

CUADERNOS TÉCNICOS UNIVERSITARIOS DE LA **DGTIC**

ISSN en trámite

Vol. 3, Núm. 1. enero-marzo 2025

DOI:10.22201/dgtic.ctud.2025.3.1



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

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 (1), páginas (N1-N2).

CUADERNOS TÉCNICOS UNIVERSITARIOS DE LA DGTIC, Año 3, No. 1, enero-marzo 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: en trámite, ambos otorgados por el Instituto Nacional del Derecho de Autor. Dra. Marcela J. Peñaloza Báez, responsable de la última actualización de este número, Circuito Exterior s/n, frente a la Facultad de Contaduría y Administración, Ciudad Universitaria, Alcaldía Coyoacán, C.P. 04510, Ciudad de México. Fecha de la última modificación, 13 de febrero de 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/>

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.

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

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

CONTENIDO

Implementación de una arquitectura modular en Laravel para sistemas web mantenibles Karla Alejandra Fonseca Márquez	8
Análisis de madurez del área de TIC Gustavo Villarreal Brito	18
Reordenamiento dinámico de elementos usando drag & drop en Laravel-11 y Livewire-3 José Antonio Ochoa Rodríguez	36
Análisis de la información de tarificación telefónica con Power BI® Isabel Georgina Gallardo Valadez.....	51
De Excel a Power BI para reducir tiempo de procesamiento de datos Areli Vázquez Padilla Díaz.....	60
Reporte de pruebas en nodo de cálculo Dell R7525 Irving Carlos Álvarez Castillo	71
Análisis estático de código fuente de aplicativos en materia de seguridad informática Angie Aguilar Domínguez.....	89

Implementación de una arquitectura modular en Laravel para sistemas web mantenibles

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:

Fonseca Márquez, K. A. (2025). Implementación de una arquitectura modular en Laravel para sistemas web mantenibles. *Cuadernos Técnicos Universitarios de la DGTIC*, 3 (1) páginas(8 - 17).

<https://doi.org/10.22201/dgtic.ctud.2025.3.1.85>

Karla Alejandra Fonseca Márquez

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

karlafm@unam.mx

ORCID: 0009-0002-6382-8182

Resumen

Para mejorar la cohesión y reducir el acoplamiento de los componentes en sistemas web complejos, se diseñó y desarrolló una arquitectura modular en Laravel que busca superar las limitaciones del enfoque tradicional basado en el patrón MVC. Para ello, se adoptaron principios de diseño que incluían la creación de componentes especializados, como acciones para la lógica de negocio, consultas separadas en clases específicas, y la utilización de objetos para la transferencia y validación de datos. Se implementó una estructura de directorios modular que facilitó el mantenimiento y escalabilidad del código, apoyándose en el estándar PSR-4 para la carga automática de clases. La metodología incluyó la adopción de estándares de codificación, como PSR-12, y el uso de herramientas de análisis estático y control de versiones para garantizar la calidad y coherencia del proyecto. En términos de resultados, se logró una arquitectura más flexible y comprensible, con un código más limpio y fácil de modificar. Como conclusión, la adopción de una arquitectura modular en Laravel contribuyó significativamente a la mejora de la calidad del código y la escalabilidad de los sistemas desarrollados. Se recomienda continuar con la implementación de buenas prácticas de codificación, así como realizar capacitaciones periódicas para mantener la coherencia en el desarrollo y facilitar la integración de nuevas funcionalidades.

Palabras clave:

Mantenibilidad, arquitectura modular, Laravel, patrón MVC.

1. INTRODUCCIÓN

El patrón Modelo-Vista-Controlador (MVC) ha sido una de las arquitecturas más adoptadas en el desarrollo de aplicaciones web debido a su capacidad para dividir la funcionalidad del software en tres capas principales: Modelo, que gestiona la lógica de datos y las interacciones con la base de datos; Vista, que presenta la interfaz de usuario y los datos al cliente; y Controlador, que actúa como intermediario, coordinando las solicitudes del usuario y facilitando la comunicación entre el modelo y la vista (Velasco-Elizondo et al., 2018). Esta separación busca simplificar el desarrollo y mantenimiento del software, promoviendo una estructura clara y organizada.

Sin embargo, la implementación extensiva del patrón MVC ha expuesto varios desafíos cuando no se aplica correctamente ni se adapta adecuadamente a las necesidades específicas de cada proyecto. Entre los desafíos, destacan controladores sobrecargados con múltiples responsabilidades, lógica compleja en modelos, así como consultas directas a la base de datos desde capas que deberían limitarse a coordinar o presentar información (Aniche, 2016; Velasco-Elizondo et al., 2018). Además, su estructura predeterminada, basada en una separación genérica de responsabilidades para aplicaciones web, carece de un enfoque orientado al dominio particular de cada sistema, lo que puede dificultar su mantenibilidad y escalabilidad (Griffin, 2021, p. 21).

Aunque el patrón MVC proporciona una base adecuada para estructurar sistemas web, su correcta implementación requiere un enfoque riguroso para evitar malas prácticas que puedan afectar la calidad del software, dificultar su mantenimiento o limitar la capacidad de los equipos para escalar y añadir nuevas funcionalidades de forma eficiente.

El presente reporte técnico tiene como objetivo describir y analizar el diseño e implementación de una arquitectura modular en Laravel, un marco de trabajo PHP ampliamente utilizado en la industria por su enfoque orientado a objetos y su versatilidad. Diseñada en 2023 para el desarrollo de aplicaciones web, esta estructura busca superar las limitaciones del patrón MVC mediante la fragmentación de funcionalidades en módulos y componentes con responsabilidades claramente definidas, optimizando así el desarrollo, la prueba y la actualización de sistemas web complejos para mejorar su mantenibilidad, escalabilidad y calidad.

2. METODOLOGÍA

La metodología seguida para el diseño e implementación de una arquitectura modular en Laravel se basó en un enfoque iterativo y sistemático que permitió adaptar, optimizar y superar las limitaciones identificadas en estructuras tradicionales basadas en el patrón Modelo-Vista-Controlador (MVC). El objetivo central fue mejorar la escalabilidad, la cohesión, y la mantenibilidad del código, garantizando al mismo tiempo la calidad del desarrollo mediante la aplicación de buenas prácticas, ampliamente documentadas en la literatura académica y técnica, de arquitectura de software y diseño basado en dominio (Fowler, 2002, Griffin, 2021; Martin, 2018; Visser et al., 2016). Además, se priorizó la mitigación de problemas típicos de las arquitecturas monolíticas asociadas al MVC (Aniche, 2016; Velasco-Elizondo et al., 2018).

El diseño de la nueva estructura modular incluyó un esquema jerárquico que organiza componentes por dominios de negocio, lo que redujo el acoplamiento y aseguró una clara separación de responsabilidades.

Se aplicaron principios de diseño limpio como el de Responsabilidad Única (SRP), que establece que cada clase o módulo debe tener una única razón para cambiar, lo cual mejora la cohesión y la simplicidad, y el principio de Cierre Común (CCP), que propone agrupar en módulos aquellas clases que cambian por las mismas razones, lo que minimiza el impacto de los cambios y facilita el mantenimiento (Martin, 2018, p. 13).

Las decisiones metodológicas estuvieron respaldadas por estudios y bibliografía académica que destacan la necesidad de modularidad y la segmentación de responsabilidades en sistemas complejos (Barde, 2023), validando además el uso de estándares de codificación y herramientas que impactan positivamente en la calidad y escalabilidad del software (Fowler, 2018). Esta fundamentación asegura que cada elección técnica realizada respondió a criterios objetivos y no al azar, maximizando el potencial de desarrollo y mitigando riesgos asociados al acoplamiento y la complejidad.

2.1 DESARROLLO

El desarrollo de la arquitectura modular en Laravel se llevó a cabo mediante la implementación de una serie de componentes y directrices que aseguran la separación de responsabilidades y la modularización del sistema. A continuación, se describen los elementos clave del proceso de desarrollo:

Componentes y su rol en la arquitectura modular

En el diseño de la arquitectura modular, se definieron varios tipos de elementos que cumplen funciones específicas para garantizar una clara separación de responsabilidades, los cuales se presentan en la Tabla 1.

Tabla 1

Tipos de elementos en la arquitectura modular y sus responsabilidades

Tipo de elemento	Responsabilidad
Modelos	Representan las tablas y relaciones de la base de datos, limitándose a operaciones básicas de manipulación de datos
QueryBuilder	Encapsulan consultas avanzadas con la base de datos, manteniendo los modelos simples.
Actions	Contienen la lógica de negocio y representan casos de uso específicos del sistema
Enums	Definen conjuntos de constantes significativas para mejorar la comprensión y mantenibilidad del código
Excepciones	Gestionan los errores de forma independiente del flujo principal
Filters	Implementan la lógica de búsqueda en los listados de información
Componentes Livewire ¹	Permite interacciones reactivas entre la vista y la lógica del sistema, reemplazando los controladores tradicionales y mejorando la experiencia del usuario.
DTOs/FormObjects	Transportan datos entre capas de la aplicación y validan los datos, utilizando spatie/laravel-data ² en Livewire 2 y clases FormObject nativas en Livewire 3.

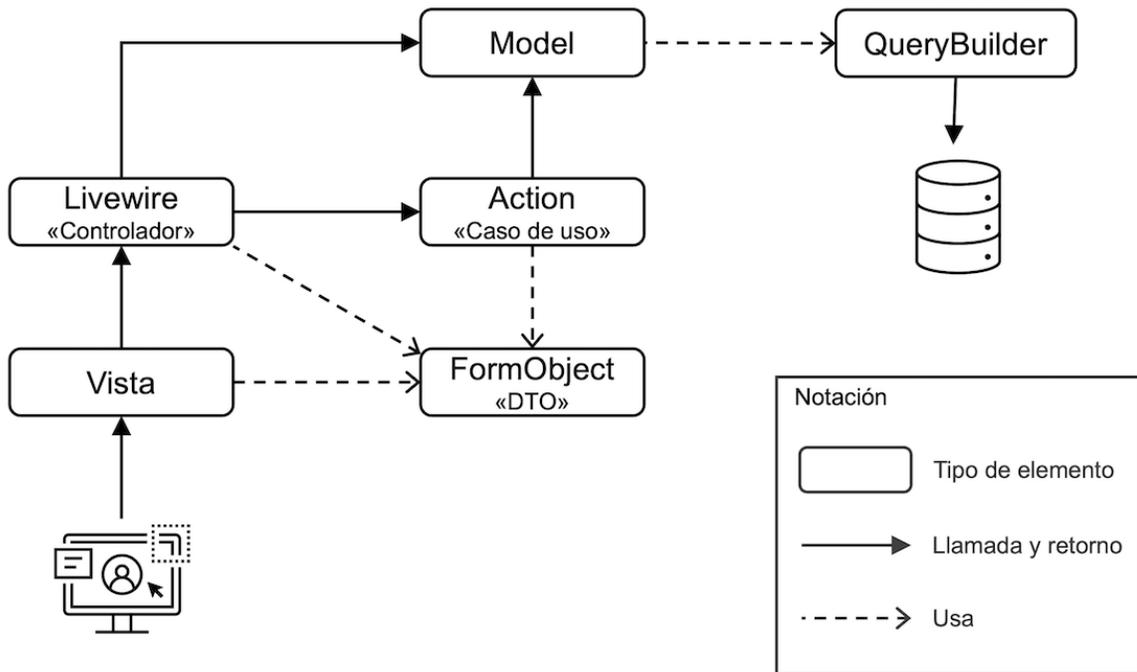
1 <https://livewire.laravel.com/>

2 <https://spatie.be/docs/laravel-data/v4/introduction>

Los diferentes tipos de elementos del sistema se interrelacionan de acuerdo con los flujos de datos definidos en el proceso, tal como se observa en la Figura 1.

Figura 1

Representación conceptual de la arquitectura modular



Nota: La interfaz de usuario recibe las solicitudes del usuario y las envía como eventos al controlador. La transferencia de datos entre la vista, el controlador y la acción se realiza mediante DTOs (Data Transfer Objects), que tienen la responsabilidad de validar que los datos sean correctos. El controlador ejecuta el caso de uso correspondiente, el cual interactúa con otros componentes como modelos, filtros, y query builders, entre otros. Finalmente, el controlador actualiza la vista con el resultado de la acción.

En la Tabla 2, se presentan las responsabilidades típicas en una aplicación web comparando su implementación en un enfoque MVC tradicional y en la arquitectura modular propuesta.

Tabla 2

Comparación de responsabilidades entre MVC tradicional y la arquitectura modular

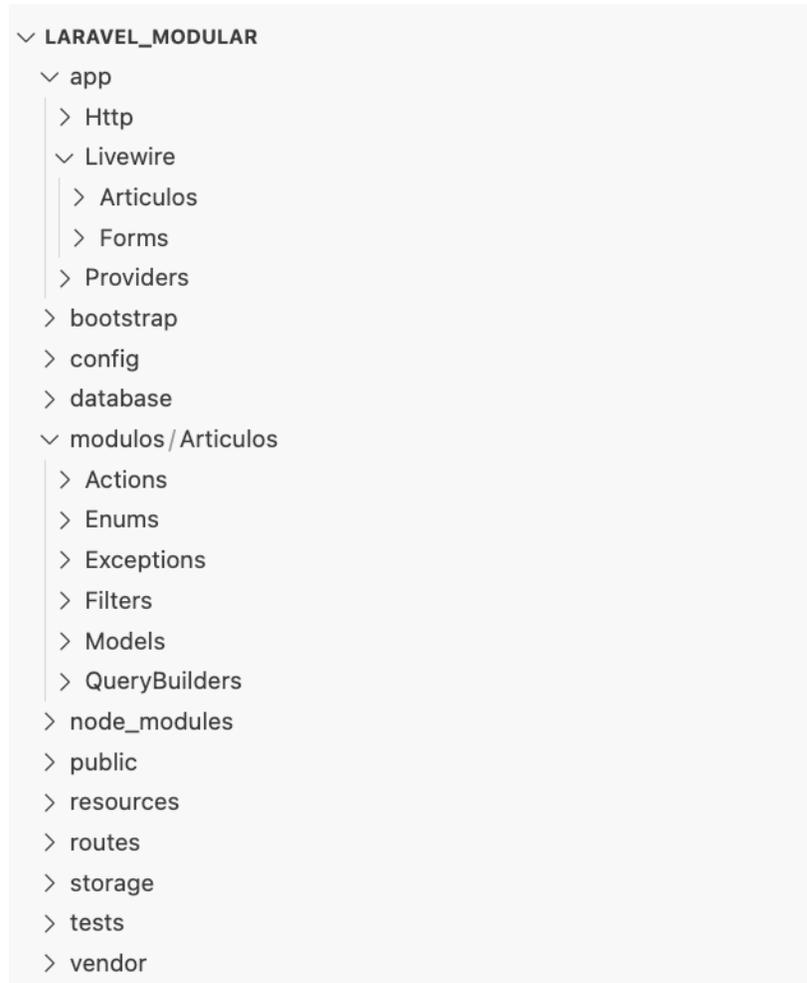
Responsabilidad	MVC tradicional	Arquitectura modular
Validación de datos del usuario	Realizada directamente en el controlador	Delegada a DTOs/FormObjects, que también transportan datos
Transformación o limpieza de datos	Realizada directamente en el controlador	Gestionada por DTOs/FormObjects
Lógica de negocio	Mezclada en el controlador y el modelo	Encapsulada en Actions, representando casos de uso específicos
Interacción con la base de datos	Realizada por el modelo y directo desde un controlador promiscuo	Gestionada por Modelos y consultas avanzadas a través de <i>QueryBuilder</i> s
Lógica de búsqueda en listados	Agregada manualmente en el controlador o modelo	Encapsulada en <i>Filters</i>
Definición de constantes significativas	Definidas como valores literales en el código	Centralizadas en <i>Enums</i> para mejorar la mantenibilidad
Manejo de errores	Captura y manejo directamente en el controlador	Delegada a Excepciones, separando el flujo principal
Generación de respuesta a la vista	Realizada directamente en el controlador	Delegada a Componentes Livewire para mayor interactividad
Actualización reactiva de la interfaz	Requiere lógica explícita en el controlador	Realizada automáticamente mediante Componentes Livewire

Estructuración de la aplicación por módulos

Para fomentar la modularidad, se creó una nueva estructura de directorios que agrupa componentes y funcionalidades en módulos independientes (ver Figura 2). Esta organización permite que cada módulo represente un dominio o conjunto de funcionalidades específicas, mejorando la analizabilidad y mantenibilidad. Al dividir la funcionalidad en múltiples módulos con reglas de acceso bien definidas, se facilita la gestión y el mantenimiento de cada módulo de manera independiente, lo que promueve un desarrollo más ágil y eficiente (Barde, 2023).

Figura 2

Estructura modular de directorios de una aplicación Laravel



Nota: Se añadió un subdirectorio denominado *modulos*, donde se organizaron subdirectorios correspondientes a los módulos definidos en el sistema. En la figura, se muestra un ejemplo utilizando el módulo “Artículos”. Los componentes de Livewire y sus formularios deben organizarse siguiendo la misma estructura por módulos.

La carga automática de clases mediante el estándar PSR-4³ facilitó el registro de nuevos módulos y su vinculación mediante *namespaces*. Este estándar establece un mapeo flexible de *namespaces* a directorios, facilitando la carga automática de clases según su ubicación, lo que simplifica la organización del código y mejora su mantenimiento.

3 <https://www.php-fig.org/psr/psr-4/>

Cumplimiento de estándares y herramientas

Se adoptó el estándar PSR-12⁴ como guía para asegurar consistencia en el estilo de programación, apoyado por herramientas de análisis estático como PHP Insights⁵ para evaluar el código en términos de calidad, complejidad y estilo. PSR-12 extiende las recomendaciones básicas de PSR-1 y PSR-2, proporcionando reglas detalladas sobre formato, indentación, espacios en blanco, declaraciones y uso de *namespaces*, entre otros aspectos, para fomentar un código claro y legible. Para el desarrollo, se utilizó Visual Studio Code con extensiones que automatizan la corrección de violaciones al estándar. Composer y NPM fueron utilizados para manejar dependencias, mientras que Laravel Sail⁶ garantizó entornos locales consistentes con Docker.

Documentación y capacitación

Se realizó un taller de capacitación para los desarrolladores, asegurando la familiarización con la arquitectura propuesta y las tecnologías utilizadas. La documentación del proyecto se mantuvo en una página colaborativa dentro de GitLab, mientras que la documentación de requerimientos y casos de uso fue almacenada en un repositorio documental. Se realizaron reuniones de seguimiento diarias, sesiones para revisión de dudas así como sesiones periódicas de retrospectiva.

Control de versiones y flujo de trabajo

El desarrollo adoptó un flujo basado en ramas, donde cada funcionalidad o corrección se trabajó de forma aislada antes de integrarse al repositorio principal. Las solicitudes de fusión (*merge requests*) facilitaron la revisión del código, para asegurar el cumplimiento tanto de la arquitectura definida como de los estándares establecidos antes de su integración final (ver Figura 3).

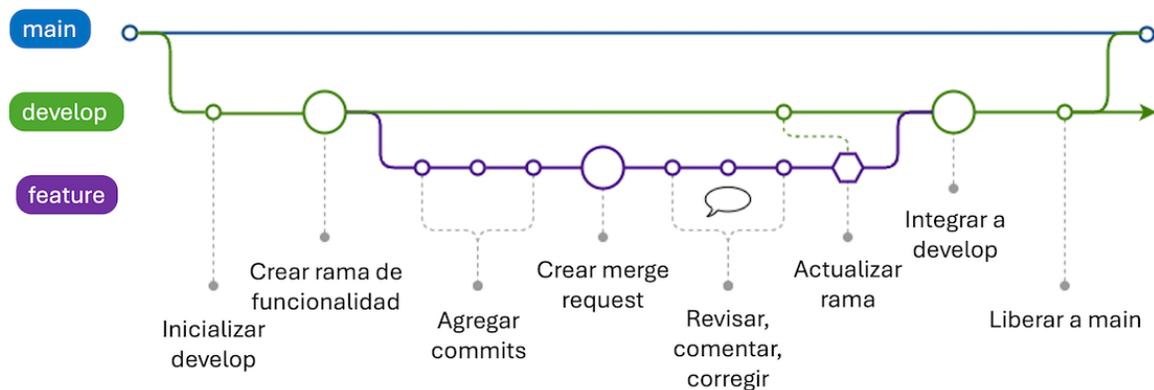
4 <https://www.php-fig.org/psr/psr-12/>

5 <https://phpinsights.com/>

6 <https://laravel.com/docs/11.x/sail>

Figura 3

Flujo de trabajo con control de versiones basado en ramas y merge requests



Nota: En el flujo de trabajo con Git, cada integrante del equipo de desarrollo crea una rama específica para cada funcionalidad que se le asigna. Una vez que completa su desarrollo, genera un merge request para que su trabajo sea revisado. El líder técnico del proyecto, o un comité de revisión designado, se encarga de evaluar los merge requests, proporcionando observaciones o retroalimentación según sea necesario. Cuando la funcionalidad cumple con los criterios establecidos y ha pasado las revisiones, se fusiona en la rama develop. Posteriormente, el líder técnico es responsable de integrar los cambios de la rama develop en la rama main como parte del proceso de liberación, asegurando que los cambios estén listos para su despliegue en el entorno de producción.

3. RESULTADOS

La implementación de la arquitectura modular en Laravel permitió resolver de manera significativa los problemas de escalabilidad y mantenibilidad identificados en proyectos previos basados en el patrón MVC tradicional. Como resultado, se obtuvo una estructura más organizada, con una clara separación de responsabilidades entre los módulos, lo que facilitó la gestión del código y la adición de nuevas funcionalidades sin afectar al sistema en su conjunto. Esta mejora también permitió un proceso de desarrollo más ágil y ordenado, reduciendo la complejidad de mantenimiento y aumentando la capacidad del equipo para implementar cambios de manera eficiente, con una notable disminución de los errores relacionados con dependencias entre módulos.

El enfoque modular evitó problemas frecuentes como controladores excesivamente grandes con múltiples responsabilidades, lo que anteriormente conducía a dificultades de mantenimiento y a código fuente más propenso a errores. Al modularizar los componentes, cada módulo pudo concentrarse en tareas específicas, mejorando la cohesión y reduciendo el acoplamiento. Esto, a su vez, simplificó la revisión y el mantenimiento del código al limitar los puntos de cambio.

Se destacaron, además, algunos hallazgos relevantes que mejoraron la experiencia de los equipos de desarrollo. La modularización del código permitió una mayor facilidad en las pruebas unitarias y en la integración de nuevas tecnologías sin comprometer la estabilidad del sistema. Sin embargo, se identificaron ciertos desafíos relacionados con la curva de aprendizaje de los desarrolladores, quienes

necesitaron tiempo para familiarizarse con la nueva estructura modular y adaptarse a los nuevos flujos de trabajo. Este aspecto incrementó ligeramente el tiempo de integración de la nueva arquitectura en los primeros proyectos implementados. Se llevaron a cabo sesiones de retrospectiva en las que los desarrolladores destacaron que la nueva arquitectura modular no sólo simplificaba el desarrollo de nuevas funcionalidades, sino que también agilizaba la integración de requerimientos adicionales y la implementación de ajustes, reduciendo el esfuerzo necesario y mejorando la productividad general del equipo.

En comparación con enfoques anteriores, como el uso del patrón MVC clásico, los resultados obtenidos confirman que la arquitectura modular no sólo resuelve los problemas iniciales de escalabilidad y mantenimiento, sino que también mejora la flexibilidad y la capacidad de adaptación de los sistemas a nuevos requerimientos y cambios tecnológicos. En este sentido, los resultados obtenidos se alinean con las expectativas iniciales y contribuyen a la mejora continua de los procesos de desarrollo.

Los resultados obtenidos reflejan un avance significativo en la calidad de los sistemas desarrollados, así como en las capacidades del equipo para abordar nuevos desafíos de manera modular, escalable y efectiva. La mejora en la calidad fue validada a través de diversas métricas e indicadores, incluyendo una reducción en el tiempo promedio de desarrollo de nuevas funcionalidades, una disminución en la cantidad y severidad de errores reportados durante la etapa de pruebas, y una mejora en la puntuación obtenida mediante herramientas de análisis estático como PHP Insights. El enfoque adoptado ha generado una base sólida para futuras mejoras, asegurando que el sistema pueda evolucionar en respuesta a las necesidades del negocio y del entorno tecnológico.

4. CONCLUSIONES

El diseño de una arquitectura modular en Laravel adaptada al patrón MVC surgió como una respuesta al problema de la complejidad y baja mantenibilidad que puede presentar una estructura convencional de modelos, vistas y controladores. Los principales retos identificados incluían controladores y modelos con responsabilidades excesivas, lo que complicaba su mantenimiento y actualización, así como la presencia de reglas de negocio dispersas que afectaban la cohesión y escalabilidad del sistema.

La solución propuesta se centró en redefinir las responsabilidades de los componentes clave mediante la creación de módulos independientes y el uso de clases con responsabilidades específicas. Este enfoque resultó en una estructura de código modular, limpia y mantenible, que aumentó la cohesión y redujo el acoplamiento. Adicionalmente, se observó una mejora significativa en la capacidad del sistema para adaptarse a nuevas funcionalidades, gracias a la organización y separación de las responsabilidades. La adopción de estándares de codificación y herramientas de análisis automatizado contribuyó a elevar la calidad del código, mientras que el uso de Livewire permitió una mejor experiencia de usuario mediante interacciones reactivas en la interfaz.

Para continuar con la mejora de la arquitectura y mitigar posibles riesgos, se recomienda mantener una supervisión constante de la modularidad y simplicidad de cada componente, evitando la acumulación de responsabilidades en clases individuales. Además, se sugiere realizar revisiones periódicas del código y capacitaciones para garantizar que todo el equipo esté alineado con los principios de desarrollo adoptados. Implementar métricas de calidad y mantenimiento de código puede facilitar la detección temprana de problemas, mejorando la capacidad de respuesta ante cambios o nuevas necesidades.

Estas acciones, junto con el uso continuo de herramientas de análisis estático, control de versiones y pruebas automatizadas, proporcionarán un marco robusto para mantener la sostenibilidad, escalabilidad y eficiencia del sistema desarrollado.

AGRADECIMIENTOS

Quiero expresar mi sincero agradecimiento a los colaboradores de la Subdirección de Sistemas Integrados y del Departamento de Desarrollo Tecnológico para la Educación por su valiosa participación en la implementación de la nueva arquitectura modular en Laravel. Su dedicación, compromiso y trabajo en equipo fueron clave para el éxito de este proyecto. Aprecio su disposición para aprender y aplicar nuevas metodologías, así como su apoyo en la resolución de desafíos técnicos. Gracias a su esfuerzo conjunto, hemos logrado implementar esta arquitectura modular, que ya se ha utilizado en varios proyectos de desarrollo a la medida durante 2023 y 2024.

REFERENCIAS

- Aniche, M., Bavota, G., Treude, C., van Deursen, A., y Gerosa, M. A. (2016). A validated set of smells in Model-View-Controller architectures. *2016 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 233-243. <https://doi.org/10.1109/ICSME.2016.12>
- Barde, K. (2023). Modular monoliths: Revolutionizing software architecture for efficient payment systems in fintech. *International Journal of Computer Trends and Technology*, 71(10), 20-27. <https://doi.org/10.14445/22312803/IJCTT-V71I10P103>
- Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional.
- Fowler, M. (2018). *Refactoring: Improving the Design of Existing Code* (2nd Edition). Addison-Wesley Professional.
- Griffin, J. (2021). *Domain-Driven Laravel: Learn to implement Domain-Driven Design using Laravel*. Apress.
- Martin, R. (2018). *Clean Architecture: A Craftsman's guide to Software Structure and Design*. Prentice Hall.
- Velasco-Elizondo, P., Castañeda-Calvillo, L., García-Fernández, A., y Vazquez-Reyes, S. (2018). Towards detecting MVC architectural smells. In J. Mejia, M. Muñoz, Á. Rocha, Y. Quiñonez, y J. Calvo-Manzano (Eds.), *Trends and applications in software engineering. CIMPS 2017. Advances in Intelligent Systems and Computing*, vol 688 (pp. 251-260). Springer. https://doi.org/10.1007/978-3-319-69341-5_23
- Visser, J., Rigal, S., van der Leek, R., van Eck, P., y Wijnholds, G. (2016). *Building Maintainable Software, Java Edition*. O'Reilly Media. <https://learning.oreilly.com/library/view/building-maintainable-software/9781491955987/>

Análisis de madurez del área de TIC

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:

Villarreal Brito, G. (2025). Análisis de madurez del área de TIC. *Cuadernos Técnicos Universitarios de la DGTIC*, 3 (18 - 35).

<https://doi.org/10.22201/dgtic.ctud.2025.3.1.84>

Gustavo Villarreal Brito

Instituto de Ciencias del Mar y Limnología
Universidad Nacional Autónoma de México

gvillarreal@cmarl.unam.mx

ORCID: 0009-0006-9119-1122

Resumen

El análisis y la implementación de un marco de referencia, que facilite la obtención de métricas y la formulación de estrategias para el gobierno de Tecnologías de Información y Comunicación, es una tarea estratégica para los profesionales y responsables de este ámbito. Estas acciones permiten contar con datos relevantes y diseñar planes efectivos a corto y mediano plazo, enfocados en la mejora continua de los productos y servicios del área. Como responsables de las TIC, resulta fundamental responder a preguntas claves como: ¿cuál es el estado actual de los servicios e infraestructura de TIC? y ¿qué estrategias se implementarán en los próximos años?

Palabras clave:

Madurez de las TIC, gestión de tecnologías, bases de gobierno de TIC, buenas prácticas, responsables de TIC.

1. INTRODUCCIÓN

Desde los inicios de las Tecnologías de Información y Comunicación (TIC), las organizaciones han buscado estrategias que les permitan rentabilidad y estabilidad en las operaciones y en los procesos de negocio. A lo largo de los años, estos conocimientos y experiencias han madurado y se han creado estándares y buenas prácticas que permiten tener una visión estratégica de las TIC; la productividad de las empresas ha crecido de la mano de la incorporación masiva de las TIC a sus procesos de negocios (Volpentesta, J. R., 2016).

El gobierno de las TIC permite estrategias alineadas con la misión y visión organizacional, aumentando la credibilidad, la rentabilidad, y reduciendo riesgos. Sin embargo, algunas organizaciones carecen de un enfoque claro en el área de las TIC, lo que dificulta un crecimiento y una planificación efectiva. Mediante marcos de referencia, se dirige y controla el uso actual y futuro de las TIC, cubriendo al menos los siguientes aspectos basados en ISACA (ISACA, 2019): alinear las decisiones de las TIC con los objetivos de negocio; mejorar la administración de las inversiones en TIC; gestión de los servicios; el uso adecuado de los recursos de las TIC, su desempeño y la administración del riesgo; garantizar los derechos de propiedad intelectual ; y continuidad y sostenibilidad del negocio.

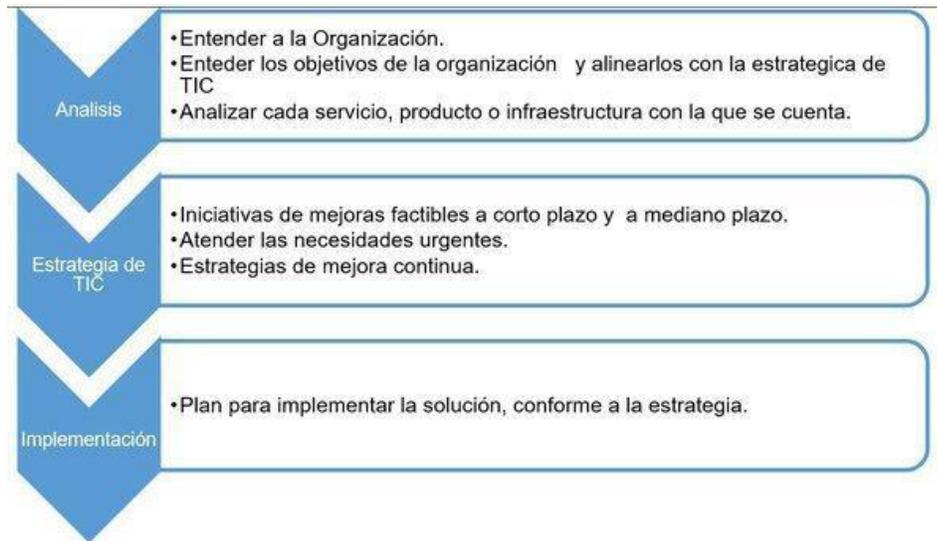
El objetivo es tener un marco de referencia que permita tener una métrica de la madurez de los productos y servicios del área de las TIC de la Unidad Académica de Sistemas Arrecifales de Puerto Morelos (UASAPM) del Instituto de Ciencias del Mar y Limnología (ICMYL) de la UNAM, junto con las bases para iniciar un gobierno de las TIC, que pueda generar una estrategia para cumplir los objetivos de la unidad académica.

2. DESARROLLO TÉCNICO

En esta sección, se mencionan los aspectos más importantes para realizar el análisis y tener información de la situación actual. En la Figura 1, se muestra una estrategia basada en marcos de referencia, o conjunto de estándares y mejores prácticas, como ITIL (*Information Technology Infrastructure Library*). Ésta es un conjunto de prácticas detalladas para la gestión de servicios de Tecnología de la Información (TI), que se enfoca en la alineación de los servicios de TI con las necesidades del negocio; COBIT (*Control Objectives for Information and Related Technologies*) es un marco para el desarrollo, implementación, supervisión y mejora de las prácticas de gobierno de TI; y PMBOK (*Project Management Body of Knowledge*) es un conjunto de estándares y mejores prácticas para la gestión de proyectos, desarrollado por el *Project Management Institute* (PMI).

Figura 1

Estrategia de análisis de madurez de TIC



Antecedentes

En 2019, se asignó a la UASAPM un técnico académico como responsable de las TIC con conocimientos afines al área, quien fue responsable de realizar el análisis de éstas con una visión clara de los objetivos, estructura y operación, considerando factores internos y externos donde la cultura organizacional jugó un papel determinante en el análisis, es decir, los valores, creencias y prácticas que predominan en una organización.

2.1 METODOLOGÍA

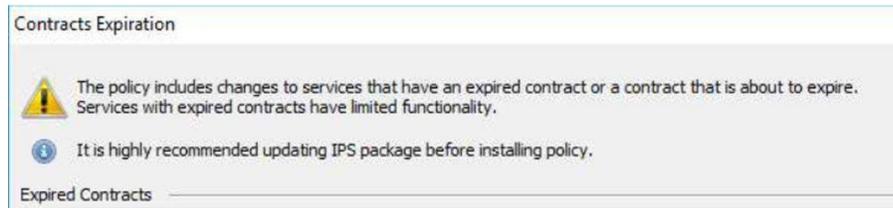
En esta sección, se habla de las áreas de oportunidad en la UASAPM que darán como resultado un marco de referencia. Se documentó y analizó cada hallazgo utilizando las tablas consultables desde el Anexo A hasta Anexo C, adaptadas a las buenas prácticas y estándares.

Gestión de Infraestructura

Este análisis tiene como ámbito las soluciones, infraestructura y procesos, identificando riesgos críticos que puedan afectar la continuidad de los servicios y la integridad de la información, como se muestra en la Figura 2.

Figura 2

Ejemplo de riesgo en seguridad perimetral de la red de datos



Se consideraron riesgos lógicos y físicos, incluyendo inspecciones visuales a toda la infraestructura.

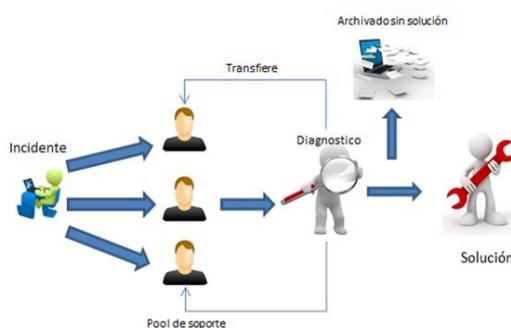
Se analizó la red de datos para asegurar el uso de buenas prácticas y estructura adecuada. Esto implica verificar si la red está segmentada y optimizada para servicios prioritarios, como videoconferencias o voz. De igual manera, pretende evaluar infraestructura como servidores, centros de datos y cuartos de telecomunicaciones, considerando aspectos como la redundancia, la seguridad en los sistemas operativos, los certificados, la capacidad de respuesta ante fallos y las medidas de seguridad física y lógica implementadas. Además, es crucial verificar la disponibilidad de respaldos de energía y sistemas de contingencia para garantizar la continuidad de los servicios críticos.

Atención a requerimientos

En este dominio, se observó la ausencia de métricas sobre tiempos de atención y casos sin solución, también la falta de herramientas que limita la capacidad de evaluar el desempeño y mejorar los servicios de manera continua como se muestra en la Figura 3.

Figura 3

Análisis de proceso de incidentes



Gestión del cambio

Otro de los factores críticos en la gestión de las TIC es el cambio que abarca todas las modificaciones realizadas a las soluciones tecnológicas. Los hallazgos de este dominio se muestran en el Anexo C.

Ponderación de los riesgos

Una vez que se realizó este análisis en los diferentes dominios y fueron documentados dichos hallazgos, se cuantificaron según el impacto en los procesos y operaciones, asignando una ponderación de 0 (no

crítico) a 5 (muy crítico), como se muestra desde el Anexo A hasta el C. La criticidad del área se calcula con la fórmula $(\text{Suma Total Ponderación} / \text{Suma Máxima}) \times 100$, restando una unidad antes de multiplicar por 100; de esta manera, se puede expresar el nivel de madurez del área. Un informe de resultados se presenta en el Anexo D.

3. RESULTADOS

Los puntos analizados en las secciones anteriores, por su naturaleza, se interrelacionan y son codependientes para garantizar servicios de calidad, donde el gobierno de las TIC es uno de los factores que coadyuvan a cumplir con los objetivos de las organizaciones. Es necesario recordar que cualquier análisis de auditoría se debe conceptualizar desde una perspectiva de mejora para identificar puntos relevantes que permitan optimizar y ajustar la operación de los procesos (Vera, H. A. P., 2024).

Como resultado del análisis, el área de TIC de la UASAPM tiene como marco de referencia las cinco directrices para un gobierno de las TIC. En el Anexo G, se pueden observar acciones implementadas que contemplan los siguientes incisos.

- 1. Las soluciones TIC** deben ser claves para apoyar los procesos de la organización, como la investigación, academia, divulgación, difusión o cualquier otro alineado a los objetivos de la UASAPM.
- 2. Cada solución debe tener una Arquitectura Conceptual**, que son las reglas básicas de diseño que deben seguir las soluciones tecnológicas.
- 3. Se debe considerar una arquitectura tecnológica**, donde se definen los modelos y estándares necesarios para construir y mejorar las soluciones.
- 4. Seguridad Informática:** se ponen las directrices que protegen la información de la organización, ayudando a reducir y gestionar los riesgos de seguridad; de esta manera, se inicia con el primer Plan estratégico de seguridad de TIC; se puede observar parte de éste en el Anexo H.
- 5. Evolución Tecnológica:** Garantizar que las soluciones puedan adaptarse y mejorar con el tiempo.

Infraestructura de TIC

De acuerdo al análisis, se atendieron los hallazgos críticos donde las soluciones tienen la capacidad, disponibilidad y adaptabilidad alineados a los objetivos de la organización UASAPM. Como resultado y siguiendo las 5 directrices, se implementan y configuran herramientas para monitorear y gestionar la infraestructura, como se muestra en el Anexo E y Anexo I.

Cómputo y sistemas

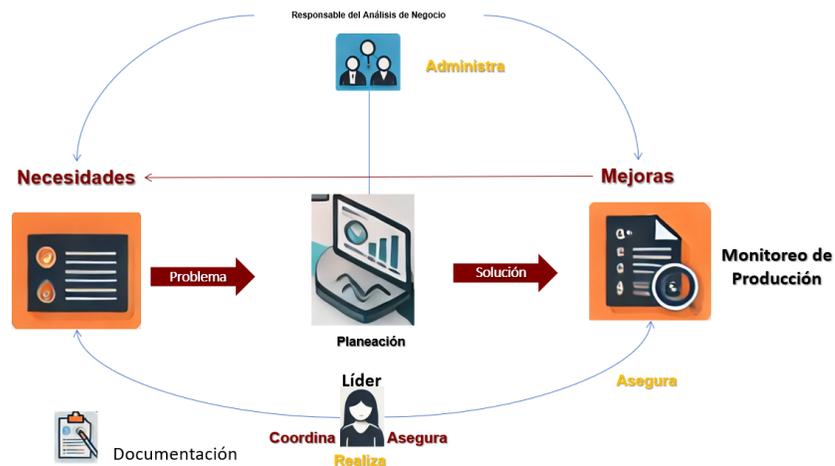
Otro marco de referencia, es que el área de cómputo y sistemas debe contar con personal y proveedores competentes para cumplir con sus funciones. Es esencial que operen de manera ordenada, siguiendo procesos y procedimientos para asegurar la calidad. Además, se inicia la documentación de procesos y sistemas de soporte según su importancia y tiempos de resolución.

Atención a requerimientos

Un factor clave para el éxito de las soluciones con base en las TIC es tener un marco de referencia para la atención de requerimientos, por lo cual, se implementa el siguiente flujo de la Figura 4.

Figura 4

Flujo para la atención de requerimientos



Nota: Basado en *Project Management Body of Knowledge, Project Management Institute (Project Management Institute, 2017)*.

De acuerdo al impacto de las implementaciones, se propone un modelo de atención de requerimientos, el cual se puede consultar en el Anexo F; se detallan los roles y responsabilidades del marco de referencia.

Administración tecnológica

Para la estrategia de los servicios de las TIC y gestionar su ciclo de vida, se inicia con la implementación de los siguientes dominios mostrados en la Figura 5.

Figura 5

Administración de TIC



Administración de Incidentes

En el Anexo I, se observa el proceso para la administración de incidentes, que busca restaurar el funcionamiento normal y minimizar riesgos mediante buenas prácticas alineadas con ITIL e ISO/IEC 20000 y 27001. El éxito de ITIL en las organizaciones se debe principalmente a la actitud y aptitud de las personas, así como al compromiso de la alta gerencia (Casanova, F. S. S. 2021).

- Mesa de Ayuda: Actúa como primer contacto, gestionado por el soporte técnico o TI, donde se registran y clasifican incidentes según prioridad y tipo, facilitando una respuesta adecuada.
- Administración del Cambio: Gestiona modificaciones en los activos TIC, evaluando, autorizando y documentando cambios para reducir riesgos y asegurar la continuidad del servicio.

Administración de Servicios y Recursos

La administración de servicios y recursos TIC, basada en ITIL, estándares ISO/IEC 20000 y 27001, permite una coordinación estratégica y técnica eficaz. Esto incluye un inventario detallado de servicios y recursos (contratos, licencias y garantías) que facilita el control y optimización. Estos elementos permiten a la organización:

- Identificar, ubicar y gestionar recursos y su ciclo de vida.
- Anticipar vencimientos y asegurar licencias.
- Detectar servicios con alta incidencia de problemas y planificar mejoras.
- Minimizar riesgos mediante evaluación de cambios.
- Cumplir con los SLA establecidos.

Estas prácticas fortalecen la capacidad de respuesta y optimizan el uso de las TIC conforme a estándares. Las métricas permiten a los responsables de las TIC justificar decisiones ante los interesados y garantizar el uso eficiente de los recursos (Axelos, 2021). En el Anexo J, se muestran las implementaciones de acuerdo con este marco de referencia.

4. CONCLUSIÓN

El análisis de madurez implementado en la UASAPM estableció un marco de referencia inicial que desencadenó el desarrollo de planes, documentación, soluciones de TIC, entre otras actividades alineadas a este marco de referencia. Aunque requirió una inversión considerable de tiempo y recursos, era esencial realizarlo para iniciar una madurez en el gobierno de las TIC. Con un solo recurso TIC en la unidad académica, el proceso exige mayor esfuerzo, pero debe llevarse a cabo de manera ordenada, cíclica y adaptada a las necesidades, posicionando al área de TIC como un pilar estratégico para la UASAPM.

AGRADECIMIENTOS

Agradecimiento a Lidia Arista Sánchez por su asesoría en PMBOOK, IIBA, SCRUM e ITIL.

Al Ing. Alfredo Landa Herrera y M. EN C. Fernando Negrete Soto por su asesoría y apoyo logístico.

REFERENCIAS

- Axelos. (2019). *ITIL Foundation, ITIL 4 Edition*. TSO (The Stationery Office).
- Axelos. (2021). *ITIL 4 Foundation*. Axelos Limited.
- Casanova, F. S. S. (2021). Implementación de ITIL versión 3 en las organizaciones: Razones del éxito y fracaso. *Revista científica de sistemas e informática*, 1(2), 54-67.
- International Organization for Standardization. (2018a). *ISO/IEC 20000-1: Information technology – Service management – Part 1: Service management system requirements*. ISO.
- International Organization for Standardization. (2018b). *ISO/IEC 27005: Information security risk management*. ISO
- ISACA. (2019). *COBIT 2019 Framework: Governance and Management Objectives*. ISACA.
- Pérez Vera, H. A. (2024). Auditorías internas en ISO 9001 en un área de proyectos de tecnología de información. *Cuadernos Técnicos Universitarios De La DGTIC*, 2(3). <https://doi.org/10.22201/dgtic.ctud.2024.2.3.65>
- Project Management Institute. (2017). *A Guide to the Project Management Body of Knowledge (PMBOK Guide) (6th ed.)*. Project Management Institute.
- Project Management Institute. (2021). *A Guide to the Project Management Body of Knowledge (PMBOK Guide) (7th ed.)*. PMI.
- Volpentesta, J. R. (2016). El impacto de las TIC sobre las estructuras organizacionales y el trabajo del hombre en las empresas. *FACES*, 22(46), 81-94.
- Zabbix. (2024). Acerca de Zabbix. Zabbix Documentación. Recuperado de <https://www.zabbix.com/documentation/current/es/manual/introduction/about>

ANEXO A

Evaluación del ámbito de infraestructura

Este anexo es un ejemplo de posibles hallazgos en el ámbito de infraestructura; el riesgo e impacto dependerá de cada organización. La evaluación de la infraestructura tecnológica considera los principios establecidos en la ISO/IEC 27001 y ISO/IEC 27002, que destacan la importancia de mantener una infraestructura segura y confiable para garantizar la disponibilidad, integridad y confidencialidad de los servicios.

Metodología:

Identificación de componentes clave: Se identificaron los elementos críticos de la infraestructura como switch, cableado, fibra óptica, y centros de datos (Site - MDF).

Evaluación de hallazgos: Cada componente fue revisado en función de las buenas prácticas internacionales y documentado con su respectivo riesgo e impacto.

Ponderación de riesgos: Los riesgos fueron clasificados en una escala del 1 al 5, con base en su impacto potencial en los servicios críticos.

En este caso, se usan tablas o matrices para documentar componentes, hallazgos y su ponderación, que es una práctica fundamentada en la gestión estructurada de riesgos y prioridades, como se muestra a continuación en la Tabla 1.

Tabla 1

Ejemplo de hallazgos ámbito infraestructura

Componente	Hallazgo	Riesgos	Ponderación
Infraestructura de switches	Se mezclan redes, sin una correcta segmentación y sin VLANs.	Esto puede provocar degradaciones en la red, no priorizar servicios como videoconferencias.	4
Cableado	Gran parte del cableado no está certificado y estructurado.	Intermitencias en la red, no existe un etiquetado e identificación de los servicios. Mayor tiempo de resolución de problemas.	4
Fibra del edificio A-B	El cableado en la distribución no se encuentra bien conectado y organizado de acuerdo a buenas prácticas.	Problemas de comunicación, intermitencias y mayor tiempo de resolución de problemas. Puede llegar a la interrupción del servicio.	5
Site -MDF	El cuarto de telecomunicaciones tiene área de oportunidad en la parte eléctrica, sistema de tierras, sistema de respaldo de energía.	Al no estar en condiciones deseables, se acorta el ciclo de vida de los componentes de los equipos.	4

Componente	Hallazgo	Riesgos	Ponderación
Respaldos	No se tiene respaldos de la información de la infraestructura y documentación.	Mayor tiempo de fallas en el servicio y no tener una recuperación rápida.	4
Asignación de IP	Gestión y control en el uso de IPs públicas.	La arquitectura y administración no permite un crecimiento, mayor riesgo de seguridad al tener equipos con IP pública. Negación del servicio a nuevos usuarios.	5
Priorizar servicios	No se priorizan servicios para la salida, compiten los servicios como Netflix y una videoconferencia.	Intermitencias en servicios importantes para la operación y trabajos académicos de la organización.	4
Internet inalámbrico	Los puntos de accesos no cubren la necesidad y operación actual de los usuarios.	Intermitencias en la red, una velocidad baja, interrupción del servicio.	5

Nota: El $=(\text{Suma Total Ponderación}/\text{Suma Máxima}) * 100 = (35/45) * 100$ Nivel de impacto en la infraestructura 77% de impacto para la degradación o interrupción en el servicio.

La ponderación asigna un valor cuantitativo al impacto y riesgo asociado a cada hallazgo, lo que permite evaluar objetivamente su criticidad. Esto se fundamenta en la importancia de identificar, analizar y evaluar los riesgos en función de su probabilidad e impacto, implementado mediante escalas ponderadas (ISO/IEC 27005, 2018b), tanto ITIL como COBIT destacan la necesidad de priorizar las iniciativas de mejora en función de su impacto en los servicios y el valor que aportan a la organización (Axelos, 2019). Los valores numéricos pueden asociarse con niveles cualitativos de impacto o criticidad:

1. Impacto muy bajo
2. Impacto bajo
3. Impacto moderado
4. Impacto alto
5. Impacto crítico

ANEXO B

La Tabla 2 se basa en las mejores prácticas de ITIL, específicamente en los procesos de Gestión de Incidentes y Gestión de Niveles de Servicio. Estos estándares resaltan la importancia de establecer acuerdos claros con los usuarios y contar con herramientas y procesos que garanticen una atención oportuna y eficiente (Axelos, 2019).

Metodología:

Definición de oportunidades: Se identificaron áreas clave como acuerdos de niveles de servicio, escalamiento, gestión de incidentes y herramientas de soporte.

Evaluación de impacto: Cada oportunidad fue ponderada en función de su contribución a la mejora de los tiempos de respuesta y la satisfacción del usuario.

Propuesta de mejoras: Se priorizaron las oportunidades con mayor impacto potencial para implementar soluciones a corto plazo.

Tabla 2

Hallazgos en el ámbito de atención a requerimientos

Componente	Oportunidades	Ponderación
Acuerdos de niveles de servicio	Acuerdos de niveles de servicio con el área usuaria.	4
Matriz de escalamiento	Matriz de escalamiento que permita recuperar en un menor tiempo los servicios.	3
Atención de incidentes	Base de conocimiento.	3
Mesa de ayuda	Un punto de contacto entre el usuario y el área de soporte.	3
Software	Algún sistema que ayude a gestionar, controlar y tener métricas para la atención al servicio.	3

Nota: Esto da una suma 38 puntos, que nos revela un nivel de impacto de 64% = (Suma Total Ponderación/Suma Máxima) * 100 = (16/25) * 100.

Los valores numéricos pueden asociarse con niveles cualitativos de impacto o criticidad:

1. Impacto muy bajo
2. Impacto bajo
3. Impacto moderado
4. Impacto alto
5. Impacto crítico

ANEXO C

Ejemplo de la documentación de hallazgos en el ámbito de gestión del cambio

Para evaluar la gestión del cambio, se utilizó como referencia la norma ISO/IEC 20000, que subraya la necesidad de procedimientos documentados para minimizar riesgos durante los cambios en entornos productivos (ISO, 2018a). Además, se consideraron principios de ITIL en su proceso de Gestión del Cambio, que busca garantizar la calidad y continuidad de los servicios.

Metodología:

Revisión de procedimientos actuales: Se analizaron los procesos existentes para realizar cambios en la configuración y en los servicios.

Identificación de deficiencias: Se documentaron las áreas donde faltan procedimientos, pruebas previas o planes de retorno.

Análisis de impacto: Se ponderaron las oportunidades según el riesgo y la criticidad del cambio en la operación diaria.

Tabla 3

Hallazgos en el ámbito de gestión del cambio

Oportunidades	Ponderación
No existe un procedimiento que regule los cambios en ambientes productivos.	2
Se realizan cambios en la configuración en horarios de operación.	3
No existe un proceso que permita mitigar riesgos.	3
Cuando se realiza un cambio, no se tiene un plan de retorno.	2
No se tiene un proceso previo para liberar un nuevo servicio o realizar un cambio.	3
No se tiene un proceso para pruebas al componente a modificar o liberar.	2

Nota: Esto da una suma de 15 puntos, que nos revela un nivel de impacto 50% = $(\text{Suma Total Ponderación} / \text{Suma Máxima}) * 100 = (15/30) * 100$.

Los valores numéricos pueden asociarse con niveles cualitativos de impacto o criticidad:

1. Impacto muy bajo
2. Impacto bajo
3. Impacto moderado
4. Impacto alto
5. Impacto crítico

ANEXO D

Impacto y riesgo en el dominio

La Figura 6 presenta un ejemplo visual del nivel de impacto y riesgo asociado a tres dominios clave dentro de la gestión de TIC: administración de infraestructura, soporte al servicio y gestión del cambio. Cada indicador utiliza una escala de riesgo que va de bajo a alto, pasando por un nivel medio, permitiendo identificar las áreas que requieren atención prioritaria. En el caso de Administración de Infraestructura, el nivel de riesgo está situado en “Medio-Alto”, lo que refleja posibles deficiencias en la configuración y mantenimiento de los componentes físicos y lógicos de la red. Para Soporte al Servicio, el indicador también apunta a un nivel de riesgo “Medio-Alto”, destacando la necesidad de mejorar los procesos relacionados con la atención de incidentes y requerimientos. Finalmente, el dominio de Gestión del Cambio muestra un nivel de riesgo similar, indicando que la implementación de cambios en entornos productivos puede estar afectada por la falta de procedimientos o estrategias de mitigación.

Figura 6

Impacto y riesgo en el dominio



Esta representación gráfica facilita la evaluación comparativa de riesgos y el enfoque en dominios con mayor impacto potencial, sirviendo como una herramienta efectiva para priorizar acciones correctivas y estratégicas.

ANEXO E

Como parte del resultado del análisis y siguiendo los objetivos planteados, se analizaron diferentes soluciones en el mercado de monitoreo, que nos permitan tener control, operación y gestión de las TIC. Al resultado de este análisis se implementó la solución Zabbix, que es un software que monitorea numerosos parámetros de una red, la salud y la integridad de servidores, máquinas virtuales, aplicaciones, servicios, bases de datos, sitios web y la nube. Zabbix ofrece excelentes informes y funciones de visualización de datos basadas en los datos almacenados. Esto hace que Zabbix sea ideal para la planificación de capacidad (Zabbix, 2024).

La implementación de Zabbix se alinea con las mejores prácticas y estándares reconocidos, específicamente con ITIL (*Information Technology Infrastructure Library*), que ofrece un marco para la gestión eficiente de los servicios de TIC. Esta solución se configuró para que exista un alertamiento vía email.

ANEXO F

Roles y responsabilidades en la Atención a Requerimientos

En el marco de la atención a requerimientos, se identifican roles clave para garantizar el éxito en la implementación de soluciones. Estos roles no son obligatorios y su participación depende del tamaño del requerimiento, de la solución y del nivel de madurez de la organización. Este enfoque está alineado con los principios descritos en el PMBOK (*Project Management Body of Knowledge*) y PMBOK for Business Analysis, los cuales destacan la importancia de definir roles y responsabilidades claras para optimizar la gestión de proyectos (véase Tabla 4).

Tabla 4

Roles y responsabilidades atención a requerimientos

Actor	Responsabilidades
Cliente	Entidades o personas que originan una necesidad que se convierte en un requerimiento o necesidad. Roles: Director General, Coordinador, Jefe de Unidad, Delegado Sindical, Usuario final.
Patrocinador	Validar definiciones, propuestas de solución y el plan del proyecto, incluyendo el alcance del producto y del proyecto. Aprobar los recursos y entregables, además de monitorear el progreso. Tomar decisiones estratégicas para asegurar el logro de los objetivos. Mantener al cliente informado sobre el avance y gestionar decisiones relevantes. Roles: Ejemplo Director General, Coordinador, Jefe de Unidad, Delegado Sindical.
Especialista en Evaluación de Necesidades	Recopilar y analizar requerimientos, asegurando que las expectativas de usuarios y partes interesadas sean claras y bien documentadas. Identificar y comunicar los beneficios esperados del proyecto, definiendo indicadores de éxito específicos y medibles. Realizar un diagnóstico de la situación actual y establecer líneas bases.

Actor	Responsabilidades
Especialista en Evaluación de Necesidades	<p>Facilitar sesiones de trabajo con usuarios e involucrados para explorar posibles alternativas de solución.</p> <p>Preparar y presentar las alternativas de solución de manera estructurada para la revisión y aprobación del patrocinador.</p> <p>Colaborar con el equipo de proyecto para asegurar que los requerimientos se traduzcan correctamente en especificaciones técnicas, sin involucrarse en la ejecución.</p> <p>Revisar y actualizar los requerimientos en caso de cambios.</p>
Líder del Proyecto	<p>Liderar el equipo de trabajo.</p> <p>Gestionar actividades en las fases de inicio, planificación, ejecución, control y cierre.</p> <p>Coordinar y validar entregables para su aprobación.</p> <p>Comunicar información clave y mantener la alineación de todos los involucrados</p>
Líder de Solución	<p>Aclarar las expectativas de los involucrados sobre la solución.</p> <p>Coordinar la planificación, ejecución y cierre de la solución aprobada, abarcando tecnología, procesos y operaciones.</p> <p>Validar que la solución cumpla con todos los requerimientos establecidos.</p> <p>Colaborar con líderes de áreas tecnológicas y reportar el avance al líder de proyecto.</p>
Involucrados	<p>Personas, normativas, leyes, internos o externos que tienen algún interés o participación en el proyecto.</p> <p>Aportan información, participan en la toma de decisiones y pueden verse afectados o beneficiados por el requerimiento.</p>
Equipo de Trabajo	<p>Conformado por los miembros asignados a los equipos bajo la coordinación de los líderes de solución.</p> <p>Ejecutan actividades específicas según sus roles, contribuyendo al logro de los objetivos del requerimiento.</p>

ANEXO G

Como resultado del análisis, se inicia con procedimientos y formatos para la atención de requerimientos, donde se contemplan los cinco aspectos descritos en la sección de resultados, alineados a marcos de referencia internacionales como ISO/IEC 20000 para la gestión de servicios de TI y el modelo de gestión de proyectos del PMBOK.

Alineados al marco de referencia descrito en la sección de resultados, estos documentos contemplan los siguientes puntos.

- Título de la necesidad
- Objetivo
- Fecha
- Historial de Edición del Documento
- Distribución del Documento Final
- Introducción
- Antecedentes
- Premisas
- Alternativas de solución: pueden ser una o varias.
- Diagramas conceptuales o arquitectura
- Gestión de la calidad
- Gestión de incidentes y riesgos
- Entregables
- Metodología
- Obligaciones y responsabilidades
- Cronograma Macro
- Monitoreo y control
- Comunicación
- Sección de firmas

Esta estructura contribuye a la estandarización y mejora continua en la atención de requerimientos, asegurando que se cumplan con las expectativas de calidad y las necesidades específicas de los usuarios.

ANEXO H

Plan estratégico de seguridad de TIC

Como parte del análisis y propuesta en la sección de resultados, se inicia con el primer plan estratégico de seguridad de las TIC de la Unidad Académica, con las buenas prácticas de ISO/IEC 27001 para la gestión de seguridad de la información, estándar ISO/IEC 27002. Ésta enfatiza la formación como una medida clave de mitigación de riesgos, ISO/IEC 19770-1 para la gestión de activos de software, lineamientos de ANSI/TIA-942 para la infraestructura de telecomunicaciones en centros de datos, metodologías como el OWASP Application Security Verification Standard (ASVS), personalizadas para la madurez inicial de nuestra entidad que engloban los siguientes puntos:

- Seguridad Informática
- Conciencia de seguridad de la información
- Protección de los derechos de autor para software
- Seguridad en el centro de datos
- Controles de acceso
- Aplicaciones web, escritorio y app móvil.
- Mejora continua.

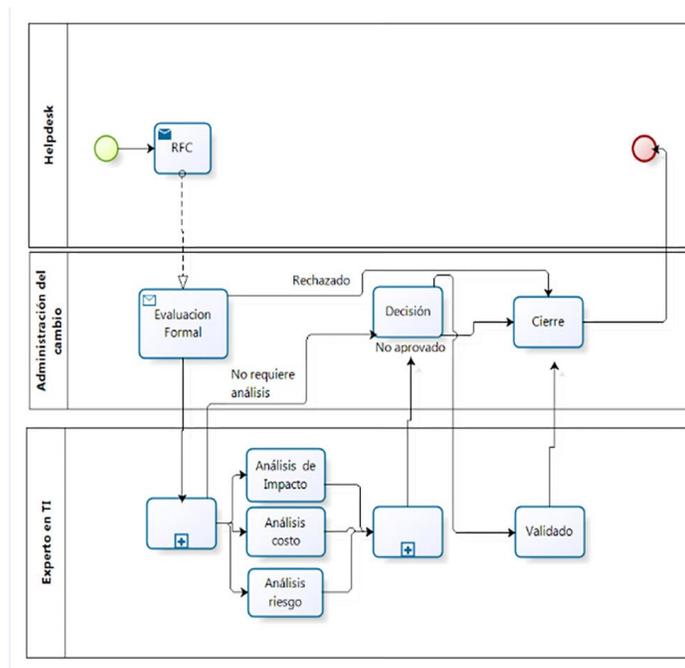
ANEXO I

Atención de incidentes o solicitudes

Con la finalidad de recuperar el nivel habitual de funcionamiento del servicio y minimizar riesgos e impactos para la UASAPM, se implementan procesos apoyados con la solución CMDBuild. Esta implementación se alinea con las mejores prácticas y estándares internacionales de gestión de servicios de TI, como ITIL y ISO/IEC 20000, que establecen marcos claros para garantizar la eficiencia y calidad en la entrega de servicios tecnológicos como se muestra en la Figura 7.

Figura 7

Diagrama lógico del proceso de atención de incidentes o solicitudes



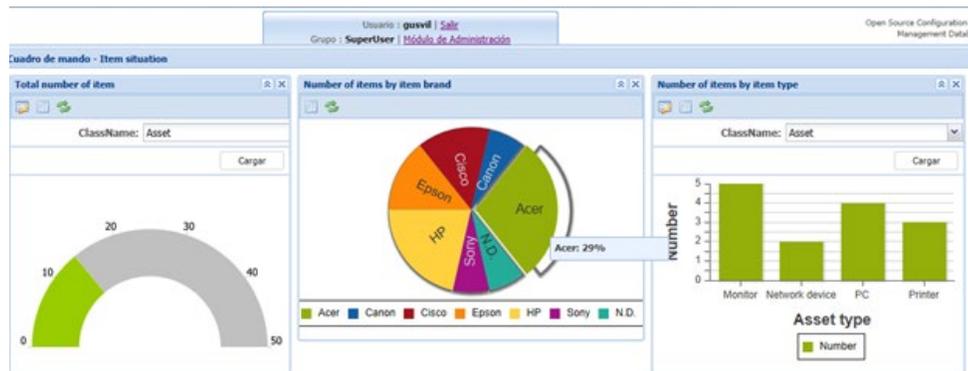
ANEXO J

Recursos de TIC de la UASAPM

CMDBuild actúa como una **base de datos de configuración** centralizada, permitiendo documentar y gestionar los recursos y sus relaciones a lo largo de su ciclo de vida. Según ITIL, este proceso es fundamental para mantener una visibilidad completa de los activos tecnológicos y su impacto en los servicios de TI (Axelos, 2019). Además, esta práctica está respaldada por **ISO/IEC 20000**, que establece la gestión de la configuración como un requisito clave para la calidad en la administración de servicios de TI (véase Figura 8).

Figura 8

Reporte de recursos de TIC de la UASAPM



La solución CMDBuild no sólo facilita la gestión de incidentes y solicitudes, sino que también contribuye al fortalecimiento de la continuidad del servicio y la resiliencia operativa, alineándose con los estándares más reconocidos en la gestión de servicios y riesgos.

Reordenamiento dinámico de elementos usando *drag & drop* en Laravel-11 y Livewire-3

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:

Ochoa Rodríguez, J. A. (2025). Reordenamiento dinámico de elementos usando drag & drop en Laravel-11 y Livewire-3. *Cuadernos Técnicos Universitarios de la DGTIC*, 3 (1) páginas (36 - 50).

<https://doi.org/10.22201/dgtic.ctud.2025.3.1.100>

José Antonio Ochoa Rodríguez

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

jantonio.ochoarz@gmail.com

ORCID: 0009-0006-1559-2346

Resumen

Cada categoría y subcategoría de software publicada en el "Portal de Software UNAM" es mostrada al usuario conforme al parámetro "orden_tienda". Este valor se define previamente en el Módulo de Gestión de Categorías y Subcategorías de Software de la herramienta web en desarrollo denominada "Censo TIC". Durante el desarrollo de dicho módulo, se implementó un control de interfaz de usuario (*widget*) que, con sólo arrastrar y soltar elementos de una lista con el puntero del ratón (*widget drag & drop*), permite actualizar de manera automática el valor de "orden_tienda" dentro de la base de datos; Esto resultó en una interfaz interactiva, fácil de usar y eficiente, que brinda la posibilidad de una menor cantidad de clics, un menor desplazamiento entre distintas pantallas para obtener datos y visualizar toda la información relacionada entre sí de cada categoría y sus subcategorías. El empleo de este tipo de controles permite una mejor experiencia de usuario (*UX*), enfocada en todo momento en la facilidad de uso, la eficiencia, la utilidad, la satisfacción y la emoción que el usuario experimenta durante la interacción con la interfaz web.

Palabras clave:

Ordenamiento dinámico, lista, *drag & drop*, Laravel, Livewire.

1. INTRODUCCIÓN

La Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) de la UNAM puso a disposición de la comunidad universitaria el “Portal de Software de la UNAM”, sitio web mediante el cual, entre otras actividades, se difunde, para su descarga e instalación, el software libre y el software adquirido con convenio para apoyar a las funciones sustantivas.

Actualmente, los responsables de TIC de la UNAM requieren del apoyo del personal de la DGTIC para gestionar los elementos publicados en dicho portal, lo que conlleva a tiempos de espera largos, dependencia técnica y velocidad de respuesta poco oportuna. Por tal motivo, la DGTIC inició el desarrollo del componente de software denominado “Censo TIC”, plataforma web modular desarrollada con el *framework* de PHP Laravel 11, Livewire 3 y TailwindCSS, que permitirá a los responsables de TIC realizar dichas tareas de forma automatizada y en tiempo real.

Al tratarse de una herramienta modular en desarrollo, fue preciso iniciar con la codificación de los componentes necesarios para los módulos de gestión de “Categorías de software” y “Subcategorías de software”. Si bien el prototipo recibido señalaba la creación de módulos independientes, se decidió crear un único componente de software denominado “Módulo de Categorías de Software”, que brinda al usuario final una única interfaz web integral que permite el registro, modificación, listado, consulta, activación, desactivación y reordenamiento de aparición “orden_tienda” en el “Portal de Software UNAM” de las categorías y subcategorías de software, de una manera simple, interactiva, coherente y de rápido aprendizaje.

El reto por resolver fue la creación de un control de interfaz de usuario (*widget*) que, con sólo arrastrar y soltar elementos de una lista con el puntero del ratón (*widget drag & drop*), permitiera actualizar de manera automática el valor de “orden_tienda” dentro de la base de datos, ya sea para una categoría o para una subcategoría contra sus similares pertenecientes a la misma categoría. Esto evita al usuario tener que navegar entre distintas pantallas, el uso de controles de formulario poco intuitivos y hacer varios clics hasta obtener el “orden_tienda” deseado.

El objetivo del trabajo presentado a continuación es el describir el proceso de implementación de un componente de software que permite reordenar los elementos de un listado y actualizar automáticamente el valor dentro de la base de datos con sólo arrastrar y soltar elementos con el puntero del ratón (*widget drag & drop*).

2. DESARROLLO TÉCNICO

Para el desarrollo del módulo, se empleó el repositorio “Censo TIC” creado por la DGTIC, que incluye un contenedor en Docker con los servicios:

- Apache: software que procesa y entrega archivos de sitios web a los usuarios desde un navegador.
- PHP: lenguaje de programación empleado para el desarrollo web y que se ejecuta del lado del servidor.
- Composer: manejador de paquetes para PHP que proporciona un estándar para administrar, descargar e instalar dependencias y librerías.

- Laravel 11: marco o esquema de trabajo de PHP para el desarrollo de software y que simplifica el trabajo repetitivo.
- Livewire 3 para Laravel: herramienta que permite crear aplicaciones interactivas con componentes Javascript para Laravel.
- TailwindCSS: marco o esquema de trabajo de CSS para aplicar estilos a interfaces de usuario HTML.
- PostgreSQL: sistema manejador de bases de datos relacional.

Barajas González (2023) señala que “el uso de contenedores para desarrollo de software facilita al programador la instalación de las herramientas necesarias para aplicaciones cuya arquitectura requiere diversos servicios adicionales. También facilita trabajar sobre sistemas legados porque encapsula los archivos binarios y librerías que estos necesitan para funcionar, en un entorno independiente al sistema operativo que lo contiene” y que “Los contenedores solucionan la incompatibilidad entre ambientes de programación en proyectos con más de un programador ya que pueden crear contenedores con las mismas características, independientemente del sistema operativo o configuraciones propias de cada uno de sus equipos de cómputo”.

De acuerdo con Adriano Escudero y Guapi Cuji (2024), “teniendo en cuenta que en el desarrollo web de buscar más eficiencia, confiabilidad, mantenibilidad y escalabilidad, se necesita de la ayuda de un *framework* que sea una parte esencial del conjunto de herramientas de desarrollo de aplicaciones web que utilizan el lenguaje PHP. Varios *framework* de PHP han demostrado ser grandes complementos del desarrollo de software con código a prueba de fallas y extensible, lo que da como resultado aplicaciones web más sólidas y seguras.”

Por otra parte, Sánchez Montes de Oca (2024) indica que “Laravel es un *framework* de desarrollo web de código abierto escrito en PHP que se utiliza para crear aplicaciones web de alta calidad y escalables, siendo uno de los *frameworks* de PHP más populares”.

Barroso Benítez, Trujillo Casanola y Millet Lombida (2021) señalan que:

En el desarrollo de software se reconoce la usabilidad como atributo de calidad para el éxito de un producto, no obstante, son pocos los procesos y profesionales que la aplican (Castilla, Hernández y González, 2017). La usabilidad constituye un factor esencial en el desarrollo de los sistemas informáticos (Ferré, 2018), si estos no son percibidos como una herramienta que ayuda al usuario a realizar sus tareas, se dificulta en gran manera la aceptación del mismo. En los últimos años se ha identificado el concepto de “Experiencia de Usuario” (UX por sus siglas en inglés). La UX está tomando mayor importancia a nivel mundial y cada vez más organizaciones se preocupan por las emociones, percepciones y respuestas que tendrán los usuarios cuando interactúen con sus productos y servicios.

En conclusión, la combinación de tecnologías seleccionadas permite desarrollar un sistema técnicamente robusto y sólido, que facilita el trabajo de los desarrolladores y asegura una experiencia de usuario positiva, respondiendo a las demandas actuales de eficiencia, seguridad y usabilidad en los sistemas informáticos.

2.1 METODOLOGÍA

2.1.1 ANÁLISIS

El modelo de entidad-relación de la base de datos indicó las siguientes entidades:

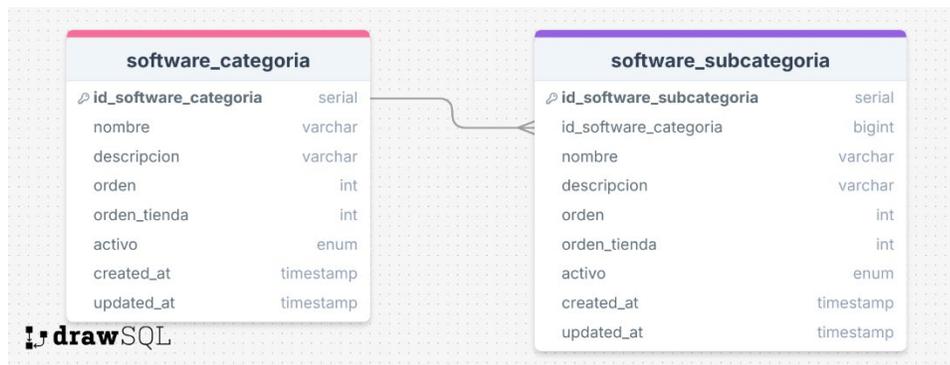
- “software_categoria” para almacenar las categorías de software, con los atributos: id_software_categoria, nombre, descripción, orden, orden_tienda, activo created_at y updated_at.
- “software_subcategoria” para almacenar las subcategorías de software, con los atributos: id_software_subcategoria, id_software_categoria, nombre, descripción, orden, orden_tienda, activo created_at y updated_at.

Podemos observar que el modelo define una relación de tipo “uno a muchos (1:M)”; es decir, que una software_categoria CONTIENE una o muchas software_subcategoria y, a su vez, una software_subcategoria PERTENECE a una software_categoria.

Como se ha definido anteriormente, el reto por resolver fue la creación de un control de interfaz de usuario (*widget*) que, con sólo arrastrar y soltar elementos de una lista con el puntero del ratón (*widget drag & drop*), permitiera actualizar de manera automática el valor de “orden_tienda” dentro de la base de datos, ya sea para una categoría o para una subcategoría contra sus similares pertenecientes a la misma categoría.

Figura 1

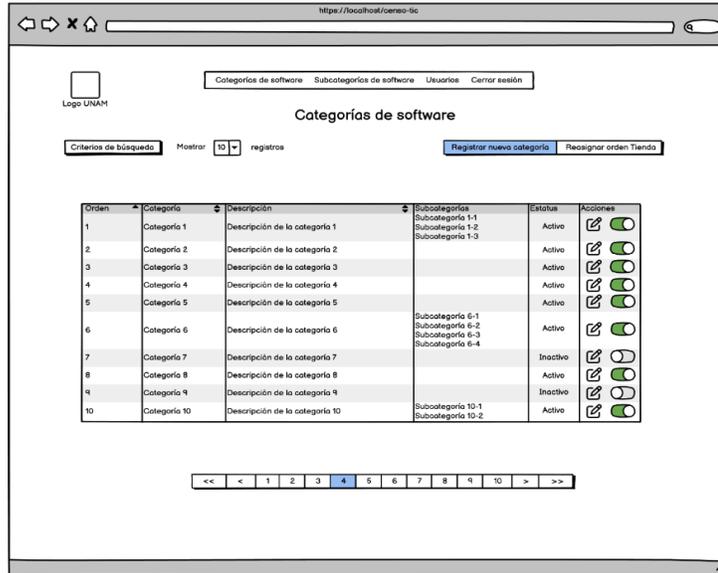
Modelo Entidad-Relación para categorías y subcategorías de software



La documentación del proyecto definió un prototipo que separa en componentes distintos la gestión de software_categoria y software_subcategoria; es decir, “Módulo de Categorías” y “Módulo de Subcategorías”; cada uno de ellos permitiendo el registro, modificación, listado, consulta, paginación, activación, desactivación y reordenamiento de aparición “orden_tienda”.

Figura 2

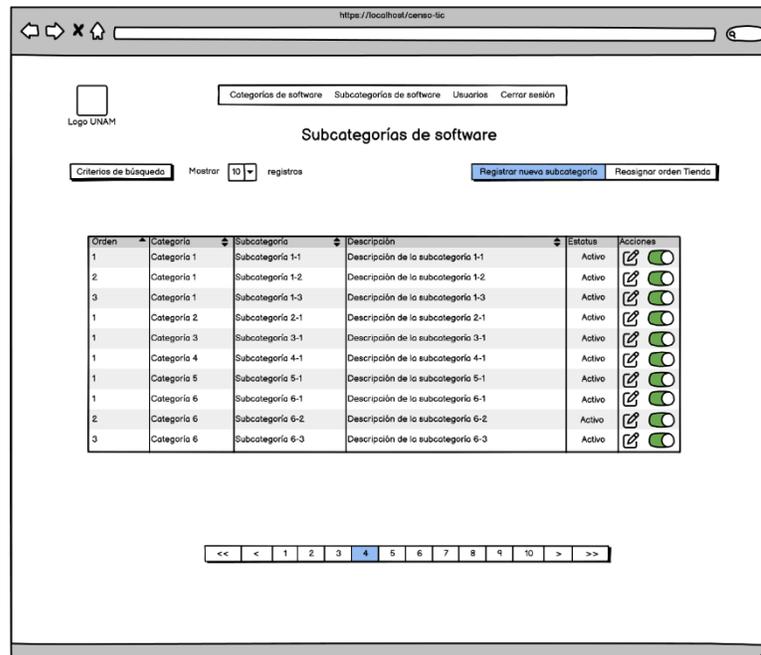
Prototipo inicial para el componente de listado de categorías de software



Y subcategorías de software:

Figura 3

Prototipo inicial para el componente de listado de subcategorías de software



2.1.2 DISEÑO

Se propusieron las siguientes interfaces de usuario para el listado y reordenamiento de categorías de software y sus subcategorías:

Figura 4

Prototipo final para el componente de listado de categorías de software

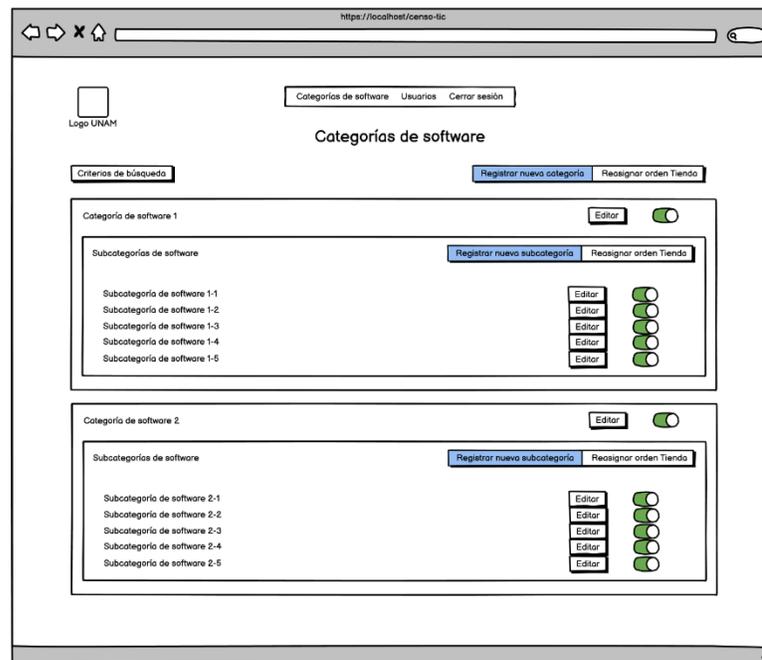


Figura 5

Prototipo final para actualizar el orden de categorías de software de manera dinámica

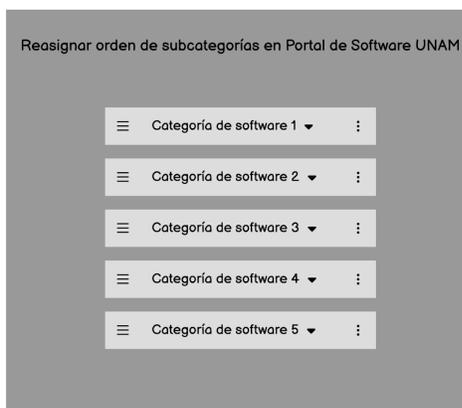
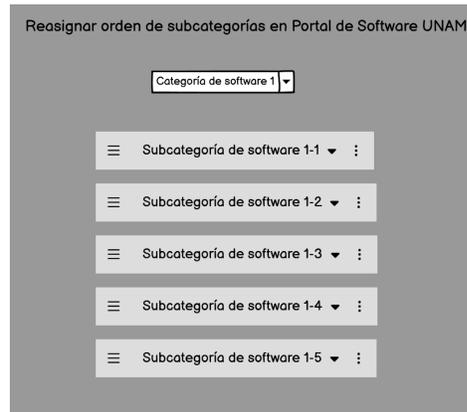


Figura 6

Prototipo final para actualizar el orden de subcategorías de software de manera dinámica



2.1.3 IMPLEMENTACIÓN

Para implementar la funcionalidad de reordenamiento dinámico de categorías y subcategorías de software mediante un *widget* de arrastrar y soltar (*drag & drop*), que además actualiza automáticamente los valores en la base de datos, se pueden observar los pasos de la implementación en los anexos A y B correspondientes.

2.1.4 PRUEBAS

Se comprobó que las vistas se representan adecuadamente en el navegador, que las acciones de los botones se ejecutan y muestran las ventanas modales esperadas, que al arrastrar un elemento del listado de categorías o subcategorías se mueve conforme se manipula el ratón y que, al soltar el elemento, se dispara la acción sobre la base de datos. Se verificaron los nuevos valores de “*tienda_orden*” en la tabla correspondiente de la base de datos.

2.1.5 MANTENIMIENTO

Al tener bien organizados todos los elementos de ambos componentes, se facilita el mantenimiento y la expansión del Módulo de Categorías y Subcategorías de Software a lo largo del tiempo; sin embargo, será necesario adaptar a los cambios que exijan las nuevas versiones de Laravel y Livewire.

3. RESULTADOS

Al finalizar el módulo de gestión de “Categorías y Subcategorías de software”, se entregó una única interfaz web que integra las acciones de registro, modificación, listado, consulta, activación, desactivación y reordenamiento de aparición en el Portal de Software UNAM. Asimismo, se creó un componente que permite el reordenamiento de aparición de las categorías y subcategorías de software en dicho portal, mediante una interfaz sencilla, simple, interactiva, coherente y de rápido aprendizaje. Esto se

logra con sólo arrastrar y soltar elementos con el puntero del ratón (*widget drag & drop*) para actualizar automáticamente el valor de “orden_tienda” dentro de la base de datos.

4. CONCLUSIONES

El desarrollo de componentes de software con diseño responsivo, atractivo, interactivo y, sobre todo, funcional permite mejorar la experiencia de usuario y un rápido aprendizaje. Al crear una interfaz que integra las acciones necesarias sobre las categorías y subcategorías de software, se evita el exceso de clics y se logra un menor desplazamiento entre distintas pantallas para obtener información y mantener el enfoque respecto a toda la información relacionada entre sí.

Asimismo, al sustituir los controles numéricos para definir el reordenamiento de aparición de las categorías y subcategorías de software en el Portal de Software UNAM, que implicaría capturar un valor numérico más un clic en el botón de “guardar”, por una interfaz que permita arrastrar y soltar elementos de una lista con el puntero del ratón (*widget drag&drop*) para actualizar automáticamente el valor de “orden_tienda” dentro de la base de datos, permitimos que dicha experiencia de usuario sea más atractiva, fluida e interactiva.

AGRADECIMIENTOS

A la Doctora Ana Yuri Ramírez Molina, Titular de la Dirección; al Maestro Hugo Alonso Reyes Herrera, Titular de la Subdirección de Sistemas Integrados y a la Licenciada Karla Fonseca Márquez, Titular del Departamento de Integración de Soluciones de Software. Todos ellos de la Dirección de Colaboración y Vinculación dependiente de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC). Gracias por brindarme la oportunidad de cumplir uno de mis más grandes sueños: servir profesionalmente a nuestra máxima casa de estudios; por su enorme apoyo, su guía y orientación; por ser el modelo a seguir, de quienes admiro todo su compromiso y esfuerzo; por su liderazgo, que me lleva a esforzarme por alcanzar mi verdadero potencial.

REFERENCIAS

- Adriano Escudero, W. & Guapi Cuji, D.(2023) Influencia del código abierto y su simplicidad en el desarrollo de sistemas web académicos. RCTU [online].vol.10, n.1 [citado 2024-11-15], pp.10-18. Recuperado de: http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S1390-76972023000100010&lng=es&nrm=iso ISSN 1390-7697. <https://doi.org/10.26423/rctu.v10i1.730>.
- Barajas González, L. D. (2023). Uso de contenedores para la construcción de productos de software. *Cuadernos Técnicos Universitarios de la DGTIC*,1(1), pp.72-81. <https://doi.org/10.22201/dgtic.ctud.2023.1.1.15>
- Barroso Benítez, Y., Trujillo Casanola, Y. & Millet Lombida, Y. (2021). Marco de trabajo de evaluación de experiencia de usuario en el desarrollo de software. *Revista cubana de Ciencias Informáticas* [online]. vol.15, n.3 [citado 2024-11-15], pp. 92-117. Recuperado de: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992021000300092&lng=es&nrm=iso Epub 01-Sep-2021. ISSN 2227-1899.
- Castilla R, L., Hernández G, D. & González P, Y. (2017). De la arquitectura de información a la experiencia de usuario: Su interrelación en el desarrollo de software de la Universidad de las Ciencias Informáticas.

E-Ciencias de la Información. Vol. 7, 1, pp.155-176.

Laravel (2011 - 2024). Recuperado de <https://laravel.com/docs/11.x>

Laravel - Artisan Console. (2011 – 2024) Recuperado de <https://laravel.com/docs/11.x/artisan>

Manual de PHP. (2001-2024). Recuperado de <https://www.php.net/manual/es/>

Sánchez Montes de Oca, P. Z. (2024). Automatización de tareas para desarrolladores de software en Laravel, por medio de comandos personalizados. *Cuadernos Técnicos Universitarios de la DGTIC*, 2 (3), pp.40-49. <https://doi.org/10.22201/dgtic.ctud.2024.2.3.61>

ANEXO A

Categorías de software

Para implementar la funcionalidad de reordenamiento dinámico de categorías de software mediante un *widget* de arrastrar y soltar (*drag & drop*), que además actualiza automáticamente los valores en la base de datos, se llevaron a cabo los siguientes pasos:

1. Se generó el componente Livewire y la vista Blade con el siguiente comando:
`php artisan make:livewire ReasignarOrdenTiendaSoftwareCategoríasComponent`
2. Dentro del componente, se generó el método `actualizarOrdenTiendaSoftwareCategorías()`, responsable de recibir el arreglo de categorías de software con el nuevo orden:

Figura 7

Código fuente del método `actualizarOrdenTiendaSoftwareCategorías()`

```
// app/Livewire/SoftwareCategorías/ReasignarOrdenTiendaSoftwareCategoríasComponent.php
public function actualizarOrdenTiendaSoftwareCategorías($softwareCategorías)
{
    ReasignarOrdenTiendaSoftwareCategoríasAction::reasignarOrdenTiendaSoftwareCategorías($softwareCategorías, Auth::id());
    $mensaje = 'software_categorías.orden-tienda.exito';
    $this->success($mensaje);
}
```

3. Luego, se creó manualmente el archivo responsable de realizar la operación dentro de la base de datos: `ReasignarOrdenTiendaSoftwareCategoríasAction.php` y, dentro de la transacción a la base de datos, se definió el código que recorre el arreglo de categorías de software con el nuevo orden y ajusta el valor dentro de la base de datos:

Figura 8

Código fuente de la acción *ReasignarOrdenTiendaSoftwareCategoriasAction()*

```
// Modulos/SoftwareCategorias/Actions
class ReasignarOrdenTiendaSoftwareCategoriasAction
{
    public static function reasignarOrdenTiendaSoftwareCategorias($softwareCategorias, int
$idUsuario)
    {
        try {
            --
            foreach ($softwareCategorias as $softwareCategoria) {
                SoftwareCategoria::updateOrCreate(
                    [
                        'id_software_categoria' => $softwareCategoria['value'],
                    ],
                    [
                        'orden_tienda' => $softwareCategoria['order'],
                    ]
                );
            }
        } catch (Exception $e) {
            Log::error($e::class . ' > ' . $e->getFile() . '(' . $e->getLine() . '): ' .
            $e->getMessage());
            throw $e;
        }
    }
}
```

4. Se generó el modelo *SoftwareCategoria* empleando el comando:

```
php artisan make:model SoftwareCategoria
```

Por default, Laravel 11 generó el archivo *SoftwareCategoria.php* en la ruta *app/Models* y se reubicó manualmente en *modulos/SoftwareCategorias/Models*.

5. Después, se modificó el código fuente de la vista en Blade:

Figura 9

Código fuente de la vista *reasignar-orden-tienda-software-categorias.blade.php*

```

<!--
resources/views/livewire/software-categorias/reasignar-orden-tienda-software-categorias.blade.php
-->

<div>
    <x-dialog-modal wire:model="modalAbierto">
        <x-slot name="title">
            Reasignar orden de aparición en Portal de Software UNAM
        </x-slot>

        <x-slot name="content">
            <div class="overflow-x-auto">
                <div class="w-96 mx-auto">
                    <ul wire:sortable="actualizarOrdenTiendaSoftwareCategorias">
                        @foreach ($this->softwareCategorias as $softwareCategoria)
                            <li wire:sortable.item="{{ $softwareCategoria->id_software_categoria
                        }}" wire:key="categoria-{{ $softwareCategoria->id_software_categoria }}" class="flex
                        justify-between border rounded mb-3 overflow-hidden">

                                <span wire:sortable.handle class="grid place-content-center
                                bg-stone-50 hover:bg-sky-50 px-3 cursor-move rounded-lg">
                                    <i class="fa-solid fa-grip-vertical"></i>
                                </span>
                                <span class="w-full flex justify-between items-center pl-2">
                                    <strong>{{ $softwareCategoria->nombre }}</strong>
                                    
                                </span>
                            </li>
                        @endforeach
                    </ul>
                </div>
            </div>
        </x-slot>

        <x-slot name="footer">
            <x-primary-button type="button" class="mr-3"
            wire:click="cerrarModalReasignarOrdenTienda" target="guardar">
                Actualizar orden
            </x-primary-button>
            <x-secondary-button type="button" class="" wire:click="$toggle('modalAbierto')"
            wire:loading.attr="disabled">Cerrar</x-secondary-button>
        </x-slot>
    </x-dialog-modal>

```

6. Por último, se modificó el código fuente del botón "Reasignar orden en Portal de Software UNAM" de las categorías, definido en la vista Blade *listar-software-categorias.blade.php*:

Figura 10

Código fuente del botón “Reasignar orden en Portal de Software UNAM” en la vista `listar-software-categorias.blade.php`

```
<!-- resources/views/livewire/software-categorias/listar-software-categorias.blade.php -->
@if (count($this->softwareCategorias) > 0)
    @can('reasignar-orden-tienda-software-categorias')
        <div class="flex justify-end">
            <x-action-button class="bg-indigo-600"
wire:click="$dispatch('abrir-modal-reasignar-orden-tienda-software-categorias');"
data-tippy="Reasignar Orden en Portal de software UNAM">
                <i class="fa-solid fa-list-squares"></i>
            </x-action-button>
        </div>
    @endcan
@endif
```

El evento “clic” del botón ejecuta “`abrir-modal-reasignar-orden-tienda-software-categorias`” para mostrar la ventana modal con el listado de Categorías de Software. Al arrastrar y soltar algún elemento del listado, se llama al evento `actualizarOrdenTiendaSoftwareCategorias`, que realiza la actualización en la base de datos de manera dinámica.

ANEXO B

Subcategorías de software

Para implementar la funcionalidad de reordenamiento dinámico de subcategorías de software mediante un *widget* de arrastrar y soltar (*drag & drop*), que además actualiza automáticamente los valores en la base de datos, se llevaron a cabo los siguientes pasos:

1. Se generó el componente Livewire y la vista Blade con el siguiente comando:

```
php artisan make:livewire ReasignarOrdenTiendaSoftwareSubcategoríasComponent
```

2. Dentro del componente, se generó el método `actualizarOrdenTiendaSoftwareSubcategorias()`, responsable de recibir el arreglo de categorías de software con el nuevo orden:

Figura 11

Código fuente del método `actualizarOrdenTiendaSoftwareSubcategorias()`

```
public function actualizarOrdenTiendaSoftwareSubcategorias($softwareSubcategorias)
{
    ReasignarOrdenTiendaSoftwareSubcategoriasAction::reasignarOrdenTiendaSoftwareSubcategorias($softwareSubcategorias, Auth::id());
    $mensaje = 'software_subcategorias_orden-tienda_exito';
    $this->success($mensaje);
}
```

3. Luego, se creó manualmente el archivo responsable de realizar la operación dentro de la base de datos: `ReasignarOrdenTiendaSoftwareSubcategoriasAction.php`, dentro de la transacción a la base

de datos, se definió el código que recorre el arreglo de categorías de software con el nuevo orden y ajusta el valor dentro de la base de datos:

Figura 12

Código fuente de la acción *ReasignarOrdenTiendaSoftwareCategoriasAction()*

```
// Modulos/SoftwareSubcategorias/Actions
class ReasignarOrdenTiendaSoftwareSubcategoriasAction
{
    public static function reasignarOrdenTiendaSoftwareSubcategorias($softwareSubcategorias, int
$idUsuario)
    {
        try {
            --
            foreach ($softwareSubcategorias as $softwareSubcategoria) {
                SoftwareSubcategoria::updateOrCreate(
                    [
                        'id_software_subcategoria' => $softwareSubcategoria['value'],
                    ],
                    [
                        'orden_tienda' => $softwareSubcategoria['order'],
                    ]
                );

                //Guardado en bitácora
                Bitacora::registrar($idAccion, $idUsuario, $softwareSubcategoria['value'],
                RegistroTipoEnum::SoftwareSubcategoria);
            }
        } catch (Exception $e) {
            Log::error($e::class . ' > ' . $e->getFile() . '('.$e->getLine().'): ' .
            $e->getMessage());
            throw $e;
        }
    }
}
```

4. Se generó el modelo *SoftwareSubcategoria* empleando el comando:

```
php artisan make:model SoftwareSubcategoria
```

Por default, Laravel 11 generó el archivo *SoftwareSubcategoria.php* en la ruta *app/Models* y se reubicó manualmente en *modulos/SoftwareSubcategorias/Models*.

5. Después, se modificó el código fuente de la vista en Blade

Figura 13

Código fuente de la vista *reasignar-orden-tienda-software-subcategorias.blade.php*

```

<!--
resources/views/livewire/software-subcategorias/reasignar-orden-tienda-software-subcategorias.blade.php -->

<div>
  <x-dialog-modal wire:model="modalAbierto">
    <x-slot name="title">
      Reasignar orden de aparición en Portal de Software UNAM
    </x-slot>

    <x-slot name="content">
      <div class="overflow-x-auto">
        <div class="mb-3 grid grid-cols-1 gap-2 w-96 mx-auto">
          <div class="flex flex-col">
            <x-input-label for="formRegistrarSoftwareSubcategoriaCategoria">Categoría
            de software <span class="obligatorio">*</span></x-input-label>
            <select class="w-full h-auto" name="rol">
              <option value="">-- Seleccione una categoría --</option>
              @foreach ($this->softwareCategorias as $softwareCategoria)
                <option value="{{ $softwareCategoria->id_software_categoria }}">
                  {{ ucfirst($softwareCategoria->nombre) }}
                </option>
              @endforeach
            </select>
            @error('formRegistrarSoftwareSubcategoria.id_software_categoria') <span
            class="error">{{ $message }}</span> @enderror
          </div>
          <div class="mb-3 grid grid-cols-1 gap-2 w-9/12 mx-auto">
            <ul wire:sortable="actualizarOrdenTiendaSoftwareSubcategorias">
              @foreach ($this->softwareSubcategorias as $softwareSubcategoria)
                <li wire:sortable.item="{{ $softwareSubcategoria->id_software_subcategoria }}"
                wire:key="categoria-{{ $softwareSubcategoria->id_software_subcategoria }}"
                class="flex justify-between border rounded mb-5">
                  <span wire:sortable.handle class="grid place-content-center
                  bg-stone-50 hover:bg-sky-50 px-3 cursor-move">
                    <i class="fa-solid fa-grip-vertical"></i>
                  </span>
                  <span class="w-full p-2 flex justify-between py-5">
                    {{ $softwareSubcategoria->nombre }}
                  </span>
                </li>
              @endforeach
            </ul>
          </div>
        </div>
      </x-slot>

      <x-slot name="footer">
        <x-primary-button type="button" class="mr-3"
        wire:click="cerrarModalReasignarOrdenTienda" target="guardar">
          Actualizar orden
        </x-primary-button>
        <x-secondary-button type="button" class="" wire:click="$toggle('modalAbierto')">
        wire:loading.attr="disabled">Cerrar</x-secondary-button>
      </x-slot>
    </div>
  </x-dialog-modal>

```

6. Por último, se modificó el código fuente del botón "Reasignar orden en Portal de Software UNAM" de las categorías, definido en la vista Blade *listar-software-categorias.blade.php*:

Figura 14

Código fuente del botón "Reasignar orden en Portal de Software UNAM" en la vista `listar-software-categorias.blade.php`

```

<!-- resources/views/livewire/software-categorias/listar-software-categorias.blade.php -->

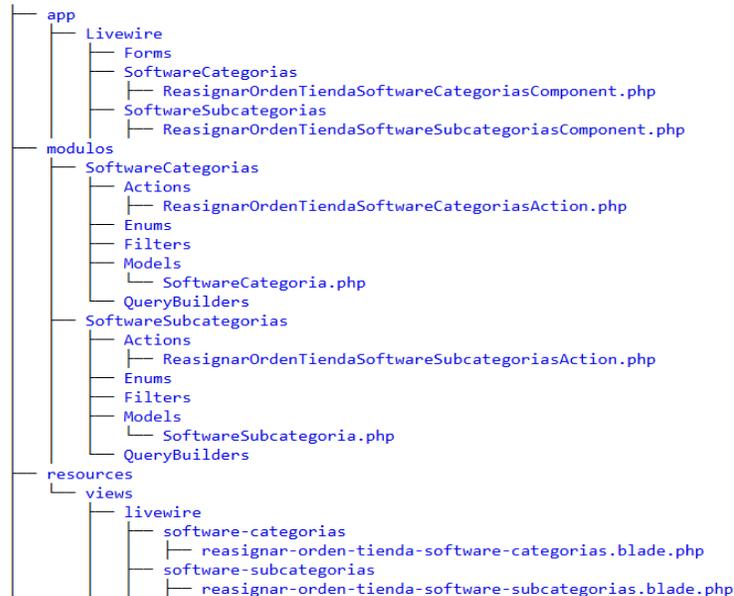
<div class="flex flex-row justify-center items-center">
  @can('registrar-software-subcategoria')
    <x-action-button class="bg-sky-600" title="Actualizar subcategoria de software"
    data-tippy="Actualizar subcategoria de software"
    wire:click="$dispatch('abrir-modal-registrar-software-subcategoria', { idSoftwareCategoria: {{
    $softwareCategoria->id_software_categoria }}, idSoftwareSubcategoria: {{
    $softwareSubcategoria->id_software_subcategoria }} }}">
      <i class="fa-solid fa-pen-to-square"></i>
    </x-action-button>
  @endcan
</div>
  
```

El evento "clic" del botón, ejecuta "abrir-modal-reasignar-orden-tienda-software-subcategoria" para mostrar la ventana modal con el listado de Subcategorías de Software pertenecientes a la misma categoría. Al arrastrar y soltar algún elemento del listado, se llama al evento `actualizarOrdenTiendaSoftwareSubcategorias`, que realiza la actualización en la base de datos de manera dinámica.

De manera general, la estructura de directorios y archivos de PHP necesarios son:

Figura 15

Estructura de directorios y archivos Módulo de Categorías y Subcategorías de Software Laravel 11 y Livewire 3



Análisis de la información de tarificación telefónica con Power BI®

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:

Gallardo Valadez, I. G. (2025). Análisis de la información de tarificación telefónica con Power BI®. *Cuadernos Técnicos Universitarios de la DGTIC*, 3 (1) páginas(51 - 59).

<https://doi.org/10.22201/dgtic.ctud.2025.3.1.92>

Isabel Georgina Gallardo Valadez

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

artisa@unam.mx

ORCID: 0009-0009-9818-1383

Resumen

En la gestión de los servicios de comunicación, es importante examinar su uso; el análisis del tráfico telefónico en la Red Universitaria de Voz de la UNAM permite conocer los patrones de comportamiento de las diversas opciones que se ofrecen a la comunidad universitaria para optimizar su utilización y reducir el gasto. Para mostrar el resultado del análisis de manera ágil y significativa, se emplearon objetos visuales utilizando Power BI® como herramienta de inteligencia de negocios a través de la construcción sencilla y rápida de gráficas que facilitan la comprensión de la información y contribuyen a la toma de decisiones.

Palabras clave:

Tarificación, tráfico telefónico, telefonía, objetos visuales.

1. INTRODUCCIÓN

En el mundo actual, impera la necesidad de conectividad en todo lugar y momento, por lo que el acceso a los servicios de comunicación en sus diversas modalidades se ha vuelto indispensable. Dentro de la Universidad Nacional Autónoma de México (UNAM), se cuenta con una variedad de opciones para solventarlo. La Red UNAM ofrece a la comunidad universitaria acceso a internet para los equipos de cómputo, tabletas, teléfonos inteligentes y otros dispositivos a través de conexiones alámbricas e inalámbricas; para el caso de la comunicación oral, ésta proporciona, a través de la red institucional de voz, miles de extensiones telefónicas que interconectan a los campus universitarios.

En la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC), se realiza la facturación de los servicios de telefonía y de conexión a internet con el objetivo de conocer e informar sobre los consumos efectuados y analizar la información para identificar patrones de uso que permitan optimizar su empleo y control presupuestal. En los servicios de telefonía, se revisan continuamente los datos de tarificación que se generan en cada llamada telefónica realizada en la red. De igual forma, se analizan las facturas emitidas por los proveedores de los servicios, ya que para las organizaciones “Analizar datos hoy en día es una tarea sumamente relevante por cuanto es un activo de gran valor”, tal y como lo menciona Arriagada-Benítez (2020, p.556).

Dentro del marco de la gestión de las Tecnologías de Información, la entrega de valor a las organizaciones es estratégica y generalmente se relaciona con el aumento de ingresos y el incremento o mejora de la satisfacción del cliente. Sin embargo, en las tareas de facturación y control del gasto, es igualmente importante la reducción de los costos asociados a la prestación de servicios y, con ello, las posibilidades de ahorro suelen ser atractivas (Quezada-Sarmiento, et al., 2018).

El uso de inteligencia de negocios o *Business Intelligence* (BI) es aplicada por las organizaciones para manejar, analizar y transformar “grandes volúmenes de datos en información significativa y útil para la toma de decisiones estratégicas y operativas” (Alvarado-Apodaca et al., 2023, p.77). En este particular, el empleo de una herramienta de inteligencia de negocios permite el manejo de una gran cantidad de registros telefónicos de forma anual, mostrando los resultados mediante objetos visuales concretos y significativos que muestren el comportamiento del tráfico de la red de voz de la UNAM.

El objetivo del presente reporte es emplear Power BI® como herramienta de Inteligencia de Negocios para el análisis y visualización de grandes volúmenes de datos en la identificación de patrones de comportamiento del tráfico telefónico.

2. DESARROLLO TÉCNICO

El tráfico de voz se compone de todas las llamadas efectuadas por medio de los dispositivos telefónicos que se encuentran instalados en las áreas universitarias, con lo que se ofrece el servicio de comunicación dentro o fuera de la misma dependencia, ya sea en el mismo campus universitario o en otra ciudad a través de la red telefónica universitaria. Así mismo, se realizan llamadas a cualquier número telefónico fijo o móvil en cualquier parte del mundo usando la red de los proveedores de servicios telefónicos.

Las categorías de tráfico son: interno (llamadas realizadas entre extensiones de la Red Universitaria de Voz) y externo (llamadas que emplean el servicio del proveedor). Derivado de lo anterior, las llamadas

telefónicas se dividen en los siguientes tipos: locales, de larga distancia nacional e internacional; a teléfonos móviles e internas. Esta información se puede analizar por tipos de llamadas, por períodos de tiempo, por áreas universitarias o por una extensión específica, entre otras opciones.

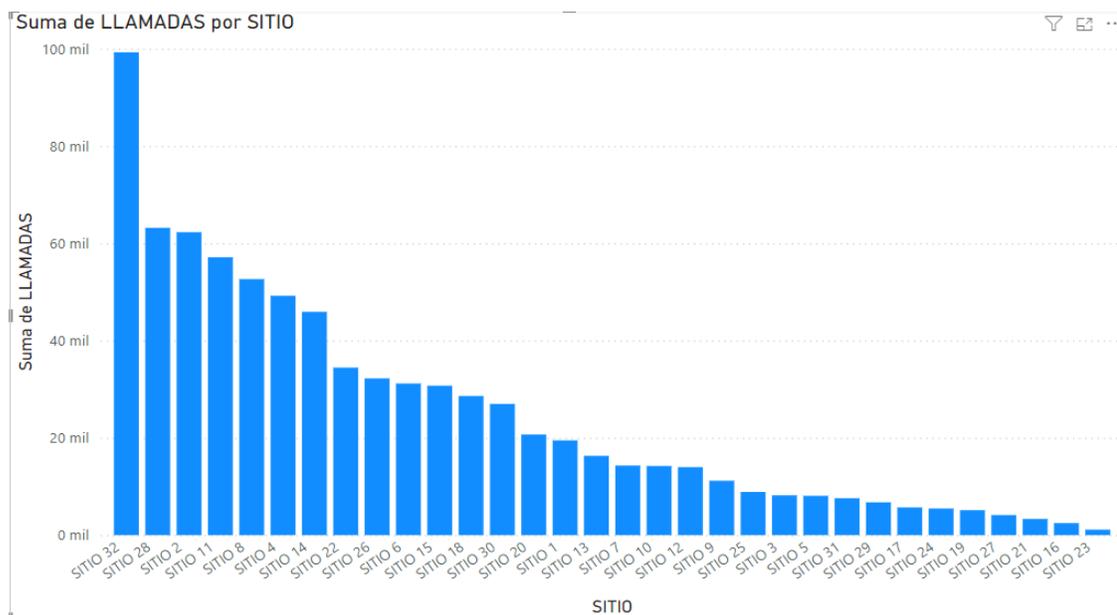
En el análisis, se eligió a Power BI de Microsoft® como plataforma de inteligencia de negocios, ya que cumple con las principales características requeridas como escalabilidad, compatibilidad, manejo de grandes volúmenes de datos, conectividad, facilidad de uso y capacitación; además, puede utilizar datos de diversas fuentes como Excel, empleado ampliamente en el área de tarificación, y ofrece gran variedad de opciones para construir de forma tanto ágil como sencilla informes accesibles e interactivos con objetos visuales representativos de la información, favoreciendo la toma de decisiones (Microsoft®, 2024). Debido a su potencial, Power BI® es también empleado como una herramienta para la ciencia de datos, procesa datos complejos para encontrar patrones y opciones de visualización que, como señalan Lemus-Delgado y Pérez, su "... propósito es facilitar la comprensión de la información de manera clara y propiciar su socialización" (2020, p. 46), ya que permite compartir los resultados en cualquier tipo de dispositivo, incluyendo teléfonos móviles.

En la elaboración de las gráficas, se reunió la información del tráfico, proporcionado por el proveedor de servicios telefónicos a la UNAM, del año 2023, así como los datos del tarificador telefónico, para lo que se crearon y ejecutaron reportes específicos de los años 2021 al 2023.

A continuación, se presenta el comportamiento del tráfico telefónico de la red. En la Figura 1, la gráfica muestra el volumen total de tráfico durante el año 2023, dividido en función de los 32 sitios de conmutación que gestionan las llamadas en la red para conectarlas con el dispositivo telefónico asignado al número marcado; como se puede observar, el sitio con mayor tráfico es el Sitio 32. Esta información puede emplearla para redistribuir el tráfico entre sitios, con la finalidad de mejorar la eficiencia de la red.

Figura 1

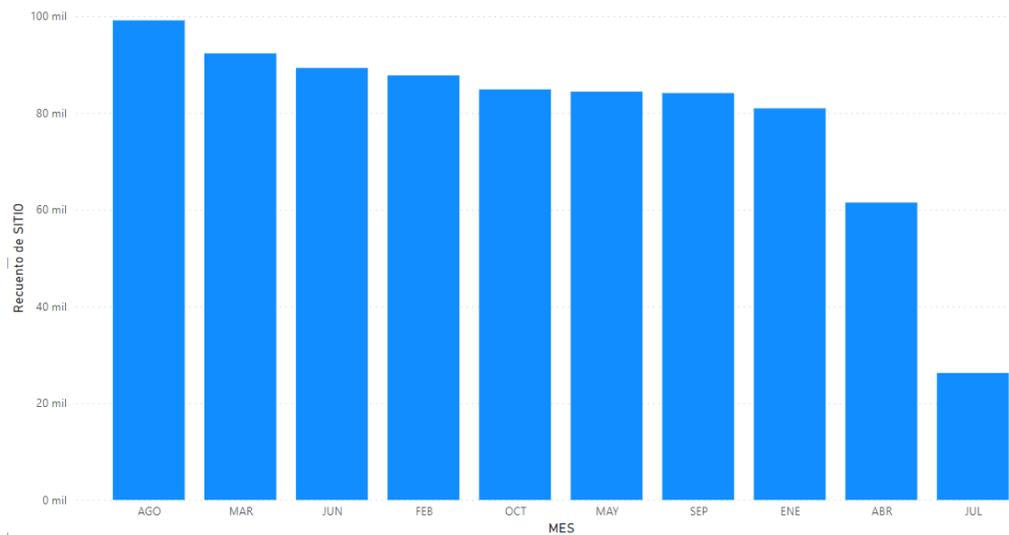
Tráfico de llamadas total por sitio



En la Figura 2, se ofrece una representación mensual del tráfico. Se observa que agosto es el mes con mayor volumen de llamadas y, en contraparte, julio tiene el menor tráfico. Esta información permite conocer el comportamiento anual de la red y es utilizada para planificar sesiones de mantenimiento del equipo tarificador en los meses de menor tráfico (diciembre y julio) e inclusive, ser usada para la Red Universitaria de Voz

Figura 2

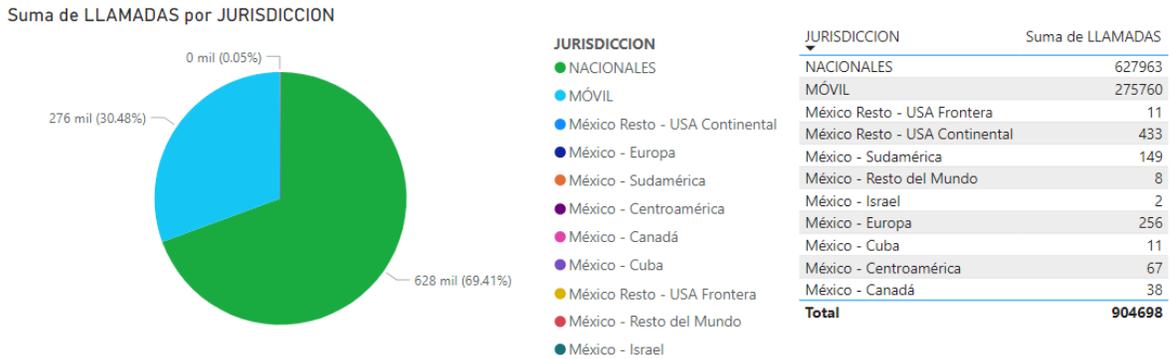
Tráfico mensual



Para ampliar el conocimiento sobre el comportamiento del tráfico, en la Figura 3, compuesta por una gráfica y una tabla de datos, la primera muestra la proporción entre los diferentes tipos de llamadas: en color verde, están las nacionales, que incluyen larga distancia en el país, y las locales efectuadas dentro del área metropolitana, con el mayor porcentaje de 69.41%; en color azul claro, se representan las llamadas a teléfonos móviles con 30.48%; y, en menor proporción, las de larga distancia internacional con 0.05%. Del lado derecho, se encuentra una tabla con el número exacto de llamadas por cada rubro con el fin de complementar la información de la gráfica.

Figura 3

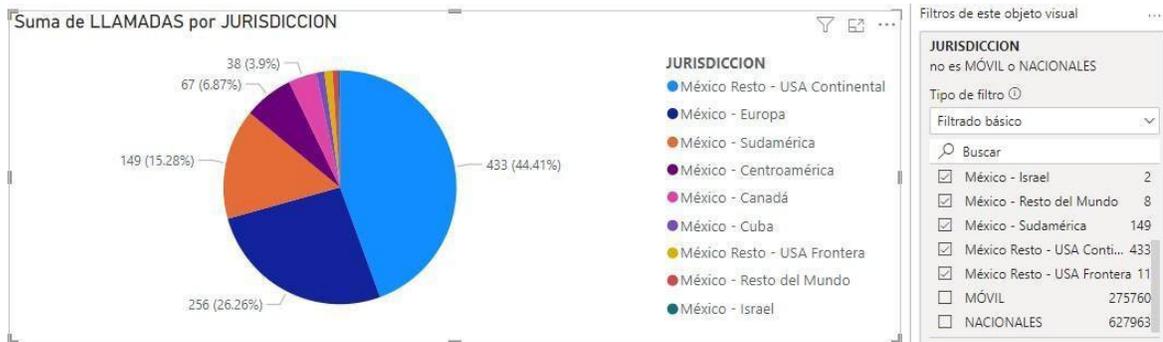
Tráfico por tipo de llamadas



Como se puede apreciar en la gráfica anterior, el detalle de las llamadas internacionales no es perceptible. Para observar mejor esta información, se aplicó simplemente un filtro, activando las regiones de larga distancia y sin activar las otras; obsérvese la Figura 4, en donde el objeto visual ofrece una vista ampliada del segmento que no se apreciaba claramente en la anterior.

Figura 4

Tráfico de Larga Distancia Internacional



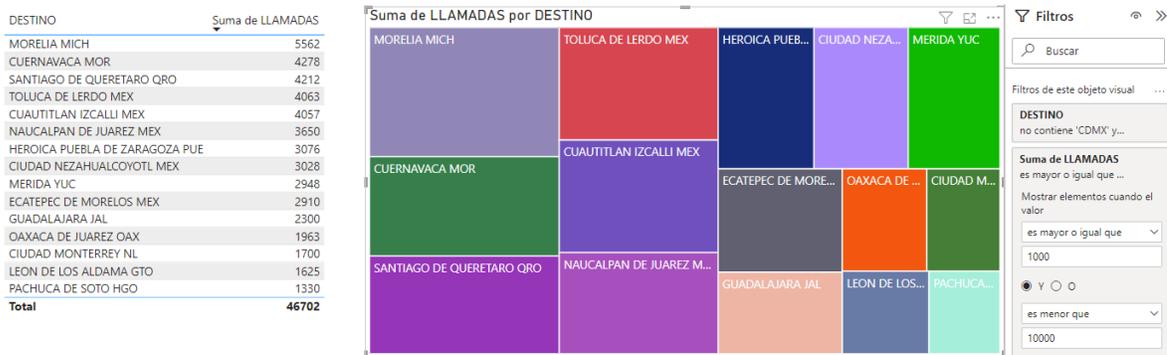
La importancia de conocer el detalle de las llamadas de larga distancia internacional se debe a que generan un costo adicional en comparación con las de larga distancia nacional, éstas últimas tienen sólo el costo de una llamada local debido a la eliminación del cobro de este servicio, promulgado en el Artículo 118 párrafo V de la Ley Federal de Telecomunicaciones y Radiodifusión en 2015, que a la letra enuncia: "Abstenerse de realizar cargos de larga distancia nacional a sus usuarios por las llamadas que realicen a cualquier destino nacional" (Cámara de Diputados del H. Congreso de la Unión, 2024, p.57). Lo anterior, consiste en un ahorro en el mayor porcentaje de llamadas realizadas dentro de la red.

Para las llamadas de tipo nacional, en la Figura 5, se muestra una lista con los lugares con mayor número de llamadas, sin incluir las realizadas en la Ciudad de México, y un filtro con un rango específico. La gráfica está compuesta de cuadriláteros de diferentes colores y tamaños correspondientes a los destinos con el mayor número de llamadas, entre los que se pueden observar a Morelia, Cuernavaca, Cuautitlán, Mérida

y Guanajuato, lugares con sedes universitarias. Esta información se puede emplear para eficientar el uso de los servicios de comunicación establecidos en estos lugares y generar un posible ahorro.

Figura 5

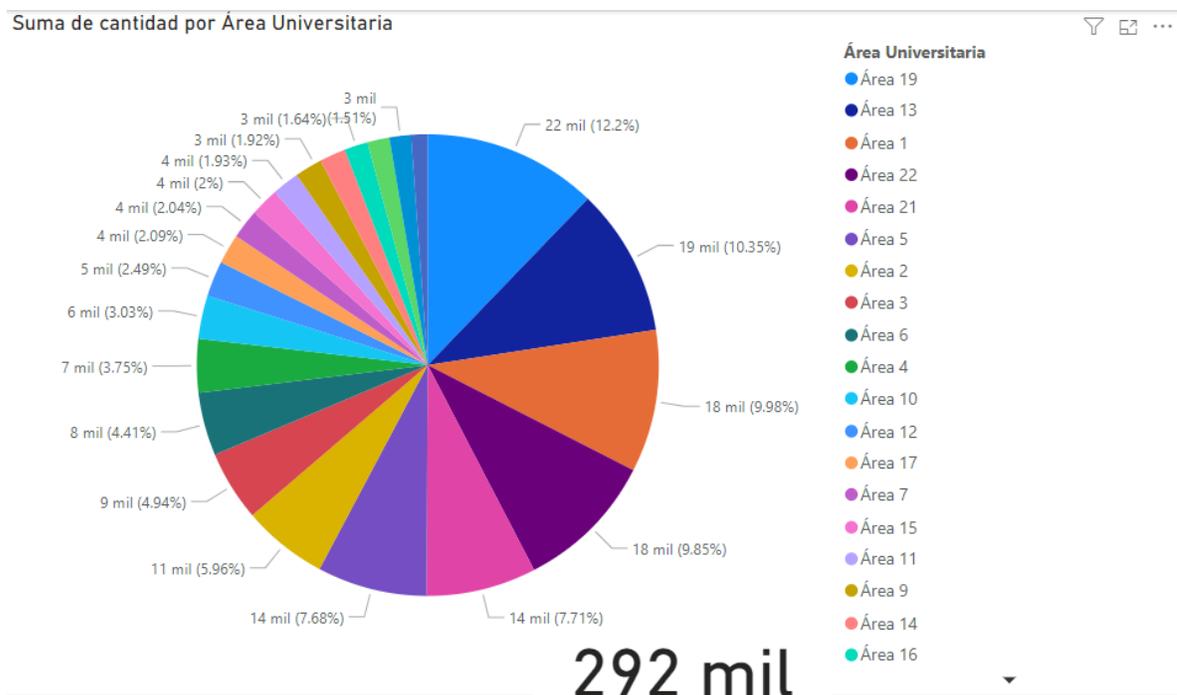
Tráfico Larga Distancia Nacional



Las llamadas internas se generan y contestan dentro de la Red Universitaria de Voz y no tienen un costo asociado, pero su uso permite conocer el aprovechamiento de la infraestructura de comunicaciones instalada; véase en la Figura 6, la gráfica que contiene las áreas que más tráfico interno reportaron en 2023.

Figura 6

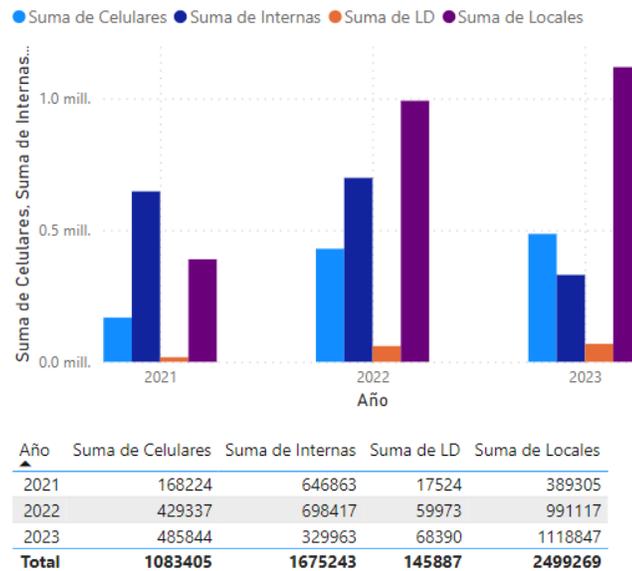
Tráfico de llamadas internas



Finalmente, para dimensionar la importancia del servicio de comunicación por voz que la Universidad ofrece a su comunidad, en la Figura 7, se presenta un comparativo de tráfico de los últimos tres años, en donde se puede observar que éste incrementó considerablemente de 2021 a 2022 en función del regreso a las actividades presenciales postpandemia. En 2023, el tráfico en general es un poco mayor con relación al año anterior.

Figura 7

Comparativo de tráfico por año



Cabe destacar que las posibilidades de creación de gráficas e información con Power BI® es inmensa. Además de los objetos visuales, se puede agregar información condensada con tablas o texto que permite fortalecer los datos representados en éstos, mejorando la comprensión de la información de una forma ágil y rápida.

3. RESULTADOS

El éxito obtenido para analizar grandes volúmenes de información, como lo es el caso del tráfico telefónico, radica en la importancia de seleccionar la herramienta correcta. La versatilidad que ofrece Power BI® para dicha tarea fue fundamental en el resultado gráfico presentado. Además del manejo eficiente de la información, cabe destacar la facilidad con la que se pueden generar diferentes objetos visuales derivados de la misma información al elegir distintas variables en la sección de filtros, detallando el comportamiento del tráfico telefónico.

En este análisis, se mostró el comportamiento del tráfico de voz en función de los tipos de llamada que realizan los usuarios. El rubro que más destaca de forma continua es el de las de tipo local y, en segundo lugar, el de las que se realizan a teléfonos móviles; ambos constituyen el mayor volumen de tráfico y de costo asociado. En menor número están las llamadas de larga distancia internacional, que son las que

generan gasto, ya que las nacionales, como ya se explicó, sólo tienen el costo de una llamada local, lo que genera un ahorro sustancial; Por otro lado, las llamadas internas se encuentran en un volumen de 292,000; éstas no generan un costo en sí, pero muestran el uso de la red, lo que puede ser considerado como un ahorro más.

Cabe resaltar que la información de este análisis es un primer acercamiento al uso de los servicios de comunicación por voz, de forma general, ofrecidos a las diferentes entidades universitarias en todo el país. Se están realizando análisis más profundos y, con ello, se pretenden establecer planes específicos de acción para aumentar la eficiencia de la red y futuros ahorros, los cuales se pueden presentar en otro documento.

4. CONCLUSIONES

En toda infraestructura de comunicación, existen gastos asociados a la misma y a los servicios de interconexión. En una red de voz como la universitaria, tan grande y compleja, es importante conocer su comportamiento con el fin de realizar las modificaciones necesarias en pro de un mejor aprovechamiento y de la contratación de los proveedores que ofrezcan las mejores condiciones para la Universidad.

De igual forma, es relevante conocer este comportamiento, ya que permite tomar fácilmente decisiones como, por ejemplo, la programación de mantenimiento de la red en períodos de menor tráfico, propuestas de reducción del gasto telefónico, balanceo de cargas de tráfico en la red, comparativos por períodos de tiempo específico, entre otras aplicaciones, que merecerán explicarse a mayor detalle en un análisis adicional.

La herramienta de inteligencia de negocios Power BI® fue de gran ayuda en el manejo de la información de forma ágil y sencilla, cumpliendo con las necesidades actuales.

AGRADECIMIENTOS

Al Ing. Ariel Candal, jefe del Departamento de Tarificación por todas las facilidades y contribuciones otorgadas en la realización de este trabajo.

REFERENCIAS

- Alvarado-Apodaca. J., Álvarez Sánchez I., Martínez-Ramírez. Y., Ramírez-Noriega. A & Tripp-Barba. C. (2023). Inteligencia de negocios en América Latina: una revisión sistemática de literatura. *Revista de Investigación en Tecnologías de Información*, 11(24), pp.77-89. <https://doi.org/10.36825/RITI.11.24.007>
- Arriagada-Benítez. M. (2020). Ciencia de datos: hacia la automatización de las decisiones. *Ingeniare. Revista chilena de ingeniería*. 28(4), pp.556-557. <https://revistas.uta.cl/pdf/2711/art01.pdf>
- Cámara de Diputados del H. Congreso de la Unión. (1º de abril 2024). *Ley Federal de Telecomunicaciones y Radiodifusión*. <https://www.diputados.gob.mx/LeyesBiblio/pdf/LFTR.pdf>
- DatData. (12 febrero 2024). *Comparativa de PowerBI con otras herramientas*. <https://www.datdata.com/blog/comparativa-de-power-bi-con-otras-herramientas>
- Intus. (2024). Power BI vs. Otras Herramientas de Business Intelligence. <https://intus.com.mx/post/>

<https://doi.org/10.22201/dgtic.ctud.2025.3.1.92>

Vol. 3, Núm. 1. enero-marzo 2025, págs. 59 - 59

[power-bi-vs-otras-herramientas-de-business-intelligence/](#)

Lemus-Delgado. D. y Pérez Navarro. R. (2020). Ciencia de datos y estudios globales: aportaciones y desafíos metodológicos. *Colombia Internacional*, (102), pp. 41-62. <https://doi.org/10.7440/colombiaint102.2020.03>

Microsoft. (2024). *Introducción a la compilación con Power BI*. <https://learn.microsoft.com/es-es/training/modules/get-started-with-power-bi/1-introduction>

De Excel a Power BI para reducir tiempo de procesamiento de datos

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:

Vázquez Padilla Díaz, A. (2025). De Excel a Power BI para reducir tiempo de procesamiento de datos. *Cuadernos Técnicos Universitarios de la DGTIC*, 3 (1) páginas(60 - 70).

<https://doi.org/10.22201/dgtic.ctud.2025.3.1.101>

Areli Vázquez Padilla Díaz

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

arelivp@unam.mx

ORCID: 0000-0002-5414-7951

Resumen

Power BI es una herramienta que cuenta con un conjunto de opciones que facilitan la transformación, modelado y visualización de datos sin afectar los datos originales. Se usó dentro de las fases de análisis de datos del cuarto estudio TIC en la verificación del procesamiento de datos reduciendo el tiempo dedicado en esta actividad, teniendo como beneficios la obtención de resultados en menor tiempo y mejor calidad del análisis con la detección oportuna de errores. Debido a esto se propone la incorporación de Power BI como herramienta de aproximación en actividades de análisis de datos, en favor de la mejora continua de los servicios ofrecidos en el área.

Palabras clave:

Análisis de datos, analítica de datos, ingeniería de datos, Power BI.

1. INTRODUCCIÓN

La ingeniería de datos es una disciplina que se centra en la implementación de infraestructura y sistemas que sirven como soporte a la recopilación, almacenamiento, procesamiento y análisis de datos. En el contexto anterior, la optimización de la gestión de datos se genera a partir del impulso a las prácticas y el uso de tecnologías para garantizar que los datos sean de alta calidad, confiables, accesibles y útiles para impulsar la toma de decisiones (Deekshith & Deekshith, 2021).

Estudiar e identificar los elementos que intervienen en el análisis de datos permitirá introducir mejoras puntuales que incrementen la calidad del análisis y de los resultados. La exploración de herramientas para su uso durante el análisis de datos será una forma de introducir dichas mejoras.

Existen diferentes herramientas para analizar datos, en este caso se abordará con mayor detalle el uso de Excel y Power BI. Excel fue una herramienta concebida originalmente como una hoja de cálculo, ha evolucionado como un conjunto cada vez más potente de herramientas, funciones y capacidades de análisis modelado de negocios y simulación de datos (Guerrero, 2018), por otra parte Power BI es una herramienta de inteligencia de negocios que tiene modelos para relacionar conjuntos de datos y aumentar las capacidades de análisis de los mismos, mejora la integración de datos y la experiencia de usuario para manejar análisis de datos complejos (Bansal, 2023). Su uso permite analizar datos de diferentes orígenes y facilita la generación de paneles de visualización e informes dinámicos (Bermeo-Moyano et al., 2020).

Enseguida se presenta un ejemplo de uso de Power BI para verificar los resultados cuantitativos del cuarto estudio del nivel de desarrollo de las TIC en las entidades y dependencias de la UNAM 2023 (Cuarto estudio TIC) (Dirección General de Cómputo y de Tecnologías de Información y Comunicación & Dirección de Colaboración y Vinculación, 2023) con lo cual se logró reducir el tiempo de análisis de datos puesto que anteriormente se realizaba con Excel.

2. DESARROLLO TÉCNICO

El cuarto estudio TIC fue realizado en el año con el objetivo de comprender el estado de las Tecnologías de Información y Comunicación (TIC) en las entidades y dependencias de la UNAM, desde una perspectiva de gobierno y gestión (Dirección General de Cómputo y de Tecnologías de Información y Comunicación & Dirección de Colaboración y Vinculación, 2023). Este es parte del proyecto 6.2.22 Integrar la Red de Responsables TIC de la UNAM, como un órgano participativo para impulsar el desarrollo y aprovechamiento eficiente de las tecnologías de la información y comunicación, del Plan de Desarrollo Institucional (PDI) 2019-2023 (Graue Wiechers, 2020).

Se brinda atención a través de 8 tipos diferentes de servicios entre los que se encuentran, asesorías y gestión de proyectos de TIC, desarrollo de software a la medida, diagnóstico de procesos informáticos, entre otros. Se colabora con diferentes entidades y dependencias de la UNAM y organizaciones externas.

En el rubro de servicios de diagnóstico de procesos informáticos es en donde se realiza la actividad de análisis de datos con mayor frecuencia, tradicionalmente se utiliza Excel como herramienta de análisis ya que la información recabada no supera la cantidad límite soportada, en este rubro se encuentra el cuarto estudio TIC. Los académicos que realizan análisis de datos cuentan con conocimientos de análisis

de datos en un nivel intermedio y la asignación de sus actividades se realiza por área de especialidad y prioridad de los proyectos por lo que no siempre el mismo académico efectúa el proceso análisis de datos completo.

En el cuarto estudio TIC se obtuvieron conclusiones a partir de la información recabada, dando seguimiento al proceso de análisis de datos que consta de las fases: preguntar, preparar, procesar, analizar, compartir y actuar (Google, 2022). En seguida se describen de manera general las actividades realizadas en cada fase para obtener resultados correctos y oportunos:

- **Preguntar:** Se habilitó un instrumento en línea compuesto por 20 preguntas respondido por Responsables TIC de 172 entidades y dependencias universitarias.
- **Preparar:** Se obtuvo una hoja de datos que fue preparada para el procesamiento de los datos.
- **Procesar:** Posteriormente, se realizó procesamiento en extenso de los datos con Excel, como en los tres estudios anteriores, en cinco unidades de tiempo. Después se verificó el procesamiento usando Power BI en una unidad de tiempo y no en cinco como anteriormente.
- **Analizar:** Tras haber procesado y verificado los datos, se realizó el análisis descriptivo y se correlacionaron los resultados numéricos para la integración del informe.
- **Compartir:** El informe revisado y ajustado fue puesto a disposición de los interesados en una página web de la UNAM en la dirección https://www.red-tic.unam.mx/recursos/2023/2023_EstudioTIC.pdf.

A continuación, se aborda con mayor detalle la fase de procesamiento de datos en el cuarto estudio TIC debido a mi participación y contribución específica en la misma.

2.1 METODOLOGÍA

Los cálculos obtenidos en el procesamiento base se realizaron con Excel, se usaron fórmulas y la copia de éstas en cada celda, todas las preguntas se procesaron en diferentes hojas de cálculo con su gráfica correspondiente y se incluyeron resultados numéricos por subsistema universitario. Se determinó que esta forma de procesamiento era la adecuada debido a que el tipo de información de la que se trataba podría requerir de comprobación y rastreo de cada cifra obtenida, y porque así se hizo en estudios anteriores. Esta actividad tuvo una duración de cinco unidades de tiempo.

La verificación de resultados requerida se hubiera llevado a cabo procesando los datos con Excel de la misma forma en la que se realizó el procesamiento inicial, para después hacer un cruce de resultados entre los académicos que participaban en el estudio.

En este estudio, el reto consistió en realizar la verificación de los resultados cuantitativos obtenidos en el procesamiento en el menor tiempo posible en comparación con el método tradicional usando Excel, es decir en menos de cinco unidades de tiempo, por lo tanto, fue necesario encontrar una estrategia diferente para lograrlo. Se exploró la verificación tradicional del procesamiento de datos, habilidades personales y la disponibilidad de herramientas en la Universidad, se concluyó que el requerimiento podía cubrirse usando una herramienta específica.

Se realizó una comparativa de herramientas que se pudieran usar en este caso, por lo que se evaluaron diferentes características de Python, Tableau y Power BI, se muestran en la Tabla 1.

Tabla 1

Resultados de la comparativa de herramientas para el procesamiento y visualización de datos

Característica evaluada	Python	Tableau	Power BI
Descarga gratuita o con acceso a comunidad UNAM	Sí	No (cuenta con una versión de prueba de 14 días)	Sí
Capacidad de mostrar resultados en corto tiempo	No	Sí	Sí
Interfaz intuitiva	No	No	Sí
Inversión de tiempo en la curva de aprendizaje acorde a las habilidades y conocimientos personales	Alto	Medio	Bajo

Python implicaba una curva de aprendizaje más lenta por el conocimiento necesario en programación, Tableau es un software con costo que tiene una versión de prueba gratuita por sólo 14 días.

Por su parte Power BI contó con la opción de uso para universidades con la que se pudo experimentar y aprender por un periodo mayor a 15 días, entre sus principales características se encontró que es una herramienta que permite trabajar, tratar y mostrar rápidamente los datos por lo que se adaptaba al requerimiento principal, además es muy útil para análisis y visualización de datos, se integra perfectamente con otras herramientas de Microsoft, es relativamente fácil de aprender, no requiere habilidades avanzadas en programación o estadística, y es adecuada para el análisis y visualización de datos básicos (Mendoza González, 2023).

Se estudiaron 17 videos tutoriales sobre Power BI (7.3 h), orientados al procesamiento de encuestas, durante tres días. Después se hizo la verificación de los resultados numéricos del cuarto estudio TIC con los siguientes pasos:

1. Carga y transformación de datos
2. Creación de un modelo de datos
3. Creación de un panel de visualización

3. RESULTADOS

A continuación se presentan los resultados obtenidos en la verificación del procesamiento del cuarto estudio TIC, se estableció la forma en la que se realizó con Excel y cómo se resolvió con algunas funcionalidades específicas de Power BI, se representó la variación en tiempo de las estrategias para el análisis de datos con Excel y con el uso de Power BI, y los problemas específicos que se resolvieron.

3.1 CARGA Y TRANSFORMACIÓN DE DATOS

Se hizo la carga de datos limpios y homogeneizados en Power BI. Al *Transformar datos* se inicia el editor de Power Query útil para hacer operaciones de revisión y limpieza de los datos antes de cargarlos en el modelo de Power BI (Microsoft Learn, s.f.), en este caso se usó la Transformación de datos con la modalidad gráfica para continuar preparando los datos (ver Figura 1). Los resultados se presentan en la Tabla 2.

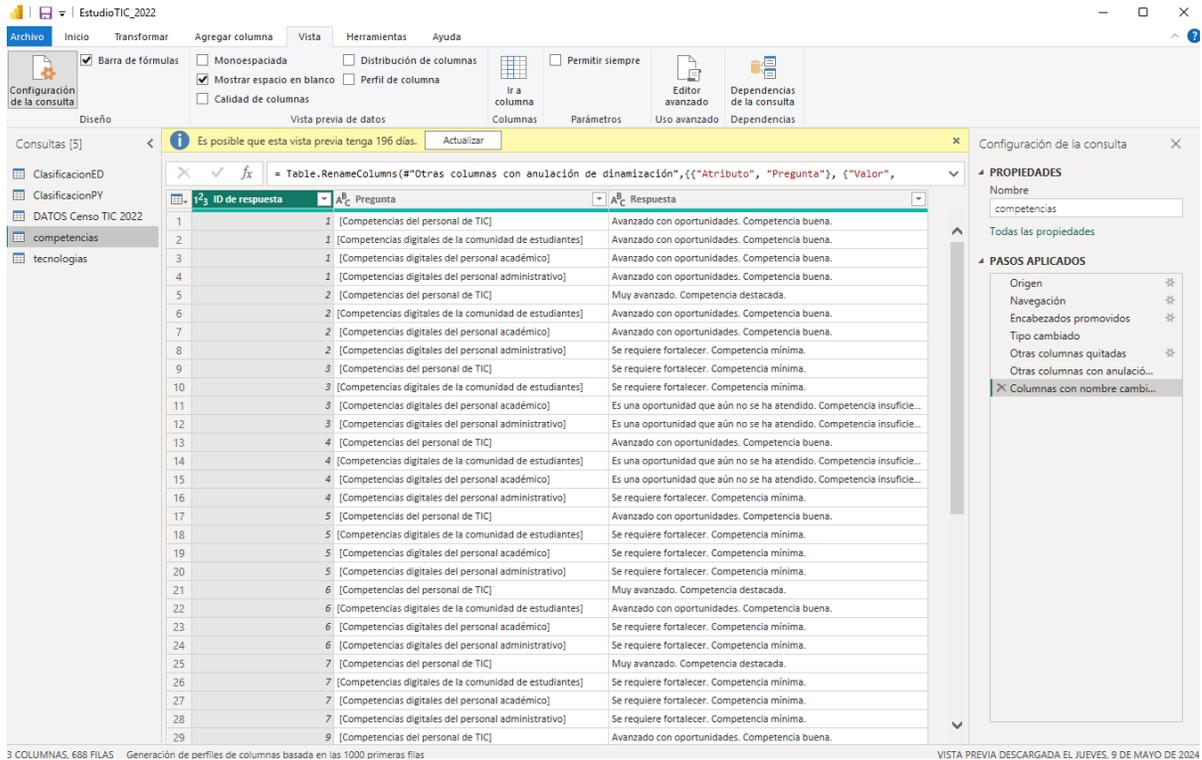
Tabla 2

Diferencias de la carga y transformación de datos en Excel y Power BI del Cuarto estudio TIC

Cuarto estudio TIC con Excel	Cuarto estudio TIC con Power BI
<ul style="list-style-type: none">• Se usaron como fuente de datos 2 archivos de Excel.• Cualquier operación sobre los datos se hace a través de fórmulas.• Las fórmulas se tienen que replicar en las celdas que lo requieran, por lo regular con arrastre manual.	<ul style="list-style-type: none">• Cada hoja de cálculo de los diferentes archivos se cargó como una tabla de una base de datos.• La transformación de datos se realiza por medio de operaciones como cambiar tipo de dato, anexar consultas, dividir columnas y reemplazar valores, las cuales quedan registradas y cuentan con la opción de revertirse o eliminarse de requerirse. En este caso se identificó automáticamente los nombres de los campos y el tipo de valor de cada columna para poder hacer operaciones sobre los datos.

Figura 1

Ejemplo de operaciones de transformación de datos



Nota: Captura de pantalla de la opción Transformación de datos de la instalación local de Power BI (2024)

3.2 CREACIÓN DE UN MODELO DE DATOS

Power BI puede establecer una relación de las hojas de cálculo a través del modelado de datos, con lo que se puede obtener un modelo de dependencias entre los archivos. En el Cuarto estudio TIC, se tenían dos archivos en Excel con una hoja de cálculo cada una (ver Figura 2), los resultados de la creación del modelo de datos se encuentran en la Tabla 3.

Tabla 3

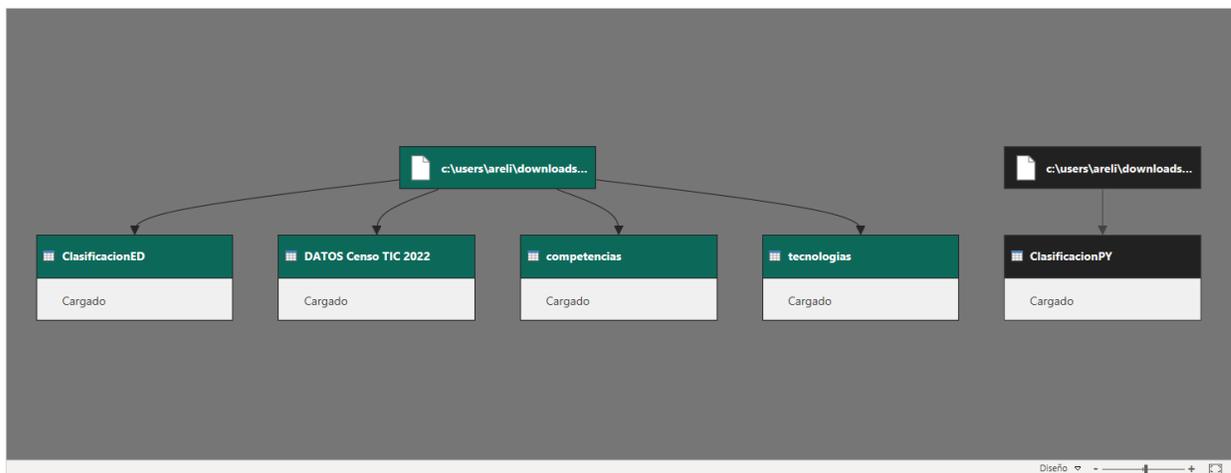
Diferencias de la creación de un modelo de datos en Excel y Power BI del Cuarto estudio TIC

Cuarto estudio TIC con Excel	Cuarto estudio TIC con Power BI
<ul style="list-style-type: none"> Para facilitar la consulta de datos de hojas de cálculo en diferentes archivos, se tuvieron que copiar en el mismo archivo. 	<ul style="list-style-type: none"> Cada hoja de cálculo de los archivos Excel se consideró como una tabla de una base de datos y se relacionaron con un identificador único. Se generaron varias tablas para analizar preguntas específicas.

Figura 2

Dependencias entre tablas de datos

Dependencias de la consulta



Nota. Captura de pantalla del modelado de datos en la versión local instalada de Power BI (2024)

3.3 CREACIÓN DE UN PANEL DE VISUALIZACIÓN

La visualización proporciona información sobre los datos de una manera interesante, sencilla y comprensible para todos sin ninguna barrera lingüística. El panel de analítica de Power BI permite representar una gran cantidad de datos en un espacio pequeño con mucha facilidad (Singh et al., 2023), seleccionar diferentes tipos de gráficos o tablas de relación para obtener elementos visuales (ver Figura 3) con estos elementos se hizo la verificación del procesamiento de datos del Cuarto estudio TIC, lo cual sería un símil de lo que se haría con tablas dinámicas de Excel, pero en una versión más avanzada y fácil de usar, más el componente gráfico. El detalle de cómo se realizó en Power BI se encuentra en la Tabla 4.

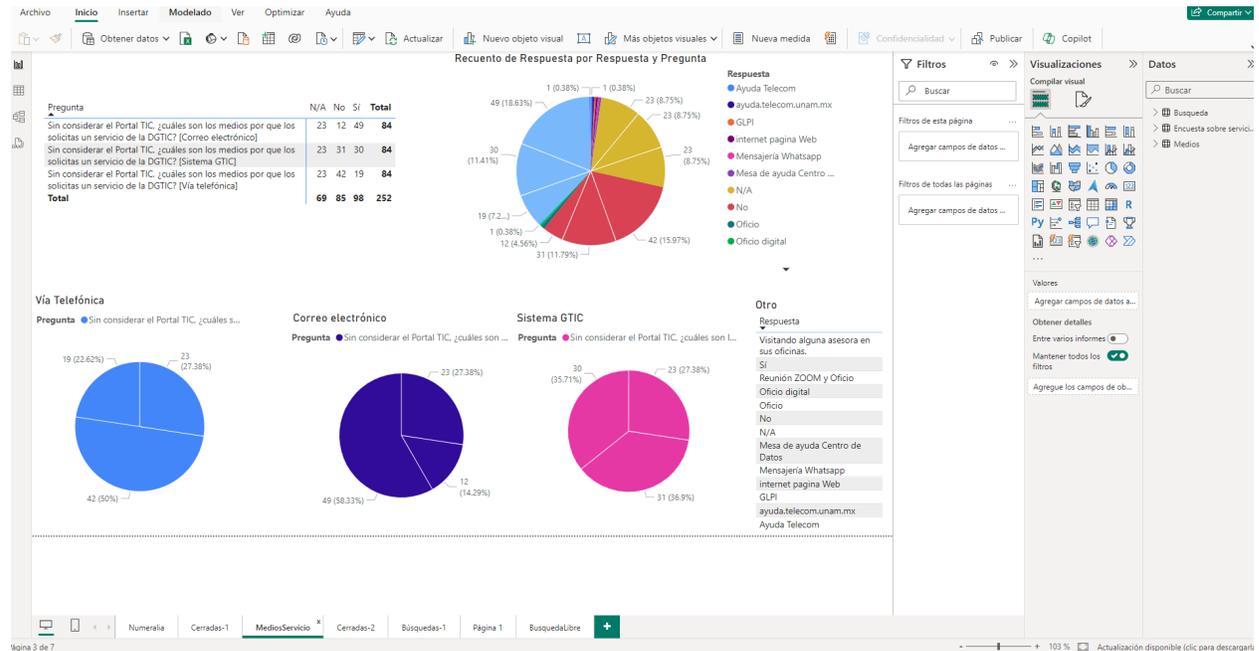
Tabla 4

Diferencias de la Creación de un panel de visualización de datos en Excel y Power BI

Cuarto estudio TIC con Excel	Cuarto estudio TIC con Power BI
<p>Cuando se genera una tabla dinámica para procesar datos de una pregunta o se agregan respuestas, se debe seleccionar el rango de datos cada vez.</p> <p>Para cada pregunta, el análisis por columnas de datos se generó en una hoja de cálculo diferente del libro de Excel.</p> <p>Para cada pregunta del Cuarto estudio TIC se generó un gráfico en una hoja de cálculo diferente del libro de Excel.</p> <p>El análisis por subsistema universitario se realizó por separado, en la misma hoja de cálculo por pregunta y en una tabla diferente, no es dinámico.</p>	<p>Se aprovechó el manejo de tablas dinámicas de una forma más eficiente para evitar la selección del rango cada vez por cada pregunta. El rango de datos se actualiza de manera automática.</p> <p>En un mismo panel de visualización se mostraron los gráficos de diferentes preguntas, se construyeron varios paneles.</p> <p>Permite cambiar de forma dinámica entre diferentes tipos de visualización la que mejor transmita los resultados y cambiar entre gráfico o tabla de resultados en el mismo panel.</p> <p>Se agregó y quitó la capa de análisis por subsistema con facilidad, de igual forma se generó el gráfico correspondiente.</p>

Figura 3

Ejemplo de panel de visualización de datos del Cuarto estudio TIC



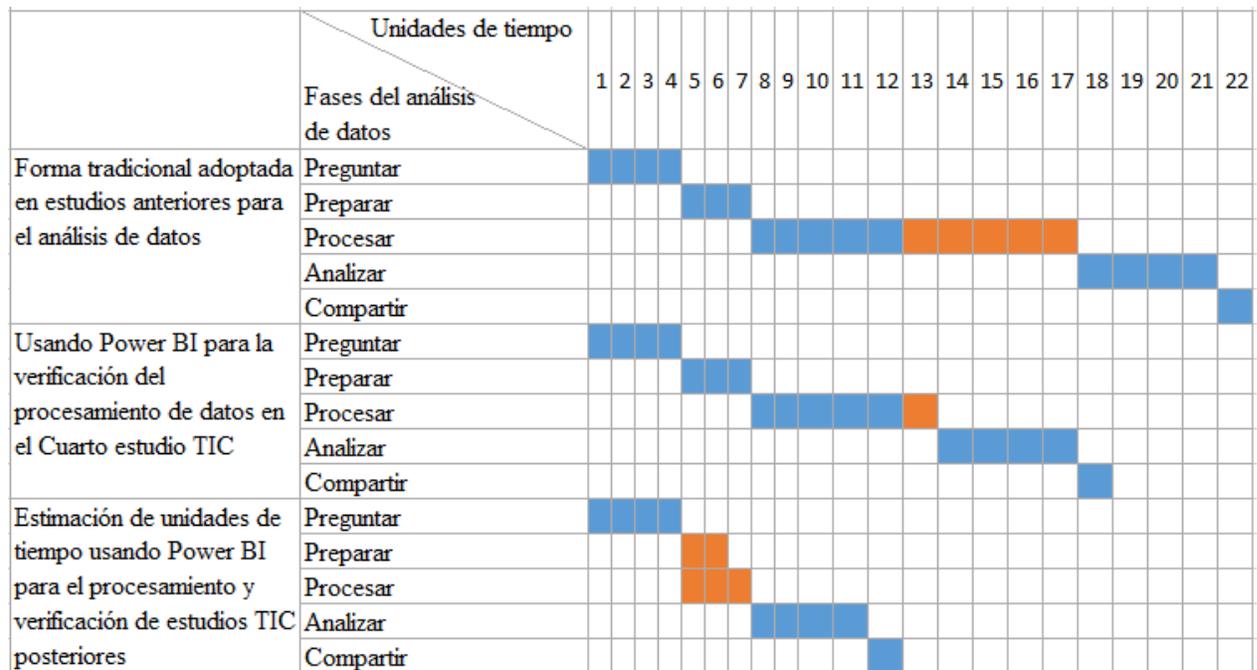
Nota: Captura de pantalla de ejemplo de un panel de visualización que se puede generar con Power BI instalado localmente (2024).

3.4 COMPARATIVA DE ESTRATEGIAS DE ANÁLISIS DE DATOS

En seguida se muestra la variación en unidades de tiempo en las dos formas de análisis de datos adoptadas en diferentes Estudios TIC. De la forma tradicional se tienen 22 unidades de tiempo utilizadas en total para el análisis de datos, en tanto que usando Power BI para la verificación del procesamiento se usaron en total 18 unidades de tiempo, reduciendo un 20% de las unidades totales empleadas en el análisis de datos (ver Figura 4).

Figura 4

Unidades de tiempo empleado para las estrategias de análisis de datos por fases abordadas para estudios TIC



Se estima que si se sustituye el uso de Excel por Power BI en la preparación, procesamiento y verificación de datos del siguiente Estudio TIC (ver Figura 4) el tiempo se pueda reducir de 22 a 12 unidades de tiempo, el 55% de las unidades totales, sin embargo, se identificaron algunas características de Power BI que en el caso particular de los estudios TIC sería necesario tomar en cuenta para su único uso:

- No se puede comprobar la fórmula u operación para la obtención de cada dato.
- Para que otros colaboradores accedan al procesamiento, es necesario tener Power BI instalado de manera local y tener una copia del origen de datos. Únicamente las licencias colaborativas admiten que varios colaboradores puedan acceder al análisis de datos.
- Si termina el licenciamiento de Power BI para comunidad UNAM no se podría acceder al procesamiento o a la visualización de este.

3.5 EJEMPLOS DE PROBLEMAS SOLUCIONADOS CON POWER BI

Transformación de datos

Dado que el origen de los datos fue a partir de una encuesta se tenían algunas preguntas específicas cuyas respuestas se tenían distribuidas en diferentes columnas, en este tipo de respuestas la transformación de los datos en Power BI se usó para:

- Hacer una copia de la tabla de datos original.
- Aislar las respuestas de la pregunta candidata eliminando columnas que no estaban involucradas en el análisis.
- Transponer las respuestas que estaban distribuidas en varias columnas para facilitar su procesamiento.

Modelado de datos

Cuando se hicieron copias de la hoja de cálculo para analizar preguntas cuyas respuestas estaban distribuidas en varias columnas compartían un identificador único, en Power BI se estableció la relación entre ambas tablas a través del modelado de datos, posteriormente se pudieron generar los gráficos o tablas que contenían información de ambas tablas.

Visualización

Para el Cuarto estudio TIC se analizaron tres preguntas con Power BI bajo la hipótesis de que se podría establecer alguna correlación entre las respuestas, se graficaron obteniendo una visualización compleja. En este caso a partir de la visualización se comprobó que no existía correlación.

También permite seleccionar un rango de datos de un gráfico y analizar su comportamiento en otras visualizaciones del mismo panel.

Omisiones al copiar fórmulas celda por celda

Cuando se copian las fórmulas en varias celdas, Excel permite replicar la fórmula en un rango de celdas. Existe la probabilidad de no seleccionar por completo el rango de datos requerido para realizar las operaciones. Power BI hace la carga de los nuevos datos y actualiza los paneles de visualización o tablas de análisis de manera automática.

4. CONCLUSIONES

La aplicación de operaciones de transformación, el modelado de datos y el tablero de visualización dinámico con Power BI en el Cuarto estudio TIC fueron clave para acortar tiempo en la verificación del procesamiento de datos y reducir el tiempo total del análisis de datos en un 20%, se obtuvieron resultados en menor tiempo, por lo que puede ser usado como una herramienta de aproximación que aporta valor en la fase de procesamiento de datos durante el análisis de estos. En el área se atienden varios servicios en paralelo, por lo que el ahorro de tiempo en los procesos representa mayor productividad.

Excel es una herramienta que se ha usado en el área como primera herramienta de análisis de datos debido a que los conjuntos de datos lo han permitido y a las opciones colaborativas con las que cuenta.

Se estima que con el uso de Power BI, las próximas actividades de análisis de datos que se realizan en el área se reduzcan hasta en un 45%, añadiendo valor al proceso y comprobando las ventajas que ofrece.

En próximos procesos de análisis de datos se propone explorar la conexión de Power BI con una encuesta en Survey Monkey en tiempo real, construir un panel de visualización y así obtener resultados preliminares en tanto se recaba la información. Por otra parte, se puede profundizar en el conocimiento de Power BI para explorar las opciones avanzadas de análisis prospectivo para el análisis de tendencias para los estudios TIC anteriores y observar los resultados.

REFERENCIAS

- Bansal, A. (2023). Power BI Semantic Models to Enhance Data Analytics and Decision-Making. *International Journal of Management (IJM)*, 14(5), 136–142. https://mylib.in/index.php/IJM/article/view/IJM_14_05_012
- Bermeo-Moyano, & Campoverde-Molina M. (2020). Implementación de Data Mart, en Power BI, para el análisis de ventas a clientes, en los Ecnegocios “Gransol”. *Polo del Conocimiento*. <https://polodelconocimiento.com/ojs/index.php/es/article/view/1242>
- Deekshith, A., & Deekshith, A. (2021). Data Engineering for AI: Optimizing Data Quality and Accessibility for Machine Learning Models. *International Journal of Management Education for Sustainable Development*, 4(4), 1–33. <https://www.ijscds.com/index.php/IJMESD/article/view/588>
- Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC), U., & Dirección de Colaboración y Vinculación, D. (2023). Cuarto estudio del nivel de desarrollo de las TIC en las entidades y dependencias de la UNAM 2023. https://www.red-tic.unam.mx/recursos/2023/2023_EstudioTIC.pdf
- Google. (2022). Aspectos básicos: Datos, datos, en todas partes | Coursera. <https://mx.coursera.org/learn/aspectos-basicos-datos-datos-en-todas-partes>
- Graue Wiechers, E. (2020, junio). Plan de Desarrollo Institucional 2019-2023. <https://www.ccg.unam.mx/divulgacion/pdi-2019-2023/>
- Guerrero, H. (2018). *Excel Data Analysis: Modeling and Simulation, Second Edition*. *Excel Data Analysis: Modeling and Simulation, Second Edition*, 1–346. <https://doi.org/10.1007/978-3-030-01279-3>
- Mendoza González, O. (2023, abril 20). Power BI en la toma de decisiones: herramientas y ventajas. Seminario TIC de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación, UNAM. <https://www.youtube.com/watch?v=Llek0s3-xPI>
- Microsoft Learn. (s/f). Obtención de datos de archivos en Power BI. Course training. Recuperado el 20 de mayo de 2024, de <https://learn.microsoft.com/es-es/training/modules/get-data/2-data-files?ns-enrollment-type=learningpath&ns-enrollment-id=learn.wwl.get-transform-data-power-bi>
- Singh, G., Kumar, A., Singh, J., & Kaur, J. (2023). Data Visualization for Developing Effective Performance Dashboard with Power BI. *International Conference on Innovative Data Communication Technologies and Application, ICIDCA 2023 - Proceedings*, 968–973. <https://doi.org/10.1109/ICIDCA56705.2023.10100169>

Reporte de pruebas en nodo de cálculo Dell R7525

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:

Álvarez Castillo, I. C. (2025). Reporte de pruebas en nodo de cálculo Dell R7525. *Cuadernos Técnicos Universitarios de la DGTIC*, 3 (1) páginas (71 - 88).

<https://doi.org/10.22201/dgtic.ctud.2025.3.1.95>

Irving Carlos Álvarez Castillo

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

irving.alvarez@unam.mx

ORCID:0009-0001-7387-1689

Resumen

Ante el cambio constante en la arquitectura de procesadores que son utilizados en las supercomputadoras, es necesario contar con información que permita conocer el comportamiento de los procesadores en diversas cargas de trabajo. Se evaluó un procesador AMD Epyc 7713 mediante estudios de escalabilidad ante cargas específicas de los códigos Gromacs y Quantum Espresso. Las aceleraciones y eficiencias obtenidas permiten conocer la escalabilidad que proporciona un nodo de cálculo con 128 núcleos de procesamiento en diferentes cargas de trabajo de *High Performance Computing* (HPC).

Palabras clave:

Supercomputadora, cómputo de alto rendimiento, AMD, evaluación de rendimiento, tiempos de ejecución, aceleración, eficiencia, Gromacs, Quantum Espresso.

1. INTRODUCCIÓN

Una supercomputadora es una máquina compuesta por diversos sistemas de cómputo, almacenamiento e interconexión; ejemplo de una supercomputadora y los diversos sistemas que la integran se pueden consultar en la sección de Miztli de la página del departamento de Supercómputo (Departamento de Supercómputo, DGTIC. UNAM, <https://www.super.unam.mx/Miztli>). El uso de las supercomputadoras ha permitido ayudar a resolver y entender diferentes problemas de diversas áreas de la ciencia como química, física e ingeniería.

El sistema de cómputo de una supercomputadora es un sistema importante porque en él se realizan los cálculos asociados a las diferentes áreas científicas. Éste está integrado por una gran cantidad (cientos o miles) de nodos de cálculo, donde cada nodo uno está compuesto de procesadores, memoria principal (RAM) y almacenamiento secundario.

Los procesadores han tenido diversos cambios tecnológicos durante la última década (Guest et al., 2021, pp. 196–199), (Suggs et al., 2020), como el aumento en la cantidad de núcleos en cada procesador (Kolpakov & Posypkin, 2020, p. 2211). En el “TOP500”(TOP500, <https://top5000.org/>), se pueden consultar las características de las supercomputadoras y de sus procesadores desde junio de 1993.

Miztli es una supercomputadora que ha servido en la generación de cientos de artículos académicos para la Universidad Nacional Autónoma de México y, en 2024, cumple doce años en operación. Su antecesora, la supercomputadora Kanbalam, estuvo en operación 6 años (2006 a 2012) y, en el mundo, los períodos de actualización de las supercomputadoras son de alrededor de 5 años. Dado lo anterior y considerando que es complicado obtener refacciones de esta infraestructura, la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) ha contemplado la adquisición de una nueva supercomputadora.

Para el proceso de adquisición de esta nueva supercomputadora es importante contar con información de los procesadores para conocer su comportamiento, sus ventajas y desventajas ante determinadas cargas de trabajo. Esta información podrá ser útil en la toma de decisiones del proyecto de adquisición de una nueva supercomputadora.

El objetivo del presente reporte es evaluar el comportamiento de procesadores AMD en un nodo de cálculo ante diversos tipos de carga de trabajo.

2. DESARROLLO TÉCNICO

2.1 METODOLOGÍA

En septiembre de 2023, se realizaron pruebas en un nodo de cálculo proporcionado por Dell modelo R7525, el cual contó con las siguientes características:

- 2 procesadores AMD EPYC 7713
- 128 núcleos de procesamiento
- 512 GB de RAM

Cada procesador AMD EPYC 7713 cuenta con 64 núcleos de procesamiento.

En el nodo de cálculo se le instaló el sistema operativo Rocky Linux versión 9.2 (versión del *kernel* 5.14) y se le deshabilitó el *Hyper-Threading*.

Las pruebas consistieron en realizar un estudio de escalabilidad utilizando dos códigos numéricos. Éste constó en ejecutar un mismo cálculo con una cantidad variable de núcleos de procesamiento; la ejecución de un caso en particular utilizando cierta cantidad de dichos núcleos fue repetida al menos cinco veces. El estudio es similar al desarrollado por Saini en "*Performance Evaluation of a Supercomputer Based on AMD Rome and Intel Cascade Lake Processors*" (Saini et al., 2022), añadiendo la variante en la cual, en lugar de utilizar comparativas de rendimiento (*benchmarks*), se utilizaron cálculos propios de algunos usuarios de la supercomputadora Miztli. Para la realización de estos cálculos, se implementaron las siguientes aplicaciones en el nodo de cálculo:

- Gromacs versión 2021.6
- Quantum Espresso versión 6.2.1

La finalidad de la implementación de ambas aplicaciones fue observar el comportamiento de los cálculos en este nodo de cálculo en particular, además de que las dos son ampliamente utilizadas en la supercomputadora Miztli.

Gromacs fue configurada utilizando el compilador GCC versión 11, la biblioteca OpenMPI versión 4 y la biblioteca FFTW versión 3.3.10. Tanto Gromacs como la biblioteca FFTW fueron construidas desde el código fuente. La entrada que se ejecutó para este código tiene 92,500 átomos y 150,000 pasos de integración.

Quantum Espresso fue construido utilizando el compilador de GCC versión 11 y la biblioteca OpenMPI versión 4. Se utilizó la biblioteca numérica Netlib proporcionada por el mismo software. Las entradas o casos utilizados para la aplicación Quantum Espresso tienen 16, 32, 48 y 64 átomos.

Se realizaron gráficas de tiempos, de aceleración y eficiencia, que se presentan en la sección de resultados. Las gráficas de aceleración y eficiencia permiten conocer qué tan eficientemente son utilizados los recursos computacionales (núcleos de procesamiento); los tiempos reportados miden el tiempo desde el inicio del cálculo hasta el final del mismo, es decir, se mide el tiempo de pared de cada cálculo. En las gráficas, una línea gris muestra el valor teórico ideal para la aceleración y la eficiencia.

3. RESULTADOS

Análisis de los resultados de Gromacs

Para el caso del código Gromacs, la gráfica de tiempos (Imagen 1) muestra que, al aumentar la cantidad de núcleos de procesamiento, fue disminuyendo gradualmente el tiempo de ejecución. En la gráfica de aceleración (Imagen 2), la tasa máxima de cambio obtenida fue 3.39 al utilizar 96 núcleos de procesamiento, para esa misma cantidad de núcleos, la aceleración teórica es de 6; para 128, la aceleración obtenida fue de 2.8 y la aceleración teórica es de 8. Sobre la gráfica de eficiencia (Imagen 3), se obtienen resultados mayores a 80% al utilizar 16, 32 y 48 núcleos de procesamiento, también se muestra que, conforme se utiliza una mayor cantidad de éstos, disminuye la eficiencia.

Las imágenes 1, 2 y 3 muestran las gráficas de tiempos, aceleración y eficiencia obtenidas (basadas en los datos de las tablas 1, 2 y 3 respectivamente del Anexo A) al ejecutar una entrada de Gromacs en diferentes cantidades de núcleos de procesamiento:

Imagen 1

Gráfica de tiempos de Gromacs



Imagen 2

Gráfica de aceleración de Gromacs

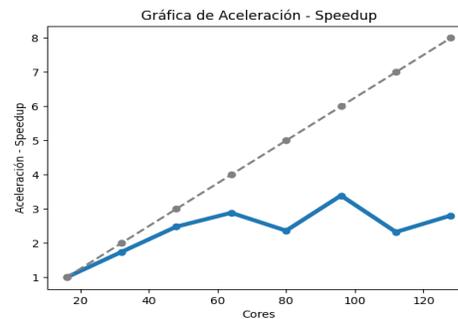
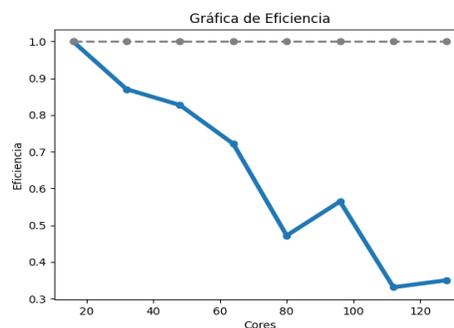


Imagen 3

Gráfica de eficiencia de Gromacs



Resultados de Quantum Espresso

Para el caso del código Quantum Espresso de 16 átomos: la gráfica de tiempos (Imagen 4) muestra que, al aumentar la cantidad de núcleos de procesamiento, fue disminuyendo gradualmente el tiempo de ejecución. En la gráfica de aceleración (Imagen 5), la tasa máxima de cambio obtenida fue 7.02 con 80 núcleos de procesamiento, para esa misma cantidad de núcleos, la aceleración teórica es de 10; para 128, la aceleración obtenida es 5.12 y la aceleración teórica es de 16. En la gráfica de eficiencia (Imagen 6), se obtienen resultados mayores a 80% al utilizar 8, 24, 32, 40 y 48 núcleos de procesamiento, también se observó que, conforme se utiliza una mayor cantidad de éstos, disminuye la eficiencia.

Las imágenes 4, 5 y 6 muestran las gráficas de tiempos, aceleración y eficiencia obtenidas (basadas en los datos de las tablas 4, 5 y 6 respectivamente del Anexo A) al ejecutar una entrada de 16 átomos de Quantum Espresso en diferentes cantidades de núcleos de procesamiento:

Imagen 4

Gráfica de tiempos de Quantum Espresso



Imagen 5

Gráfica de aceleración de Quantum Espresso

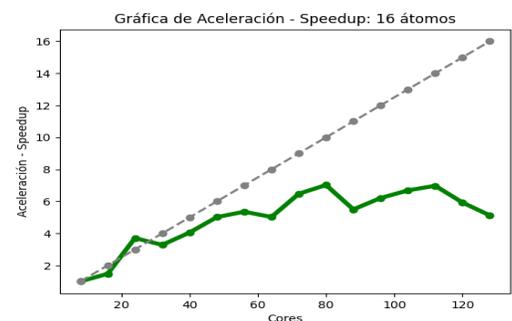
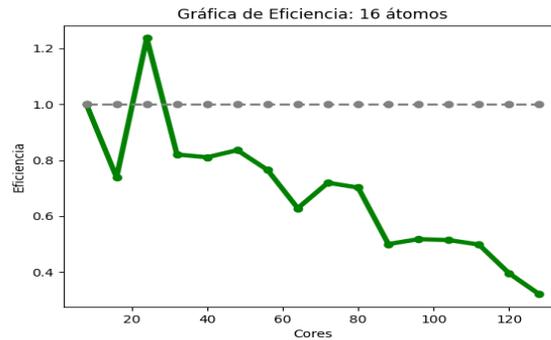


Imagen 6

Gráfica de eficiencia de Quantum Espresso



Para el caso del código Quantum Espresso (32 átomos), al aumentar la cantidad de núcleos de procesamiento, fue disminuyendo el tiempo de ejecución (Imagen 7) y se observan algunas variaciones en 64, 72, 88, 104 y 128 núcleos. En la gráfica de aceleración (Imagen 8), la tasa máxima de cambio fue 5.42, utilizando 112 núcleos de procesamiento, para esa cantidad, la aceleración teórica es de 14; en 128, la aceleración obtenida fue de 2.16 y su aceleración teórica es de 16. En la gráfica de eficiencia (Imagen 9), se obtienen resultados mayores al 80% al utilizar solo 8 núcleos de procesamiento, también se observa que, conforme se utiliza una mayor cantidad de éstos, disminuye la eficiencia.

Las imágenes 7, 8 y 9 muestran las gráficas de tiempos, aceleración y eficiencia obtenidas (basadas en los datos de las tablas 7, 8 y 9 respectivamente del Anexo A) al ejecutar una entrada de 32 átomos de Quantum Espresso en diferentes cantidades de núcleos de procesamiento:

Imagen 7

Gráfica de tiempos del cálculo II de Quantum Espresso



Imagen 8

Gráfica de aceleración del cálculo II de Quantum Espresso

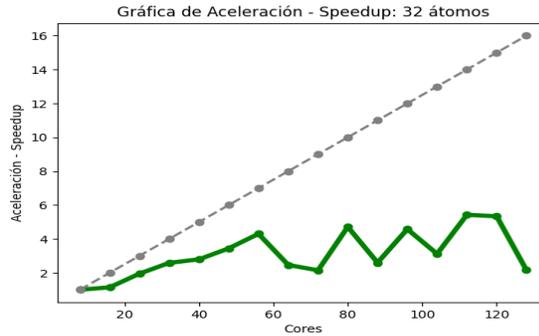


Imagen 9

Gráfica de eficiencia del cálculo II de Quantum Espresso



Para el caso del código Quantum Espresso (48 átomos), se observó que, al aumentar la cantidad de núcleos de procesamiento, fue disminuyendo gradualmente el tiempo de ejecución (Imagen 10). En la gráfica de aceleración (Imagen 11), la tasa máxima de cambio fue 5.71, obtenida con 104 núcleos, para esa misma cantidad de éstos, la aceleración teórica esperada es 13; en 128, la aceleración obtenida fue de 4.25 y la aceleración teórica esperada es de 16. En la gráfica de eficiencia (Imagen 12), se obtienen resultados mayores a 80% al utilizar solo 8 núcleos, también se observa que, conforme se utiliza una mayor cantidad de éstos, disminuye la eficiencia.

Las imágenes 10, 11 y 12 muestran las gráficas de tiempos, aceleración y eficiencia obtenidas (basadas en los datos de las tablas 10, 11 y 12 respectivamente del Anexo A) al ejecutar una entrada de 48 átomos de Quantum Espresso en diferentes cantidades de núcleos de procesamiento:

Imagen 10

Gráfica de tiempos del cálculo III de Quantum Espresso

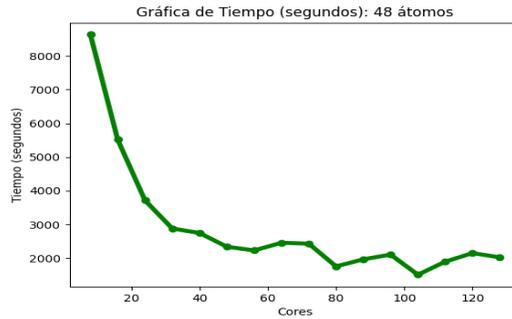


Imagen 11

Gráfica de aceleración del cálculo III de Quantum Espresso

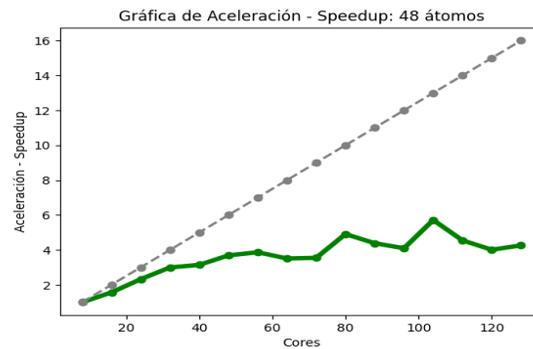
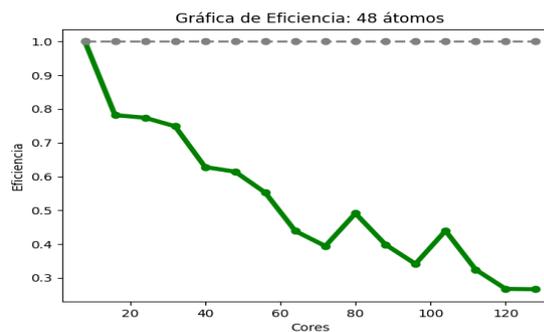


Imagen 12

Gráfica de eficiencia del cálculo III de Quantum Espresso



Para el caso del código Quantum Espresso (64 átomos), se observó que, al aumentar la cantidad de núcleos de procesamiento, fue disminuyendo gradualmente el tiempo de ejecución (Imagen 13). En la gráfica de aceleración (Imagen 14), la tasa máxima de cambio fue 4.99, obtenida con 112 núcleos, para esa misma cantidad de éstos la aceleración teórica esperada es 6; en 128, la aceleración obtenida fue de 3.11 y la aceleración teórica esperada es de 8. Sobre la gráfica de eficiencia (Imagen 15), se obtienen resultados mayores a 80% al utilizar sólo 16, 32, 48 y 64 núcleos, también se observa que, conforme se utiliza una mayor cantidad de éstos, la eficiencia disminuye.

Las imágenes 13, 14 y 15 muestran las gráficas de tiempos, aceleración y eficiencia obtenidas (basadas en los datos de las tablas 13, 14 y 15 respectivamente del Anexo A) al ejecutar una entrada de 64 átomos de Quantum Espresso en diferentes cantidades de núcleos de procesamiento:

Imagen 13

Gráfica de tiempos del cálculo IV de Quantum Espresso

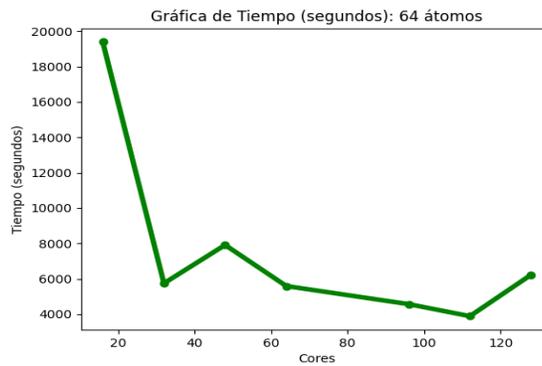


Imagen 14

Gráfica de aceleración del cálculo IV de Quantum Espresso

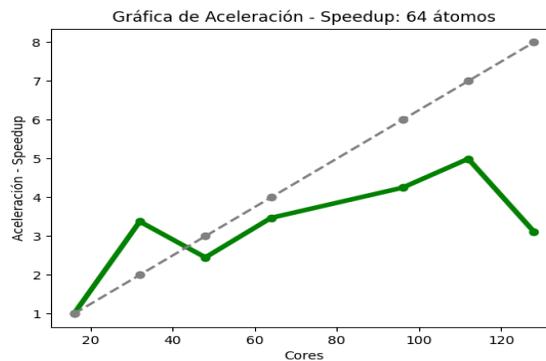
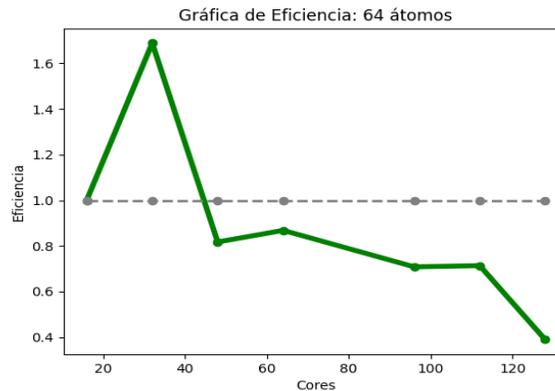


Imagen 15

Gráfica de eficiencia del cálculo IV de Quantum Espresso



4. CONCLUSIONES

Dependiendo de la cantidad de núcleos de procesamiento utilizados las gráficas de aceleración y eficiencia se mantienen cerca del valor teórico los siguientes tres casos:

1. Gromacs
2. Quantum Espresso (caso 16 átomos)
3. Quantum Espresso (caso 64 átomos)

Específicamente, para 1) y 2), las aceleraciones y eficiencias se mantienen cerca del teórico hasta 48 núcleos de procesamiento; al aumentar una cantidad mayor de núcleos, las aceleraciones y eficiencias de ambos casos se alejan del teórico. Por otro lado, para el caso 3), la aceleración y eficiencia se mantiene cerca del teórico hasta 64 núcleos de procesamiento y, al incrementar un mayor número de núcleos, la aceleración y eficiencia se distancia de la referencia teórica.

Al realizar las pruebas tanto del código Gromacs como el de Quantum Espresso, se observó que la distribución de los núcleos de procesamiento no fue secuencial, es decir, si un cálculo requería utilizar 64 núcleos, no se asignaban del núcleo 0 al 63, se distribuían 32 en un procesador y los otros 32 en otro.

Al realizar un cálculo con cierta cantidad de núcleos de procesamiento, se utilizó el nodo de cálculo de forma dedicada, es decir, no se realizaron otros cálculos aun habiendo disponibilidad de núcleos.

En el análisis de los resultados, se puede observar que hay un escalamiento limitado (las aceleraciones y eficiencias obtenidas son cercanas al teórico al utilizar pocos núcleos de procesamiento y son distantes del teórico al utilizar la totalidad de los núcleos del nodo de cálculo) para las cargas de trabajo probadas en este procesador. Se recomienda realizar más pruebas en diferentes configuraciones tanto de hardware (procesadores con menor cantidad de núcleos de procesamiento) como de software (*suites* de desarrollo) para contar con información suficiente en cuanto a los procesadores AMD.

La información obtenida nos permitió evaluar el comportamiento del procesador AMD en un nodo de cálculo con varios tipos de cargas de trabajo, por lo anterior, se cumplió con el objetivo de este trabajo.

AGRADECIMIENTOS

Se agradece a la empresa DELL por el nodo de cálculo prestado para la realización de pruebas. Se agradece a S. Frausto, E. Ortega y A. Aparicio del departamento de Supercómputo de la DGTIC por el apoyo brindado para la realización de este reporte.

REFERENCIAS

- Departamento de Supercómputo, DGTIC. UNAM. (n.d.). *Supercomputadora Miztli*. Retrieved November 15, 2024, from <https://www.super.unam.mx/miztli>
- DGTIC. (n.d.). Retrieved November 15, 2024, from <https://www.tic.unam.mx/>
- Guest, M. F., Elena, A. M., & Chalk, A. B. G. (2021). DL_POLY - A performance overview analysing, understanding and exploiting available HPC technology. *Molecular Simulation*, 47(2-3), 194-227. Scopus®. <https://doi.org/10.1080/08927022.2019.1603380>
- Kolpakov, R., & Posypkin, M. (2020). The scalability analysis of a parallel tree search algorithm. *Optimization Letters*, 14(8), 2211-2226. Scopus®. <https://doi.org/10.1007/s11590-020-01547-6>
- Saini, S., Baron, J., Chang, J., Hood, R., & Jin, H. (2022). *Performance Evaluation of a Supercomputer Based on AMD Rome and Intel Cascade Lake Processors*. 2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2022 IEEE International, IPDPSW, 848-859. IEEE Xplore Digital Library. <https://doi.org/10.1109/IPDPSW55747.2022.00141>
- Suggs, D., Subramony, M., & Bouvier, D. (2020). The AMD "Zen 2" Processor. *IEEE Micro*, Micro, IEEE, 40(2), 45-52. IEEE Xplore Digital Library. <https://doi.org/10.1109/MM.2020.2974217>
- Top 500. (<https://top500.org/>). Retrieved November 15, 2024, from <https://top500.org/lists/top500/>

ANEXO A. TABLAS

Tabla 1

Tabla de tiempos de Gromacs

Núcleos	Tiempo (segundos)
16	1503.29
32	862.832
48	605.71
64	520.64
80	635.67
96	443.109
112	646.451
128	535.222

Tabla 2

Tabla de aceleración de Gromacs

Núcleos	Aceleración (<i>Speedup</i>)
16	1
32	1.7422
48	2.4818
64	2.8873
80	2.3648
96	3.3925
112	2.3254
128	2.8087

Tabla 3

Tabla de eficiencia de Gromacs

Núcleos	Eficiencia
16	1
32	0.8704
48	0.8272
64	0.7216
80	0.472
96	0.5648
112	0.3312
128	0.3504

Tabla 4

Tabla de tiempos de Quantum Espresso

Núcleos	Tiempo (segundos)
8	1643.75
16	1112.54
24	442.956
32	500.78
40	405.502
48	327.65
56	306.906
64	327.062
72	253.852
80	234.058
88	298.842
96	264.642
104	245.834
112	235.594
120	277.258
128	320.682

Tabla 5

Tabla de aceleración de Quantum Espresso

Núcleos	Aceleración (<i>Speedup</i>)
8	1
16	1.47747
24	3.71087
32	3.28238
40	4.05362
48	5.01679
56	5.35588
64	5.02581
72	6.47524
80	7.02284
88	5.5004
96	6.21123
104	6.68643
112	6.97705
120	5.9286
128	5.1258

Tabla 6

Tabla de eficiencia de Quantum Espresso

Núcleos	Eficiencia
8	1
16	0.738737
24	1.23696
32	0.820596
40	0.810724
48	0.836132
56	0.765126
64	0.628226
72	0.719471
80	0.702284
88	0.500037
96	0.517602
104	0.514341
112	0.498361
120	0.39524
128	0.320363

Tabla 7

Tabla de tiempos del cálculo II de Quantum Espresso

Núcleos	Tiempo (segundos)
8	3198.46
16	2801.77
24	1642.22
32	1236.44
40	1144.38
48	929.498
56	743.606
64	1298.8
72	1489.39
80	678.892
88	1229.82
96	698.916
104	1026.23
112	589.746
120	599.358
128	1477.45

Tabla 8

Tabla de aceleración del cálculo II de Quantum Espresso

Núcleos	Aceleración (Speedup)
8	1
16	1.14159
24	1.94764
32	2.58682
40	2.79492
48	3.44106
56	4.30128
64	2.46262
72	2.1475
80	4.71129
88	2.60075
96	4.57632
104	3.11671
112	5.42345
120	5.33648
128	2.16486

Tabla 9

Tabla de eficiencia del cálculo II de Quantum Espresso

Núcleos	Eficiencia
8	1
16	0.570794
24	0.649214
32	0.646705
40	0.558985
48	0.57351
56	0.614469
64	0.307827
72	0.238611
80	0.471129
88	0.236431
96	0.38136
104	0.239747
112	0.38739
120	0.355765
128	0.135304

Tabla 10

Tabla de tiempos del cálculo III de Quantum Espresso

Núcleos	Tiempo (segundos)
8	8632.4
16	5519.4
24	3716.8
32	2881.87
40	2748.33
48	2341.46
56	2232.04
64	2459.25
72	2433.06
80	1758.26
88	1968.25
96	2107.33
104	1510.73
112	1898.77
120	2152.01
128	2026.51

Tabla 11

Tabla de aceleración del cálculo III de Quantum Espresso

Núcleos	Aceleración (Speedup)
8	1
16	1.56401
24	2.32254
32	2.99542
40	3.14096
48	3.68676
56	3.86749
64	3.51018
80	4.90964
88	4.38583
96	4.09638
104	5.71406
112	4.54632
120	4.01132
128	4.25974

Tabla 12

Tabla de eficiencia del cálculo III de Quantum Espresso

Núcleos	Eficiencia
8	1
16	0.782005
24	0.774179
32	0.748855
40	0.628191
48	0.61446
56	0.552499
64	0.438772
72	0.394217
80	0.490964
88	0.398712
96	0.341365
104	0.439543
112	0.324737
120	0.267421
128	0.266234

Tabla 13

Tabla de tiempos del cálculo IV de Quantum Espresso

Núcleos	Tiempo (segundos)
16	19391.4
32	5740
48	7909.2
64	5584.6
96	4562.8
112	3882
128	6225.8

Tabla 14

Tabla de aceleración del cálculo IV de Quantum Espresso

Núcleos	Aceleración (<i>Speedup</i>)
16	1
32	3.37829
48	2.45175
64	3.4723
96	4.24989
112	4.99521
128	3.11468

Tabla 15

Tabla de eficiencia del cálculo IV de Quantum Espresso

Núcleos	Eficiencia
16	1
32	1.68915
48	0.817251
64	0.868075
96	0.708315
112	0.713601
128	0.389336

Análisis estático de código fuente de aplicativos en materia de seguridad informática

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 Domínguez, A. (2025). Análisis estático de código fuente de aplicativos en materia de seguridad informática. *Cuadernos Técnicos Universitarios de la DGTIC*, 3 (1) páginas(89 - 96).

<https://doi.org/10.22201/dgtic.ctud.2025.3.1.93>

Angie Aguilar Domínguez

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

angie.aguilar@comunidad.unam.mx

ORCID: 0009-0007-7590-0678

Resumen

El análisis estático de código fuente se realizó para detectar vulnerabilidades conocidas que pudieran llegar a ser explotadas, afectando tanto a aplicativos en entornos web como en dispositivos móviles. La revisión de seguridad permitió llevar a cabo el descubrimiento de fallas de seguridad, accionando la implementación de las correcciones necesarias previo a la publicación en entornos productivos. La Dirección General de Cómputo y de Tecnologías de Información y Comunicación implementó una metodología que combinó el uso de herramientas automatizadas específicas para este fin; éstas trabajaron en conjunto con la verificación manual de los hallazgos para detectar y descartar posibles falsos positivos, además de considerar elementos correspondientes a la calidad del código que pudieran llegar a interferir en la seguridad de la información. Los hallazgos detectados y verificados fueron clasificados de acuerdo con el nivel de criticidad, siguiendo la metodología de calificación de riesgos propuesta por el *Open Web Application Security Project*. Se emitió un reporte técnico con los hallazgos, dirigido al equipo de desarrollo, para su adecuada atención y así solventar mejoras a los aplicativos. Posteriormente, se verificó la adecuada implementación de correcciones. El análisis estático de código fuente permitió detectar y remediar vulnerabilidades conocidas a nivel de código en los aplicativos que fueron analizados, colaborando a mejorar la seguridad de la información que con ellos se maneja.

Palabras clave:

Análisis estático, código fuente, seguridad informática, vulnerabilidades, SAST.

1. INTRODUCCIÓN

Debido a la gran cantidad de actividades que se realizan hoy en día mediante el uso de distintas herramientas de tecnologías de información, es necesario tener presente las vulnerabilidades existentes en el software que pueden ser explotadas por distintas amenazas, ya que, de acuerdo con Candau (2021), “las vulnerabilidades son consustanciales al software. Todos los años se publican alrededor de 5000 en diferentes tecnologías” (p.13).

Por lo anterior, es importante considerar la seguridad como un elemento esencial del ciclo de vida del desarrollo de software, independientemente del tipo, tamaño, cantidad de usuarios o finalidad de uso de aplicativo del que se trate, incluyéndola en las etapas de diseño e implementación (desarrollo).

Como indican Assal y Chiasson (2018) en la investigación que realizaron sobre la inclusión de seguridad en los proyectos de software:

...encontramos una gran brecha en las prácticas de seguridad descritas por nuestros participantes en la etapa de diseño. En esta etapa, había equipos en todos los puntos del espectro de priorización de la seguridad; sin embargo, la mayoría de los participantes indicaron que sus equipos no consideraban la seguridad como parte de esta etapa (p. 284).

Así, durante la etapa de implementación, también es necesario tener presente que “la seguridad se considera una responsabilidad del desarrollador durante la implementación, y los participantes explicaron que son conscientes de las vulnerabilidades introducidas por los errores al escribir el código” (Assal y Chiasson, 2018, p. 184).

Dada la necesidad de la realización de pruebas de seguridad sobre aplicativos web y móviles, la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) puso en marcha la realización de pruebas de revisión de seguridad en código fuente previo a la liberación en entornos productivos de los desarrollos realizados dentro de la Universidad.

El objetivo de realizar pruebas de análisis estático de código fuente en materia de seguridad consiste en encontrar posibles vulnerabilidades y fallos en términos de seguridad dentro de la programación, los cuales pueden ser mitigados siguiendo las recomendaciones propuestas antes de la publicación de los aplicativos para su uso por parte de los usuarios finales.

2. DESARROLLO TÉCNICO

De acuerdo con *Open Web Application Security Project (OWASP)*, “El análisis de código estático (también conocido como análisis de código fuente) se realiza generalmente como parte de una revisión de código (también conocida como prueba de caja blanca) y se lleva a cabo en la fase de implementación de un ciclo de vida de desarrollo de seguridad (SDL)”. (*Static Code Analysis | OWASP Foundation, s.f.*). Dichas pruebas analizan el código (sin estar en ejecución) durante la fase de desarrollo o, en el caso de seguir alguna metodología ágil, cuando se cuenta con un producto mínimo viable.

Para la realización de estas pruebas, se hizo uso de dos herramientas de análisis estático de código fuente; por una parte, una herramienta comercial *on premise* (*Fortify Static Code Analyzer, Fortify SCA*) y, por otra, una herramienta de código abierto *on premise* (*Mobile Security Framework, MobSF*).

Se hace uso de herramientas automatizadas, ya que, de acuerdo con (Bermejo Higuera et al., 2020, p. 1559): “una de las ventajas más importantes de las herramientas SAST (*Static Application Security Testing*) es que analizan toda la aplicación cubriendo todas las fuentes de entrada. En combinación con una verificación manual de los resultados”.

También se hace uso de la metodología *OWASP Risk Rating* (*OWASP Risk Rating Methodology* | *OWASP Foundation, s.f.*) para ponderar los hallazgos. Esta metodología permite realizar una estimación del riesgo asociado a la vulnerabilidad detectada en las etapas tempranas del ciclo de vida. Adicionalmente, ofrece dos ventajas importantes: primero, ésta es sencilla de implementar, mientras que, por otro lado, permite personalizar algunos elementos de la ponderación para adecuarse más a las necesidades particulares de cada proyecto, por ejemplo, la priorización al reportar un tipo de hallazgo en específico.

Así, se revisa la totalidad de los archivos del código fuente desarrollado, lo que permite tener presente la seguridad en la etapa de implementación; esto ayudará a que tanto el aplicativo web como el de dispositivos móviles (disponible para dispositivos Android o bien IOS) tome en cuenta la detección de vulnerabilidades en su implementación.

2.1 METODOLOGÍA

La metodología de análisis estático de seguridad de código fuente, propuesta y desarrollada de acuerdo con las necesidades planteadas para llevar a cabo las revisiones, comprende cuatro actividades base: diseño de las pruebas, ejecución de las pruebas, entrega de resultados y verificación de las correcciones.

Adicionalmente, la metodología empleada para la realización de las pruebas considera la aplicación de dos etapas de éstas, las cuales se estructuraron de la siguiente manera:

- Primera etapa. Se realiza el análisis estático de código fuente en materia de seguridad mediante el uso de herramientas automatizadas y verificación manual de los resultados.
- Segunda etapa. Se aplican pruebas de verificación, nuevamente con las herramientas automatizadas y verificación manual, sobre el código que contiene las correcciones implementadas, para verificar la mitigación de los hallazgos detectados en la primera etapa.

Para el diseño de las pruebas, se consideró realizar el análisis del código fuente en materia de seguridad, tomando como referencia las buenas prácticas de codificación segura de OWASP, siguiendo tanto el Top 10 para aplicativos web (*OWASP Top Ten* | *OWASP Foundation, s.f.*), como el Top 10 Mobile (*OWASP Mobile Top 10* | *OWASP Foundation, s.f.*), para aplicativos que se despliegan en dispositivos móviles.

El análisis se realiza en la totalidad de archivos de código fuente, los cuales son proporcionados por la entidad o dependencia interesada en revisar su aplicativo. Éstos se entregan en un archivo comprimido, lo que permite corroborar que se pueda llevar a cabo la revisión sobre la última versión estable.

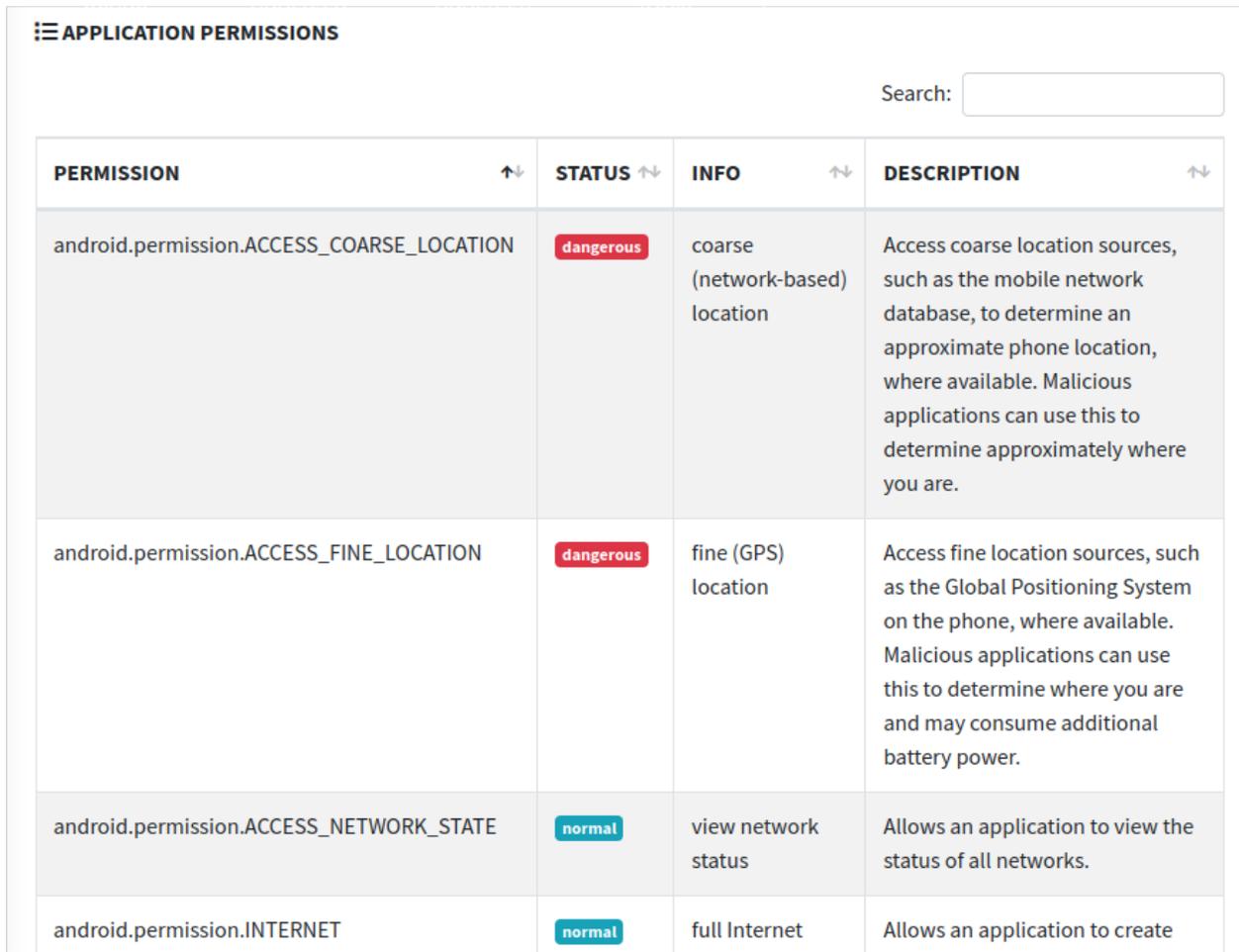
La ejecución de las pruebas comprende la realización de los análisis estáticos de caja blanca sobre los archivos que fueron compartidos, los cuales son examinados por las dos herramientas previamente seleccionadas. Posteriormente, se verifica que la salida proporcionada por las herramientas haya finalizado correctamente su ejecución, para con ello, hacer una revisión y depuración de los resultados.

En las imágenes, de la 1 a la 3, puede observarse un ejemplo del resultado de la ejecución de las pruebas. En la Imagen 1, se cuenta con la salida proporcionada por la ejecución de la herramienta *MobSF* para un

aplicativo móvil en su versión para Android, mostrando los permisos declarados para su ejecución y el estatus asignado a cada uno de ellos, así como una breve descripción del permiso.

Imagen 1

Salida proporcionada por la herramienta MobSF al verificar permisos de ejecución



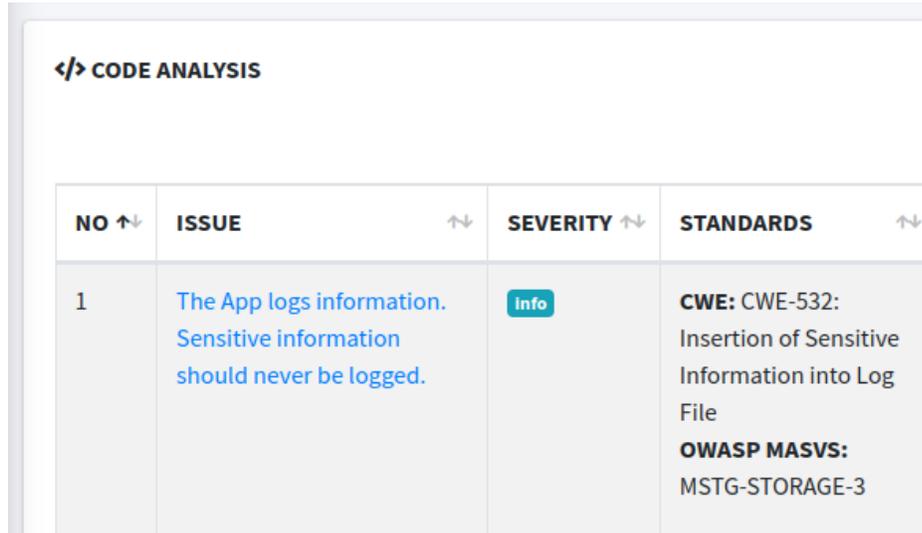
The screenshot shows the 'APPLICATION PERMISSIONS' section of the MobSF tool. It features a search bar at the top right and a table with four columns: PERMISSION, STATUS, INFO, and DESCRIPTION. The table lists four permissions: ACCESS_COARSE_LOCATION (dangerous), ACCESS_FINE_LOCATION (dangerous), ACCESS_NETWORK_STATE (normal), and INTERNET (normal). Each row provides a brief description of the permission's function and potential risks.

PERMISSION	STATUS	INFO	DESCRIPTION
android.permission.ACCESS_COARSE_LOCATION	dangerous	coarse (network-based) location	Access coarse location sources, such as the mobile network database, to determine an approximate phone location, where available. Malicious applications can use this to determine approximately where you are.
android.permission.ACCESS_FINE_LOCATION	dangerous	fine (GPS) location	Access fine location sources, such as the Global Positioning System on the phone, where available. Malicious applications can use this to determine where you are and may consume additional battery power.
android.permission.ACCESS_NETWORK_STATE	normal	view network status	Allows an application to view the status of all networks.
android.permission.INTERNET	normal	full Internet	Allows an application to create

En la Imagen 2, se observa la salida proporcionada por la herramienta *MobSF* en la sección donde se analiza el código fuente y el primer hallazgo detectado, el cual proporciona una breve descripción y las referencias a las vulnerabilidades asociadas que se encuentran reportadas en distintas bases de datos. En este ejemplo, se encontró que el aplicativo guarda en bitácora información sensible que no debe ser almacenada ahí.

Imagen 2

Salida proporcionada por la herramienta MobSF al analizar el código fuente

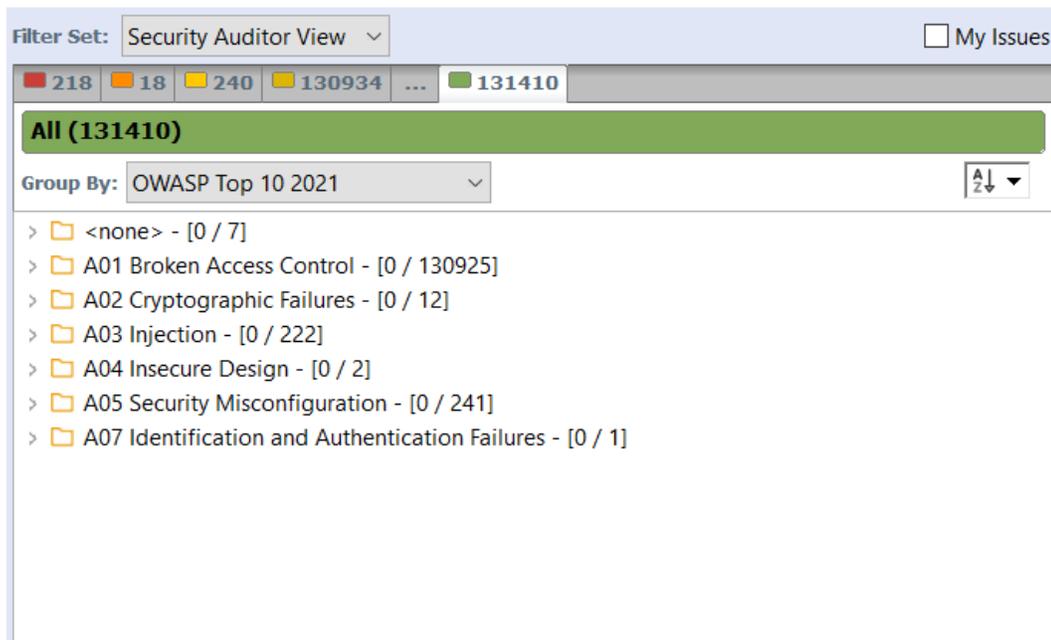


NO ↕	ISSUE ↕	SEVERITY ↕	STANDARDS ↕
1	The App logs information. Sensitive information should never be logged.	Info	CWE: CWE-532: Insertion of Sensitive Information into Log File OWASP MASVS: MSTG-STORAGE-3

En la Imagen 3, puede observarse la clasificación de hallazgos realizada por la herramienta Fortify SCA, tomando como referencia a OWASP Top 10; los presenta agrupados en carpetas después de haber realizado el análisis del código fuente. En este ejemplo, para el caso de Inyección, existen 222 hallazgos.

Imagen 3

Salida proporcionada por la herramienta Fortify SCA de acuerdo con el Top 10 de OWASP



Filter Set: Security Auditor View My Issues

218 18 240 130934 ... 131410

All (131410)

Group By: OWASP Top 10 2021

- > <none> - [0 / 7]
- > A01 Broken Access Control - [0 / 130925]
- > A02 Cryptographic Failures - [0 / 12]
- > A03 Injection - [0 / 222]
- > A04 Insecure Design - [0 / 2]
- > A05 Security Misconfiguration - [0 / 241]
- > A07 Identification and Authentication Failures - [0 / 1]

Con los resultados previamente obtenidos, es necesario que se lleve a cabo la verificación manual de la salida proporcionada por las herramientas para descartar falsos positivos, ya que, como indica de Barros (2022), “se sabe que las herramientas SAST producen falsos positivos (encuentran una vulnerabilidad donde no existe y, por lo tanto, no requieren el esfuerzo de los desarrolladores para solucionarla)” (p. 2).

Al descartar falsos positivos, se corrobora que los resultados informados al equipo de desarrollo no incluyan hallazgos erróneos y contengan sólo aquellos que representan un problema de seguridad si no son mitigados, evitando compartir información errónea.

Para la entrega de resultados, se redacta un reporte técnico con la finalidad de que el equipo responsable del aplicativo lo analice y considere la implementación de las medidas recomendadas o bien, realice una justificación que sustente la razón que impide la implementación de la corrección.

Así, el reporte entregable contiene los hallazgos, agrupados, clasificados y priorizados, acompañados de un conjunto de recomendaciones para solucionarlos, las referencias a las fuentes de información para profundizar la recomendación emitida y la calificación asignada al hallazgo de acuerdo con la metodología para ponderar conocida como *OWASP Risk Rating* (*OWASP Risk Rating Methodology* | *OWASP Foundation, s.f.*).

Las recomendaciones emitidas para solventar los hallazgos toman como referencia guías y mejores prácticas de la industria. Un ejemplo de esto son aquellas que han sido emitidas por el *National Institute of Standards and Technology* (NIST), por *MITRE Corporation*, o bien, por las distintas bases de datos de vulnerabilidades existentes.

Finalmente, en caso de que sea solicitado, se revisa la atención de las observaciones durante la segunda etapa de pruebas y se emite un segundo reporte actualizado, el cual contiene los hallazgos persistentes (aquellos que no fueron solventados), así como otros nuevos que hayan surgido durante los últimos cambios que requieran ser corregidos.

3. RESULTADOS

Como resultado del diseño y puesta en marcha de estas pruebas, se cuenta con un catálogo de los hallazgos y recomendaciones de seguridad asociadas con información unificada, el cual permitió estandarizar la información de los hallazgos que se incluye en la emisión de los reportes.

Durante el periodo de realización de las pruebas, se emitieron un total de 12 reportes, los cuales, en su mayoría, fueron revisiones de primera etapa, lo que indica que los equipos de desarrollo que solicitaron la revisión de su aplicativo aún presentan dificultades para incluir prácticas de seguridad embebidas en el ciclo de vida de sus proyectos de software.

Los hallazgos más frecuentes hacen referencia a los siguientes elementos:

- Permisos de ubicación en aplicativos para dispositivos móviles cuando no es necesario hacer uso de ellos, ya que comparten la ubicación en tiempo real del dispositivo.
- Ausencia de canales de comunicación cifrados para la transmisión de datos, lo que permitiría a un usuario malintencionado intentar interceptar datos enviados en texto claro.

- Uso de bibliotecas y software de terceros que se encuentran desactualizados, o bien, que cuentan con vulnerabilidades conocidas, lo que se podría emplear como vector de ataque contra el aplicativo.

La realización de análisis estático de código fuente sirvió como un apoyo a las entidades y dependencias de la Universidad para mejorar el aseguramiento de datos sensibles empleados por los distintos aplicativos, por ejemplo: datos de colaboradores de una dependencia, o bien, datos de estudiantes de alguna facultad.

La entrega de los resultados de la revisión permitió a los responsables de los aplicativos tomar las medidas necesarias para mejorar o incluir la seguridad (en caso de no haberla considerado) en sus proyectos de software, en la etapa de implementación y, sobre todo, a tomar en cuenta y atender las vulnerabilidades detectadas, así como a reducir el riesgo de que sean explotadas.

4. CONCLUSIONES

Como indican Assal y Chiasson (2018): “Las mejores prácticas a menudo se ignoran, simplemente porque el cumplimiento aumentaría la carga sobre el equipo; en su opinión, los equipos están haciendo un equilibrio razonable entre costo y beneficio” (p. 292).

Por lo anterior, la seguridad en el desarrollo de software debe considerarse a lo largo de las diferentes etapas, buscando aligerar la carga de su inclusión, y considerarse desde las primeras etapas de diseño de la solución que se implementará, con lo que se busca proteger no sólo la información que se maneja, sino a los usuarios y la infraestructura de TI involucrada.

La realización de las pruebas de seguridad en el código fuente de los aplicativos permitió encontrar vulnerabilidades, las cuales, al corregirse, mejoraron la seguridad en el manejo de la información utilizada, mitigando riesgos que pueden ser detectados por este tipo de pruebas antes de ser explotados.

De acuerdo con De Barros Reigada (2021): “el análisis estático tiene algunas limitaciones debido a la falta de información sobre el tiempo de ejecución. En las herramientas SAST, esta falta de información puede afectar gravemente a la calidad de los hallazgos, es decir, a su precisión” (p. 2)

Por lo anterior, es importante tomar en cuenta que el análisis estático de código fuente es sólo uno de los elementos que pueden ayudar a mejorar la seguridad de los aplicativos, ya que debe acompañarse de una revisión dinámica del aplicativo, así como la realización de análisis más completos como la aplicación de pruebas de penetración o *pentest* y la implementación de mejores prácticas sobre configuraciones para el robustecimiento de la infraestructura de tecnologías de información involucrada en su despliegue.

Finalmente, la revisión de seguridad también fungió como un facilitador para la publicación en entornos productivos de los aplicativos. Para el caso de los aplicativos web, disminuyendo la cantidad de hallazgos realizados en el análisis dinámico de seguridad y, en el caso de las aplicaciones para dispositivos móviles, facilitando su publicación en la tienda de aplicaciones correspondiente (App Store para el caso de Apple y Google Play Store para el caso de Android).

REFERENCIAS

- Assal, H., & Chiasson, S. (2018). *Security in the Software Development Lifecycle*. USENIX Association. <https://www.usenix.org/system/files/conference/soups2018/soups2018-assal.pdf>
- Bermejo Higuera, J. R., et. al (2020). Benchmarking Approach to Compare Web Applications Static Analysis Tools Detecting OWASP Top Ten Security Vulnerabilities. *Computers, Materials & Continua*, 64(3), 1555–1577. <https://doi.org/10.32604/cmc.2020.010885>
- Candau, J. (2021). Ciberseguridad: Evolución y tendencias. Bie3: *Boletín IEEE*, 23, 460–494. <https://dialnet.unirioja.es/servlet/articulo?codigo=8175398>
- De Barros Reigada, A. (2021). Generic SAST tool Comparer. *Universidade Do Minho*. <https://repositorium.sdum.uminho.pt/bitstream/1822/84173/1/Alexandra%20de%20Barros%20Reigada.pdf>
- OWASP Mobile Top 10 | OWASP Foundation. (s.f.). <https://owasp.org/www-project-mobile-top-10/>
- OWASP Risk Rating Methodology | OWASP Foundation. (s.f.). https://owasp.org/www-community/OWASP_Risk_Rating_Methodology
- OWASP Top Ten | OWASP Foundation. (s.f.). <https://owasp.org/www-project-top-ten/>
- Static Code Analysis | OWASP Foundation. (s.f.). https://owasp.org/www-community/controls/Static_Code_Analysis

CUADERNOS TÉCNICOS UNIVERSITARIOS DE LA **DGTIC**

ISSN en trámite



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