DDR5 por nodo. Bajo el criterio de asignar un OSD por cada unidad NVMe, se estableció un diseño conceptual de 84 OSD en total. Esta definición permitiría plantear una base física suficiente para distribuir las cargas de almacenamiento entre distintos servidores, reducir la dependencia de un único dispositivo, aislar fallas a nivel de nodo y proyectar crecimiento de manera ordenada. Además, cada nodo contempló dos tarjetas de red de cuatro puertos de 25 GbE, lo que permite disponer de ocho puertos de 25 GbE por servidor y facilita la separación del tráfico asociado al acceso de las plataformas consumidoras y a las operaciones internas del clúster, como se representa en la Figura 1.

Figura 1

Arquitectura física propuesta del clúster Ceph RBD y plataformas consumidoras



Con base en esta arquitectura, se realizó el dimensionamiento de capacidad para estimar el alcance operativo del clúster. La capacidad bruta por nodo se calculó en 184.32 TB y la capacidad bruta total en 1,290.24 TB. Bajo un esquema de réplica 3, la capacidad útil aproximada fue de 430.08 TB; sin embargo, este valor no debía considerarse como capacidad totalmente disponible para consumo.

Debido a que Ceph requiere espacio libre para procesos de recuperación, *backfill*, rebalanceo, crecimiento y tolerancia ante fallas, se propuso una ocupación operativa máxima de 80%, equivalente a 344.06 TB.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.188 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 51 - 59

Esta estimación permitiría establecer un margen de operación más seguro para la planeación de la infraestructura, como se resume en la Tabla 1.

Tabla 1

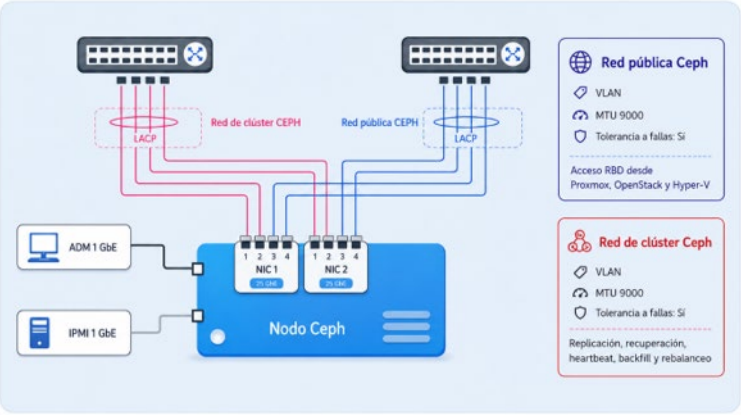
Dimensionamiento de capacidad bruta, replicada y operativa del clúster Ceph RBD

Concepto	Valor propuesto
Número de nodos	7
Discos por nodo	12 NVMe
Capacidad por disco	15.36 TB
OSD totales	84
Capacidad bruta por nodo	184.32 TB
Capacidad bruta total	1,290.24 TB
Factor de réplica	3
Capacidad útil aproximada	430.08 TB
Capacidad operativa al 80 %	344.06 TB

Posteriormente, se definió la configuración conceptual de red como un componente esencial para la operación del clúster. La red pública se destinó al acceso de las plataformas consumidoras hacia RBD, mientras que la red de clúster se reservó para procesos internos de Ceph, como replicación, recuperación, *heartbeat*, *backfill* y rebalanceo entre OSD. Asimismo, se consideraron criterios de *bonding,* LACP (*Link Aggregation Control Protocol*), MTU (*Maximum Transmission Unit*), VLAN, rutas, latencia, tolerancia a fallas y separación de dominios de *broadcast*. Esta separación permitió reducir el riesgo de que las operaciones internas del clúster afectaran el acceso de las plataformas de virtualización, especialmente en escenarios de recuperación, crecimiento o redistribución de datos, como se muestra en la Figura 2.

Figura 2

Diseño lógico de red de un nodo Ceph: separación entre tráfico público RBD y tráfico interno del clúster



https://doi.org/10.22201/dgtic.30618096e.2026.4.3.188 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 52 - 59

En continuidad con el diseño físico y de red, se propuso desplegar cinco monitores para favorecer el *quorum* y mejorar la resiliencia ante fallas. Esta distribución permitió considerar la continuidad de los servicios de control del clúster, ya que los monitores son necesarios para mantener los mapas del estado general de Ceph, incluyendo OSD, *pools*, *placement groups*, monitores y CRUSH.

Tabla 2

Distribución propuesta de servicios MON y MGR

Nodo	Función MON	Función MGR	Rol propuesto	Observaciones de disponibilidad
ceph-node01	Sí	Sí	MON + MGR activo inicial	Nodo participante en *quorum* y punto inicial de administración del clúster
ceph-node02	Sí	Sí	MON + MGR *standby*	Nodo participante en *quorum* y respaldo del servicio MGR
ceph-node03	Sí	Sí	MON + MGR *standby*	Nodo participante en *quorum* y segundo respaldo del servicio MGR
ceph-node04	Sí	No	MON	Nodo participante en *quorum*
ceph-node05	Sí	No	MON	Nodo participante en *quorum*
ceph-node06	No	No	OSD	Nodo dedicado principalmente a servicios de almacenamiento
ceph-node07	No	No	OSD	Nodo dedicado principalmente a servicios de almacenamiento

Asimismo, se consideraron tres instancias MGR (Ceph Manager) distribuidas en nodos físicos distintos, una activa y dos en espera, debido a que este servicio aporta funciones de administración, monitoreo y exposición de métricas. Con ello, se buscó evitar la concentración de servicios críticos en un único servidor físico y fortalecer la disponibilidad operativa de la arquitectura, como se sintetiza en la Tabla 2.

2.2 DESARROLLO DE LA ARQUITECTURA PROPUESTA

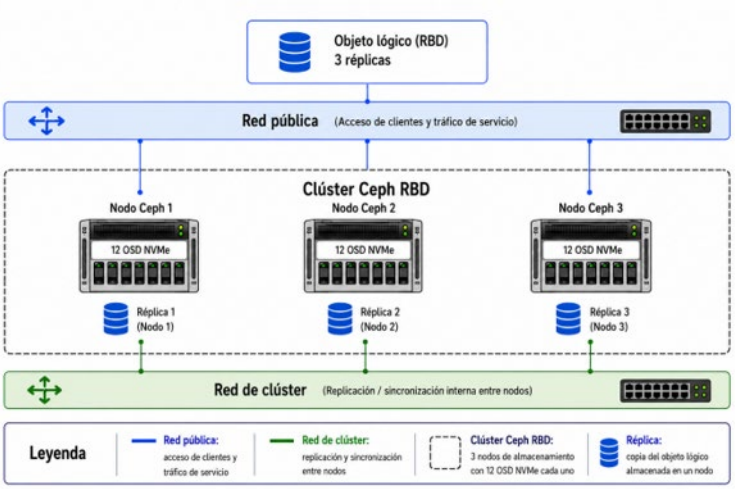
El *backend* de almacenamiento se definió sobre BlueStore, debido a que se consideró una arquitectura de *object store* diseñada para la capa RADOS de Ceph. Esta elección fue pertinente porque BlueStore permite almacenar los datos directamente en dispositivos de bloque y gestionar metadatos mediante RocksDB, evitando la dependencia de un sistema de archivos convencional en la ruta principal de datos y reduciendo limitaciones asociadas a FileStore, como la sobrecarga por *journaling* (Lee *et al.*, 2017). A partir de ello, se diseñó la distribución lógica mediante CRUSH *map*, mecanismo que permite organizar la dispersión de datos, reglas de replicación y dominios de falla sin depender de un servicio centralizado

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.188 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 53 - 59

de localización de objetos (Goggin *et al*., 2025). En consecuencia, se propuso una estructura por *buckets* de *host* y una regla *replicated_rule* con dominio de falla por *host*, para ubicar tres réplicas en nodos físicos distintos y fortalecer la resiliencia de la arquitectura, como se ilustra en la Figura 3.

Figura 3

Distribución conceptual de réplicas RBD mediante CRUSH map con dominio de falla por host



Con base en el uso exclusivo de RBD, se propusieron *pools* separados por plataforma o función, a fin de facilitar la operación, el monitoreo y la administración del almacenamiento. Esta segmentación permitiría aplicar permisos, cuotas y políticas diferenciadas según la plataforma consumidora, además de mejorar el aislamiento operativo, la trazabilidad y el control del crecimiento. Los parámetros considerados fueron *size 3*, *min_size 2*, *pg_autoscale_mode on* y *aplicación rbd*, como se muestra en la Tabla 3.

Tabla 3

Diseño conceptual de pools RBD por plataforma consumidora

Pool	Propósito	Plataforma	Réplica	Autoscaling	Aplicación
rbd_proxmox	Discos de VM	Proxmox	3 max – 2 min	on	rbd
rbd_openstack_volumes	Volúmenes persistentes	OpenStack Cinder	3 max – 2 min	on	rbd
rbd_openstack_images	Imágenes	OpenStack Glance	3 max – 2 min	on	rbd
rbd_openstack_vms	Discos de instancias	OpenStack Nova	3 max – 2 min	on	rbd
rbd_hyperv	Discos para Hyper-V	Hyper-V	3 max – 2 min	on	rbd

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.188 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 54 - 59

Respecto a los *placement groups*, no se definió una asignación fija, ya que su cantidad depende del número de OSD, los *pools* creados, el consumo esperado, el tamaño relativo de cada *pool* y el comportamiento real del clúster. Por ello, se consideró *pg_autoscale_mode on*, puesto que la documentación oficial de Ceph indica que el *autoscaling* permite ajustar o recomendar el número de PG por *pool* con base en la utilización observada y esperada (Ceph Documentation, s. f.). Esta decisión evitaría una configuración inicial arbitraria y favorecería una administración flexible del almacenamiento en escenarios productivos institucionales, como se muestra en la Tabla 4.

Tabla 4

Estimación de placement groups por pool RBD

Pool	Uso estimado	Autoscaling	PG inicial sugerido	Criterio de ajuste	Observaciones
rbd_proxmox	Alto	on	Validar por autoscaler	Uso real y crecimiento	No fijar arbitrariamente
rbd_openstack_volumes	Alto	on	Validar por autoscaler	Capacidad asignada a Cinder	Considerar cuotas por proyecto
rbd_openstack_images	Medio	on	Validar por autoscaler	Cantidad y tamaño de imágenes	Puede crecer menos que volúmenes
rbd_openstack_vms	Alto	on	Validar por autoscaler	Uso de Nova con RBD	Considerar impacto de arranque
rbd_hyperv	Alto	on	Validar por autoscaler	Integración no nativa	Considerar impacto de arranque

Como parte de los criterios operativos, se consideraron umbrales de capacidad *nearfull*, *backfillfull*, *full* y *failsafe_full*, junto con acciones preventivas y correctivas orientadas a conservar margen de operación y evitar condiciones críticas de llenado.

En materia de seguridad, se contempló el uso de CephX con usuarios separados por plataforma o servicio, bajo el principio de privilegios mínimos. Esta decisión mejoraría el control de acceso, protegería los *keyrings*, separaría responsabilidades y mantendría trazabilidad sobre el uso del almacenamiento, como se muestra en la Tabla 5.

Tabla 5

Matriz de control de acceso CephX por plataforma

Usuario CephX	Plataforma	Pool autorizado	Permisos	Alcance
client.proxmox	Proxmox	rbd_proxmox	R/W sobre *pool* asignado	Discos VM
client.cinder	OpenStack Cinder	rbd_openstack_volumes	R/W sobre volúmenes	Volúmenes persistentes

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.188 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 55 - 59

Usuario CephX	Plataforma	Pool autorizado	Permisos	Alcance
client.glance	OpenStack Glance	rbd_openstack_images	R/W o R según flujo	Imágenes
client.nova	OpenStack Nova	rbd_openstack_vms	R/W sobre *pool* de instancias	Discos de VM
client.openstack	OpenStack general	*Pools* requeridos	Restringido por servicio	Uso administrativo limitado
client.hyperv	Hyper-V	rbd_hyperv	R/W sobre *pool* asignado	Integración Hyper-V

Finalmente, se revisaron los mecanismos de integración con las plataformas consumidoras. Proxmox se planteó mediante RBD nativo; OpenStack, a través de Cinder, Nova y Glance; e Hyper-V, mediante *rbd-wnbd*.

Tabla 6

Matriz comparativa de integración de Ceph RBD con plataformas de virtualización y nube privada

Plataforma	Servicio	Uso	Integración	Ventajas
Proxmox	QEMU/KVM	Discos de VM	RBD nativo en Proxmox	Integración directa y almacenamiento distribuido
OpenStack	Cinder	Volúmenes persistentes	Driver Ceph RBD	Volúmenes distribuidos y escalables
OpenStack	Nova	Discos de instancias	libvirt/QEMU con RBD	Reduce dependencia de disco local
OpenStack	Glance	Imágenes	*Backend* RBD opcional	Uso eficiente con Cinder/Nova
Hyper-V	VM	Discos para VM	*rbd-wnbd*	Exposición de RBD a Windows

Esta revisión permitió identificar las condiciones técnicas necesarias para incorporar el almacenamiento en bloque a los entornos de virtualización y nube privada considerados en el diseño, como se compara en la Tabla 6.

3. RESULTADOS

Se logró, como resultado principal, definir un clúster de siete nodos con doce unidades NVMe de 15.36 TB por nodo, equivalente a 84 OSD bajo el criterio de un OSD por dispositivo. Este dimensionamiento permitió estimar 1,290.24 TB de capacidad bruta y 430.08 TB útiles con réplica 3; no obstante, se identificó una capacidad operativa cercana a 344.06 TB para conservar margen ante recuperación, *backfill*, rebalanceo

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.188 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 56 - 59

y crecimiento. El análisis consolidó una propuesta de arquitectura Ceph RBD para virtualización y nube privada institucional. Los resultados deben interpretarse como producto de un diseño conceptual y de dimensionamiento técnico.

Desde la perspectiva de disponibilidad, el uso propuesto de réplica 3 y una regla CRUSH con dominio de falla por *host* permitiría distribuir copias en nodos físicos distintos, por lo que el beneficio esperado sería reducir el riesgo ante fallas de disco o servidor. Este planteamiento es coherente con antecedentes que señalan la importancia de evitar dispositivos aislados y presentar almacenamiento distribuido como recurso único para plataformas de virtualización (Higuera-Balderrama *et al*., 2023). Asimismo, la separación entre red pública y red de clúster se identificó como una decisión de diseño orientada a disminuir interferencias entre el acceso de las plataformas consumidoras y el tráfico interno de replicación o recuperación.

Finalmente, los beneficios planteados, como tolerancia a fallas, administración por *pools*, separación de responsabilidades y crecimiento controlado, deben entenderse como beneficios esperados y no como mejoras cuantitativamente demostradas. Por ello, antes de una implementación productiva será necesario validar rendimiento con cargas reales, ajustar PG mediante *autoscaling*, proteger credenciales CephX, revisar umbrales de llenado y comprobar la integración de Hyper-V.

4. CONCLUSIONES

El análisis desarrollado permitió establecer que una arquitectura de almacenamiento distribuido basada en Ceph RBD representa una alternativa técnicamente viable para fortalecer la infraestructura de virtualización y nube privada institucional del Centro de Datos de la DGTIC-UNAM. Más que centrarse únicamente en la capacidad disponible, el diseño evidenció la importancia de articular de manera coherente la disponibilidad, la distribución de datos, la separación de tráfico, la seguridad y la operación continua como elementos indispensables para una plataforma de almacenamiento institucional.

Asimismo, el trabajo identificó que la solidez de la propuesta no depende de un único componente, sino de la correcta integración entre la arquitectura física, la lógica de distribución mediante CRUSH, la administración de *pools*, el control de acceso y la planeación operativa. La propuesta contribuyó a definir una base técnica ordenada para orientar decisiones futuras sobre crecimiento, integración con plataformas de virtualización y continuidad del servicio.

Se concluyó que la arquitectura propuesta debe acompañarse de una planeación operativa gradual, documentación técnica, monitoreo constante y políticas claras de administración. Para ello, se recomendó revisar el comportamiento del clúster ante cargas reales, escenarios de falla, ajustes de red, políticas de llenado, monitoreo y mecanismos de integración, con el propósito de reducir riesgos operativos y consolidar una implementación segura, escalable y administrable.

REFERENCIAS

Bocchi, E., Lekshmanan, A., Valverde, R., & Goggin, Z. (2024). Enabling storage business continuity and disaster recovery with Ceph distributed storage. *EPJ Web of Conferences, 295*, 01021. https://doi.org/10.1051/epjconf/202429501021

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.188 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 57 - 59

Ceph Foundation. (s. f.). *Ceph Block Device*. Ceph Documentation. https://docs.ceph.com/en/squid/rbd/

Goggin, Z., Lekshmanan, A., Valverde, R., & Bocchi, E. (2025). Ceph at CERN in the multi-datacentre era. *EPJ Web of Conferences, 337*, 01331. https://doi.org/10.1051/epjconf/202533701331

Higuera-Balderrama, D., Morejón-López, D., Canosa-Reyes, R. M., Alonso-Labrada, R., & Marín-Hernández, F. (2023). Diseño e implementación de una red de almacenamiento distribuido basada en CEPH. *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI, 11*(Especial 2), 55–60. https://doi.org/10.29057/icbi.v11iEspecial2.10839

Jelten, J., Wollek, A., Frank, D., & Lasser, T. (2023). *Equilibrium: Optimization of Ceph cluster storage by size-aware shard balancing*. arXiv. https://doi.org/10.48550/arXiv.2310.15805

Lee, D.-Y., Jeong, K., Han, S.-H., Kim, J.-S., Hwang, J.-Y., & Cho, S. (2017). *Understanding write behaviors of storage backends in Ceph object store*. MSST 2017. https://msstconference.org/MSST-history/2017/Papers/CephObjectStore.pdf

Miao, Y., Fan, Z., Zhang, M., & Yang, L. (2024). A data balanced distribution algorithm based on Ceph storage. *Microelectronics & Computer, 41*(3), 90–97. https://doi.org/10.19304/J.ISSN1000-7180.2023.0180

Zhao, L., Hu, G., & Xu, Y. (2024). Educational resource private cloud platform based on OpenStack. *Computers, 13*(9), 241. https://doi.org/10.3390/computers13090241

ANEXO A. GLOSARIO DE TÉRMINOS TÉCNICOS

El presente glosario reúne los principales conceptos técnicos utilizados en el reporte con el propósito de facilitar la comprensión de la arquitectura de almacenamiento distribuido basada en Ceph RBD para plataformas de virtualización y nube privada institucional. Los términos incluidos corresponden a componentes, servicios, procesos y criterios operativos abordados en el diseño propuesto para el Centro de Datos de la DGTIC-UNAM.

Almacenamiento distribuido. Modelo de almacenamiento en el que los datos se reparten entre varios nodos interconectados por una red, es decir, no se guardan en un solo servidor o dispositivo.

Backfill. Proceso mediante el cual Ceph completa o redistribuye datos hacia otros OSD después de una falla, expansión o cambio en la distribución del clúster.

backfillfull. Umbral que puede limitar procesos de recuperación o redistribución cuando no existe suficiente espacio disponible para realizar *backfill* de manera segura.

BlueStore. Es el *backend* de almacenamiento utilizado por Ceph para guardar datos directamente en los discos administrados por los OSD.

Bonding. Técnica que permite agrupar varias interfaces de red físicas para mejorar redundancia, disponibilidad o capacidad agregada.

Bucket. Elemento jerárquico dentro del CRUSH *map* que agrupa dispositivos o nodos. Puede representar niveles como *host*, *rack*, sala o centro de datos.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.188 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 58 - 59

Ceph. Plataforma de almacenamiento distribuido de código abierto que permite integrar múltiples servidores y discos en un clúster unificado, capaz de ofrecer almacenamiento de bloque, archivos u objetos.

CephFS. Es el sistema de archivos distribuido de Ceph que permite almacenar y acceder a datos en forma de archivos y carpetas, similar a un sistema de archivos tradicional, pero distribuido entre varios nodos del clúster.

CephX. Mecanismo de autenticación de Ceph que permite controlar el acceso de usuarios y servicios a los recursos del clúster.

Cinder. Servicio de OpenStack encargado de administrar volúmenes persistentes.

CRUSH. Algoritmo de Ceph que determina la ubicación de los datos y sus réplicas dentro del clúster, sin depender de una tabla centralizada de localización.

CRUSH *map*. Mapa lógico que describe la jerarquía del clúster, incluyendo dispositivos, *hosts*, *buckets* y reglas de distribución de datos.

Failsafe_full. Umbral adicional de protección para evitar condiciones extremas de llenado que puedan comprometer la operación del clúster.

FileStore. *Backend* anterior de Ceph que dependía de un sistema de archivos convencional y de mecanismos de *journaling*. Se menciona como antecedente frente a BlueStore.

Full. Estado crítico en el que un OSD o el clúster alcanza un nivel de llenado que puede impedir nuevas escrituras.

Glance. Servicio de OpenStack encargado de administrar imágenes de máquinas virtuales.

Heartbeat. Comunicación interna entre componentes del clúster utilizada para verificar disponibilidad, detectar fallas y actualizar estados operativos.

Keyring. Archivo que contiene credenciales de autenticación CephX. Debe protegerse adecuadamente para evitar accesos no autorizados al almacenamiento.

LACP (*Link Aggregation Control Protocol*). Protocolo utilizado para la agregación dinámica de enlaces de red, comúnmente empleado en configuraciones de *bonding*.

Libvirt. Capa de administración de virtualización utilizada para gestionar hipervisores como KVM/QEMU.

MDS (*Metadata Server*). Es el servicio de Ceph encargado de administrar los metadatos de CephFS, como nombres de archivos, carpetas, permisos y rutas. No almacena directamente los datos del usuario, sino la información necesaria para organizar y acceder correctamente a los archivos dentro del sistema de archivos distribuido.

MGR (*Manager*). Es el servicio de Ceph encargado de recopilar información operativa del clúster y proporcionar funciones de administración, monitoreo y visualización. Complementa a los monitores al ofrecer métricas, módulos de gestión, panel web y herramientas que facilitan la supervisión del estado, rendimiento y capacidad del clúster.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.188 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 59 - 59

MON (*Monitor*). Servicio de Ceph encargado de mantener los mapas del clúster y participar en el *quorum*. Su función es esencial para conservar la consistencia del estado general del clúster.

MTU (*Maximum Transmission Unit*). Tamaño máximo de paquete que puede transmitirse por una interfaz de red. Su correcta configuración ayuda a evitar problemas de fragmentación o rendimiento.

Nearfull. Umbral de advertencia que indica que un OSD o el clúster se aproxima a un nivel alto de ocupación. Permite tomar acciones preventivas antes de llegar a una condición crítica.

Nova. Servicio de OpenStack encargado de la gestión de instancias de cómputo.

NVMe (*Non-Volatile Memory Express*). Es una tecnología de almacenamiento de alta velocidad utilizada principalmente en unidades de estado sólido.

OSD (*Object Storage Daemon*). Es el servicio de Ceph encargado de almacenar los datos en los discos del clúster. Cada OSD administra un disco o unidad de almacenamiento y participa en la distribución, réplica, recuperación y balanceo de los datos, permitiendo que Ceph mantenga la disponibilidad y tolerancia a fallos.

PG (*Placement Group*). Unidad lógica que agrupa objetos dentro de un *pool* y facilita su distribución entre los OSD. Su cantidad influye en el balance, desempeño y administración del clúster.

Pool. Agrupación lógica de almacenamiento dentro de Ceph. Permite separar datos por plataforma, servicio o función, por ejemplo, Proxmox, Cinder, Glance, Nova o Hyper-V.

Quorum. Mecanismo de consenso utilizado por los monitores de Ceph para mantener una visión consistente del estado del clúster. Requiere la disponibilidad de la mayoría de los monitores configurados.

RADOS (*Reliable Autonomic Distributed Object Store*). Es el servicio de Ceph que permite acceder al almacenamiento de objetos mediante interfaces compatibles con S3 y Swift.

RBD (RADOS *Block Device*). Proporciona almacenamiento en bloque y permite presentar discos virtuales o volúmenes persistentes a plataformas como Proxmox, OpenStack e Hyper-V.

Rrbd-wnbd. Herramienta que permite exponer dispositivos RBD de Ceph en sistemas Windows, y facilita su uso en entornos como Hyper-V.

RGW (RADOS *Gateway*). Es el servicio de Ceph que permite acceder al almacenamiento de objetos mediante interfaces compatibles con S3 y Swift.

RocksDB. Es una base de datos interna utilizada por Ceph, especialmente con BlueStore, para almacenar metadatos del OSD, tales como información sobre objetos, ubicaciones y estado del almacenamiento.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.185 **Vol. 4, Núm. 3.** julio-septiembre 2026, págs. 60 - 73

Implementación de la clase espejo como estrategia de internacionalización del currículo en Ciencias de la Educación

Implementation of the mirror class as a curriculum internationalization strategy in Educational Sciences

Información del reporte:

Para citar este reporte técnico:

Moreno Alcaraz, F., Castañeda García, H. I., Aguirre Sánchez, K. M., y Puga Villarreal, J. C. (2026). Implementación de la clase espejo como estrategia de internacionalización del currículo en Ciencias de la Educación. *Cuadernos Técnicos Universitarios de la DGTIC, 4*(3), páginas (60 - 73). https://doi.org/10.22201/dgtic.30618096e.2026.4.3.185

Flérida Moreno Alcaraz
Facultad de Ciencias de la Educación
Universidad Autónoma de Sinaloa
flerida@uas.edu.mx
ORCID: 0009-0001-3778-3147

Karla Marisol Aguirre Sánchez
Unidad Académica Ciencias de la Educación
Universidad Autónoma de Sinaloa
karlaaguirre@uas.edu.mx
ORCID: 0009-0007-7243-0498

Héctor Ignacio Castañeda García
Unidad Académica Ciencias de la Educación
Universidad Autónoma de Sinaloa
hcastaneda@uas.edu.mx
ORCID: 0000-0002-0275-9286

Juan Carlos Puga Villarreal
Facultad de Ciencias de la Educación
Universidad Autónoma de Sinaloa
pugavillarreal@uas.edu.mx
ORCID: 0009-0006-2887-1141

Resumen

Como estrategia de internacionalización del currículo, se implementó una clase espejo en la Licenciatura en Ciencias de la Educación de la Universidad Autónoma de Sinaloa durante el ciclo escolar 2025–2026. La intervención pedagógica se llevó a cabo en colaboración con la Universidad Nacional de Catamarca y contó con la

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.185 **Vol. 4, Núm. 3.** julio-septiembre 2026, págs. 61 - 73

participación de 27 estudiantes de México y cuatro de Argentina. La propuesta respondió a las limitaciones de movilidad física derivadas de las situaciones de riesgo que afectaron a Sinaloa durante el ciclo 2024-2025, periodo en el que las actividades presenciales fueron suspendidas. La metodología incluyó la planeación conjunta de sesiones sincrónicas y el uso de herramientas digitales como Padlet y Classroom, y la aplicación del método cualitativo de historia de vida centrado en la entrevista como instrumento de investigación, así como la viabilidad del modelo como alternativa de internacionalización sin movilidad física. Los resultados evidenciaron el desarrollo de competencias investigativas, interculturales y éticas en los participantes. Se concluye que la clase espejo constituyó una herramienta de resiliencia pedagógica que permitió sostener la calidad formativa en contextos de adversidad, y cuyo modelo resultó transferible a otras disciplinas de las ciencias sociales en instituciones de educación superior.

Palabras clave: Herramientas colaborativas, interculturalidad educativa, movilidad académica.

Abstract

As part of a strategy to internationalize the curriculum, a mirror class was implemented in the Licenciatura en Ciencias de la Educación program at the Universidad Autónoma de Sinaloa *during the 2025–2026 academic year. The pedagogical intervention was carried out in collaboration with the* Universidad Nacional de Catamarca *and involved 27 students from Mexico and four from Argentina. The initiative responds to the limitations on physical mobility resulting from the security risks that affected Sinaloa during the 2024–2025 academic year, a period in which in-person activities were suspended. The methodology included the joint planning of synchronous sessions and the use of digital tools such as Padlet and Classroom, as well as the application of the interview-centered qualitative life history method as a research instrument, and an assessment of the model's viability as an alternative for internationalization without physical mobility. The results demonstrated the development of research, intercultural, and ethical competencies among the participants. It is concluded that the mirror class served as a tool for pedagogical resilience that made possible to maintain educational quality in adverse contexts, and that this model proved transferable to other social science disciplines in college institutions.*

***Keywords**: Collaborative tools, intercultural education, academic mobility.*

1. INTRODUCCIÓN

En las últimas décadas, la internacionalización del currículo representó un eje estratégico para la mejora de la calidad educativa en las instituciones de educación superior (IES). De acuerdo con Knight (2021), dicho proceso trascendió la movilidad estudiantil para abarcar la integración de dimensiones interculturales e internacionales en los planes de estudio, los métodos de enseñanza y los resultados de aprendizaje. En el contexto latinoamericano, Gacel *et al*. (2024) señaló que la internacionalización en casa —modalidad que no requiere desplazamiento físico— emergió como una alternativa viable y equitativa para las comunidades académicas con limitaciones económicas o de seguridad. Bajo esta perspectiva, la Licenciatura en Ciencias de la Educación de la Universidad Autónoma de Sinaloa (UAS) adoptó la clase espejo como herramienta pedagógica para vincular su comunidad estudiantil con pares de Argentina.

Inicialmente, la UAS contaba con un programa de licenciatura orientado a la formación de profesionales capaces de analizar la realidad socioeducativa y generar conocimiento aplicado mediante la investigación

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.185 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 62 - 73

(UAS, 2023). Sin embargo, el ciclo escolar 2024-2025 estuvo marcado por situaciones de riesgo en el estado de Sinaloa que obligaron a suspender las actividades presenciales y a migrar a la virtualidad. Esta circunstancia evidenció brechas tecnológicas y socioeconómicas significativas entre los estudiantes, al tiempo que planteó el desafío de mantener la calidad formativa en condiciones de adversidad. Frente a este escenario, la comunidad académica de la Facultad de Ciencias de la Educación respondió con resiliencia tecnológica y pedagógica, entendida por autores como Espinosa *et al*., (2023) como la capacidad institucional de adaptarse positivamente ante la adversidad mediante ecosistemas digitales para sostener procesos de aprendizaje significativos.

Por su parte, la clase espejo —estrategia de colaboración sincrónica entre grupos de distintas instituciones y países que comparten temáticas comunes— fue identificada como un mecanismo pertinente para articular las necesidades formativas locales con una perspectiva comparada internacional.

En el ciclo 2025-2026, se diseñó e implementó una clase espejo entre la UAS y la Universidad Nacional de Catamarca (Argentina), en torno al método cualitativo de historia de vida y la entrevista como herramienta de investigación. Este esfuerzo conjunto se materializó gracias a la colaboración académica del profesorado y al soporte técnico provisto por el área de Tecnologías de la Información (TIC) de ambas instituciones. El objetivo del presente reporte técnico es describir la contribución del personal especialista en TIC en el diseño, configuración y soporte de la infraestructura tecnológica que sustentó dicha estrategia de clase espejo, con el propósito de sistematizar un modelo de implementación digital replicable en contextos de IES con restricciones de movilidad física.

2. DESARROLLO TÉCNICO

La Licenciatura en Ciencias de la Educación de la UAS contaba, previo al proyecto, con una experiencia acumulada de virtualización forzada durante el ciclo 2024-2025, periodo en que diversas situaciones de riesgo en Sinaloa obligaron a suspender las actividades presenciales e implementar modalidades en línea de forma abrupta. Ese antecedente reveló con nitidez dos brechas estructurales: una brecha tecnológica, expresada en el acceso desigual de los estudiantes a Internet y dispositivos de cómputo; y una brecha didáctica, manifestada en la ausencia de experiencias formativas con perspectiva internacional que pudieran desarrollarse sin movilidad física. Ambas brechas fueron el punto de partida que justificó el diseño del presente proyecto.

De acuerdo con Gacel *et al.* (2024), la internacionalización en casa representa la vía más equitativa para universidades públicas latinoamericanas que no disponen de fondos suficientes para sostener programas de movilidad estudiantil, por lo que su implementación era congruente tanto con las necesidades del contexto como las posibilidades reales de la institución.

En cuanto a los antecedentes académicos del modelo, la clase espejo —denominada en la literatura especializada como *virtual exchange* o *telecollaboration*— ha sido documentada desde la década de 1990; aunque su expansión significativa en América Latina ocurrió a partir de 2015, con la consolidación de plataformas digitales de acceso libre (O'Dowd, 2019). En el ámbito específico de las ciencias sociales, Machwate *et al*. (2021) documentaron que los proyectos de intercambio virtual centrados en métodos cualitativos de investigación generan aprendizajes más profundos que las modalidades exclusivamente teóricas, dado que colocan a los estudiantes ante la necesidad de negociar significados culturales en situaciones auténticas. Este hallazgo sustentó la decisión de articular la clase espejo alrededor del

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.185 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 63 - 73

método de historia de vida y a la entrevista como instrumento de investigación, dado que ambas herramientas requieren, por definición, el contacto con sujetos cuyas experiencias están culturalmente situadas. Asimismo, la convergencia curricular entre la materia Métodos Cualitativos de Investigación Educativa de la UAS y el Taller Metodológico: La historia de vida de la Universidad Nacional de Catamarca (UNCA) proporcionó una base temática sólida para el diseño de actividades compartidas sin necesidad de modificar los programas de estudio de ninguna de las dos instituciones.

Ante la necesidad de implementar una experiencia de internacionalización sin movilidad física, se analizaron cuatro modalidades alternativas: a) la movilidad virtual asíncrona, consistente en el intercambio de materiales grabados entre instituciones sin interacción en tiempo real; b) el proyecto COIL (*Collaborative Online International Learning*) estándar, que integra actividades sincrónicas y asincrónicas a lo largo de un semestre completo; c) la clase espejo en forma intensiva, con sesiones sincrónicas concentradas en pocas semanas; y d) el seminario internacional virtual, basado en conferencias impartidas por docentes invitados de otras instituciones, sin participación activa de los estudiantes. La Tabla 1 sintetiza los criterios de comparación considerados para la selección de la modalidad más adecuada al contexto de la UAS en el ciclo 2025-2026.

Tabla 1

Comparación de modalidades de internacionalización sin movilidad física evaluadas para el proyecto UAS-UNCA

Criterio	Movilidad virtual asíncrona	COIL semestral	Clase espejo intensiva (modalidad elegida)	Seminario virtual con invitados
Interacción sincrónica entre pares	No	Sí	Sí	Limitada
Requiere acuerdo interinstitucional formal	No	Sí	No	No
Adaptable a contexto de conectividad variable	Alta	Media	Alta	Alta
Desarrolla competencias interculturales activas	Baja	Alta	Alta	Baja
Integrable en el calendario académico existente	Alta	Media	Alta	Alta
Costo de implementación	Bajo	Medio	Bajo	Bajo

Nota: Elaborada con base en O'Dowd (2019), Rubin y Guth (2023) y Machwate *et al*. (2021).

La comparación evidenció que la clase espejo en formato intensivo era la modalidad que mejor conciliaba los criterios prioritarios para el contexto de la UAS: alto potencial para el desarrollo de competencias interculturales activas, adaptabilidad a la variabilidad de conectividad, bajo costo de implementación e integración directa en el calendario académico sin necesidad de acuerdos institucionales formales previos. La movilidad virtual asíncrona fue descartada porque, si bien presentaba ventajas en

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.185 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 64 - 73

términos de flexibilidad, no permitía el tipo de interacción en tiempo real necesaria para el desarrollo de competencias interculturales activas. En concordancia con Deardorff (2019), la construcción de la competencia intercultural requiere situaciones de contacto auténtico y negociación de sentido que difieren cualitativamente de la interacción diferida. El COIL semestral fue descartado por su demanda de coordinación interinstitucional formal, que no era viable en los tiempos del ciclo escolar 2025-2026. El seminario virtual con invitados fue desestimado por su naturaleza pasiva, que no propicia la participación voluntaria y activa del estudiante como sujeto del intercambio intercultural.

El proyecto se enmarcó en el enfoque metodológico del aprendizaje colaborativo internacional en línea (COIL), descrito por Rubin y Guth (2023) como una modalidad que integra equipos de estudiantes de distintos países mediante tecnologías digitales para el desarrollo conjunto de proyectos académicos. Dentro de ese marco, la clase espejo fue estructurada siguiendo un diseño instruccional en tres fases —diseño, implementación y evaluación— que se describe en detalle en la sección de desarrollo de este reporte técnico.

La etapa de diseño (mayo-agosto de 2025): las docentes responsables de ambas instituciones realizaron seis reuniones de planeación conjunta mediante videoconferencia para identificar los contenidos curriculares compartidos y definir la secuencia didáctica. El primer paso de esa fase fue un mapeo comparativo de los programas de las dos materias involucradas: la asignatura Métodos Cualitativos de Investigación Educativa, de segundo año de la Licenciatura en Ciencias de la Educación de la UAS, y el Taller Metodológico: Historia de vida, de cuarto año de las licenciaturas en Antropología Social y Cultural, Arqueología y Patrimonio Cultural de la UNCA. El mapeo reveló que ambas materias compartían el objetivo de desarrollar competencias para la aplicación del método biográfico-narrativo en la investigación social, con la entrevista en profundidad como técnica central. Esto fundamentaba la articulación temática del intercambio sin necesidad de modificar ningún programa de estudios.

2.1 SELECCIÓN DE LAS HERRAMIENTAS DIGITALES

La selección de las herramientas digitales se realizó con base en cuatro criterios establecidos por el equipo docente: 1) accesibilidad desde dispositivos móviles de gama media[1], dado que una proporción significativa de los estudiantes de la Licenciatura en Ciencias de la Educación de la UAS accede a Internet exclusivamente mediante teléfono celular, de acuerdo con las observaciones realizadas durante las sesiones en clase; 2) funcionamiento aceptable con anchos de banda reducidos, en respuesta a las condiciones de conectividad documentadas; 3) curva de aprendizaje mínima, pues el objetivo del proyecto era la formación en investigación cualitativa y no en el manejo de plataformas tecnológicas; y 4) gratuidad, en congruencia con la naturaleza de institución pública de ambas universidades. Con base en esos criterios, la elección no fue al azar; se basó en consideraciones de accesibilidad, consumo de datos y compatibilidad de *hardware*.

1 Las herramientas digitales (dispositivos móviles de gama media) se definen como una categoría de celulares que se encuentra entre gama baja (entry-level) y gama alta (high-end). Sus criterios están en los celulares que tienen mejores especificaciones que los de gama baja y no son costosos como los de gama alta; es decir, las personas que obtienen un dispositivo de gama media es porque se adapta a sus necesidades a un precio accesible (Halawa *et al*., 2024)

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.185 **Vol. 4, Núm. 3.** julio-septiembre 2026, págs. 65 - 73

El proceso de selección de la infraestructura digital se rigió por un análisis de viabilidad técnica y accesibilidad. Se evaluaron herramientas alternativas de alta demanda gráfica (como Miro o Teams); sin embargo, se optó por el ecosistema de Google y Padlet debido a su alta optimización para dispositivos de gama media, predominantes en la muestra estudiantil. Las pruebas funcionales, previas al ciclo 2025-2026, confirmaron que esta configuración minimizaba el uso de memoria RAM y mitigaba la exclusión tecnológica, permitiendo una transición fluida entre las sesiones síncronas y el trabajo colaborativo asíncrono.

Se seleccionó Google Classroom como sistema de gestión de aprendizaje principal, en tanto que el personal adscrito a ambas instituciones ya contaba con cuentas institucionales en la *suite* de Google Workspace for Education, lo que eliminó la necesidad de registro previo de los estudiantes y simplificó la integración del aula compartida entre países. Para la colaboración en línea, se adoptó Padlet.com, herramienta que permitía la creación de tableros colaborativos accesibles sin registro obligatorio y con interfaz intuitiva para usuarios sin experiencia previa. La sesión sincrónica se realizó mediante Google Meet, servicio integrado en el ecosistema ya disponible en ambas instituciones, que ofrecía funcionalidades suficientes —videollamada grupal, compartición de pantalla y salas pequeñas— sin costo adicional.

También fue importante atender la gestión de la diferencia horaria —cuatro horas entre México (UTC-7) y Argentina (UTC-3)—, la cual fue resuelta mediante la fijación de un horario de sesión sincrónica de 8 a 10 horas (hora del Pacífico), equivalente a 12:00 a 14:00 horas en Catamarca. Lo anterior garantizaba que los estudiantes argentinos pudieran participar dentro de su jornada universitaria regular. Ese acuerdo horario se formalizó en un calendario compartido vía Google Calendar y fue comunicado a los estudiantes de ambas instituciones con cuatro semanas de anticipación al inicio de las sesiones. En paralelo, se elaboró una carta de intención académica firmada por las dos docentes responsables, en la que se especificaban los objetivos, la metodología, los criterios de evaluación y los compromisos de cada institución; dicho documento funcionó como marco normativo del intercambio en ausencia de un convenio interinstitucional formal. Un elemento metodológico central del diseño fue la elaboración conjunta de las rúbricas de evaluación. Estas fueron construidas por las dos docentes a lo largo de tres sesiones de trabajo colaborativo, con el propósito de asegurar que los criterios de evaluación reflejaran tanto las competencias disciplinares específicas de cada programa como las competencias interculturales comunes al proyecto.

En el proceso, se diseñaron cuatro rúbricas: a) participación en la sesión sincrónica; b) calidad de las aportaciones en Padlet; c) diseño y aplicación de la guía de entrevistas; y d) reflexión intercultural final. Cada rúbrica establecía cuatro niveles de desempeño y fue socializada con los estudiantes al inicio del proyecto, en consonancia con las recomendaciones de Creswell (2023) sobre la transparencia evaluativa como factor de validez en los procesos formativos.

Para asegurar que los 31 estudiantes (México y Argentina) pudieran participar sin asimetrías técnicas, se realizó una fase de pilotaje o pruebas de estrés técnico previo en dispositivos con características estándar de gama media, como procesadores octa-core básicos, 4 GB-6 GB de memoria RAM y redes móviles 4G/Wi-Fi estándar.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.185 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 66 - 73

2.2 PRUEBAS TÉCNICAS

El equipo técnico decidió aplicar pruebas de concurrencia y multitarea, de carga de contenido multimedia y de usabilidad intercultural (sincronización transfronteriza). A continuación, se comentan los resultados obtenidos.

En la prueba de concurrencia y multitarea, se evaluó la ejecución de una sesión síncrona en Google Meet en primer plano mientras el estudiante minimizaba la app para escribir o consultar el muro de Padlet. En los dispositivos de gama media, la memoria RAM gestionó eficazmente ambas aplicaciones sin cerrar la videollamada de fondo, aunque se detectó un incremento térmico moderado en llamadas superiores a 60 minutos.

En lo que respecta a la prueba de carga de contenido multimedia, con la herramienta de entrevista, se aprobó la velocidad de carga del instrumento de investigación (guión de entrevista) y recursos complementarios en Classroom desde redes móviles. Los tiempos de respuesta obtenidos fueron óptimos (menores a 3 segundos para PDFs y documentos de texto), lo que garantizó que la limitación de *hardware* de gama media no fuera un obstáculo para el aprendizaje.

Se diseñó una prueba de usabilidad intercultural (sincronización transfronteriza) para evaluar la interacción en tiempo real en Padlet entre los nodos de Sinaloa (México) y Catamarca (Argentina), con el fin de medir el desfase y así evitar quedarse atrás en una sesión síncrona. La latencia en la actualización de los comentarios fue casi imperceptible (menos de 1.5 segundos), lo que validó que las plataformas elegidas eran inclusivas para la infraestructura técnica disponible por los alumnos.

En este punto, es relevante comentar lo que se consideró como "funcionamiento aceptable" en este proceso. Las condiciones de conectividad se documentaron a partir del reporte institucional de la Universidad Autónoma de Sinaloa, el cual detalla las limitaciones de infraestructura tecnológica y acceso a Internet en la región centro de la universidad. Además, se complementaron y validaron cualitativamente en tiempo real a través de observaciones directas en clase, es decir, la experiencia empírica del docente actúa como testigo de los fallos de red.

Previo al despliegue de las actividades en septiembre de 2025, se ejecutaron protocolos de pilotaje y control de calidad tecnológicos. Las pruebas no fueron improvisadas: se dividieron en dos entornos controlados orientados a evaluar la concurrencia y carga del sistema, así como la resiliencia de la infraestructura ante condiciones críticas de red.

En primer lugar, se evaluó el comportamiento del servidor ante peticiones simultáneas en el muro de Padlet, mediante simulaciones de carga masiva concurrente con inserción simultánea de contenido multimedia. El objetivo era identificar el punto de quiebre (*bottleneck*) del servidor y verificar que la plataforma no se congelara ni rechazara las peticiones de los estudiantes durante las actividades síncronas.

En segundo lugar, se evaluó el rendimiento de la red en el peor escenario posible, considerando variables de latencia y pérdida de paquete bajo un ancho de banda asimétrico, intencionalmente limitado a 1 Mbps. Con este fin, se llevaron a cabo pruebas de estrés de conectividad transcontinental entre los administradores TIC de la UAS en Sinaloa y los enlaces técnicos de la UNCA en Argentina, con el propósito de probar la viabilidad de la comunicación bajo condiciones críticas de red.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.185 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 67 - 73

El resultado esperado era determinar si el protocolo de contingencia asíncrona —desarrollado por Héctor Ignacio Castañeda García— reaccionaba correctamente cuando la latencia superaba los límites tolerables o cuando ocurría una pérdida severa de paquetes. Estas pruebas permitieron validar dicho protocolo, basado en el uso de formularios alternos en Forms y en un margen de 48 horas en Classroom, asegurando así la resiliencia tecnológica del ecosistema ante cortes o fluctuaciones en los servicios de Internet locales.

2.3 METODOLOGÍA

La explicación detallada de los procedimientos técnicos y didácticos aplicados para estructurar la solución digital se distribuyó en un cronograma ejecutado a lo largo del mes de septiembre de 2025, el cual comprendió una sesión de presentación inaugural y cuatro semanas de actividades consecutivas. Con el fin de fundamentar científicamente el proceso y asegurar que las decisiones tecnológicas no fuesen azarosas, la metodología se organizó de forma fluida en cuatro etapas secuenciales: diseño, configuración, instalación y pruebas.

En la fase inicial, se determinó la arquitectura lógica, la secuencia cronológica y los requerimientos instruccionales del entorno virtual. El diseño técnico se fundamentó en el modelo de Aprendizaje Colaborativo Internacional en Línea (COIL) (Rubin y Guth, 2023) y se articuló metodológicamente en una línea de tiempo estructurada en cuatro fases operativas:

- Fase inaugural (Semana 1): se planificó la introducción teórico-conceptual del método biográfico-narrativo de historia de vida y una actividad rompehielos síncrona orientada a contrastar las experiencias situadas de los estudiantes (contextos de riesgo de la UAS frente al trabajo de campo arqueológico rural de la UNCA).
- Fase de ideación colectiva (Semana 2): se diseñó una dinámica de lluvia de ideas interactiva para la identificación de líneas temáticas transversales (identidad, migración, educación en crisis), sustentada en los postulados de Rodríguez *et al.* (2024) sobre la internacionalización del currículo en casa.
- Fase de co-diseño y aplicación (Semana 3): Se estructuró el trabajo asíncrono en duplas internacionales (México-Argentina) para la creación de guías de entrevista cualitativa y su posterior aplicación local, bajo los principios éticos de consentimiento informado y confidencialidad formulados por Flick (2018).
- Fase de evaluación y cierre (Semana 4): Se diseñó un espacio de socialización síncrona para la presentación de hallazgos comparativos, promoviendo la práctica reflexiva (Espinosa *et al*., 2023).

Asimismo, se planificó la sistematización del proceso a través de la triangulación de tres fuentes analíticas: rúbricas semanales de desempeño, notas de observación docente e instrumentos de autoevaluación estudiantil, siguiendo las recomendaciones metodológicas de Creswell (2023) para incrementar la validez de los procesos cualitativos. Este diseño analítico previo a la evaluación contempló dimensiones como la reflexión intercultural, la responsabilidad ética y el rigor metodológico. Para mitigar la variable crítica de conectividad (estimada en un 30% de vulnerabilidad técnica en la UAS debido a la situación regional), el diseño incorporó proactivamente las pautas del Diseño Universal para el Aprendizaje (CAST, 2024), proyectando rutas alternativas síncronas y asíncronas de participación que garantizaran la inclusión de la totalidad de los 27 estudiantes.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.185 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 68 - 73

La etapa de configuración consistió en la parametrización lógica, asignación de permisos y estructuración de la *suite* de *software* seleccionada para dar soporte a las fases diseñadas, que incluyó el uso de Google Classroom, Google Meet, Padlet, Google Docs y Google Forms. A continuación, se comentan los parámetros empleados en cada herramienta digital.

Para Google Classroom, se configuró este LMS principal mediante la creación de aulas virtuales y subcarpetas organizadas por semanas. Se programaron los módulos para la publicación automatizada de las grabaciones de video en un lapso no mayor a dos horas posteriores a las sesiones en vivo y se fijó técnicamente un margen de tolerancia de 48 horas para la entrega diferida de tareas.

Se parametrizó el sistema de videoconferencia de Google Meet activando controles de encriptación, salas de espera y la función de grabación nativa en la nube. Se configuró la herramienta de subsalas de discusión (*breakout rooms*), requiriendo una asignación manual por parte del responsable técnico (Héctor Ignacio Castañeda García) para conformar de manera exacta los grupos mixtos binacionales.

Se configuró un muro digital interactivo en Padlet bajo el formato de "Lienzo", habilitando los permisos lógicos específicos para que los usuarios externos realizaran aportaciones y comentarios en tiempo real, restringiendo los privilegios de edición o borrador de publicaciones ajenas para salvaguardar la integridad de las más de 40 propuestas temáticas previstas.

En lo que respecta a Google Docs y Google Forms, se crearon plantillas con permisos de edición compartida por el co-diseño asíncrono de las guías de entrevista de las duplas, y se programaron formularios estructurados en Forms como mecanismo técnico alternativo y equivalente para la recolección de aportaciones didácticas de aquellos alumnos que no lograsen ingresar a las subsalas síncronas.

En la etapa de instalación, al implementarse herramientas basadas en el modelo de *Software* como Servicio (SaaS), los procedimientos se centraron en el aprovisionamiento de identidades digitales y la verificación de compatibilidad en *hardware*. Se gestionó el alta, matriculación y vinculación masiva de los correos institucionales de los 27 estudiantes de la UAS y los cuatro participantes de la UNCA dentro de las plataformas configuradas. Asimismo, se distribuyó un manual técnico con las especificaciones de optimización para la instalación de las aplicaciones móviles ligeras de Google Meet, Classroom y Docs en dispositivos con sistemas operativos Android e iOS. Esto garantizó que el *hardware* de gama media y baja de los alumnos fuera capaz de procesar los flujos de trabajo sin congelamientos del sistema.

3. RESULTADOS

Los hallazgos técnicos y pedagógicos derivados de la implementación del ecosistema digital diseñado son altamente resilientes. Esto demuestra que el éxito del proyecto no depende de una conectividad óptima o de alta velocidad, ya que los mecanismos asíncronos absorben eficazmente las fallas de infraestructura local, garantizando que el estudiante nunca pierda la continuidad de su aprendizaje. Se organizaron y analizaron a través de tres dimensiones fundamentales, permitiendo evaluar el grado de resolución de la problemática inicial y la estabilidad de la infraestructura diseñada por el personal especialista en TIC.

Los productos técnicos como las simulaciones validaron que los servidores soportaran la carga masiva. En la prueba transcontinental a 1 Mbps, se registraron fluctuaciones y pérdida de paquetes esperados, lo que activó con éxito el protocolo de contingencia (formularios alternos en Forms y el margen de 48 horas en Classroom). Se demostró que la arquitectura de red e instruccional diseñada garantizó la continuidad

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.185 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 69 - 73

académica del 100% de los 27 estudiantes de la UAS, superando la interrupción educativa provocada por las situaciones de riesgo locales.

En la dimensión procedimental, el soporte asíncrono provisto a través de Google Docs permitió el co-diseño exitoso de las guías de entrevista binacionales. El análisis de las rúbricas digitales automatizadas reveló que el 85% de los participantes interactuó de forma óptima con la plataforma, identificando y corrigiendo sesgos metodológicos potenciales antes de la fase de campo. Este resultado se sustenta en los criterios de evaluación aplicados a través de dos instrumentos complementarios: la rúbrica de participación en entrevistas y la rúbrica para evaluar mapa colaborativo de problemáticas de investigación social y educativa, ambas estructuras en una escala de cuatro niveles de desempeño (0 a 4 puntos, de Insuficiente a Sobresaliente).

En particular, el criterio de aplicación de conceptos teóricos y metodológicos resultó determinante para constatar la capacidad de los estudiantes de vincular los fundamentos teóricos de la entrevista con su práctica, distinguiendo entre quienes lograban aplicar los conceptos de forma crítica y creativa (nivel sobresaliente) y quienes incurrían en imprecisiones o no lograban vincularlos claramente con la práctica (niveles satisfactorios o en desarrollo). De manera complementaria, el criterio de preparación y rigor académico permitió valorar si los estudiantes habían realizado las lecturas previas y reflexionado críticamente sobre los temas a tratar, elemento que explica de forma directa la identificación temprana de sesgos metodológicos antes del trabajo de campo.

En el caso del mapa colaborativo, los criterios de profundidad y argumentación de las problemáticas, de articulación y conexión entre ellas, aportaron evidencia adicional sobre la calidad del análisis crítico de los estudiantes, al evaluar tanto la solidez de los argumentos como la coherencia con que relacionaban las distintas problemáticas identificadas. El 85% de los participantes se ubicó en los niveles de desempeño sobresaliente y notable en el conjunto de estos criterios, lo que constituye evidencia de una interacción óptima con la plataforma y una apropiación reflexiva de las herramientas metodológicas.

Estos datos confirmaron una apropiación reflexiva de las herramientas, validando los postulados de González (2018), quien afirmó que las mediaciones tecnológicas rigurosas son fundamentales para capturar la complejidad de las experiencias humanas en sus entornos específicos sin perder rigor metodológico.

El entorno síncrono y asíncrono configurado facilitó la apertura a la diversidad y la transferencia conceptual analítica a distancia. Los registros de observación técnica y docente documentaron que las subsalas programadas en la plataforma de videoconferencia y el lienzo dinámico de Padlet, que acumuló más de 40 propuestas temáticas compartidas, permitieron a los estudiantes de la UAS asociar de forma sistémica las situaciones de riesgo en Sinaloa con procesos similares analizados por sus pares de la UNCA en comunidades rurales de Argentina. De este modo, se establecieron categorías de análisis complejas como la vulnerabilidad institucional y la resiliencia comunitaria. Este nivel de transferencia conceptual interactiva es catalogado por Deardorff (2019) como un indicador de competencia intercultural avanzada, el cual, analizado frente a los antecedentes, demostró que la infraestructura tecnológica expandió el horizonte formativo de un problema local hacia una comprensión global comparada.

Respecto a la viabilidad institucional del proyecto, los indicadores de éxito confirmaron que el modelo digital es altamente sostenible y replicable. La infraestructura operó exclusivamente mediante *software* de acceso libre y licencias corporativas preexistentes (Google Workspace), anulando los costos financieros asociados a la movilidad internacional física. Asimismo, la inversión de tiempo técnico y pedagógico para

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.185 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 70 - 73

la coordinación transcontinental se optimizó a un indicador de eficiencia de aproximadamente 12 horas de planificación conjunta distribuidas a lo largo de cuatro meses.

Al realizar un análisis comparativo entre estos hallazgos y los referentes teóricos documentados por Ramírez *et al.* (2020) en proyectos COIL de universidades europeas y latinoamericanas, se constató que el modelo tecnológico desarrollado en las UAS es plenamente consistente con las mejores prácticas internacionales de educación digital. No obstante, el presente proyecto aportó un valor incremental a la literatura especializada al demostrar que la arquitectura de red e instruccional descrita a lo largo de este documento —caracterizada por operar mediante plataformas SaaS independientes entre sí (Google Meet, Classroom, Docs, Padlet y Forms), sin depender de un servidor central único, y articuladas mediante el protocolo de contingencia asíncrona desarrollado por Castañeda García— constituye en realidad una arquitectura digital descentralizada, y que este tipo de arquitecturas poseen la resiliencia necesaria para operar con éxito en contextos de riesgo e inestabilidad civil generalizada.

A pesar del cumplimiento de los objetivos, la fase de explotación del sistema expuso vulnerabilidades críticas en la infraestructura de conectividad de los usuarios finales. El principal fallo técnico detectado se manifestó durante las sesiones síncronas semanales, donde un 30% de la muestra estudiantil de la UAS experimentó caídas intermitentes de la señal, congelamiento de pantallas y degradación del flujo de audio debido a las restricciones de ancho de banda y al agotamiento de los planes de datos móviles en las zonas de riesgo.

Si bien esta vulnerabilidad fue mitigada de forma paliativa mediante la ruta de contingencia asíncrona (grabaciones en Classroom dentro de un margen de dos horas y tareas alternas en Google Forms), el análisis del comportamiento del sistema evidenció la necesidad de introducir mejoras en el procesamiento técnico para futuros ciclos. Como solución definitiva para optimizar la replicabilidad del modelo en condiciones de conectividad degradada, se identificó la urgencia de sustituir las videoconferencias extensas por microcápsulas de video pre-grabadas con codificación de baja tasa de bits, así como migrar las dinámicas de debate síncrono hacia foros de discusión asíncronos, optimizados para redes móviles 2G/3G.

4. CONCLUSIONES

La suspensión imprevista de las actividades académicas presenciales, derivada de las situaciones de riesgo en Sinaloa, planteó una problemática de interrupción formativa crítica, la cual se vio agravada por una marcada brecha digital y heterogeneidad en el acceso a Internet entre los estudiantes dispersos geográficamente. Frente a este escenario de vulnerabilidad, el objetivo general de este reporte técnico, centrado en describir y evaluar la contribución del personal especialista en TIC mediante el diseño, configuración y soporte de una infraestructura tecnológica ligera, se alcanzó en un grado total de cumplimiento. La arquitectura digital implementada demostró que es posible sostener la calidad del proceso formativo en condiciones adversas y que la internacionalización del currículo no requiere de movilidad física transcontinental para generar aprendizajes significativos y transferencia conceptual avanzada en el método de historia de vida.

Los hallazgos demostraron que la *suite* tecnológica seleccionada (Google Classroom, Padlet y sistemas de videoconferencia) resultó técnicamente suficiente y eficiente para propiciar el desarrollo de competencias investigativas e interculturales. La gestión generalizada del entorno virtual, mediante la asignación manual

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.185 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 71 - 73

de subsalas binacionales y la carga de rúbricas automatizadas, permitió que los estudiantes transitaran de la teoría cualitativa a la práctica reflexiva en entornos reales. Asimismo, el aprovisionamiento oportuno de rutas tecnológicas asíncronas y alternativas mitigó de forma efectiva las restricciones operativas del entorno, logrando un indicador de éxito del 100% de inclusión y participación efectiva de la matrícula (27 estudiantes), lo que confirmó la sostenibilidad financiera y operativa de este modelo para instituciones de educación superior con presupuestos limitados.

Para resolver de manera definitiva los riesgos y fallos de conectividad identificados en el 30% de los usuarios, debido a la degradación de las redes móviles en zonas de riesgo, se formularon recomendaciones técnicas orientadas a fortalecer la resiliencia preventiva del sistema. Se sugirió diseñar protocolos instruccionales híbridos que disminuyan la dependencia de la presencialidad virtual, lo que implica sustituir las videoconferencias extensas por microcápsulas de video pregrabadas con codificación de baja tasa de bits y migrar las dinámicas de debate en tiempo real hacia foros de discusión asíncronos optimizados para redes móviles de baja velocidad. El nivel de esfuerzo general requerido para la implementación de estas medidas preventivas y correctivas se estimó como bajo, debido a que no demanda inversiones en *hardware* especializados, sino la optimización de las funciones lógicas ya instaladas en el sistema de gestión del aprendizaje institucional y la capacitación docente en producción multimedia básica.

Declaración de contribución de autoría

Flérida Moreno Alcaraz: Especialista en investigación educativa e internacionalización del currículo. Lideró el diseño del proyecto de clase espejo desde su conceptualización hasta su implementación. Por medio de la Dirección General de Vinculación y Relaciones Internacionales de la UAS, se estableció el contacto institucional con la Universidad Nacional de Catamarca y se coordinaron las reuniones de planeación conjunta realizadas entre mayo y agosto de 2025. Diseñó la secuencia didáctica centrada en el método de historia de vida y la entrevista como herramienta de investigación; elaboró las rúbricas de evaluación y supervisó su aplicación durante las sesiones de septiembre. Asimismo, sistematizó los registros de observación y redactó el presente reporte técnico, asegurando la congruencia entre los objetivos del proyecto, la metodología implementada y los resultados.

Héctor Ignacio Castañeda García: Especialista en tecnologías de la información aplicadas a la educación. Se encargó de la selección, configuración y administración de las plataformas digitales empleadas en el proyecto. Configuró el entorno de Google Classroom para el grupo de la UAS, diseñó los tableros colaborativos en Padlet y gestionó los accesos de los estudiantes a ambas herramientas. Ante las dificultades de conectividad reportadas por aproximadamente el 30% del grupo, desarrolló versiones asíncronas de las actividades sincrónicas, garantizando la inclusión de todos los participantes. También coordinó las pruebas técnicas previas a cada sesión para verificar la estabilidad de la conexión con Catamarca y elaboró el protocolo de contingencia tecnológica utilizado durante la implementación.

Juan Carlos Puga Villarreal: Especialista en metodología de la investigación cualitativa. Participó en la definición del fundamento epistemológico y metodológico del proyecto. Seleccionó y adaptó los materiales de lectura sobre historia de vida utilizados en ambas instituciones, asegurando su pertinencia para los niveles de formación de los estudiantes involucrados. Diseñó las guías de práctica de entrevista y los instrumentos de reflexión ética empleados durante las sesiones. Además, apoyó el proceso de análisis de los resultados cualitativos obtenidos a partir de las rúbricas y los registros de observación, así como contribuyó a la redacción de las secciones metodológicas y de resultados del presente reporte técnico.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.185 **Vol. 4, Núm. 3.** julio-septiembre 2026, págs. 72 - 73

Karla Marisol Aguirre Sánchez: Especialista en evaluación educativa y mediación pedagógica en entornos virtuales. Fue responsable del acompañamiento psicopedagógico y del aseguramiento de la permanencia de los 27 estudiantes de la UAS, atendiendo de manera directa las necesidades críticas derivadas de las situaciones de riesgo regionales. Se encargó del diseño, aplicación y procesamiento técnico del instrumento de autoevaluación estudiantil administrado durante la cuarta semana, aportando los insumos cuantitativos y cualitativos esenciales para realizar la triangulación metodológica de los resultados del proceso formativo. Su intervención facilitó la resolución de dudas procedimentales respecto al codiseño de las guías de entrevista cualitativa en la nube, lo que garantizó la retención total de la matrícula y la viabilidad operativa de la secuencia didáctica transcontinental.

REFERENCIAS

CAST (2024). *CAST Universal Design for Learning Guidelines* version 3.0. https://udlguidelines.cast.org

Creswell, J. W. y Creswell, J. D. (2023). *Research design: Qualitative, quantitative, and mixed methods approaches* (5.ª ed.). SAGE. https://repositorio.ciem.ucr.ac.cr/bitstream/123456789/514/1/Research%20Design.pdf

Deardorff, D. K. (2019). *Manual for developing intercultural competencies: Story circles*. Routledge. https://doi.org/10.4324/9780429244612

Espinosa Izquierdo, J., Villamar Bravo, J., Quijije Acosta, K., y Mesa Vázquez, J. (2023). Ecosistemas digitales de aprendizaje y educación 4.0 una aproximación a las pedagogías emergentes. *Polo del Conocimiento, 8*(9), 134-158. https://doi.org/10.23857/pc.v8i9.6005

Flick, U. (2018). *An introduction to qualitative research* (4.ª ed.). Morata.

Gacel-Ávila, J., Villalón-de-la-Isla, E. M., y Vázquez-Niño, G. (2024). La internacionalización de la educación superior en América Latina: una visión comparada. *Revista Educación Superior Y Sociedad (ESS)*, *36*(1), 310-334. https://doi.org/10.54674/ess.v36i1.912

González Giraldo, O. E. (2018). La narrativa biográfica como una prometedora experiencia (auto)formativa en el trayecto de formación docente. *Latinoamericana de Estudios Educativos, 15*(1), 68-90. https://doi.org/10.17151/rlee.2019.15.1.5

Halawa, A. N., Sihite, H. H., & Syahrizal, M. (2024). Sistem Pendukung Keputusan Pemilihan Smartphone Kelas Midrange Menerapkan Metode Multi-Attibute Utility Theory (MAUT). *Journal of Computing and Informatics Research, 3(2)*, 173-181. https://doi.org/10.47065/comforch.v3i2.1201

Knight, J. (2021). Higher Education Internationalization: Concepts, Rationales, and Frameworks. *Revista REDALINT. Universidad, Internacionalización E Integración Regional*, *1*(1), 65-88. https://revele.uncoma.edu.ar/index.php/redalint/article/view/3090

Machwate, S., Bendaoud, R., Henze, J., Berrada, K., y Burgos, D. (2021). Virtual Exchange to Develop Cultural, Language, and Digital Competencies. *Sustainability*, *13*(11), 5926. https://doi.org/10.3390/su13115926

O'Dowd, R., y O'Rourke, B. (2019). New developments in virtual exchange in foreign language education. *Language Learning & Technology, 23*(3), 1–7. https://doi.org/10.64152/10125/44690

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.185 **Vol. 4, Núm. 3.** julio-septiembre 2026, págs. 73 - 73

Ramírez-Marín, F., Núñez-Figueroa, L. C., y Blair, N. (2020). Collaborative Online International Learning: Language and Cross-Cultural Experiences of University Students. *Matices en Lenguas Extranjeras, 14*(1), 118-162. https://doi.org/10.15446/male.v14n1.92144

Rodríguez Loaiza, D. C., Mira Fernández, C., Saldarriaga Molina, J. C., y Areiza Pérez, L. M. (2024). La internacionalización en casa como estrategia de fortalecimiento del currículo en la Facultad de Ingeniería de la Universidad de Antioquia. *En Encuentro Internacional de Educación en Ingeniería*. https://doi.org/10.26507/paper.4157

Rubin, J. y Guth, S. (Eds.). (2023). *The guide to COIL virtual exchange: Implementing, growing, and sustaining collaborative online international learning*. Routledge. https://doi.org/10.4324/9781003447832

ANEXO A. SITIOS WEB (PUBLICACIÓN DE LA CLASE ESPEJO)

https://dcs.uas.edu.mx/noticias/12124/estudiantes-y-docentes-hacen-clase-espejo-entre-la-facultad-de-ciencias-de-la-educacion-de-la-uas-y-la-universidad-nacional-de-catamarca-argentina

https://paherportal.com/estudiantes-y-docentes-hacen-clase-espejo-entre-la-facultad-de-ciencias-de-la-educacion-de-la-uas-y-la-universidad-nacional-de-catamarca-argentina/

https://www.facebook.com/share/18qsEugD1w/

https://www.facebook.com/share/p/14Wgv17QfXu/

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.190 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 74 - 82

Uso de la herramienta OPNsense para la gestión de seguridad unificada de un centro de datos

Use of the OPNsense tool for unified threat management in a data center

Pedro Temachti Pérez Santillán
Dirección General de Cómputo y de Tecnologías de Información y Comunicación
Universidad Nacional Autónoma de México
ptsantillan@unam.mx
ORCID: 0009-0000-2626-9073

Información del reporte:

Para citar este reporte técnico:

Pérez Santillán, P. T. (2026). Uso de la herramienta OPNsense para la gestión de seguridad unificada de un centro de datos. *Cuadernos Técnicos Universitarios de la DGTIC, 4*(3), páginas (74 - 82). https://doi.org/10.22201/dgtic.30618096e.2026.4.3.190

Resumen

La seguridad de la infraestructura tecnológica en los centros de datos requiere esquemas robustos que controlen el acceso y protejan los servicios críticos contra amenazas externas. Ante la exposición directa de servidores de administración con direcciones públicas y la falta de conexiones remotas seguras en una organización, se planteó la necesidad de implementar una solución de gestión unificada de amenazas que permitiera centralizar el tráfico de red, ocultar los equipos internos y habilitar accesos cifrados para el personal técnico.

La metodología se desarrolló a través de un enfoque que abarcó el diseño lógico de una arquitectura segmentada y la posterior instalación de una plataforma de código abierto sobre un entorno virtualizado. Se configuraron reglas estrictas para aislar los segmentos de operación, se habilitó la traducción de direcciones uno a uno para gestionar las salidas a la red externa y se crearon servidores virtuales independientes de red privada para el acceso remoto. Finalmente, se ejecutaron pruebas de conectividad y auditoría de eventos para validar la efectividad de las políticas establecidas antes de la puesta en marcha definitiva. Una limitación del estudio fue la imposibilidad de validar empíricamente el bloqueo geográfico, debido a la falta de acceso a direcciones IP de origen extranjero durante la etapa de pruebas.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.190 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 75 - 82

Los resultados demostraron un ocultamiento de los equipos internos y un control del tráfico saliente. Los intentos de acceso no autorizado fueron mitigados de manera automática y registrados para su posterior auditoría, mientras que las conexiones de red privada funcionaron de forma estable en diversos sistemas operativos. Se concluyó que la solución adoptada resolvió las vulnerabilidades iniciales al reducir la superficie de ataque, recomendando para el futuro la integración de sistemas de inspección profunda y la centralización de registros.

Palabras clave: OPNsense, OpenVPN, *firewall*, VPN, ciberseguridad.

Abstract

The security of technological infrastructure in data centers requires robust frameworks that control access and protect critical services against external threats. Given the direct exposure of administration servers with public addresses and the lack of secure remote connections in an organization, the need arose to implement a unified threat management solution to centralize network traffic, hide internal equipment, and enable encrypted access for technical personnel.

The methodology was developed through a structured approach that encompassed the logical design of a segmented architecture and the subsequent installation of an open-source platform within a virtualized environment. Strict filtering rules were configured to isolate operational segments, one-to-one address translation was enabled to regulate outbound traffic to the external network, and independent virtual private network servers were created for remote access. Finally, connectivity tests and event auditing were executed to validate the effectiveness of the established policies before final deployment. One limitation of this study was the inability to empirically validate the geographic blocking feature, due to the lack of access to foreign-origin IP addresses during the testing phase.

The results demonstrated a total concealment of internal servers and precise control of outbound traffic. Unauthorized access attempts via remote administration protocols were automatically mitigated and logged for subsequent auditing, while the private network connections functioned stably across various operating systems. It was concluded that the adopted solution resolved the initial vulnerabilities by reducing the attack surface, recommending for the future the integration of deep inspection systems and the centralization of logs

Keywords: *OPNsense, OpenVPN, firewall, VPN, cibersecurity.*

1. INTRODUCCIÓN

La seguridad de la información constituye uno de los pilares fundamentales en la operación de cualquier infraestructura tecnológica moderna. La protección de los sistemas que soportan servicios críticos dejó de ser opcional para convertirse en una prioridad. Las oficinas del centro de datos de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) albergan la administración de servicios que requieren un esquema robusto de control de acceso, segmentación de red y conectividad remota segura.

La gestión unificada de amenazas, conocida por sus siglas en inglés como UTM (*Unified Threat Management*), surgió porque la diversidad de herramientas de seguridad tradicionales requería administrar de forma separada múltiples sistemas como *firewalls*, sistemas de detección de intrusos, segmentación de redes,

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.190 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 76 - 82

entre otras. Según la página de Zscaler (2024), los sistemas UTM integran estas capacidades en una sola plataforma de administración centralizada, simplificando la operación y reduciendo la superficie de ataque. En este mismo sentido, investigaciones recientes publicadas en el *International Journal of Research Publication and Reviews* (Purvant y Sowmya, 2024) señalaron que el análisis de tráfico y la automatización en los *firewalls* se están convirtiendo en herramientas de seguridad indispensables. Paralelamente, el trabajo de Majid (2025) demostró que herramientas como OPNsense son capaces de prevenir accesos no autorizados, registrar eventos de seguridad y bloquear ataques en entornos de red, confirmando su eficacia como plataforma de seguridad.

Ante la ausencia de un mecanismo centralizado para regular el tráfico saliente de las máquinas de administración del centro de datos de la DGTIC, se identificó la necesidad de implementar una solución que proporcionara salida controlada mediante direcciones IP privadas, así como acceso remoto cifrado para el personal técnico.

El objetivo del trabajo es implementar la herramienta OPNsense como solución de gestión unificada de amenazas (UTM) para regular el tráfico saliente de las máquinas de administración del centro de datos de la DGTIC, mediante el control de acceso basado en direcciones IP privadas y el establecimiento de un mecanismo de conectividad remota cifrada para el personal técnico.

2. DESARROLLO TÉCNICO

Selección de la solución

Previo a la implementación descrita en este documento, las máquinas de administración de servicios del centro de datos operaban con una asignación directa de direcciones IP públicas. Adicionalmente, no existía un mecanismo formal para que el personal técnico se conectara de forma cifrada desde ubicaciones remotas.

Ante esta situación, se evaluaron distintas alternativas. A pesar de que las soluciones comerciales, como FortiGate y Cisco ASA, ofrecen capacidades avanzadas de inspección de tráfico, sus costos elevados de licenciamiento pueden representar una limitación. Por tal motivo, pfSense constituyó una opción viable al ser una plataforma de código abierto.

Huda *et al.* (2023) realizaron una comparativa de calidad de servicio entre pfSense y OPNsense y concluyeron que ambas plataformas ofrecen un desempeño equiparable. OPNsense fue seleccionado gracias a que ofrece una interfaz web más moderna, ciclos de actualización de seguridad más frecuentes y complementos que pueden aumentar sus capacidades y alcance, como la integración con sistemas de detección de intrusiones como Suricata y soporte nativo para múltiples protocolos VPN. En investigaciones recientes, Tymoshchuk *et al.* (2024) confirmaron la eficiencia de OPNsense para entornos de producción y realizaron pruebas que demostraron su capacidad para detectar y prevenir accesos no autorizados, evidenciando su efectividad en aplicaciones reales de seguridad de redes.

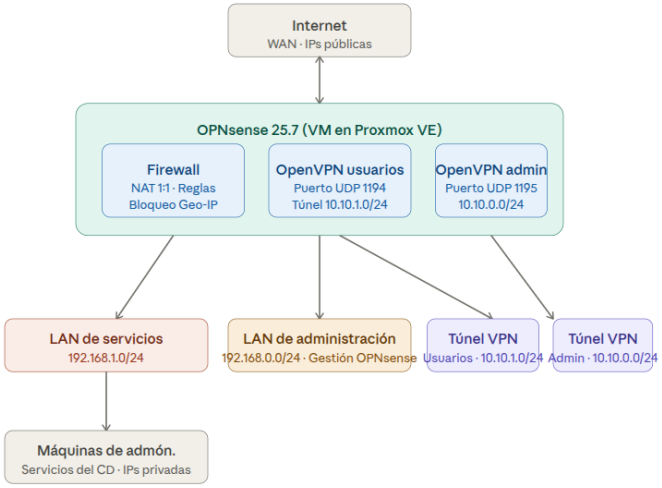
https://doi.org/10.22201/dgtic.30618096e.2026.4.3.190 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 77 - 82

Arquitectura de red diseñada

La arquitectura implementada se fundamentó en la segmentación de red mediante el uso de OPNsense como punto centralizado de control del tráfico. Se definieron las siguientes interfaces lógicas dentro del sistema, representadas en el diagrama de la Figura 1.

Figura 1

Diagrama de la arquitectura de red implementada con OPNsense



WAN: Interfaz de cara a Internet, a través de la cual OPNsense recibe el tráfico externo y gestiona la salida de las máquinas internas hacia la red pública.

LAN de servicios: Red privada a la que se conectan las máquinas de administración del centro de datos. Las máquinas de esta red dejan de tener exposición directa en Internet y obtienen salida controlada a través de OPNsense.

LAN de administración: Segmento exclusivo para la gestión del propio OPNsense, separado lógicamente de la red de servicios para impedir que usuarios puedan acceder a la consola de administración del *firewall*.

Interfaces VPN: Interfaces virtuales creadas por OpenVPN para la conexión remota cifrada del personal técnico.

2.1 METODOLOGÍA

La implementación se llevó a cabo en cinco etapas: diseño, instalación, configuración, pruebas y puesta en producción.

Etapa 1 – Diseño

La etapa de diseño consistió en la definición de la arquitectura lógica de red que daría soporte a los servicios de *firewall* y VPN. A partir del análisis de los requerimientos de seguridad del centro de datos, se

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.190 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 78 - 82

elaboró un diagrama conceptual de red en el que se identificaron los segmentos de red necesarios, los flujos de tráfico permitidos y los filtros de seguridad entre ellos.

Se definió el esquema de direccionamiento IP privado como se muestra en la Tabla 1.

Tabla 1

Esquema de direccionamiento IP de la infraestructura implementada

Segmento	Red	Propósito
LAN de servicios	192.168.1.0/24	Máquinas de administración de servicios
LAN de administración	192.168.0.0/24	Gestión de OPNsense
Túnel VPN de usuarios	10.10.1.0/24	Acceso remoto cifrado para usuarios
Túnel VPN de administración	0.10.0.0/24	Acceso remoto cifrado para administradores
WAN	Direcciones IP públicas	Salida a Internet de los usuarios

La separación de los segmentos de red cubre la necesidad de mantener aislados entre sí a los usuarios de la administración para limitar el impacto de un eventual incidente de seguridad. Asimismo, la definición de dos instancias VPN independientes —una para usuarios y otra para administradores— siguió la premisa de que cada usuario debe tener acceso únicamente a los recursos estrictamente necesarios para el cumplimiento de sus funciones.

Etapa 2 – Instalación

Se creó una máquina virtual en Proxmox VE con los recursos necesarios para ejecutar OPNsense, asignando interfaces de red virtuales correspondientes a los segmentos WAN y LAN. Se descargó la imagen ISO oficial de OPNsense y se procedió a la instalación del sistema. Una vez completada, se accedió a la interfaz web de administración para iniciar la configuración del sistema.

Etapa 3 – Configuración

En esta etapa, se configuraron las principales funcionalidades que se requerían del sistema.

- *Configuración de NAT 1 a 1.* Se implementó traducción de direcciones de red uno a uno (*one-to-one NAT*) para que cada máquina mantuviera su dirección IP pública al salir hacia Internet, conservando los accesos asociados a su dirección IP, pero eliminando la exposición directa de los equipos Internet.
- *Reglas de firewall y bloqueo regional.* Se definieron reglas de filtrado para separar el tráfico de la LAN de servicios respecto de la LAN de administración, impidiendo que, desde la red de servicios, se pudiera acceder a la interfaz de gestión de OPNsense. Adicionalmente, se configuró el bloqueo regional de IPs utilizando las listas de geolocalización MaxMind GeoLite2 disponibles para OPNsense, de modo que todo el tráfico entrante proveniente de rangos de direcciones IP fuera del territorio mexicano se descartara en la interfaz WAN. El bloqueo geográfico constituye una de las múltiples herramientas de una estrategia de ciberseguridad integral cuando opera como parte de las reglas del *firewall*, brindando una mayor protección (Miller, 2025).
- *Configuración de OpenVPN.* Se desplegó un servidor OpenVPN dentro de OPNsense para proveer acceso remoto cifrado al personal del centro de datos. Se generó una infraestructura de llave pública dentro de OPNsense para emitir certificados de servidor y de cliente, garantizando que únicamente

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.190 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 79 - 82

usuarios autenticados mediante un certificado válido y contraseña pudieran establecer una conexión.

Etapa 4 – Pruebas

Las pruebas funcionales se diseñaron para validar cada uno de los componentes configurados. Éstas se describen con detalle en la sección de Resultados.

3. RESULTADOS

La implementación de OPNsense como plataforma de seguridad unificada en el centro de datos produjo resultados satisfactorios en todos los puntos dentro del alcance del proyecto. A continuación, se presentan los resultados y su respectivo análisis.

Segmentación de red y ocultamiento de equipos

Una vez desplegada la arquitectura de red, se migraron todas las máquinas de administración de servicios a un esquema de direccionamiento con direcciones IP privadas. Como resultado, ninguno de los equipos quedó "visible" desde Internet, dado que sus direcciones privadas no son enrutables. Esto resolvió el problema de tener expuestos los equipos a actores externos.

NAT uno a uno y salida controlada a Internet

Se verificó desde cada equipo, mediante una consulta a la página web *ifconfig.me*, que cada uno salía a Internet con su dirección IP pública correspondiente. Esto validó que la traducción de direcciones operaba correctamente.

La consola de OPNsense proporcionó visibilidad en tiempo real del tráfico que atravesaba el *firewall*, permitiendo identificar los flujos activos y los eventos de seguridad registrados. Esta capacidad de observabilidad representó una gran mejora de seguridad.

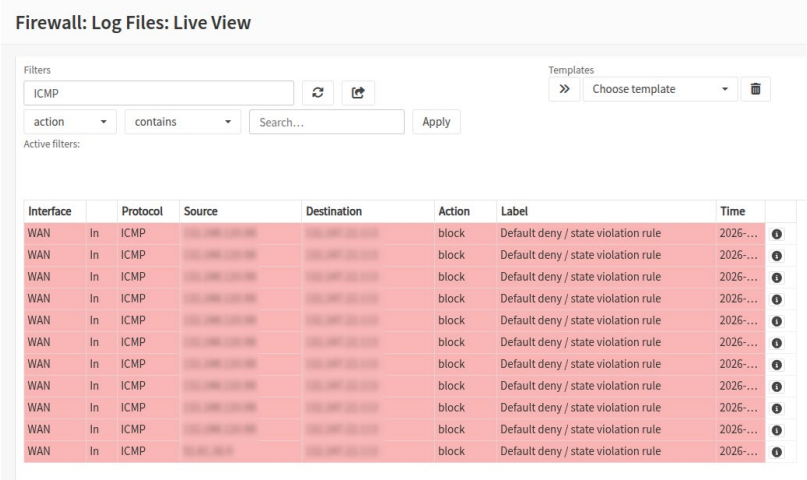
Reglas de *firewall* y bloqueo de accesos externos

Los *gateways* de los segmentos de red configurados dejaron de responder a solicitudes ICMP. Como se puede ver en la Figura 2, se confirmó que las reglas de filtrado aplicadas sobre las interfaces correspondientes operaron conforme a lo diseñado.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.190 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 80 - 82

Figura 2

Verificación de que las solicitudes ICMP son bloqueadas y registradas en los logs



Los intentos de conexión SSH realizados desde direcciones IP externas fueron bloqueados y registrados automáticamente en el log del *firewall*, lo que demostró tanto la efectividad del filtrado de tráfico entrante como la capacidad de auditoría de la plataforma. Este comportamiento es consistente con lo reportado por Majid (2025), quien documentó que OPNsense es capaz de detectar, registrar y bloquear accesos no autorizados en entornos de red reales.

El aislamiento entre la LAN de usuarios y la LAN de administración fue verificado mediante intentos de acceso a la consola web de OPNsense desde equipos en la red de usuarios. En todos los casos, los intentos de conexión fueron bloqueados y reportados en el *firewall*.

Respecto al bloqueo geográfico mediante las listas MaxMind GeoLite2, no fue posible realizar pruebas funcionales directas porque no se cuenta con acceso a direcciones IP de origen extranjero durante el período de validación; no obstante, se verificó que los alias fueron creados correctamente en OPNsense y que las reglas asociadas se encontraban activas en la interfaz WAN.

Conectividad VPN y acceso remoto cifrado

Ambas instancias de OpenVPN funcionaron correctamente en plataformas Linux y Windows. El sistema exigió la contraseña de usuario para completar el proceso de autenticación. Cada intento de conexión, tanto exitoso como fallido, quedó registrado en el log de la VPN, lo que garantiza la visibilidad de todos los intentos de acceso a la misma.

Análisis de resultados

Los resultados obtenidos demuestran que OPNsense resolvió la situación planteada inicialmente. Las máquinas de administración de sistemas dejaron de estar expuestas directamente a Internet, el tráfico saliente quedó centralizado y controlado, los accesos no autorizados fueron bloqueados y registrados, así como el personal técnico dispuso de un mecanismo de acceso remoto cifrado y autenticado.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.190 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 81 - 82

La capacidad de observabilidad aportada por el *firewall* constituye una mejora significativa, ya que permite a los administradores detectar actividad sospechosa, auditar accesos y fundamentar decisiones de ajuste de políticas de seguridad con base en evidencia.

Implementaciones a futuro

En una etapa futura, se buscará incorporar un sistema de detección y prevención de intrusiones (IDS/IPS) mediante Suricata, complemento nativo de OPNsense, para añadir una capa de inspección profunda de paquetes que permita identificar patrones de ataque más sofisticados que los contemplados por el *firewall* con reglas estáticas. Veerasingam *et al*. (2023) demostraron la viabilidad de Suricata como sistema de detección y prevención de intrusiones incluso en entornos con recursos limitados, confirmando que su integración con plataformas de *firewall* existentes añade una capa de inspección profunda de tráfico que amplía significativamente las capacidades de detección frente a amenazas más sofisticadas que las cubiertas por reglas estáticas.

4. CONCLUSIONES

El centro de datos presentaba un esquema de red en el que las máquinas de administración de servicios operaban con direcciones IP públicas. Esta condición representaba una superficie de ataque mayor, situación por la que se decidió implementar OPNsense como plataforma de seguridad unificada.

La implementación de OPNsense brinda servicios integrados de *firewall* y VPN que resolvieron los planteamientos originales, el ocultamiento de las máquinas de administración respecto a Internet mediante el uso de direcciones IP privadas y NAT uno a uno, el control y filtrado del tráfico entrante y saliente mediante reglas basadas en el principio de mínimo privilegio, y el acceso remoto cifrado y autenticado para el personal técnico a través de dos instancias independientes de OpenVPN. Los intentos de conexión SSH no autorizados desde el exterior fueron bloqueados y registrados, el aislamiento entre los segmentos de red de usuarios y de administración se verificó correctamente, y ambas instancias VPN operaron de forma estable en plataformas Linux y Windows.

El único componente que no pudo validarse empíricamente fue el bloqueo geográfico mediante listas MaxMind GeoLite2. Su configuración fue completada, pero la prueba funcional quedó pendiente debido a que no se contó con direcciones IP extranjeras.

Para etapas posteriores se considera la implementación de un sistema centralizado de gestión de *logs* que consolide los registros del *firewall* y de las instancias VPN en una plataforma de análisis externa. Esto facilitaría la detección de patrones anómalos y la respuesta ante incidentes de seguridad a medida que la infraestructura del centro de datos crezca.

REFERENCIAS

Huda, A. S., Prihantoro, C., y Pranata, M. (2023). *Analisis perbandingan QoS pfSense dan OPNsense menggunakan metode load balancing* [Comparative analysis of QoS in pfSense and OPNsense using the load balancing method]. *Media Informatika, 22*(2), 87–95. https://doi.org/10.37595/mediainfo.v22i2.196

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.190 **Vol. 4, Núm. 3.** julio-septiembre 2026, págs. 82 - 82

Majid, J. (2025). Building a firewall and intrusion detection system based network security system using OPNsense tools. *Iraqi Journal of Intelligent Computing and Informatics (IJICI)*, *4*(1), 66–76. https://doi.org/10.52940/ijici.v4i1.96

Miller, J. (2025). *Enhancing network security: Can geo-blocking be the answer?* BitLyft. https://www.bitlyft.com/resources/enhancing-network-security-can-geo-blocking-be-the-answer

Purvant, S., y Sowmya, K. S. (2024). Advancements in network security. *International Journal of Research Publication and Reviews*, *5*(1), 5122–5126. https://ijrpr.com/uploads/V5ISSUE1/IJRPR22156.pdf

Tymoshchuk, V., Pakhoda, V., Dolinskyi, A., y Karnaukhov, A. (2024). Modelling cyber threats and evaluating the performance of intrusion detection systems. *Grail of Science*, (46), 638–646. https://doi.org/10.36074/grail-of-science.29.11.2024.081

Veerasingam, P., Abd Razak, S., Abidin, A. F. A., Mohamed, M. A., y Mohd Satar, S. D. (2023). Intrusion detection and prevention system in SME's local network by using Suricata. *Malaysian Journal of Computing and Applied Mathematics*, *6*(1), 21–30. https://doi.org/10.37231/myjcam.2023.6.1.88

Zscaler. (2024). *What is unified threat management?* https://www.zscaler.com/zpedia/what-unified-threat-management

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 83 - 105

Solicitudes de visitas guiadas mediante agentes de software

Guided visit requests using software agents

Información del reporte:

Para citar este reporte técnico:

Villarreal Brito, G. (2026). Solicitudes de visitas guiadas mediante agentes de software. *Cuadernos Técnicos Universitarios de la DGTIC, 4*(3), páginas (83 - 105). https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186

Gustavo Villarreal Brito
Instituto de Ciencias del Mar y Limnología
Universidad Nacional Autónoma de México
gvillarreal@cmarl.unam.mx
ORCID: 0009-0006-9119-1122

Resumen

Las visitas guiadas de la Unidad Académica de Sistemas Arrecifales coadyuvan a las actividades de divulgación científica y vinculación institucional. Antes de la mejora, la recepción de solicitudes dependía principalmente de correos, mensajes y revisiones manuales de disponibilidad, lo que generaba tiempos de respuesta variables, riesgo de omisiones y dificultad para conocer el estado de cada trámite. El objetivo fue diseñar, implementar y validar una arquitectura basada en agentes de *software* para apoyar la recepción, validación, notificación y seguimiento de solicitudes. Estos agentes operan como componentes de automatización basada en reglas, sin utilizar modelos de inteligencia artificial ni sustituir la decisión humana del coordinador.

La metodología incluyó la revisión del proceso manual, la identificación de reglas operativas, el diseño modular y la programación de componentes especializados para consultar agenda, validar formularios, registrar solicitudes, enviar notificaciones, apoyar el agendamiento y emitir confirmaciones.

Como resultado, las solicitudes quedaron centralizadas y asociadas a un estado de seguimiento. En el proceso manual, los tiempos de primera respuesta oscilaron entre 2 h 4 min y 28 días 2 h 50 min, con un promedio de 8.9 días; en el caso automatizado, la confirmación ocurrió en 42 minutos. Se concluyó que la arquitectura propuesta fue adecuada para un servicio con entradas estructuradas,

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 84 - 105

reglas claras y necesidad de trazabilidad. Como limitación, la decisión final de agenda siguió dependiendo de condiciones logísticas y académicas.

Palabras clave: Agenda digital, automatización administrativa, captura estructurada de solicitudes, gestión de solicitudes, seguimiento operativo.

Abstract

Guided visits at the Unidad Académica de Sistemas Arrecifales are part of its scientific outreach and institutional engagement activities. Before the improvement, request intake relied mainly on emails, messages, and manual availability checks, which generated variable response times, risk of omissions, and difficulty in tracking the status of each request. The objective was to design, implement, and validate a software agent-based architecture to support the intake, validation, notification, and follow-up of guided visit requests. These agents operate as rule-based automation components, without using artificial intelligence models or replacing the coordinator's human decision-making.

The methodology included reviewing the manual process, identifying operational rules, designing a modular architecture, and programming specialized components to query the calendar, validate forms, register requests, send notifications, support scheduling, and issue confirmations. As a result, requests were centralized and associated with a follow-up status. In the manual process, first-response times ranged from 2 h 4 minutes to 28 days 2 h 50 min, with an average of 8.9 days; in the documented automated case, confirmation occurred within 42 minutes. It was concluded that the architecture was suitable for a service with structured inputs, clear rules, and traceability needs. As a limitation, the final scheduling decision continued to depend on logistical and academic conditions.

Keywords: *Digital calendar, administrative automation, structured request intake, request management, operational follow-up.*

1. INTRODUCCIÓN

Las instituciones de educación superior realizan actividades de vinculación que requieren atención al público, coordinación administrativa y apoyo de plataformas tecnológicas. En estos servicios, la recepción de solicitudes por distintos medios, la revisión manual de disponibilidad y el seguimiento mediante mensajes aislados pueden generar tiempos de respuesta variables, duplicidad de información y dependencia de personas específicas. La literatura sobre automatización de procesos ha señalado que las tareas repetitivas, basadas en reglas y con entradas estructuradas, son candidatas adecuadas para ser apoyadas mediante componentes de *software* especializados (Van der Aalst *et al*., 2018; Syed *et al*., 2020).

En la Unidad Académica de Sistemas Arrecifales (UASA) del Instituto de Ciencias del Mar y Limnología (ICML) de la Universidad Nacional Autónoma de México (UNAM), las visitas guiadas formaron parte de las actividades de divulgación científica y vinculación con escuelas, instituciones y público general. Desde 2001, la comisión responsable atendió a más de 7,000 personas mediante recorridos, pláticas y actividades relacionadas con los ecosistemas arrecifales y el trabajo académico desarrollado en la Unidad.

Antes de la mejora, el servicio dependía principalmente de la supervisión manual de cada paso: revisar solicitudes, verificar restricciones operativas, validar datos de contacto, consultar disponibilidad de

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 85 - 105

espacios, coordinar fechas, comunicar confirmaciones a las instituciones interesadas y registrar la información en los sistemas de apoyo. Este esquema permitía atender las solicitudes, pero aumentaba el riesgo de omisiones en periodos de alta demanda y dificultaba conocer el estado actualizado de cada trámite. Esta situación mostró la necesidad de contar con un mecanismo más ordenado para recibir, validar, notificar y dar seguimiento a las solicitudes.

La solución se basó en agentes de *software*, los cuales operan como componentes programados dentro de una automatización basada en reglas; cabe aclarar que estos agentes no emplean modelos de inteligencia artificial ni toman decisiones autónomas sobre la agenda. Su función consiste en recibir eventos, validar datos, consultar disponibilidad, enviar notificaciones y actualizar estados conforme a reglas previamente definidas.

Estudios sobre bots y agentes conversacionales han mostrado que estos componentes pueden apoyar tareas de coordinación, respuesta y seguimiento cuando se integran de forma controlada a procesos institucionales (Lebeuf *et al*., 2018; Diederich *et al*., 2022). Asimismo, en servicios de atención, se ha destacado la importancia de conservar la claridad en la interacción, la trazabilidad y el control humano sobre las decisiones relevantes (Androutsopoulou *et al*., 2019).

En este contexto, el objetivo fue diseñar, implementar y validar una arquitectura basada en agentes de *software* para automatizar la recepción, validación, notificación y seguimiento de solicitudes de visitas guiadas en la UASA; se mantuvo únicamente la decisión final del coordinador sobre la agenda institucional.

2. DESARROLLO TÉCNICO

2.1 METODOLOGÍA

El trabajo se desarrolló a partir de la revisión del proceso de solicitudes de visitas guiadas, desde el primer contacto de la institución interesada hasta la confirmación final. Se identificaron actividades repetitivas, como recibir mensajes, revisar fechas disponibles, validar datos, consultar al coordinador, registrar la visita y responder al solicitante; asimismo, se identificaron las decisiones que debían mantenerse bajo control humano, especialmente la confirmación definitiva de la fecha. La secuencia general del flujo se resume en el Anexo A.

Con base en el análisis, se definieron las reglas operativas que debía respetar el sistema. Entre ellas, se incluyeron la anticipación mínima para solicitar una visita, los días permitidos, los meses con restricciones, el cupo máximo mensual y los periodos no laborables. Estas reglas ya formaban parte de la operación cotidiana del servicio, pero no estaban integradas en un solo mecanismo automatizado. Por ello, se documentaron y se usaron como base para programar la consulta de disponibilidad y la validación inicial de las solicitudes, como se muestra en la matriz del Anexo C.

Después, se diseñó una arquitectura modular en la que cada agente cumple una función específica: el agente de disponibilidad consulta la agenda; el agente de captura valida el formulario; el agente notificador avisa al coordinador; el agente de agendamiento asistido actualiza el estado de la visita; y los agentes complementarios atienden consultas por correo y generan reportes. La coordinación entre

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 86 - 105

componentes se realizó mediante una base de datos, de acuerdo con el esquema presentado en el apartado 5 del Anexo B.

La implementación se realizó con herramientas compatibles con la infraestructura disponible en la Unidad. El formulario y la consulta de disponibilidad se desarrollaron en PHP por su integración con el servidor web institucional. Los agentes de notificación, agendamiento y resumen se programaron en Node.js, porque requerían ejecutar tareas programadas, comunicarse con servicios externos y atender eventos de mensajería. Por otro lado, el agente de correo se desarrolló en Python debido a la disponibilidad de bibliotecas estables para consultar buzones y enviar respuestas mediante protocolos estándar. Los fragmentos principales de estos componentes se incluyeron en el Anexo B para documentar la implementación.

Durante el desarrollo, se realizaron pruebas funcionales por componente. Se verificó que el formulario rechazara registros incompletos, que la consulta excluyera fechas ocupadas o bloqueadas, que el agente notificador detectara solicitudes pendientes y que el flujo de confirmación actualizara el estado de la visita. En estas pruebas, se identificó que la validación de fechas era el punto más sensible, porque debía combinar reglas institucionales, disponibilidad real de agenda y confirmación humana.

La validación operativa se centró en comprobar que el sistema respetara las reglas del servicio y que mantuviera la intervención humana en la decisión final. No se buscó sustituir al coordinador, sino reducir las tareas repetitivas previas y posteriores a su decisión. Con ello, la metodología permitió pasar de un proceso distribuido entre correos, mensajes y revisiones manuales, a un flujo documentado mediante agentes especializados y estados persistentes en una base de datos.

2.2 IMPLEMENTACIÓN

El formulario captura los datos mínimos de la solicitud: institución, responsable, contacto, participantes, nivel académico, fecha solicitada y comentarios. La validación incluyó método de envío, formato de correo y teléfono, campos cerrados, longitudes máximas, CAPTCHA y limitación de tasa. La persistencia de estos datos se vinculó con el esquema de la tabla principal descrito en el apartado 5 del Anexo B.

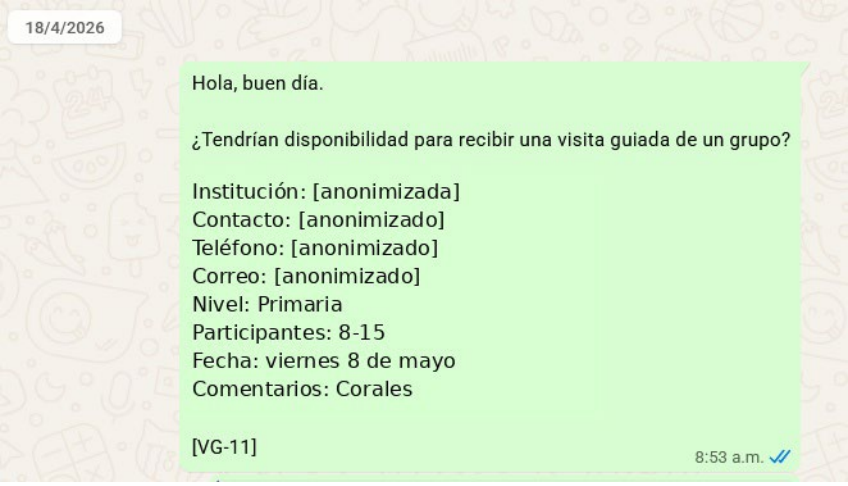
El agente de disponibilidad consulta la agenda antes del envío de la solicitud y genera una lista de viernes disponibles dentro de un horizonte de doce semanas. Para ello, se aplica reglas institucionales de anticipación mínima, restricción de meses, cupo mensual y bloqueo de días no laborables. La consulta se ejemplifica en el apartado 1 del Anexo B y corresponde a las reglas del Anexo C.

El manejo de seguridad se incorporó desde el formulario: encabezados HTTPS de protección, sanitización de campos de texto y consultas parametrizadas para reducir la exposición a inyección de código. El agente notificador revisa periódicamente las solicitudes pendientes y genera un aviso al coordinador por mensajería instantánea. En la Figura 1 se muestra el mensaje que fue redactado automáticamente con los datos de la solicitud.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 87 - 105

Figura 1

Mensaje de solicitud automatizado

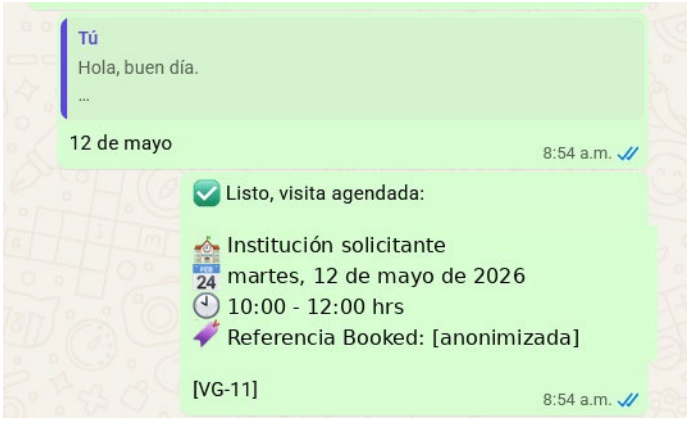


Nota. Mensaje al coordinador para su validación con los datos mínimos de la solicitud.

El diseño conservó la intervención humana en la selección definitiva de fecha, ya que esa decisión dependía de condiciones logísticas y de la disponibilidad de los recursos. El responsable debe responder con una afirmación o con una fecha específica. En la Figura 2 se observa la interacción de confirmación con el agente, en la que se seleccionó una fecha distinta a la propuesta por éste.

Figura 2

Confirmación de visita validada en la agenda



Nota. La figura muestra la respuesta humana del coordinador y la confirmación generada por el agente después de registrar la visita en la agenda institucional y la notificación a los interesados.

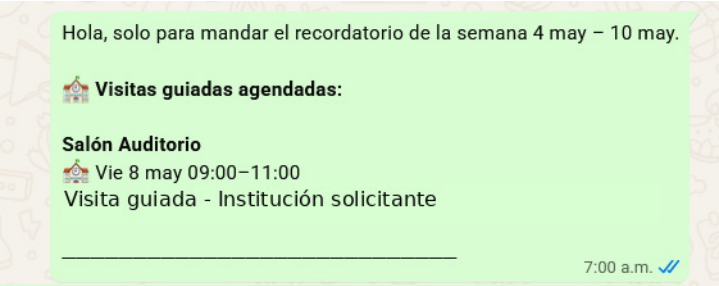
La lógica de notificación por WhatsApp se documentó en el apartado 2 del Anexo B; la recepción de respuestas y el agendamiento asistido, en el apartado 7 del Anexo B; el agente de correo, en el apartado 4 del Anexo B; el resumen semanal enviado por WhatsApp, en el apartado 3 del Anexo B; y el bloqueo de

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 88 - 105

días inhábiles, en el apartado 6 del Anexo B. De forma complementaria, la Figura 3 muestra un ejemplo del resumen semanal enviado al coordinador.

Figura 3

Resumen semanal de visitas



La solución se implementó como un conjunto de agentes especializados y no como un único flujo centralizado en una herramienta de automatización visual. Cada componente respondió a un evento concreto, aplicó reglas del servicio y actualizó el estado de la solicitud sin depender directamente de los demás módulos. Así, el sistema está bajo acoplamiento, facilita el mantenimiento individual de cada agente y conserva evidencia técnica mediante código fuente revisable. La Tabla 1 resume esta distribución funcional.

Tabla 1

Componentes funcionales de la solución de agentes

Componente	Tecnología	Disparador	Función principal
Captura web	PHP	Solicitud HTTPS	Valida y registra solicitudes de visita
Disponibilidad	PHP	Consulta del formulario	Publica fechas disponibles con reglas de cupo
Notificación	Node.js	Tarea programada	Avisa al coordinador sobre nuevas solicitudes
Agendamiento asistido	Node.js	Respuesta del coordinador	Actualiza estados, agenda y confirmación
Correo y resumen	Python / Node.js	Tareas programadas	Atiende consultas y genera reportes semanales

3. RESULTADOS

La solución centralizó la entrada de solicitudes mediante un formulario controlado y redujo la dispersión de información entre correos, mensajes y consultas manuales. El registro en base de datos permitió

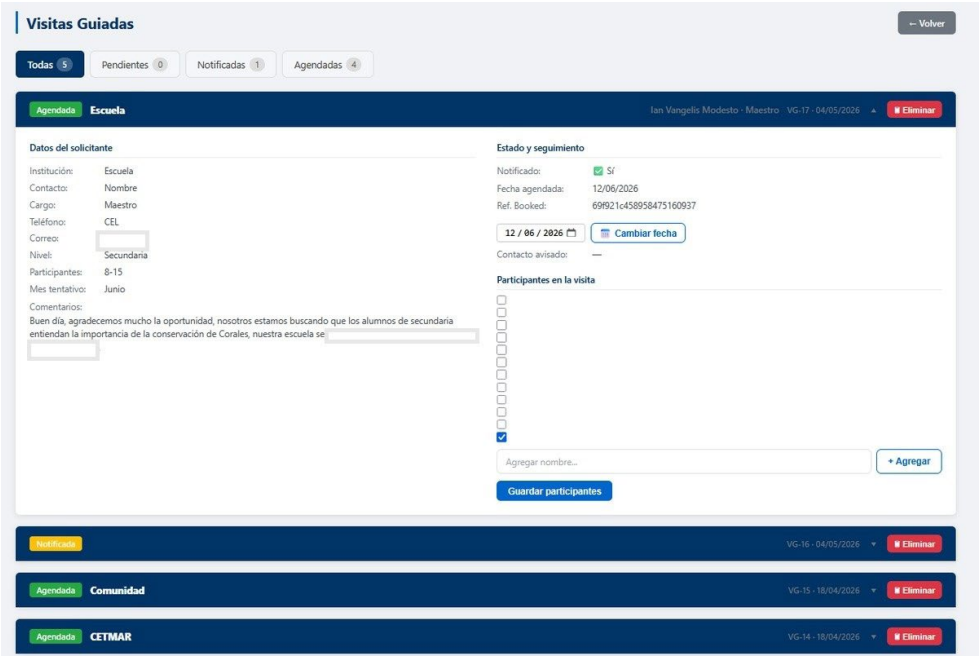
https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 89 - 105

conservar un estado único por solicitud, con lo cual se facilitó el seguimiento y disminuyó la probabilidad de duplicidad. Este resultado se relacionó con el flujo general del Anexo A y con el esquema de base de datos documentado en el apartado 5 del Anexo B.

La pizarra de seguimiento permitió visualizar en una sola interfaz el estado de cada solicitud, la fecha agendada, el avance de notificación y los participantes asignados a la visita. Como se muestra en la Figura 4, esta vista facilita la supervisión operativa del proceso sin consultar directamente la base de datos.

Figura 4

Seguimiento de solicitudes de visitas guiadas



La arquitectura basada en agentes permitió incorporar reglas de operación documentadas en el Anexo C, validaciones de seguridad, control de cupo, manejo de fechas no disponibles y persistencia de estados.

El tiempo de aviso al coordinador quedó acotado por el intervalo de ejecución programada del agente notificador. En el diseño evaluado, la notificación pudo emitirse dentro de una ventana máxima de cuatro minutos desde el registro de la solicitud. La evidencia técnica de esta operación se vinculó con la consulta de solicitudes pendientes y la actualización de estado, descritas en el apartado 2 del Anexo B, así como con el servidor interno de mensajería del apartado 7 del Anexo B.

Para dimensionar el cambio operativo, se comparó una muestra de hilos de correo previos con un caso posterior a la implementación, cuya evidencia anonimizada se presenta en el Anexo D. Como se resume en la Tabla 2, los tiempos de primera respuesta observados en el proceso manual variaron entre 2 h 4 min y 28 días 2 h 50 min, con un promedio de 8.9 días. En el flujo automatizado documentado, la confirmación de fecha se realizó en 42 minutos y el registro en agenda ocurrió en el mismo minuto de la validación humana.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 90 - 105

Tabla 2

Comparación de indicadores operativos antes y después de la automatización

Indicador	Proceso manual	Proceso automatizado
Tiempo mínimo observado de respuesta	2 h 4 min	41 min
Tiempo máximo observado de respuesta	28 días 2 h 50 min	42 min en el caso documentado
Promedio observado	8.9 días	42 min en el caso documentado
Registro en agenda	Manual, posterior a la confirmación	En el mismo minuto de la validación humana
Trazabilidad	Hilos de correo dispersos	Folio de solicitud y referencia de agenda

El impacto operativo se observó en tres dimensiones: oportunidad de respuesta, al pasar de revisiones manuales a avisos programados; consistencia del servicio, porque todas las solicitudes siguieron las mismas reglas; y capacidad de supervisión, ya que el estado de cada solicitud quedó disponible para consultas posteriores. Esta reducción fue relevante porque la coordinación de visitas guiadas es una actividad complementaria del personal técnico académico.

4. CONCLUSIONES

El proceso de visitas guiadas presentó condiciones adecuadas para su automatización, debido a que combinaba reglas explícitas, entradas estructuradas, actividades repetitivas y necesidad de seguimiento. La solución implementada cumplió el objetivo de automatizar la recepción, validación, notificación y seguimiento de solicitudes, sin sustituir la decisión final del coordinador sobre la agenda institucional. De esta manera, la automatización redujo tareas operativas y fortaleció la atención ordenada del servicio.

Los resultados mostraron que el desacoplamiento entre componentes facilitó la integración del formulario web, la agenda institucional, el correo electrónico y la mensajería instantánea. Esta organización permitió conservar trazabilidad por solicitud, reducir dependencias entre módulos y mantener la intervención humana en la validación final de la fecha.

La principal aportación del trabajo fue demostrar que un servicio universitario de vinculación podía ser transformado en un flujo documentado y verificable, sin perder la flexibilidad necesaria para la toma de decisiones académicas y logísticas. La implementación permitió aplicar reglas de disponibilidad de forma uniforme, disminuir el riesgo de omisiones y establecer una base técnica replicable para otros servicios administrativos de apoyo académico.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 **Vol. 4, Núm. 3.** julio-septiembre 2026, págs. 91 - 105

AGRADECIMIENTOS

Se agradece a la comisión de visitas guiadas de la Unidad Académica de Sistemas Arrecifales del Instituto de Ciencias del Mar y Limnología por la colaboración brindada para documentar los procesos y validar los requerimientos funcionales de la solución.

REFERENCIAS

Androutsopoulou, A., Karacapilidis, N., Loukis, E., y Charalabidis, Y. (2019). Transforming the communication between citizens and government through AI-guided chatbots. *Government Information Quarterly*, *36*(2), 358–367. https://doi.org/10.1016/j.giq.2018.10.001

Diederich, S., Brendel, A. B., Morana, S., y Kolbe, L. M. (2022). On the design of and interaction with conversational agents: An organizing and assessing review of human-computer interaction research. *Journal of the Association for Information Systems*, *23*(1), 96–138. https://doi.org/10.17705/1jais.00724

Lebeuf, C., Storey, M. A., y Zagalsky, A. (2018). Software bots. *IEEE Software, 35*(1), 18–23. https://doi.org/10.1109/MS.2017.4541027

Syed, R., Suriadi, S., Adams, M., Bandara, W., Leemans, S. J. J., Ouyang, C., ter Hofstede, A. H. M., van de Weerd, I., Wynn, M. T., y Reijers, H. A. (2020). Robotic process automation: Contemporary themes and challenges. *Computers in Industry*, *115*, 103162. https://doi.org/10.1016/j.compind.2019.103162

Van der Aalst, W. M. P., Bichler, M., y Heinzl, A. (2018). Robotic process automation. *Business & Information Systems Engineering*, *60*(4), 269–272. https://doi.org/10.1007/s12599-018-0542-4

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 92 - 105

ANEXO A. ARQUITECTURA FUNCIONAL RESUMIDA

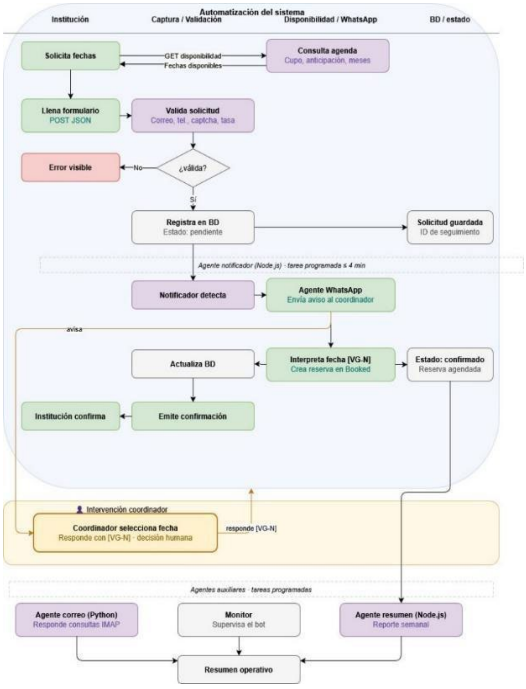
El proceso inicia cuando una institución consulta las fechas disponibles. El agente de disponibilidad revisa la agenda institucional, aplica las reglas del servicio —día permitido, anticipación mínima, meses restringidos y cupo mensual— y devuelve únicamente las fechas válidas. Con la fecha seleccionada, la institución envía el formulario en línea. El agente de captura valida los datos, registra la solicitud en la base de datos con estado "pendiente" y asigna un identificador de seguimiento. Si alguna validación falla, el error se notifica de inmediato.

En un ciclo programado de hasta cuatro minutos, el agente notificador detecta las solicitudes nuevas y envía un aviso automático al coordinador a través del agente WhatsApp, incluyendo los datos de la visita y un *token* de referencia [VG-N]. El coordinador revisa la información y responde con la fecha confirmada. El agente WhatsApp interpreta la respuesta, crea la reserva en el sistema de agenda, actualiza el estado a "confirmado" en la base de datos y envía la confirmación por correo electrónico a la institución solicitante.

De forma complementaria, el agente de correo atiende consultas entrantes respondiendo automáticamente con el enlace al formulario, mientras que el agente de resumen genera un reporte semanal de visitas programadas para el coordinador. Ambos agentes operan en modo lectura sin alterar el flujo principal.

Figura 5

Diagrama conceptual del proceso automatizado



Nota. El diagrama distingue las actividades ejecutadas por agentes de *software*, las interacciones con la base de datos y la intervención del coordinador en la validación final de la fecha.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 93 - 105

ANEXO B. FRAGMENTOS REPRESENTATIVOS DE IMPLEMENTACIÓN

Los fragmentos incluidos en este anexo son extractos representativos de la implementación y no corresponden al código fuente completo del sistema. Su propósito es mostrar la lógica técnica principal de cada componente, como validación, consulta de disponibilidad, notificación, actualización de estados y agendamiento asistido. Para proteger la seguridad de la infraestructura institucional, los fragmentos fueron depurados y ajustados con el fin de omitir credenciales, rutas internas, direcciones, identificadores sensibles, parámetros de conexión y cualquier configuración que pudiera comprometer los servicios tecnológicos.

1. Agente de disponibilidad — *disponibilidad.php* (PHP)

Consulta la API REST de Booked Scheduler y filtra los viernes disponibles. El fragmento muestra la lógica central: autenticación, consulta de reservas existentes y cálculo de fechas libres.

Consulta a Booked y cálculo de los viernes disponibles:

```
$auth = booked_post(BOOKED_BASE . '/Authentication/Authenticate',
    ['username' => BOOKED_USER, 'password' => BOOKED_PASS]);

$token  = $auth['sessionToken'];
$userId = $auth['userId'];
$authHeaders = [
    'X-Booked-SessionToken: ' . $token,
    'X-Booked-UserId: ' . $userId,
];

// Rango: mínimo 2 semanas desde hoy, 12 semanas adelante
$inicio = (new DateTime())->modify('+14 days');
$fin    = (clone $inicio)->modify('+12 weeks');

$url = BOOKED_BASE . '/Reservations/?'
    . 'startDateTime=' . urlencode($inicio->format('Y-m-d\\
T00:00:00'))
    . '&endDateTime='  . urlencode($fin->format('Y-m-d\\T23:59:59'))
    . '&resourceId='   . RESOURCE_ID;

$reservas = booked_get($url, $authHeaders)['reservations'] ?? [];

while ($cursor <= $limite && count($disponibles) < 8) {
    $yyyymmdd = $cursor->format('Y-m-d');
    $mes_key  = $cursor->format('Y-m');

    if (($visitas_por_mes[$mes_key] ?? 0) >= 2) {
        $cursor->modify('+7 days'); continue;
    }
    // Verificar traslape con 10:00-12:00 (slot UTC 15:00-17:00)
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 94 - 105

```
    $ocupado = /* comparación de rangos de tiempo */ false;
    if (!$ocupado && !esta_bloqueado($yyyymmdd, $DIAS_BLOQUEADOS,
$RANGOS))
        $disponibles[] = ['value'=>$yyyymmdd, 'label'=>"viernes
..."];
    $cursor->modify('+7 days');
}
echo json_encode(['fechas' => $disponibles]);
```

2. Agente notificador — *visitas_notifier.js* (Node.js)

Lee solicitudes no notificadas en la BD, limpia los campos antes de construir el mensaje y lo envía por mensajería instantánea. Una vez confirmado el envío, marca el registro como notificado.

Sanitización de campos y composición del mensaje:

```
function sanitizarCampo(val) {
    return String(val ?? '')
        .replace(/[\x00-\x08\x0B\x0C\x0E-\x1F\x7F]/g, '')
        .replace(/ignore\\s+(previous|all)\\s+instructions?/gi,
'[eliminado]')
        .replace(/system\\s*:/gi, '[eliminado]')
        .trim();
}

function mensajeDefault(v, fechasDisponibles) {
    const fechasStr = fechasDisponibles?.map(f => {
        const dia = DIAS_ES[f.getDay()];
        return `${dia} ${f.getDate()} de ${MESES_ES[f.getMon-
th()+1]}`;
    }).join(', ') ?? null;

    return `Hola, buen día.

¿Tendrían disponibilidad para recibir una visita guiada?

Institución: ${v.escuela}
Contacto: ${v.responsable} (${v.cargo})
Teléfono: ${v.telefono}
Nivel: ${v.nivel_academico}
Fecha: ${fechasStr ?? v.mes_tentativo}

Gracias.

[VG-${v.id}]`;
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 95 - 105

```
}
```

Ciclo principal: consulta, envío y actualización de estado:

```
async function main() {
    const { rows } = await pool.query(
        "SELECT * FROM visitas_guiadas"
        + " WHERE notificado = FALSE ORDER BY creado_en ASC"
    );

    for (const visita of rows) {
        const mensaje = await componerMensaje(visita, fechasDisponi-
bles);

        const res = await enviarWhatsApp(mensaje);
        if (res.ok) {
            await pool.query(
                "UPDATE visitas_guiadas SET notificado = TRUE WHERE
id = $1",
                [visita.id]
            );
        }
        await new Promise(r => setTimeout(r, 3000)); // pausa entre
envíos
    }
    await pool.end();
}
```

3. Agente de resumen — *agenda_semanal.js* (Node.js)

Consulta los tres recursos del auditorio en Booked Scheduler, filtra las reservas cuyo título contiene la palabra "visita" y envía el resumen semanal. Si no hay visitas registradas, no genera mensaje.

```
// Consultar los 3 recursos en paralelo
const consultas = Object.keys(RECURSOS).map(rid =>
    httpsGet(`${BOOKED_BASE}/Reservations/?` +
        `startDateTime=${inicio}&endDateTime=${fin}&resourceI-
d=${rid}`,
        headers
    ).then(data => ({ rid, reservas: data.reservations || [] }))
);
const resultados = await Promise.all(consultas);

const visitasPorRecurso = {};
for (const { rid, reservas } of resultados) {
    visitasPorRecurso[rid] = reservas.filter(
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 96 - 105

```
        r => /visita/i.test(r.title || '')
    );
}

if (totalVisitas === 0) { log('Sin visitas esta semana.'); return; }

let msg = `Hola, recordatorio semana ${semanaStr}.\n\n`;
for (const [rid, visitas] of Object.entries(visitasPorRecurso)) {
    if (!visitas.length) continue;
    msg += `*${RECURSOS[rid]}*\n`;
    for (const r of visitas) {
        msg += `  ${fmtFecha(r.startDate)} ${fmtHora(r.startDa-
te)}-${fmtHora(r.endDate)}\n`;
        msg += `    ${(r.title||'').slice(0,60)}\n`;
    }
}
await enviarWhatsApp(msg);
```

4. Agente de correo — *visitas_mail.py* (Python)

Revisa la bandeja institucional por IMAP, detecta correos que pidan informes sobre visitas guiadas y responde indicando el formulario. Los correos automáticos (*noreply*, *mailer-daemon*) son omitidos.

Detección de solicitudes de informe**s:**

```
PALABRAS_INFORMES = [
    r'inform[eo]s?',        r'informaci[oó]n',
    r'solicitar?\\s+visita', r'visita\\s+guiada',
    r'disponibilidad',      r'c[óo]mo\\s+(?:solicitar|agendar)',
    r'requisitos?',         r'nos?\\s+gustar[íiI]a',
]

def pide_informes(asunto: str, cuerpo: str) -> bool:
    texto = (asunto + ' ' + cuerpo).lower()
    return any(re.search(p, texto) for p in PALABRAS_INFORMES)
```

Ciclo principal de revisión:

```
for uid in uids:                          # solo correos no leídos
    msg    = email.message_from_bytes(...)
    asunto = sanitizar_header(decode_str(msg.get('Subject', '')))
    _, addr = parseaddr(msg.get('From', ''))

    # Omitir correos automáticos
    if re.search(r'noreply|no-reply|mailer-daemon', addr, re.I):
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 97 - 105

```
            mail.store(uid, '+FLAGS', '\\\\Seen'); continue

        if pide_informes(asunto, extraer_cuerpo(msg)):
            enviar_respuesta(addr, nombre, asunto, message_id)
            logging.info(f'Respuesta enviada a {addr}')

        mail.store(uid, '+FLAGS', '\\\\Seen')   # marcar leído
```

5. Esquema de base de datos — *schema.sql* (PostgreSQL)

La tabla principal del módulo de visitas guiadas. El campo "notificado" es el mecanismo de coordinación entre el agente de captura y el agente notificador; los demás campos registran la trazabilidad completa de cada solicitud.

```
CREATE TABLE IF NOT EXISTS visitas_guiadas (
    id                SERIAL PRIMARY KEY,
    correo            VARCHAR(200) NOT NULL,
    escuela           VARCHAR(300) NOT NULL,
    responsable       VARCHAR(200) NOT NULL,
    telefono          VARCHAR(30)  NOT NULL,
    cargo             VARCHAR(200),
    participantes     VARCHAR(20),
    nivel_academico   VARCHAR(200),
    mes_tentativo     VARCHAR(30),
    fecha_solicitada  DATE,
    fecha_agendada    DATE,
    comentarios       TEXT,
    notificado        BOOLEAN DEFAULT FALSE,
    estado            VARCHAR(30) DEFAULT 'pendiente',
    creado_en         TIMESTAMP DEFAULT NOW()
);
```

6. *Script* auxiliar — *bloquear_dias.js* (Node.js)

Bloquea días inhábiles y periodos vacacionales en Booked, creando reservas administrativas en el Salón Auditorio. Se ejecuta manualmente antes de cada ciclo escolar para sincronizar la agenda pública con el calendario institucional.

```
// Días individuales y rangos a bloquear
const DIAS = [
    { y:2026, m:1,  d:1,  titulo:'Día inhábil – Año nuevo' },
    { y:2026, m:5,  d:1,  titulo:'Día inhábil – Día del trabajo' },
    // ...
];
const RANGOS = [
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 98 - 105

```
    { ini:[2026,3,30], fin:[2026,4,3],  titulo:'Asueto académico' },
    { ini:[2026,7,6],  fin:[2026,7,24], titulo:'Vacaciones adminis-
trativas' },
];

// Para cada fecha, crear reserva con título 'BLOQUEADO — <motivo>'
const res = await httpsPost(`${BOOKED_BASE}/Reservations/`, {
    resourceId:    RESOURCE_ID,
    startDateTime: `${fecha}T09:30:00`,
    endDateTime:   `${fecha}T15:30:00`,
    title:         `BLOQUEADO — ${t.titulo}`,
}, headers);

if (/conflicto|conflict/i.test(res.errors?.join(';') || '')) {
    log(`Ya tiene reserva, omitiendo: ${fecha}`);
    continue;
}
```

7. Agente de WhatsApp — *index.js* (Node.js)

Es el componente central del sistema de notificación. Implementa tres responsabilidades en un mismo proceso: (a) un servidor HTTP interno que recibe mensajes de los demás agentes y los envía por WhatsApp; (b) el *listener* de mensajes entrantes que detecta respuestas del coordinador mediante el *token* [VG-N]; y (c) el agente de agendamiento asistido que, al recibir una fecha válida, crea la reserva en Booked, actualiza la base de datos y envía la confirmación por correo al solicitante.

Servidor HTTP interno:

```
// Los demás agentes (notifier, agenda_semanal) llaman a este en-
dpoint
app.post('/enviar', async (req, res) => {
    const { numero, mensaje } = req.body;
    if (!sock) return res.json({ error: 'Bot no conectado' });

    const jid = numero.includes('@')
        ? numero
        : `${numero}@s.whatsapp.net`;
    await sock.sendMessage(jid, { text: mensaje });
    log(`ENVIADO → ${jid}: ${mensaje}`);
    res.json({ ok: true });
});

app.get('/mensajes', (req, res) => {
    const lineas = fs.existsSync(LOG_FILE)
        ? fs.readFileSync(LOG_FILE,'utf8').trim().split('\n').sli-
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 99 - 105

```
ce(-50)
        : [];
    res.json(lineas);
});
```

Detección de respuestas del coordinador:

```
sock.ev.on('messages.upsert', async ({ messages, type }) => {
    if (type !== 'notify') return;
    for (const msg of messages) {
        const texto  = msg.message?.conversation
                    || msg.message?.extendedTextMessage?.text || '';

        const ctxInfo  = msg.message?.extendedTextMessage?.contex-
tInfo || null;
        const citado   = ctxInfo?.quotedMessage?.conversation
                      || ctxInfo?.quotedMessage?.extendedTextMessa-
ge?.text || '';

        const textoCompleto = texto + (citado ? ' ' + citado : '');

        if (textoCompleto.includes('[VG-')) {
            await procesarRespuesta(msg.key.remoteJid, textoComple-
to);
        }
    }
});
```

Extracción de fecha en español:

```
function parsearFecha(texto) {
    const m = texto.match(
        /(\d{1,2})\s+de\s+(enero|febrero|marzo|abril|mayo|junio|
         julio|agosto|septiembre|octubre|noviembre|diciembre)/i
    );
    if (!m) return null;
    const fecha = new Date(
        new Date().getFullYear(),
        MESES_PARSE[m[2].toLowerCase()] - 1,
        parseInt(m[1])
    );
    if (fecha < new Date()) fecha.setFullYear(fecha.getFullYear() +
1);
    return fecha;
}
```

Agendamiento asistido: reserva en Booked + confirmación por correo:

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 100 - 105

```
async function procesarRespuesta(jid, texto) {
    const tokenMatch = texto.match(/\[VG-(\d+)\]/i);
    if (!tokenMatch) return;

    const visitaId = parseInt(tokenMatch[1]);
    const fecha    = parsearFecha(texto);

    if (!fecha) {
        await sock.sendMessage(jid, {
            text: `No pude identificar la fecha [VG-${visitaId}].`
                + ` Indica: "viernes 12 de junio [VG-${visitaId}]"`
        });
        return;
    }

    const { rows } = await pool.query(
        'SELECT * FROM visitas_guiadas WHERE id = $1', [visitaId]
    );
    const visita = rows[0];

    const bookedRes = await crearReservaBooked(visita, fecha);
    const ref = bookedRes.referenceNumber || bookedRes.reservatio-
nId;

    await pool.query(
        'UPDATE visitas_guiadas SET fecha_agendada=$1, referencia_
booked=$2 WHERE id=$3',
        [fecha, ref, visitaId]
    );

    // Confirmar al coordinador por WhatsApp
    const diaStr = fecha.toLocaleDateString('es-MX',
        { weekday:'long', day:'numeric', month:'long', year:'nume-
ric' });
    await sock.sendMessage(jid, {
        text: ` Visita agendada:\n ${visita.escuela}\n ${diaStr}`
            + `\n Referencia: ${ref}  [VG-${visitaId}]`
    });

    // Notificar al solicitante por correo
    await enviarEmailContacto(visita, diaStr);
}
```

Conexión a WhatsApp y reconexión automática:

```
async function conectar() {
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 101 - 105

```
    const { state, saveCreds } = await useMultiFileAuthState(AUTH_
DIR);
    const { version }          = await fetchLatestBaileysVersion();

    sock = makeWASocket({
        version, auth: state,
        logger: pino({ level: 'silent' }),
        printQRInTerminal: false,
    });

    sock.ev.on('connection.update', ({ connection, lastDisconnect,
qr }) => {
        if (qr) qrcode.generate(qr, { small: true });  // escanear
al iniciar

        if (connection === 'close') {
            const code = lastDisconnect?.error?.output?.statusCode;
            const noReconectar = [
                DisconnectReason.loggedOut,
                DisconnectReason.connectionReplaced,  // otra ins-
tancia activa
            ];
            if (!noReconectar.includes(code))
                setTimeout(conectar, 3000);  // reconexion automáti-
ca
        }
    });
    sock.ev.on('creds.update', saveCreds);
}
conectar();
```

ANEXO C. MATRIZ DE REGLAS OPERATIVAS DEL SISTEMA

Las reglas operativas son las condiciones, restricciones y criterios explícitos que gobiernan el comportamiento del sistema y delimitan qué solicitudes son válidas, cuándo se pueden atender y cómo deben procesarse. Para efectos de la implementación, se clasificaron en reglas del servicio y reglas técnicas.

Reglas del servicio

Estas reglas definieron las condiciones institucionales bajo las cuales podía atenderse una visita guiada:

1. Las visitas se realizan únicamente los viernes.
2. Se requiere una anticipación mínima de dos semanas entre la solicitud y la fecha de visita.
3. No se realizan visitas durante julio ni en el mes de diciembre.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 102 - 105

4. El cupo máximo es de dos visitas por mes.
5. El horario ordinario de visita es de 10:00 a 12:00 horas.
6. Los días inhábiles y periodos vacacionales institucionales se bloquean en la agenda antes de cada ciclo escolar.

Reglas técnicas

Estas reglas definieron la forma en que el sistema valida, procesa y protege las solicitudes:

1. Sólo se aceptan solicitudes por método POST con contenido JSON.
2. El tamaño máximo de la petición se limita para prevenir el abuso del formulario.
3. El correo electrónico debe tener formato válido.
4. El teléfono debe cumplir con un patrón de dígitos definido.
5. Los campos cerrados, como nivel académico y cargo, se validan contra listas blancas.
6. Cada campo tiene una longitud máxima permitida.
7. Se requiere verificación CAPTCHA para confirmar que la solicitud proviene de una persona.
8. Se aplica limitación de tasa con un máximo de cinco intentos por hora por dirección de origen.
9. Las operaciones en base de datos se realizan mediante consultas parametrizadas.
10. El agente notificador ejecuta su ciclo en intervalos de hasta cuatro minutos.
11. La respuesta del coordinador debe incluir el *token* [VG-N] y una fecha en lenguaje natural para activar el agendamiento.
12. Los correos automáticos, como *noreply* o *mailer-daemon*, son ignorados por el agente de correo.
13. La caché de disponibilidad tiene una vigencia de treinta minutos para reducir la carga sobre Booked Scheduler.

ANEXO D. EVIDENCIA OPERATIVA DE TIEMPOS DE RESPUESTA

Este anexo presenta evidencia operativa utilizada para estimar los tiempos de respuesta antes y después de la implementación del sistema. Para proteger la privacidad de las instituciones solicitantes, se anonimizaron nombres de escuelas, personas responsables, correos electrónicos, teléfonos y referencias internas. Las fechas y horas se conservaron únicamente con fines de análisis operativo.

1. Evidencia del proceso manual previo

Antes de la implementación, las solicitudes de visitas guiadas se recibían principalmente por correo electrónico. La atención dependía de la revisión manual de mensajes, la consulta de disponibilidad en agenda, la validación de fechas posibles y la comunicación posterior con la institución solicitante. Esta forma de operación generaba tiempos de respuesta variables, especialmente en periodos con alta

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 103 - 105

demanda o cuando era necesario ajustar fechas por cupo, días inhábiles o disponibilidad del personal responsable.

En la muestra revisada, se identificaron respuestas que ocurrieron desde el mismo día hasta varias semanas después de la solicitud inicial. También se observaron hilos con varios intercambios de correo antes de llegar a una confirmación o ajuste de fecha. Esta evidencia mostró que el proceso manual dependía de la revisión periódica de mensajes y de la consulta caso por caso de la disponibilidad.

Figura 6

Ejemplo anonimizado de solicitud atendida mediante correo electrónico con tiempo de respuesta extendido

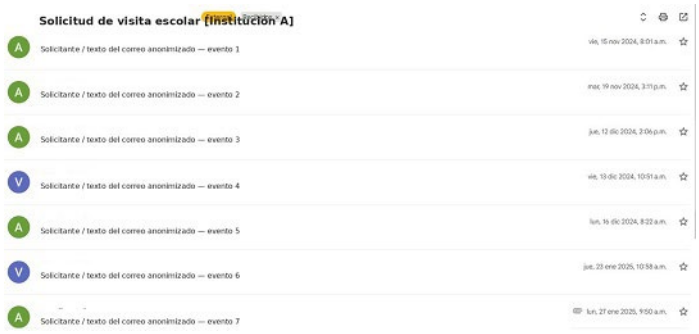


Figura 7

Ejemplo anonimizado de coordinación manual para revisar disponibilidad y ajustar fechas

2. Evidencia del proceso posterior a la implementación

Después de la implementación, el sistema permitió recibir solicitudes mediante un formulario controlado, registrar los datos en una base de datos, generar un folio de seguimiento y notificar automáticamente al coordinador. La respuesta humana continuó siendo necesaria para confirmar la fecha definitiva, pero el flujo redujo las tareas manuales previas y posteriores a esa decisión.

En el caso documentado, el agente notificó una nueva solicitud al coordinador a las 5:04 p.m. La primera respuesta humana se recibió a las 5:45 p.m. y la confirmación de fecha ocurrió a las 5:46 p.m. En ese mismo minuto, el sistema registró la visita en la agenda y generó la referencia correspondiente. Esto

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 Vol. 4, Núm. 3. julio-septiembre 2026, págs. 104 - 105

permitió observar un tiempo de 42 minutos entre la notificación automática y la confirmación de fecha, así como un registro en agenda realizado en el mismo minuto de la validación humana.

Figura 8

Ejemplo anonimizado de notificación automática, respuesta humana y confirmación del flujo

Figura 9

Ejemplo anonimizado de confirmación humana y registro automatizado en agenda

3. Síntesis de la comparación operativa

La evidencia revisada permitió comparar dos condiciones de operación. En el proceso manual, la primera respuesta dependía de la revisión de correos y podía variar entre algunas horas y varias semanas. En el proceso automatizado, la solicitud quedó registrada desde el inicio, se notificó al coordinador en una ventana controlada y se conservó trazabilidad mediante folio y referencia de agenda.

https://doi.org/10.22201/dgtic.30618096e.2026.4.3.186 **Vol. 4, Núm. 3.** julio-septiembre 2026, págs. 105 - 105

La comparación no tuvo como propósito sustituir la decisión del coordinador, sino reducir actividades repetitivas alrededor de esa decisión: búsqueda de mensajes, revisión inicial de datos, aviso de solicitudes pendientes, seguimiento de estado y registro posterior. Esta mejora fue relevante porque la coordinación de visitas guiadas no correspondía a un recurso dedicado exclusivamente a dicha actividad, sino a una función complementaria realizada por personal técnico académico junto con sus actividades principales. En este sentido, la automatización favoreció la continuidad del servicio sin que el personal asignado descuidara las funciones técnicas y académicas para las cuales fue contratado o designado.

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