CUADERNOS TÉCNICOS UNIVERSITARIOS DE LA DGTIC

ISSN-e: 3061-8096

Vol. 3, Núm. 2. abril-junio 2025

DOI:10.22201/dgtic.30618096e.2025.3.2

DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Cuadernos Técnicos Universitarios de la DGTIC

DOI: 10.22201/dgtic.30618096e.2025.3.2

Vol. 3, Núm. 2. abril-junio 2025

Cuadernos Técnicos Universitarios de la DGTIC

DOI: 10.22201/dgtic.30618096e.2025.3.2

Vol. 3, Núm. 2. abril-junio 2025

CUADERNOS TÉCNICOS UNIVERSITARIOS DE LA DGTIC

Editor Responsable Héctor Benítez Pérez • Editora Académica Marcela J. Peñaloza Báez • Asistente Editorial Pamela Valdés Reséndiz • Corrector de estilo Pablo Vázquez Castellanos

Comité Editorial de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación

Héctor Benítez Pérez • Luz María Castañeda de León • Rafael Fernández Corro • Alfredo Hernández Mendoza • Marina Kriscautzky Laxague • Lorena Cárdenas Guzmán • Ana Yuri Ramírez Molina • Eprin Varas Gabrelian • Juan Voutssás Márquez

Para citar un reporte técnico de la obra: Apellidos 1 Apellidos 2, Iniciales nombres. (2025). Título del reporte técnico. *Cuadernos Técnicos Universitarios de la DGTIC, 3* (2), páginas (N1-N2).

DOI: 10.22201/dgtic.30618096e.2025.3.2

Vol. 3, Núm. 2. abril-junio 2025

CUADERNOS TÉCNICOS UNIVERSITARIOS DE LA DGTIC, Año 3, No. 2, abril-junio 2025, 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 mayo de 2025.

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

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

AVISO DE PRIVACIDAD
https://www.tic.unam.mx/avisosprivacidad/

Cuadernos Técnicos Universitarios de la DGTIC

DOI: 10.22201/dgtic.30618096e.2025.3.2

Vol. 3, Núm. 2. abril-junio 2025

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 de Colaboración y Vinculación 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 • Miguel Ángel Islas Flores, Diseño gráfico y editorial • Jonathan Cedillo Castro, Maquetador.

Cuadernos Técnicos Universitarios de la DGTIC

DOI: 10.22201/dgtic.30618096e.2025.3.2

Vol. 3, Núm. 2. abril-junio 2025

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

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

Cuadernos Técnicos Universitarios de la DGTIC

DOI: 10.22201/dgtic.30618096e.2025.3.2

Vol. 3, Núm. 2. abril-junio 2025

CONTENIDO

Retos en la implementación de exámenes de lenguas en línea con Moodle
Sonia Cruz Techica, Jesús Tonatihu Sánchez Neri 8

Clúster de software libre del Laboratorio Nacional de Observación de la Tierra
Alejandro Aguilar Sierra 24

Implementación de servidor para publicación de proyectos de construcción de sitios web
Margarita González Trejo 35

Utilización de Dockers para desplegar en producción sistemas de información heredados
José Othoniel Chamú Arias 45

Desarrollo de aplicaciones a la medida con apoyo de modelos GPT
Luz María Ramírez Romero 54

Revisores julio 2024 a junio 2025 81

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112

Vol. 3, Núm. 2. abril-junio 2025, págs. 8 - 23

Retos en la implementación de exámenes de lenguas en línea con Moodle

Sonia Cruz Techica
Escuela Nacional de Lenguas, Lingüística y Traducción
Universidad Nacional Autónoma de México
sonia@enallt.unam.mx
ORCID: 0000-0002-2891-5259

Jesús Tonatihu Sánchez Neri
Escuela Nacional de Lenguas, Lingüística y Traducción
Universidad Nacional Autónoma de México
tsanchez@enallt.unam.mx
ORCID: 0000-0001-9202-5202

Información del reporte:

Licencia Creative Commons

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

Para citar este reporte técnico:

Cruz Techica, S. Sánchez Neri, J. T. (2025). Retos en la implementación de exámenes de lenguas en línea con Moodle. *Cuadernos Técnicos Universitarios de la DGTIC*, 3 (2) páginas(8 - 23).

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112

Resumen

La aplicación de exámenes en línea ofrece varios beneficios en comparación con las evaluaciones tradicionales, entre los que destacan la reducción de costos, la flexibilidad en cuanto a ubicación y horarios, así como la optimización de todo el proceso, al permitir la inclusión de preguntas más dinámicas e interactivas y hacer más eficiente la entrega de resultados. No obstante, en algunos casos no basta con el uso de los recursos y actividades provistos por el sistema de gestión de contenidos elegido como soporte para estos exámenes, sino que es preciso generar soluciones a la medida para satisfacer necesidades particulares. Éste fue el caso de los exámenes de admisión para distintos proyectos en la Escuela Nacional de Lenguas, Lingüística y Traducción, en donde las secciones de comprensión lectora y auditiva requirieron ajustes ya fuera en la configuración o en la programación de Moodle. Para el diseño de las soluciones tecnológicas, se partió de una metodología basada en ingeniería de software para delimitar los requerimientos y proponer soluciones acordes con los recursos disponibles en la institución, así como el análisis de las ventajas y

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112

Vol. 3, Núm. 2. abril-junio 2025, págs. 9 - 23

desventajas de emplear una solución comercial. Específicamente, la programación requerida para el control del número de reproducciones de audios, en las secciones de comprensión auditiva, se enmarcó en una metodología ágil que ha permitido un ciclo de mejora continua y ajustes con base en las áreas de oportunidad detectadas. Los resultados reportados muestran la capacidad de adaptación de Moodle para tareas específicas, a la vez que dan cuenta de las lecciones aprendidas y trazan nuevas rutas por explorar en un trabajo futuro. Al mismo tiempo, sugieren recomendaciones y reflexiones que pueden ser de utilidad para resolver problemáticas similares en otros contextos.

Palabras clave:

Exámenes en línea, enseñanza de lenguas, Moodle, reproducción de audio.

Abstract

The application of online exams provides several benefits compared to traditional examinations, especially cost reduction, flexibility in terms of locations and schedules, and the optimization of the entire process when the inclusion of more dynamic and interactive questions is facilitated and the delivery of results becomes more efficient. However, in some cases it is not enough to use the resources and activities provided by the content management system that was chosen to support these exams; it is necessary to generate customized solutions to meet particular needs. This was the case of the admission exams for different projects at the Escuela Nacional de Lenguas, Lingüística y Traducción, where the reading and listening comprehension sections required adjustments either in the configuration or in the Moodle programming. The design of the technological solutions used a software engineering-based methodology to define the requirements and propose solutions within the available institutional resources, as well as the analysis of the advantages and disadvantages of using a commercial solution. Specifically, the programming that was required to control the number of audio playbacks in the listening comprehension sections was developed within an agile methodology that promoted a continuous improvement cycle with adjustments based on the areas of opportunity that were detected. The reported results show how adaptable Moodle is for specific tasks, while they provide lessons learned and outline new paths to be explored in future work. At the same time, the results suggest recommendations and reflections that may be useful for solving similar problems in other contexts.

Key words:

Online testing, language teaching, Moodle, audio playback.

1. INTRODUCCIÓN

La contingencia sanitaria ocasionada por el COVID-19 llevó a la Escuela Nacional de Lenguas, Lingüística y Traducción (ENALLT) a buscar alternativas para mantener sus actividades académicas y garantizar la continuidad de los procesos de evaluación. En este contexto, surgió la necesidad de emplear la plataforma Moodle como entorno virtual para la aplicación de exámenes en línea, particularmente en los procesos de admisión a licenciaturas, diplomados y en la asignación de niveles para algunos cursos de idiomas. Esta iniciativa contó con la colaboración de cuatro técnicos académicos del área de Sistemas

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112 Vol. 3, Núm. 2. abril-junio 2025, págs. 10 - 23

del Departamento de Cómputo y dos técnicos académicos del área de Sistemas, así como un técnico académico del área de Diseño gráfico y Comunicación visual de la Coordinación de Educación a Distancia (CED), quienes trabajaron conjuntamente para la implementación del servidor, la configuración (técnica y visual) de la plataforma, la integración de exámenes y el soporte técnico.

Tras el retorno de la comunidad académica de la Universidad Nacional Autónoma de México (UNAM) a las actividades presenciales en marzo del año 2022, las condiciones especiales sobre la determinación de aforos en espacios cerrados, publicados en los lineamientos generales para las actividades universitarias en el marco de la pandemia de COVID-19 (UNAM, 2021), favorecieron la continuidad de la aplicación de exámenes en línea. Además, con la obtención de otros beneficios significativos como el ahorro de recursos económicos y materiales, la reducción de tiempos administrativos y la automatización de los procesos de evaluación, esta forma de trabajo se ha consolidado en la ENALLT como una práctica vigente y eficiente.

Aunque no hay un examen estandarizado para todos los procesos de admisión de la ENALLT, sí es posible identificar algunas características comunes dado que todos son exámenes que evalúan el dominio de un idioma en particular. De esta manera, fue posible identificar secciones que se trabajaron de manera unificada, como las que corresponden a la comprensión lectora y a la comprensión auditiva. En particular, en esta última se enfrentaron retos que llevaron a una solución a la medida que fue más allá de realizar ajustes en la configuración de Moodle, pues se necesitaba que los archivos de audio sólo se reprodujeran un determinado número de veces en ciertas preguntas, sin recurrir a opciones de software comercial.

Así, el objetivo de este reporte es exponer de manera general el proceso de desarrollo de los exámenes en línea de dominio de lenguas, desde el proceso de planeación en Moodle, la configuración del servidor y los retos de integración del contenido, poniendo énfasis en la solución implementada a partir del año 2022 para el control de reproducción de audios. Esto tiene la finalidad de mejorar el funcionamiento de la sección de comprensión auditiva con base en las primeras lecciones aprendidas durante la pandemia.

2. DESARROLLO TÉCNICO

El proceso inició en las reuniones de trabajo que se llevaron a cabo con la participación de personal de las áreas involucradas en la aplicación de exámenes: Coordinaciones de Licenciaturas, Departamentos de Lengua, Sección Escolar, Cómputo y Educación a Distancia. Desde el punto de vista de ingeniería de software, los resultados de esta fase inicial fueron los requerimientos para la implementación de un sistema de exámenes en línea.

Tras contar con la documentación de los requisitos del sistema, se hizo un análisis de las opciones de implementación y se optó por Moodle (acrónimo de *Modular Object-Oriented Dynamic Learning Environment*) al considerar varias razones. En primer lugar, Moodle era la plataforma que ya se utilizaba en la entidad académica para la gestión de cursos semipresenciales y a distancia. Otra razón fue el breve período del que se dispuso para el desarrollo de la solución tecnológica, la programación de un sistema propio requería más tiempo. Además, se contaba con el antecedente del estudio realizado en 2015 en el marco del Seminario de Plataformas Libres para la Educación Mediada por TIC, donde se llegó a la conclusión de que “Moodle era la mejor opción hasta ese momento al ser la plataforma con mayores capacidades tecnológicas sin demasiada complejidad en su instalación y mantenimiento” (De Mendizábal y Valenzuela, 2015), y aún para el año 2019, esta plataforma seguía siendo la mejor opción

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112 **Vol. 3, Núm. 2.** abril-junio 2025, págs. 11 - 23

como plataforma de aprendizaje virtual para la mayoría de las instituciones de educación superior en el país (Ponce-López, 2019).

Moodle, al ser un producto de código abierto, ofrece la flexibilidad para diseñar un entorno virtual a la medida de las necesidades de cada institución educativa y posibilita realizar modificaciones que abarcan desde ajustes en la interfaz visual hasta la integración de nuevas funcionalidades. Esto también ha propiciado tanto la incorporación de otros desarrollos de software que pueden integrarse a la plataforma para aumentar la gama de recursos y actividades que ofrece, como la disponibilidad de paquetes de traducción de su interfaz para más de 100 idiomas, que facilita la gestión de contenidos en distintas lenguas en el mismo sistema. En particular, para la evaluación en línea, es posible destacar las siguientes ventajas: posibilidad de retroalimentación inmediata para los estudiantes, mayor fiabilidad con las calificaciones automatizadas en reactivos cerrados, estilos de preguntas más dinámicos e interactivos, la creación de bancos de reactivos y la generación de informes de resultados personalizados (Koneru, 2017).

Con relación a la satisfacción de la experiencia de uso de Moodle, también se han reportado buenos resultados en la literatura. En una revisión de trabajos de investigación publicados entre 2001 y 2019, para conocer el grado de satisfacción de los principales actores dentro del proceso de enseñanza-aprendizaje relacionado con el uso de herramientas tecnológicas (Garcia-Murillo et al., 2020), los autores encontraron que Moodle proporciona a sus usuarios un elevado nivel de satisfacción tecnológica y que esto no es influenciado por factores como la cantidad de participantes, la aplicación de un estándar de evaluación ni la comparación de Moodle con otros sistemas de gestión del aprendizaje. Por otro lado, en un estudio más enfocado a la evaluación de Moodle como sistema base para la aplicación de exámenes en línea a gran escala (Hillier et al., 2018), los estudiantes reportaron una experiencia positiva al ser cuestionados sobre aspectos de usabilidad y confiabilidad.

Por lo anterior, una vez que se retomaron las actividades presenciales, se decidió continuar con el uso de Moodle, ahora como un sistema de gestión de exámenes que se configuró de manera independiente a las plataformas existentes para cursos en la ENALLT, y trabajar en mejoras a las adaptaciones realizadas para cubrir las necesidades específicas de los exámenes de lenguas.

2.1 CONFIGURACIÓN DEL SERVIDOR E INSTALACIÓN DE MOODLE

El sistema de exámenes en línea se encuentra instalado en un servidor virtualizado que cuenta con 64 GB de memoria RAM y 50 GB de espacio de almacenamiento. Como sistema operativo, se eligió la distribución GNU/Linux "Debian", debido a su estabilidad, seguridad, madurez y a la facilidad que otorga para mantener actualizado el sistema. Asimismo, se eligió a "Apache HTTP Server" como servidor web y a "MariaDB" como manejador de bases de datos. Si bien en un inicio se trabajó con sub-versiones de PHP7, actualmente se emplea PHP8 debido a los requerimientos de la versión 4.4 de Moodle.

Con la finalidad de facilitar las actualizaciones de Moodle, se decidió emplear el control de cambios con *Git*[1] para la instalación de la plataforma (Moodle, 2024b). Mediante esta modalidad, las actualizaciones pueden realizarse de manera rápida y sin muchos contratiempos, ya sea para versiones menores o incluso para actualizaciones de versiones principales (como la que se llevó a cabo para pasar de la versión 3.x a la 4).

1 ***Git*** es un sistema de control de versiones.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112

Vol. 3, Núm. 2. abril-junio 2025, págs. 12 - 23

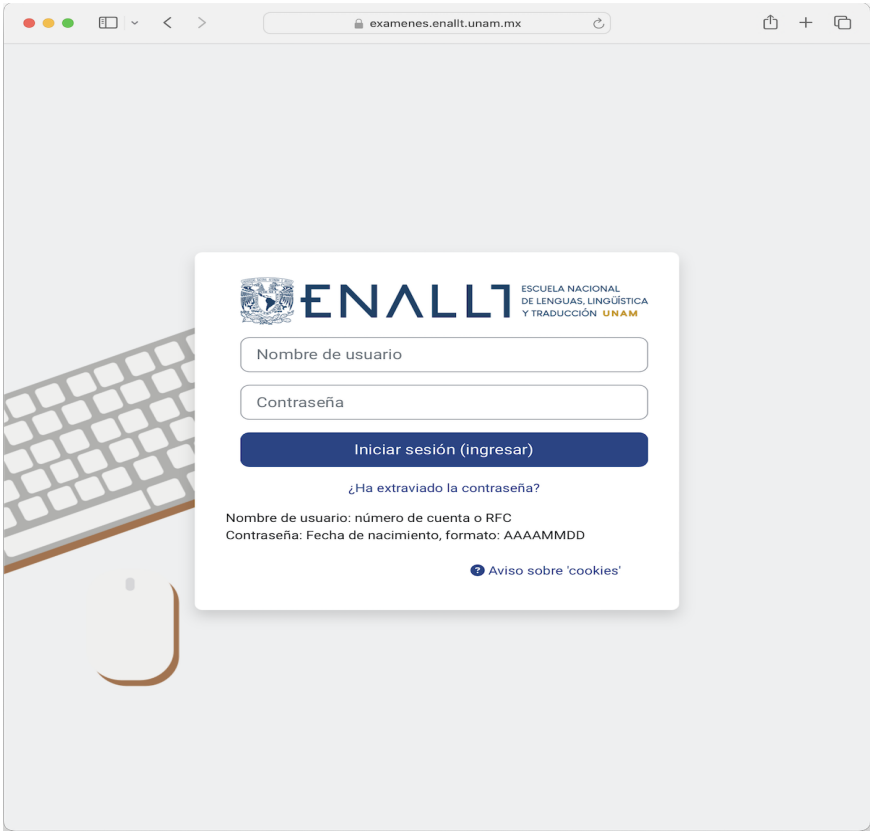
Por otra parte, para asegurar la eficiencia del servidor, en particular del desempeño de Moodle, se configuró el servidor web para emplear el MPM *event*, así como PHP-FPM en lugar del módulo de apache para PHP, de acuerdo con las recomendaciones de desempeño publicadas en el sitio oficial de Moodle (Moodle, 2024c).

Asimismo, se realizaron pruebas de estrés con ayuda de la herramienta JMeter, lo cual permitió determinar que la aplicación de exámenes podía llevarse a cabo con fluidez hasta con mil usuarios simultáneos. No obstante, como medida preventiva, se decidió tener sesiones de aplicación de examen con un máximo de 700 sustentantes concurrentes.

Por último, se aplicaron configuraciones de seguridad (Moodle, 2024d) y se siguieron otras recomendaciones, como el registro del sitio en Moodle.org (Ally, 2022). En la Figura 1, se muestra la página de inicio del sitio de Exámenes de la ENALLT.

Figura 1

Sitio electrónico de exámenes de admisión en línea, ENALLT



2.2 ESTRUCTURA GENERAL DE LOS EXÁMENES

Los exámenes de admisión para los cursos de lenguas usualmente se dividen en secciones que corresponden a las habilidades que el alumno debe dominar para poder comunicarse en una lengua

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112

Vol. 3, Núm. 2. abril-junio 2025, págs. 13 - 23

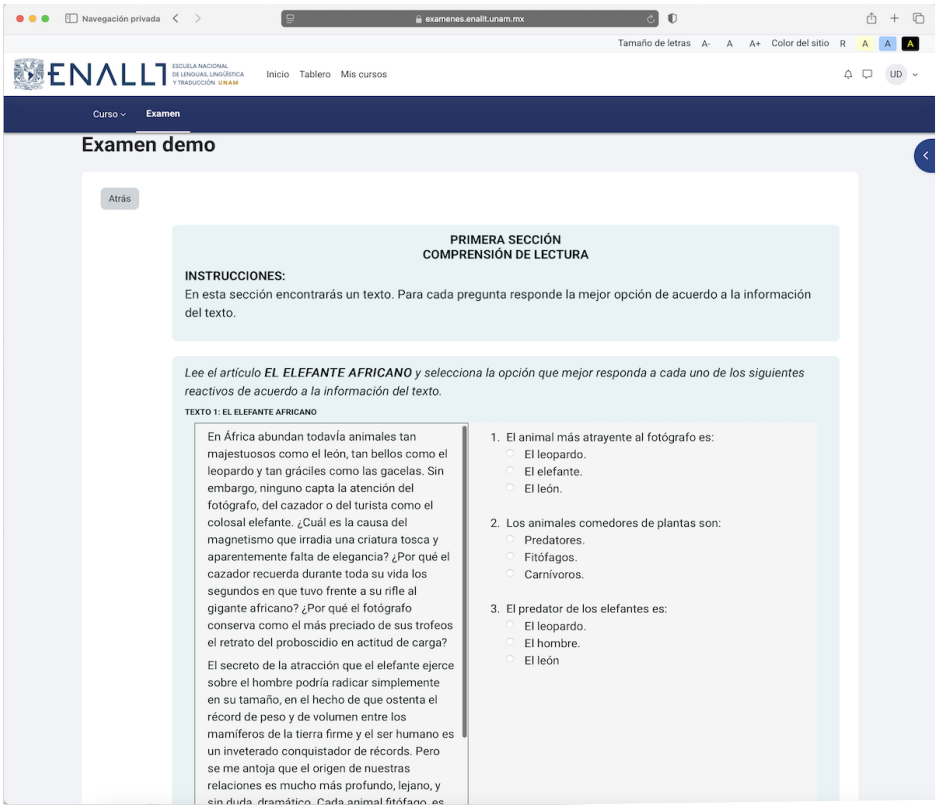
extranjera: leer, escuchar, escribir y hablar. Dependiendo de las necesidades de cada examen, en ocasiones se prescinde de alguna sección o se incluye otra en la que se evalúa vocabulario o aspectos de gramática. Sin embargo, las secciones que invariablemente se mantienen son las de comprensión lectora y comprensión auditiva, las cuales requirieron ajustes especiales para su implementación.

De manera general, cada examen corresponde a un curso en Moodle, configurado con formato de actividad única de tipo "examen". Los reactivos fueron creados con el banco de preguntas de cada examen, organizados en tantas categorías como secciones posea su diseño. Para la captura de preguntas, se emplearon distintos tipos de éstas, entre los que destacan: opción múltiple, completar espacios con respuesta corta o menú plegable, falso y verdadero, relación de columnas y preguntas de tipo ensayo.

En el caso de las secciones de comprensión lectora, se solicitó una navegación independiente entre secciones y reactivos, pero organizada en columnas que permitiera a los sustentantes identificar las respuestas de las preguntas dentro del texto de manera más cómoda. Para ello, se optó por emplear el tipo de pregunta *cloze*, que permite la inclusión de diferentes tipos de preguntas en una sola (Moodle, 2024a), con lo cual, se puede dar el formato especial solicitado en los reactivos que así lo requieran. Como se puede apreciar en la Figura 2, se diseñaron dos columnas con una altura fija y con desplazamiento vertical habilitado para permitir el desplazamiento en el texto sin perder de vista las preguntas o, en caso contrario, desplazarse por las preguntas sin perder de vista el texto.

Figura 2

Vista previa de una sección de comprensión de lectura en un examen demo



https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112

Vol. 3, Núm. 2. abril-junio 2025, págs. 14 - 23

Para la comprensión auditiva, se requería que los sustentantes escucharan audios, limitando el número de reproducciones y restringiendo el acceso a los controles de reproducción para evitar que se pudieran pausar o descargar.

De manera nativa, Moodle permite la inclusión de recursos multimedia en cualquier parte del contenido editable con un editor de texto enriquecido, pero no tiene las opciones de configuración que se necesitaban para estos exámenes. Cuando se inició con los exámenes en línea para admisión en la ENALLT, no se contaba con una alternativa en el directorio de *plugins* de terceros para esta plataforma. Más adelante, en agosto de 2022, se publicó una solución de software comercial para Moodle denominada *Strict Audio Player,* el cual permitía incrustar un *widget* con un reproductor de audio que limitaba la cantidad de veces que podía reproducirse un recurso (Poodll, 2022). No obstante, requería del uso de complementos para un editor de HTML específico, Atto, y de los servicios de almacenamiento en la nube de Poodll, además de que no resolvía la personalización del control de reproducción para evitar la pausa una vez iniciadas las reproducciones.

A pesar de que la integración de complementos de Poodll podía resolver parcialmente el requerimiento, uno de los inconvenientes fue la adquisición de la suscripción ya que, como institución pública, en la universidad se privilegia la búsqueda de soluciones basadas en software libre. Además, al depender de un *plugin* desarrollado por otros, la actualización de la plataforma quedaría condicionada a las actualizaciones de estos complementos, los cuales no siempre son actualizados con la misma frecuencia con la que evoluciona el código de Moodle.

En consecuencia, se optó por trabajar en un desarrollo propio para satisfacer las configuraciones en los reactivos que necesitaran de la inserción de audios. Adicionalmente, se evaluó la forma de minimizar la carga del servidor, causada por las peticiones simultáneas durante la realización de los exámenes, para lo cual, el ahorro de ancho de banda se volvió un objetivo crucial para garantizar un servicio eficiente en cada evaluación. Por lo anterior, una propuesta fue alojar los recursos de audio en alguna plataforma para almacenar y compartir videos como YouTube o Vimeo. Asimismo, al tratarse de contenido de exámenes para procesos de admisión, se concluyó que era necesario cuidar la confidencialidad de los archivos de audio, con lo cual, surgió una siguiente propuesta: almacenar los archivos de audio en el servicio de Google Drive. La decisión por esta última opción se debió en gran medida a que la ENALLT cuenta con una suscripción a *Google Workspace for Education*, la cual es suficiente para el trabajo requerido, aún cuando no se tenga almacenamiento ilimitado.

Asimismo, fue necesario realizar pruebas de integración y diseñar e implementar un *script* integral que no solo cambiara la apariencia del reproductor multimedia y limitara el número de reproducciones, sino que también tuviera la capacidad de validar sesiones y guardar el registro de las peticiones en la base de datos.

De esta forma, los exámenes pueden:

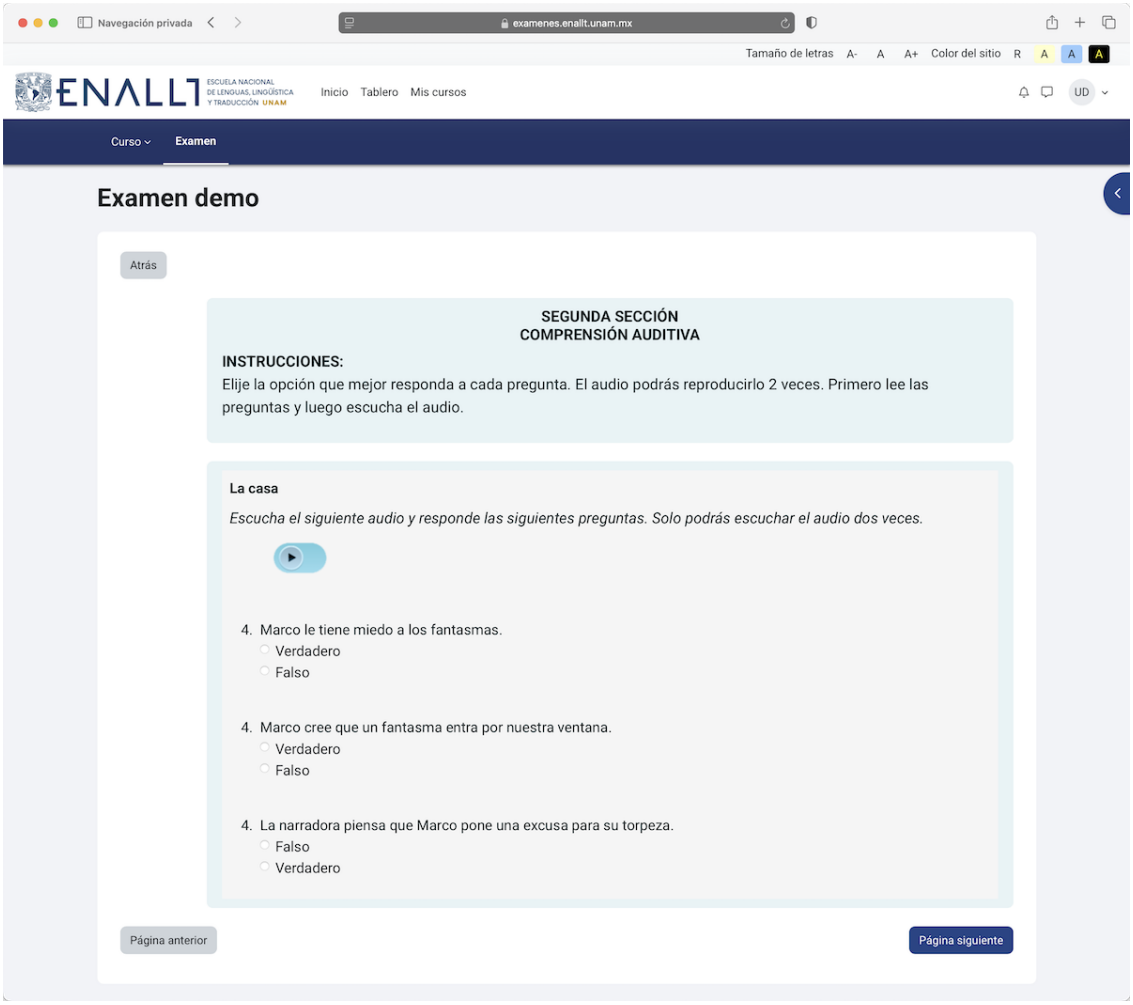
- Incluir recursos de audio de forma incrustada en el contenido del examen, es decir, sin necesidad de utilizar enlaces para visualizar o reproducir los archivos externos en otras ventanas y sin importar el tipo de pregunta en Moodle.
- En el caso de los audios, se limita el uso de controles de reproducción para que no se pueda pausar la reproducción y tampoco sea posible descargar el archivo como se muestra en la Figura 3; mientras se lleva a cabo la reproducción, aparece un contador de tiempo.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112

Vol. 3, Núm. 2. abril-junio 2025, págs. 15 - 23

Figura 3

Vista previa de una sección de comprensión auditiva en un examen demo



- Asimismo, para los audios es posible limitar el número de reproducciones por sesión, de manera que, una vez agotados los intentos indicados, es imposible volver a reproducirlo y así se emula la aplicación que se hacía en los exámenes presenciales.

2.3 METODOLOGÍA PARA LA IMPLEMENTACIÓN DEL SISTEMA DE EXÁMENES DE LENGUAS EN LÍNEA CON CONTROL DE REPRODUCCIONES DE AUDIOS

Tanto para la implementación del sistema de exámenes en línea en general como para el desarrollo del *script* de control de reproducciones de audios en particular, se trabajó con la metodología ágil, ya que ésta tiene como parte de sus principios la "entrega temprana y continua de software con valor" (Beck et al., 2001) y, con ello, ha sido posible mantener flexibilidad y rapidez en el diseño e implementación de la solución para esta necesidad específica.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112

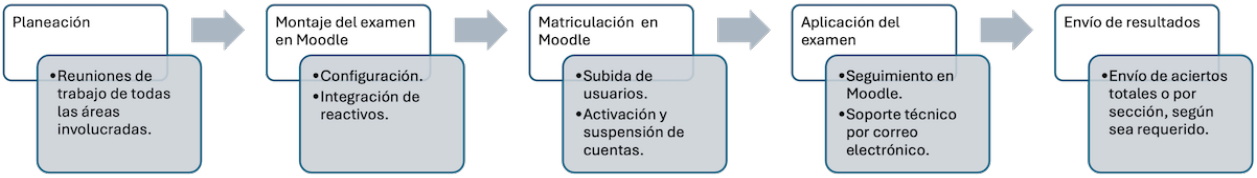
Vol. 3, Núm. 2. abril-junio 2025, págs. 16 - 23

Asimismo, esta forma de trabajo ha permitido que el equipo técnico obtenga retroalimentación en cada sesión de aplicación de exámenes, lo que a su vez, ha favorecido el proceso de mejora continua, corrección de errores y ajustes a las funcionalidades para obtener mejores resultados.

De manera general, se identificaron las etapas desglosadas en la Figura 4 para el sistema de exámenes.

Figura 4

Etapas de la implementación de los exámenes en línea de la ENALLT



Cabe recordar que la etapa de la instalación de la plataforma Moodle implica una instalación totalmente nueva en un servidor independiente, como se explica en la sección 2.1 de este reporte.

En la etapa de "Montaje del examen en Moodle", tanto las coordinaciones de licenciaturas como los departamentos de lengua involucrados enviaron los reactivos y las especificaciones para cada uno de los exámenes. Un equipo de tres Técnicos Académicos pertenecientes a la CED fue responsable de la configuración general e integración de reactivos.

Respecto a la etapa "Matriculación en Moodle", la gestión de usuarios se realizó mediante la carga de archivos en formato CSV (valores separados por comas).

Como estrategia de soporte técnico para la etapa de "Aplicación del examen", se habilitó un correo electrónico en donde se atienden dudas y problemáticas de ingreso al sistema o de reproducciones de audios.

Para finalizar, se envían los resultados para el procesamiento posterior por parte de las coordinaciones de licenciatura, departamentos de lengua y sección escolar. Estos resultados normalmente se envían en un archivo de Microsoft Excel y, en algunos casos, la evaluación de los reactivos de producción escrita se realizan directamente en Moodle; en consecuencia, el reporte de evaluaciones se consulta también en el sistema.

El uso del *script* de control de reproducciones de audios pertenece tanto a la etapa de "Montaje del examen en Moodle" como al de "Aplicación del examen". Los detalles de la implementación de éste se describen en la siguiente sección.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112

Vol. 3, Núm. 2. abril-junio 2025, págs. 17 - 23

2.4 IMPLEMENTACIÓN DEL CONTROL DE REPRODUCCIONES DE AUDIOS

La solución consiste en insertar un *iframe* por cada recurso de audio que se requiere en alguna de las preguntas de examen. De manera general, el código HTML de dicho *iframe* tiene la siguiente sintaxis:

```
<iframe src="URL_SERVIDOR/audio.php?source=ARCHIVO&n=N">
</iframe>
```

Como puede observarse, mediante el *iframe* se carga un *script* en PHP que recibe dos parámetros:

- **ARCHIVO** es el nombre del archivo de audio (.mp3), ya sea como un identificador del archivo en Google Drive, o como nombre del archivo almacenado en el servidor.
- **N** es el número máximo de reproducciones que se permitirán.

Al cargar el *iframe,* se presenta al usuario un botón con el ícono universal de *play* ( ) que, cuando se da clic sobre él, comienza la reproducción del audio mostrando el tiempo de reproducción transcurrido.

Desde que se inició con la aplicación de exámenes en línea en la ENALLT en agosto de 2020, el *script* ha pasado por varias versiones, las cuales se explican brevemente a continuación:

- Versión 1 (agosto de 2020): permitía la inserción de audios a partir de un identificador de recurso compartido en Google Drive y de un parámetro para indicar el número máximo de reproducciones. Este *script* no tenía comunicación con Moodle, por lo que el contador de reproducciones era reiniciado al actualizar la página, lo cual era funcional para los exámenes que tenían navegación restringida y no podían navegar libremente entre las secciones del examen.
- Versión 2 (octubre de 2020): limitaba el número de reproducciones mediante los datos de la sesión de los usuarios en Moodle, de manera que, aún con la navegación libre[2] o actualizando la página, sólo permitía el número de reproducciones establecida.
- Versión 3 (diciembre de 2022): se realizaron mejoras en la notificación al usuario cuando alcanzaba el número máximo de reproducciones permitidas. Se consideraron los casos en los que el *script* se insertaba más de una vez en una misma página para verificar si había un audio reproduciéndose al iniciar una nueva reproducción, evitando así la reproducción simultánea. Se modificó la carga del archivo para intentar una carga desde el servidor del recurso alojado en Google para mejorar la disponibilidad de los recursos; además, se cambió el API Key de Google para evitar problemas de reproducción en audios con una duración mayor a 4 minutos. Finalmente, se agregó más detalle en la bitácora de registros de reproducciones durante las aplicaciones de exámenes.
- Versión 4 (septiembre de 2024): se actualizó el código de comunicación con la API de Moodle debido a la actualización a Moodle de la versión 3.x a la versión 4.

En 2020, el primer acercamiento se hizo con ayuda de las bibliotecas de JavaScript "JQuery" (que facilitó el acceso y manipulación de elementos DOM) y "MediaElement" (que facilitaba la configuración de características para la reproducción de audio). En esa primera versión, la verificación sobre el máximo de reproducciones se realizaba del lado del cliente, con JavaScript, llevando un contador inicializado en cero,

2 Tipo de navegación de Moodle en exámenes donde es posible navegar entre todas las páginas de manera libre, es decir, no hay un orden secuencia obligatorio por lo que los usuarios pueden regresar a cualquier sección mientras no hayan dado por terminado su examen.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112

Vol. 3, Núm. 2. abril-junio 2025, págs. 18 - 23

que se incrementaba cada vez que el audio terminaba de reproducirse. Esto implicaba el problema de que, con sólo recargar la página, el sustentante tenía nuevamente N posibles reproducciones de audio.

Previo al desarrollo de la versión 3 del *script*, se presentaron diversos problemas que se mencionan a continuación:

- Imposibilidad para reproducir los audios.

  - En algunos casos, se identificó que podría deberse a problemas con la conexión de red del usuario.

  - En otros casos, se detectó el escenario en el que de manera simultánea, se hacían demasiadas peticiones a Google Drive, lo que ocasionaba la denegación del acceso a algunos sustentantes.

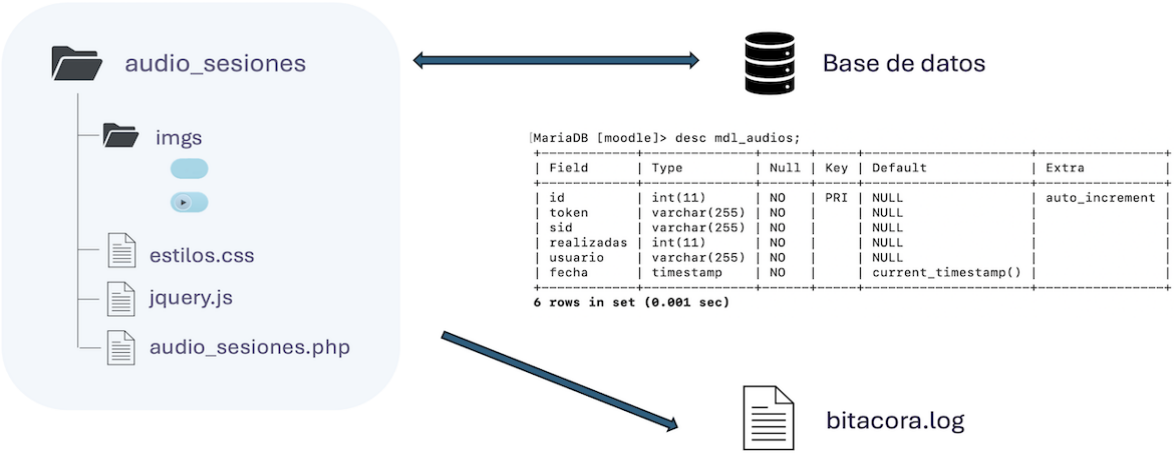
- Si al estarse reproduciendo un audio se daba clic en otro, se escuchaban los dos audios simultáneamente mientras el primero no finalizara, lo cual impedía la comprensión del audio para responder las preguntas correspondientes de esa sección en el examen.

Debido a lo anterior, al retornar a las actividades presenciales en la UNAM, se decidió mantener en uso esta plataforma y dar continuidad al desarrollo del *script* de control de reproducciones de audio. A partir de entonces, se trabajó en una tercera versión más robusta, que permitiera llevar un registro del número de reproducciones desde el servidor y que corrigiera la mayor cantidad de problemas que se habían enfrentado anteriormente.

La estructura de la solución actual se compone de un *script* PHP, un archivo JS con el código de la biblioteca "JQuery", una hoja de estilos para personalizar la visualización del control en los diferentes estados de reproducción de los audios y una carpeta de imágenes que forma parte del diseño del botón. El *script* interactúa con el API de Moodle para verificar la información de la sesión de los usuarios y con la base de datos para revisar el conteo de reproducciones realizadas por sesión, a la vez que registra los eventos asociados a las reproducciones en una bitácora en el servidor, tal y como se muestra en la Figura 5.

Figura 5

Diagrama de las interacciones de la solución implementada y su estructura de archivos



https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112 **Vol. 3, Núm. 2.** abril-junio 2025, págs. 19 - 23

El código del *script* inicia con la verificación de la existencia de la sesión del usuario. Adicionalmente, se obtiene un identificador de sesión llamado *token* que utiliza las funciones provistas por la documentación de Moodle. Este dato se obtenía con la expresión **`($SESSION->logintoken)['core _ auth _ login']['token']`**, sin embargo, con la migración a la versión 4 de Moodle, fue necesario cambiar el código para obtenerlo mediante la función **`core\session\manager::get _ login _ token()`** (*Data Manipulation API|Moodle Developer Resources*, 2024).

Dicho *token* se utilizó para crear registros en una nueva tabla, a la que se denomina "mdl_audios", en la cual se tienen campos para el conteo del número de reproducciones realizadas y una marca de tiempo que es útil para la solución de problemas (ver Figura 5).

Para los casos en los que el examen tuviera más de un audio en la misma sección o página, se agregó código para que, al dar clic en un audio, se verificara la existencia de audios en reproducción por cada *iframe* de dicha sección o página, con el fin de detenerlos antes de comenzar la reproducción del audio en el que se dio clic.

Por otra parte, se decidió implementar un método de redundancia para los casos en los que fallara la carga del audio desde Google Drive; es decir, en caso de error (atributos *onerror* y *onstaled* en la etiqueta de audio), mediante código en JavaScript, se hace una petición con AJAX[3] (*Asynchronous JavaScript and XML*) al servidor para que realice un *switch* en el código de la etiqueta del audio y así cargar el audio desde el servidor en lugar del alojamiento externo.

A su vez, se agregó al *script* una clave de API de Google Drive[4] para que todas las peticiones de recursos fueran autenticadas y autorizadas sin las limitaciones que suele tener Google Drive, si se emplea de manera anónima.

Asimismo, se implementó una bitácora en el servidor que permitiera el registro de cada reproducción, tanto al iniciar como al finalizar, así como de los errores y el uso del *switch* mencionado anteriormente para cambiar la carga de audios desde Google Drive a la carga desde el servidor y viceversa. En dicha bitácora, además de la marca de tiempo de cada evento, se guarda el nombre de usuario, el *token* de sesión, el nombre del audio en cuestión y una leyenda que indica si se cargó un audio, si finalizó o si hubo un error.

Por último, se aplicó un seguimiento y control de cambios con ayuda de la herramienta *Git*, lo cual ha facilitado el trabajo colaborativo y el registro del historial de los cambios realizados.

3. RESULTADOS

La colaboración entre los distintos equipos de trabajo de las áreas involucradas en la aplicación de exámenes en línea de la UNAM ha sido exitosa, en principio, porque se logró dar respuesta inmediata a una necesidad emergente que le permitiera a la ENALLT continuar con sus procesos de admisión aún durante el confinamiento. Por otro lado, las etapas de implementación expuestas en la metodología empleada muestran las interacciones entre los procesos y la organización que se realiza con cada

3 AJAX se refiere a un grupo de tecnologías que se utilizan en aplicaciones web para agilizar la interacción al permitir la comunicación del cliente con el servidor web sin recargas de páginas (IBM, 2015).

4 La API de Google Drive es una interfaz de programación que permite gestionar archivos en dicha plataforma.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112 **Vol. 3, Núm. 2.** abril-junio 2025, págs. 20 - 23

aplicación de examen, lo cual ha permitido mantener un diálogo fluido con los académicos que diseñan los exámenes y una mejora continua de acuerdo con las necesidades específicas de cada examen y con la retroalimentación obtenida a través de los mensajes de correo electrónico recibidos durante el soporte técnico.

Así, desde julio de 2020 a la fecha, se han configurado 18 exámenes en línea: nueve exámenes de requisito para ingreso a licenciaturas y diplomados, cinco exámenes de colocación para cursos de idiomas y cuatro exámenes departamentales de inglés. En promedio, cada año se realizan 30 sesiones de aplicación de estos exámenes con un total de 5,000 aspirantes evaluados. De todas las sesiones programadas, se ha proporcionado soporte técnico vía correo electrónico para que los usuarios reporten los problemas técnicos experimentados durante el examen.

Además de los beneficios asociados con el ahorro de recursos, como el uso de papel y la ocupación de espacios para cada examen, los departamentos de lenguas han experimentado otras ventajas al poder diversificar sus exámenes con la ayuda de los bancos de reactivos de Moodle. Un ejemplo de esto es el examen del Departamento de Alemán, en donde antes se elaboraba una versión de examen para cada semestre y, ahora, al alimentar el banco de reactivos con cuatro versiones de exámenes distintas, la configuración aleatoria por sección les permite incrementar la cantidad de versiones de exámenes; de manera que, tan sólo para la sección de vocabulario, se tienen 15,504 combinaciones distintas, lo cual es el resultado obtenido de la combinación de 20 reactivos tomados en grupos de cinco, es decir, C(20,5); como también se tienen cuatro versiones de bloques de reactivos de gramática, cuatro de comprensión de lectura y tres de comprensión auditiva, el total de posibilidades por nivel es de 15,504 * 4 * 4 * 3 =744,192; y dado que este examen consta de cinco niveles, se tendrían en total $(744,192)^5$ versiones.

En cuanto a los errores detectados durante la aplicación de los exámenes, la mayoría de los reportes que se han registrado corresponden a problemas relacionados con la conexión a internet de los usuarios y con la reproducción de audios. Como se detalló en la descripción de la implementación del *script*, gran parte de los problemas reportados en ese sentido se han mitigado con los ajustes realizados, particularmente, a partir de la adición de la clave API de Google Drive, pues, de acuerdo con los registros de la bitácora en el servidor, de un total de 66,407 reproducciones registradas desde el año 2022, previo a la adición de la clave API, se tuvieron 25,538 reproducciones con 1,798 errores, que corresponde al 7% de los intentos de reproducción; mientras que, después de la adición de la clave API, se registraron 40,869 reproducciones con 2,005 errores, que equivale al 4.9%. Esto representa una disminución de poco más del 2% en los errores generados.

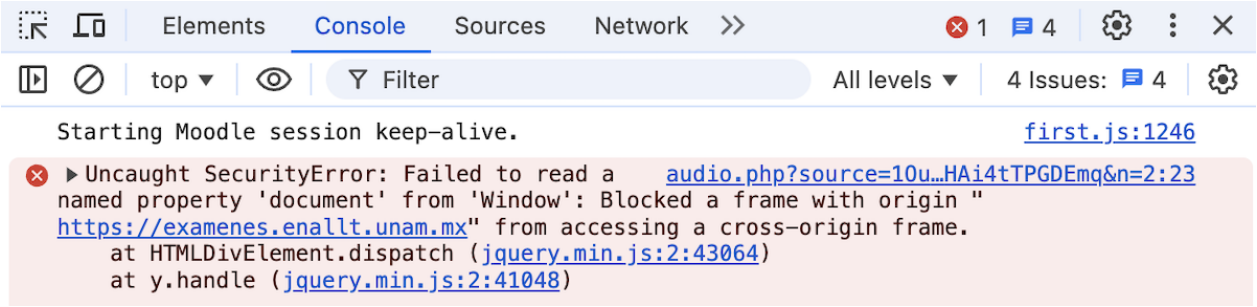
Actualmente, con la última versión del *script,* se han resuelto la mayor parte de los problemas que se han presentado en experiencias previas. No obstante, aún persisten algunas fallas que afectan a un número reducido de sustentantes. Esto motivó la realización de nuevas pruebas del *script,* tratando de igualar las distintas variables involucradas (sistema operativo, navegador, perfil del navegador, extensiones, etc.) tanto como sea posible. De estas pruebas, se detectó una extensión del navegador Google Chrome que genera una excepción del lado del cliente e impide la carga del audio (ver Figura 6).

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112

Vol. 3, Núm. 2. abril-junio 2025, págs. 21 - 23

Figura 6

Error causado por la extensión "Zotero" visto desde la Consola de las herramientas de desarrollador del navegador (Google Chrome)



En estos casos, como solución alternativa, se le indica al sustentante que cambie de navegador o que utilice una sesión de invitado o ventana/pestaña "de incógnito" para iniciar sesión en la plataforma de exámenes, sin que interfieran las configuraciones de extensiones y temas de sus perfiles de usuario.

4. CONCLUSIONES

El proceso de aplicación de exámenes en línea en la ENALLT ha sido el resultado de un trabajo de equipo entre diversas áreas, que, mediante una organización sistemática que pasa por distintas etapas, ha logrado transformar los exámenes de dominio de lenguas del papel al formato electrónico dentro del entorno virtual de la plataforma Moodle.

Aún cuando las circunstancias que originaron el proyecto fueron precipitadas por la emergencia sanitaria, se encontró una solución robusta y adaptable a las necesidades específicas de los exámenes de lenguas basada en Moodle. Además de la flexibilidad en el diseño de los exámenes, se han observado otras ventajas como el ahorro de recursos humanos y económicos al eliminar el uso de papel y la necesidad de espacios físicos para cada sesión de exámenes. Asimismo, se ha diversificado el contenido de los exámenes mediante los bancos de preguntas y se ha reducido también el trabajo de elaboración de reactivos que se hacía anteriormente.

Con relación a los retos de integración de los exámenes en la plataforma virtual, destaca la solución implementada para la limitación de reproducciones de audio. Aunque el uso de soluciones de terceros suele implicar un menor tiempo de integración, la experiencia y resultados obtenidos con el desarrollo de software propio mostró en este caso que, sin la necesidad de una gran cantidad de recursos de desarrollo, ni la adquisición de licenciamiento, se puede dar solución a problemáticas específicas de manera exitosa. Si bien la metodología ágil empleada no necesariamente es la mejor para cualquier proyecto de desarrollo de software, para este caso, probó ser la adecuada porque permitió lograr una solución de manera expedita y con posibilidad de adaptación a los cambios que se han presentado, tal y como se enumeró en las versiones que se han realizado hasta el día de hoy. Además, al estar ya en una etapa de desarrollo medianamente madura y dados los buenos resultados obtenidos, se tiene la motivación para generar una solución más robusta e integrada con la plataforma, de manera que pueda

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112 **Vol. 3, Núm. 2.** abril-junio 2025, págs. 22 - 23

estar disponible para el público en general (como parte de los módulos oficiales de Moodle), pero en particular, para la comunidad de nuestra universidad.

AGRADECIMIENTOS

Queremos agradecer al Mtro. Juan Manuel García Morales, jefe del Departamento de Cómputo, por el apoyo en la coordinación del proyecto, así como a Román Sánchez y Luis Vázquez, miembros del mismo departamento, por su apoyo en la planeación y en la realización de pruebas y monitoreo en las aplicaciones iniciales de exámenes en línea. De manera similar, al Mtro. Víctor Martínez de Badereau, jefe de la Coordinación de Educación a Distancia, por su apoyo en la coordinación del proyecto; a Rodolfo García por su colaboración en el diseño y comunicación visual de la interfaz de la plataforma y su participación junto a Alejandro Ortiz en la gestión de la plataforma en cada una de las aplicaciones de exámenes. Por supuesto, los exámenes en línea tampoco serían posibles sin la colaboración de las Coordinaciones de Licenciaturas y Diplomados, así como los departamentos de lenguas que han participado en la elaboración del contenido de los exámenes.

REFERENCIAS

Ally, S. (2022). Review of Online Examination Security for the Moodle Learning Management System. *International Journal of Education and Development using Information and Communication Technology*, 18(1), 107-124.

Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland y J., Thomas, D. (2001). Principios del Manifiesto Ágil. Recuperado de: https://agilemanifesto.org/iso/es/principles.html

Data manipulation API|Moodle Developer Resources. (2024, 16 diciembre). https://moodledev.io/docs/4.4/apis/core/dml

De Mendizábal, M., y Valenzuela, R. (2015). *Plataformas libres para la educación mediada por las TIC*. S y G Editores.

García-Murillo, G., Novoa-Hernández, P., y Rodríguez, R. S. (2020). Technological satisfaction about Moodle in higher education—A meta-analysis. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, *15*(4), 281-290. https://doi.org/10.1109/RITA.2020.3033201

Hillier, M., Grant, S., y Coleman, M. (2018). Towards authentic e-Exams at scale: robust networked Moodle. In M. Campbell, J. Willems, C. Adachi, D. Blake, I. Doherty, S. Krishnan, S. Macfarlane, L. Ngo, M. O'Donnell, S. Palmer, L. Riddell, I. Story, H. Suri, y J. Tai (Eds.), *Open oceans: learning without borders: proceedings ASCILITE 2018 Geelong* (pp. 131-141). ASCILITE. https://doi.org/10.14742/apubs.2018.1919

IBM. (2015). ¿Qué es Ajax?. https://www.ibm.com/docs/es/rational-soft-arch/9.7.0?topic=page-asynchronous-javascript-xml-ajax-overview

Koneru, İ. (2017). Exploring Moodle Functionality for Managing Open Distance Learning E-Assessments. *Turkish Online Journal of Distance Education*, 18(4), 129-141. https://doi.org/10.17718/tojde.340402

Moodle. (2024a). *Embedded Answers (Cloze) question type - MoodleDocs*. https://docs.moodle.org/405/en/Embedded_Answers_(Cloze)_question_type

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.112

Vol. 3, Núm. 2. abril-junio 2025, págs. 23 - 23

Moodle. (2024b). *Git for Administrators - MoodleDocs*. https://docs.moodle.org/405/en/Git_for_Administrators

Moodle. (2024c). *Performance recommendations - MoodleDocs*. https://docs.moodle.org/404/en/Performance_recommendations

Moodle. (2024d). *Security - MoodleDocs*. https://docs.moodle.org/404/en/Security

Poodll. (11 de agosto de 2022). *Strict Audio Player for Moodle.* [Archivo de video]. Youtube. https://youtu.be/4SVBSAlxBZY

Ponce-López, J.L.(Coord.). (2019). *Estado actual de las tecnologías de la información y la comunicación en las instituciones de educación superior en México: Estudio 2019.* ANUIES.

UNAM. (2021, 16 noviembre). Lineamientos generales para las actividades universitarias en el marco de la pandemia de COVID-19. *UNAM Gaceta.* https://www.cseguimientocovid19.unam.mx/Docs/211116-Lineamientos-generales-para-las-actividades-universitarias-en-el-marco-de-la-pandemia-de-covid-19-161121.pdf

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.107

Vol. 3, Núm. 2. abril-junio 2025, págs. 24 - 34

Clúster de software libre del Laboratorio Nacional de Observación de la Tierra

Alejandro Aguilar Sierra
Instituto de Ciencias de la Atmósfera
y Cambio Climático
Universidad Nacional Autónoma de México
asierra@unam.mx
ORCID: 0000-0003-1018-8521

Información del reporte:

Licencia Creative Commons

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

Para citar este reporte técnico:

Aguilar Sierra, A. (2025). Clúster de software libre del Laboratorio Nacional de Observación de la Tierra. *Cuadernos Técnicos Universitarios de la DGTIC*, 3 (2) páginas(24 - 34).

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.107

Resumen:

Se implementó un clúster de procesamiento y distribución de datos satelitales para el Laboratorio Nacional de Observación de la Tierra, con el objetivo de contar con un sistema de alta disponibilidad para brindar datos de manera robusta y continua a los usuarios. El laboratorio recibe imágenes de satélite constantemente y con ellas se generan productos útiles para estudios territoriales, meteorología y prevención de desastres. El proceso de datos, desde la recepción, la generación de productos, el almacenamiento y la distribución a usuarios finales, requiere un sistema organizado, eficiente y tolerante a fallas. El clúster de procesamiento y distribución de productos satelitales, con base en software libre de código abierto, fue implementado para cubrir estas necesidades.

Palabras clave:

Clúster heterogéneo, cómputo de alto rendimiento, tolerancia a fallas, alta disponibilidad.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.107

Vol. 3, Núm. 2. abril-junio 2025, págs. 25 - 34

Abstract

A cluster for satellite data processing and distribution was implemented for the National Laboratory for Earth Observation, to provide a high-availability system so that robust and continuous data could be offered to users. The laboratory constantly receives satellite images, which are used to generate products that are useful for territorial, meteorology and disaster prevention studies. From reception, product generation, storage, and distribution to end users, data processing operations require an organized, efficient, and fault-tolerant system. A cluster for the processing and distribution of satellite products was implemented to meet these needs, based on open-source software.

Keywords:

Heterogeneous cluster, high-performance computing, fault tolerance, high availability

1. INTRODUCCIÓN

El Laboratorio Nacional de Observación de la Tierra (LANOT), cuya sede es el Instituto de Geografía de la Universidad Nacional Autónoma de México (UNAM), fue creado en 2017 con apoyo del entonces Consejo Nacional de Ciencia y Tecnología y de un consorcio de instituciones de la UNAM y externas, con el fin de proporcionar datos e información de satélite para prevención de riesgos, estudios territoriales y meteorológicos (Aguirre, 2018). Desde entonces ha crecido el número de socios del consorcio, con instituciones tan importantes para la seguridad nacional como el Centro Nacional de Prevención de Desastres y la Secretaría de la Marina. Conjuntando información de diversas fuentes en tiempo casi real, es posible coadyuvar a generar alertas tempranas y prevenir riesgos, monitorear tormentas, actividad volcánica, incendios y posibles accidentes industriales. Además, el LANOT es un espacio para la docencia y la difusión de las ciencias, al que continuamente se incorporan estudiantes, becarios y tesistas.

El objetivo del proyecto presentado en este reporte técnico es contar con un sistema confiable de alta disponibilidad (*High Avalability*) tolerante a fallas (*fail over*) y prevención de denegación de servicio (*Denial of service*), para brindar datos de manera robusta y continua a los usuarios del consorcio LANOT. En este trabajo se reporta el estado actual de un proyecto activo y en desarrollo.

2. DESARROLLO TÉCNICO

2.1 INFRAESTRUCTURA Y NECESIDADES

El LANOT cuenta con tres antenas (véase la Figura 1): una con dirección fija al satélite geoestacionario GOES-East (NOAA, 2017), que abarca el hemisferio americano; una móvil que sigue a algunos satélites de órbita polar como los JPSS de la Oficina Nacional de Administración Oceánica y Atmosférica (NOAA por sus siglas en inglés) (NOAA, 2024); y una pequeña antena para el sistema GeonetCast Américas (NOAA, 2023).

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.107

Vol. 3, Núm. 2. abril-junio 2025, págs. 26 - 34

Figura 1

Antenas del LANOT ubicadas en la azotea del edificio anexo del Instituto de Geografía en Ciudad Universitaria.

Nota. De izquierda a derecha, antena móvil polar, antena de GeonetCast y antena para el satélite geoestacionario GOES-East.

La cantidad de información que se recibe diariamente es considerable, más de 1 TB, y debe ser procesada, publicada y almacenada. El clúster de software libre del LANOT, llamado así porque utiliza únicamente sistemas de código abierto de alta calidad, desarrollados en instituciones como la Universidad de Wisconsin en Madison o la misma UNAM y patrocinado por instituciones como la NOAA y la Administración Nacional de Aeronáutica y el Espacio, (NASA por sus siglas en inglés), fue implementado con este propósito.

Un clúster se compone de un conjunto de servidores, llamados nodos, que unen fuerzas para ejecutar un conjunto de tareas de manera eficiente y robusta. Entre las ventajas que proporciona un clúster están las siguientes:

- Alta disponibilidad: minimiza puntos débiles por medio de redundancia y replicación.
- Alto rendimiento: acelera procesos por medio de paralelismo, concurrencia y distribución de procesos.
- Balanceo de carga: distribuye tareas de acuerdo a la demanda, de modo que haya un equilibrio entre los nodos y ninguno se sature.

Dada la variedad de tareas que se requieren para proporcionar un buen servicio a los usuarios del LANOT y el enorme volumen de productos satelitales, se consideró que un clúster era la solución apropiada. Es heterogéneo porque fue construido en diferentes etapas desde mediados de 2018, con diferentes

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.107 Vol. 3, Núm. 2. abril-junio 2025, págs. 27 - 34

marcas pero al mismo tiempo, sus necesidades son muy específicas, al contrario de un clúster de uso general y multiusuario, como los descritos en la implementación de Grid UNAM (Frausto, 2024) .

Cada nodo participa en alguna de las siguientes operaciones:

1. **Adquisición.** Proporciona la materia prima de todo el proceso: los datos crudos obtenidos en las estaciones receptoras conectadas a las antenas externas.
2. **Procesamiento.** Esos datos crudos deben procesarse para obtener productos útiles para distintas aplicaciones a partir de ellos. Para una descripción detallada de esta operación, así como los productos y sus niveles, ver la sección 2.4.
3. **Publicación.** Los productos ya procesados se ponen al alcance de los usuarios del LANOT por FTP y al público en general por HTTP.
4. **Almacenamiento.** Tanto los datos crudos como los productos de mayor nivel pueden almacenarse para su consulta posterior.

Aunque son de marcas diferentes y con hardware ligeramente distinto, todos los nodos tienen memoria RAM de 250 GB, arreglos RAID de discos con capacidad total de 20 a 40 TB, procesadores Intel Xeon con múltiples núcleos y dos puertos Ethernet de 10 Gb.

2.2 ANTECEDENTES

Originalmente se contaba solamente con los servidores bajo licencia comercial de la empresa *Sea Space*. A mediados de 2018 ya estaban saturados y no podían generar las animaciones que se muestran en la página principal del LANOT (https://www.lanot.unam.mx/), además de que la geolocalización presentaba errores que no se podían corregir porque no se contaba con el código fuente. En 2018 se agregó un primer servidor de software libre con sistema operativo GNU/Linux con el que se empezaron a crear imágenes con geolocalización correcta y animaciones de disco completo y de Estados Unidos de América (EUA) continental (véase la Figura 3). Más tarde en ese mismo año se construyó el sistema de almacenamiento masivo (Aguilar, 2020). Con este sistema se implementó una red local con un *switch* de alta velocidad de 10Gb, donde posteriormente se armaría el clúster.

Antes de la pandemia ya se contaba con dos servidores de software libre: el principal se dedicaba al procesamiento de imágenes del satélite GOES-16 y el secundario al procesamiento de las imágenes polares, pero ambos eran capaces de asumir las tareas del otro en caso de falla o mantenimiento, con lo que ya se contaba con un método básico de redundancia y disponibilidad.

Durante la pandemia se adquirieron nuevos servidores, con lo que se tuvo el suficiente número de nodos para armar un sistema con base en clúster. Al principio se implementó un sistema sofisticado con *Pacemaker* (Pacemaker, 2024) como gestor del clúster y GlusterFS (Gluster, 2024) como un sistema de archivos distribuido, que funcionó en pruebas. Al poner esos servidores en producción, empezaron a fallar, y ante la urgencia de mantener el servicio activo, se tomó la decisión de simplificarlo y armar un sistema con prestaciones similares pero sin herramientas sofisticadas, programas en shell en lugar del *Pacemaker* y el clásico sistema de archivos en red NFS en lugar de Gluster. Este sistema con base en recursos convencionales es lo que se reporta en este trabajo.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.107

Vol. 3, Núm. 2. abril-junio 2025, págs. 28 - 34

2.3 OPERACIÓN

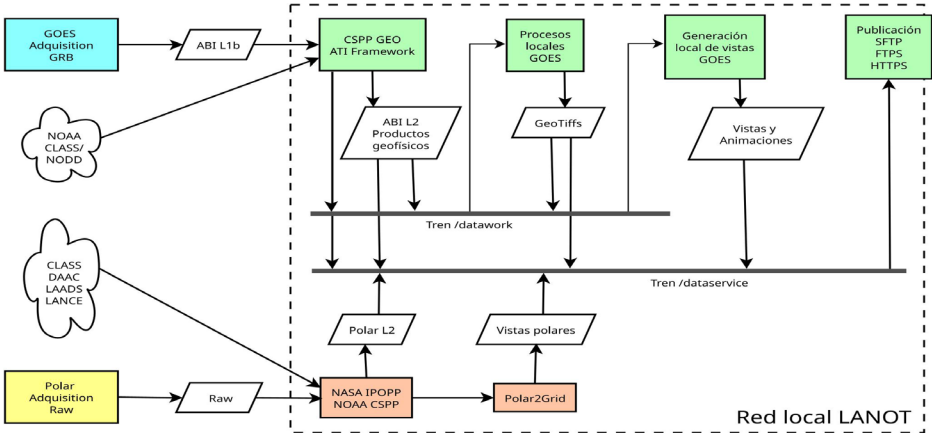
Todos los nodos usan una versión reciente del sistema operativo GNU Linux y se comunican entre sí sin necesidad de contraseña, por medio de llaves públicas con *secure shell*. Un solo usuario virtual realiza todas las operaciones por medio de un conjunto de programas y scripts ejecutables en carpetas replicadas en todos los nodos, con control de versiones `git` en el nodo principal. Los scripts están principalmente escritos en Bash y Python3 y los programas ejecutables en los lenguajes C y C++. El área de trabajo local de cada nodo es su arreglo de discos RAID con 20 a 40 TB de capacidad total.

Al realizar cualquiera de las operaciones descritas en la sección 2.1, los nodos producen nuevos datos que son almacenados. Todos tienen acceso a dos discos virtuales distribuidos, montados por NFS, llamados trenes porque surten de materia prima y transportan productos generados en los distintos nodos (véase la Figura 2). El primero de ellos se llama `/datawork` porque se usa principalmente para trabajo de procesamiento de datos. En su carpeta `input` se guardan los productos de bajo nivel y conforme se generan productos de mayor nivel, se almacenan en la carpeta `output`. En el segundo tren, llamado `/dataservice` pues se dedica al servicio de datos, se almacenan todos los productos que serán publicados. Los nodos de publicación ponen el contenido de `/dataservice` accesible al exterior del clúster por medio de los protocolos FTPS, SFTP y HTTP.

Un primer uso de redundancia en el sistema permite reconstruir cualquiera de los trenes en caso de falla, pues todos los productos generados en los nodos también se encuentran en sus respectivos espacios de disco locales. Toda esta información almacenada en esta parte del sistema es efímera ya que se está renovando constantemente, y sólo permanece unos días o un par de semanas, a lo mucho, pero un conjunto de datos e imágenes selectos se almacenan a largo plazo en el sistema de almacenamiento masivo.

Figura 2

Diagrama de flujo de datos del LANOT



Nota. Del lado izquierdo se muestran los sistemas propietarios de adquisición conectados directamente a las antenas externas, así como proveedores externos de datos auxiliares. Dentro del rectángulo punteado se muestran todos los procesos en la red local aislada y el uso de los trenes de datos. Los únicos servidores con acceso a la red externa son los de publicación.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.107 Vol. 3, Núm. 2. abril-junio 2025, págs. 29 - 34

Toda la operación del sistema es automatizada, impulsada por eventos de recepción de imágenes, que ocurren cada 10 minutos (disco completo), 5 minutos (sector EUA continental) y 1 minuto en el caso geoestacionario, y a intervalos de horas o días en el caso polar. Todas las actividades se monitorean por medio de eventos programados a distintos horarios y cada uno genera un mensaje local de correo electrónico, lo que constituye en conjunto una bitácora del sistema.

En las siguientes secciones se detallan las operaciones definidas en la sección 2.1, con excepción de adquisición, realizada por el sistema propietario TeraScan en los servidores de las estaciones receptoras conectadas a las antenas.

2.4 PROCESAMIENTO

No se pueden usar directamente los datos crudos que se reciben de las antenas, por lo que deben ser procesados. De acuerdo con las convenciones en Percepción Remota (NASA, 2025), cada etapa de procesamiento genera productos de mayor nivel. Los de nivel 1 son usualmente datos ya legibles pero aún no muy útiles pues están en unidades de radiancia. Los de nivel 2 ya están calibrados y georreferenciados y convertidos a unidades geofísicas como albedo, temperatura o presión. Los de nivel 3 ya pasaron por un proceso de control de calidad y están mapeados en una cuadrícula uniforme y una proyección geográfica adecuada a su aplicación. Los de nivel 4 fueron combinados con modelos y datos de otros instrumentos.

Las vistas son productos finales que ya no se pueden procesar más, exclusivos para visualización humana. En la Figura 2 se ilustra el proceso de cómo, a partir de datos crudos o de nivel 1, se generan productos de mayor nivel y vistas útiles. Una descripción más detallada del conjunto de productos que se generan en el LANOT, se encontrará en una nota técnica actualmente sometida a la revista Investigaciones Geográficas (Aguilar y Jiménez, 2025).

Los paquetes de software libre más importantes usados en el LANOT son las bibliotecas NetCDF, GDAL (Warmerdam, 2024), los Paquetes de Procesamiento Satelital de la Comunidad (CSPP GEO, 2024 para procesar imágenes geoestacionarias, CSPP LEO, 2024 para imágenes polares), desarrollados por la NOAA y la UW Madison y el Paquete Internacional de Procesamiento de Observación Planetaria (IPOPP por sus siglas en inglés, detenido en 2024), desarrollado por la NASA. Dentro de lo posible, todas las tareas están configuradas para aprovechar los núcleos de los servidores, lo que confiere alto rendimiento con paralelismo local (no distribuido entre servidores).

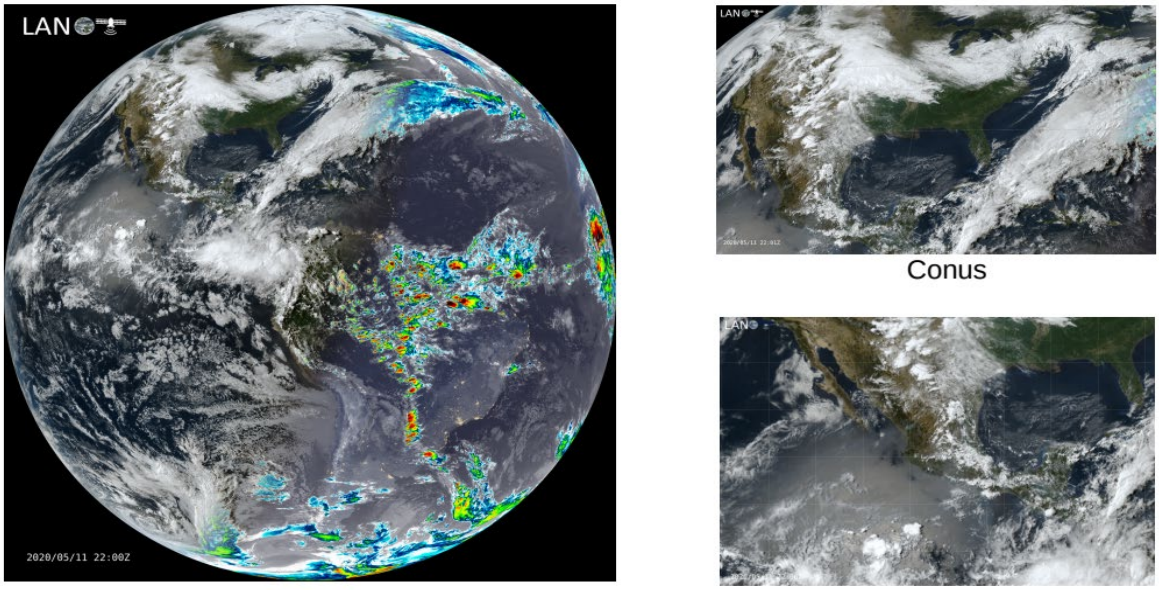
También se desarrollaron programas originales para generar versiones de los mismos productos en el formato GeoTiff (GeoTiff, 2024) y reproyectados, que son más útiles para algunos usuarios que el original NetCDF, por ejemplo, para uso en sistemas de información geográfica, así como recortes de diferentes sectores de interés. Se generan diversas vistas, como los compuestos RGB (combinaciones de distintas bandas para cada color primario) notablemente en color verdadero, que se publican en disco completo cada 10 minutos, y en CONUS (región continental de EUA que incluye a casi todo México) cada 5 minutos, tanto en imagen fija como en animación de las últimas 24 horas, en su proyección geoestacionaria original. También se generan recortes derivados del disco completo y reproyectados de distintos sectores de interés, como los sectores A1 a A5 de la Secretaría de Marina, como se muestra en la Figura 3.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.107

Vol. 3, Núm. 2. abril-junio 2025, págs. 30 - 34

Figura 3

Compuesto RGB resultado de la combinación de las imágenes nocturna y color verdadero, en disco completo, Estados Unidos continental y uno de los recortes en proyección geográfica que abarca México



Conus

Disco completo

México

Actualmente, cuatro nodos dentro del clúster se dedican a tareas exclusivas de procesamiento. Todos tienen una configuración similar, con el mismo hardware, por lo que son intercambiables. Si cualquiera de ellos falla o se desactiva para mantenimiento, sus tareas son asumidas por los nodos restantes. Conforme se adoptan programas con requerimientos más especializados, está en consideración como mejora futura el uso de contenedores tipo docker, con la misma filosofía de que pueda activarse en cualquiera de los nodos de procesamiento.

2.5 PUBLICACIÓN

Los productos procesados son puestos a disposición de los usuarios por medio de nodos especializados en publicación. Todos los nodos de publicación se conectan por un lado a la red local privada del LANOT y por otro a la red externa de 10Gb de la UNAM. El principal método de tolerancia a fallas es la redundancia pues las tareas las realizan dos nodos idénticos en cada caso.

Dos nodos idénticos se especializan en FTPS. Se implementó una configuración a modo del servidor PureFTP (PureFTP, 2024) por eficiencia y seguridad. Si falla uno de los dos, el que queda ofrece exactamente los mismos servicios. También provee balance de carga manual. Si un usuario encuentra uno de los servidores saturado, puede intentar entrar al otro. Cada nodo admite un máximo de 70 usuarios. El balance de carga automático se había implementado con Pacemaker y funcionaba asignando la IP pública al servidor menos activo, pero se suspendió su uso. Una mejora futura obvia será reimplementarlo junto con otras prestaciones que puede ofrecer un gestor de clúster moderno.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.107

Vol. 3, Núm. 2. abril-junio 2025, págs. 31 - 34

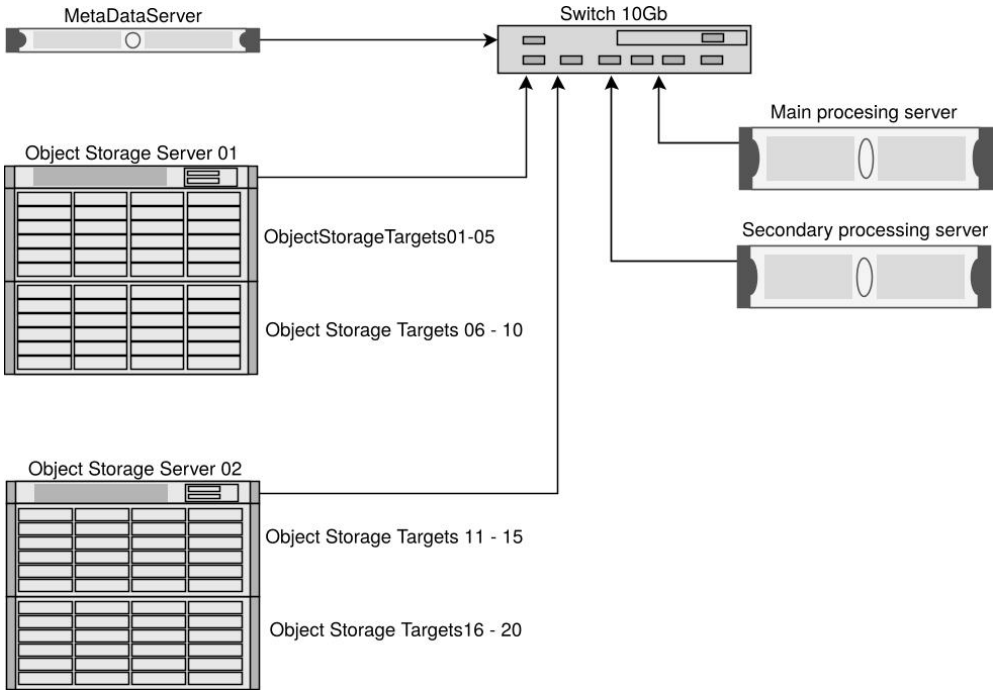
Otros dos nodos se especializan en publicación web con los protocolos HTTP y HTTPS. Usan Apache 2, Postgresql, Django para la página principal interactiva y GeoServer para visualización interactiva de mapas. Si cae uno de los servidores, el otro puede asumir sus funciones, pero al igual que en los servidores FTP, aún no está automatizado su sistema de balance de carga y queda para mejora futura.

2.6 ALMACENAMIENTO

La información guardada en el Sistema de Almacenamiento es persistente, no se borra hasta que sea necesario. El criterio para almacenar información en este sistema es que sean productos originales adquiridos por antena o internet, o que sea costoso volverlos a generar en procesamiento. Se usa un sistema de archivos distribuido con base en Lustre y sus componentes se ilustran en la Figura 4.

Figura 4

Componentes y diagrama de red del Sistema de Almacenamiento del LANOT



Se guardan 24 discos físicos de 12 TB en un gabinete o "cajón" y se organizan en arreglos RAID, cada cajón cuenta con una tarjeta controladora. El sistema Lustre se encarga de organizar estos arreglos para lograr una capacidad total aproximada de un PB con 4 cajones (una explicación más detallada de su armado y funcionamiento se encuentra en Aguilar, 2020).

2.7 SEGURIDAD

El *site* donde se encuentran las estaciones receptoras y los racks con los servidores, se encuentra en el primer piso del edificio anexo del Instituto de Geografía. Aparte de que el acceso al edificio está controlado con una estación de vigilancia, el *site* cuenta con acceso controlado mediante reconocimiento facial y de huella dactilar.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.107

Vol. 3, Núm. 2. abril-junio 2025, págs. 32 - 34

Se implementaron medidas de seguridad para los siguientes riesgos:

- Falla eléctrica. Se cuenta con generadores con base en gasolina, que se activan a los pocos segundos de que haya un corte eléctrico. Dentro del site, se cuenta con UPS redundantes (dos por rack) con batería de media hora que entran de inmediato ante el evento de una falla eléctrica, ya sea corte o descarga, y avisan por correo electrónico cualquier situación. Todos los nodos tienen fuentes redundantes, conectadas cada una a un UPS.
- Falla en discos. Los discos duros del sistema de almacenamiento se organizan con una tarjeta controladora en un arreglo RAID 6 que incluye dos discos de paridad y uno de repuesto por cajón, por lo que puede soportar un máximo de tres discos fallidos. Se monitorea automáticamente cada hora y cuando se daña un disco se envía un reporte por correo electrónico. Si se dañan más de dos discos en el mismo gabinete, se detiene el sistema para evitar daños mayores. Los servidores cuentan también con un sistema local que va de 4 a 10 discos organizados en arreglo RAID 5, lo que permite tener discos virtuales de 20 o 30 TB que soportan que falle un disco físico.
- Prevención de ataques. Los nodos con acceso al exterior tienen una estricta cortina de fuego en su sistema operativo, administrado con *nftables*, que permiten acceso solamente a los puertos autorizados para comunicación con SSH, FTP o HTTP y HTTPS. La mayoría de los nodos están en red local y no es posible entrar a ellos desde el exterior. El servicio sshd está estrictamente configurado para reducir riesgos, como por ejemplo, intentos de acceso directo al usuario root, lo cual no es posible. Para descarga de datos, sólo es posible entrar por FTP seguro, ya sea SFTP (basado en ssh) o FTPS (con base en TLS), y cuenta además con mecanismos para prevenir ataques de denegación de servicio y saturación; hay un límite preestablecido de accesos simultáneos, no se permite la entrada al usuario anónimo, es preciso contar con un usuario y una contraseña y, además, la IP desde la que se quiera entrar debe estar en una lista restringida de IPs.

3. RESULTADOS

El funcionamiento del clúster permite la distribución de más de 100 productos satelitales de varios niveles, que pueden usarse en tareas de monitoreo ambiental, meteorología, prevención de desastres y estudios ambientales.

El LANOT es sede de los siguientes proyectos activos en los que participan distintas instituciones:

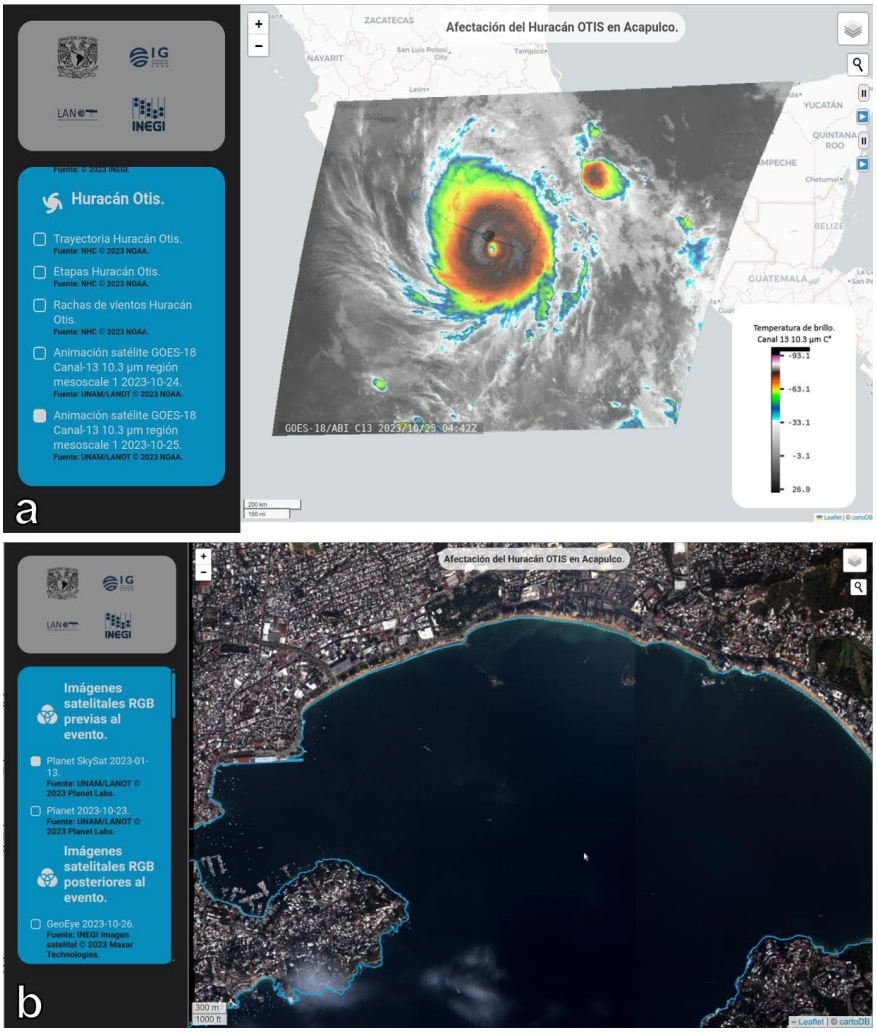
- Monitoreo de ceniza volcánica.
- Monitoreo del sargazo en el Caribe mexicano.
- Detección de puntos de calor para la prevención y el combate de incendios forestales.
- Monitoreo de tormentas tropicales. Un caso especial fue el huracán Otis que devastó Acapulco en octubre de 2023. En la Figura 5 se muestra el visualizador interactivo con el que se puede ver el mismo fenómeno con diferentes niveles de resolución, desde el satélite geoestacionario (5a, 1 km por pixel) hasta un detalle de 0.5 m con la constelación de satélites Planet (5b).
- Monitoreo de los mares territoriales de México.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.107

Vol. 3, Núm. 2. abril-junio 2025, págs. 33 - 34

Figura 5

Ejemplo de cobertura de eventos importantes, como el huracán Otis que devastó Acapulco en el otoño de 2023.



Nota. Se pueden ver vistas de diversas escalas, desde los 2 km por pixel del sector de mesoescala del GOES (a) hasta los mosaicos de 0.5 m por pixel de la constelación Planet (b), así como animaciones, vistas nocturnas y comparaciones de antes y después del evento.

Además de los datos recibidos directamente por las antenas, algunos de estos proyectos usan datos de otros satélites, como los mencionados de la constelación *Planet* o, en particular para el monitoreo del sargazo, imágenes de alta resolución de la constelación Sentinel, de la Unión Europea. En el caso de almacenamiento, antes de que se armara la infraestructura descrita en este reporte, simplemente no era posible almacenar la información recibida y procesada y debía desecharse a los pocos días.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.107

Vol. 3, Núm. 2. abril-junio 2025, págs. 34 - 34

4. CONCLUSIONES

Desde la introducción del primer servidor de software libre en el LANOT, en 2018, hasta la puesta en marcha del actual clúster, los mecanismos para procesar y distribuir datos de origen satelital han mejorado continuamente y atienden una demanda de información cada vez mayor. La construcción de un clúster heterogéneo con base en software libre ha permitido mantener el sistema actualizado al ritmo de las nuevas tecnologías a un costo bajo y satisfacer la demanda de productos satelitales para usuarios del LANOT.

En este trabajo se reportó el estado actual del clúster, cuya prioridad ha sido mantener el suministro de datos continuo y oportuno a los usuarios del consorcio. Entre las mejoras futuras se contempla utilizar contenedores que simplifiquen la administración de los nodos y adoptar un gestor de clúster apropiado a las necesidades del LANOT tipo *Pacemaker* o alguno más actual.

REFERENCIAS

Aguirre, R. (2018). Laboratorio Nacional de Observación de la Tierra (LANOT). Investigaciones Geográficas, 96.

Aguilar, A. (2020). Sistema de Almacenamiento Masivo para el Laboratorio Nacional de Observación de la Tierra (LANOT). Especialidad en Cómputo de Alto Rendimiento, IIMAS, UNAM.

Aguilar, A. y Jiménez, V. (2025). Implementación de una infraestructura de procesamiento satelital con software libre en el Laboratorio Nacional de Observación de la Tierra. Sometido a Investigaciones Geográficas.

CSPP Geo (2024).Community Satellite Processing Package for Geostationary Data. University of Wisconsin-Madison, Space Science and Engineering Center (SSEC). Consultado el 12 de noviembre de 2024. https://cimss.ssec.wisc.edu/csppgeo/

CSPP Leo (2024).Community Satellite Processing Package for Polar orbiting satellite data processing. University of Wisconsin-Madison, Space Science and Engineering Center (SSEC). Consultado el 12 de noviembre de 2024. https://cimss.ssec.wisc.edu/cspp/.

Frausto del Río, S. E. et al. (2024). Grid UNAM, la experiencia en su implementación. Cuadernos Técnicos Universitarios de la DGTIC, 2 (4).

Gluster (2024). Un sistema de archivos de red escalable. Consultado el 12 de noviembre de 2024. https://www.gluster.org/

NASA (2025). ARSET - Fundamentals of Remote Sensing. Consultado el 8 de mayo de 2025. https://appliedsciences.nasa.gov/get-involved/training/english/arset-fundamentals-remote-sensing

NOAA (2017). GOES-R Mission Overview. Consultado el 10 de noviembre de 2024. https://www.goes-r.gov/missiont/mission.html

NOAA (2024). The Joint Polar Satellite System. Consultado el 10 de noviembre de 2024. https://www.noaasis.noaa.gov/POLAR/JPSS/jpss.html

Pacemaker (2024). Pacemaker, un sistema de gestión de clústers altamente disponible. Consultado el 12 de noviembre de 2024. https://clusterlabs.org/projects/pacemaker/

Warmerdam, F. et al. (2024). Geospatial Data Abstraction Library, GDAL. Consultado el 6 de noviembre de 2024. https://gdal.org/

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.88

Vol. 3, Núm. 2. abril-junio 2025, págs. 35 - 44

Implementación de servidor para publicación de proyectos de construcción de sitios web

Margarita González Trejo
Dirección General de Cómputo y de Tecnologías de Información y Comunicación
Universidad Nacional Autónoma de México
mar@unam.mx
ORCID: 0000-0002-6682-8420

Información del reporte:

Licencia Creative Commons

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

Para citar este reporte técnico:

González Trejo, M. (2025). Implementación de servidor para publicación de proyectos de construcción de sitios web. *Cuadernos Técnicos Universitarios de la DGTIC*, 3 (2) páginas (35 - 44).

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.88

Resumen

Se detalla la participación en programas de educación continua en modalidades a distancia y presencial, sustentados en una infraestructura tecnológica robusta adaptada a los requerimientos técnicos definidos de cada uno. Se aborda la implementación de un servidor web como infraestructura de soporte para la publicación de proyectos finales del diplomado "Planeación y construcción de sitios web," donde se realizó la instalación y configuración de servidores con librerías y servicios para lenguajes de programación, bases de datos y gráficos. Asimismo, se destaca la creación de máquinas virtuales como estrategia para facilitar la operación y recuperación de sistemas, especialmente en entornos de desarrollo. Proporcionar esta infraestructura para la capacitación en tecnologías de la información y comunicación presenta desafíos como la rápida evolución del software, lo que exige una comunicación estrecha con los coordinadores académicos para identificar requerimientos, realizar implementaciones adecuadas y brindar soporte técnico continuo. Este compromiso garantiza la operatividad y funcionalidad de los recursos, demostrando la capacidad y experiencia en la provisión de infraestructura tecnológica esencial para programas de capacitación de alto nivel, adaptándose a las necesidades específicas de cada iniciativa lo que asegura un entorno de aprendizaje efectivo.

Palabras clave:

Servidor web, servidor Apache, implementación de servidores, Linux, entorno de aprendizaje efectivo.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.88

Vol. 3, Núm. 2. abril-junio 2025, págs. 36 - 44

Abstract

The present report details continuing education programs participation both in distance learning and in-person modalities, supported by a robust technological infrastructure adapted to the specific technical requirements of each course. The implementation of a web server as a support infrastructure for the publication of final projects of the diploma course "Planeación y construcción de sitios web" is discussed, where servers were installed and configured with libraries and services for programming languages, databases, and graphics. Likewise, the creation of virtual machines stands out as a strategy to facilitate system operation and recovery, especially in development environments. Providing this infrastructure for training in information and communications technologies presents challenges such as the rapid evolution of software, which requires close communication with academic coordinators to identify requirements, deliver appropriate implementations, and provide ongoing technical support. This commitment guarantees the operability and functionality of resources, demonstrating the capacity and experience in providing essential technological infrastructure for high-level training programs, adapting to the specific needs of each initiative to ensure an effective learning environment.

Keywords:

Web server, Apache server, server deployment, Linux, effective learning environment.

1. INTRODUCCIÓN

Como parte de sus funciones, la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) participa en la formación y actualización, en el ámbito de las tecnologías de información y comunicación, de los miembros de la comunidad universitaria y de la sociedad en general a través del desarrollo de diversos programas de educación continua que son impartidos por sus Centros de Extensión Académica en modalidades a distancia y presencial (DGTIC, 2025).

El soporte tecnológico para el desarrollo de estos programas, categorizados en cursos y diplomados, lo constituye una infraestructura de cómputo, comunicaciones y servicios, que se proporciona con la finalidad de facilitar y promover el aprendizaje significativo en los participantes, (Mendoza et al, 2024). Esta infraestructura se asigna con base en los requerimientos técnicos de cada curso; en casos de programas de especialización, como diplomados en los que se desarrollan proyectos funcionales, es necesario proporcionar infraestructura que incluya la instalación y configuración de servidores con librerías específicas y con los servicios necesarios que permitan el despliegue adecuado de lenguajes de programación, bases de datos y gráficos involucrados en los proyectos.

En el caso particular del diplomado "Planeación y construcción de sitios web", el Comité Académico indicó el desarrollo de un proyecto final en el cual los participantes integraran las herramientas que les fueron proporcionadas a lo largo del mismo. Para la publicación de estos proyectos, solicitó un servidor web con las características y configuraciones de hardware y software necesarias para dar soporte a todos los proyectos. La implementación de este servidor significa un valor agregado en el aprendizaje, en relación a las primeras emisiones, mediante el cual los participantes pueden experimentar el trabajo directamente en éste y realizar pruebas de compatibilidad en distintos dispositivos y navegadores web.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.88

Vol. 3, Núm. 2. abril-junio 2025, págs. 37 - 44

Un servidor web es un sistema integrado por elementos de hardware y software que permite almacenar, procesar y entregar archivos de sitios web a través del protocolo de Transferencia de Hipertexto (HTTP) (ComputerWeekly, 2025).

Del lado del hardware, se encuentran equipos de cómputo con recursos de conectividad, procesamiento y memoria suficiente para atender el volumen calculado de información a transferir y de peticiones de usuarios (Balkhi, 2024); del lado del software, existen diversas tecnologías como Internet Information Services de Microsoft, NGINX y Apache, siendo este último el más antiguo y de los más utilizados ya que es de código abierto y puede implementarse en sistemas Windows, Mac OS y Linux (Ramírez, 2019). Los servidores web normalmente se complementan con otras herramientas como gestores de bases de datos y lenguajes de programación con los cuales se construyen sitios que permiten el almacenamiento de datos y el intercambio de información con los usuarios.

Este reporte se presenta con el objetivo de describir la implementación del servidor web para la publicación de los proyectos finales de los participantes en el diplomado "Planeación y construcción de sitios web".

2. DESARROLLO TÉCNICO

La implementación de un servidor para un propósito específico presenta retos como:

- Seleccionar el equipo de cómputo que cuente con los recursos necesarios y suficientes de procesamiento, disco duro y memoria.
- Garantizar la disponibilidad del servidor: esto implica contar con un servicio de respaldo de energía eléctrica, una conexión a Internet estable con suficiente ancho de banda y un sistema cortafuegos para protección de la red interna.
- Seleccionar el sistema operativo adecuado junto con las librerías requeridas para el despliegue correcto de la información que el servidor entregará.
- Verificar la integración de las diferentes herramientas de software: sistema operativo, servidor web, gestor de base de datos y lenguaje de programación.
- Configurar apropiadamente la presentación de errores, considerando que se trata de un servidor de desarrollo.
- Otorgar los permisos necesarios a cada cuenta de usuario que permitan la publicación de los proyectos.

En este caso, al tratarse de un servidor web para la publicación de proyectos finales del diplomado "Planeación y construcción de sitios web", son los coordinadores académicos quienes, al haber analizado cada proyecto, concentran la información de los requerimientos técnicos. A partir de este momento y en el marco de una metodología, se inician los trabajos correspondientes con el propósito de lograr que la infraestructura proporcionada satisfaga los requisitos para la publicación de los proyectos realizados por los participantes en el diplomado.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.88

Vol. 3, Núm. 2. abril-junio 2025, págs. 38 - 44

2.1 METODOLOGÍA

2.1.1 SOLICITUD DE LA INFRAESTRUCTURA

En la solicitud que se recibió, se especificaron los siguientes requisitos de software y librerías: Servidor Apache versión 2.4 con el módulo mod_rewrite activado; lenguaje de programación PHP versión 8.2 con los módulos curl, intl, mbstring; pdo_mysql; mysql; mysqli y soap; gestor de bases de datos MySQL versión 8 o MariaDB 10; habilitación del protocolo secure shell para conexión remota; cuentas de acceso para los participantes con permisos para publicación web; una cuenta de acceso con privilegios para publicación web; una cuenta de acceso con privilegios de administración del sistema operativo y una cuenta para la administración del gestor de la base de datos.

En la solicitud no se indicó un sistema operativo distinto a Ubuntu 20.04, que es el que se emplea por política institucional; se trata de una distribución de GNU/Linux con la que ya se tiene experiencia en otros proyectos, basada en el modelo de desarrollo de software libre y de código abierto compatible con gran variedad de hardware sin exigir grandes recursos; además, proporciona actualizaciones de seguridad sin costo, cuenta con características de robustez, seguridad, estabilidad, trabajo multitarea y multiusuario necesarias para dar soporte a los servicios requeridos y para la cual existe disponibilidad del software solicitado.

2.1.2 DETERMINACIÓN DEL HARDWARE NECESARIO

Para la instalación del sistema operativo y los servicios solicitados, se empleó una computadora de escritorio con las siguientes características: procesador Intel Core i5-4460 a 3.20 GHz, memoria RAM de 8 gigabytes, disco duro de 250 gigabytes y tarjeta de red RTL8111/8168/8411 PCI express gigabit ethernet controller. Estos recursos son superiores a los solicitados para esta versión del sistema operativo, el cual recomienda, como mínimo, un procesador de un solo núcleo a 2 GHz, 2 gigabytes en memoria RAM y 10 gigabytes de espacio en disco duro. El enlace de datos del lugar en donde se ubicó el servidor es de 50 megabits por segundo tanto para descarga como para subida de datos.

2.1.3 INSTALACIÓN Y CONFIGURACIÓN DEL SOFTWARE

Antes de la instalación y configuración del software solicitado, se instaló Ubuntu, se configuró la cuenta de administrador, se verificó la conectividad del equipo, se realizó la actualización del sistema operativo y se configuraron los repositorios de descarga. Posteriormente, se ejecutó el siguiente procedimiento para la instalación del servidor Apache, del gestor de bases de datos MariaDB y del lenguaje de programación PHP:

1. Se descargó e instaló el software en las versiones solicitadas en los requerimientos: Apache, MariaDB y PHP, así como sus dependencias; las dependencias se instalaron de manera automática previa confirmación de autorización solicitada por el sistema operativo.

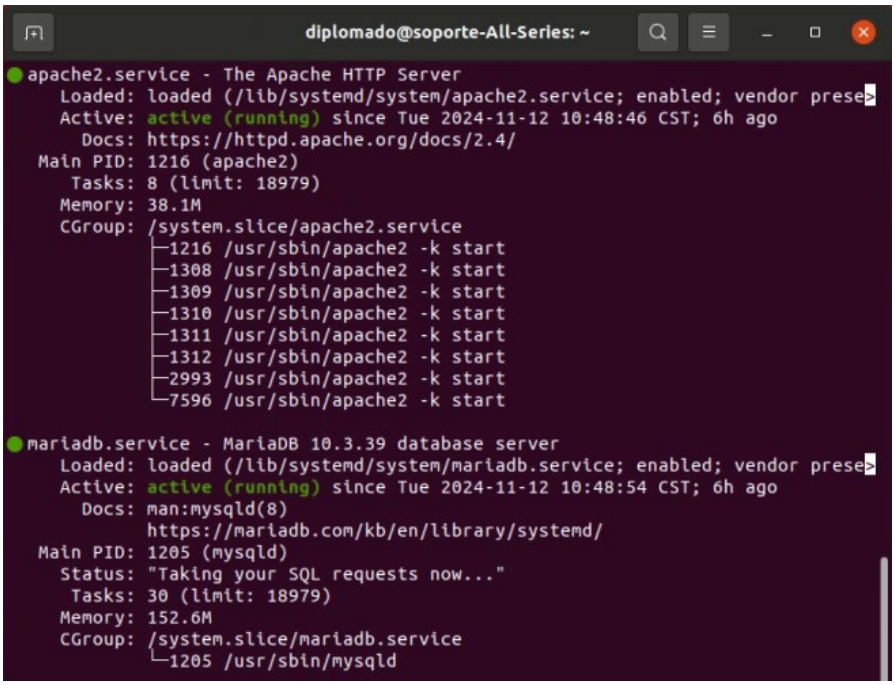
2. Se realizó la prueba de funcionamiento de los servicios de Apache y MariaDB para verificar que se encontraran correctamente instalados y activos, como se muestra en la Figura 1.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.88

Vol. 3, Núm. 2. abril-junio 2025, págs. 39 - 44

Figura 1

Prueba de los servicios Apache y MariaDB



3. Se realizó la configuración del *firewall* de Ubuntu para permitir las conexiones externas al servicio web.

4. Se configuró el servidor Apache: se editó el archivo */etc/httpd/conf/httpd.conf* para indicar el nombre del servidor (directiva ServerName) y se habilitó el servicio de *Directorios Web por usuario* (directiva UserDir) con el propósito de permitir la publicación web a cada usuario a través de una URL del tipo *http://servidor.com/~nombredeusuario* (Apache Software Foundation, 2024).

5. Se estableció la contraseña para el usuario root del gestor de bases de datos MariaDB y se eliminaron los usuarios anónimos.

6. Se editó el archivo de configuración del lenguaje de programación PHP (php.ini): se estableció la zona horaria correspondiente a la Ciudad de México, se habilitó el acceso para la ejecución de aplicaciones en consola, y se habilitó la directiva para el tratamiento de errores, considerando que se trata de un servidor de desarrollo.

7. Se reiniciaron los servicios de Apache para actualizar las configuraciones realizadas.

8. Se hizo una prueba de configuración de los servicios por medio de un script en PHP, la salida se presenta en la Figura 2.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.88

Vol. 3, Núm. 2. abril-junio 2025, págs. 40 - 44

Figura 2

Prueba de configuración del servidor web

PHP Version 8.2.13

System	Linux soporte-All-Series 5.15.0-102-generic #112~20.04.1-Ubuntu SMP Thu Mar 14 14:28:24 UTC 2024 x86_64
Build Date	Nov 24 2023 08:46:50
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.2/apache2
Loaded Configuration File	/etc/php/8.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.2/apache2/conf.d
Additional .ini files parsed	/etc/php/8.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/8.2/apache2/conf.d/10-opcache.ini, /etc/php/8.2/apache2/conf.d/10-pdo.ini, /etc/php/8.2/apache2/conf.d/20-bcmath.ini, /etc/php/8.2/apache2/conf.d/20-bz2.ini, /etc/php/8.2/apache2/conf.d/20-calendar.ini, /etc/php/8.2/apache2/conf.d/20-ctype.ini, /etc/php/8.2/apache2/conf.d/20-curl.ini, /etc/php/8.2/apache2/conf.d/20-exif.ini, /etc/php/8.2/apache2/conf.d/20-ffi.ini, /etc/php/8.2/apache2/conf.d/20-fileinfo.ini, /etc/php/8.2/apache2/conf.d/20-ftp.ini, /etc/php/8.2/apache2/conf.d/20-gettext.ini, /etc/php/8.2/apache2/conf.d/20-iconv.ini, /etc/php/8.2/apache2/conf.d/20-imagick.ini, /etc/php/8.2/apache2/conf.d/20-intl.ini, /etc/php/8.2/apache2/conf.d/20-mbstring.ini, /etc/php/8.2/apache2/conf.d/20-mysqli.ini, /etc/php/8.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.2/apache2/conf.d/20-phar.ini, /etc/php/8.2/apache2/conf.d/20-posix.ini, /etc/php/8.2/apache2/conf.d/20-readline.ini, /etc/php/8.2/apache2/conf.d/20-shmop.ini, /etc/php/8.2/apache2/conf.d/20-sockets.ini, /etc/php/8.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.2/apache2/conf.d/20-sysvsem.ini, /etc/php/8.2/apache2/conf.d/20-sysvshm.ini, /etc/php/8.2/apache2/conf.d/20-tokenizer.ini, /etc/php/8.2/apache2/conf.d/20-zip.ini
PHP API	20220829
PHP Extension	20220829
Zend Extension	420220829
Zend Extension Build	API420220829,NTS
PHP Extension Build	API20220829,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
Zend Max Execution Timers	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, bzip2.*, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v4.2.13, Copyright (c) Zend Technologies
with Zend OPcache v8.2.13, Copyright (c), by Zend Technologies

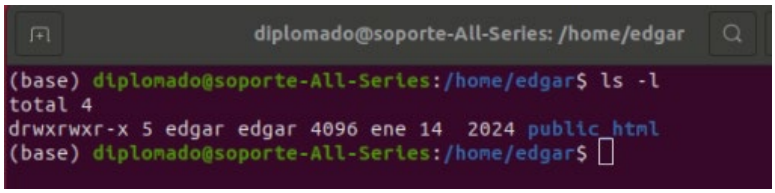
9. Se creó una cuenta para pruebas y el directorio para publicación de proyectos. Este directorio debe ubicarse debajo de la ruta /home/username en cada cuenta de usuario y su nombre debe ser public_html; posteriormente, se asignaron permisos de lectura, escritura y ejecución para el propietario del directorio, y de lectura y ejecución para otros usuarios, como se muestra en la Figura 3.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.88

Vol. 3, Núm. 2. abril-junio 2025, págs. 41 - 44

Figura 3

Ubicación del directorio de publicación del sitio web



10. Se enviaron las credenciales de acceso al instructor para que realizara las pruebas de funcionamiento; una vez que se recibió su aprobación, se procedió a crear la cuenta de usuario y el directorio de publicación del sitio web para cada participante.

Desde el inicio del desarrollo de los proyectos y hasta su publicación, se proporcionó soporte técnico para realizar los ajustes de configuración necesarios, asistir a los participantes en el acceso a sus cuentas y verificar la disponibilidad 24/7 del servidor.

2.1.4 INHABILITACIÓN DEL SERVIDOR

Una vez recibida la notificación de que los proyectos habían sido evaluados y la confirmación de que los participantes del diplomado habían realizado sus respaldos finales, se procedió a deshabilitar el servidor: esto como política de seguridad para evitar accesos no permitidos en un futuro y para permitir que la infraestructura empleada en su implementación pudiera destinarse a otros proyectos. El proceso se documentó para tenerlo como referencia ante futuras solicitudes, pero teniendo en cuenta que los requerimientos pueden cambiar debido a la evolución natural de las diversas herramientas de software involucradas, por lo que las configuraciones deberán ser las que se requieran y correspondan en su momento.

3. RESULTADOS

Con los servicios que se habilitaron en el servidor, los participantes realizaron la publicación de sus proyectos, en los cuales integraron herramientas de diseño gráfico, emplearon el gestor de base de datos para almacenar información que se ingresó desde formularios de captura y realizaron programas para gestionar dicha información.

Durante el proceso de desarrollo de los proyectos, se analizó el comportamiento del servidor, encontrando que respondió al 100% de las conexiones solicitadas.

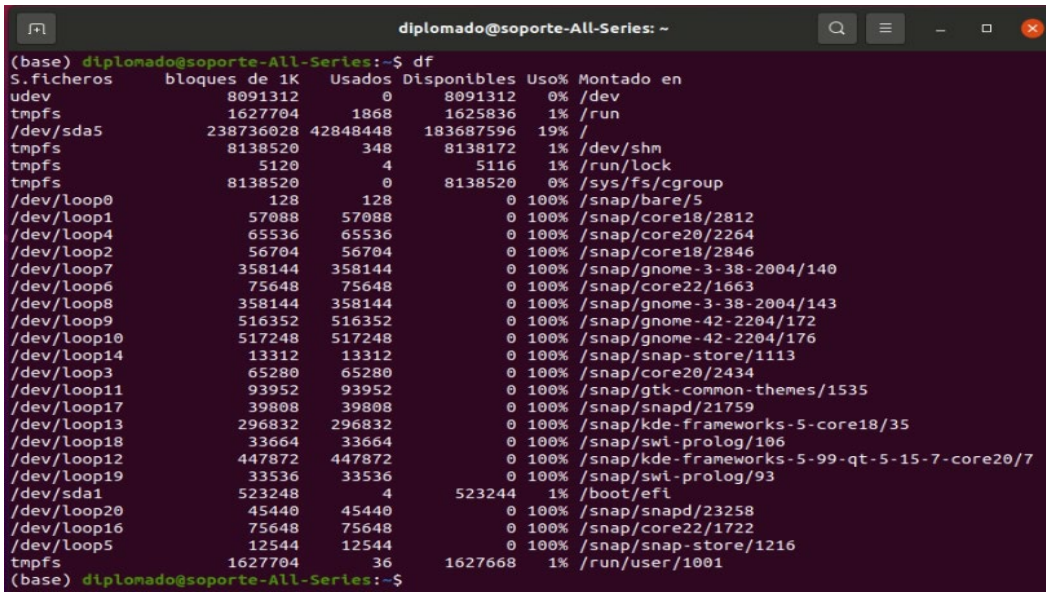
En relación a las características físicas del servidor, que se proporcionaron superiores a los requerimientos mínimos indicados por el sistema operativo, no se encontraron problemas para la atención a las peticiones procesamiento recibidas ni de memoria. En la partición de las cuentas de desarrollo (/dev/sda5), el uso de disco duro no superó el 20%. Esto se aprecia en la Figura 4.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.88

Vol. 3, Núm. 2. abril-junio 2025, págs. 42 - 44

Figura 4

Gráfico que muestra el porcentaje de uso de la partición de trabajo.



```
diplomado@soporte-All-Series: ~
(base) diplomado@soporte-All-Series:~$ df
S.ficheros     bloques de 1K   Usados Disponibles Uso% Montado en
udev                8091312        0     8091312   0% /dev
tmpfs               1627704     1868     1625836   1% /run
/dev/sda5         238736028 42848448   183687596  19% /
tmpfs               8138520      348     8138172   1% /dev/shm
tmpfs                  5120        4        5116   1% /run/lock
tmpfs               8138520        0     8138520   0% /sys/fs/cgroup
/dev/loop0               128      128           0 100% /snap/bare/5
/dev/loop1             57088    57088           0 100% /snap/core18/2812
/dev/loop4             65536    65536           0 100% /snap/core20/2264
/dev/loop2             56704    56704           0 100% /snap/core18/2846
/dev/loop7            358144   358144           0 100% /snap/gnome-3-38-2004/140
/dev/loop6             75648    75648           0 100% /snap/core22/1663
/dev/loop8            358144   358144           0 100% /snap/gnome-3-38-2004/143
/dev/loop9            516352   516352           0 100% /snap/gnome-42-2204/172
/dev/loop10           517248   517248           0 100% /snap/gnome-42-2204/176
/dev/loop14            13312    13312           0 100% /snap/snap-store/1113
/dev/loop3             65280    65280           0 100% /snap/core20/2434
/dev/loop11            93952    93952           0 100% /snap/gtk-common-themes/1535
/dev/loop17            39808    39808           0 100% /snap/snapd/21759
/dev/loop13           296832   296832           0 100% /snap/kde-frameworks-5-core18/35
/dev/loop18            33664    33664           0 100% /snap/swi-prolog/106
/dev/loop12           447872   447872           0 100% /snap/kde-frameworks-5-99-qt-5-15-7-core20/7
/dev/loop19            33536    33536           0 100% /snap/swi-prolog/93
/dev/sda1             523248        4      523244   1% /boot/efi
/dev/loop20            45440    45440           0 100% /snap/snapd/23258
/dev/loop16            75648    75648           0 100% /snap/core22/1722
/dev/loop5             12544    12544           0 100% /snap/snap-store/1216
tmpfs               1627704       36     1627668   1% /run/user/1001
(base) diplomado@soporte-All-Series:~$
```

En la Tabla 1, se presenta la distribución del uso de disco duro por cada cuenta de trabajo.

Tabla 1

Distribución de uso de disco duro.

Cuenta	Uso de disco duro (MB)
/home/aaron/	50
/home/adrian/	27
/home/Alejandro/	27
/home/citlali/	106
/home/edgar/	258
/home/gerardo/	74
/home/jose/	5
/home/juan/	46
/home/marco/	8
/home/michell/	3
/home/ruben/	14
/home/sonia/	221
/home/vania/	373
/home/victor	133

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.88

Vol. 3, Núm. 2. abril-junio 2025, págs. 43 - 44

3.1 AUTOMATIZACIÓN DEL PROCESO

Proporcionar servidores como infraestructura de desarrollo para diferentes proyectos implica un riesgo importante de que se produzcan errores que pueden derivar en la inhabilitación del servidor, principalmente porque, al entregar credenciales de administración de los distintos componentes de software, existe la posibilidad de que se realicen cambios que afecten la operación. Por lo anterior, es importante, en primera instancia, recomendar que cada usuario realice respaldos de sus proyectos periódicamente y, en segunda instancia, contar con un medio automatizado que permita la rápida recuperación del servidor como la creación de máquinas virtuales del sistema una vez que este ha sido configurado y probado. Esto permitirá, en caso de ser necesario, poner en operación rápidamente un nuevo sistema.

Un punto importante a considerar es que, si bien las máquinas virtuales representan grandes ventajas como la facilidad de exportación, también tienen algunas desventajas como la disminución del rendimiento en comparación con un sistema en un equipo físico, pero, sin duda, son una herramienta útil para su implementación en entornos de desarrollo.

4. CONCLUSIONES

Proporcionar servicios de infraestructura de cómputo para programas de capacitación en tecnologías de información y comunicación representa un reto por la acelerada evolución de las herramientas de software, lo que implica estudiar de manera permanente los cambios entre sus versiones, así como los requerimientos de hardware.

Un aspecto relevante es identificar claramente los requerimientos técnicos necesarios a través de una estrecha comunicación con los coordinadores académicos, lo que permite realizar la selección adecuada de los recursos para realizar una primera implementación, en este caso en particular, de un servidor de propósito específico y, a partir de ese punto, realizar las pruebas de operación suficientes. Otro aspecto a destacar, es considerar el soporte técnico adecuado durante todo el proceso para resolver las incidencias que se presenten tanto para mantener la funcionalidad como la operatividad del equipo y de los servicios.

AGRADECIMIENTOS

Al personal de la Coordinación de Infraestructura de la Dirección de Sistemas y Servicios Institucionales de la DGTIC y al personal de la Dirección de Docencia en el Centro Mascarones, por el trabajo conjunto que nos permite continuar realizando proyectos que impactan en el desarrollo profesional de los individuos que confían su actualización académica y profesional en los programas de capacitación continua de la DGTIC.

REFERENCIAS

Apache Software Foundation (2024). *Directorios web por usuario*. Recuperado el 11 de noviembre de 2024, de https://httpd.apache.org/docs/2.4/es/howto/public_html.html

Balkhi, S. (2024). *Cómo determinar el tamaño ideal de un servidor web para su sitio web.* Recuperado el 9 de noviembre de 2024, de https://www.wpbeginner.com/es/beginners-guide/how-to-determine-

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.88

Vol. 3, Núm. 2. abril-junio 2025, págs. 44 - 44

the-ideal-size-of-a-web-server-for-your-website/

ComputerWeekly (2025). *Definición. Servidor Web*. Recuperado el 12 de marzo de 2025, de https://www.computerweekly.com/es/definicion/Servidor-web#:~:text=Adem%C3%A1s%20de%20HTTP%2C%20los%20servidores,de%20archivos%20y%20el%20almacenamiento.

DGTIC (2025). *Funciones*. Recuperado el 3 de marzo de 2025, de https://www.tic.unam.mx/funciones/

Mendoza, A. J., Vera, M. J., Quimis, W. R., Chiriboga, I. A., Cevallos, B. M., & Encarnación, M. M. (2024). Implementación de Tecnologías de la Información y la Comunicación en la Educación: Diagnóstico de problemas y propuestas de intervención: Implementation of Information and Communication Technologies in Education: Diagnosis of problems and proposals for intervention. *LATAM Revista Latinoamericana De Ciencias Sociales Y Humanidades*, 5(4), 859 – 867. https://doi.org/10.56712/latam.v5i4.2298

Ramírez, M. (2019). *Análisis comparativo de rendimiento a servidores Web de distribución libre utilizando Apache benchmark,* 13. Ecuador: UTMACH, Facultad de Ingeniería Civil. Tesis de licenciatura en Ingeniería de Sistemas.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.96

Vol. 3, Núm. 2. abril-junio 2025, págs. 45 - 53

Utilización de Dockers para desplegar en producción sistemas de información heredados

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

Información del reporte:

Licencia Creative Commons

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

Para citar este reporte técnico:

Chamú Arias, J. O. (2025). Utilización de Dockers para desplegar en producción sistemas de información heredados. *Cuadernos Técnicos Universitarios de la DGTIC*, 3 (2) páginas (45 - 53).

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.96

Resumen

Se aborda el análisis de un caso particular en el que se requería extraer un sistema de información heredado que estaba alojado dentro de una computadora personal y transferirlo a un ambiente de trabajo en producción, en máquinas virtuales. Se presentaron incompatibilidades a nivel de sistema operativo, versiones de librerías y lenguajes utilizados ya que, al estar en un equipo personal, esos elementos no se actualizaban, dando como resultado versiones de software antiguas y sin soporte. Se intentó la virtualización sobre Rocky Linux versión 9, pero, al ser reciente el sistema operativo, no incluía el lenguaje de programación PHP 7.4 en el que estaba desarrollado el sistema de información, lo que ocasionó una reacción en cadena en los demás componentes que dependían de dicho módulo, ocasionando que la virtualización no fuera suficiente para resolver la situación; además, utilizar versiones anteriores no era la mejor opción por riesgos de seguridad y falta de soporte. Por ello, se exploró la utilización de la plataforma de contenedores "Dockers", que es un entorno ejecutable que se monta sobre un sistema operativo principal (Rocky Linux) y que reúne los elementos necesarios para poner en operación una aplicación o un sistema de información. Sobre Docker, se montó el sistema operativo original (Debian Linux versión 11) del sistema de información, así como el paquete de las librerías y herramientas necesarias para el funcionamiento de la aplicación. Al estar encapsulado, permitió que el

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.96

Vol. 3, Núm. 2. abril-junio 2025, págs. 46 - 53

peso de la seguridad se transfiriera al sistema operativo principal y que se pudieran tener los diferentes ambientes de trabajo contenerizados[1]: desarrollo, pruebas y producción dentro de la misma máquina virtual; se ganó tiempo para actualizar la información y facilitar la ejecución de pruebas funcionales y no funcionales al permitir la copia del contenedor principal.

Palabras clave:

Docker, compatibilidad, desarrollo, pruebas, producción, ambiente de trabajo, contenedor, Linux, máquinas virtuales.

Abstract

This report addresses a specific case in which a legacy information system hosted on a personal computer needed to be extracted and transferred to a production environment using virtual machines. Since the operating system, library versions, and languages used were not updated on a personal computer, some incompatibilities were found related to these components, resulting in outdated and unsupported software versions. Virtualization on Rocky Linux version 9 was attempted, but, since the operating system was new, it did not include the PHP 7.4 programming language used to develop the information system. This caused a chain reaction in the other components that depended on that module, getting to an insufficient level of virtualization to solve the problem. Using older versions was not the best option due to security risks and lack of support. Therefore, the use of the "Dockers" container platform was explored. This is an executable environment that runs on a main operating system (Rocky Linux) and brings together the necessary elements to run an application or information system. The original operating system (Debian Linux version 11) of the information system was installed on Docker, along with the package of libraries and tools necessary for the application's operation. Encapsulation allowed the security burden to be transferred to the main operating system and allowed the different work environments to be containerized: development, testing, and production within the same virtual machine. This saved time for updating information and facilitated the execution of functional and non-functional tests by allowing the main container to be copied.

Keywords:

Docker, compatibility, development, testing, production, work environment, container, Linux, virtual machines.

1. INTRODUCCIÓN

Durante los procesos de desarrollo, pruebas, liberación a producción y mantenimiento de las aplicaciones, los grupos de trabajo se enfrentan a problemas de compatibilidad entre versiones del sistema operativo y de las herramientas utilizadas, así como a las diferencias en las configuraciones de cada uno de los ambientes de trabajo como diferentes configuraciones de seguridad requeridas o el sincronizado de configuraciones de las herramientas en cada uno.

1 La contenerización se refiere al empaquetado de código de software con las bibliotecas del sistema operativo y las dependencias necesarias para ejecutar dicho código, a fin de crear un único ejecutable ligero, llamado contenedor, que se ejecuta en cualquier infraestructura.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.96 Vol. 3, Núm. 2. abril-junio 2025, págs. 47 - 53

Por ejemplo, al tener que preparar los ambientes de desarrollo para aplicaciones que utilizan la versión de PHP 7.4, se evidenció que éstas ya no cuentan con soporte en Rocky Linux 9.x y, al ser un requerimiento para el funcionamiento de la aplicación, no podrían actualizarse a la versión de PHP más reciente, por lo que se tuvo que explorar la utilización de plataformas tecnológicas que permitieran el uso de la versión de PHP con versiones anteriores del sistema operativo, sin comprometer la seguridad.

Al utilizar los Dockers como una plataforma tecnológica que apoyara el desarrollo de aplicaciones, hubo resistencia al cambio por parte de algunos desarrolladores al tener que modificar la forma de trabajo respecto al despliegue de las aplicaciones, además de introducir la curva de aprendizaje que se tiene que enfrentar al comprender el uso de contenedores y de los comandos que se tienen que utilizar para su gestión.

Por lo anterior, se comparte la experiencia a través de este reporte técnico, cuyo objetivo es mostrar el uso de Docker como una plataforma tecnológica viable en el despliegue de aplicaciones nuevas o heredadas, que ofrece portabilidad, escalabilidad y una mejor eficiencia en el uso de los recursos utilizados, además de permitir agilizar el desarrollo y liberación de las mismas al quedar empaquetadas con sus dependencias en contenedores portátiles.

2. DESARROLLO TÉCNICO

2.1 ANTECEDENTES

Actualmente, algunos de los sistemas operativos Linux han tenido un cambio en el modelo de licenciamiento por parte de los dueños de los derechos, o bien, son fusionados a otros con licenciamiento comercial, lo que obliga a cambiar los desarrollos o los sistemas de información a otras plataformas. Tal es el caso de CentOS, que era distribuido por la empresa Red Hat como software libre y de código abierto, pero que fue adquirida por IBM, la cual cambió el modo de distribución.

El tema que se presenta en este reporte técnico deriva de los cambios de licencia que han afectado a los sistemas operativos y, por consiguiente, al desarrollo de las aplicaciones. Para evitar el riesgo de quedarse sin soporte, se buscaron opciones diferentes a CentOS, entre ellas: Ubuntu, FreeBSD, Debian, Rocky y AlmaLinux. Tener múltiples opciones trae consigo complicaciones de incompatibilidad en los ambientes de trabajo, los cuales, en ocasiones, se colocan en equipos de cómputo personales, trayendo como consecuencia que la aplicación funcione solamente en su ambiente de trabajo local debido a las diferentes versiones de librerías, plataformas y sistemas operativos que se usan, lo que afecta la puesta en producción.

Para subsanar las situaciones antes mencionadas, se evaluaron algunas características que se muestran a continuación:

Máquinas virtuales

Las máquinas virtuales presentan varias ventajas, entre las que destaca que alojan un sistema operativo completo con manejo de errores y reporte de logs, lo que permite la creación de una plataforma completa. Además, el hardware completo es emulado y así se facilita la creación de tantas instancias como se requieran; además, pueden ser transferidas a diferentes equipos, siempre y cuando trabajen Linux con QEMU/KVM.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.96 **Vol. 3, Núm. 2.** abril-junio 2025, págs. 48 - 53

Sin embargo, también tienen desventajas: reservan recursos exclusivamente para el funcionamiento del sistema operativo huésped, como CPU, memoria RAM y disco duro; además, se requiere un proceso de conversión de formato de disco duro en caso de que un elemento del equipo tenga otro virtualizador; consumen una cantidad considerable de recursos de red para poder compartir la máquina entre diferentes equipos, lo cual requiere reservar espacio para emular el disco duro; y, asimismo, es más complejo controlar las versiones de las diferentes máquinas y equipos en la nube o en la empresa.

Dockers

La utilización de Dockers presenta varias ventajas, entre las que destaca que ésta ayuda a homologar los ambientes de trabajo de desarrollo y mejora la precisión del control de versiones; además, ofrece mayor comodidad a los equipos de desarrollo para emplear el sistema operativo que mejor les funcione; La creación, actualización y borrado de las imágenes es más ágil y centralizada, lo que permite abordar el problema de la incompatibilidad: eliminar los efectos de las diferencias entre los equipos y mitigar los riesgos al colocar el sistema en producción; también ofrece características que permiten crear múltiples instancias de los contenedores, esto es, incrementar o decrementar su número según se requiera.

En caso de cambios de licenciamiento o de políticas del proveedor del software, la comunidad de desarrollo cuenta con varias alternativas de plataformas compatibles que ofrecen el uso de contenedores. De acuerdo con Bruno y Bruno (2020), entre otras opciones, destacan RKT, PodMan, Singularity, Linux Containers – LXC y OpenVz. El Docker es utilizado por grandes compañías como ING, PayPal, ADP y Spotify (Novoseltseva, 2023). Por otro lado, presenta un comportamiento similar a Java, ya que es multiplataforma y consume pocos recursos. "Para ejecutar un contenedor no se necesita hypervisor porque no se ejecuta un sistema operativo invitado y no hay que simular hardware" (Gallego & Chico Guzmán, 2017).

Los contenedores son elementos ligeros en el uso de recursos del sistema operativo y su despliegue es más ágil una vez creada la imagen, es decir, la plantilla con las instrucciones que definen un contenedor. Además, permite desplegar ambientes heredados y, de acuerdo con Dock, M. (2025) "no tiene costo para organizaciones pequeñas o medianas con menos de 250 empleados y que generen ingresos menores a 10 millones de dólares anuales. Para organizaciones mayores, implica usar algún plan empresarial". En términos de seguridad, al estar encapsulado en un sistema operativo huésped, se delegan las actualizaciones y medidas de seguridad perimetrales como son el cierre de puertos, las actualizaciones de librerías, la segmentación de redes internas, el uso de proxy para administrar el certificado o el uso de un *firewall* de aplicaciones web; la escalabilidad que se tiene con esta tecnología es más práctica al aumentar el número de contenedores en lugar de crear más máquinas virtuales. Al no tener que emular el hardware con todos sus componentes, el rendimiento es mucho más ligero, pues se consumen menos recursos.

Sin embargo, el uso de Docker también presenta desventajas, como la curva de aprendizaje para su utilización, que es más alta, ya que requiere conocimientos básicos de servidores y *firewalls* por parte de los equipos de desarrollo. Los contenedores, al ser sistemas recortados, pueden apagarse al generarse un error, debido a que no tienen todos los elementos para generar un *core dump*[2]; aunado a esto, modificar un archivo de configuración es una tarea que requiere mayor experiencia técnica, pues implica

2 Core dump: También conocido como volcado de memoria, es una copia del estado de la memoria que se crea cuando se produce un error o fallo de un programa o aplicación.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.96

Vol. 3, Núm. 2. abril-junio 2025, págs. 49 - 53

muchos pasos para recrear el contenedor con las nuevas características, lo cual implica borrarlo, crearlo, respaldarlo, restaurarlo, ajustar puertos y, finalmente, verificar que todo funcione adecuadamente.

La protección de la seguridad es más compleja ya que, dependiendo de cómo se crea un contenedor, hay que cerrar o abrir múltiples puertos; también se debe considerar lo que se requiere para que, en caso de intrusión, ésta no absorba todos los recursos del equipo huésped, por mencionar algunos.

A partir de lo anterior, se presenta la Tabla 1 para visualizar las ventajas de seleccionar Dockers en lugar de máquinas virtuales:

Tabla 1

Comparativa entre un Docker y una máquina virtual

Tecnología	Emulación de hardware	Ejecución de múltiples sistemas operativos en un mismo equipo físico	Separar componentes de hardware	Ejecución de una aplicación en ambiente productivo	Ejecutar en diferentes tipos de hardware y sistemas operativos	Capacidad de Escalamiento
Máquina Virtual	Sí	Sí	Sí	Sí	Sí	Sí
Docker	No	No	Sí	Sí	Sí	Sí

Es importante destacar que Docker se ejecuta en diversas plataformas de hardware sin emular todo el componente físico, dando como resultado menor consumo de recursos.

Asimismo, si se cambian las reglas del *firewall*, éstas se borran y hay que reiniciar los servicios y contenedores. Si se requiere un cambio en la imagen del contenedor, es probable que se tenga que reconstruir. La compatibilidad de las versiones antiguas está atada a la versión del cliente con la que se creó y sobre la que se requiere ejecutar. Para evitar problemas como los mencionados, es necesario un orquestador, ya sea Kubernetes, Docker Swarm, entre otros.

Docker, la herramienta seleccionada, funciona en los sistemas operativos más populares y utilizados en la UNAM. Racero & Som (2022) mencionan que "los contenedores son un paquete de elementos que permiten ejecutar una aplicación determinada en cualquier sistema operativo", ya que permiten crear un ambiente desde un repositorio central, donde se puede controlar y administrar el control de las versiones. Al estar centralizado, se puede tener un control más preciso de los cambios en los ambientes de los diferentes equipos y procesos. Además, esta herramienta permite que los equipos de trabajo usen los sistemas operativos que tienen en sus equipos personales, por ejemplo: personal de desarrollo que trabaje en Windows, personal de aseguramiento de calidad que ocupe MacOS, expertos en seguridad que usen Ubuntu y los responsables del ambiente de producción que trabajen en Debian. Dicho de otra manera, el motor y la imagen funcionan siempre y cuando se encuentren dentro de la lista de los sistemas operativos más usados por las comunidades de desarrollo.

Se utiliza Dockers cuando se requieren despliegues de ambientes en poco tiempo, independientes de plataforma, con facilidad de escalamiento tanto vertical como horizontalmente. Por ello, la aplicación debe estar preparada para funcionar en dicho entorno, por ejemplo, si es una aplicación que guarda el

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.96 **Vol. 3, Núm. 2.** abril-junio 2025, págs. 50 - 53

estado del usuario, debe almacenarlo en base de datos y las credenciales de conexión a los recursos deben ser configurados con variables de ambiente para reducir o evitar modificaciones a los contenedores.

Sobre los ambientes de trabajo, Tirado (2017) destaca a "Docker y docker-compose, herramienta que nos ayuda a gestionar aplicaciones multicontenedor, en este caso, nos ayudan a la administración de los ambientes de desarrollo, facilitando la forma de trabajo y minimizando los recursos que necesitamos para la estandarización de los ambientes de trabajo".

Como plataforma, ayuda a poner en marcha sistemas monolíticos, y permite un mejor aprovechamiento de los recursos de hardware de los servidores. Conforme pasa el tiempo, se va adoptando cada vez más: un estudio de 2018 arroja que hay "alrededor de 700 millones de contenedores en uso" (MuyComputerPro, 2018).

3. IMPLEMENTACIÓN

Para el uso de Dockers dentro de los ambientes de trabajo (desarrollo, pruebas, pre-producción o producción), se requiere de la instalación en una máquina virtual o servidor físico y del desarrollo de las siguientes etapas:

- Creación de la imagen del Docker
- Ejecución del contenedor
- Publicación y distribución de la imagen

3.1 CREACIÓN DE LA IMAGEN DEL DOCKER

Para la creación de la imagen del Docker, se eligió como sistema operativo Rocky Linux 9, que es una versión libre compatible con Red Hat del cual derivan diversas distribuciones de Linux y una alternativa a CentOS, el cual fue descontinuado y que se utilizaba en los desarrollos dentro del área donde trabajo. Sin embargo, también puede utilizarse en otras distribuciones de Linux como son Ubuntu y Debian.

Se instaló Rocky Linux 9 en una máquina virtual, siguiendo los pasos predeterminados del instalador, sin modificaciones ni configuraciones adicionales. Para la creación de la imagen del Docker, es necesario instalar algunas librerías y clientes (Docker Engine) que son descargados del Repositorio de Docker[3] de acuerdo con el sistema operativo utilizado.

A continuación, se debe configurar el ambiente del contenedor en el sistema operativo para otorgar privilegios a la cuenta de usuario que va a gestionar el Docker, y se abren los puertos de comunicación que utilizará el contenedor de acuerdo a los requerimientos de la aplicación a desarrollar o que residirá en él.

Posteriormente se creará el contenedor a partir de una imagen de prueba que se encuentra en el Repositorio de Docker, tomando como base el archivo llamado Dockerfile, el cual contiene la receta para crear la imagen como se muestra a continuación (Creación de Imágenes A Partir de un Dockerfile, s. f.):

```
FROM debian:buster-slim
```

3 Repositorio de Docker: https://docs.docker.com/engine/

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.96 Vol. 3, Núm. 2. abril-junio 2025, págs. 51 - 53

```
MAINTAINER José Domingo Muñoz "josedom24@gmail.com"
RUN apt-get update  && apt-get install -y  apache2
COPY index.html /var/www/html/
CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

La línea "MAINTAINER" puede ser modificada u omitida por quien genere la imagen del Docker.

Para ejecutar el archivo Dockerfile se utiliza el comando:

```
docker build -t usuario/Prueba:1.0
```

con el cual se leen las instrucciones del archivo. Si no encuentra errores, crea la imagen del contenedor.

Se puede verificar su creación a través de:

```
docker image ls
```

3.2 EJECUCIÓN DEL CONTENEDOR

Una vez creado el contenedor, se tiene que poner en funcionamiento a través del siguiente comando:

```
docker run -d  --name  incripcionesweb1    usuario/Prueba:1.0
```

Para verificar que se encuentre en ejecución el Docker, se puede verificar con el comando:

```
docker ps
```

Éste mostrará una lista de los contenedores que se encuentran en ejecución.

3.3 PUBLICACIÓN Y DISTRIBUCIÓN DE LA IMAGEN

Una vez realizado el paso anterior, se puede publicar el resultado en un repositorio privado como GitHub, o bien, en Docker Hub, que es un sitio público o privado donde se concentran las imágenes para compartirlas desde la nube, el cual es "el repositorio en la nube de Docker donde se encuentran todas las imágenes de los contenedores" (Ponsico Martín, 2017). Por ejemplo, la liga https://hub.docker.com/r/jchamu/php8-usuario3 contiene una imagen para ejecutar ambientes de PHP y es accesible para todas las áreas de trabajo. De igual forma, cuenta con características de versionado, descripción de contenido, instrucciones para crear las imágenes e instrucciones para su puesta en operación.

La implementación realizada se usó en el portal web de un proyecto a cargo de la DGTIC, el cual tenía las limitantes de rescatar un sistema de información con las versiones exactas, que para este caso eran CouchDB 2.3.1, Directus 9.12.1, PHP 7.4 y Mysql 8.0, con restricciones de tiempo. Se trabajó en dos ambientes (desarrollo y producción), la meta del tiempo se logró y todas las versiones del software funcionaron correctamente con el uso de dicha tecnología, lo que permitió simplificar los retos técnicos derivados de que la versión "PHP 7.4" ya no tenía soporte en Rocky Linux 9, que era la versión mínima que se tenía en el ambiente de producción. La forma de subsanar estas restricciones y evitar más incompatibilidades fue el uso de los contenedores. La seguridad se distribuyó teniendo un sistema operativo reciente con todas

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.96 **Vol. 3, Núm. 2.** abril-junio 2025, págs. 52 - 53

las actualizaciones, como es el caso de Rocky 9 y en el contenedor incorporando un *firewall* web en el Apache como lo es mod_security[4].

4. RESULTADOS

El uso de Docker, en el caso mencionado anteriormente, permitió el funcionamiento de una aplicación heredada que requería versiones anteriores, manteniendo la seguridad a través del sistema operativo de la máquina virtual que lo aloja y permitiendo su traslado a un ambiente de producción sin modificaciones, es decir, sin tener que agregar o quitar librerías, así como cambiar valores en variables de ambiente.

Asimismo, se lograron reducir los tiempos de uno a dos días a una hora por ambiente, debido a que se cuenta con un ambiente estandarizado en el contenedor que se podría portar sin modificaciones a desarrollo, pruebas y producción, es decir, al no tener que estar instalando y configurando cada tipo de ambiente, ajustando variables y permisos en directorios, validar versiones de librerías, entre otras cosas.

Lo anterior simplifica las labores de administración de las máquinas virtuales, porque agiliza la atención de solicitudes de ambientes de desarrollo, pruebas y producción, evitando tener solicitudes de ambientes de aplicaciones encoladas por más de dos días.

De igual manera, el uso de Dockers ayudó a reducir las incompatibilidades que se tenían entre los diferentes ambientes de trabajo, a centralizar las imágenes del sistema en un repositorio, controlar los cambios al tener versionado de dichas imágenes y abrió las puertas a un nuevo tipo de despliegue de aplicaciones distribuidas. Hoy, se cuenta con una nueva opción que no implica solamente máquinas virtuales y que ha sido una experiencia favorable en casi un año de uso de los contenedores. En contraste, en el modelo de trabajo anterior, se tenían problemas de compatibilidad entre ambientes de trabajo en todos los proyectos de desarrollo de software.

Esto permitió mantener el control del versionado del contenedor al momento de entregarlo a los diferentes equipos de trabajo, donde participan al menos 15 personas con equipos de cómputo mixtos, sin generar conflictos. Esta característica no está limitada a este número de participantes, ya que puede ser mayor.

5. CONCLUSIONES

Las bondades del despliegue ágil de esta tecnología han sido aplicadas de manera exitosa en otras aplicaciones de la Universidad, a cargo de la DGTIC. Si bien el uso de Docker y sus bondades de encapsulado para controlar diferentes ambientes de trabajo no es nuevo, ha aportado mejoras significativas, reduciendo los tiempos de despliegue y la posibilidad de incompatibilidades.

Respecto al uso de Dockers, se encontró que, si no se tienen conocimientos de sistemas operativos, configuración de la red y nociones básicas de la operación de *firewalls*, se enfrenta una curva de aprendizaje más alta.

4 Proyecto de mod_security: https://modsecurity.org/

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.96 **Vol. 3, Núm. 2.** abril-junio 2025, págs. 53 - 53

Los Dockers son una herramienta útil para agilizar la labor en la administración de los ambientes de trabajo de desarrollo, pruebas, pre-producción y producción, que permite reducir tiempos en la implementación, optimizar recursos hardware al admitir la ejecución de múltiples instancias de una aplicación en diferentes contenedores y permitir que las aplicaciones puedan ejecutarse en diferentes sistemas operativos y entornos sin modificaciones.

De igual forma, proporcionan el aislamiento de recursos entre contenedores, lo que permite evitar que un contenedor afecte a otros en su operación al tener diferentes sistemas operativos, versiones de herramientas y librerías o configuraciones de seguridad.

Por último, los Dockers son una plataforma tecnológica que se utilizará más ampliamente a futuro, integrada con el uso de Kubernetes y de nuevas herramientas que permitan implementar aplicaciones que se centran en tareas de alto impacto, que implican la gestión de grandes cantidades de datos, que requieren un nivel de disponibilidad que permita a una aplicación seguir funcionando mientras una de sus partes se encuentra en actualización o reparación, o que requiera un esquema robusto de seguridad. De igual forma, se tendrá una mayor integración en nubes públicas o privadas o híbridas.

REFERENCIAS

Bruno, V., & Bruno, V. (2020, 30 marzo). Top 5 Alternativas a Docker. Infranetworking. https://blog.infranetworking.com/top-5-alternativas-a-docker/

Creación de imágenes a partir de un Dockerfile. (s. f.). Docker. https://iesgn.github.io/curso_docker_2021/sesion6/dockerfile.html.

Dock, M. (2025, 23 febrero). Pricing | Docker. Docker. https://www.docker.com/pricing/

Gallego, M., & Chico Guzmán, P. (2017). Seminarios OfiLibre Docker y Kubernetes. Universidad Rey Juan Carlos. https://tv.urjc.es/uploads/material/5fb81bcbd68b14ac6f8b4be2/Docker.pdf

MuyComputerPro. (2018, Julio 3). Estadísticas de uso de Docker año 2018. https://www.muycomputerpro.com/2018/07/03/adopcion-docker-estadisticas

Novoseltseva, E. (2023, 22 junio). Utilizar Docker: beneficios, estadísticas y historias de éxito | Apiumhub. Apiumhub. https://apiumhub.com/es/tech-blog-barcelona/beneficios-de-utilizar-docker/

Ponsico Martín, P. (2017). Tecnología de Contenedores Docker [Tesis de grado de Ingeniería Telemática, Universitat Politècnica de Catalunya]. https://upcommons.upc.edu/bitstream/handle/2117/113040/Degree_thesis.pdf?sequence=1

Racero, A., & Som, A. (2022, mayo). Contenedores: Iniciación a Docker y casos de uso prácticos. Oficina de Software Libre, Universidad de Granada. https://osl.ugr.es/wp-content/uploads/2022/05/Contenedores-Iniciacio%CC%81n-a-Docker-y-casos-de-uso-pra%CC%81cticos-Mayo2022.pptx.pdf

Tirado, L. M. (2017). Implementación de Docker en la gestión del entorno de desarrollo [Tesina de Ingeniería, Universidad Politécnica de Sinaloa, Programa Académico de Ingeniería en Informática]. http://repositorio.upsin.edu.mx/formatos/TesinaLiliaMariaTirado3453.pdf

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 54 - 80

Desarrollo de aplicaciones a la medida con apoyo de modelos GPT

Luz María Ramírez Romero
Dirección General de Cómputo y de Tecnologías de Información y Comunicación
Universidad Nacional Autónoma de México
luzrr@unam.mx
ORCID: 0000-0002-8867-9374

Información del reporte:

Licencia Creative Commons

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

Para citar este reporte técnico:

Ramírez Romero, L. M. (2025). Desarrollo de aplicaciones a la medida con apoyo de modelos GPT. *Cuadernos Técnicos Universitarios de la DGTIC*, 3 (2)(54 - 80).

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Resumen

Con base en la necesidad de mejorar la productividad en el desarrollo de aplicaciones a la medida, en la Dirección General de Cómputo y Tecnologías de Información y Comunicación se exploraron diferentes metodologías, técnicas y herramientas. Se revisó que los modelos GPT inciden en la mejora del aprendizaje de los *frameworks* y lenguajes de programación de los desarrolladores de aplicaciones a la medida, en aras de reducir el tiempo de desarrollo de las diferentes piezas de software que conforman dichas aplicaciones. En este reporte, se explican los elementos que se pueden integrar en el diseño del *prompt* o solicitud y cómo se puede refinar el *prompt* para que los modelos GPT brinden mejores respuestas, más fáciles de integrar a la aplicación en desarrollo y menos complejas en su lógica. Se tomó en cuenta la complejidad de los componentes con la herramienta *laravel sail insights*. Se concluyó que al aplicar los modelos GPT de Inteligencia Artificial generativa se reduce el tiempo de construcción de funciones, componentes y piezas de software, así como el número de incidencias por resolver; además, las incidencias detectadas por el equipo de aseguramiento de la calidad del software se solucionan en menor tiempo. El desarrollador con experiencia puede diseñar los mejores *prompts* para obtener las mejores respuestas de los modelos GPT, y analizar los resultados para decidir la incorporación del código generado, además de tomar en cuenta el marco de ética de los derechos de autor.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 55 - 80

Palabras clave:

Productividad, desarrollo de aplicaciones, ingeniería de prompts, modelos GPT, ingeniería de software

Abstract

Based on the need to improve productivity in custom application development, experts at Dirección General de Cómputo y de Tecnologías de Información y Comunicación explored different methodologies, techniques and tools. It was reviewed that GPT models contribute to improve the learning curve of custom application developers in programming frameworks and languages, thereby reducing the development time of the various software components that make up custom applications. This report explains the elements that can be integrated into the design of the prompt or request and how the prompt can be refined so that GPT models provide better responses that are easier to integrate into the application under development and less complex in their logic. The complexity of the components was taken into account using the Laravel Sail Insights tool. It was concluded that applying generative Artificial Intelligence GPT models reduces the construction time of functions, components and software components, as well as the number of issues to be solved. Furthermore, issues detected by the software quality assurance team are dealt with in less time. An experienced developer can design the best prompts to elicit the best responses from GPT models and analyze the results to decide whether to incorporate the generated code, while also taking into account the ethical framework of copyright issues.

Keywords:

Productivity, application development, prompt engineering, GPT models, software engineering.

1. INTRODUCCIÓN

Las aplicaciones a la medida son algunas de las herramientas que utilizan las organizaciones para acercarse al logro de sus objetivos. El proceso de desarrollo de dichas aplicaciones requiere la inversión de recursos humanos y financieros, así como el tiempo.

En la Dirección General de Cómputo y Tecnologías de Información y Comunicación (DGTIC), se construyen aplicaciones para apoyar las actividades académico-administrativas de entidades y dependencias universitarias que lo solicitan a la Dirección General; es importante revisar e integrar métodos, técnicas y herramientas que mejoren la eficiencia en el desarrollo de esta tarea. Por lo tanto, una de las líneas de análisis es el uso de los modelos de inteligencia artificial generativa para este propósito, en los ámbitos de: reducción de incidencias, disminución del tiempo de desarrollo y del tiempo de corrección de incidencias.

El objetivo de este reporte técnico es compartir cómo se ha utilizado el modelo GPT-4 de inteligencia artificial generativa a través del diseño de *prompts* (solicitudes o instrucciones para los modelos GPT), a fin de disminuir el tiempo de desarrollo de software y reducir las incidencias en los sistemas de información, así como acortar el tiempo de corrección de defectos. En específico, se utilizó el diseño de *prompts* en el

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 56 - 80

Sistema de Censo de Tecnologías de Información y Comunicaciones, durante el periodo de abril de 2024 a febrero de 2025.

2. ANTECEDENTES

El concepto de Inteligencia Artificial (IA) surge en 1950, con el artículo escrito por Alan Turing denominado *Computing Machinery and Intelligence* (Computadoras e inteligencia), en el que propuso que el procesamiento de las computadoras fuera tan avanzado y tan similar a la forma de procesamiento del cerebro humano, que se pudieran confundir las respuestas de la computadora con las que daría una persona. A dicho planteamiento, se le conoce como la *Prueba de Turing* (Robisco, 2024).

La IA tuvo varios desarrollos y aportes, sin embargo, este análisis se centrará en el modelo denominado GPT, desarrollado en la empresa *OpenAI*, que se trabajó a partir de 2018. El modelo GPT ha sido diseñado para generar textos, imágenes, traducciones lingüísticas, conversaciones, resúmenes y código de programación a partir de incorporar técnicas como el procesamiento de lenguaje natural (PLN) y de *Machine Learning* (aprendizaje automático), entre otras (Brown et al., 2020).

El procesamiento de lenguaje natural permite introducir en el modelo un *prompt*, solicitud o instrucción en lenguaje natural y que el modelo pueda transformarlo en *tokens* o representaciones numéricas que pueda interpretar (Martí et al., 2002).

La técnica *Machine Learning* habilita al modelo para recibir o buscar en grandes fuentes de datos (*Big Data*[1]), o en diferentes fuentes o sitios especializados en Internet, y detectar patrones y/o reglas que le permitirán "aprender", de tal forma que el modelo puede generar respuestas sin haber sido programado específicamente para ello, basado en los patrones que reconoció (Ramana et al., 2024).

Además, se consultó el estudio *Stackoverflow Developer Survey* 2024, como referencia para saber qué tanto se están utilizando los modelos de IA GPT en el desarrollo de software a la medida. La encuesta indica que el 76% de desarrolladores a nivel mundial ya están utilizando la IA, o planean hacerlo, con aplicaciones en el desarrollo de aplicaciones a la medida (*2024 Stack Overflow Developer Survey*, s/f).

Otra estadística que muestra la encuesta es que el 82.1% de los desarrolladores están usando ChatGPT, seguidos por el 41.2% de programadores que están utilizando Microsoft Copilot.

2.1 METODOLOGÍA

Se revisaron recomendaciones y buenas prácticas de ingeniería y diseño de *prompts*, mismas que se probaron y se aplicaron en el desarrollo del *Sistema de Censo de Tecnologías de Información y Comunicaciones* en la DGTIC. A partir de los resultados obtenidos, se analizaron los *prompts* que producían el código más preciso. También se fue refinando el código a partir de la mejora de los *prompts*, además de que se cuidó el aspecto ético en este proceso. Ver Figura 1.

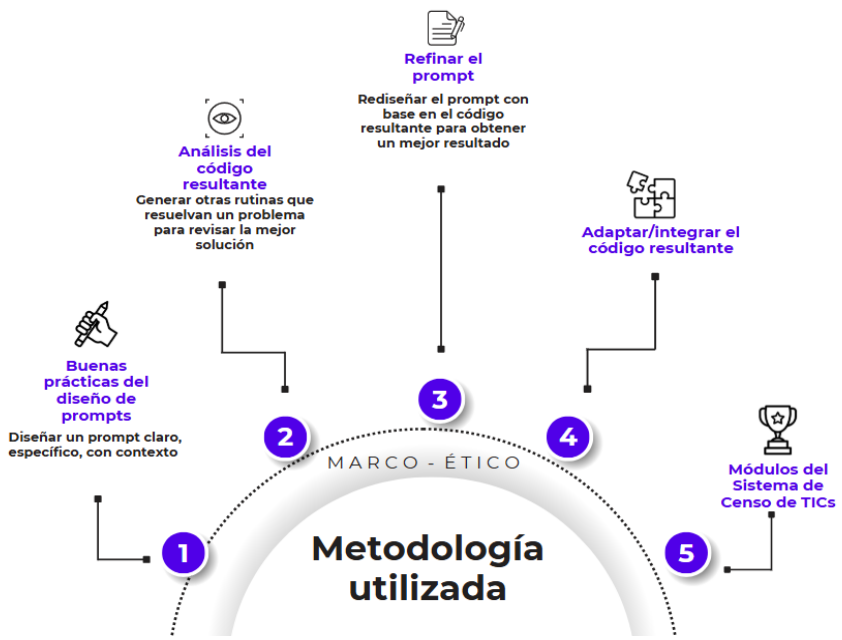
1 *Big Data*: Término que se utiliza para referirse a grandes volúmenes de datos generados cada segundo, que deben almacenarse y procesarse con técnicas diferentes a las tradicionales como las bases de datos SQL, ya que su procesamiento debe ser en tiempo real; además, los datos pueden estar estructurados, no-estructurados o puede haber combinación de ambos.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 57 - 80

Figura 1

Representación gráfica de la metodología utilizada



Además de efectuarse la corrección de incidencias de los componentes desarrollados, el trabajo de diseño de *prompts* se aplicó en los siguientes componentes para el desarrollo del sistema:

- Registro, edición y borrado de equipos periféricos de digitalización e impresión.
- Consulta tabular, registro, edición y eliminación de equipos de red con contrato de mantenimiento, equipos de telefonía, multilíneas/conmutadores, fibra óptica, puntos Ethernet y servicios de red, así como la integración de nuevas categorías de telefonía y Ethernet.
- Registro y edición de la cantidad de equipo de cómputo con mantenimiento preventivo y correctivo, en la categoría de equipos de cómputo.
- Componentes para registro de la terminación del censo, con la generación del acuse de recibo y la notificación al usuario correspondiente (Responsable TIC).

Las recomendaciones y prácticas se probaron para el desarrollo de código sobre la base tecnológica de los *frameworks* Laravel 11, Livewire 3 y PHP Orientado a Objetos versión 9 y para la construcción de sentencias SQL para el manejador de base de datos PostgreSQL, que es la tecnología que se utilizó en la implementación del sistema en mención.

Se emplearon dos modelos de IA: Microsoft Copilot y ChatGPT, ambos basados en el modelo GPT-4. Se seleccionaron porque han sido entrenados utilizando una gran base mundial de código fuente (GitHub), tomando en cuenta sólo el código abierto, respetando las licencias y derechos de autor de los programadores (iac, 2024).

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 58 - 80

2.2 APLICACIÓN DEL MODELO GPT-4 DE IA EN DIFERENTES ASPECTOS DEL DESARROLLO DE SOFTWARE A LA MEDIDA

Se aplicó el modelo de IA en el desarrollo del *Sistema de Censo de Tecnologías de Información y Comunicaciones*, durante el periodo de abril de 2024 a febrero de 2025. Se obtuvieron reducciones en el número de incidencias (ver Tabla 1), así como mejoras tanto en los tiempos de desarrollo de componentes como en la corrección de incidencias del código.

Tabla 1

Incidencias registradas en sistemas desarrollados del 2023 al 2024[2]

Sistema	Incidencias con causa-raíz: desarrollo-código
Censo * (Usando *prompts* y modelo GPT)	44
Sistema 2 (Cor)	55
Sistema 3 (JG)	54

El modelo GPT-4 fue aplicado en las siguientes áreas del desarrollo del sistema en mención:

1. Corrección de errores: Al desarrollar aplicaciones, los programadores pueden introducir defectos en el código. Los más sencillos de corregir por el modelo de IA GPT-4 son los errores de sintaxis, basta con copiar la sentencia que ha generado el error y obtendremos el código modificado. Si brindamos más contexto al modelo de IA, la respuesta para corregir el error será más acertada. Ver ejemplo en el Anexo 1.

2. Generación de código para atender rutinas comunes: Resulta conveniente contar con un ayudante tecnológico que genere funciones que atiendan problemáticas comunes; así, la mente del programador estará más ocupada en resolver problemas más complejos. Ver ejemplo en el Anexo 2.

3. Generación de consultas SQL (*Structured Query Language*): Los modelos de datos de sistemas reales comúnmente conllevan una fuerte complejidad para obtener cierta información que implique relacionar varias entidades, agrupar varios campos, filtrar la información a partir de algún resultado parcial (*subqueries*) y procesar los datos para dar algún formato al resultado de la consulta. El modelo de IA es de gran apoyo para generar sentencias SQL complejas. Ver ejemplo en el Anexo 3.

4. Búsqueda de alternativas de código: En el ámbito de desarrollo de aplicaciones, siempre es importante llegar a la implementación idónea en eficiencia del uso de recursos de cómputo y de red. Por lo anterior, se puede interrogar al modelo de IA en la generación de código alternativo que encuentre vías más eficientes que las implementadas en el sistema, dentro de las cuales se detecte que se está sobrecargando algún recurso de cómputo o red; esto se puede lograr con herramientas de depuración como *Debugbar de Laravel* o la exploración de los elementos enviados a la red en la *consola del navegador*, o bien utilizando comandos como *sail artisan insights* en *Laravel Sail*. Ver

2 Los datos fueron tomados de la herramienta Mantis Bug Tracker, que sirve para la gestión de incidencias y errores de proyectos de desarrollo de software.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 59 - 80

ejemplos en el Anexo 4 y Anexo 5.

5. Generación de código para *front-end:* Aunque ha resultado un poco limitado en cuanto al control de posiciones fijas y/o relativas de los diferentes elementos en la interfaz de usuario, es posible apoyarse en los modelos de IA GPT-4 para generar elementos complejos en los archivos .blade.php que alojan la interfaz de usuario, como la construcción de vistas y datos agrupados en tablas. Ver ejemplo en el Anexo 6.

6. Búsqueda de librerías de código: Muchas rutinas o procedimientos recurrentes en los sistemas de información ya tienen su solución implementada y compartida por otros programadores en librerías de código disponibles en Internet. Se puede interrogar al modelo GPT-4 para que sugiera las librerías más actuales, o que cuenten con soporte, para integrar al sistema y resolver de forma rápida ciertos problemas comunes y repetitivos. Ver ejemplo en el Anexo 7.

7. Acortar la curva de aprendizaje: Resulta muy ágil interrogar al modelo GPT-4 en la búsqueda de documentación, presentación de ejemplos de uso de determinadas funciones, sentencias o componentes. Asimismo, se puede copiar alguna función y pedirle al modelo de IA que brinde la explicación desglosada de dicha función. Ver ejemplo en el Anexo 8.

La Figura 2 resume los diferentes productos o resultados de los modelos de IA para dar soporte al desarrollo de aplicaciones a la medida.

Figura 2

Productos obtenidos de modelos IA para el desarrollo de aplicaciones a la medida

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 60 - 80

2.3 ANÁLISIS DE PRÁCTICAS PARA EL DISEÑO DE PROMPTS PARA HACER MÁS EFICIENTE LA FASE DE DESARROLLO DE SOFTWARE A LA MEDIDA

El diseño de *prompts* requiere conocimientos técnicos de quien los desarrolla; si se sabe hacer la solicitud al modelo, la respuesta será de mayor ayuda y, posiblemente, el código resultante requiera menos adaptaciones para su integración al sistema (Solohubov et al., s/f).

- Se requieren conocimientos técnicos para poder desarrollar *prompts* que generen código más acertado y correcto. Es mejor brindar estructura técnica al *prompt* para obtener una respuesta que resuelva mejor el problema.
- Se debe conocer el requerimiento del componente, función o rutina para poder expresar la tarea a realizar por el código. Al igual que en la redacción de un documento, se deben evitar las repeticiones innecesarias o pleonasmos que dificulten al modelo la interpretación de la instrucción.
- Usar lenguaje sencillo en la redacción del *prompt* para evitar agregar complejidad en el procesamiento de la instrucción.
- Utilizar verbos más descriptivos o claros.
- *Prompts,* tareas a realizar o código a obtener por el modelo de GPT-4 que incluyan detalles muy concretos y específicos del resultado que se desea obtener. Por ejemplo, se puede dar el nombre de las tablas y el nombre de los campos para generar el *query* de *joins* complejos y se puede pedir al modelo de IA que el *query* brinde formato a alguno o algunos de los campos del resultado (Jarrell, s/f).
- Refinación del *prompt* - Cuando el código generado por la IA se aprecia complejo, se puede mejorar el *prompt* agregando más contexto como, por ejemplo, indicarle al modelo que se desea que el código generado cumpla con un patrón de desarrollo como el *Return early pattern*[3] (Dev, 2020).
- Si los resultados obtenidos después de varias iteraciones en la mejora del *prompt* no son los deseados, se puede probar con otro modelo GPT-4. En el caso práctico del sistema en desarrollo, se usó inicialmente Microsoft Copilot, pero algunas respuestas no daban solución a lo solicitado en el *prompt* y se interrogó a ChatGPT. Se observó que, en ocasiones, Copilot brindaba la mejor respuesta, pero, a veces, la mejor respuesta la generaba ChatGPT, lo que podría abordarse en un estudio posterior.
- Se interrogó tanto a ChatGPT como a Microsoft Copilot si se obtienen mejores respuestas preguntando en algún idioma. La respuesta obtenida fue que es indistinto el idioma en el que se interrogue al modelo. En la refinación de la pregunta, el nuevo *prompt* fue si la mayor parte del entrenamiento del modelo estaba en inglés, ambos modelos respondieron que sí es posible que la mayor parte de los datos de entrenamiento estén en inglés.
- Cuando el código que sugiere el modelo GPT-4 marca errores de ejecución, es posible hacer un nuevo *prompt* incluyendo el mensaje de error obtenido, lo cual brinda al modelo mayor contexto. En la mayoría de los casos, la corrección que brindó el modelo funcionó correctamente y solucionó el error de ejecución.

La Figura 3 esquematiza la aplicación de prácticas para el diseño de *prompts*.

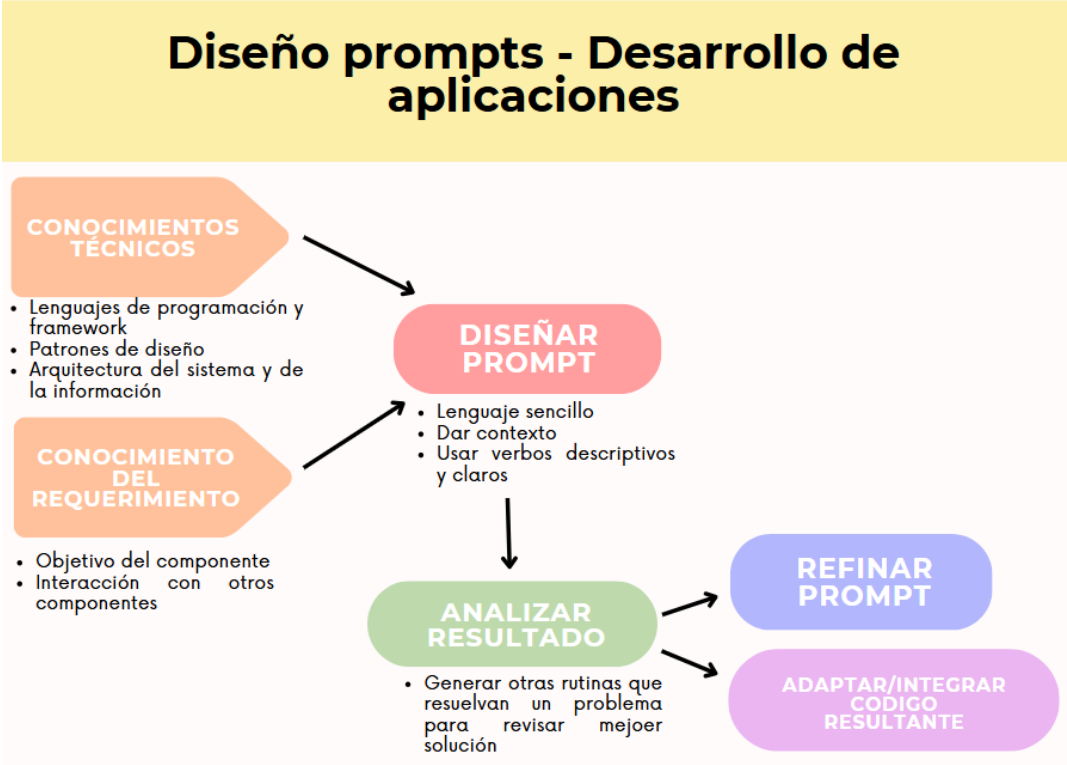
3 Return early pattern - técnica de programación que mejora la legibilidad y la mantenibilidad del código al manejar casos especiales o errores lo antes posible en una función o método.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 61 - 80

Figura 3

Diseño de prompts para el desarrollo de aplicaciones



2.4 ÉTICA EN EL DISEÑO DE PROMPTS PARA EL DESARROLLO DE SOFTWARE

Se debe revisar el aspecto ético al utilizar cualquier modelo de IA generativa, en el sentido de no incluir código protegido por derechos de autor, mismo que se puede identificar porque incluye muchos comentarios detallados de lo que está realizando el código, referencias a fuentes externas, así como explicación detallada de variables y estructuras de datos.

En su mayoría, Copilot tiene como fuente de aprendizaje el código abierto de GitHub y patrones y ejemplos abiertos en la red (iac, 2024). En el caso de ChatGPT, sus fuentes de aprendizaje son sitios y repositorios públicos en Internet, por lo cual, no tiene acceso a repositorios privados.

Sin embargo, si el código parece muy complejo y muy específico, es muy probable que haya sido desarrollado por un programador humano (*Developers Warned*, s/f).

Para evitar caer en la utilización de código fuente protegido por derechos de autor, una práctica sugerida es dividir el problema en unidades funcionales más simples y diseñar los *prompts* para obtener rutinas de estas unidades más simples y más genéricas. Estas rutinas genéricas serían integradas con la habilidad de programación del humano para resolver el problema original. De esta manera, el desarrollador está obteniendo apoyo de la herramienta sólo en implementar funciones comunes y ordinarias, siendo así su intelecto el que está resolviendo el problema.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 62 - 80

Unidades de funcionalidad más específicas que parezcan protegidas por derechos de autor deben de ser programadas por el humano, a fin de respetar la originalidad y la autoría, aunque el humano puede también aprender de estas rutinas. Asimismo, resulta importante detallar el código que se desea obtener con el modelo GPT-4, ya que, si no se especifica lo suficiente, el modelo puede generar prácticas que impliquen sesgos indeseables, aprendidos por el modelo a partir de otros sistemas. (Degli-Esposti, 2023).

Otro aspecto ético a cuidar es analizar la sensibilidad de los procesos a automatizar. Por ejemplo, sería irresponsable interrogar al modelo GPT-4 para que genere rutinas relacionadas con la verificación de la identidad de un usuario, ya que el modelo aprende y podría utilizar el *prompt* y el resultado para resolver algún problema de la misma temática en otra organización. Tampoco deben incluirse datos sensibles en el diseño del *prompt*.

Por ello, resulta una buena práctica no incluir la información, ni los procedimientos sensibles, o bien, instalar algún modelo GPT de manera local, a fin de que estos no queden expuestos en algún momento, o bien, desarrollar y construir, sin apoyo de modelos GPT, este tipo de automatizaciones. (Javier Garzás en LinkedIn, s.f).

A continuación, se enlistan algunas aplicaciones de IA generativa que son gratuitas, de código abierto y que pueden instalarse localmente y que no necesitan de conexión a Internet para generar código:

- Llama (*Large Language Model Meta AI*)
- Qwen de Alibaba Cloud
- Gemma de Google AI
- Ollama de la empresa homónima
- Phi de Microsoft

3. RESULTADOS

Se observó que la utilización de modelos GPT y el diseño de *prompts* contextualizados y breves ayudaron a reducir el número de incidencias encontradas atribuibles a defectos en el proceso de desarrollo de código, como se mostró en la Tabla 1.

En seguimiento a los ámbitos mencionados al inicio de este reporte técnico, se puede comentar que:

1. Reducción de incidencias

La Tabla 1 muestra el número de incidencias registradas en tres sistemas desarrollados, cuya causa-raíz es directamente el desarrollo de componentes. Usando el modelo GPT, se observa una disminución en el número de incidencias cuya causa-raíz es desarrollo-código.

2. Disminución del tiempo de desarrollo de software

   Se observó una disminución de un 20 - 30% en el tiempo de desarrollo de componentes con el uso de modelos GPT[4]. Cabe mencionar que en esta estadística personal se ha considerado el tiempo

4 Estimaciones basadas en *Personal Software Process* (PSP), en donde cada desarrollador registra sus estadísticas

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111 **Vol. 3, Núm. 2.** abril-junio 2025, págs. 63 - 80

que conlleva diseñar el *prompt*, analizar el código generado e integrar el código al sistema.

3. Reducción del tiempo de corrección de incidencias

   Se registró una disminución de un 20 - 40% en el tiempo de corrección de defectos con el uso de modelos GPT[5].

Se observó una disminución en el tiempo de desarrollo de componentes de un 20 - 30%, si se considera la inversión de tiempo en el diseño del *prompt,* en su refinamiento a través de varias iteraciones para su mejora, en el análisis del código generado, en la revisión de la eficiencia de éste y en la adaptación para la integración.

A pesar de esta inversión de tiempo, se obtienen resultados eficientes, ya que los modelos GPT se basan en el código desarrollado por expertos humanos, con lo cual, el código generado puede contener funciones que el programador humano desconozca y que acorten o hagan más eficiente la automatización de algún proceso.

También hubo una disminución del tiempo para la corrección de incidencias entre un 20 - 40%. En ocasiones, no tiene tanto impacto el uso de los modelos GPT en la corrección de incidencias, ya que son demasiado específicas y requieren mayor trabajo por parte del desarrollador; sin embargo, en los mejores casos, se disminuye hasta en un 40% de tiempo, ya que las respuestas del modelo son inmediatas y no se requiere leer tanta documentación para poder corregir el código.

Se observó un ahorro de tiempo al obtener resultados inmediatos en la generación de código por los modelos GPT, ya que se basan en soluciones ya implementadas que van incluyendo en su modelo de entrenamiento.

Hubo mejora en la calidad del código, puesto que los componentes generados toman en cuenta las mejores prácticas y estándares de la industria, especialmente si desde el *prompt* se solicita este factor. Además, en muchas ocasiones, fue más fácil resolver errores y problemas del código desarrollado por programadores humanos, siguiendo las recomendaciones que brindaban los modelos GPT. Esta tarea fue más rápida que consultando documentación y foros de discusión o bases de conocimiento como *Stackoverflow*[6].

La generación de elementos, como expresiones regulares para validar datos de entrada del sistema, fue más rápido y eficiente al obtenerlos del modelo GPT y después analizarlos y probarlos, que construyéndolos de manera tradicional. Asimismo, el modelo GPT puede generar el código de validaciones hechas a la medida para integrarlas en Laravel y agregarlas a la base de validaciones que ya tiene preconstruidas el *framework*.

individuales para utilizarlas en la medición y estimación de su trabajo.

5 Estimaciones basadas en *Personal Software Process* (PSP).

6 Stackoverflow – es una plataforma en Internet en dónde programadores publican preguntas y otros programadores publican las respuestas, constituyendo una base de conocimientos viva, en constante actualización.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 64 - 80

Mediciones de los componentes

La herramienta *Php Insights de Sail*, para análisis de código en proyectos PHP genera mediciones como la calidad del código (organización, legibilidad y coherencia del código), complejidad, arquitectura y estilo de codificación.

Aplicando la herramienta a algunos de los componentes que incluyen métodos con código adaptado obtenido de los modelos GPT, se obtuvieron métricas que se encuentran en semáforo verde y amarillo, como las que se muestra a continuación, y que permiten confirmar la validez y utilidad del código obtenido de los modelos:

Figura 4

Métricas del componente RegistrarCensoEthernetComponent.php

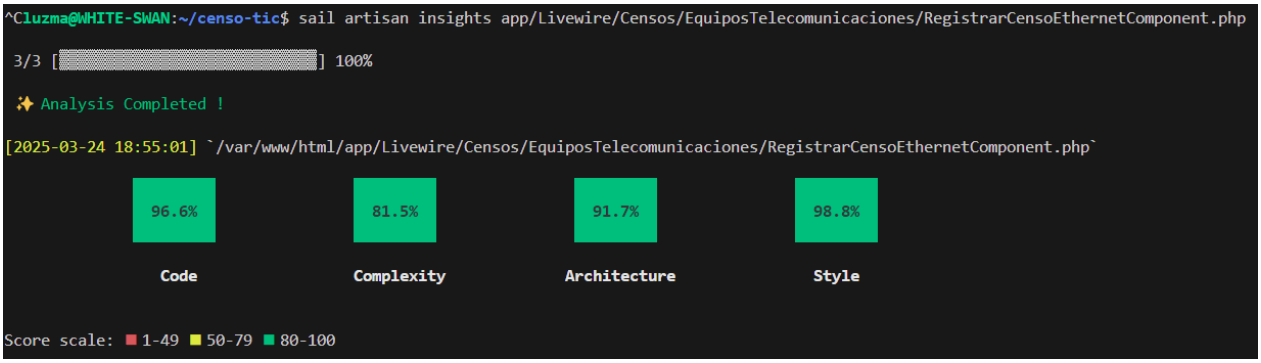


Figura 5

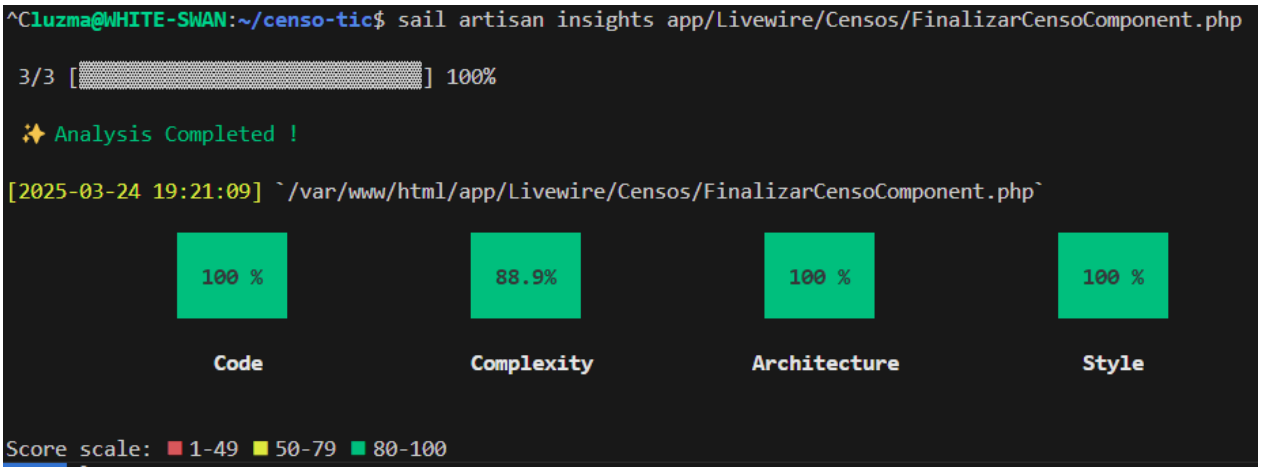
Métricas del componente RegistrarFibraOpticaComponent.php

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 65 - 80

Figura 6

Métricas del componente FinalizarCensoComponent.php



4. CONCLUSIONES

Dado que los beneficios que se obtuvieron en el desarrollo del Sistema de Censo utilizando modelos GPT también pueden alcanzarse en la construcción de cualquier software a la medida, se vuelve importante utilizar esta técnica en el desarrollo de software.

Al mismo tiempo, resulta imprescindible analizar y cuestionar el código que genera el modelo GPT ya que, en muchas ocasiones, el código generado no funcionará dentro del contexto del sistema en el que se está desarrollando la funcionalidad requerida.

El programador, que crea el *prompt* para generar código a través de los modelos GPT-4, debe contar con el marco de conocimientos técnicos, a fin de desarrollar un *prompt* específico, que brinde contexto y sea claro en el código que se espera, lo cual puede incluso involucrar el especificar tipos de datos de la entrada y la salida esperada. Mientras más información técnica brinde el *prompt*, el código generado será más cercano al requerimiento, de otra manera, se puede caer en muchas iteraciones, tratando de refinar el *prompt*, sin conseguir un resultado que funcione. Por esta razón, el programador debe seguir actualizándose y capacitándose, a fin de utilizar de manera eficiente los modelos de IA, en beneficio de la organización y de sus usuarios. Además, el programador analizará la eficiencia del código generado por el modelo de IA, a fin de obtenerla en el desempeño del sistema en el cuál se integrará para lo anterior, se requiere que el humano cuente con los conocimientos técnicos necesarios.

Los modelos GPT-4 ayudan a reducir los tiempos de desarrollo y corrección de incidencias desde el momento de escribir el código, ya que ponen automáticamente el nombre de las variables en las funciones, mismas que fueron declaradas como propiedades en las clases del componente, sugieren código (si uno declara un arreglo, el modelo sugiere el código para recorrerlo, o para desplegarlo en el blade) y ayudan en la creación de rutinas específicas, interrogando al modelo por medio de los *prompts*.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 66 - 80

Es importante resaltar que el ingenio de crear el *prompt* para solicitarle código al modelo GPT-4 pertenece al programador humano. En el desarrollo del *prompt*, se vierte la creatividad del programador para solicitar diferentes formas de construir el código que contribuya a resolver un problema de automatización y también se aplica el discernimiento humano para seleccionar la mejor solución generada y su adaptación para su integración al resto del sistema, como se ilustra en el Anexo 4.

El Anexo 5 es otra muestra de refinación del *prompt* con conocimientos técnicos, aunque, en este ejemplo, la mejor opción de código en cuanto a rendimiento de uso de memoria la brindó el modelo GPT. No obstante, es útil explorar estos cambios en el *prompt*, a fin de analizar diferentes caminos de resolver un mismo problema.

El proceso de aprendizaje y el entrenamiento de la mente humana para incrementar las capacidades lógica, analítica, creativa y orientadas a la solución de problemas debe ser un proceso permanente del ingeniero de software. También, en este proceso, los modelos GPT inciden, ya que es posible estudiar la explicación que brinda el modelo de cada solución que propone, pues en cada resultado generado por los modelos GPT, el programador humano puede aprender diferentes tácticas para resolver un mismo problema con el anhelado crecimiento profesional de cada programador humano.

REFERENCIAS

2024 Stack Overflow Developer Survey. (s.f). Recuperado el 15 de enero de 2025, de https://survey.stackoverflow.co/2024/

Brown, T., Mann, B., et al. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901. https://papers.nips.cc/paper_files/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

Degli-Esposti, S. (2023). *La ética de la inteligencia artificial*. Los Libros De La Catarata.

De, las H. del D., Rafael, Alonso, Á. G., & Carmen, L. G. (2018). *Métodos Ágiles. Scrum, Kanban, Lean*. Comercial Grupo ANAYA, S.A.

Dev, L. M. (2020, agosto 7). Return Early Pattern. *The Startup*. https://medium.com/swlh/return-early-pattern-3d18a41bba8

Developers warned: GitHub Copilot code may be licensed | TechTarget. (s/f). Search Software Quality. Recuperado el 24 de marzo de 2025, de https://www.techtarget.com/searchsoftwarequality/news/252526359/Developers-warned-GitHub-Copilot-code-may-be-licensed

iac. (2024, julio 24). *Microsoft Copilot con GitHub: ¿Cómo funciona?* Ingeniería Asistida Por Computador. https://www.iac.com.co/microsoft-copilot-con-github/

Jarrell, G. (s.f). *Chatgpt: Hacia Una Nueva Conciencia La Ingeniería De Prompts Para Chatgpt Que Va a Hacer De Ti El Genio Que Quieres Ser (La Superinteligencia Y Cómo Chatgpt Está Cambiando El Software La Comunicación Y La)*. Gordon Jarrell.

Javier Garzás en LinkedIn: ¿MIEDO subir tus datos a una IA? Pues instálate una en local, sin acceso…. (s.f). Recuperado el 15 de enero de 2025, de https://es.linkedin.com/posts/jgarzas_miedo-subir-tus-datos-a-una-ia-pues-instalate-activity-7272954391580594176-uee_

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 67 - 80

Martí, M. A., Antonín, M. A. M., Llisterri, J., & Boix, J. L. (2002). *Tratamiento del lenguaje natural: Tecnología de la lengua oral y escrita*. Edicions Universitat Barcelona.

Martin, R. C. (2018). *Arquitectura limpia*. Editorial Anaya.

Martínez, R. N. (2015). *El Proceso de Desarrollo de Software*. IT Campus Academy.

Ramana, T. V., Ghantasala, G. S., Sathiyaraj, R., & Khan, M. (2024). *Artificial Intelligence and Machine Learning for Smart Community: Concepts and Applications*. CRC Press.

Reilly, E. (2024). *The No-Code Startup: The complete guide to building apps without code*. Practical Inspiration Publishing.

Robisco, S. (2024). *Historia de la Inteligencia Artificial*. Guadalmazán.

Solohubov, I., Moroz, A., Tiahunova, M. Y., Kyrychek, H. H., & Skrupsky, S. (s/f). *Accelerating software development with AI: exploring the impact of ChatGPT and GitHub Copilot*. CEUR Workshop Proceedings, 3679, 76-86. https://ceur-ws.org/Vol-3679/paper17.pdf

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 68 - 80

ANEXO 1

Ejemplo de correcciones de errores de sintaxis

En este ejemplo puede verse cómo el *prompt* es breve, con una tarea concreta y se debe contextualizar proporcionando el fragmento de código que causa el error, o bien, el fragmento de código que el programador detecta que tiene el problema, con ayuda de diferentes técnicas de depuración.

Figura 7

Prompt y respuesta del modelo GPT de ejemplo de corrección de errores de sintaxis

Encuentra el error en: $buscarSede =
Sede::where('id_area_universitaria', $idAreaUniversitaria)
->where('nombre', $nombreSede);

El error en el código que has proporcionado está en la falta del método `get()` o `first()` para ejecutar la consulta y obtener los resultados. Aquí tienes el código corregido:

Php Copiar

```php
$buscarSede = Sede::where('id_area_universitaria', $idAreaUniversitaria)
                ->where('nombre', $nombreSede)
                ->first(); // o ->get()
```

Explicación:

- `first()`: Obtiene el primer resultado que coincida con la consulta.
- `get()`: Obtiene todos los resultados que coincidan con la consulta.

Dependiendo de si esperas uno o varios resultados, puedes usar `first()` o `get()`. Si necesitas más ayuda o tienes alguna otra pregunta, ¡estoy aquí para ayudarte!

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 69 - 80

ANEXO 2

Ejemplo de generación de código para atender rutinas comunes

Figura 8

Prompt de ejemplo de generación de código para atender rutinas comunes

Genera en laravel un método para ordenar ascendentemente por id_herramienta, los elementos de un array asociativo que contiene id_herramienta, nombre, tipo_licenciamiento, proveedor.

En Laravel, puedes usar la función `usort()` de PHP para ordenar un array asociativo por `id_herramienta` de forma ascendente. Aquí tienes un método que lo hace dentro de una clase de servicio o controlador:

Php Copiar

```php
class HerramientaService
{
    public function ordenarPorIdHerramienta(array $herramientas): array
    {
        usort($herramientas, function ($a, $b) {
            return $a['id_herramienta'] <=> $b['id_herramienta'];
        });

        return $herramientas;
    }
}

// Ejemplo de uso
$herramientas = [
    ['id_herramienta' => 3, 'nombre' => 'Herramienta C', 'tipo_licenciamient
    ['id_herramienta' => 1, 'nombre' => 'Herramienta A', 'tipo_licenciamient
    ['id_herramienta' => 2, 'nombre' => 'Herramienta B', 'tipo_licenciamient
];

$service = new HerramientaService();
$herramientasOrdenadas = $service->ordenarPorIdHerramienta($herramientas);

print_r($herramientasOrdenadas);
```

Este método utiliza `usort()` y el operador de comparación `<=>`, lo que proporciona una ordenación eficiente y clara. ¿Quieres integrarlo en algún contexto específico dentro de Laravel, como una consulta de base de datos o un recurso de API? ¡Estoy aquí para ayudar! 🚀

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 70 - 80

ANEXO 3

Ejemplo de generación de consultas SQL

Figura 9

Prompt de ejemplo de generación de consultas SQL

El modelo CensoAntivirus tiene esta relación: public function software_herramienta(): BelongsTo { return $this->belongsTo(SoftwareHerramienta::class, 'id_software_herramienta', 'id_software_herramienta'); } Genera la consulta para obtener el campo nombre que está en la tabla SoftwareHerramienta dado un determinado $idCensoAntivirus

Puedes utilizar una consulta en Eloquent para obtener el campo `nombre` de la tabla `SoftwareHerramienta`, dado un determinado `$idCensoAntivirus`. Aquí tienes cómo hacerlo:

Php Copiar

```php
$nombreSoftwareHerramienta = CensoAntivirus::where('id', $idCensoAntivirus)
    ->first()
    ->software_herramienta
    ->nombre;

echo $nombreSoftwareHerramienta;
```

Esta consulta primero busca el registro en `CensoAntivirus` con el `id` especificado. Luego accede a la relación `software_herramienta` y obtiene el campo `nombre` de la tabla `SoftwareHerramienta`.

Si necesitas validar que la relación existe antes de acceder al campo, puedes hacerlo con:

Php Copiar

```php
$censoAntivirus = CensoAntivirus::where('id', $idCensoAntivirus)->first();

if ($censoAntivirus && $censoAntivirus->software_herramienta) {
    $nombreSoftwareHerramienta = $censoAntivirus->software_herramienta->nomb
    echo $nombreSoftwareHerramienta;
} else {
    echo "No se encontró el software herramienta para el ID proporcionado.";
}
```

Así evitamos posibles errores en caso de que no haya coincidencias. ¿Necesitas agregar algún filtro adicional o modificar la consulta? Estoy aquí para ayudar.

Nota de la autora: El desarrollador debe adaptar el código puesto que seguramente las instrucciones con 'echo' no son necesarias en el contexto del sistema. También el parámetro $idCensoAntivirus debe pasarse de manera adecuada al método que se está implementando.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 71 - 80

ANEXO 4

Ejemplo 1 de búsqueda de alternativas de código

Figura 10

Prompt de ejemplo de búsqueda de alternativas de código

Tengo la consulta: $this->listadoFibrasOpticas = VistaFibraOptica::where('id_censo_area', $this->idCensoArea)->get();
Genera una consulta más eficiente.

Puedes mejorar la eficiencia de tu consulta agregando filtros específicos y seleccionando solo los campos que necesitas en lugar de recuperar toda la tabla. Aquí tienes una versión optimizada:

Php Copiar

```php
$this->listadoFibrasOpticas = VistaFibraOptica::select('id', 'nombre', 'long
    ->where('id_censo_area', $this->idCensoArea)
    ->get();
```

Si `VistaFibraOptica` es una tabla grande, considera utilizar `pluck()` o `chunk()` dependiendo del caso de uso:

Php Copiar

```php
$this->listadoFibrasOpticas = VistaFibraOptica::where('id_censo_area', $this
    ->pluck('id', 'nombre'); // Solo recupera los valores necesarios
```

También podrías usar `chunk()` si estás procesando muchos registros para mejorar la gestión de memoria:

Php Copiar

```php
VistaFibraOptica::where('id_censo_area', $this->idCensoArea)
    ->chunk(100, function ($fibras) {
        // Procesar en lotes de 100 registros
    });
```

Si trabajas con modelos que no cambian frecuentemente, considera agregar `cache()` para evitar consultas repetitivas. ¿Necesitas más mejoras en el rendimiento o tienes otro contexto que quieras optimizar?

Nota de la autora:

Los desarrolladores en Laravel saben que hay métodos más eficientes para generar la consulta anterior que el primer resultado que ofreció el modelo GPT, así que se buscará refinar el *prompt* como sigue a continuación.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 72 - 80

Refinación del *prompt*, usando conocimientos técnicos

Figura 11

Ejemplo de refinación de prompt

Tengo la consulta: $this->listadoFibrasOpticas = VistaFibraOptica::where('id_censo_area', $this->idCensoArea)->get();
Se puede generar una consulta más eficiente sin usar eloquent?

Sí, puedes optimizar la consulta utilizando SQL puro en Laravel mediante el constructor de consultas de Laravel (`DB::select`) o `DB::table`, lo que reducirá la sobrecarga de Eloquent. Aquí te dejo una alternativa eficiente:

Php Copiar

```php
use Illuminate\Support\Facades\DB;

$this->listadoFibrasOpticas = DB::select("SELECT * FROM vista_fibra_optica W
```

Otra opción es usar `DB::table`, que sigue siendo más eficiente que Eloquent pero mantiene la fluidez de Laravel:

Php Copiar

```php
use Illuminate\Support\Facades\DB;

$this->listadoFibrasOpticas = DB::table('vista_fibra_optica')
    ->where('id_censo_area', $this->idCensoArea)
    ->get();
```

Ambos enfoques eliminan la carga de Eloquent al manejar modelos y mejoran el rendimiento en consultas con grandes volúmenes de datos. ¿Necesitas más detalles o ayuda con índices y optimización en la base de datos?

Nota de la autora:

Se observa que, si el desarrollador brinda más detalles técnicos al modelo GPT, la respuesta del modelo mejora y se acerca a lo que espera el desarrollador humano.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 73 - 80

ANEXO 5

Ejemplo 2 de búsqueda de alternativas de código

Figura 12

Prompt para solicitar una función de obtención de totales por tipo de periférico

Genera una consulta a una tabla de censo_equipo_computo que tenga id_tipo_periferico y cantidad y que calcule el total por id_tipo_periferico

Aquí tienes una consulta SQL que te ayudará a calcular el total de cantidad por `id_tipo_periferico` en la tabla `censo_equipo_computo`:

Sql Copiar

```sql
SELECT id_tipo_periferico, SUM(cantidad) AS total_cantidad
FROM censo_equipo_computo
GROUP BY id_tipo_periferico;
```

Esta consulta utiliza la función `SUM()` para sumar las cantidades de cada `id_tipo_periferico`, agrupándolas con `GROUP BY` para obtener el total por cada tipo de periférico.

Si necesitas filtrar por alguna condición adicional o modificar la consulta para adaptarla a tu sistema, dime qué requisitos tienes y con gusto te ayudo a ajustarla. ¿Quieres agregar algún otro criterio?

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 74 - 80

Figura 13

Prompt para la búsqueda de una alternativa con el método map

Genera una consulta en laravel 11 a una tabla de censo_equipo_computo que tenga id_tipo_periferico y cantidad y que calcule el total por id_tipo_periferico. Usa el método map.

En Laravel 11, puedes usar el método `map()` de las colecciones para agrupar y calcular el total por `id_tipo_periferico`. Si estás obteniendo los datos de la tabla `censo_equipo_computo` con Eloquent, podrías hacer algo así:

Php Copiar

```php
use App\Models\CensoEquipoComputo;

$datos = CensoEquipoComputo::select('id_tipo_periferico', 'cantidad')->get()

$resultado = $datos->groupBy('id_tipo_periferico')->map(function ($items) {
    return $items->sum('cantidad');
});

dd($resultado); // Mostrar el resultado para depuración
```

Aquí:

- Se obtiene la información de la base de datos con `select()`.
- `groupBy('id_tipo_periferico')` agrupa los registros por tipo de periférico.
- `map()` itera sobre cada grupo y usa `sum('cantidad')` para obtener el total de cada uno.

Nota de la autora:

En este ejemplo, el modelo GPT brindó de primera instancia la mejor solución. Se observa que el programador humano intentó refinar la respuesta a través del uso de otro método, sin embargo, según el modelo GPT, ésta no era la mejor opción en uso de recursos de memoria.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 75 - 80

ANEXO 6

Generación de código para *front-end*

Las Figuras 14-A, 14-B y 14-C están relacionadas con el prompt solicitando la generación de un archivo de front-end (blade.php) para Livewire versión 3, incorporando una tabla html con el resultado de una consulta a una base de datos.

Figura 14-A

Primera parte del resultado del modelo ChatGPT para generar una tabla en un archivo .blade.php a partir de una consulta a un modelo de base de datos en Laravel-Livewire

En livewire 3 genera un .blade.php para mostrar en una tabla el resultado de la consulta $cursos = Curso::all();.
El resultado obtiene los datos de los campos id_nombre_curso, nombre, area

En Livewire 3, puedes generar una vista en `.blade.php` para mostrar los resultados de la consulta en una tabla. Aquí tienes un ejemplo de cómo podrías hacerlo:

Paso 1: Componente Livewire

Primero, asegúrate de tener un componente Livewire. Puedes crearlo con el siguiente comando en Laravel:

Sh Copiar

```sh
php artisan make:livewire CursosTable
```

Esto generará dos archivos:

- `app/Http/Livewire/CursosTable.php`
- `resources/views/livewire/cursos-table.blade.php`

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 76 - 80

Figura 14-B

Segunda parte del resultado del modelo ChatGPT para generar una tabla en un archivo .blade.php a partir de una consulta a un modelo de base de datos en Laravel-Livewire

Paso 2: Controlador Livewire (`CursosTable.php`)

Edita el archivo `CursosTable.php` para obtener los cursos desde la base de datos:

Php Copiar

```php
namespace App\Http\Livewire;

use Livewire\Component;
use App\Models\Curso;

class CursosTable extends Component
{
    public $cursos;

    public function mount()
    {
        $this->cursos = Curso::all();
    }

    public function render()
    {
        return view('livewire.cursos-table');
    }
}
```

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 77 - 80

Figura 14-C

Tercera y última parte del resultado del modelo ChatGPT para generar una tabla en un archivo .blade.php a partir de una consulta a un modelo de base de datos en Laravel-Livewire

Paso 3: Vista Blade (`cursos-table.blade.php`)

Edita `cursos-table.blade.php` para mostrar los cursos en una tabla:

Blade Copiar

```
<div>
    <table class="table-auto border-collapse border border-gray-400 w-full">
        <thead>
            <tr class="bg-gray-200">
                <th class="border border-gray-400 px-4 py-2">ID</th>
                <th class="border border-gray-400 px-4 py-2">Nombre</th>
                <th class="border border-gray-400 px-4 py-2">Área</th>
            </tr>
        </thead>
        <tbody>
            @foreach($cursos as $curso)
                <tr>
                    <td class="border border-gray-400 px-4 py-2">{{ $curso->
                    <td class="border border-gray-400 px-4 py-2">{{ $curso->
                    <td class="border border-gray-400 px-4 py-2">{{ $curso->
                </tr>
            @endforeach
        </tbody>
    </table>
</div>
```

Paso 4: Incluir el Componente en la Vista Principal

Para mostrar la tabla en alguna vista de Laravel, incluye el siguiente código en `welcome.blade.php` o en la vista donde lo necesites:

Blade Copiar

```
@livewire('cursos-table')
```

Nota de la autora:

En este ejemplo, se observa que el modelo GPT no sólo generó el archivo .blade.php solicitado. Además, sugirió el contenido del componente de Livewire, con la consulta a la base de datos y mostró cómo incluir el componente de livewire en una vista del framework Laravel, con el que trabaja Livewire.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

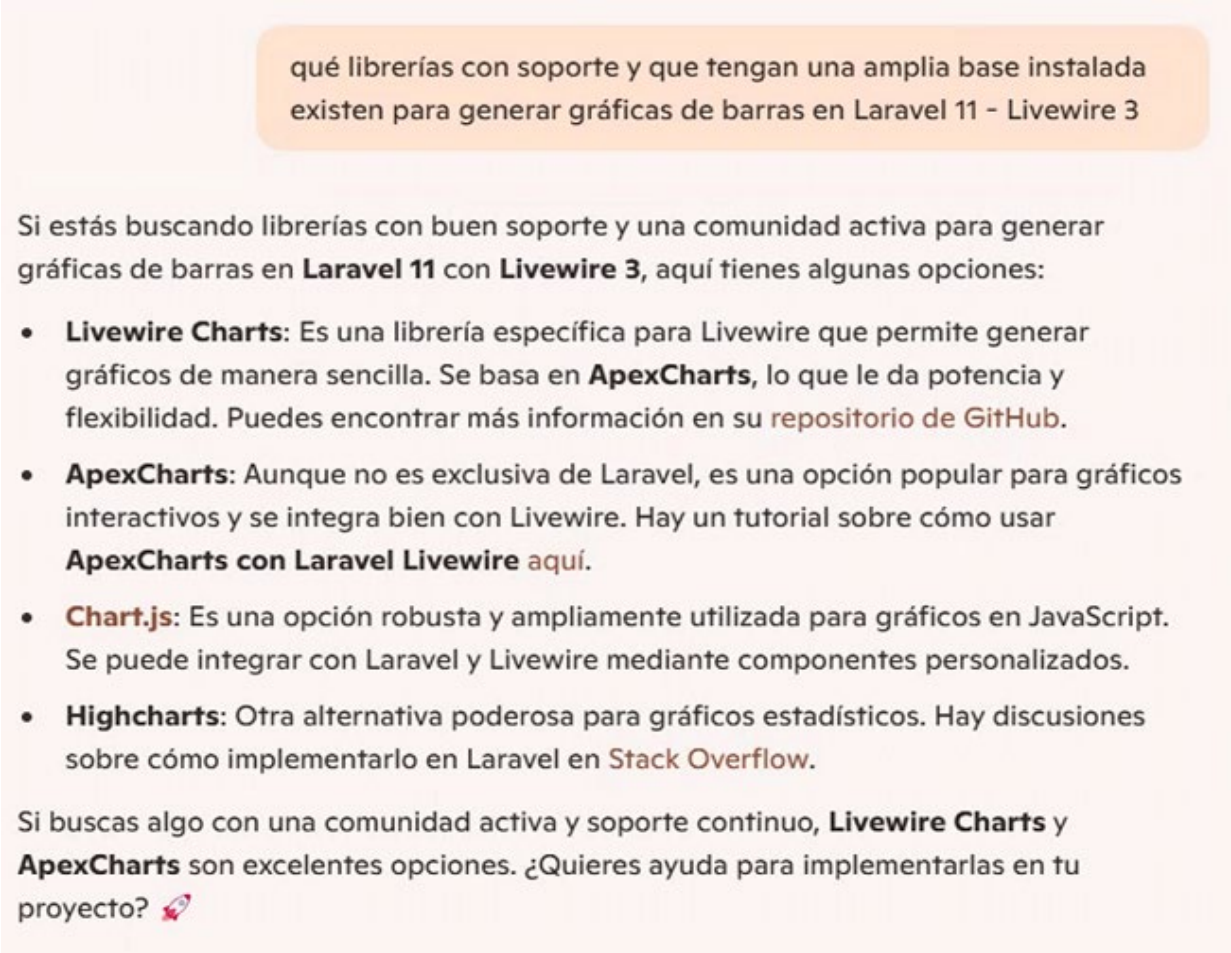
Vol. 3, Núm. 2. abril-junio 2025, págs. 78 - 80

ANEXO 7

Búsqueda de librerías de código

Figura 15

Prompt solicitando librerías para generar gráficas



qué librerías con soporte y que tengan una amplia base instalada existen para generar gráficas de barras en Laravel 11 - Livewire 3

Si estás buscando librerías con buen soporte y una comunidad activa para generar gráficas de barras en **Laravel 11** con **Livewire 3**, aquí tienes algunas opciones:

- **Livewire Charts**: Es una librería específica para Livewire que permite generar gráficos de manera sencilla. Se basa en **ApexCharts**, lo que le da potencia y flexibilidad. Puedes encontrar más información en su repositorio de GitHub.
- **ApexCharts**: Aunque no es exclusiva de Laravel, es una opción popular para gráficos interactivos y se integra bien con Livewire. Hay un tutorial sobre cómo usar **ApexCharts con Laravel Livewire** aquí.
- **Chart.js**: Es una opción robusta y ampliamente utilizada para gráficos en JavaScript. Se puede integrar con Laravel y Livewire mediante componentes personalizados.
- **Highcharts**: Otra alternativa poderosa para gráficos estadísticos. Hay discusiones sobre cómo implementarlo en Laravel en Stack Overflow.

Si buscas algo con una comunidad activa y soporte continuo, **Livewire Charts** y **ApexCharts** son excelentes opciones. ¿Quieres ayuda para implementarlas en tu proyecto? 🚀

Nota de la autora:

Sólo se incluyó la respuesta completa del modelo GPT en la primera librería recomendada para no hacer muy extenso el texto. En cada recomendación de librerías, el modelo GPT incluyó información sobre cómo hacer su instalación y configuración, así como un ejemplo de uso para hacer más fácil el proceso de selección de la librería. Es de resaltar que toda la información está recopilada en la respuesta del modelo GPT, lo que ahorra tiempo en la búsqueda de la información y de los ejemplos de uso en la documentación técnica de cada librería.

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 79 - 80

ANEXO 8

Acortar la curva de aprendizaje

Las Figuras 16-A y 16-B están relacionadas con el *prompt* para el aprendizaje del uso de #[On para el manejo de eventos en la nueva versión de livewire (3) para el desarrollo de aplicaciones a la medida

Figura 16-A

Primera parte de la respuesta del modelo GPT para el aprendizaje del uso de #[On

explícame la notación #[On para el manejo de eventos en livewire 3

La notación `#[On]` en **Livewire 3** se utiliza para **escuchar eventos** dentro de un componente. Es una forma elegante y declarativa de manejar eventos sin necesidad de escribir código adicional en la plantilla Blade.

Cómo funciona

Cuando un evento es **emitido** desde un componente, otro componente puede **escucharlo** utilizando `#[On]`. Esto permite que el método correspondiente se ejecute automáticamente cuando el evento ocurra.

Ejemplo básico

Supongamos que tienes un componente que emite un evento cuando se crea un nuevo post:

Php Copiar

```php
use Livewire\Component;

class CreatePost extends Component
{
    public function save()
    {
        // Guardar el post...
        $this->dispatch('post-created');
    }
}
```

Ahora, otro componente puede **escuchar** ese evento y reaccionar a él:

Php Copiar

```php
use Livewire\Component;
use Livewire\Attributes\On;

class Dashboard extends Component
{
    #[On('post-created')]
    public function updatePostList()
    {
        // Código para actualizar la lista de posts...
    }
}
```

https://doi.org/10.22201/dgtic.30618096e.2025.3.2.111

Vol. 3, Núm. 2. abril-junio 2025, págs. 80 - 80

Figura 16-B

Segunda parte de la respuesta del modelo GPT (paso de parámetros) para concluir la explicación del uso de #[On

- **Pasar parámetros**

Si el evento envía datos adicionales, estos se pueden recibir en el método:

Php Copiar

```php
use Livewire\Component;
use Livewire\Attributes\On;

class Dashboard extends Component
{
    #[On('post-created')]
    public function updatePostList($title)
    {
        // Actualizar la lista con el nuevo título...
    }
}
```

- **Escuchar eventos dinámicos**

También puedes **dinamizar** los nombres de los eventos, por ejemplo, si quieres que un evento solo afecte a un modelo específico:

Php Copiar

```php
use Livewire\Component;

class UpdatePost extends Component
{
    public function update()
    {
        $this->dispatch("post-updated.{$post->id}");
    }
}
```

Y luego escucharlo en otro componente:

Php Copiar

```php
use Livewire\Component;
use App\Models\Post;
use Livewire\Attributes\On;

class ShowPost extends Component
{
    public Post $post;

    #[On('post-updated.{post.id}')]
    public function refreshPost()
    {
        // Código para actualizar el post...
    }
}
```

Nota de la autora:

La información y ejemplos del uso de la directiva #[On está concentrada en la respuesta del modelo para facilidad de quién estudia esta actualización al *framework* Livewire en su versión 3 .

DOI: 10.22201/dgtic.30618096e.2025.3.2 Vol. 3, Núm. 2. abril-junio 2025

REVISORES JULIO 2024 A JUNIO 2025

Desde la editorial de Cuadernos Técnicos Universitarios de la DGTIC expresamos nuestro más amplio reconocimiento a los especialistas del área de tecnologías de información y comunicación que contribuyeron en la revisión de los reportes recibidos en los números 3 y 4 del volumen 2, así como los números 1 y 2 del volumen 3.

Gracias a ustedes es posible publicar reportes técnicos de TI, de calidad y pertinencia. Sabemos del tiempo y esfuerzo que invierten en este trabajo y valoramos enormemente sus recomendaciones para la mejora de cada reporte.

Rocío Alejandra Aldeco Pérez
Alfredo Alonso Peña
Rubén Aquino Luna
Mario Alberto Arredondo Guzmán
Ricardo Arroyo Mendoza
Juventino Ávila Ramos
Héctor Bautista Vázquez
Patricia Canela Manzo
Lorena Cárdenas Guzmán
Luz María Castañeda de León
Juan Manuel Castillejos Reyes
Antonio Baruch Cuevas Ortiz
Max Ulises de Mendizábal Carrillo
Carmen Díaz Novelo
Joel Espinosa Longi
Isabel Galina Rusell
Alma Rosa García Martínez
Alberto González Guízar
Pablo Antonio González Peñaloza
Fernando Israel González Trejo
Jesús Hernández Cabrera
Griselda Berenice Hernández Cruz
María Teresa Hernández Elenes
Alfredo Hernández Mendoza
Alejandra Herrera Mendoza
Erick S. Huesca Morales
Salma Leticia Jalife Villalón
Marina Kriscautzky Laxague
Eduardo López Molina
María Esther Judith Martínez González
Alejandro Ulises Mendoza Pérez
Gerardo Elías Navarrete Terán
Jesús Ochoa Méndez
Hanna Jadwiga Oktaba
Juan Carlos Olivares Rojas
Israel Ortega Cuevas
Jorge Palacios Elizalde
Ana Cecilia Pérez Arteaga
Marcelo Pérez Medel
Pedro Temachti Pérez Santillán
Víctor Rangel Licea
Hugo Alonso Reyes Herrera
Luis Raúl Rivera Herrera
Sandra Roan Cano
Ana Berenice Rodríguez Lorenzo
Lily Sacal Neumann
Rubén Saenz González
José Antonio Salazar Carmona
Aurelio Sánchez Vaca
Edith Tapia Rangel
Francisco Valdés Souto
María Teresa Ventura Miranda
Ricardo Federico Villarreal Martínez

CUADERNOS TÉCNICOS UNIVERSITARIOS DE LA DGTIC

ISSN-e: 3061-8096

DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN