Vol. 4, Núm. 1. enero-marzo **2026**
DOI:10.22201/dgtic.30618096e.2026.4.1

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.2026.4.1

Vol. 4, Núm. 1. enero-marzo 2026

Cuadernos Técnicos Universitarios de la DGTIC

DOI: 10.22201/dgtic.30618096e.2026.4.1

Vol. 4, Núm. 1. enero-marzo 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* (1), páginas (N1-N2). https://doi.org/10.22201/dgtic.30618096e.2026.4.1

Cuadernos Técnicos Universitarios de la DGTIC

DOI: 10.22201/dgtic.30618096e.2026.4.1

Vol. 4, Núm. 1. enero-marzo 2026

CUADERNOS TÉCNICOS UNIVERSITARIOS DE LA DGTIC, Año 4, No. 1, enero-marzo 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 febrero de 2026.

Cuadernos Técnicos Universitarios de la DGTIC

DOI: 10.22201/dgtic.30618096e.2026.4.1

Vol. 4, Núm. 1. enero-marzo 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 • Lilia Nataly Vaca Tapia, Jefa de Gestión de Revistas Académicas • Jorge Pérez García, Jefe del Departamento de Soporte Técnico de Sistemas Editoriales • Juan Manuel Rodríguez Martínez, Jefe de Desarrollo • Victor Daniel Haro Gómez, Diseñador web • Jaqueline Segura Bautista, Gestión de recursos.

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.2026.4.1

Vol. 4, Núm. 1. enero-marzo 2026

Cuadernos Técnicos Universitarios de la DGTIC

DOI: 10.22201/dgtic.30618096e.2026.4.1 Vol. 4, Núm. 1. enero-marzo 2026

CONTENIDO

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.148

Vol. 4, Núm. 1. enero-marzo 2026, págs. 8 - 18

Integración de pruebas visuales automatizadas con Applitools en el proceso de pruebas funcionales

Integration of automated visual testing with Applitools in the functional testing process

Información del reporte:

Para citar este reporte técnico:

Rangel Cano, L. (2026). Integración de pruebas visuales automatizadas con Applitools en el proceso de pruebas funcionales. *Cuadernos Técnicos Universitarios de la DGTIC, 4* (1), páginas (8 - 18). https://doi.org/10.22201/dgtic.30618096e.2026.4.1.148

Liliana Rangel Cano

Dirección General de Cómputo y de Tecnologías de Información y Comunicación

Universidad Nacional Autónoma de México

lilianarc@unam.mx

ORCID: 0009-0001-4100-4011

Resumen

En el marco de las pruebas funcionales de un sistema de información, se llevaron a cabo pruebas visuales con Applitools, herramienta de pruebas visuales automatizadas que facilita la verificación de interfaces gráficas en múltiples navegadores, dispositivos y resoluciones. Ésta utiliza inteligencia artificial para detectar diferencias visuales. La metodología aplicada permitió confirmar su efectividad como apoyo en el proceso de pruebas, particularmente en la verificación visual y en pruebas de regresión sobre interfaces gráficas. La herramienta permite comparar versiones del software o sitios web con información estática, pero su uso en aplicaciones con contenido dinámico requiere mayor personalización para evitar falsos positivos. Applitools genera líneas base de comparación específicas para cada ambiente, aunque se puede configurar una línea base unificada para facilitar la detección de inconsistencias visuales derivadas por diferencias en la renderización entre navegadores. Su implementación implica una curva de aprendizaje y demanda recursos tecnológicos adecuados, además de considerar limitaciones de la versión gratuita. La interpretación de los resultados depende del criterio del probador, quien determina si las diferencias corresponden a errores o cambios esperados. Una vez superada la curva de aprendizaje, su uso contribuye a reducir significativamente los tiempos de ejecución en comparación con las pruebas manuales. Asimismo, es necesario considerar aspectos

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.148 **Vol. 4, Núm. 1.** enero-marzo 2026, págs. 9 - 18

de seguridad de la información, la reutilización de la automatización y la conveniencia de adquirir una licencia institucional para su uso.

Palabras clave: Applitools, calidad del software, pruebas funcionales de caja negra, pruebas visuales.

Abstract

Within the framework of functional testing of an information system, visual tests were carried out using Applitools, an automated visual testing tool that facilitates the verification of graphical user interfaces across multiple browsers, devices, and resolutions. This tool uses artificial intelligence to detect visual differences. The applied methodology confirmed its effectiveness in supporting the testing process, particularly for visual verification and regression testing of graphical interfaces. The tool enables the comparison of software versions or websites with static information; however, its use in applications with dynamic content requires additional customization to avoid false positives. Applitools generates specific baselines for each environment, although a unified baseline can be configured to facilitate the detection of visual inconsistencies resulting from differences in browser rendering. Its implementation involves a learning curve and requires adequate technological resources, as well as consideration of the limitations of the free version. The interpretation of the results depends on the tester's judgment, who determines whether the differences correspond to errors or expected changes. Once the learning curve is overcome, its use significantly reduces execution times compared to manual testing. Furthermore, it is necessary to consider information security, the reusability of the automation, and the advisability of acquiring an institutional license for its use.

Keywords: *Applitools, black box functional testing, software quality, visual testing.*

1. INTRODUCCIÓN

Las pruebas funcionales son fundamentales para verificar que los sistemas de información cumplan con los requerimientos establecidos y respondan correctamente a las necesidades de los usuarios. Dentro de este proceso, se verifica tanto la funcionalidad como el diseño gráfico del software en distintos navegadores y tamaños de pantalla. Esto garantiza la correcta visualización y satisfacción del usuario al utilizar las diversas opciones disponibles en diferentes sistemas operativos, resoluciones y ambientes tecnológicos (Pressman & Maxim, 2020; Sommerville, 2016).

El *International Software Testing Qualifications Board* (ISTQB) define las pruebas funcionales como aquellas que "se centran en las funciones que realiza un sistema o componente, mediante la verificación de que la salida producida por el software coincide con la esperada, basada en las especificaciones funcionales" (ISTQB, 2024).

Como parte del proceso de pruebas funcionales, se realizan revisiones visuales de la interfaz de usuario, una tarea que demanda tiempo y recursos considerables. Para optimizar esta actividad, se evaluó la automatización de la validación visual, considerando a Applitools como una alternativa de herramienta tecnológica (Myers, Sandler & Badgett, 2011; Applitools, 2025).

Applitools es una plataforma especializada en pruebas visuales automatizadas para aplicaciones web, móviles y de escritorio. Utiliza inteligencia artificial para comparar las interfaces de una aplicación y detectar diferencias visuales no deseadas, como desplazamientos, cambios en color, alineación o tamaño

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.148 Vol. 4, Núm. 1. enero-marzo 2026, págs. 10 - 18

de elementos, sin necesidad de definir selectores o ubicaciones específicas en el código (Applitools, 2025; Nguyen, 2023).

Las pruebas visuales se definen como un proceso mediante el cual se comparan imágenes o representaciones gráficas de una interfaz de usuario con una versión base o esperada, con el fin de identificar diferencias que puedan afectar la experiencia del usuario (Myers, Sandler & Badgett, 2011). Se consideran un complemento a las pruebas funcionales, ya que garantizan la coherencia y consistencia estética de la interfaz, además de su correcto comportamiento (Applitools, 2025).

Con el objetivo de determinar el grado en que esta herramienta puede apoyar las actividades sustanciales del proceso de pruebas de software, se implementó una prueba piloto en un sistema de información institucional. La funcionalidad evaluada consistió en el acceso a tres formularios de captura de datos disponibles para usuarios de la comunidad universitaria y público externo. Dos formularios solicitan los mismos tres datos, mientras que el tercero requiere únicamente dos de ellos e incluye una funcionalidad que modifica la distribución de los campos. Una vez ingresada la información, el sistema valida los datos y, en caso de ser correctos, despliega un segundo formulario para capturar información complementaria. Por el alcance específico de la funcionalidad evaluada, no se considera necesario profundizar en los detalles del sistema utilizado.

Dado que Applitools, al basarse en Inteligencia Artificial, tiene riesgos inherentes asociados al manejo de información, la prueba se enfocó exclusivamente en la comparación visual de formularios, sin capturar ni almacenar datos sensibles, con lo que se garantiza la confidencialidad y el cumplimiento de políticas de privacidad institucionales (UNAM, 2023; UNESCO, 2021).

En cumplimiento a las recomendaciones emitidas por la UNAM sobre el uso ético de inteligencia artificial generativa en la docencia, se atendieron los lineamientos que enfatizan la transparencia en el uso de datos, el control por parte de los usuarios y la recopilación mínima necesaria de información, con lo que se respetan los principios éticos y de protección de la dignidad humana (Grupo Académico de Inteligencia Artificial Generativa en Educación, 2023; UNESCO, 2021).

Asimismo, el documento “Recomendaciones para el uso educativo de la Inteligencia Artificial Generativa en la UNAM” (Grupo Académico de Inteligencia Artificial Generativa en Educación, 2025) destaca que la incorporación adecuada de herramientas basadas en IA requiere identificar procesos susceptibles de mejora y promover un uso crítico, responsable y experimental, derivado de experiencias individuales e institucionales. En ese sentido, la evaluación de Applitools se enmarca en un ejercicio controlado y ético para mejorar procesos de aseguramiento de calidad.

2. DESARROLLO TÉCNICO

Las pruebas visuales dentro del contexto de las pruebas funcionales cumplen una función complementaria, ya que aseguran que la presentación gráfica y la interfaz de usuario sean coherentes, estables y libres de defectos visuales. Detectan inconsistencias como desalineaciones, cambios inadvertidos en el diseño, o diferencias en la renderización entre navegadores que podrían afectar la experiencia del usuario.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.148 Vol. 4, Núm. 1. enero-marzo 2026, págs. 11 - 18

2.1 METODOLOGÍA

La metodología empleada, representada en la Figura 1, está basada en enfoques recomendados por estándares de gestión de proyectos y pruebas de software. Ésta se estructuró en cuatro fases: planeación, preparación del ambiente, ejecución, así como análisis y evaluación (Project Management Institute, 2021; ISTQB, 2024). Cada fase estuvo orientada a verificar la utilidad, precisión y aplicabilidad de la herramienta en escenarios reales de pruebas funcionales y visuales.

Figura 1

Metodología aplicada en la prueba piloto del uso de Applitools

Fase 1. Planeación

La planeación de la prueba piloto para revisar la funcionalidad de Applitools fue tratada como un proyecto, donde se definieron objetivos, alcance, recursos y entregables (Project Management Institute, 2021). Sin embargo, dadas las limitaciones de recursos, esta planeación no incluyó un calendario exhaustivo ni una gestión detallada de riesgos, enfocándose principalmente en establecer objetivos específicos, criterios de éxito y métricas de evaluación.

Los objetivos específicos definidos fueron:

I. Determinar el costo-beneficio en cuestión de tiempo de ejecución respecto a las pruebas manuales.

II. Evaluar la precisión de la herramienta en la detección de diferencias visuales.

III. Analizar la viabilidad técnica y ética para integrar Applitools en las actividades de pruebas funcionales.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.148 Vol. 4, Núm. 1. enero-marzo 2026, págs. 12 - 18

Debido a la demanda, la mayoría de los proyectos evaluados por el equipo de trabajo de pruebas de software corresponden a sistemas de información, por lo que se delimitó el alcance técnico a la comparación visual en un sistema institucional con funcionalidad genérica y se excluyó la captura o almacenamiento de información sensible o confidencial. Se seleccionó un módulo representativo sin funcionalidad de autenticación, con formularios que presentan variaciones estructurales y de contenido, con el fin de evaluar la capacidad de Applitools para detectar diferencias visuales en diseño, espaciado y disposición de campos, aspectos cruciales para la experiencia del usuario (Myers et al., 2011).

La prueba piloto se integró al flujo habitual del proceso de pruebas funcionales, se le asignó a un probador la aplicación de la prueba en su equipo de trabajo regular, se le proporcionaron 24 horas adicionales para actividades de automatización, ejecución y análisis de resultados. La capacitación en Applitools no formó parte del piloto, ya que el *tester* contaba con conocimientos básicos previos.

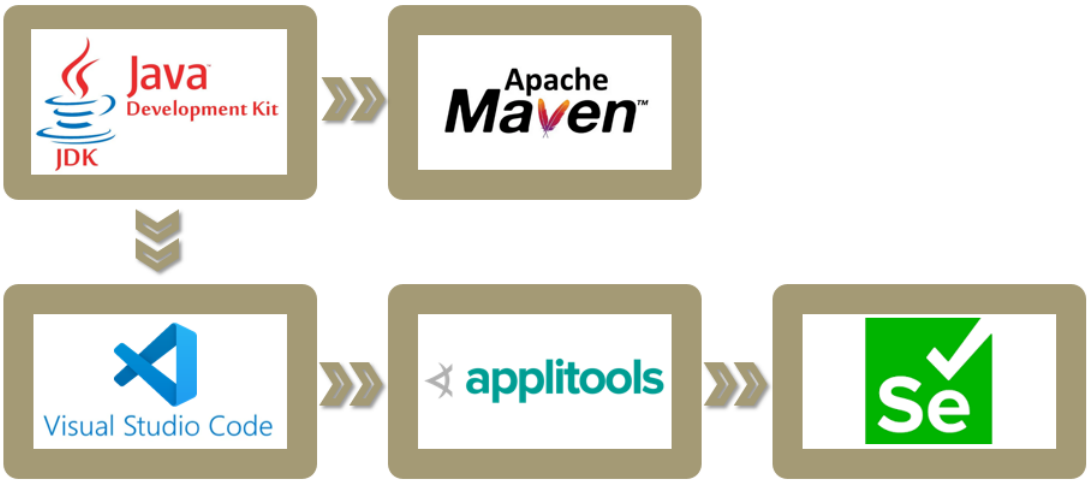
Conforme a buenas prácticas internacionales, se establecieron criterios y métricas de evaluación para decidir sobre la adopción de Applitools. Además, se diseñaron casos de prueba específicos para validar funcionalidades, que incluyen la comparación entre diferentes formularios del mismo sistema, del mismo formulario en distintos tamaños de pantalla y de la interfaz en diversos navegadores.

Fase 2. Preparación del ambiente

Para garantizar la validez técnica y la reproducibilidad de los resultados de la prueba piloto, se configuró un ambiente controlado y se instalaron los componentes mínimos requeridos en sus versiones más recientes. Se aseguró la compatibilidad con las librerías y dependencias más actuales (ISTQB, 2024; SeleniumHQ, 2022). La Figura 2 presenta el software que compone el ambiente para utilizar Applitools.

Figura 2

Ambiente con los recursos de software



Posteriormente, se configuraron las variables de entorno y las dependencias del proyecto, se incluyó la definición de la clave API de Applitools y la actualización de los archivos de configuración de Maven.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.148 Vol. 4, Núm. 1. enero-marzo 2026, págs. 13 - 18

Como buena práctica, se verificó la conectividad con los servicios externos (Applitools Eyes Server, repositorios Maven y el navegador).

Durante esta fase, se identificaron errores derivados de actualizaciones en el SDK de Applitools, que modificaron la forma de definir las áreas de captura, lo cual requirió ajustes para estabilizar la automatización. Este hallazgo destacó un riesgo técnico de compatibilidad de versiones, que debe considerarse en futuras implementaciones.

Se estableció la *baseline* visual o imagen de referencia, la cual corresponde al estado esperado de un formulario del sistema.

Fase 3. Ejecución

En esta fase, se desarrollaron y ejecutaron los casos de prueba automatizados con la herramienta Applitools. El proceso inició al retomar la *baseline* visual, que sirvió como punto de comparación para detectar diferencias.

Las pruebas consistieron en las siguientes tareas estructuradas:

I. Ajustar el caso de prueba

De acuerdo con el propósito de la herramienta Applitools, una vez automatizada la prueba, ésta debería ejecutarse de forma repetible sin necesidad de modificaciones, incluso ante ajustes menores en el código fuente o el ambiente visual (Applitools, 2025). No obstante, en el contexto de la prueba piloto, fue necesario realizar ciertos ajustes manuales para verificar el comportamiento de la herramienta bajo condiciones diversas, tales como cambiar la URL del formulario a evaluar, modificar los navegadores o el tamaño de la pantalla para validar la consistencia visual, e incorporar datos dinámicos para evitar inconsistencias. Estas configuraciones permitieron simular escenarios, considerando la diversidad de ambientes en los que los usuarios finales interactúan con los sistemas institucionales.

II. Ejecutar el caso de prueba

Se enviaron las ejecuciones al panel de Applitools, donde el motor de validación visual asistido por inteligencia artificial procesó las capturas y comparó con la *baseline* visual. El enfoque de Applitools combina *machine learning* y análisis perceptual, lo que reduce falsos positivos y mejora la eficacia respecto a herramientas tradicionales (Myers, Sandler & Badgett, 2011; Applitools, 2025). Como resultado de las ejecuciones, Applitools presentó un informe visual con las diferencias detectadas. Esta funcionalidad permitió una evaluación objetiva y reproducible de la interfaz gráfica.

III. Analizar las diferencias visuales

Una vez obtenidos los resultados, el probador fue responsable de analizar las diferencias detectadas por Applitools. Se observó que la herramienta puede identificar una gran cantidad de diferencias, incluyendo los cambios esperados, lo cual requiere una revisión cuidadosa por parte del especialista en pruebas para distinguir entre defectos reales y variaciones por los cambios aplicados.

IV. Determinar la causa de las diferencias

Finalmente, el probador evaluó si las diferencias eran esperadas (por ejemplo, asociados a los cambios en el contenido) o no esperadas (errores visuales o inconsistencias). Esta tarea, aunque automatizada parcialmente, puede demandar más tiempo de análisis que una revisión manual cuando se presentan

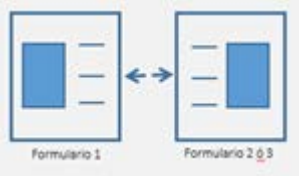
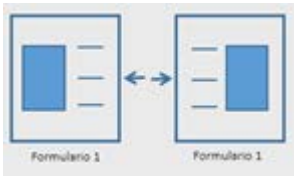
https://doi.org/10.22201/dgtic.30618096e.2026.4.1.148 Vol. 4, Núm. 1. enero-marzo 2026, págs. 14 - 18

numerosas diferencias, especialmente si la interfaz es dinámica o dependiente de datos variables. Por lo anterior, al analizar los resultados, el probador generó ajustes en la automatización del caso de prueba para que, al aplicar nuevamente la prueba, Applitools eliminará inconsistencias inexistentes.

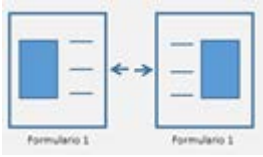
En la Tabla 1 se presentan los resultados obtenidos en las pruebas aplicadas:

Tabla 1

Resultados obtenidos en las pruebas realizadas

Prueba	Resultados	
	Esperados	Obtenidos
Comparar diferentes formularios para conocer la funcionalidad de la herramienta. 	Se espera que Applitools reconozca consistentemente los mismos elementos visuales en los diferentes formularios evaluados. Applitools genera una línea base (baseline) a partir de las imágenes de referencia de los formularios. Los recuadros rojos destacan las diferencias visuales identificadas por el usuario entre el baseline y la versión actual del formulario. 	Applitools identificó las diferencias en los textos y las resaltó en ambas pantallas con un sombreado rosa. Como parte de la prueba, se configuró la automatización para ignorar los desfases de posición, por lo que no se mostraron diferencias en ese aspecto, como en el caso del recuadro de solicitud de datos. 
Comparar un mismo formulario en diferentes tamaños de pantalla para determinar las diferencias que presenta la herramienta. 	Se espera que Applitools reconozca consistentemente los elementos en diferentes tamaños de pantalla, a partir de la línea base, y marcar únicamente las diferencias entre elementos.	A partir de la línea base generada por Applitools para un tamaño de pantalla específico, se identificaron diferencias al comparar con otros tamaños, las cuales fueron resaltadas mediante un sombreado en color rosa. Al realizar la comparación con un tamaño distinto, prácticamente la totalidad de la pantalla fue marcada como diferente. 

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.148 Vol. 4, Núm. 1. enero-marzo 2026, págs. 15 - 18

Prueba	Resultados	
	Esperados	Obtenidos
Comparar un mismo formulario en diferentes navegadores respecto a uno establecido para determinar las diferencias que presenta la herramienta. 	Se espera que Applitools reconozca consistentemente los elementos en diferentes navegadores, a partir de la línea base, y marcar únicamente las diferencias entre elementos.	A partir de la línea base generada por Applitool, no se detectaron diferencias entre navegadores cuando se mantuvo el mismo tamaño de pantalla; en caso contrario, se presentaron variaciones en la distribución de los elementos. 

Durante la ejecución, se identificaron aspectos positivos de la herramienta, como la posibilidad de realizar múltiples comparaciones en una sola ejecución, la capacidad de agrupar pruebas en lotes, lo que facilita la gestión de resultados, ignorar diferencias causadas por desplazamientos y la flexibilidad en la generación de la *baseline* visual del área visible en pantalla o de la página completa al desplazarse de manera vertical. Éstos elementos confirman que Applitools es una herramienta robusta para la automatización de pruebas visuales en escenarios donde la interfaz gráfica es crítica para la calidad.

Fase 4. Análisis y evaluación

Finalmente, se realizó un análisis entre los resultados obtenidos mediante Applitools y los derivados de las pruebas manuales tradicionales. Esta fase tuvo como objetivo determinar la viabilidad técnica y operativa para incorporar la herramienta dentro del proceso institucional de pruebas de software. Se evaluó su precisión, eficiencia y aplicabilidad en distintos escenarios.

En la Tabla 2 se presenta la comparativa del tiempo invertido en la prueba automatizada respecto a la manual, así como los recursos de software necesarios.

Tabla 2

Comparativa del tiempo invertido y recursos necesarios en la prueba automatizada respecto a la manual

Elemento de evaluación	Prueba	
	Manual	Automatizada
	Tiempo invertido	
Automatización	No aplica	La automatización del caso de prueba: 2 h

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.148 Vol. 4, Núm. 1. enero-marzo 2026, págs. 16 - 18

Elemento de evaluación	Prueba	
	Manual	Automatizada
Tiempo invertido		
Ajustes aplicados en cada prueba	No aplica	Ajustes por cada prueba: de 5 a 30 min (depende del ajuste a realizar)
Aplicación de la prueba Identificación de defectos asociados al diseño	Por cada comparación entre 2 navegadores o tamaños: 10 min Consideraciones: La identificación de defectos depende tanto de la experiencia del probador como de su nivel de observación.	Inmediato Consideraciones: La identificación de diferencias es exacta, sin embargo, determinar si es algo esperado o no depende del probador.
Recursos de software		
Consumo de recursos	Navegadores y emuladores de dispositivos o herramientas para desarrolladores de Chrome se simularon diferentes tamaños de pantalla	Java Development Kit (JDK), para ejecutar scripts de prueba. Maven o Gradle, administrador de dependencias del proyecto. IDE de desarrollo (IntelliJ IDEA, Visual Studio Code o Eclipse), ambiente para codificación e integración. Applitools SDK, biblioteca para integrar Applitools con frameworks de prueba. Framework de automatización (Selenium, Cypress, Playwright o WebdriverIO), herramienta base para ejecutar pruebas funcionales. Cuenta en Applitools Cloud (API Key activa), plataforma principal para almacenar, analizar y visualizar resultados. Dashboard de Applitools Eyes (Accesible desde navegador -Chrome, Edge, Firefox-), interfaz web para revisión y gestión de resultados visuales.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.148 Vol. 4, Núm. 1. enero-marzo 2026, págs. 17 - 18

3. RESULTADOS

Applitools se enfoca en comparar una imagen o un conjunto de imágenes base en diferentes navegadores y resoluciones respecto a sus diversas versiones, conforme evoluciona el *software*. Permite comparar pantallas con funcionalidad similar mediante la automatización de casos de prueba, centrándose en la revisión de regiones específicas (elementos fijos de pantalla) como encabezados, menús y títulos, e ignorando diferencias en datos dinámicos o en desplazamientos. Al comparar una imagen base con distintos tamaños de pantalla o diferentes formularios produce inconsistencias causadas por diferencias estructurales y de contenido.

Por defecto, Applitools crea una línea base distinta para cada navegador y resolución (además de diferentes tamaños que consideren o no un diseño responsivo), aunque es posible unificar estas referencias visuales para todos los ambientes. Resultó muy efectiva para comparar una misma interfaz entre diferentes navegadores, siempre que se mantengan la resolución y el tamaño de pantalla. Se recomienda generar un conjunto de imágenes base en distintos navegadores y resoluciones para compararlas durante las pruebas de regresión, las cuales verifican qué cambios en el software no introducen errores en funcionalidades previamente validadas (ISTQB, 2024).

Por lo anterior, se concluye que la herramienta resulta especialmente útil en pruebas de regresión cuando se mantiene el mismo contexto, usuario, elementos de pantalla y volumen de información. También es adecuada para revisar sitios y portales web. No obstante, puede requerir personalización al comparar funcionalidades con contenido dinámico, como carruseles, noticias o elementos de interacción con los usuarios. Asimismo, puede integrarse en pruebas funcionales de sistemas de información, ya que permite comparar diferentes resultados esperados según las entradas ingresadas. Sin embargo, su uso exige un conocimiento previo del sistema y una inversión considerable de tiempo en la automatización.

En cuanto al tiempo total de ejecución y revisión, se observó un incremento inicial debido a la curva de aprendizaje y la necesidad de ajustar la automatización para los casos de prueba específicos. Es importante mencionar que, para probadores con menor experiencia, la curva de aprendizaje puede ser significativa. Es fundamental contar con equipos de cómputo adecuados (Procesador Intel i5 o superior, 8 GB RAM, 10 GB de espacio libre) que aseguren pruebas fluidas.

Una vez comprendido su alcance y uso, las pruebas automatizadas con Applitools reducen significativamente el tiempo invertido respecto a las pruebas manuales, especialmente al ejecutar múltiples validaciones visuales simultáneamente y automatizar la detección de inconsistencias. Es posible realizar pruebas en paralelo en distintos navegadores y dispositivos, lo que amplía la cobertura.

Aunque Applitools genera comparaciones entre pantallas en diferentes navegadores y resoluciones, la interpretación final de las diferencias recae en el probador, quien debe determinar si se trata de errores o de cambios esperados.

4. CONCLUSIONES

La metodología aplicada permitió validar de manera objetiva la efectividad de Applitools como herramienta de apoyo en las actividades del proceso de pruebas, destacando su valor en la verificación visual.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.148 Vol. 4, Núm. 1. enero-marzo 2026, págs. 18 - 18

Applitools resulta particularmente viable y eficiente para pruebas de regresión en ambientes con interfaces gráficas estables, facilita la detección precisa de diferencias entre versiones o en la revisión de sitios y portales web con información estática.

La aplicación de pruebas visuales en sistemas de información con contenido dinámico presenta retos importantes, que requieren una planificación detallada, un manejo controlado de las líneas base y una capacitación técnica adecuada para maximizar su eficacia.

Debido a que el uso de Applitools se hace en la nube, se recomienda extremar precauciones respecto a las imágenes a generar, con el fin de evitar la captura de información sensible y confidencial. Por lo anterior, se sugiere el uso de datos de prueba para garantizar la privacidad durante las pruebas de regresión.

La creación de casos de prueba bien diseñados permitirá la reutilización de la automatización en distintas secciones y sistemas, lo que reducirá los tiempos de generación y ejecución en general.

Para incorporar la herramienta en las actividades de pruebas de software, es importante considerar que la versión gratuita de Applitools está limitada tanto en funcionalidad como en número de ejecuciones; por ello, para su uso en todos los proyectos, se observa la necesidad de evaluar la adquisición de una licencia de pago.

REFERENCIAS

Applitools. (2025). *AI-powered Visual Testing Overview*. Recuperado de https://applitools.com

Grupo Académico de Inteligencia Artificial Generativa en Educación UNAM. (2025). *Recomendaciones para el uso educativo de la inteligencia artificial generativa en la UNAM*. Coordinación de Evaluación, Innovación y Desarrollo Educativos (CEIDE). https://www.ceide.unam.mx/index.php/recomendaciones-para-el-uso-educativo-de-la-inteligencia-artificial-generativa-en-la-unam/

International Software Testing Qualifications Board (ISTQB). (2024). *ISTQB glossary* (Versión 4.6.2). Recuperado de https://glossary.istqb.org/en_US/search?term=&exact_matches_first=true

Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing* (3rd ed.). John Wiley & Sons.

Pressman, R. S., & Maxim, B. R. (2020). *Ingeniería de Software*: Un enfoque práctico (9a ed.). McGraw-Hill.

Project Management Institute. (2021). *A guide to the project management body of knowledge* (PMBOK® Guide) (7ª ed.). Project Management Institute.

Selenium HQ. (2022). *WebDriver* [Documentación]. Recuperado de https://www.selenium.dev/documentation/webdriver/

UNESCO. (2021). *Recommendation on the Ethics of Artificial Intelligence* (SHS/BIO/PI/2021/1). Paris: UNESCO. https:// unesdoc.unesco.org/ark:/48223/pf0000381137

Universidad Nacional Autónoma de México. (2023, Octubre). *Recomendaciones para el uso de la inteligencia artificial generativa en la docencia* [PDF]. Recuperado de https://cuaed.unam.mx/descargas/recomendaciones-uso-iagen-docencia-unam-2023.pdf

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 19 - 36

Modelo de infraestructura tecnológica para la gestión y resguardo de información académica: experiencia de implementación en la UASA

Technological infrastructure model for the management and safeguarding of academic information: implementation experience at UASA

Información del reporte:

Para citar este reporte técnico:

Villarreal Brito, G. (2026). Modelo de infraestructura tecnológica para la gestión y resguardo de información académica: experiencia de implementación en la UASA. *Cuadernos Técnicos Universitarios de la DGTIC, 4* (1), páginas (19 - 36). https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147

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

La gestión eficiente y segura de la información académica representó un reto creciente en la Unidad Académica de Sistemas Arrecifales debido al aumento en la generación de datos científicos y administrativos. Con el fin de responder a esta necesidad, se diseñó e implementó un modelo de infraestructura tecnológica orientado a fortalecer el resguardo, la continuidad operativa y la colaboración académica. El modelo se basó en la integración de tres componentes principales: una plataforma de colaboración en nube privada con software libre, un sistema de almacenamiento en red con redundancia y una herramienta de monitoreo de servicios. La metodología consideró diseño, configuración, instalación y pruebas en condiciones de uso real. Se demostró la viabilidad de un modelo replicable, contribuyendo a la soberanía de datos y autonomía operativa, así como a la resiliencia en la gestión de datos académicos.

Palabras clave: Almacenamiento en red, gestión de datos académicos, infraestructura tecnológica, monitoreo de servicios, nube privada.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 **Vol. 4, Núm. 1.** enero-marzo 2026, págs. 20 - 36

Abstract

The efficient and secure management of academic information represented an increasing challenge at the Reef Systems Academic Unit, due to the growth in scientific and administrative data generation. To address this need, a technological infrastructure model was designed and implemented to strengthen data safeguarding, operational continuity, and academic collaboration. The model was based on the integration of three main components: a private cloud collaboration platform using open-source software, a redundant network storage system, and a service monitoring tool. The methodology considered design, configuration, installation, and testing under real operating conditions. The results showed improved information access times, reduced failure risks, and early incident detection. The feasibility of a replicable model was demonstrated, contributing to institutional data sovereignty and resilience in academic information management.

Keywords: *Academic data management, network storage, private cloud, service monitoring, technological infrastructure.*

1. INTRODUCCIÓN

La gestión de la información académica se ha consolidado como un eje estratégico en las instituciones de educación superior, donde la seguridad, la continuidad operativa y la soberanía de los datos deben abordarse de manera integral junto con la colaboración y el acceso a la información (Guerschberg, 2025). Sin embargo, los servicios comerciales de almacenamiento en la nube presentan limitaciones relacionadas con el control institucional, la privacidad y los costos a largo plazo (Cabral Vargas, 2018).

Durante el periodo 2024-2025, la Unidad Académica de Sistemas Arrecifales (UASA) del Instituto de Ciencias del Mar y Limnología de la Universidad Nacional Autónoma de México implementó un modelo de infraestructura tecnológica orientado a integrar almacenamiento seguro, servicios de colaboración y monitoreo en tiempo real, con el fin de garantizar la disponibilidad, integridad y resguardo de los datos utilizados en actividades de investigación, docencia y administración.

La solución se fundamenta en el uso de software libre y hardware institucional, fortaleciendo la autonomía en la gestión de la información y la sostenibilidad operativa de la infraestructura. El Anexo B presenta el análisis comparativo de herramientas colaborativas que sustenta la selección de la plataforma implementada, mientras que el Anexo C incluye el análisis del sistema de almacenamiento en red y su integración con la plataforma de monitoreo de infraestructura.

El objetivo fue implementar un modelo de infraestructura tecnológica que integre Nextcloud, Synology NAS y Zabbix[1], con el propósito de optimizar la gestión, intercambio y resguardo de información académica, mejorar los tiempos de acceso y disponibilidad del servicio, así como garantizar la continuidad operativa mediante monitoreo y respaldo automatizado.

2. DESARROLLO TÉCNICO

La implementación del modelo de infraestructura tecnológica se estructuró en cuatro etapas: diseño, configuración, instalación y pruebas. Durante el diseño, se evaluaron enfoques relacionados con la

1 Para la definición de éste y otros conceptos técnicos, véase el glosario contenido en el Anexo A.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 21 - 36

transformación digital de las universidades, donde la gestión de información y los datos se han convertido en recursos estratégicos (Komljenovic, 2024). Se consideraron servicios comerciales de nube (Google Workspace, Dropbox), que ofrecen facilidad de administración y alta disponibilidad, pero presentan limitaciones en el control institucional, la privacidad de los datos y costos a largo plazo (Cabral Vargas, 2018).

I. Plataformas libres como Seafile y ownCloud, que proporcionan independencia tecnológica, aunque con menor madurez comunitaria y soporte limitado.

II. Soluciones híbridas basadas en Nextcloud y almacenamiento NAS, que equilibran seguridad, soberanía tecnológica y escalabilidad (Aydın, 2021).

Tras el análisis comparativo del Anexo B, se eligió la opción híbrida por ser la más adecuada para el entorno académico de la UASA, en el que la protección de datos, la interoperabilidad y la autonomía institucional constituyen prioridades estratégicas. Este enfoque coincide con experiencias documentadas en otras universidades, que demuestran la viabilidad de las nubes privadas para mantener el control de la información, reducir la dependencia tecnológica y garantizar la continuidad operativa, como señalan Haurech (2020), Mero-Terán & Macías-Arias (2025).

2.1 INSTALACIÓN E INTEGRACIÓN DE NEXTCLOUD Y SYNOLOGY NAS

La infraestructura se implementó sobre un hipervisor VMware en su versión gratuita, disponible previamente en la unidad, el cual aloja múltiples máquinas virtuales dedicadas a servicios institucionales. Entre ellas, se encuentra la máquina virtual de Nextcloud, configurada con ocho unidades de procesamiento virtual (vCPU), 10 GB de memoria RAM y discos virtuales sobre un sistema operativo Ubuntu Server. Con el fin de garantizar eficiencia y estabilidad en las comunicaciones, se implementó una segmentación de red virtual mediante conmutadores virtuales (vSwitch, virtual switch; conmutador virtual) asociados a adaptadores físicos (vmnic, *virtual machine network interface*; interfaz de red de máquina virtual), permitiendo aislar el tráfico administrativo, de almacenamiento y de usuarios finales.

Para la instalación de Nextcloud, se utilizó la imagen oficial, siguiendo las recomendaciones del Nextcloud Server Administration Manual (Nextcloud Server,2025). El entorno se configuró con Apache como servidor web y PostgreSQL como motor de base de datos, logrando una operación estable e integrándose con servicios institucionales como grupos académicos, almacenamiento compartido, monitoreo de infraestructura y procesos de respaldo automatizado.

La integración con el sistema de almacenamiento se realizó mediante una NAS Synology DS1821+, configurada con ocho discos de 14 TB en un arreglo RAID 5 (arreglo redundante de discos independientes), proporcionando una capacidad útil aproximada de 84 TB, destinada al almacenamiento de información académica de usuarios. La comunicación se estableció a través del protocolo NFS (*Network File System*) versión 4.1, mediante un vSwitch dedicado, lo que permitió reducir la congestión de la red principal y mejorar la seguridad y estabilidad del entorno. Los volúmenes se exportaron desde el sistema DSM (*DiskStation Manager*) y se montaron de forma persistente en el servidor Nextcloud mediante el archivo /etc/fstab.

El directorio de datos de usuarios se reconfiguró para enlazarse directamente al almacenamiento NAS, aprovechando un esquema centralizado y redundante, con parámetros optimizados descritos en el Anexo D. Se calibraron opciones como escritura síncrona, tamaño de bloque de 256 KiB, conexiones

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 22 - 36

paralelas y desactivación de actualizaciones de metadatos, lo que redujo significativamente la latencia en transferencias de archivos superiores a 10 GB. Adicionalmente, se separó el directorio temporal hacia un almacenamiento local en SSD, mejorando el rendimiento en operaciones intermedias y de procesamiento interno del sistema.

Monitoreo mediante Zabbix: Se instaló la plataforma de monitoreo Zabbix en un servidor alterno y el agente en el servidor Nextcloud con la habilitación del protocolo simple de administración de red (SNMP), permitiendo la integración con la plantilla oficial de Nextcloud en Zabbix. Como se muestra en el Anexo E, se habilitó el monitoreo continuo de uso de CPU, espacio en disco, carga de red, actividad de usuarios, cantidad de archivos sincronizados, entre otros parámetros que proporciona alertas preventivas y diagnósticos en tiempo real del estado del sistema.

Automatización y respaldo: Se configuraron *scripts* de respaldo incremental y procesos de verificación por *hash*, garantizando la integridad de los archivos copiados. El sistema ejecuta tareas programadas de respaldo y monitoreo, manteniendo la operación continua y auditabilidad de los servicios.

3. RESULTADOS TÉCNICOS DE LA INTEGRACIÓN

La integración de esta plataforma publicada en instancia institucional de nube privada (acceso restringido) se está consolidando como un eje central del sistema de colaboración y resguardo institucional, garantizando la disponibilidad, seguridad y accesibilidad de la información entre los distintos grupos de trabajo. Actualmente, el sistema cuenta con 56 cuentas activas, organizadas en grupos vinculados al instituto y sus diferentes unidades académicas, con cuotas individuales que oscilan entre 100 GB a 1 TB según su rol y requerimientos operativos.

Durante las pruebas con sistema de archivos compartidos en red (CIFS/SMB), se observaron bloqueos al transferir archivos de gran tamaño (>4 GB) desde diferentes sistemas a la plataforma de Nextcloud. El montaje mediante `mount.cifs` resultó inestable, quedando procesos colgados. Este caso evidenció la necesidad de evaluar otras rutas de transferencia más robustas para usuarios con cargas elevadas. En el Anexo F, se describen las formas de conexión y acceso, incluyendo los distintos métodos disponibles para la transferencia y sincronización de archivos: cliente web, cliente de escritorio y móvil, WebDAV, NFS, CIFS/SMB, rsync y SCP/SFTP. Este análisis comparativo permitió establecer los canales de acceso más adecuados para cada tipo de usuario, optimizando el rendimiento, la estabilidad de conexión y la eficiencia en el intercambio de información localmente o hacia internet.

Finalmente, es indispensable realizar optimizaciones de APCu y OPcache**,** configuradas a nivel del entorno PHP del servidor Nextcloud, mediante los archivos de configuración del intérprete PHP, junto con el uso de un directorio temporal local, descrito en el Anexo E, para soportar operaciones de lectura y escritura intensiva.

3.1 LIBERACIÓN DEL SERVICIO Y OPERACIÓN INICIAL

Tras la implementación técnica, se realizó la liberación formal del servicio mediante la notificación a los usuarios académicos a través de correo institucional y mensajería instantánea, incluyendo instrucciones

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 23 - 36

de acceso, manuales de uso y políticas del servicio. Se configuraron cuentas de usuario individuales con cuotas asignadas y se instruyó a los usuarios a realizar el cambio de contraseña desde la plataforma web.

Asimismo, se estableció un procedimiento de atención de incidentes, atendido por el responsable de tecnologías de la información de la unidad. El servicio fue evaluado inicialmente por un grupo piloto de aproximadamente 30 usuarios de perfiles académico, administrativo y técnico, lo que permitió realizar ajustes menores antes de su adopción generalizada, alcanzando posteriormente hasta 56 usuarios activos concurrentes.

3.2 ANÁLISIS DEL MONITOREO

El monitoreo continuo mediante Zabbix permitió evaluar de forma integral el desempeño de la plataforma Nextcloud, analizando los principales indicadores operativos del servidor, incluyendo uso de CPU (porcentaje de utilización de los núcleos virtuales asignados), consumo de memoria RAM (porcentaje respecto al total disponible), uso de almacenamiento, tráfico de red y número de usuarios activos (véanse Figuras 1 a 5, en el Anexo E). La utilización promedio de CPU se mantuvo cercana al 4 %**,** con picos de hasta 56 %**,** asociados principalmente a tareas de sincronización y respaldos automáticos. El uso de memoria permaneció estable alrededor del 22 % del total de 10 GB asignados**,** lo que refleja una gestión eficiente de procesos concurrentes.

En cuanto al almacenamiento, se registraron picos de latencia de lectura, medidos en milisegundos (ms)**,** de hasta aproximadamente 434 ms en momentos de alta demanda de operaciones de Entrada/Salida, lo que motivó la reprogramación de tareas de respaldo en horarios no críticos, optimizando la disponibilidad de la plataforma.

Respecto al monitoreo de red, los picos de transferencia alcanzaron aproximadamente 1.1 Mbps, medidos como tasa de tráfico saliente en la interfaz de red principal del servidor, sin incidencias de error ni pérdida de paquetes, confirmando la estabilidad del enlace de datos. La carga máxima del sistema (≈5.2), correspondiente al *load average* del servidor, se mantuvo dentro de los parámetros esperados para un entorno con 8 núcleos virtuales, y con hasta 56 usuarios activos concurrentes y más de 20 TB almacenados, sin observarse una degradación perceptible del servicio.

En términos de resultados técnicos, la infraestructura mostró una mejora aproximada del 30 % en los tiempos de acceso, respecto a la configuración inicial, derivada de las optimizaciones aplicadas en el servidor mediante APCu y OPcache como mecanismos de caché, así como el uso de un directorio temporal local.

4. CONCLUSIONES

El modelo de infraestructura tecnológica implementado en la UASA demostró que es posible gestionar información académica de forma eficiente y segura mediante la integración de software libre, almacenamiento en red con redundancia y monitoreo continuo, manteniendo el control institucional de los datos y reduciendo la dependencia de proveedores externos.

El uso de hardware institucional y plataformas de código abierto permitió optimizar los recursos de cómputo, almacenamiento y red, reflejándose en una mejora en los tiempos de acceso a la información, estimada a partir de la comparación de métricas de latencia y carga del sistema antes y después de

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 24 - 36

las optimizaciones, descritas en la sección de resultados y el Anexo D, sin afectar la disponibilidad ni la seguridad del servicio.

La adopción de prácticas de administración, respaldo y monitoreo permitió mantener la continuidad operativa del servicio bajo escenarios de alta demanda, con hasta 56 usuarios activos concurrentes y más de 20 TB de información académica, confirmando la viabilidad de un modelo replicable para instituciones de educación superior con requerimientos similares.

AGRADECIMIENTOS

Se agradece al Instituto de Ciencias del Mar y Limnología y a la Unidad Académica de Sistemas Arrecifales de la Universidad Nacional Autónoma de México por el respaldo institucional brindado para el desarrollo de esta solución tecnológica. Asimismo, al Servicio Académico de Monitoreo Meteorológico y Oceanográfico (https://sammo.icmyl.unam.mx/) por proporcionar el NAS, infraestructura tecnológica que dio soporte a esta solución, y al proyecto PAPIME PE207521 por su apoyo en el fortalecimiento de las capacidades tecnológicas. De igual manera, al Ing. Alfredo Landa Herrera por su asesoría y consultoría en la plataforma Nextcloud y al M. en C. Fernando Negrete Soto por su orientación académica durante el desarrollo del proyecto.

REFERENCIAS

Aydın, H. (2021). *A* study of cloud computing adoption in universities as a model for technology integration. *SAGE Open, 11*(3), 1–12. https://doi.org/10.1177/21582440211030280

Cabral Vargas, B. C. (2018). Consideraciones para el almacenamiento de archivos digitales en la nube informática en bibliotecas universitarias. *Investigación Bibliotecológica, 32*(74), 55-75. https://doi.org/10.22201/iibi.24488321xe.2018.74.57909

Guerschberg, L. (2025*).* Impacto de la soberanía digital en poblaciones vulnerables: el rol del software libre en educación para cerrar la brecha digital. *Sapiens Discoveries International Journal, 3*(1), 1–22.

Haurech, H. R. (2020). *Metodología para la selección de recursos computacionales gestionados con tecnologías de cloud computing en ambientes educativos* [Tesis de Maestría, Facultad de Ciencias Exactas, Químicas y Naturales, Universidad Nacional de Misiones]. https://rid.unam.edu.ar/bitstream/handle/20.500.12219/2764/Haurech%20HR_2020_Metodolog%C3%ADa%20para%20la%20selecci%C3%B3n.pdf

Komljenovic, J. (2024). Turning universities into data-driven organisations: seven propositions. *Higher Education, 1–20*. https://doi.org/10.1007/s10734-024-01277-z

Nextcloud Server. (2025). *Administration manual* (versión "latest"). Recuperado de https://docs.nextcloud.com/server/latest/admin_manual/

Mero-Terán L. F., & Macías-Arias E. J. (2025). Modelos tecnológicos de computación en la nube en la transformación digital de la educación superior: Una revisión sistemática de literatura. *593 Digital Publisher CEIT, 10*(1), 29-53.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 25 - 36

ANEXO A. GLOSARIO DE TÉRMINOS TÉCNICOS

APCu. Sistema de caché en memoria para aplicaciones PHP que mejora el rendimiento del servidor al reducir accesos repetitivos al disco y a la base de datos.

Hipervisor. Capa de software que permite ejecutar múltiples máquinas virtuales sobre un mismo hardware físico, asignando y aislando recursos de cómputo, memoria y red de forma controlada.

Monitoreo de infraestructura. Conjunto de prácticas y herramientas destinadas a supervisar de forma continua el estado, desempeño y disponibilidad de sistemas de cómputo, almacenamiento y red.

NAS (*Network Attached Storage*). Sistema de almacenamiento conectado a la red que centraliza grandes volúmenes de datos y permite el acceso concurrente desde múltiples servidores o estaciones de trabajo.

Nextcloud. Plataforma de colaboración y almacenamiento de archivos autoalojada que permite la sincronización, compartición y gestión de información académica en entornos institucionales, integrándose con distintos servicios y sistemas de autenticación.

NFS (*Network File System*). Protocolo de red que permite montar directorios remotos como sistemas de archivos locales, ampliamente utilizado en entornos Linux y Unix para almacenamiento centralizado.

NFS versión 4.1. Versión del protocolo NFS que incorpora mejoras en la gestión de bloqueos, estabilidad de conexión y rendimiento, siendo adecuada para el manejo de archivos de gran tamaño y accesos concurrentes.

Nube privada. Modelo de computación en la nube en el que la infraestructura es operada de manera exclusiva por una institución, permitiendo un control directo sobre el almacenamiento, el acceso y las políticas de seguridad de la información.

NVMe (*Non-Volatile Memory Express*). Protocolo de acceso a almacenamiento de alta velocidad que utiliza el bus PCI Express (PCIe), empleado para mejorar el rendimiento de lectura y escritura mediante caché o dispositivos de estado sólido.

OPcache. Mecanismo de optimización de PHP que almacena en memoria el *bytecode* compilado, disminuyendo los tiempos de carga y el consumo de CPU en aplicaciones web.

RAID 5. Esquema de almacenamiento redundante que distribuye datos y paridad entre varios discos, permitiendo la tolerancia a la falla de un disco sin pérdida de información y manteniendo un equilibrio entre capacidad y seguridad.

Synology NAS. Solución comercial de almacenamiento en red que combina hardware dedicado con el sistema operativo DSM, facilitando la administración de volúmenes, usuarios, respaldos y servicios de red.

VMware. Plataforma de virtualización utilizada para la creación y administración de máquinas virtuales que alojan servicios institucionales en entornos controlados.

vSwitch (conmutador virtual). Componente de red virtual que permite segmentar, aislar y gestionar el tráfico de datos entre máquinas virtuales y adaptadores físicos dentro de un entorno de virtualización.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 26 - 36

Zabbix. Plataforma de monitoreo que permite la recolección de métricas, generación de alertas y visualización del estado de servicios e infraestructura en tiempo real.

ANEXO B. ANÁLISIS COMPARATIVO DE HERRAMIENTAS COLABORATIVAS

En la Tabla 1, se presenta una comparación entre diferentes alternativas tecnológicas y la solución adoptada:

Tabla 1

Comparativa de alternativas de almacenamiento en red y servicios en la nube

Requisito	*Google Workspace*	*Nextcloud + NAS*	*Dropbox Business*	*Seafile CE + NAS*
Almacenamiento local	☒	☑	☒	☑
Control total de privacidad	☒	☑	☒	☑
Personalización	Limitada	Alta	Limitada	Media
Costos a largo plazo	Elevados	Mantenimiento	Elevados	Mantenimiento
Integración con red local	☒	☑	☒	☑
Automatización de respaldos	Limitada	☑	Limitada	☑
Requisitos de internet	Siempre	Solo sincronización externa	Siempre	Solo sincronización externa

La elección de Nextcloud con una NAS Synology se justificó por la necesidad institucional de mantener un control local de los datos, minimizar costos operativos y asegurar independencia tecnológica. A diferencia de las soluciones basadas en la nube pública como Google Workspace o Dropbox, la combinación de Nextcloud y almacenamiento local ofrece:

- Alto nivel de control y privacidad: al no depender de terceros para el almacenamiento, se favorece el cumplimiento con políticas internas y normativas institucionales.
- Alta personalización: se puede adaptar la plataforma a flujos de trabajo específicos, integrar módulos adicionales y monitorear el sistema en detalle.
- Costos predecibles: una inversión inicial única en infraestructura y software libre elimina las suscripciones recurrentes.
- Disponibilidad sin depender de Internet: permite operar y acceder a los datos incluso ante

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 27 - 36

interrupciones del servicio externo.

- Resiliencia y respaldo: mediante una arquitectura RAID y copias programadas locales y remotas.

Frente a otras soluciones autoalojadas como **Seafile CE**, Nextcloud ofrece una comunidad más activa, más aplicaciones compatibles (como Forms, Deck, Office), y una interfaz más amigable para usuarios no técnicos. Además, la integración con herramientas como Zabbix permite monitorear el sistema con métricas nativas gracias a la instalación del agente de Zabbix y el uso del protocolo SNMP, sin necesidad de configuraciones manuales adicionales.

ANEXO C. ANÁLISIS DEL SISTEMA DE ALMACENAMIENTO EN RED

La elección de la plataforma Synology NAS DS1821+ como sistema de almacenamiento centralizado tiene como elementos claves rendimiento, confiabilidad y facilidad de administración. El modelo DS1821+, con procesador AMD Ryzen V1500B (4 núcleos a 2.2 GHz) configurado con 8 discos de 14 TB, permitió configurar volúmenes con RAID 5, optimizando la capacidad útil y garantizando redundancia frente a fallos en un disco sin sacrificar desempeño. En la Tabla 2, se observan estas cualidades en comparación con otras soluciones.

Ventajas principales:

Integración nativa con servicios NFS, SMB y AFP, facilitando la interoperabilidad con servidores Linux, macOS y Windows.

Sistema operativo DSM 7.2 con interfaz gráfica avanzada y soporte para copias de seguridad automáticas.

Mantenimiento reducido mediante monitoreo de salud de discos, temperatura y ventiladores.

Escalabilidad mediante expansiones y conectividad PCIe para tarjetas de red de alta velocidad o caché NVMe (*Non-Volatile Memory Express*), utilizada como memoria de alto rendimiento para acelerar operaciones de lectura y escritura en el sistema de almacenamiento.

Tabla 2

Cuadro comparativo de plataformas NAS y de almacenamiento

Criterio	Synology DS1821+ (RAID 5)	QNAP TS-873A (RAID 5)	TrueNAS CORE (Servidor personalizado)	Servidor Linux con RAID manual (mdadm)
CPU / RAM	AMD Ryzen V1500B, 4 núcleos, 4 GB (ampliable a 32 GB)	AMD Ryzen V1500B, 8 GB (ampliable)	Depende del hardware, flexible	Depende del hardware, flexible
Sistema operativo	DSM 7.2 (propietario)	QTS 5.x (propietario)	TrueNAS CORE (FreeBSD)	Ubuntu/Debian con utilidades Linux

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 28 - 36

Criterio	Synology DS1821+ (RAID 5)	QNAP TS-873A (RAID 5)	TrueNAS CORE (Servidor personalizado)	Servidor Linux con RAID manual (mdadm)
Facilidad de administración	Muy alta (GUI intuitiva)	Alta (más opciones avanzadas)	Media (requiere conocimientos técnicos)	Baja (requiere configuración manual)
Compatibilidad NFS / SMB / AFP / iSCSI	Completa	Completa	Completa	Completa (manual)
Seguridad y control de acceso	ACL, LDAP, Active Directory, 2FA	ACL, LDAP, 2FA	ACL, certificados TLS, ZFS snapshots	Depende del administrador
Gestión de copias de seguridad	Integrada (Hyper Backup, Snapshots)	Integrada (Hybrid Backup Sync)	Snapshots automáticos ZFS	Manual o scripts personalizados
Rendimiento en RAID 5 (8x14 TB)	~700 MB/s lectura, ~650 MB/s escritura	~700 MB/s	Depende del hardware (puede superar 1 GB/s)	Variable según configuración
Escalabilidad / expansión	Hasta 18 discos (con DX517)	Hasta 18 discos (con UX-800P)	Ilimitada según hardware	Ilimitada según hardware
Soporte técnico / garantía	3 años (extensible)	3 años (extensible)	Comunidad activa, sin soporte comercial	Sin soporte oficial
Costo aproximado (sólo chasis)	25,000–30,000 MXN	27,000–32,000 MXN	Variable (~20,000 MXN)	Bajo (~15,000 MXN)
Ideal para	Instituciones académicas y de investigación	PYMEs y entornos multimedia	Centros de datos avanzados	Laboratorios o entornos de prueba

ANEXO D. DESCRIPCIÓN DE LOS PARÁMETROS OPTIMIZADOS

1. Optimización del Montaje NFS y Configuración Interna de Nextcloud

El servidor Nextcloud, con almacenamiento principal en una NAS fue montado mediante el protocolo Network File System versión 4.1, optimizado para manejo de archivos de gran tamaño y acceso concurrente. En la Tabla 3, se describe cada parámetro utilizado para su montaje.

2. Configuración del montaje NFS

Archivo /etc/fstab:

```
192.X.X.X:/volume1/CloudFile /mnt/ncdata2 nfs4 \
rw,sync,noatime,nodiratime,vers=4.1,proto=tcp,rsize=262144,wsize=262144,\
nconnect=4,hard,timeo=150,retrans=5,sec=sys,clientaddr=192.168.0.2,local_lock=none 0 0
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 29 - 36

Tabla 3

Descripción de parámetros aplicados en el punto de montaje

Parámetro	Descripción	Beneficio técnico
vers=4.1	Usa NFS versión 4.1 con conexión TCP única y mejor control de locks	Estabilidad y rendimiento
rw,sync	Escrituras síncronas garantizan integridad de datos	Fiabilidad ante cortes o fallos
noatime,nodiratime	Desactiva actualización de metadatos en accesos	Reduce operaciones I/O
rsize/wsize=	Tamaño de bloque de lectura/escritura	Buen equilibrio entre throughput y latencia
proto=tcp	Fuerza uso de TCP en lugar de UDP	Mayor fiabilidad en red gigabit
nconnect=4	Usa 4 conexiones TCP paralelas al NAS	Aumenta el ancho de banda efectivo
timeo=150,retrans=5	Espera 15s y reintenta 5 veces	Previene errores por micro cortes
local_lock=none	Evita bloqueos locales innecesarios	Mayor compatibilidad con NFSv4

La calibración anterior dio como resultado una reducción sustancial en la latencia de lectura/escritura y estabilidad completa durante subidas de archivos de más de 10 GB.

3. Separación del directorio temporal

Para reducir la latencia derivada del NFS en operaciones intermedias, se configuró un directorio temporal local en SSD:

```
sudo mkdir -p /var/www/nextcloud/data/tmp
sudo chown -R www-data:www-data /var/www/nextcloud/data/tmp
```

Y en el archivo config.php:

```
'tempdirectory' => '/var/www/nextcloud/data/tmp',
```

I. Configuración avanzada de Nextcloud

Fragmento relevante de config.php:

```
'memcache.local'    => '\OC\Memcache\APCu',
'memcache.locking' => '\OC\Memcache\APCu',
'filelocking.enabled' => true,
'tempdirectory'     => '/var/www/nextcloud/data/tmp',
'maintenance_window_start' => 1,
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 30 - 36

II. Ajustes de PHP y OPcache

```
Archivo /etc/php/8.x/apache2/php.ini:
memory_limit = 10G
upload_max_filesize = 400G
post_max_size = 20G
max_execution_time = 16500
max_input_time = 16600
extension = apcu.so
apc.enable_cli = 1

opcache.enable = 1
opcache.memory_consumption = 256
opcache.interned_strings_buffer = 24
opcache.max_accelerated_files = 10000
opcache.validate_timestamps = 1
```

III. Resultados observados

La Tabla 4 muestra los resultados de la combinación de NFSv4.1 optimizado, caché APCu, directorio /tmp local y parámetros PHP ajustados. Esto consolidó un entorno Nextcloud de alto rendimiento, estable y escalable, adecuado para la gestión de grandes volúmenes de información científica, incluyendo datos estructurados y no estructurados derivados de actividades de investigación, monitoreo y análisis académico, en una infraestructura académica local.

Tabla 4

Resultados observados, performance del servidor

Prueba	Antes de la optimización	Después	Resultado
occ files:scan (≈ 2 TB)	> 1 h	40–50 min	☑ La reducción proviene de `nconnect=4`, uso local de `/tmp` y mejora en I/O de NFS.
Subida de 10 GB	Fallaba por timeout	Exitosa	☑ Antes, la subida HTTP se interrumpía por límite de PHP o latencia; con `chunked upload`, `/tmp` local y buffers mayores, se estabiliza.
Inicio de sesión web	6–8 s	2–3 s	☑ APCu + OPcache reducen la carga inicial de PHP y lectura de metadata.
Generación de previews	Muy lenta	Instantánea	☑ La cache local y CPU dedicada permiten preprocesar sin bloqueo de NFS.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 31 - 36

ANEXO E. FIGURAS DE MONITOREO ZABBIX

Figura 1

Saltos de CPU y uso porcentual en un periodo de 4 horas con 30 usuarios activos

Figura 2

Utilización de CPU y espacio libre en almacenamiento

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 32 - 36

Figura 3

Tráfico de red en interfaces ens32 y ens192

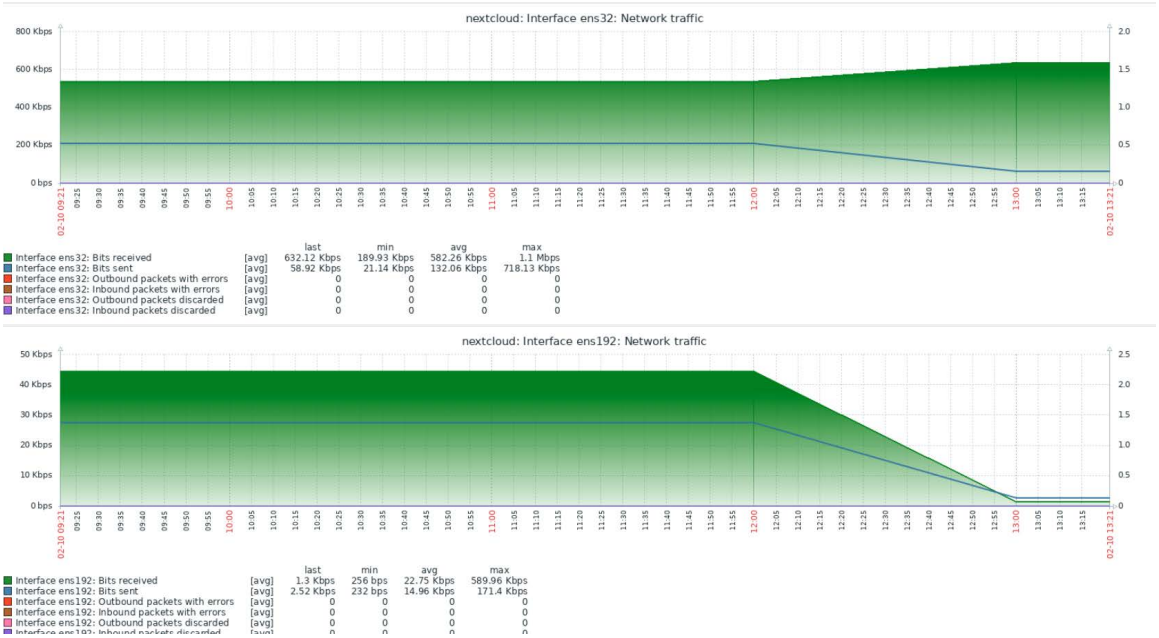
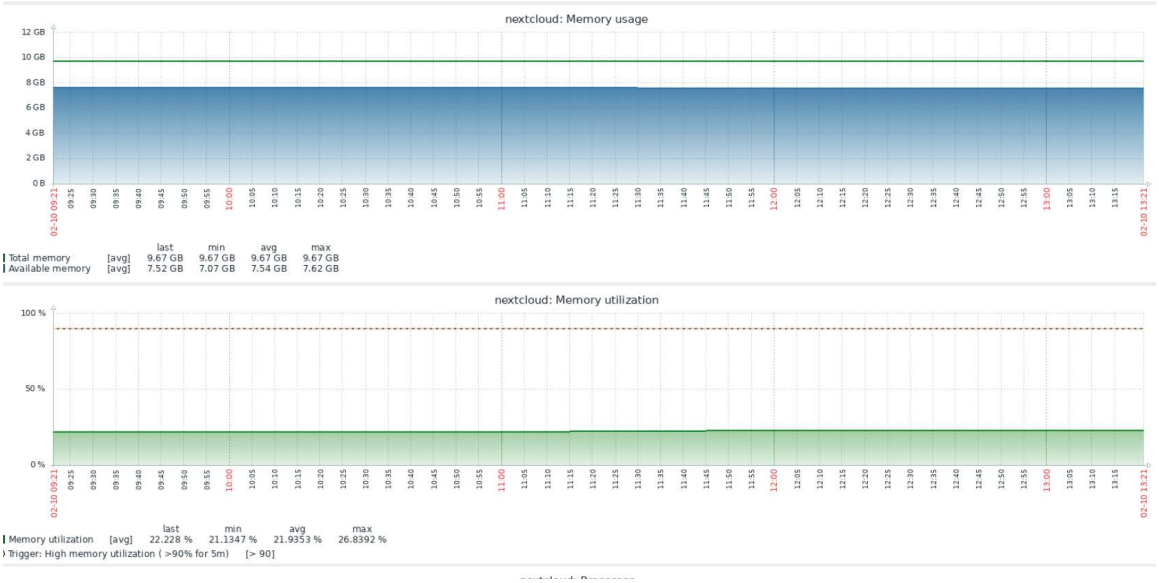


Figura 4

Uso y disponibilidad de memoria RAM



https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 33 - 36

Figura 5

Tiempos de espera y tasas de E/S en disco

ANEXO F. DESCRIPCIÓN DE LAS FORMAS DE CONEXIÓN Y ACCESO

Comparativa de métodos de montaje y transferencia hacia Nextcloud

De acuerdo a los diferentes usos de la plataforma Nextcloud (instancia institucional de nube privada con acceso restringido), se analizaron y se especificaron diferentes métodos de transferencia y acceso, cada uno con sus ventajas y limitaciones que se describen en la Tabla 5.

Tabla 5

Comparativa de métodos de montaje y transferencia de datos para Nextcloud

Método / Protocolo	Cuando utilizarlo	Ventaja	Consideración
NFS (Network File System v4.1)	Cuando el almacenamiento está en un NAS (Synology o Linux) dentro de la misma LAN. Ideal para el almacenamiento principal de datos de Nextcloud.	Alta velocidad y baja latencia; soporte total para archivos grandes; integración directa; estabilidad.	Requiere red estable; UID/GID coherentes; si se copian archivos fuera de la web, ejecutar: sudo -u www-data php /var/www/nextcloud/occ files:scan --all.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 34 - 36

Método / Protocolo	Cuando utilizarlo	Ventaja	Consideración
CIFS / SMB (Windows share)	Cuando los archivos están en un PC o NAS compartido por Windows. Ideal para copias o accesos desde Windows.	Compatible con Windows; autenticación por usuario; fácil de montar.	Rendimiento medio; puede fallar con archivos > 4 GB si el cliente usa FAT32 o SMBv1; requiere occ files:scan tras copias manuales.
WebDAV (Nextcloud nativo)	Para acceso remoto o integración sin un protocolo de acceso remoto seguro (SSH). Ideal para usuarios externos o sincronización ligera.	Cifrado HTTPS; multiplataforma; integración con clientes y apps.	Limitado según navegador y PHP; lento en archivos grandes; no apto para sincronía masiva.
rsync (por SSH)	Para respaldo o sincronización incremental de grandes volúmenes entre servidores.	Eficiente; soporta archivos grandes; compresión y reanudación.	No actualiza índice automáticamente — ejecutar occ files:scan; requiere clave SSH; no montaje interactivo.
scp / protocolo seguro de transferencia de archivos (SFTP) (manual)	Transferencias puntuales o de emergencia.	Cifrado SSH; soporta archivos grandes; simple.	Manual; no incremental; requiere occ files:scan.
Nextcloud Sync Client (oficial)	Para sincronizar carpetas de escritorio con Nextcloud (Windows/macOS/Linux).	Automático, bidireccional, con control de versiones.	Depende del ancho de banda; puede saturar CPU; apto hasta el límite PHP configurado.
Synology Nextcloud (NFS o rsync)	Cuando el almacenamiento principal o respaldo está en un NAS Synology.	Permite RAID, snapshots, estabilidad, soporte completo de archivos grandes.	UID/GID coherentes; evitar CIFS; ejecutar occ files:scan tras cambios manuales.
Cliente Web (instancia institucional de nube privada (acceso restringido)	Para subir o descargar archivos de forma interactiva, compartir enlaces y gestionar versiones desde cualquier navegador.	Indexación automática; control de versiones y etiquetas; cifrado HTTPS; apps integradas (Office, PDF, multimedia).	Usa subida fragmentada *(chunked)*; limitada por PHP/ Apache/navegador; puede fallar si conexión inestable; ideal para archivos medianos.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 35 - 36

Actualización manual del índice de archivos en Nextcloud

Cuando los archivos se copian fuera de la interfaz web o cliente, Nextcloud no los muestra automáticamente hasta que se reindexan. Los comandos principales utilizados son:

- Reindexar todo: `sudo -u www-data php /var/www/nextcloud/occ files:scan --all`
- Reindexar usuario: `sudo -u www-data php /var/www/nextcloud/occ files:scan --path="usuario/files"`
- Limpiar caché: `sudo -u www-data php /var/www/nextcloud/occ files:cleanup`

Calibración de parámetros de PHP para mejorar el performance y permitir subir archivos grandes desde la interfaz web; para archivos grandes, se debe de utilizar otro método.

```
===== Básicos de rendimiento =====
memory _ limit = 768M
max _ execution _ time = 3600
max _ input _ time      = 3600
post _ max _ size       = 8G
upload _ max _ filesize= 8G
file _ uploads        = On
max _ file _ uploads    = 200
output _ buffering    = Off
expose _ php          = Off
default _ charset     = "UTF-8"
date.timezone       = America/Cancun
upload _ tmp _ dir      = /var/www/nextcloud/data/tmp
; ===== OPcache =====
opcache.enable                  = 1
opcache.enable _ cli              = 1
opcache.memory _ consumption      = 256
opcache.interned _ strings _ buffer= 32
opcache.max _ accelerated _ files   = 50000
opcache.validate _ timestamps     = 1
opcache.revalidate _ freq         = 60
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.147 Vol. 4, Núm. 1. enero-marzo 2026, págs. 36 - 36

```
opcache.save_comments           = 1
opcache.jit                    = 0
opcache.jit_buffer_size         = 0
; ===== APCu =====
apc.enabled      = 1
apc.shm_size     = 256M
apc.ttl          = 7200
apc.enable_cli   = 1
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.145

Vol. 4, Núm. 1. enero-marzo 2026, págs. 37 - 49

Simulación de reflejos anidados en tiempo real mediante Render to Texture en el proyecto Museo de la Luz

Simulation of nested real-time reflections using Render to Texture in the project Museo de la Luz

José Larios Delgado

Dirección General de Cómputo y de Tecnologías de Información y Comunicación

Universidad Nacional Autónoma de México

jlarios@unam.mx

ORCID: 0009-0003-5663-9962

Información del reporte:

Para citar este reporte técnico:

Larios Delgado, J. (2026). Simulación de reflejos anidados en tiempo real mediante Render to Texture en el proyecto Museo de la Luz. *Cuadernos Técnicos Universitarios de la DGTIC, 4* (1), páginas (37 - 49). https://doi.org/10.22201/dgtic.30618096e.2026.4.1.145

Resumen

Se aborda la implementación en Unity de una simulación de reflejos anidados en entornos tridimensionales interactivos mediante la técnica *Render to Texture*. Con el objetivo de obtener un modelo capaz de reproducir en tiempo real el comportamiento físico de un periscopio, se priorizaron la exactitud física del fenómeno y el rendimiento del sistema. Se utilizó una metodología que consistió en generar cámaras asociadas a planos reflectantes, las cuales producen texturas de renderizado dinámicas, ajustando su posición y orientación en función del cálculo del vector reflejado, la dirección del vector de vista y la normal del plano que representa al espejo. El rendimiento del sistema se evaluó incrementando el número de reflejos y analizando métricas como el número de cuadros por segundo, el tiempo de renderizado y el uso de memoria. Se observó una reducción promedio de 40 cuadros por segundo y un aumento en el tiempo de renderizado de 1.28 milisegundos por cuadro por cada reflejo adicional. En cuanto a la exactitud física, la comparación con un periscopio real confirmó que la orientación final de los reflejos en la simulación coincide con la orientación observada físicamente, lo que permitió validar el algoritmo implementado. Se concluye que la simulación propuesta constituye una solución factible para representar reflejos anidados en tiempo real dentro de entornos interactivos para la web, ofreciendo un equilibrio adecuado entre exactitud visual y desempeño.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.145 Vol. 4, Núm. 1. enero-marzo 2026, págs. 38 - 49

Palabras clave: Aprendizaje autodirigido, periscopio virtual, reflejo en tiempo real, renderizado a textura.

Abstract

The implementation of nested reflection simulation in interactive three-dimensional environments using the Render to Texture technique within Unity was addressed. The objective was to develop a model capable of reproducing the physical behavior of a periscope in real time, prioritizing both the physical accuracy of the phenomenon and system performance. The methodology consisted of generating cameras associated with reflective planes, which produced dynamic rendering textures by adjusting their position and orientation based on the calculation of the reflected vector, the view direction vector, and the normal of the plane representing the mirror. System performance was measured by increasing the number of reflections and analyzing metrics such as frames per second, rendering time, and memory usage. An average reduction of 40 frames per second and an increase of 1.28 milliseconds per frame for each additional reflection were observed. Regarding physical accuracy, comparison with a real periscope confirmed that the final orientation of the simulated reflections matched the physically observed orientation, thereby validating the implemented algorithm. It was concluded that the simulation represents a feasible solution for rendering nested reflections in real time within web-based interactive environments, offering an appropriate balance between visual accuracy and performance.

Keywords: *Real time reflection, render to texture, self–directed learning, virtual periscope.*

1. INTRODUCCIÓN

En la actualidad, los entornos digitales interactivos son una herramienta clave para comunicar conceptos de manera accesible y atractiva en espacios educativos y museísticos (Serrano Moral, 2014). Este tipo de herramientas permiten al público experimentar y fomentar el aprendizaje a través de la exploración activa.

En este contexto, el Museo de la Luz identificó la necesidad de contar con una exposición permanente que coadyuve a la comprensión del fenómeno de la reflexión de la luz, por lo que surgió la propuesta de crear un espacio digital interactivo que permitiera experimentar este fenómeno de manera remota y continua. Para ello, durante el 2025, un equipo multidisciplinario de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación definió los objetivos y alcances para el proyecto, proponiendo un interactivo web que fomente la libre experimentación y el aprendizaje autónomo sin mediación docente, en concordancia con los enfoques contemporáneos que promueven la autorregulación y la exploración activa del conocimiento en entornos educativos digitales (Kemp, Davidson, Hurse, & Naik, 2024).

Se planteó un reto interactivo en donde el usuario se enfrenta a la tarea de construir un periscopio digital dentro de un ambiente simulado, un concierto multitudinario. Utilizando componentes como espejos y extensores, el usuario descubre los principios ópticos que le permitan visualizar el escenario.

Este desarrollo presentó dos desafíos técnicos centrales: la simulación de reflejos en tiempo real y su anidamiento, es decir, lograr que un espejo refleje la imagen de otro espejo sucesivamente. Se empleó el motor Unity, seleccionado por su capacidad de exportación a WebGL, su amplio conjunto de herramientas

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.145 Vol. 4, Núm. 1. enero-marzo 2026, págs. 39 - 49

para entornos 3D interactivos y su compatibilidad con técnicas avanzadas de renderizado[1], características ampliamente documentadas en estudios comparativos sobre motores de juego y entornos WebGL aplicados a la visualización interactiva (Zheng et al., 2023).

El objetivo del proyecto fue implementar una simulación de reflejos anidados en tiempo real, como los de un periscopio, dentro de un entorno educativo digital desarrollado en Unity para el proyecto Museo de la Luz.

2. DESARROLLO TÉCNICO

Simulación de reflejos en tiempo real

Existen múltiples técnicas para la simulación de reflejos, por ejemplo, el trazado de rayos en tiempo real, el cual presenta limitaciones importantes en contextos donde el rendimiento y la compatibilidad multiplataforma son prioritarios. En los últimos años, se han realizado numerosas mejoras en este algoritmo y se ha alcanzado un punto en el que los resultados son viables para tiempo real, gracias a técnicas avanzadas y hardware dedicado específicamente al trazado de rayos (Clemenz y Weydemann, 2021). Aunque tiene ventajas como la precisión de los reflejos que produce —incluyendo interreflexiones de manera nativa—, su dependencia de hardware especializado, como las GPU con núcleos para *Ray Tracing* (RT) (Tatzgern, et al., 2024), restringe su implementación en plataformas como WebGL .

Por esta razón, el uso de la técnica de *Render to Texture* (RTT) representó una alternativa eficiente y práctica para el proyecto, ya que permite obtener reflejos visualmente coherentes en tiempo real sin requerir hardware de aceleración de RT.

RTT permite capturar la escena desde una perspectiva específica y proyectar el resultado como una textura sobre una superficie, creando la ilusión de un reflejo coherente con el entorno y la posición del observador (Unity Technologies, 2023; Epic Games, 2024), la cual consiste en redirigir la salida del renderizado hacia una textura almacenada temporalmente en memoria. En lugar de mostrar la imagen directamente en pantalla, el motor gráfico la guarda en un *buffer*. Dicha textura puede aplicarse después a otro objeto dentro de la escena y actualizarse de forma dinámica en cada cuadro (Möller, Haines y Hoffman, 2018).

RTT en la simulación de reflejos

Para implementar reflejos en tiempo real usando RTT, se siguieron los siguientes pasos:

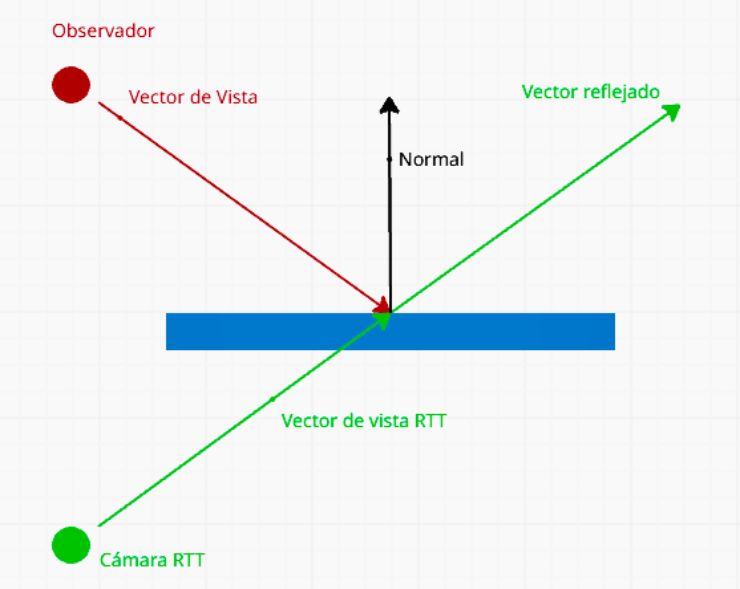
1. **Creación de una cámara para hacer el RTT**, cuya vista representa la "vista reflejada" de la escena. Esta cámara se posiciona de manera simétrica respecto al plano que simula el espejo.
2. **Dibujo de la escena de la cámara** a una textura, guardando lo que se "vería" en el reflejo. La dirección del vector de vista de la cámara debe coincidir con la dirección del vector reflejado, calculado a partir del vector de vista del observador y la normal del plano reflectante (ver Figura 1).
3. **Asignación de la textura (reflejo) como material del objeto reflectante,** textura que se le aplica al espejo.

1 Para la definición de éste y otros conceptos técnicos, véase el glosario contenido en el Anexo A.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.145 Vol. 4, Núm. 1. enero-marzo 2026, págs. 40 - 49

Figura 1

Esquema de Render to Texture



Esta técnica permite obtener un reflejo coherente, que se actualiza en tiempo real, corresponde al movimiento del usuario o de los objetos en la escena.

Ventajas: Generar reflejos dinámicos y coherentes con el entorno (Shirley et al., 2021), permitir el reflejo de objetos que se mueven, de luces y efectos visuales en tiempo real, sin requerir del uso de técnicas de *ray tracing*; se puede implementar en múltiples plataformas, incluyendo WebGL.

Limitaciones: Implica dibujar la escena por lo menos dos veces cada cuadro, lo que incrementa el costo computacional (Akenine-Möller et al., 2018), en superficies curvas o con múltiples reflejos, el manejo de las cámaras encargadas del RTT se vuelve más complejo y propenso a errores (Akenine-Möller et al., 2018).

Simulación de reflejos anidados

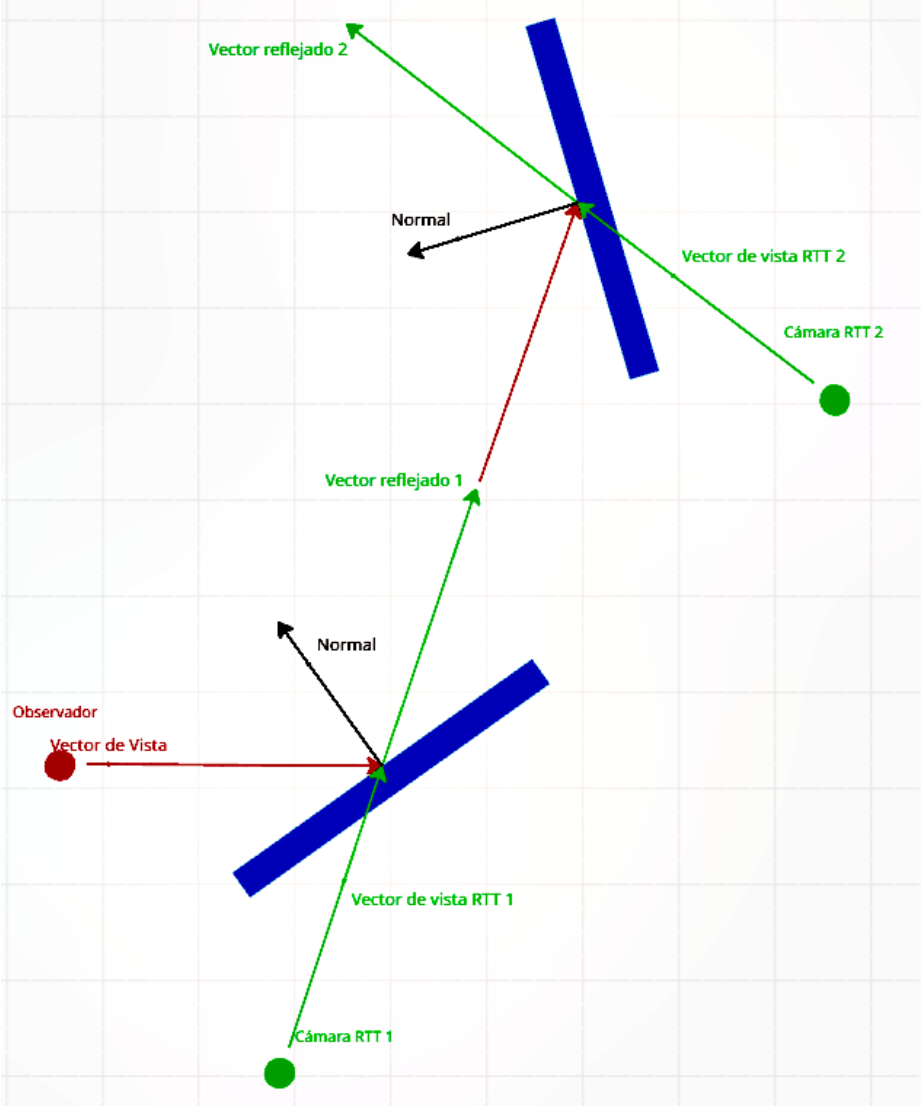
Una vez resuelto el reflejo de un espejo, se abordó el desarrollo de una solución para simular el funcionamiento del periscopio de forma digital. Esto requirió el anidamiento de los reflejos simulados de múltiples espejos.

El anidamiento de reflejos significa que, en tiempo real, el vector reflejado del primer espejo debe actualizarse en función de la dirección del vector de vista de la cámara principal (observador). El vector resultante actúa a su vez como el nuevo vector de vista para calcular el reflejo del segundo espejo, y así sucesivamente con los demás planos reflectantes (ver Figura 2).

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.145 Vol. 4, Núm. 1. enero-marzo 2026, págs. 41 - 49

Figura 2

Esquema reflejos anidados



Nota. Resalta la necesidad de controlar el orden de dibujo de las cámaras que realizan el RTT, ya que se observa que el primer reflejo simulado debe corresponder al último espejo, mientras que el último reflejo debe corresponder al primer espejo. Esto se debe a que este orden (inverso) garantiza la coherencia de la simulación y evita que ocurran errores por falta de sincronización en el dibujo de las texturas.

2.1 METODOLOGÍA

La implementación se realizó siguiendo un enfoque de prototipado evolutivo, ya que permite construir y refinar progresivamente las funcionalidades de un sistema, facilitando la validación temprana de resultados y la corrección iterativa del diseño (Pressman & Maxim, 2020).

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.145 Vol. 4, Núm. 1. enero-marzo 2026, págs. 42 - 49

A continuación, se describe el procedimiento que se siguió para desarrollar un prototipo en Unity que utiliza la técnica de *Render to Texture* para que simule el reflejo de un espejo.

Implementación de Render to Texture en Unity

Se creó un recurso de tipo RenderTexture y se le nombró *MirrorRT_A*; se configuró con los siguientes parámetros:

- **Resolución:** 1024 × 1024 (modificable para optimizar el rendimiento).
- ***Depth Buffer*:** 24 bits (información de profundidad).
- ***Color Format*:** R8G8B8A8.

En la jerarquía de la escena, se creó una nueva cámara, *RTCameraA*. La textura *MirrorRT_A* se le asignó en el campo *TargetTexture* de la cámara *RTCameraA* en el editor de Unity, con esto se logra que la salida del renderizado se guarde en MirrorRT_*A*.

Aplicación del reflejo

Se creó un nuevo material con el *shader* estándar de Unity y se le asignó *MirrorRT_A* como la textura base (*Base Map*). Este material se aplicó al plano que funciona como espejo. En tiempo de ejecución, la textura se actualiza, mostrando el punto de vista de la cámara *RTCameraA* (Möller, Haines y Hoffman, 2018).

Actualización de la posición y orientación del reflejo

La cámara que simula el reflejo (*RTCameraA*) se ubica y orienta sobre el vector reflejado de la vista del observador (*MainCamera*) respecto al plano del espejo (ver Figura 1). Su posición se calcula reflejando la posición de la cámara principal a través del plano reflejante y a la misma distancia que el observador está del espejo. Asimismo, su orientación, correspondiente al vector de vista, coincide con la dirección del vector reflejado.

El cálculo se realiza mediante el siguiente pseudocódigo:

```
posicion_camara_reflejo = Reflejo(posicion_camara_principal - posicion_espejo, normal_espejo) + posicion_espejo

orientacion_camara_reflejo = Reflejo(posicion_camara_principal - posicion_espejo, normal_espejo)
```

Recordando que la función de reflejo se define como (Akenine-Möller et al., 2018):

$$R = I - 2(I \cdot N)N$$

donde:

R-> vector reflejado

I-> vector incidente

N-> vector normal del plano reflectante (normalizado)

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.145 Vol. 4, Núm. 1. enero-marzo 2026, págs. 43 - 49

Desarrollo del anidamiento de reflejos

Para obtener el anidamiento de reflejos, se desarrolló un *script* nombrado MirrorController, el cual se asignó como un componente de la cámara encargada del reflejo (*RTCameraA*). Este *script* se encarga de actualizar la posición y orientación de la cámara en el método *Update()*, e implementa el pseudocódigo antes descrito en lenguaje C#.

El *script MirrorController* depende de parámetros configurables:

- **MainCam:** cámara desde la cual se observa el reflejo.
- **MirrorGO:** plano que representa el espejo.

De esta forma, el anidamiento se logra encadenando cámaras y espejos de la siguiente manera (ver Figura 2):

Primer espejo (RTCameraA), parámetros en *MirrorController***:**

MainCam: Main Camera (observador)

MirrorGO: MirrorA

Segundo espejo (RTCameraB), parámetros en *MirrorController***:**

MainCam: RTCameraA (primer espejo)

MirrorGO: MirrorB

Finalmente, para controlar el orden de dibujo de las cámaras, se configuró la propiedad *Priority* en el editor. Una prioridad más alta indica que la cámara se renderiza antes que las cámaras que tienen una prioridad menor. Así pues, la cámara del último reflejo se configuró con la prioridad más alta (5) y se asignaron valores decrecientes hasta llegar a la cámara del primer reflejo con la prioridad más baja (1).

3. RESULTADOS

Como resultado de la implementación anteriormente descrita, se identificaron dos aspectos fundamentales: Primero, el rendimiento de la simulación y sus límites (cuántos espejos pueden simularse y cuál es el impacto cada vez que se agrega un espejo más). Segundo, la correspondencia de la simulación respecto al fenómeno real de reflexión.

Rendimiento de la simulación

Se buscó determinar el impacto de agregar más espejos a la simulación, analizando la variación en el rendimiento (cuadros por segundo [FPS] y tiempo de renderizado por cuadro) al incrementar el número de cámaras que usan RTT en una escena al usar la misma configuración de hardware para todos los casos (ver Anexo A).

A continuación, se describe el procedimiento que se siguió: se tomó como base una escena desarrollada como prototipo para la aplicación y se hizo la implementación de cuatro espejos, se configuraron según lo descrito en el desarrollo. Posteriormente, al correr la simulación, se midió el rendimiento con un espejo

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.145 Vol. 4, Núm. 1. enero-marzo 2026, págs. 44 - 49

activo durante 30 segundos y se registraron los valores promedio de FPS, CPU y uso de memoria, que reportó el Profiler de Unity. Este proceso se repitió con dos, tres y cuatro espejos para generar la siguiente tabla.

Tabla 1

Resultados prueba de rendimiento

Nº de RTT	FPS promedio	CPU ms/frame	Uso de memoria (RTT) mb
1	246	4.05	341.5
2	195	5.12	395
3	159	6.27	446
4	126	7.9	497.5

3.1 ANÁLISIS

Se observó una disminución de aproximadamente 40 FPS por cada espejo adicional, también hubo un incremento promedio de 1.28 ms en cada cuadro (ver Tabla 1). Esto fue previsible debido a que cada RTT extra requiere un renderizado completo adicional. Con esta información, se proyectó un punto de inflexión a partir del cual la incorporación de más espejos degrada notoriamente el rendimiento y se estimó que, a partir de alrededor de seis espejos, la tasa de cuadros por segundo desciende al rango de 40 a 30 FPS.

Si se compara con el uso de técnicas como *Real-Time Rendering of Glossy Reflections*, el trazado de rayos por sí mismo tiene un costo significativo en el desempeño; se observan tiempos de renderizado por *frame* de ~5.83 ms a 1080p usando una GPU moderna como la AMD Radeon RX 7900 XTX. (Eto, Meunier, Harada, & Boissé, 2023).

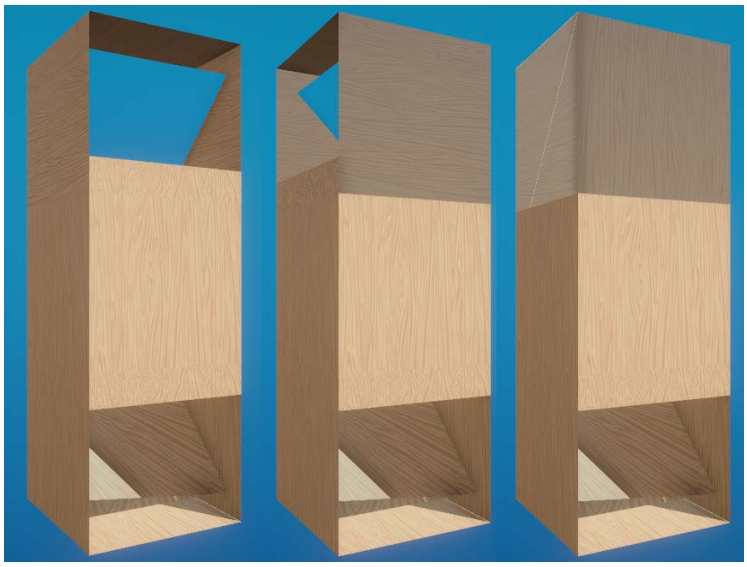
Exactitud de la simulación

Para evaluar la exactitud del modelo con el fenómeno físico, se ensamblaron tres configuraciones diferentes (ver Figura 3) usando dos espejos en un periscopio real y se observó la orientación del reflejo final (reflejo del primer espejo).

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.145 Vol. 4, Núm. 1. enero-marzo 2026, págs. 45 - 49

Figura 3

Configuraciones del periscopio virtual y real



Se observó que:

- En la primera configuración, el reflejo aparece invertido (rotado 180°).
- En la segunda, el reflejo se ve rotado 90° en sentido antihorario.
- En la tercera, el reflejo se muestra correctamente orientado.

Estas mismas configuraciones se reprodujeron en la aplicación, obteniéndose los siguientes resultados :

Figura 4

Reflejo simulado, primera configuración

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.145 Vol. 4, Núm. 1. enero-marzo 2026, págs. 46 - 49

Figura 5

Reflejo simulado, segunda configuración

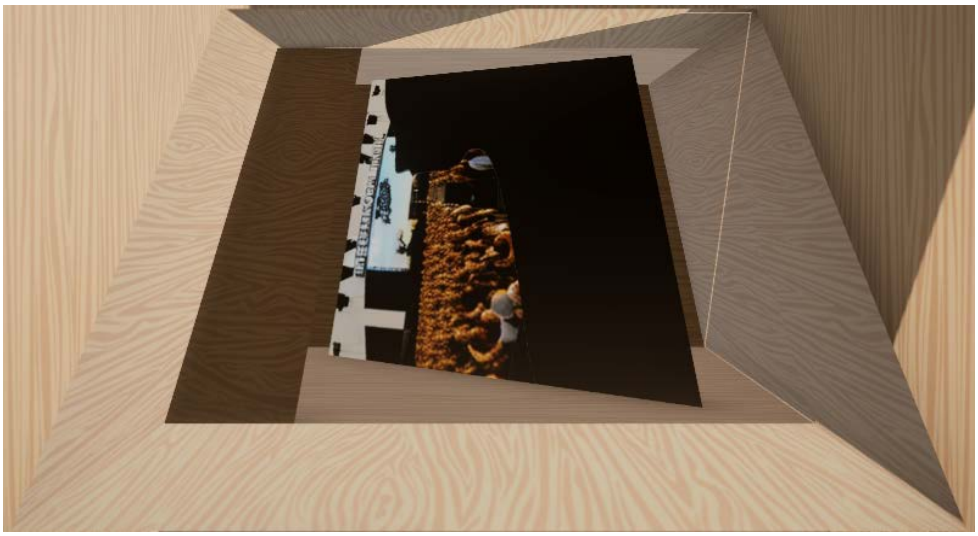
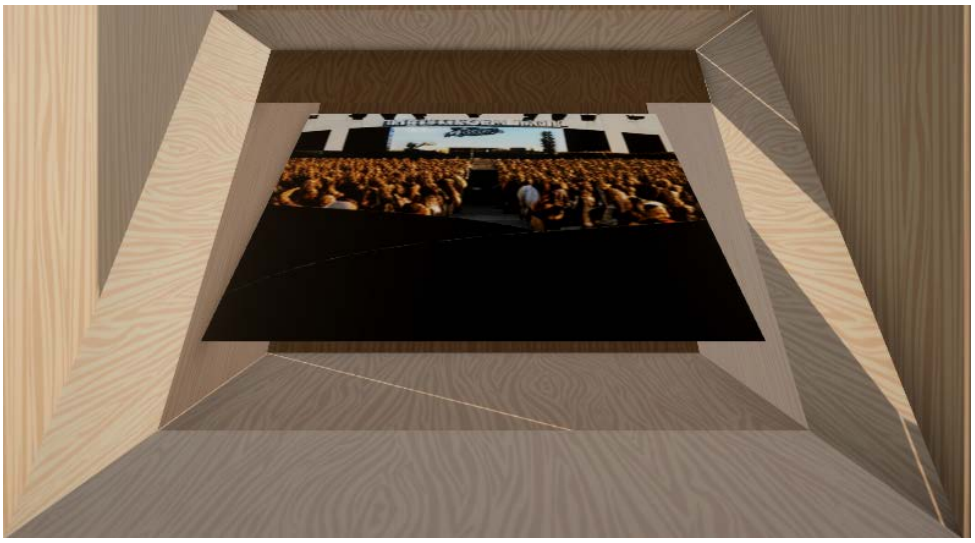


Figura 6

Reflejo simulado, tercera configuración



Observando las Figuras 4, 5 y 6, es evidente que los resultados de la simulación concuerdan fielmente con las características de los reflejos observados en el periscopio real. Esta prueba confirmó la validez del modelo de reflexión anidada implementado en el proyecto.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.145 Vol. 4, Núm. 1. enero-marzo 2026, págs. 47 - 49

4. CONCLUSIONES

Con base en los resultados, se puede afirmar que se desarrolló un sistema capaz de simular reflejos anidados en tiempo real, y se resolvieron dos principales desafíos técnicos: la representación precisa del comportamiento físico de un periscopio y la optimización del rendimiento de la simulación para su correcta ejecución en un entorno WebGL.

Las pruebas de exactitud física (ver Anexo B) mostraron que la orientación del reflejo que se observó en la simulación coincide con el reflejo en el periscopio físico, esto sirvió para validar el modelo de reflejos anidados propuesto. En cuanto al rendimiento, se pudo determinar que 4 cámaras con RTT es el máximo número factible para usar en el proyecto sin afectar la tasa de cuadros mínima deseada (30FPS). Con este número máximo de espejos, aún es posible crear escenarios interesantes para la experimentación del usuario y el rendimiento es aceptable para funcionar en aplicaciones WebGL.

Por lo anterior, se concluyó que la implementación cumplió satisfactoriamente con el objetivo planteado, al lograr una simulación que combina exactitud física, interactividad y visualización en tiempo real en entornos WebGL. Como trabajo a futuro, se prevén oportunidades de mejora en la programación de optimizaciones en la gestión de los recursos gráficos que puedan incrementar el desempeño de la implementación en las pruebas de rendimiento.

REFERENCIAS

Akenine-Möller, T., Haines, E., & Hoffman, N. (2018). *Real-Time Rendering* (4.ª ed.). CRC Press.

Clemenz, C., & Weydemann, L. (2021). Reflection techniques in real-time computer graphics. *KoG : Journal of the Croatian Computer Graphics Society, 25*(25), 87–95. https://doi.org/10.31896/k.25.10

Epic Games. (s. f.). *Textures in Unreal Engine. Epic Games Documentation.* https://dev.epicgames.com/documentation/en-us/unreal-engine/textures-in-unreal-engine

Eto, K., Meunier, S., Harada, T., & Boissé, G. (2023). Real-Time Rendering of Glossy Reflections Using Ray Tracing and Two-Level Radiance Caching. In *SIGGRAPH Asia 2023 Technical Communications*. Association for Computing Machinery (ACM). https://doi.org/10.1145/3610543.3626167

Kemp, K. L., Davidson, C. J., Hurse, D. N., & Naik, A. R. (2024). Exploring educational experiences that correlate with self-directed learning in college students seeking to pursue science, technology, engineering, math, and medical (STEMM) fields. *Frontiers in Education, 9*, Article 1393493. https://doi.org/10.3389/feduc.2024.1393493

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

Serrano Moral, C. (2014). ¿Museos del futuro? Comunicación, educación e interactividad. *International Journal of Educational Research and Innovation (IJERI), 2*, 129-140. Recuperado de https://www.upo.es/revistas/index.php/IJERI/article/view/1142

Shirley, P., Ashikhmin, M., Marschner, S., & Gleicher, M. (2021). *Fundamentals of Computer Graphics* (5th ed.). CRC Press.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.145 **Vol. 4, Núm. 1.** enero-marzo 2026, págs. 48 - 49

Tatzgern, W., Weinrauch, A., Stadlbauer, P., Mueller, J. H., Winter, M., & Steinberger, M. (2024). Radiance caching with on-surface caches for real-time global illumination. *Proceedings of the ACM on Computer Graphics and Interactive Techniques, 7* (3), Article 38. https://doi.org/10.1145/3675382

Unity Technologies. (s. f.). *Render Texture* (Unity Manual). Recuperado de https://docs.unity3d.com/Manual/class-RenderTexture.html

Zheng, Y., Merchant, A., Laninga, J., Xiang, Z. X., Alshaebi, K., Arellano, N., Romaniuk, H., Fai, S., & Sun, D. H. (2023). Comparison of characteristics of BIM visualization and interactive application based on WebGL and game engine. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLVIII-M-2-2023,* 1671–1677. https://doi.org/10.5194/isprs-archives-XLVIII-M-2-2023-1671-2023

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.145 Vol. 4, Núm. 1. enero-marzo 2026, págs. 49 - 49

ANEXO A. GLOSARIO

Buffer. Área de memoria temporal donde se almacenan datos (como imágenes, vértices o texturas) antes de ser procesados o enviados a la GPU durante el renderizado.

Cuadros por segundo (FPS). Medida que indica cuántas imágenes completas se muestran en pantalla cada segundo. A mayor FPS, mayor fluidez visual en juegos y aplicaciones gráficas.

Trazado de Rayos (*Ray Tracing*). Técnica avanzada de renderizado que simula el comportamiento real de la luz mediante el trazado de rayos, logrando reflejos, sombras e iluminación más realistas.

Renderizado. Proceso mediante el cual la computadora genera una imagen o animación a partir de modelos, texturas, luces y cámaras.

WebGL. API basada en JavaScript que permite renderizar gráficos 2D y 3D acelerados por hardware directamente en navegadores web, sin necesidad de plugins.

ANEXO B. CARACTERÍSTICAS DEL EQUIPO CON EL QUE SE HICIERON LAS PRUEBAS DE RENDIMIENTO

Procesador	Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz 3.60 GHz
RAM instalada	64.0 GB
Tarjeta gráfica	NVIDIA GeForce RTX 2080 (8 GB), Intel(R) HD Graphics 630 (128 MB)
Tipo de sistema	Sistema operativo de 64 bits, procesador basado en x64

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152

Vol. 4, Núm. 1. enero-marzo 2026, págs. 50 - 70

Optimización de las transmisiones *streaming* a partir de los dispositivos periféricos asociados

Optimization of streaming broadcasts from associated peripheral devices

Información del reporte:

Para citar este reporte técnico:

Romo Zamudio, J.F. (2026). Optimización de las transmisiones streaming a partir de los dispositivos periféricos asociados. *Cuadernos Técnicos Universitarios de la DGTIC, 4* (1), páginas (50 - 70). https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152

José Fabián Romo Zamudio
Dirección General de Cómputo y de Tecnologías de Información y Comunicación
Universidad Nacional Autónoma de México
fabian.romo@unam.mx
ORCID: 0009-0003-9269-8185

Resumen

Con el propósito de incrementar la calidad y estabilidad de las transmisiones de eventos académicos vía *streaming* producidos por la Dirección General de Cómputo y de Tecnologías de Información y Comunicación de la Universidad Nacional Autónoma de México durante 2025, se realizó un análisis de la problemática existente. Dicho análisis se integró a una propuesta inicial para la mejora del cableado, conexiones y equipos periféricos de audio y video, lo que permitió ejecutar pruebas físicas en diversos escenarios con los equipos disponibles y agregar componentes nuevos de bajo costo para reducir los potenciales puntos de falla. A raíz de la simplificación alcanzada en la ingesta y procesamiento de señales digitales, se alcanzó el objetivo de proporcionar transmisiones en línea a un mayor número de participantes, a la vez que se identificaron los componentes y recomendaciones básicas para la producción de *streaming* en espacios como aulas y auditorios.

Palabras clave: Audio digital y analógico, transmisiones por internet, video digital y analógico.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 **Vol. 4, Núm. 1.** enero-marzo 2026, págs. 51 - 70

Abstract

With the aim of improving the quality and stability of streaming transmissions for academic events produced by the General Directorate of Computing and Information and Communication Technologies of Universidad Nacional Autónoma de México *during 2025, an analysis of the underlying issues was conducted. This analysis was integrated into an initial proposal to enhance cabling, connections, and audio-video peripheral equipment. This approach enabled the execution of physical tests across multiple scenarios using available equipment, as well as the incorporation of new, low-cost components to reduce potential points of failure. As a result of the simplification achieved in the ingestion and processing of digital signals, the objective of delivering online broadcasts to a larger number of participants was met, while also identifying key components and basic recommendations for streaming production in spaces such as classrooms and auditoriums.*

Keywords: *Broadcast over the internet, digital and analog audio, digital and analog video.*

1. INTRODUCCIÓN

Las tecnologías de *streaming* transformaron el ambiente educativo universitario en los últimos cinco años, al evolucionar de ser un recurso complementario hacia una infraestructura esencial para la colaboración académica, proceso catalizado por la pandemia de COVID-19, y se consolidaron como un elemento permanente debido a su eficacia y flexibilidad. La Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO) (2020) recomendó el uso de herramientas tecnológicas para impartir clases en video y emplear las plataformas integradas de aprendizaje ante el reto de dar continuidad a la educación en todos los niveles durante la emergencia sanitaria. El requerimiento fue de tal magnitud que, así como se hicieron ajustes pedagógicos, los avances tecnológicos en todo el mundo alcanzaron niveles de desarrollo no previstos, con el fin de garantizar la disponibilidad de las herramientas. La Universidad Nacional Autónoma de México (UNAM) no es ajena a esta transformación y, en lo particular, la Dirección de Innovación en Tecnologías para la Educación (DITE) de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) ha usado el *streaming* para la difusión de sus actividades académicas.

El *streaming*, definido como "una tecnología de transmisión de datos que permite consumir contenido multimedia (audio, video, etc.) de forma continua a través de una red, sin necesidad de descargar previamente el archivo completo al dispositivo del usuario" (Rueda & Rueda, 2020, p. 45) se consolidó apoyado en tres pilares: componentes técnicos, usuarios y contenidos. En lo tecnológico, la maduración de plataformas y la expansión de la banda ancha permitieron que las transmisiones en vivo y bajo demanda fueran más estables y accesibles (Adedoyin & Soykan, 2020). Por su lado, los usuarios demandaron flexibilidad y acceso asincrónico a los contenidos, razón adicional que consolidó al *streaming* como una solución que trascendió las barreras del aula física. Finalmente, los contenidos se diversificaron: además de la grabación de clases para posterior consulta (*podcast* o "*Portable On Demand broadCAST*"), se incluyeron conferencias, presentaciones y mesas de trabajo en vivo.

En ese ámbito, el *streaming* producido por la DITE presentó algunas problemáticas que incidieron en una menor asistencia de usuarios en línea, además de limitar la participación en los segmentos interactivos. Entre las causales, estaban la baja calidad del audio, las resonancias y los ecos, interrupciones de la transmisión y poca variación de fuentes de video. En febrero de 2025, se presentó a la DITE un análisis con recomendaciones técnicas que incluyó la estimación inicial de la inversión necesaria partiendo de

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 52 - 70

un estudio de mercado. En éste, se definió como objetivo incrementar sustancialmente la participación de estudiantes, personal académico y personas afines a las actividades de la DITE emitidas vía *streaming,* mediante la mejora en la calidad y estabilidad de las transmisiones, empleando herramientas tanto físicas (*hardware*) como *lógicas* (software) con la menor erogación económica posible.

2. DESARROLLO TÉCNICO

Para cumplir el objetivo, se ejecutaron tres etapas: a) identificación de problemática y causas, b) definición de elementos para la mejora de transmisiones y c) aplicación y evaluación práctica de la solución propuesta. La Tabla 1 del Anexo A resume los resultados de esas etapas y el Anexo B describe a detalle los procedimientos en cada una. A partir del análisis inicial y la elección de los componentes más viables en el corto plazo, se diseñó una propuesta para dar solución a la problemática. A continuación, se detallan los elementos técnicos de la solución.

2.1 CABLEADO DE LAS CÁMARAS

Considerando solo las cámaras disponibles, la mayoría con puertos de tipo *HDMI*[1], se adquirieron adaptadores *HDMI – Ethernet* [2] pasivos, lo que permitió alcanzar distancias de hasta 30 metros sin degradación perceptible de la señal. Esta decisión se justifica en las limitaciones físicas de los cables *HDMI* pasivos, cuya longitud efectiva sin atenuación notable rara vez supera los 15 metros, dependiendo del estándar y la calidad del cable. De acuerdo con las guías técnicas de Anker (2024) y Cable Matters (2023), los cables *HDMI* pasivos garantizan una transmisión confiable de señales 4K únicamente hasta alrededor de 5 metros, y su desempeño comienza a degradarse progresivamente con longitudes mayores, requiriendo soluciones activas o basadas en fibra para tramos extendidos. En contraste, el cable *Ethernet* categoría 5e o 6 permite aprovechar un medio físico diseñado para enlaces de hasta 100 metros, conforme a las normas *IEEE 802.3* (2022) y *TIA-568* (2024), con control de atenuación y diafonía adecuados para la transmisión de señales digitales balanceadas. El empleo de extensores *HDMI* sobre cable *Ethernet* representó una solución técnicamente viable para instalaciones audiovisuales que requieren continuidad de señal en tramos superiores a los que soporta un cable *HDMI* convencional, manteniendo la integridad eléctrica y compatibilidad con las resoluciones de trabajo necesarias para el *streaming*. En la Tabla 2 del Anexo A se muestra la inversión realizada tanto en los adaptadores de extensión sobre Ethernet como en los respectivos cables, lo que significó un ahorro del 80% de la estimación inicial.

2.2 CAPTURA DE VIDEO

Se adquirieron adaptadores/capturadores HDMI hacia USB[3] 3.0 (tipo "*capture dongles*") como interfaz de bajo costo entre la salida HDMI de las cámaras y las computadoras para codificación. Técnicamente, estos dispositivos aprovechan el enlace SuperSpeed USB 3.0 (5 Gbit/s teóricos) para transportar tramas sin compresión o con compresión mínima a rangos de hasta 1080p (Alta definición) y 60 cuadros por segundo, con una latencia y rendimiento adecuados para la producción en vivo (Uitto & Heikkinen, 2020; Yan & Yi, 2022). La elección de USB 3.0 permite alimentar varias interfaces sin necesidad de adaptadores

1 HDMI: *High Definition Multimedia Interface*

2 *Ethernet*: Tecnología que conecta dispositivos a una red de área local (*LAN*) mediante cableado físico.

3 USB. *Universal Serial Bus*.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 53 - 70

externos, ya que el estándar básico de USB 3.x proporciona 5 V con hasta ≈900 mA (≈4.5 W). No obstante, capturadores de mayor demanda (como 4K a 60 cuadros sin compresión o con electrónica activa avanzada) requieren alimentación auxiliar con puertos con mayor capacidad (*Power delivery*) para garantizar estabilidad y evitar caídas de rendimiento (USB-IF, 2022). Debido a esto, se conectaron los adaptadores HDMI hacia un concentrador USB 3.0 con fuente de poder de 5V a 3 amperes. Con ello, se consiguió un enlace estable y sin pérdida de calidad para la ingesta hacia el software de codificación (OBS). En la Tabla 3 del Anexo A, se describen los límites de escalabilidad de fuentes de video a partir de las pruebas realizadas. Como se indica en la Tabla 2, del mismo anexo, la inversión por tres capturadoras de video con concentrador USB externo implicó erogar solo el 5% de lo calculado en el primer análisis.

2.3 MICROFONÍA

Se optó por usar micrófonos de solapa (*lavalier*) inalámbricos, dado el equilibrio entre practicidad, movilidad y bajo costo. Tienen respuesta en frecuencia plana y disponen de la adecuada relación señal/ruido para la voz hablada, además de una mínima interferencia con la movilidad del presentador, factores que los hacen idóneos para entornos de *streaming* profesional (Zheng et al., 2024). La mayoría de estos sistemas opera en banda UHF entre 470 y 698 MHz, espectro asignado a la radiodifusión digital terrestre y compartido con servicios móviles, por lo que se debe usar en el rango correcto (Taha et al., 2024). Del mismo modo, es indispensable evitar interferencias con redes Wi-Fi y LTE cercanas, particularmente para micrófonos en la banda ISM de 2.4 GHz, la cual es susceptible a la congestión en entornos académicos, empresariales y urbanos (Ofcom, 2012). Estos micrófonos fueron eficientes al conectar sus receptores en el concentrador USB, adquirido para los convertidores de video, además de económicamente viables, como se puede verificar en la Tabla 2 del Anexo A, ya que representaron sólo el 7% del costo inicial estimado. Adicionalmente, en la Tabla 3 del mismo anexo, se resume la escalabilidad derivada de las pruebas que se efectuaron.

2.4 CAPTURA Y MEZCLA DE AUDIO

Todos los dispositivos de entrada se integraron como fuentes digitales hacia la computadora de codificación para asegurar la consistencia en la sincronización y la calidad del sonido durante el *streaming*. Asimismo, la unidad receptora de los micrófonos de solapa se conectó al concentrador USB 3.0 y los micrófonos amplificados (como los presentes en auditorios, analógicos en su mayoría) se conectaron a través de una mezcladora de cuatro canales con interfaz digital USB 3.0, lo que unificó las señales en formato PCM[4]. Esto anuló las potenciales pérdidas por conversiones analógico-digitales sucesivas y redujo tanto la diafonía como el ruido inducido, típicos de conexiones analógicas de extensa longitud. En consecuencia, se aseguró una mayor estabilidad en la tasa de muestreo y profundidad de bits, factores críticos para la inteligibilidad y fidelidad de la voz (Mori et al., 2023). Diversos estudios recientes destacan que las arquitecturas basadas en interfaces USB 3.0 y audio digital sobre bus de alta velocidad proporcionan un flujo síncrono de baja latencia, adecuado para aplicaciones de transmisión en vivo y entornos de producción multimedia (Huang & Wang, 2021; West et al., 2023). La conversión inmediata a digital evitó la degradación por interferencias electromagnéticas, un fenómeno especialmente relevante en escenarios de codificación portátiles o con múltiples fuentes simultáneas (Boddeda et al., 2022). Paralelamente, seleccionar una mezcladora con menor número de canales implicó un ahorro significativo, como se verifica en la Tabla 2 del Anexo A.

4 PCM. *Pulse Code Modulation*

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 **Vol. 4, Núm. 1.** enero-marzo 2026, págs. 54 - 70

2.5 CODIFICACIÓN

La estación principal fue una laptop con sistema operativo Windows 10, procesador Intel Core i5, 16 GB de RAM y GPU Nvidia, a la que se conectó el concentrador USB 3.0 que reunía las capturadoras de vídeo, los receptores de micrófonos *lavalier* y la mezcladora de audio. Como respaldo, se configuró una laptop MacBook Air 13″ con chip Apple M1 y 8 GB de RAM. Esta arquitectura de ingesta totalmente digital facilitó que el software OBS Studio recibiera las señales en un formato uniforme y redujo al máximo posible tanto las conversiones analógico–digital adicionales como la cantidad de puntos de conexión, mitigando en el CPU la carga por correcciones de sincronía y posibles ruidos de línea (OBS Project, 2021). Para las transmisiones requeridas por la DITE en 1080p, las guías de ingeniería de transmisión recomiendan —como mínimo— un procesador del rango de un Intel Core i5, o equivalente, y 16 GB de RAM para mantener una codificación estable con tasas de bits para YouTube (de 8 a 12 Mbps para video HD a 1080p), así como también para aceptar múltiples dispositivos USB simultáneos y procesar escenas complejas en OBS, al que se configuró para delegar a la GPU Nvidia la codificación, aliviando así la carga de la *CPU* y reduciendo la latencia de codificación en vivo, acciones recomendadas en estudios y por la investigación sobre arquitecturas de *streaming* en tiempo real (Drosos et al., 2022).

3. RESULTADOS

Durante dos eventos de alto impacto (EDUCATIC 2025 y sesión final SEMINOVA 2025), se alcanzó una mayor participación, estabilidad y calidad en la transmisión, como se puede verificar en la Tabla 4 en el Anexo A. No se registraron desconexiones de los equipos de codificación, el flujo de la señal fue constante y esto permitió que los asistentes participaran de manera más dinámica e interactiva. Como subproducto, se elaboró una guía para la mejora de las transmisiones con diagramas para facilitar la implementación; éstos se incluyen en el Anexo B. También se detectaron áreas de oportunidad adicionales, como la sustitución de los conectores HDMI para prevenir fallas, la futura migración hacia adaptadores activos de HDMI a Ethernet para extender la distancia hacia las cámaras hasta 150 metros, mejorar la calidad de la imagen a resolución 4K e incorporar subtitulado por Inteligencia Artificial en tiempo real.

4. CONCLUSIONES

El proyecto resolvió deficiencias técnicas en las transmisiones de *streaming*, en específico en cuanto a la calidad del audio, la estabilidad del video y confiabilidad de la codificación. En el diagnóstico inicial y las pruebas para la construcción de una solución entre febrero y julio de 2025, se constató que los problemas provenían de pérdidas de señal por limitaciones físicas del cableado HDMI, interferencias en los sistemas de audio y cargas excesivas en la computadora de codificación. El principal aporte técnico del proyecto fue concentrar las fuentes de audio y video en una arquitectura digital, lo que redujo la latencia y los errores de sincronía. La digitalización completa de las líneas de captura y codificación elevó la calidad del *streaming*, con equipos de bajo costo a partir de una configuración simple. El proyecto alcanzó su objetivo de manera satisfactoria y estableció una base replicable para la mejora continua de las transmisiones académicas.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 55 - 70

AGRADECIMIENTOS

El autor agradece por su apoyo en el diagnóstico, la evaluación de alternativas y la puesta en operación del plan de mejora con los dispositivos periféricos seleccionados durante los eventos EDUCATIC 2025 y SEMINOVA sesión del 5 de septiembre de 2025, organizados por la DITE DGTIC UNAM, a las siguientes personas: Dra. Marina Kriscautzky, Mtro. Stephen García, Mtra. Lizbeth Heras, Mtro. Emilio Quiroz, D.G. Emmanuel Rojón, Mtro. Arturo Muñiz, Lic. César Ordoñez, Lic. Daniel González.

REFERENCIAS

Adedoyin, O. B., & Soykan, E. (2020). Covid-19 pandemic and online learning: the challenges and opportunities. *Interactive Learning Environments*, *31*(2), 863–875. https://doi.org/10.1080/10494820.2020.1813180

Anker. (2024, 31 julio). What Is the Longest HDMI Cable? Understanding the Max HDMI Cable Length. *Anker Blog*. https://www.anker.com/blogs/cables/max-hdmi-cable-length

Boddeda, R., Nandakumar, R., Gollakota, S., & Choudhury, R. R. (2022). Digital audio systems with low-latency wireless links for interactive media applications. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 18*(2), 1–21. https://doi.org/10.1145/3462635

Cable Matters. (2023, 11 septiembre). How Long Can an HDMI Cable Be? - The maximum length of an HDMI cable. *Cable Matters Blog*. https://www.cablematters.com/Blog/HDMI/how-long-can-an-hdmi-cable-be

Drosos, I., & Guo, P. J. (2022). The design space of livestreaming equipment setups: Trade-offs, challenges, and opportunities. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference (DIS '22)* (pp. 835-848). Association for Computing Machinery. https://doi.org/10.1145/3532106.3533489

Huang, J., & Wang, Y. (2021). Research and application of high-speed and adjustable synchronous data transfer system based on USB 3.0 peripheral controller. *Journal of Circuits, Systems and Computers*. https://doi.org/10.1142/S0218126621501188

IEEE Standards Association. (2022). *IEEE Standard for Ethernet: IEEE Std 802.3-2022*. https://standards.ieee.org/ieee/802.3/10422/

Mori, F., Santoni, A., Fausti, P., Pompoli, F., Bonfiglio, P., & Nataletti, P. (2023). The effectiveness of least mean squared-based adaptive algorithms for active noise control system in a small confined space. *Applied Sciences*, *13*(20), 11173. https://doi.org/10.3390/app132011173

The OBS Project. (2021, 25 agosto). System requirements [Documentación]. https://obsproject.com/kb/system-requirements

Ofcom. (2012). *Potential for LTE interference to wireless audio.* https://www.ofcom.org.uk/siteassets/resources/documents/consultations/uncategorised/8180-technical-licence-conditions/associated-documents/secondary-documents/wireless_audio_testing.pdf?v=320008

Rueda, L. A. C., & Rueda, O. L. C. (2020). *Tecnologías de la información y la comunicación: Apropiación y uso en la educación*. Editorial Universidad del Cauca.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 **Vol. 4, Núm. 1.** enero-marzo 2026, págs. 56 - 70

Taha, H., Vári, P., & Nagy, S. (2024). Adjacent-channel compatibility analysis of international mobile telecommunications downlink and digital terrestrial television broadcasting reception in the 470–694 MHz frequency band using Monte Carlo simulation. *Electronics*, *13*(3), 575. https://doi.org/10.3390/electronics13030575

Telecommunications Industry Association. (2024). *ANSI/TIA-568 Commercial building telecommunications cabling standards*. https://store.accuristech.com/standards/tia-ansi-tia-568-set?product_id=2594410

Uitto, M., & Heikkinen, A. (2020). Exploiting and evaluating live 360° low latency video streaming using CMAF. In *2020 European Conference on Networks and Communications* (*EuCNC 2020*) (pp. 276-280). IEEE. https://doi.org/10.1109/EuCNC48522.2020.9200954

UNESCO. (2020). *COVID-19: 10 recommendations to plan distance learning solutions*. Recuperado de https://www.unesco.org/en/articles/covid-19-10-recommendations-plan-distance-learning-solutions

USB Implementers Forum. (2022). *USB 3.2 Revision 1.1 – June 2022*. USB-IF. https://www.usb.org/document-library/usb-32-revision-11-june-2022

Yan, Z., & Yi, J. (2022). Dissecting latency in 360° video camera sensing systems. *Sensors*, *22*(16), 6001. https://doi.org/10.3390/s22166001

West, R., Golchin, A., & Njavro, A. (2023). Real-time USB networking and device I/O. *ACM Transactions on Embedded Computing Systems*, *22*(4), 1-28. https://doi.org/10.1145/3604429

Zheng, Z., Wang, C., Wang, L., Ji, Z., Song, X., Mak, P.-I., Liu, H., & Wang, Y. (2024). Micro-electro-mechanical systems microphones: a brief review emphasizing recent advances in audible spectrum applications. *Micromachines*, *15*(3), 352. https://doi.org/10.3390/mi15030352

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 57 - 70

ANEXO A. RESULTADOS DE ESAS ETAPAS

Tabla 1

Ámbitos de análisis, áreas de oportunidad y soluciones propuestas para la mejora del streaming en la DITE

Ámbito	Área de oportunidad	Solución propuesta
Captura de video	Usar más de una cámara y varias escenas con la menor cantidad de operadores.	Agregar cámaras del tipo PTZ (pan-tilt-zoom) reutilizadas o de bajo costo.
	Agregar cámaras de escena fija.	Incorporar cámaras estáticas preexistentes.
	Estabilizar el video de origen.	Utilizar extensiones de cableado digital desde cada fuente de video.
	Variar los orígenes de video durante la transmisión.	Agregar mezcladora de video con corte directo y desvanecimiento y homologar las entradas a la computadora de *streaming*.
Captura de audio	Emplear micrófonos de solapa	Adquirir micrófonos *lavalier* recargables y de bajo costo.
	Incorporar micrófonos ambientales.	Sumar micrófonos de mesa o poste activos.
	Eliminar resonancias, ecos e interferencias en el audio.	Usar mezcladora analógica con salida digital hacia la computadora de *streaming* y puerto de monitoreo.
Iluminación	Mejorar la calidad de imagen con iluminación directa e indirecta.	Reubicar lámparas frías en función del espacio donde se genere la señal. Homologar la temperatura de color.
Conectividad	Estabilizar las transmisiones hacia Internet.	Usar conexiones alámbricas, reducir al mínimo las de radio frecuencia.
Computadora de *streaming*	Reforzar la capacidad de procesamiento y codificación.	Emplear dos estaciones de codificación con prestaciones de cómputo superiores.
Software de transmisión	Asegurar que el software para codificación y transmisión opere de forma estable.	Utilizar la última versión de una plataforma estándar y abierta para la transmisión hacia YouTube.
Subtitulaje	Proporcionar mayor accesibilidad al contenido con subtitulaje preciso.	Combinar herramientas automatizadas, de inteligencia artificial y revisor humano para el subtitulaje de las grabaciones en podcast.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 58 - 70

Tabla 2

Comparativo de costos de los componentes físicos requeridos para el streaming mejorado

Estimación inicial de costos (Febrero de 2025)			Inversión en la solución implementada (Julio – agosto de 2025)		
Cantidad	**Descripción**	**Costo total (*)**	**Cantidad**	**Descripción**	**Costo total (*)**
Transmisión de video					
6	Cable HDMI 4K de 15 metros alta velocidad	$8400.00	3	Cable Ethernet CAT 6 de 30 metros	$1104.00
3	Repetidor – extensor activo HDMI 4K	$1188.00	3	Extensor HDMI sobre Ethernet / pasivo	$600.00
Captura de video					
1	Mixer – switcher grabador de video ATEM Mini Pro	$8000.00	3	Convertidor HDMI a USB 3.0	$600.00
1	Tarjeta de video Nvidia GeForce RTX 4070	$22000.00	1	Concentrador USB 3.0 de 8 puertos con fuente de poder	$500.00
Microfonía					
2	Kit de dos micrófonos de solapa RODE Wireless Go II	$10000.00	2	Kit de dos micrófonos de solapa en banda 2.4GHz con receptor USB	$700.00
Mezcla de audio					
1	Mezcladora de audio BEHRINGER Xenyx X1622. 16 canales con interfaz USB	$5400.00	1	Mezcladora de audio de 4 canales con interfaz USB	$600.00
Gran total estimado			**Gran total invertido**		
$54,988.00			$4,104.00		
Ahorro general			97%		

(*) Los costos se expresan en pesos mexicanos e incluyen IVA

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 59 - 70

Tabla 3

Límites de escalabilidad resultado de pruebas de fuentes de audio y video digital

Prueba	Equipo de cómputo	Puertos USB 3.x	Puertos USB 2.x	Máx. de fuentes de audio	Máx. de fuentes de video
1	Escritorio, Intel i5, 4 GB RAM.	2	3	2	1
2	Portátil, Intel i5, 4 GB RAM	2	2	2	1
3	Escritorio, Intel i5, 8 GB RAM	3	3	2	2
4	Portátil, Intel i5, 8 GB RAM	2	2	2	2
5	Portátil, Intel i5, 8 GB RAM	8 Con concentrador externo USB 3.0 y fuente de poder	2	3	3
6	Escritorio, Intel i7, 8 GB RAM	3	3	3	3
7	Portátil, Intel i5, 16 GB RAM	8 Con concentrador externo USB 3.0 y fuente de poder	2	4	4
8	Escritorio, Intel i7, 32 GB RAM.	6	3	4	5

Tabla 4

Relación de sesiones donde se usaron los componentes de hardware y software y las configuraciones mejoradas para streaming

Evento	Fecha, sesión y URL	Duración	Participantes / vistas totales
EDUCATIC 2025	30 JUL 2025. Sesión 1. https://www.youtube.com/live/q-pifuKbIgs?si=11hQwYrz5Qr3AnH1	2:28:15	250/ 1,546

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 60 - 70

Evento	Fecha, sesión y URL	Duración	Participantes / vistas totales
EDUCATIC 2025	31 JUL 2025. Sesión 2. https://www.youtube.com/live/V4YKof8Io_c?si=heHKJjQos_ztYliP	1:42:11	210/ 987
	1 AGO 2025. Sesión 3. https://www.youtube.com/live/b5csG2nAFl8?si=oJjd5H8hglqyNx_c	2:22:18	225/ 993
SEMINOVA sesión final 2025	5 SEP 2025. Sesión "¿Cómo se decide el futuro de la IA generativa? Política, sociedad y academia en debate" https://www.youtube.com/live/teSmiEriTKc?si=f4xDd32hyLUcgvyR	1:52:25	70 / 468

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 61 - 70

ANEXO B. PROCEDIMIENTOS PARA LA INTEGRACIÓN DE SOLUCIÓN A LA MEJORA DEL STREAMING

Procedimiento de la etapa 1. Identificación de la problemática y sus causas

Objetivo: Verificar el correcto funcionamiento e integración de los componentes de captura, procesamiento y envío de una transmisión en vivo, asegurando estabilidad, calidad audiovisual y continuidad del servicio, antes y durante la emisión.

Componentes:

- Video: Dos cámaras de video con salida HDMI y tarjeta de digitalización externa
- Audio: Dos micrófonos inalámbricos con salida analógica (miniplug)
- Codificación: Computadora Intel i5 con 4 GB RAM y OBS Studio
- Conectividad: Red inalámbrica compartida 2.4 GHz (Wi-Fi)
- Plataforma para transmisión de streaming: YouTube

Sección I. Fases y tareas generales

En esta sección del procedimiento, se describen las cuatro fases generales aplicables a cualquier espacio, con ajustes según el aforo:

Fase 1. Inspección física y de conexiones

Tareas:

1.1 Verificar la integridad de los cables HDMI y de los conectores, asegurando la holgura que evite tensiones en el cable, así como evitando falsos contactos.

1.2 Confirmar que tengan la suficiente carga los dispositivos que emplean baterías, y preferentemente emplear eliminadores – cargadores todo el tiempo.

1.3 Revisar las conexiones de los dispositivos analógicos de audio, asegurando que los conectores plug o mini estén firmemente fijados a la computadora de codificación.

1.4 Identificar posibles fuentes de interferencia eléctrica o inalámbrica, tales como motores, ventiladores, aires acondicionados, microondas, teléfonos inalámbricos.

Fase 2. Verificación de las señales de video

Tareas:

2.1 Confirmar que se detectan correctamente las señales de las cámaras en OBS por medio de la tarjeta de digitalización externa que está conectada por USB a la computadora de codificación.

2.2 Verificar que la resolución, la frecuencia de cuadros y la sincronía de imagen sea uniforme en OBS.

2.3 Evaluar que la imagen sea estable al menos durante 10 minutos.

2.4 Detectar cortes, congelamientos o pérdida de señal HDMI.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 62 - 70

Fase 3. Verificación de señal de audio

Tareas:

3.1 Confirmar que se recibe señal de ambos micrófonos en OBS.

3.2 Verificar los niveles de audio con la ayuda de audífonos conectados a la computadora de codificación, con el fin de evitar saturación y ruido de fondo.

3.3 Detectar eco, realimentación o interferencias de radio frecuencia.

3.4 Validar que exista sincronización entre el audio y el video.

Fase 4. Verificación de red y transmisión

Tareas:

4.1 Medir la estabilidad de la red Wi-Fi, identificando que sea estable la conectividad hacia el servicio de publicación de streaming.

4.2 Realizar la transmisión privada o de prueba en YouTube desde OBS.

4.3 Evaluar la tasa de bits sostenida que indique tanto YouTube como OBS en su monitor de transmisión, verificando que no existan cuadros perdidos.

4.4 Realizar la simulación durante al menos 20 minutos sostenidos.

Sección II. Tareas específicas y resultados por tipo de espacio

En esta sección del procedimiento, se describen las tareas específicas aplicadas a cada espacio que se verificó para la integración de la solución, así como las fallas detectadas:

Espacio 1. Aula en DGTIC para 20 personas

Condiciones generales: Espacio reducido con acústica controlable y una mínima interferencia inalámbrica.

Tareas específicas:

1.1 Ubicar las cámaras a corta distancia y evitar el uso de zoom digital.

1.2 Colocar los micrófonos al menos a 1 metro de distancia de las bocinas para impedir la generación de eco.

1.3 Configurar los micrófonos con niveles bajos a medios de ganancia.

1.4 Usar la configuración a la Wi-Fi institucional con señal fuerte (superior al 70 %).

1.5 Realizar la prueba de transmisión de 15 minutos.

Fallas detectadas:

- Eco por la cercanía entre micrófonos y altavoces.
- Saturación de audio por la ganancia excesiva.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 63 - 70

- Desconexiones por el peso de los conectores HDMI hacia el dispositivo capturador.
- Cortes de transmisión por la saturación de red inalámbrica
- Cortes en la codificación por la cantidad de memoria RAM en la computadora con OBS.

Espacio 2. Auditorio de DGTIC para 60 personas

Condiciones generales: Mayor distancia entre periféricos, con acústica variable y más dispositivos conectados a la red.

Tareas específicas:

2.1 Verificar la longitud y la calidad de los cables HDMI.

2.2 Ajustar los encuadres amplios y revisar la iluminación del auditorio.

2.3 Probar los micrófonos mientras se camina por el estrado para detectar zonas muertas.

2.4 Revisar las posibles interferencias de otros micrófonos inalámbricos en operación.

2.5 Ejecutar una prueba de transmisión de al menos 20 minutos.

Fallas detectadas:

- Contaminación de audio por reverberación.
- Desconexiones momentáneas hacia la red Wi-Fi.
- Desfases (errores de sincronía) de audio–video.
- Atenuaciones de señal de video por la longitud de los cables HDMI.
- Desconexiones por el peso de los conectores HDMI hacia el dispositivo capturador.
- Cortes en la codificación por la cantidad de memoria RAM en la computadora con OBS.

Espacio 3. Auditorio de la Escuela Nacional de Trabajo Social para 200 personas

Condiciones generales: Espacio más amplio con una acústica compleja y una elevada variabilidad en señal inalámbrica.

Tareas específicas

3.1 Verificar la estabilidad térmica de la computadora codificadora.

3.2 Ajustar la compresión de audio en OBS para reducir el ruido ambiental.

3.3 Fijar una tasa de bits de codificación reducida para evitar caída de transmisión.

3.4 Deshabilitar la Wi-Fi del público cercano al punto de transmisión o emplear una red Wi-Fi exclusiva.

3.5 Ejecutar prueba de transmisión por al menos 30 minutos.

Fallas detectadas:

- Pérdida intermitente de la señal de red inalámbrica.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 64 - 70

- Interferencias por radio frecuencia en los micrófonos.
- Atenuaciones de la señal de video por la longitud de los cables HDMI.
- Desconexiones por el peso de los conectores HDMI hacia el dispositivo capturador.
- Cortes de video por sobrecarga del sistema.

Conclusiones generales de la etapa 1. Identificación de la problemática y sus causas

A partir de las fallas detectadas, se considera que la transmisión de streaming será aceptable cuando:

- No se presenten cortes de audio o video durante la prueba.
- El audio sea inteligible, sin eco ni distorsión.
- El video sea continuo y sincronizado.
- La transmisión en YouTube se mantenga estable sin alertas de tasa de transferencia de bits.

Procedimiento de la etapa 2. Definición de los elementos para la mejora de las transmisiones.

Objetivo: Definir, de manera sistemática y justificable, los elementos técnicos, operativos y de configuración que permitan mejorar el desempeño de las transmisiones de streaming, mitigando fallas recurrentes como caídas de señal, cortes de video, eco y contaminación de audio, así como también inestabilidad en la transmisión hacia la plataforma de publicación (YouTube).

Fases y tareas generales.

Este procedimiento consta de cuatro fases, orientadas a la toma de decisiones técnicas con base en la evidencia obtenida durante pruebas reales de transmisión.

Fase 1. Análisis de resultados de la verificación

Objetivo específico: Identificar las fallas recurrentes y condiciones críticas observadas durante la fase de pruebas.

Tareas:

1.1 Clasificar las incidencias detectadas por tipo, agrupando los elementos de video, audio, red y codificación.

1.2 Determinar la frecuencia de las incidencias según el tipo de espacio analizado originalmente: aula (20 personas), auditorio mediano (60 personas), auditorio grande (200 personas)

1.3 Evaluar el impacto en la experiencia del usuario final.

Entregable: Lista por prioridades de los problemas técnicos reales, no hipotéticos.

Fase 2. Identificación de los puntos de origen de las fallas

Objetivo específico: Determinar en qué parte de la cadena de transmisión del streaming se genera la degradación del servicio.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 65 - 70

Tareas

2.1 Analizar los componentes en secuencia: Captura (cámaras, micrófonos), interconexión (HDMI, audio analógico), procesamiento (computadora codificadora, OBS Studio), envío (red inalámbrica), plataforma de destino (YouTube).

2.2 Relacionar cada falla con su punto de origen.

2.3 Establecer distinciones entre las fallas partiendo de tres criterios: Técnicas (de hardware o software), operativas (por configuración o uso incorrecto), contextuales (como acústica o saturación de red).

Entregable: Mapa conceptual con relaciones de causas raíz clasificadas por tipo de falla.

Fase 3. Definición de los criterios para la mejora del streaming

Objetivo específico: Establecer los parámetros para decidir cuáles elementos deben mejorarse o agregarse a la solución.

Tareas

3.1 Evaluar el impacto en la calidad de transmisión.

3.2 Considerar la frecuencia de las fallas.

3.3 Definir la facilidad de implementación.

3.4 Identificar el costo asociado.

3.5 Describir la potencial escalabilidad en distintos espacios.

Entregable: Esquema de decisión técnica para justificar las mejoras e inversiones.

Fase 4. Identificación de elementos de mejora

Objetivo específico: A partir de los resultados de las fases previas, definir los elementos concretos que permitirán mejorar las transmisiones.

Tareas:

4.1 Descripción de componentes para la mejora en video.

4.2 Relación de elementos para la mejora en audio.

4.3 Acciones y componentes para la mejora en conexión de red.

4.4 Elementos para la mejora en la capacidad de procesamiento y codificación.

4.5 Consideraciones generales de eficiencia operativa.

Entregable: Diagramas (véase Figuras 1, 2 y 3 al final del presente Anexo), análisis y listados de componentes de la solución.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 66 - 70

Procedimiento de la etapa 3. Aplicación y evaluación práctica de la solución propuesta

Objetivo: Aplicar de forma controlada la solución técnica de mejora y evaluar su efectividad mediante pruebas prácticas, comparables y repetibles, en transmisiones realizadas con OBS Studio hacia YouTube, en distintos tipos de espacios.

Fases y tareas generales

El procedimiento se divide en cinco fases operativas, cada una con tareas, responsables y productos verificables.

Fase 1. Preparación e implementación de la solución

Objetivo específico. Instalar y configurar los elementos de mejora definidos, asegurando compatibilidad y funcionamiento inicial.

Tareas:

1.1 Integrar los nuevos componentes definidos en la etapa 2 (capturadoras HDMI, interfaces de audio digital, extensores de cableado de HDMI por Ethernet, mezcladora digital de audio).

1.2 Aplicar configuraciones estandarizadas en OBS en los ámbitos de resolución, cuadros por segundo, tasa de bits y perfil de codificación.

1.3 Configurar perfiles diferenciados según el tipo de espacio, por medio de la verificación de todas las fuentes.

1.4 Documentar todos los perfiles base de las configuraciones.

Entregable: Sistema de transmisión actualizado y funcional, listo para pruebas.

Fase 2. Ejecución de las pruebas piloto controladas

Objetivo específico: Probar la solución en condiciones reales, pero sin exposición pública.

Tareas:

2.1 Realizar transmisiones privadas o no listadas en YouTube.

2.2 Mantener la transmisión activa durante un periodo mínimo de 15 minutos para el aula, 20 minutos para el auditorio de DGTIC y 30 minutos para el auditorio de la Escuela Nacional de Trabajo Social.

2.3 Durante las transmisiones, simular condiciones reales de uso de micrófonos, movimientos en el escenario y cambios de cámara.

2.4 Monitorear y documentar los indicadores en tiempo real de cuadros perdidos, uso del procesador central en la computadora de codificación y estado de la red inalámbrica o alámbrica.

Entregable: Evidencia documentada del comportamiento de la solución de streaming mejorada.

Fase 3. Evaluación técnica del desempeño de la solución

Objetivo específico: Medir objetivamente si la solución implementada mejora la transmisión del streaming.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 67 - 70

Tareas:

3.1 Registrar los siguientes datos con relación al video: continuidad de señal sin cortes, estabilidad de los cuadros por segundo, ausencia de congelamientos o pixelación excesiva.

3.2 Documentar los siguientes valores relativos al audio: nivel adecuado y constante, ausencia de eco y realimentación, reducción de ruido ambiente.

3.3 Registro de los siguientes valores de red: tasa de bits sostenida sin caídas críticas. latencia estable y nulas desconexiones.

3.4 Verificar los siguientes indicadores de procesamiento y codificación: uso de CPU dentro de rangos aceptables, ausencia de sobrecalentamiento de la computadora de codificación, estabilidad del software de codificación.

Entregable: Tabla comparativa de configuraciones previas y estado mejorado.

Fase 4. Ajustes correctivos y optimización

Objetivo específico: Refinar la solución con base en los resultados obtenidos y de ser necesario.

Tareas:

4.1 Ajustar la tasa de bits y resolución según estabilidad real de la transmisión.

4.2 Optimizar los filtros y ajustes de audio en la mezcladora.

4.3 Simplificar las escenas y las fuentes en OBS si es necesario.

4.4 Repetir las pruebas cortas tras cada ajuste.

4.5 Actualizar documentación de la configuración final.

Entregable: Configuración optimizada y estable para operación continua.

Fase 5. Validación operativa final

Objetivo específico: Confirmar que la solución puede utilizarse de forma regular.

Tareas:

5.1 Ejecutar una transmisión real con audiencia.

5.2 Monitorear durante toda la sesión.

5.3 Registrar incidencias (si las hubiera).

5.4 Recabar retroalimentación básica de usuarios.

Entregable: reporte de la operación en transmisión con audiencia.

Conclusión. Criterios de aceptación

La solución se considera validada cuando:

A) No se presentan interrupciones críticas.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 68 - 70

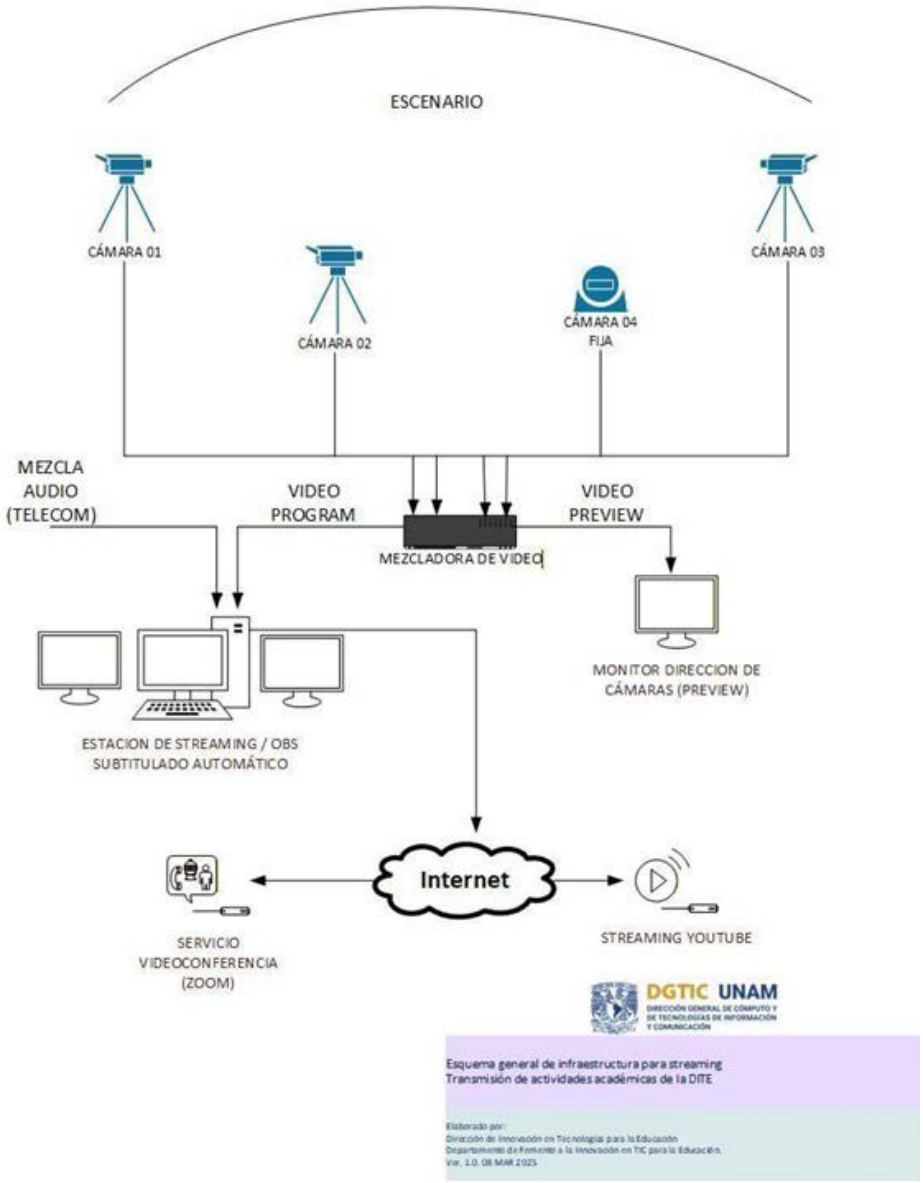
B) El audio es claro y estable.

C) El video mantiene continuidad y sincronía.

D) La transmisión se sostiene sin alertas en YouTube.

Figura 1

Diagrama tipo de las conexiones de periféricos para transmisión de streaming mejorada. Caso Aula DGTIC

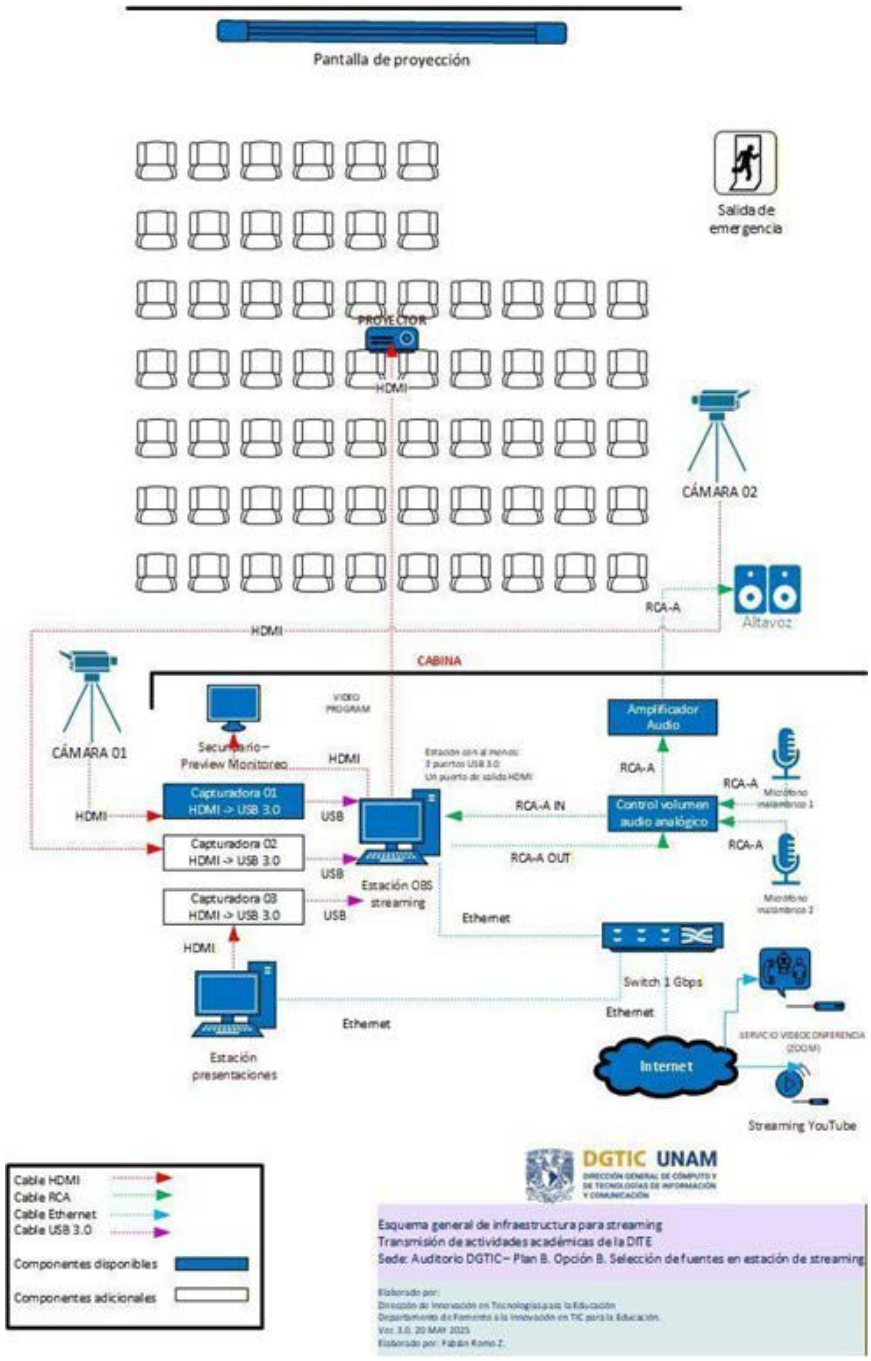


Nota. El software empleado para realizar el diagrama fue Microsoft Visio 2016.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 69 - 70

Figura 2

Diagrama tipo de las conexiones de periféricos para transmisión de streaming mejorada. Caso Auditorio de la DGTIC

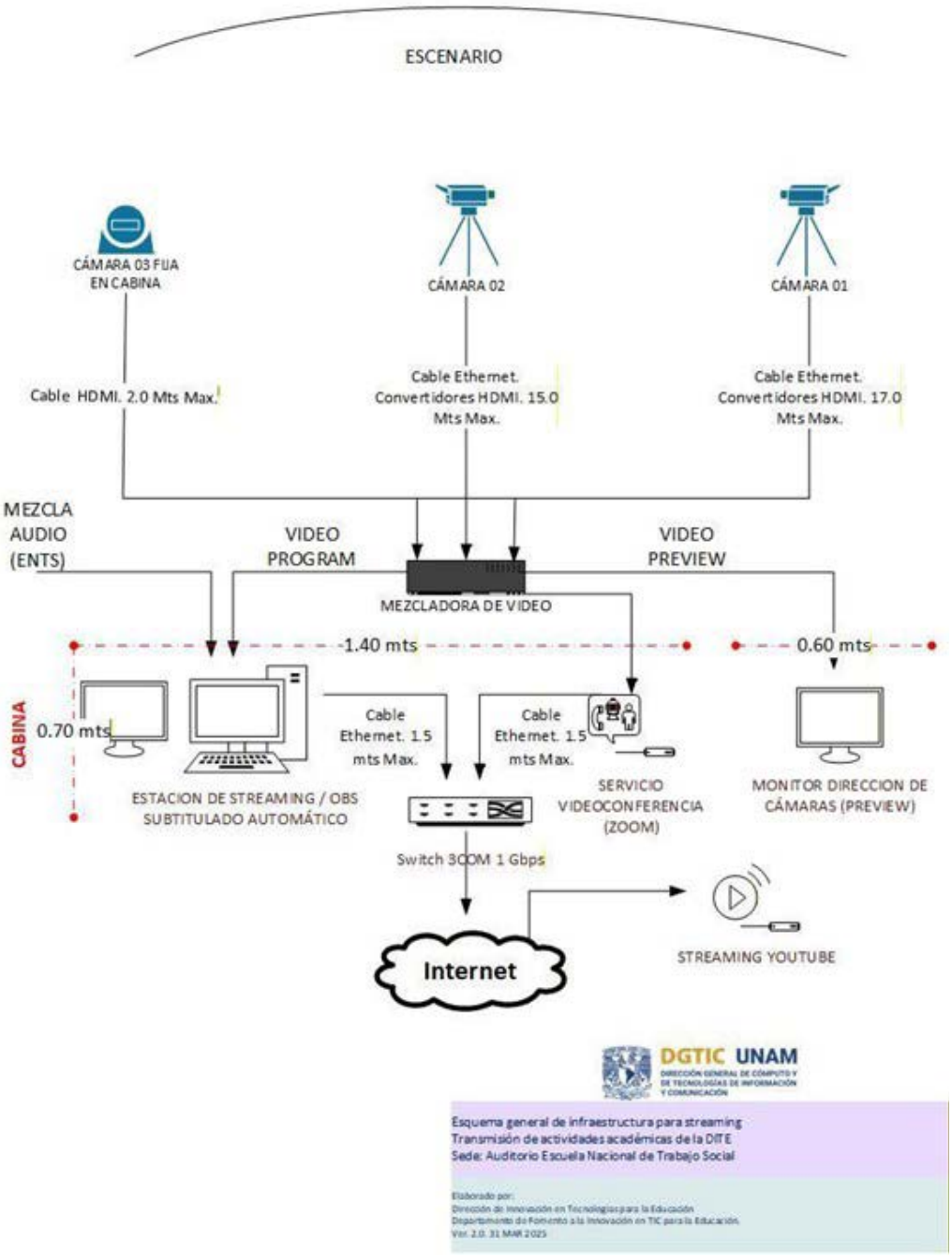


Nota. El software empleado para realizar el diagrama fue Microsoft Visio 2016.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.152 Vol. 4, Núm. 1. enero-marzo 2026, págs. 70 - 70

Figura 3

Diagrama tipo de las conexiones de periféricos para transmisión de streaming mejorada. Caso Auditorio de la Escuela Nacional de Trabajo Social



Nota. El software empleado para realizar el diagrama fue Microsoft Visio 2016.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.153

Vol. 4, Núm. 1. enero-marzo 2026, págs. 71 - 80

Automatización de auditorías de seguridad en sistemas operativos mediante OpenSCAP

Automation of security audits for operating systems using OpenSCAP

Información del reporte:

Para citar este reporte técnico:

Hernández Fuentes, F.A. (2026). Automatización de auditorías de seguridad en sistemas operativos mediante OpenSCAP. *Cuadernos Técnicos Universitarios de la DGTIC, 4* (1), páginas (71 - 80). https://doi.org/10.22201/dgtic.30618096e.2026.4.1.153

Félix Alejandro Hernández Fuentes

Dirección General de Cómputo y de Tecnologías de Información y Comunicación

Universidad Nacional Autónoma de México

felixhdz@ciencias.unam.mx

ORCID: 0000-0002-4791-485X

Resumen

La auditoría de configuraciones de seguridad en sistemas operativos, a cargo de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación, representaba un reto debido al tiempo considerable requerido por las revisiones efectuadas de forma manual y a la necesidad de garantizar altos niveles de cumplimiento con prácticas y recomendaciones de seguridad reconocidas internacionalmente. En este contexto, se evaluó el uso de la herramienta OpenSCAP para automatizar tanto la revisión como la corrección de configuraciones de seguridad. La evaluación se llevó a cabo en un entorno de pruebas con sistemas operativos de uso común en la Universidad. La herramienta permitió realizar auditorías automatizadas conforme a buenas prácticas, aplicar correcciones de forma autónoma y medir objetivamente el nivel de cumplimiento antes y después de las acciones correctivas. Los resultados mostraron que el uso de OpenSCAP redujo significativamente el tiempo de ejecución de las auditorías. Además, las correcciones automáticas aplicadas mejoraron notablemente el cumplimiento de las recomendaciones de seguridad, alcanzando niveles superiores al 90%. No obstante, se identificaron configuraciones que requirieron intervenciones manuales, sobre todo aquellas relacionadas con la gestión de contraseñas de arranque, la administración de particiones y el control de acceso a servicios críticos. Estos resultados mostraron que la automatización permitió disminuir la carga operativa

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.153 Vol. 4, Núm. 1. enero-marzo 2026, págs. 72 - 80

y hacer más uniformes los procesos de verificación, sin sustituir completamente la necesidad de supervisión técnica en casos específicos. Se concluyó que integrar mecanismos de automatización en los procedimientos institucionales contribuye a un modelo más eficiente de gestión de seguridad de la información en la Universidad.

Palabras clave: Automatización de auditorías de seguridad, CIS *benchmarks*, configuración segura de sistemas operativos, OpenSCAP, seguridad informática.

Abstract

The security configuration audit of operating systems, managed by the Dirección General de Cómputo y de Tecnologías de Información y Comunicación, posed a challenge due to the considerable time required for manual reviews and the need to ensure high levels of compliance with internationally recognized security practices and recommendations. In this context, the use of the OpenSCAP tool was evaluated to automate both the review and correction of security configurations. The evaluation was conducted in a test environment with commonly used operating systems at the University. The tool enabled automated audits based on best practices, autonomously applied corrections, and objectively measured compliance levels before and after corrective actions. The results showed that the use of OpenSCAP significantly reduced the execution time of audits. Additionally, the automated corrections applied improved compliance with security recommendations, reaching levels above 90%. However, certain configurations were identified that required manual intervention, particularly those related to boot password management, partition administration, and access control to critical services. These results demonstrated that automation helped reduce operational workload and made verification processes more uniform, without completely replacing the need for human supervision in specific cases. It was concluded that integrating automation mechanisms into institutional procedures contribute to a more efficient model for information security management at the University.

Keywords: *CIS benchmarks, information security, OpenSCAP, secure operating systems configuration, security audit automation.*

1. INTRODUCCIÓN

Hoy en día, la seguridad de la información es señalada como un aspecto fundamental para proteger las infraestructuras tecnológicas, tanto en el sector público como en el privado. Con el aumento de la complejidad de los entornos digitales y la frecuencia de los incidentes de seguridad, resulta indispensable asegurar que los sistemas operativos que gestionan servicios críticos estén correctamente configurados.

En este contexto, la auditoría de configuraciones de seguridad se entiende como el proceso sistemático orientado a revisar el cumplimiento de las configuraciones de un sistema operativo con respecto a un conjunto de reglas establecidas en buenas prácticas de seguridad. Dentro de este proceso, en opinión del autor, la verificación corresponde a la comprobación técnica del estado real de cada configuración frente a una regla específica, mientras que la evaluación se refiere al análisis del grado de cumplimiento global del sistema a partir de los resultados obtenidos.

Diversos estudios han destacado la importancia de utilizar herramientas que automaticen las auditorías de configuraciones y permitan verificar su cumplimiento con buenas prácticas reconocidas en seguridad.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.153 Vol. 4, Núm. 1. enero-marzo 2026, págs. 73 - 80

Webb, Henderson y Webb (2019) mostraron que el uso de software de código abierto, como OpenSCAP, facilita la automatización del monitoreo y la verificación de cumplimiento en entornos de prueba automatizados, reduciendo tanto costos como carga operativa. Liu y Hu (2023) destacaron la efectividad de aplicar el *Security Content Automation Protocol* (SCAP) [definido por el National Institute of Standards and Technology (NIST, 2025) como un estándar para automatizar la evaluación de configuraciones de seguridad] en equipos informáticos locales, evidenciando mejoras en la consistencia y confiabilidad de los procesos de auditoría.

Por su parte, Bandara *et al.* (2024) resaltaron cómo el uso de OpenSCAP y otras herramientas de automatización puede simplificar las auditorías de seguridad, permitiendo detectar y corregir vulnerabilidades de forma más rápida y mantener el cumplimiento con estándares vigentes. Asimismo, Urtamo y Costin (2023) demostraron la viabilidad del uso de herramientas basadas en SCAP, como OpenSCAP, para automatizar la verificación de políticas de seguridad y el escaneo de vulnerabilidades en sistemas operativos compatibles con este estándar. Finalmente, Bakhtiyarov y Mammadov (2024) analizaron la aplicabilidad de OpenSCAP en entornos empresariales, destacando su capacidad para generar reportes reproducibles y consistentes con estándares de seguridad reconocidos.

La Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) llevó a cabo auditorías de configuraciones de sistemas operativos siguiendo las recomendaciones del *Center for Internet Security* (CIS), una organización reconocida internacionalmente por sus guías de buenas prácticas de seguridad, conocidas como *CIS Benchmarks* (CIS, 2025). Estas auditorías forman parte de los servicios institucionales de seguridad informática que ofrece la DGTIC, orientados a fortalecer la protección de los sistemas operativos y aplicaciones utilizados en la Universidad, en concordancia con la *Política General de Seguridad de la Información en la DGTIC* (DGTIC, 2023; DGTIC, 2024).

No obstante, este proceso, realizado de manera manual, implicaba un esfuerzo considerable por parte del personal técnico y presentaba limitaciones, principalmente asociadas al tiempo requerido para completar cada revisión. Ante este escenario, surgió el propósito de automatizar las auditorías de configuraciones con el fin de reducir el trabajo manual, facilitar la reproducción de los resultados y mantener el cumplimiento con las recomendaciones de seguridad reconocidas.

En este contexto, el objetivo de este proyecto fue evaluar la viabilidad de utilizar OpenSCAP como herramienta de apoyo para la DGTIC en la automatización de auditorías de configuraciones de seguridad, con el propósito de hacer más eficiente y uniforme este proceso.

2. DESARROLLO TÉCNICO

La auditoría de configuraciones de seguridad en sistemas operativos es un proceso complejo y susceptible a errores humanos cuando se realiza de manera manual. En múltiples organizaciones, se siguen lineamientos como los *CIS Benchmarks*, cuya implementación permite establecer un nivel mínimo de seguridad aceptable en los sistemas evaluados.

Sin embargo, cuando el número de sistemas auditados es elevado, la revisión manual no resulta eficiente. Este es el caso de la Universidad, donde las tareas de auditoría requieren un tiempo considerable por parte del personal técnico y pueden presentar variaciones en los resultados, lo que dificulta la estandarización de los procesos de verificación.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.153 Vol. 4, Núm. 1. enero-marzo 2026, págs. 74 - 80

Existen diversas soluciones para la automatización de auditorías de configuraciones de seguridad. Algunas herramientas propietarias están disponibles, pero su uso suele verse limitado por el costo de licencias, la falta de flexibilidad en la personalización de políticas y la dependencia de un proveedor específico. En contraste, tal como lo mencionan Webb *et al.* (2019), las herramientas de código abierto han mostrado mayor flexibilidad y adaptabilidad a distintos contextos organizacionales. Esta idea se aplicó en el contexto institucional a través de la adopción de una solución de código abierto capaz de ejecutar auditorías automatizadas y alineadas con buenas prácticas de seguridad.

Entre las alternativas disponibles dentro del software libre, destaca OpenSCAP (OpenSCAP, 2025), un proyecto desarrollado por la comunidad, respaldado por Red Hat y alineado con el estándar SCAP. Tanto Bandara *et al.* (2024) como Bakhtiyarov y Mammadov (2024) coincidieron en que OpenSCAP puede aplicarse con buenos resultados en entornos empresariales, facilitando auditorías automatizadas, generando reportes consistentes y contribuyendo al cumplimiento de estándares reconocidos de seguridad.

A diferencia de otras herramientas, OpenSCAP permite integrar políticas de seguridad preexistentes, como los *CIS Benchmarks*, y adaptarlas a las necesidades específicas de cada organización. Lo anterior lo convierte en una opción adecuada para el contexto de la Universidad, donde la diversidad tecnológica y el volumen de sistemas requieren soluciones escalables y sostenibles.

El presente trabajo se desarrolló a partir de un proceso institucional de auditoría manual de configuraciones de sistemas operativos, llevado a cabo por la DGTIC conforme a los *CIS Benchmarks*. Con base en este contexto, se diseñó una metodología orientada a evaluar, de manera objetiva, el impacto de la automatización de auditorías y la aplicación de remediaciones automáticas mediante OpenSCAP. Dicha metodología se fundamentó en la ejecución comparativa de auditorías iniciales y posteriores a las acciones correctivas, cuyos resultados se presentan en la Sección 3.

2.1 METODOLOGÍA

El trabajo se estructuró siguiendo un enfoque sistemático dividido en cuatro etapas: diseño, configuración, instalación y pruebas. Este enfoque permitió operacionalizar el objetivo planteado, asegurando un proceso automatizado, reproducible y técnicamente consistente para la auditoría de configuraciones de sistemas operativos mediante OpenSCAP.

En la etapa de diseño, se seleccionaron tres sistemas operativos de uso común en la Universidad: Ubuntu 24.04, Debian 12 y Fedora 42. Esta selección se basó en la disponibilidad de perfiles de seguridad alineados con los *CIS Benchmarks* y en su compatibilidad con OpenSCAP. Asimismo, la planificación se apoyó en estudios recientes sobre automatización de auditorías y buenas prácticas en la verificación de configuraciones de seguridad.

Los perfiles utilizados correspondieron al nivel 1 (*Level* 1) de los *CIS Benchmarks*, ya que éste está orientado a proporcionar configuraciones de seguridad recomendadas que pueden aplicarse sin afectar de manera significativa la funcionalidad del sistema. En el contexto institucional de la Universidad, este nivel ofrece un equilibrio adecuado entre fortalecimiento de la seguridad y estabilidad operativa.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.153 Vol. 4, Núm. 1. enero-marzo 2026, págs. 75 - 80

Durante la configuración, se prepararon máquinas virtuales con la instalación por defecto de cada sistema operativo, sin aplicar ajustes de seguridad previos. Todas las pruebas se realizaron en un entorno virtualizado mediante VMware Workstation 17 Pro, utilizando máquinas virtuales independientes para cada plataforma y conectividad de red estándar en modo NAT, sin exposición a servicios externos, con el fin de evitar interferencias durante la evaluación.

En todos los casos, se asignó la misma configuración base de recursos de hardware: un procesador con dos núcleos, 4GB de memoria RAM y 40GB de almacenamiento, con el fin de mantener un entorno homogéneo de evaluación y asegurar que las variaciones observadas en los resultados se debieran exclusivamente a la aplicación de las auditorías y remediaciones, y no a diferencias en la infraestructura subyacente.

Las características técnicas específicas del entorno experimental, incluyendo las versiones exactas de OpenSCAP y los perfiles CIS utilizados en cada sistema operativo, se resumen en la Tabla 1, con el objetivo de asegurar la reproducibilidad del procedimiento.

Tabla 1

Características técnicas del entorno experimental

Sistema operativo	Versión OpenSCAP	Perfil CIS utilizado
Ubuntu 24.04 (64-bit)	1.3.9	CIS Ubuntu Linux 24.04 LTS Benchmark for Level 1 – Workstation
Debian 12 (64-bit)	1.3.7	CIS Debian Benchmark for Level 1 – Workstation
Fedora 42 (64-bit)	1.4.3	DRAFT – CIS Fedora Benchmark for Level 1 – Workstation

Nota. Todas las máquinas virtuales fueron desplegadas mediante virtualización de tipo hosted sobre VMware Workstation 17 Pro (versión 17.6.2 build-24409262), con configuraciones de hardware equivalentes.

En la etapa de instalación, se desplegó OpenSCAP en las máquinas virtuales configuradas. Se instalaron los paquetes necesarios, así como las dependencias y herramientas auxiliares requeridas para llevar a cabo las auditorías. Tanto la configuración como la instalación se documentaron, asegurando que el proceso pudiera reproducirse y verificarse en el futuro.

En la fase de pruebas, se realizaron dos auditorías por cada sistema operativo. La primera estableció la línea base de cumplimiento y la segunda se llevó a cabo después de aplicar las correcciones automáticas. Ambas auditorías se ejecutaron en condiciones controladas para asegurar que los resultados obtenidos fueran comparables.

OpenSCAP generó reportes que incluyeron:

- el porcentaje de cumplimiento global del sistema,
- el número de reglas cumplidas y no cumplidas,
- el detalle individual de cada regla evaluada,
- recomendaciones de corrección,

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.153 Vol. 4, Núm. 1. enero-marzo 2026, págs. 76 - 80

- la clasificación de severidad de las reglas (*high, medium, low*), y
- las referencias asociadas a cada control.

Cabe señalar que el porcentaje de cumplimiento reportado corresponde a una puntuación ponderada, calculada conforme al modelo de evaluación utilizado por OpenSCAP, el cual considera la severidad de las reglas evaluadas y no únicamente su número total.

Las auditorías de seguridad se realizaron utilizando perfiles de evaluación incluidos en los *CIS Benchmarks*, los cuales definen un conjunto de reglas orientadas a verificar configuraciones específicas del sistema operativo, tales como políticas de contraseñas, permisos de archivos, parámetros de red, servicios activos y controles de privilegios. De este modo, cada regla fue evaluada automáticamente por OpenSCAP, determinando su cumplimiento o incumplimiento con base en el estado real del sistema al momento de la auditoría.

Las remediaciones automáticas se aplicaron únicamente en aquellos casos en los que el perfil de auditoría incluía acciones de corrección explícitamente definidas. Estas acciones consistieron principalmente en la modificación de parámetros del sistema, ajustes en archivos de configuración y la habilitación o deshabilitación de servicios. No todas las reglas contaban con mecanismos de corrección automática, por lo que algunas configuraciones requirieron análisis o intervención manual posterior, con el fin de evitar afectaciones operativas.

Es importante señalar que la aplicación de remediaciones automáticas puede implicar riesgos operativos, como la modificación inadvertida de configuraciones críticas o la posible afectación de servicios. Por esta razón, las pruebas realizadas en este trabajo se llevaron a cabo exclusivamente en entornos controlados, utilizando máquinas virtuales y sin impacto sobre sistemas productivos. Asimismo, se consideró que la adopción de este tipo de mecanismos en entornos reales debe acompañarse de buenas prácticas, tales como la validación previa en ambientes de prueba, la realización de respaldos, la aplicación gradual de cambios y la supervisión técnica de las remediaciones implementadas.

El método utilizado garantizó transparencia y consistencia, tomando como referencia prácticas descritas en la literatura académica y en estándares internacionales, como los definidos por el NIST para SCAP (NIST, 2025). Gracias a esta base, fue posible obtener resultados medibles y confiables, alineados con el objetivo de este trabajo.

3. RESULTADOS

Las auditorías realizadas permitieron evaluar el impacto de OpenSCAP en los sistemas operativos seleccionados. El análisis se centró en tres plataformas de uso común en la Universidad: Ubuntu 24.04, Debian 12 y Fedora 42. El proceso incluyó una auditoría inicial para establecer la línea base, la aplicación de remediaciones automatizadas y una auditoría final para medir el cumplimiento alcanzado. Los reportes completos para los sistemas auditados (Ubuntu 24.04, Debian 12 y Fedora 42, línea base y post-remediación), generados por OpenSCAP y que detallan cada auditoría y sus resultados, se encuentran disponibles como material complementario en línea[1]. Cada archivo incluye el porcentaje de cumplimiento,

1 En la carpeta ubicada en https://drive.google.com/drive/folders/1LhdVMakP4duhh7Ro2Snh-5-69aLKvtlw que el autor se compromete a mantener disponible para consulta de los lectores de este reporte técnico.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.153 Vol. 4, Núm. 1. enero-marzo 2026, págs. 77 - 80

reglas cumplidas y no cumplidas, detalles de cada regla, recomendaciones de corrección, clasificación de severidad y referencias asociadas. Este material permite consultar los resultados directamente y profundizar en los detalles técnicos de las auditorías, asegurando transparencia y seguimiento del proceso.

La auditoría inicial estableció la línea base de cumplimiento para los sistemas operativos evaluados. Los resultados mostraron niveles parciales de conformidad, que se resumen en la Tabla 2.

Tabla 2

Resultados de la auditoría inicial de cumplimiento de configuraciones de seguridad

Sistema operativo	Reglas cumplidas	Reglas no cumplidas	Cumplimiento (%)
Ubuntu 24.04	242	113	65.76
Debian 12	235	112	68.56
Fedora 42	144	113	71.54

Nota. Los resultados corresponden a la auditoría inicial efectuada mediante OpenSCAP, con base en los *CIS Benchmarks*.

Los reportes generados por OpenSCAP mostraron que los tres sistemas evaluados presentaban configuraciones no conformes en áreas críticas y operativas. Entre los principales hallazgos, se identificaron los siguientes:

- la gestión de contraseñas,
- los parámetros de red,
- el control de privilegios mediante *sudo*,
- los permisos de tareas programadas, y
- la configuración de herramientas de integridad como AIDE.

Todas estas no conformidades fueron identificadas de acuerdo con las reglas definidas en los *CIS Benchmarks*, lo que permitió precisar los incumplimientos más relevantes en los sistemas auditados.

Después de aplicar las remediaciones automáticas con OpenSCAP, se realizó otro ciclo de auditoría para medir el impacto de las correcciones. Los resultados se muestran en la Tabla 3.

Tabla 3

Resultados de la auditoría posterior a las remediaciones automáticas

Sistema operativo	Reglas cumplidas	Reglas no cumplidas	Cumplimiento (%)
Ubuntu 24.04	347	6	92.39
Debian 12	334	16	93.04
Fedora 42	245	12	93.93

Nota. Los resultados corresponden a la segunda auditoría tras remediaciones automáticas con OpenSCAP, basada

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.153 Vol. 4, Núm. 1. enero-marzo 2026, págs. 78 - 80

en los CIS Benchmarks.

Los resultados representan mejoras de 26, 24 y 22 puntos porcentuales, respectivamente. Esto demuestra la efectividad de OpenSCAP para corregir configuraciones inseguras y uniformizar el proceso de auditoría en distintas plataformas.

La ejecución de las revisiones se volvió más rápida y eficiente gracias al uso de OpenSCAP. Además, permitió aplicar remediaciones automáticas y mejorar el cumplimiento de los sistemas auditados. Esto redujo la carga operativa sobre el personal técnico y garantizó resultados más consistentes. No obstante, algunos casos aún requirieron atención manual, particularmente en configuraciones relacionadas con la administración de particiones, seguridad física y control de accesos. Lo anterior confirma que la supervisión técnica sigue siendo indispensable.

En términos operativos, la auditoría manual de configuraciones requería, en promedio, alrededor de 13 días de trabajo, con la participación de dos integrantes del equipo técnico. En contraste, la auditoría automatizada con OpenSCAP permitió evaluar el sistema, aplicar las remediaciones disponibles y generar los reportes correspondientes en menos de una hora, requiriendo la intervención de una sola persona. Esta diferencia evidencia una reducción sustancial en el tiempo de ejecución y en el esfuerzo humano necesario para llevar a cabo el proceso de auditoría.

Los resultados obtenidos, con niveles de cumplimiento superiores al 90%, demostraron la eficacia de la herramienta para mejorar la conformidad de las configuraciones y fortalecer los procesos de auditoría institucionales. Asimismo, el procedimiento permitió generar reportes estructurados y verificables, incrementando la capacidad de la DGTIC para identificar y corregir desviaciones en configuraciones críticas. En conjunto, estos hallazgos confirman que la automatización mediante OpenSCAP puede aplicarse de manera efectiva en entornos con diversidad tecnológica, ofreciendo una base sólida para su adopción a mayor escala dentro de la Universidad.

4. CONCLUSIONES

En este reporte técnico, se expuso la necesidad de automatizar las auditorías de configuraciones de sistemas operativos basadas en las recomendaciones de los *CIS Benchmarks* en la Universidad. El propósito fue reducir la carga operativa, obtener resultados más uniformes y reproducibles, así como elevar el nivel de cumplimiento de las configuraciones evaluadas. El proceso previo requería de mucho tiempo y esfuerzo por parte del personal técnico, que debía revisar y documentar los resultados de cada regla de seguridad de manera manual.

Los resultados demostraron que la implementación de OpenSCAP redujo de manera significativa el tiempo necesario para completar las auditorías. Además, permitió alcanzar niveles de cumplimiento superiores al 90% en los sistemas evaluados. La herramienta ejecutó auditorías alineadas con buenas prácticas de seguridad reconocidas internacionalmente, aplicó remediaciones automáticas y generó reportes consistentes y reproducibles. No obstante, se identificaron configuraciones que requirieron intervención manual, particularmente aquellas relacionadas con la gestión de contraseñas, la administración de particiones y el control de accesos a servicios críticos. Esto confirma la necesidad de mantener supervisión técnica en casos específicos.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.153 **Vol. 4, Núm. 1.** enero-marzo 2026, págs. 79 - 80

Se recomienda seguir incorporando la automatización en los procesos institucionales de auditoría y ampliar la aplicación de los perfiles *CIS Benchmark* a más sistemas operativos. Además, sería importante que todos los servidores sean auditados antes de su puesta en marcha, complementando este proceso con auditorías periódicas durante su operación. Estas medidas ayudarían a fortalecer los procesos de auditoría, haciéndolos más eficientes, replicables y sostenibles. Así, se contribuiría a mantener y mejorar la seguridad de la información en la Universidad.

AGRADECIMIENTOS

Quiero expresar mi sincero agradecimiento a la Coordinación de Seguridad de la Información de la DGTIC por su generosa acogida y por brindarme la oportunidad de realizar este trabajo. También agradezco profundamente a los revisores anónimos por sus valiosos comentarios y sugerencias, los cuales han sido fundamentales para mejorar de manera sustancial este reporte técnico.

REFERENCIAS

Bakhtiyarov, B., & Mammadov, V. (2024). OpenSCAP technology application to enterprise. *PAHTEI-Proceedings of Azerbaijan High Technical Educational Institutions, 42*(05), 221-228. https://doi.org/10.36962/pahtei42072024-24

Bandara, E., Shetty, S., Rahman, A., Mukkamala, R., Foytik, P., & Liang, X. (2024). RMF-GPT — OpenAI GPT-3.5 LLM, Blockchain, NFT, Model Cards and OpenSCAP Enabled Intelligent RMF Automation System. *2024 International Conference on Computing, Networking and Communications (ICNC)*, 653-658. https://doi.org/10.1109/ICNC59896.2024.10555963

Center for Internet Security. (2025). *CIS Benchmarks: Security configuration guides*. https://www.cisecurity.org/cis-benchmarks/

Dirección General de Cómputo y de Tecnologías de Información y Comunicación. (2023). *Política General de Seguridad de la Información en la DGTIC*. Universidad Nacional Autónoma de México. https://www.tic.unam.mx/politica-general-seguridad-informacion/

Dirección General de Cómputo y de Tecnologías de Información y Comunicación. (2024). *Seguridad informática para sistemas operativos y aplicaciones*. Universidad Nacional Autónoma de México. https://sistemas.tic.unam.mx/index.php/seguridad-informatica-sistemas-operativos-aplicaciones/

Liu, Y., & Hu, B. (2023). Security baseline verification technology for domestic computer terminal based on SCAP. *2023 IEEE 5th International Conference on Power, Intelligent Computing and Systems (ICPICS)*, 171-174. https://doi.org/10.1109/ICPICS58376.2023.10235334

National Institute of Standards and Technology. (30 de septiembre de 2025). *Security Content Automation Protocol (SCAP)*. https://csrc.nist.gov/projects/security-content-automation-protocol

OpenSCAP. (2025). *OpenSCAP project*. https://www.open-scap.org/

Urtamo, I., & Costin, A. (2023). On tools for practical and effective security policy management and vulnerability scanning. En B. Shishkov (Ed.), *Business Modeling and Software Design. BMSD 2023*.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.153 **Vol. 4, Núm. 1.** enero-marzo 2026, págs. 80 - 80

Lecture Notes in Business Information Processing (Vol. 483, pp. 375-382). Springer. https://doi.org/10.1007/978-3-031-36757-1_28

Webb, J. A., Henderson, M. W., & Webb, M. L. (2019). An open source approach to automating surveillance and compliance of automatic test systems. *2019 IEEE AUTOTESTCON*, 1-8. https://doi.org/10.1109/AUTOTESTCON43700.2019.8961077

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154

Vol. 4, Núm. 1. enero-marzo 2026, págs. 81 - 107

Desarrollo de la interfaz gráfica para la plantilla de actividades del Taller Integral de Arquitectura

Design and Development of a Graphical User Interface for the Activity Template Used in the Taller Integral de Arquitectura (Comprehensive Architecture Workshop)

Información del reporte:

Para citar este reporte técnico:

Hernández Serra, I. y Zúñiga González, M. (2026). Desarrollo de la interfaz gráfica para la plantilla de actividades del Taller Integral de Arquitectura. *Cuadernos Técnicos Universitarios de la DGTIC, 4* (1), páginas (81 - 107). https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154

Itzel Hernández Serra
Dirección General de Cómputo y de Tecnologías de Información y Comunicación
Universidad Nacional Autónoma de México
iheserra@unam.mx
ORCID: 0009-0003-1754-0919

Miguel Zúñiga González
Dirección General de Cómputo y de Tecnologías de Información y Comunicación
Universidad Nacional Autónoma de México
mzuniga@unam.mx
ORCID: 0009-0007-3305-5496

Resumen

Se diseñó una interfaz gráfica para la plantilla general de actividades del curso en Moodle, en formato de Ambiente Virtual de Aprendizaje, del Taller Integral de Arquitectura I y II del taller "Jorge González Reyna" de la Facultad de Arquitectura de la Universidad Nacional Autónoma de México, basada en el modelo sistemático de Bruce Archer. El equipo de diseño realizó una serie de actividades de diagnóstico con el fin de contextualizar la identidad visual del taller. A partir de esta información, se desarrollaron *wireframes* y prototipos iniciales que fueron validados en un proceso iterativo con los distintos equipos participantes. Se definieron criterios gráficos como la disposición de los elementos, el uso de tipografía y una paleta de color para facilitar la lectura, navegación y coherencia visual entre los componentes. Para asegurar la legibilidad del contenido, se utilizaron herramientas digitales que permitieron validar el contraste entre texto y fondo, conforme a las pautas de accesibilidad web. Esta validación también contribuyó a

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 82 - 107

optimizar el desarrollo técnico, al reducir el número de clases CSS necesarias para su implementación en la plataforma educativa. El resultado fue una plantilla funcional, accesible y estéticamente alineada con los objetivos institucionales y pedagógicos del taller, a pesar de las limitantes de tiempo disponible, inconsistencias en la interpretación de CSS entre navegadores y la necesidad de trabajar en paralelo con las diseñadoras instruccionales que montaban los contenidos en cada curso.

Palabras clave: Accesibilidad web, interfaz gráfica, metodología Bruce Archer, Moodle.

Abstract

A graphical interface was designed for the general activities template of a Moodle-based virtual learning environment for the "Taller de Arquitectura Integral I y II" (Comprehensive Architecture Workshop I and II) subject, offered by the "Jorge González Reyna" Workshop at the Facultad de Arquitectura of the Universidad Nacional Autónoma de México, based on Bruce Archer's systematic model. The design team carried out a series of diagnostic activities to contextualize the workshop's visual identity. From this information, wireframes and initial prototypes were developed and validated in an iterative process with the various participating teams. Graphic criteria were defined, such as the arrangement of elements, the use of typography, and a color palette to facilitate reading, navigation, and visual consistency among components. To ensure accessibility, digital tools were used to validate the contrast between text and background, guaranteeing the readability of the content in accordance with web accessibility guidelines. This validation also helped optimize the technical development by reducing the number of CSS classes required for implementation on the educational platform. The result was a functional, accessible template that was aesthetically aligned with the workshop's institutional and pedagogical objectives, even with time constraints, inconsistencies in interpreting CSS between browsers, as well as the need to work in parallel with the instructional designers who assembled the content for each course.

Keywords: *Bruce Archer methodology, graphical interface, Moodle, web accessibility.*

1. INTRODUCCIÓN

El ambiente para el Taller Integral de Arquitectura I y II de la Facultad de Arquitectura fue una de las iniciativas que impulsaron la puesta en marcha del proyecto general "Ambientes Digitales de Aprendizaje" (ADA), desarrollado por la Dirección General de Cómputo y de Tecnologías de Información y Comunicación de la Universidad Nacional Autónoma de México (DGTIC). Los ADA son ambientes o espacios virtuales de aprendizaje (Díaz Díaz & Castro Arévalo, 2017), que tienen el objetivo de ofrecer a los docentes la posibilidad de proponer y realizar actividades mediante el desarrollo de una planeación didáctica (San Vicente Parada, 2021). Dichas planeaciones están disponibles para su consulta por parte de los estudiantes, otros docentes y público en general desde cualquier dispositivo conectado a Internet. Los contenidos se encuentran licenciados mediante *Creative Commons*, lo que permite a profesores interesados incorporarlos parcial o completamente en sus asignaturas o estrategias particulares de enseñanza-aprendizaje. En esta colaboración entre profesores, diseñadoras instruccionales y equipo tecnológico, se realizaron diferentes propuestas de ambientes a partir de las situaciones de aprendizaje del taller, en el cual los diseñadores gráficos fueron los responsables de generar y consolidar la

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 83 - 107

identidad visual de éste, el estilo de navegación y la presentación de componentes de cada actividad a implementarse en el gestor de contenidos educativos Moodle.

Cabe mencionar que los ADA desarrollados en Moodle conservan la estructura de navegación de un curso estándar en esta misma plataforma: una disposición general semejante en las portadas y plantillas de actividades, utilizando colores e iconografía adaptada a las diferentes áreas de conocimiento y entidades de la UNAM. Además, sus contenidos se muestran de forma abierta, sin necesidad de un usuario y contraseña. Para cada propuesta de ADA, se cuidaron los estándares de accesibilidad y buenas prácticas de Moodle en la presentación de actividades (CAST, 2025; Pineda, Valdivia y Ciraso, 2015; Yunos, Muslim, Hussain, 2024). Entre éstas, se incluyen la integración de los títulos de secciones que permite generar un índice de contenidos accesible para los lectores y asistentes de pantalla; la disposición con mayor o menor detalle en la descripción de actividades para el estudiante; y la incorporación de un bloque para docentes que ofrece recursos explicativos de la motivación de las actividades, sugerencias de implementación y consideraciones para adaptarlas a cada grupo específico. Cada ADA se diseñó como un paquete flexible que puede funcionar como un proyecto, un caso específico, un módulo parcial o un curso completo. Además, estos ambientes permiten obtener una copia, exportarlos a otros entornos Moodle y extenderse, modificarse y conducirse sin herramientas o conocimientos técnicos adicionales.

En el caso concreto del ADA del Taller de Arquitectura I y II, se planteó la alternativa de desarrollar una propuesta de diseño de interfaz a la medida y personalizada. Esta decisión surgió ante la inquietud de los profesores, quienes consideraban que Moodle era muy rígida visualmente para sus necesidades pedagógicas, para el proceso creativo e iterativo, así como para la dinámica en las sesiones presenciales del taller, que conjuntan tres cursos en un solo taller. Además, debían gestionar la cantidad de material disponible y la secuencia con que querían presentarlo, lo cual era una oportunidad inigualable de mostrar las posibilidades y niveles de personalización de la interfaz que ofrece la plataforma y los *frameworks* gráficos que incluye.

Con el propósito de generar la interfaz gráfica pertinente, se realizaron visitas guiadas a las instalaciones de la Facultad de Arquitectura y se asistió a clases del taller de la materia, se mantuvieron pláticas con los docentes y se revisaron publicaciones sobre temas relacionados con la arquitectura en su proceso de composición. Esto tuvo la finalidad de contrastar los estilos de diseño editorial y multimedia más usados en esta área para sensibilizar al equipo de diseño sobre el ambiente, la comunidad estudiantil de la facultad y todos los aspectos que pudieran servir de referente para tener un mayor acercamiento al proyecto.

La identidad visual del taller contempló el diseño de:

- Un logotipo, imagotipo o isologo.
- Portada de la Facultad de Arquitectura
- Portada de acceso al taller
- Programa general y desglose de actividades
- Plantilla de actividades

Para la definición de estos elementos, se consideraron fundamentos de diseño aplicados para el entorno de la web vinculados con la experiencia de usuario UX y la interfaz de usuario UI (Bidart, 2023). El desarrollo

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 84 - 107

de cada uno de estos conceptos requirió de un proceso específico para cumplir con un objetivo particular de comunicación visual (Paredes-Calderón & Nájera-Galeas, 2020).

El objetivo del ambiente virtual de aprendizaje del Taller Integral de Arquitectura I y II fue construir dos aulas virtuales en Moodle donde se guiara al estudiante en su aprendizaje de forma auxiliar y durante el desarrollo del taller, pero, además, que sirviera de fuente de consulta para el público interesado, en una propuesta que intercala contenidos disciplinares, ejemplos de trabajo e ideas y referencias para profundizar en aspectos alrededor de los proyectos estructurados en la modalidad de Aprendizaje Basado en Proyectos (ABP). Para lograrlo, la Plantilla de actividades del Taller integral de Arquitectura I y II, objeto de este trabajo, tuvo el propósito de dar un marco de referencia para que las diseñadoras instruccionales y tecnopedagógicas pudieran trasladar los contenidos del temario en secuencias de enseñanza-aprendizaje, donde ellas, los profesores titulares del taller y la gente interesada en el Ambiente Virtual de Aprendizaje pudiera modificarlo y extenderlo utilizando los editores Atto o TinyMCE, que incluye Moodle de paquete, sin conocimiento de HTML o CSS. Para este fin, la propuesta de plantilla se comprobó mediante el recurso Comprobador de contraste de la página web "WebAIM" (*Webaim,* s. f.), el cual valida la legibilidad de los contenidos, así como su traslado en una plantilla duplicable y editable que pueda modificarse sin necesidad de conocimiento específico de CSS y HTML. De esta manera, se pueden aprovechar los elementos con algún tipo de interacción o efectos visuales sin temor a romper su código e interactividad.

2. METODOLOGÍA

Para el desarrollo de la solución gráfica de la plantilla de actividades, se tomó como referencia la metodología sistemática de Bruce Archer (Song & Wang, 2025), llamada "Modelo sistemático para diseñadores". En ésta, se propone "Seleccionar materiales correctos y darles forma para satisfacer las necesidades de función y estética, dentro de las limitaciones de los medios de producción disponibles" y consta de tres fases principales: identificar el problema, definir criterios, así como analizar opciones y seleccionar la mejor solución para su implementación, como se muestra en la Figura 1.

Figura 1

Esquema de las fases de la metodología de Bruce Archer

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 85 - 107

Esta metodología se adaptó a la naturaleza del proyecto en un contexto educativo, en donde la interfaz gráfica debería cumplir con criterios funcionales, técnicos, pedagógicos y estéticos.

2.1 FASE ANALÍTICA

En esta fase, se identificaron las necesidades y las limitaciones tecnológicas de la plataforma Moodle, obteniendo la información siguiente:

- Se debía generar un ADA para el taller "Jorge González Reyna", de la carrera de Arquitectura en la UNAM.
- El nombre del taller sería "Taller Integral de Arquitectura" curso I y II, impartido en el primer año de la carrera.
- El taller debería estar disponible en línea, accesible para consulta de la comunidad universitaria en todo momento.
- El diseño de la interfaz gráfica debería integrar el encabezado y pie institucional acorde a los lineamientos gráficos para los sitios web de la DGTIC (Dirección General de Tecnología de la Información y Comunicaciones, 2022).
- El concepto gráfico de la identidad visual del taller debería tener referencias visuales con la Facultad de Arquitectura y el Taller "Jorge González Reyna" con el fin de lograr una identificación y sentido de pertenencia por parte de los alumnos.
- El taller se conforma de tres componentes: Investigación, Proyectos y Construcción, los cuales contienen secuencias didácticas llamadas "Actividades", vinculadas de manera integral en el curso.
- La información de cada actividad se desplegaría en una sola página.
- El equipo tecnológico implementaría la plantilla a partir de la definición del diseño de interfaz en la plataforma Moodle mediante el uso del *framework* Bootstrap. Dicho *framework* permite incorporar un diseño de interfaz gráfica diferente a la que presenta la plataforma por defecto bajo un marco responsivo (Chávez Calderón, et al., 2022), pues se basa en el uso de HTML y hojas de estilo CSS.
- El total de actividades sería de 37 para el primer curso y 39 para el segundo curso, resultado del trabajo realizado por los docentes y los especialistas en didáctica.

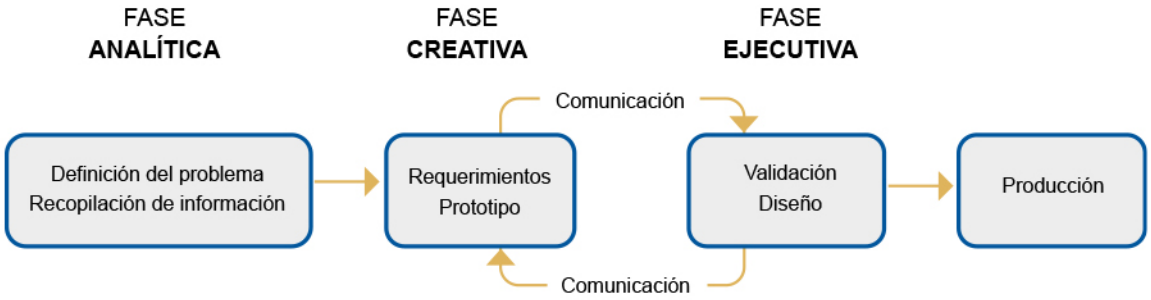
2.2 FASE CREATIVA

En esta fase, se utilizó una de las herramientas de usabilidad en el diseño de interfaces para la elaboración de *wireframes* (Murillo, 2016) y prototipos a nivel de imagen. Estos se presentaron al equipo didáctico y tecnológico con el fin de obtener realimentación y realizar ajustes necesarios para su validación con los docentes, asegurando la pertinencia pedagógica del diseño, como se muestra en la Figura 2.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 86 - 107

Figura 2

Esquema sobre el proceso iterativo



Consecuentemente se revisaron los documentos de texto de las actividades con la finalidad de identificar el tipo de información contenida, como se muestra en la Figura 3, comparando unas con otras para poder establecer elementos en común e importancia de los mismos, y así poder bocetar las primeras ideas. El resultado fue una propuesta de *wireframe* para validar la arquitectura, organización y jerarquización de la información, como se muestra en la Figura 4, así como la elaboración de un primer prototipo para mostrar a los docentes un primer acercamiento a lo que podría ser la plantilla de actividades, como se puede ver en la Figura 5.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 87 - 107

Figura 3

Actividad desarrollada en texto plano

Actividad 2.3
Cimentación, castillos, dalas, trabes

Objetivo de aprendizaje:
- Identifica los conceptos y procedimientos relativos a los diferentes tipos de cimentación.

Producto esperado:
- Respuesta de preguntas guía

Recursos / Materiales:
- Documento de "Apuntes de Construcción"

Forma de trabajo:
- Individual y grupal

Indicaciones:

1. Revisa con tu docente los siguientes elementos de la cimentación y la proyección de imágenes que los ilustran:
   - Excavación – a mano – con maquinaria.
   - Protección de colindancias.
   - Revisión del trazo y ubicación de ejes.
   - Zarpeo de cepas.
   - Contención de cepas o taludes.
   - Elaboración de plantillas.
   - Colocación de silletas, para recibir armados.
   - Armado de contratrabes, trabes de liga y dalas.
   - Cimientos de mampostería (piedra).
   - Cimientos de concreto ciclópeo.
   - Cimientos de concreto armado.
   - Cimientos corridos (zapatas).
   - Cimientos aislados (zapatas).
   - Losas de cimentación.
   - Colados de elementos estructurales.
   - Anclajes para columnas de acero.
2. Utiliza la tabla de dosificación de concreto que buscaste en internet, determina con tus compañeros qué mezcla tendrías que preparar para colar una cimentación.
3. Revisa las siguientes páginas de la 41 a la 44, 53 a 57 y 58 a 61 del documento "Apuntes de Construcción" para comentarlas en la siguiente clase.

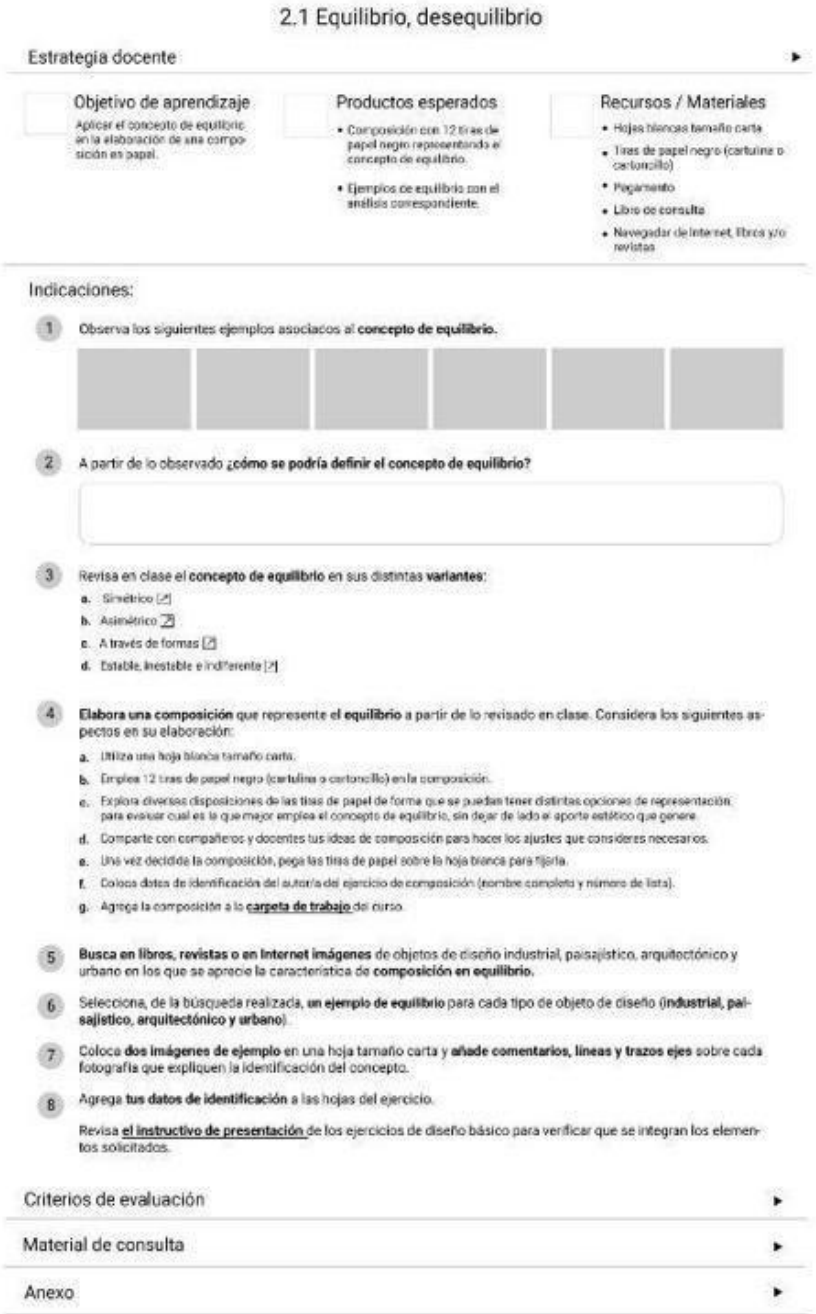
Autor del ejercicio: Arq. Ernesto L. Natarén

Criterios sugeridos de evaluación

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 88 - 107

Figura 4

Primer wireframe de la actividad



https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 89 - 107

Figura 5

Primera propuesta de prototipo para intercambiar puntos de vista entre el equipo de trabajo y los docentes con el fin de definir los alcances del diseño

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 90 - 107

De esta revisión, se tomaron decisiones que delimitaron aspectos que impactaron en la definición del diseño de la plantilla de las actividades, considerando que se integrarían un total de 76 actividades, además de incluir algún elemento que visualmente identificara el tipo de componente al que pertenecían, manteniendo la coherencia visual con las demás páginas del taller.

Aunque en algún momento se planteó la posibilidad de generar una plantilla con algunos elementos en las actividades con un diseño personalizado, se descartó de inmediato esta posibilidad debido a las limitaciones de no contar con los recursos humanos técnicos suficientes ni el tiempo para implementar esa alternativa de acuerdo con el cronograma del proyecto.

Una vez definidos los requerimientos y las necesidades, el equipo de diseño revisó los principios de UX y UI relacionados con la arquitectura de la información, los principios de usabilidad de Nielsen Nielsen (Panimboza Deleg, 2025), principios de percepción de la Gestalt (Olalde Ramos, 2024) y principios de accesibilidad web a considerar para el desarrollo de las propuestas de interfaz gráfica de la plantilla.

2.3 FASE DE EJECUCIÓN

Los profesores del taller "Jorge González Reyna" de la Facultad de Arquitectura de la UNAM integraron los tres cursos que lo componen, investigación, proyecto y construcción; en propuestas modulares de proyectos arquitectónicos con diferentes grados de desarrollo en términos de entregables: desde análisis de la forma y volumetría, un partido arquitectónico (el concepto fundamental, la visión y la intención del arquitecto, que guía el diseño de un edificio) y un proyecto ejecutivo, como lo ocupan los profesionales de la construcción (Natarén, 2022). En la página principal del curso, se muestran los proyectos integrales en términos de la seriación de actividades para llegar a los resultados esperados. Asimismo, en cada una de las actividades conformantes, se asignaron códigos de color, tanto en la pleca del encabezado como en los contornos de los bordes de las tablas, para identificar las actividades que correspondían a cada curso del taller. De esta manera, al final de cada proyecto, el calificador de Moodle permite obtener los promedios de los tres cursos que componen el taller para cada proyecto, los cuales corresponden administrativamente a la calificación de cada parcial del primer y segundo semestre de la carrera de Arquitectura, donde se imparte el taller.

Esta dinámica hizo patente el reto de diferenciar visualmente las actividades de los diseños instruccionales de cada curso del taller, integrando las instrucciones específicas, los criterios y rúbricas de evaluación de cada una y su contribución al proyecto que componen, para que los estudiantes pudieran realizarlas y entregarlas en tiempo, aún sin haber estado presentes en clase cuando éstas fueron descritas. Para contribuir a la claridad en el alcance de los entregables, la plantilla debía ser lo suficientemente adaptable para reflejar de forma rápida el curso al que pertenecían, los objetivos y las instrucciones con ejemplos de proyectos semejantes.

La plantilla de actividades contempló estas necesidades de diseño, integrando los aspectos técnicos de implementación con las decisiones gráficas para generar la propuesta final de las diferentes plantillas. Cada una contempló las secciones pertinentes según los requerimientos de cada actividad particular, tales como carruseles, galerías, imágenes colapsables, etcétera. Asimismo, se adaptaron al *framework* de desarrollo web seleccionado, se optimizó el código y se alineó a las clases CSS del *framework* correspondientes, asegurando la accesibilidad tanto en aspectos de legibilidad como en coherencia visual. De este modo, se generó una propuesta que respondió a los principios de UX y UI que persigue este proyecto.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 91 - 107

Al momento de elegir el *framework* de trabajo, Moodle disponía de tres marcos de construcción de interfaces y de carga de AJAX: jQuery UI, el Moodle *UI Component library* y Bootstrap, todos con patrones para adaptar pantallas de la interfaz de Moodle y con comunidades grandes y consolidadas. Aunque JQuery proporciona más efectos visuales y facilita la inclusión de Javascript, dando más dinamismo a las páginas, se optó por Bootstrap ya que ofrecía la posibilidad de editar directamente desde el editor HTML de elección, TinyMCE o Atto, sin necesidad de intercalar entre la vista de edición y la vista de código para las porciones interactivas.

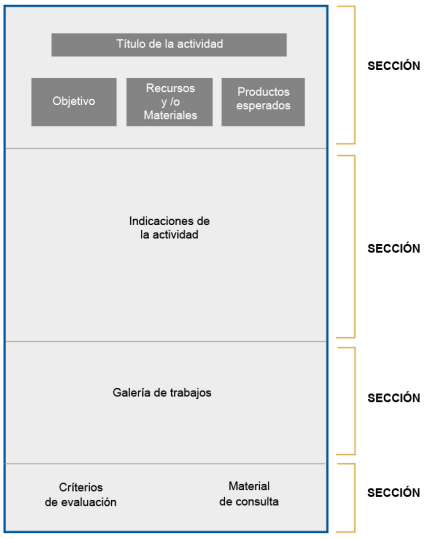
Por la experiencia obtenida en el equipo de trabajo, relacionada con la personalización de temas visuales en diferentes proyectos, se decidió que, durante la etapa de desarrollo y en paralelo a la construcción de actividades "tipo" que materializarían las propuestas de diseño, se tendrían clases generales declaradas en el tema visual con la finalidad de tener cambios de forma ágil sin entorpecer el trabajo de las diseñadoras instruccionales y tecnopedagógicas. Esto permitió cambiar estilos para revisiones sin tener que editarlos masivamente en páginas para el posterior traslado a la plantilla general de actividades. Una vez aprobada y validada la plantilla, cerca de la culminación de la primera etapa del proyecto, los estilos CSS definidos inicialmente en la sección «CSS adicional» del tema visual habilitado en Moodle, se trasladaron ya bien a estilos *inline* dentro de cada etiqueta HTML o a una zona de estilos generales en el bloque cero del curso. De esta manera, fue posible trasladar la plantilla a otros entornos Moodle sin tener que pedir al administrador de la plataforma destino tener que incluir clases adicionales en su tema visual. En el apartado 2.4, se describe ese proceso a nivel operativo.

2.3.1 DISPOSICIÓN

Con base en la propuesta de *wireframe* y las reuniones con los docentes, se integraron elementos para facilitar la identificación de cada tipo de actividad, así como la agrupación de la información en cuatro secciones principales, como se ejemplifica en la Figura 6.

Figura 6

Wireframe de la página de actividades



https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 92 - 107

Esta disposición se adapta tanto a monitores anchos como a pantallas móviles sin alterar significativamente sus secciones, cumpliendo tanto con pautas de navegación y accesibilidad web, con un diseño responsivo y acorde a los requerimientos del proyecto y de Moodle.

2.3.2 INTEGRACIÓN DE IDENTIDAD GRÁFICA

Considerando que el contenido de la actividad eran instrucciones en texto plano, y algunas de ellas eran de considerable extensión, se decidió crear un concepto gráfico minimalista que visualmente pudiera apoyar al usuario en la identificación de una separación y la diferenciación entre las secciones, para una lectura más ágil del contenido.

De esta manera, en la propuesta se integraron trazos de líneas rectas con formas simples para las secciones con información complementaria de la actividad y para la sección de "indicaciones", se asignó un color de fondo sólido para destacar la misma, generando un área de mayor atención visual y jerarquía que, a su vez, serviría como el elemento diferenciador entre las actividades de los tres componentes de cada curso, lo anterior, reforzado con la integración de una pleca en la barra superior del encabezado con el nombre del componente y asignación del color de fondo correspondiente.

2.3.3 TIPOGRAFÍA

Se optó por utilizar dos fuentes tipográficas para diferenciar y jerarquizar los elementos textuales, agregando un estilo personal a los textos, ya que el 90% del contenido de la plantilla estaría conformado por texto plano.

- **Roboto:** se definió como la fuente primaria en sus variantes *light* y regular, que, por su trazo, es una de las fuentes más utilizadas y recomendadas para los bloques de texto, además de que se considera como una fuente muy flexible y adaptable a diversos dispositivos.
- **Barlow:** se definió como la fuente secundaria para integrar los títulos y subtítulos que, por su trazo regular, redondeado y alargado, presentan una semejanza visual con la tipografía usada en los pies de planos de los arquitectos.

2.3.4 PALETA DE COLOR

La paleta de color se basó en tres colores para mantener una relación cromática con la utilizada en los sitios web institucionales de la DGTIC, como se muestra en la Figura 7.

Figura 7

Paleta de color

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 93 - 107

2.3.5 GENERACIÓN DE LA PALETA DE COLOR COMPLEMENTARIA PARA LAS ACTIVIDADES DE LOS COMPONENTES

Una vez definida la propuesta, el reto a resolver era encontrar una paleta de colores para el fondo de la sección "Indicaciones" de la plantilla de actividades correspondiente a los tres componentes del curso I y a los tres componentes del curso II. Esta selección debía garantizar un contraste suficiente con respecto a los colores definidos para la tipografía, pues, de lo contrario, habría sido necesario buscar y definir nuevos colores para la tipografía de las actividades de cada componente, ya que cada uno tiene un color de fondo asignado.

Para poder definir los seis colores de fondo, se usó la herramienta web "Comprobador de contraste" de la página WebAIM (Webaim: comprobador de contraste, s. f.), la cual permite hacer una comparación de códigos de color entre el color de fuente y fondo, verificando que exista un contraste suficiente para que sean legibles los textos. Cabe mencionar que actualmente existen diversas herramientas web de acceso gratuito para hacer estas validaciones, sin embargo, se eligió ésta por el nivel de resultados que muestra, por lo que se hicieron diversas pruebas hasta encontrar los colores adecuados para la validación de contraste, atendiendo las recomendaciones de las pautas de accesibilidad para el contenido en la web *Web Content Accessibility Guidelines* WGAC 2.1 (World Wide Web Consortium, 2018).

2.4 CODIFICACIÓN DE PLANTILLAS

Entre las posibilidades de construcción para las plantillas que dispone Moodle o pueden incorporarse, se eligió Bootstrap, un marco de referencia con herramientas para prototipar y estandarizar interfaces Web de grandes proyectos. Esta selección se fundamentó gracias a su arquitectura modular y personalizable, que permite importar sólo los componentes necesarios en cada ocasión y habilitar opciones globales como degradados y sombras, CSS con variables estandarizadas y una importante cantidad de funciones, mapas y efectos, que funcionan entre diferentes navegadores (Bootstrap Team, 2023). Además, Bootstrap tiene la ventaja de estar incluido en el paquete de distribución estándar de Moodle, tiene una considerable cantidad de ejemplos de uso en la red y una comunidad grande de desarrolladores que comprueban su compatibilidad con numerosos equipos de escritorio y móviles, lo que facilita la construcción de vistas y sitios Web.

Se retomó uno de los aspectos definidos en la fase analítica, en el que se menciona incluir el encabezado y pie de página institucional conforme a los lineamientos de imagen para los sitios web de la DGTIC, práctica que en el equipo de trabajo se ha implementado en otros proyectos con Moodle, como Tu aula virtual, Formación Académica, Retos TIC, etcétera, se adaptó el encabezado y pie de página a esta disposición. Ahora bien, hacia el interior de los cursos o ADA, los profesores editores deciden cómo se integrarán los elementos dentro de éste. Aunque tienen relativamente pocas opciones para cambiar el desplegado de las actividades, pueden decidir colocar los enlaces a las mismas en forma de lista, bloques, gradilla o mosaicos, lo que cambia el formato de curso. De igual forma, tienen la posibilidad de intercalar contenido enriquecido mediante Etiquetas de Moodle (*mod_label*) llamadas Zonas de texto y medios desde Moodle 4.0.

En el interior del curso, tanto en la página principal como en las páginas de actividades y plantillas, los editores HTML impiden de manera predeterminada incluir etiquetas <script>, <embed>, <style> e <iframe>, con el fin de evitar cualquier forma de secuencias de comandos entre sitios potencialmente

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 94 - 107

maliciosa (*cross-side scripting*), pero se puede configurar para que ciertos perfiles puedan incrustarlas mediante la opción de seguridad del sitio «Habilitar contenido confiable», disponible en Administración del sitio > Seguridad > Políticas de seguridad del sitio Moodle. (Moodle HQ, s. f.). Con esta opción habilitada, se agregó una etiqueta <style> en la sección de encabezado de la página de la actividad base con clases para que se dispusiera del cien por ciento de la página del curso y así mostrar las secciones pertinentes con fondo de color del sitio a pantalla completa.

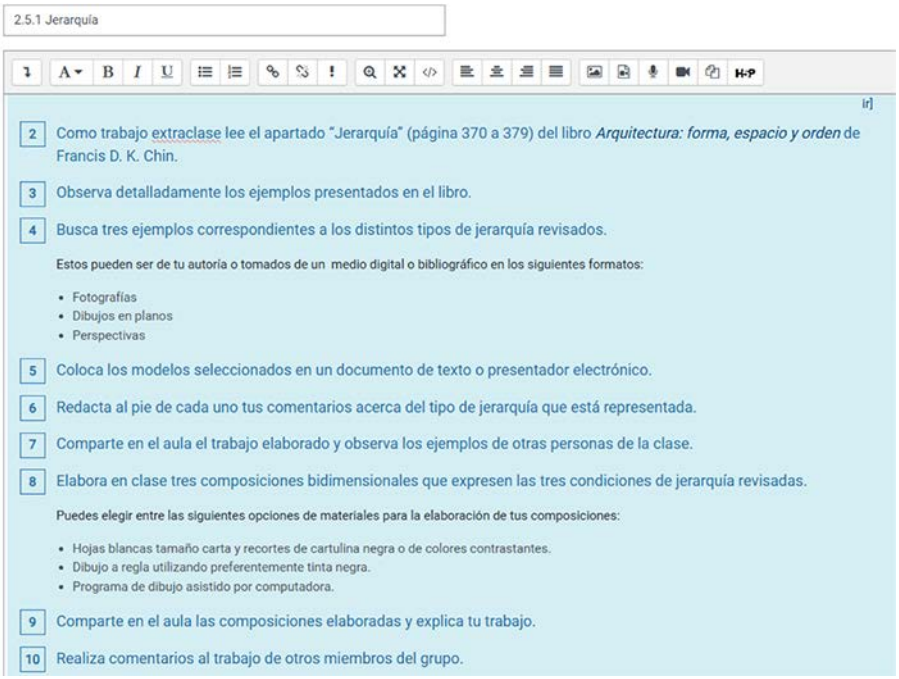
Como se comentó en el apartado de ejecución, durante el desarrollo y consolidación de las plantillas de actividades, se utilizaron clases generales de Bootstrap y se mantuvo la convención de la biblioteca de componentes de Moodle (como los nombres de categorías y de cursos declarados en cada etiqueta <body>) para minimizar la personalización de clases adicionales para conseguir efectos o desplegados específicos. Una vez aprobada la plantilla general de actividades que comprendieron el proyecto, se trasladaron los estilos personalizados a etiquetas <style> en la primera parte de cada respectiva página de actividad, o bien, en los bloques cero o encabezados de curso, con el objetivo de poder trasladar los cursos o parte de ellos a otros sitios Moodle. Así, se consiguió que todas y cada una de las actividades se puedan modificar mediante los editores HTML de Moodle desde el navegador web, sin necesidad de modificar los roles de profesores editores, de incrustar código ni de intercalar la vista de diseño y la de código HTML fuente mediante la edición, como se aprecia en la parte derecha de la Figura 8, pues, al alinearnos a las convenciones de Moodle, las plantillas son compatibles con las disposiciones de desplegado de Moodle.

Se agregaron clases estandarizadas para los componentes de cada sección de página a desplegarse y se cuidó que el código HTML de las plantillas permaneciera al editarse desde las pantallas de Atto y de TinyMCE.

Figura 8

Detalles de plantilla de actividades, con un bloque colapsable de Bootstrap (izquierda) que puede editarse directamente desde el editor HTML de Moodle sin entorpecer (derecha)

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 95 - 107



Se ocupó el contenedor estándar de Bootstrap (con la clase *container*, diferente a *fluid-container*) para la plantilla general de actividades, marcando las zonas para poder agregar secciones. Esta selección se fundamentó en que éste permite manejar anchos fijos según la resolución de cada pantalla, no de manera fluida a nivel de píxel, sino en secciones de anchos de pantalla (Bootstrap Team, 2023). Para los efectos de muestra de secciones, se ocupó el efecto *collapse* estándar de Bootstrap (Bootstrap Team, 2022), como se muestra en la Figura 8 arriba, obteniendo elementos editables desde la interfaz web. Para las galerías de trabajos, se implementó un carrusel con miniaturas en la parte inferior que facilita la selección de elementos individuales (Bootstrap Team, 2022), ajustado a la rejilla de Bootstrap, lo que permite adaptar todos los elementos a diferentes tamaños de pantalla. Las únicas clases adicionales que se agregaron fueron los bordes de las plantillas de instrucciones, el formato para las viñetas numeradas y la sección de objetivos, recursos y productos, que, si bien se formaron en un contenedor estándar de Bootstrap, dichas clases permiten personalizar el espacio de relleno (*padding*) y la posibilidad de mostrarlo adaptativamente según el ancho de pantalla mediante una tabla, por compatibilidad CSS en navegadores anteriores, que puede mostrarse en fila en pantallas reducidas.

Para mostrar los cuadros de la tabla en forma de lista, se colocaron anchos relativos en los cuadros externos (*width: 30vw;*) que, en tamaños de pantalla pequeños, se ajustan al 100% con márgenes a la izquierda y derecha. Para mantener la proporción de cuadros internos, al que contiene el texto dentro de cada cuadro se le agregó el atributo de «aspect-ratio: 1», ancho de bloque completo (*width: 100%;*) y desbordamiento automático (*overflow: auto;*), de tal modo que, en los casos donde el texto ocupe más espacio del disponible, aparezca una barra de desplazamiento y se mantenga el tamaño de cuadrado, tal cual se aprecia en la Figura 9. Un ejemplo completo del código del esquema de la plantilla se encuentra en la figura del Anexo A de este documento.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 96 - 107

Figura 9

Detalle de desplegado de cuadrados en las secciones de objetivos, que se mantengan cuadrados sin importar el ancho de la pantalla

3. RESULTADOS

Como resultado de la validación de contraste, se obtuvieron los seis colores requeridos, cumpliendo con el nivel AA de accesibilidad web que permite la legibilidad de textos de tamaño normal y grande con respecto al color de fondo, como se muestra en la Figura 10.

Figura 10

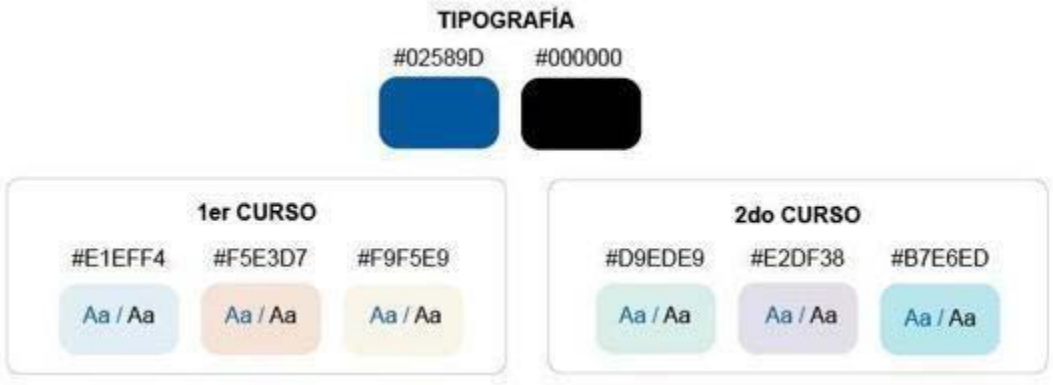
Captura de pantalla de la herramienta de validación de colores

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 97 - 107

Al tener definidos los seis colores con su respectivo código hexadecimal, únicamente se requirió definir seis nuevas clases para la hoja de estilos CSS correspondiente a la plantilla de actividades, ya que se mantuvieron los mismos colores definidos para las fuentes tipográficas, ver Figura 11.

Figura 11

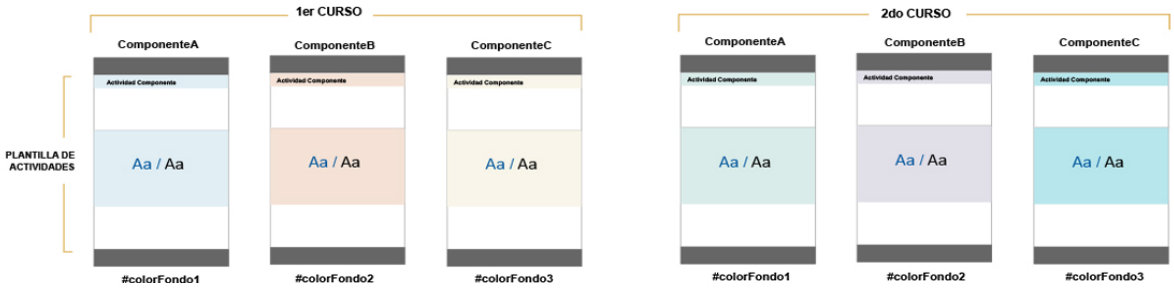
Paleta de color de fondos para las actividades de los dos cursos



De esta manera, el personal asignado para la integración de cada actividad solamente tuvo que hacer la referencia a la clase con el color de fondo correspondiente a la actividad y, para el equipo de diseñadoras instruccionales, fue transparente, de tal modo que la dinámica de integración fue la misma para todas las actividades, proceso que redujo considerablemente el tiempo de integración, como se muestra en la Figura 12.

Figura 12

Esquema de representación de asignación color de fondo de cada componente de ambos cursos



El equipo de diseñadoras instruccionales y tecnopedagógicas evaluaron los resultados del material y diseño en tres dimensiones: técnica, usabilidad y percepción del usuario (García García, González Alarcón y Martínez Sánchez, 2024) mediante la comparación de impresiones y sugerencias en encuestas semiestructuradas entre dos generaciones de estudiantes, quienes terminaban el curso anterior y quienes tomaban el curso propedéutico previo al inicio del primer semestre, donde se ocupó una copia de estos cursos. En el estudio, participaron los 114 estudiantes que tomaron o comenzaban el taller de

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 98 - 107

arquitectura con los profesores Natarén, Quezada y Cueto Meza, así como 121 estudiantes de las mismas dos generaciones de grupos de los otros profesores que componen el taller Arq. Jorge González Reyna. A partir de los resultados obtenidos, los profesores manifestaron su deseo de incorporarse y aportar con materiales y variaciones a una siguiente revisión de los ambientes virtuales.

Gracias al número de reuniones donde se presentaron avances a los diferentes equipos, se documentaron problemas técnicos y se superaron inconsistencias en la herencia de estilos CSS y diferencias con estilos predeterminados de Moodle entre el sitio de ADA y en Tu aula virtual, donde se restauró una copia de los cursos.

Para evitar dejar clases globales en el resultado final a exportarse y compartirse o utilizar etiquetas <style> al inicio que pudieran omitirse al exportar el curso, se revisó que la opción de contenido confiable (Moodle HQ, s. f.) permitiera mostrar la personalización del curso en otras instalaciones de Moodle mediante atributos *style* en las etiquetas de cada página. Esta operación fue ciertamente más tardada, pero asegura que los estilos se desplegarán correctamente en cualquier sitio donde se importen y que se pueden editar tanto con Atto, como con TinyMCE, los editores que tiene disponibles Moodle.

La aplicación de la metodología de Archer demostró ser útil para estructurar el proceso técnico y justificar decisiones basadas en evidencia. Su enfoque racional y secuencial favoreció la documentación y comunicación entre equipos multidisciplinarios. Sin embargo, su estructura lineal exigió ajustes para integrarse con procesos de validación iterativa, típicos de los entornos digitales (Reinoso Yunga, & Zhindón Duarte, 2023). Comparada con metodologías como *Design Thinking*, Archer aportó mayor trazabilidad y rigor técnico, aunque menor agilidad en la ideación.

El rediseño de la plantilla de actividades basado en la metodología de Bruce Archer permitió mantener las pautas de accesibilidad que incluye Moodle y mejorar la legibilidad en las propuestas de distribución conforme a estándares internacionales (WCAG 2.1), reducir tiempos de carga y simplificar el código CSS, al tiempo que se incrementó la satisfacción y eficiencia de navegación de los usuarios. La solución implementada representa una práctica replicable para otras áreas académicas dentro de la UNAM.

4. CONCLUSIONES

En el lado del diseño, se creó una identidad visual coherente con la Facultad de Arquitectura, alineada con los principios de UX y UI, dentro de las limitaciones de tiempo, recursos humanos y medios tecnológicos disponibles.

Con base en los resultados obtenidos de la aplicación del proceso metodológico para el diseño de la plantilla de actividades del Taller Integral de Arquitectura, se identificaron necesidades específicas mediante el trabajo colaborativo entre diseñadores, pedagogas y equipo técnico. Se generaron prototipos que fueron validados de manera iterativa, cuidando la jerarquización de la información, el uso de elementos gráficos diferenciados por tipo de actividad y un uso correcto del color para garantizar la legibilidad, obteniendo una plantilla adaptable, accesible y visualmente consistente, que facilitó su implementación en estilos CSS persistentes y transparentes en la edición para diseñadores instruccionales y docentes mediante la interfaz web. Con la validación del contraste de color, se aseguró el cumplimiento de accesibilidad web AA, sin comprometer el aspecto visual del diseño.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 99 - 107

Por la parte técnica, el trabajo se centró en valorar las alternativas que ofrece Moodle para implementar las vistas generales y los componentes interactivos en cada propuesta de actividad y de secuencia de aprendizaje. Se buscó que las plantillas y sus secciones tuvieran etiquetas que cumplieran con las disposiciones de accesibilidad y sus atributos estuvieran dentro del espectro válido de Moodle; esto garantizó que, al utilizar cualquiera de los editores HTML de Moodle, se mantuviera la edición y se pudieran agregar, cambiar o eliminar partes de cada actividad sin necesidad de conocer HTML, CSS o Javascript ni alternar entre la vista de edición estándar y la de código fuente.

De igual manera, se ordenaron las actividades en el libro de calificaciones de Moodle para que se pudiera obtener agrupadamente las calificaciones de cada curso que compone cada Taller de Arquitectura Integral por parcial, mediante ponderación de categorías dentro de una misma aula en Moodle. Al utilizar solamente tipos de actividades existentes en la distribución estándar de Moodle, se maximizó la posibilidad de que pudieran exportarse y utilizarse en otros sitios Moodle, como se comprobó con la copia que se realizaron de los ADA a Tu aula virtual para ser utilizados en los talleres curriculares de los ciclos a partir de 2024-1. La elección de Bootstrap fue la que el mismo equipo de desarrollo de Moodle eligió en 2024 como marco preferido para el desarrollo de vistas en sus módulos estándar y para agregados de terceros (Moodle HQ, 2024). En los últimos dos años, se han publicado módulos para extender los editores HTML de Moodle y aprovechar elementos de Bootstrap incrustables y utilizables sin necesidad de utilizar código, como Bootstrap Elements (*mod_bootstrapelements*), HTML Bootstrap Editor (Tool) (*tool_htmlbootstrapeditor*) o Bootstrap Grid (*atto_bootrsapgrid*), disponibles en la página de descarga de agregados de Moodle.

Para proyectos futuros, se recomienda identificar tempranamente los requerimientos técnicos y pedagógicos, y validar los prototipos con el equipo de trabajo y docentes para reducir problemas de interpretación del contenido; así como aplicar herramientas de validación de accesibilidad web y definir avances parciales para tener un progreso escalonado. Es relevante incluir una fase de investigación participativa y trabajo colaborativo multidisciplinario con conocimientos técnicos sobre el proyecto para optimizar los resultados, entender la naturaleza del desarrollo de las actividades, retos, casos o proyectos que se van a llevar a cabo, cómo se les presentan a los estudiantes y qué productos se esperan obtener, las formas de evaluar y calificar, el tipo de acompañamiento que se realiza a lo largo del semestre y el estilo de enseñanza-aprendizaje que tienen los profesores autores, con el fin de reflejar tales aspectos en las composiciones gráficas y en las pantallas que utilicen a lo largo de sus cursos. Aunque sea un producto abierto y disponible para todo público, más que ser un repositorio de actividades, un ADA es una propuesta que le es útil a profesores y estudiantes específicos. Conocerlos ayudará a buscar soluciones encaminadas a ellos, sus necesidades y expectativas.

Declaración de contribución de autoría

Itzel Hernández Serra: Diseñadora de interfaces gráficas para web, generó la identidad visual del taller, su manual y la propuesta del logotipo. Realizó diversas propuestas de interfaz gráfica para los distintos diseños solicitados que se tomaron como base para la creación de las versiones definitivas. Desarrolló la plantilla de actividades. **Miguel Zúñiga González:** Especialista en plataformas educativas, codificador de las plantillas. Montó el Moodle que albergó el sitio de ADA productivo y de pruebas, realizó la propuesta de *framework* web para la plantilla general de actividades, homologó el desplegado de secciones y atendió las eventualidades durante la edición, publicación y copia del taller en otro sitio Moodle.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 **Vol. 4, Núm. 1.** enero-marzo 2026, págs. 100 - 107

REFERENCIAS

Bidart, M. (2023, 23 de octubre). *Productos digitales, interfaces – UX/UI* [Teórico, Licenciatura en Comunicación Visual, Facultad de Artes, Universidad Nacional de La Plata]. Repositorio SEDICI. https://sedici.unlp.edu.ar/handle/10915/161396

Bootstrap Team (2023). Grid system. *Bootstrap 4.6 Docs*. https://getbootstrap.com/docs/4.6/layout/grid/

Bootstrap Team (2022). Collapse and Carrousel Components. *Bootstrap 4.6 Docs*. https://getbootstrap.com/docs/4.6/components

CAST, Center for Applied Special Technology. (2025). Methods for response, navigation, and movement. *Design Options for Interaction, Universal Design for Learning (UDL)*. https://udlguidelines.cast.org/action-expression/interaction/response-navigation-movement/

Chávez Calderón, J. X., Zambrano Romero, W. D., Cedeño Palma, E. A., Zambrano Zambrano, D. M., y Cotera Ramírez, G. A. (2022). El Frontend: Diseño web adaptativo y diseño web responsivo para el desarrollo de aplicaciones web. *Informática y Sistemas: Revista de Tecnologías de la Informática y las Comunicaciones, 6*(1), 79-95. https://doi.org/10.5281/zenodo.6859127

Díaz Díaz, F., y Castro Arévalo, A. L. (2017). Requerimientos pedagógicos para un ambiente virtual de aprendizaje. *Cofin Habana, 11*(1), 1-13. Recuperado en 11 de agosto de 2025, de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2073-60612017000100004&lng=es&tlng=es

Dirección General de Tecnología de la Información y Comunicaciones (2022). *Lineamientos gráficos para los sitios web de la DGTIC*. Universidad Nacional Autónoma de México. https://www.tic.unam.mx/lineamientos-graficos-sitios-web-dgtic/

García García, A., González Alarcón, G. P., y Martínez Sánchez, M. E. (2024). Un ambiente digital de aprendizaje para la enseñanza de la arquitectura. Propuesta didáctica con una metodología activa. *Edunovatic 2024: Conference proceedings*. REDINE, Red de Investigación e Innovación Educativa. Adaya Press, España, 475-482. https://dialnet.unirioja.es/servlet/articulo?codigo=9970998

Moodle HQ. (2025). Styles. Core CSS organisation. *Moodle API Guides*. https://moodledev.io/docs/4.5/apis/plugintypes/theme/styles

Moodle HQ. (2024). Boost presets. *Moodle developer documentation*. https://docs.moodle.org/dev/Boost_Presets

Moodle HQ. (s. f.). *Configuraciones de seguridad del sitio -Habilitar contenido confiable*. https://docs.moodle.org/all/es/Configuraciones_de_seguridad_del_sitio#Habilitar_contenido_confiable

Murillo, D. (2016). Uso de esquemas o wireframes. *El Tecnológico*, *25*(1), 21-23. https://revistas.utp.ac.pa/index.php/el-tecnologico/article/view/67

Natarén de la Rosa, E. L. (2022). *Apuntes de Construcción*. Taller Jorge González Reyna, Facultad de Arquitectura UNAM. https://ada.educatic.unam.mx/pluginfile.php/9548/mod_assign/intro/Apuntes%20de%20Construccion.pdf

Olalde Ramos, M. T. (2024). *La buena composición visual*. Universidad Autónoma Metropolitana. https://doi.org/10.24275/uama.352.10714

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 **Vol. 4, Núm. 1.** enero-marzo 2026, págs. 101 - 107

Paredes-Calderón, B. A., y Nájera-Galeas, C. E. (2020). Comunicación visual, procesos de construcción de los signos visuales en los mensajes: Diseño y diseñador gráfico. *Dominio de las Ciencias*, *6*(2 [Especial junio]), 995-1006. https://doi.org/10.23857/dc.v6i2.1262 (dominiodelasciencias.com)

Pineda i Herrero, P., Valdivia, A., Ciraso, C. (2015). *Actividades de Moodle : manual de buenas prácticas pedagógicas*. Universitat Autònoma de Barcelona. https://core.ac.uk/outputs/78536186/

Reinoso Yunga, E. N. y Zhindón Duarte, J. A. (2023). Método investigativo-creativo aplicado al diseño de interior residencial generado mediante contenedores marítimos. *Revista de Investigación y Pedagogía del Arte*, 13, 1-16. https://doi.org/10.18537/ripa.13.06

San Vicente Parada, A. D. C. (2021). La planeación didáctica para la enseñanza en línea. SCIAS - *Educação, Comunicação e Tecnologia*, *3*(1), 158-179. https://doi.org/10.36704/sciaseducomtec.v3i1.4982

Song, Y., y Wang, M. (2025). Revisiting the systematic method of Bruce Archer: A review on design as the third culture. *The Design Journal, 28*(5), 929–946. https://doi.org/10.1080/14606925.2025.2522557

WebAIM. (s.f.). *Contrast checker*. https://webaim.org/resources/contrastchecker/

World Wide Web Consortium (W3C). (2018, 5 de junio). *Web content accessibility guidelines (WCAG) 2.1*. https://www.w3.org/TR/WCAG21/

Yunos, N., Muslim, N., Hussain, A. H., Hasim, N. A., Nazri, N. S., y Hamsan, M. H.(2024). Best Practice of Moodle Implementation for E-Learning: A Perspective of Public University Lecturers. *Journal of Ecohumanism*, *3*(5), 261–268. https://doi.org/10.62754/joe.v3i5.3899

ANEXO A. PROPUESTA DE MAQUETA DE ACTIVIDADES CON ZONAS GENERALES Y EJEMPLO DE SECCIÓN DE CARRUSEL

Figura 13

Ejemplo de código

```
<style>
.barraArq {
  color:#02589d;
  font-family: Roboto;
  font-size: 1em;
  font-weight: medium;
  min-height: 60px;
  .titb {
    display: inline-block;
    margin: 1.2em 0;
  }
  .icondoc {
    display: inline-block;
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 102 - 107

```
      vertical-align: top;
      text-align: left;
      float: right;
      margin-top: 5px;
    }
}
.bordedoc {
   border: 2px solid #09203f;
   border-radius: 5px;
   padding: 1em;
   margin: 0px auto;
}.course-2 .barraInvestigacion {
    background-color:#f9f5e8;
}
.course-2 .barraProyectos {
    background-color:#ebd7c9;
}
.course-2 .barraConstruccion {
    background-color:#e1eff4;
}
.course-2 h2 {
    color:#02589d;
    font-family: Barlow;
    font-size: 2em;
    text-align: center;
    font-weight: bold;
}
.course-2 #region-main a {
    color:#ff9933;
    font-family: Roboto;
    font-size: .8em;
    font-weight: bold;
}
.course-2 th, .course-2 td {
  text-align: center;
  vertical-align: top;
  border: #d9d9d9 solid 1px;
  padding: 2em;
}
.course-2 td > div {
   text-align: left;
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 103 - 107

```
  width: 85%;
  margin: 1em auto;
}
table.arqcuadrosup {
  td {
    width: 30vw;
    border: 0;
    .content {
      aspect-ratio: 1 / 1 ;
      width: 100%;
      margin: 0;
      padding: 0.5em;
      overflow: auto;
      h4 {
        font-size: large;
        font-family: Barlow;
        font-weight: bold;
        color: #212529;
      }
  td.mini {
    width: 4%;
  }
  td.mleft {
    border-top: #01589d 1px solid;
    border-right: #01589d 1px solid;
    border-bottom: #01589d 1px solid;
  }
  td.mright {
    border-top: #01589d 1px solid;
    border-left: #01589d 1px solid;
    border-bottom: #01589d 1px solid;
  }
  td.cinfo {
    border: #01589d 1px solid;
    padding: 2em;
    h4 {
      margin: 0 auto 1em;
    }
    .mconsulta {
      font-size: medium;
      color: rgb(34, 109, 167);
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 104 - 107

```
        font-family: Barlow;
      }
    }
  }
}
/* ocultar actividad anterior y siguiente */
.course-2 .activity-navigation {
  display: none;
}
/* fin arquitectura */
</style>

<div class=”barraArq barraProyectos”>
   <div class=”container”>
      <span><strong>Proyectos</strong></span>
   </div>
</div>
<p>
</p>
<!-- comienza acordeón -->
<div id=”general” class=”col-sm-6 bordedoc collapse” style=””>
   <h4>Estrategia didáctica</h4>
   <p></p>
   <p>Descripción de estrategia.</p>
   <p></p>
</div>
<!-- finaliza acordeón -->

<br clear=”left”>

<div class=”container”>
   <div class=”row”>
      <div class=”col-sm-12”>
         <p><br></p>
         <h4>Ejemplo de actividad</h4>
      </div>
   </div>
</div>

<br clear=”left”>
<p><br></p>
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 105 - 107

```
<!-- comienzan cuadrados -->
<div class=”container”>

  <table class=”arqcuadrosup”>
    <tbody>
      <tr>
        <td class=”mini”></td>
        <td class=”cinfo”>
          <div class=”content”>
            <h4>Objetivos de aprendizaje</h4>
            <ul>
              <li>Objetivo</li>
            </ul>
          </div>
        </td>
        <td> </td>
        <td class=”cinfo”>
          <div class=”content”>
            <h4>Productos esperados</h4>
            <ul>
              <li>Entregable</li>
            </ul>
            <ul>
            </ul>
          </div>
        </td>
        <td class=”mini”></td>
      </tr>
      <tr>
        <td class=”mini mleft”> </td>
        <td> </td>
        <td class=”cinfo”>
          <div class=”content”>
            <h4>Recursos/materiales</h4>
            <ul>
              <li>Ejemplo de material</li>
            </ul>
          </div>
        </td>
        <td> </td>
        <td class=”mini mright”> </td>
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 106 - 107

```
        </tr>
      </tbody>
    </table>
</div>

<!-- comienzan indicaciones -->
<div class=”arqfcian”>
  <div class=”container”>
    <div class=”col-sm-12 pindicaciones”>
      <h4>INDICACIONES</h4>
      <!-- Inciso 1 -->
      <div class=”row”>
        <div class=”tab”><span class=”num”><strong>1</strong></span></div>
        <div class=”right”>
          <p>Descripción del primer inciso.</p>
        </div>
      </div>
      <!-- Inciso 2 -->
      <div class=”row”>
        <div class=”tab”><span class=”num”><strong>2</strong></span></div>
        <div class=”right”>
          <p>Descripción del segundo inciso</p>
        </div>
      </div>

      <p><br clear=”left”></p>
      <p style=”text-align: right;”><strong><span style=”font-size: small;”>Autor del ejercicio:
Arq. Ernesto L. Natarén.</span></strong><br></p>
      <p style=”text-align: right”><a href=”#”>[Subir]</a></p>
    </div>
  </div>
</div>
<!-- terminan indicaciones -->

<p><br></p>

<div class=”container”>

</div>
<p><br></p>
<p><br></p>
```

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.154 Vol. 4, Núm. 1. enero-marzo 2026, págs. 107 - 107

```
<!-- comienzan cuadrados -->
<div class=”container”>

  <div class=”container”>

    <table class=”arqcuadrosup”>
      <tbody>
        <tr>
          <td class=”mini”></td>
          <td class=”cinfo”>
            <h4>CRITERIOS SUGERIDOS DE EVALUACIÓN</h4>
            <div><span class=”” style=”font-size: small;”>Conocimientos teóricos, procedimentales y actitudinales</span></span><br></div>
          </td>
          <td> </td>
          <td class=”mini mright”></td>
        </tr>
        <tr>
          <td class=”mini mleft”> </td>
          <td> </td>
          <td class=”cinfo”>
            <p><span><strong><span class=”mconsulta”>MATERIAL DE CONSULTA</span></strong>
              </span>
            </p>
            <ul>
              <li>Ejemplo</li>
            </ul>
          </td>
          <td class=”mini”> </td>
        </tr>
      </tbody>
    </table>
    <p><br></p>
    <p style=”text-align: right”><a href=”#”>[Subir]</a></p>
  </div>

</div>
```

En este ejemplo, se asume que la actividad se dio de alta en el curso con ID 2.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.157 Vol. 4, Núm. 1. enero-marzo 2026, págs. 108 - 117

Aplicación de pruebas de seguridad informática para sitios web institucionales

Application of cybersecurity testing for institutional websites

Información del reporte:

Para citar este reporte técnico:

Aguilar Domínguez, A. (2026). Aplicación de pruebas de seguridad informática para sitios web institucionales. *Cuadernos Técnicos Universitarios de la DGTIC, 4* (1), páginas (108 - 117). https://doi.org/10.22201/dgtic.30618096e.2026.4.1.157

Angie Aguilar Domínguez
Dirección General de Cómputo y de Tecnologías de Información y Comunicación
Universidad Nacional Autónoma de México
angie.aguilar@comunidad.unam.mx
ORCID: 0009-0007-7590-0678

Resumen

Como parte de los servicios institucionales proporcionados por la Dirección General de Cómputo y de Tecnologías de Información y Comunicación hacia la comunidad de TI universitaria, se cuenta con el servicio de realización de pruebas de penetración web, que forman parte de las medidas de seguridad que ayudan a robustecer la protección tanto de los sitios y los usuarios finales, como de la información que se maneja. La metodología empleada para la realización de las pruebas es una adecuación a partir de las metodologías existentes en la industria, en la que se combina la ejecución de herramientas automatizadas que se sumaron a la revisión manual por parte del equipo de trabajo. Los resultados obtenidos son ponderados de acuerdo con la calculadora *Common Vulnerability Scoring System* del *Forum of Incident Response and Security Teams*, para establecer su nivel de criticidad. Como resultado de la prestación de este servicio, se apoya a distintas áreas universitarias a reforzar la seguridad no sólo de sus aplicaciones web, sino también del entorno en el que se despliegan y de la información que manejan. La importancia de la realización de las pruebas de seguridad web puede verse en el aumento de revisiones realizadas en el periodo 2024-2025, sobre todo en las de seguimiento. Se destaca el interés por parte de las áreas universitarias en mejorar este aspecto de la infraestructura tecnológica que gestionan para beneficio de la comunidad universitaria.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.157 Vol. 4, Núm. 1. enero-marzo 2026, págs. 109 - 117

Palabras clave: Despliegue de aplicaciones web institucionales, *pentest* de aplicaciones web, seguridad informática en aplicaciones web, vulnerabilidades en aplicaciones web.

Abstract

As part of the institutional services provided by the Dirección General de Cómputo y de Tecnologías de Información y Comunicación to the university's IT community, a web penetration testing service is offered. These tests are part of the security measures that help strengthen the protection of websites, end users, and the information they handle. The methodology used for conducting the tests is an adaptation of existing industry methodologies, combining the execution of automated tools with manual review by the team. The results obtained are weighted according to the Common Vulnerability Scoring System calculator of the Forum of Incident Response and Security Teams to establish their level of criticality. As a result of providing this service, various university areas are supported in strengthening the security not only of their web applications but also of the environment in which they are deployed and the information they handle. The importance of conducting web security tests can be seen in the increase in reviews carried out in the 2024-2025 period, especially in follow-up reviews, highlighting the interest on the part of university areas in improving this aspect of the technological infrastructure they manage for the benefit of the university community.

Keywords: *Computer security in web applications, deployment of institutional web applications, pentest of web applications, vulnerabilities in web applications.*

1. INTRODUCCIÓN

El propósito de la prestación del servicio de pruebas de seguridad web para la UNAM es "fortalecer la seguridad de las redes sociales y las páginas oficiales de la Universidad" (Universidad Nacional Autónoma de México, Dirección General de Cómputo y de Tecnologías de Información y Comunicación, 2022). Por ello se debe entender que la revisión en materia de seguridad informática para los sitios web de la Universidad es una medida preventiva necesaria, ya que busca disminuir la exposición de vulnerabilidades una vez que los sitios sean publicados, las cuales, si no son atendidas de forma preventiva, pueden derivar en incidentes de seguridad.

Adicionalmente, en el entorno de la seguridad en aplicaciones web, es importante tener presente que "el proceso de aplicación de pruebas de penetración puede ser una forma de evaluar el nivel de seguridad de un sistema. Cuanto más potente sea la prueba de penetración, más completa será la evaluación de las debilidades/fortalezas del sistema" (Bertoglio y Zorzo, 2017, p.2).

Dado lo anterior, el servicio proporcionado se fundamenta en marcos de referencia como el Proyecto Abierto de Seguridad de Aplicaciones Web (OWASP) y el *Forum of Incident Response and Security Teams* (FIRST), que son empleados en la industria para la realización de las pruebas. Esto permite sistematizar su ejecución y entregar resultados ordenados por nivel de criticidad para su atención.

Finalmente, es relevante recordar que, aunque las pruebas podrían aplicarse a sitios que ya se encuentran en un entorno productivo, en este caso se abordan sólo aquellos que se encuentran próximos a su publicación.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.157 Vol. 4, Núm. 1. enero-marzo 2026, págs. 110 - 117

El objetivo de este reporte técnico es describir y comparar el servicio de realización de las pruebas de penetración en aplicaciones web durante el periodo 2024-2025, con el fin de dar a conocer un servicio institucional, así como evidenciar su contribución al interior de la Universidad en el reforzamiento de aspectos de seguridad informática en sitios web antes de su publicación.

2. DESARROLLO TÉCNICO

La Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) proporciona el servicio de Pruebas de penetración y análisis de vulnerabilidades en aplicaciones web, el cual ayuda al adecuado despliegue de aplicaciones web mediante la revisión de seguridad informática, para lo cual se sigue la metodología de *pentest* establecida en la industria.

De acuerdo con Tudosi, Graur, Balan y Potorac (2023): "La principal diferencia entre un ataque informático y una prueba de penetración es que los hackers irrumpen para robar y causar daños, mientras que los pentesters alertan sobre vulnerabilidades de seguridad explotables y ayudan a corregirlas". Así pues, las pruebas de penetración que se llevan a cabo por parte de los revisores se encuentran limitadas al objetivo previamente establecido, se realizan en entornos controlados, en ventanas de tiempo que han sido acordadas y, sobre todo, cuentan con la autorización de los responsables del sitio.

Cuando una entidad o dependencia busca publicar un sitio web, requiere cubrir una serie de requisitos a nivel técnico. Al solicitar el servicio, lo primero que debe realizarse es un respaldo del sitio web y su base de datos, así como evitar modificar el sitio durante el periodo de revisión. Posteriormente, se debe agendar la fecha para llevar a cabo las pruebas de seguridad necesarias. Actualmente, la solicitud puede realizarse mediante un ticket en la plataforma de Gestión de Servicios TIC o mediante la Oficina Central de Correspondencia. En ambos casos se deben proporcionar los datos solicitados para dar inicio al proceso de revisión.

Lo anterior, junto con la autorización para la realización de las pruebas, permite su desarrollo de manera coordinada entre el solicitante y el equipo de revisores, lo que habilita una comunicación constante para lograr el seguimiento adecuado de posibles complicaciones como, por ejemplo, la pérdida de conectividad hacia el sitio.

De esta forma, la revisión de seguridad se lleva a cabo considerando que: "Mediante la ejecución de pruebas que simulan ataques activos, imitando el comportamiento real de un actor malicioso, se pueden evaluar las vulnerabilidades en diversos activos de información, ya sea de forma manual o automática. Este proceso, en última instancia, facilita la identificación de las estrategias de defensa y respuesta más eficaces" (Moreno et al., 2025, p. 2).

Esto es de gran ayuda para identificar cuáles son los puntos débiles que pueden ser aprovechados por usuarios maliciosos, así como para determinar las medidas de mitigación y recomendaciones de seguridad que serán entregadas a los responsables de las aplicaciones para solventar los hallazgos reportados.

A partir de lo anterior, se genera un reporte con los hallazgos derivados de la revisión, los cuales se ponderan de acuerdo con el impacto que pueden tener y cómo éste afecta la confidencialidad, disponibilidad e integridad del sitio web, así como la información que contiene, los usuarios y la infraestructura tecnológica relacionada.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.157 Vol. 4, Núm. 1. enero-marzo 2026, págs. 111 - 117

El documento contiene una serie de recomendaciones que se apoyan en los estándares y buenas prácticas realizadas por autoridades en materia de seguridad informática como, por ejemplo, el *National Institute of Standards and Technology* (NIST), MITRE, el top 10 de vulnerabilidades web publicado por OWASP, así como referencias a la documentación oficial en el rubro de seguridad de CMS, lenguajes de programación como PHP o Java, bibliotecas de terceros, *frameworks*, repositorios, entre otros.

Una vez entregado el reporte a los responsables de la aplicación web para su atención, se espera la implementación de medidas que ayuden a solventar los puntos señalados en el documento. Cuando el equipo responsable de solventar las vulnerabilidades indica que han sido atendidas, es necesario llevar a cabo una revisión de seguimiento para verificar la adecuada implementación de las configuraciones de seguridad. No se define un periodo de tiempo específico para la atención de los hallazgos, debido a que cada entidad o dependencia universitaria gestiona sus actividades de acuerdo con su carga de trabajo, infraestructura y prioridades institucionales, por lo que son los responsables de atender el reporte tomando en consideración lo anterior.

Durante la revisión de seguimiento, no se realiza nuevamente un escaneo en busca de vulnerabilidades, sino que el trabajo se centra en verificar la solución de los hallazgos reportados. En caso de existir hallazgos persistentes debido a que no se tomaron medidas o a que su aplicación no fue adecuada, se entrega un segundo reporte de seguimiento, el cual contiene tanto los hallazgos que no fueron atendidos como los nuevos que se encuentren durante la verificación y que pueden afectar la seguridad del sitio web.

Adicionalmente, se comunica a los responsables de la aplicación que, en caso de llevar a cabo una migración de sistema operativo o una versión mayor en cuanto a los componentes empleados (por ejemplo, el *framework* de desarrollo o bibliotecas de terceros), es necesario llevar a cabo una nueva prueba de seguridad debido a los cambios realizados.

Finalmente, en caso de que los hallazgos del primer o segundo reporte sean solventados favorablemente, el sitio revisado puede continuar con su proceso de liberación en el entorno productivo.

2.1 METODOLOGÍA

La metodología empleada obedece a una propuesta de adaptación propia de acuerdo con las necesidades de seguridad de la comunidad de TI de la Universidad, la cual es una versión modificada de la metodología empleada en la industria para las pruebas de penetración (*pentest*). Se tomó como referencia la información proporcionada por OWASP en su apartado referente al *Web Application Security Testing* (OWASP, s. f.), así como las 7 fases propuestas por el *Penetration Testing Execution Standard* (PTES) (Penetration Testing Execution Standard, s. f.).

Se realizan pruebas de caja negra que "consisten en imitar las acciones de un atacante que no tiene información sobre la empresa ni su red corporativa. Así es precisamente como la gran mayoría de los hackers se ven obligados a actuar: utilizando todas las herramientas disponibles para identificar y explotar vulnerabilidades en el sistema de seguridad. En la mayoría de los casos, encuentran fallos de seguridad" (Khamdamov y Ibrokhimov, 2021, p.1). Por lo anterior, se desarrollan las siguientes etapas:

- Planificación y determinación del alcance: Se notifica que se lleva a cabo una revisión de caja negra. Adicionalmente, se establece de común acuerdo qué otros elementos descubiertos durante la revisión quedarán fuera del alcance.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.157 Vol. 4, Núm. 1. enero-marzo 2026, págs. 112 - 117

- Reconocimiento: Acercamiento al sitio web, información pública (si existe), posibles usuarios, entre otros.
- Análisis y escaneo: Revisión con distintas herramientas (manuales y automatizadas).
- Explotación: En ocasiones, se trabaja con pruebas de concepto para evitar afectaciones mayores o a otros elementos fuera del alcance.
- Generación del reporte: Se documentan las vulnerabilidades y se emite un reporte técnico.

Las etapas anteriores evitan directamente la explotación de la vulnerabilidad; tampoco se mantiene el acceso en caso de lograrse. Esto obedece a que la revisión debe mantenerse dentro del alcance acordado con la entidad o dependencia universitaria, y descartar otros elementos que puedan encontrarse durante la revisión.

En cuanto a las herramientas empleadas para llevar a cabo el proceso de revisión de seguridad, se emplea Acunetix, en conjunto con la suite de Kali, para tener un entorno controlado durante la realización de las pruebas. De la suite de Kali, se emplean aquellas herramientas que puedan reportar con mayor precisión algún hallazgo, por ejemplo, si se revisa un sitio en WordPress, entonces se hace uso de WPScan.

En la Tabla 1, se pueden observar las características que permitieron la selección de Acunetix como una de las herramientas que se emplean en la etapa automatizada y que se complementan con otras de la suite de Kali y la revisión manual:

Tabla 1

Elementos considerados para la selección de Acunetix

	Acunetix	Burp Suite Professional	OWASP ZAP
Tiempo de respuesta	Rápido y automatizado	Depende de la configuración	Lento en aplicaciones grandes
Capacidad de procesar múltiples sitios	Alto	Medio (depende de la capacidad de procesamiento del equipo donde está instalado)	Limitado-Medio (depende de la capacidad de procesamiento del equipo donde está instalado)
Soporte y mantenimiento	Soporte del fabricante (de acuerdo con lo contratado)	Soporte del fabricante y comunidad amplia	Comunidad *open source*
Falsos positivos	Bajo	Bajo-Moderado (requiere configuración adicional)	Moderado

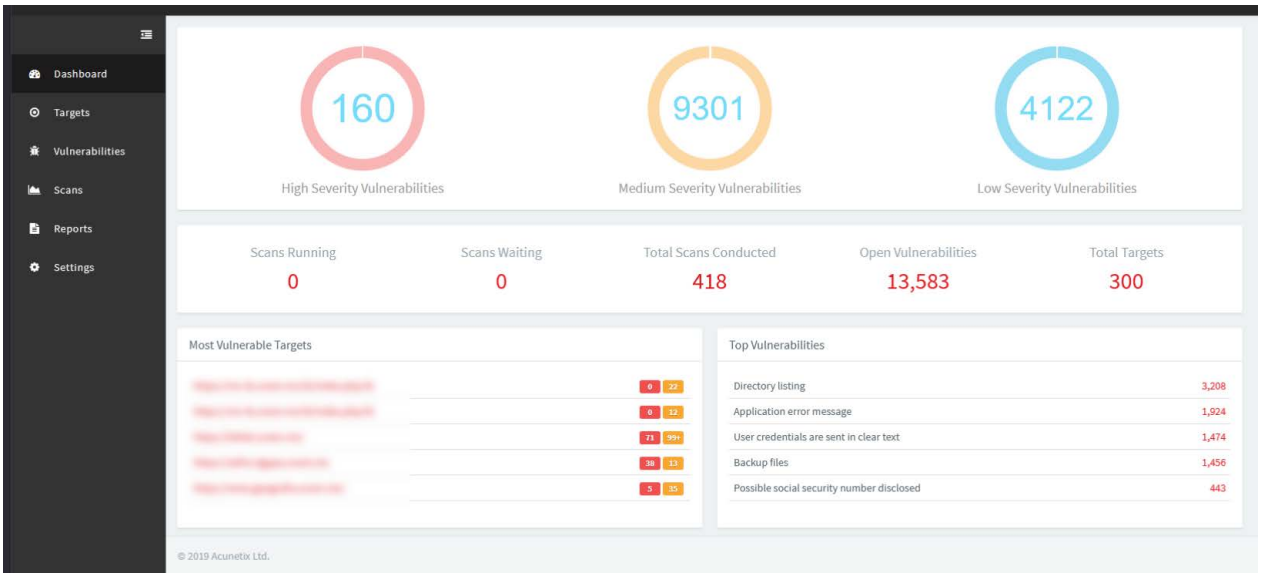
Nota. La tabla se elaboró tomando en consideración la experiencia en el uso de las herramientas durante la realización de las revisiones.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.157 Vol. 4, Núm. 1. enero-marzo 2026, págs. 113 - 117

En la Figura 1 se puede observar el tablero con el resumen de la ejecución de la herramienta automatizada empleada durante la revisión.

Figura 1

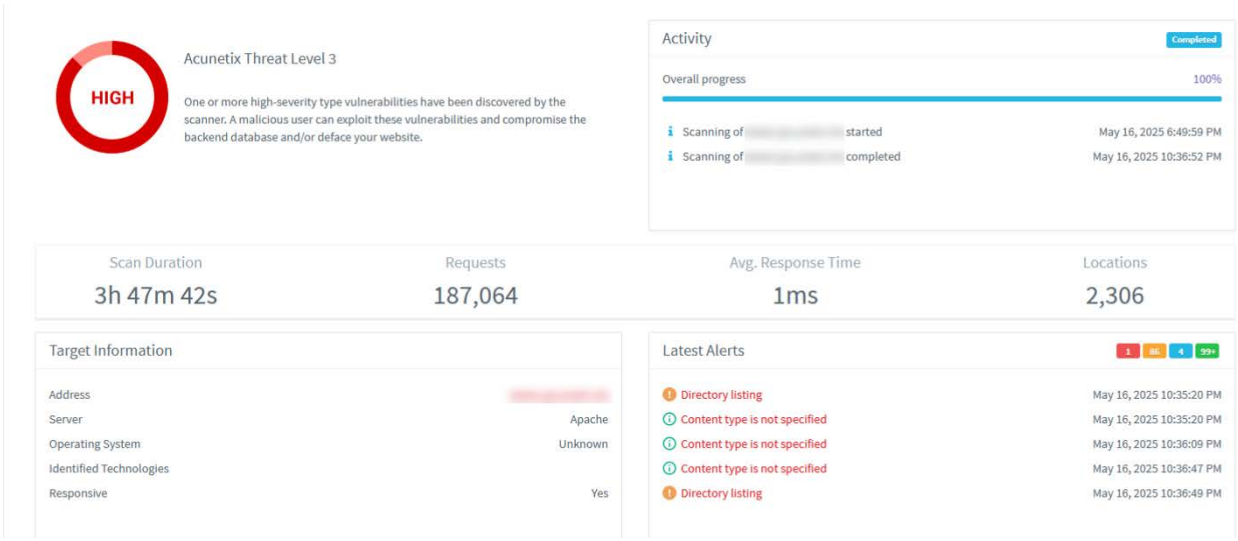
Salida proporcionada por la herramienta Acunetix durante la realización de las pruebas



En la Figura 2 puede observarse el detalle de la ejecución de la herramienta para la búsqueda de vulnerabilidades en un sitio web institucional.

Figura 2

Salida de la ejecución de la herramienta Acunetix para un sitio web



https://doi.org/10.22201/dgtic.30618096e.2026.4.1.157 Vol. 4, Núm. 1. enero-marzo 2026, págs. 114 - 117

Para aprobar una revisión de seguridad se requiere que el reporte emitido (primera revisión o de seguimiento) no cuente con hallazgos ponderados como Críticos, Altos o Medios de acuerdo con las calificaciones proporcionadas por la calculadora *Common Vulnerability Scoring System (CVSS)* de FIRST en su versión 3.1 (FIRST, 2025).

Las recomendaciones de seguridad buscan la implementación de configuraciones que ayuden a proteger no sólo al código fuente, sino también al manejador de bases de datos, al sistema operativo, a los usuarios y a los privilegios asignados, así como al software empleado como servidor de aplicaciones web.

Las pruebas despliegan un conjunto de herramientas automatizadas que se deben configurar y poner en marcha sólo para el objetivo previamente establecido; la salida proporcionada debe ser verificada para descartar falsos positivos, así como para obtener la evidencia correspondiente.

3. RESULTADOS

La realización de pruebas de caja negra permite llevar a cabo una exploración de sitio web sin conocer su contexto, lo que evita tener un sesgo al conocer previamente el funcionamiento, apartados, tipos de usuarios o tecnologías empleadas en su implementación. Los resultados mostrados corresponden sólo a las pruebas realizadas para los sitios web institucionales que solicitaron el servicio. Adicionalmente, es mandatorio para aquellos que se alojan en el Centro de Datos de la DGTIC y, en todos los casos, se llevan a cabo antes de su publicación.

Si bien el servicio se ha proporcionado con anterioridad, este reporte se centra en el periodo 2024-2025 debido al aumento en su demanda. Se presentó un incremento cercano al 11% en las revisiones solicitadas por primera vez y otro de aproximadamente 14% en las revisiones de seguimiento, lo que refleja un mayor interés por parte de las áreas universitarias por solventar temas de seguridad en los sitios web antes de su publicación. De esta manera, a lo largo del periodo que abarca tanto el año 2024 como el año 2025, se obtuvieron los resultados de la Tabla 2.

Tabla 2

Comparativa del número de revisiones realizadas entre 2024 y 2025

	2024	2025 (enero - octubre)
Primera revisión	74	81
Revisión de seguimiento	41	57

Derivado de los datos proporcionados en la Tabla 2, se puede observar que se ha presentado un incremento del número de revisiones realizadas, las cuales reflejan el interés no sólo en cubrir el requisito institucional de llevar a cabo la verificación correspondiente, sino en fortalecer las implementaciones realizadas a los sitios.

Este proceso representa un beneficio real para la Universidad, ya que las correcciones y mejoras pueden ser asimiladas y, posteriormente, aplicadas por los responsables en otras aplicaciones web que pertenecen a las mismas áreas, lo que contribuye a aumentar la seguridad de la infraestructura tecnológica.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.157 Vol. 4, Núm. 1. enero-marzo 2026, págs. 115 - 117

Todas las aplicaciones revisadas se encontraban próximas a su publicación en entornos productivos y las cifras proporcionadas corresponden a la totalidad de solicitudes de revisión recibidas y atendidas.

Por otra parte, a pesar de que los sitios web institucionales son heterogéneos, ya que cada uno soluciona una necesidad en particular, en la Tabla 3 se pueden observar los porcentajes de las tecnologías más empleadas para su implementación:

Tabla 3

Tecnologías más empleadas en los sitios web institucionales

Lenguaje de programación	Servidor de aplicaciones	CMS empleado
PHP 80%	Apache 90%	Wordpress 70%
Java 15%	NginX 5%	OJS 15%
Otras soluciones 5%	Otras soluciones 5%	Drupal 10%
		Otras soluciones 5%

El uso de estas tecnologías se encuentra orientado a proporcionar soluciones a las necesidades de sitios web de la comunidad universitaria, por lo que se puede notar una preferencia por herramientas de distribución libre, que además cuentan con soporte, actualizaciones constantes y módulos o complementos de seguridad por parte de los desarrolladores y la comunidad que les da seguimiento.

Se ha observado que las pruebas de caja negra se pueden fortalecer con una visión integral de seguridad donde se complementen con otras tecnologías y prácticas de seguridad como el análisis estático de código fuente (SAST) y la implementación de defensa en profundidad. Asimismo, es importante el uso de un *firewall* como elemento adicional de seguridad, pues aporta información relevante para el monitoreo y análisis posterior a la revisión realizada, lo que permite dar seguimiento a los hallazgos reportados que no pudieron ser atendidos.

De esta manera, "Una función esencial de un firewall es registrar los detalles de cada intento de conexión, como quién lo realizó y cuándo. Estos datos son valiosos para la resolución de problemas, el análisis de seguridad y otras aplicaciones" (Tudosi et al., 2023, p.11). Esta información, al ser analizada, es de ayuda para fortalecer los resultados de las revisiones y mejorar de manera continua la seguridad web.

Cabe destacar que todas las solicitudes recibidas fueron atendidas. Los únicos retrasos registrados en el inicio de las pruebas corresponden a las aplicaciones que presentaron complicaciones en su despliegue, situación que, una vez solventada, permitió iniciar con el proceso de revisión sin mayores afectaciones.

Adicionalmente, las aplicaciones web que fueron revisadas, en su mayoría, cumplieron con las medidas de seguridad que ayudan no sólo a protegerlas, sino también a evitar afectaciones en los otros sitios que están coubicados en la misma infraestructura.

Finalmente, al considerar la incorporación de las pruebas de *pentest* como parte del ciclo de vida del desarrollo seguro de software (SSDLC), se obtienen elementos para entender el incremento observado en las revisiones solicitadas, ya que su adopción favorece una "mayor eficiencia del proceso de seguridad del software, mejor rendimiento en las métricas de seguridad del software y seguridad general de los productos de software" (Umeugo et al., 2023, p.2).

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.157 Vol. 4, Núm. 1. enero-marzo 2026, págs. 116 - 117

4. CONCLUSIONES

Las revisiones de seguridad realizadas durante el periodo 2024-2025 contribuyeron a mitigar la exposición de vulnerabilidades de los sitios web antes de su liberación y uso por parte de la comunidad universitaria, lo que se refleja en la protección de la información y la continuidad de los servicios institucionales.

El análisis del periodo permitió identificar un incremento en la realización de pruebas de seguimiento respecto a años anteriores, lo que refleja una mayor comprensión y compromiso por parte de los responsables de los sitios respecto a los hallazgos reportados y las medidas a implementar orientadas al fortalecimiento de la seguridad de sus aplicaciones web.

No obstante, los resultados obtenidos pueden optimizarse, ya que el escenario deseable sería que cada revisión inicial cuente con una revisión de seguimiento en donde se corrobore la mitigación total de los hallazgos reportados en el primer reporte, ayudando así a un cierre efectivo de las pruebas de seguridad web.

Resulta fundamental considerar a futuro la necesidad de impulsar el servicio para otras entidades y dependencias de la Universidad, con el fin de fortalecer la postura institucional que promueve la seguridad en aplicaciones web, así como promover una cultura de la seguridad que apoye acciones preventivas que beneficien tanto a los sitios web como a los usuarios.

AGRADECIMIENTOS

Por su apoyo, al Ing. Sergio Anduin Tovar Balderas, al Dr. Manuel I. Quintero Martínez y al Mtro. Juan López Morales.

REFERENCIAS

Bertoglio, D. D., & Zorzo, A. F. (2017). Overview and open issues on penetration test. *Journal of the Brazilian Computer Society, 23*(1), 2. https://doi.org/10.1186/s13173-017-0051-1

Forum of Incident Response and Security Teams. (2025). *Common Vulnerability Scoring System Version 3.1 Calculator*. Recuperado de https://www.first.org/cvss/calculator/3-1/

Khamdamov, R. K., & Ibrokhimov, A. (2021). Techniques and methods of black box identifying vulnerabilities in web servers. En *2021 International Conference on Information Science and Communications Technologies (ICISCT)* (Vol. 00, pp. 1–4). IEEE. https://doi.org/10.1109/ICISCT52966.2021.9670263

Moreno, A. C., Hernandez-Suarez, A., Sanchez-Perez, G., Toscano-Medina, L. K., Perez-Meana, H., Portillo-Portillo, J., Olivares-Mercado, J., & García Villalba, L. J. (2025). Analysis of Autonomous Penetration Testing Through Reinforcement Learning and Recommender Systems. *Sensors, 25*(1), 211. https://doi.org/10.3390/s25010211

Open Web Application Security Project. (s. f.). *OWASP Web Security Testing Guide*. OWASP. https://owasp.org/www-project-web-security-testing-guide/

Penetration Testing Execution Standard. (s. f.). *Penetration Testing Execution Standard (PTES)*. https://www.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.157 **Vol. 4, Núm. 1.** enero-marzo 2026, págs. 117 - 117

pentest-standard.org/index.php/Main_Page

Tudosi, A.-D., Graur, A., Balan, D. G., & Potorac, A. D. (2023). Research on Security Weakness Using Penetration Testing in a Distributed Firewall. *Sensors*, *23*(5), 2683. https://doi.org/10.3390/s23052683

Umeugo, W., Lowrey, K., & Pandya, S. Y. (2023). Factors affecting the adoption of secure software practices in small and medium enterprises that build software in-house. *International Journal of Advanced Research in Computer Science, 14*(2), 1–8. https://doi.org/10.26483/ijarcs.v14i2.6955

Universidad Nacional Autónoma de México, Dirección General de Cómputo y de Tecnologías de Información y Comunicación. (2022). *Lineamientos de seguridad de la información en sitios web de la UNAM*. Red de responsables TIC. https://www.red-tic.unam.mx/recursos/2022/2022_Lineamientos_DGTIC_02.pdf

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 118 - 130

Integración de pruebas DAST en el desarrollo seguro de aplicaciones web

Integration of DAST testing into secure web application development

Información del reporte:

Para citar este reporte técnico:

Chavarría Fernández, R. (2026). Integración de pruebas DAST en el desarrollo seguro de aplicaciones web. *Cuadernos Técnicos Universitarios de la DGTIC, 4* (1), páginas (118 - 130). https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156

Ricardo Chavarría Fernández
Dirección General de Cómputo y de Tecnologías de Información y Comunicación
Universidad Nacional Autónoma de México
rick@unam.mx
ORCID: 0009-0000-3817-6827

Resumen

Se llevó a cabo una serie de evaluaciones de seguridad sobre aplicaciones web institucionales en tiempo de ejecución, mediante la simulación de ataques reales utilizando el método de Pruebas Dinámicas de Seguridad de Aplicaciones (*Dynamic Application Security Testing*, DAST). El objetivo fue identificar vulnerabilidades durante la fase de pruebas del Ciclo de Vida del Desarrollo de Software (SDLC), antes de la implementación y de la realización de auditorías de seguridad. Para ello, se diseñó un entorno de evaluación controlado y se seleccionó y configuró la herramienta especializada OWASP Zed Attack Proxy (ZAP), con la que se realizaron escaneos automatizados y manuales para detectar vulnerabilidades comunes, tales como inyecciones de código malicioso (XSS) y configuraciones inseguras. Durante las pruebas, se identificaron hallazgos importantes, principalmente relacionados con errores en la validación de entradas y configuraciones por defecto. Los resultados se compararon con estándares actuales de seguridad web, lo que hizo posible establecer medidas de mitigación y prevención adecuadas. En conjunto, la integración de DAST durante la fase de pruebas del SDLC permitió fortalecer el aseguramiento continuo de la seguridad de los sistemas evaluados, al identificar vulnerabilidades en tiempo de ejecución antes de su despliegue en entornos productivos.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 119 - 130

Palabras clave: Análisis dinámico de seguridad, gestión de vulnerabilidades, OWASP-ZAP, seguridad web.

Abstract

A series of runtime security assessments were conducted on institutional web applications by simulating real-world attacks using Dynamic Application Security Testing (DAST). The objective was to identify vulnerabilities during the testing phase of the Software Development Life Cycle (SDLC), prior to implementation and security audits. To this end, a controlled testing environment was designed, and the specialized OWASP Zed Attack Proxy (ZAP) tool was selected and configured. Automated and manual scans were performed to detect common vulnerabilities, such as cross-site scripting (XSS) and insecure configurations. During the tests, significant findings were identified, primarily related to errors in input validation and default configurations. The results were compared with current web security standards, enabling the establishment of appropriate mitigation and prevention measures. Overall, the integration of DAST during the SDLC testing phase strengthened the ongoing security assurance of the evaluated systems by identifying runtime vulnerabilities before their deployment in production environments.

Keywords: *Dynamic security analysis, OWASP-ZAP, vulnerability management, web security.*

1. INTRODUCCIÓN

La proliferación y adopción de tecnologías en el desarrollo de software ha expandido significativamente los vectores de ataque en los entornos digitales, particularmente en aplicaciones web. Ante esta perspectiva, las organizaciones se han visto obligadas a adoptar estrategias de seguridad más sólidas para salvaguardar su información y asegurar los principios de confidencialidad, integridad y disponibilidad de la misma (International Organization for Standardization [ISO], 2022; National Institute of Standards and Technology [NIST], 2020).

En la práctica, las revisiones o auditorías de seguridad suelen realizarse al concluir el ciclo de desarrollo de software, lo que conduce al descubrimiento de hallazgos tardíos que pueden exponer a las organizaciones a riesgos innecesarios como la pérdida o robo de información, así como incrementar los costos de mantenimiento. Por esta razón, resulta prioritario adoptar enfoques —como DevSecOps o el Ciclo de Vida del Desarrollo de Software (SDLC) (NIST, 2022)— que integren la seguridad desde fases intermedias y continuas del ciclo de desarrollo, incentivando la evaluación periódica de riesgos y la aplicación proactiva de controles de seguridad desde la fase de diseño, desarrollo y pruebas, lo que contribuye a reducir la superficie de ataque y mejorar la resiliencia de las aplicaciones.

En el ámbito del aseguramiento continuo de los sistemas web, las Pruebas Dinámicas de Seguridad en Aplicaciones (DAST) (OWASP Foundation, 2023a) representan un proceso proactivo para identificar vulnerabilidades mientras las aplicaciones están en ejecución, interactuando directamente con las interfaces del sistema sin necesitar acceso al código fuente; además, permiten emular ataques reales desde la perspectiva de un usuario externo. Por esta razón, las DAST se han posicionado como un componente metodológico esencial dentro de las estrategias de seguridad de aplicaciones en línea (AppSec), particularmente en fases donde las aplicaciones ya cuentan con funcionalidad operativa,

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 120 - 130

complementando otras técnicas empleadas en el ámbito de la seguridad (Aydos & Baykara, 2022; Jennings, 2025; Dencheva, 2022).

En el entorno institucional, específicamente en la Dirección General de Cómputo y de Tecnologías de Información y Comunicación de la UNAM, ha sido necesario incorporar paulatinamente mecanismos de defensa que fortalezcan la seguridad de los sistemas de servicios web en la fase de pruebas. Por ello, entre 2023 y 2025, se realizaron evaluaciones dinámicas de seguridad en dos aplicaciones institucionales: el Sistema Institucional de Consultas Universitarias (SICU) y el Sistema de Votaciones Electrónicas (SVE), con el propósito de identificar vulnerabilidades y establecer acciones de mejora continua, sentando las bases para evaluaciones de seguridad más exhaustivas en etapas posteriores.

El objetivo principal consistió en implementar un proceso de Pruebas Dinámicas de Seguridad de Aplicaciones como parte del desarrollo seguro de sistemas web institucionales, mediante la ejecución de evaluaciones en entornos de pruebas controlados, con el fin de analizar el comportamiento de las aplicaciones en tiempo de ejecución, validar la eficacia de los controles de seguridad existentes y reducir riesgos antes de su despliegue en entornos productivos, sirviendo además como preparación para auditorías de seguridad internas y externas posteriores.

Adicionalmente, se buscó alinear dichas pruebas con estándares internacionales de seguridad, como OWASP Top 10 e ISO/IEC 27034 (OWASP Foundation, 2021; ISO, 2018); de esta manera, no sólo se fortaleció la seguridad de las aplicaciones, sino que también se impulsó la madurez del enfoque institucional hacia el desarrollo seguro de aplicaciones web en el equipo de trabajo.

2. DESARROLLO TÉCNICO

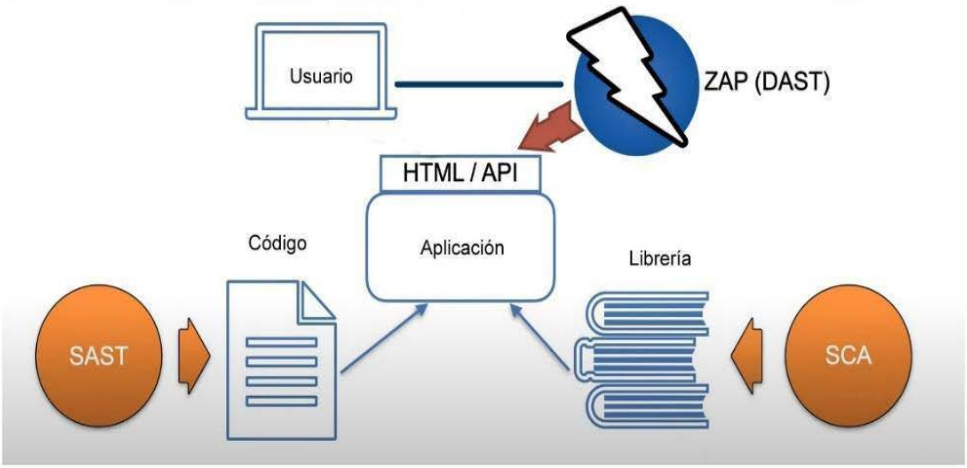
Para la planificación y ejecución de las pruebas, se empleó OWASP Zed Attack Proxy (OWASP ZAP), una herramienta gratuita de análisis dinámico reconocida por su capacidad para realizar escaneos activos, interceptar tráfico y detectar vulnerabilidades en tiempo de ejecución. Los análisis fueron configurados cuidadosamente para no afectar la disponibilidad del entorno: se inició con escaneos pasivos y, posteriormente, se ejecutaron pruebas activas autorizadas y pruebas manuales de comprobación (Chorell & Ekberg, 2024; Kondraciuk et al., 2022).

OWASP ZAP actúa como un proxy entre el navegador y la aplicación web, lo que permite interceptar e inspeccionar las solicitudes y respuestas para modificar su contenido y así reenviar los paquetes interceptados para descubrir vulnerabilidades y comportamientos inesperados; (véase la Figura 1).

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 121 - 130

Figura 1

Intercepción e inspección de solicitudes y respuestas



Las evaluaciones automatizadas se complementaron con pruebas de penetración manuales (*pentesting*), las cuales simulan el comportamiento de un atacante externo con el fin de identificar fallos en los sistemas o en las comunicaciones, así como evaluar riesgos de filtración de datos, denegación de servicio o falsos positivos. Aunque las pruebas manuales pueden ser más lentas que los escaneos automáticos, proporcionan hallazgos de mayor precisión y ayudan a validar los mecanismos de defensa, los procedimientos de respuesta y el cumplimiento de políticas.

Después de identificar las vulnerabilidades, se priorizaron y aplicaron las correcciones necesarias, según su criticidad, mediante ajustes de configuración, mejoras en el código y refuerzo de controles. Posteriormente, se verificó la efectividad de estas correcciones con escaneos de verificación, empleando la misma herramienta de detección inicial. Para concluir, se documentaron los hallazgos y se proporcionaron recomendaciones adicionales para prevenir recurrencias y fortalecer la seguridad del sistema en desarrollo.

2.1 METODOLOGÍA

La metodología de pruebas dinámicas de seguridad se desarrolló siguiendo las guías y marcos establecidos por la OWASP Foundation, que representan estándares abiertos reconocidos en la industria para la seguridad de aplicaciones web. La estructura metodológica se organizó en seis etapas:

I. Planeación

En esta etapa, se definieron los objetivos del proceso de evaluación, el alcance del análisis dinámico y las restricciones operativas. Así mismo, se seleccionaron las pruebas a realizarse con la herramienta OWASP Zed Attack Proxy (ZAP).

Como marco de referencia para la identificación de vulnerabilidades, se adoptó el OWASP Top 10 2021, documento que establece las diez categorías de riesgos más críticos en aplicaciones web y que sirve como guía para priorizar esfuerzos de mitigación (OWASP Foundation, 2021); (véase la Figura 2).

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 122 - 130

Figura 2

OWASP TOP 10 2021

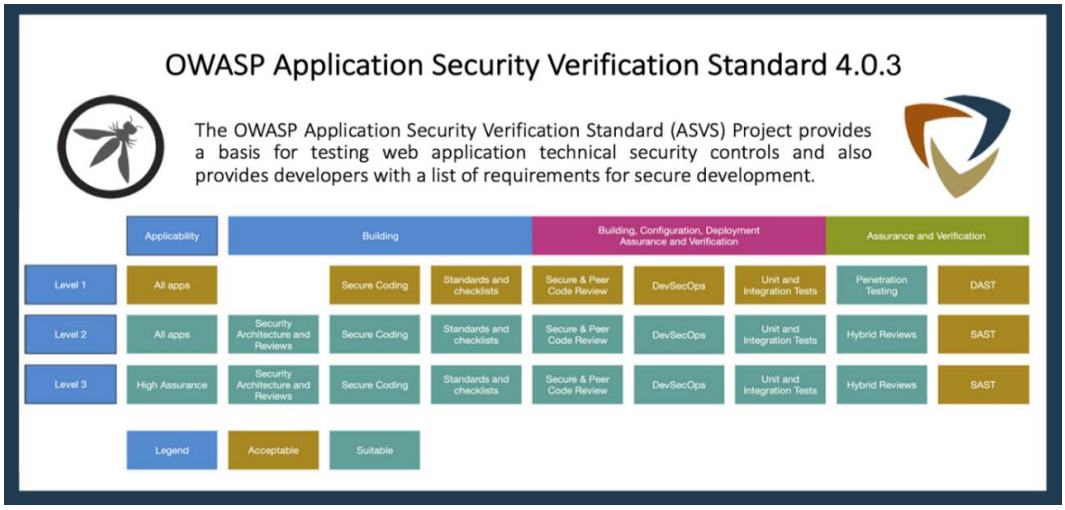
II. Preparación

La instalación de OWASP ZAP se realizó en un entorno aislado y controlado, utilizando máquinas virtuales para la aplicación y la base de datos a probar, lo que garantizó la reproducibilidad del análisis. Adicionalmente, se habilitaron extensiones de ZAP disponibles en su sitio oficial, lo que permitió ampliar la capacidad de exploración y generar reportes más específicos (OWASP Foundation, 2023a).

La selección de herramientas y la forma de despliegue se alinearon con el OWASP Application Security Verification Standard (ASVS) v4.0.3, que establece controles de seguridad requeridos para aplicaciones modernas y sugiere pruebas dinámicas en entornos dedicados (OWASP Foundation, 2023); véase la Figura 3.

Figura 3

OWASP Application Security Verification Standard v4.0.3



Nota. Tomado de https://blog.secureflag.com/2024/11/15/understanding-owasp-asvs/

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 123 - 130

III. Descubrimiento

El entorno de pruebas se diseñó para replicar condiciones reales de operación sin comprometer la disponibilidad del sistema. Se configuró OWASP ZAP con parámetros ajustados a las necesidades del aplicativo, incluyendo:

- Perfiles de escaneo pasivo y activo
- Credenciales de prueba con permisos controlados
- Exclusión de ciertos recursos sensibles para evitar interrupciones
- Límites de solicitudes y profundidad de análisis

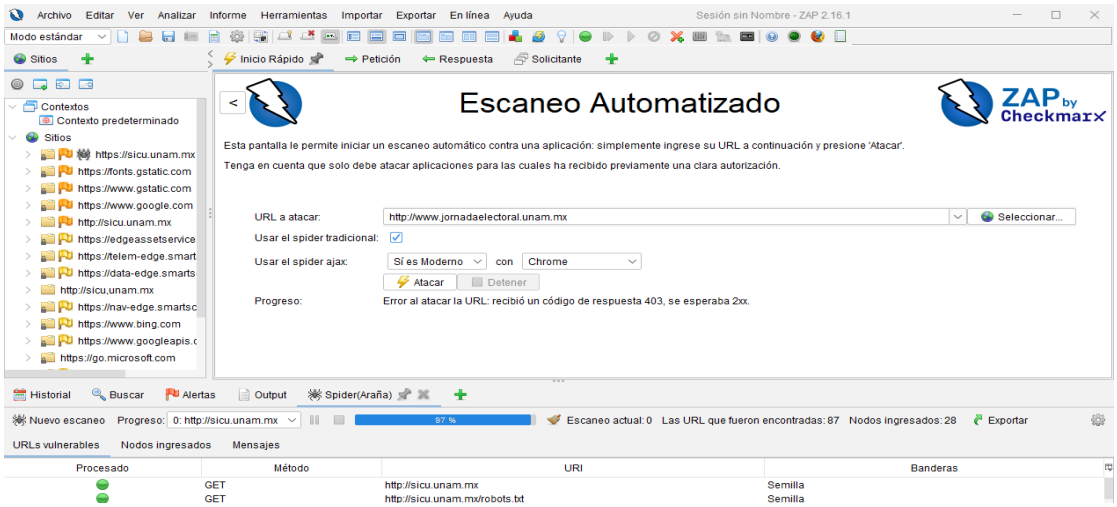
IV. Ejecución de pruebas

Las pruebas se ejecutaron en dos modalidades:

1. Escaneo activo, en el que se simularon ataques controlados para confirmar vulnerabilidades detectadas durante la etapa pasiva (véase la Figura 4).

Figura 4

Escaneo automatizado ZAP

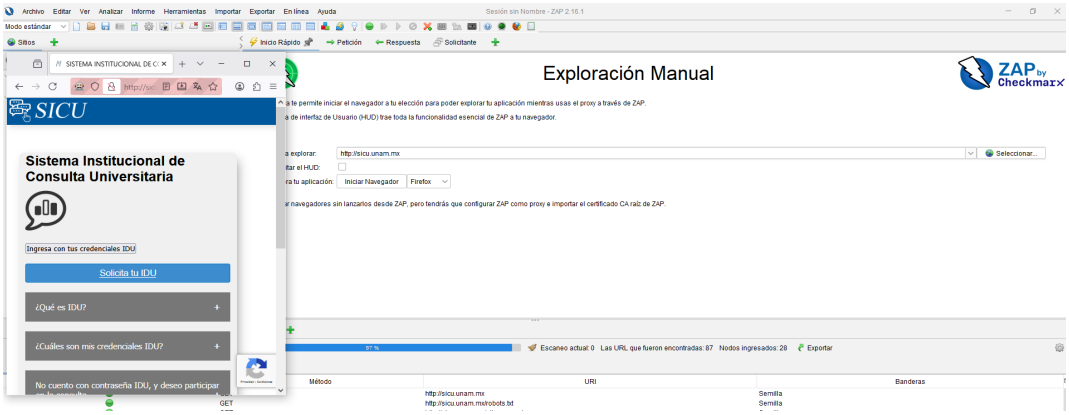


2. Escaneo manual, en el cual se registraron las solicitudes y respuestas sin modificar el tráfico, con el fin de encontrar vulnerabilidades que los escaneos automatizados pasan por alto, especialmente aquellas relacionadas con la lógica de negocio y el contexto específico de la aplicación (véase la Figura 5).

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 124 - 130

Figura 5

Escaneo manual OWASP ZAP



V. Validación, análisis de impacto y priorización contextual de hallazgos

Para la priorización de los hallazgos se aplicó la OWASP Risk Rating Methodology (OWASP Foundation, 2023b), complementando la clasificación automática generada por OWASP ZAP mediante un proceso estructurado de dos etapas.

Etapa 1: Clasificación inicial automatizada

OWASP ZAP proporcionó una valoración técnica preliminar basada en:

- Patrones predefinidos de vulnerabilidad.
- Severidad estándar de la industria.
- Heurísticas automatizadas de detección.
- Resultado: Clasificación genérica, (véase Figura 6).

Figura 6

Reporte de alertas

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 125 - 130

Etapa 2: Evaluación contextual manual (Matriz OWASP)

Cada hallazgo fue reevaluado considerando el contexto institucional del sistema, aplicando los factores de valoración siguientes (véase Figura 7), así como la matriz de severidad de riesgo OWASP Risk; (véase Figura 8).

Figura 7

Reporte de alertas OWASP ZAP

Figura 8

Matriz de severidad, OWASP Risk Rating Methodology

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

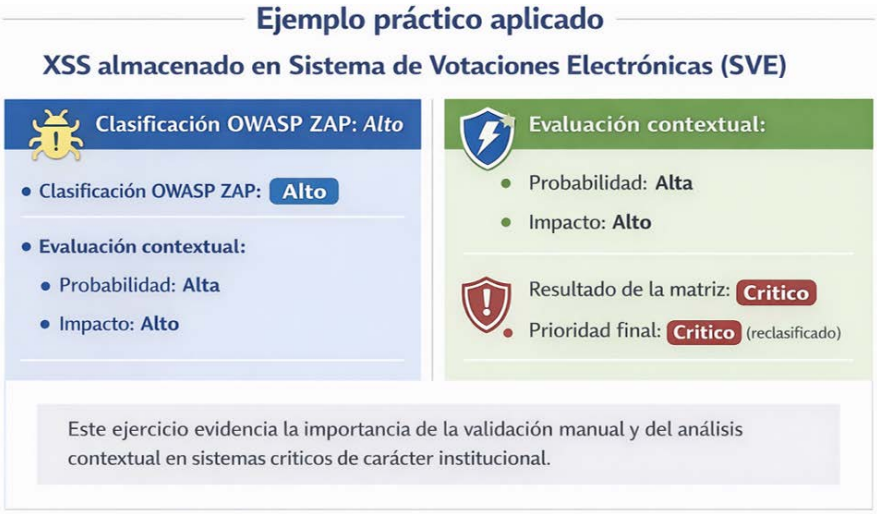
Nota. Riesgo = Probabilidad x Impacto = Likelihood x Impact. Recuperado de https://secumantra.com/owasp-top-ten-risk-rating/

A continuación, se muestra un ejemplo práctico de validación, análisis de impacto y priorización, (véase Figura 9).

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 126 - 130

Figura 9

Evaluación de hallazgo cross-site scripting (XSS)



VI. Mitigación y revalidación

Los hallazgos confirmados fueron reportados mediante evidencias, nivel de riesgo y recomendaciones de mitigación. Posteriormente, se contempló la ejecución de re-escaneos para verificar la efectividad de las acciones correctivas aplicadas.

3. RESULTADOS

La integración de Pruebas Dinámicas de Seguridad de Aplicaciones, durante la fase de pruebas del SDLC, permitió identificar, corregir y mitigar vulnerabilidades en aproximadamente el 80% de los hallazgos identificados —principalmente aquellos clasificados con severidad media, alta y crítica— en la primera iteración de corrección. Esta práctica redujo significativamente los costos de corrección en comparación con la detección tardía durante auditorías posteriores al despliegue, ya que se realizaron en la fase de pruebas del SDLC con sistemas funcionales, pero aún en proceso de ajuste y mejora.

Los hallazgos obtenidos, además de proporcionar información relevante, retroalimentaron de manera inmediata las actividades de diseño y codificación en iteraciones subsecuentes, contribuyendo a la prevención de repeticiones y a la mejora progresiva de los controles de seguridad implementados. Asimismo, la adopción de estas pruebas dinámicas, alineadas con los estándares OWASP, fortaleció un ciclo robusto de mejora continua en la seguridad de aplicaciones web.

Como parte de una estrategia de mejora continua, los resultados de las pruebas dinámicas con OWASP ZAP en los sistemas de Consultas Universitarias y de Voto Electrónico revelaron hallazgos relacionados con aspectos de seguridad que no fueron considerados durante el diseño y desarrollo inicial, tales como configuraciones inseguras por defecto, cabeceras de seguridad faltantes (Política de Seguridad de Contenidos y Seguridad Estricta de Transporte HTTP), y controles insuficientes en mecanismos de autenticación, mismos que fueron subsanados en cada iteración (véase la Figura 10 y 11).

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 127 - 130

Figura 10

Figura extraída de la auditoría externa realizada al Sistema de Votaciones Electrónicas 2024

NIVEL DE SEGURIDAD EN EL SISTEMA DE VOTACIONES.

ENTENDIMIENTO DE RESULTADOS

LOS RESULTADOS DETERMINAN UN NIVEL **ALTO** DE SEGURIDAD EN EL SISTEMA DE VOTACIONES ELECTRÓNICAS EN SU VERSIÓN 5, CON BASE EN EL HECHO DE QUE FUERON **DETECTADAS 0 (CERO)** "VULNERABILIDADES" DE ACUERDO CON EL MARCO OWASP.

LO ANTERIOR NO EXIME DE MANTENER CONTROLES DE SEGURIDAD Y EVALUAR CONTINUAMENTE EL SISTEMA, DADO QUE, PARA UN INTERNO CON CONOCIMIENTO, RESULTARÍA MÁS SENCILLO CAUSAR ALGÚN DAÑO A LOS ACTIVOS DE INFORMACIÓN Y A LA INFORMACIÓN MISMA DE LA UNIVERSIDAD Y SUS VOTANTES

Figura 11

Figura extraída de la auditoría interna realizada al Sistema Institucional de Consultas Universitarias 2024

DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN
COORDINACIÓN DE SEGURIDAD DE LA INFORMACIÓN

INFORME DE PRUEBAS DE SEGURIDAD A SITIO WEB SICU.UNAM.MX

DGTIC

RESUMEN EJECUTIVO

En el presente informe se documentan los resultados de las pruebas y, con base en ellos, se emiten recomendaciones que incluyen acciones correctivas y preventivas orientadas a fortalecer el esquema de seguridad actual.

Como resultado de las pruebas se identificaron 6 hallazgos, 1 medio, 2 bajos y 3 sin impacto, según la calificación establecida a través del *Common Vulnerability Scoring System* Versión 3.1 (Anexo C).

La siguiente tabla muestra la cantidad de hallazgos identificados por activo.

HALLAZGOS IDENTIFICADOS DURANTE LA REVISIÓN						
ID ACTIVO	SITIO	CRITICOS	ALTO	MEDIO	BAJO	SIN IMPACTO
ACT01	sicu.unam.mx	0	0	1	2	3

Como resultado, las auditorías de seguridad internas y externas, realizadas posteriormente a los sistemas, reflejaron una madurez institucional, evidenciada por una superficie de ataque reducida y una gestión eficaz de los hallazgos residuales de acuerdo con los informes presentados por las auditorías respectivas.

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 128 - 130

4. CONCLUSIONES

Las Pruebas Dinámicas de Seguridad de Aplicaciones (DAST), aplicadas a los sistemas institucionales, permitieron identificar vulnerabilidades críticas que no siempre pueden ser detectadas mediante otros tipos de análisis, tales como inyecciones SQL, *cross-site scripting* (XSS) y debilidades en los mecanismos de autenticación, lo cual se reflejó en la corrección de los hallazgos identificados. Al evaluar la aplicación en ejecución y bajo condiciones reales, las DAST proporcionan una medición más precisa de su nivel de seguridad, con una tasa significativamente menor de falsos positivos en comparación con técnicas que no analizan aplicaciones en funcionamiento (Putra et al., 2024; Qadir et al., 2025).

No obstante, desde una perspectiva metodológica, las DAST presentan ciertas limitaciones, ya que, al enfocarse exclusivamente en componentes accesibles desde el exterior, no permiten examinar en profundidad elementos internos como la lógica de negocio o procesos no expuestos. Asimismo, al requerir que el sistema se encuentre operativo y generalmente en entornos de producción, su aplicación en fases tardías del ciclo de desarrollo puede incrementar el esfuerzo y los costos de corrección (Abdulghaffar et al., 2023).

En el presente contexto, estas limitaciones fueron reducidas al integrar pruebas dinámicas antes del despliegue en entornos productivos, permitiendo que los hallazgos se atendieran durante etapas aún controladas en la fase de pruebas, aunque se reconoce que una incorporación de pruebas unitarias por parte de los desarrolladores, complementada con Pruebas Estáticas de Seguridad de Aplicaciones (SAST), podría optimizar aún más los resultados.

Por estas razones, se recomienda utilizar DAST en conjunto con pruebas SAST, ya que esta combinación permite analizar el código fuente desde la fase de desarrollo y ofrece una visión más integral del estado de seguridad de una aplicación (Koman & Janiszewski, 2025).

Para que las pruebas DAST sean realmente eficaces, es indispensable configurar la herramienta de acuerdo con las características específicas del sistema a evaluar, prestando atención a la autenticación, gestión de sesiones y exposición de información sensible del aplicativo o sistema. En este sentido, se aconseja implementar validaciones robustas, mecanismos de autenticación más estrictos y políticas que limiten la divulgación de información operativa del software.

Asimismo, la incorporación de DAST como parte del proceso de desarrollo favorece una cultura de seguridad continua en las organizaciones, al complementar otras técnicas de análisis y contribuir a una mejor preparación de los sistemas ante amenazas de seguridad en sus entornos productivos.

REFERENCIAS

Abdulghaffar, K., Elmrabit, N., & Yousefi, M. (2023). Enhancing web application security through automated penetration testing with multiple vulnerability scanners. *Computers,* 12(11), 235. https://www.mdpi.com/2073-431X/12/11/235

Aydos, M., & Baykara, M. (2022). Security testing of web applications: A systematic mapping. *Journal of Information Security and Applications*, 63, 103005. https://www.sciencedirect.com/science/article/pii/S131915782100269X?via%3Dihub

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 Vol. 4, Núm. 1. enero-marzo 2026, págs. 129 - 130

Chorell, I., & Ekberg, C. (2024). *A comparative analysis of open source dynamic application security testing tools* [Tesis de maestría, Linköping University, Departamento de Ciencias de la Computación e Información]. Linköping University.

https://www.diva-portal.org/smash/get/diva2:1868722/FULLTEXT01.pdf

Dencheva, L. (2022). *Comparative analysis of static application security testing (SAST) and dynamic application security testing (DAST).*

https://norma.ncirl.ie/5956/1/lyubkadencheva.pdf

International Organization for Standardization. (2018). *Information technology — Security techniques — Application security — Part 3: Application security management process* (ISO/IEC Standard No. 27034-3:2018*)*. https://www.iso.org/standard/55583.html

International Organization for Standardization. (2022). *ISO/IEC 27001:2022 Information technology—Security techniques—Information security management systems—Requirements*. ISO. – Requirements. ISO. https://www.iso.org/standard/27001

Jennings, T. (2025, 5 de mayo). *Dynamic Application Security Testing: DAST Basics*. Mend.io. https://www.mend.io/blog/dast-dynamic-application-security-testing/

Koman, J., & Janiszewski, M. (2025). SCAnME – scanner comparative analysis and metrics for evaluation. *International Journal of Information Security*, 24, Article 147. https://doi.org/10.1007/s10207-025-01054-8

Kondraciuk, A., Bartos, A., & Pańczyk, B. (2022). Comparative analysis of the effectiveness of OWASP ZAP, Burp Suite, Nikto and Skipfish in testing the security of web applications. *Journal of Computer Sciences Institute*, 24, 176-180. https://doi.org/10.35784/jcsi.2929

National Institute of Standards and Technology. (2020). *Security and privacy controls for information systems and organizations* (NIST Special Publication 800-53, Revision 5). https://doi.org/10.6028/NIST.SP.800-53r5

National Institute of Standards and Technology. (2022). *Secure software development framework (SSDF) version 1.1: Recommendations for mitigating the risk of software vulnerabilities* (NIST Special Publication 800-218). U.S. Department of Commerce. https://doi.org/10.6028/NIST.SP.800-218

OWASP Foundation. (2021). *OWASP Top Ten:2021*. https://owasp.org/Top10/

OWASP Foundation. (2023). *Application Security Verification Standard (ASVS)* v4.0.3. https://owasp.org/ASVS/

OWASP Foundation. (2023a). *OWASP Zed Attack Proxy (ZAP)*. https://www.zaproxy.org/

OWASP Foundation. (2023b). *OWASP Risk Rating Methodology*. OWASP Foundation. https://owasp.org/www-community/OWASP_Risk_Rating_Methodology

Putra, F. P. E., Ubaidi, U., Hamzah, A., Pramadi, W. A., & Nuraini, A. (2024). Systematic Literature Review: Security Gap Detection On Websites Using OWASP ZAP. *Brilliance: Research of Artificial Intelligence,* 4(1), 348-355. https://doi.org/10.47709/brilliance.v4i1.4227

https://doi.org/10.22201/dgtic.30618096e.2026.4.1.156 **Vol. 4, Núm. 1.** enero-marzo 2026, págs. 130 - 130

Qadir, S., Waheed, E., Khanum, A., & Jehan, S. (2025). Comparative evaluation of approaches & tools for effective security testing of Web applications. *PeerJ Computer Science*, 11, e2821. https://doi.org/10.7717/peerj-cs.2821

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