

CUADERNOS TÉCNICOS UNIVERSITARIOS DE LA DGTIC

ISSN en trámite

Vol. 1, Núm. 1. octubre-diciembre 2023

DOI:10.22201/dgtic.ctud.2023.1.1



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**

Vol. 1, Núm. 1. octubre-diciembre 2023

DOI: 10.22201/dgtic.ctud.2023.1.1



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Dirección General de Cómputo y de Tecnologías de Información y Comunicación

ISSN en trámite

CUADERNOS TÉCNICOS UNIVERSITARIOS DE LA DGTIC

Editor Responsable [Héctor Benítez Pérez](#) • Editora Académica [Marcela J. Peñaloza Báez](#) • Editora Académica Asociada [Luz María Castañeda de León](#) • Asistente Editorial [Pamela Valdés Reséndiz](#)

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](#) • [Marcela J. Peñaloza Báez](#)
• [Ana Yuri Ramírez Molina](#) • [Eprin Varas Gabrelian](#) • [Juan Voutssás Márquez](#)

Para citar un reporte técnico de la obra: Apellidos 1 Apellidos 2, Iniciales nombres. (2023). Título del reporte técnico. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (N1-N2).

CUADERNOS TÉCNICOS UNIVERSITARIOS DE LA DGTIC, Año 1, No. 1, octubre-diciembre 2023, 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 noviembre de 2023.

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/>

AGRADECIMIENTOS

La editorial desea expresar su agradecimiento por hacer posible la primera edición de esta publicación digital a la 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 • Jaqueline Segura Bautista, Gestora de Recursos • Víctor Daniel Haro Gómez, Diseñador web .

Agradecemos el apoyo de la Dirección de Colaboración y Vinculación, de la Subdirección de Comunicación e Información y de la Unidad Jurídica de la DGTIC:

Dirección de Colaboración y Vinculación: 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 • Mario Alberto Arredondo Guzmán, Soporte del gestor editorial OJS • Miguel Ángel Islas Flores, Diseño gráfico y editorial.

Subdirección de Comunicación e Información: Eprin Varas Gabrelian, Diseño editorial • Gabriela Morales Naranjo, Deena Luz Cruz Segura, Formación editorial • Rodolfo Cano Ramírez, Página web.

Unidad Jurídica: Elizabeth Rangel Gutiérrez, Jefa de la Unidad • Rossana Olivia Palomares León, Jefa de Departamento de Gestión y Análisis Contractuales • José Luis Chávez Sánchez.

Agradecemos también a las y los revisores que dictaminaron los reportes técnicos que son publicados en este número. Sus nombres aparecerán en el número 2 del volumen 2.

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Dr. Enrique Graue Wiechers, Rector • Dr. Leonardo Lomelí Vanegas, Secretario General • Mtro. Hugo Concha Cantú, Abogado General • Dr. Luis Álvarez Icaza Longoria, Secretario Administrativo • Dra. Patricia Dolores Dávila Aranda, Secretaria de Desarrollo Institucional • Dr. Héctor Benítez Pérez, Director General de Cómputo y de Tecnologías de Información y Comunicación.

CONTENIDO

Presentación.....	8
Implementación del ambiente de desarrollo de un sistema mediante máquinas virtuales Mario Alberto Hernández Mayorga	9
Tratamiento de datos personales en la contratación de servicios en la nube José Luis Chávez Sánchez.....	17
Interfaces web dinámicas: solución a las limitaciones que presentan Vue y React Pablo Enrique Zenil Rivas	25
Gestión de activos de la infraestructura de cómputo Margarita González Trejo.....	35
Implementación de la técnica de Mapeo Objeto-Relacional en el desarrollo de sistemas Cristian Ricardo Ortega Ramírez.....	52
Diseño de una cola de hilos Alejandro Talavera Rosales.....	62
Uso de contenedores para la construcción de productos de software Luis Daniel Barajas González.....	72
Análítica de negocio desde la oficina de gestión de proyectos Areli Vázquez Padilla Díaz.....	82
Integración de una aplicación PHP con el servicio de Identidad Digital Universitaria Hugo Germán Cuéllar Martínez	94
Examen diagnóstico como requisito de ingreso en educación continua en TIC Nubia Lizbeth Marina Fernández Grajales	104
Diseño, limpieza y carga de datos para un sistema de consulta Susana Laura Corona Correa	111
Análisis del proceso de integración de los cursos en línea de capacitación continua Sarai Llanos Navarrete	126
Rehabilitación del sistema de proyección del Museo Universitario de Ciencias y Arte Araceli Casas Cordero	139
Estrategias y herramientas para depuración de código en el back-end Luz María Ramírez Romero	153
Implementación de técnicas de observabilidad en el Centro de Monitoreo de la Red Esteban Roberto Ramírez Fernández.....	169
Uso de GnuPG como herramienta para la confidencialidad de la información Ricardo Chavarría Fernández	185
Directorio Activo en laboratorio virtual de la Facultad de Estudios Superiores Aragón Laura Martínez Sierra	196

PRESENTACIÓN

Desde el mes de julio de 2020, he orientado el trabajo en la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) para reafirmar y consolidar nuestra vida académica, en el entorno del servicio a la comunidad universitaria alrededor del cómputo y de las TIC, y así presentar respuestas creativas a las necesidades de la Universidad, con servicios que reafirmen nuestra contribución a la UNAM y el impacto de la actividad de la DGTIC.

En este contexto, Cuadernos Técnicos Universitarios de la DGTIC (CTUD) nace como parte de una estrategia de impulso a la madurez de la comunidad de técnicos académicos de la DGTIC. A partir del mes de noviembre de 2023, CTUD es un espacio arbitrado para compartir conocimiento especializado que cumple con criterios editoriales y que ha sido aplicado a proyectos y servicios institucionales de TIC.

En el diseño de esta nueva publicación digital, se ha definido un reporte técnico como un documento expositivo que detalla aspectos asociados a los resultados de los trabajos realizados por personal técnico académico en el ejercicio de sus funciones.

Se busca que cada autor lleve a cabo un proceso de reflexión académica para construir un texto donde demuestre su contribución individual y refleje su experiencia técnica, así como el dominio y la aplicación de conceptos propios de las diferentes áreas de especialización que se emplean en los proyectos desarrollados y los servicios prestados en torno a las TIC, para su revisión por especialistas académicos.

En este primer número se presentan 17 reportes técnicos revisados por pares bajo modalidad de arbitraje doble ciego. Invito a la comunidad de técnicos académicos de la DGTIC para que hagan suyo este espacio, que ha sido construido para ustedes, y sumar así a otras acciones impulsadas en beneficio del personal académico en los últimos años, en el marco de nuestros valores y nuestro compromiso universitario.

Agradezco la colaboración de la Dirección General de Publicaciones y Fomento Editorial, con su apoyo se buscará posicionar esta nueva publicación digital y mejorar su calidad en los tiempos por venir.

Dr. Héctor Benítez Pérez

Presidente del Comité Editorial y titular de la
Dirección General de Cómputo y de Tecnologías de Información y Comunicación

Noviembre 2023

Implementación del ambiente de desarrollo de un sistema mediante máquinas virtuales

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:

Hernández Mayorga, M. A. (2023). Implementación del ambiente de desarrollo de un sistema mediante máquinas virtuales. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (9 - 16).

<https://doi.org/10.22201/dgtic.ctud.2023.1.1.12>

Mario Alberto Hernández Mayorga

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

mariohm@unam.mx

ORCID: 0009-0002-5504-6009

Resumen:

Se describe cómo se utilizó la herramienta *Vagrant* para automatizar la creación de los ambientes de trabajo y el proceso realizado para replicar la implementación del ambiente de desarrollo mediante máquinas virtuales para la segunda versión del Sistema Integral de Personal Académico en la Dirección General de Cómputo y de Tecnologías de Información y Comunicación, así como los beneficios obtenidos, tales como facilitar la incorporación de miembros al equipo de desarrollo y la gestión de los ambientes de desarrollo, integración y pruebas.

Palabras clave:

Ambientes de desarrollo, máquinas virtuales.

1. INTRODUCCIÓN

Un ambiente de desarrollo es un espacio de trabajo con un conjunto de procesos y herramientas usadas para programar, integrar, probar, validar y ejecutar un producto de *software* (Software-und System-Entwicklung, 2023). Estos ambientes facilitan el proceso de desarrollo, el control de versiones y la realización de pruebas, lo cual agiliza el flujo de trabajo. Históricamente, la forma de crear dichos ambientes ha implicado instalar y configurar manualmente los requerimientos técnicos del producto de software —base de datos, lenguaje de programación, bibliotecas o librerías, servidor web, entre otros— en la computadora local de cada uno de los integrantes del equipo de trabajo (Hashimoto, 2013).

Sin embargo, dado que cada proyecto tiene requerimientos específicos que pueden ser incompatibles con los de otros, entre ellos, la versión de *PHP* o la configuración del servidor web, la instalación de dichos requerimientos para varios proyectos a la vez en una máquina local puede convertirse en una tarea difícil de lograr. Además, se puede incrementar la dificultad del proceso si se usan diferentes sistemas operativos en las computadoras locales, ya que puede suceder que alguno de los requerimientos o una versión en específico no exista para alguno, o bien que necesite configuraciones particulares.

Por otra parte, la virtualización de los ambientes de trabajo permite construir ambientes independientes para cada proyecto, por lo cual cada uno puede contar con su propio servidor web, manejador de base de datos y versión del lenguaje de programación, así como cualquier otra dependencia, además de que es posible simular el ambiente de producción (Peacock, 2015).

Con base en lo anterior, en este reporte técnico se describe el contexto de la implementación de los ambientes de trabajo para la segunda versión del Sistema Integral de Personal Académico y el proceso realizado para automatizar, mediante máquinas virtuales, su creación. Dicha automatización facilitó la gestión de los ambientes de desarrollo, integración y pruebas, así como la incorporación de miembros al equipo de desarrollo.

2. OBJETIVO

Automatizar la creación de ambientes de trabajo (desarrollo, integración y pruebas) para la segunda versión del Sistema Integral de Personal Académico con el fin de proveer un mecanismo de replicación del ambiente de desarrollo.

3. DESARROLLO

3.1 CONTEXTO

Parte importante de la gestión de la información del personal académico de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) de la Universidad Nacional Autónoma de México (UNAM) es la creación, evaluación, revisión y firma de informes y programas de trabajo. El Sistema Integral del Personal Académico (SIPA) permite la gestión de dicha información.

La primera versión del SIPA fue responsabilidad de una sola persona que gestionó todo el proceso de implementación del sistema, el cual contó con un ambiente de desarrollo local. Debido a que no había

más personas involucradas en el proyecto, la gestión de los ambientes de trabajo no representaba un problema. Para el desarrollo de la segunda versión, la integración de un equipo de trabajo generó nuevas condiciones a considerar, entre ellas:

- La incorporación de miembros de diferentes perfiles (desarrollo y diseño gráfico).
- La necesidad de compartir código fuente (control de versiones).
- El acondicionamiento de un ambiente de desarrollo homogéneo, con los requerimientos técnicos del sistema; para este caso: Sistema Operativo *Linux*, Lenguaje de programación *PHP* (*framework Symfony*), base de datos *MariaDB* y servidor web *Apache*.
- La creación de ambientes de trabajo para desarrollo y pruebas.

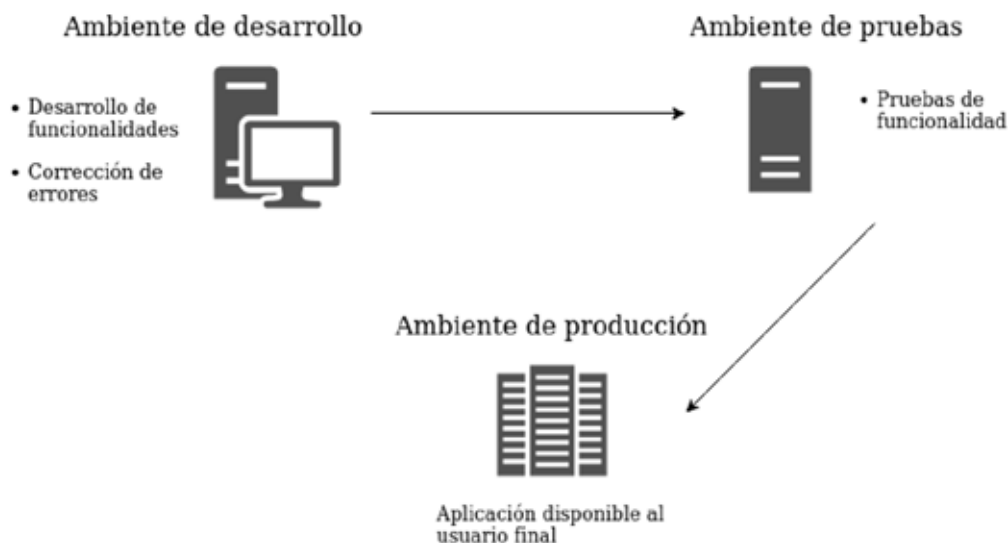
Cabe señalar que la duración del proyecto se definió a partir de los plazos establecidos para la entrega de la documentación del personal académico, por lo que la optimización del tiempo se convirtió en una prioridad. En este sentido, para poder comenzar a codificar, era necesario que los requerimientos técnicos del sistema estuvieran instalados y configurados en las computadoras de cada integrante del equipo, lo que podía ser una tarea repetitiva y que consumiría tiempo.

También se necesitaba instalar una versión del sistema donde el equipo de pruebas pudiera validar su correcto funcionamiento y reportar errores, para ello se necesitaba una nueva instalación y configuración de las herramientas para ejecutar una instancia del sistema.

Finalmente, también era necesario instalar los requerimientos en el servidor de producción, para liberar las versiones estables del sistema. En la figura 1 se muestran los ambientes de trabajo identificados.

Figura 1

Ambientes de trabajo



3.2 VIRTUALIZACIÓN DE LOS AMBIENTES DE TRABAJO

Es importante gestionar los ambientes de trabajo para desarrollo, pruebas y producción, y mantenerlos separados, para que se puedan realizar las tareas correspondientes en cada uno de ellos, sin afectar a los demás (Chamú Arias & González Guízar, 2020), propósito que se cumple a través de la virtualización de los ambientes de trabajo, lo que adicionalmente permite obtener los siguientes beneficios:

- Evitar la instalación y configuración de requerimientos directamente en la máquina local de cada uno de los integrantes del equipo.
- Replicar el ambiente de desarrollo de forma consistente.
- Emular la configuración del servidor de producción.
- Minimizar errores al desplegar el sistema en el servidor de producción.

En general se encontraron dos formas de gestionar los ambientes: con máquinas virtuales y con contenedores. Una máquina virtual es una representación virtual o emulación de una computadora física. Un contenedor es un paquete de aplicación que contiene todos los elementos necesarios para ejecutarse.

A continuación se muestra una tabla comparativa de las principales diferencias entre contenedores y máquinas virtuales:

Tabla 1

Tabla comparativa entre contenedores y máquinas virtuales

Característica	Contenedores	Máquinas virtuales
Sistema Operativo	Compartido con el del equipo donde se ejecutan	Propio
Velocidad	Inicio y apagado en segundos	Inicio y apagado en minutos
Uso de recursos	Menor	Mayor
Tamaño	MB	GB
Aislamiento	Aislamiento a nivel de proceso.	Aislamiento completo del equipo donde se ejecuta.

Nota. Adaptada de Containers vs VMs (virtual machines): What are the differences? de Google Cloud, s.f., <https://cloud.google.com/discover/containers-vs-vm>.

Derivado del análisis de las dos formas de virtualizar descritas previamente, el uso de contenedores se presenta como la opción más ligera para virtualizar el ambiente de trabajo. Por ello, se exploró la opción de implementar el ambiente de desarrollo mediante contenedores, y se usó *Docker* que es la herramienta más popular y ampliamente usada para su manejo.

Durante su instalación se presentaron dificultades que comenzaron a consumir tiempo, la principal fue que las computadoras del equipo de trabajo tenían distintas características y sistemas operativos y no todas contaban con los requerimientos mínimos para la instalación de la herramienta. Se intentó instalar

Docker mediante el proyecto *Docker Toolbox*¹ (ya obsoleto), pero se comenzó a invertir más tiempo en la instalación y configuración particular de las computadoras. Además, se presentaron incompatibilidades en las versiones instaladas con *Docker Toolbox*.

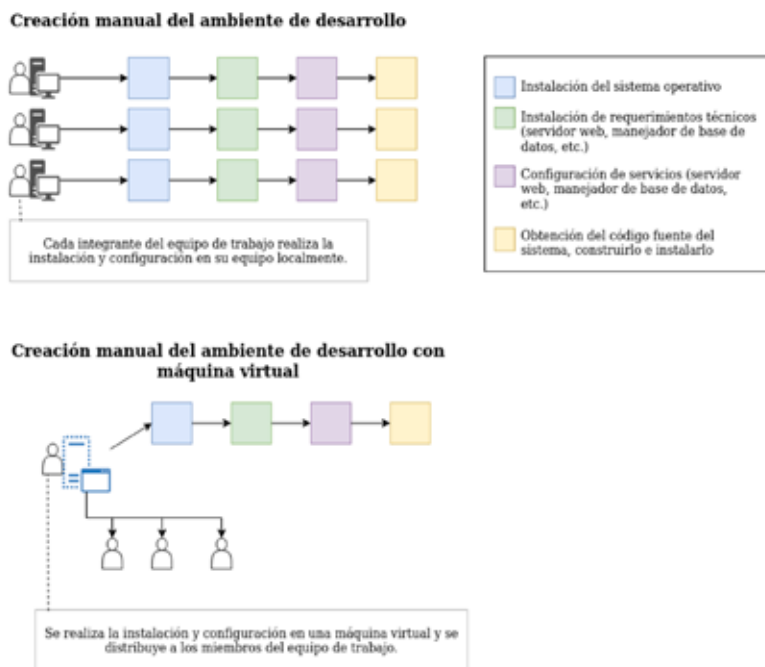
También se tomó en cuenta que el despliegue en producción del sistema, en ese momento, no se realizaba con contenedores, sino en un servidor virtual en el Centro de Datos de la DGTIC. Como dicho ambiente se simula de mejor manera con una máquina virtual, se optó por virtualizar los ambientes de trabajo a través de ella, considerando también la experiencia previa de algunos miembros del equipo en la materia.

3.3 AUTOMATIZACIÓN DE LA CREACIÓN DE LOS AMBIENTES DE TRABAJO

Como se mencionó en la introducción, históricamente, la forma de crear los ambientes de trabajo era mediante la instalación y configuración manual de los requerimientos técnicos del producto de *software*. Con máquinas virtuales, se realiza la instalación y configuración de una máquina virtual con los requerimientos necesarios para ejecutar el sistema, y después se distribuye a cada integrante del equipo de trabajo (figura 2). Así se tiene la certeza de que todos cuentan con las mismas versiones y configuraciones de las herramientas necesarias (Ramani, 2019).

Figura 2

Creación manual vs Creación con máquina virtual



¹ *Docker Toolbox* es un proyecto que permite ejecutar *Docker* en versiones de sistemas operativos *Windows* o *MacOS* que no cumplen con los requerimientos de la herramienta.

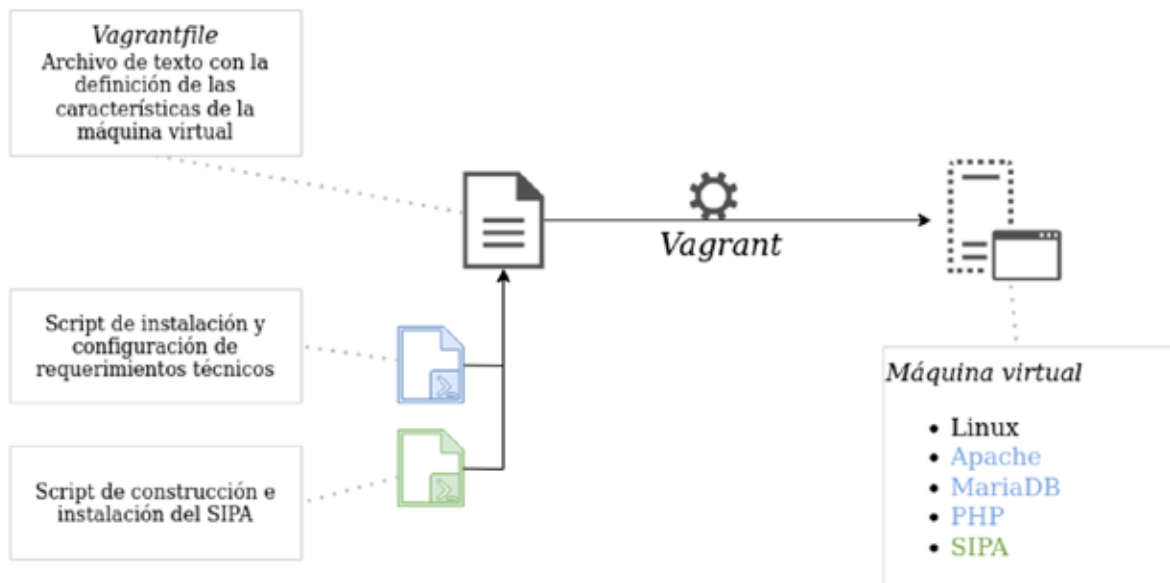
Sin embargo, un aspecto que se presentó fue que el tamaño del archivo que contiene la máquina virtual es grande (del orden de GB), por lo que su distribución a los integrantes del equipo se volvió tardada y aún era necesario instalar manualmente los requerimientos técnicos, aunque fuera una sola vez. Para atender estas situaciones, se buscaron herramientas de automatización, dentro de las cuales se encontró *Vagrant* como una buena opción, ya que es una herramienta para construir y administrar entornos de máquinas virtuales en un solo flujo de trabajo, fácil de usar y con un enfoque hacia la automatización.

La creación de la máquina virtual con *Vagrant* se realiza por medio de un archivo de texto, llamado *Vagrantfile*, que contiene la información necesaria para crear y replicar el ambiente. Además, *Vagrant* brinda opciones para aprovisionar la máquina virtual, desde *scripts de shell* hasta sistemas de administración de configuración.

Para aprovechar esta característica de la herramienta, para el desarrollo del SIPA, se crearon dos *scripts de shell*, uno con el fin de instalar los requerimientos del sistema, y el segundo, para la construcción e instalación de éste; los *scripts* se reutilizaron para instalar el sistema en producción (figura 3).

Figura 3

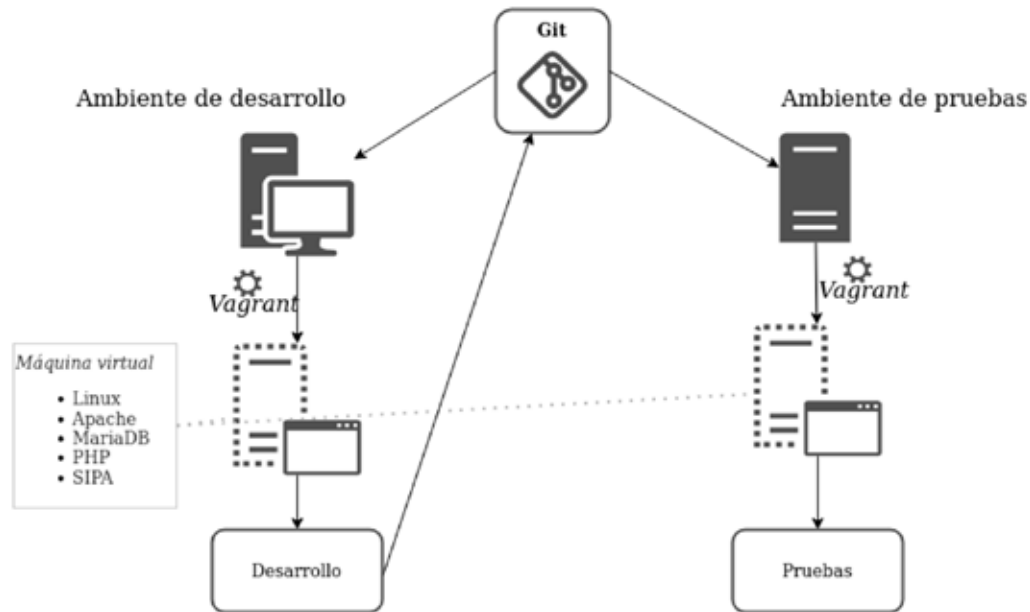
Uso de *Vagrant* para la creación automatizada de la máquina virtual



Finalmente, el archivo de texto para generar la máquina virtual y los *scripts* para configurar los requerimientos técnicos se agregaron al sistema de control de versiones del proyecto, permitiendo que las actualizaciones al ambiente de desarrollo también pudieran tener control de cambios. De esta forma, al obtener o actualizar el código fuente en el sistema de control de versiones, también se incluyen los archivos necesarios para crear la máquina virtual con el ambiente de desarrollo local, y así, comenzar el desarrollo por parte de cada uno de los miembros del equipo, en cuestión de minutos. Los ambientes para las etapas de integración y pruebas se crearon de la misma forma (figura 4).

Figura 4

Creación de ambientes de desarrollo y pruebas



4. RESULTADOS

Se logró automatizar la creación de ambientes de trabajo con máquinas virtuales para el desarrollo del SIPA, lo cual permitió:

- Replicar el ambiente de desarrollo de forma consistente para cada uno de los integrantes del equipo (desarrollo y diseño gráfico).
- Reducir el tiempo de creación y configuración del ambiente de desarrollo en aproximadamente un 70 por ciento.
- Crear los ambientes necesarios para las etapas de integración y pruebas.
- Reutilizar el *script* de instalación del sistema en el servidor de producción.

Como trabajo a futuro, se propone explorar la posibilidad de usar tecnologías como *Puppet* o *Chef* para la instalación de los servicios o requisitos técnicos en la máquina virtual en lugar de *scripts* de *shell*. Dichos programas permiten definir los servicios a instalar y configurar el servidor de forma transparente.

5. CONCLUSIONES

Automatizar la creación de los ambientes de trabajo para cada etapa (desarrollo, integración y pruebas) permite realizar las actividades de cada una de ellas en un ambiente aislado y permite que el flujo de trabajo se ejecute de forma más ágil.

Se identificó que la virtualización de ambientes de desarrollo con máquinas virtuales puede ser una alternativa cuando por el contexto de un proyecto no es posible llevarlo a cabo con contenedores.

El uso de *Vagrant* para la creación de ambientes en máquinas virtuales facilitó la gestión de los ambientes de desarrollo, integración y pruebas, permitió la replicación consistente de los ambientes, y redujo el tiempo de acondicionamiento del ambiente de desarrollo así como la distribución del mismo en el control de versiones del proyecto.

REFERENCIAS BIBLIOGRÁFICAS

- Chamú Arias, J. O., y González Guízar, A. (2020). *La importancia de separar los ambientes de trabajo en el desarrollo de los productos de software*. Boletín Informativo para el Personal de la DGTIC, 13. <https://www.tic.unam.mx/boletin-13/>
- Google Cloud. (s.f.). *Containers vs VMs (virtual machines): What are the differences?* Temas Google Cloud. <https://cloud.google.com/discover/containers-vs-vm>
- Hashimoto, M. (2013). *Vagrant: Up and running*. (1ra. ed). O'Reilly Media, Inc.
- Peacock, M. (2015). *Creating development environments with vagrant: Leverage the power of Vagrant to create and manage virtual development environments with Puppet, Chef, and VirtualBox*. (2a. Ed). Packt Publishing.
- Ramani, M. (2019). *Moving towards devops with Vagrant*. https://education.dell.com/content/dam/dell-emc/documents/en-us/2019KS_Ramani_Moving_Towards_DevOps_with_Vagrant.pdf
- Software- und System-Entwicklung (2023). Development Environment. *Technology Definitions | SUSE Defines*. <https://www.suse.com/suse-defines/definition/development-environment/>

Tratamiento de datos personales en la contratación de servicios en la nube

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:

Chávez Sánchez, J. L. (2023). Tratamiento de datos personales en la contratación de servicios en la nube. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (17 - 24).

<https://doi.org/10.22201/dgtic.ctud.2023.1.1.8>

José Luis Chávez Sánchez

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

jchavez@comunidad.unam.mx

ORCID: 0009-0008-5891-3984

Resumen:

Se presenta un análisis de la normatividad universitaria y nacional que regula las obligaciones de los proveedores que brindan servicios en la nube y realizan el tratamiento de datos personales en posesión de las áreas universitarias y se proponen recomendaciones para su incorporación en los instrumentos consensuales que formalizan la contratación de dichos servicios.

Palabras clave:

Datos personales, servicios de nube, instrumentos consensuales.

1. INTRODUCCIÓN

Con el fin de asegurar la continuidad de sus actividades, las organizaciones han contemplado la migración de sus servicios a la nube, esta realidad no es ajena a las diferentes entidades y dependencias de la Universidad Nacional Autónoma de México sin embargo, requieren especial atención los servicios que llevan a cabo el *“tratamiento de la información que se realiza en la nube... por un tercero y... sobre todo aquella que se refiere a las personas...”* (Chávez Sánchez, 2022 p 16), en *“aspectos como nombre, domicilio, teléfono, edad, sexo, escolaridad, estado civil, religión, filiación política, ocupación, amigos, familia, cuentas bancarias, pasatiempos, estado de salud, etcétera”* (Meraz Espinoza, 2018 p 301), esta información comprende los “datos personales”.

De acuerdo con lo establecido en el artículo 3 fracción IX de la Ley General de Protección de Datos Personales en Posesión de Sujetos Obligados (2017) y la fracción VI del Artículo 2 del “Reglamento de Transparencia y Acceso a la Información Pública de la Universidad Nacional Autónoma de México”, la denominación legal de los datos personales es “cualquier información concerniente a una persona física identificada o identificable”.

Cabe señalar que, para efectos de este análisis, se entiende como los “servicios de nube pública” al “modelo de servicio de tecnología de información adquirida bajo demanda a terceros, operada en infraestructura ajena a la Universidad”, de acuerdo con lo indicado en la fracción XXXI del artículo 2 de las “Normas complementarias sobre medidas de seguridad técnicas, administrativas y físicas para la protección de datos personales en posesión de la Universidad”, en lo sucesivo, “Normas Complementarias”.

Para que un área universitaria realice la contratación de servicios de nube pública, es importante conocer en primer lugar el contenido de la normatividad universitaria y nacional vigente, e identificar las obligaciones mínimas que por disposición de la ley deben cumplir los proveedores que brindan estos servicios, para posteriormente incorporarlas en el instrumento consensual que contiene los derechos y obligaciones de los implicados durante y después de la prestación del servicio.

Por lo anterior y dada la importancia de los temas antes expuestos, estos puntos son desarrollados con el fin de aportar recomendaciones puntuales que puedan ser objeto de consideración por parte de aquellas áreas universitarias que estén contemplando contratar dichos servicios.

2. OBJETIVOS

2.1. OBJETIVO GENERAL

Analizar la normatividad vigente relacionada con las obligaciones que deben cumplir los proveedores que brindan servicios en la nube a áreas universitarias y realizan el tratamiento de datos personales que están en su posesión.

2.2. OBJETIVO ESPECÍFICO

Proponer recomendaciones para fortalecer los instrumentos consensuales que se suscriben para la contratación de los servicios en la nube.

3. DESARROLLO

Los datos personales son considerados como información confidencial y sólo puede tener acceso a ella la persona que provee la información, ya sea por sí misma o por medio de sus representantes, y/o el personal universitario facultado para ello, de acuerdo con lo establecido en el Aviso de Privacidad o bien en el documento donde la persona manifiesta su consentimiento sobre el uso que se realizará de sus datos personales.

De acuerdo con lo señalado, son las áreas universitarias quienes fungen como **Responsables** de los datos personales, y tienen la obligación de manejar, resguardar y/o decidir sobre el tratamiento de dichos datos. A su vez, sólo los **Responsables** pueden encomendar dicho tratamiento a una tercera persona, la cual se denomina Encargado; esta relación se formaliza por medio de la suscripción de instrumentos consensuales, de conformidad con lo establecido en el artículo 45 de los "Lineamientos para la Protección de Datos Personales en Posesión de la Universidad Nacional Autónoma de México".

En el numeral 46 de estos lineamientos se indica que el instrumento consensual que se suscribe entre el **Responsable** y el **Encargado** debe contener las siguientes cláusulas: i) el instrumento consensual se debe regir por lo establecido en el Aviso de Privacidad que haya publicado el **Responsable**; ii) indicar que por ninguna razón el **Encargado** puede decidir por sí mismo sobre el tratamiento de dichos datos, por lo que no podrá tratarlos para una finalidad distinta de las indicadas; iii) contener instrucciones claras y precisas de los tratamientos que puede realizar el **Encargado**; iv) establecer la obligatoriedad para que el **Encargado** tenga implementadas las medidas de seguridad requeridas por la normativa universitaria, y si es factible, se sugiere indicar las acciones o medios para verificar este punto; la obligación para que el **Encargado** garantice guardar confidencialidad respecto a los datos personales tratados; v) una vez concluido el servicio, el **Encargado** debe eliminar o devolver los datos personales objeto de tratamiento; y vi) señalar que no se podrán subcontratar dichos servicios.

Es relevante considerar asimismo lo establecido en el artículo 21 de las "Normas Complementarias", el cual indica que se permite el resguardo y tratamiento de datos personales en los sistemas universitarios en servicios de nube pública, siempre y cuando estén cifrados.

La Ley General de Protección de Datos Personales en Posesión de Sujetos Obligados (2017), establece regulaciones similares a las contempladas en la normatividad universitaria, principalmente en los artículos 59, 63 y 64. De esta revisión, destacan tres supuestos que indican otras obligaciones, las cuales pueden ser consideradas por las áreas universitarias:

- El **Encargado** debe dar a conocer cambios en sus políticas de privacidad o condiciones del servicio que presta,
- el **Encargado** deberá informar al Responsable cuando ocurra una vulneración a los datos personales que trata por sus instrucciones, y
- el **Encargado** debe abstenerse de incluir condiciones en la prestación del servicio que le autoricen o permitan asumir la titularidad o propiedad de la información sobre la que preste el servicio.

4. RESULTADOS

Con base en el análisis realizado, se pone a consideración de las áreas universitarias una serie de recomendaciones para la contratación de servicios en la nube, cuya inclusión en los instrumentos consensuales está debidamente fundamentada y se presentan en las tablas siguientes, donde:

- En la primera columna se incluye el artículo de la disposición normativa.
- En la segunda columna se incluye el requisito textual de la norma.
- En la tercera columna se incluyen algunas sugerencias para ser consideradas durante la elaboración del contenido de los instrumentos consensuales:

NORMATIVIDAD UNIVERSITARIA LINEAMIENTOS PARA LA PROTECCIÓN DE DATOS PERSONALES EN POSESIÓN DE LA UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO		
Fundamento legal: Artículo 46.- Son obligaciones de los encargados del tratamiento de datos personales, además de las asentadas en los instrumentos consensuales que al efecto se suscriban		
FUNDAMENTO	REQUISITO	SUGERENCIAS
Artículo 46 fracción I	Realizar actividades del tratamiento de los datos personales sin ostentar poder alguno de decisión sobre el alcance y contenido del tratamiento.	<i>Incluir una cláusula que estipule el alcance del tratamiento de los datos personales por parte del Encargado.</i> <i>En caso de contar con un Aviso de Privacidad contemplar su contenido para delimitar el alcance de dicho tratamiento.</i>
Artículo 46 fracción II	Tratar únicamente los datos personales conforme a las instrucciones de las áreas universitarias.	<i>Incluir un anexo con los manuales, procedimientos y documentos que contengan las instrucciones para que el Encargado realice el tratamiento de datos personales.</i>
Artículo 46 fracción III	Abstenerse de tratar los datos personales para finalidades distintas a las instruidas por las áreas universitarias.	<i>Incluir una cláusula con la restricción indicada para el Encargado (proveedor) de no tratar los datos personales con fines diferentes a los señalados por el área universitaria.</i>
Artículo 46 fracción IV	Implementar las medidas de seguridad conforme a los presentes lineamientos y las demás disposiciones legales aplicables.	<i>Indicar la obligación para que el proveedor cumpla con las medidas de seguridad indicadas por la normativa universitaria.</i>

NORMATIVIDAD UNIVERSITARIA LINEAMIENTOS PARA LA PROTECCIÓN DE DATOS PERSONALES EN POSESIÓN DE LA UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO		
Fundamento legal: Artículo 46.- Son obligaciones de los encargados del tratamiento de datos personales, además de las asentadas en los instrumentos consensuales que al efecto se suscriban		
FUNDAMENTO	REQUISITO	SUGERENCIAS
Artículo 46 fracción V	Guardar confidencialidad respecto de los datos personales tratados.	Indicar una cláusula en la cual se haga referencia a la confidencialidad de los datos tratados. Adicionalmente se puede agregar la posibilidad de que el representante legal y el personal del proveedor suscriban cartas de confidencialidad.
Artículo 46 fracción VI	Suprimir o devolver los datos personales objeto de tratamiento una vez cumplida la relación jurídica con el área universitaria o por instrucciones de ésta, siempre y cuando no exista una previsión legal que exija la conservación de los datos personales.	Se recomienda incluir un procedimiento para la eliminación de datos con herramientas de borrado seguro y documentar su eliminación mediante la elaboración y suscripción de un acta.
Artículo 46 fracción VII	Abstenerse de transferir los datos personales relativos a su encargo.	La normatividad universitaria no permite que el proveedor transfiera datos personales a terceros.
Artículo 46 fracción VIII	No podrá subcontratar servicios que impliquen el tratamiento de datos personales por cuenta de la Universidad.	Incluir una cláusula que indique que no se pueden realizar subcontrataciones para brindar el servicio en la nube.

NORMATIVIDAD UNIVERSITARIA NORMAS COMPLEMENTARIAS SOBRE MEDIDAS DE SEGURIDAD TÉCNICAS, ADMINISTRATIVAS Y FÍSICAS PARA LA PROTECCIÓN DE DATOS PERSONALES EN POSESIÓN DE LA UNIVERSIDAD		
FUNDAMENTO	REQUISITO	SUGERENCIAS
Artículo 21	Realizar actividades del tratamiento de los datos personales sin ostentar poder alguno de decisión sobre el alcance y contenido del tratamiento.	Indicar que, en el caso de tratarse de servicios en la nube que impliquen sistemas automatizados, la normatividad universitaria solo permite el uso de nube pública para resguardo de datos personales siempre y cuando estén cifrados.

NORMATIVIDAD NACIONAL LEY GENERAL DE PROTECCIÓN DE DATOS PERSONALES EN POSESIÓN DE SUJETOS OBLIGADOS		
<p>Fundamento legal: Artículo 64.- Para el tratamiento de datos personales en servicios, aplicaciones e infraestructura de cómputo en la nube y otras materias, en los que el responsable se adhiera a los mismos mediante condiciones o cláusulas generales de contratación, sólo podrá utilizar aquellos servicios en los que el proveedor:</p>		
FUNDAMENTO	REQUISITO	SUGERENCIAS
Artículo 64, fracción I, inciso c)	<p>Cumpla, al menos, con lo siguiente:</p> <p>Abstenerse de incluir condiciones en la prestación del servicio que le autoricen o permitan asumir la titularidad o propiedad de la información sobre la que preste el servicio.</p>	<p><i>Indicar que no se autorizará que el proveedor asuma la titularidad o propiedad de la información sobre la que presta el servicio</i></p>
Artículo 64, fracción II, inciso a)	<p>Cuente con mecanismos, al menos, para:</p> <p>dar a conocer cambios en sus políticas de privacidad o condiciones del servicio que presta.</p>	<p><i>Pormenorizar en una cláusula por qué medios se dan a conocer los cambios, cada cuándo y a quién se dará a conocer dicha información.</i></p> <p><i>Se recomienda evitar aceptar cláusulas en las que se indique que los cambios se realizan en un sitio y que el proveedor no tiene la obligación de avisar a la dependencia o entidad universitaria.</i></p>
<p>Fundamento legal: Artículo 59.- La relación entre el responsable y el encargado deberá estar formalizada mediante contrato o cualquier otro instrumento jurídico que decida el responsable, de conformidad con la normativa que le resulte aplicable, y que permita acreditar su existencia, alcance y contenido.</p> <p>En el contrato o instrumento jurídico que decida el responsable se deberá prever, al menos, las siguientes cláusulas generales relacionadas con los servicios que preste el encargado:</p>		
FUNDAMENTO	REQUISITO	SUGERENCIAS
Artículo 59 fracción IV	<p>Informar al responsable cuando ocurra una vulneración a los datos personales que trata por sus instrucciones.</p>	<p><i>¿Cómo y a quién se informa, por qué medio, en cuánto tiempo se debe informar la vulneración? ¿Qué debe de contener el aviso?</i></p>

Adicionalmente, se recomienda la revisión de documentos que apoyan y fortalecen la selección y contratación de los proveedores de servicios de nube, elaborados por el Instituto Nacional de Transparencia, Acceso a la Información y Protección de Datos Personales, como sigue:

- Guía breve para Sujetos Obligados para la contratación de servicios de cómputo en la nube que impliquen el tratamiento de datos personales.
- Criterios mínimos para la contratación de servicios de cómputo en la nube que impliquen el tratamiento de datos personales.
- Conformidad de contratos de adhesión de servicios de cómputo en la nube vs los criterios mínimos para la contratación de servicios de cómputo en la nube que impliquen el tratamiento de datos personales.

Estas disposiciones se enfocan en características y buenas prácticas adoptadas por los proveedores, sin embargo, queda a consideración su uso, al no ser obligatorias, se observa:

- Facilidad para negociar términos y condiciones del servicio.
- Identificar los procesos y datos que se quieren migrar a la nube.
- Evaluar su reputación y considerar si tienen certificaciones en materia de seguridad de la información, manejo de incidentes, riesgos, entre otras.
- Priorizar aquellos proveedores que señalen la resolución de controversias en territorio nacional.
- Que los proveedores tengan buenas prácticas para informar sobre casos de vulnerabilidades y fallas en los servicios, así como de las acciones de remediación implementadas.

5. CONCLUSIONES

Identificar en la normatividad aplicable las acciones que deben cumplir los proveedores de servicios en la nube que realicen tratamiento de datos personales, e incorporarlas como obligaciones en los instrumentos consensuales, permitirá contar con reglas más precisas respecto de los alcances del servicio entre las partes implicadas y resolver controversias o situaciones de incumplimiento que se pudieran presentar.

6. AGRADECIMIENTOS

Agradezco las observaciones realizadas por la Mtra. Elizabeth Rangel Gutiérrez, Jefa de la Unidad Jurídica de la DGTIC, para impulsar la elaboración de este reporte y fortalecer el contenido expuesto.

REFERENCIAS BIBLIOGRÁFICAS

- Chávez Sánchez, J. L. (2022). *Tratamiento de datos personales en la nube, conforme el marco jurídico mexicano*. Revista Digital EDI, (39)15–22. https://issuu.com/elderechoinformatico.com/docs/revista_39
- Instituto Nacional de Transparencia, Acceso a la Información y Protección de Datos Personales. (2018). *Criterios mínimos sugeridos para la contratación de servicios de cómputo en la nube que impliquen el tratamiento de datos personales*. INAI. <https://home.inai.org.mx/wp-content/documentos/DocumentosSectorPrivado/ComputoEnLaNube.pdf>
- Instituto Nacional de Transparencia, Acceso a la Información y Protección de Datos Personales. (2021a). *Conformidad de contratos de adhesión de servicios de cómputo en la nube vs los criterios mínimos de contratación de servicios de cómputo en la nube que impliquen el tratamiento de datos personales*. INAI. https://home.inai.org.mx/wp-content/uploads/ContratosASCN_CN.pdf
- Instituto Nacional de Transparencia, Acceso a la Información y Protección de Datos Personales. (2021b). *Guía breve para sujetos obligados para la contratación de servicios de cómputo en la nube que impliquen el tratamiento de datos personales*. INAI. https://home.inai.org.mx/wp-content/uploads/Guia_SO_CC.pdf
- Ley General de Protección de Datos Personales en Posesión de Sujetos Obligados [LGPDPSSO], Nueva Ley, Diario Oficial de la Federación [DOF], 26 de enero de 2017, (México). <https://www.diputados.gob.mx/LeyesBiblio/pdf/LGPDPPSO.pdf>
- Meraz Espinoza, A. I. (2018). *Empresa y privacidad: el cuidado de la información y los datos personales en medios digitales*. Revista Ius, 12(41), 293–310. <https://doi.org/10.35487/rius.v12i41.2018.313>
- Universidad Nacional Autónoma de México. (2019). *Acuerdo por el que se establecen los Lineamientos para la Protección de Datos Personales en posesión de la Universidad Nacional Autónoma de México*, Gaceta UNAM. 25 de febrero de 2019, 1-9. https://www.red-tic.unam.mx/recursos/2019/2019_Acuerdo_Rectoria_02.pdf
- Universidad Nacional Autónoma de México. (2020). *Normas complementarias sobre medidas de seguridad técnicas, administrativas y físicas para la protección de datos personales en posesión de la Universidad Nacional Autónoma de México*, Gaceta UNAM. https://www.red-tic.unam.mx/recursos/2020/2020_Norma_ComiteTransparencia_01.pdf
- Universidad Nacional Autónoma de México. (2022). *Reglamento de Transparencia y Acceso a la Información Pública de la Universidad Nacional Autónoma de México*, Gaceta UNAM., 21 de septiembre de 2022, 11-22 https://www.red-tic.unam.mx/recursos/2022/2022_Reglamento_ComTransp_01.pdf

Interfaces web dinámicas: solución a las limitaciones que presentan *Vue* y *React*

Información del reporte:

Licencia Creative Commons



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

Para citar este reporte técnico:

Zenil Rivas, P. E. (2023). Interfaces web dinámicas: solución a las limitaciones que presentan *Vue* y *React*. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (25 - 34).

<https://doi.org/10.22201/dgtic.ctud.2023.1.1.22>

Pablo Enrique Zenil Rivas

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

pablozr@unam.mx

ORCID: 0009-0007-3956-0194

Resumen:

Las herramientas como *Vue* y *React* son muy populares en el desarrollo eficiente y escalable de Interfaces de Usuario; son utilizadas para implementar aplicaciones de gran tamaño y complejidad. Sin embargo, cuando se requiere de una experiencia de usuario altamente dinámica y responsiva, estas herramientas presentan algunas limitaciones como: la comunicación entre componentes, el registro de cambios en tiempo real y la manipulación directa del *DOM*. En este reporte técnico se propone e implementa la solución a las limitaciones que se presentaron en el desarrollo del proyecto *Laboratorio Virtual de Química*. La solución conjunta, además del despliegue declarativo de componentes reactivos, los paradigmas de desarrollo web: comunicación vía emisor-suscriptor de eventos y manipulación del Modelo de Objetos del Documento *HTML (DOM)*. Se implementó una interfaz web intuitiva, altamente dinámica y responsiva a las interacciones del usuario, que permite la comunicación, interacción y seguimiento en tiempo real de los componentes que representan los distintos materiales y sustancias del laboratorio.

Palabras clave:

Interfaces web dinámicas, *Vue*, *React*.

1. INTRODUCCIÓN

Las herramientas como *Vue* (Vue team, 2022) y *React* (React team, s./f.) cubren la mayoría de las necesidades de un sistema o aplicación en el desarrollo de interfaces gráficas web. Estas herramientas se basan en los estándares de *HTML*, *CSS* y *JavaScript* y proveen un modelo de programación declarativo (Wikipedia contributors, 2023a) basado en componentes reactivos, que permite el desarrollo eficiente y escalable de interfaces de usuario (W3Schools, s.f). Lo anterior las hace ideales para implementar aplicaciones de gran tamaño y complejidad. Por esta razón, cada vez más desarrolladores y equipos de desarrollo utilizan este tipo de herramientas.

Hay dos conceptos centrales que se deben tomar en cuenta al utilizar estas herramientas:

- **Despliegue declarativo.** Describe el estado final de los componentes basados en *HTML* para cualquier estado de *JavaScript*.
- **Reactividad.** Cada componente realiza el seguimiento de los cambios en sus dependencias reactivas (el estado de *JavaScript*), y cuando suceden, se actualiza de forma eficiente. Estos cambios se ven reflejados en el Modelo de Objetos del Documento (MDN Web Docs, 2019). A este tipo de componentes se les llama componentes reactivos.

La interfaz de usuario se compone de unidades pequeñas como botones, texto e imágenes. Esto permite combinarlas en componentes anidables y reutilizables. Desde sitios web hasta aplicaciones de celular, toda interfaz puede descomponerse en componentes reactivos. Cada componente encapsula en un bloque su comportamiento, sus datos, su lógica y su estado, lo que lo hace reutilizable y fácil de mantener.

Sin embargo, cuando se requiere de una experiencia de usuario altamente dinámica, estas herramientas presentan las siguientes limitaciones:

- **Comunicación entre componentes.** El flujo de la comunicación entre componentes solamente se da entre padre e hijo. Los componentes anidados que no guardan esta relación, no pueden comunicarse directamente entre sí.
- **Registro de cambios continuos en los componentes.** La reactividad de cada componente lleva el seguimiento de cambios discretos, es decir, cambios que se derivan de eventos aislados como las interacciones del usuario (presionar un botón, teclear texto, entre otros). Los cambios continuos en el tiempo, como posiciones derivadas de una transición animada o la variación de temperatura en una simulación, requieren de un tratamiento especial.
- **Manipulación directa del Modelo de Objetos del Documento *HTML* (*DOM*).** El despliegue declarativo y la reactividad de los componentes se encargan de la manipulación del *DOM*. Sin embargo, algunas interacciones de usuario como arrastrar y soltar, implican una manipulación directa del *DOM* que debe ser implementada de manera independiente.

El presente reporte técnico describe el proceso de solución de las limitaciones anteriormente mencionadas, que se implementó en el desarrollo del proyecto *Laboratorio Virtual de Química*: un ambiente web de simulación que contribuye a promover la experimentación para la resolución de problemas de diluciones en agua destilada y titulación de ácidos/bases fuertes (De Levie, 2001; McCord y Stanton, 2005).

2. OBJETIVO

Implementar una interfaz web intuitiva, altamente dinámica y responsiva a las interacciones del usuario que permita la comunicación, interacción y seguimiento en tiempo real de los componentes que representan los distintos materiales y sustancias del Laboratorio Virtual de Química y proponer una solución a las limitaciones entre componentes reactivos que presentan las herramientas como *Vue* y *React*.

3. DESARROLLO

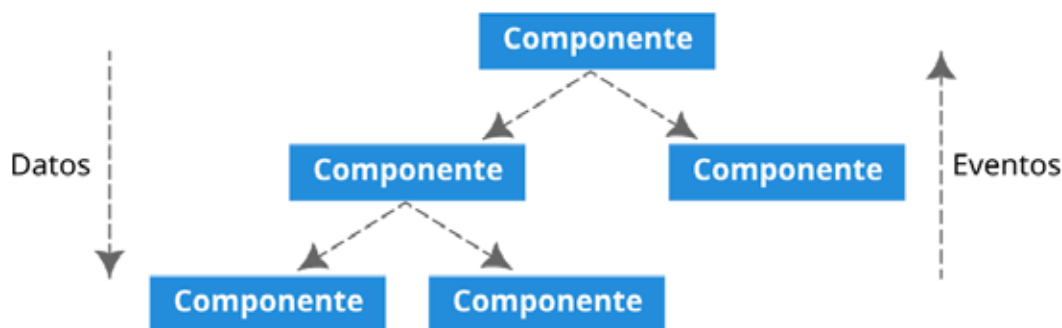
El tipo de interacción que se necesitó implementar implica que el usuario pueda posicionar distintos materiales en la mesa de trabajo, arrastrar y soltar componentes unos sobre otros, para activar interacciones entre ellos, por ejemplo, el vaciado de sustancias y la medición de sus propiedades fisicoquímicas.

3.1 COMUNICACIÓN ENTRE COMPONENTES

En los componentes anidados, la comunicación se da entre padre e hijos, como se muestra en la figura 1.

Figura 1

Flujo de comunicación entre componentes



Nota. La comunicación entre componentes, con este paradigma, se da de manera directa

Cada hijo recibe datos directamente de su padre, y le responde mediante eventos. Esto limita la interactividad a los elementos que tienen una relación de padre e hijo. Si bien la reactividad y el despliegue declarativo de los componentes facilitan el desarrollo de sistemas complejos, al integrar al árbol nuevos componentes, muy probablemente romperán con la relación padre-hijo de otros ya establecidos y, con esto, su comunicación. Una solución a este problema es proveer un mecanismo de comunicación directa entre cualquier componente. Esto se logra mediante la implementación del paradigma de emisor-suscriptor de eventos (Wikipedia contributors, 2023b). En este paradigma, un componente, a través de un objeto global Emisor, emite eventos con un nombre definido, lo que ocasiona que los componentes receptores previamente suscritos a ese evento dado ejecuten una acción pertinente.

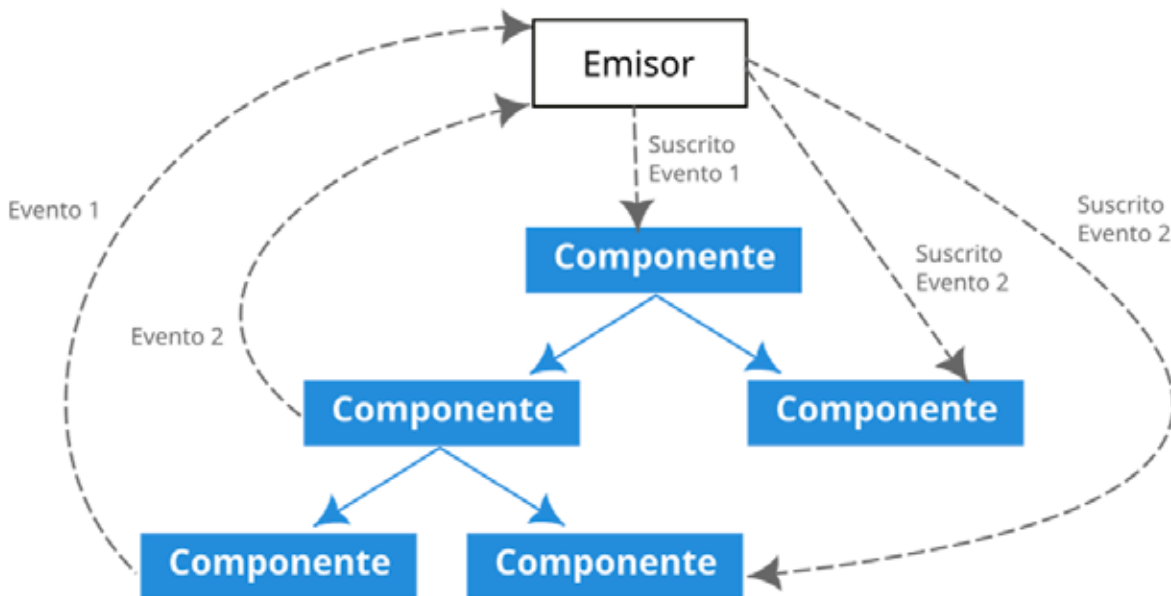
El objeto emisor cumple con dos funciones:

- Emitir eventos.
- Registrar y eliminar funciones receptoras.

Lo anterior habilita la comunicación directa entre cualquier par de componentes. Por ejemplo, como se muestra en la figura 2, un componente es capaz de emitir el Evento 1 y otro componente distinto emite el Evento 2. Los componentes que requieran reaccionar a estos eventos se suscriben al emisor para escuchar a cada uno de ellos. Los componentes se pueden suscribir y también pueden emitir cualquier número de eventos.

Figura 2

Utilizando el paradigma emisor-suscriptor



Nota. La comunicación entre componentes, con este paradigma, se da de manera directa.

La limitante de la comunicación entre componentes no anidados queda resuelta, y se eleva así el grado de interacción. Bajo este paradigma, la complejidad y el número de componentes puede escalar fácilmente, ya que pueden ser incorporados en cualquier nivel del árbol sin romper la comunicación de componentes ya establecidos. El emisor puede ser implementado de manera nativa con *JavaScript*, sin embargo, en el desarrollo del laboratorio se utilizó la herramienta de terceros *Mitt* (Miller, s/f), al ser una implementación ligera, funcional y fácil de utilizar.

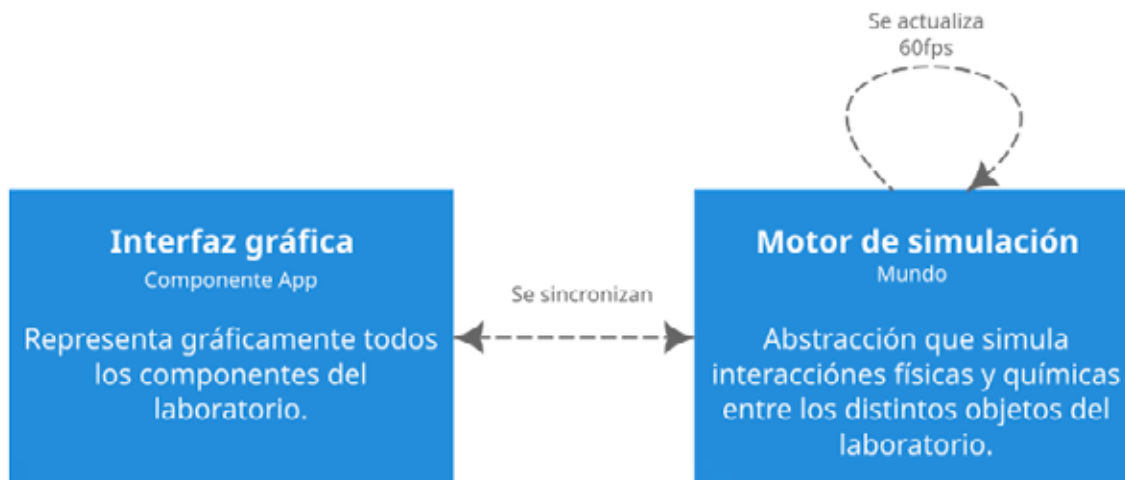
3.2 REGISTRO DE CAMBIOS CONTINUOS EN LOS COMPONENTES

La aplicación consta de dos elementos principales, como se muestra en la figura 3:

- 1. Motor de simulación de reacciones químicas.** Abstrae al mundo, el cual modela a las distintas sustancias en términos de su composición química mediante la *Molaridad* (De Levie, 2001; McCord & Stanton, 2005), temperatura y volumen. El simulador depende del bucle de ejecución del navegador y 60 veces por segundo, calcula en tiempo real los avances de reacción (De Levie, 2001; McCord & Stanton, 2005; Saker Neto, 2018), entalpía de mezclado/reacción (De Levie, 2001; McCord & Stanton, 2005) y enfriamiento (Downey, 2022), para poder representar de forma dinámica y realista las distintas propiedades fisicoquímicas, como pH y temperatura, de las sustancias y contenedores que interactúan entre sí.
- 2. Interfaz gráfica.** Tiene como raíz al componente *App* y representa gráficamente, mediante componentes, a los elementos del mundo, así como sus propiedades fisicoquímicas. Además, permite que el usuario los manipule e interactúe con ellos.

Figura 3

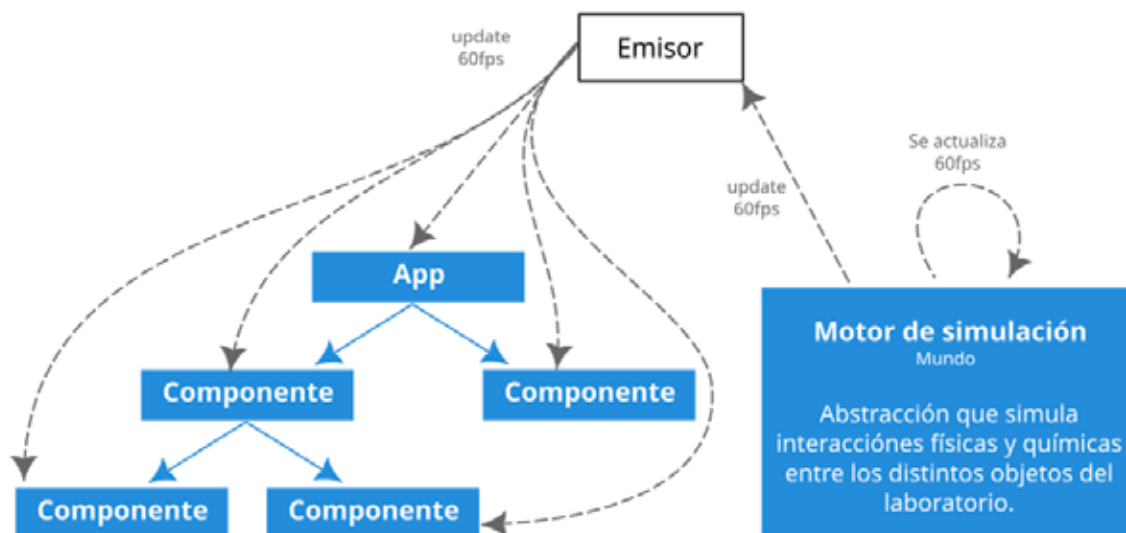
Elementos principales de la aplicación



Tras haber implementado el emisor, resolver la sincronización es relativamente fácil, como se ilustra en la figura 4, pues basta con suscribir a todos los componentes al evento *update*, emitido por el mundo (60 veces por segundo), que es el encargado de notificar que se ha ejecutado un cuadro de tiempo del bucle de ejecución. Aquí cabe resaltar que no solo los componentes reactivos pueden emitir y escuchar eventos, sino que también otros elementos (como el mundo) de la aplicación pueden comunicarse con los componentes.

Figura 4

Sincronización entre el motor de simulación y los componentes



Nota. El emisor se encarga de notificar a todos los componentes para que se actualicen en tiempo real.

3.3 INTERACCIONES ENTRE COMPONENTES

Los componentes principales representan materiales de laboratorio como el vaso de precipitado, matraz, pipeta, entre otros. El usuario puede arrastrarlos y colocarlos en cualquier parte de la mesa de trabajo que es el componente padre y representa casi toda el área visible de la ventana del navegador. Cada envase puede contener un volumen determinado de alguna sustancia y al ser arrastrado y soltado sobre otro envase, se puede verter parte de esa sustancia. Parte importante de la interacción del usuario está en arrastrar y soltar los componentes de la mesa de trabajo. Cada componente notifica los siguientes eventos a cada suscriptor, los que se derivan de la interacción del usuario:

- Se está arrastrando.
- Ha sido soltado.
- Le ha sido soltado otro componente.
- Ha sido seleccionado (se presionó sobre él sin ser arrastrado).

Además, también notifica en tiempo real los siguientes eventos del mundo:

- Cuánto volumen de sustancia contiene (si es envase).
- La temperatura de la sustancia que contiene (si es envase).
- La molaridad de la sustancia que contiene (si es envase).

- Los valores de sensor (pH en el caso del pH-metro).
- El ángulo de inclinación (si aplica).
- Cuánto volumen de sustancia está vertiendo (si es envase).
- En qué zona de la mesa de trabajo se encuentra.

Cada componente suscriptor reacciona al evento que le corresponde. Por ejemplo, si un vaso de precipitado es arrastrado y soltado en un matraz, el matraz sabe que puede recibir la sustancia del vaso y a su vez, el vaso sabe a quién verter la sustancia.

El arrastrado y soltado de componentes es una interacción especial que requiere del paradigma de manipulación del *DOM*, que nos permite: agregar y eliminar elementos al documento *HTML*, modificar atributos de elementos y ejecutar funciones que reaccionan a los distintos eventos, entre otros. En la implementación de esta interacción en el laboratorio, utilizamos la herramienta de terceros *jQuery* (*jQuery team, s/f*), una herramienta eficiente y ligera que facilita la manipulación del *DOM*.

Con todo lo anterior, solo falta asegurar que todos los componentes representen gráficamente su estado en tiempo real (volumen de líquido, inclinación, temperatura, color), como se muestra en la figura 5. Lo anterior se logra al actualizar el estado visual de cada componente en el evento *update*, manipulando la parte gráfica vía *CSS* y con la herramienta de terceros *Paper.js* (*Lehni & Puckey, s/f*), la cual provee una funcionalidad ligera y muy poderosa en la creación y manipulación de gráficos.

Figura 5

Representación gráfica en tiempo real, del estado de algunos componentes



4. RESULTADOS

Se implementó el proyecto Laboratorio Virtual de Química donde se usó *Vue* para el manejo de componentes reactivos. El sistema cuenta con una interfaz web intuitiva, altamente dinámica y responsiva a las interacciones del usuario, que permite la comunicación, interacción y seguimiento en tiempo real de los componentes que representan los distintos materiales y sustancias del laboratorio; es el resultado de la conjunción de tres principales paradigmas de programación web:

- 1.Despliegue declarativo de componentes reactivos:** permite el desarrollo de una interfaz intuitiva y responsiva a los eventos de usuario, mediante el encapsulamiento de cada componente y su funcionalidad.
- 2. Comunicación mediante el emisor-suscriptor de eventos:** permite una comunicación directa entre componentes y elementos del sistema, lo que eleva el nivel de interacción y escalabilidad del mismo.
- 3. Manipulación directa de los atributos y propiedades del *DOM*:** permite el desarrollo de interacciones que el uso de componentes reactivos no contempla, como el arrastrado y soltado de componentes.

La experiencia en el desarrollo de sistemas web permitió proponer una solución sencilla y escalable en la que:

- Nuevos componentes reactivos como materiales, menús o ventanas emergentes, pueden ser incorporados fácilmente en cualquier nivel del árbol de componentes, sin interferir en lo que ya es funcional.
- Nuevos elementos que representen objetos del mundo como sustancias, sensores o materiales, pueden ser incorporados con la misma facilidad.
- Se pueden definir nuevos eventos e interacciones sin interferir con lo ya implementado.
- Cualquier nuevo componente o elemento solo tiene que definir y/o suscribirse a los eventos de las interacciones que le corresponden e implementarlas según sea el caso.
- Cabe destacar que esta solución es aplicable en cualquier escenario en el que se requiera usar cualquier tecnología basada en componentes reactivos. Una captura de pantalla del producto final se puede apreciar en la figura 6.

Figura 6

Captura de pantalla del Laboratorio Virtual de Química



5. CONCLUSIONES

Herramientas basadas en componentes reactivos como *Vue* y *React*, cubren la mayoría de las necesidades de un sistema o aplicación en el desarrollo de interfaces gráficas web. Sin embargo, cuando se requiere de una experiencia de usuario altamente dinámica y responsiva, estas herramientas presentan las siguientes limitaciones:

- **Comunicación entre componentes:** los componentes anidados que no guardan una relación de padre e hijo no pueden comunicarse directamente entre sí.
- **Registro de cambios continuos en los componentes:** los cambios continuos en el tiempo, como la variación de temperatura en una simulación, requieren de un tratamiento especial.
- **Manipulación directa del DOM:** las interacciones de usuario como el arrastrado y soltado, implican una manipulación directa del DOM que debe ser implementada de manera independiente.

En el proyecto Laboratorio Virtual de Química, la experiencia en el desarrollo de sistemas web permitió proponer e implementar una solución a dichas limitaciones, sencilla, escalable y que conjunta los siguientes paradigmas de desarrollo web:

- 1.Despliegue declarativo de componentes reactivos:** permite el desarrollo de una interfaz intuitiva y responsiva a los eventos de usuario.
- 2. Comunicación mediante el emisor-suscriptor de eventos:** permite una comunicación directa entre componentes y elementos del sistema.
- 3. Manipulación directa de los atributos y propiedades del DOM:** permite el desarrollo de interacciones que no contemplan el uso de componentes reactivos.

Lo anterior permitió implementar una interfaz web intuitiva, altamente dinámica y responsiva a las interacciones del usuario, que permite la comunicación, interacción y seguimiento en tiempo real de los componentes que representan los distintos materiales y sustancias del laboratorio. Esta solución es aplicable en cualquier escenario en el que se requiera usar una tecnología similar basada en componentes reactivos y que presente las mismas limitaciones.

REFERENCIAS BIBLIOGRÁFICAS

- De Levie, R. (2001). *How to use Excel in analytical chemistry and in general scientific data analysis*. Cambridge University Press.
- Downey, A. (2022). *Modeling And Simulation In Python*. O'reilly Media.
- jQuery team. (s.f.). *jQuery*. Recuperado el 7 de septiembre de 2023, de <https://jquery.com>
- Lehni, J., & Puckey, J. (s.f.). *Paper.js*. paperjs.org. Recuperado el 7 de septiembre de 2023, de <http://paperjs.org>
- McCord, P., & Stanton, E. (2005). *pH, Titrations, and Dilutions. Course: General Chemistry I; The University of Texas at Austin*. <http://mccord.cm.utexas.edu/courses/spring2005/ch301/concentrations.html>
- MDN Web Docs. (2019, agosto 16). *Introduction to the DOM*. MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction
- Miller, J. (s.f.). *Mitt*. GitHub. Recuperado el 7 de septiembre de 2023. <https://github.com/developit/mitt>
- React team. (s.f.). *Describing the UI – React*. react.dev. Recuperado el 7 de septiembre de 2023. <https://react.dev/learn/describing-the-ui>
- Saker Neto, N. (2018). *The reason behind the steep rise in pH in the acid base titration curve*. Chemistry Stack Exchange. <https://chemistry.stackexchange.com/q/8074>
- Vue team. (2022). *Introduction | Vue.js*. vuejs.org. <https://vuejs.org/guide/introduction.html>
- W3Schools. (s.f.). *What is a Front-End Developer*. www.w3schools.com. Recuperado el 7 de septiembre de 2023. https://www.w3schools.com/whatis/whatis_frontenddev.asp
- Wikipedia contributors. (2023a, septiembre 7). *Declarative programming*. Wikipedia. https://en.wikipedia.org/w/index.php?title=Declarative_programming&oldid=1174296946
- Wikipedia contributors. (2023b, septiembre 7). *Event-driven programming*. Wikipedia. https://en.wikipedia.org/w/index.php?title=Event-driven_programming&oldid=1174274696

Gestión de activos de la infraestructura de cómputo

Información del reporte:

Licencia Creative Commons



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

Para citar este reporte técnico:

González Trejo, M. (2023). Gestión de activos de la infraestructura de cómputo mediante GLPI. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (35 - 51).

<https://doi.org/10.22201/dgtic.ctud.2023.1.1.14>

Margarita González Trejo

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

mar@unam.mx

ORCID: 0000-0002-6682-8420

Resumen:

La gestión de activos de la infraestructura de cómputo es una actividad que toma especial relevancia cuando forma parte de la prestación de servicios de educación continua, particularmente por el volumen de equipos que se administra. Conocer en detalle sus características, configuración y localización para poder consultarlas en forma rápida y precisa, permite ubicar en qué estado dentro del ciclo de vida se encuentran los activos; determinar su uso para proyectos presentes; identificar necesidades a futuro con el fin de actualizar componentes; identificar otros usos o realizar su baja cuando ya no satisfacen los requerimientos de las nuevas tecnologías de los sistemas operativos o aplicaciones de *software*. Son numerosas las aplicaciones que incrementan la eficiencia de estas tareas, al recopilar automáticamente la información de los activos de cómputo y concentrarlos en bases de datos que pueden visualizarse vía web, de manera inmediata. La efectividad de la implementación de estas herramientas se basa en una clara visión de las necesidades a atender y en una sólida evaluación para seleccionar la que mejor se ajuste a los requerimientos.

Palabras clave:

Infraestructura de cómputo, administración de activos, GLPI.

1. INTRODUCCIÓN

Mantener un inventario actualizado de los activos que conforman la infraestructura de cómputo es una problemática frecuente en las áreas de tecnologías de información (TI) responsables de su administración y resguardo, que se manifiesta en aspectos como los errores de captura debidos al factor humano, o la falta de actualización de datos y de medios para concentrarlos en un mismo formato que los visibilice de manera oportuna. Como consecuencia, la elaboración de reportes de evaluación técnica de la infraestructura se transforma en un proceso lento y, en ocasiones, poco confiable debido a la falta de información precisa.

Esta problemática puede resolverse mediante la implementación de aplicaciones para la gestión de servicios de TI que permiten automatizar la captura del inventario, dar seguimiento al ciclo de vida del equipo y mantener organizada la información, además de proporcionar otras funcionalidades como el seguimiento a los servicios de soporte técnico. La experiencia en la selección de una herramienta para la gestión de activos de infraestructura de cómputo se expone en este documento, mismo que se ha dividido en cuatro secciones: antecedentes, problemática a resolver, fundamentos teóricos y propuesta de solución.

2.OBJETIVO

Realizar la implementación de la aplicación de gestión de servicios *GLPI*, como herramienta para incrementar la eficiencia de la administración y el levantamiento del inventario de activos de infraestructura de cómputo.

3. APLICACIONES PARA LA GESTIÓN DE ACTIVOS

3.1 ANTECEDENTES

Las áreas de TI especializadas en proporcionar servicios de infraestructura y soporte técnico para programas de educación continua en tecnologías de información y comunicación, tienen como una de sus principales funciones gestionar, administrar y mantener en óptimas condiciones de uso la infraestructura de cómputo tanto de áreas operativas como en aulas de capacitación. De acuerdo con las necesidades de los usuarios, estas aulas pueden ser: fijas para los programas que se imparten de manera presencial, móviles para los programas impartidos en las instalaciones de empresas o instituciones que así lo solicitan, o virtuales para los programas a distancia que hacen uso de escritorios remotos. Regularmente, el equipamiento de cómputo es de diferentes características y capacidades, su distribución obedece a los requerimientos de *hardware* y *software* específicos para cada curso, por lo que suele tener alta movilidad y el mantenimiento al *software* instalado debe ser permanente. Tener un control total y preciso de todos estos elementos puede representar un reto importante que hace necesaria la incorporación de alguna aplicación para la automatización de las tareas y para la presentación de la información.

3.2 PROBLEMA A RESOLVER

Se consideraron los antecedentes presentados en el punto anterior, para determinar los elementos deseables de la herramienta a incorporar para resolver la problemática de la gestión de los activos de la infraestructura de cómputo:

- Automatizar el levantamiento del inventario de la infraestructura de cómputo.
- Organizar la información por áreas funcionales para facilitar la administración de los activos.
- Resguardar la información en una base de datos.
- Facilitar la visualización de los datos contenidos.
- Contar con métodos para la seguridad de la información.
- Preferentemente, hacer uso de aplicaciones de *software* libre.

Para lograr cumplir con estos objetivos se evaluaron tres herramientas de *software* libre cuyos resultados se describen a continuación.

3.3 FUNDAMENTOS TEÓRICOS

Los estándares oficiales para la gestión de activos de TI están definidos por la Organización Internacional de Normalización (*ISO*, por las siglas en inglés de *International Organization for Standardization*) en la familia *ISO 19770*, derivada de la norma *ISO 55000* en la cual se definen los aspectos generales, principios y terminología para la gestión de activos (*ISO*, 2023).

Por su parte, el marco de referencia de mejores prácticas *ITIL* (2019) en su versión 4 destaca la importancia de la gestión de activos a través de la Práctica de Gestión de activos de *TI*¹ (*AXELOS*, 2023), cuyo propósito es *Gestionar el Ciclo de Vida de los Activos de TI* para:

- Maximizar el valor de los activos.
- Controlar el gasto.
- Gestionar el riesgo.
- Soportar la toma de decisiones.
- Cumplir con los requerimientos contractuales y normativos de la organización.

El modelo también pone de manifiesto la importancia que tiene para las organizaciones la incorporación de tecnologías para la automatización de tareas a través del principio *Optimizar y Automatizar* (*Invgate*, 2023).

3.4 METODOLOGÍA APLICADA

De entre el universo de herramientas de *software* libre para la gestión de activos de TI, se encontraron tres con las cuales se realizaron pruebas de funcionamiento:

1. *FusionInventory*
2. *GLPI*
3. *OCS Inventory*

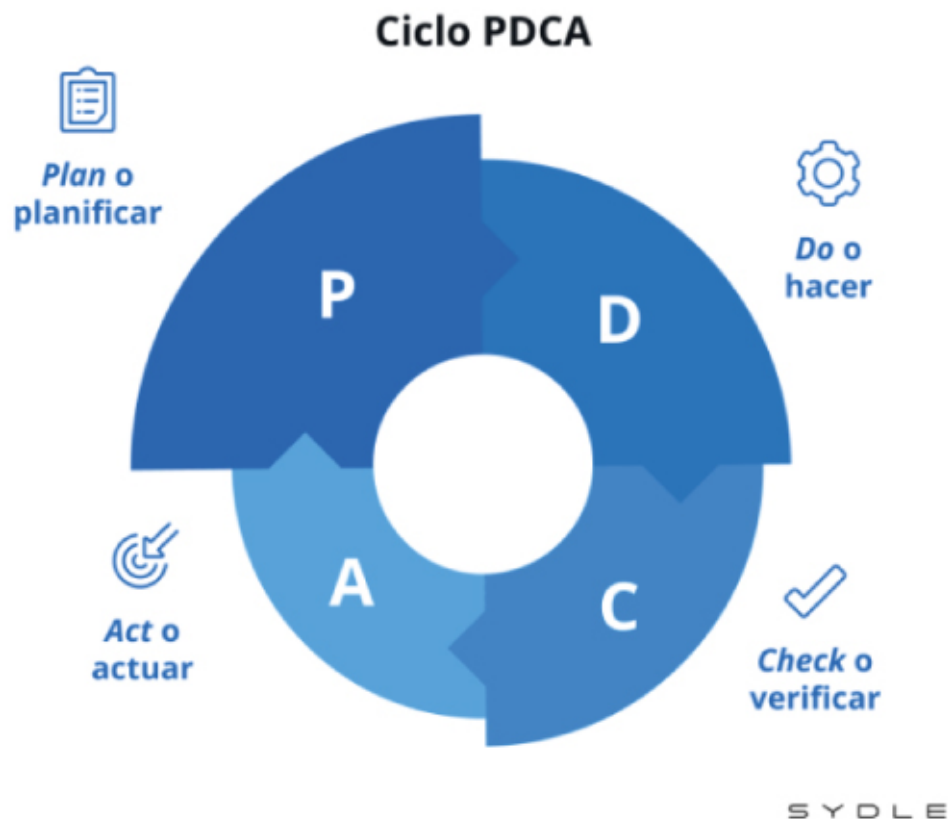
¹ En el modelo *Information Technology Infrastructure Library (ITIL)* en su versión 4 se define una Práctica como “un conjunto de recursos organizativos diseñados para realizar un trabajo o cumplir un objetivo (*AXELOS*, 2023)”.

Las actividades de soporte técnico y aprovisionamiento de los recursos de infraestructura de cómputo (*hardware* y *software*) para dar atención a las áreas operativas y a los programas de educación continua en TI, requieren de una actualización técnica constante, especialmente por la amplia variedad de temáticas de los programas académicos y la velocidad de desarrollo de las tecnologías aplicadas. La implementación de metodologías como el ciclo de *Deming* hacen posible establecer un sistema de mejora continua que, en este caso en particular, permitió realizar la evaluación de las herramientas de *software* para la gestión de los activos de infraestructura de cómputo.

El ciclo de Deming o PDCA consta de cuatro fases: *Plan* (planear), *Do* (hacer), *Check* (verificar) y *Act* (actuar) que se ejecutan de manera iterativa con el propósito de acelerar y mejorar la calidad de los procesos, lo cual se muestra en la figura 1.

Figura 1

Ciclo Deming



Nota: Adaptado de SYDLE (2023) Ciclo PDCA: ¿cuáles son los pasos y cómo funciona? Conoce algunos ejemplos | (<https://www.sydle.com/blog/assets/post/ciclo-pdca-61ba2a15876cf6271d556be9/ciclo-pdca.png?w=720>).

La forma en la cual se aplicó esta metodología en la evaluación de las herramientas de *software* para la gestión de activos de infraestructura de cómputo se explica en la tabla 1.

Tabla 1

Aplicación del ciclo PDCA a la evaluación de las herramientas de software

Fase ciclo PDCA	Aplicación en la evaluación de las herramientas
Planear	Definición de los objetivos de la herramienta para la gestión de activos
Hacer	Implementación de las herramientas y pruebas de funcionamiento
Verificar	Verificación del funcionamiento de la herramienta seleccionada de acuerdo con los objetivos propuestos
Actuar	Implementación de la herramienta, e inicio nuevamente del ciclo para evaluar las mejoras de nuevas versiones de la herramienta

Como resultado de las evaluaciones se obtuvo lo siguiente:

OCS Inventory (Open Computer and Software Inventory Next Generation) es un *software* libre para la gestión del inventario de activos de *TI* que obtiene la información del *hardware* y *software* de equipos en una red a través de un agente denominado *OCS*, (*OCS Inventory*, 2023). Es una aplicación que requiere de un servidor, dentro de sus principales características se encuentran:

- Sistema para implementar instalaciones de *software* o ejecutar comandos en computadoras en red.
- Detección de redes.
- Soporte para múltiples sistemas operativos.
- Plugin para integración con *GLPI* para ampliación de sus funcionalidades.

FusionInventory. Software para realizar el inventario del *hardware* y *software* de activos de *TI* (*FusionInventory*, 2023). Funciona a través de agentes que son instalados como complementos de operación en servidores *GSIT* y *GLPI*, sus características más relevantes son las siguientes:

- Detección de redes.
- Inventario de conmutadores.
- Implementación de *software*.
- Inventario de máquinas virtuales y contenedores.

GLPI (Gestionnaire Libre de Parc Informatique) es un *software* de gestión de servicios de código abierto desarrollado en Francia en el año 2003 por la asociación *INDEPNET* y desde 2009 es la empresa *Teclib* quien se hace cargo del desarrollo de la aplicación. Permite realizar el inventario de activos de *TI* mediante el agente *GLPI Inventory* (*GLPI*, 2023). Actualmente *GLPI* cuenta con diversas funcionalidades como son:

- *CMDB* para la Gestión de *hardware*, *software* y centros de datos.
- Mesa de ayuda.
- Herramientas para gestión de gastos, contratos, proveedores, licencias de *software*.
- Gestión de proyectos mediante *Kanban board*, diagrama de *Gantt*.

3.5 PROPUESTA DE SOLUCIÓN

El resultado de las evaluaciones realizadas permitió conocer que tanto *FusionInventory* como *OCS Inventory* cumplen con el requisito de levantamiento de inventario automatizado, sin embargo sólo *GLPI* tiene todas las características solicitadas en el punto 3.2; además cuenta con mesa de ayuda y gestión de proyectos que permiten que el objetivo inicial de gestión de activos de cómputo evolucione y favorezca a otros servicios tales como la asistencia técnica basada en la creación de *tickets*, la gestión de licencias de *software* y la construcción de conocimiento basado en la información de los incidentes atendidos.

Una vez que se encontró que la herramienta *GLPI* es la que satisface de mejor manera los objetivos establecidos, se procedió a realizar su implementación para el levantamiento automatizado del inventario de la infraestructura de cómputo.

3.5.1. REQUISITOS DE INSTALACIÓN

GLPI es una aplicación web disponible para su descarga en: <https://glpi-project.org/es/>, para su instalación requiere de un servidor que cumpla con los siguientes requisitos:

- Servidor web: Apache 2 mínimo, compatible con *PHP* versión 8 o superior.
- Gestor de base de datos *MySQL* o *MariaDB*.
- Extensiones *PHP* esenciales para el correcto funcionamiento de *GLPI*: *Sesión*, *MySQL*, *JSON* (*JavaScript Object Notation*), *Mbstring* (*Cadena MultiByte*), *GD* (*Gif Draw*).

3.5.2 PROCEDIMIENTO DE INSTALACIÓN

La instalación se realizó en un servidor con sistema operativo *Linux CentOS 7*, procesador *Intel(R) Core(TM) i3-4150 CPU @ 3.50GHz*, 8 GB en memoria *RAM* y disco duro *SATA* de 250 GB; se llevó a cabo el siguiente procedimiento:

1. Instalar herramientas y utilidades.

```
$ sudo yum install -y policycoreutils-python wget
```

2. Descargar *GLPI*.

```
$ wget -q https://github.com/glpi-project/glpi/releases/download/10.0.6/glpi-10.0.6.tgz
```

3. Descomprimir y copiar el programa en la ubicación correcta.

```
$ sudo tar xf glpi-10.0.6.tgz -C /var/www/html
```

4. Proporcionar los permisos de escritura al directorio y archivos de configuración.

```
$ chown -R apache:apache /var/www/html/glpi
```

```
$ chmod -R 777 /var/www/html/glpi
```


5. Instalar extensiones de *PHP* necesarias para *GLPI*.

```
$ sudo yum install -y php-{gd,imap,intl,ldap,mbstring,opcache,pear-CAS,pecl-apcu,xmlrpc,pecl-zip}
```

6. Crear la configuración para *GLPI* en *Apache*.

```
$ sudo nano /etc/httpd/conf.d/glpi.conf
```

Con el contenido:

```
<Directory /var/www/html/glpi>
```

```
AllowOverride all
```

```
</Directory
```

7. Crear la base de datos y el usuario para *GLPI*.

```
$ mysql -u root -p
```

```
> create database glpi charset utf8mb4 collate utf8mb4_unicode_ci;
```

```
> create user glpi@localhost identified by 'X';
```

```
> grant all privileges on glpi.* to glpi@localhost;
```

```
> grant select on mysql.time_zone_name to glpi@localhost;
```

```
> exit
```

8. Iniciar la instalación de *GLPI*, desde un navegador web se indica la URL del servidor añadiendo la ruta */glpi*, el proceso de instalación iniciará como se observa en la figura 2.

Figura 2

Inicio de Instalación de GLPI



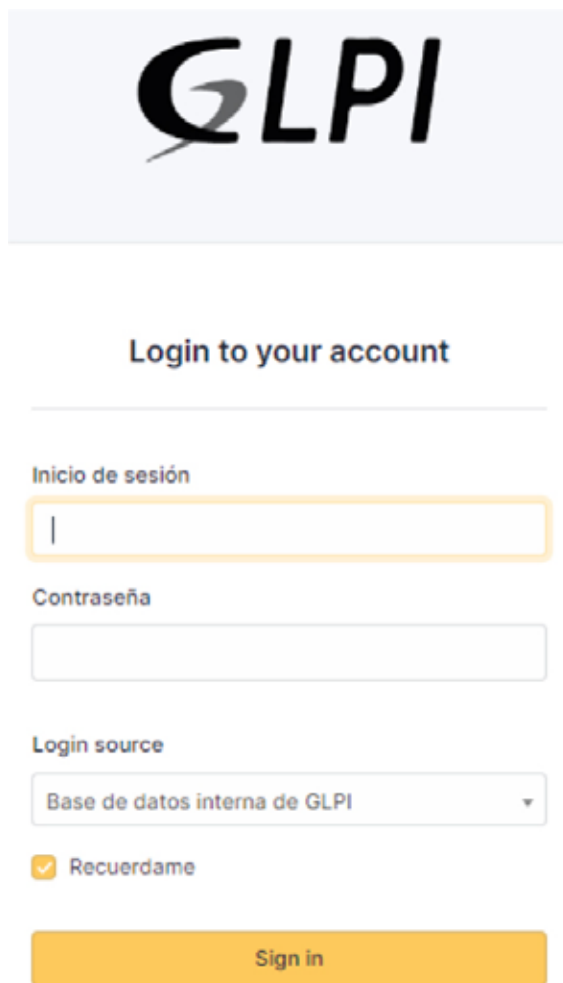
- 8.1. Seleccionar el idioma de instalación.

- 8.2. Aceptar los términos de licencia.

- 8.3. *GLPI* verifica los requisitos del sistema, si no encuentra problema continua con el proceso.
- 8.4. Se ingresan los datos de acceso al servicio de base de datos y de ser correctos procede con la instalación.
9. Finalizada la instalación se presenta la página donde se solicitan los datos de inicio de sesión del usuario, como se muestra en la figura 3. *GLPI* permite crear usuarios con perfiles de acceso asociados al rol que se tendrá en la operación del sistema.

Figura 3

Inicio de sesión en GLPI



GLPI

Login to your account

Inicio de sesión

Contraseña

Login source

Base de datos interna de GLPI

Recuérdame

Sign in

10. Iniciar la configuración de *GLPI* con el registro de las entidades en el menú Administración. *GLPI* basa su operación en *Entidades*, es decir, subconjuntos que reflejan la forma en la cual está organizada la infraestructura en una organización, y las presenta en forma de árbol.

Figura 4

Configuración de Entidades en GLPI



11. Instalar el agente en un equipo de prueba: en este punto es muy importante verificar que se ha indicado correctamente la ruta del servidor para el registro del inventario en la base de datos de GLPI.
12. Verificar que la información del inventario del equipo de prueba se ha ingresado con éxito, desde el menú *Activos -> Computadoras*. Un ejemplo de la información recolectada por GLPI se presenta en la figura 5.

Figura 5

Información del Inventario de una computadora en GLPI

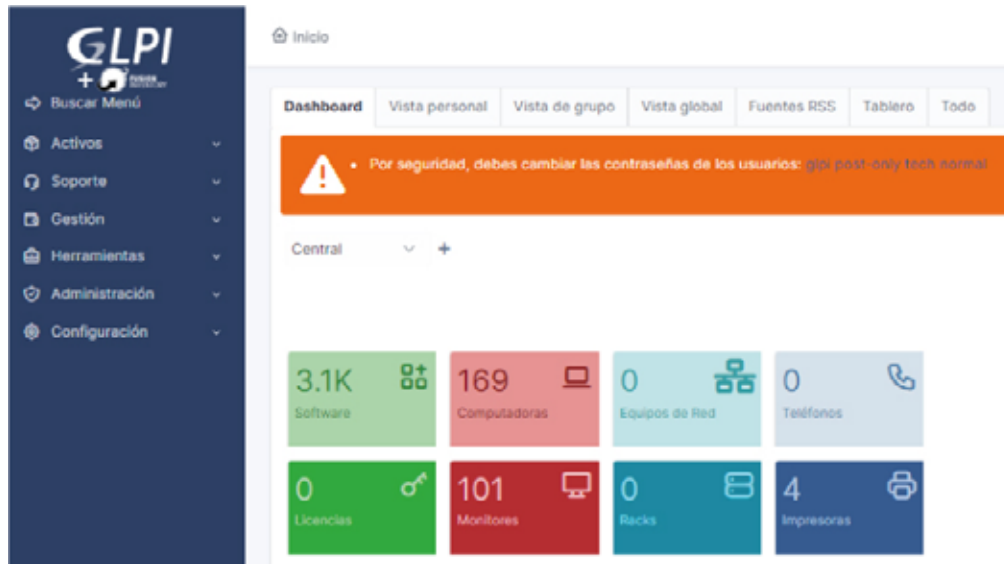
NOMBRE	ENTIDAD	ESTADO	FABRICANTE	NÚMERO DE INVENTARIO	NÚMERO DE SERIE	TIPO	MODELO	SISTEMA OPERATIVO	UBICACIÓN	ÚLTIMA MODIFICACIÓN	COMPONENTES - PROCESADOR	SOFTWARE ANTIVIRUS	SUBENTIDADES	COMPONENTES - MEMORIA
CM-AUUA-PC20	Entidad principal > DSSI > CIST > ISTSE > CM-AUUA1		LENOVO	2432527	MU1DE1T	Desktop	ThinkCentre M93z	Windows		2023-03-06 16:29	Intel(R) Core(TM) i5-4430S CPU @ 2.70GHz	Windows Defender	No	8 GB

4. RESULTADOS

El proceso para la implementación y configuración de la aplicación GLPI consistió en una serie de pruebas que se realizaron de manera iterativa bajo el marco de la metodología del ciclo de *Deming*. Como resultado se cuenta con un servidor configurado con las *Entidades* y *Usuarios* requeridos para la operación así como con la información del inventario del equipo que forma parte de la infraestructura de cómputo, como se muestra en la figura 6.

Figura 6

Panel informativo del inventario de la Infraestructura de cómputo en GLPI



La información obtenida es muy amplia, presenta detalles de las características de los componentes físicos, los programas instalados, la configuración de la red y de monitores e impresoras conectados directamente a las computadoras, como se refleja en la figura 7.

Figura 7

Detalle de la información obtenida por los agentes de captura de inventario de GLPI

Computadora		Agregar nuevo componente							Agregar		
Análisis de impacto		Componentes							Inventario automático		
Sistemas Operativos 1		Tipo de componente							Acciones		
Componentes 23		FIRMWARE	FABRICANTE	TIPO	VERSIÓN	FECHA DE LANZAMIENTO					
Almacenamiento 3	<input type="checkbox"/>	Dell Inc. BIOS +	Dell Inc.	BIOS	A09	2015-07-27	Actualizar	Si	<input type="checkbox"/>		
Software 341	<input type="checkbox"/>	PROCESADOR		FABRICANTE	VELOCIDAD (MHZ)	NÚCLEOS	PROCESADORES LÓGICOS				
Conexiones	<input type="checkbox"/>	Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz +		Intel	Actualizar	1700	2	4	Si	<input type="checkbox"/>	
Puertos de red 2	<input type="checkbox"/>	MEMORIA		TIPO	VELOCIDAD	TAMAÑO (MB)	NÚMERO DE SERIE	POSICIÓN DEL DISPOSITIVO			
Sockets	<input type="checkbox"/>	DDR3 - SODIMM +		DDR3	1600	Actualizar	4096	12191FE8	2	Si	<input type="checkbox"/>
Gestión remota	<input type="checkbox"/>	DISCO DURO		FABRICANTE	INTERFAZ	CAPACIDAD (MB)	NÚMERO DE SERIE				
Gestión	<input type="checkbox"/>	TOSHIBA MQ01ABF050 +		(Standard disk drives)	IDE	Actualizar	500107	84185DWPS		Si	<input type="checkbox"/>
Contratos											
Documentos											
Virtualización											
Software Antivirus 1											
Base de conocimiento											

La implementación de esta herramienta ha facilitado la captura del inventario de los activos de cómputo y se han eliminado errores de captura. Con la definición de entidades la información queda organizada por áreas funcionales y contenida en una base de datos visible a través de un sitio web al cual se accede a través de cuentas de acceso definidas por perfiles, lo que da seguridad en el manejo de la información de los activos.

A partir de la información del inventario es posible hacer uso de otras funcionalidades de *GLPI* para proyectos subsecuentes relacionados con el soporte técnico a usuarios y la gestión de proyectos, de igual manera, al documentar las soluciones de los incidentes se puede construir una base de conocimiento de apoyo para los usuarios.

Cabe mencionar que aun cuando *OCS Inventory* y *FusionInventory* pueden integrarse con *GLPI* a través de *Plugins*, la característica de la función de inventario nativo de *GLPI* (disponible desde la versión 10.0.0) y la integración de todos sus elementos en un solo servidor, son factores determinantes que influyeron en su selección como una herramienta que satisface en mayor medida las necesidades que se identificaron al inicio del proyecto.

5. CONCLUSIONES

La administración de la infraestructura de cómputo es una actividad que puede resultar complicada cuando el volumen de activos es grande, por lo que es necesario implementar herramientas de software como *GLPI*, que permite automatizar el levantamiento del inventario, conocer sus características, organizar la información y poder emplearla para dar soporte a otras actividades como son: el soporte técnico a usuarios y el seguimiento del ciclo de vida de los activos para la toma de decisiones en torno a nuevas adquisiciones, actualizaciones o corrimiento del equipo hacia áreas en donde sea funcional por el tipo y grado de complejidad de las tareas que se ejecutan.

Dentro de los hallazgos encontrados, y que se considera importante tomar en cuenta al realizar la implementación de *GLPI*, se encuentran los siguientes:

- Realizar un análisis cuidadoso acerca de la distribución de los activos y la forma en la que se requiere se presente la información de los mismos. Como se mencionó anteriormente, *GLPI* trabaja con base en *Entidades* y regularmente representan la organización funcional de la entidad que implementa *GLPI*.
- Definir la información que se requiere y que se necesita recolectar, esto evitará saturar la base de datos con información poco relevante.
- En la instalación del agente debe indicarse de manera correcta la dirección del gestor de la base de datos en donde se almacenará la información recolectada.

REFERENCIAS BIBLIOGRÁFICAS

AXELOS (2023). *ITIL4 Practitioner*. <https://www.axelos.com/certifications/itil-service-management/itil-practices-manager/itil-4-specialist-plan-implement-and-control/itil-4-practitioner-it-asset-management>

FusionInventory (2023). *Welcome to FusionInventory*. <https://fusioninventory.org/>

GLPI (2023). *Características de GLPI*. <https://glpi-project.org/es/caracteristicas/>

Invgate (2023). *ITIL 4 La guía definitiva*. <https://invgate.com/es/guides/itil/>

International Organization for Standardization (2023). ISO 55000:2014(es) *Gestión de activos - Aspectos generales, principios y terminología*. <https://www.iso.org/obp/ui#iso:std:iso:55000:ed-1:v2:es>

ITIL (2019). *ITIL 4, Foundation*. United Kingdom: AXELOS Limited.

OCS Inventory (2023). About OCS Inventory. https://ocsinventory-ng.org/?page_id=118&lang=en

SYDLE (2023). *Ciclo PDCA: ¿cuáles son los pasos y cómo funciona? Conoce algunos ejemplos | Blog SYDLE*. <https://www.sydle.com/blog/assets/post/ciclo-pdca-61ba2a15876cf6271d556be9/ciclo-pdca.png?w=720>.

ANEXO A. INSTALACIÓN DEL AGENTE PARA LEVANTAMIENTO DE INVENTARIO

El agente para el levantamiento del inventario de la infraestructura de cómputo se descarga del sitio <https://github.com/glpi-project/glpi-agent/releases/tag/1.5> y debe instalarse en cada computadora de la siguiente manera:

1. Ejecutar el instalador del agente, se presenta la pantalla de inicio, dar clic en *Next*. Figura 8.

Figura 8

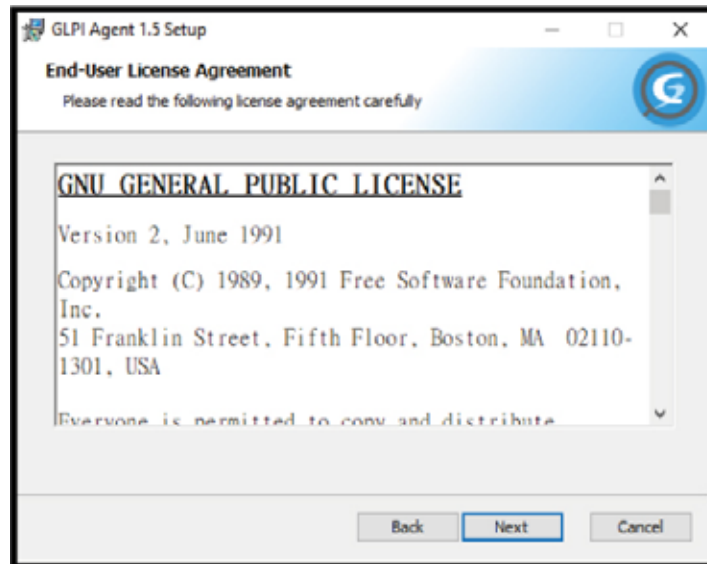
Inicio de instalación del Agente GLPI



2. Se muestra el acuerdo de licencia GNU v2, dar clic en *Next*.

Figura 9

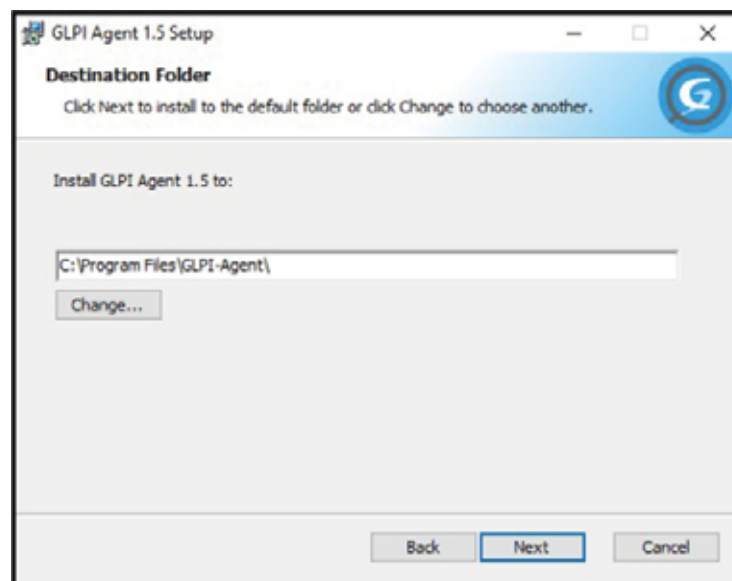
Acuerdo de licencia



3. Se indica la ruta de instalación del agente, dar clic en *Next*.

Figura 10

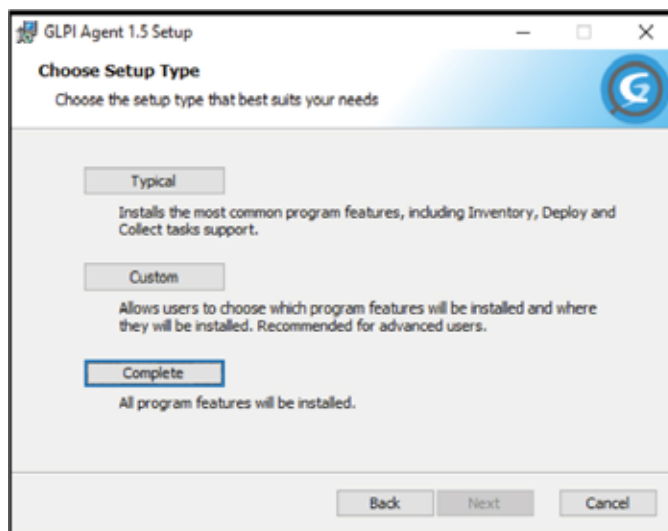
Ruta de instalación



4. Se presentan las *Opciones de instalación*. Seleccionar *Complete* y dar clic en *Next*.

Figura 11

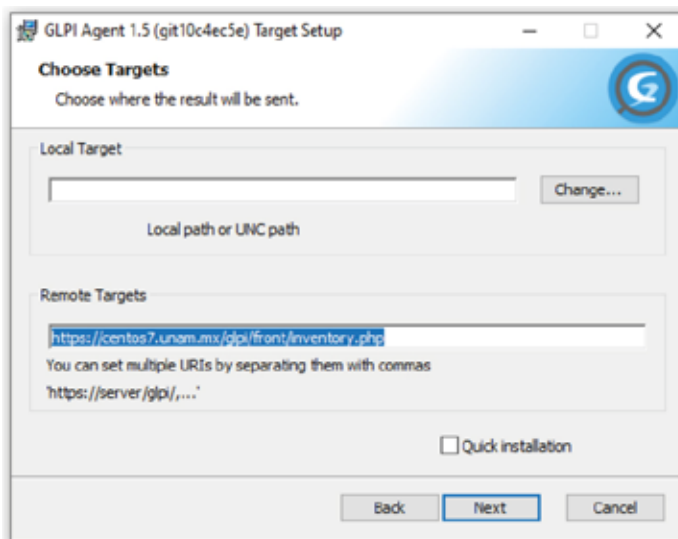
Opciones de instalación del Agente



5. Se presenta la pantalla para indicar la ruta del servidor en donde se encuentra la base de datos que alojará la información capturada en el levantamiento del inventario. Es muy importante proporcionar la ruta correcta, dar clic en *Next*.

Figura 12

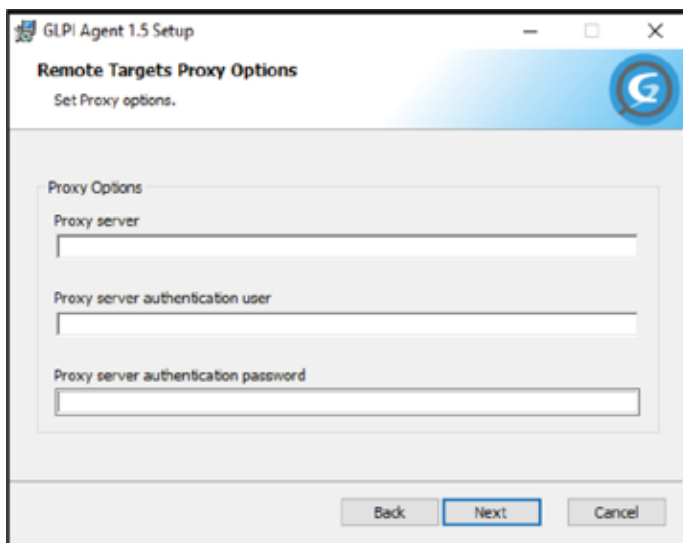
Ruta de acceso al servidor



- Indicar si se requiere alguna configuración especial de SSL en el servidor, dar clic en Next.

Figura 13

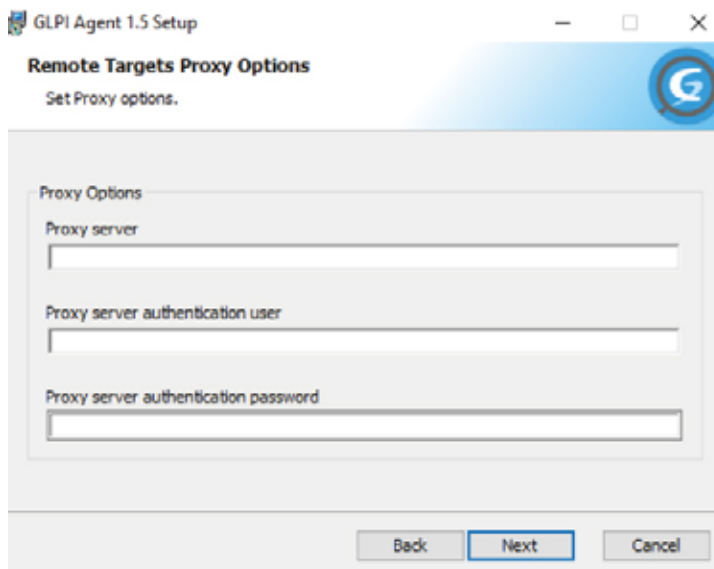
Configuración de SSL



- Indicar si se requiere configuración *Proxy* en el servidor, dar clic en *Next*.

Figura 14

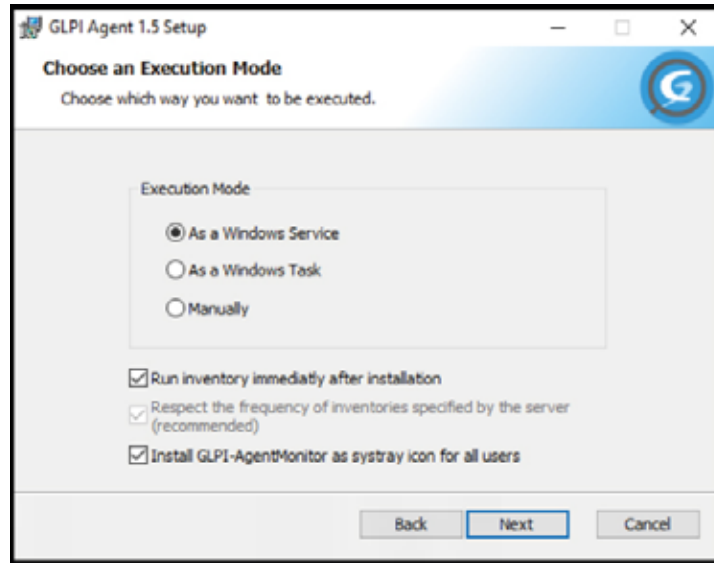
Configuración Proxy



- Elegir en *Modo de ejecución*, seleccionar *As a Windows Service, Run inventory immediatly after installation e Install GLPI-AgentMonitor as systray icon for all users*, dar clic en *Next*.

Figura 15

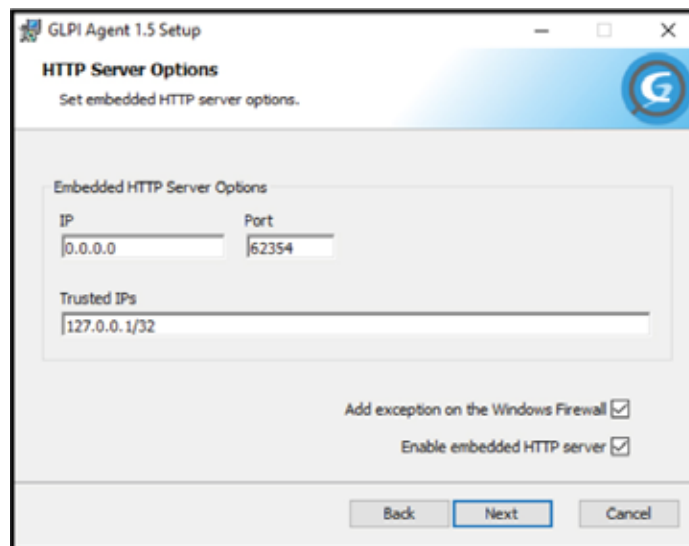
Modo de ejecución del Agente



- Indicar si se requiere alguna configuración adicional para el servicio HTTP en el servidor, dar clic en *Next*.

Figura 16

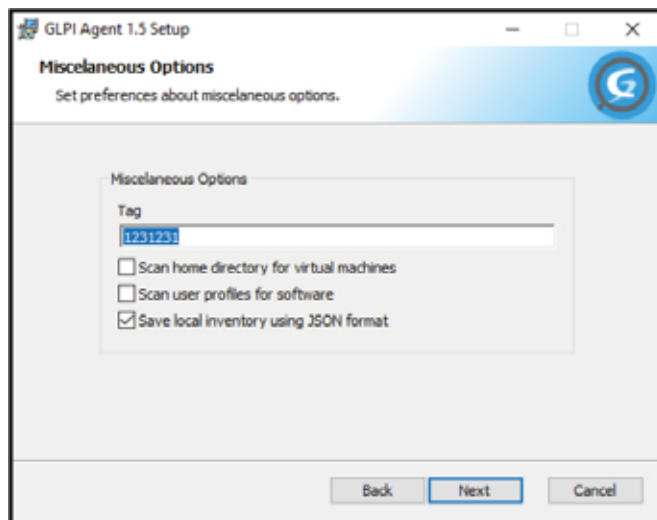
Configuración de acceso al servidor HTTP



10. Proporcionar información adicional para registrar en la base de datos, este es un campo especial llamado *Tag* y regularmente se usa para indicar el número de inventario del equipo, dar clic en *Next*.

Figura 17

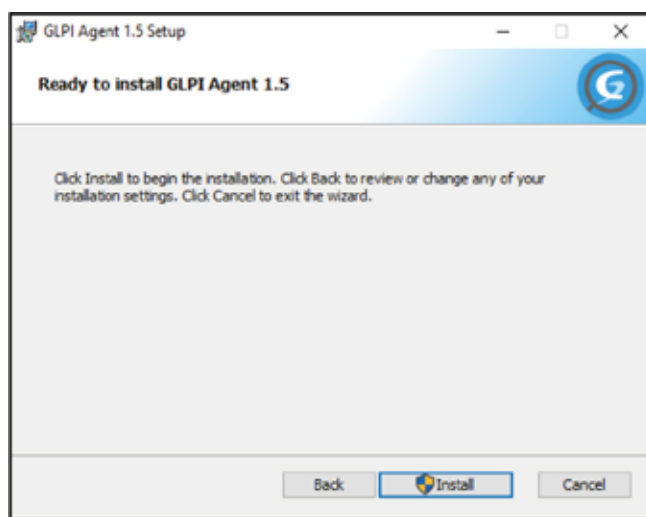
Información del Tag



11. La configuración del agente ha finalizado, dar clic en *Install*.

Figura 18

Inicio de la instalación del Agente



Al finalizar la instalación, se llevará a cabo el levantamiento de inventario y la información recolectada se enviará al servidor en donde podrá visualizarse.

Implementación de la técnica de Mapeo Objeto-Relacional en el desarrollo de sistemas

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:

Ortega Ramírez, C. R. (2023). Implementación de la técnica de Mapeo Objeto-Relacional en el desarrollo de sistemas. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (52 - 61).

<https://doi.org/10.22201/dgtic.ctud.2023.1.1.16>

Cristian Ricardo Ortega Ramírez

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

cristian.ortega@comunidad.unam.mx

ORCID: 0009-0003-1073-6492

Resumen:

Se abordan los principales conceptos implicados en la técnica *ORM (Object Relational Mapping)*. Se describe el proceso realizado para su implementación en el desarrollo de la segunda versión del Sistema Integral de Personal Académico con el *framework Symfony* y la librería *Doctrine*, mediante su interfaz de línea de comandos. La incorporación de esta técnica brindó mayor flexibilidad a la hora de manipular los datos, y permitió la reducción del tiempo empleado para la codificación del sistema en aproximadamente un 37% con respecto a la primera versión.

Palabras clave:

Mapeo Objeto-Relacional, programación orientada a objetos, bases de datos, *Symfony*, *Doctrine*.

1. INTRODUCCIÓN

Un sistema de información es comúnmente descrito como un conjunto de componentes que interactúan entre sí para recoger, procesar, almacenar y proveer la información necesaria para apoyar la toma de decisiones y el control de una organización. Entre los componentes que conforman un sistema de información destacan la tecnología de procesos, que viene determinada principalmente por los lenguajes y técnicas de programación utilizadas, así como la tecnología empleada para la gestión de los datos (Laudon y Laudon, 2006; Whitten, Bentley y Dittman, 2004, citados por Fernández Alarcón, 2006, p. 13).

Gómez-Díaz et al. (2022) sostienen que uno de los principales desafíos de los sistemas de información, refiere a la complejidad de alinear el código de programación con las estructuras de la base de datos, ya que un desarrollador además de dominar el lenguaje de programación empleado, deberá de comprender y aplicar mediante el lenguaje de consulta estructurado (*SQL*), las sentencias y el código necesarios para la conexión e interacción de su aplicación con una base de datos.

Es así que cobra especial relevancia la implementación de una técnica que permita mapear las estructuras de base de datos sobre una estructura lógica de objetos, con el fin de simplificar, y acelerar el desarrollo del sistema, permitir a los desarrolladores “liberarse” de la escritura o generación manual de código *SQL* y realizar las operaciones *CRUD* (creación, recuperación, actualización y eliminación, por sus siglas en inglés) sobre los datos de forma indirecta (Muro, 2023).

La primera versión del Sistema Integral de Personal Académico fue desarrollada por un equipo de trabajo conformado por un solo programador y un diseñador. Posteriormente, nuevos requerimientos que impactaron de forma importante en la estructura de datos dio como resultado una nueva versión del SIPA, incorporando al equipo de trabajo a un especialista en bases de datos con la finalidad de implementar un mecanismo que brindara mayor flexibilidad a los procesos de manipulación e interacción con la estructura de la base y los datos en sí mismos para acelerar el desarrollo del sistema. Adicionalmente, se integró al equipo a un especialista en *DevOps*¹ para agilizar el despliegue del sistema en entornos de desarrollo, pruebas y producción.

A continuación, se describe el proceso de implementación de la técnica *ORM* para el desarrollo de la segunda versión del SIPA, misma que contribuyó a proveer a los desarrolladores con una capa de abstracción que encapsula la lógica de los datos, lo que facilita su gestión a través de objetos, así como la reutilización de código, para acelerar el desarrollo y puesta en marcha del sistema.

2. OBJETIVO

Reportar los resultados y beneficios obtenidos al implementar la técnica de Mapeo Objeto-Relacional en el desarrollo de la segunda versión del SIPA.

¹ El ingeniero de *DevOps* incorpora procesos, herramientas y metodologías para equilibrar las necesidades durante todo el ciclo de vida del desarrollo de *software*, desde la programación y la implementación hasta el mantenimiento y las actualizaciones

3. DESARROLLO

3.1 SELECCIÓN DE TECNOLOGÍA

3.1.1 LENGUAJE DE PROGRAMACIÓN

Para la codificación del sistema se optó por utilizar *PHP*, lenguaje empleado en la versión previa del SIPA y con el que el programador principal tenía mayor experiencia, además de ser el lenguaje de programación del lado del servidor más utilizado en sistemas web (Web Technology Surveys [W3Techs], s. f.). El lenguaje de programación *PHP* está basado en el paradigma de programación orientada a objetos (*POO*), con un gran impacto en el desarrollo de sistemas en tanto que facilita la simplificación de la escritura, mantenimiento y reutilización de código. Asimismo, *PHP* permite representar un programa como una colección de objetos que colaboran entre sí para realizar determinadas tareas. Se entiende por objeto a una entidad que se encuentra en situaciones o problemas de nuestro mundo real, y que está conformado tanto por atributos que representan sus datos o características, como por métodos que responden a las acciones u operaciones que puede realizar. La representación abstracta de una colección de objetos que comparten una misma estructura y un mismo comportamiento se denomina clase, y es a partir de ella que el objeto se materializa (Pérez Montero y Hernández Pérez, 2019).

En la figura 1 se muestra una representación visual de una clase y un objeto derivado de dicha clase, ambos con sus correspondientes atributos y métodos.

Figura 1

Representación visual de una clase y un objeto



3.1.2 FRAMEWORK PHP

Se optó por el uso del *framework*² *Symfony*, debido a su capacidad para realizar actualizaciones de forma segura en el esquema de la base de datos a través de migraciones, es decir, de bloques de código generados de manera automática por el framework al comparar la definición de las clases con la estructura actual real de la base de datos (Altube Vera, 2021).

3.1.3 SISTEMA MANEJADOR DE BASES DE DATOS (SMBD)

Como los sistemas de información requieren almacenar datos relacionados entre sí de una manera estructurada es imprescindible el uso de un Sistema Manejador de Bases de Datos. De acuerdo con el DB-Engines (2023), el grupo de las bases de datos relacionales es el más consolidado del sector de las TIC y el de mayor uso hoy en día. Una base de datos relacional estructura información en tablas conformadas por campos y filas, además de relacionar dicha información mediante la unión de tablas, lo que facilita la comprensión sobre la relación entre varios datos (Google, s. f.).

En la figura 2 se muestra la representación visual de una tabla con sus correspondientes campos y filas.

Figura 2

Representación visual de una tabla en una base de datos

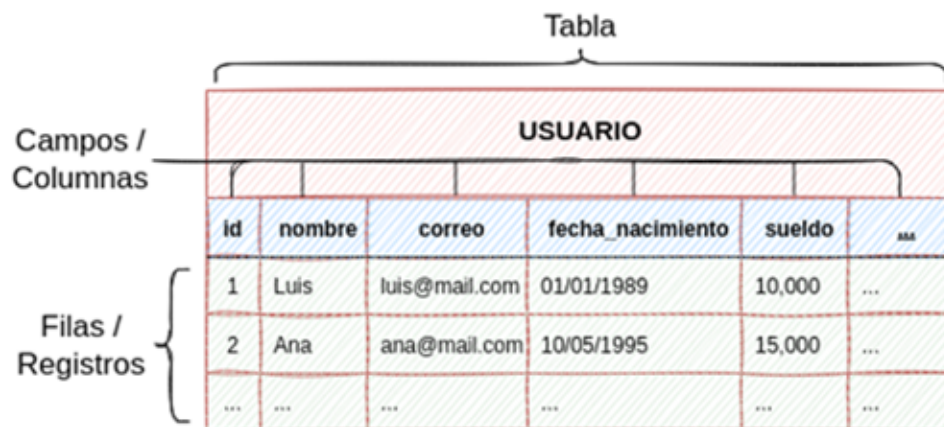


Tabla					
USUARIO					
id	nombre	correo	fecha_nacimiento	sueldo	...
1	Luis	luis@mail.com	01/01/1989	10,000	...
2	Ana	ana@mail.com	10/05/1995	15,000	...
...

La herramienta a utilizar para la gestión de los datos fue el *SMBD* el cual se empleó en la primera versión del sistema. Dado que ya se contaba con información almacenada del SIPA en su primera versión, utilizar el mismo *SMBD* permitió acelerar el tiempo de migración de los datos existentes, y garantizar su integridad. Dicha migración de datos implicó ejecutar un proceso de *ETL* (Extracción, Transformación y Carga, por sus siglas en inglés) que se llevó a cabo paralelamente a la codificación del sistema.

² Un *framework* es un marco de trabajo que proporciona una estructura base para desarrollar un proyecto con objetivos específicos, en otras palabras, es un tipo de plantilla que sirve como punto de partida para organizar y crear *software*.

3.1.4 LIBRERÍA ORM

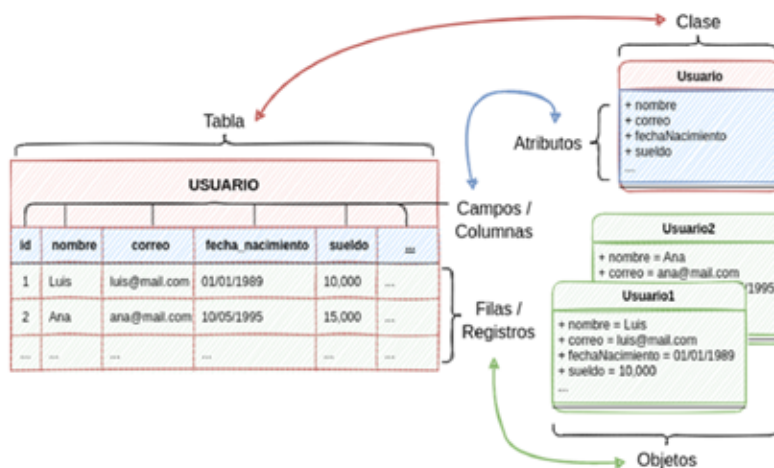
Para la gestión y acceso a los datos se implementó la técnica *ORM*, que brinda mejores niveles de abstracción respecto a otras técnicas de acceso a bases de datos. Existen dos patrones de diseño utilizados en el Mapeo Objeto-Relacional (*Active Record* y *Data Mapper*), y si bien ambos patrones tienen el mismo propósito, difieren en la forma en que abordan la asignación y la responsabilidad de las operaciones de base de datos (Kolodka, 2023).

Finalmente, para la implementación de la técnica *ORM* se decidió utilizar la librería *Doctrine*, la cual utiliza el patrón *Data Mapper* y es la utilizada por defecto para un proyecto creado con *Symfony*, lo cual permite aprovechar el amplio abanico de herramientas que ofrece para la implementación de dicha librería. Asimismo, se consideró la experiencia de los nuevos integrantes del equipo de trabajo en el uso de otras librerías que utilizan el patrón *Data Mapper*.

Doctrine permite realizar un Mapeo Objeto-Relacional al trasladar los datos de una base de datos relacional a un sistema de clases y de objetos, donde las tablas, campos y filas serían el equivalente a clases, atributos y objetos, respectivamente. En la figura 3 se muestra una representación visual del mapeo realizado por un *ORM*.

Figura 3

Representación visual de un Mapeo Objeto-Relacional



3.2 IMPLEMENTACIÓN DE LA TÉCNICA ORM

Los cambios en la funcionalidad requerida para la segunda versión del SIPA implicaron realizar una reestructura de la base de datos original, mediante la implementación de *Doctrine* desde un enfoque *Model First* para generar un Modelo Entidad-Relación que cubriera dichos requerimientos, lo que permitió trabajar de forma paralela la migración de los datos existentes y la codificación del sistema.

Cabe señalar que el enfoque *Model First* consiste en realizar, en primera instancia, el modelado de la aplicación mediante alguna herramienta como *UML*, y posteriormente generar la estructura de la base de datos y el código a partir de dicho modelo.

Para iniciar con la implementación del *ORM*, se debe crear un nuevo proyecto *Symfony* que, a su vez, precisa de la instalación previa de la paquetería³ que requiere el *framework* en el entorno de desarrollo. A continuación, se integran las instrucciones que generarán la estructura de un nuevo proyecto *Symfony*, a través de la *CLI (Command-Line Interface)*:

```
$ composer create-project symfony/skeleton[:<symfony_version>] <app_name>
$ cd <app_name>
$ composer require webapp
```

Una vez creada la estructura del proyecto *Symfony*, se procede a la instalación de la librería *Doctrine* y del paquete "*symfony/maker-bundle*", para dar inicio a la generación asistida de código para la implementación del *ORM*:

```
$ composer require symfony/orm-pack
$ composer require --dev symfony/maker-bundle
```

Finalmente, se configura la conexión con el *SMBD* lo que implica editar el archivo ".env" para agregar a la siguiente línea sustituyendo cada campo con los valores correspondientes:

```
DATABASE_URL="<smbd_name>://<db_user>:<db_password>@<db_host>:<db_port>/<db_name>?serverVersion=<smbd_version>"
```

En este punto, se puede proceder con la implementación del *ORM* partiendo del modelo diseñado para el sistema, como se muestra en la figura 4.

Figura 4

Modelo Entidad-Relación



³ Los requerimientos técnicos para comenzar a trabajar con *Symfony* son: *PHP* versión 8.1 o superior; las librerías *Ctype*, *iconv*, *PCRE*, *Session*, *SimpleXML* y *Tokenier*; *Composer* para el manejo de dependencias para *PHP*.

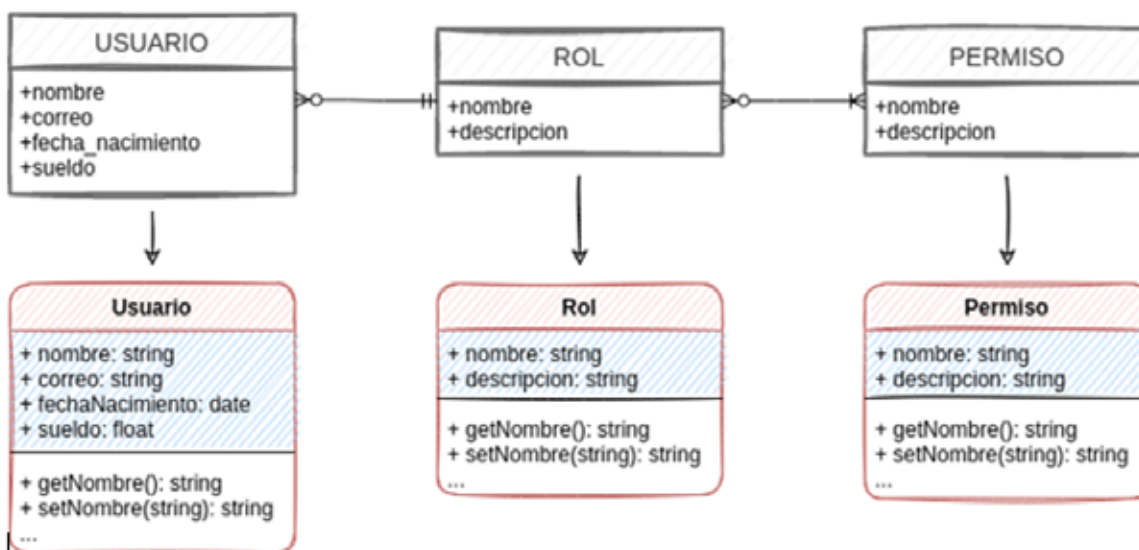
Para comenzar a generar el código *PHP* de las clases con sus correspondientes atributos, se ejecuta un comando proporcionado por el paquete "symfony/maker-bundle":

```
$ php bin/console make:entity
```

Una vez ejecutado el comando, un asistente será el encargado de guiar la generación del código de una *Clase PHP* que servirá para mapear dicha entidad, lo que requiere especificar: nombre de la clase, nombre de los atributos, tipo de dato a utilizar para cada atributo, longitud (para los tipos de datos que apliquen) e indicar si el valor del atributo puede ser nulo en la base de datos. En la figura 5, se muestra una representación de las clases generadas a partir de las entidades del modelo.

Figura 5

Representación visual de las clases generadas a partir de las entidades del modelo



Una vez creadas las clases para cada entidad es necesario crear sus relaciones considerando los requerimientos del sistema y preponderando si se deben mapear uno o los dos sentidos que se integran en una relación.

Por ejemplo, en el modelo mostrado, existe la entidad *USUARIO* que se relaciona con la entidad *ROL*, vista la relación desde la primera entidad hacia la segunda, significa que un *USUARIO* tiene un *ROL* asignado, pero si se observa la relación en el sentido inverso, implica que un *ROL* puede estar asignado a uno o más *USUARIOS*. Entonces, si los requerimientos del sistema indican que es preciso conocer los usuarios y sus roles asignados, convendrá mapear la relación de *USUARIO* a *ROL*, en el caso contrario, si se deben conocer los usuarios a los que se ha asignado cada rol, sería apropiado mapear la relación desde el *ROL* hacia *USUARIO*; otro escenario posible es aquel en que las dos situaciones sean requerimientos del sistema, en cuyo caso mapear ambos sentidos de la relación será lo indicado.

Para generar las relaciones, se ejecuta nuevamente el comando previo y cuando el asistente pregunta el nombre de la clase que se desea crear, se ingresa el nombre de la clase existente para modificarla; posteriormente se ingresa el nombre del atributo que servirá para mapear la relación, y se especifica como tipo de dato *relation*. En este punto el asistente preguntará por el nombre de la entidad con la que se desea relacionar la entidad actual, tras lo cual se indica el tipo de relación (muchos a uno, uno a muchos, muchos a muchos o uno a uno); opcionalmente el asistente permitirá crear la relación inversa.

Finalmente, para la construcción de la estructura de la base de datos, se genera una migración y se aplican dichos cambios en la base de datos a través de los siguientes comandos:

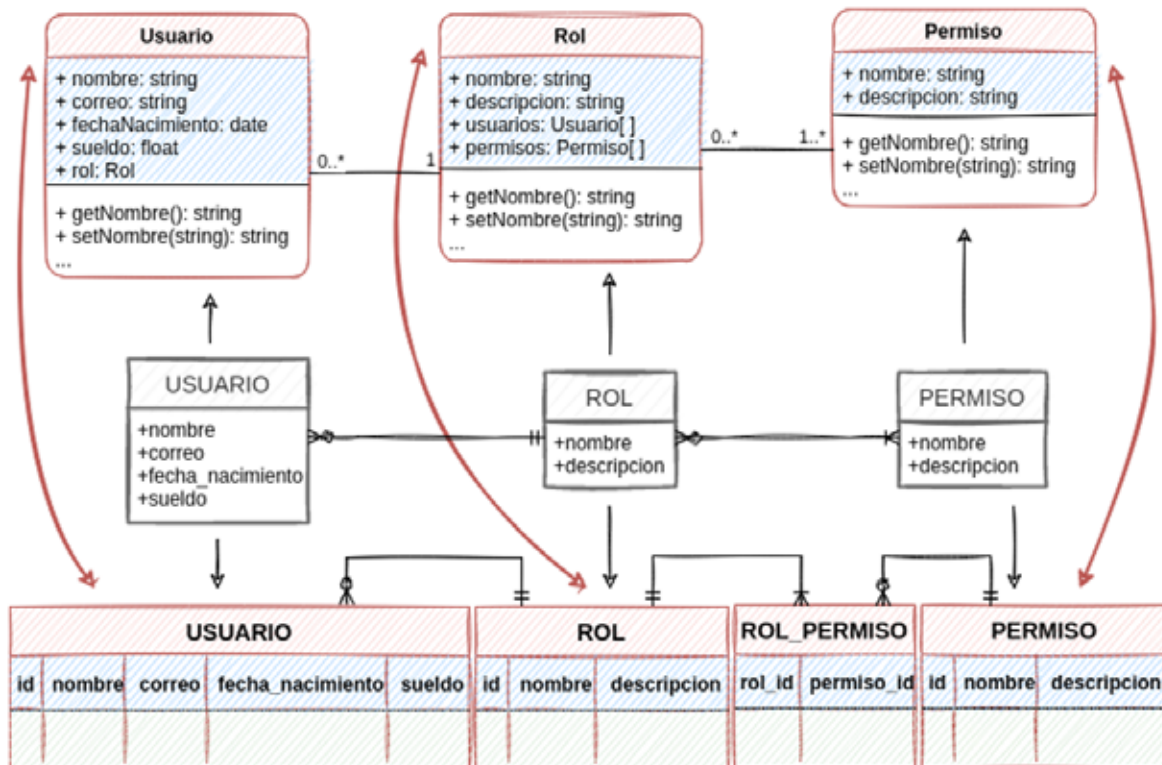
```

$ php bin/console make:migration
$ php bin/console doctrine:migration:migrate
  
```

Lo anterior da como resultado el conjunto de elementos necesarios (clases y tablas) para comenzar a desarrollar el sistema con la implementación de la técnica *ORM* como se muestra en la figura 6.

Figura 6

Mapeo Objeto-Relacional a partir del enfoque model first



4. RESULTADOS

Se logró implementar de forma exitosa la técnica *ORM* en el desarrollo del SIPA con los criterios de selección tecnológica anteriormente descritos (*PHP + Symfony + MariaDB + Doctrine*), de manera que fue posible separar las tareas de desarrollo entre los integrantes del equipo y trabajar de forma paralela en diferentes aspectos del sistema (generación de entornos, programación y migración de datos).

Fue posible validar los beneficios ofrecidos por la técnica *ORM*, en cuanto a agilizar el desarrollo del sistema mediante una programación centrada en la lógica de objetos, lo que permitió realizar el Mapeo Objeto-Relacional de 18 tablas y 25 relaciones, en un tiempo estimado de cuatro horas y reducir en, aproximadamente, un 37% el tiempo efectivo empleado en la codificación de la segunda versión del SIPA (cinco meses) con respecto al tiempo que tomó la codificación de la primera versión (ocho meses), la cual no implementó la técnica *ORM*.

Finalmente, la mayoría de la literatura disponible sobre la implementación de la técnica *ORM* sugiere utilizar una librería bajo el enfoque *Active Record* cuando se desea priorizar el tiempo de desarrollo por considerar que el enfoque *Data Mapper* resulta más intimidante debido a su elevada curva de aprendizaje y configuración más compleja; sin embargo, los beneficios obtenidos en el desarrollo del SIPA dada la implementación exitosa de la técnica *ORM* con una librería que sigue el enfoque *Data Mapper*, enfatizan la importancia de considerar las capacidades del equipo de desarrollo por encima de las características de la tecnología a implementar.

4.1 TRABAJOS FUTUROS

Si bien al seleccionar la tecnología empleada para el almacenamiento de los datos, se optó por utilizar *MariaDB* priorizando la optimización de tiempo para la migración de los datos existentes, otra de las principales ventajas que ofrece la técnica *ORM* es la de proporcionar independencia del *SMBD*, por lo que, en una futura actualización, se podría cambiar de forma transparente en caso de ser necesario. La alternativa más conveniente de acuerdo con un análisis realizado por Choina y Skublewska-Paszkowska (2022) sería utilizar *PostgreSQL* al ser el manejador de bases de datos que presentó el mejor rendimiento en conjunto con *Doctrine*.

5. CONCLUSIONES

Implementar una técnica de Mapeo Objeto-Relacional en el desarrollo de un sistema brinda mayor flexibilidad a la hora de manipular los datos al proveer mecanismos para la generación de código y sentencias *SQL* de forma automatizada, lo que permite aumentar la productividad de los programadores y evita la escritura repetitiva de código para ejecutar operaciones *CRUD*.

Si bien, la implementación de la técnica *ORM* en el SIPA resultó exitosa y por ello se sugiere emplear en el desarrollo de otros sistemas que requieran gestionar información en una base de datos, siempre se debe considerar la magnitud del proyecto, los plazos para su entrega, las capacidades del equipo de desarrollo y las tecnologías a emplear, para determinar su viabilidad.

REFERENCIAS BIBLIOGRÁFICAS

- Altube Vera, R. (2021, mayo 21). *Laravel vs Symfony: Qué framework PHP elegir*. *openwebinars.net*. <https://openwebinars.net/blog/laravel-vs-symfony-que-framework-php-elegir/>
- Choina, M., & Skublewska-Paszowska, M. (2022). *Performance analysis of relational databases MySQL, PostgreSQL and Oracle using Doctrine libraries*. *Journal of Computer Sciences Institute*, 24, 250-257. <https://doi.org/10.35784/jcsi.3000>
- DB-Engines. (2023). *DB-Engines ranking*. DB-Engines. Recuperado el 18 de septiembre de 2023, de <https://db-engines.com/en/ranking>
- Fernández Alarcón, V. (2006). *Desarrollo de sistemas de información: Una metodología basada en el modelado* (1a ed.). UPC, S.L., Edicions. <https://doi.org/10.5821/ebook-9788498800708>
- Gómez-Díaz, M. S., Casillas-Rodríguez, F. J., Juárez Guerra, L., Castellanos Nolasco, E., y Uribe López, U. (2022). *Análisis e implicaciones de la implementación del mapeo relacional de objetos en la programación orientada a objetos*. *Innovación y Desarrollo Tecnológico Revista Digital*, 14(4), 983-988.
- Google. (s.f.). *¿Qué es una base de datos relacional?* Google Cloud. <https://cloud.google.com/learn/what-is-a-relational-database?hl=es-419>
- Kolodka, P. (2023, julio 5). *Understanding abstraction levels in database interactions: DAL, DAO, Raw Queries, Query Builder, ORM and Repository*. Medium. <https://levelup.gitconnected.com/understanding-abstraction-levels-in-database-interactions-dal-dao-raw-queries-query-builder-4819d607b0d6>
- Laudon, K. C., y Laudon, J. P. (2012). *Sistemas de información gerencial* (12a ed.). Pearson Educación.
- Muro, J. A. (2023). *¿Qué es un ORM?* Deloitte Spain. <https://www2.deloitte.com/es/es/pages/technology/articles/que-es-orm.html>
- Pérez Montero, E. L., & Hernández Pérez, F. de M. (2019). *Object oriented programming: Easy to create*. *I+ T+ C- Research, Technology and Science*, 1(13), 96-100.
- Web Technology Surveys. (s.f.). *Usage statistics and market share of server-side programming languages for websites*. Recuperado el 25 de octubre de 2023, de https://w3techs.com/technologies/overview/programming_language

Diseño de una cola de hilos

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:

Talavera Rosales, A. (2023). Diseño de una cola de hilos. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (62 - 71).

<https://doi.org/10.22201/dgtic.ctud.2023.1.1.13>

Alejandro Talavera Rosales

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

badboy@unam.mx

ORCID: 0009-0005-5489-2854

Resumen:

Desde hace décadas los sistemas operativos permiten el desarrollo de programas que aprovechen las capacidades de ejecución de varias tareas a la vez, lo que se conoce como tareas concurrentes. Si se incorpora correctamente esta cualidad a una aplicación, se puede lograr una reducción en el tiempo de respuesta para distintas tareas que se van realizando. Sin embargo, existen circunstancias donde no es posible realizar estas tareas, como es el caso de equipos con recursos computacionales limitados como los celulares de gama baja, en los que el uso de la programación de tareas concurrentes puede comprometer la ejecución de una aplicación, llegando incluso a la aparición de las denominadas *Screen of Death*¹, con la subsecuente falla del equipo. En estos casos se debe limitar el uso de programación concurrente para no exceder las capacidades computacionales, lo que puede llevarse a cabo mediante el uso de un componente diseñado para limitar el número de tareas que se ejecutan a la vez.

Palabra clave:

Threads, Java, Estructuras de datos, hilos de ejecución.

¹ *Screen of Death* o Pantalla de la muerte, es el término técnico que se da a una falla catastrófica que interrumpe la ejecución de un sistema y presenta un mensaje indicando esta condición. Por lo regular, la única ruta de acción es reiniciar el dispositivo. Unas de las más conocidas son las pantallas azules del sistema operativo *Windows*.

1. HILOS DE EJECUCIÓN

La programación concurrente emplea el concepto de hilos de ejecución, mismo que proviene del término en inglés *Thread*. Un hilo de ejecución es el mecanismo que permite la ejecución de una tarea en un equipo de cómputo. Dicha ejecución se realiza en distintas fases, incluyendo su preparación, la ejecución y manejo de recursos de cómputo, hasta concluir su ejecución. Todo programa requiere al menos un hilo de ejecución.

Si un programa consta de varias tareas, éstas se ejecutan una después de otra dentro del mismo hilo de ejecución. Dicho hilo² sigue la ejecución secuenciada de cada tarea hasta que termina. Cuando las tareas pueden ejecutarse de manera independiente, sin depender una de otra, es posible asignar uno o más hilos de ejecución paralelos.

La ejecución concurrente de tareas establece nuevos elementos que deben ser considerados por parte de los desarrolladores a fin de lograr su correcta implementación.

2. DESARROLLO CON HILOS

En el desarrollo de aplicaciones uno de los elementos más relevantes a conseguir es el uso óptimo de los recursos computacionales disponibles. Esto se logra al identificar claramente los procesamientos de información que se llevarán a cabo usando la aplicación, así como tener en cuenta los tiempos de respuesta de las distintas opciones que se implementan. Con esto en mente resulta lógico incluir el uso de las capacidades de concurrencia para su aprovechamiento. Estas capacidades permiten que más de una tarea pueda ser ejecutada a la vez en lugar de la estrategia básica de disponer de una lista de actividades que van ejecutándose una después de otra. Como ejemplo de este escenario se percibe cuando un navegador web carga una página que contiene varias imágenes. Los navegadores actuales van descargando distintas imágenes a la vez y no así una por una.

En principio, integrar el manejo de la concurrencia a una aplicación resulta muy simple mediante el empleo de bibliotecas destinadas para este propósito como lo son el paquete `java.lang.Thread` del lenguaje *Java*, o la biblioteca `thread` de *C++*. La problemática en la que los desarrolladores deberán centrarse será la del uso de los recursos computacionales que la aplicación requiere para su operación.

Ahora bien, el uso libre y sin planeación de este tipo de programación puede afectar negativamente a una aplicación en lugar de beneficiarla. En el caso de equipos muy limitados de memoria y/o poder de procesamiento, ejecutar varias tareas a la vez puede resultar en problemas de rendimiento notorios tales como la lentitud en la respuesta de un procesamiento o una falla catastrófica que bloquee al propio equipo, lo que además pueda derivar en la pérdida de información, entre otros efectos adversos.

Continuando con el ejemplo de los navegadores, en equipos muy limitados lanzar varias peticiones a la vez para la descarga de distintas imágenes puede resultar en una demanda de recursos que el propio equipo no será capaz atender apropiadamente y, en algunos casos, el sistema operativo del dispositivo al verse excedido por la carga de trabajo que tiene que atender, detendrá su operación de forma abrupta y en varios se presentará una pantalla indicando esta falla también conocida como *Pantalla de la muerte*.

² Se usará el término *Hilo* como sinónimo de *Hilo de Ejecución* para simplificar las explicaciones a lo largo de las siguientes secciones.

3. COLA DE HILOS

Para controlar el número de hilos que se ejecutan de forma concurrente se diseñó un componente de *software*, al cual se ha denominado *Cola de hilos*. El componente incorpora para su operación dos estructuras de datos clásicas: la *Lista* y la *Cola*.

Una lista representa una colección de elementos ordenados, entendiendo en este caso el orden como el lugar o posición que ocupan dichos elementos en la estructura. El objetivo para la *Cola de hilos* será el de contener las tareas que se están ejecutando al mismo tiempo en un momento determinado, en otras palabras, los elementos de esta lista serán hilos.

Con la estructura de datos cola, se aprovecharán sus cualidades de operación donde el primer elemento que se agrega será el primero en salir. El propósito de esta estructura será la de alojar las distintas tareas que se desean ejecutar, previo a su asignación a un hilo.

Con las tareas de cada estructura claramente identificadas, la operación de la *Cola de hilos* será la siguiente:

1. Se indica el número de hilos que se pueden ejecutar a la vez.
2. Se cargarán todas las tareas a ser ejecutadas.
3. Cuando se indique la ejecución de la *Cola de hilos*:
 - a. Se obtienen las primeras tareas de la cola, el mismo número que el indicado en el punto 1.
 - b. Por cada tarea se crea un hilo para su ejecución y se agregan a la lista.
 - c. Se llama a ejecución a cada uno de los hilos en la lista.
4. Cuando un hilo en la lista termina su ejecución:
 - a. Se libera de la lista el lugar del hilo que terminó su ejecución.
 - b. Si la cola de tareas no está vacía:
 - i. Sacar el siguiente elemento de la cola de tareas.
 - ii. Crear un nuevo hilo y asociarlo a la tarea que se obtuvo.
 - iii. Ocupar el espacio libre en la lista con el nuevo hilo.
 - iv. Ejecutar este hilo.
 - c. Si la cola de tareas está vacía, termina la ejecución de la *Cola de hilos*.

La operación de la *Cola de hilos* que se ha definido limita el número de tareas que pueden ejecutarse a la vez, permitiendo al mismo tiempo mantener a las demás tareas en espera a su ejecución. Una vez que alguno de los hilos en la lista termine su ejecución, se verificará si no hay más tareas pendientes de ejecutar. En caso de haber una tarea, se asocia a un nuevo hilo para su ejecución. Cuando la cola de tareas no tenga elementos se termina la ejecución de la cola de hilos.

4. DESARROLLO

Para la implementación de la *Cola de hilos* se eligió el lenguaje de programación *Java* a fin de aprovechar varias de sus cualidades, tales como la independencia de la plataforma, la implementación de las estructuras de datos clásicas y su biblioteca para la programación de hilos.

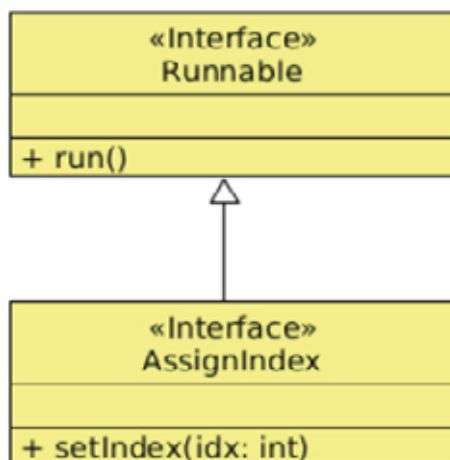
Estos elementos se usaron para el modelado de los distintos componentes de la *Cola de hilos* entre los que se pueden identificar las siguientes: las tareas que se ejecutarán, la terminación de una tarea que se ha ejecutado y la propia cola.

4.1 TAREAS A EJECUTAR

Se configuró la interfaz *AssignIndex* a fin de representar las tareas que se ejecutarán en la *Cola de hilos*. Esta interfaz hereda su funcionalidad de la interfaz *Runnable* por lo que en términos muy simples cualquier tarea (definida en un clase de *Java*) solamente deberá incluir obligatoriamente los métodos *run()* y *setIndex()*; el primero, como el método que ejecutará el hilo al que finalmente se asocie, mientras que el segundo es para asignar la posición en la lista de hilos que están ejecutándose. La figura 1 muestra el diseño de la interfaz.

Figura 1

Diseño de la interfaz AssignIndex



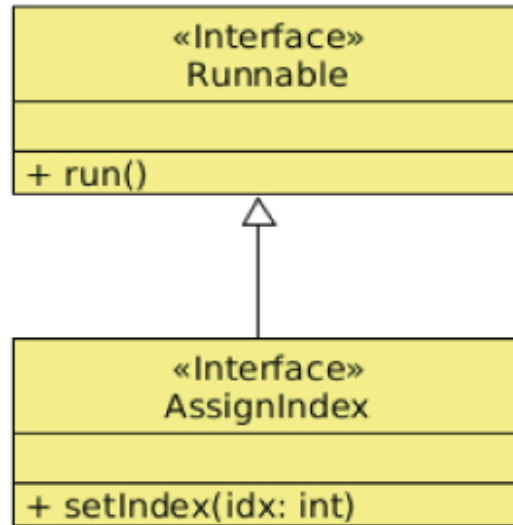
Nota. La interfaz *Runnable* es parte de los paquetes estándar de *Java* (*Oracle Java Documentation 2022* y *Oracle Java Documentation, 2023a*). Para más información de su definición se recomienda al lector consultar la documentación oficial de *Oracle*: <https://docs.oracle.com/javase/8/docs/api/java/lang/Runnable.html>

4.2 TERMINACIÓN DE UNA TAREA

La interfaz *Callback* define un solo método denominado *callback(int)*, el cual será llamado cuando una tarea concluye ya que se debe indicar a la *Cola de hilos* sobre este acontecimiento con el propósito de llevar a cabo las acciones para verificar si hay tareas por procesar, o bien, concluir con la ejecución de la *Cola de hilos*. La figura 2 muestra el diseño de la interfaz.

Figura 2

Diseño de la interfaz Callback



4.3 COLA DE HILOS

La clase *ThreadQueue* representa la propia *Cola de hilos*, misma que para su operación implementa las interfaces *Queue*, del lenguaje *Java*, y *Callback* que se definió.

ThreadQueue consta de tres atributos: *queue*, *activeThreadsList* y *totalActiveThreads*, donde:

- **queue:** es la cola de tareas a ejecutar.
- **activeThreadsList:** la lista de hilos ejecutándose.
- **totalActiveThreads:** número de tareas que se pueden ejecutar a la vez.

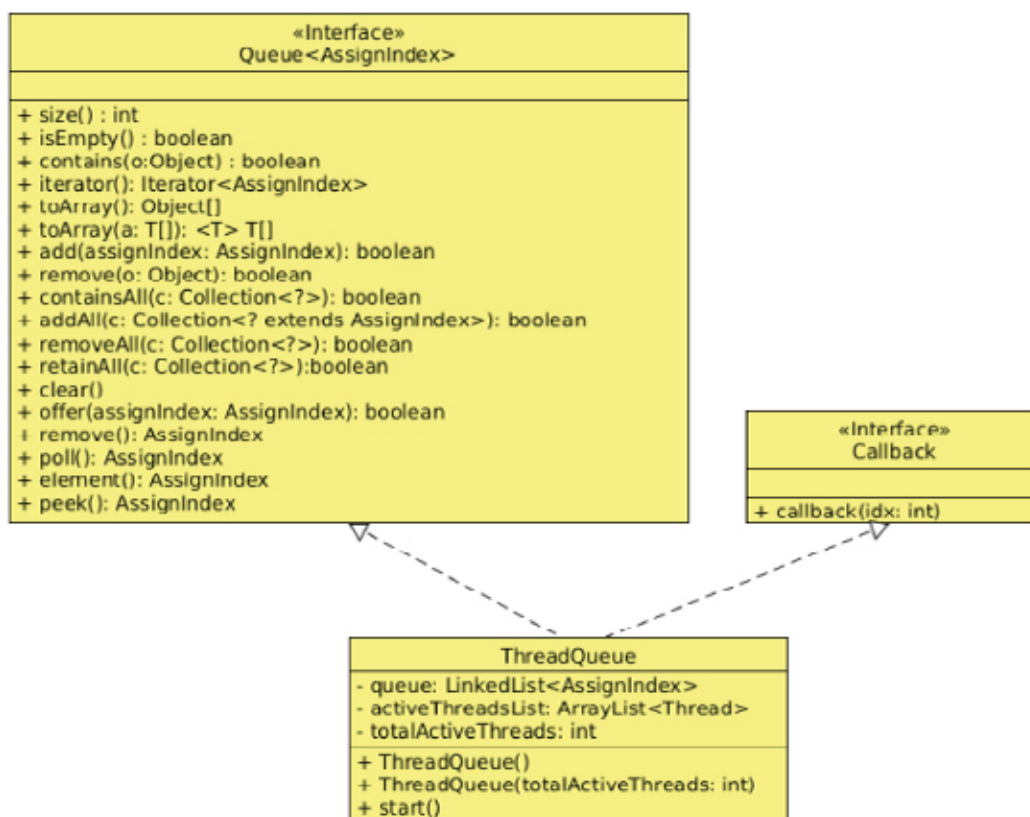
Asimismo, cuenta con los siguientes métodos:

- **ThreadQueue():** constructor por omisión.
- **ThreadQueue(int):** constructor que indica el número de hilos activos.
- **start():** inicia la ejecución de la *Cola de hilos*.
- **callback(int):** método que cada tarea ejecuta cuando concluye su propia ejecución.

La figura 3 muestra el diseño completo de la clase *ThreadQueue*.

Figura 3

Diseño de la clase ThreadQueue



Nota. La interfaz *Queue* es parte de los paquetes estándar de *Java* (Friesen, J. 2015, Liang, Y., 2021 y *Oracle Java Documentation. 2023b*). Para más información de su definición se recomienda al lector consultar la documentación oficial de *Oracle*: <https://docs.oracle.com/javase/8/docs/api/java/util/Queue.html>

5. CREANDO TAREAS PARA LA COLA DE HILOS

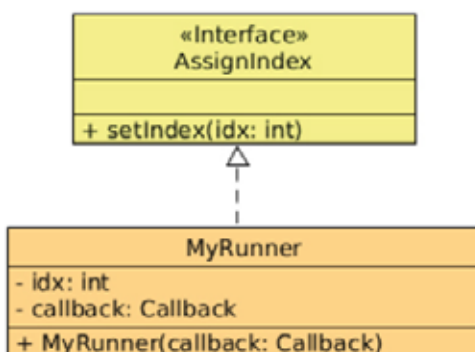
Para que una tarea pueda agregarse y posteriormente ejecutarse dentro del control de la *Cola de hilos* como se ha mencionado anteriormente, es necesario que implemente la interfaz *AssignIndex*. En otras palabras, la clase que represente una tarea deberá incluir el método *run()* que tendrá las operaciones que se ejecutarán mediante un hilo que se le asignará dentro de la *Cola de hilos*. El segundo método que debe programarse es *setIndex()*, que será invocado cuando la tarea pase a la lista de hilos que se están ejecutando, entonces el *ThreadQueue* asociado indicará con este método el lugar que ocupará para su control.

Cuando una tarea concluye su ejecución, deberá invocar al método *callback()* de la *Cola de hilos*, por lo que se debe tener una referencia desde la tarea. El parámetro que se envía a este método deberá ser la misma posición asignada cuando la tarea se agregó a la lista de hilos en ejecución.

El diseño de una tarea básica puede apreciarse en la figura 4.

Figura 4

Diseño de una tarea



La clase *MyRunner* puede ser programada para la ejecución de una tarea particular permitiendo que su diseño atienda las especificaciones de operación que se definan en su contexto de operación.

Como resultado del diseño una misma *Cola de hilos* puede ejecutar tareas muy distintas como resultado de la relación de funcionalidad mediante el uso de interfaces del lenguaje *Java*.

Retomando el ejemplo del navegador, para tener una tarea de descarga de imágenes se debe crear una clase para este propósito. Al igual que en el caso de *MyRunner* la nueva clase debe indicar la implementación de la interfaz *AssignIndex*, permitiendo la asignación de su posición en la *Cola de hilos* y al terminar la descarga indicar dicha posición.

6. USO DE LA COLA DE HILOS

Para hacer uso de la *Cola de hilos* solo se requiere tres pasos: crear un *ThreadQueue* a la programación, agregar las tareas que se quieren ejecutar al objeto *ThreadQueue* y finalmente iniciar la ejecución de la cola. La figura 5 ejemplifica estas tres acciones.

Figura 5

Diseño de una tarea

```

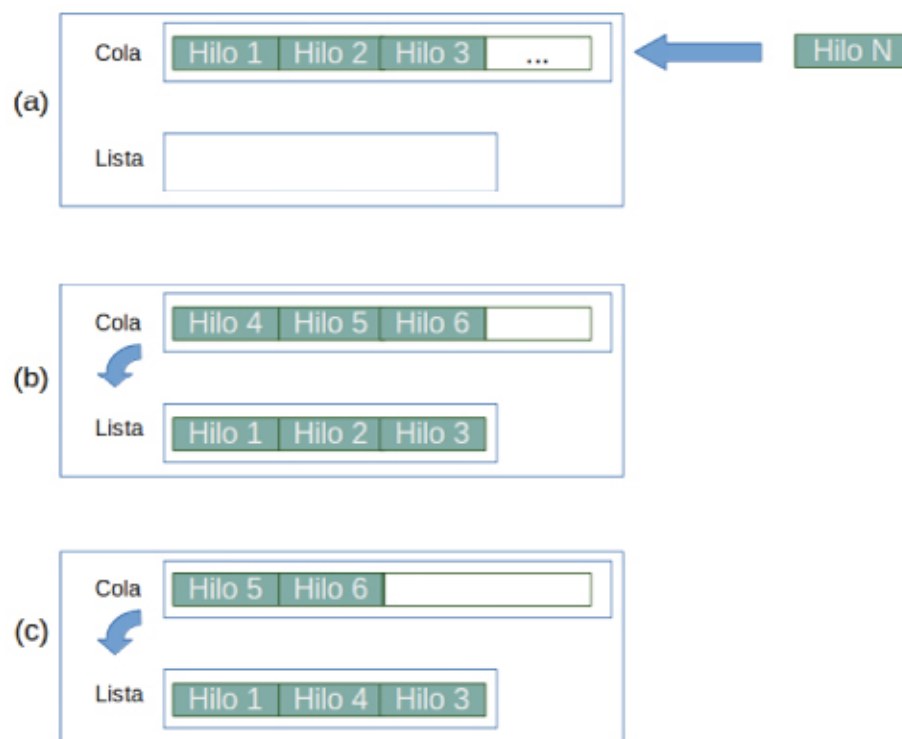
1  ThreadQueue threadQueue = new ThreadQueue(5);
2
3  threadQueue.add(new MyRunner(threadQueue));
4  threadQueue.add(new MyRunner(threadQueue));
5  threadQueue.add(new MyRunner(threadQueue));
6  threadQueue.add(new MyRunner(threadQueue));
7
8  threadQueue.start();
    
```

En la línea 1 se indica la creación de una *Cola de hilos* que en este caso tendrá a lo más cinco hilos ejecutándose a la vez. Las líneas 3 a 6 muestran la carga de tareas, en este caso mediante la creación de objetos *MyRunner* según su especificación. La línea 8 se llama a la ejecución de la *Cola de hilos*, misma que comenzará la ejecución de las tareas indicadas y conforme vayan concluyendo estas tareas se irán ejecutando las pendientes.

La Figura 6 resume la operación de la *Cola de hilos* que se ha diseñado. En primer lugar los hilos que se van a ejecutar se cargan en la cola interna (a). Una vez completada la carga, los primeros hilos se sacan de la cola interna y pasan a la lista donde éstos comienzan su ejecución (b). Cuando una tarea en la lista completa su ejecución, su lugar será ocupado por el siguiente hilo en la cola interna (c), repitiendo esta acción hasta que ya no haya más hilos en espera de ejecución.

Figura 6

Operación de la Cola de hilos



7. CONCLUSIONES

El uso de la *Cola de hilos* que se diseñó previene la demanda excesiva de recursos de cómputo en aplicaciones donde existen diversas tareas ejecutándose en paralelo. Su uso en una aplicación proporciona diversos beneficios tales como evitar el bloqueo de la aplicación, controlar la cantidad de tareas concurrentes que se ejecutan, impedir la lentitud del equipo o prevenir la interrupción catastrófica de equipos con recursos limitados.

El diseño presentado se puede mejorar según el contexto donde se necesite, por ejemplo, en lugar de una cola simple se puede complementar su programación para implementar una cola con prioridad donde las tareas se ejecutarán con base en un criterio de precedencia cada vez que una nueva tarea se ejecute; o bien, extender la funcionalidad incluyendo métodos para pausar la cola o incluir mejoras sobre el proceso de excepciones.

Con el uso de esta nueva estructura se alcanzan varios beneficios ya que contribuye a la ejecución ordenada de tareas en un equipo de cómputo, el número de hilos que se pueden ejecutar al mismo tiempo se limita desde la construcción de la *Cola de hilos* y conforme éstas van concluyendo su ejecución, su lugar será ocupado por una nueva hasta que no haya más.

REFERENCIAS BIBLIOGRÁFICAS

Liang, Y. (2021). *Introduction to Java™ Programming and Data Structures*. (12a ed.). USA: Pearson

Friesen, J. (2015). *Java Threads and the Concurrency Utilities*. (1a. ed.). USA: Apress

Oracle Java Documentation (2022). *Concurrency*. Recuperado el 21 de octubre de 2023, de <https://docs.oracle.com/javase/tutorial/essential/concurrency/index.html>

Oracle Java Documentation (2023a). *Interface Runnable*. Recuperado el 21 de octubre de 2023, de <https://docs.oracle.com/javase/8/docs/api/java/lang/Runnable.html>

Oracle Java Documentation (2023b). *Interface Queue*. Recuperado el 21 de octubre de 2023, de <https://docs.oracle.com/javase/8/docs/api/java/util/Queue.html>

ANEXO A

Una implementación libre en el lenguaje de programación *Java* de la *Cola de hilos* presentada, se ubica en el repositorio institucional de la DGTIC, la cual puede accederse en la siguiente dirección:

<https://kwira-kaab.unam.mx/badboy/threadqueue>

Este desarrollo cuenta con los siguientes archivos:

- *App.java*, archivo principal de ejecución.
- *AssignIndex.java*, interfaz para las tareas que se ejecutarán.
- *Callback.java*, interfaz de la llamada de vuelta para la conclusión de una tarea.
- *ThreadQueue.java*, implementación de la *Cola de hilos*.

Se incluyen además dos ejemplos de implementación de la interfaz *AssignIndex*, las cuales son:

- *MyRunner.java*, muestra una tarea de consumo variable de tiempo.
- *DownloadRunner.java*, muestra una tarea destinada a la descarga de imágenes.

Para la programación del componente se usaron las siguientes tecnologías:

- *Eclipse 2023-09 (4.29.0)*.
- *JDK 20*.

La clase *DownloadRunner* usa la biblioteca *Apache Commons IO™* para simplificar la descarga de archivos que se encuentran disponibles desde algún sitio Web usando su *URL* mediante la clase *FileUtils*.

Existen otras alternativas a esta tecnología como es el uso del paquete *Java NOI*, para la construcción de componentes a bajo nivel, o bien, se pueden emplear otras bibliotecas como el caso de *AsyncHttpClient (AHC)* de *Netty*.

Uso de contenedores para la construcción de productos de software

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:

Barajas González, L. D. (2023). Uso de contenedores para la construcción de productos de software. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (72 - 81).
<https://doi.org/10.22201/dgtic.ctud.2023.1.1.15>

Luis Daniel Barajas González

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

ldanielbg@comunidad.unam.mx

ORCID: 0009-0001-7515-4667

Resumen:

La tecnología de contenedores ha surgido como una alternativa efectiva para agilizar el desarrollo de *software*. Encapsular las herramientas y librerías de programación en entornos de ejecución virtuales, permite que sean compartidos en el equipo de trabajo sin requerir tiempo o esfuerzo excesivo para su instalación y configuración. Lo anterior crea ambientes de trabajo homogéneos que previenen problemas de incompatibilidad.

Palabras clave:

Contenedores, Docker, ambientes, software.

1. INTRODUCCIÓN

Las herramientas de software mínimas que conforman un ambiente de desarrollo para sistemas web transaccionales son:

- a) un servidor web que habilite el acceso al sistema vía Internet,
- b) un manejador de bases de datos para almacenar y recuperar datos,
- c) un lenguaje de programación para construir componentes ejecutables de *software*,
- d) un servidor de correo para el envío y recepción de mensajes del sistema,
- e) un editor de código, y
- f) un programa que permita interactuar con la base de datos.

Figura 1

Conjunto de herramientas mínimas para desarrollo de un sistema web



Todos estos elementos deben ser instalados en el equipo de cómputo de cada programador que participe en el desarrollo de un sistema web transaccional.

Los principales retos a los que se enfrenta un programador al trabajar en un esquema como el antes descrito son:

1. Instalar y configurar cada uno de estos programas en el equipo de cómputo asignado.
2. Lograr que todos los programadores cuenten con un ambiente de trabajo configurado con los mismos elementos de *software*.

Surgen problemas de incompatibilidad en los componentes de *software* construidos cuando los ambientes de programación divergen, por ello, la adopción de prácticas y herramientas de trabajo permite superar los retos antes planteados a través del encapsulamiento de programas de *software* dentro de contenedores.

2. OBJETIVO

Compartir el conocimiento adquirido sobre el uso de contenedores que permite reducir el tiempo requerido para instalar y configurar ambientes de programación para sistemas web en equipos de trabajo.

3. ANTECEDENTES

La herramienta *XAMPP* era utilizada para programar sistemas web en la Subdirección de Sistemas Integrados de la DGTIC; esta herramienta es un paquete de *software* que simplifica la instalación de un servidor web *Apache* configurado para interpretar al lenguaje *PHP*. También se instalaba un manejador relacional de bases de datos como *PostgreSQL* o *MySQL* para persistir los datos del sistema. Estos programas eran instalados directamente sobre el sistema operativo del equipo y sus versiones podían variar conforme a los requerimientos de cada proyecto.

Constantemente se desarrollaban nuevos sistemas y se modificaba la funcionalidad de sistemas legados. Por lo tanto, era necesario tener disponibles distintas versiones de las herramientas. Por ejemplo, para modificar un sistema construido con *PHP* versión 7 y al mismo tiempo trabajar en un nuevo sistema con *PHP* versión 8.2, se necesita contar la instalación de ambas versiones del lenguaje.

3.1 PROBLEMÁTICA

A. Instalar y configurar ambientes locales para sistemas distintos.

Para utilizar diferentes versiones de *PHP* en un mismo equipo era necesario instalar más de una instancia de *XAMPP*, configurar los puertos de los servidores web *Apache* y *PostgreSQL* para evitar colisiones con instalaciones anteriores de ambos programas, así como establecer la ruta de ejecución de *PHP* en el sistema operativo para que coincida con el proyecto en turno.

Lo anterior presentaba algunos problemas, como el uso de espacio en disco para programas duplicados y la tarea adicional de modificar la configuración de variables del sistema operativo cuando era necesario pasar de un proyecto a otro para que funcionara sobre el ambiente adecuado. En ocasiones era importante desinstalar herramientas de versiones nuevas y reemplazarlas con otras antiguas para mantener la compatibilidad.

B. Ambientes de programación en equipos de trabajo

Al trabajar con otros programadores, un reto es lograr que todos y cada uno de ellos tenga los elementos de *software* instalados y configurados con las mismas versiones y parámetros. Aunque esto pudiera parecer solamente una cuestión de buena comunicación entre personas, en la realidad el ambiente de cada integrante del equipo es diferente: pueden tener distintos sistemas operativos (*Windows*, *Linux* o *MacOS*), o bien, diferentes versiones de un mismo sistema operativo (*Windows 10* o *Windows 11*); también pueden tener instaladas y configuradas versiones antiguas del lenguaje de programación, debido a su participación en proyectos previos.

Cuando divergen demasiado los ambientes de programación de un mismo equipo de trabajo, aparece un fenómeno en el cual la funcionalidad construida por un programador solo funciona en su equipo, lo cual es inaceptable porque los componentes de *software* construidos entre todos los miembros del equipo deben funcionar en armonía sobre un ambiente tecnológico homogéneo.

3.2 ALTERNATIVAS DE SOLUCIÓN

A continuación, se desglosan algunas soluciones que se exploraron para subsanar la problemática antes descrita.

A. Uso de máquinas virtuales

El aislamiento de ambientes de ejecución de sistemas de *software* tiene sus orígenes en diversos esfuerzos orientados a la virtualización de equipos de cómputo donde destacan nombres como *VMWare*, *Oracle VirtualBox* y *Virtuozzo*. La principal ventaja que ofrece la virtualización es compartir recursos de almacenamiento, memoria y procesador de un solo equipo físico entre muchos equipos virtuales cuya demanda de recursos sea significativamente moderada (Younge, et al, 2011).

El análisis de esta solución concluyó que el uso de máquinas virtuales para el desarrollo de *software* es una solución parcial al problema de instalar los componentes del ambiente de desarrollo sin afectar al equipo anfitrión. Adicionalmente, se identificó que esta opción consume demasiados recursos de almacenamiento y memoria debido a que implica la instalación de un sistema operativo completo, que se ejecutará sobre el equipo anfitrión. Para el desarrollo de *software*, no son necesarias muchas de las librerías, programas y funcionalidades instaladas como parte del sistema operativo de la máquina virtual, y constituyen un desperdicio de recursos. Es por lo anterior, que la virtualización completa de equipos no es la mejor solución para aislar ambientes de desarrollo de *software*.

B. Tecnología de Contenedores

La tecnología de contenedores ha sido un parteaguas para el desarrollo de *software*, porque permite aislar el ambiente de ejecución de una aplicación, del resto del sistema sobre el que funciona. Esto permite conformar arquitecturas de sistemas conectando servidores web, manejadores de bases de datos, lenguajes de programación y cuanto *software* sea necesario sin afectar directamente el ambiente operativo del equipo anfitrión.

Respecto al uso de recursos, los contenedores ocupan menos espacio que las máquinas virtuales debido a que no instalan programas adicionales innecesarios para el desarrollo de *software*. Además, todos los contenedores comparten el mismo *kernel* del sistema operativo para acceder a los recursos computacionales del equipo donde se ejecutan (Preeth, Mulerickal, et al., 2015).

C. Paquete de software XAMPP

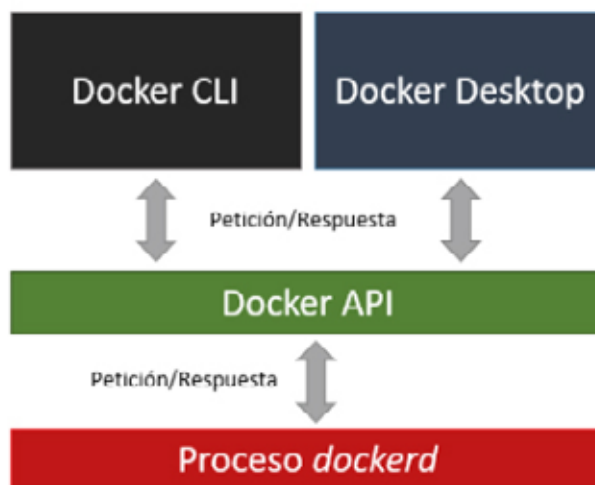
La herramienta *XAMPP* es un apilamiento (*stack*) de tecnologías que se usan para desarrollar sistemas web, que pueden instalarse fácilmente a través de un programa que guía paso a paso el proceso. La ventaja principal de este tipo de herramientas es que simplifican la puesta en marcha del ambiente de desarrollo, y conjunta un servidor web que además interpreta *scripts* de *PHP* y con múltiples librerías adicionales al lenguaje. La desventaja está en que los programas se instalan directamente sobre el equipo de cómputo y quedan acoplados al equipo, a su ruta de ejecución (*PATH*) y a los puertos de red. En caso de necesitar versiones más nuevas o antiguas de *XAMPP* es mandatorio instalar una instancia diferente y modificar la configuración del equipo.

4. LA PLATAFORMA DOCKER

La plataforma que popularizó el uso de contenedores fue *Docker* en el 2013 al lanzar su "motor" de contenedores conocido como "*docker engine*", el cual permite crear y ejecutar contenedores de aplicaciones. *Docker* funciona como una arquitectura cliente-servidor, donde existe un proceso servidor conocido como "*dockerd*" que se ejecuta en segundo plano y es el responsable de construir y ejecutar los contenedores. Los clientes utilizan la interfaz de programación de aplicaciones (API) de *Docker* para crear y utilizar los contenedores. Existen clientes gráficos tales como *Docker Desktop*, pero también es posible utilizar la interfaz de línea de comandos (CLI) de *Docker* para interactuar con la herramienta.

Figura 2

Arquitectura cliente-servidor de la plataforma Docker



4.1 COMPONENTES DE DOCKER COMO PLATAFORMA

Docker proporciona tres herramientas principales. La primera de ellas es el motor (*Docker engine*) que funciona como el gestor principal de otros elementos relacionados con la tecnología de componentes, tales como las imágenes, los contenedores, los volúmenes y las redes virtuales. La segunda herramienta son los clientes que se mencionaron antes: la API de línea de comandos y el cliente gráfico *Docker Desktop*, los cuales permiten el acceso a la funcionalidad del motor. La tercera herramienta es el servicio de alojamiento de imágenes conocido como *Docker Hub*, el cual es de uso público y permite subir y compartir imágenes con la comunidad de usuarios (Overview of Docker Hub, 2023).

4.1.1 OBJETOS DE DOCKER

Al trabajar con contenedores para el desarrollo de sistemas se identifican cuatro tipos de objetos comunes que son: 1) las imágenes, 2) los contenedores, 3) los volúmenes, y 4) las redes.

Las imágenes pueden comprenderse como una especie de plantillas a partir de las cuales se crean los contenedores. Las imágenes contienen los archivos binarios y librerías necesarias para ejecutar un tipo

específico de aplicación. A partir de una imagen es posible crear múltiples contenedores para aplicaciones distintas. Haciendo un símil, las imágenes son parecidas a los archivos *ISO* de las distribuciones *Linux* que pueden descargarse de Internet para generar un disco de arranque ya sea sobre un disco compacto o sobre un dispositivo *USB*. Estos archivos *ISO* contienen el *software* necesario para instalar y hacer funcionar *Linux* en una computadora personal. Es posible crear imágenes personalizadas a partir de archivos conocidos como *Dockerfiles*, los cuales son archivos de texto plano que contienen las instrucciones de instalación de los archivos binarios y las librerías con la imagen en cuestión. Posteriormente, estos archivos son procesados por un comando de *Docker* para generar la imagen.

Los contenedores son ambientes de ejecución para aplicaciones, estos nos permiten “encapsular” un sistema a la medida junto con sus dependencias para que se ejecute correctamente sin ser afectado por los cambios en la configuración del equipo anfitrión. Los contenedores se crean a partir de una imagen base y es posible configurar variables propias de cada instancia, por ejemplo, los puertos de red que mantendrán abiertos, o las contraseñas de los servicios que contienen.

Los volúmenes son espacios de almacenamiento administrados por *Docker* que permiten almacenar archivos de trabajo tales como el código fuente para usarlos desde un contenedor. Para usarlos, es necesario crearlos y “montarlos” sobre algún directorio del contenedor durante la creación del mismo. El contenedor tendrá acceso a los archivos del volumen de forma transparente.

Docker permite crear **redes virtuales** con el propósito de comunicar contenedores entre sí. Por ejemplo, un contenedor “A” que ejecuta una aplicación web, conectado a la misma red virtual que un contenedor “B” que ejecuta un manejador de bases de datos, podrá establecer conexión para consultar y almacenar datos en este último.

Figura 3

Objetos de Docker



5. CASO DE USO

A continuación, se presenta la creación de un ambiente de desarrollo basado en contenedores de *Docker* para una aplicación web. Está conformado por una red virtual de *Docker*, un contenedor para la base de datos, otro contenedor para un servidor de correo electrónico y uno más para ejecutar el sistema web. Este ambiente sirvió para el desarrollo de un sistema para el archivo histórico de un cuerpo colegiado de la Universidad, el cual fue desarrollado con el marco de trabajo *Laravel* que utiliza los lenguajes de programación *Javascript* y *PHP*.

Primeramente se creó una red virtual para conectar y comunicar todos los contenedores del ambiente de desarrollo.

Figura 4

Instrucción para crear la red virtual con Docker

```
docker network create local-net
```

Se utilizó una imagen del manejador de bases de datos *PostgreSQL* versión 15 para crear un contenedor para la base de datos. Se utilizó un volumen de *Docker* para que los datos quedaran almacenados independientemente del contenedor, lo que permite destruir y volver a generar el mismo sin perder información del sistema.

Figura 5

Instrucciones para la creación del contenedor de la base de datos

```
docker run -itd --name postgres-15 \  
  --network local-net \  
  -e POSTGRES_USER=postgres \  
  -e POSTGRES_PASSWORD=s3cret \  
  -p 5432:5432 \  
  -v postgres-data:/var/lib/postgresql/data \  
  -h postgres.hostname \  
  postgres-15:latest
```

También se creó un servidor de correo electrónico falso utilizando la imagen *Docker* del *software Mailhog*. Este *software* recibe las peticiones de envío de correo electrónico del sistema y muestra los mensajes en una página web simulando ser un cliente de correo real.

Figura 6

Instrucciones para la creación del contenedor de Mailhog

```
docker run -d \  
  -e "MH_STORAGE=maildir" \  
  -h mailhog.hostname \  
  --network local-net \  
  -p 1025:1025 \  
  -p 8025:8025 \  
  -v mailhog-maildir:/maildir \  
  mailhog/mailhog
```

El contenedor que ejecuta la aplicación web está basado en una imagen personalizada que conjunta un servidor web *Apache*, el lenguaje de programación *PHP* en su versión 8, el entorno de ejecución *Node JS*, el manejador de dependencias *Composer* y diversas librerías para que funcione el marco de desarrollo *Laravel*. Esta imagen personalizada está disponible para descarga desde *Docker Hub* a través de la dirección electrónica <https://hub.docker.com/repository/docker/danielbg1409/webphp-8.2/general>.

Figura 7

Instrucciones para crear el contenedor de la aplicación web

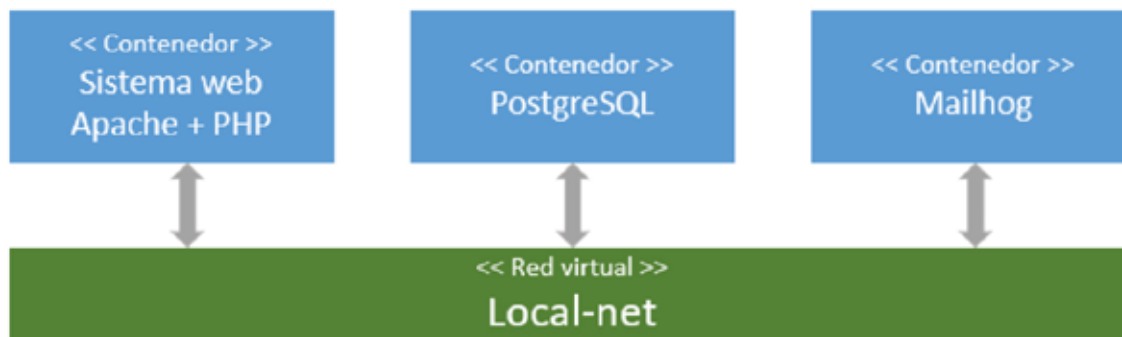
```
docker run -itd \  
  --name junta-gobierno \  
  -v v-junta-gobierno:/var/www/html \  
  --network local-net \  
  -p 80:80 \  
  -p 5173:5173 \  
  -h junta-gobierno.hostname \  
  danielbg1409/web-php-8.2:latest
```

Este último contenedor también utilizaba un volumen de *Docker* para mantener el código fuente del sistema desacoplado del contenedor.

Al momento de crear cada contenedor se conectaba a la red mediante el parámetro "*network*", esto permitió que el contenedor del sistema web estableciera conexión con la base de datos y con el servidor de envío de correo.

Figura 8

Los contenedores del ambiente conectados entre sí



Una vez creados los contenedores, se utilizaron editores de código y clientes de bases de datos para trabajar directamente sobre ellos. El código fuente, así como los datos quedaban guardados en volúmenes de *Docker*, de tal forma que, si el contenedor era destruido, no había ninguna pérdida de información porque el contenedor y el volumen son gestionados por *Docker* como objetos independientes.

Otra de las ventajas de *Docker* fue que al separar los componentes en tres contenedores distintos se reutilizaron los contenedores de *PostgreSQL* y de *Mailhog* en otros proyectos. También se reutilizó la imagen personalizada del servidor web en otros proyectos que utilizan el mismo conjunto de tecnologías.

Debido a que los programas quedan encapsulados en contenedores, no afectan a otros programas instalados en el equipo anfitrión, tampoco alteran sus variables de entorno ni la ruta de ejecución del sistema operativo. Esto permite tener contenedores con ambientes de desarrollo para el mantenimiento de sistemas legados junto con ambientes actualizados sin que se afecten entre ellos.

Los programadores del equipo de trabajo instalaron *Docker* en sus computadoras y posteriormente ejecutaron los comandos antes mencionados para habilitar su ambiente de desarrollo con los mismos elementos de *software*, evitando así problemas de incompatibilidad en los artefactos de *software* que construyeron.

6. CONCLUSIONES

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, a diferencia de otro tipo de soluciones basadas en *stacks* de tecnologías como *XAMPP*. 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.

REFERENCIAS BIBLIOGRÁFICAS

- Overview of Docker Hub. (2023, September 19). *Docker Documentation*. <https://docs.docker.com/docker-hub/>
- Preeth, E. N., Mulerickal, F. J. P., Paul, B. and Y. Sastri, (2015). *Evaluation of Docker containers based on hardware utilization*, International Conference on Control Communication & Computing India (ICCC), Trivandrum, India, 2015, pp. 697-700, doi: [10.1109/ICCC.2015.7432984](https://doi.org/10.1109/ICCC.2015.7432984)
- Younge, A. J., Henschel, R. Brown, J. T. Laszewski, G. von. Qiu, J. and G. C. Fox. (2011). *Analysis of Virtualization Technologies for High Performance Computing Environments*, IEEE 4th International Conference on Cloud Computing, Washington, DC, USA, 2011, pp. 9-16, doi: [10.1109/CLOUD.2011.29](https://doi.org/10.1109/CLOUD.2011.29)

Analítica de negocio desde la oficina de gestión de proyectos

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. (2023). Analítica de negocio desde la oficina de gestión de proyectos. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (82 - 93).

<https://doi.org/10.22201/dgtic.ctud.2023.1.1.11>

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:

En un mercado dinámico y competitivo y una sociedad cada vez más tecnológica, los datos se consideran uno de los activos más importantes a nivel mundial para las organizaciones, son los impulsores de los negocios en estos días y la base de casi todas las decisiones importantes. En nuestro entorno universitario será importante tomar conciencia de las actividades necesarias para analizar datos a niveles operativos, es por ello que se describe una iniciativa interna para la obtención de éstos en proyectos de un área de la Dirección General de Cómputo y Tecnologías de Información y Comunicación, la Dirección de Colaboración y Vinculación. Para ello, la iniciativa fue gestionada y documentada en el marco de la Guía *PMBOK*, conocimientos de indicadores para el sistema de gestión de calidad basado en la norma *ISO 9001:2015*, análisis de datos y analítica de negocio. Entre las principales aportaciones se encuentra la relevancia de la gestión de alcance, tiempo, conocimientos y recursos. Se plantea como posible línea de acción para implantar una cultura de datos, la adopción del concepto de las oficinas de gestión de proyectos en la Universidad así como la necesidad de un gobierno de datos institucional.

Palabras clave:

Gestión de proyectos, análisis de datos, oficina de gestión de proyectos, PMO, analítica de negocio.

1. INTRODUCCIÓN

La cultura para la toma de mejores decisiones, informadas y basadas en datos a nivel dirección, estrategia u operación, es la base de la **analítica de negocio**, la cual ha sido impulsada desde hace algunos años por las tecnologías de la información, los modelos de la ciencia de datos y el interés de líderes de alta dirección que ponen énfasis en las decisiones basadas en datos, para ser adoptada, incorporada y respaldada en toda la organización. En el anexo A se presenta terminología asociada con la analítica de negocio.

Asimismo, el aprovechamiento de los **datos** representa beneficios estratégicos para ofrecer productos de calidad, optimizar recursos e incrementar las probabilidades de éxito en la gestión de proyectos, es por ello que entre las tendencias actuales que han tomado mayor impulso se encuentra la analítica de negocio para mejorar la toma de decisiones, favorecer el crecimiento empresarial y como consecuencia, la competitividad.

De acuerdo con los hallazgos del informe anual de 2021 sobre el estado de la gestión de proyectos (Wellintone, 2021), en el que participaron 214 organizaciones de Reino Unido y de otras partes del mundo, se observa lo siguiente:

- El 86% de las organizaciones reportaron que cuentan con una o más Oficinas de Gestión de Proyectos (*Project Management Office – PMO*) (con tendencia incremental desde 2016) y el 25% de las mismas tienen menos de dos años de haberse establecido.
- El 47% de las *PMO* no contaron con información en tiempo real de sus indicadores clave de rendimiento (*Key Performance Indicator - KPI*); cabe señalar que esta tendencia ha disminuido respecto al 2020, por eso es importante que los responsables de la toma de decisiones en las organizaciones tengan acceso a la información actualizada de los proyectos, que sea precisa y confiable para la obtención de los *KPI*.
- Entre sus responsabilidades, más del 50% de las *PMO* invierten uno o más días en la recolección de información y la generación de reportes sobre el estado de sus proyectos, siendo este informe una de las actividades que se realiza de manera recurrente junto con el mantenimiento de la cartera de proyectos y el seguimiento de la metodología de gestión de proyectos.

Por otra parte, se reconoce que en el futuro, los **administradores profesionales de proyectos** tendrán que cubrir habilidades actualmente demandadas, entre ellas una comprensión básica de cómo leer, comprender, crear datos, además de poder extraer información significativa y comunicar con confianza (Le Manh, P., 2022).

Ante este panorama general sobre el aprovechamiento de los datos, ¿qué pasa en el ámbito universitario? Al respecto, un estudio realizado por la investigadora Christine L. Borgman, de la Universidad de California en Los Ángeles y la directora y editora de *MIT Press*, Amy Brand del *Massachusetts Institute of Technology* en el que entrevistaron a 12 líderes universitarios de Norte América y Canadá, quienes aportaron su perspectiva en la gestión de datos en el ámbito universitario, donde se concluyó que la toma de decisiones basada en datos brinda oportunidades para promover una gobernanza transparente, la justicia y la equidad, así como el ahorro de recursos. Además, alientan a los líderes universitarios a adoptar modelos más objetivos y transparentes basados en datos para la toma de decisiones, y se expone que existen universidades que cuentan con datos, otras que no los tienen y otras que los tienen pero no los explotan (Borgman, C. L., & Brand, A., 2022).

De acuerdo con información del Cuarto Estudio del Nivel de Desarrollo de las TIC en las entidades y dependencias de la Universidad Nacional Autónoma de México (Dirección General de Cómputo y de Tecnologías de Información y Comunicación, y Secretaría de Desarrollo Institucional, 2023), en general en el año 2023 se observa una mejoría en la madurez de los sistemas universitarios, lo cual podría deberse a una de las respuestas a las necesidades por las que se enfrentan las entidades y dependencias universitarias al desarrollar mecanismos que les permitan organizar la información para su uso y explotación. Además, desde la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) se ha promovido la visión de contar con información que pueda ser analizada para su explotación correctiva y preventiva en la mejora de los servicios que ofrece, por ello este reporte se centra en una iniciativa de análisis de datos de proyectos de la DGTIC, a cargo del área denominada Dirección de Colaboración y Vinculación (DCV).

La DCV atiende distintos servicios y cuenta con equipos de trabajo que están a cargo de la gestión de proyectos acorde a sus áreas de especialidad, por lo que cuenta con lo que se puede considerar como una oficina de gestión de proyectos. Cabe mencionar que, las oficinas de proyecto se pueden identificar como una de las principales fuentes de datos cuya utilización permite la obtención de hallazgos significativos para resolver problemas organizacionales complejos, y tomar mejores decisiones de manera oportuna y con mayor precisión, que es uno de los campos de la analítica de negocio.

2. OBJETIVO

Presentar las implicaciones y los hallazgos obtenidos en la prueba piloto realizada para la obtención periódica de datos de proyectos alineados a la planeación estratégica de la DGTIC, susceptibles de convertirse en indicadores en el Sistema de Gestión de Calidad (SGC) de la DCV; para lo que se consideraron, como marco de referencia, algunas buenas prácticas descritas en la Guía de los Fundamentos para la Dirección de Proyectos (Guía *PMBOK*), así como conocimientos sobre indicadores para el SGC basados en la norma *ISO 9001:2015*, y conceptos de análisis de datos y de analítica de negocio.

3. SOBRE EL DESARROLLO

La DCV es un área que cuenta con recursos humanos que tienen experiencia y conocimientos en herramientas y gestión de proyectos, por lo que el lenguaje está normalizado y se aplican buenas prácticas establecidas en su SGC apegado a la norma *ISO 9001:2015*, que tiene indicadores definidos a nivel táctico. La prueba piloto tiene la intención de explorar datos de los proyectos que gestiona la DCV y que puedan ser candidatos a evolucionar a indicadores de nivel operativo adecuados al área.

3.1. METODOLOGÍA

Como se ha mencionado, la gestión de la prueba piloto para la obtención de datos de proyectos de la DCV tomó como base la aplicación de buenas prácticas descritas en el *PMBOK*, por lo que las actividades realizadas se distribuyeron en las fases de inicio, planeación, ejecución, seguimiento y control, y cierre.

En la fase de **inicio** se hizo la definición de la propuesta y se asignó la prioridad de atención, el periodo que las actividades abarcarían, los precedentes y el objetivo de la propuesta. En la fase de **planeación** se definieron tres ciclos para el alcance (figura 1), y un calendario de actividades que incluyó el desarrollo de un informe cuatrimestral, además, se especificó el equipo de trabajo considerando el involucramiento del

personal operativo y directivo y se describieron las actividades para recabar los datos de los proyectos, los mecanismos a utilizar, así como las revisiones y validaciones requeridas. Se **ejecutó** el plan de trabajo para obtener datos de proyectos del 2022 en cada ciclo y se dio **seguimiento** a las actividades y documentos técnicos a generar. Durante el cierre se hizo el ejercicio de lecciones aprendidas y se construyó una memoria de las actividades realizadas. La documentación generada y productos se conservan en un expediente dentro del repositorio digital de la DCV.

Figura 1

Estructura desglosada de trabajo para la prueba



3.1.1. COMPRENSIÓN DEL CONTEXTO

Para poder enfocar los esfuerzos de la iniciativa del área en relación con el propósito y visión de la Dirección General, se analizaron las funciones sustantivas de la DGTIC, el Manual de Organización aprobado de la DGTIC (Universidad Nacional Autónoma de México, 2022) y el Manual de Calidad de la

DCV (Dirección General de Cómputo y de Tecnologías de Información y Comunicación, 2023), así como otras fuentes de información, entre las que se encuentran algunos informes entregados a lo largo del año 2021, publicaciones y documentación de definición de los proyectos.

Se realizó un análisis inicial para lo cual se dividieron las actividades a realizar en cinco grupos de trabajo de acuerdo con sus áreas de especialidad. A partir de la extracción de información de las fuentes disponibles, se propuso un primer listado de datos por grupo, para lo que se consideró el reflejo de características y actividades específicas de los proyectos de 2022, y se pusieron a disposición con documentos compartidos que facilitaron su revisión. Se llevaron a cabo sesiones ágiles de forma presencial con los grupos y se convocó a diferentes actores clave identificados para complementar la propuesta inicial.

3.1.2. IDENTIFICACIÓN DE DATOS CON IMPACTO SIGNIFICATIVO

Para el éxito de la prueba, se diseñó una actividad considerada como una estrategia competitiva para asegurar la centralización de esfuerzos en puntos clave: los grupos analizaron y discutieron la propuesta inicial de datos con lo que se obtuvo un gran conjunto de datos cuantificables. Con base en su experiencia, los actores involucrados discriminaron aquellos datos que tenían mayor importancia, seleccionaron los que podrían obtener y monitorear de manera continua a lo largo del año y que podrían contribuir a los objetivos estratégicos de la Dirección General.

3.1.3. ANÁLISIS DE DATOS

En los tres ciclos trimestrales ejecutados, se generaron instrumentos colaborativos para favorecer la recopilación y el registro ordenado de los datos de cada grupo usando hojas de cálculo compartidas por la facilidad que ofrecen para contar con datos estructurados accesibles para todos, dado el volumen de información obtenida en cada ciclo. Tras cada recolección, los datos se consolidaron en una estructura ordenada en la hoja de cálculo, conservando el grupo y el proyecto como identificadores; se realizó la limpieza, homogeneización y cruces de datos con funciones avanzadas, y se obtuvieron gráficas como parte del análisis descriptivo. A partir de lo anterior se generaron informes de seguimiento con elementos visuales para compartir los resultados. En esta actividad se aplicaron conocimientos del proceso de análisis de datos que consiste en: preguntar, preparar, procesar, analizar, compartir y evaluar los resultados.

3.1.4. SESIONES DE REVISIÓN DE DATOS

Se realizaron sesiones virtuales específicas con los subdirectores y la directora del área durante las cuales se revisaron los informes de resultados que plasmaron de manera clara lo que se obtuvo, e incluyeron elementos gráficos para facilitar su comprensión, revisión y discusión.

3.1.5 REGISTRO DE PROYECTOS EN EL SISTEMA GTIC

Se usaron las funcionalidades y campos del módulo de proyectos provisto por el sistema GTIC, el cual es utilizado para la gestión de servicios TIC ofrecidos por diferentes áreas de la DGTIC y está desarrollado con el *software* GLPI (*Gestionnaire Libre de Parc Informatique*).

Durante 2022, personal de la DCV realizó el registro de 42 proyectos en dicho sistema y actualizó la información semanalmente. Las características de los proyectos se distribuyeron en las siguientes secciones:

- 1.Registro de datos generales del proyecto:** Identificador, nombre, fecha de inicio, fecha de fin, responsable, porcentaje de avance, estado, prioridad, tipo, descripción y comentarios.

2. **Actividades:** Nombre, estado, tipo, duración, asignación, descripción, comentarios.
3. **Subproyectos:** Proyectos que dependen del proyecto principal.
4. **Documentos:** Productos documentales con nombre, fecha, descripción, comentario, asunto o enlace web.
5. **Recursos materiales:** Registro de licencias o certificados asociados al proyecto, en caso de aplicar.
6. **Notas:** Registro de información adicional, con el nombre del autor, fecha y hora.
7. **Gantt:** Diagrama de *Gantt* generado automáticamente por proyecto, y global.
8. **Histórico:** Registro cronológico automático de movimientos en el proyecto, con fecha, hora, usuario que hizo el cambio, campo cambiado y descripción del cambio.

4. RESULTADOS

Se ejecutaron tres ciclos para recabar información de los proyectos. En el primer ciclo se obtuvieron:

- Datos generales: aquellos que están presentes en todos los proyectos,
- datos transversales: aquellos que están presentes en varios proyectos, y
- datos específicos: aquellos que están enfocados en determinadas actividades por área.

Se realizó el primer informe con las siguientes secciones: numeralia detallada inicial del trabajo realizado por subáreas para revisión interna de la DCV, matriz de datos cualitativos y cuantitativos por sección para visualizar posibles cruces de información, gráficas con la información relevante, resumen de resultados a partir de los datos recopilados y recomendaciones.

Los principales resultados en el primer ciclo fueron los siguientes:

- El valor de llevar a cabo el proceso de análisis de datos para preparar, procesar, analizar y compartir los resultados, con el conocimiento adecuado y suficiente control.
- Las implicaciones del tiempo requerido para reunir datos manualmente de diferentes grupos en un solo instrumento estructurado, si se considera al perfil asignado para el seguimiento y la cantidad de información a integrar de las fuentes disponibles.
- La influencia que puede tener el uso de elementos gráficos para facilitar la visualización de datos en la presentación de resultados para su revisión.

En el segundo ciclo se hicieron ajustes respecto del anterior: se simplificó la recolección, el análisis y la presentación de los datos sobre la productividad orientada a los servicios que atiende el área; se realizó un segundo informe que contenía: información de la cartera de proyectos, gráficas relevantes, información registrada en GTIC, proyectos por tipo de servicio, datos de impacto institucional y de usuarios, así como una sección de hallazgos y recomendaciones. Asimismo, se construyó un panel colaborativo con tablas dinámicas en hojas de cálculo compartidas para facilitar la visualización del estado de los proyectos de la cartera 2022. Entre los resultados más destacados de este segundo ciclo, se encuentran los siguientes:

- La importancia de que los datos recabados de los proyectos cuenten con elementos de soporte para su revisión detallada, justificación o comprobación en todas las fases de la prueba.
- Contar con una gran cantidad de datos no es necesariamente bueno, es más importante contar con aquellos que tienen mayor impacto estratégico.
- La relevancia de la evaluación periódica del impacto de los resultados obtenidos para reorientar la prueba y dirigir los esfuerzos.

El tercer ciclo consistió en una actualización de la información para lo cual se utilizó una versión reciente del catálogo de servicios de la DCV y se profundizó en datos específicos por tipo de servicio. Los hallazgos y productos más importantes que se lograron fueron los siguientes:

- La selección de datos que se puede mantener o monitorear en el año, debe responder a la situación actual del área, en cuanto a recursos, tiempos y prioridad.
- El involucramiento de personas de diferentes niveles en la estructura organizacional tuvo los siguientes beneficios:
 - Las propuestas de cada ciclo se enriquecieron con la participación de perfiles específicos del área.
 - La revisión a cargo de los subdirectores y la directora del área fue fundamental para orientar los esfuerzos.
- La identificación de datos que son útiles y lo que se requiere para obtenerlos en la capa operativa, ejecutiva y directiva.
- Documentar lo que se realizó fue una excelente práctica a realizar por el seguimiento y control durante la prueba y como precedente para siguientes iniciativas. Los productos más importantes que se generaron son: plan y calendario de trabajo, instrumentos para el registro de datos ajustados a cada grupo de trabajo en los diferentes periodos, datos recabados, integrados, limpios, homogeneizados y estructurados para su procesamiento y análisis, tres informes ejecutivos con gráficos, una presentación de lecciones aprendidas y la memoria técnica de actividades y productos obtenidos.

En cuanto al uso del sistema GTIC, se construyó un documento de convenciones para estandarizar el registro de la información de los proyectos de la DCV en esa plataforma. Los principales hallazgos obtenidos son los siguientes:

- Se pueden establecer relaciones padre-hijo entre los proyectos en una estructura jerarquizada, en cuyo caso se tiene que considerar la cantidad de áreas, proyectos y niveles internos de agrupación, los cuales si bien son útiles, pueden incrementar la complejidad de la visibilidad, comprensión de la información o la explotación de la misma.
- La información extraída de los proyectos registrados permite la construcción de informes y gráficos, lo cual se hace de forma manual a través de una consulta de toda la información de los proyectos, que se descarga y procesa con el apoyo de una herramienta local o colaborativa.

5. CONCLUSIONES

El ejercicio realizado constituye un precedente de la obtención de datos ordenados y estructurados de proyectos y detalla las actividades de administración de la iniciativa realizada por la DCV para el análisis de datos, tras su recolección manual en el nivel operativo, como un camino para la exploración de datos que pudieran ser integrados al SGC del área. Se usaron herramientas para comprender el estado, las tendencias, el historial y la información de los proyectos; la implantación de prácticas formales de recolección de datos y analítica de negocio quedó fuera del alcance acordado en el área.

Asimismo, se destacan los siguientes resultados a tomar en cuenta por su aplicabilidad en líneas de trabajo relacionadas con el proceso de recolección de datos:

- Estimar los recursos humanos y materiales para recabar los datos en un proceso controlado y documentado.
- El involucramiento de personas con diferentes niveles en la estructura organizacional que contribuyan a la recolección de datos y a orientar los esfuerzos con la evaluación periódica de la iniciativa.
- Conocer y gestionar adecuadamente el proceso de análisis de datos para integrar elementos de visualización, de soporte y transparencia a los mismos, identificar aquellos que aportan valor y que se puedan obtener de manera periódica, además de ser útiles para diferentes niveles organizacionales.

La experiencia adquirida puede servir como punto de partida para ampliar el alcance en una siguiente fase, en la cual los datos se conviertan en métricas, luego en indicadores y posteriormente en metas, es decir, ampliar y formalizar el sistema de indicadores dentro del SGC del área, al considerar los siguientes pasos: 1) seleccionar e identificar indicadores, 2) asignar nombre y tipo de indicador, 3) detallar la forma de cálculo, periodicidad y fuentes de información para su obtención, 4) representación, 5) definición de roles y responsabilidades y, 6) definición de umbrales y objetivos.

A partir de las actividades realizadas se reconoce la relevancia de la identificación del **propósito de la analítica de negocio** desde las primeras etapas de las iniciativas relacionadas, pues a partir de eso, las PMO podrán estandarizar los procesos relacionados con la gobernabilidad de los proyectos de un área universitaria, que incluyen datos, métricas o indicadores en sus SGC. Por ello se propone que las áreas universitarias adopten un enfoque orientado a proyectos, lo que implica: acuñar el concepto de las oficinas de gestión de proyectos, analizar el valor que éstas pueden aportar en la recolección de datos al definir convenciones para obtener resultados comparables, además de dar importancia al tiempo y esfuerzo colectivo para la obtención de datos, para potenciar el uso de la información recabada y la difusión de los logros como incentivos del trabajo grupal.

Tomando en cuenta lo anterior, se incrementa el impacto de las PMO en la alineación estratégica y en la entrega de valor organizacional, debido a que los rubros más importantes en donde tienen gran influencia es en la integración de datos e información de proyectos estratégicos organizacionales, así como en la evaluación del cumplimiento de los objetivos estratégicos de nivel superior, por su relación con los sistemas de medición de la organización.

Para apoyar el uso y aprovechamiento de los datos a nivel institucional se recomienda analizar y trabajar en los siguientes puntos:

- Promover el alcance y los beneficios que la analítica de negocio puede ofrecer a la Universidad,

mediante el uso de los datos para apoyar la toma de decisiones informadas a nivel estratégico, impulsar el crecimiento y ayudar en la elección de herramientas y roles adecuados para llevarla a cabo (Programa *Harvard Business Analytics*, 2021).

- Maximizar el impacto de la analítica de datos para propiciar respuestas cada vez más ágiles y oportunas, acorde a las necesidades digitales modernas; hacer frente a los retos actuales y a una mejor comprensión del comportamiento, necesidades y habilidades de las entidades, dependencias y de la comunidad universitaria.
- Propiciar un cambio cultural profundo en la adopción de una cultura institucional de datos, debido a que no es suficiente adoptar soluciones tecnológicas: también se requiere invertir recursos en el gobierno de datos.
- Abordar con liderazgo la infraestructura o las políticas que permitan adquirir la experiencia necesaria y coordinar de manera efectiva los esfuerzos para cambiar la manera en la que se gestionan los datos (Borgman, C. L., & Brand, A., 2022): en su estructura, almacenamiento, interconectividad en los sistemas y tableros de visualización.
- Tomar conciencia de la forma actual en la que las áreas de la Universidad dan respuesta a las necesidades institucionales para el aprovechamiento de información estructurada y centralizada, así como los esfuerzos que se deben realizar para asegurar el impacto previsto.

Por último, como se mencionó anteriormente, se hace hincapié en que no es suficiente la adopción de soluciones tecnológicas; el uso y selección de herramientas de *software* para la recopilación de datos debe responder a necesidades estratégicas determinadas desde la alta dirección de entidades y dependencias universitarias y a planteamientos que se quieran solucionar con su uso. La implementación de *software* dedicado a la gestión de proyectos usado en las organizaciones para recabar información de los proyectos constituye la base dentro de la analítica de negocio. Si no existen datos seleccionados y coherentes no se podrán obtener los resultados deseados en su análisis.

REFERENCIAS BIBLIOGRÁFICAS

- Borgman, C. L., & Brand, A. (2022). *Data blind: Universities lag in capturing and exploiting data*. *Science*, 378(6626), 1278–1281. [10.1126/ciencia.add2734](https://doi.org/10.1126/ciencia.add2734).
- Deloitte. (2021). *Convertirse en una organización alimentada por IA, Perspectivas de Deloitte*. <https://www2.deloitte.com/us/en/insights/focus/cognitive-technologies/state-of-ai-and-intelligent-automation-in-business-survey.html>
- Dirección General de Cómputo y de Tecnologías de Información y Comunicación, y Secretaría de Desarrollo Institucional. (2023). *Cuarto estudio para el desarrollo de las TIC en la UNAM*. https://www.red-tic.unam.mx/recursos/2023/2023_EstudioTIC.pdf
- Dirección General de Cómputo y de Tecnologías de Información y Comunicación [DGTIC]. (2023). *Manual de Calidad*.

- Google Career Certificates. (2022). *Certificado profesional de Google Data Analytics*. Aspectos básicos: Datos, datos, en todas partes | Coursera. <https://mx.coursera.org/learn/aspectos-basicos-datos-datos-en-todas-partes>
- Le Manh, P. (2022, octubre 5). *Principales habilidades de gestión de proyectos*. El blog oficial del Project Management Institute. https://community.pmi.org/blog-post/73107/the-top-10-project-management-skills-you-need-in-your-toolbelt#_=_
- Programa Harvard Business Analytics. (2021). *Business Intelligence vs Business Analytics - Harvard Business*. <https://analytics.hbs.edu/blog/business-intelligence-vs-business-analytics/>
- Project Management Institute. (2017). *A Guide to the PROJECT MANAGEMENT BODY OF KNOWLEDGE (PMBOK® GUIDE) Sixth Edition*. <http://www.PMI.org>
- Rodríguez, P., Palomino, N., & Mondaca, J. (2017, mayo). *El uso de datos masivos y sus técnicas analíticas para el diseño e implementación de políticas públicas en Latinoamérica y el Caribe*. Banco Interamericano de Desarrollo. <https://publications.iadb.org/publications/spanish/document/El-uso-de-datos-masivos-y-sus-tecnicas-analiticas-para-el-diseno-e-implementacion-de-politicas-publi.pdf>
- Sharda, R., Turban, E., Delen, D., Aronson, J. E., Liang, T. P., & King, D. (2014). *Business Intelligence and Analytics: Systems for Decision Support*. Pearson. <https://books.google.com.mx/books?id=FLYDnwEACAAJ>
- Shende, V. (2018). *Literature review of Applications of Business Intelligence, Business Analytics and Competitive Intelligence*. International Journal of Scientific and Research Publications, 8(8), 782. <https://doi.org/10.29322/IJSRP.8.8.2018.p8099>
- Universidad Nacional Autónoma de México. (2022). *Manual de organización de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación*. <https://presupuesto.unam.mx/organi/biblioteca/522013A.pdf>
- Wellintone. (2021). *The State of Project Management Annual Report Wellington*. <https://wellintone.co.uk/publications/state-of-project-management-research/>
- Yalcin, A. S., Kilic, H. S., & Delen, D. (2022). *The use of multi-criteria decision-making methods in business analytics: A comprehensive literature review*. Technological Forecasting and Social Change, 174, 121193. <https://doi.org/10.1016/J.TECHFORE.2021.121193>

ANEXO A. TERMINOLOGÍA

Analítica de datos

La **analítica de datos** es la observación de los datos, a través de diferentes métodos, técnicas y herramientas. Se puede aplicar casi a cualquier área, por ejemplo: en procesos industriales, análisis de negocio, biomedicina, procesamiento de lenguaje natural, control de tráfico, seguros de vida, construcción, ingeniería, astronomía, entre otros (Yalcin, A. S., Kilic, H. S., & Delen, D., 2022a). El proceso de análisis de datos es un proceso iterativo y consta de seis fases o pasos dependiendo del autor: preguntar, preparar, procesar, analizar, compartir y actuar (*Google Career Certificates*, 2022), que se definen con mayor detalle a continuación:

- **Preguntar:** Hacer preguntas efectivas, formular hipótesis, explorar los datos disponibles, definir el problema a resolver, aplicar el pensamiento estructural, entender las expectativas de los involucrados.
- **Preparar:** Comprende la determinación de los datos necesarios, la generación y recopilación de los datos, así como identificar y usar diferentes formatos, tipos y estructuras, asegurar datos imparciales, consistentes y creíbles, organizados y protegidos.
- **Procesar:** Consiste en crear y transformar datos, mantener su integridad, probarlos y limpiarlos con la finalidad de que estén completos y sean correctos, verificar y compartir los resultados de la limpieza.
- **Analizar:** En esta fase se usan herramientas para transformar datos, se ordenan y filtran, se seleccionan, se aplican modelos, se definen algoritmos a utilizar, se identifican patrones y se obtienen conclusiones, también se interpretan los resultados y se extraen conclusiones útiles.
- **Compartir:** Comprende la visualización para ayudar a otras personas a comprender los resultados; a partir de la creación de imágenes efectivas.
- **Actuar:** En esta fase se resuelven los problemas planteados y se realiza la toma de decisiones basada en los resultados.

Cuando el análisis de datos se hace sobre un gran volumen de datos, se almacenan, procesan con gran velocidad y existe variación en las fuentes de datos, se denomina *Big Data*, debido a estas características se hace necesario adoptar aspectos de otras disciplinas científicas, como: estadística, aprendizaje automático, reconocimiento de patrones, teoría de sistemas, investigación de operaciones o inteligencia artificial (*Artificial Intelligence-AI*) (Rodríguez, Palomino, & Mondaca, 2017).

Analítica de negocio (*BA- Business Analytics*)

La **analítica de negocio** es la aplicación de herramientas, enfoques y principios de la analítica de datos para transformar datos en hallazgos claves para resolver problemas empresariales y organizativos complejos, y tomar mejores decisiones de manera oportuna y con mayor precisión. Investiga por qué sucedió y predice lo que puede suceder en el futuro (Shende, V., 2018). Se usa para la toma de decisiones basadas en datos históricos que provienen de distintas fuentes, incluso externas a la organización, y tiene los siguientes componentes (Yalcin, A. S., Kilic, H. S., & Delen, D., 2022a):

- **Analítica descriptiva:** Hace uso de estadísticas descriptivas, KPI así como herramientas para gestionar datos como informes, tableros, paneles u otros elementos de visualización de datos que son usados para analizar datos e identificar problemas potenciales y determinar las posibles acciones para resolverlos.
- **Analítica predictiva:** Se caracteriza por elementos como marcos predictivos y probabilísticos, estimaciones, análisis estadísticos y modelos de puntuación. Este tipo de analítica usa los datos recopilados en el tiempo para calcular posibles escenarios futuros, aquí se puede ubicar la minería de datos y texto, así como el aprendizaje automático el cual ya se usa en un 67% de las organizaciones de las economías con competencia en estrategias e inversiones de AI, de acuerdo a una encuesta de 2020 de Deloitte en la que entrevistó a 2,875 ejecutivos (Deloitte, 2021). La analítica predictiva se está volviendo muy popular en las organizaciones por su potencial.
- **Analítica prescriptiva:** Con su aplicación se sugiere una o más soluciones que aporten mayores beneficios a futuro, puedan demostrar los resultados posibles y sean el resultado del análisis de

grandes cantidades de datos. Combina los resultados del análisis predictivo y utiliza la AI, métodos y algoritmos de optimización, sistemas expertos en un entorno estocástico y redes neuronales artificiales para la toma de decisiones adaptativas, automatizadas, limitadas en tiempo y óptimas.

Inteligencia de negocio (BI- Business Intelligence)

La **inteligencia de negocio** es un término que combina arquitecturas, herramientas, bases de datos, herramientas analíticas, aplicaciones y metodologías empleadas para dirigir la gestión operativa de una organización. Se basa en la transformación de datos de la organización a información, luego a decisiones y finalmente a la implementación de acciones (Sharda, R., Turban, E., Delen, D., Aronson, J. E., Liang, T. P., & King, D., 2014). Informa sobre lo que sucedió en el pasado o lo que sucede en tiempo real (Shende, V., 2018).

Hay cinco componentes de la inteligencia de negocio: OLAP (procesamiento analítico en línea), análisis avanzado, gestión del desempeño corporativo (portales, cuadros de mando, tableros), inteligencia de negocio en tiempo real, almacén de datos, y fuentes de datos (Shende, V., 2018).

Tanto la analítica de negocio como la inteligencia de negocio apoyan la toma de decisiones informadas, impulsan el crecimiento de las organizaciones y ayudan en la elección de herramientas y roles adecuados para llevarla a cabo (Programa *Harvard Business Analytics*, 2021). La inteligencia de negocio se necesita para administrar el negocio, en tanto que la analítica de negocio se necesita para cambiar el negocio a lo largo de los años, esta última está siendo más estudiada (Shende, V., 2018).

Project Management Office (PMO)

De acuerdo con el *PMBOK*, una *PMO* es una entidad organizativa que estandariza los procesos de gobierno relacionados con los proyectos y facilita el intercambio de recursos, metodologías, herramientas y técnicas, se compone por personas expertas en la dirección de proyectos (*Project Management Institute*, 2017).

La función principal de una *PMO* es apoyar a los gerentes de proyectos en la gestión de recursos compartidos en todos los proyectos, desarrollando, administrando y supervisando el cumplimiento de las normas, políticas, procedimientos y plantillas de gestión de proyectos, coordinando la comunicación entre proyectos. Cabe mencionar que las responsabilidades de una *PMO* pueden variar y proporcionar funciones de apoyo a la gestión de proyectos, o bien ser realmente responsable de la gestión directa de un proyecto (*Project Management Institute*, 2017).

Integración de una aplicación PHP con el servicio de Identidad Digital Universitaria

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:

Cuéllar Martínez, H. C. (2023). *Integración de una aplicación PHP con el servicio de Identidad Digital Universitaria*. Cuadernos Técnicos Universitarios de la DGTIC, 1 (1), páginas (94 - 103).

<https://doi.org/10.22201/dgtic.ctud.2023.1.1.18>

Hugo Germán Cuéllar Martínez

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

hugo.cuellar@comunidad.unam.mx

ORCID: 0009-0005-1346-276X

Resumen:

Se presenta el procedimiento técnico para realizar la interconexión entre una aplicación *PHP* y el servicio de Identidad Digital Universitaria (*IDU*), el cual ofrece un registro de identidad biunívoca para alumnos y trabajadores universitarios.

Palabras clave:

OAuth, Laravel, interconexión entre sistemas.

1. INTRODUCCIÓN

Muchos servicios *TIC* universitarios cuentan con sus propios procesos para la identificación de personas, lo que origina que un usuario tenga muchos datos para iniciar sesión en los diversos sistemas que ofrece la UNAM. En contraste, el servicio de *IDU* toma como identidad digital la información que reconoce de manera unívoca a una persona como miembro de la UNAM, a partir de datos como la *CURP* y el *RFC*, con lo que se busca homologar los criterios para el control de acceso y la identificación de los usuarios.

Este reporte detalla el procedimiento técnico para realizar la comunicación entre un sistema desarrollado en el lenguaje de programación *PHP* y el marco de trabajo *Laravel*, a fin de integrarlo al uso del servicio de Identidad Digital Universitaria (*IDU*).

Este procedimiento técnico fue aplicado para la interconexión del sistema denominado “Sistema de seguimiento del proceso de titulación universitaria”, el cual permite que en las entidades académicas se genere el expediente de titulación del alumno, quien puede consultar el avance del proceso de titulación desde el envío de su solicitud hasta el inicio del proceso de emisión del título.

El sistema propuesto cumple con la característica de ser un sistema institucional que maneja entre sus usuarios, a los alumnos de las entidades académicas, así como a trabajadores de las áreas de servicios escolares que dan seguimiento a cada trámite. Partiendo de este universo definido, en este reporte se describen los pasos y las rutas necesarias para la obtención de los datos de cada usuario, alumno o trabajador.

2. OBJETIVO

Documentar el procedimiento necesario para replicar la incorporación del servicio de Identidad Digital Universitaria (*IDU*) al Sistema de seguimiento del proceso de titulación universitaria a fin de contribuir con el manejo y protección de los datos del usuario, obteniendo solamente los datos necesarios para el uso del sistema.

3. DESARROLLO

3.1 ANTECEDENTES

El servicio de Identidad Digital Universitaria (*IDU*) permite que cada miembro de la comunidad universitaria cuente con una identidad digital única, con la visión de evitar el manejo de múltiples cuentas de usuarios y de contraseñas por cada servicio *TIC* ofrecido en la Universidad Nacional Autónoma de México (UNAM). Promueve además un manejo adecuado de la información, al prevenir su duplicidad y/o pérdida.

En su origen, el servicio *IDU* fue desarrollado en su primera etapa con el lenguaje de programación Java, con un diseño y componentes complejos, lo que hacía difícil su actualización y ponía en riesgo la posibilidad de mantener altos estándares de seguridad. Posteriormente, el área a cargo de *IDU* implementó una herramienta capaz de administrar las identidades y los accesos a los usuarios de la comunidad universitaria a través de los diferentes servicios *TIC*, dando como resultado el surgimiento del actual servicio *IDU*.

3.2 PROBLEMAS POR RESOLVER

Identificar e integrar el flujo de autorización y autenticación del servicio *IDU* al *Sistema de seguimiento del proceso de titulación universitaria*, implementado en el lenguaje de programación *PHP* y el marco de trabajo (*framework*) de *Laravel*. Adicionalmente, documentar las referencias del proceso de interconexión para su uso en futuras intercomunicaciones con otros sistemas universitarios.

3.3 OAUTH

La interconexión de *IDU* con el *Sistema de seguimiento de titulación universitaria*, se realiza mediante el protocolo denominado *OAuth*, cuya versión más reciente y estable es la 2.0 (*OAuth 2.0*, 2012). Este protocolo permite la autorización segura de forma estándar, lo que garantiza que el usuario comparta su información con otra aplicación sin revelar toda su identidad.

OAuth funciona mediante el intercambio de un token que genera el proveedor y este es enviado al cliente, que en este caso es el *Sistema de seguimiento de titulación universitaria*. Este token está integrado por una cadena de caracteres que identifican al usuario, al cliente y a la duración del acceso.

Entre las ventajas de utilizar *OAuth* se encuentran:

- Evitar que el usuario comparta sus credenciales con aplicaciones o sitios web de terceros, lo que reduce el riesgo de robo o filtración.
- Permite definir el nivel de información; es decir, determinar los datos que se van a compartir, así como la duración del acceso, lo que reduce el riesgo de abuso o mal uso de la información del usuario.
- Es un protocolo muy utilizado por grandes compañías como son *Google*, *Facebook*, *Microsoft*, *GitHub* entre otros, ya que permite compartir información de sus cuentas con otras aplicaciones.

3.4 INTEGRACIÓN DE IDU CON EL SISTEMA DE SEGUIMIENTO DE TITULACIÓN UNIVERSITARIA

Una vez revisado el funcionamiento del protocolo *OAuth*, a continuación se detalla cómo se implementó la intercomunicación de *IDU* con el sistema de seguimiento.

Antes de empezar a integrar el proceso de autenticación y autorización de *IDU* al *Sistema de seguimiento de titulación universitaria*, es importante solicitar el servicio al área responsable, que requiere la siguiente información:

- **Nombre del cliente:** Corresponde al nombre de la aplicación.
- **URL¹ de redirección:** Se trata de la dirección electrónica (*URL*) a la que el servidor de autorización regresará la información cuando se realice el inicio de sesión.
- **URL de redirección de cierre de sesión:** Es la dirección electrónica (*URL*) a la que el servidor de autorización regresará la información al cierre de la sesión.
- **Scopes:** Especifica los grupos de parámetros a los que la aplicación solicitará el acceso, entre otros: nombre, *CURP*, correo electrónico.

¹ *URL* es el acrónimo que significa Localizador Uniforme de Recursos por sus siglas en inglés. Una *URL* es una secuencia de caracteres que identifica y localiza un recurso en Internet.

- **Consentimiento:** Define si se muestra o no la página de consentimiento de *IDU* al usuario, en el que pueda ver e indicar explícitamente los datos que compartirá con el sistema.
- **Descripción:** Se refiere a la información adicional que se quiera mostrar referente a la aplicación.

Con estos datos el servicio *IDU* identificará a la aplicación, así como el *URL* para recibir y enviar los datos que sean solicitados.

Una vez concluido el registro de la aplicación al servicio *IDU* es importante resguardar la información enviada, ya que incluye los datos para realizar la autenticación, su autorización y consulta, como sigue:

- **Issuer:** Es la dirección electrónica (*URL*) donde está el servidor *IDU* (<https://idu.unam.mx/sso/oauth2/unam>).
- **ID del cliente:** Corresponde al identificador único de la aplicación registrada en *IDU*.
- **Clave del cliente:** Es una cadena secreta (similar a una contraseña) que solamente es conocida por la aplicación y el servidor *IDU*.

IDU cuenta con tres procesos o servicios para obtener la información:

- El primer servicio consiste en solicitar una autorización al usuario, donde explícitamente se le informa qué datos se van a compartir (*scopes*); el usuario debe dar su consentimiento, el cual puede ser requerido por una única ocasión al ingresar la primera vez al *Sistema de seguimiento de titulación universitaria*, y la respuesta es almacenada dentro de *IDU* para próximas conexiones. Si este consentimiento es aceptado, se recibirá una concesión de autorización (*authorization grant*).
- El segundo servicio implica la obtención de un *token* de acceso (*access token*); para solicitar este *token* es necesario enviar la concesión de autorización previamente solicitada, lo que demuestra que es un cliente válido y que el usuario decidió compartir su información. Si todo lo anterior se cumple, el servicio emite el *token* de acceso (*access token*), y se considera que la autorización ha sido completada.
- El tercer servicio se refiere a recibir los recursos, es decir, a obtener los datos de la persona, para eso se necesita enviar el *token* de acceso que fue proporcionado en el servicio anterior; *IDU* se encargará de verificar y validar el *token*, y devolverá la información de la persona.

Figura 1

Diagrama donde se muestra la interacción del sistema de titulación con *IDU*



Nota. Se muestran los procesos para obtener la información del usuario descrita anteriormente.

Una vez revisado el flujo para la obtención de datos, se trabajó en la integración con *Laravel* para lograr la comunicación entre ambos sistemas.

En un primer paso se revisó *Guzzle* (2021), la librería que provee *Laravel* para hacer peticiones "http"², y se resolvió el envío de los datos solicitados al servicio de *IDU*. El uso correcto de esta librería requiere definir las rutas y cabeceras manualmente, y al no contar con toda esa información se recibían respuestas de error, además de tener que implementar la comunicación con las librerías de autenticación, manejo de usuarios y permisos.

En la búsqueda de otras librerías o componentes que realizaran la conexión, se encontró la librería llamada "*Socialite*" la cual está pensada precisamente para autenticarse con *OAuth* para servicios como *Google*, *GitHub*, *Facebook*, *LinkEdit*, *Slack*, entre otros, permitiendo personalizar el propio conector para un servicio personalizado como lo es *IDU*, e integrándose correctamente a las demás librerías de autenticación y manejo de usuarios que ofrece *Laravel*.

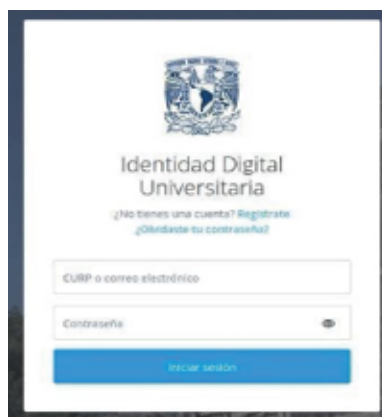
Para solicitar la autorización es necesario realizar una petición *GET*³ al servidor; se envían como parámetros el identificador del cliente, la URL de redirección, los alcances o permisos (*scopes*) y el tipo de respuesta o tipo de autorización que se requiere que devuelva el servicio *IDU*. Un ejemplo de esta petición sería:

https://idu.unam.mx/sso/oauth2/unam/authorize?client_id=sistema-firma-documentos&redirect_uri=http://localhost:5000/callback&scope=openid+profile+email&response_type=code

Aquí se puede notar que en el parámetro *response_type* tiene como valor la cadena "*code*", el cual indica al servicio *IDU* que queremos que nos devuelva un flujo de Autorización (para obtener recursos y confirmar que se concede dicha autorización). Si todo es correcto se obtiene lo que se muestra en la figura 2.

Figura 2

Pantalla de autenticación *IDU*



Nota: Adaptado de Iniciar sesión [Captura de pantalla], por IDU-UNAM, 2023, *IDU* (https://idu.unam.mx/sso/oauth2/unam/authorize?client_id=seguimiento-algo&redirect_uri=https%3A%2F%2Fseguimientotitulacionpre.unam.mx%2Flogin%2Fidu%2Fcontrol&scope=openid+profile+email&response_type=code)

² *HTTP* es un protocolo de comunicación que permite las transferencias de información a través de archivos.

³ *GET* es un método *HTTP* que se utiliza para solicitar datos al servidor web. El método *GET* envía los datos a través de la *URL*, lo que significa que se pueden ver en la barra de direcciones del navegador.

Cabe aclarar que todos los parámetros deben estar registrados previamente, si alguno de ellos no se encuentra registrado se obtendría el mensaje de error `"invalid_client Client authentication failed"` que se muestra en la figura 3.

Figura 3

Pantalla de error al enviar un dato diferente a los capturados por IDU



Nota: Adaptado de Iniciar sesión [Captura de pantalla], por IDU-UNAM, 2023, IDU (https://idu.unam.mx/sso/oauth2/unam/authorize?client_id=seguimiento-algo&redirect_uri=https%3A%2F%2Fseguimientotitulacionpre.unam.mx%2Flogin%2Fidu%2Falumno&scope=openid+profile+email&response_type=code)

En este punto el usuario captura sus datos de ingreso y si es la primera vez que lo hace, el servicio IDU notificará al usuario la información que quiere ver, permitiendo que acepte o deniegue el acceso a su información, como se muestra en la figura 4.

Figura 4

Pantalla donde se muestra la autorización y concesión de los datos.



Nota: Adaptado de Iniciar sesión [Captura de pantalla], por IDU-UNAM, 2023, IDU (https://idu.unam.mx/sso/oauth2/unam/authorize?client_id=seguimiento-algo&redirect_uri=https%3A%2F%2Fseguimientotitulacionpre.unam.mx%2Flogin%2Fidu%2Falumno&scope=openid+profile+email&response_type=code)

Si el usuario previamente lo ha aprobado, se podría omitir este paso y enviar directamente la concesión a la URL dada para la redirección y un parámetro llamado "code". Un ejemplo del resultado es el siguiente:

http://localhost:5000/callback?code=MSEzR6ErbpglHhZJuTE25fNk6ho&iss=http://idu.unam.mx:8081/sso/oauth2/unam&client_id=sistema-firma-documentos

Para este punto ya se ha logrado la primera parte del flujo del protocolo para la concesión de autorización, ahora se solicita el token de acceso (access token) lo que demuestra que es un cliente válido y que se puede obtener la información a la que el usuario dio acceso.

Para obtener este token de acceso se realiza una petición POST⁴ al servidor con los siguientes datos:

URL: https://idu.unam.mx/sso/oauth2/unam/access_token

Cabeceras (headers):

Content-Type: application/x-www-form-urlencoded

Authorization: Basic c2lzdGVtYS1hY3Rhcy10aXR1...

Cuerpo (body):

grant_type: authorization_code

code:MSEzR6ErbpglHhZJuTE25fNk6ho

redirect_uri: <http://localhost:5000/callback>

Donde:

content-type: define que se están enviando los datos desde un formulario.

Autorization: es el tipo de autorización que define el servidor, en este caso está compuesta de la palabra "Basic" seguido de una cadena *base64* donde incluye el identificador del cliente, dos puntos y la contraseña del cliente "id_cliente:client_secret".

Grant_type: cadena de texto "authorization_code" que especifica el avance en el flujo de autorización.

Code: código que se recibe de la petición anterior, que solo se puede utilizar una vez para obtener el token.

Redirect_uri: esta es la URL donde se va a enviar la información, que debe coincidir con la que se envió en el proceso de autorización.

Ejemplo:

https://idu.unam.mx/sso/oauth2/unam/access_token

Cabeceras:

Authorization: Basic c2lzdGVtYS1hY3Rhcy10aXR1...

Content-Type: application/x-www-form-urlencoded

⁴ POST es un método que utiliza HTTP para enviar datos: los transmite en el cuerpo de la petición.

Cuerpo:

```
grant_type:authorization_code  
code:MSEzR6ErbpglHhZJuTE25fNk6ho  
redirect_uri:http://localhost:5000/callback
```

Lo anterior debería devolver como resultado un objeto json⁵ como se muestra en la figura 5, con la información del token de acceso, los permisos (*scopes*) del *token*, el tipo de *token* y el tiempo de validez del *token* (*expires_in*).

Figura 5

Respuesta recibida al solicitar el token de acceso (access token)



```
"access_token": "jq_bzfsyE8iVnlyq7DyB6ajHz7c",  
"scope": "openid profile email",  
"id_token": "eyJ0eXAiOiJKV1QiLCJraWQiOiJ3VTNpZk1JYUxPVUFSZVJC  
eyJhdF9oYXNoIjoieYjoiYUxPVUFSZVJCeyJhdF9oYXNoIjoieYjoiYUxPVUFSZVJC  
ZmZGZi0WRhYjZkLTc2NDU4Iiwic3VibmFtZSI6IjY4YzRkNWVhLTM5NjY  
c2lkIjoieEJwZWdUZW5ZZDZpMnVPR3FMYkRUZTAwNHFcTJLNmI4dFNOT  
JvY2sub3B1bm1kY29ubmVjdC5vcHMiOiJFT0hLMmQ4UWZMSDEtMm15SFZ  
NDEsInRva2VuVHlwZSI6IkpXVFRva2VuIiwiaWF0IjoxNjY3NTg5NDQxf  
SqUXBqBlizatn_UZwtYFrPFBUFLx1r00ljIkiya0yk1L9-LLBxrFh9bSR  
VRWiUFzjsJPdHHMCrgSttR5VjHzzRSX3kDxxcX04aUkwla0Hf0pz4Tnsm  
"  
"token_type": "Bearer",  
"expires_in": 3599
```

Al llegar a este punto se puede considerar que la autorización ha sido completada.

Una vez que se cuenta con la autorización, se pueden solicitar los recursos al servidor, en este caso el recurso será la información del usuario. Para este caso, se realiza una petición *GET* a la dirección <https://idu.unam.mx/sso/oauth2/unam/userinfo>, para lo que se agrega la siguiente cabecera:

Authorization: Bearer jq_bzfsyE8iVnlyq7DyB6ajHz7c

A esta cabecera de autorización se le agrega el *token* que devolvió el servidor *IDU*, antecediendo la cadena "Bearer", lo que devuelve como respuesta la información del usuario en formato json, como se muestra en la figura 6. Se pueden realizar tantas consultas como se necesiten mientras el token esté activo.

⁵ *JSON* es una notación de objetos *JavaScript* por sus siglas en inglés, formato que sirve para el intercambio de datos basándose en la sintaxis de los objetos de *JavaScript*.

Figura 6

Respuesta que se recibe con los datos de la persona

```
{
  "firstName": "Juan",
  "lastName": "Pérez",
  "secondLastName": "García",
  "name": "Juan Pérez García",
  "curp": "GARCJ051010000000000000",
  "rfc": "GARCJ051010000000000000",
  "employeeNumber": "000000000000000000",
  "studentNumber": "000000000000000000",
  "email": "correo@email.com",
  "unamEmail": "correo@unam.unam.mx",
  "sub": "(user:f826fb10-4271-45b8-9429-4b8391b5c425)",
  "subname": "f826fb10-4271-45b8-9429-4b8391b5c425"
}
```

Para terminar la sesión e invalidar el token se tendrán que realizar las siguientes peticiones:

Una petición *POST* a la siguiente dirección:

https://idu.unam.mx/sso/oauth2/unam/token/revoke?token=jq_bzfsyE8iVnlyq7D yB6ajHz7c

Se enviará como parámetro el *token* de acceso (*access_token*) que se recibió cuando fue solicitado (figura 4) y se agrega la cabecera *Autorization* compuesta de la palabra *"Basic"* seguido de una cadena *base64* donde se incluye el identificador del cliente, dos puntos y la contraseña del cliente *"id_client:client_secret"*; esta petición no regresará alguna respuesta. Una vez realizada esta petición se invalidará el *token* y como último paso para terminar la sesión se envía una petición *GET* a la siguiente URL:

https://idu.unam.mx/sso/oauth2/unam/connect/endSession?id_token_hint=eyJ0eXAiOiJKV1Qi...&client_id=sistema-firma documentos&post_logout_redirect_uri=http://localhost:5000/end

Siendo *id_token_hint*: el identificador del *token* de acceso (*id_token*) que se recibió al inicio (figura 4), el *client_id* que corresponde al *id* del cliente y el *post_logout_redirect* la ruta a la cual se redireccionará después de terminar la sesión.

4. RESULTADOS

Se realizó la integración del servicio *IDU* con el Sistema de seguimiento de titulación universitaria por medio de dos librerías. La primera, llamada *"Guzzle"*, se recomienda para proyectos que no tengan algún marco de trabajo, ya que facilita el uso de las peticiones HTTP hacia *IDU*.

La segunda librería se llama *"Socialite"* y se integra con el marco de trabajo *Laravel*, lo que facilita el manejo de sesiones y de usuarios, propios de dicho marco de trabajo.

Se generó un manual de interoperabilidad revisado junto con la Coordinación de Servicios de Identidad y de Firma Electrónica Universitaria de la DGTIC, donde se detalla la implementación.

Adicionalmente, dado que el sistema de seguimiento ya tenía un módulo para registro de usuarios, asignación de permisos y autenticación, éste fue adaptado para recibir los datos de *IDU*, para ello se eliminaron datos como el nombre de usuario y contraseña, y se agregó el campo RFC que sirve como punto de conexión entre ambos sistemas.

Para el caso de los alumnos, al no contar con su información académica se implementó otra conexión al servicio web de la Dirección General de Administración Escolar para traer su trayectoria académica y verificar el cumplimiento de los requisitos para iniciar su trámite

5. CONCLUSIONES

El servicio *IDU* permite que un usuario proporcione su información con otras aplicaciones sin revelar toda su identidad, lo que mejora la privacidad y la seguridad de los usuarios al no tener que compartir sus credenciales con las aplicaciones, y se permite el acceso delegado y restringido a sus recursos, para controlar qué acciones puede realizar la aplicación en su nombre y poder revocar el acceso en cualquier momento.

IDU facilita la integración y la interoperabilidad entre las aplicaciones y servicios reduciendo la complejidad y el mantenimiento del sistema, y se encarga de la gestión de la información de los usuarios, la emisión de los tokens de acceso y la comunicación entre los componentes.

- Algunas de las áreas de oportunidad a considerar son:
- La implementación cuidadosa y un seguimiento de las buenas prácticas para evitar vulnerabilidades o ataques de seguridad, como el robo o la suplantación de tokens, la suplantación de identidad (*phishing*) o la falsificación de solicitud entre sitios *CSRF* (*cross-site request forgery*).
- La evaluación del nivel de dependencia de los servidores de autorización y recursos, que pueden tener fallos, retrasos o cambios en sus políticas o especificaciones lo que puede provocar que no puedan ingresar a los sistemas.
- El diseño de la interfaz de usuario para evitar que se genere confusión o desconfianza al mostrarle la información que se compartirá con la aplicación.

Al elaborar esta memoria técnica se busca dejar un antecedente para que las áreas universitarias que requieran información para obtener la autenticación y obtención de datos de *IDU* lo puedan desarrollar de manera sencilla.

6. AGRADECIMIENTOS

Se reconoce y agradece la colaboración recibida del personal de la Coordinación de Servicios de Identidad y de la Firma Electrónica Universitaria y del equipo de aseguramiento de la calidad de la Dirección de Colaboración y Vinculación, ambas áreas pertenecientes a la DGTIC.

REFERENCIAS BIBLIOGRÁFICAS

Guzzle. (2021). *Guzzle Documentation*. Recuperado el 22 de septiembre de 2023, de <https://docs.guzzlephp.org/en/stable/index.html>

OAuth 2.0 — OAuth. (2012). *OAuth.net*. Recuperado el 12 de septiembre de 2023, de <https://oauth.net/2/>

Examen diagnóstico como requisito de ingreso en educación continua en 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:

Fernández Grajales, N.L.M. (2023). Examen diagnóstico como requisito de ingreso en educación continua en TIC. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (104 - 110).

<https://doi.org/10.22201/dgtic.ctud.2023.1.1.10>

Nubia Lizbeth Marina Fernández Grajales

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

nulizma@comunidad.unam.mx

ORCID: 0000-0001-6547-1899

Resumen:

¿Cómo seleccionar a los aspirantes que desean participar en los eventos de educación continua en el ámbito de las tecnologías de la información y la comunicación (TIC) ofrecidos por la UNAM? En este reporte se presenta el diseño e implementación de un examen de evaluación diagnóstica de conocimientos previos, que se ha aplicado como requisito de ingreso desde el año 2020. El examen tiene el propósito de medir el nivel de competencia de los aspirantes en los temas y subtemas relacionados con cada evento, así como brindar retroalimentación durante el proceso de ingreso. Para diseñar el examen, se adaptó una metodología basada en una guía de orientación y un conjunto de instrumentos que contemplan aspectos como el perfil de referencia, la tabla de especificaciones, la construcción y validación de los reactivos, la aplicación y el análisis del examen piloto y definitivo. El examen se aplica mediante una plataforma en línea, Moodle, que facilita el acceso, la administración y la calificación del mismo. Los resultados obtenidos muestran que el examen cumple con los criterios de objetividad, validez y confiabilidad, y que se adapta al perfil de ingreso de los aspirantes a los eventos de educación continua en TIC. También identifica algunas limitaciones que pueden afectar la calidad y la pertinencia del examen, y propone algunas recomendaciones para superarlas. El examen de evaluación diagnóstica de conocimientos previos es una herramienta útil y eficaz para mejorar el proceso de selección y formación de los asistentes a los eventos de educación continua en TIC.

Palabras clave:

Examen objetivo, tabla de especificaciones, banco de reactivos.

1. INTRODUCCIÓN

En el año 2020, por indicaciones de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación se inició la integración de comités consultivos para el proceso de creación de nuevos diplomados, conformados por expertos en la materia. Se decidió enfocarse en un requisito de ingreso: la generación de un examen de evaluación diagnóstica de conocimientos previos, para fortalecer lo señalado en el capítulo VI, artículo 23, fracción IV del Reglamento General de Educación Continua de la UNAM, aprobado el 18 de marzo de 2016, sobre el programa de actividades en educación continua, que contempla la evaluación del participante. (Red de Educación Continua [REDEC], 2016, p. 22).

2. OBJETIVO

Adoptar una metodología para el diseño de un examen de evaluación diagnóstica de conocimientos previos, como requisito de ingreso para los aspirantes que desean participar como asistentes en los eventos de educación continua en TIC, que contemple instrumentos y bancos de preguntas idóneos para la adecuada selección de los aspirantes.

3. DESARROLLO

3.1 ANTECEDENTES

En el marco de los programas y los proyectos descritos en el Plan de Desarrollo Institucional 2019-2023 de la Universidad, en su proyecto estratégico 2.4.1 el Dr. Enrique Graue Wiechers (2020) expone que se debe “incrementar y diversificar la oferta de programas, cursos y actividades de educación continua de la UNAM”, (p. 16).

Por otra parte, en el Reglamento General de Educación Continua de la UNAM se señala que la educación continua “tiene como finalidad complementar la formación curricular, profundizar y ampliar conocimientos en todos los campos del saber, capacitar y actualizar profesionalmente para contribuir al bienestar y desarrollo individual y social, bajo los criterios de calidad y pertinencia distintivos de la Institución” (REDEC, 2016, p. 19).

La Dirección de Docencia en TIC (DDTIC) de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC), tiene el objetivo de “coordinar las acciones y mecanismos, mediante los cuales se desarrollen y promuevan los servicios de formación, capacitación y actualización en el área de las TIC, tanto en la modalidad presencial como semipresencial y en línea, como parte de la educación continua”. (DGTIC, 2022, p. 31).

3.2 PROBLEMAS A RESOLVER

¿Cómo se puede diseñar e implementar un examen de evaluación diagnóstica de conocimientos previos en el ámbito de las TIC, como requisito de ingreso para los aspirantes que desean participar como asistentes a los eventos de educación continua ofrecidos por la DGTIC, donde se utilice una metodología para la elaboración y aplicación de examen objetivo (Burruezo Ordoñez et al., 2014) y se considere el tipo de conocimiento y procesos cognitivos (Kriscautzky *et al.*, 2021), que contemple instrumentos válidos y confiables, y bancos de preguntas alineados con los contenidos y las competencias de los eventos académicos (Downing y Haladyna, 2006), para la adecuada selección y orientación de los aspirantes?

3.3 FUNDAMENTOS TEÓRICOS

En el material del curso Evaluación del y para el aprendizaje: enfoque cuantitativo del programa especializado en evaluación educativa, Martínez y Herrera (2020) definen como examen objetivo a “un instrumento de evaluación que se presenta en forma escrita y está integrado por una variedad de reactivos; se le denomina objetivo debido a que solamente existe una respuesta correcta para cada uno de los reactivos que lo integran y el examinado debe elegirla de entre un conjunto de opciones, lo cual impide que el juicio o las opiniones del evaluador interfieran al momento de analizar los resultados” (p. 55). Para identificar los conocimientos esperados asociados con los temas y subtemas identificados en el examen objetivo, la taxonomía de aprendizaje propuesta por Anderson y sus colaboradores expone la dimensión de los conocimientos, como son: Factual¹, Conceptual² y Procedimental³, así como la dimensión de los procesos cognitivos involucrados: Recordar⁴, Comprender⁵, Aplicar⁶, Analizar⁷, Evaluar⁸ y Crear⁹.

3.4 METODOLOGÍA APLICADA

Martínez, Herrera, *et al.* (2022), proponen una metodología de 3 fases: Planeación, Aplicación y Análisis de resultados. En la primera se genera el perfil de referencia y la tabla de especificaciones, se construyen los reactivos y se verifica su validez. Posteriormente, se integra un examen piloto y se hacen los ajustes necesarios antes de su aplicación.

El perfil de referencia contiene: tipo y propósito del examen, temas y subtemas a evaluar y resultados de aprendizaje esperados, mientras que la tabla de especificaciones contiene, adicionalmente, la ponderación y el número de reactivos.

3.5 PROPUESTAS DE SOLUCIÓN

Se adaptó, aplicó y mejoró un procedimiento, considerando las prácticas alineadas a las metodologías vigentes en la Universidad Nacional Autónoma de México, para la creación de un examen de evaluación diagnóstica de conocimientos previos, en colaboración con la Coordinación Académica de los Diplomados.

¹ Los elementos básicos que los estudiantes deben conocer para estar familiarizados con una disciplina o resolver problemas con ellas.

² Las interrelaciones entre los elementos básicos con una estructura mayor que les permita funcionar en conjunto.

³ Cómo hacer algo: metodología o investigación, así como los criterios para utilizar habilidades, algoritmos, técnicas y métodos.

⁴ Recuperar conocimiento relevante de la memoria a largo plazo

⁵ Determinar el significado de los mensajes institucionales, incluyendo comunicación oral, escrita y gráfica

⁶ Llevar a cabo o usar un procedimiento en una situación dada

⁷ Separar el material en sus partes constituyentes y detectar cómo esas partes se relacionan unas con otras y con una estructura o propósito general.

⁸ Realizar juicios basados en criterios y estándares.

⁹ Colocar elementos juntos para confirmar un todo coherente y novedoso o elaborar un producto original.

3.6 PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS

El procedimiento aplicado consta de siete actividades secuenciales. Cada una es realizada por cuatro roles. La figura 1 muestra las actividades para la planeación, la aplicación y el análisis de resultados del examen de evaluación diagnóstica de conocimientos previos.

Adicionalmente, se diseñaron los instrumentos para la tabla de especificaciones y la estructura del examen.

Figura 1

Diagrama de actividades de la evaluación diagnóstica de conocimientos previos



4. RESULTADOS

Tras aplicar la metodología definida, los hallazgos técnicos son los siguientes: se diseñó un examen de evaluación diagnóstica de conocimientos previos, mediante una guía de orientación y un conjunto de instrumentos, que cumple con los criterios de objetividad, validez y confiabilidad, y que se adapta a las características y necesidades de los aspirantes a los eventos de educación continua en TIC. El examen se diseñó con el propósito de evaluar los conocimientos previos de los aspirantes a los diplomados Ciencia

de datos e Inteligencia artificial aplicada, así como al curso Introducción a la Computación Cuántica, utilizando una plataforma en línea, *Moodle*, que facilitó el acceso, la administración y la calificación del examen. Se obtuvo un banco de preguntas de opción múltiple con una sola respuesta correcta, con diferentes niveles de dificultad y complejidad, que abarcan los temas y subtemas relevantes para cada evento, según la tabla de especificaciones que se elaboró previamente y que permitió medir los conocimientos esperados.

La información que se utilizó para diseñar el examen, se obtuvo de fuentes confiables y actualizadas como libros, artículos, sitios web, cursos en línea relacionados y de la experiencia de expertos en la materia. La información se clasificó y organizó según los temas y subtemas a evaluar en la tabla de especificaciones, y se utilizó para construir los reactivos del examen. La validez de los reactivos se verificó mediante una revisión por parte de expertos en cada tema.

Los reactivos se integraron en un examen piloto que se aplicó a un grupo reducido de personas, y se hicieron los ajustes necesarios antes de su aplicación definitiva, entre los que se encuentran reactivos con más de una respuesta correcta, mal redactados, con errores ortográficos o gramaticales, que no estaban alineados con la tabla de especificaciones, con un nivel de dificultad o complejidad muy alto o muy bajo, que eran irrelevantes o se encontraban desactualizados, lo que ayudó a evitar la confusión y la frustración en los aspirantes, y propiciar la correcta discriminación y la credibilidad del examen.

El resultado del examen de evaluación diagnóstica de conocimientos previos tiene un impacto positivo tanto para los aspirantes como para las coordinaciones académicas de los eventos. Para los aspirantes, el examen les permite conocer su nivel de conocimientos previos sobre los temas que se abordarán en cada evento, así como identificar sus fortalezas y debilidades. Además, el examen les ayuda a decidir si están preparados para asistir al evento o si necesitan reforzar sus conocimientos mediante otras fuentes o programas académicos. Para las coordinaciones académicas, en colaboración con los comités académicos, el examen les permite seleccionar a los aspirantes que cumplan con el perfil requerido para cada evento, así como adaptar el contenido y la metodología de los eventos según el nivel de conocimientos previos de los asistentes. Asimismo, el examen les facilita el seguimiento y la evaluación de los conocimientos esperados. Para el equipo de soporte les permite ahorrar tiempo y recursos en el proceso de diseño, aplicación y análisis del examen de evaluación diagnóstica de conocimientos previos, gracias a la automatización y la optimización de las tareas y los instrumentos involucrados.

5. CONCLUSIONES

Se buscó diseñar un examen que permitiera evaluar los conocimientos previos de los aspirantes a diplomados y cursos de TIC, usando una metodología que incluyera instrumentos y preguntas adecuadas para seleccionarlos. Se obtuvo un diseño de examen de evaluación diagnóstica de conocimientos previos que cumple con los criterios de objetividad, validez y confiabilidad, y que se adapta al perfil de ingreso de los aspirantes a los eventos de educación continua en TIC; un banco de preguntas con diferentes niveles de dificultad y complejidad, que abarcan los temas y subtemas establecidos en la tabla de especificaciones, y que permiten medir los conocimientos esperados; la aplicación del examen a una muestra representativa de aspirantes mediante *Moodle*, lo que facilitó el acceso, la administración y la calificación del examen. Estos hallazgos tuvieron un alto grado de impacto, ya que cubrieron todos los aspectos del proceso de diseño, aplicación y análisis del examen, y se ajustaron a las especificaciones y expectativas definidas para cada evento.

Las limitaciones que pueden afectar la validez y la confiabilidad de los resultados se encuentran en la posibilidad de que los aspirantes respondan al azar y acierten a la respuesta correcta, lo cual no implica que posean el conocimiento requerido para participar en el evento. Esta situación puede generar una sobreestimación del nivel de competencia de los aspirantes y dificultar el proceso de selección. La imposibilidad de conocer el proceso cognitivo que siguen los aspirantes para llegar al resultado final impide identificar las fortalezas y debilidades de sus aprendizajes previos, así como las posibles causas de sus errores. Esta situación puede limitar la retroalimentación a los aspirantes.

Se recomienda revisar y corregir periódicamente los reactivos del examen; actualizar el banco de preguntas con base en los cambios o las tendencias en los contenidos y las competencias de los eventos de educación continua en TIC; aplicar nuevamente el examen piloto a un grupo reducido de personas, y verificar su validez y confiabilidad mediante técnicas estadísticas. Estas tareas son necesarias para mejorar la calidad y la pertinencia del examen, y evitar riesgos como la insatisfacción o el abandono por parte de los aspirantes, así como la pérdida de credibilidad o reputación por parte de la coordinación académica y el equipo de soporte. El nivel general de esfuerzo requerido para su implementación es moderado, ya que se trata de tareas que se pueden realizar en un plazo razonable, con los recursos disponibles y con la colaboración de los expertos en cada tema.

6. AGRADECIMIENTOS

Expreso mi más sincero agradecimiento a la Mtra. Cristina Múzquiz Fragoso, por su apoyo y orientación, así como a la Mtra. Patricia Garcés Natera, por su profesionalismo y disposición al logro de los objetivos planteados. Asimismo, expreso mi reconocimiento al trabajo y la dedicación de todos los integrantes del equipo. Sin su esfuerzo y compromiso, el examen de evaluación diagnóstica de conocimientos previos no hubiera sido posible.

REFERENCIAS BIBLIOGRÁFICAS

- Burruezo Ordóñez, M., Cortés de las Heras, J., Martínez Soriano, V y Moreno Agud, A. P. (2014). *Diseño de Pruebas Objetivas*. Cuadernos de instrumentos de evaluación #01.
- DGTIC. (2022). *Manual de organización Dirección General de Cómputo y de Tecnologías de Información y Comunicación, UNAM*. <https://presupuesto.unam.mx/organi/biblioteca/522012C.pdf>
- Downing, S. M. & Haladyna, T. M. (2006). *Handbook of test development*. <https://fatihegitim.files.wordpress.com/2014/03/hndb-t-devt.pdf>
- Graue, E. (2020). *Plan de Desarrollo Institucional 2019-2023, UNAM*. <https://www.rector.unam.mx/doctos/PDI2019-2023.pdf>
- Kriscautzky L., M., Martínez F., N. P., Ortega R., C. R., Ramírez B., A. M., Rodríguez A., G. (2021). *Guía digital para el uso de TIC y TAC por área de conocimiento*. (pp. DGTIC, UNAM. <http://educatic.unam.mx>
- Martínez, A. y Herrera, C. (2020). *Examen objetivo*. En M. Sánchez y A. Martínez (Ed.), *Evaluación del y para el Aprendizaje: Instrumentos y Estrategias* (pp. 55-74). Imagia Comunicación. https://cuaieed.unam.mx/descargas/investigacion/Evaluacion_del_y_para_el_aprendizaje.pdf

- Martínez, A., García, M. y Herrera, C. (2022) *Examen objetivo*. En M. Sánchez y A. Martínez (Ed.), *Evaluación y aprendizaje en educación universitaria: estrategias e instrumentos* (pp. 181-197). Imagia Comunicación. <https://cuaieed.unam.mx/publicaciones/libro-evaluacion/>
- REDEC. (2016). Reglamento General de Educación Continua de la UNAM, UNAM. http://abogadogeneral.unam.mx/legislacion/abogen/documento.html?doc_id=87

Diseño, limpieza y carga de datos para un sistema de consulta

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:

Corona Correa, S. L. (2023). Diseño, limpieza y carga de datos para un sistema de consulta. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (111 - 125).

<https://doi.org/10.22201/dgtic.ctud.2023.1.1.23>

Susana Laura Corona Correa

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

susanacc@unam.mx

ORCID: 0009-0003-0560-1745

Resumen:

La existencia de nuevas necesidades de consultas y reportes dio origen al desarrollo del Sistema de Consulta de Información de un cuerpo colegiado de la Universidad Nacional Autónoma de México, lo que implicó el diseño de su base de datos utilizando prácticas de metodologías ágiles. Se presentan las actividades realizadas en el análisis de requerimientos de los datos, el entendimiento de la problemática actual, el análisis exploratorio de la información, el diseño de la base de datos, la identificación y validación de los catálogos, así como el proceso de limpieza de datos, carga y verificación de la información inicial para su operación. El resultado principal fue contar con un nuevo diseño de base de datos, poblado con datos revisados, que cumpliera con las necesidades del cuerpo colegiado para facilitar su consulta y aprovechamiento.

Palabras clave:

Limpieza de datos, carga de datos, diseño de bases de datos.

1. INTRODUCCIÓN

Se solicitó a la Dirección General de Cómputo y de Tecnologías de Información y Comunicación realizar un proyecto conjunto con un cuerpo colegiado de la Universidad Nacional Autónoma de México para actualizar su sistema para consultar datos de sus expedientes y miembros. Para ello se realizó el diseño de una nueva base de datos y se hizo la limpieza y carga de los datos relevantes de los expedientes a consultar, tales como: fecha, lugar, tipo de proceso atendido, participantes de un proceso, persona designada, asistencia, entre otros.

Asimismo, mediante un proceso de análisis se identificaron oportunidades de mejora en la calidad de los datos¹ de la información histórica del registro de los expedientes de este cuerpo colegiado.

2. OBJETIVO

Mejorar la calidad de los datos de los expedientes y sus procesos, así como de los datos históricos, a través del diseño de la base de datos para cumplir con las necesidades establecidas por el usuario sobre la consulta, registro y generación de reportes de los expedientes de las sesiones de un cuerpo colegiado de la Universidad Nacional Autónoma de México.

3. ANTECEDENTES

La documentación de las sesiones de un cuerpo colegiado de la Universidad Nacional Autónoma de México (UNAM) se concentra en expedientes generados a partir de su instauración hasta la fecha, lo cual se ha hecho en carpetas físicas estructuradas en uno o varios tomos. Los expedientes contienen principalmente el orden del día, la lista de asistencia, el acta de la sesión y sus anexos.

Se digitalizaron los expedientes históricos a partir de una iniciativa previa con la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC), con lo que la oficina de este cuerpo colegiado incorporó la digitalización de su contenido como un proceso al finalizar la integración de cada uno de ellos.

El control de los datos relevantes de los expedientes de las sesiones ordinarias y extraordinarias de este cuerpo colegiado se ha realizado por medio de su registro en una hoja de cálculo; posteriormente este archivo era importado de forma periódica por personal técnico en la base de datos del sistema anterior.

Debido a que los usuarios del sistema anterior no contaban con la posibilidad de agregar su propia información, los datos que podían consultar estaban incompletos y desactualizados, lo que ha dificultado su uso de forma oportuna.

Adicionalmente, se han requerido búsquedas y salidas de información para cubrir peticiones de información, debido a que los filtros del sistema anterior sólo permiten buscar por número de expediente, año, tipo de sesión, proceso, cargo y entidad.

1 La calidad de los datos se refiere al grado en el cual los datos satisfacen “los requisitos definidos por la organización a la que pertenece el producto” de acuerdo con la norma ISO 25012:2008 (ISO 25000, s.f.)

Adicionalmente, la información obtenida en los listados no cumplía con el formato, calidad y datos necesarios para resolver las consultas y reportes solicitados a este cuerpo colegiado, debido a que no permitía su exportación en archivos de hoja de cálculo; únicamente se mostraba el resultado vía web, por lo que el usuario del sistema no podía usarlo fácilmente para generar sus reportes.

Otras oportunidades de mejora del diseño de la base de datos anterior surgen a partir de la falta de catálogos, ya que se tenían datos sin normalizar y los datos importados de hojas de cálculo con los expedientes presentaban diversos errores, tales como: datos incompletos, errores ortográficos, valores incorrectos, entre otros.

Lo anterior ha ocasionado que los usuarios del sistema usaran su archivo de hoja de cálculo y la carpeta física del expediente para poder dar respuesta a las solicitudes de información y reportes, debido a que el sistema no cubría sus necesidades.

Por lo anterior, se tomó la decisión de desarrollar una aplicación y diseño de la base de datos nuevos, para lo cual se acordaron diversas actividades con la oficina del cuerpo colegiado para alcanzar los objetivos y considerar filtros adicionales.

4. PROCESO REALIZADO PARA EL DISEÑO DE LA BASE DE DATOS, LA LIMPIEZA Y CARGA DE DATOS

4.1 ANÁLISIS DE REQUERIMIENTOS Y ENTENDIMIENTO DE LA PROBLEMÁTICA ACTUAL

En esta etapa hubo sesiones de trabajo acerca de los datos de los expedientes para entender las necesidades de información del área, la problemática para la consulta y generación de reportes, así como el uso de la información y los objetivos a alcanzar respecto de los datos.

Estas sesiones permitieron comprender la estructura de los datos, el flujo de información que sigue el usuario desde su registro hasta su consulta y la periodicidad con la que se realizan estas actividades. De esta manera se contribuyó a definir mejoras en el diseño de la base de datos para el nuevo sistema, así como para definir validaciones y consideraciones en la limpieza de su información.

Durante todo el proyecto se incorporaron prácticas de metodologías ágiles como la generación de un *backlog* o lista detallada de actividades, participación en reuniones diarias (*dailys*), retrospectivas de los periodos de trabajo (*sprints*) para identificar mejoras en cada ciclo de desarrollo, entre otras.

4.2 ANÁLISIS EXPLORATORIO DE LA INFORMACIÓN

En esta etapa se revisó el archivo de hoja de cálculo que contenía 1,200 filas con los datos de expedientes y procesos atendidos por el cuerpo colegiado, desde su instalación. Dado que el proceso se ejecutaría sólo una vez antes de iniciar la operación del sistema, se eligió crear fórmulas directamente en una copia del archivo de hoja de cálculo del usuario del sistema para generar comandos INSERT con el fin de agregar los datos en una tabla en la base de datos con la finalidad de facilitar su análisis como se muestra en el Anexo A. Las tablas usadas para el análisis fueron eliminadas al final del proceso de revisión.

En este caso se utilizó Microsoft Excel para generar las fórmulas y PgAdmin como cliente para crear las tablas en donde se va a analizar la información y ejecutar los comandos INSERT; se muestra un ejemplo en el Anexo B.

Tras la creación y población de la tabla para el análisis de datos, se revisaron duplicados y la longitud máxima de los datos, se identificaron los valores que podían ser homogeneizados en catálogos, valores que no tenían una correspondencia exacta con los nombres de los miembros del cuerpo colegiado, fechas que no tenían un formato adecuado, datos inconsistentes de acuerdo con el contexto, entre otros; por ejemplo, se ubicaron campos vacíos que debían contener algún valor, fechas inconsistentes, etc. En el Anexo C se muestra un ejemplo de la consulta utilizada en este paso del proceso.

Los resultados de este análisis se comunicaron a la oficina del cuerpo colegiado y se acordaron los cambios que se aplicaron en DGTIC, tras considerar los catálogos validados y las indicaciones obtenidas del usuario del sistema. Por su parte la oficina realizó la revisión de los datos contrastando la información faltante o incorrecta con los expedientes físicos.

4.3 IDENTIFICACIÓN Y VALIDACIÓN DE CATÁLOGOS

Los datos históricos contenían algunos datos sucios, es decir, datos que “tienen una estructura incorrecta, numerosas imprecisiones o simplemente están incompletos” (Tableau, s.f.), los cuales pudieron ser generados por errores en la captura, cambio de requerimientos, cambio de políticas para almacenar los datos, entre otros.

Como resultado del análisis exploratorio se identificó que la información era de captura libre, sin considerar catálogos o validaciones, lo cual dificultaba la generación de reportes por medio de filtros, por ejemplo, tenía diferentes direcciones para la misma ubicación como se puede ver en la figura 1.

Figura 1

Ejemplo de nombres de lugar diferentes para la misma ubicación

OFICINAS DE LA RECTORÍA DE LA UNAM
OFICINAS DE RECTORÍA
OFICINAS DE RECTORIA
RECTORÍA DE LA UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

En esta etapa se identificaron los datos que eran candidatos a ser catálogos, los cuales se revisaron, modificaron y validaron con la ayuda y experiencia de los usuarios del sistema.

Se elaboró la equivalencia que tenía el dato original en el archivo de hoja de cálculo con el valor final en la base de datos aprobado por la oficina del cuerpo colegiado, con la finalidad de homologar los datos y facilitar la generación de estadísticas y reportes. Un ejemplo de esto se puede ver en la figura 2.

Figura 2

Ejemplo de correspondencia de nombres de lugar diferentes y la ubicación aprobada por el usuario

OFICINAS DE LA RECTORÍA DE LA UNAM	OFICINAS DE LA RECTORÍA DE LA UNAM
OFICINAS DE RECTORÍA	OFICINAS DE LA RECTORÍA DE LA UNAM
OFICINAS DE RECTORIA	OFICINAS DE LA RECTORÍA DE LA UNAM
RECTORÍA DE LA UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO	OFICINAS DE LA RECTORÍA DE LA UNAM

Se insertaron los registros del catálogo de lugares con los valores validados usando el cliente de bases de datos, así como los datos equivalentes entre el valor actual del archivo de hoja de cálculo y el valor revisado en una tabla creada en la base de datos para tal fin, a través del cliente PGAdmin.

Tanto las equivalencias de los lugares como el catálogo final fueron utilizados en la consulta para agregar los datos de las sesiones con los lugares en los que se llevaron a cabo, usando consultas de tipo INSERT as SELECT en scripts de SQL; un ejemplo de esta consulta se puede ver en el Anexo D.

Del mismo modo se identificaron 12 catálogos más, mediante consultas para obtener los valores distintos de los datos capturados de los expedientes y en tres de ellos con las equivalencias revisadas manualmente y validadas por el usuario del sistema. Estos catálogos tienen en su mayoría menos de 10 registros, a excepción del catálogo de entidades con 96 registros.

Los scripts mencionados se generaron y ejecutaron con el cliente PGAdmin:

- 10 *scripts* de SQL para crear las tablas usadas para almacenar temporalmente los datos obtenidos de los archivos de hoja de cálculo de expedientes, notas, personas, miembros, comunicados, entre otros, y poblarlas.
- 3 *script* con los valores equivalentes de los catálogos.
- 17 *scripts* de SQL para poblar las tablas principales del sistema como expedientes, procesos, personas, miembros del cuerpo colegiado, entre otras.

Con base en este contexto se realizó un nuevo diseño de bases de datos para el sistema de consulta, así como un proceso de limpieza y carga de los registros.

4.4 DISEÑO DE LA BASE DE DATOS

El diseño de la base de datos consistió en la identificación de las tablas necesarias para almacenar los datos de los expedientes, sus procesos, personas relacionadas con los procesos, miembros del cuerpo colegiado, sesiones y registros de asistencia, así como los catálogos y otras tablas necesarias para el funcionamiento del sistema.

Posteriormente se determinaron las relaciones entre las tablas identificadas y se consideraron los requerimientos de los usuarios del sistema acerca del registro de la información, consultas y reportes; se

consideró también el conocimiento obtenido de las características de los datos en el análisis exploratorio y la aplicación de buenas prácticas de diseño de bases de datos, por ejemplo:

- Todas las tablas del diseño de la base de datos tienen llave primaria².
- La integridad relacional se fortaleció usando constraints o restricciones de llave foránea³.
- Identificación de los datos obligatorios y establecimiento de restricciones de tipo NOT NULL.
- Identificación de los valores por defecto necesarios.
- Uso de nombres cortos y descriptivos para las tablas y campos.
- Uso de la nomenclatura elegida por el equipo de desarrollo en el nombrado de los objetos de la base de datos consistente en usar nombres cortos, sin usar ñ ni acentos, nombre de la tabla en singular, llave primaria usando id_nombre_tabla para identificar el registro; lo anterior se muestra en la figura 3.

Figura 3

Ejemplo de la estructura de la tabla lugar y la nomenclatura usada en los nombres del objeto y campos



- Incorporación de la descripción de cada campo y tabla como parte de su estructura usando el comando COMMENT de postgresQL.
- Normalización⁴ de la información que lo requería.
- Identificación de las características de los datos para establecer tipos de datos adecuados al contenido; algunos casos fueron limpiados antes de poder ser agregados, debido a que no tenían valores adecuados al formato o contenido esperados, por ejemplo fechas erróneas, tales como "19 DE ABRIL AL 12 DE NOVIEMBRE DE 1999", lo que no corresponde a una fecha sino a un periodo.
- Documentación del diseño en un diagrama entidad relación, el cual se incorporó a la documentación técnica.

2 En términos generales, la llave primaria (PK) es un atributo (o identificación de atributos) que de manera única identifica a cualquier renglón dado (Coronel et. al., 2011).

3 Una llave foránea (FK) es un atributo cuyos valores corresponden con los valores de la llave primaria de la tabla relacionada (Coronel et. al., 2011).

4 La normalización es un proceso para evaluar y corregir estructuras de tablas a fin de minimizar redundancias de datos, con lo cual se reduce la probabilidad de anomalías de datos (Coronel et. al., 2011).

Se hizo la actualización de la última versión del diagrama y la generación de su diccionario de datos a partir de la estructura de la base de datos. Este proceso, conocido como ingeniería en reversa, incluye el uso de algunas herramientas como DBeaver, Navicat, MySQL Workbench, entre otros. En la herramienta se indican los datos de conexión a la base de datos para generar el diagrama entidad relación o el diccionario de datos a partir de los objetos de la base de datos a la cual se conectó. La obtención del diccionario de datos también se puede realizar por medio de consultas, como se muestra en Anexo E.

Se obtuvo la retroalimentación del equipo de trabajo acerca del diseño de la base de datos, y se incorporaron las mejoras detectadas por el equipo, por ejemplo: relacionar directamente los estudios académicos con los datos generales de la persona, agregar el campo que permite relacionar un miembro con su antecesor para poder generar el reporte cronológico solicitado por el usuario del sistema, entre otros. Finalmente se generó el *script* de SQL para la creación de los objetos como tablas, secuencias y vistas.

4.5 REVISIÓN Y VALIDACIÓN DE DATOS DE PERSONAS

Durante el análisis exploratorio se identificó que había oportunidades de mejora en los nombres de las personas que se encontraban en los datos de los expedientes con diferentes roles; por ejemplo, en algunos se tenía el nombre completo, en otros casos se presentaba el nombre con abreviaturas o bien con errores de ortografía. Con la ayuda de la oficina del cuerpo colegiado se determinó cuál era el valor correcto que debía estar en el registro del expediente en el sistema, y con la equivalencia indicada se corrigieron los datos.

4.6 PROCESO DE CARGA

Se hizo una relación de los datos que proporcionó el usuario con la tabla y campo en la base de datos, se estableció el orden para ingresar los datos, se identificaron las correcciones a realizar por medio de scripts de SQL, se realizaron pruebas del proceso de carga y limpieza de datos, y finalmente se ejecutó el proceso en el ambiente de producción usando el comando `psql` para ejecutar los scripts.

4.7 VERIFICACIÓN DE LOS DATOS

Durante todo el proceso se realizaron revisiones de los datos almacenados en las tablas para verificar:

- Los valores válidos de acuerdo con el formato y características establecidas,
- el número de registros respecto de la información proporcionada por el usuario,
- los valores de texto libre modificados por catálogos aprobados.

Adicionalmente a la revisión de los datos por medio de consultas, el proceso de ingeniería de software incluyó pruebas funcionales; durante este proceso los probadores reportaron incidencias relacionadas con los datos, las cuales fueron analizadas y atendidas. Respecto a las incidencias reportadas se determinó la acción o acciones a seguir, tales como:

- El dato requería alguna modificación o corrección,

- la aplicación tenía que ser actualizada por el desarrollador del módulo reportado para que funcionara adecuadamente con los datos ingresados, o bien,
- era necesaria más información para confirmar los valores adecuados en la base de datos.

5. RESULTADOS

Como resultado del proceso de diseño, análisis, limpieza y carga se logró mejorar la calidad de los datos del registro de los expedientes y los datos históricos de los miembros del cuerpo colegiado.

El nuevo diseño de la base de datos está compuesto por 48 tablas. Las tablas principales almacenan los datos generales de los expedientes. Se agregaron los datos de 982 expedientes, 1,203 procesos y la asistencia a 1,195 sesiones. Se verificó, como indicador de éxito de la carga de cada tabla, que la cantidad de registros de la fuente de datos fuera la misma que el número de registros en el nuevo modelo; por ejemplo, si en el archivo de hoja de cálculo se tenían 982 expedientes, esa misma cantidad tenía que estar en la base de datos final. Otro indicador verificado fue que el número de registros en la tabla final en producción debía corresponder con un catálogo y tener la misma cantidad de registros que los valores iniciales sin catálogo.

Figura 4

Ejemplo de datos de expediente con valores en entidad, comparado con la tabla final proceso con valores del catálogo de entidades

```
SELECT count(*) FROM proceso where id_entidad is not null;  
-- 1083  
SELECT count(*) FROM z_expediente_nuevo where trim(entidad) !='';  
-- 1083
```

De acuerdo con los valores verificados, se homologaron y limpiaron los nombres de 1,686 personas relacionadas con el proceso en uno o más roles. Por ejemplo, si el primer apellido de una misma persona estaba escrito de formas distintas, la oficina del cuerpo colegiado revisaba manualmente los datos para indicar cuál era el valor adecuado. Con este dato verificado se limpiaron los datos relacionados con esa persona mediante *scripts* de SQL.

Se identificaron 14 catálogos, entre ellos: lugares, tipos de proceso y entidades. Con apoyo de estos catálogos se asignaron los valores correspondientes para sustituir y estandarizar los datos capturados de forma libre en la base de datos final.

6. CONCLUSIONES

Ante la problemática inicial se decidió realizar un nuevo diseño de bases de datos que permitiera realizar las consultas y reportes que le son solicitados al cuerpo colegiado, y llevar a cabo los procesos necesarios para contar con datos poblados de acuerdo con la nueva estructura y con criterios de calidad de datos.

Las mejoras implementadas en el diseño de la base de datos contribuyeron a fortalecer aspectos de calidad de los datos tales como la precisión⁵, al detectar y corregir datos inconsistentes con los valores reales o que no tuvieran el formato adecuado.

El proceso de carga se fortaleció usando algunas prácticas de limpieza de datos, con el fin de mejorar aspectos como la consistencia de la información, el uso de valores correctos mediante la implementación de catálogos, la completitud al evitar la omisión de valores obligatorios, entre otros.

El nuevo diseño y los procesos de limpieza de datos ejecutados en procesos de migración o en carga de datos contribuyeron a que la información obtenida por los usuarios del sistema para hacer consultas, generar reportes y estadísticas se genere a partir de datos más precisos. Por ejemplo, para obtener estadísticas acerca de en cuántas sesiones había participado un miembro del cuerpo colegiado, y dado que sus nombres no estaban estandarizados en las listas de asistencia, se podía considerar como personas distintas a una misma persona. Estos datos fueron estandarizados mediante consultas en *scripts* de SQL a partir del catálogo de miembros y la equivalencia revisada con el usuario del sistema.

Es recomendable revisar los atributos de calidad de datos mencionados en el estándar ISO 25012:2008, identificar los datos relevantes en el proyecto y definir las mejoras necesarias de los datos para cumplir con los atributos de calidad. Adicionalmente, se sugiere incluir procesos de pruebas de los sistemas con los datos con la finalidad de confirmar que funcionan adecuadamente, o en su caso establecer las acciones correctivas necesarias.

Finalmente, respecto al diseño de bases de datos es recomendable establecer lineamientos generales y una nomenclatura para nombrar los objetos en conjunto con el equipo de desarrollo con la finalidad de estandarizar el diseño.

REFERENCIAS BIBLIOGRÁFICAS

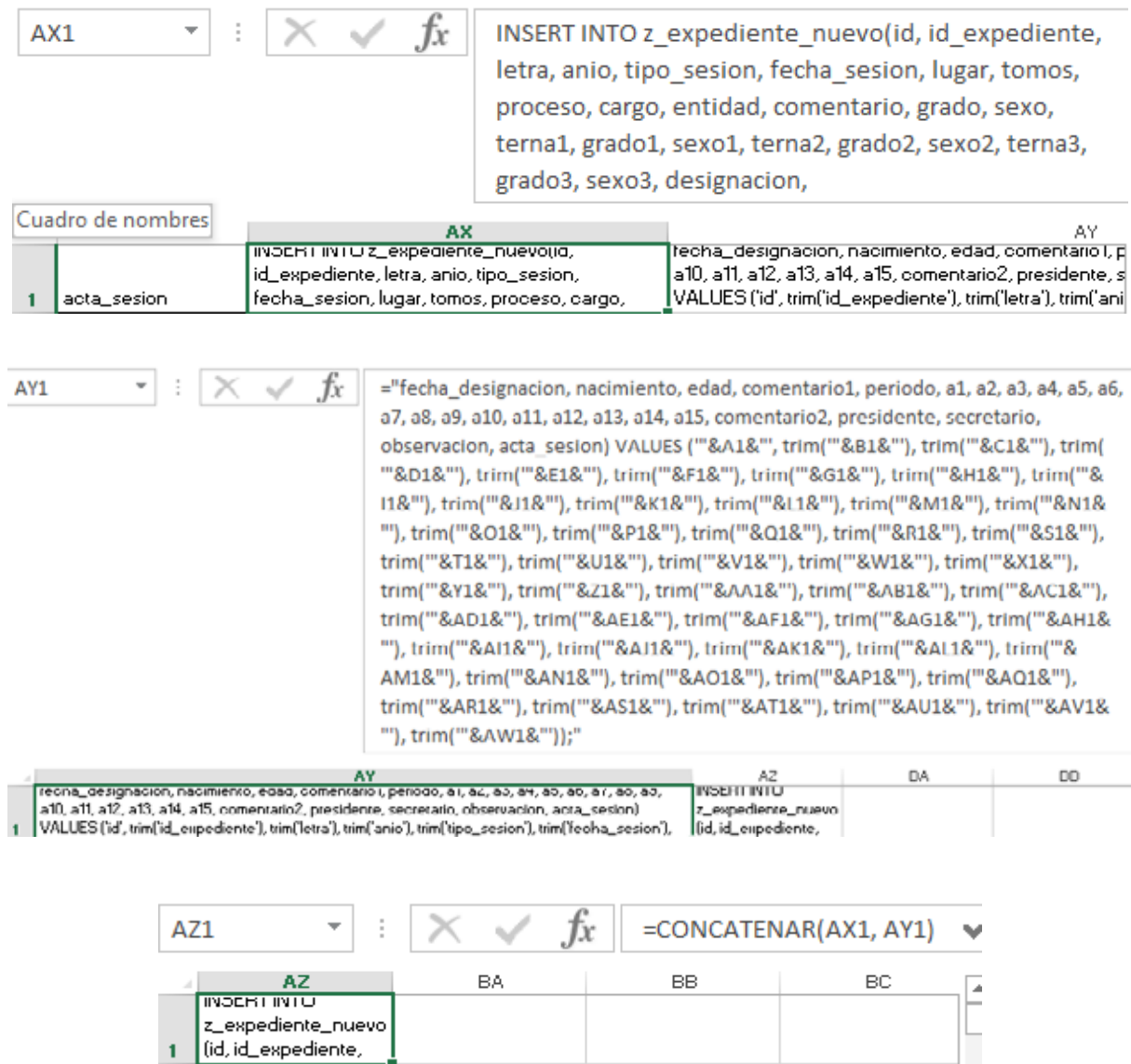
- Coronel, C. et. al. (2011). *Bases de datos. Diseño, implementación y administración*. México, Cengage Learning.
- González, M.F. (2013). Aplicación del estándar ISO/IEC 9126-3 en el modelo de datos conceptual entidad-relación. *Revista Facultad de Ingeniería, UPTC*, julio - diciembre de 2013, Vol. 22, No. 35. Revisado el 6 de septiembre de 2023. Obtenido de: <https://revistas.uptc.edu.co/index.php/ingenieria/article/view/2519/2374>
- ISO 25000 (s.f.). *ISO 25000 calidad de software y datos*. Recuperado el 6 de septiembre de 2023 de <https://iso25000.com/index.php/normas-iso-25000/iso-25012>
- Tableau (s.f.). *Los datos "sucios" tienen consecuencias: Cómo solucionar los problemas más comunes de preparación de datos*. Recuperado el 6 de septiembre de 2023 de: <https://www.tableau.com/es-es/learn/whitepapers/costs-of-dirty-data>

5 La precisión es el grado en el cual el dato tiene atributos que son exactos o que proporcionan la discriminación en un contexto específico de uso (González, 2013).

ANEXO A. EJEMPLO DE LAS FÓRMULAS DE EXCEL

Figura 5

Ejemplo de las fórmulas usadas en el archivo de Excel con los expedientes



The image shows three examples of Excel formulas used in a spreadsheet:

- Cell AX1:** The formula bar shows a partial formula: `INSERT INTO z_expediente_nuevo(id, id_expediente, letra, anio, tipo_sesion, fecha_sesion, lugar, tomos, proceso, cargo, entidad, comentario, grado, sexo, terna1, grado1, sexo1, terna2, grado2, sexo2, terna3, grado3, sexo3, designacion,`
- Cell AY1:** The formula bar shows a long concatenation formula: `= "fecha_designacion, nacimiento, edad, comentario1, periodo, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12, a13, a14, a15, comentario2, presidente, secretario, observacion, acta_sesion) VALUES (" & A1 & ", trim(" & B1 & "), trim(" & C1 & "), trim(" & D1 & "), trim(" & E1 & "), trim(" & F1 & "), trim(" & G1 & "), trim(" & H1 & "), trim(" & I1 & "), trim(" & J1 & "), trim(" & K1 & "), trim(" & L1 & "), trim(" & M1 & "), trim(" & N1 & "), trim(" & O1 & "), trim(" & P1 & "), trim(" & Q1 & "), trim(" & R1 & "), trim(" & S1 & "), trim(" & T1 & "), trim(" & U1 & "), trim(" & V1 & "), trim(" & W1 & "), trim(" & X1 & "), trim(" & Y1 & "), trim(" & Z1 & "), trim(" & AA1 & "), trim(" & AB1 & "), trim(" & AC1 & "), trim(" & AD1 & "), trim(" & AE1 & "), trim(" & AF1 & "), trim(" & AG1 & "), trim(" & AH1 & "), trim(" & AI1 & "), trim(" & AJ1 & "), trim(" & AK1 & "), trim(" & AL1 & "), trim(" & AM1 & "), trim(" & AN1 & "), trim(" & AO1 & "), trim(" & AP1 & "), trim(" & AQ1 & "), trim(" & AR1 & "), trim(" & AS1 & "), trim(" & AT1 & "), trim(" & AU1 & "), trim(" & AV1 & "), trim(" & AW1 & ");"`
- Cell AZ1:** The formula bar shows the formula: `=CONCATENAR(AX1, AY1)`

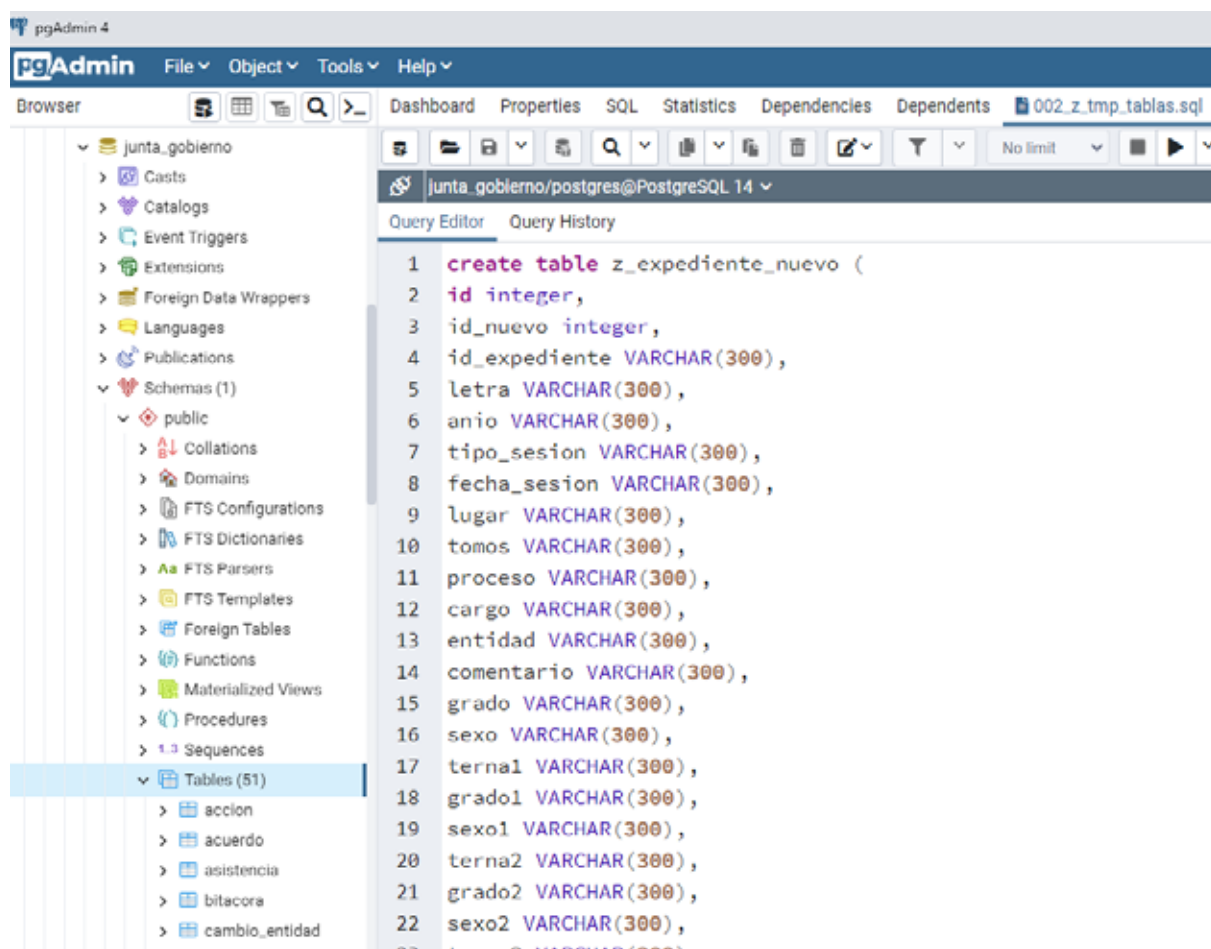
The spreadsheets shown below the formula bars illustrate the data structure:

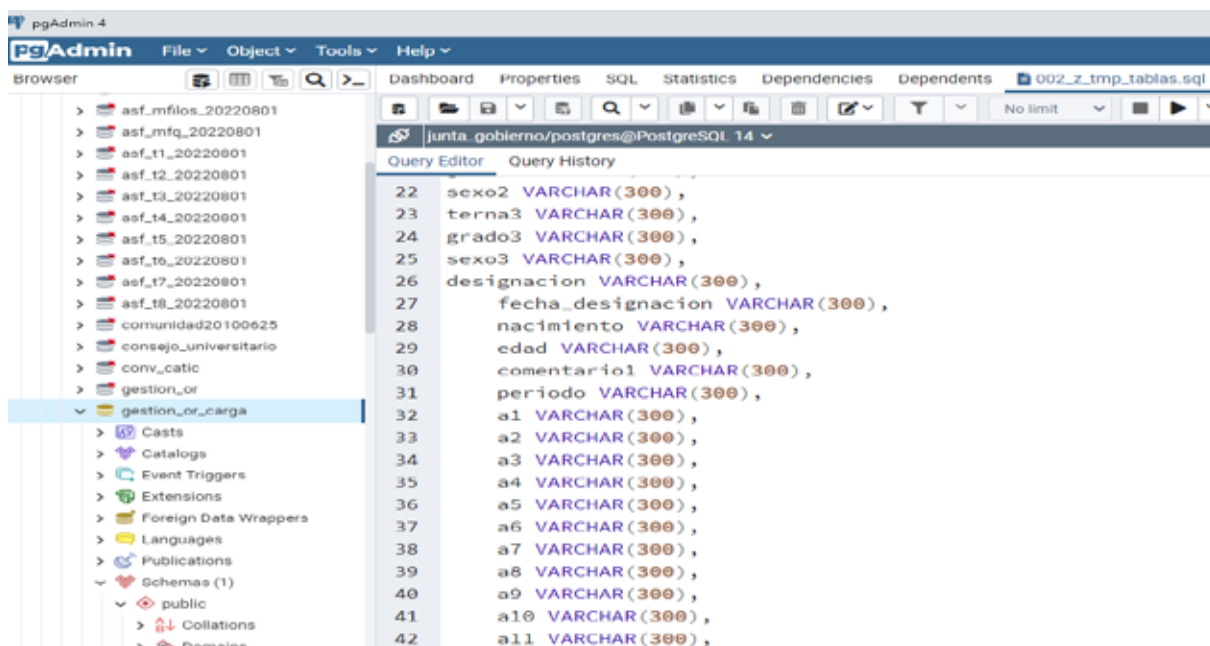
- Spreadsheet 1 (AX):** Column 1 contains 'acta_sesion'. Column 2 contains the start of the SQL insert statement: `INSERT INTO z_expediente_nuevo(id, id_expediente, letra, anio, tipo_sesion, fecha_sesion, lugar, tomos, proceso, cargo,`
- Spreadsheet 2 (AY):** Column 1 contains the rest of the SQL insert statement: `fecha_designacion, nacimiento, edad, comentario1, periodo, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12, a13, a14, a15, comentario2, presidente, secretario, observacion, acta_sesion) VALUES ('id', trim('id_expediente'), trim('letra'), trim('anio'), trim('tipo_sesion'), trim('fecha_sesion'),`
- Spreadsheet 3 (AZ):** Column 1 contains the start of the SQL insert statement: `INSERT INTO z_expediente_nuevo(id, id_expediente,`

ANEXO B. EJEMPLO DE USO DE PGADMIN

Figura 6

Uso del cliente PGAdmin para ejecutar el script de SQL para crear la tabla que fue usada para insertar posteriormente los datos del archivo de Excel de expedientes para su revisión



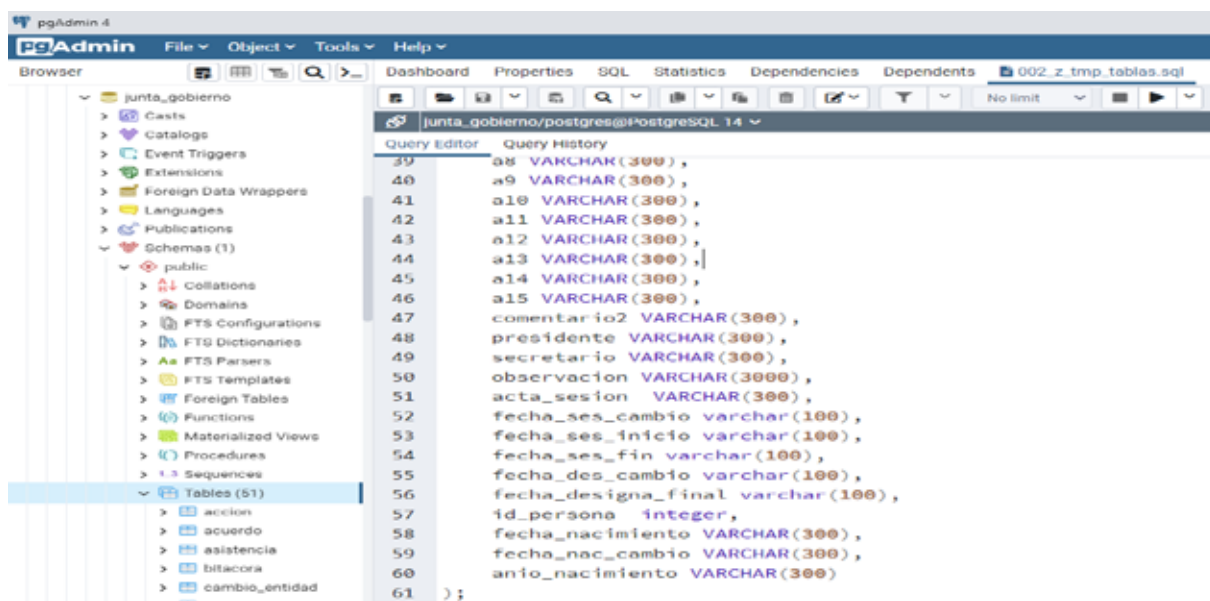


The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of the database structure, with 'gestion_or_carga' selected. The main window shows the 'Query Editor' with the following SQL code:

```

22 sexo2 VARCHAR(300),
23 terna3 VARCHAR(300),
24 grado3 VARCHAR(300),
25 sexo3 VARCHAR(300),
26 designacion VARCHAR(300),
27   fecha_designacion VARCHAR(300),
28   nacimiento VARCHAR(300),
29   edad VARCHAR(300),
30   comentario1 VARCHAR(300),
31   periodo VARCHAR(300),
32   a1 VARCHAR(300),
33   a2 VARCHAR(300),
34   a3 VARCHAR(300),
35   a4 VARCHAR(300),
36   a5 VARCHAR(300),
37   a6 VARCHAR(300),
38   a7 VARCHAR(300),
39   a8 VARCHAR(300),
40   a9 VARCHAR(300),
41   a10 VARCHAR(300),
42   a11 VARCHAR(300),

```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of the database structure, with 'Tables (51)' selected. The main window shows the 'Query Editor' with the following SQL code:

```

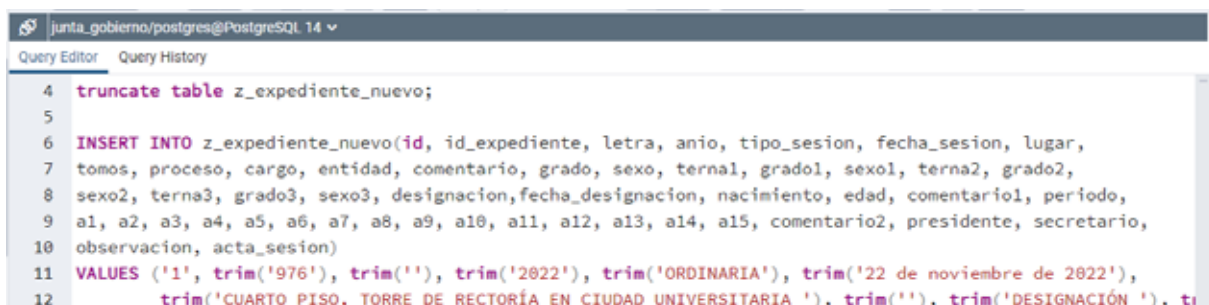
39   a8 VARCHAR(300),
40   a9 VARCHAR(300),
41   a10 VARCHAR(300),
42   a11 VARCHAR(300),
43   a12 VARCHAR(300),
44   a13 VARCHAR(300),
45   a14 VARCHAR(300),
46   a15 VARCHAR(300),
47   comentario2 VARCHAR(300),
48   presidente VARCHAR(300),
49   secretario VARCHAR(300),
50   observacion VARCHAR(3000),
51   acta_sesion VARCHAR(300),
52   fecha_ses_cambio varchar(100),
53   fecha_ses_inicio varchar(100),
54   fecha_ses_fin varchar(100),
55   fecha_des_cambio varchar(100),
56   fecha_designa_final varchar(100),
57   id_persona integer,
58   fecha_nacimiento VARCHAR(300),
59   fecha_nac_cambio VARCHAR(300),
60   anio_nacimiento VARCHAR(300)
61 );

```

Nota: La instalación de postgresQL incluye el cliente PGAdmin y se puede descargar del sitio <https://www.postgresql.org/download/>

Figura 7

Uso del cliente PGAdmin para ejecutar scripts de SQL para insertar datos



```

4 truncate table z_expediente_nuevo;
5
6 INSERT INTO z_expediente_nuevo(id, id_expediente, letra, anio, tipo_sesion, fecha_sesion, lugar,
7 tomos, proceso, cargo, entidad, comentario, grado, sexo, ternal, gradol, sexo1, terna2, grado2,
8 sexo2, terna3, grado3, sexo3, designacion, fecha_designacion, nacimiento, edad, comentario1, periodo,
9 a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12, a13, a14, a15, comentario2, presidente, secretario,
10 observacion, acta_sesion)
11 VALUES ('1', trim('976'), trim(''), trim('2022'), trim('ORDINARIA'), trim('22 de noviembre de 2022'),
12 trim('CUARTO PISO, TORRE DE RECTORÍA EN CIUDAD UNIVERSITARIA '), trim(''), trim('DESIGNACIÓN '), tr

```

ANEXO C. EJEMPLO DE CONSULTA PARA INSERTAR SESIONES

Figura 8

Ejemplo de la consulta usada para insertar los datos de las sesiones

```

INSERT INTO sesion(id_expediente, id_lugar, fecha)
SELECT DISTINCT e.id_expediente, id_lugar, ze.fecha_ses_inicio::date
FROM expediente e JOIN z_expediente_nuevo ze on (e.id_expediente = ze.id_nuevo)
LEFT JOIN
z_tmp_eq_lugar zl ON
(REPLACE(TRIM(LOWER(zl.lugar_s)), ' ', '') = REPLACE(TRIM(LOWER(ze.lugar)), ' ', ''))
LEFT JOIN
lugar l on (REPLACE(TRIM(LOWER(zl.lugar_c)), ' ', '') = REPLACE(TRIM(LOWER(l.nombre) ), ' ', ''))
ORDER BY id_expediente
;

```

ANEXO D. EJEMPLO DE CONSULTA PARA REVISAR FECHAS

Figura 9

Ejemplo de la consulta usada para revisar las fechas de designación y término de un miembro con sus fechas de asistencia mínima y máxima en los campos 1 y 2 de la lista de asistencia

```
SELECT '1', min(fecha_ses_inicio::date) as fec_minima, max(fecha_ses_fin::date) as fec_maxima,
m.id_miembro , p.nombre, p.primer_apellido, p.segundo_apellido, m.fecha_designacion,
    m.toma_protesta, m.fecha_termino
FROM z_expediente_nuevo ze JOIN z_tmp_eq_asistente_miembro za
    ON (trim(ze.a1)= za.asistente) JOIN miembro m ON (za.id_miembro = m.id_miembro) JOIN
    sesion s ON (ze.id = s.id_sesion) JOIN persona p ON (p.id_persona = m.id_persona)
    group by m.id_miembro , m.fecha_designacion, m.toma_protesta, m.fecha_termino,
    p.nombre, p.primer_apellido, p.segundo_apellido

UNION

SELECT '2', min(fecha_ses_inicio::date) as fec_minima, max(fecha_ses_fin::date) as fec_maxima,
m.id_miembro , p.nombre, p.primer_apellido, p.segundo_apellido, m.fecha_designacion,
    m.toma_protesta, m.fecha_termino
FROM z_expediente_nuevo ze JOIN z_tmp_eq_asistente_miembro za
    ON (trim(ze.a2)= za.asistente) JOIN miembro m ON (za.id_miembro = m.id_miembro) JOIN
    sesion s ON (ze.id = s.id_sesion) JOIN persona p ON (p.id_persona = m.id_persona)
    group by m.id_miembro , m.fecha_designacion, m.toma_protesta, m.fecha_termino,
    p.nombre, p.primer_apellido, p.segundo_apellido
```

ANEXO E. EJEMPLO DE CONSULTA PARA DICCIONARIO DE DATOS

Figura 10

Ejemplo de la consulta y descripción de los campos obtenida de la base de datos en postgresQL excluyendo las tablas de configuración de Laravel

```
select t1.table_name as tabla_nombre ,
       t1.COLUMN_NAME AS columna_nombre,
       t1.COLUMN_DEFAULT AS columna_defecto,
       t1.IS_NULLABLE AS columna_nulo,
       t1.DATA_TYPE AS columna_tipo_dato,
       COALESCE(t1.NUMERIC_PRECISION,
               t1.CHARACTER_MAXIMUM_LENGTH) AS columna_longitud,
       PG_CATALOG.COL_DESCRIPTION(t2.OID,|
       t1.DTD_IDENTIFIER::int) AS columna_descripcion
FROM
  INFORMATION_SCHEMA.COLUMNS t1
  INNER JOIN PG_CLASS t2 ON (t2.RELNAME = t1.TABLE_NAME)
WHERE
  t1.TABLE_SCHEMA = 'public' AND t1.table_name NOT IN
  ('failed_jobs', 'role_has_permissions', 'roles', 'sessions',
   'flyway_schema_history', 'jobs, migrations', 'model_has_permissions',
   'model_has_roles', 'password_resets', 'permissions', 'personal_access_tokens',
   'jobs', 'migrations')
ORDER by
  t1.table_name ,
  t1.ORDINAL_POSITION;
```

tabla_nombre	columna_nombre	columna_defec	columna_nu	columna_tipo_dato	columna_longitud	columna_descripcion
lugar	id_lugar		NO	integer	32	Identificador del lugar.
lugar	nombre		NO	character varying	200	Nombre del lugar donde se realizó la sesión de la Junta de Gobierno.
lugar	created_at	now()	NO	timestamp without time zone		Fecha y hora de creación del registro.
lugar	updated_at		YES	timestamp without time zone		Fecha y hora de última modificación de registro.

Análisis del proceso de integración de los cursos en línea de capacitación continua

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:

Llanos Navarrete, S. (2023). Análisis del proceso de integración de los cursos en línea de capacitación continua. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (126 - 138).
<https://doi.org/10.22201/dgtic.ctud.2023.1.1.25>

Sarai Llanos Navarrete

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

sarai.llanos@unam.mx

ORCID: 0009-0003-5492-4719

Resumen:

El análisis de un proceso enriquece a otras áreas, ya que pueden replicarlo, mejorarlo o adaptarlo de acuerdo con las necesidades que se tengan. Este análisis se centra en el proceso para la integración de cursos en línea a cargo de la Coordinación de Capacitación Continua a Distancia en la Dirección General de Cómputo y de Tecnologías de Información y Comunicación y que, a través de los años, se ha ido robusteciendo. Durante el análisis se hace énfasis en cada una de las etapas a cumplir y su justificación, con base en los lineamientos y estándares establecidos para cuidar la calidad de los productos. Parte importante de este proceso es el uso de un sistema de desarrollo propio, con más de diez años de madurez, y que entre otras ventajas se crea para cubrir las necesidades de creciente demanda, al centralizar todos los cursos en línea, y lograr homogeneidad de manera rápida y funcional, así como mantener el modelo de los cursos y la estructura de sus respectivas Unidades Didácticas, antes de ser implementados en un sistema de gestión de aprendizaje.

Palabras clave:

Cursos en línea, diseño instruccional, integración, unidades didácticas, sistema de contenidos.

1. INTRODUCCIÓN

En la Coordinación de Capacitación Continua a Distancia (CCCD) en la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) se cuenta con lineamientos y estándares establecidos para que el desarrollo de un curso en línea cumpla los criterios de calidad necesarios para ser implementado e impartido en el sistema de gestión de aprendizaje (*Learning Management System*, LMS), por ejemplo, Moodle. Por ello es vital elaborar un diseño instruccional y ejecutarlo durante la integración de un curso, y contar con un proceso definido de integración que ayudará a lograr que un curso en línea de capacitación continua cumpla con los criterios de calidad que se requieren. ¿Cuáles son estos criterios? ¿Qué es lo mínimo que debe tener un curso en línea? Como lo menciona en su tesis Vieyra (2020) “para valorar sus elementos que lo integran se utilizará el instrumento de evaluación del EC0366 “Desarrollo de cursos de formación en línea” del CONOCER” (p. 107). Este instrumento de evaluación que se menciona en el caso de uso citado es el mismo al que se apega el personal de la CCCD para el desarrollo de los cursos.

Asimismo, Lira y Brunett (2021) señalan que en México la educación universitaria mediada por tecnología ha adoptado recientemente criterios que tienden a establecer principios y estándares únicos para la evaluación y acreditación de los programas educativos en modalidad a distancia y mixta, en el que se reconoce la experiencia de diversos organismos internacionales. No obstante, para el caso de los cursos de capacitación, esta información es responsabilidad absoluta de las instituciones, así como también lo es la definición de los estándares, de los parámetros y de los instrumentos de verificación.

En este sentido, la integración del curso resulta de suma importancia porque es donde finalmente se verá plasmado el diseño instruccional y los contenidos didácticos desarrollados con base en lo establecido y avalados por un experto.

Sin embargo, además de mantener la calidad en la integración, ante la creciente demanda de cursos y procesos que se podrían sistematizar, surgió la pregunta: ¿Cómo se puede optimizar el proceso de integración de cursos en línea considerando los aspectos técnicos, pedagógicos y de calidad que se requieren?

Para ello, en 2011 se creó un sistema de desarrollo propio para la Creación de Unidades Didácticas en Línea, al que se nombró SCUDeL, que cumple con casi todos los criterios del instrumento EC0366, y lo que no cumple se satisface con el LMS en el que se implemente el curso. Desde 2012, se utiliza SCUDeL para integrar todos los cursos en línea de la CCCD, antes de ser implementados en Moodle. Se utiliza un sistema fuera del LMS para realizar esta integración ya que permite una independencia de cualquier LMS, por lo cual es posible implementar un curso en línea integrado en SCUDeL dentro de cualquier LMS sin importar el tipo, versión o ubicación del sistema de gestión de aprendizaje que se utilice. Lo anterior es posible por la manera en que se encuentran estructurados los archivos que se obtienen al finalizar la integración, los cuales funcionan a modo de micrositio que contiene todo lo necesario para un curso en línea.

Al utilizar el proceso de integración mediado por SCUDeL, y definido dentro de la CCCD, se ha logrado reducir los tiempos de integración, obtener cursos homogéneos y facilitar mejoras o actualizaciones de materiales que son requeridos por la temática relacionada con el cómputo y las TIC.

2. OBJETIVOS

General

- Analizar el proceso a seguir para integrar cursos en línea en un sistema de desarrollo propio con la finalidad de mantener la consistencia y uniformidad en el diseño, funcionalidad y apariencia, bajo el modelo que se sigue para los cursos en línea a cargo de la CCCD de la DGTIC. SCUDeL

Específicos

- Describir el proceso de integración de los cursos en línea de capacitación continua a cargo de la CCCD de la DGTIC, con el fin de mostrar los pasos necesarios para obtener un curso de calidad.
- Implementar el proceso de integración para sistematizar los métodos utilizados, con el propósito de optimizar tiempos y recursos.

3. PROCESO DE INTEGRACIÓN DE LOS CURSOS EN LÍNEA

Se entiende por **integración** a la creación de la presentación y de cada una de las unidades didácticas que componen un curso en línea. Cada una de estas unidades incluyen en el Sistema de Creación de Unidades Didácticas en Línea, una Introducción, Objetivos, Actividades (con sus respectivos materiales didácticos), Evaluación (y sus respectivos instrumentos), Conclusiones y Referencias.

Esta integración es la línea que divide al Desarrollo de la Implementación, con base al modelo clásico de diseño instruccional ADDIE, cuyo nombre obedece al acrónimo *analyze* (analizar), *design* (diseñar), *develop* (desarrollar), *implement* (implementar) y *evaluate* (evaluar), y que forma parte del modelo utilizado en la CCCD. Astudillo (2017) señala que, dentro de las diferentes fortalezas de este modelo, se puede determinar que conecta constantemente las características de todos los factores que pueden influir en un proceso efectivo de aprendizaje; de esta manera, en su análisis inicial se establece el perfil previo del sujeto, enfocando desde ahí la transformación del contenido de aprendizaje.

Para poder realizar una integración óptima de un curso en línea con base al modelo mencionado, se debe cumplir con las siguientes etapas:

- 1.Verificación de los formatos y criterios establecidos en los materiales didácticos e instrumentos de evaluación, antes de ser integrados.
- 2.Integración de los contenidos de un curso en línea en el Sistema para la Creación de Unidades Didácticas en Línea (SCUDeL).
- 3.Respaldo de materiales digitales fuente.

3.1 VERIFICACIÓN DE LOS FORMATOS Y CRITERIOS ESTABLECIDOS EN LOS MATERIALES DIDÁCTICOS E INSTRUMENTOS DE EVALUACIÓN, ANTES DE SER INTEGRADOS

Cada uno de los materiales didácticos e instrumentos de evaluación debe seguir una nomenclatura, lo que ayuda a corregir o actualizar rápidamente los sistemas correspondientes, cuando se soliciten cambios por errores o la actualización de algún contenido en un curso en línea.

Los contenidos se entregan en formato editable y publicado¹, con la siguiente nomenclatura:

Editable (sin acentos y sin espacios):

t##_TipoMaterial##_nombreMaterial_versionSoftware_añoCreacionEdicion_inicialesEditor.extOriginal

Publicado (sin acentos y sin espacios):

t##_TipoMaterial##_nombreMaterial.extPublicada

Donde:

- **t##**: corresponde al tema del que pertenece dicho material.
- **TipoMaterial##**: corresponde a la(s) letra(s) en paréntesis del tipo de material (tabla 1), seguido de un número consecutivo de acuerdo con el orden en que se integrará en la unidad. El número se reinicia en cada unidad.
- **nombreMaterial**: nombre breve que haga referencia al contenido del material.
- **versionSoftware**: en caso de haber utilizado alguna herramienta que no tenga compatibilidad entre versiones, es necesario especificar la versión de desarrollo para que pueda ser editado.
- **añoCreacionEdicion**: año en que se crea el material didáctico o cada vez que tenga alguna edición.
- **inicialesEditor**: iniciales del desarrollador o editor que realizó los últimos ajustes al material.

Los materiales didácticos e instrumentos de evaluación más comunes encontrados en el desarrollo de un curso en línea, así como los formatos y criterios requeridos antes de su integración se resumen a continuación.

¹ Hay algunos casos en los que no es necesaria la publicación del material.

Tabla 1

Descripción del formato y criterios por tomar en cuenta de acuerdo con el tipo de material a integrar.

Tipo de material	Formato	Criterios ²
Documento (d)	Editable: *.docx Publicado: *.pdf	<p>Documentos protegidos con una contraseña asignada por la CCCD, para evitar la edición y el copiado de los textos, y permitiendo una impresión de alta calidad.</p> <p>El tamaño de las páginas debe ser carta y con los metadatos requeridos (referencia a material desarrollado para la UNAM).</p> <p>Ejemplo del nombre del archivo (editable y publicado respectivamente):</p> <p>t01_d01_caracteristicasMacros_2023_SLN.docx</p> <p>t01_d01_caracteristicasMacros.pdf</p>
Video (v)	Editable: .cptx, .approj, entre otros Publicado: .mp4	<p>Las dimensiones se han ido modificando con el tiempo, actualmente se sigue la proporción 4:3, por tener una mejor adaptación en casi cualquier pantalla; y con los metadatos requeridos (referencia a material desarrollado para la UNAM). Asimismo, se verifica que el video tenga un peso menor a los 10MB por la velocidad de descarga intermitente o baja que algunos de los participantes pueden llegar a tener, y para no saturar el espacio asignado del servidor para los cursos en línea.</p> <p>Ejemplo del nombre del archivo (editable y publicado respectivamente):</p> <p>t01_v01_planeacion_AP8_2023_SLN.approj</p> <p>t01_v01_planeacion.mp4</p>
Interactivo (i)	Editable: enlace en un *.txt del material editable. Publicado: *.zip	<p>Se pueden utilizar diversas plataformas para generarlos, pero, en lo posible, es importante contar con los archivos descargados para que se tenga la copia del recurso a implementar y no depender de la plataforma en la que se creó.</p> <p>Ejemplo del nombre del archivo (editable y publicado respectivamente):</p> <p>t01_i01_aspectosDuranteGrabacionMacros_2023_SLN.txt</p> <p>t01_i01_aspectosDuranteGrabacionMacros.zip</p>

2 Los criterios se basan en los lineamientos de desarrollo establecidos en la Coordinación.

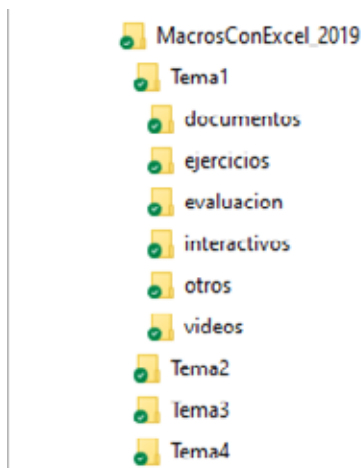
Tipo de material	Formato	Criterios ²
URL (u)	Enlace en un *.txt o *.docx	<p>Entregar el enlace a utilizar en un documento editable.</p> <p>Verificar que sea accesible y se considere un recurso educativo.</p> <p>Ejemplo del nombre del archivo (solo se requiere el editable): t01_u01_historiaMacros.txt</p>
Foro (f)	.txt o .docx	<p>Entregar el texto para foro en un documento editable.</p> <p>Se prefiere que se inicie con un caso y posteriormente las preguntas de debate (2 a 4 preguntas), las cuáles deben ser evaluativas, que fomenten el pensamiento crítico y la argumentación.</p> <p>Máximo 890 caracteres (que es lo que permite el sistema).</p> <p>Ejemplo del nombre del archivo (solo se requiere el editable): t01_f01_grabacionMacros_2023_SLN.txt</p>
Ejercicio (e)	<p>Editable: *.docx</p> <p>Publicado: *.pdf</p>	<p>Documentos protegidos con una contraseña asignada por la CCCD, para evitar la edición y el copiado de los textos, a excepción de que se necesite copiar la información por la actividad a realizar, y permitir una impresión de alta calidad.</p> <p>El tamaño de las páginas debe ser carta y con los metadatos requeridos (referencia a material desarrollado para la UNAM).</p> <p>Ejemplo del nombre del archivo (editable y publicado respectivamente): t02_e01_conceptosBasicosDeVBA_2023_SLN.docx t02_e01_conceptosBasicosDeVBA.pdf</p>
Material (m)	.zip o formato fuente a trabajar	<p>Se recomienda .zip si se va a trabajar con varios archivos.</p> <p>Nombre del archivo (solo se requiere el editable): t##_m##_nombreMaterial.zip/extensiónArchivoFuente</p> <p>En caso de tratarse de un material para ejercicio(e) o práctica (p) se le agrega la letra según corresponda y el número, después del tema, ejemplo: t02_e01_m01_conceptosBasicosDeVBA.zip t02_p01_m01_conceptosBasicosDeVBA.zip</p>

Tipo de material	Formato	Criterios ²
Cuestionario (c)	<p>Editable: *.docx</p> <p>No se entrega publicado.</p>	<p>Las preguntas con sus respectivas respuestas son entregadas en un documento editable.</p> <p>En caso de ser preguntas de opción múltiple, cada pregunta deberá tener 4 incisos de opción múltiple y siempre con el primer inciso como respuesta correcta, para facilitar la implementación en el sistema de gestión de aprendizaje Moodle.</p> <p>Ejemplo del nombre del archivo (solo se requiere el editable): t02_c01_conceptosBasicosDeVBA_2023_SLN.docx</p>
Práctica (p)	<p>Editable: *.docx</p> <p>Publicado: *.pdf</p>	<p>Documentos protegidos con una contraseña asignada por la CCCD, para evitar la edición y el copiado de los textos, a excepción de que se necesite copiar la información por la actividad a realizar, y permitir una impresión de alta calidad.</p> <p>El tamaño de las páginas debe ser carta y con los metadatos requeridos (referencia a material desarrollado para la UNAM).</p> <p>Ejemplo del nombre del archivo (editable y publicado respectivamente): t01_p01_grabacionMacros_2023_SLN.docx t01_p01_grabacionMacros.pdf</p>
Lista de cotejo (lc)	<p>Editable: *.xlsx</p> <p>No se entrega publicado.</p>	<p>Utilizar el formato establecido y verificar que se evalúen todos los puntos solicitados en la práctica que le corresponde.</p> <p>El texto de cada elemento a evaluar debe contener máximo 200 caracteres (que es lo que permite el sistema).</p> <p>Se debe cuidar la redacción de cada uno de los puntos a evaluar, de tal manera que demuestre que si es correcto lo que se desea evaluar, es el puntaje que alcanzará.</p> <p>Nombre del archivo (solo se requiere el editable): t01_p01_lc01_grabacionMacros_2023_SLN.xlsx</p>

Finalmente, los materiales se acomodan en carpetas por tema y tipo, para facilitar la localización al momento de proceder con la integración y respaldo.

Figura 1

Organización de las carpetas por temas y tipo de materiales por utilizar durante la integración



Es necesario cumplir con todos los criterios detallados, para proceder con la integración de los contenidos digitales en el sistema interno.

3.2 INTEGRACIÓN DE LOS CONTENIDOS DE UN CURSO EN LÍNEA EN EL SISTEMA PARA LA CREACIÓN DE UNIDADES DIDÁCTICAS EN LÍNEA (SCUDeL)

El modelo que se utiliza para la integración de cursos en línea tiene en sus Unidades Didácticas la siguiente estructura:

- Introducción.
- Objetivos.
- Actividades.
- Evaluaciones.
- Conclusiones.
- Referencias.

Con base en este modelo, y a partir de la organización de las unidades didácticas y la necesidad de homologar la imagen y estructura de los cursos, es que surge la idea de sistematizar la creación de Unidades Didácticas para la integración de los cursos, por lo que se crea **SCUDeL**, siglas que provienen de **Sistema para la Creación de Unidades Didácticas en Línea**³.

3 La autora de este reporte es coautora del desarrollo de la primera versión de SCUDeL (2011), junto con el académico Alejandro F. Zárate.

Este sistema se ha venido utilizando desde el año 2012 para la integración de todos los cursos en línea, y su propósito principal es generar un micrositio del curso con los materiales integrados, el cual puede ser implementado en cualquier lugar, en particular, los cursos generados por personal de la CCCD se despliegan dentro de un LMS Moodle porque es la plataforma utilizada para impartir los cursos.

Para acceder al sistema se realiza lo siguiente:

1. Ir a la dirección: <https://nefertete.tic.unam.mx/scudel/>
2. Se ingresa el **Nombre de usuario** y **Contraseña** y, dependiendo de los privilegios que se tengan, se muestran cursos y accesos.

Para integrar un curso, se requieren permisos de **Integrador**, de lo contrario solamente se pueden visualizar los cursos asignados. Con este permiso el integrador puede ingresar a su curso para comenzar a construir la Presentación y cada una de las Unidades Didácticas con la respectiva estructura que lo compone.

Figura 2

Captura de pantalla de SCUDeL que muestra la asignación de un usuario como integrador en el curso de Macros con Excel



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

Dirección de Docencia en
 Tecnologías de Información y Comunicación

EL USUARIO HA SIDO AGREGADO CORRECTAMENTE AL CURSO

HOJAS DE CÁLCULO
MACROS CON EXCEL 2019

NÚMERO DE TEMA	NOMBRE DEL TEMA	VERSIÓN FINAL
0	Presentación del curso	presentacion.html
1	Agrega nombre de tema	tema1.html

Reordenar Agrega un tema Obtener versión final Eliminar

El detalle del tipo y la cantidad de materiales en este curso se muestra en la siguiente tabla:

Tipo	Cantidad
Total	0

Los usuarios que tienen acceso a este curso y los permisos correspondientes se muestran a continuación.

Nombre	Permiso
Saraí Llanos Navarrete	Integrador

Agregar usuario al curso

En la Figura 2 se muestran claramente tres secciones:

- En la sección A se encuentra la Presentación del curso por editar, así como la opción para agregar tantos temas como Unidades Didácticas se requieran.
- En la sección B se muestra el detalle del tipo y la cantidad de materiales didácticos que se van a ir integrando en el curso. Inicialmente está en cero (0), pero conforme se va subiendo en la tabla se va actualizando con el tipo y cantidad de recursos totales que se tienen en el curso.
- En la sección C se muestran los usuarios que tienen acceso al curso y los permisos correspondientes. En este ejemplo se encuentra un usuario con permisos de Integrador.

El procedimiento recomendado para realizar la integración del curso en este sistema es:

1. Agregar tantos temas como requiera el curso y asignarles el nombre en mayúsculas.
2. Ir a la Presentación del curso y realizar el llenado de cada uno de los campos requeridos. Es importante comenzar con la creación de temas ya que en la Presentación se retoman todos los que fueron creados para la construcción del temario correspondiente.
3. Construir cada una de las Unidades Didácticas, con los materiales didácticos e instrumentos de evaluación que previamente se verificaron y validaron.
4. Establecer el calendario del curso: temas por semanas a revisar.
5. Revisión final de la integración.

El contar con un Sistema para la Creación de Unidades Didácticas en Línea (SCUDeL) apoya sin duda la creación de cursos en línea, al automatizar procesos y tener la disponibilidad de manera rápida para implementar el curso en cualquier sistema de gestión de aprendizaje (*LMS*) y aún para la búsqueda de los materiales disponibles.

Figura 3

Vista de Administrador de SCUDeL



3.3 RESPALDO DE MATERIALES DIGITALES FUENTE

Al inicio se puntualizó en la necesidad de mantener un orden en el manejo de los archivos tanto editables como publicados en carpetas, para realizar el proceso de la integración; lo anterior es importante porque servirá también como la copia de materiales digitales fuente que se debe entregar al terminar con el proceso de integración del curso en SCUDeL, lo cual tiene, entre otras ventajas, las siguientes:

- Asegurar la integridad y disponibilidad de los materiales digitales fuente para su posterior consulta, actualización e integración en los mismos o en nuevos cursos.
- Corregir, modificar o hacer las adecuaciones necesarias con base en las retroalimentaciones de participantes y asesores.
- Respaldo de materiales digitales fuente (actualizados o de nueva creación) de cada curso en línea desarrollado a cargo de la CCCD, para el inventario anual.

Sin embargo, a lo largo del tiempo se ha visto que la entrega de la carpeta final de los materiales digitales fuente algunas veces está incompleta, por lo que para verificar la entrega y determinar qué materiales específicamente faltan, se elabora un documento con el reporte de lo entregado y lo faltante, a través de listas de cotejo por cada tema que componen el curso entregado, lo cual se muestra en el Anexo A.

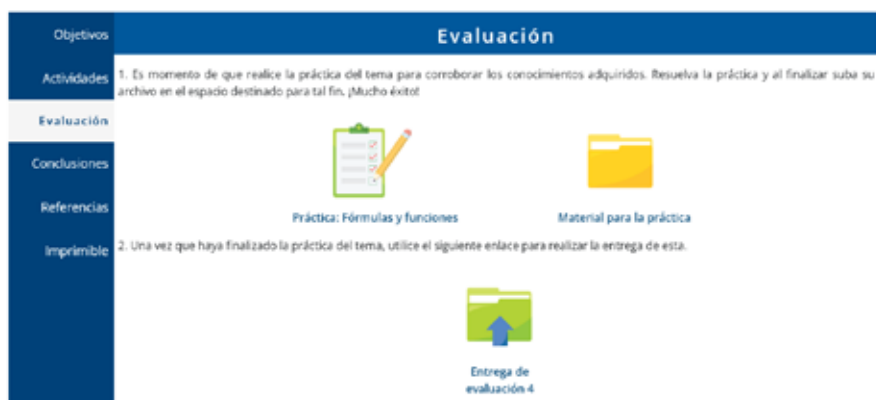
4. RESULTADOS

El Sistema SCUDeL permite integrar cursos en línea con una identidad propia, homogéneos y alineados al modelo establecido en la CCCD de la DGTIC; además, facilita el almacenamiento, la búsqueda y la reutilización de los materiales didácticos y la actualización de los cursos.

Al integrar los materiales en el sistema SCUDeL se ofrece la posibilidad de personalizar su tipo, mediante el uso de iconos, lo que contribuye a generar un mayor interés y claridad en los estudiantes. Estos cambios de iconos se realizan de manera global dentro del sistema y se actualizan en todos los cursos, de manera automática.

Figura 4

Ejemplo de curso integrado en SCUDeL



Asimismo, la centralización del proceso dentro del sistema ha permitido implementar los cursos en diversas plataformas LMS en corto tiempo. Los proyectos de cursos requieren que se utilicen versiones dedicadas de LMS por proyecto, por las actualizaciones mismas de los sistemas de gestión de aprendizaje y hasta por las políticas establecidas por el área de plataformas de la CCCD de la DGTIC.

Finalmente, las cifras del uso de SCUDeL para la integración de cursos son las siguientes:

Desde 2012 a la fecha, se han integrado 513 cursos con un total de 2,529 unidades didácticas, dentro de las cuales se han utilizado 1,853 archivos de trabajo, 6,846 documentos, 9,967 videos y 613 ejercicios; para la evaluación de estas unidades se han integrado 126 foros, 1,080 cuestionarios y 2,149 prácticas. En resumen, a través de SCUDeL se han utilizado 22,334 materiales para 2,529 unidades didácticas en 513 cursos de capacitación continua en TIC.

5. CONCLUSIONES

La integración es un proceso vital para llevar a buen término un curso en línea, de este proceso depende la calidad y por lo tanto el éxito durante su impartición.

Sin embargo, con base en la experiencia en el desarrollo de cursos no basta con tener documentado este proceso, es necesario sistematizar el proceso completo o la mayor cantidad posible del mismo. En este sentido con el sistema SCUDeL se ha logrado gran parte de esta sistematización ya que con este sistema los cursos se ven homogéneos, tanto en imagen como en estructura, son fáciles de implementar en cualquier LMS y son independientes de los propios LMS, lo cual permite concentrar los esfuerzos en la integración de los cursos más que en la herramienta que se utilizará para impartirlos.

Asimismo, el proceso de integración de cursos se ha fortalecido a lo largo de los años para abarcar la mayor cantidad de detalles, desde la nomenclatura de materiales hasta el resguardo de los materiales didácticos, lo cual permite mejoras en su administración, para procesos de búsqueda, actualización y reutilización, lo que sin duda se puede fortalecer en busca de la calidad y mejora continua.

En el último año, se utilizó este proceso para validar y verificar que estuviera alineado con los criterios contenidos dentro del instrumento de certificación "EC0366 Desarrollo de cursos de formación en línea": se cumplió con los puntos solicitados y se obtuvo la certificación correspondiente.

Sin embargo, el éxito en la integración del curso, sobre todo cuando se trata del desarrollo de un curso nuevo, recae en la persona que se encarga de integrarlo porque, aunque el sistema facilita el proceso, es tarea del integrador: la forma en que se integra, el orden de las actividades y la manera de apropiar los materiales didácticos conforme a la guía didáctica y la guía de actividades del curso, para asegurar el cumplimiento de los objetivos de enseñanza-aprendizaje.

Por lo tanto, se puede concluir que, aunque la sistematización de los procesos optimiza los recursos disponibles, siempre será necesario de una persona experta que realice la integración conforme a los requisitos solicitados para cada curso en particular.

REFERENCIAS BIBLIOGRÁFICAS

- Astudillo Ganora, B. (2017). *Modelo ADDIE como apoyo al desarrollo docente instrumental en competencias TIC como plan de certificación en estándares internacionales* [Tesis de Magister en Educación. Universidad de Chile]. Repositorio Académico de la Universidad de Chile <https://repositorio.uchile.cl/handle/2250/167803>
- Lira García, A. A. & Brunett Zarza, K. (2021). *Indicadores para evaluar la calidad en un curso de capacitación e-learning en México. Revista Tecnología, Ciencia y Educación*, (20), 83-102. <https://doi.org/10.51302/tce.2021.610>
- Vieyra González, F. J. (2020). *Propuesta de intervención para mejorar la asesoría académica de la MADEMS Español a Distancia* [Tesis de maestría, Universidad de Guadalajara, Sistema de Universidad Virtual]. Repositorio Institucional de la Universidad de Guadalajara. <https://riudg.udg.mx/bitstream/20.500.12104/82492/1/MSUV10102FT.pdf>
- Zárate Pérez, Alejandro Felipe & Llanos Navarrete, Sarai. (2011). *Sistematización para la creación de unidades didácticas para cursos en línea*. [Tesis de licenciatura. Facultad de Ciencias. Universidad Nacional Autónoma de México]. Repositorio Institucional de la UNAM. <https://hdl.handle.net/20.500.14330/TES01000692901>

ANEXO A

Reporte del respaldo de cursos en línea de la CCCD						
A. DATOS BÁSICOS						
1. Curso:					# de temas:	
2. Nombre del(a) responsable del curso:						
3. Fecha próxima de impartición:						
4. Fecha de entrega de FUENTES:						
5. Ruta de entrega de FUENTES:						
6. Ruta de integración del curso (SCUDeL):						
7. Ruta de implementación en el LMS (Moodle):						
8. Revisores(as):						
9. Fecha inicio de revisión:						
B. DATOS DE FINALIZACIÓN						
10. Fecha final de revisión:						
11. Ruta de respaldo:						
12. Curso completo en respaldo:			<input type="checkbox"/>			
13. Observaciones generales:						
C. LISTA DE COTEJO POR TEMA						
Tema 1						
Tipo de recurso	# en SCUDEL	# en respaldo	Elemento a evaluar en el respaldo	Completo	En revisión	Observaciones
Documento	0	0	Los documentos deben proporcionarse en formato PDF y en Word.	<input type="checkbox"/>	<input type="checkbox"/>	
Video	0	0	Los videos deben proporcionarse en formato MP4 y el archivo editable (.ostc, .apnre, etc)	<input type="checkbox"/>	<input type="checkbox"/>	
Interactivo	0	0	Debe estar el archivo .zip con todo el contenido así como la información relevante (al menos el programa utilizado, fecha de creación y si es posible la cuenta utilizada) en un documento de texto.	<input type="checkbox"/>	<input type="checkbox"/>	
Foro	0	0	La redacción del foro en un documento de texto.	<input type="checkbox"/>	<input type="checkbox"/>	
Material	0	0	Si es necesario se debe proporcionar el editable del material.	<input type="checkbox"/>	<input type="checkbox"/>	
Cuestionario	0	0	Los cuestionarios deben proporcionarse en formato de Word.	<input type="checkbox"/>	<input type="checkbox"/>	
Práctica	0	0	Las prácticas deben proporcionarse en formato PDF y en Word.	<input type="checkbox"/>	<input type="checkbox"/>	
Tarea	0	0	Así como sus materiales y listas de cotejo en caso de existir.	<input type="checkbox"/>	<input type="checkbox"/>	
Tarea	0	0	Las tareas deben proporcionarse en formato PDF y en Word.	<input type="checkbox"/>	<input type="checkbox"/>	
Wiki	0	0		<input type="checkbox"/>	<input type="checkbox"/>	

Rehabilitación del sistema de proyección del Museo Universitario de Ciencias y Arte

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:

CasasCordero, A. (2023). Consideraciones para rehabilitar el sistema de proyección "Maqueta interactiva CU". *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (139 - 152).
<https://doi.org/10.22201/dgtic.ctud.2023.1.1.26>

Araceli Casas Cordero

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

casas@unam.mx

ORCID: 0009-0006-3884-0517

Resumen:

La experiencia que se documenta está basada en la intervención para reactivar el proyecto "Maqueta interactiva" realizado por la Dirección General de Patrimonio Universitario y la Facultad de Arquitectura (2017, *Ciudad Universitaria ya tiene una maqueta interactiva*). El proyecto actualmente se encuentra en exhibición al público general en un espacio del Museo Universitario de Ciencias y Arte (MUCA), Ciudad Universitaria. En este trabajo se analizan los recursos de hardware del proyecto de *video mapping*, de manera integral, y la descripción de los componentes se focaliza en destacar que el reemplazo de adaptadores, convertidores y cables puede reducir la calidad del trabajo gráfico que se presenta, en el caso de elegir componentes incompatibles.

Palabras clave:

Sistemas *video mapping*, video proyección, conectores, adaptadores de señal de A/V.

1. INTRODUCCIÓN

El *video mapping* es una técnica visual, que combina sistemas de visualización con recursos mixtos. A través de proyectores de video se despliega imagen fija o en movimiento sobre superficies reales, entornos aumentados y contiene audio (Fischnaller, 2018). Los recursos proyectados deben contar con una narrativa que estructure el contenido visual, el audio y la ambientación con el objetivo de difundir determinada información. Las características de la instalación y los componentes de hardware tienen características específicas relacionadas al espacio empleado y el tema que se desarrolle.

Se recomienda al lector revisar el espectáculo 'Memoria luminosa', el cual se presentó en el Zócalo de la Ciudad de México en el mes de agosto de 2021. El sistema utiliza diferentes recursos visuales que se proyectan en una maqueta edificada en una proporción del 35 % del tamaño original (EFE., 2021), con la finalidad de narrar la historia de la Gran Tenochtitlán (Serrano, 2021). El montaje se realizó en un espacio abierto, para recibir a una audiencia numerosa.

Figura 1

Memoria luminosa

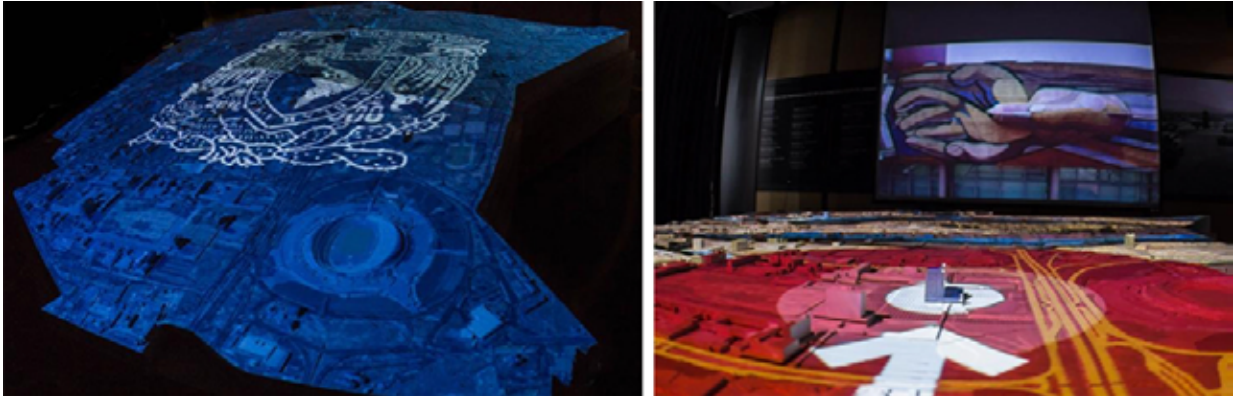


Nota: fuente de (Guerrero, H., 2021) (Estrella, A., 2021). Las proyecciones se realizan en la maqueta de la pirámide que se observa en las imágenes. El destello de luz que se observa en la imagen derecha (en el cuadrante superior derecho), es uno de los proyectores.

Otro trabajo a destacar, es el sistema de video proyección de la 'Maqueta interactiva de Ciudad Universitaria' que se encuentra en las instalaciones del Museo Universitario de Ciencias y Artes (MUCA). Las video imágenes se proyectan en una maqueta del campus universitario realizada en cartón a una escala 1:1000 que contiene diferentes narrativas relacionadas a la historia del patrimonio universitario (DGCS-UNAM, 2017).

Figura 2

Video proyección



Nota: Fuente tomada de DGCS-UNAM, 2017. Sistema de Video Mapping de la Maqueta Interactiva de Ciudad Universitaria, la coloración corresponde a la proyección; a la derecha se observa una flecha que señala el edificio de Rectoría en donde se encuentra la pintura que se muestra en la pantalla al frente.

Se realizó la revisión de trabajos anteriores con la finalidad de observar que es en relación con las características de la temática, el espacio disponible y la tecnología, entre otros factores, que los proyectos tienen necesidades particulares, en los que se involucran los recursos de hardware, software y elementos adicionales para la proyección. A partir de lo anterior, se explica que en el desarrollo de un proyecto de *video mapping*, se distinguen tres componentes: por un lado la realización del guión, la elaboración de la gráfica y la selección de los componentes de hardware. El guión define la estructura de la información y se complementa con los elementos gráficos, así también los componentes de hardware como proyectores, audio, luces o cualquier otro, deben estar comunicados con la tecnología adecuada para que el orden de aparición y la secuencia se presente sin retardos de tiempo y con la misma calidad visual; esta descripción explica la naturaleza de la sincronización.

1.1 DEFINICIÓN DEL PROBLEMA

La Maqueta Interactiva de Ciudad Universitaria presentó problemas diversos, a partir de lo cual, participé en su reactivación. Una de las situaciones que se presentó fue que los proyectores dejaron de recibir señal de audio y video (A/V) de la computadora y con ello se canceló la posibilidad de mostrarlo al público por un tiempo prolongado.

El sistema está conformado con los siguientes componentes: una computadora Mac Pro 2013, cuatro videoproyectores y un equipo de audio (subwoofer y cinco bocinas); los datos gráficos (imagen y video) están integrados en el software MadMapper V. 2.2.2, a través del cual se asigna la secuencia y el momento de aparición de cada uno de los elementos proyectados, además de asignar a cada proyector la información a mostrar (figura 3).

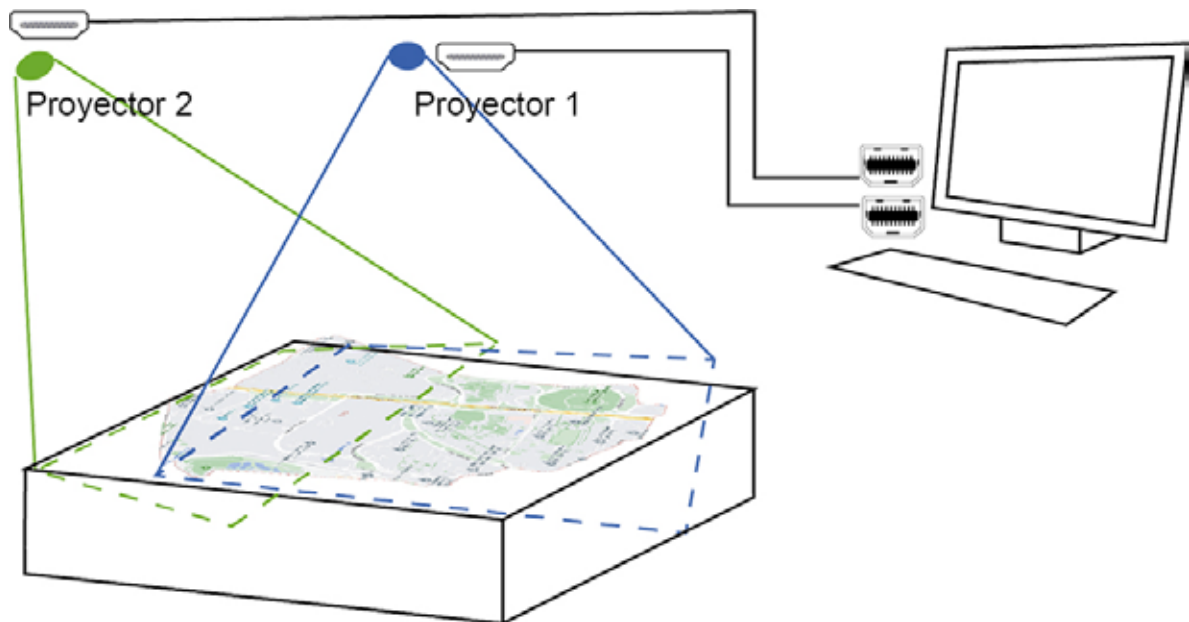
El equipo que se describe está en un límite muy cercano a la obsolescencia, por lo que se analizó la posibilidad de reemplazar la computadora por un modelo actual, sin embargo, la versión del software no sería compatible; adicionalmente, actualizar el software implicaba sustituir los videoproyectores con

tecnología compatible, por lo que la propuesta no era viable debido a que su costo estaba fuera de presupuesto, por lo que la relevancia del trabajo se enfocaría en encontrar realmente la naturaleza del problema con la intención de reemplazar algunos de los componentes.

El escenario planteado motivó a realizar una revisión exhaustiva de la tecnología del sistema, analizar los componentes de comunicación entre los dispositivos con el propósito de buscar un reemplazo o añadir algún componente, como es el caso de los conectores, adaptadores y cables, además de elegir aquellos que garantizaran la máxima calidad gráfica.

Figura 3

Sistema hipotético, con dos proyectores



Nota: Cada uno de ellos plasma un segmento de imagen sobre una superficie horizontal, esta información es la que se envía el ordenador a través del puerto Mini DisplayPort a HDMI

2. OBJETIVOS

- Comprender la naturaleza de los sistemas de video mapping, desde el punto de vista de la tecnología que se emplea.
- Identificar las conexiones y adaptadores que son compatibles, así como sus características más importantes.

3. SOBRE EL DESARROLLO

3.1 PROCEDIMIENTO DIAGNÓSTICO

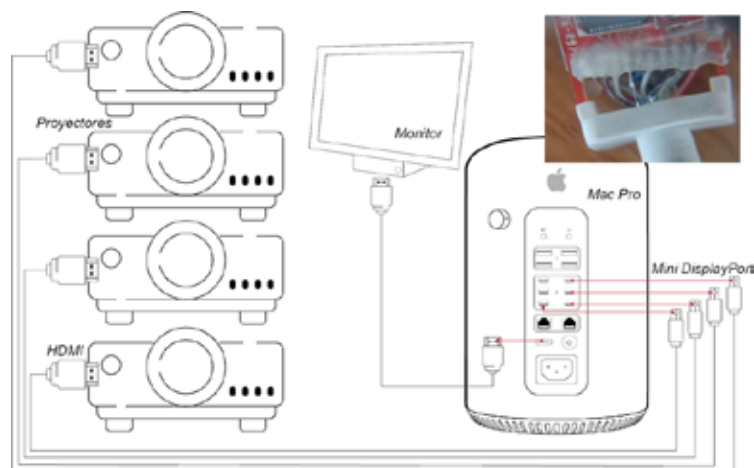
En una primera evaluación se confirmó que no se realizó actualización de sistema operativo en la computadora, adicionalmente el software MadMapper instalado, correspondía a la versión de la época en la que se realizó el proyecto. Los archivos de trabajo se abrieron sin presentar problema, lo cual determinó que la afectación tenía una relación inequívoca con los componentes de hardware, es decir, no había comunicación con los dispositivos de proyección.

Para aislar el problema se probó cada uno de los proyectores y se observó que funcionaban correctamente, así también se confirmó que los cables transmitían la señal, lo que llevó a determinar que el problema estaba en la conexión de la computadora (Mini DisplayPort a HDMI), como se muestra en la figura 4.

Al realizar pruebas de funcionamiento se observó un calentamiento inusual en el adaptador conectado al Mini DisplayPort; en consecuencia, tenía un funcionamiento inestable.

Figura 4

Mini DisplayPort



Nota: Las conexiones señaladas con un círculo rojo en el CPU presentaron el problema de comunicación. El interior del adaptador de señal Mini DisplayPort a HDMI (imagen superior-derecha), presenta cables sin recubrimiento debido al calentamiento.

3.2 EL VIAJE DE LOS DATOS A TRAVÉS DE CABLES Y ADAPTADORES

El hardware empleado en los sistemas de video mapping presenta un desgaste por el uso continuo, como se pudo demostrar al probar el deterioro del conector HDMI (figura 4) debido al calentamiento que ocurre debido a que el adaptador empleado transporta imagen con mayor velocidad del puerto Mini DisplayPort al HDMI; a pesar de que son dos estándares de buena calidad, se deterioran con el tiempo (figura 5).

Figura 5

Adaptador que lleva datos del extremo Mini DisplayPort (izq) a HDMI (derecho)

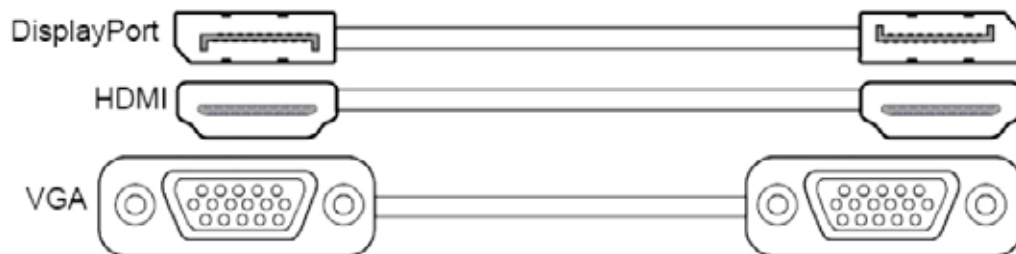


El conector Mini DisplayPort es una versión de tamaño reducido del Display Port y tiene mayores cualidades que el HDMI por sus frecuencias de actualización de pantalla, por su capacidad de salida a múltiples pantallas a través de Multi-Stream Transport (MST) y por el ancho de banda de 48 Gbps contra 7.4 Gbps (Fernández, Y., 2022), en este sentido, el volumen de información que transporta se merma cuando llega a la terminal HDMI porque no responde con la misma velocidad, y entonces se calienta.

La mejor estrategia para lograr el mejor rendimiento y compatibilidad es utilizar la conexión directa, donde el conector de entrada y salida es el mismo, como se muestra en la figura 6; en este caso, no existe la posibilidad de pérdida en la calidad de transmisión de la señal.

Figura 6

Conexión directa que implica que la señal de origen no tiene ninguna conversión hasta su destino







La complejidad se incrementa cuando se requiere de un cableado de mayor longitud (más de 30 metros), o cuando se conectan equipos que requieren adaptadores. Un cable de mayor longitud reduce la calidad de la señal que transporta, sin embargo, para resolverlo, se añaden componentes electrónicos al interior de los adaptadores y cables para amplificar o convertir la señal, lo que se reconoce como **adaptadores activos** (DELL, 2022). Es importante mencionar que un adaptador activo no es bidireccional.




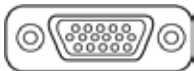
3.2.1 CARACTERÍSTICAS DE LOS CONECTORES DE LOS DISPOSITIVOS DE PROYECCIÓN

En la tabla 1 se documentan las características más destacadas de los conectores que transfieren A/V. Esta información es útil en una situación como la siguiente: al proyectar información desde una computadora, tableta, smartphone, etc., a otro dispositivo como proyector, monitor, smartphone, pantalla de televisión, etc., se requiere saber el tipo de conector a través del cual salen los datos y reconocer el conector por el cual entran para proyectarlos; una vez que se identifican se puede predecir la compatibilidad, por ejemplo, un conector HDMI transporta audio y video, pero si el destino es un conector DVI-I, entonces el audio se perderá debido a que este último no soporta esa información.

Tabla 1

Características destacadas de los conectores que transmiten audio y video; el orden en el que se presentan implica que se inicia con los más actuales

 <p>USB-C</p>	<p>El conector admite una velocidad de transferencia de datos de 10 Gbps y transfiere a través de una variedad de interfaces, como HDMI, DisplayPort y Thunderbolt 3 (Lee, H, et. all. s/f), (modo de funcionamiento Alt). Tiene la capacidad de convertir audio digital en sonido analógico (BenQ Ibérica, s/f). Es compatible con dispositivos Thunderbolt 3, maneja sistema de video de 8K a 30 Hz (Coughlin, T. M., 2017). Los conectores USB-C indican con una gráfica impresa su función: ⚡, indica que es Thunderbolt 3 y transmite señal de video, D DisplayPort, transmite señal de video estas dos variantes trabajan en modo Alt. Esta señalización SS indica que la función es de carga, no transmite señal de video (Club 3D).</p>
 <p>DisplayPort</p>	<p>A través de esta interfaz se pueden interconectar diversos dispositivos, como: Computadoras, Blue-Ray, televisores, etc. (2017, Coughlin, T. M.). Es compatible con USB-C, existen varias versiones y la más actual ofrece 77.4 Gbit/s, con una resolución de 8K a 60 Hz, se cataloga con mejor desempeño que HDMI, por sus frecuencias de actualización de pantalla y por su capacidad de salida a múltiples pantallas a través de Multi-Stream Transport (MST) (2022, Fernández, Y.). Soporta 6 a 16 bits de profundidad de color.</p>
 <p>Mini DisplayPort / Thunderbolt</p>	<p>Esta interfaz se reconoce como MiniDP o mDP, es una versión compacta del puerto DisplayPort y su uso es para la conexión de monitores y dispositivos de audio y video. El conector es idéntico al puerto Thunderbolt, tiene capacidades de alta velocidad en la transferencia de datos. Para identificar qué tipo de puerto es, hay que observar los iconos en el cable o conector, esta gráfica ⚡ indica que funciona como Thunderbolt o , indica que tiene la función de Mini DisplayPort (Mini DisplayPort Y thunderbolt: Compatibilidad Y USO – Faq-Mac.). Las últimas versiones trabajan con resoluciones 4K.</p> <p>Las versiones Thunderbolt 1 y 2, son compatibles con el Mini DisplayPort</p>









 HDMI	<p>High Definition Multimedia Interface o HDMI es una interfase para transmitir datos digitales (audio y video) de alta resolución sin compresión (2017, Coughlin, T. M.). Esta interfaz se diseñó para ser el sucesor de VGA y DVI. Existen diferentes versiones, la más actual 2.1 trabaja con una resolución de 8k a 120 Hz (2022, Fernández, Y.).</p>
 DVI-I	<p>El conector lleva a la vez una señal analógica y una digital, es funcional para tecnología de display LCD o CRT (Nvidia, security check, s/f) Solo transmite video</p>
 DVI-D	<p>Esta interfaz, envía únicamente señal digital (Security check) a una resolución de 1920 x 1200 px</p>
 VGA	<p>Solo transmite video análogo a una resolución de 640 x 480 px, un cable activo puede tener hasta 30 m de longitud (Tecnopc, 2023).</p>







3.2.2 COMPATIBILIDAD DE ADAPTADORES

La relación de comunicación en los conectores de la primera fila con los de la columna inicial de la tabla 2, permite analizar la siguiente situación: al proyectar un video realizado a una resolución de 3840 x 2160 pixeles desde una computadora con salida HDMI 2.1 (7680 x 4320 px) a un proyector con puerto VGA (640 x 480 px), será posible proyectarlo, a pesar de ser diferentes tecnologías (Roca, 2023).

Tabla 2

La relación de lectura: el conector de la primera fila se puede comunicar con cada uno de los conectores de la columna inicial y su funcionamiento se describe en la casilla en la que se interseca

							
 Conexión directa	Conexión directa	Res. 4K, transmite A/V, bidirecc. convertidor, Modo	Res. 4K, transmite A/V, bidirecc., compatible HDR.	Res. 4K, unidireccional, adaptador activo.	---	---	---

	Res. 8k, modo Alt, transmite A/V.	Conexión directa	Res. 8K, modo extendido y de espejo	Res. 4K, Activo y pasivo, unidireccional	---	Res. 1920 x 1080 px, adaptador activo.	Res. 1920 x 1080 px, adaptador activo, solo video.
	Res. 4K, transmite A/V, modo extendido y espejo.	Res. 4K, transmite A/V, soporta Multi Stream.	Conexión directa	Res. 4K, adaptador activo	---	Res. 1920 x 1080 px, convertidor pasivo, no soporta audio.	---
	Res. 8K, transmite A/V, compat. HDR.	Res. 4K, adaptador activo, unidirecc.	Transmite A/V, adaptador pasivo, max. longitud de cable 15.2m	Conexión directa	Res. 1920 x 1080 px, Unidirecc. solo video	Res. 1920 x 1080 px, bidirecc. No audio	Res. 1920 x 1080 px, solo transmite video.
	Res. 2560 x 1600 px, adaptador activo, modo Alt.	Res. 1920 x 1200 px, adap. activo, unidirecc.	Res. 1920 x 1200 px, no audio, unidireccional, compatible con Thunderbolt	Adaptador activo, bidireccional, no transfiere sonido.	Conexión directa	Bidireccional,	Res. 1920 x 1200 px, Solo transmite video.
	Res. 1920 x 1200 px, adaptador funcional hasta una distancia de 3m, modo Alt.	Res. 1920 x 1200 px, adaptador pasivo, unidirecc, 24 bits, no soporta audio.	Res. 1920 x 1200 px, compatible con Thunderbolt, adaptador pasivo	Res. 1920 x 1080 px, adaptador bidireccional - pasivo.	Adaptador bidireccional.	Conexión directa	Res. 1920 x 1200 px, solo transmite video.
	Res. 1920 x 1080 px, adaptador activo, modo extendido y dual	Res. 1920 x 1080 px, adaptador activo, unidireccional.	Res. 1920 x 1200, transmite solo video, unidireccional	Res. 1920 x 1080 adaptador activo, unidireccional.	Res. 1600 x 1200 px, señal analógica, Adaptador pasivo.	Res. 1920 x 1080 px, adaptador activo, señal digital a análoga, Unidirecc.	Conexión directa

Algunos fabricantes de adaptadores y cables incorporan un cable adicional para alimentación eléctrica externa, debido a que los chips y componentes electrónicos en su interior, demandan mayor energía y al proporcionar una alimentación independiente se mejora el desempeño de los componentes electrónicos.

3.2.3 APROXIMACIÓN A LA SOLUCIÓN

Al realizar la revisión de las tecnologías de “Compatibilidad de adaptadores” se observó que los fabricantes de adaptadores activos se alimentan de energía eléctrica con cables externos; esto se debe a que los componentes electrónicos que transforman la señal requieren de mayor potencia para realizar el trabajo. Lo anterior explica el porqué los adaptadores Mini DisplayPort a HDMI empleados en la “Maqueta Interactiva de Ciudad Universitaria” presentaron un calentamiento significativo mientras se utilizaban; se puede entender que había un exceso de trabajo en la transferencia de video datos.

El reemplazo de estos adaptadores no representaba una solución definitiva debido a que se mantuvo la sospecha de que los puertos de salida Mini DisplayPort del CPU tuvieron alguna afectación (figura 4).

4. RESULTADOS

En la búsqueda de un sistema de conexiones con una fuente de alimentación independiente, se analizó la tecnología de los **Docking Station** o **Estación de Acoplamiento**, debido a que es un componente de hardware que amplía la conexión de periféricos externos, ya sea monitores, videoproyectores, teclado, mouse, bocinas, puertos ethernet, concentradores USB, entre otros, además de añadir funciones de carga eléctrica. Estos componentes tienen una fuente de alimentación propia, lo que es conveniente porque logra estabilizar la transmisión de señales A/V. Aunque puede considerarse un equipo costoso, en realidad el gasto no se compara con el reemplazo de la computadora, sumado al costo de los proyectores y la adquisición de una licencia del software MadMapper.

El modelo recomendado es marca Wavlink (figura 7), con capacidades 4K, 100 W y con 19 entradas para diferentes dispositivos. Al compararlo con otros componentes, se validó la compatibilidad con el equipo de cómputo Mac Pro (2013) y los videoproyectores, y se verificó la disponibilidad de la empresa Wavlink para establecer contacto con el equipo de soporte técnico, lo cual fue útil porque se realizaron diversas consultas. El dispositivo solo requiere de una conexión USB con la computadora y la alimentación eléctrica, así como cuatro conexiones HDMI para los videoproyectores de la Maqueta Interactiva de Ciudad Universitaria.

Asimismo, se destacan los siguientes resultados:

1. Se estabiliza la señal de audio y video, debido a que los proyectores se conectan en el formato de conexión directa, como se recomienda en la sección 3.2.
2. Se elimina el calentamiento en las terminales HDMI, debido a que la alimentación eléctrica es independiente y con una especificación de 100 W.
3. Se pueden conectar otros dispositivos para propósitos diversos, sin que afecte el funcionamiento entre los dispositivos conectados.

Figura 7

Imagen de la estación de acoplamiento recomendada, desde la cual se obtienen cuatro conexiones HDMI

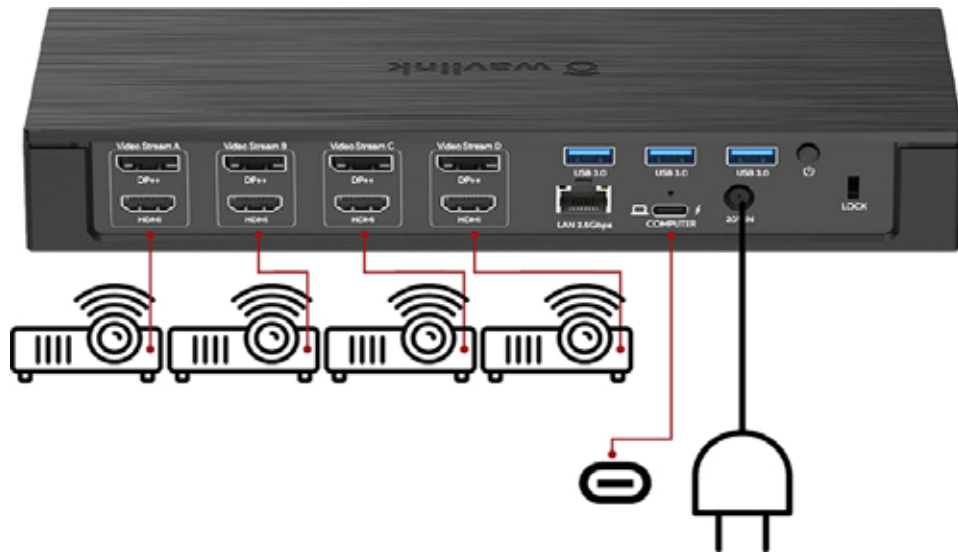


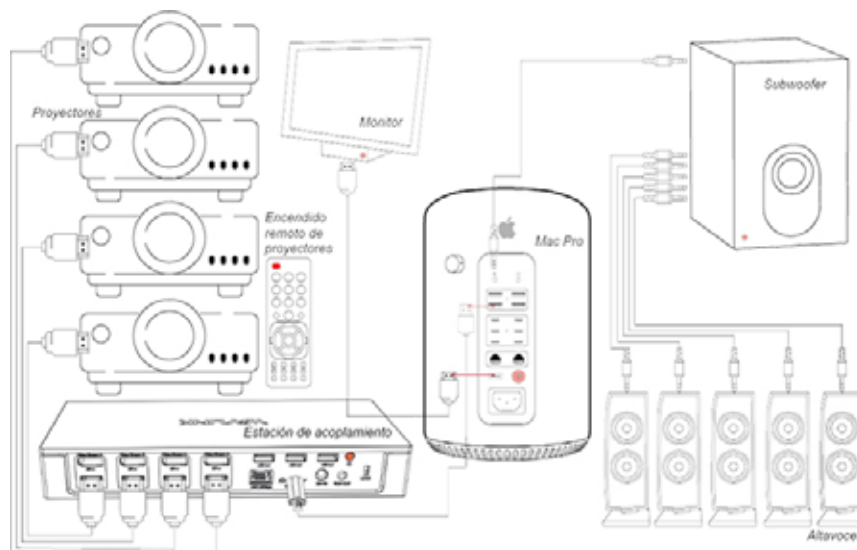
Figura 8

La imagen muestra la proyección de imagen satelital del área de CU, que procede de tres proyectores sobre una maqueta con la estación de acoplamiento funcionando



Figura 9

Maqueta Interactiva de Ciudad Universitaria



Nota: En la imagen se puede apreciar la conexión de todos los dispositivos empleados en el sistema de video mapping “Maqueta Interactiva de Ciudad Universitaria”; el dispositivo de la estación de acoplamiento abajo – izquierda, es el componente que se integró.

5. CONCLUSIONES

5.1 LECCIONES APRENDIDAS

La revisión exhaustiva de los adaptadores, la compatibilidad e interconexión entre dispositivos permitió profundizar y entender la manera en que interactúan los dispositivos empleados al comunicarse. Este conocimiento puede ser útil en la realización de las aplicaciones de video juego, realidad virtual, realidad aumentada, inteligencia artificial y, por supuesto, video mapping, enmarcadas en el área de graficación por computadora.

Es a partir del trabajo diagnóstico que se pudieron evaluar objetivamente los riesgos al intervenir el sistema de la maqueta interactiva, principalmente los relacionados a la compatibilidad hardware / software. En forma concluyente resultó muy eficiente buscar las alternativas de solución desde el análisis de hardware, como lo fue el reemplazo de conectores y adaptadores.

Es necesario promover que proyectos como el de la maqueta interactiva, en el cual participó un equipo de profesionales interdisciplinario, documente de forma muy precisa el proyecto, con la finalidad de que esa memoria técnica permita a acortar el tiempo de trabajo en el caso de intervenir el proyecto.

El éxito en la rehabilitación de la *maqueta interactiva* está basado en la exhaustiva revisión técnica de los componentes de hardware.

REFERENCIAS BIBLIOGRÁFICAS

- BenQ Ibérica, (s/f), *Introducción al USB-C: ¿Que es DisplayPort USB-C (DP en modo alt)?* <https://www.benq.eu/es-es/knowledge-center/knowledge/usb-c-introduction-what-is-dp-alt-mode.html>
- Club 3D | *USB C over ALT mode*. (n.d.). Club 3D. https://www.club-3d.com/es/technology/15/usb_c_over_alt_mode/
- Coughlin, T. M. (2017). *Digital storage in consumer electronics: The essential guide*. Springer.
- DGCS-UNAM (2017). *Inauguran en el Museo Universitario de Ciencias y Arte, Maqueta Interactiva de Ciudad Universitaria* (Número 2017/049). Dirección General de Comunicación Social de la Universidad Nacional Autónoma de México. https://www.dgcs.unam.mx/boletin/bdboletin/2017_049.html.
- DELL (2022, marzo). *Información general sobre Una o varias pantallas de video* | Dell México. <https://www.dell.com/support/kbdoc/es-mx/000052929/informaci%C3%B3n-general-sobre-una-o-varias-pantallas-de-video>
- EFE. (2021, agosto 14). *Así fue la inauguración de la maqueta del Templo Mayor en el Zócalo*. El Financiero. <https://www.elfinanciero.com/cdmx/2021/08/13/asi-fue-la-inauguracion-de-la-maqueta-del-templo-mayor-en-el-zocalo/>
- Estrella, A. (2021). *Maqueta de una pirámide de la antigua Tenochtitlán para la conmemoración del 500 aniversario*. El País. https://elpais.com/elpais/2021/12/26/album/1640546442_873467.html
- Fernández, Y. (2022, Noviembre 4). *DisplayPort vs HDMI: Cuáles son las diferencias*. Xataka.com; Xataka Basics. <https://www.xataka.com/basics/displayport-vs-hdmi-cuales-diferencias>
- Fischnaller F. (2018, junio 5-6). *New Media Exhibit at the Musée des Civilisations de L'Europe et de la Méditerranée*. In: Luigini, A. (eds) *Proceedings of the 1st International and Interdisciplinary Conference on Digital Environments for Education, Arts and Heritage*. EARTH 2018, Italy.
- Guerrero, H. (2021). *Vista general de la maqueta pirámide de la antigua Tenochtitlan en el Zócalo de Ciudad de México*. El País. <https://elpais.com/mexico/2021-08-13/los-ecos-de-la-caida-de-tenochtitlan-500-anos-despues.html>
- Instrumentic.info (s/f), *Mini DisplayPort Y thunderbolt: Compatibilidad Y USO – Faq-Mac*. (n.d.). <https://www.faq-mac.com/2014/08/mini-displayport-y-thunderbolt-compatibilidad-y-uso/>
- Instrumentic.info (s/f), *MiniDisplayPort - Todo sobre*. (2023, January 15). instrumentic.info. <https://instrumentic.info/es/apple/mini-displayport.html#gsc.tab=0>
- J. Lee, H. -h. Hsu, C. -h. Chen, X. Li and X. K. Cai. (2018, april). *System-level coexistence impact of USB type-C and type-A connectors with WiFi radio* [Conference session]. *EEE 26th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, San Jose, CA, USA.
- Roca, J. (2023, September 18). *HDMI: Version, hercios Y resolución de esta interfaz de video*. HardZone. <https://hardzone.es/tutoriales/rendimiento/hdmi-version-resolucion-hercios/>
- Nvidia (s/f) *¿Cuál es la diferencia entre DVI-I y DVI-D?*. <https://support.nvidia.eu/hc/es/articles/201073551--Cu%C3%A1l-es-la-diferencia-entre-DVI-I-y-DVI-D>

- Serrano, R. (2021, septiembre 15). *La historia de Tenochtitlán contada en video mapping en el zócalo de la CDMX*. Iluminet revista de iluminación. <https://iluminet.com/tenochtitlan-en-video-mapping/>
- Tecnopc. (2023, mayo 15). *¿Que es VGA?, la clásica interfaz • Tarjetas Gráficas PC*. Tarjetas Gráficas PC: La Plataforma Hardware para Computadoras. <https://tarjetasgraficasp.com/monitores/que-es-vga/>
- Universidad Nacional Autónoma de México (2017, febrero 2), *Ciudad Universitaria ya tiene una maqueta interactiva*. <https://www.fundacionunam.org.mx/pumarte/ciudad-universitaria-ya-tiene-una-maqueta-interactiva/>

Estrategias y herramientas para depuración de código en el back-end

Información del reporte:

Licencia Creative Commons



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

Para citar este reporte técnico:

Ramírez Romero, L. M. (2023). Estrategias y herramientas para depuración de código en el back-end. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (153 - 168).

<https://doi.org/10.22201/dgtic.ctud.2023.1.1.24>

Luz María Ramírez Romero

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

luzrr@unam.mx

ORCID: 0000-0002-8867-9374

Resumen:

El desarrollo de módulos y componentes libres de errores o bugs es uno de los objetivos medulares de la construcción de sistemas. Esto forma parte de entregar y/o liberar en ambientes de producción, sistemas de calidad, libres de problemas o vicios ocultos que impidan cubrir los requerimientos de los usuarios y que eviten cumplir los objetivos del sistema. Se exponen algunas estrategias y herramientas que el programador puede utilizar para depurar los sistemas de información, basados tanto en cambios de metodologías o paradigmas, como es la aplicación de *Scrum*, como en el uso de la Inteligencia Artificial para encontrar rutinas de código probadas, o incluso para la refactorización del código. Asimismo, se ha observado útil la integración de técnicas como el *Test Driven Development (TDD)*, que propone automatizar las pruebas de módulos individuales, además del uso de herramientas propias de *Laravel/Livewire*, como sus archivos de depuración, la herramienta *Tinker* y la barra de depuración de *Laravel*, o bien el uso de extensiones integradas al editor de código Visual Studio Code.

Palabras clave:

Herramientas de depuración, bug, error de codificación, asistentes de codificación basados en Inteligencia Artificial, pruebas unitarias.

1. INTRODUCCIÓN

Los programadores introducen defectos en el código que desarrollan o que actualizan y esto ocurre de manera involuntaria porque errar y equivocarse es una situación inherente a la naturaleza humana. El programador al cometer un error, introduce defectos en el código. Estos defectos o desperfectos harán que el sistema falle eventualmente, producirán malestar en los usuarios, problemas en el dominio o contexto donde se ejecuta el sistema y generarán un costo para la organización.

El especialista en TI José M. Huerta, menciona en su *blog Gestión de TI*, las siguientes causales de defectos en el *software*, mismas que propone analizar para intentar reducir los defectos y fallos en las aplicaciones desarrolladas (Huerta, J., 2028): Los programadores introducen defectos por no comprender los requerimientos de los usuarios, o al encontrarse con errores de diseño que llevan a un planteamiento equivocado o confuso.

Otros defectos son los directamente atribuibles al código por sí mismo, consistentes en los llamados *errores de dedo*, que agregan caracteres extras o equivocados al escribir el código, o que son ocasionados por no conocer con precisión la sintaxis del lenguaje, o *framework*, o marco de trabajo de programación

También se producen defectos con los consecuentes fallos en el software por cambios de versión de la tecnología de soporte o tecnología subyacente, con la que se conforman los sistemas.

Hay varias estrategias para contrarrestar y disminuir la introducción de defectos en los sistemas de información, tales como:

- El uso de herramientas que ayuden al programador a detectar defectos de sintaxis de manera temprana.
- Codificar utilizando un conjunto de buenas prácticas, configuradas a través de la experiencia acumulada en el mundo de la ingeniería de software, que permitan hacer más fácil la depuración del código y que además faciliten su actualización y evolución posterior.
- Contar con rutinas, funciones y componentes ya probados, que resuelvan problemas menores y que puedan reutilizarse.
- Disponer de herramientas que permitan detectar puntos de mejora y refactorización¹ del código.

En este reporte técnico se tratará el tema de la producción de *software* de calidad, dentro de la metodología y enfoque ágil de desarrollo de sistemas *Scrum*. También se hablará sobre la construcción del *back-end*² de sistemas o aplicaciones desarrolladas bajo los marcos de trabajo o *frameworks* *Laravel-Livewire*, lenguaje de programación *PHP*, a través del editor de código *Visual Studio Code (VSC)*³.

¹ *Refactorización* consiste en re-estructurar o re-escribir algunas partes de código para hacerlo más comprensible, mejorar su estructura, eliminar código muerto, con el fin de hacerlo más mantenible y para no introducir defectos durante su actualización posterior.

² *Back-end* es la programación del lado del servidor para implementar la lógica de negocio (Luna et al., 2018). En el *Modelo Vista-Controlador (MVC)* del *framework Laravel*, el *Back-end* lo constituyen los controladores de la aplicación.

³ *Visual Studio Code (VSC)* es uno de los editores de código fuente, ampliamente conocido y utilizado por programadores en todo el mundo. Es una herramienta gratuita

2. OBJETIVOS

- Revisar cómo contribuye la metodología del enfoque ágil *Scrum* en la obtención temprana de los resultados de pruebas sobre el *software* en desarrollo para aplicar las técnicas y herramientas de depuración del código.
- Compartir estrategias para la depuración de código en el *back-end* de una aplicación bajo el paradigma orientado a objetos de *PHP-Laravel*.
- Interpretar la información que proporcionan las herramientas de depuración de código en los *frameworks* o marco de trabajo *Laravel-Livewire*.
- Proponer nuevos esquemas para producir código con menos defectos basados en tecnologías actuales como las propuestas por los modelos de Inteligencia Artificial (*IA*).

3. CICLO DE VIDA CLÁSICO DE DESARROLLO DE SOFTWARE VS. EL ENFOQUE ÁGIL DE SCRUM.

La construcción de aplicaciones de *software* ocurre dentro de un proceso metodológico y estructurado, denominado *Ciclo de Vida Clásico* del desarrollo de *software* (*Ingeniería de Software*). Este proceso inicia con la fase de *Análisis*, que indica que la etapa más importante del desarrollo de sistemas, no es la programación, sino que consiste en comprender el objetivo del sistema y las necesidades o requerimientos del usuario.

Partiendo de ese *Análisis*, continúa el *Diseño del software*, que puede ser conformado a través de diagramas y prototipos basados en satisfacer las necesidades del usuario. A continuación del *Diseño*, sigue la etapa de *Programación*, en la que se implementan los prototipos que responderán a las necesidades del usuario. Posteriormente, está la etapa de *Pruebas* que consiste en que un equipo de personas diferentes al equipo de desarrolladores, revise el sistema y sus módulos para asegurar que todas las piezas desarrolladas cumplan con los requerimientos del usuario. Finalmente, llega la etapa de *Liberación* o *Despliegue* del *software*, que colocará el sistema en un ambiente de producción, lo que permitirá a los usuarios ocupar el sistema para lograr sus objetivos iniciales.

A pesar de que el desarrollo de *software* ocurre dentro de este *Ciclo de Vida Clásico*, metodológico y estructurado, en 1968, durante la primera conferencia sobre desarrollo de *software* de la *OTAN*, se definieron una serie de problemas denominados como la *Crisis del Software*, que es un término que se refiere a que comúnmente los sistemas a la medida se concluían mucho después del tiempo planeado y que al final los sistemas no cubrían las necesidades de los usuarios (Noriega, M., 2015).

Por lo tanto, en 1995 un grupo de ingenieros de *software*, entre los que destaca Jeff Sutherland, crean el *Manifiesto Ágil*, a partir del cual se desarrolla la *Metodología Scrum*, cuya principal aportación es la creación del *software* en varios ciclos cortos. En cada ciclo, se elige desarrollar algunas de las funcionalidades que requiere el usuario, de tal manera que al finalizar cada uno, se obtiene una parte del sistema probada y funcionando, lista para que el usuario la revise y determine si es lo que esperaba.

El otro gran cambio que introduce *Scrum* es la realización de las pruebas en cada ciclo de desarrollo, con lo cual se obtiene rápidamente retroalimentación sobre los componentes desarrollados en el ciclo, para

que el código sea depurado, a diferencia del *Ciclo de Vida Clásico* de desarrollo de *software*, en el que las pruebas comienzan hasta que es concluido todo el sistema.

Las técnicas y herramientas documentadas en este texto dan soporte a la etapa de *Desarrollo y Pruebas* del *software*.

En la tarea de depuración de *software*, no se ha encontrado en la literatura de ingeniería de *software*, ni en la de métodos ágiles, alguna metodología definida para llevar a cabo la depuración del *software* (Trigás, G. (2012). Dependiendo de las herramientas de desarrollo seleccionadas, se utilizan diversas herramientas y técnicas de depuración.

A continuación, se abordarán las que son aplicables al desarrollo *back-end* (es decir, de la implementación de las reglas de negocio de las aplicaciones de *software* a la medida), para los *frameworks Laravel/Livewire*, lenguaje *PHP* y para el editor de código *VSC*.

4. USO DE ASISTENTES DE CODIFICACIÓN BASADOS EN INTELIGENCIA ARTIFICIAL (IA)

Estas herramientas surgen a partir del año 2021, y son las más novedosas y recientes para ayudar al desarrollo y depuración de código. Estas piezas de *software* inteligentes, sugieren a los programadores humanos, desde simples líneas de código individuales, hasta pequeñas rutinas que resuelven problemas sencillos, específicos y granulares (*IA para desarrolladores*).

El primer asistente de código conocido es **Codex**, que fue desarrollado por la empresa *OpenAI*, que de acuerdo con la definición que brinda su sitio web, *Codex* es capaz de traducir el lenguaje natural a líneas de código (OpenAI, 2023). Este es un hecho importante en el mundo del desarrollo de aplicaciones porque con ayuda de estos asistentes, los programadores pueden crear módulos y componentes con menos defectos, además de que estos asistentes ayudan a desarrolladores tanto principiantes, como expertos.

A partir de *Codex*, se desarrolló el complemento *GitHub Copilot*, que se puede instalar en varios editores de código tales como *VSC*, *Vim*, *JetBrains* o *Neovim*, potenciando estos editores con el soporte mencionado para los programadores. Otros asistentes de código son *Blackbox AI Code Generator* y *GitHub Copilot*.

Blackbox AI Code generator. Se trata de un asistente de programación de uso gratuito, que se le instala al *VSC*. El programador le puede preguntar al *Blackbox* por rutinas de programación específicas, por ejemplo, cómo conectarse a un servidor de base de datos específico o cómo ordenar una lista de números de mayor a menor, tras lo cual propondrá el código correspondiente. *Blackbox* realmente buscará en diversas fuentes de Internet dichas rutinas de código, tales como *Stack Overflow* y brindará los resultados de la búsqueda como propuestas. El programador revisará y adaptará la rutina que mejor convenga al problema a resolver (*Blackbox AI*, s.f.). Un ejemplo de funcionamiento de *Blackbox* se muestra en el Anexo A.

GitHub Copilot. Esta extensión también está potenciada con *IA*, y basa su núcleo en *OpenAI Codex* que es capaz de sugerir líneas de código, construir expresiones regulares, funciones completas y proponer comentarios para documentar los programas. *Copilot* realiza sus búsquedas de código en los repositorios de *GitHub*. El programador se encargará de revisar cuál opción conviene más al problema y adaptará el código a su módulo. *Copilot* es una extensión de pago, ya que sus funciones de *IA* le permiten proponer código más adaptado al contexto que tiene en el editor de *VSC* en ese momento (GitHub, 2022).

5. INSTALACIÓN DE OTROS COMPLEMENTOS EN EL EDITOR DE CÓDIGO VISUAL STUDIO CODE (VSC)

El VSC cuenta actualmente con la función de detectar el lenguaje de programación de cada archivo, a través de la extensión del mismo. Por ejemplo, los archivos con extensión *.blade.php* son asociados a código *HTML* con directivas del motor blade de Laravel. Los archivos con extensión *.php* son asociados con código *PHP*. Una vez detectado el lenguaje en el que está escrito cada archivo, VSC marca de manera gráfica con color rojo, los errores de sintaxis para que el programador los identifique y los pueda corregir rápidamente.

Asimismo, VSC es capaz de identificar cuando se abre una función, o ciclo, o bucle de programación y marca en rojo cuando no se detecta el cierre correspondiente para su corrección. Además de esta funcionalidad propia del VSC, se pueden agregar complementos o extensiones para apoyar la tarea de programación de componentes y sistemas, tales como los siguientes:

Run on Save. Revisa estilos de codificación y eficiencia de las líneas codificadas en el momento de guardar el archivo del módulo, de tal manera que elimina cosas innecesarias, como líneas en blanco excesivas y propone cierto grado de refactorización en el código, cambiando las líneas de código, o eliminando variables innecesarias, por ejemplo.

PHP Intelephense. Realiza autocompletado de código, mostrando en una pequeña ventana emergente diferentes opciones de selección, ante una instrucción PHP. Lo anterior apoya al programador para tener a la mano la estructura y sintaxis de funciones básicas de PHP.

Laravel Extra Intellisense. Esta extensión proporciona un autocompletado de las rutas de Laravel.

Laravel Goto Controller. Permite abrir controladores de Laravel desde el archivo de rutas *web.php*.

6. EL ANÁLISIS DE LA BITÁCORA DE ERRORES DE LARAVEL

Los proyectos del marco de trabajo *Laravel*, cuentan con archivos de bitácora de errores, dentro de la carpeta *storage/logs* del proyecto, en dónde típicamente se registran los errores ocurridos en tiempo de ejecución. Estos archivos de errores son *laravel.log*, *errors.log* y *debug.log*.

El programador de *back-end* debe revisar estos archivos de bitácora, los cuales brindarán información más específica de los errores producidos, junto con mensajes del *framework*, que le ayudarán a determinar la causa de la falla (problemas en la conexión con la base de datos, nombres de campos o de variables mal nombradas o que no existen, falta de instalación de componentes, entre otros).

Asimismo, el programador que utiliza *Laravel* puede enviar mensajes y el contenido de variables, arreglos, clases y colecciones con la directiva *Log::debug*, mismos que se registrarán en el archivo *laravel.log*, para su revisión y análisis correspondiente. (Aguirre, S., 2022) En el Anexo B se presenta un resumen del uso de la bitácora de *Laravel*.

7. DIRECTIVA DD (DUMP AND DIE) DE LARAVEL

La función `dd` ayuda al programador a revisar el contenido de la variable especificada en el navegador web y además detiene la ejecución. La siguiente figura muestra cómo imprimir el contenido de la variable `$this->sesion` y detener la ejecución del programa.

Figura 1

Código para imprimir el contenido

```
dd($this->sesion);
```

Se incluye en el Anexo C un ejemplo de la respuesta de la directiva `dd` de *Laravel*.

8. USO Y ANÁLISIS DE LA BARRA DE DEPURACIÓN DE LARAVEL

Este elemento se muestra en la parte inferior del navegador web, una vez instalado, y proporciona información valiosa sobre el tiempo de ejecución de las consultas a la base de datos por cada interfaz de la aplicación, así como el número y nombre de los modelos (tablas) consultadas de la base de datos.

Es un paquete que debe instalarse en el proyecto de *Laravel* para poder hacer uso de sus características y ocuparlas para la depuración del código.

La barra de depuración de *Laravel* se muestra en el Anexo D.

9. TINKER, LA CONSOLA DE LARAVEL

Dice la definición técnica que *Tinker* es una consola de *Laravel* (Blastcoding, s.f. y Aguirre, S. 2022), que permite revisar de manera inmediata y directa los resultados a obtener de una instrucción o conjunto de instrucciones y así poder integrar el código a los controladores y componentes del *back-end*, con la certeza de no haber insertado defectos en el código, puesto que el fragmento de código ya fue probado en la consola *Tinker*.

Usos recomendados de la consola *Tinker*:

- Probar algún método o clase propios de *Laravel*, a fin de incluirlos en los controladores que se estén desarrollando.
- Interactuar con cualquier clase que se haya programado.
- Revisar funcionalidades de librerías que se hayan integrado a *Laravel*, como el manejo de fechas a través de la librería (Carbon, s.f.).
- Revisar la interacción entre los controladores de *Laravel* con la base de datos, ejecutando las sentencias de su *ORM* (*Object Relational Mapping*) *Eloquent*, *SQL* nativo o a través del constructor de consultas *Query Builder*.

10. PRUEBAS UNITARIAS

Otra estrategia para la depuración de código es que el programador desarrolle código para probar de manera automática su programación.

Esta es una estrategia denominada *Test-Driven Development (TDD)* (Paradigma, s.f.), que permite ejecutar las pruebas automatizadas cuantas veces sea necesario, cada vez que el código es re-factorizado, mejorado, modificado o actualizado. Cabe mencionar que las pruebas unitarias no reemplazan otro tipo de pruebas como las funcionales de caja negra, pruebas de integración o pruebas de regresión y tampoco sustituyen a los probadores o *testers*.

El enfoque *TDD* también propone alterar el *Ciclo de Vida tradicional* de desarrollo de *software*, indicando que primero se deben definir y automatizar las pruebas de un módulo y después el desarrollador o programador debe construir su módulo, haciendo que éste vaya cumpliendo con las pruebas definidas para él. Una vez que se ha terminado de codificar el módulo, se ejecuta la prueba automatizada, que indicará si se está cumpliendo con los objetivos para los que fue construido (Beck, K, 2003).

Los programas correspondientes a las pruebas unitarias deben colocarse en el directorio *tests/Unit* de la estructura del proyecto de *Laravel*.

Para una referencia rápida sobre las pruebas unitarias en *Laravel* se sugiere revisar el Anexo E.

11. RESULTADOS

Las herramientas y estrategias revisadas se han aplicado en el desarrollo de sistemas y aplicaciones a la medida por el grupo de desarrolladores de la Subdirección de Sistemas Integrados de la DGTIC. Se han obtenido resultados de mejora en los tiempos de desarrollo y codificación, así como la reducción del número de incidencias atribuidas a errores por código.

Uno de los resultados más importantes es el cambio de metodología para el desarrollo de *software*. La experiencia obtenida brinda como resultado que es mejor el desarrollo de aplicaciones bajo el enfoque ágil, que enmarcar el proyecto en el *Ciclo de Vida Clásico*.

Otro hallazgo importante, es la aplicación de herramientas de Inteligencia Artificial en el desarrollo y depuración de código. Mediante el uso de estas herramientas, se puede incluir en el código de los sistemas a la medida, la implementación de algoritmos genéricos en el lenguaje de programación elegido. Otra ventaja adicional es que esas implementaciones ya han sido utilizadas y probadas en otros sistemas de código abierto publicados en repositorios tales como *GitHub*. Entre los algoritmos genéricos nombrados se puede mencionar los de ordenamiento y las rutinas para quitar espacios en blanco, entre otros.

Tradicionalmente primero se construía el *software* y después se hacían las pruebas al mismo. Actualmente es más útil involucrar al equipo de pruebas desde el inicio del proyecto, de tal manera que se diseñan las pruebas, junto con la etapa de análisis. Posteriormente, se construyen los módulos de la aplicación; y una vez construidos los módulos, se pueden probar de manera inmediata con las pruebas unitarias previamente construidas. Con ello, se sabe rápidamente si los módulos de manera individual, cumplen con los objetivos y requerimientos del usuario. Además, este enfoque de *TDD* permite que cada vez que se realicen cambios en los módulos de *software*, se pueda correr de manera automática las pruebas unitarias para saber si los módulos afectados siguen brindando los resultados esperados.

Asimismo, durante el proceso de codificación se ocupan herramientas como *Tinker*, que permiten al desarrollador probar paso a paso el código escrito y saber si se están obteniendo los resultados esperados.

Todas estas técnicas apoyan un proceso eficiente de desarrollo de *software*, dentro de un círculo virtuoso que tiene como propósito final, la producción de *software* de calidad en tiempos competitivos.

Adicionalmente, se observa que la elección de un editor de código no es una tarea trivial, ya que actualmente existen editores muy especializados, con extensiones que pueden agregarse a la funcionalidad básica del editor y que permite ir analizando el código, al tiempo en que es escrito, con la finalidad de detectar de manera pronta y expedita la introducción de defectos sintácticos en el código. Incluso hay extensiones que ofrecen cierto grado de refactorización del código, proponiendo líneas más eficientes de código.

12. CONCLUSIONES

Debido a que se sabe de antemano que todo código tiene defectos, que pueden traducirse en fallas de los sistemas, es obligatorio que todo grupo de desarrollo de aplicaciones cuente con una serie definida de estrategias y herramientas que apoyen la depuración del código, además de contar con un equipo de personas especializadas en pruebas de *software* que trabaje de la mano con ingenieros de requerimientos y desarrolladores de *software*, para apoyar la labor de depuración del código.

Dentro de la actividad de depuración de código, es importante integrar los cambios de metodologías y de paradigmas, tales como el enfoque ágil desde la organización del proyecto, la obtención de resultados de pruebas y detección de defectos de forma pronta, dentro del ciclo de vida del desarrollo de *software*.

Una de las principales ventajas que se ha detectado en el uso del enfoque ágil es la obtención y entrega al usuario final, de módulos que funcionan en tiempos muy cortos de hasta 1 o 2 semanas, lo cual refuerza la credibilidad en el equipo de desarrollo de *software* y permite que el usuario brinde retroalimentación sobre dichos módulos.

Otra técnica que ha resultado importante integrar en la depuración de código es el *TDD*, que permite probar automática y rápidamente los módulos del sistema de manera individual, cuantas veces sea necesario, lo que resulta útil cada vez que se cambia algo en el *software*, debido a alguna actualización, mejora o por atender nuevos requerimientos de los usuarios.

Asimismo, es importante considerar el probar e incorporar de manera continua y permanente nuevas herramientas en la construcción y depuración del código fuente de módulos y aplicaciones, en el marco del lenguaje de programación y *frameworks* seleccionados para el desarrollo del *software*.

Un ejemplo de este análisis y búsqueda constante de herramientas y técnicas de depuración, es la aplicación de herramientas en el campo de la *IA*, la cual está apoyando varios campos de las Tecnologías de Información y uno de ellos es el desarrollo de *software*. Resulta muy útil contar con este tipo de soporte tecnológico, ya que con ello se acortan los tiempos de análisis y depuración de código, pues estas herramientas novedosas proponen rutinas sencillas de código ya probado en otros sistemas de *software* abierto, o incluso sugieren la refactorización o re-escritura de código, en aras de lograr eficiencia.

Asimismo, no se debe soslayar las herramientas tradicionales de depuración que también resultan útiles en la detección de defectos y que permiten a los desarrolladores revisar los resultados de líneas de código, de manera parcial, para integrarlas al código, una vez que se ha comprobado que logran objetivos atómicos o más granulares, que se suman al objetivo global del módulo y del sistema en su conjunto.

Todo paradigma, técnica y herramienta enfocada en la depuración de código, debe ser considerada para lograr la confiabilidad en el sistema desarrollado, especialmente para asegurar que el sistema brindará los resultados requeridos por los usuarios, para superar la llamada *Crisis del Software*.

REFERENCIAS BIBLIOGRÁFICAS

- Aguirre, S. (2022). *Laravel Curso completo*. Ra-Ma Editorial.
- Beck, K. (2003). *Test-Driven Development by Example*. Addison-Wesley Signature Series.
- Blackbox IA, (s.f.). *Blackbox*. Recuperado 21 de septiembre de 2023, de <https://www.useblackbox.io/>
- [B]lastcoding.com (s.f.). *Qué es y cómo usar Laravel Tinker*. (1556204739). Blastcoding. https://blastcoding.com/?post_type=post&p=2163
- Carbon (s.f.). *A simple PHP API extension for DateTime*. Recuperado 22 de septiembre de 2023, de <https://carbon.nesbot.com/docs/>
- Huerta, J. (2028). *Coste de los errores en proyectos de software*. Recuperado 21 de septiembre de 2023, de <https://josehuerta.es/gestion/proyectos/calidad/coste-de-los-errores-en-proyectos-de-software>
- Gamarra, F. (2023). *IA para desarrolladores*. Editorial RedUsers.
- GitHub (Director). (2022, noviembre 15). *What is GitHub Copilot?* <https://www.youtube.com/watch?v=lqXNhakuwVc>
- Luna, F., Millahual, C. P., & Iacono, M. (2018). *PROGRAMACION WEB Full Stack 13 - PHP: Desarrollo frontend y backend - Curso visual y práctico*. Editorial RedUsers.
- Noriega, M. (2015). *El proceso de desarrollo de software*. Editorial IT Campus Academy.
- OpenAI. (2023). *OpenAI Codex*. Recuperado 21 de septiembre de 2023, de <https://openai.com/blog/openai-codex>
- Pantaleo, G y Rinaudo, L. (2015). *Ingeniería de Software*. Alfaomega editores.
- Paradigma (s.f.). *TDD como metodología de diseño de software*. Recuperado 22 de septiembre de 2023, de <https://www.paradigmadigital.com/dev/tdd-como-metodologia-de-diseno-de-software/>
- Trigás, G. (2012). *Metodología Scrum: El proceso de desarrollo de software*. Editorial Universitat Oberta de Catalunya.

ANEXO A. EJEMPLO DE FUNCIONAMIENTO DE BLACKBOX

En las imágenes 2 y 3, se muestra como ejemplo una pregunta que se le hizo a la extensión de *Blackbox* que se instaló en el VSC. Se le pidió el código para ordenar una lista de número de mayor a menor en lenguaje *PHP* y la respuesta fue una explicación de cada paso que debe realizar el programador para lograr el resultado deseado, así como el código completo.

Figura 2

Ejemplo de forma de consultar a Blackbox y primera parte de la respuesta del software

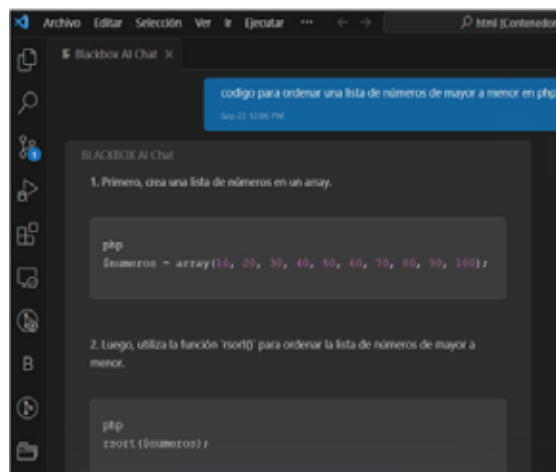
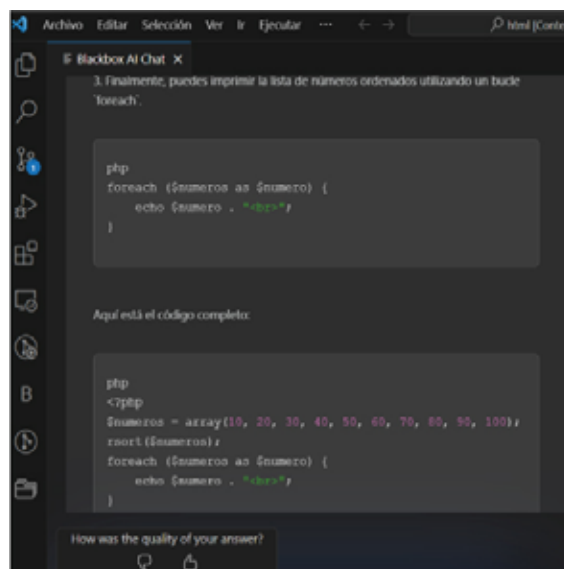


Figura 3

Parte 2 de la respuesta de Blackbox



ANEXO B. RESUMEN DE USO DE LA BITÁCORA DE LARAVEL

Para utilizar la directiva `Log::debug` primero hay que importar la clase `Log` de la siguiente manera:

```
use Illuminate\Support\Facades\Log;
```

Un ejemplo de uso de `Log::debug` es el siguiente: imprimir el mensaje "Id de sesión =" y enseguida se imprimirá el contenido de la variable `$id_sesion`:

```
Log::debug("Id de sesion = " . $id_sesion);
```

En la siguiente figura se muestra un ejemplo de un error registrado en la bitácora de *Laravel*, ocasionado por la falta de instalación del paquete *DomPDF* que ayuda a generar archivos en formato *PDF* desde *Laravel*.

Figura 4

Ejemplo de error en la bitácora `errors.log` de *Laravel*

```
storage > logs > errors.log
1 [2023-09-13 18:53:44] local.ERROR: Class "Barryvdh\DomPDF\ServiceProvider" not found {"exception":"[obj
2 [stacktrace]
3 #0 /var/www/html/vendor/laravel/framework/src/Illuminate/Foundation/ProviderRepository.php(142): Illumi
4 #1 /var/www/html/vendor/laravel/framework/src/Illuminate/Foundation/ProviderRepository.php(61): Illumin
5 #2 /var/www/html/vendor/laravel/framework/src/Illuminate/Foundation/Application.php(745): Illuminate\F
6 #3 /var/www/html/vendor/laravel/framework/src/Illuminate/Foundation/Bootstrap/RegisterProviders.php(17)
7 #4 /var/www/html/vendor/laravel/framework/src/Illuminate/Foundation/Application.php(261): Illuminate\F
8 #5 /var/www/html/vendor/laravel/framework/src/Illuminate/Foundation/Http/Kernel.php(186): Illuminate\F
9 #6 /var/www/html/vendor/laravel/framework/src/Illuminate/Foundation/Http/Kernel.php(170): Illuminate\F
10 #7 /var/www/html/vendor/laravel/framework/src/Illuminate/Foundation/Http/Kernel.php(144): Illuminate\F
11 #8 /var/www/html/public/index.php(51): Illuminate\Foundation\Http\Kernel->handle(Object(Illuminate\F
12 #9 {main}
13 }
```

ANEXO C. EJEMPLO DE LA RESPUESTA DE LA DIRECTIVA DD DE LARAVEL

Como ejemplo de uso de `dd` se tiene la siguiente línea de código, en la cual se pide a *Laravel* imprimir el contenido de la variable `$this->sesión`, que contiene el resultado de consultar el registro que corresponda al `$this->id_sesion` de la tabla *Sesion*.

```
$this->sesion = Sesion::find($this->id_sesion);
dd($this->sesion);
```

La respuesta que se obtiene en el navegador web se muestra en la figura 5. En dicha salida se observa información diversa, como la base de datos a la que se está conectando (*Postgresql – pgsq1*), la tabla sesión, la llave primaria de la tabla y de manera especial se muestra el contenido de los atributos que conforman la variable *\$this->sesion*, que consiste en un arreglo de 8 atributos (*id_sesion, id_tipo_sesion, fecha, id_sede, codigo, activa* y dos atributos propios de *Laravel: created_at* y *updated_at*, que son la fecha y hora de creación y actualización del registro).

Figura 5

Respuesta del dump and die (dd) de Laravel

```
App\Models\Sesion {#658 ▼ // app/Http/Livewire/Consultar
  #connection: "pgsq1"
  #table: "sesion"
  #primaryKey: "id_sesion"
  #keyType: "int"
  +incrementing: true
  #with: []
  #withCount: []
  +preventsLazyLoading: false
  #perPage: 15
  +exists: true
  +wasRecentlyCreated: false
  #escapeWhenCastingToString: false
  #attributes: array:8 [▼
    "id_sesion" => 2
    "id_tipo_sesion" => 1
    "fecha" => "2023-10-09"
    "id_sede" => 1
    "codigo" => "555555"
    "activa" => true
    "created_at" => "2023-09-12 12:42:21.075096"
    "updated_at" => null
  ]
  #original: array:8 [▶]
  #changes: []
  #casts: []
  #classCastCache: []
  #attributeCastCache: []
}
```

ANEXO D. BARRA DE DEPURACIÓN DE LARAVEL

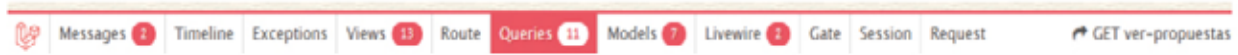
Para la instalación se debe ejecutar el siguiente comando en la consola:

```
composer require barryvdh/laravel-debugbar
```

En la siguiente figura se muestra la barra de depuración de *Laravel*.

Figura 6

Barra de depuración de Laravel



La barra de depuración abierta y con la pestaña de "Queries" seleccionada se pueden apreciar en la siguiente figura, donde se observan 11 consultas a la base de datos y sus correspondientes sentencias SQL y del lado derecho de cada renglón, se aprecia el tiempo de ejecución de cada consulta. De esta manera, el programador se puede dar cuenta si necesita re-escribir algún *query* para optimizar su tiempo de respuesta y hacer más eficiente su aplicación.

Figura 7

Barra de depuración abierta con la pestaña Queries



ANEXO E. PRUEBAS UNITARIAS EN LARAVEL

A continuación, se muestran una serie de comandos para la consola relacionados con las pruebas unitarias.

Para crear una prueba unitaria se ejecuta el comando siguiente de *php artisan*:

```
PS C:\xampp\htdocs\peliculas\peliculas> php artisan make:test AcercaDeTest --unit
```

Para ejecutar todas las pruebas automatizadas que se encuentren en la carpeta *tests* de *Laravel* se ejecuta el siguiente comando:

```
PS C:\xampp\htdocs\peliculas\peliculas> php artisan test
```

Para ejecutar solamente las pruebas unitarias que se encuentren en el directorio *tests/Unit* de *Laravel*, se corre el siguiente comando:

```
PS C:\xampp\htdocs\peliculas\peliculas> ./vendor/bin/phpunit
```

Para ejecutar únicamente una prueba (en este caso la prueba llamada “acercade”), se ejecuta el siguiente comando:

```
PS C:\xampp\htdocs\peliculas\peliculas> php artisan test --filter acercade
```

En la figura 7 se muestra el código de una prueba unitaria básica. En este ejemplo, se observa el uso de las clases *Tests\TestCase*, que contienen una serie de métodos para probar funciones específicas de *Laravel*.

Después se aprecia cómo se pide la carga de la página con la ruta */usuarios* con la directiva:

\$this->get('/usuarios')

Una vez solicitada la carga de la página, con el método *assertStatus*, se revisa si el código que devuelve la carga de la página es un código 200, que equivale a la carga exitosa. Si se puede afirmar que el código devuelto es igual a 200 (código de éxito), entonces la prueba es exitosa y comprueba que se pudo cargar correctamente la página solicitada.

Después se comprueba con el método *assertSee* que la página incluya en su contenido la palabra *'Usuarios'*.

Figura 7

Ejemplo del código de una prueba unitaria básica en *Laravel*

```
tests > Unit > UsersModuleTest.php > UsersModuleTest > test_carga_un_nuevo_usuario
1  <?php
2
3  namespace Tests\Unit;
4
5  use Tests\TestCase;
6
7  class UsersModuleTest extends TestCase
8  {
9      public function test_carga_pagina_usuarios(): void
10     {
11         $response = $this->get('/usuarios')
12             ->assertStatus(200)
13             ->assertSee('Usuarios');
14     }
15
```

GLOSARIO

Back-end. En el *Modelo Vista-Controlador (MVC)* se le conoce como *back-end* a todos los complementos y funciones que se colocan en el servidor y que se encargan de implementar la lógica de negocio. Su contraparte lo constituye el *front-end* o interfaz de usuario. Parte del *back-end* son los manejadores de bases de datos que almacenan los datos de las aplicaciones.

Blackbox AI Code generator. Es un asistente de codificación de uso libre que permite codificar 10 veces más rápido. Permite convertir cualquier pregunta del lenguaje natural a código de programación.

Carbon. Librería para la manipulación y operaciones con fechas.

Codex. Asistente de codificación basado en Inteligencia Artificial, desarrollado por la empresa *OpenAI*.

Complementos/extensiones de Visual Studio Code. Son unidades de *software* que se instalan al editor de código (*Visual Studio Code*) para agregar funcionalidad diversa, apoyando la labor de programación.

Defecto. Desperfecto en un componente que puede causar que el mismo falle en sus funciones.

Depuración. Acción de diferentes modalidades de revisión para la detección de defectos con el propósito de efectuar su corrección.

Eloquent. Es el *ORM (Object-Relational Mapping)* de *Laravel* que permite la interacción entre el sistema desarrollado en *Laravel* y la base de datos.

Error. Acción humana que produce un resultado incorrecto.

Falla. Manifestación visible de un defecto.

Framework de programación. Es un conjunto de herramientas para el desarrollo de sistemas o aplicaciones de manera más rápida y eficiente.

Front-end. En el *Modelo Vista-Controlador (MVC)* se le llama *front-end* a todos los elementos que se encuentran en el cliente y que son el punto de contacto con el usuario final. A través del *front-end*, el usuario ingresa datos para obtener información por parte del *back-end*.

GitHub. Es un repositorio del código fuente de diversas aplicaciones que es publicado por la comunidad *GitHub* con el propósito de compartirlo (si se coloca como público) y/o para obtener aportaciones en funcionalidad, en documentación y que además permite controlar las versiones de las aplicaciones que ahí se alojan.

GitHub Copilot. Es un asistente para desarrollo de código cuyo núcleo está basado en *OpenAI Codex* y que busca rutinas de código en el repositorio *GitHub*.

Inteligencia artificial. Es una disciplina tecnológica cuyo propósito es desarrollar e implementar algoritmos que permitan emular algunas capacidades intelectuales humanas, a partir de mecanismos de entrenamiento de los algoritmos y con el objetivo de resolver problemas.

Laravel. Es un *framework* o marco de trabajo de código abierto, basado en el *Modelo Vista-Controlador*, y soportado por el lenguaje de programación *PHP*, que brinda organización, estructura y ciertas funcionalidades para el desarrollo eficiente de sistemas y aplicaciones informáticas para la web.

Postgresql. Es un manejador de bases de datos basado en el modelo relacional, en el cual los datos se estructuran en entidades y tuplas, que se relacionan a través de llaves foráneas que permiten estas relaciones.

Pruebas funcionales de caja negra. Son pruebas que no toman en cuenta cómo está construida una aplicación, sino que se basa solamente en que a ciertos datos de entrada, se deben obtener un determinado conjunto de datos de salida.

Pruebas de integración. Son pruebas en las que se revisa que los diferentes módulos trabajen correctamente en su conjunto, satisfaciendo los requerimientos del usuario.

Pruebas de regresión. Estas pruebas se realizan sobre todo el sistema cada vez que alguna parte del código se modifica para aumentar o modificar la funcionalidad.

Query. Se refiere a las sentencias que se utilizan para consultar o interrogar a la base de datos.

Query Builder. Es una forma alternativa de métodos que ofrece *Laravel* para crear y ejecutar consultas a la base de datos.

Stack Overflow. Se trata de un foro en internet en el que los programadores publican problemas de programación y que otros programadores solucionan. Todas las conversaciones y código quedan disponibles para toda la comunidad y usuario interesado, lo que va conformando una base de conocimientos sobre programación en diferentes lenguajes y marcos de trabajo.

Tester (probador). Es un rol informático dedicado a planificar y llevar a cabo pruebas de *software* para comprobar que las aplicaciones cumplan con los requerimientos funcionales y no funcionales.

Visual Studio Code. Aplicativo especializado para escribir y editar código de *software*.

Implementación de técnicas de observabilidad en el Centro de Monitoreo de la Red

Información del reporte:

Licencia Creative Commons



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

Para citar este reporte técnico:

Ramírez Fernández, E. R. (2023). Implementación de técnicas de observabilidad en el Centro de Monitoreo de la Red. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (169 - 184).

<https://doi.org/10.22201/dgtic.ctud.2023.1.1.17>

Esteban Roberto Ramírez Fernández

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

esteban.ramirez@unam.mx

ORCID: 0000-0002-2169-6233

Resumen:

El presente reporte muestra el resultado de la evaluación de aplicaciones orientadas al análisis de información, para conformar una arquitectura de monitoreo que contribuya al análisis de eventos que afectan la operación de la red de datos al interior de la UNAM operada por el Centro de Monitoreo de la Red (NOC) de RedUNAM. Se incorpora el concepto de "observabilidad" que permite incrementar la visibilidad a partir de la concentración de información proveniente de bitácoras y sistemas de monitoreo, entre otros, para lo cual se considera cualquier fuente de información relevante para la operación de las redes, y se crean tableros de control con gráficas que permiten la consulta en tiempo real de los datos recolectados. La integración de software emplea aplicaciones de monitoreo activo, pasivo, e información de protocolos de comunicación para el análisis de incidentes en la red. El desarrollo se enfoca en la problemática de identificación y solución de incidentes de comunicaciones en los enlaces de la red de datos de la UNAM, para lo que se concentran datos de *hardware*, *software* y de aplicaciones operativas que se comunican al interior de la UNAM y a Internet, y así atender la responsabilidad del NOC de RedUNAM de mantener la conectividad a Internet y redes académicas en óptimas condiciones.

Palabras clave:

Sistemas de monitoreo, técnicas de observabilidad, incidentes en redes de datos.

1. INTRODUCCIÓN

La implementación de mecanismos de monitoreo para conocer el estado de operación de las redes de datos es una actividad clave debido a la importancia de conocer la disponibilidad de los enlaces de comunicaciones que influyen de forma directa en la percepción de uso de los servicios de Tecnologías de Información (TI); estos mecanismos han evolucionado impulsados por la migración de procesos críticos a sistemas informáticos. Se requiere de una detección inmediata de los eventos que afectan a cualquier nivel las comunicaciones, si se toma en cuenta que las redes se componen de infraestructura física y virtual y se necesita obtener información de todos los elementos involucrados en una falla.

En el Centro de Monitoreo de la Red (NOC) de RedUNAM de la DGTIC se busca constantemente mejorar la visibilidad de las conexiones de red debido a que el NOC es responsable de vigilar su operación y garantizar la comunicación a las entidades y dependencias universitarias por medio de enlaces dedicados al campus central de la UNAM con apoyo de proveedores de telecomunicaciones en el área metropolitana y en todo el territorio de México, tanto para las conexiones a Internet como a través de redes académicas. Contar con información completa de las comunicaciones desde que se originan por el cliente hasta ser atendidas por un servidor y a lo largo de todo el trayecto involucra soluciones tecnológicas en constante evolución. Esta complejidad, donde confluyen diferentes elementos de comunicación físicos y virtuales, motiva al cambio constante en la atención de incidentes de red.

Hasta ahora el diagnóstico de incidentes ha empleado la información operativa que ofrecen todos los dispositivos mediante el protocolo simple de administración de red (SNMP, *Simple Network Management Protocol*) y la tecnología llamada Netflow (propietaria de la marca CISCO para análisis de flujos de red), así como el estándar IPFIX para la misma tarea.

Buscar la causa raíz de incidentes de red para diagnosticar y resolverlos de forma ágil implica una estrategia para la recolección de información de la arquitectura de la red de la UNAM y datos sobre las aplicaciones que hacen uso de ella en centros de datos y servidores dedicados, así como los protocolos de comunicación de red, para estar en posibilidad de atender fallas de comunicación y detectar de manera rápida, por ejemplo, los problemas de mal funcionamiento físico.

Para apoyar esta tarea se han explorado técnicas que emergen del mundo de la operación de servicios en la nube y desarrollo de *software*. Entre ellas se destaca el concepto "*Observabilidad*"¹, el cual está relacionado con la capacidad de obtener datos de distintas fuentes y formatos que sean de interés, para proveer de un panorama completo de información que en este caso se aplica con un enfoque hacia la solución de incidentes de red.

¹ Betsy Beyer, Niall Richard Murphy, David K. Resin, Kent Kawahara and Stephen Thorne. "The Site Reliability Workbook" Google. Recuperado el 1 de noviembre de 2023, de <https://sre.google/workbook/table-of-contents/> (SER), editado por de la compañía Google.

2. OBJETIVO

2.1 OBJETIVO GENERAL

Generar una arquitectura compuesta por soluciones de software que permita la implementación de estrategias para el análisis de información de diferentes fuentes de datos, obtenidos de equipos y aplicaciones que operan en la red de datos de la UNAM para la mejora en la atención de incidentes de red.

2.2 OBJETIVOS ESPECÍFICOS

Conjuntar datos para analizar el comportamiento de eventos asociados a las comunicaciones, para identificar incidentes, establecer su causa y permitir darles solución inmediata y registrar la causa raíz de problemas frecuentes.

Integrar *software* y herramientas que permitan concentrar información de eventos (alertas de monitoreo) como resultado del monitoreo tradicional, por medio de *SNMP* de red y el análisis de bitácoras de equipos para relacionar y concentrar la información en un solo punto.

Generar tableros de control y reportes con apoyo de fuentes de datos concentradas, con apoyo de funciones de información que permitan hacer filtros para presentar el análisis de comportamientos anormales en un formato útil tanto para personal operativo como directivo.

3. DESARROLLO

Para el desarrollo de este reporte técnico se consideró importante contextualizar el origen de la iniciativa, por lo que a continuación se abordan los antecedentes y posteriormente se describe la metodología empleada para la implementación de las pruebas de *software* con apoyo de tableros de control.

3.1. ANTECEDENTES

El *NOC* de RedUNAM es el área responsable de operar y monitorear servicios de comunicaciones (enlaces de datos) que concentran diferentes equipos de red y tecnologías de comunicación, cuyo correcto funcionamiento contribuye a determinar el nivel de disponibilidad de la red de datos de la UNAM.

Para mejorar la tarea de monitoreo de enlaces de área amplia (*WAN*, *Wide-Area Network*) y así mantener la disponibilidad de los enlaces de datos en procesos cada vez más críticos tales como exámenes y procesos administrativos que operan sobre estas redes de datos, se volvió aún más importante contar con información del estado de todos los elementos que conforman los servicios de comunicación, en tiempo real.

Para incrementar la capacidad del monitoreo al ampliar la visibilidad y las relaciones de los datos disponibles, se exploraron alternativas al esquema tradicional que opera solicitando el estado de un elemento cada determinado tiempo. Este proceso fue realizado en el periodo de mayo a octubre de 2021, en el que se probaron mecanismos de monitoreo y manejo de la información alternativos al funcionamiento del protocolo *SNMP*.

Una motivación principal fue complementar la información consultable por medio de un agente SNMP, debido a que si un elemento no cuenta con soporte de este protocolo, pierde la capacidad de ser monitoreado y es únicamente compensado con la experiencia de los operadores en eventos previos. Este es el caso de las aplicaciones que operan sobre las redes de datos para las comunicaciones, como redes virtuales o las propias aplicaciones que ofrecen servicios en red como son los servicios web. Sin embargo, el protocolo sigue siendo la principal fuente de información de los equipos de red.

Para resaltar las técnicas de monitoreo exploradas y compararlas con las tradicionales, se agrupan a continuación las técnicas empleadas; se toma como referencia su enfoque de consulta (monitoreo activo) o de escucha, así como la notificación de un evento (monitoreo pasivo), como sigue:

Técnicas de Monitoreo tradicional empleadas en el NOC de RedUNAM

Monitoreo activo de incidentes:

- Monitoreo que resulta de la recolección de información de estado de los enlaces por medio de protocolos *ICMP* y *SNMP*.

Monitoreo pasivo de incidentes:

- Recolección de información por medio del protocolo *SNMP*, a través del uso de la función “traps” previamente configurada en los equipos de red a monitorear.
- Monitoreo de bitácoras de equipos por medio del protocolo *Syslog* definido por la *IETF* en los documentos *Request For Comments* 5424, 426 y 6587.
- Implementación de *RMON* (*Remote Network Monitoring*) para monitoreo del estado de elementos que soportan esta tecnología.

Técnicas exploradas y actualmente empleadas para el diagnóstico de incidentes de red:

Monitoreo pasivo de incidentes

- Monitoreo de bitácoras de los equipos de red y aplicaciones concentradas en bases de datos de series de tiempo.
- Monitoreo pasivo de protocolo *BGP* para la detección de cambios de rutas; concentra eventos para ayudar a identificar caídas de rutas lógicas.
- Monitoreo basado en trazas: Rastreo de los eventos a través de una trayectoria en las redes de datos.

Además de las técnicas de monitoreo, se exploraron técnicas de almacenamiento que permitieron ampliar las posibilidades de concentración de información en un solo punto.

Se han explorado al momento estrategias de almacenamiento del tipo binario, Round Robin y relacional, para conjuntar los diferentes mensajes en una sola base con capacidad suficiente para la retención, lectura y escritura requerida (con un mínimo de cinco años).

3.2. PROBLEMÁTICA POR RESOLVER

La complejidad de los sistemas de comunicaciones ha ampliado las fuentes de información, desde la infraestructura hasta el software que realiza las comunicaciones. La estrategia más común aplicada al interior del NOC de RedUNAM para el diagnóstico de un incidente ha sido encontrar la relación de la información del monitoreo de un incidente proveniente de estas fuentes de información, por medio de los distintos sistemas de apoyo, comparar los resultados del diagnóstico y concluir con base en la experiencia técnica del operador la causa de un incidente. Esta actividad cuenta con diferentes tipos de fuentes de información (enlaces, servidores, protocolos y aplicaciones) consultados de forma aislada; para el diagnóstico se involucran los siguientes elementos:

- Enlaces de datos (compuestos de equipos, interfaces y medios de comunicación).
- Servidores que ofrecen servicios (monitoreo de valores operativos de los mismos).
- Protocolos de comunicación (si se cuenta con información de estados y comunicación).
- Datos de aplicaciones (monitoreadas bajo solicitud).

Cada elemento genera información con un formato distinto que se consulta de forma independiente para el diagnóstico de un evento con posible afectación al interior de la red de datos de la UNAM, a la que se agrega información valiosa que tal vez no ha sido identificada porque se analiza en su propio entorno.

Encontrar las relaciones en el proceso de análisis de los datos anteriormente mencionados refleja la importancia de integrar nuevas técnicas que presenten reportes gráficos de la información recolectada por distintos mecanismos, para facilitar la identificación visual de un evento y apoyar la toma de decisiones acerca de un incidente de red por parte del personal especializado y directivos.

3.3. FUNDAMENTOS TEÓRICOS

El término “*Observabilidad*”² fue desarrollado para contar con nuevas formas de mantener vigilada la operación de los elementos virtuales que conforman los servicios de TI en la nube.

Anya Bragin (2019) menciona que “la observabilidad no es algo que un proveedor entrega en una caja, es un atributo de un sistema que creas, similar a la facilidad de uso, la alta disponibilidad y la estabilidad”. El objetivo de diseñar y crear un sistema “observable” es asegurarse de que cuando se ejecute en producción, los operadores responsables puedan detectar comportamientos no deseados y tengan información procesable para localizar la causa raíz de manera eficaz.

Con base en este razonamiento, se desarrolla la presente propuesta con un enfoque de observabilidad a las redes de datos para apoyar la necesidad de integración de información relativa a eventos operativos.

A partir de este término, se buscó implementar este enfoque centrado en las comunicaciones de las redes de datos, ya que como se mencionó, originalmente surge de entornos de desarrollo de *software* para tecnologías de información de la nube, donde todos los elementos son virtuales y la capacidad de ser observados se genera a partir de la unificación de los eventos recolectados e indicadores de cada aplicación.

² Se rastrea el origen del término “observabilidad” del libro S.R.E.(2018), donde Betsy Beyer, Niall Richard Murphy, David K. Rensin, Kent Kawahara y Stephen Thorne al libro “*The Site Reliability Workbook*” (SRE) (2018) editado por la compañía *Google*, mencionan los principios relacionados con lo que se conoce actualmente como observabilidad.

Para la empresa de soluciones de telecomunicaciones CISCO (s.f), en su portal comercial se menciona que: “la observabilidad es un proceso que utiliza herramientas de *software* para detectar problemas mediante la observación de las entradas y las salidas de la oferta tecnológica”. En el mismo artículo se menciona que “las herramientas de observabilidad recopilan y analizan una amplia gama de datos, incluido el estado y el rendimiento de las aplicaciones, las métricas de negocios como las tasas de conversión, la asignación de la experiencia del usuario, y la telemetría de la infraestructura y la red para resolver problemas antes de que afecten a los KPI empresariales”.

4. METODOLOGÍA APLICADA

Siendo relevante para el análisis de información la selección de *software* para el análisis de datos, la cantidad de espacio requerido para concentrar la información y la generación de una propuesta que otorgue valor a la operación del monitoreo de la red de la UNAM, se dividió la metodología en cuatro etapas proponiendo una estrategia donde se evaluaron diferentes técnicas de monitoreo y se conformó una arquitectura operativa para explotar la información en el proceso actual de análisis de incidentes de red.

En el desarrollo de estas etapas para la selección de *software* y análisis de información se emplearon dos estrategias que contribuyeron a definir arquitectura y evaluar qué información contribuye a la solución de incidentes de red. Se eligió como método de selección de *software* al método **MoSCoW**, complementado con el uso de los métodos **inductivo y deductivo** para centrar la elección de las capacidades consideradas como deseables en las soluciones de *software* y la identificación de información para la solución de incidentes de red considerando la presentación, flexibilidad y navegación en tiempo real de los datos.

Método MoSCoW:

El método para la selección de *software* MoSCoW es un mecanismo empleado como parte de las mejores prácticas de TI provistas por la biblioteca de buenas prácticas ITIL en su versión 4 (2019) .

MoSCoW permite separar los requisitos o características en cuatro categorías:

- M (**MUST**) La solución tiene que cumplir la capacidad. Si no se cumple se verá comprometido el éxito del proyecto.
- S (**SHOULD**) La solución debería cumplir este requisito siempre que sea posible. El éxito del proyecto no depende del cumplimiento, aunque es de alta prioridad.
- C (**COULD**) Sería relevante incluir la capacidad. Es una característica opcional.
- W (**WON'T**) Estos requisitos no se consideran, aunque podrán incluirse a futuro.

En complemento, los métodos inductivo y deductivo para selección de *software* de acuerdo con (Rodríguez Jiménez, 2017), son métodos científicos de indagación y de construcción del conocimiento, y resulta útil su empleo para la generación de criterios que permitan establecer diferencias entre varias opciones de *software*, en el caso que nos ocupa.

Se hace uso del método *inductivo* en la selección de *software* por medio de la observación de los tipos de soluciones más reconocidos, mencionados en estudios de empresas reconocidas como *Gartner*, identificando patrones de operación de acuerdo con las características ofrecidas.

En cambio, el método *deductivo* apoya el desarrollo de los criterios de selección con base en las capacidades que se buscan para integrar una arquitectura que ofrezca características similares, permitiendo la selección de *software* final derivado del análisis de los criterios generados.

4.1. ETAPA 1: ANÁLISIS DE LA INFORMACIÓN

En la primera etapa, para delimitar el alcance de una propuesta de *software* se analizó el tipo de información disponible partiendo de la información enviada por los equipos de red:

Concentrar y analizar la información

El análisis requiere concentrar la información, para ello se evaluaron las fuentes de información operativa física y virtual importantes para una comunicación completa, a través de una red de datos, identificando los siguientes elementos de comunicación:

- Origen de la comunicación (cliente y aplicación que puede consumir otra aplicación).
- Medios de transmisión que emplean (sirve para identificar fuentes de información).
- Equipos de red involucrados (sirve para definir tecnologías de monitoreo disponibles).
- Destino de la comunicación (unidireccional, bidireccional) y protocolos empleados.

Fuentes de información:

La aplicación completa del concepto de observabilidad implica obtener información de todos los elementos que componen un servicio; sin embargo, centrándose en el alcance de esta propuesta, se busca recuperar información de todos los elementos que ofrezcan datos del proceso de las telecomunicaciones, para ello se identifica cualquier proceso que sea parte de una comunicación; para ello se emplean protocolos *SNMP*, *ICMP*, información de flujos de red, bitácoras de información, excepciones de aplicación y cualquier fuente que brinde datos relevantes.

Derivado del análisis de la etapa 1, se consideran:

- Fuentes de información (protocolos, bitácoras y alertas disponibles)
 - *NETFLOW*, *IPFIX*, *SYSLOG*, *SNMP*.
- Interfaces para la consulta (cuáles serían los medios para concentrar la información)
 - Puertos lógicos de *SNMP* (161 y 162 *UDP*) y *SYSLOG* (514 *TCP*).
- Métodos de recolección (pasiva y activa)
 - *SYSLOG* y *SNMP*.
- Almacenamiento de la información (tipo de bases de datos y cómo se relacionarían)
 - Bases de datos basadas en series de tiempo.

4.2. ETAPA 2: EVALUACIÓN DE SOLUCIONES DE SOFTWARE

Para reducir el tiempo de evaluación de soluciones de monitoreo y observabilidad dentro de la gran cantidad de soluciones que ofrecen características de valor para el diagnóstico y solución de incidentes de red, se consideró como punto de comparación el análisis de la empresa referente de estudios de *software* propietario *Gartner*.

Considerando las características de las soluciones de *software* propietarias que son de gran valor en las tareas de concentración y análisis de información, destacan las compañías *Elastic, New Relic, Datadog, Splunk* y *Dynatrace*. Sin embargo, derivado de una evaluación inicial del costo de uso de los productos con información de las páginas oficiales, extrapolando el costo por *gigabyte* de información procesado, se consideraron no costeadables, pero sirven de base para elaborar una arquitectura compuesta de soluciones de software libre integrando capacidades destacadas de las propietarias.

La evaluación de características que se identificaron en el software propietario disponibles en *software* libre y aportan valor a las necesidades del NOC de RedUNAM son:

- Soporte para base de datos libre de alta velocidad de lectura y escritura,
- Interfaz o mecanismos de conexión con fuentes de información externas, y
- Tableros de control web con facilidad de uso para los usuarios expertos y no expertos.

El listado detallado se encuentra en el **Anexo A, tabla 1: Evaluación MoSCoW**.

Las características antes mencionadas posteriormente se evaluaron en las soluciones con mayor comunidad de soporte realizando la selección de entre las aplicaciones listadas, destacando: *Grafana, Loki, PNP, Telegraf* y *Rsyslog*. En el **Anexo A, tabla 2: Selección de Software** se ubican también las aplicaciones evaluadas.

4.3. ETAPA 3: ALMACENAMIENTO DE LA INFORMACIÓN

Durante la evaluación del *software* para cumplir con las necesidades de almacenamiento se buscó conformar una arquitectura con bases de datos no propietarias, puesto que la arquitectura se compone de varias fuentes de datos y se pretende que concentre gran cantidad de la información, además de permitir la consulta de información en tiempo real para generar tableros.

Esta tarea, con colectores de datos propietarios, requiere muchos recursos y licencias. Por esta razón se eligieron colectores de información basados en series de tiempo que son los incluidos en la evaluación anterior.

Para esta tarea se considera la implementación de más de un colector de información para poder incorporar la mayor cantidad de datos posible y posteriormente poderla relacionar en tableros de control. Los colectores de información en forma de eventos con marcas de tiempo seleccionados fueron:

- *FluentD*
- *Rsyslog*
- *PNP NAGIOS*
- *LOKI*

Se le llaman colectores debido a la función que realizan de captura, almacenamiento y posibilidad de consulta de información guardada; cada *software*, como el caso de *FluentD*, es un sistema que genera su estructura para gestionar la información recuperada, habilitando interfaces de comunicación y también interfaces de usuario (*UI*) para la administración de la base de información en su conjunto. La ventaja de usar este tipo de colectores proviene de que permiten el registro de cada evento empleando una marca de tiempo.

4.4. ETAPA 4: DEFINICIÓN DE ARQUITECTURA

Derivado del análisis de aplicaciones, se generó una arquitectura compuesta de aplicaciones libres para alimentar el *software* elegido para tableros de control *Grafana*, cuyo mayor diferenciador fue la capacidad de integrar como fuentes de datos al sistema de monitoreo de disponibilidad, a través de interfaces.

La arquitectura se estructuró como sigue:

- Sistema operativo *Linux Base Debian*.
- Implementación de *software PNP4Nagios* (para obtener el dato de monitoreo SNMP).
- Implementación de *FluentD* (concentrar información hacia base de datos *Json*).
- *Software* de recolección de eventos *RSYSLOG*.
 - Duplicación de bitácoras seleccionadas hacia el *software FluentD*.
 - Configuración de equipos remotos para envío de bitácoras *SYSLOG*.
- Implementación de *software LOKI* (filtrado y consultas desde bitácoras).
- Manejador de tableros de control *Grafana*.
 - Fuentes de datos integradas para recuperación de información.
 - Fuente *LOKI*.
 - Fuente *FluentD*.
 - Fuente *PNP for Nagios Firewall* local integrado (*IPfilter*).
- Servidor de correo para liberar alertas *Postfix* (para envío de alertas).

5. RESULTADOS

Se logró la implementación y vinculación del sistema de monitoreo *Nagios* permitiendo consultar el estatus de los servicios monitoreados a nivel de disponibilidad por medio de la interfaz *PNP 4 Nagios* a *Grafana*; lo anterior se muestra en la figura 1, con las fuentes de datos configuradas en el *software Grafana*, en la arquitectura implementada para la evaluación realizada.

Figura 1

Fuentes de datos integradas al sistema *Grafana*



Se envía información de los servicios monitoreados de acceso a Internet y monitoreo de protocolos de red que podría apoyar a la identificación de incidentes.

Se implementó el servicio de captura de bitácoras, primero por medio de la aplicación *Rsyslog* y su posterior replicación a una segunda aplicación para su tratamiento, *FluentD*, permitiendo la consulta desde *Grafana* por medio de la solución *Loki* para generar tableros de control. Se implementaron las interfaces para el acceso a información de monitoreo *PNP4Nagios* y *LOKI*, como se muestra en la figura 2.

Figura 2

Consulta de información y gráfica de conteo de eventos con Grafana + Loki



También se configuraron tableros de control con la información de las fuentes de datos para concentrarlos en una misma vista y comparar información de distintas fuentes, como se ve en la figura 3, con un ejemplo de la contabilización de eventos de diferentes fuentes de datos (*Nagios + PNP + Grafana, RSYSLOG + LOKI + Grafana*) y el cambio de rango a minuto y día.

Figura 3

Conteo de bitácoras con la palabra "intruder" con Loki + tiempos de respuesta a Internet



Se lograron generar tableros de control en *Grafana*:

- Gráficas de los valores de respuesta consultados con el sistema *Nagios + PNP4Nagios*.
- Comparación en una sola gráfica de hasta tres elementos monitoreados en *Nagios*.
- Creación de umbrales para monitoreo de patrones que salen de lo que se espera.

Se hicieron filtros en los tableros de control de *Grafana + Loki*:

- Se obtuvo un reporte gráfico por medio de *Grafana + LOKI* sobre las bitácoras recolectadas en el servidor.
- Se logró realizar la gráfica del comportamiento de incidentes para rechazos de autenticación en equipos inalámbricos con *LOKI + Grafana* empleando *Logql*.

Sobre los límites en la correlación de información para su presentación gráfica, se identificó que la información recolectada se puede presentar aplicando métodos de:

- Conteo de mensajes.
- Graficación numérica.
- Suma por periodo de tiempo.
- Filtrado para enfocar resultados.
- Multiplicación de indicadores para resaltar información.

Se observan limitantes en la cantidad de información contabilizada por periodos de tiempo. Como se muestra en la figura 4, se comprobó la posibilidad de generar conteos de ocurrencias en eventos registrados, pero se observó que entre mayor es la cantidad existente, se requiere de más procesamiento para generar los gráficos, por lo que es necesario considerarlo al plantear el monitoreo de grandes volúmenes como puede ser el consultar sobre un servicio del que se requiere graficar un tipo de acceso en particular. Es el caso de la figura 4, que muestra el análisis de datos de eventos de red, con un evento donde se eleva el tiempo de respuesta hacia *Google* y la posible relación con el incremento para un servicio de traducción de nombres de dominio.

Figura 4

Grafana, Gráficas numéricas y de calor + Umbrales



En cuanto al filtrado de información, se detecta que el lenguaje *LogQL* tiene una semántica limitada para aplicar filtros que permitan graficar cualquier por medio de filtros eventos contabilizados; sin embargo, acepta la integración de expresiones regulares con la lentitud que acarrea su uso para la generación de gráficas.

Finalmente, como se observó en la figura 4, la representación de gráficas se acota a las existentes al interior del sistema *Grafana*, y aunque pueden personalizarse, no se encontró posibilidad para generar una versión completamente personalizada.

Sobre recomendaciones a futuro y siguientes pasos, se identificó que algunos plugins para *Grafana* solamente están disponibles para la versión comercial. Asimismo, para el almacenamiento se recomienda buscar explotar de una mejor forma la base *FluentD* ya que permite la integración de información de más fuentes de datos (aplicaciones, servidores e información de monitoreo), pero se sugiere considerar que su configuración es más compleja.

Es importante hacer una evaluación de las limitaciones de los tableros de control, ya que hasta el momento se identifica la necesidad de buscar superponer información de diferentes fuentes de datos en una misma gráfica y en más de tres gráficas, para ampliar la visibilidad de la información en un solo tablero.

Sobre la generación de reportes automatizados de la información, se sugiere considerar que actualmente se puede compartir la información en tiempo real, pero para un reporte ejecutivo podría ser importante generar un reporte con formato con una descripción de los elementos graficados.

6. CONCLUSIONES

A partir de las pruebas realizadas y la conformación de la arquitectura para observabilidad de los eventos de red, se concluye que la arquitectura evaluada permite concentrar y presentar información operativa relevante y en tiempo real para la solución de incidentes.

La elección del *software* para la presentación gráfica de información recolectada tiene como base el sistema *Grafana* que se reconoce por la facilidad de crear tableros de control a partir de diferentes fuentes de información, apoyando la identificación de posibles incidentes de red además de permitir el establecimiento de umbrales gráficos, permitiendo identificar su relación con otros eventos y ayudando a visualizar el impacto de los eventos graficados a primera vista.

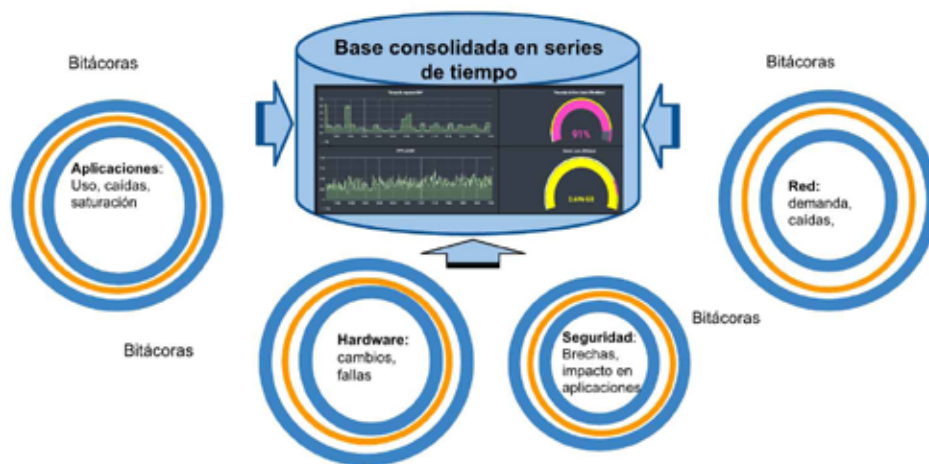
Se confirma también que con apoyo de las interfaces de *software* es posible relacionar distintos tipos de fuentes de información generando paneles gráficos sobre sucesos que ocurren en tiempo real, gracias a que los eventos se reciben y almacenan en línea en bases de datos que emplean marcas de tiempo para identificar su información.

Se concluye también que la personalización de distintos paneles contribuye a la agilidad de acceso a la información tanto al personal operativo como directivos, debido a que las gráficas pueden adecuarse para presentarse de acuerdo con las necesidades de los usuarios con apoyo de filtros y aplicación de formatos de presentación de *Grafana* para destacar información relevante.

A partir de la cantidad de información que la arquitectura puede concentrar, se sugiere plantear un proyecto donde distintas áreas puedan concentrar información operativa creando un indicador de disponibilidad global no solo con la intención de identificar incidentes sino generando métricas de uso y saturación conjuntas como se representa en la figura 5:

Figura 5

Fuentes de datos que podrían integrarse para evaluar comportamientos



El trabajo conjunto puede facilitar también la identificación de las causas que originan un incidente al contar con información de los servicios a diferentes niveles, apoyando así la mejora de la disponibilidad de los servicios en red de la UNAM y el acceso a Internet.

REFERENCIAS BIBLIOGRÁFICAS

- Chuck Lane and Joerg Linge, *PNP4Nagios*. Recuperado 30 de agosto de 2023 de <https://github.com/pnp4nagios>
- Cisco Systems, Inc., (s.f). *¿Qué es la observabilidad?* Recuperado 30 de agosto de 2023, de https://www.cisco.com/c/es_mx/solutions/full-stack-observability/what-is-observability.html
- Cloud Architecture Center, GoogleCloud, *Medición de DevOps: supervisión y observabilidad*, (2023). *Cloud Architecture Center*, Última actualización: 2023-08-19 (UTC). Recuperado 30 de agosto de 2023 de <https://cloud.google.com/architecture/devops/devops-measurement-monitoring-and-observability?hl=es-419>
- Grafana Labs (2023). *Grafana Loki documentation*. Recuperado 30 de agosto de 2023, de <https://grafana.com/docs/loki/latest/>
- Grafana Labs (2023). *Technical documentation for Grafana Labs products and services*. Recuperado 30 de agosto de 2023, de <https://grafana.com/docs/>
- FluentD Project team, *Guides and Receipts, Log Analytics*. Recuperado 30 de agosto de 2023, de <https://www.fluentd.org/guides>

Marko Hartikainen. (2023). Bachelor thesis Degree Programme in Business Information Technology 2023, Häme University of Applied Sciences. Recuperado de https://www.theseus.fi/bitstream/handle/10024/804045/Hartikainen_Marko.pdf?sequence=2

Rainer Gerhards (GroBrinderfeld), rsyslog. *The rocket-fast system for log processing*. Recuperado 30 de agosto de 2023, de <https://www.rsyslog.com/doc/master/index.html>

ANEXO A

Tabla 1

Evaluación MoScow. Evaluación de las características para selección de software

Característica disponible	Evaluación MoScow	Importancia
Soporte de almacenamiento por medio de base de datos libre y de alta velocidad de lectura y escritura	Debe tener la capacidad (Must)	<ol style="list-style-type: none"> 1. La base de datos creciente de información. 2. Acceso múltiple para crear reportes personalizados. 3. Escritura constante 24x7x365
Licencia libre y posibilidad de escalar a servicio de paga	Debe tener la capacidad (Must)	Debido a que la Universidad maneja tráfico de usuario final superior a 20 Gigabits por segundo distribuidos adicional a los enlaces al interior de la república se generan datos operativos en su mayoría de poca relevancia operativa, no es viable pagar por cada evento registrado
Interfaz o mecanismos de conexión con fuentes de información externas.	Debe tener la capacidad (Must)	Es necesario que la información operativa de eventos relacionados con incidentes de red pueda vincularse con información de diferentes fuentes para generar reportes útiles.
Tableros de control web con facilidad de uso para los usuarios expertos y no expertos	Debe tener la capacidad (Must)	La presentación de los informes es tan importante como la generación de información debido a la importancia de contar con ellos al momento de identificar un comportamiento anómalo
Cambio de formato de información recolectada	Puede tener, pero no necesaria (Should)	Para presentar resultados es importante contar con un formato que refleje los hallazgos y se pueda identificar fácilmente a la vista del personal.
Alertas vía correo	Puede tener, pero no necesaria (Should)	El envío de información relevante por medio de correo electrónico es de gran ayuda cuando se encuentra el personal fuera del horario laboral.

Característica disponible	Evaluación MoSCoW	Importancia
Alertas por cualquier medio	Debe tener la capacidad (Must)	El envío de información relevante por medio de correo electrónico es de gran ayuda cuando se encuentra el personal fuera del horario laboral.
Módulos para ubicar geográficamente los dispositivos	Debe tener la capacidad (Must)	Esta característica es útil para ubicar si un incidente afecta geográficamente a otros sin una relación directa
Cálculos sobre información generar nueva información	Debe tener, pero no necesaria (Should)	Esta función ayuda a resaltar dentro del contexto de un reporte gráfico los elementos más importantes al presentarse un incidente de red.
Reportes personalizados	Debe tener, pero no necesaria (Should)	Importante para cuando se requiere generar un reporte ejecutivo a personal directivo y a usuarios finales.
Base de datos basada en series de tiempo circular	No debería tener (Won't)	Es una característica deseable para datos que crecen muy rápidamente, esta característica sin embargo puede sustituirse por depuraciones periódicas.
Monitoreo en tiempo real	Debería tener (Could)	Entre más rápido se tenga la información de identificación de un incidente es mejor. Sin embargo, es aceptable el retraso hasta de un minuto dependiendo la cantidad de información analizada.
Tablero de control mostrando resultados en tiempo real	Debería tener (Could)	Entre más rápido se tenga la información de identificación de un incidente es mejor. Sin embargo, es aceptable el retraso hasta de un minuto dependiendo la cantidad de información analizada.
Operaciones con información obtenida para generación de gráficas.	Debería tener (Should)	Importante para cuando se requiere generar un reporte específico, ejecutivo a personal directivo y usuarios finales.
Alta disponibilidad	Puede tener, pero no necesaria (Could)	Es necesario que la información siempre esté disponible y es común que el sistema de monitoreo se vea afectado ante un incidente de red; es de gran utilidad contar con un respaldo activo del sistema.
Alertas gráficas al alcanzar umbrales	Puede tener, pero no necesaria (Could)	La preparación de un escenario en que debiera ser notificado el personal cuando se alcance el consumo de algún servicio presenta gran utilidad al momento del diagnóstico operativo de un incidente de red.

Tabla 2

Selección de software. Resultado de la evaluación de software para observabilidad

Software	Función principal	Resultado MoSCoW	Observaciones
Fluentbit	Almacenar información	NO INCLUIR	Complejidad alta y bajas posibilidades de integración
FluentD	Almacenar información	INCLUIR	Cumple con necesidades
NXlog	Recolectar Syslog	NO INCLUIR	Complejidad alta
Graylog	Recolectar Syslog	NO INCLUIR	Sistema parcialmente propietario
Octopussy	Recolectar Syslog	NO INCLUIR	No se adapta a otras soluciones
Telegraf	Recolecta información de sistema	INCLUIR	Se puede adaptar a otras soluciones
Grafana	Genera tableros de control	INCLUIR	Cumple con necesidades
Kibana	Genera tableros de control	NO INCLUIR	Sistema parcialmente propietario
Rsyslog	Recolectar Syslog	INCLUIR	Cumple con necesidades
Loki	Recolectar Syslog	INCLUIR	Cumple con necesidades
PNP Nagios	Recolecta información del sistema de monitoreo Nagios	INCLUIR	Cumple con necesidades

Uso de GnuPG como herramienta para la confidencialidad de la información

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:

Chavarría Fernández, R. (2023). Uso de GnuPG como herramienta para la confidencialidad de la información. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (185 - 195).

<https://doi.org/10.22201/dgtic.ctud.2023.1.1.19>

Ricardo Chavarría Fernández

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

rick@unam.mx

ORCID: 0009-0000-3817-6827

Resumen:

Se ha incrementado la importancia del uso de *GnuPG* como herramienta criptográfica para la preservación de la privacidad y confidencialidad en el mundo digital, debido a la necesidad de salvaguardar la información contra divulgación no autorizada. El proceso implica el uso de algoritmos criptográficos para cifrar y descifrar archivos o mensajes, y su implementación asegura la privacidad y confidencialidad de los archivos cifrados. Sin embargo, al igual que otras herramientas criptográficas, *GnuPG* también puede presentar vulnerabilidades o desafíos técnicos, por lo tanto, es crucial utilizarla adecuadamente para garantizar la seguridad de la información.

Palabras clave:

Pretty Good Privacy, PGP, OpenPGP, GNU Privacy Guard, GnuPG.

1. INTRODUCCIÓN

La pérdida, fuga o robo de información es una amenaza constante que enfrentan las organizaciones. La información puede ser extraída por personal no autorizado, ciberdelincuentes o como objetivo económico de gobiernos y organizaciones. Este tipo de amenaza se puede mitigar mediante la aplicación de controles de acceso y herramientas criptográficas que minimicen la fuga y divulgación de la misma.

La confidencialidad es un pilar importante de la seguridad y una garantía de que la información será resguardada o protegida para evitar que se divulgue sin consentimiento, y así procurar su cumplimiento mediante medidas de seguridad, y grupos de reglas o parámetros que limiten el acceso a los usuarios.

Independientemente del sistema en el que se encuentre la información, los responsables deben establecer y mantener las medidas de seguridad de carácter administrativo, físico y técnico, como lo establece el estándar *ISO/IEC 27001:2013*, Anexo A.10 Cifrado, o el estándar *ISO 27001:2022*, Anexo 8.24 Uso de criptografía: "Garantizar el uso adecuado y eficaz de la criptografía para proteger la confidencialidad, autenticidad o integridad de la información..." con la finalidad de proteger contra daño, pérdida, destrucción, alteración o tratamiento no autorizado.

La criptografía permite hacer ilegible el contenido semántico o el significado de cierta información que se desea salvaguardar o compartir mediante el uso de una llave única para cifrar y descifrar (cifrado simétrico) o un par de llaves relacionadas (pública y privada) que permiten cifrar y descifrar respectivamente (cifrado asimétrico). Del mismo modo, la implementación adecuada y precisa del cifrado es extremadamente crítica para su eficacia, debido a que un fallo en la configuración dará como resultado un cifrado poco seguro, y en consecuencia la pérdida de la protección o confidencialidad. Por ello, es importante considerar las siguientes vulnerabilidades en su implementación: mal manejo de llaves, adopción de algoritmos de cifrado y generadores aleatorios de números computacionalmente débiles.

Si se considera el uso adecuado y eficaz de la criptografía, es recomendable implementar una herramienta que mitigue las vulnerabilidades asociadas a su implementación y garantice un algoritmo de cifrado sólido, atendiendo el empleo de estándares, interoperabilidad, capacidad de gestión de claves y una amplia adopción en la comunidad de seguridad informática.

Una herramienta criptográfica que proporciona servicios de integridad y confidencialidad mediante firma digital, cifrado y compresión, e incluso cumple con los criterios descritos anteriormente es *GNU Privacy Guard (GnuPG)* (*The GnuPG Project*, s/f). *GnuPG* es un *software* de código abierto que implementa el cifrado de llave pública de *Pretty Good Privacy (PGP)* (OpenPGP, 2023) desarrollado originalmente en 1991 por Phil Zimmermann y el estándar abierto de cifrado de email basada en la tecnología *PGP (OpenPGP)* (About,2023a), que cumple con los estándares **RFC4880** (J. Callas, 2007), **RFC 3156** (M Elkins, 2015) y **RFC 6091** (N.Mavrogiannopoulos, 2011).

El uso de *GnuPG* permite cifrar la información contenida en archivos digitales tales como: datos personales, informes financieros, propiedad intelectual, entre otros; y ayuda a evitar que una organización tenga pérdida de información, afectaciones de tipo económico o de reputación.

2. OBJETIVO

Implementar mecanismos con *GnuPG* mediante la generación y administración de llaves, cifrado y descifrado de archivos con información sensible, crítica, clasificada o privada para preservar la confidencialidad de la información.

3. DESARROLLO TÉCNICO

A continuación, se presenta la instalación de la herramienta y su uso para cifrar y descifrar archivos en el sistema operativo *Linux*, específicamente en la distribución *CentOS 7*. La instalación se llevó a cabo utilizando el comando “*yum*,” que es un sistema de gestión de *software* para instalar, actualizar y eliminar grupos de paquetes, incluyendo sus dependencias necesarias, también conocidas como aplicaciones auxiliares.

Con ello se busca prevenir el acceso a la información por usuarios no autorizados y evitar:

- a) Robo, copia o extravío.
- b) Uso o tratamiento.
- c) Daño o alteración.

3.1 INSTALACIÓN EN SISTEMA OPERATIVO LINUX (CENTOS 7)

El primer paso fue instalar la herramienta *GnuPG* y el generador de entropía, para lo cual se ejecutó la sentencia o línea de código “*yum*” (PGP Command Line User Guide, 2020):

Figura 1

Línea de código de instalación

```
# yum install gnupg y # yum install rng-tools
```

En los enlaces <https://www.gnupg.org/download/index.html> y <https://github.com/nhorman/rng-tools>, se deben descargar los paquetes de *GnuPG* y el generador de entropía respectivamente.

3.2 GENERACIÓN DE ENTROPÍA

La entropía (Del Rio Mateos, s/f) es la generación de una gran muestra de números o datos pseudoaleatorios que recopila un sistema operativo para generar llaves criptográficas utilizadas para cifrar información. Esta pseudoaleatoriedad frecuentemente se obtiene de componentes de *hardware* tales como los movimientos del ratón, electrónicos o, si se dispone, de generadores de azar.

Una vez instalado, se generó la entropía que sería utilizada por *GnuPG* para la creación de claves pública y privada al ejecutar la siguiente sentencia (*Red Hat Customer Portal*, s.f.):

Figura 2

Línea de código para la generación de entropía

```
# sudo rngd -r /dev/urandom
Initalizing available sources
Initalizing entropy source Hardware RNG Device
Enabling RDSEED rng support
Initalizing entropy source Intel RDRAND Instruction RNG
Enabling JITTER rng support
Initalizing entropy source JITTER Entropy generator
```

Figura 3

Entropía



Nota. Adaptado de Acevedo, Ó. y Romero D. (febrero, 2021). El significado profundo de los números. Revista Esfinge. Recuperado de <https://revistaesfinge.com/2010/08/el-significafo-profundo-de-los-numeros/>

3.3 GENERACIÓN DE LLAVES

Las claves se generaron al ejecutar la siguiente sentencia (PGP Command Line User Guide, 2020):

Figura 4

Línea de código para la generación de llaves

```
# sudo gpg --gen-key
```

Luego, el sistema solicitó proporcionar la siguiente información:

a) Tipo de cifrado

Figura 5

Selección del tipo del algoritmo de cifrado

```
gpg (GnuPG) 2.0.22; Copyright (C) 2013 Free Software Foundation, Inc.  
This is free software: you are free to change and redistribute it.  
There is NO WARRANTY, to the extent permitted by law.  
  
Por favor seleccione tipo de clave deseado:  
(1) RSA y RSA (por defecto)  
(2) DSA y ElGamal  
(3) DSA (sólo firmar)  
(4) RSA (sólo firmar)  
Su elección: 1
```

Es recomendable utilizar la opción 1 o 2, *RSA* y *ElGamal*, ya que ambos son algoritmos sumamente robustos, estandarizados y aceptados por la comunidad internacional (David Trujillo Gradit, 2020).

b) Tamaño de la llave

Figura 6

Solicitud del tamaño de la llave

```
las claves RSA pueden tener entre 1024 y 4096 bits de longitud.  
¿De qué tamaño quiere la clave? (2048) 3072 (Enter)
```

Por seguridad, se proporcionó un valor superior a 2048.

c) Período de validez

Figura 7

Solicitud del periodo de validez

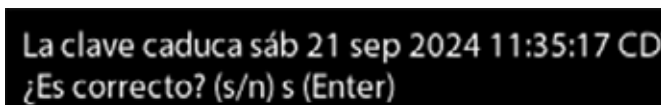
```
Por favor, especifique el período de validez de la clave:  
0 = la clave nunca caduca  
<n> = la clave caduca en "N" días  
<n>w = la clave caduca en "N" semanas  
<n>m = la clave caduca en "N" meses  
<n>y = la clave caduca en "N" años  
¿Validez de la clave (0)? 1y (Enter)
```

Para fines ilustrativos, la validez proporcionada fue de un año, no obstante, los tiempos de validez se deben establecer de acuerdo con las necesidades propias de la organización, la criticidad de la información y la actualización de los algoritmos.

Posteriormente, la solicitud se confirmó al presionar la letra "s" (si).

Figura 8

Confirmación del periodo de validez

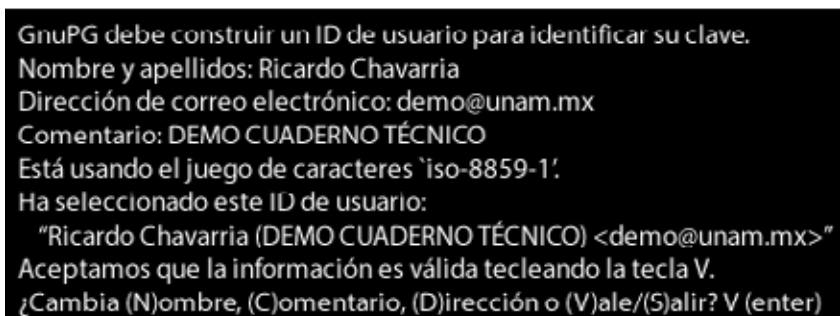


d) ID de usuario

En esta sección se solicitó que se proporcionará la información del propietario de las claves:

Figura 9

Solicitud de información del propietario de las llaves



e) Contraseña del llavero y creación de llaves

El sistema solicitó la introducción y confirmación de la contraseña, tal como se muestra en las figuras 10 y 11.

Figura 10

Solicitud de contraseña para proteger las llaves GNP

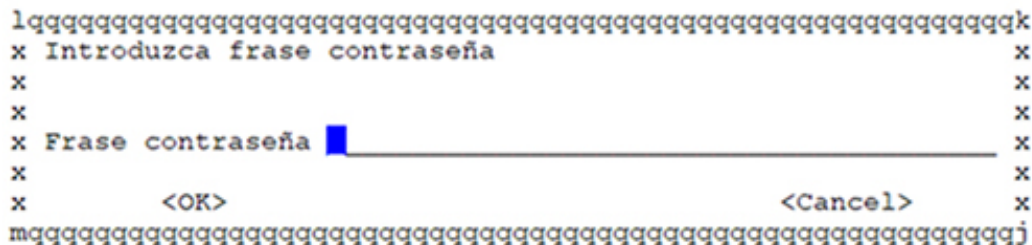


Figura 11

Solicitud de confirmación de contraseña para proteger las llaves GNP

```

lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x Introduzca frase contraseña x
x x x
x x x
x Frase contraseña [█] x
x x x
x <OK> <Cancel> x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
  
```

Tras proporcionar la contraseña, la herramienta *GnuPG* creó el par de llaves utilizando los valores generados por la entropía, y mostró los siguientes mensajes:

Figura 12

Confirmación de generación del par de llaves

```

Es necesario generar muchos bytes aleatorios...

gpg: siguiente comprobación de base de datos de confianza el: 2024-09-21
pub 3072R/D631B8D4 2023-09-22 [caduca: 2024-09-21]
Huella de clave = 9D96 BAF5 7062 2852 94F3 25B1 C428 4495 D631 B8D4
uid Ricardo Chavarria (DEMO CUADERNO TÉCNICO) <demo@unam.mx>
sub 3072R/915B41FA 2023-09-22 [caduca: 2024-09-21]
  
```

3.4 FINALIZAR EL PROCESO DE ENTROPÍA

Para finalizar el proceso de entropía, se debió ejecutar la siguiente sentencia (Red Hat Customer Portal, s.f.):

Figura 13

Línea de código para finalizar el proceso de entropía

```
# sudo pkill rngd
```

3.5 LISTADO DE LAS LLAVES

Se proporcionó la siguiente sentencia para visualizar las claves generadas y contenidas en el llavero GnuPG (PGP Command Line User Guide, 2020):

Figura 14

Línea de código para listar las llaves generales

```
# sudo gpg --list-keys
/root/.gnupg/pubring.gpg
-----
pub 2048R/7A6D1180 2023-09-22
uid          dsf isdf sdfsd sdf ds
sub 2048R/3D252456 2023-09-22

pub 3072R/D631B8D4 2023-09-22 [caduca: 2024-09-21]
uid          Ricardo Chavarria (DEMO CUADERNO TÉCNICO) <demo@unam.mx>
sub 3072R/915B41FA 2023-09-22 [caduca: 2024-09-21]
```

3.6 CIFRADO Y DESCIFRADO DE ARCHIVOS DESDE CONSOLA

Para **cifrar** un archivo, se requirió ejecutar la siguiente sintaxis:

```
sudo gpg --encrypt --armor --recipient [ID_USUARIO] [ARCHIVO_A_CIFRAR] (PGP Command Line User Guide, 2020)
```

Figura 15

Línea de código para cifrar un archivo

```
# sudo gpg --encrypt --armor --recipient demo@unam.mx demo.txt
```

Como resultado, se obtuvo un archivo cifrado con la extensión .asc, tal como se muestra en la figura 17:

Figura 16

Contenido archivo demo.txt

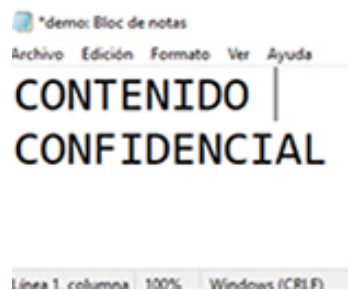


Figura 17

Contenido archivo cifrado demo.txt.asc



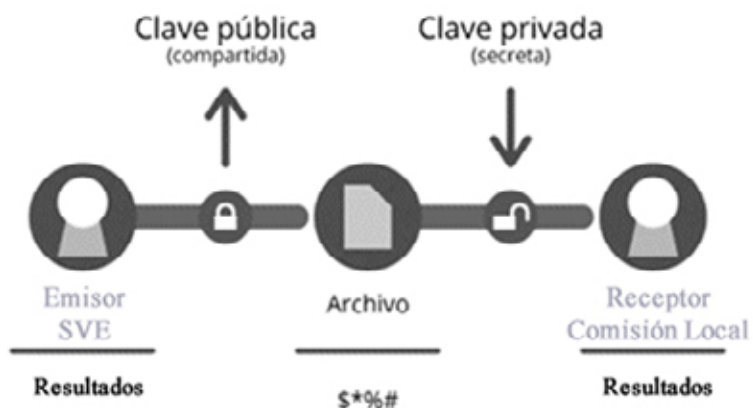
Para **descifrar** un archivo fue necesario ejecutar la siguiente sintaxis:

```
sudo gpg --output [Nombre archivo en texto claro] --decrypt [Nombre archivo cifrado] (PGP Command Line User Guide, 2020).
```


En particular, en la etapa de entrega de resultados preliminares, se ha utilizado la herramienta *GnuPG* para preservar la integridad y confidencialidad de los archivos en formato *PDF* que contienen el recuento total de boletas y votos, así como la participación del padrón electoral. La instalación y configuración detallada en la sección 3. *Desarrollo técnico*, junto con la compatibilidad del lenguaje de programación utilizado en el sistema, garantizaron la privacidad y secrecía de los archivos en el servidor, y así las *Comisiones Locales de Vigilancia de la elección* accedieran al contenido de los archivos de manera exclusiva a través de la llave privada. Esto se resume en la figura siguiente:

Figura 22

Proceso de emisión, cifrado y recepción de resultados



En resumen, la importancia de la seguridad en la distribución de resultados y conteos de boletas en el sistema de votaciones electrónicas radica en su papel fundamental para preservar la integridad del proceso electoral, fomentar la confianza pública y garantizar elecciones seguras.

5. RESULTADOS

GnuPG presentó una serie de hallazgos notables en su implementación. En primer lugar, la instalación en *Linux CentOS 7* se realizó de manera nativa a través del comando *yum*, eliminando la necesidad de requisitos técnicos avanzados. Destaca su capacidad de utilizar algoritmos criptográficos fuertes, como *RSA*, *ElGamal* y *DSA*, basados en funciones matemáticas complejas y difíciles de calcular. En cuanto a la generación de llaves, permitió seleccionar longitudes adecuadas, como llaves *RSA* de 2048 bits o más, proporcionando una capa adicional de seguridad haciendo que sea difícil para los atacantes descifrar algún tipo de información. Por último, implementa el estándar de cifrado de mensajes *OpenPGP*, lo que permitió la interoperabilidad con otras aplicaciones y la ejecución del cifrado y descifrado de archivos.

A pesar de sus ventajas técnicas, *GnuPG* presentó desafíos que deben ser considerados. Los problemas de instalación, que surgen debido a la falta de dependencias necesarias durante este proceso, y los conflictos potenciales entre diferentes versiones de *GnuPG*, complican su implementación. La generación

de claves se ve afectada por la falta de entropía, lo que, en ocasiones, puede bloquear la generación de datos aleatorios, además de experimentar problemas en el cifrado y descifrado por la ausencia de la llave pública del destinatario o al riesgo de olvidar la contraseña de la llave privada. De la misma manera, si la fecha del sistema no está configurada correctamente, emergen problemas en la verificación de la información. Finalmente, en términos de rendimiento, la alta demanda de recursos de *CPU* en operaciones criptográficas resulta en un cifrado y descifrado más lentos, especialmente al trabajar con grandes volúmenes de datos o en sistemas con recursos limitados.

6. CONCLUSIONES

Al abordar los desafíos y aprovechar al máximo sus ventajas, los mecanismos de seguridad proporcionados por *GnuPG* para la gestión de claves, que incluyen la capacidad de generar, importar, exportar, revocar, proteger y administrar tanto llaves públicas como privadas, junto con la gestión de certificados y la implementación de algoritmos de cifrado seguro (ya sea asimétrico o simétrico), garantizaron que los archivos cifrados preservaran su privacidad y confidencialidad a lo largo del tiempo. Esto significa que el contenido permaneció ilegible, preciso, consistente y confiable, independientemente del período de almacenamiento o la frecuencia de acceso. Sin embargo, es fundamental abordar los desafíos técnicos para maximizar su utilidad en la preservación de la confidencialidad de la información. Se recomienda prestar especial atención a la instalación, garantizando la disponibilidad de dependencias y la compatibilidad de versiones, así como aumentar la entropía del sistema con movimientos del ratón, ejecución de aplicaciones o de generadores de azar. Además, es esencial mantener un registro seguro de las contraseñas de llaves privadas y asegurarse de que la fecha del sistema esté correctamente configurada para evitar problemas en la verificación de información. La elección de los recursos computacionales adecuados contribuirá a un rendimiento óptimo de la herramienta. En resumen, abordar estos desafíos técnicos requiere un esfuerzo que se traducirá en una mayor seguridad y confidencialidad de la información, así mismo se debe considerar la disposición de los usuarios a invertir tiempo en aprender y configurar la herramienta.

REFERENCIAS BIBLIOGRÁFICAS

- About. (2023a, junio 27). *OpenPGP*. Recuperado de <https://www.openpgp.org/about/>
- Del Rio Mateos, A. (s.f.). *Introducción a la Criptología*. <http://www.um.es>. Recuperado el 23 de octubre de 2023, de <https://www.um.es/adelrio/Docencia/Criptografia/Criptografia.pdf>
- J. Callas, (2007). *RFC4880*, Rfc-editor.org. Recuperado el 20 de octubre de 2023, de <https://www.rfc-editor.org/rfc/rfc4880.txt>
- PGP Command Line User Guide. (2020). Broadcom.com. Recuperado de https://techdocs.broadcom.com/content/dam/broadcom/techdocs/symantec-security-software/information-security/pgp-solutions/10-4-2/generated-pdfs/pgpCmdline_usersguide_en.pdf
- Red Hat Customer Portal (s.f.). *Using the random number generator Red Hat Enterprise Linux 6* https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/security_guide/sect-security_guide-encryption-using_the_random_number_generator
- The GnuPG Project. (s.f.). *The GNU Privacy Guard*. Recuperado de <https://www.gnupg.org/index.html>

Directorio Activo en laboratorio virtual de la Facultad de Estudios Superiores Aragón

Información del reporte:

Licencia Creative Commons



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

Para citar este reporte técnico:

Martínez Sierra, L. (2023, Noviembre). Directorio Activo en laboratorio virtual de la Facultad de Estudios Superiores Aragón. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (196 - 204)

<https://doi.org/10.22201/dgtic.ctud.2023.1.1.6>

Laura Martínez Sierra

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

laura.martinez@unam.mx

ORCID: 0009-0004-2850-2384

Resumen:

Se realizó el proyecto de instalación y configuración del *Directorio Activo* como parte de la infraestructura necesaria para el funcionamiento de laboratorios virtuales requeridos por la Facultad de Estudios Superiores Aragón. El proyecto consistió en ofrecer veinte laboratorios virtuales a la comunidad estudiantil que proporcionan acceso a equipos de cómputo vía remota, que incluyen características de *software* y *hardware* eficientes para la realización de sus actividades académicas. Esta iniciativa surge a partir de la necesidad de los alumnos que no contaban con equipos que tuvieran las características de procesamiento o *software* para llevar a cabo sus actividades, situación que tuvo más impacto en la modalidad de educación a distancia impartida temporalmente en la Universidad Nacional Autónoma de México durante la pandemia provocada por el virus SARS-CoV-2.

Palabras clave:

Datos personales, servicios de nube, instrumentos consensuales.

1. INTRODUCCIÓN

La pandemia del *COVID 19* transcurrida entre los años 2020 y 2023 generó numerosos problemas en los ámbitos laboral, cultural, de salud y educación, puesto que no sólo se pusieron en jaque a los sistemas de salud, sino que también a la infraestructura tecnológica, la cual se convirtió en parte fundamental de la vida de la población.

Derivado de la emergencia sanitaria el gobierno mexicano implementó una serie de medidas de seguridad para evitar la propagación de virus, entre las cuales se encontraba la suspensión de clases en todos los niveles educativos, ante lo cual la Universidad Nacional Autónoma de México acató dichas medidas en las actividades presenciales de todos sus planteles a partir del 17 de marzo de 2020, motivo por el cual se adoptó temporalmente la educación remota y digital como una alternativa a la educación escolarizada.

Estas alternativas de educación presentaron obstáculos como la falta de acceso de los estudiantes a las tecnologías, ya que en la educación remota es fundamental el uso de dispositivos electrónicos como computadoras, tabletas digitales o teléfonos celulares, los cuales no están al alcance de todos los alumnos por sus altos costos.

Esta situación se agrava si el estudiante requiere de un *software* especial para el desarrollo de sus actividades académicas y de aprendizaje. Un ejemplo de ello se suscitó en las facultades de Artes y Diseño y de Ingeniería, las cuales requerían de recursos tecnológicos especializados acordes a sus carreras. Ante tal situación, se desarrolló un primer proyecto piloto el cual consistió en la creación de laboratorios virtuales con acceso remoto.

En el caso concreto de la Facultad de Estudios Superiores Aragón (FES Aragón) se detectó que un número importante de alumnos no contaba con equipos de cómputo con el procesamiento necesario para el correcto desarrollo académico, es decir, las computadoras de los alumnos carecían de la capacidad requerida para el funcionamiento adecuado de determinados programas, así como de *software* especializado de costo elevado.

Por ejemplo, *CorelDRAW* es un *software* que la población estudiantil requiere y este tiene un precio anual de \$5,865.00 pesos mexicanos, con la posibilidad de realizar una compra única de \$11,815.00 pesos mexicanos, lo cual es inaccesible para la mayoría de los estudiantes, considerando que en México se tiene un ingreso diario de \$158.80 pesos por perceptor, según la Encuesta Nacional de Ingresos y Gastos de los Hogares 2020 (INEGI, 2021).

Para atender esta problemática, la FES Aragón solicitó la creación de un laboratorio virtual como una solución que, adicionalmente, fungió como una oportunidad para fortalecer las habilidades tecnológicas de los alumnos.

Un laboratorio virtual es una simulación de prácticas manipulativas que pueden ser hechas por el/la estudiante, lejos de la universidad y del docente (Monge-Nájera, Rivas Rossi y Méndez Estrada, 2002). Es decir, es un espacio virtual al que los alumnos pueden acceder para realizar prácticas y actividades académicas, y para poder ingresar al laboratorio se requiere que los estudiantes tengan acceso a los equipos de cómputo, vía remota.

Para lograr que los alumnos tengan este acceso se desarrollaron una serie de pasos técnicos, dentro de los cuales se encuentra la instalación y configuración del *Directorio activo*.

Un directorio es una estructura jerárquica que almacena información sobre objetos en la red. Un servicio de directorio, como los Servicios de Dominio de *Active Directory (AD DS)*, proporciona los métodos para almacenar datos de directorio y ponerlos a disposición de los usuarios y administradores de la red. Por ejemplo, *AD DS* almacena información sobre cuentas de usuario, como nombres, contraseñas, números de teléfono, etc., y permite que otros usuarios autorizados en la misma red accedan a recursos compartidos en el dominio.

Active Directory almacena información sobre objetos en la red y hace que esta información sea fácil de encontrar y usar para administradores y usuarios, utiliza un almacén de datos estructurados como base, para una organización lógica y jerárquica de la información del directorio (Microsoft, 2022).

La instalación y uso del *Directorio Activo* en un ambiente de laboratorio virtual es el objeto de estudio del presente reporte, y es de vital importancia para su pleno funcionamiento, ya que a través del directorio se proporciona la administración segura a equipos de cómputo que se encuentran dentro de su dominio.

2. OBJETIVO

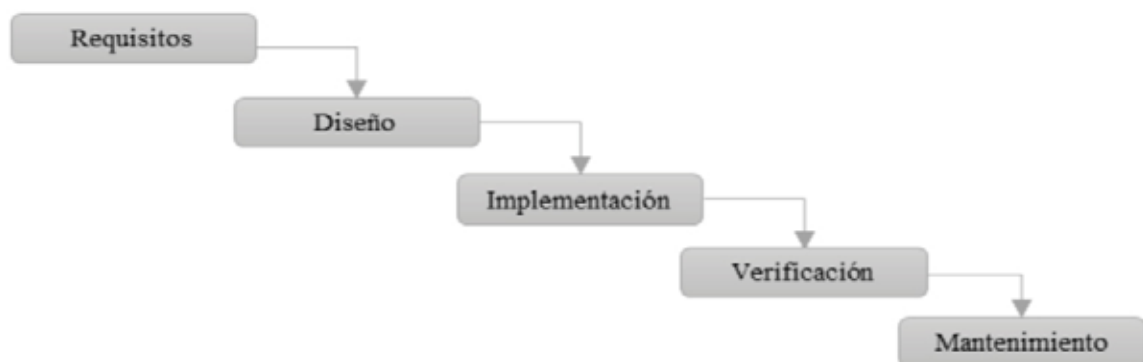
Implementar un *Directorio Activo* como parte de la infraestructura necesaria para el funcionamiento del laboratorio virtual que ofreciera a los estudiantes de la FES Aragón equipos de cómputo con características de *software* y *hardware* eficientes para sus actividades académicas.

3. METODOLOGÍA

Para la implementación del *Directorio Activo* se empleó el modelo en cascada (figura 1), ya que cada fase debe completarse antes de que pueda iniciar la siguiente.

Figura 1

Adaptado del Modelo en cascada, de Barry Boehm e Ian Sommerville, 1985



De manera general, en la primera fase se definieron los requisitos necesarios para la implementación del laboratorio virtual; durante la fase de diseño se establecieron especificaciones tales como el *hardware* a utilizar, tomando en cuenta los recursos disponibles en el Centro de Datos de la DGTIC; en la fase de implementación se realizó la instalación y configuración del *Directorio Activo*, equipos de cómputo y *Red Privada Virtual* conocida como *VPN*; en la fase de verificación se ejecutaron diversas pruebas y validaciones y, finalmente, durante la fase de mantenimiento se detectaron algunas características a modificar, y se aplicaron las correcciones necesarias.

4. DESARROLLO

En la fase de especificación de requisitos, se definieron los requerimientos del laboratorio virtual, se planteó la necesidad de contar con equipos de cómputo, una *VPN* y un directorio para la administración de los equipos.

Una vez definidos los requerimientos, se comenzó con la fase de diseño donde se establecieron las especificaciones de *hardware* y *software* a utilizar, y se determinó generar veinte instancias que son el número de equipos que el Centro de Datos tiene capacidad de otorgar con un funcionamiento óptimo. El sistema operativo que se eligió instalar en estas veinte instancias fue *Microsoft Windows*, que es el sistema operativo más conocido por los usuarios finales, y es compatible con el *software* de aplicación a instalar (*Corel Draw Graphics Suite* y *Adobe Creative Suite*).

Posteriormente, se seleccionó el servicio de directorio que se utilizaría, entre dos opciones que brindan el mismo servicio: *Open LDAP* como solución de código abierto, y el *Directorio Activo*. La elección se basó en la interoperabilidad con el sistema operativo de la infraestructura, a nivel de administración y de equipos finales (instancias con acceso remoto); también se buscó contar con una interfaz gráfica conocida y de manejo fácil e intuitivo para su gestión. Estas características se encontraron en el *Directorio Activo*, por lo que fue seleccionado para este proyecto.

Posteriormente, se revisó la documentación oficial de *Microsoft* para preparar la instalación y configuración del *Directorio Activo*. Además de la consulta en la red de documentación técnica y de casos de instalación exitosos, se contó con la información de contenidos de capacitación, tales como el curso "*Instalación y configuración de Microsoft Windows Server*" (Gómez, 2022) impartido por *Udemy*¹.

Al terminar de recabar información se continuó con el proceso de instalación y configuración, definiendo el direccionamiento necesario para todos los equipos y dos diferentes segmentos de red, uno para el *Directorio Activo* y los equipos, y otro para el túnel de la *VPN*.

4.1 CONFIGURACIÓN DEL DIRECTORIO ACTIVO

En la fase de implementación se realizó la instalación y configuración del *Directorio Activo* (*Active Directory*).

Un *Active Directory* almacena información sobre objetos en la red y hace que esta información sea fácil de encontrar y de usar para administradores y usuarios. El *Active Directory* utiliza un almacén de datos estructurados como base para una organización lógica y jerárquica de la información del directorio (Microsoft, 2021).

1 Udemy es una tienda virtual de aprendizaje y enseñanza en línea (Udemy, 2023)

Este almacén de datos contiene información sobre los objetos del *Directorio Activo*; suelen incluir recursos compartidos, tales como servidores, volúmenes, impresoras y las cuentas de usuario y de equipo de la red.

La seguridad se integra en *Active Directory* mediante la autenticación de inicio de sesión y el control de acceso a los objetos del directorio. Con un único inicio de sesión de red, los administradores pueden administrar los datos del directorio y la organización a través de su red, y los usuarios de red autorizados pueden tener acceso a los recursos en cualquier parte de la red. La administración basada en la directiva facilita la administración de incluso la redes más complejas (Microsoft, 2022).

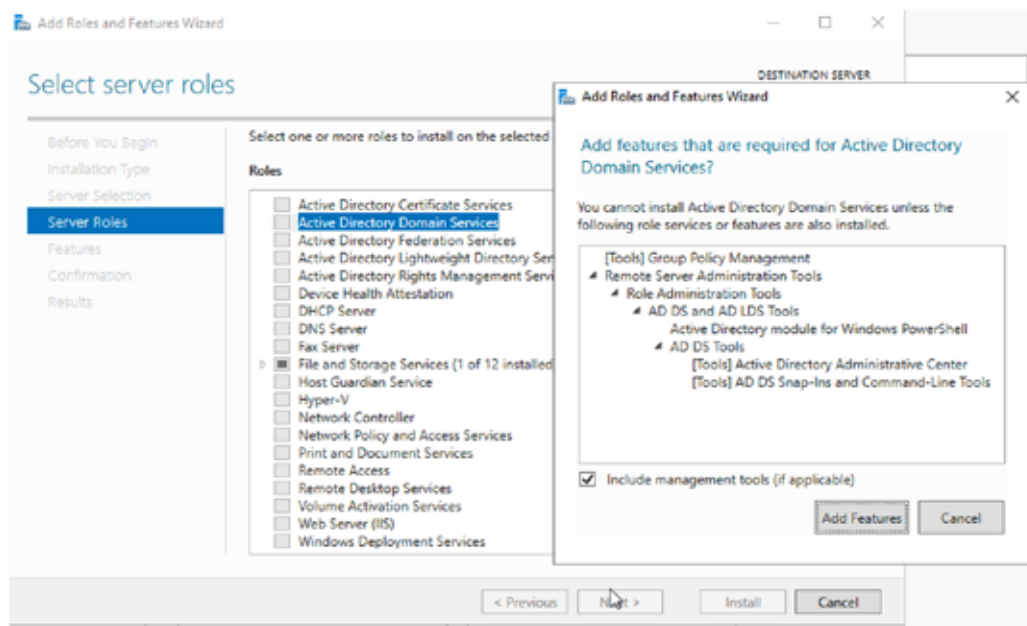
Para el proceso de instalación del *Directorio Activo*, se generó una máquina virtual en *VMware* configurada con memoria *RAM* de 8 GB y disco duro de 100 GB, en la que se instaló el Sistema Operativo *Windows Server 2019*. El siguiente paso que se ejecutó fue cambiar el nombre de la máquina, ya que es un requisito indispensable para la instalación de un controlador de dominio, en este caso el nombre seleccionado para el equipo fue "AD".

En las configuraciones de *Red* para el *Directorio Activo*, como se mencionó anteriormente, se definieron dos segmentos de red y direccionamiento utilizado: se estableció una dirección *IP* para el *Directorio Activo*, que también se configuró para la dirección del *DNS*.

A continuación, se realizó la instalación del *AD*, se agregó el rol de *Servicio de Directorio Activo*, a través del asistente, el cual consiste en una serie de ventanas que indican los pasos a seguir. La principal función de este asistente es habilitar el rol del *Servicio del Directorio Activo* (figura 2), así como las características de administración.

Figura 2

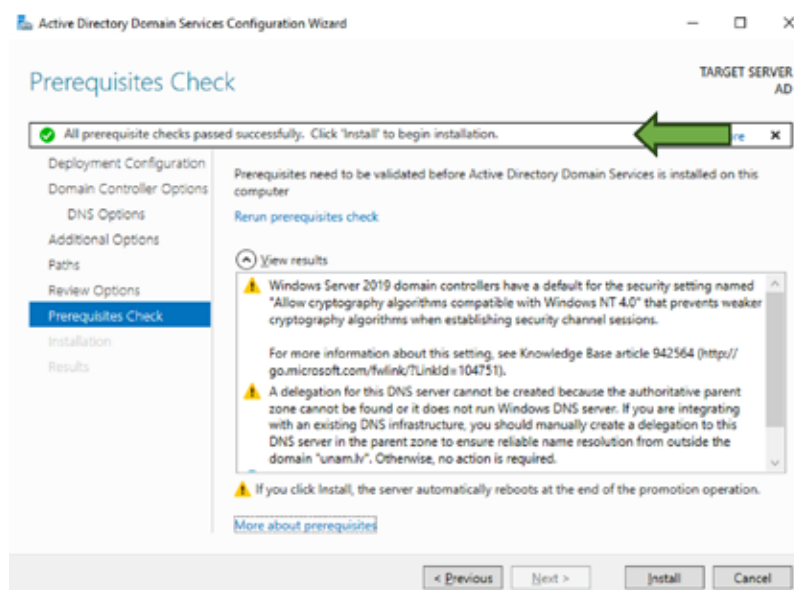
Rol de Servicio de Directorio Activo (captura de pantalla)



Tras finalizar la configuración de las características se promovió el equipo como controlador de dominio, antes de terminar este proceso, se revisó una ventana, la cual indicó si los pre-requisitos fueron cumplidos. En caso favorable, permite seguir con la instalación; en caso contrario mostraría un mensaje de error e impediría continuar con el proceso.

Figura 3

Validación de pre-requisitos (captura de pantalla)



Cuando se realizó este paso por primera vez, no se habían configurado los parámetros de red, por lo que se tuvo que cancelar la instalación. Se volvió a configurar los parámetros de red para iniciar nuevamente a partir del paso para promover el equipo como controlador de dominio. En el segundo intento, el resultado fue exitoso y se completó la instalación.

4.2 GENERACIÓN DE USUARIOS, GRUPOS, UNIDADES ORGANIZATIVAS Y OBJETOS DE DIRECTIVA DE GRUPO

Una vez instalado el *Directorio Activo*, se configuró la unidad organizativa llamada *fesaragon*, la cual funciona como contenedor para usuarios y grupos.

Se generaron tres grupos dentro de la unidad organizativa, denominados: Alumnos, Profesores y TI. Se crearon 24 usuarios: 20 proporcionados a los alumnos para ingresar a los equipos, dos asignados a profesores y dos usuarios de gestión para personal de Tecnologías de la Información. Se observó la siguiente nomenclatura para la notación de cada uno:

- Grupo Alumnos: Usuario0X
- Grupo Profesores: Profesor0X
- Grupo TI: TI0X

Esta organización y asignación de usuarios permitió la aplicación correcta de políticas.

Posteriormente, se crearon los objetos de directiva de grupo conocidos como *GPO*, referidos a las políticas que controlan el entorno de trabajo de cuentas de usuario y cuentas de equipo. Su principal función en el laboratorio virtual es aplicar permisos a los usuarios dependiendo del grupo al que pertenecen. Se configuraron y aplicaron tres *GPO* a la unidad organizativa *fesaragon*, que fueron nombrados como: *GPO_NoControlPanelFes*, *GPO_NoShurDownFes*, *GPO_WPFes*. La descripción de cada *GPO*, es la siguiente:

- *GPO_NoControlPanelFes* se aplicó a los usuarios que pertenecen al grupo Alumnos. Bloquea el acceso al Panel de Control para evitar que se realicen cambios que afecten el funcionamiento de los equipos.
- *GPO_NoShurDownFes* se aplicó a los grupos Alumnos y Profesores. Evita que apaguen los equipos.
- *GPO_WPFes* se aplicó a los grupos Alumnos, Profesores y TI. Otorga a cada usuario un fondo de pantalla específico.

4.3 INCORPORACIÓN DE EQUIPOS A DIRECTORIO ACTIVO

Una vez instalado el *Directorio Activo*, se continuó con el proceso para agregar los equipos al directorio. Para esto se generaron 20 instancias para laboratorios virtuales, que se instalaron en *VMware* con Sistema Operativo *Windows 10 Education N*. Cada equipo cuenta con 10 GB de *RAM* y disco duro de 250 GB; los programas con los que cuentan son la suite de *Adobe* y *Corel*. En los equipos se habilitó el *Escritorio Remoto* para acceder por el protocolo *RDP*.

Después de configurar los parámetros de red, se incorporaron los equipos al *Directorio Activo*. Se comprobó que existía comunicación entre *PC* y *Directorio Activo* a través de la utilidad *ping*. Al verificar la comunicación, se agregó el equipo de cómputo al dominio y se reinició el equipo para aplicar el cambio. Este proceso se realizó en los 20 equipos de cómputo, y se verificó que cada máquina estuviera unida correctamente al dominio.

4.4 ACCESO REMOTO

Los usuarios ingresan de manera remota a los laboratorios virtuales a través de una *VPN*, para lo cual deben instalar el *software OpenVPN*, así como agregar un certificado y usar las credenciales otorgadas por el Centro de Datos. Una vez conectados a la *VPN* se accede a los equipos por medio de *Escritorio remoto*, herramienta incluida en el sistema operativo de *Windows*, por lo que no es necesario que el usuario realice algún proceso de instalación, sólo se ingresa tanto la *IP* del equipo como el usuario asignados.

Este método de acceso proporciona dos factores de autenticación: el primer factor consta de las credenciales que solicita la *VPN*, esto es, el usuario y la contraseña que solicita para la conexión. El segundo factor son las credenciales que el usuario ingresa para acceder al equipo de cómputo a través del *Escritorio remoto*, con lo que se logra un acceso seguro a los laboratorios virtuales.

Se inició la fase de verificación de la operación y funcionalidad, a través de la ejecución de pruebas al ingresar credenciales para comprobar el acceso e inicio de sesión a los equipos vía remota.

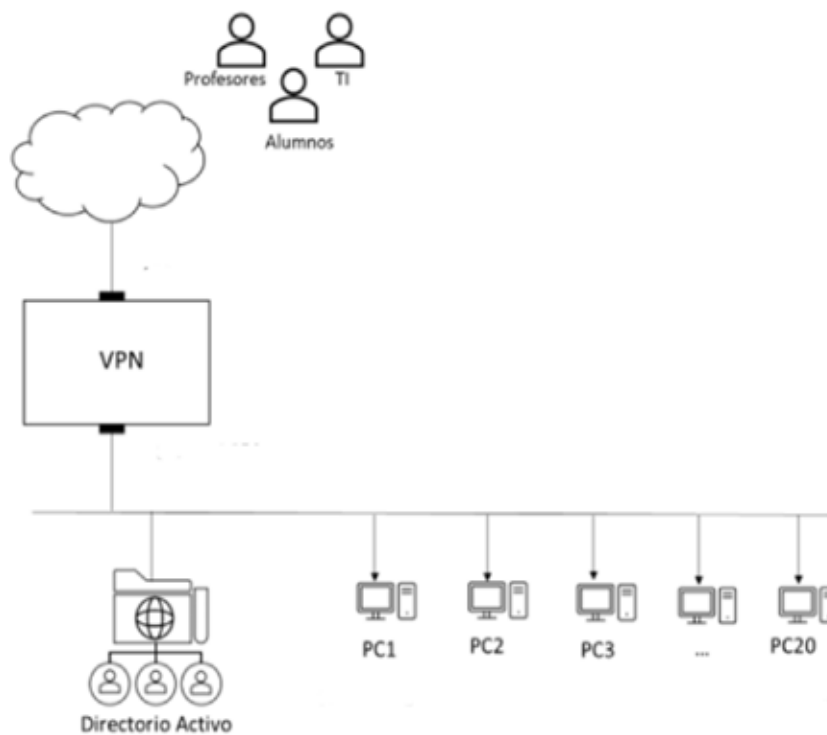
Finalmente, en la fase de mantenimiento se entregaron las credenciales al personal de la FES Aragón para la verificación de los accesos y aclaración de dudas, y se revisó el procedimiento para contar con un asesoramiento más detallado.

5. RESULTADOS

La infraestructura de laboratorios virtuales para la FES Aragón se implementó como se definió en los requerimientos iniciales. Se instaló y configuró el *Directorio Activo*, así como veinte equipos con las especificaciones de *software* establecidas y el servicio de VPN (figura 4).

Figura 4

Infraestructura del laboratorio virtual de la FES Aragón



Se realizaron pruebas de acceso para validar que los usuarios pudieran ingresar a los equipos de cómputo y que al acceder cada usuario obtuviera los permisos adecuados a su rol, con resultados satisfactorios.

Cabe mencionar que en la instalación y configuración del *Directorio Activo* se encontró un error provocado al no cumplir con los pre-requisitos de instalación: la falta de asignación de dirección IP en el directorio antes de instalar. Se completó esta configuración para cumplir con los requisitos solicitados, y se continuó la instalación hasta la implementación exitosa del *Directorio Activo* para el laboratorio virtual.

Es indispensable aplicar políticas de seguridad a través del *Directorio Activo*, ya que permiten asegurar que cada usuario cumpla con su rol asignado y limitar sus permisos; de esta manera, la información y el equipo es resguardado ante acciones inapropiadas o protegido de posibles errores. Por ejemplo, en el caso del laboratorio virtual, al perfil de un alumno no se le permite apagar los equipos de cómputo, ya que causaría que el siguiente alumno asignado al equipo no pudiera ingresar y perdiera el horario asignado.

6. CONCLUSIONES

El *Directorio Activo* juega un papel clave en toda la infraestructura del laboratorio virtual de la FES Aragón, ya que permite la administración de equipos y usuarios, y proporciona seguridad al obstaculizar o permitir el ingreso a los equipos de cómputo según el rol asignado a cada usuario.

Entre las ventajas de la ejecución del *Directorio Activo* para la comunidad estudiantil de la FES Aragón se destaca que los alumnos son beneficiados con equipos de cómputo óptimos para poder realizar sus prácticas y actividades académicas desde cualquier lugar por medio de un acceso remoto, seguro y confiable, para que cuenten con una infraestructura que les da las posibilidades de lograr un mejor desempeño académico.

7. AGRADECIMIENTOS

Agradezco a la Universidad Nacional Autónoma de México por brindar la infraestructura necesaria para realizar el proyecto que ayudará a estudiantes a realizar sus actividades académicas. Asimismo, agradezco al Ing. Pedro Bautista Fernández, M.T.I.A. Jesús Salvador Fernández Rauda y compañeros del Centro de Datos de la DGTIC por su apoyo y el aprendizaje obtenido.

REFERENCIAS BIBLIOGRÁFICAS

- Gómez, J. (s.f.) *Instalación y configuración de Microsoft Windows Server*. Udemy. <https://www.udemy.com/course/cursos-instalacion-y-configuracion-de-windows-server-2016/>.
- INEGI. (2021). *Encuesta Nacional de Ingresos y Gastos de los Hogares 2020* (ENIGH).
- Microsoft. (2022). *Active Directory Domain Services Overview*. Recuperado 26 de junio de 2023. <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>
- Microsoft. (2021). *AD DS Getting Started*. Recuperado el 29 de julio de 2023. <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/ad-ds-getting-started?source=recommendations>.
- Monge-Nájera, J. Rivas Rossi, M. & Méndez-Estrada, V. (2002). *La evolución de los laboratorios virtuales durante una experiencia de cuatro años con estudiantes a distancia*. San José, Costa Rica. Centro para el Mejoramiento de los Procesos Académicos (CEMPA), UNED. Recuperado el 26 de junio de 2023. <https://observatoriotecedu.uned.ac.cr/media/evollab4.pdf>
- Singh, A. (2021). *Agile & Scrum* ([edition unavailable]). Babelcube Incorporated. <https://www.perlego.com/book/2984684/agile-scrum-pdf>
- Udemy. (2022). *¿Quiénes somos?* Recuperado el 26 de junio de 2023. <https://www.udemy.com>.



**Cuadernos Técnicos Universitarios
de la DGTIC**



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

ISSN en trámite