

DISEÑO E IMPLEMENTACIÓN DE LA MIGRACIÓN DEL PORTAL WEB DEL LABORATORIO
DE ENSEÑANZA Y PRÁCTICAS HACIA UNA ARQUITECTURA BASADA EN
CONTENEDORES ORQUESTADOS CON KUBERNETES

MONICA NATALIA HERREÑO FANDIÑO
FACULTAD DE LAS TICS
INGENIERIA DE TELECOMUNICACIONES

UNIVERSIDAD SANTO TOMAS

TUTOR ACADEMICO: FERNANDO PRIETO BUSTAMANTE
BOGOTA 2026

RESUMEN

El presente proyecto tiene como objetivo llevar a cabo la migración de un portal web utilizado para pruebas y diseños con clientes, hacia una arquitectura más robusta basada en contenedores orquestados con Kubernetes, desplegada sobre una plataforma de virtualización empresarial propia de Huawei (FusionCompute)[37]. Esta iniciativa surge ante la necesidad de mejorar la estabilidad, disponibilidad y eficiencia del servicio, considerando las limitaciones del entorno actual, el cual presenta fallas ocasionales y dificultades en su gestión.

El portal web, desarrollado bajo una arquitectura full stack que integra frontend, backend y base de datos, es una herramienta fundamental para la ejecución de actividades de entrenamiento y pruebas técnicas por parte de ingenieros y aliados. Sin embargo, su despliegue actual en un entorno contenedorizado de terceros (Portainer), gestionado mediante herramientas limitadas y no garantiza niveles adecuados de continuidad operativa, lo que impacta directamente en la experiencia de los usuarios y en la eficiencia del laboratorio.

Para dar solución a esta problemática, se propone una arquitectura basada en Kubernetes, que permite automatizar la gestión de contenedores, mejorar la escalabilidad de los servicios, implementar mecanismos de alta disponibilidad y facilitar la administración del sistema. Esta nueva solución se integra con la plataforma de virtualización FusionCompute, permitiendo aprovechar mejor los recursos tecnológicos y alinearse con los estándares empresariales.

Finalmente, el desarrollo del proyecto no solo contribuye a la optimización de los servicios tecnológicos del laboratorio, sino que también fortalece las competencias del estudiante en áreas como virtualización, contenedores y gestión de infraestructuras IT. Los resultados esperados evidencian mejoras significativas en la estabilidad, disponibilidad y gestión del portal web, consolidando la viabilidad de la solución propuesta en entornos corporativos.

TABLA DE CONTENIDO

RESUMEN	2
INTRODUCCIÓN	5
CAPÍTULO I. MARCO GENERAL DEL PROYECTO	6
1.1 PLANTEAMIENTO DEL PROBLEMA.....	6
1.2 ALCANCE	7
1.3 JUSTIFICACIÓN	8
1.4 OBJETIVOS	9
CAPÍTULO II. MARCO TEÓRICO Y CONCEPTUAL	10
2.1 BASES TEORICAS	11
2.2 BASES LEGALES.....	19
2.3 ESTADO DEL ARTE	23
CAPITULO III. MARCO METODOLÓGICO	26
3.1 ENFOQUE Y TIPO DE INVESTIGACIÓN	26
3.2 DISEÑO METODOLOGICO	28
3.3 HERRAMIENTAS UTILIZADAS	36
3.4 VARIABLES Y CONSIDERACIONES.....	37
3.5 CONCLUSIÓN METODOLÓGICA	39
CAPÍTULO IV. DESARROLLO DE LA PROPUESTA.....	40
4.1 DESARROLLO OBJETIVO GENERAL.....	40
4.2 DESARROLLO PRIMER OBJETIVO GENERAL	40
4.3 DESARROLLO SEGUNDO OBJETIVO GENERAL	41
4.4 DESARROLLO TERCER OBJETIVO GENERAL	42
4.5 DESARROLLO CUARTO OBJETIVO GENERAL.....	43
4.6 DESARROLLO QUINTO OBJETIVO GENERAL	44
CAPÍTULO V. ANALISIS	45
CAPÍTULO VI. RESULTADOS	47
CAPÍTULO VII. DISCUSION	49
CAPÍTULO VIII. CONCLUSIONES	51
CAPÍTULO IX. RECOMENDACIONES.....	53
REFERENCIAS BIBLIOGRAFICAS.....	55
ANEXO A.....	57

INDICE DE TABLAS

Tabla 1: Componentes principales del backend	32
Tabla 2: Componentes principales del frontend	33
Tabla 3: Elementos necesarios para la configuración de kubernetes	34
Tabla 4: Recursos usados para la comunicación entre backend y frontend	34
Tabla 5: Problemas durante el despliegue	35

INDICE DE FIGURAS

Ilustración 1 Arquitectura de página web en portainer	29
Ilustración 2. Arquitectura final desplegada en fusioncompute.	30

INTRODUCCIÓN

En el contexto actual de la transformación digital, las organizaciones dependen cada vez más de plataformas tecnológicas robustas que garanticen la disponibilidad, estabilidad y eficiencia de sus servicios digitales. En particular, los entornos de laboratorio y entrenamiento tecnológico cumplen un papel fundamental al permitir la práctica y capacitación a los ingenieros de Huawei como usuarios externos, aliados estratégicos, mediante el acceso a herramientas y recursos especializados a través de plataformas web. Durante este tiempo se identificó que uno de los servicios críticos del laboratorio corresponde a un portal web utilizado para la gestión y ejecución de actividades de entrenamiento y pruebas técnicas. Dicho portal está soportado por una arquitectura de tipo full stack, que contiene componentes de frontend, backend y bases de datos, desplegados mediante tecnologías de contenedorización sobre infraestructura virtualizada. No obstante, el despliegue actual presenta limitaciones en términos de estabilidad y disponibilidad del servicio, además de que es una plataforma de terceros y no propia de la empresa. Esta situación genera la necesidad de evaluar y proponer una alternativa tecnológica que permita optimizar el entorno de ejecución del portal, alineándolo con plataformas más robustas y acordes con los estándares de infraestructura empresarial. En este sentido, el presente proyecto propone el diseño y ejecución de una migración del portal web hacia una arquitectura basada en contenedores orquestados mediante Kubernetes, ejecutándose sobre una plataforma de virtualización propia (FusionCompute), con el objetivo de mejorar la estabilidad, disponibilidad y gestión de los servicios del laboratorio. El desarrollo de esta propuesta busca aportar tanto a la optimización de los procesos tecnológicos de la organización como al fortalecimiento de las competencias profesionales adquiridas durante la formación académica.

CAPÍTULO I. MARCO GENERAL DEL PROYECTO

1.1 PLANTEAMIENTO DEL PROBLEMA

En el departamento de Service Partner Enablement se dispone de un laboratorio al que se accede a través de un portal web. Dicho portal permite el ingreso de clientes, así como de ingenieros internos y externos, con el fin de realizar pruebas, prácticas, entrenamientos y otros usos relacionados. A través de esta plataforma es posible realizar reservas para el acceso a dispositivos de almacenamiento, direcciones IP, equipos de acceso y transmisión. Asimismo, el portal ofrece acceso a plataformas de Huawei, recursos como demostraciones de productos, laboratorios prácticos y entornos especializados, los cuales resultan fundamentales para los procesos de capacitación. Por esta razón, la disponibilidad y el correcto funcionamiento del portal son factores críticos para garantizar la continuidad de las actividades y la calidad del servicio ofrecido por el laboratorio.

Actualmente, el portal web se encuentra desplegado bajo una arquitectura de aplicación full stack, compuesta por frontend, backend y base de datos, ejecutada en un entorno contenedorizado sobre infraestructura virtualizada mediante la herramienta Portainer [36], la cual corresponde a una plataforma de terceros. Si bien esta solución permite una administración básica de contenedores, no fue diseñada para soportar de manera robusta escenarios de alta disponibilidad ni para gestionar de forma eficiente entornos con múltiples usuarios concurrentes y servicios críticos.

Como consecuencia de lo anterior, se han presentado problemas recurrentes de estabilidad y disponibilidad, evidenciados en interrupciones inesperadas del servicio, caídas del portal y fallos que afectan el acceso de los usuarios. Estas situaciones obligan a realizar una supervisión constante del entorno por parte del personal técnico, incrementando la carga operativa y generando retrasos en la ejecución de entrenamientos y pruebas. Adicionalmente, las interrupciones impactan negativamente la experiencia del usuario, disminuyen la confiabilidad del laboratorio y afectan la percepción de calidad del servicio ofrecido.

Otro aspecto relevante del problema es que, al tratarse de una aplicación desplegada sobre una plataforma de terceros, se limita la implementación de mecanismos avanzados de orquestación, escalabilidad automática y tolerancia a fallos. Esta dependencia tecnológica dificulta la adopción de estándares empresariales y restringe el control sobre la infraestructura, impidiendo realizar ajustes que permitan responder de manera eficiente a picos de demanda o a fallos parciales del sistema.

En este contexto, se evidencia la necesidad de migrar el portal web hacia una arquitectura más estable, controlada y alineada con una infraestructura de virtualización empresarial propia, que permita mejorar la gestión de los servicios, reducir las interrupciones y garantizar la continuidad operativa del laboratorio. La falta de esta migración representa un riesgo para la operación del laboratorio y para el cumplimiento de sus objetivos de capacitación, lo que justifica el desarrollo del presente proyecto como una solución tecnológica orientada a fortalecer la estabilidad, disponibilidad y confiabilidad del portal web.

1.2 ALCANCE

El presente proyecto abarca el análisis detallado del entorno actual en el cual se encuentra desplegado el portal web del laboratorio, incluyendo la identificación de su arquitectura de aplicación full stack, la forma en que están integrados los componentes de frontend, backend y base de datos, así como las condiciones de estabilidad y disponibilidad del servicio. Este análisis permite comprender el funcionamiento del sistema existente y establecer los requerimientos técnicos necesarios para su migración hacia una arquitectura más robusta.

El alcance del proyecto incluye la migración del portal web hacia una arquitectura basada en contenedores orquestados mediante Kubernetes, ejecutados sobre una plataforma de virtualización empresarial desarrollada por Huawei, específicamente FusionCompute. Esta migración contempla el despliegue de los componentes de frontend, backend y base de datos en contenedores, garantizando que la aplicación conserve su funcionalidad original y su correcto desempeño tras el proceso de migración.

Asimismo, el proyecto incluye la configuración básica del entorno de orquestación, la validación del funcionamiento de los servicios y la realización de pruebas orientadas a verificar mejoras en estabilidad, disponibilidad y gestión del entorno. Como resultado, se espera obtener un entorno más controlado, confiable y alineado con estándares de infraestructura empresarial, que reduzca las interrupciones del servicio y mejore la experiencia de los usuarios del laboratorio.

Es importante resaltar que el proyecto no contempla el desarrollo de nuevas funcionalidades, modificaciones en la lógica de la aplicación ni cambios en la interfaz del portal web. Tampoco incluye la expansión de la solución a otros laboratorios, entornos productivos externos o infraestructuras diferentes a las definidas para el proyecto. El enfoque se limita exclusivamente a la migración y optimización de la infraestructura tecnológica que soporta el portal web del laboratorio, con el fin de generar un caso de éxito que evidencie la aplicación de tecnologías de contenedorización y virtualización empresarial de Huawei en Colombia.

1.3 JUSTIFICACIÓN

El desarrollo del presente proyecto se justifica, en primer lugar, por la necesidad de mejorar la estabilidad y disponibilidad del portal web del laboratorio, el cual cumple un papel fundamental en la prestación de servicios de entrenamiento y pruebas técnicas para aliados estratégicos. Las interrupciones frecuentes del servicio afectan directamente la continuidad de las actividades, generan retrasos en los procesos de capacitación y disminuyen la confiabilidad del laboratorio como entorno tecnológico. Por tanto, resulta indispensable optimizar la infraestructura que soporta el portal para garantizar un funcionamiento continuo y confiable.

Adicionalmente, la infraestructura actual presenta limitaciones importantes al estar basada en una plataforma de terceros que no permite implementar de manera eficiente mecanismos avanzados de orquestación, escalabilidad y tolerancia a fallos. Esta situación obliga a una gestión manual constante y dificulta la respuesta ante fallos o incrementos en la demanda de usuarios. La adopción de una arquitectura basada en contenedores orquestados sobre una plataforma de virtualización empresarial conocida por la organización permitirá mejorar la gestión de los recursos, reducir la dependencia de soluciones externas y fortalecer el control sobre el entorno de ejecución del portal.

Desde el punto de vista organizacional, la implementación de esta solución contribuye a la estandarización de la infraestructura tecnológica, alineándola con las plataformas empresariales utilizadas por la compañía, como FusionCompute, desarrollada por la misma, y con tecnologías de orquestación como Kubernetes. Esto no solo mejora la calidad de los servicios ofrecidos, sino que también facilita futuras ampliaciones, mantenimientos y la replicación del modelo en otros entornos o laboratorios, generando un impacto positivo a nivel operativo y estratégico.

Finalmente, desde el enfoque académico y formativo, este proyecto representa una oportunidad significativa para aplicar conocimientos teóricos en un entorno real, integrando conceptos de virtualización, contenedorización y gestión de servicios de TI. La ejecución de esta solución fortalece competencias técnicas, analíticas y profesionales propias de la ingeniería en telecomunicaciones, al tiempo que permite documentar un caso práctico que evidencia la aplicación de tecnologías empresariales modernas. De esta manera, el proyecto no solo aporta a la mejora del laboratorio, sino que también contribuye al proceso de formación integral de la estudiante.

1.4 OBJETIVOS

Objetivo general

Diseñar e implementar la migración de un portal web que contiene frontend, backend y base de datos, hacia una arquitectura basada en contenedores orquestados con Kubernetes que será ejecutada sobre una plataforma de virtualización (FusionCompute), con el fin de mejorar la estabilidad y disponibilidad de los servicios ofrecidos.

Objetivos específicos

- Analizar la arquitectura actual del portal web del laboratorio y su esquema de despliegue, con el fin de documentar sus componentes, tecnologías utilizadas, forma de interconexión y principales limitaciones en términos de estabilidad, disponibilidad y gestión del entorno.
- Identificar las principales causas de inestabilidad y fallas en la disponibilidad del servicio.
- Diseñar una arquitectura objetivo basada en contenedores que permita una gestión más eficiente de los servicios.
- Documentar el proceso de migración del portal web al nuevo entorno de contenedores sobre la plataforma de virtualización empresarial.
- Evaluar los resultados obtenidos a partir de la migración, considerando mejoras en estabilidad, disponibilidad y gestión del entorno.

CAPÍTULO II. MARCO TEÓRICO Y CONCEPTUAL

El presente proyecto se desarrolla en el contexto de un laboratorio tecnológico perteneciente a la empresa Huawei, ubicado en Colombia, el cual está orientado a la ejecución de actividades de entrenamiento, capacitación y pruebas técnicas dirigidas a ingenieros, aliados estratégicos y usuarios externos. Este entorno tiene como finalidad proporcionar escenarios prácticos que simulan condiciones reales de operación en el ámbito de las telecomunicaciones y los sistemas de información, permitiendo fortalecer habilidades técnicas mediante el uso de plataformas especializadas.

El laboratorio dispone de una infraestructura tecnológica basada en entornos virtualizados, lo que facilita la ejecución de múltiples servicios y aplicaciones, así como el acceso remoto a recursos de cómputo, redes y almacenamiento. Dentro de estos servicios, se destaca un portal web que actúa como plataforma central para la gestión de las actividades académicas y técnicas, permitiendo a los usuarios acceder a materiales de formación, desplegar prácticas y realizar entrenamientos en tiempo real.

Dicho portal web ha sido desarrollado bajo una arquitectura de tipo full stack, compuesta por frontend, backend y base de datos, elementos que en conjunto permiten la interacción con los usuarios, el procesamiento de solicitudes y el almacenamiento de información generada durante las diferentes actividades del laboratorio. Actualmente, la aplicación se encuentra desplegada en un entorno contenedorizado sobre infraestructura virtualizada, empleando herramientas de gestión de contenedores de terceros, como Portainer.

No obstante, durante el desarrollo de las prácticas profesionales se identificaron diversas problemáticas asociadas al funcionamiento del portal, principalmente relacionadas con la estabilidad, disponibilidad y gestión del servicio. Entre los principales inconvenientes se evidencian interrupciones ocasionales, necesidad de supervisión constante del entorno, y limitaciones en el manejo eficiente de los recursos. Estas condiciones afectan directamente la continuidad de las actividades del laboratorio y la experiencia de los usuarios, quienes dependen de la plataforma para el desarrollo de sus procesos de formación.

Adicionalmente, el uso de una solución de terceros para el despliegue de la aplicación limita significativamente la capacidad de implementar mecanismos avanzados de orquestación, escalabilidad y alta disponibilidad. Esto restringe el control sobre la infraestructura tecnológica y dificulta la adopción de estándares empresariales alineados con las políticas y herramientas propias de la organización.

En este contexto, surge la necesidad de modernizar la infraestructura del laboratorio mediante la migración del portal web hacia una solución tecnológica más robusta, flexible y alineada con las tendencias actuales de la industria. Para ello, se propone la implementación de una arquitectura basada en contenedores orquestados mediante Kubernetes, desplegados sobre la plataforma de virtualización corporativa FusionCompute, desarrollada por Huawei.

FusionCompute permite la gestión eficiente de recursos de cómputo, red y almacenamiento en entornos empresariales, brindando mayor control sobre la infraestructura y facilitando la implementación de soluciones con altos niveles de disponibilidad, escalabilidad y automatización.

Por su parte, Kubernetes contribuye a la orquestación de contenedores, optimizando la administración de aplicaciones, la distribución de cargas de trabajo y la recuperación ante fallos.

La adopción de esta nueva arquitectura responde a la creciente necesidad de contar con plataformas tecnológicas resilientes, capaces de soportar múltiples usuarios concurrentes y garantizar la continuidad de los servicios digitales. Asimismo, esta transición se alinea con las tendencias actuales de transformación digital, donde la virtualización, la computación en la nube y la contenerización juegan un papel fundamental en la optimización de infraestructuras tecnológicas.

Finalmente, este proyecto se desarrolla en el marco de las prácticas profesionales de la estudiante de ingeniería en telecomunicaciones, constituyendo una oportunidad para aplicar conocimientos teóricos en un entorno real. A través de su ejecución, se fortalecen competencias en áreas como virtualización, administración de sistemas, redes y tecnologías de contenedores. De igual forma, el proyecto busca generar valor tanto a nivel organizacional, mediante la mejora del servicio del laboratorio, como a nivel académico, evidenciando un caso práctico del uso de tecnologías Huawei en Colombia.

2.1 BASES TEORICAS

Transformación Digital

La transformación digital se ha consolidado como un eje estratégico fundamental para las organizaciones en el contexto actual, caracterizado por la acelerada evolución tecnológica y la creciente digitalización de los entornos económicos y sociales. Desde una perspectiva teórica, la transformación digital no se limita a la adopción de tecnologías digitales, sino que implica un proceso profundo de cambio organizacional, que abarca estrategias, estructuras, capacidades, cultura y modelos de negocio [12].

De acuerdo con la literatura analizada en el documento de Miranda Torrez (2023), la transformación digital puede entenderse como un proceso estratégico mediante el cual las organizaciones integran tecnologías digitales para crear valor, mejorar su desempeño y responder de manera más eficaz a las demandas del entorno. En este sentido, la tecnología actúa como un habilitador, pero no como el único factor determinante del proceso [12].

Desde un enfoque basado en recursos y capacidades, la transformación digital requiere que las organizaciones desarrollen capacidades digitales, entendidas como la habilidad para combinar recursos tecnológicos, humanos y organizacionales con el fin de innovar y adaptarse. Estas capacidades permiten a las organizaciones aprovechar herramientas digitales como plataformas, sistemas de información y analítica de datos para redefinir procesos internos y externos [12].

El documento plantea que la transformación digital está impulsada por diversos factores internos y externos. Entre los factores internos se destacan el liderazgo estratégico, la cultura organizacional orientada a la innovación y la disponibilidad de recursos digitales. Por su parte, los factores externos incluyen la presión competitiva, los cambios en el entorno tecnológico, la globalización y eventos disruptivos como la pandemia, que aceleraron la adopción de soluciones digitales en múltiples sectores.

Optimización De Recursos Tecnológicos

La optimización de recursos tecnológicos es un componente clave dentro de los procesos de transformación digital, pues permite a las organizaciones utilizar sus infraestructuras tecnológicas de manera más eficiente y orientada a generar valor empresarial. En este contexto, la optimización de Tecnologías de la Información (TI) se define como la capacidad de maximizar el rendimiento de los recursos, sistemas y tecnologías existentes para mejorar el desempeño global y alcanzar mejores resultados organizacionales [13].

En la era digital, la transformación digital impulsa a las organizaciones a aprovechar al máximo sus recursos tecnológicos, lo que incluye el uso eficiente de plataformas en la nube, virtualización y modelos híbridos de infraestructura. Esta optimización permite reducir costos, aumentar la eficiencia operativa y mejorar la productividad, ya que los recursos disponibles se emplean de manera estratégica para soportar las actividades críticas del negocio [13].

Además, la optimización de TI contribuye a mejorar la flexibilidad y adaptabilidad de las organizaciones frente a las demandas cambiantes del mercado. Al contar con sistemas y tecnologías que responden eficientemente a nuevas necesidades, las empresas pueden adaptarse con mayor rapidez, reforzar su estabilidad operativa y ofrecer mejores experiencias tanto a clientes internos como externos [13].

Entre las principales ventajas de optimizar recursos tecnológicos se encuentran:

- **Eficiencia:** permite aprovechar al máximo los recursos existentes, reduciendo costos y eliminando desperdicios [13].
- **Flexibilidad:** facilita la adaptación a cambios y nuevas necesidades del negocio [13].
- **Estabilidad:** disminuye las interrupciones y fallas, mejorando la confiabilidad de los sistemas [13].
- **Agilidad:** potencia la capacidad de responder rápidamente a nuevos retos u oportunidades en el mercado [13].

Una estrategia común para lograr la optimización de recursos tecnológicos es la adopción de enfoques híbridos de infraestructura, combinando nubes públicas, privadas y entornos tradicionales de TI. Esta estrategia permite gestionar cargas de trabajo de forma más eficiente, ubicando datos y aplicaciones donde sean más eficaces y aporten mayor valor, lo que a su vez se traduce en un uso optimizado de los recursos disponibles.

Gestión De Servicios de TI (IT Service Management)

La Gestión de Servicios de Tecnologías de la Información (IT Service Management, ITSM) se define como el conjunto de prácticas y procesos orientados a diseñar, entregar, operar y mejorar de manera continua los servicios de TI que una organización ofrece a sus usuarios. De acuerdo con Atlassian, el enfoque de ITSM no se centra únicamente en la infraestructura o en los componentes tecnológicos, sino en la forma en que la tecnología se gestiona como un servicio que aporta valor al negocio y satisface las necesidades de los usuarios finales. Este enfoque permite que las organizaciones estructuren sus operaciones de TI de manera ordenada, repetible y alineada con sus objetivos estratégicos [15].

Desde la perspectiva de ITSM, los servicios de TI deben gestionarse mediante procesos estandarizados que faciliten la atención de solicitudes, la resolución de incidentes, la administración del conocimiento y la gestión de cambios. Estas prácticas buscan garantizar la continuidad del servicio, reducir errores operativos y mejorar la eficiencia en la prestación de los servicios tecnológicos. Atlassian resalta que la adopción de ITSM contribuye a mejorar la productividad de los equipos de TI, ya que proporciona claridad sobre roles, responsabilidades y flujos de trabajo, permitiendo una respuesta más rápida y controlada ante fallos o requerimientos de los usuarios [15].

Hablando de una migración de una página web, la Gestión de Servicios de TI aporta un soporte teórico clave para justificar la forma en que se administran las actividades del proyecto. La migración de un sitio web implica cambios significativos en infraestructura, aplicaciones y servicios, por lo que la aplicación de principios de ITSM permite gestionar adecuadamente los riesgos, atender posibles incidentes durante el proceso y asegurar que el servicio web continúe cumpliendo con los niveles de calidad esperados por los usuarios.

Infraestructura Empresarial

La arquitectura empresarial provee un marco amplio que permite entender cómo la infraestructura tecnológica se integra con otros componentes organizacionales para apoyar la misión y los objetivos estratégicos de la empresa. En este marco, la infraestructura no se considera de forma aislada, sino como parte de un sistema interrelacionado que incluye plataformas computacionales, redes de datos, dispositivos de almacenamiento y las aplicaciones que operan sobre ellos, así como los procesos para administrar y evolucionar estos recursos tecnológicos [16].

La arquitectura tecnológica es la perspectiva que define la estrategia de las plataformas y servicios tecnológicos que deben soportar las soluciones y procesos del negocio. Esta perspectiva contempla no solo los activos tecnológicos, sino también los mecanismos de almacenamiento de datos, redes de comunicación, centros de procesamiento y las tecnologías que facilitan la operación y la innovación dentro de la organización [16].

De esta manera, la infraestructura empresarial se vuelve un componente clave para lograr una alineación efectiva entre la tecnología y los objetivos organizacionales, permitiendo que los sistemas de información sean robustos, escalables y sostenibles. La correcta implementación y gestión de esta infraestructura contribuye a la mejora de la eficiencia operativa, la reducción de riesgos tecnológicos y la capacidad de responder ante cambios del entorno empresarial [16].

Transformación Digital y Servicios Tecnológicos

La transformación digital es un proceso integral mediante el cual una organización utiliza tecnologías digitales para modificar fundamentalmente la forma en que opera, crea valor y

satisface las necesidades de sus clientes. Según el artículo “Digital transformation: A multidisciplinary reflection and research agenda”, la transformación digital no solo consiste en adoptar herramientas tecnológicas, sino en redefinir los modelos de negocio, los procesos internos y la estructura organizacional para aprovechar de manera estratégica los recursos digitales disponibles. En este sentido, transforma cómo las empresas compiten, interactúan con sus clientes y responden a los cambios del entorno tecnológico y económico [17].

El estudio identifica tres fases principales dentro del proceso de transformación digital: digitización, digitalización y transformación digital propiamente dicha. La digitización se refiere a la conversión de información analógica en digital, habilitando la automatización de tareas y la gestión electrónica de datos. La digitalización implica que las tecnologías digitales alteran y mejoran procesos existentes, como la comunicación entre la empresa y sus clientes o la gestión de información interna. Finalmente, la transformación digital representa un cambio profundo que conduce a la creación de nuevos modelos de negocio mediante la integración y uso estratégico de tecnologías digitales en toda la organización [17].

En el contexto de los servicios tecnológicos, la transformación digital tiene un impacto directo, dado que impulsa la creación, entrega y mejora continua de servicios digitales que antes no existían o eran poco eficientes. La adopción de tecnologías digitales permite a las organizaciones ofrecer servicios más personalizados, eficientes y orientados al usuario, utilizando plataformas digitales, análisis de datos y automatización para responder a las necesidades del mercado con mayor rapidez. De esta manera, los servicios tecnológicos se convierten en un componente fundamental de la propuesta de valor de la organización y en un motor de innovación y competitividad.

Arquitectura Full Stack

Una arquitectura full stack es un enfoque de diseño de sistemas en el que una aplicación se construye y gestiona de forma integral, abarcando tanto la parte de cliente (frontend) como la parte de servidor (backend), y conectándolas con capas de datos y servicios adicionales. En este contexto, el término stack se refiere a la combinación de tecnologías y componentes que trabajan juntos para ofrecer una solución completa, desde la interfaz de usuario hasta la lógica de negocio y el almacenamiento de datos [18].

En una arquitectura full stack, el frontend se encarga de la interacción con el usuario final y se desarrolla con tecnologías que pueden incluir HTML, CSS y frameworks de JavaScript, mientras que el backend maneja la lógica de negocio, las operaciones del servidor, la autenticación, la comunicación con la base de datos y la coordinación de procesos internos. La capa de base de datos se encarga del almacenamiento, consulta y gestión de la información que sustenta las funciones de la aplicación. Este enfoque permite que todos los elementos de una solución trabajen de forma cohesionada, lo cual facilita la evolución y escalabilidad del Sistema [18].

Además, una arquitectura full stack bien diseñada adopta mejores prácticas para asegurar que cada componente sea eficiente, mantenible y seguro. Estas mejores prácticas incluyen el uso de patrones de diseño que separan responsabilidades entre las distintas capas, la implementación de APIs bien estructuradas para comunicar frontend y backend, la utilización de mecanismos de autenticación y autorización sólidos, y la adopción de procesos automatizados de despliegue y pruebas para mejorar la calidad general del software [18].

La importancia de una arquitectura full stack radica en su capacidad para integrar y orquestar

todos los componentes de una aplicación web, reduciendo la complejidad asociada a fragmentaciones tecnológicas, mejorando la eficiencia del desarrollo y facilitando futuras actualizaciones, mantenimiento y escalabilidad de la solución. Este enfoque resulta especialmente relevante en proyectos de desarrollo web completo, donde la interfaz de usuario, la lógica de negocio y la gestión de datos deben operar con coherencia para brindar un servicio de calidad a los usuarios.

Contenedorización

La contenedorización es una técnica de despliegue de software que empaqueta el código de una aplicación junto con todas sus librerías, dependencias y archivos necesarios para que se ejecute de manera consistente en distintos entornos y plataformas. A diferencia de los métodos tradicionales que requieren adaptar el software a un sistema operativo específico, los contenedores permiten que una aplicación se ejecute sin necesidad de reconfiguraciones en cada infraestructura, lo que mejora su portabilidad y facilita el desarrollo y la operación [19].

Esta tecnología aporta diversas ventajas, entre las que se encuentran la portabilidad, la escalabilidad y la agilidad en el ciclo de vida del software. La portabilidad se logra porque los contenedores contienen todos los elementos que necesita la aplicación para funcionar, permitiendo que esta se despliegue en diferentes sistemas operativos sin cambios en el código. En cuanto a la escalabilidad, los contenedores son componentes ligeros que pueden ejecutarse de manera eficiente en el mismo servidor, lo que permite añadir o remover instancias de forma rápida según las necesidades de carga. Asimismo, al estar aislados del sistema operativo subyacente, los contenedores proporcionan mayor agilidad para realizar pruebas, actualizaciones o ajustes sin interferir con otros servicios [19].

La contenedorización opera mediante la creación de imágenes de contenedores, que son paquetes inmutables que contienen toda la información necesaria para ejecutar un programa. Estas imágenes se ejecutan mediante un motor de contenedores (container runtime) que gestiona la asignación de recursos y el aislamiento entre contenedores, haciendo que múltiples contenedores puedan convivir en un mismo sistema sin interferir entre sí. Este enfoque no solo favorece la eficiencia de recursos, sino que también genera entornos consistentes desde el desarrollo hasta la producción [19].

La migración de una página web, los contenedores ofrecen una forma eficiente de empacar, trasladar y desplegar servicios web sin que se vean afectados por cambios en la infraestructura subyacente. Por ejemplo, si tu sitio web está compuesto por varios servicios (como servidor web, base de datos o microservicios), la contenedorización te permite encapsular cada uno de estos componentes y mantenerlos funcionando de manera consistente durante la migración y después de ella. Esto facilita pruebas, facilita el mantenimiento del sistema y reduce el riesgo de errores por dependencias o incompatibilidades de entorno durante el proceso de migración.

Orquestación de contenedores con Kubernetes

La orquestación de contenedores es el proceso automatizado que permite aprovisionar, implementar, escalar y administrar aplicaciones ejecutadas dentro de contenedores de software en sistemas distribuidos. Esta práctica se vuelve esencial cuando se trabaja con múltiples contenedores en producción, ya que facilita la gestión del ciclo de vida completo de las

aplicaciones y asegura que estas respondan correctamente ante variaciones en la demanda, fallos o cambios de infraestructura [20].

En este contexto, Kubernetes se ha consolidado como la plataforma de orquestación de contenedores de código abierto más utilizada en el mundo. Desarrollado inicialmente por Google y actualmente mantenido por la Cloud Native Computing Foundation (CNCF), Kubernetes automatiza tareas complejas como el despliegue de contenedores, la escalabilidad automática, la asignación de recursos, el balanceo de carga y la autorreparación de servicios cuando ocurren fallos [20].

Kubernetes opera mediante la agrupación de máquinas (físicas o virtuales) llamadas nodos, organizadas en un clúster. Un componente central llamado plano de control se encarga de tomar decisiones globales sobre la carga de trabajo, como qué contenedores ejecutar, dónde hacerlo y cuántas réplicas mantener en ejecución. Los contenedores se organizan en unidades más grandes llamadas pods, que son las entidades mínimas que Kubernetes gestiona para ejecutar aplicaciones [20].

La orquestación con Kubernetes también incorpora herramientas y procesos que permiten la automatización declarativa: en lugar de ejecutar comandos manualmente para cada contenedor, los equipos definen el “estado deseado” de la aplicación mediante archivos de configuración (por ejemplo, en YAML), y Kubernetes se encarga de alcanzar y mantener ese estado por sí mismo. Esto simplifica enormemente las tareas de desarrollo y operaciones, reduce errores humanos y mejora la consistencia entre entornos de prueba y producción [20].

Además, Kubernetes no se limita al manejo de contenedores aislados: facilita características avanzadas como el escalado automático, la recuperación ante fallos, el descubrimiento de servicios, el aprovisionamiento de almacenamiento y la integración con pipelines de entrega continua e integración continua (CI/CD). Estas capacidades hacen que Kubernetes sea una herramienta fundamental en estrategias de desarrollo moderno, especialmente en proyectos que requieren alta disponibilidad, flexibilidad y agilidad operativa.

Importancia de la migración tecnológica

La migración tecnológica en el contexto de las organizaciones se refiere al proceso mediante el cual una empresa actualiza, traslada o reemplaza sus sistemas, aplicaciones, datos e infraestructura tecnológica de un entorno obsoleto a uno moderno y más eficiente. Este proceso puede implicar la transición de sistemas locales a ambientes en la nube, la actualización de software anticuado o la reestructuración completa de la infraestructura de TI para adoptar tecnologías más avanzadas. La migración tecnológica es un componente esencial de la gestión y evolución de los sistemas de información, permitiendo a las organizaciones responder a la rápida transformación del entorno digital [21].

Una de las ventajas principales de la migración tecnológica es la modernización y mejora de la eficiencia operativa. Al adoptar tecnologías más actuales, las empresas pueden optimizar el rendimiento de sus sistemas, reducir tiempos de respuesta y disminuir las cargas operativas asociadas con el mantenimiento de infraestructuras antiguas. La migración también permite mejorar la seguridad tecnológica, ya que las plataformas y aplicaciones más recientes incorporan mejores mecanismos de protección ante amenazas cibernéticas y vulnerabilidades que no pueden atender los sistemas heredados [21].

La migración tecnológica favorece la escalabilidad y flexibilidad de los sistemas de información, facilitando que las organizaciones ajusten recursos tecnológicos según las necesidades de crecimiento o variaciones en la carga de trabajo, sin depender de infraestructura física limitada. Asimismo, cuando la migración incluye la adopción de soluciones en la nube, se logra una mejor disponibilidad y continuidad del servicio, permitiendo un acceso más seguro y confiable a los datos y aplicaciones desde diferentes ubicaciones [21].

Finalmente, la migración tecnológica constituye una pieza clave dentro de un proceso más amplio de transformación digital, donde las organizaciones no solo cambian sus plataformas técnicas, sino que también revisan sus procesos, capacidades y estrategias para generar valor a partir del uso de tecnología. En este sentido, la migración tecnológica no solo mejora la infraestructura de TI, sino que impulsa una evolución estructural en la forma en que una organización opera, se conecta con sus clientes y aprovecha los datos para la toma de decisiones.

Computación En La Nube

El uso de la nube ha revolucionado la forma en que las organizaciones y los usuarios finales acceden y gestionan los recursos tecnológicos. Al eliminar la necesidad de invertir en infraestructura física propia, la computación en la nube reduce costos, facilita la colaboración remota, mejora la disponibilidad de servicios y permite escalar capacidades tecnológicas según la demanda. Estos beneficios han hecho que tanto pequeñas como grandes empresas adopten la nube como un componente clave dentro de sus estrategias de transformación digital [22].

El cloud computing se ofrece a través de tres modelos de servicio fundamentales, que representan diferentes niveles de abstracción y control [22].

- **Software como Servicio (SaaS):** el proveedor ofrece aplicaciones listas para usar a través de la nube, accesibles por los usuarios sin necesidad de instalación local ni administración de infraestructura [22].
- **Plataforma como Servicio (PaaS):** el proveedor entrega plataformas completas para el desarrollo, ejecución y gestión de aplicaciones, permitiendo a los desarrolladores concentrarse en la lógica del software sin gestionar los recursos subyacentes [22].
- **Infraestructura como Servicio (IaaS):** en este modelo se proporciona acceso a recursos de infraestructura básicos como servidores, almacenamiento y redes, con mayor control por parte del usuario sobre los sistemas operativos y aplicaciones que se ejecutan [22].

Frontend

El frontend es la parte visual de una aplicación o sistema web con la que interactúan directamente los usuarios. Se encarga del diseño, la estructura, los botones, formularios, menús y todos los elementos gráficos que permiten navegar y utilizar una plataforma digital de manera intuitiva. Para su desarrollo se utilizan tecnologías como HTML, CSS y JavaScript, además de frameworks modernos que mejoran la experiencia del usuario y la velocidad de interacción [29].

Backend

El backend corresponde a la parte interna y lógica de una aplicación, donde se procesan los datos y se ejecutan las funcionalidades principales del sistema. Este componente administra la

comunicación con las bases de datos, la autenticación de usuarios, la seguridad y las reglas de negocio. El backend trabaja en segundo plano para que el frontend pueda mostrar información correctamente y responder a las acciones de los usuarios [29].

Virtualización

La virtualización es una tecnología que permite crear versiones virtuales de recursos físicos como servidores, sistemas operativos, redes y almacenamiento. Gracias a esta técnica, un mismo servidor físico puede ejecutar múltiples máquinas virtuales de manera independiente, optimizando recursos y reduciendo costos de infraestructura tecnológica [34].

FusionCompute

FusionCompute es una plataforma de virtualización desarrollada por Huawei que permite administrar recursos computacionales, almacenamiento y redes dentro de una infraestructura virtualizada. Esta solución facilita la creación de máquinas virtuales, mejora el aprovechamiento del hardware y proporciona herramientas para alta disponibilidad y gestión eficiente de centros de datos [35].

Balanceo de carga

El balanceo de carga es una técnica que distribuye el tráfico o las solicitudes entre varios servidores para evitar sobrecargas y mejorar el rendimiento del sistema. Su objetivo principal es optimizar la disponibilidad, velocidad y estabilidad de las aplicaciones, asegurando que ningún servidor trabaje más que los demás [35].

Migración tecnológica

La migración tecnológica es el proceso de trasladar sistemas, aplicaciones o infraestructuras desde tecnologías antiguas hacia plataformas más modernas y eficientes. Este proceso busca mejorar rendimiento, seguridad, compatibilidad y capacidad de crecimiento dentro de una organización [34].

Infraestructura virtualizada

La infraestructura virtualizada es un entorno tecnológico en el que los recursos físicos, como servidores, almacenamiento y redes, son gestionados de manera lógica a través de plataformas especializadas. Este enfoque permite optimizar el uso de los recursos, reducir los costos operativos y facilitar la administración centralizada de los sistemas. [35].

Portal web

Un portal web es una plataforma digital que centraliza información, servicios y funcionalidades accesibles desde internet mediante un navegador. Los portales web suelen ofrecer acceso a contenidos, herramientas, consultas y sistemas de interacción para usuarios internos o externos de una organización [38].

2.2 BASES LEGALES

El desarrollo e implementación de soluciones tecnológicas en entornos empresariales y académicos debe enmarcarse dentro de la normativa vigente que regula el uso, gestión y prestación de los servicios de Tecnologías de la Información y las Comunicaciones (TIC). En Colombia, el marco legal en materia de TIC establece los lineamientos para garantizar la eficiencia, calidad, seguridad y continuidad de los servicios tecnológicos, aspectos directamente relacionados con el presente proyecto.

Ley 1341 de 2009

La Ley 1341 de 2009 establece el marco legal general para el sector de las Tecnologías de la Información y las Comunicaciones (TIC) en Colombia. Su objetivo es orientar la formulación de políticas públicas que regulen este sector, incluyendo la planificación, gestión, competencia, protección al usuario, acceso, calidad de los servicios y promoción de la inversión en tecnologías que permitan el acceso a la Sociedad de la Información [23].

La Ley reconoce que las TIC son fundamentales para el desarrollo educativo, social, económico y cultural del país y, por ello, establece principios como el acceso universal a las TIC, la libre competencia, la protección de los derechos de los usuarios, la neutralidad tecnológica, el despliegue eficiente de infraestructura, así como la promoción de contenidos, aplicaciones y la masificación del Gobierno en Línea para facilitar trámites y servicios digitales [23].

Además, esta norma define el papel del Ministerio de Tecnologías de la Información y las Comunicaciones, la Agencia Nacional del Espectro y la Comisión de Regulación de Comunicaciones, encargados de implementar, regular y controlar las políticas del sector. Establece también la creación de fondos para financiar proyectos que impulsen la conectividad y el acceso a las TIC en todo el territorio nacional [23].

La Ley 1341 de 2009 es fundamental para el desarrollo de este proyecto porque establece el marco legal que regula el sector de las Tecnologías de la Información y las Comunicaciones (TIC) en Colombia, definiendo principios como el acceso universal a los servicios, la protección de los derechos de los usuarios, la libre competencia y la promoción de infraestructura tecnológica. Esta norma también determina las funciones del Estado y de los organismos encargados de regular, planificar y controlar el desarrollo del sector, lo cual respalda jurídicamente las decisiones técnicas y estratégicas adoptadas en este trabajo, asegurando que la implementación de soluciones y la atención a necesidades tecnológicas se ajusten a la regulación vigente y contribuyan al fortalecimiento del sistema de TIC en el país [23].

Ley 1978 de 2019

La Ley 1978 de 2019 modifica y moderniza el marco normativo del sector de las Tecnologías de la Información y las Comunicaciones (TIC) en Colombia con el propósito de promover la inversión,

ampliar la cobertura, mejorar la calidad de los servicios y reducir la brecha digital. Esta ley reorganiza la institucionalidad del sector al fortalecer el rol del Estado como promotor del acceso a las TIC y crear un regulador único, con el fin de garantizar mayor eficiencia, neutralidad y transparencia en la regulación [24].

Asimismo, la norma establece principios orientados a priorizar el acceso y uso de las TIC como habilitadores del desarrollo social y económico, fomenta el despliegue de infraestructura de telecomunicaciones, incentiva la competencia en el mercado y protege los derechos de los usuarios. La ley también introduce cambios en la gestión del espectro radioeléctrico y en los mecanismos de financiación del sector, buscando asegurar la sostenibilidad de los servicios y la inclusión digital, especialmente en zonas rurales y apartadas del país [24].

La Ley 1978 de 2019 proporciona el marco normativo actualizado que orienta el diseño, la implementación y la gestión de soluciones tecnológicas en el sector TIC, garantizando que las decisiones técnicas propuestas se alineen con los principios de acceso, calidad, eficiencia y sostenibilidad establecidos por el Estado colombiano. Su aplicación permite sustentar jurídicamente el proyecto, evidenciando coherencia con las políticas públicas que buscan fortalecer la infraestructura tecnológica, fomentar la conectividad y contribuir a la reducción de la brecha digital, aspectos directamente relacionados con los objetivos y el alcance del presente trabajo académico.

Ley 2108 de 2021

Denominada “Ley de Internet como servicio público esencial y universal”, modifica la Ley 1341 de 2009 y otras normas del sector de las Tecnologías de la Información y las Comunicaciones (TIC) para establecer el acceso a Internet como un servicio público esencial en Colombia. Esta ley busca garantizar que la prestación del servicio de Internet sea eficiente, continua y permanente, promoviendo su universalidad, especialmente en zonas rurales, apartadas o poblaciones en situación de vulnerabilidad social o étnica, con el fin de cerrar brechas digitales y asegurar la conectividad de todos los habitantes. Además, incorpora principios de universalidad en la regulación del sector, establece obligaciones para los proveedores en casos de emergencia y regula condiciones en planes móviles pre y pospago para asegurar el acceso a contenidos educativos y de interés público. La norma también introduce medidas para incentivar el acceso y despliegue de infraestructura, así como incentivos para pequeños proveedores de Internet, integrando en su articulado disposiciones que apoyan la masificación de servicios en todo el territorio nacional [25].

La Ley 2108 de 2021 establece el acceso a Internet como un servicio público esencial y universal dentro del marco normativo colombiano, lo cual respalda la importancia de garantizar conectividad eficiente y permanente en las soluciones propuestas. Su incorporación permite fundamentar jurídicamente las decisiones técnicas orientadas al cierre de brechas digitales, la inclusión de servicios en zonas rurales y vulnerables, y la provisión de conectividad de calidad dentro del alcance del proyecto. Esto asegura que los objetivos planteados se alineen con las políticas públicas vigentes que promueven la cobertura, calidad y equidad en el acceso a servicios TIC en Colombia [25].

Ley 1581 de 2012

La Ley 1581 de 2012 es una norma colombiana que regula la protección de los datos personales, garantizando el derecho que tienen todas las personas a conocer, actualizar y rectificar la información que se tenga sobre ellas en bases de datos. Esta ley desarrolla el derecho fundamental a la privacidad y busca que el manejo de la información personal respete la dignidad y los derechos de los ciudadanos. [26]

Además, la ley aplica a todas las entidades públicas y privadas que recolecten, almacenen o utilicen datos personales. Establece principios fundamentales como la legalidad, la finalidad (uso específico de los datos), la libertad (requiere autorización del titular), la veracidad, la transparencia, la seguridad y la confidencialidad, los cuales deben cumplirse en todo tratamiento de información personal. [26]

También define los derechos de los titulares de los datos, como acceder a su información, corregirla, solicitar pruebas de autorización, conocer el uso que se le da y pedir su eliminación cuando sea necesario. Igualmente, impone obligaciones a las organizaciones, como solicitar permiso previo para usar los datos, garantizar su protección y permitir que los usuarios realicen consultas o reclamos. [26]

Es importante tener en cuenta esta ley en el proyecto porque permite asegurar un manejo adecuado y legal de la información personal, evitando sanciones y problemas jurídicos. Además, contribuye a proteger la privacidad de las personas, genera confianza en quienes interactúan con el proyecto y promueve un uso responsable y ético de los datos, lo cual es fundamental para la credibilidad y el buen desarrollo de cualquier iniciativa.

Decreto 1377 de 2013

El Decreto 1377 de 2013 reglamenta parcialmente la Ley 1581 de 2012, que es el marco general de protección de datos personales en Colombia. Este decreto establece definiciones y reglas específicas sobre cómo se deben tratar los datos personales cuando son recolectados, almacenados, utilizados, administrados y eliminados. Regula aspectos como la autorización que debe otorgar el titular para el tratamiento de sus datos, las políticas de tratamiento que los responsables deben diseñar y comunicar, los mecanismos para ejercer derechos como conocer, rectificar o suprimir información, y las obligaciones de transparencia y de conservación de evidencias sobre dicho tratamiento. El decreto también establece requisitos especiales para el manejo, comunicación y revocación de la autorización de tratamiento de datos personales, así como estándares para la protección de datos sensibles y los mecanismos que aseguran el cumplimiento de estas obligaciones por parte de quienes gestionan información personal dentro de organizaciones públicas o privada [27].

El Decreto 1377 de 2013 es relevante para este proyecto porque complementa la Ley 1581 de 2012 al establecer cómo deben llevarse a cabo los procesos de tratamiento de datos personales, asunto esencial al diseñar e implementar soluciones tecnológicas que gestionen información de usuarios o de sistemas. Incluir este decreto dentro del marco legal del proyecto permite demostrar que el manejo de datos se ajustará a principios como autorización, transparencia, seguridad y respeto por los derechos de los titulares, lo cual fortalece la validez jurídica y ética

del tratamiento de la información dentro del desarrollo e implementación de la propuesta investigativa [27].

Ley 1273 de 2009

La Ley 1273 de 2009 modifica el Código Penal colombiano para establecer normas que protejan la información, los datos y los sistemas informáticos, electrónicos y telemáticos frente a accesos o usos no autorizados y conductas que atenten contra su confidencialidad, integridad y disponibilidad. Esta ley crea un título nuevo dentro del Código Penal denominado “De la protección de la información y de los datos”, donde tipifica delitos como el acceso abusivo a sistemas informáticos, la interceptación de datos, el daño o la alteración de información, la suplantación de sitios web y la utilización del software malicioso para causar daño o infringir la seguridad de sistemas de información. Además, incorpora circunstancias de agravación para estas conductas y adopta medidas punitivas concretas para quienes cometan estos delitos, reforzando así la seguridad digital en Colombia [28].

La Ley es pertinente porque establece el marco penal que protege la información, los sistemas de comunicaciones y los datos frente a amenazas o accesos indebidos, lo cual es esencial cuando se diseñan, implementan o gestionan soluciones tecnológicas relacionadas con comunicaciones o gestión de datos. Incluir esta ley dentro del marco normativo demuestra que el proyecto no solo se ajusta a los principios técnicos actuales, sino que también toma en cuenta la seguridad informática, la prevención de delitos digitales y la protección de la información en el contexto del uso de tecnologías y redes. Esto aporta un respaldo jurídico sólido a las medidas de seguridad y control que se plantean en el desarrollo de la propuesta investigativa.

Normas ISO/IEC 27001

La ISO/IEC 27001 es una norma internacional que establece los requisitos para implementar, mantener y mejorar un Sistema de Gestión de Seguridad de la Información (SGSI) dentro de una organización. Su objetivo principal es proteger la información mediante la gestión de riesgos asociados a la confidencialidad, integridad y disponibilidad de los datos. Esta norma proporciona un enfoque basado en procesos y en la gestión del riesgo, permitiendo identificar amenazas, evaluar vulnerabilidades y aplicar controles de seguridad adecuados para mitigar posibles incidentes [29].

Asimismo, la ISO/IEC 27001 promueve la adopción de buenas prácticas en seguridad informática, incluyendo políticas de seguridad, control de accesos, gestión de incidentes, continuidad del negocio y cumplimiento normativo. Su implementación es aplicable a cualquier tipo de organización, independientemente de su tamaño o sector, que maneje información sensible o crítica [29].

La norma ISO/IEC 27001 es relevante para este proyecto de grado debido a que proporciona un marco internacionalmente reconocido para la gestión de la seguridad de la información, lo cual es fundamental en el diseño e implementación de soluciones tecnológicas que involucren el manejo de datos. Su aplicación permite asegurar que los procesos desarrollados en el proyecto

incorporen controles adecuados para proteger la información contra accesos no autorizados, pérdidas o alteraciones, garantizando así la confidencialidad, integridad y disponibilidad de los datos. De esta manera, el proyecto se alinea con estándares internacionales de seguridad, fortaleciendo su solidez técnica y su enfoque en la gestión del riesgo.

2.3 ESTADO DEL ARTE

El desarrollo de aplicaciones web y la gestión de infraestructuras tecnológicas han experimentado una transformación significativa en las últimas décadas, impulsada por la necesidad de construir sistemas más escalables, resilientes y fáciles de administrar. En este contexto, tecnologías como la virtualización, la contenedorización y la orquestación de servicios se han consolidado como pilares fundamentales tanto en entornos empresariales como académicos. El estado del arte en estas áreas permite comprender la evolución de dichas tecnologías y su impacto en la modernización de plataformas digitales, proporcionando un sustento técnico sólido para la toma de decisiones en procesos de migración tecnológica.

En sus primeras etapas, las aplicaciones web eran desplegadas en arquitecturas monolíticas directamente sobre servidores físicos, lo cual generaba limitaciones importantes en términos de escalabilidad, mantenimiento y disponibilidad. Con el aumento en la demanda de servicios digitales, estas arquitecturas comenzaron a mostrar problemas relacionados con tiempos de inactividad, dificultad para realizar actualizaciones y un uso ineficiente de los recursos. Como respuesta a estas limitaciones, surgió la virtualización, tecnología que permite abstraer los recursos físicos y ejecutar múltiples sistemas operativos en un mismo hardware, mejorando considerablemente la utilización de los recursos disponibles.

La virtualización ha evolucionado hasta convertirse en un componente esencial de la infraestructura moderna. Plataformas empresariales como VMware vSphere y Microsoft HyperV ofrecen capacidades avanzadas como alta disponibilidad, migración en caliente y balanceo de carga, lo que permite garantizar la continuidad del servicio y optimizar el uso de infraestructura [12]. Además, esta tecnología proporciona beneficios clave como flexibilidad, escalabilidad bajo demanda, reducción de costos operativos y aislamiento seguro entre entornos, lo que la convierte en una base fundamental para los centros de datos actuales [2]. En este sentido, la virtualización es hoy en día prácticamente obligatoria en entornos empresariales modernos.

Sin embargo, a pesar de sus múltiples ventajas, las máquinas virtuales presentan una sobrecarga considerable debido a que cada instancia incluye un sistema operativo completo. Esto motivó el desarrollo de soluciones más eficientes, dando lugar a la contenedorización. Los contenedores permiten encapsular aplicaciones junto con sus dependencias en unidades ligeras que comparten el sistema operativo del host, reduciendo el consumo de recursos y facilitando su ejecución en distintos entornos.

La contenedorización ha sido ampliamente adoptada gracias a herramientas como Docker, que permiten empaquetar aplicaciones de manera estandarizada, garantizando consistencia entre ambientes de desarrollo, pruebas y producción [3]. Entre sus principales ventajas se destacan la portabilidad, la eficiencia en el uso de recursos y la capacidad de despliegue rápido.

Asimismo, los contenedores facilitan la transición hacia arquitecturas basadas en microservicios, donde cada componente del sistema funciona de manera independiente, lo que mejora la escalabilidad y la resiliencia de las aplicaciones [4]. Este enfoque ha sido adoptado

por más del 80 % de grandes organizaciones, evidenciando su relevancia en el desarrollo moderno de software [4].

No obstante, el aumento en la cantidad de contenedores genera nuevos desafíos relacionados con su gestión. En este contexto, surgen herramientas de orquestación, siendo Kubernetes la más destacada y adoptada a nivel global. Kubernetes permite automatizar tareas críticas como el despliegue, la gestión, el escalado y la recuperación de aplicaciones, convirtiéndose en un estándar en la industria para la implementación de sistemas distribuidos [5]. Gracias a capacidades como el auto escalado, el balanceo de carga y la tolerancia a fallos, Kubernetes mejora significativamente la disponibilidad y el rendimiento de los servicios [6].

Una de las características más relevantes de Kubernetes es su capacidad de autorrecuperación. El sistema supervisa constantemente el estado de los contenedores y, en caso de fallos, los reemplaza automáticamente sin intervención humana. Además, distribuye la carga de trabajo entre los nodos disponibles y ajusta dinámicamente los recursos según la demanda, lo que garantiza una alta disponibilidad del servicio. Estas funcionalidades permiten construir sistemas resilientes capaces de adaptarse a condiciones cambiantes y minimizar el impacto de fallos.

La integración entre virtualización y contenedorización representa una práctica ampliamente validada en entornos empresariales. Diversos estudios han demostrado que ejecutar contenedores sobre máquinas virtuales permite combinar la eficiencia de los contenedores con la seguridad y el aislamiento de la virtualización, logrando una infraestructura más robusta y flexible [7]. Este enfoque híbrido facilita la consolidación de servicios, la optimización de recursos y la reducción de costos operativos, además de ofrecer un mayor control sobre la infraestructura en comparación con soluciones completamente basadas en la nube.

En paralelo, las prácticas de desarrollo han evolucionado hacia metodologías DevOps, las cuales buscan integrar de manera continua el desarrollo y la operación de software. En este contexto, la contenedorización y Kubernetes desempeñan un papel fundamental al facilitar la implementación de pipelines de integración y despliegue continuo (CI/CD). Estas prácticas automatizan procesos como la construcción, prueba y despliegue de aplicaciones, reduciendo errores humanos y acelerando los ciclos de entrega [8]. Kubernetes, en particular, potencia estos procesos al ofrecer despliegues automatizados, escalabilidad dinámica y mecanismos de recuperación ante fallos, lo que mejora la confiabilidad del software [9].

En cuanto a las tendencias actuales, se observa una fuerte adopción de arquitecturas basadas en microservicios y entornos híbridos que combinan infraestructura local con soluciones empresariales escalables. Kubernetes se ha consolidado como el estándar de facto para la orquestación de contenedores, siendo utilizado por una gran mayoría de organizaciones en entornos de producción, lo que demuestra su madurez y confiabilidad [10]. Asimismo, la combinación de virtualización, contenedorización y orquestación permite a las organizaciones construir sistemas altamente eficientes, flexibles y preparados para enfrentar los retos tecnológicos actuales.

En el contexto específico del presente proyecto, el análisis del estado del arte evidencia que la migración de aplicaciones web hacia infraestructuras basadas en contenedores y gestionadas mediante Kubernetes, soportadas por plataformas de virtualización empresarial, constituye una práctica ampliamente validada. Esta estrategia no solo mejora la disponibilidad, la escalabilidad y el rendimiento de los servicios, sino que también optimiza la gestión de recursos y reduce los tiempos de recuperación ante fallos.

En conclusión, la evolución de la virtualización, la contenedorización y la orquestación ha transformado profundamente el desarrollo y la operación de sistemas digitales. Estas tecnologías representan actualmente una de las mejores prácticas en la industria de tecnologías de la información, especialmente en entornos donde la continuidad del servicio, la eficiencia operativa y la escalabilidad son factores críticos. Su adopción permite construir infraestructuras modernas, robustas y alineadas con las demandas actuales del entorno digital.

CAPITULO III. MARCO METODOLÓGICO

3.1 ENFOQUE Y TIPO DE INVESTIGACIÓN

Enfoque de la investigación

El presente proyecto se desarrolla bajo un enfoque de investigación aplicada, dado que su finalidad principal es proponer e implementar una solución práctica a una problemática real identificada en el entorno de un laboratorio tecnológico. En este caso, la investigación se orienta a mejorar la infraestructura tecnológica que soporta el portal web utilizado para la gestión de entrenamientos, pruebas técnicas y actividades académicas, buscando optimizar su funcionamiento, disponibilidad y eficiencia operativa. Este enfoque permite que el conocimiento generado esté directamente vinculado con la resolución de necesidades concretas del contexto analizado.

Además, la investigación adopta un enfoque cualitativo–técnico, ya que el análisis se centra en la comprensión del comportamiento del sistema, la evaluación de la arquitectura tecnológica existente y la identificación de fallas o limitaciones en su diseño e implementación. En lugar de basarse en el análisis estadístico de grandes volúmenes de datos, el estudio prioriza la observación técnica, el análisis funcional de los componentes del sistema y la interpretación de los resultados obtenidos a partir de pruebas, configuraciones y validaciones realizadas sobre la infraestructura tecnológica.

El enfoque metodológico incorpora un componente estudiantil, propio del desarrollo académico del proyecto, el cual se manifiesta en la aplicación práctica de los conocimientos adquiridos durante la formación profesional en ingeniería. Desde esta perspectiva, el proyecto no solo busca generar una mejora técnica en la infraestructura del portal web, sino también fortalecer las competencias del estudiante en áreas como análisis de arquitecturas tecnológicas, toma de decisiones técnicas, resolución de problemas reales y aplicación de buenas prácticas en entornos de TI. Este enfoque permite articular la teoría aprendida en el aula con su aplicación en un escenario real, promoviendo un aprendizaje significativo y contextualizado.

Por último, la investigación se concibe como un proceso formativo–aplicado, en el que el estudiante asume un rol activo en el análisis, diseño y evaluación de la solución propuesta. La metodología empleada favorece la comprensión integral del sistema, el desarrollo de habilidades técnicas y el fortalecimiento del criterio profesional, al tiempo que contribuye a la mejora del entorno tecnológico objeto de estudio. De esta manera, el enfoque de la investigación integra de forma coherente los objetivos académicos y técnicos del proyecto, garantizando tanto su pertinencia metodológica como su utilidad práctica.

Tipo de investigación

El presente proyecto se enmarca dentro de una investigación aplicada de tipo descriptivo y propositivo, orientada a analizar una situación real dentro de un entorno tecnológico específico y a plantear una solución que contribuya a su mejora. Este tipo de investigación resulta pertinente cuando el objetivo no es únicamente generar conocimiento teórico, sino aplicar dicho conocimiento para resolver problemas concretos en contextos reales, como es el caso del

laboratorio tecnológico objeto de estudio.

Desde el punto de vista descriptivo, la investigación busca caracterizar el estado actual de la infraestructura tecnológica que soporta el portal web del laboratorio. Para ello, se realiza un análisis detallado de la arquitectura existente, los componentes que la conforman, su funcionamiento, así como las principales limitaciones y fallas identificadas en términos de rendimiento, disponibilidad y escalabilidad. Este proceso descriptivo permite comprender el comportamiento del sistema y establecer una base técnica sólida sobre la cual se sustenta la propuesta de mejora.

De manera complementaria, la investigación adopta un carácter propositivo, ya que no se limita a describir la problemática identificada, sino que plantea una solución tecnológica orientada a optimizar la infraestructura existente. A partir del análisis realizado, se diseña una propuesta de mejora que busca fortalecer la arquitectura del portal web, incorporando buenas prácticas en el uso de tecnologías de la información y alineándose con los principios de eficiencia, confiabilidad y sostenibilidad tecnológica. Este enfoque propositivo permite transformar los hallazgos del análisis en acciones concretas que aportan valor al entorno estudiado.

Asimismo, el proyecto puede clasificarse como una investigación de campo, debido a que el estudio se desarrolla directamente en el entorno real del laboratorio tecnológico, considerando sus condiciones operativas, restricciones técnicas y necesidades específicas. La información recopilada proviene de la observación directa del sistema, la revisión de su configuración, la ejecución de pruebas técnicas y el análisis de su comportamiento en condiciones reales de uso, lo que garantiza la pertinencia y validez de los resultados obtenidos.

Desde una perspectiva académica, el tipo de investigación también posee un componente formativo, propio de los proyectos desarrollados en el ámbito estudiantil. En este sentido, la investigación permite al estudiante aplicar conocimientos teóricos adquiridos durante su formación profesional, desarrollar habilidades analíticas y técnicas, y fortalecer su capacidad para diseñar soluciones tecnológicas en contextos reales. Este componente formativo no resta rigurosidad al estudio, sino que refuerza su carácter aplicado y su contribución tanto al aprendizaje como a la mejora del entorno tecnológico analizado.

En conclusión, el proyecto se define como una investigación aplicada, descriptiva, propositiva y de campo, orientada a la solución de una problemática real mediante el análisis técnico de una infraestructura tecnológica y la formulación de una propuesta de mejora, integrando objetivos académicos y prácticos de manera coherente.

Método de investigación

El método de investigación empleado en el presente proyecto es el método analítico-sistémico, el cual resulta adecuado para el estudio y mejora de infraestructuras tecnológicas complejas. Este método permite descomponer el sistema objeto de estudio en sus diferentes componentes para analizarlos de manera individual, y posteriormente integrarlos nuevamente con el fin de comprender su funcionamiento global y proponer una solución coherente y estructurada.

En una primera etapa, se aplica el análisis técnico del sistema, mediante la identificación y evaluación de los elementos que conforman la infraestructura que soporta el portal web del laboratorio tecnológico. Este análisis incluye la revisión de la arquitectura existente, los servicios desplegados, los flujos de comunicación, los recursos tecnológicos utilizados y las dependencias entre los distintos componentes. A través de este proceso, se identifican fallas, limitaciones y oportunidades de mejora relacionadas con el desempeño, la disponibilidad y la escalabilidad del sistema.

Posteriormente, se emplea un enfoque sistémico, que permite comprender el sistema como un conjunto interrelacionado de elementos que interactúan entre sí y con su entorno. Este enfoque resulta fundamental para evaluar cómo los cambios propuestos en la infraestructura pueden impactar el funcionamiento general del portal web, evitando soluciones aisladas que no consideren el comportamiento integral del sistema. De esta manera, la propuesta de mejora se diseña teniendo en cuenta la interacción entre los componentes tecnológicos, los usuarios y el entorno operativo del laboratorio.

El método de investigación también incorpora un proceso iterativo, en el cual se alternan fases de análisis, diseño, implementación y evaluación. Durante este proceso, las decisiones técnicas se ajustan de acuerdo con los resultados obtenidos en cada etapa, permitiendo refinar la solución propuesta y asegurar su alineación con los objetivos del proyecto. Este enfoque iterativo facilita la validación progresiva de la propuesta y reduce el riesgo de fallos en la implementación final.

Desde una perspectiva académica, el método utilizado integra un componente formativo, propio del desarrollo estudiantil del proyecto. El estudiante participa activamente en cada una de las fases metodológicas, aplicando conocimientos teóricos, desarrollando habilidades prácticas y fortaleciendo su capacidad de análisis crítico. Este proceso contribuye tanto a la solución de la problemática identificada como al desarrollo de competencias profesionales en el área de tecnologías de la información.

En conclusión, el método analítico–sistémico, apoyado en un enfoque iterativo y técnico, permite abordar de manera estructurada la problemática del proyecto, garantizando una comprensión integral del sistema y el diseño de una solución viable, eficiente y alineada con las necesidades reales del entorno tecnológico analizado.

3.2 DISEÑO METODOLOGICO

3.2.1 Fase 1: Análisis del entorno actual

En esta fase se llevó a cabo un análisis del entorno actual del portal web del laboratorio tecnológico, con el objetivo de comprender su funcionamiento, arquitectura y condiciones operativas. El portal se encuentra implementado bajo una arquitectura de aplicación full stack, compuesta por los componentes de frontend, backend y base de datos, los cuales están desplegados en un entorno contenedorizado.

Durante el análisis se identificó el esquema de despliegue utilizado, así como la forma en que interactúan los distintos componentes de la aplicación dentro del entorno existente. Adicionalmente, se evaluaron aspectos clave relacionados con la estabilidad, disponibilidad y gestión operativa del servicio, evidenciando fallas recurrentes asociadas a la administración de

los contenedores, dificultades en la escalabilidad y limitaciones en los mecanismos de monitoreo y control del entorno.

Asimismo, se detectaron restricciones en la gestión del ciclo de vida de la aplicación, lo que impacta negativamente en los procesos de actualización, mantenimiento y recuperación ante fallos. Estas limitaciones generan riesgos operativos que afectan la continuidad del servicio y la experiencia de los usuarios que acceden al portal del laboratorio para actividades de capacitación y pruebas técnicas.

Como resultado de este análisis, se establecieron los principales puntos críticos que comprometen el correcto funcionamiento del portal web, lo cual permitió justificar de manera técnica la necesidad de migrar hacia una arquitectura más robusta, estable y eficiente, capaz de garantizar una mayor disponibilidad del servicio y una mejor gestión del entorno tecnológico.

Con el fin de facilitar la comprensión del entorno actual, se presenta un esquema general de la arquitectura existente del portal web, el cual permite visualizar la interacción entre sus principales componentes, tal como se muestra en [Figure 1].

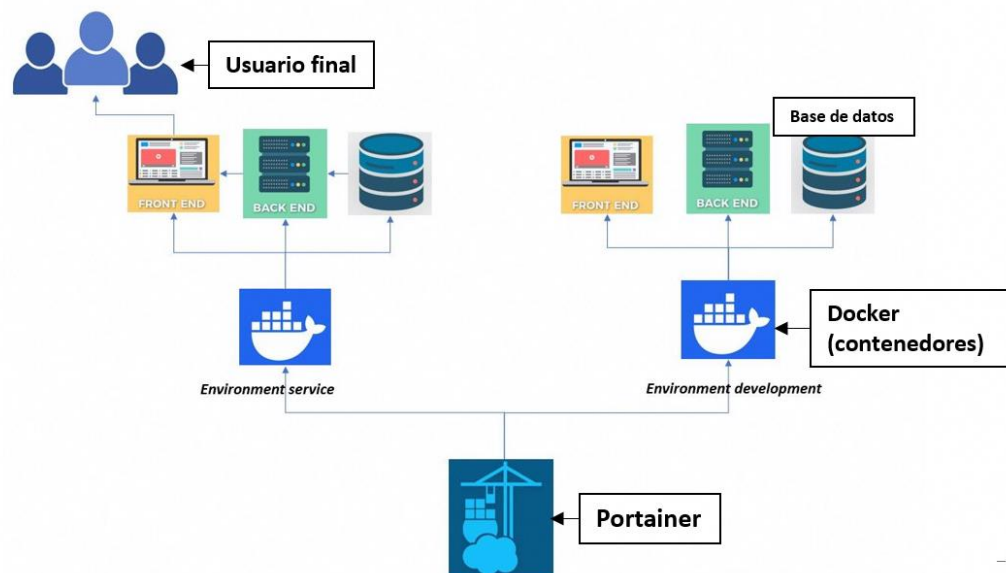


Ilustración 1 Arquitectura de página web en portainer

Como se muestra en la Figura 1, el despliegue del portal web se encontraba implementado en un entorno basado en contenedores Docker, administrado mediante la herramienta Portainer. La arquitectura estaba compuesta por dos entornos principales: un entorno de desarrollo y un entorno de servicio.

El entorno de desarrollo estaba destinado a la realización de pruebas y modificaciones, e incluía contenedores independientes para el frontend, el backend y la base de datos, permitiendo validar cambios antes de su puesta en producción. Por su parte, el entorno de servicio correspondía al entorno operativo del portal web, desde el cual el usuario final accedía a la aplicación a través de la interfaz web, consumiendo los servicios provistos por el backend y la información almacenada en la base de datos.

Ambos entornos se ejecutaban sobre la misma infraestructura Docker y se comunicaban a través de la red interna del entorno contenedorizado. La administración y supervisión de los contenedores se realizaba mediante Portainer, el cual proporcionaba una gestión básica de los servicios desplegados. No obstante, esta arquitectura carecía de mecanismos avanzados de orquestación, balanceo de carga y tolerancia a fallos, lo que generaba dependencias críticas entre los componentes y afectaba la disponibilidad y escalabilidad del servicio.

3.2.2 Fase 2: Diseño de la arquitectura objetivo

A partir de los resultados obtenidos en la fase de análisis del entorno actual, se planteó el diseño de una arquitectura objetivo orientada a la modernización del portal web del laboratorio tecnológico, con énfasis en la escalabilidad, la disponibilidad y la facilidad de administración. Para ello, se adoptó un enfoque basado en contenedores y orquestación con Kubernetes, desplegados sobre una plataforma de virtualización empresarial.

La arquitectura propuesta se concibe para ejecutarse sobre FusionCompute[37], plataforma desarrollada por Huawei, la cual permite la gestión centralizada de máquinas virtuales, redes y almacenamiento en entornos corporativos. Esta elección responde a la necesidad de contar con un mayor control sobre los recursos de cómputo, garantizar la estandarización del entorno de ejecución y alinearse con las políticas de infraestructura empresarial del laboratorio.

Desde el punto de vista lógico, la arquitectura se estructura en capas claramente definidas. En la capa de infraestructura se encuentra FusionCompute, encargado de proveer las máquinas virtuales sobre las cuales se despliega el clúster de Kubernetes. Esta capa abstrae los recursos físicos y facilita su asignación dinámica según la demanda de la aplicación. Sobre dicha infraestructura se ubica la capa de orquestación, donde Kubernetes administra los contenedores, gestiona el ciclo de vida de los servicios, supervisa su estado y permite la escalabilidad automática [figure 2].

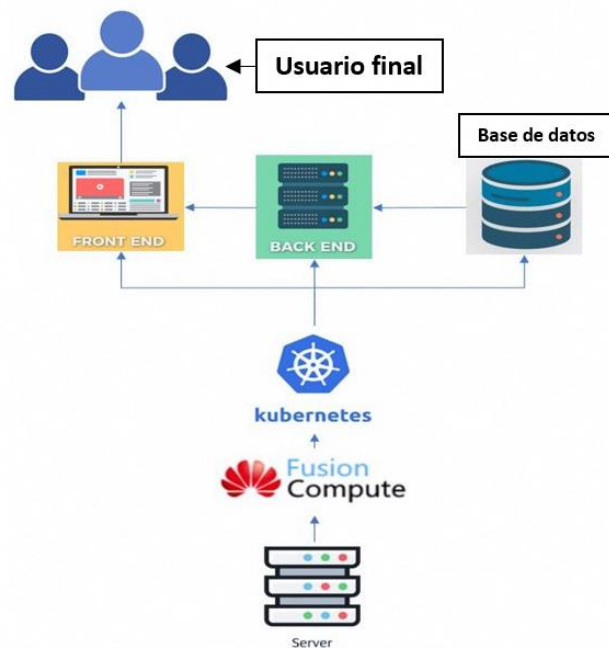


Ilustración 2. Arquitectura final desplegada en fusioncompute.

Como se ve en la imagen, la capa de aplicación, la arquitectura se implementa bajo un enfoque de microservicios, donde cada componente se ejecuta de forma independiente dentro de su propio contenedor. El frontend se despliega como un servicio separado y es responsable de la interfaz gráfica y la interacción directa con el usuario final. Este componente consume las APIs expuestas por el backend para mostrar y enviar información. El backend se implementa como otro servicio independiente y concentra la lógica de negocio del sistema. Se encarga de procesar las solicitudes provenientes del frontend, aplicar reglas de negocio y gestionar la comunicación con la base de datos. La comunicación entre ambos componentes se realiza a través de servicios internos de Kubernetes (ClusterIP), lo que permite un intercambio de información estable, controlado y no expuesto directamente al exterior. Para el acceso externo del sistema, se utilizan mecanismos de exposición como NodePort o Ingress, los cuales permiten publicar el frontend hacia los usuarios finales, manteniendo el backend protegido dentro del clúster.

Esta separación de responsabilidades permite escalar, actualizar y mantener cada componente de manera independiente, reduciendo el impacto de fallos y mejorando la disponibilidad general del sistema.

3.2.3 Fase 3: Documentación de la solución

3.2.3.1 Preparación del entorno de trabajo

En la fase inicial de implementación se llevó a cabo la preparación del entorno de trabajo, con el propósito de garantizar que la infraestructura disponible soportara de manera adecuada la ejecución de contenedores y la orquestación mediante Kubernetes. Esta etapa resultó fundamental para asegurar la estabilidad del proceso de implementación y prevenir errores derivados de incompatibilidades técnicas.

Se realizó una validación de los recursos de hardware disponibles, incluyendo memoria RAM, capacidad de procesamiento y almacenamiento, con el fin de determinar su viabilidad para la ejecución de múltiples contenedores simultáneamente. Además, se verificó la compatibilidad del sistema operativo con las herramientas requeridas, asegurando el soporte para virtualización y tecnologías de contenedorización.

Posteriormente, se configuraron las herramientas base necesarias, incluyendo el motor de contenedores (Docker) y el entorno de Kubernetes en modalidad local, lo cual permitió simular un entorno de despliegue controlado. Esta simulación fue clave para realizar pruebas sin afectar entornos productivos y facilitó la detección temprana de errores.

Finalmente, se validó el acceso al clúster de Kubernetes y su correcto funcionamiento, comprobando que los nodos se encontraban en estado operativo y que las herramientas de administración respondían adecuadamente. Esta verificación permitió confirmar que el entorno estaba listo para avanzar hacia las siguientes etapas del proceso de implementación.

3.2.3.2 Proceso de contenerización del backend

Una vez preparado el entorno, se procedió a la contenerización del backend, con el objetivo de encapsular la aplicación en una imagen que permitiera su ejecución de manera independiente del entorno subyacente. Este enfoque facilita la portabilidad y garantiza la consistencia en diferentes infraestructuras.

Para ello, se realizó un análisis previo de la estructura del proyecto backend, identificando elementos clave como dependencias, archivos de configuración, puntos de entrada y puertos expuestos [Tabla 1]. Este análisis permitió determinar los requerimientos necesarios para construir una imagen funcional y optimizada.

Componente	Descripción	Función
.git	Repositorio de control de versiones	Gestiona cambios del código
Controllers	Módulos de control	Manejan peticiones HTTP
Db	Base de datos	Conexión y consultas
Middleware	Funciones intermedias	Autenticación y validaciones
Smtplib	Configuración de correo	Envía correos
.env	Variables de entorno	Configuración del sistema
Dockerfile	Archivo de construcción	Define imagen
docker-compose.yml	Orquestación local	Ejecuta servicios
package.json	Dependencias	Configura Node.js
server.js	Punto de entrada	Inicia servidor

Tabla 1: Componentes principales del backend

A partir de dicha información, se diseñó y ajustó el archivo Dockerfile, en el cual se definió la imagen base, el entorno de ejecución, la instalación de dependencias y la copia del código fuente. Además, se configuró el comando de inicio del servicio, asegurando que el contenedor ejecutara correctamente la aplicación al ser desplegado.

Durante este proceso, se realizaron pruebas de construcción de la imagen con el fin de validar la correcta ejecución de las instrucciones definidas en el Dockerfile. Esto permitió identificar y corregir errores relacionados con dependencias, versiones de software y configuración interna. El resultado de esta etapa fue una imagen de contenedor completamente funcional del backend, capaz de ejecutarse de manera consistente y lista para ser desplegada en un entorno orquestado.

3.2.3.3 Contenerización del frontend

De manera paralela, se llevó a cabo la contenerización del frontend, considerando sus particularidades en cuanto a la entrega de contenido estático y su interacción con el backend.

Se realizó un análisis de la estructura del proyecto frontend, identificando componentes principales, configuración de entorno, dependencias y mecanismos de consumo de la API [

Tabla 2]. Este análisis permitió comprender cómo el frontend interactúa con el backend y qué configuraciones debían ser ajustadas para su despliegue en contenedores.

Componente	Descripción	Función
.git	Control de versiones	Gestiona cambios
public	Archivos estáticos	Imágenes y recursos
src	Código fuente	Componentes y lógica
index.html	Punto de entrada	Carga app
package.json	Dependencias	Configura frontend
vite.config.ts	Config Vite	Build app
tsconfig.json	Tipado	Reglas TS
.env	Variables	Conexión backend
Dockerfile	Construcción	Empaquetado
nginx.conf	Servidor	Entrega contenido

Tabla 2: Componentes principales del frontend

Para la construcción del contenedor, se definió un Dockerfile que incluye tanto la fase de compilación de la aplicación (build) como su posterior despliegue utilizando un servidor web, en este caso Nginx. La configuración de Nginx fue especialmente relevante, ya que permitió gestionar correctamente las rutas de la aplicación y garantizar el comportamiento adecuado en aplicaciones tipo SPA (Single Page Application).

Asimismo, se configuraron las variables de entorno necesarias para establecer la comunicación con el backend, asegurando que la aplicación frontend pudiera consumir los servicios expuestos por este. El resultado de este proceso fue una imagen de contenedor del frontend capaz de entregar contenido de forma eficiente y de integrarse correctamente con los servicios backend.

3.2.3.4 Construcción y validación de imágenes

Una vez definidos los archivos de construcción (Dockerfiles), se procedió a la generación de las imágenes correspondientes para el backend y el frontend.

Este proceso permitió validar que ambas aplicaciones podían ser empaquetadas correctamente y que cumplieran con los requisitos definidos previamente. Durante la construcción, se presentaron algunos inconvenientes relacionados con la descarga de imágenes base y la conectividad con repositorios externos, los cuales fueron solucionados mediante ajustes en la configuración del entorno y selección de imágenes compatibles.

Posteriormente, se realizó la ejecución local de las imágenes generadas, con el fin de comprobar que los servicios iniciaban correctamente, que los puertos estaban expuestos de manera adecuada y que no existían errores críticos en tiempo de ejecución. Esta validación fue esencial para asegurar la calidad de las imágenes antes de su despliegue en Kubernetes.

3.2.3.5 Despliegue en el clúster de Kubernetes

Con las imágenes validadas, se procedió al despliegue de los componentes en el clúster de Kubernetes, utilizando recursos de tipo Deployment. En esta etapa se definieron los archivos de configuración en formato YAML, especificando aspectos clave como los que se ven en la siguiente tabla [Tabla 3].

Elemento	Backend	Frontend
Imagen base	Node.js	Node.js + Nginx
Tipo	API REST	Aplicación web
Puerto	Configurable	80
Proceso	server.js	Nginx
Dependencias	Node.js	Node + Vite

Tabla 3: Elementos necesarios para la configuración de kubernetes

Este enfoque permitió automatizar el despliegue de los servicios y garantizar su escalabilidad y disponibilidad dentro del clúster. Además, se verificó el estado de los pods generados, confirmando que se encontraban en ejecución y sin errores. El uso de Kubernetes facilitó la gestión de los contenedores, proporcionando mecanismos para el control del ciclo de vida de las aplicaciones y la recuperación ante fallos.

3.2.3.6 Configuración de servicios y comunicación interna

Una vez desplegados los componentes, se procedió a la configuración de los recursos Service dentro de Kubernetes, con el fin de habilitar la comunicación entre frontend y backend. Se creó un servicio interno para el backend, el cual permitió exponer el sistema mediante un nombre DNS dentro del clúster. Este mecanismo elimina la dependencia de direcciones IP estáticas y facilita la comunicación entre servicios de manera dinámica.

Por otro lado, se configuró un servicio para el frontend que permitió su acceso desde el exterior, posibilitando la interacción de los usuarios con la aplicación. Adicionalmente, se estableció la comunicación entre frontend y backend mediante el uso del nombre del servicio interno, lo cual aseguró una integración más robusta y alineada con las buenas prácticas de Kubernetes, en la siguiente tabla se explica mejor cuales fueron los recursos definidos [Tabla 4].

Recurso	Descripción	Función
Deployment	Despliegue	Gestión contenedores
Pod	Unidad mínima	Ejecuta contenedores
Service (ClusterIP)	Servicio interno	Comunicación interna
Service (NodePort)	Servicio externo	Acceso externo
ReplicaSet	Réplicas	Alta disponibilidad

Tabla 4: Recursos usados para la comunicación entre backend y frontend

3.2.3.7 Pruebas y validación del sistema

Con el sistema desplegado, se ejecutaron diferentes pruebas con el objetivo de validar su correcto funcionamiento, se realizaron pruebas de conectividad internas dentro del clúster, verificando que el frontend pudiera comunicarse con el backend sin inconvenientes. Asimismo, se evaluó la disponibilidad de los servicios y la respuesta de los endpoints definidos en el backend. También se llevaron a cabo pruebas funcionales básicas, enfocadas en comprobar que la aplicación mantenía su comportamiento original tras la migración a contenedores. Esto incluyó la validación de funcionalidades clave y la navegación dentro del sistema. Los resultados obtenidos confirmaron que la solución implementada es funcional y que cumple con los requerimientos establecidos.

3.2.3.8 Problemas encontrados durante la implementación

Durante el proceso de implementación, se identificaron varios inconvenientes que afectaron la ejecución del proyecto, principalmente relacionados con la compatibilidad entre arquitecturas (x86 y ARM) y la integración con la plataforma FusionCompute a continuación los errores y especificaciones presentados [tabla 5].

Problema	Causa	Solución
Error arquitectura	x86 vs ARM	Uso entorno local
Fallo FusionCompute	Incompatibilidad	Ajustes config
Descarga imágenes	Red	Config repositorios
Despliegue	Config YAML	Corrección archivos

Tabla 5: Problemas durante el despliegue

Se evidenció que las imágenes generadas no eran compatibles directamente con el entorno de destino, lo que generó errores durante su carga y ejecución. A pesar de los intentos por ajustar la arquitectura de las imágenes, las limitaciones del entorno de desarrollo impidieron resolver completamente este problema.

Como solución alternativa, se optó por realizar el despliegue en un entorno local, dejando preparados los archivos de configuración necesarios (Deployments y Services) para su posterior implementación en FusionCompute una vez se resuelvan las limitaciones actuales.

3.2.3.9 Documentación y registro de evidencias

Finalmente, se realizó la documentación del proceso de implementación, incluyendo las actividades desarrolladas, configuraciones realizadas, problemas encontrados y soluciones aplicadas.

Adicionalmente, se recopilaron evidencias visuales del proceso, tales como capturas del estado de los pods, servicios desplegados y resultados de pruebas. Estas evidencias se organizan en un anexo al final del documento, donde se describen de manera detallada con el fin de respaldar el trabajo realizado y facilitar su validación.

3.2.4 Fase 4: Pruebas y validación

Una vez implementada la arquitectura basada en contenedores y orquestada mediante Kubernetes, se llevó a cabo una fase de pruebas orientada a validar el correcto funcionamiento, la estabilidad operativa y la disponibilidad de los componentes desplegados. Esta fase tuvo como objetivo confirmar que la aplicación continuara funcionando de manera adecuada tras la migración al nuevo esquema de infraestructura.

Las pruebas realizadas se centraron inicialmente en la verificación del estado de los pods correspondientes al frontend y al backend, asegurando que estos alcanzaran el estado Running y se mantuvieran estables durante su ejecución. Asimismo, se validó la correcta creación y funcionamiento de los servicios de Kubernetes, comprobando que el backend estuviera accesible internamente mediante un servicio de tipo ClusterIP y que el frontend pudiera ser expuesto de forma controlada mediante un servicio de tipo NodePort.

Como parte del proceso de validación, se realizaron pruebas de conectividad entre los componentes de la aplicación, verificando que el frontend pudiera comunicarse correctamente con el backend a través del nombre del servicio definido en Kubernetes. Esta prueba permitió confirmar el correcto funcionamiento de la red interna del clúster y la resolución de servicios.

Adicionalmente, se evaluó el comportamiento del sistema ante la eliminación y recreación de pods, observando que Kubernetes gestionara automáticamente la reposición de los contenedores afectados, manteniendo la continuidad del servicio. Este comportamiento evidenció la capacidad del clúster para manejar fallos parciales y garantizar la disponibilidad de los componentes.

Finalmente, los resultados obtenidos fueron comparados de manera cualitativa con el entorno anterior, basado en despliegues más estáticos y menos automatizados. Esta comparación permitió evidenciar mejoras en términos de gestión, recuperación ante fallos y organización de la infraestructura, destacando el valor añadido de la nueva arquitectura orquestada frente al esquema previo.

3.3 HERRAMIENTAS UTILIZADAS

Para la ejecución del proyecto se emplearon diversas herramientas tecnológicas orientadas a la virtualización, contenedorización y orquestación de servicios, las cuales permitieron llevar a cabo la migración y validación del portal web. A continuación, se describen las principales herramientas utilizadas:

- **Docker**

Se utilizó para la creación, construcción y gestión de imágenes de contenedores correspondientes a los componentes del frontend y backend del portal web. Esta herramienta permitió encapsular las aplicaciones junto con sus dependencias,

garantizando portabilidad y consistencia en los entornos de despliegue.

- **Kubernetes**

Fue empleado como plataforma de orquestación de contenedores, permitiendo el despliegue, la administración y la comunicación entre los distintos servicios de la aplicación. Kubernetes facilitó la gestión de pods, deployments y services, así como la recuperación automática ante fallos y la exposición controlada del frontend.

- **FusionCompute**

Se utilizó como plataforma de virtualización para soportar la infraestructura donde se desplegó el clúster de Kubernetes. Esta herramienta proporcionó el entorno virtualizado necesario para la ejecución de los nodos y la administración de los recursos computacionales del proyecto.

- **Portainer**

Se tomó como referencia del entorno previo al proceso de migración. Portainer permitió analizar la arquitectura inicial basada en contenedores, identificar sus limitaciones y servir como punto de comparación frente a la nueva solución orquestada mediante Kubernetes.

3.4 VARIABLES Y CONSIDERACIONES

3.4.1. Disponibilidad del servicio

Definición: Porcentaje de tiempo en que el sistema permanece accesible.

Métrica: $(\text{Tiempo operativo} / \text{Tiempo total}) \times 100$.

Indicador esperado: $\geq 99\%$ de disponibilidad.

Validación: Monitoreo de disponibilidad de servicios (pods y endpoints) mediante Kubernetes.

3.4.2. Estabilidad del sistema

Definición: Número de fallos o reinicios inesperados del sistema.

Métrica: Número de caídas o reinicios de pods por hora/día.

Indicador esperado: ≤ 2 reinicios no controlados por día.

Validación: Logs de Kubernetes y estado de pods (CrashLoopBackOff, RestartCount).

3.4.3. Rendimiento

Definición: Tiempo de respuesta entre solicitudes frontend-backend.

Métrica: Tiempo promedio de respuesta (ms).

Indicador esperado: ≤ 500 ms por solicitud.

Validación: Pruebas de API (Postman o curl) y medición de latencia.

3.4.4 Escalabilidad

Definición: Capacidad del sistema para soportar aumento de carga.
Métrica: Número de pods activos bajo carga.
Indicador esperado: Escalamiento horizontal sin pérdida de servicio.
Validación: Pruebas de carga y verificación de replicación automática (ReplicaSet).

3.4.5. Gestión del entorno

Definición: Facilidad de administración y control del sistema.
Métrica: Tiempo promedio de despliegue y recuperación ante fallos.
Indicador esperado: Despliegue < 5 minutos, recuperación automática inmediata.
Validación: Uso de manifiestos YAML y comandos kubectl.

3.4.6. Seguridad de la información

Se implementa aislamiento por red interna (ClusterIP) y exposición controlada (NodePort).
Se evita acceso directo al backend desde el exterior.

3.4.7. Protección de datos

Uso de variables de entorno para credenciales.
No exposición de datos sensibles en imágenes Docker.

3.4.8. Integridad en la migración

Validación funcional post-despliegue (frontend y backend operativos).
Comparación de comportamiento con entorno anterior.

3.4.9. Uso correcto de herramientas

Docker para empaquetado, Kubernetes para orquestación.
Configuraciones basadas en buenas prácticas (Deployments, Services).

3.4.10 Cumplimiento organizacional

Arquitectura alineada con virtualización empresarial (FusionCompute).
Separación por capas: infraestructura, orquestación y aplicación.

3.5 CONCLUSIÓN METODOLÓGICA

La metodología aplicada en el presente proyecto permitió abordar de manera estructurada el proceso de análisis, diseño, implementación y validación de la migración del portal web hacia una arquitectura basada en contenedores orquestados con Kubernetes. El uso del enfoque analítico–sistémico facilitó la descomposición del sistema en sus componentes principales (frontend, backend y base de datos), permitiendo identificar de forma precisa las limitaciones del entorno actual, especialmente en términos de disponibilidad, escalabilidad y gestión operativa.

A partir de este análisis, fue posible definir requerimientos técnicos concretos que orientaron el diseño de la arquitectura objetivo, basada en una separación clara de capas (infraestructura, orquestación y aplicación) y en el uso de mecanismos automatizados de despliegue y recuperación ante fallos. La metodología no se limitó a una fase descriptiva, sino que permitió la transición hacia una fase propositiva y aplicada, donde se materializó la solución mediante la contenerización de servicios y su despliegue en un clúster de Kubernetes.

El carácter iterativo del método permitió realizar ajustes progresivos durante la implementación, particularmente en la construcción de imágenes, configuración de manifiestos YAML y validación de la comunicación entre servicios. Este enfoque contribuyó a reducir errores y asegurar la coherencia entre el diseño teórico y su ejecución práctica.

En la fase de validación, la metodología permitió verificar el cumplimiento de los objetivos mediante variables medibles como disponibilidad, estabilidad y rendimiento, evidenciando mejoras frente al entorno previo. Asimismo, se comprobó la capacidad del sistema para recuperarse automáticamente ante fallos y mantener la continuidad del servicio, lo cual confirma la efectividad del enfoque metodológico adoptado.

En conclusión, la metodología empleada no solo permitió cumplir con los objetivos del proyecto, sino que también demostró ser adecuada para proyectos de modernización de infraestructura tecnológica, al integrar análisis técnico, diseño estructurado, implementación práctica y validación basada en métricas. Este enfoque garantiza que la solución propuesta no sea únicamente conceptual, sino funcional, medible y replicable en entornos reales.

CAPÍTULO IV. DESARROLLO DE LA PROPUESTA

4.1 DESARROLLO OBJETIVO GENERAL

Con el fin de dar cumplimiento al objetivo general planteado, se llevó a cabo el diseño e implementación de la migración del portal web del laboratorio hacia una arquitectura basada en contenedores orquestados con Kubernetes, Este proceso se desarrolló de manera estructurada, garantizando la estabilidad, disponibilidad y correcta operación de los servicios involucrados.

Durante la ejecución del proyecto, se contempló inicialmente el despliegue del portal web migrado sobre la plataforma de virtualización empresarial FusionCompute, con el propósito de ejecutar el clúster de Kubernetes directamente sobre dicha infraestructura. Esta decisión se fundamentó en la intención de aprovechar las capacidades de virtualización, gestión centralizada y alta disponibilidad que ofrece la plataforma.

No obstante, durante la fase de implementación se presentaron limitaciones técnicas asociadas a la arquitectura del entorno y a la gestión de imágenes de contenedores, específicamente un error en el proceso de carga y despliegue de imágenes que impidió su correcta ejecución dentro del entorno administrado por FusionCompute. A pesar de múltiples intentos de validación y ajuste, esta restricción no pudo ser superada dentro del alcance y los tiempos establecidos para el proyecto.

Ante esta situación, se tomó la decisión de realizar el despliegue del portal web en un entorno local de Kubernetes, lo cual permitió garantizar la correcta ejecución de los contenedores, validar el funcionamiento de la arquitectura propuesta y cumplir con los objetivos técnicos del proyecto. Este entorno local permitió implementar y probar satisfactoriamente los componentes de frontend, backend y base de datos, asegurando la comunicación entre servicios, la estabilidad de la aplicación y la correcta orquestación de los contenedores.

Es importante resaltar que, aunque el despliegue final no se realizó directamente sobre la plataforma FusionCompute, la arquitectura diseñada se mantiene compatible con dicho entorno, por lo que la solución puede ser escalada o migrada a una infraestructura empresarial una vez superadas las limitaciones técnicas identificadas. De esta manera, el proyecto cumple con su objetivo general al demostrar la viabilidad técnica de la migración hacia una arquitectura basada en contenedores orquestados con Kubernetes, evidenciando mejoras en estabilidad, disponibilidad y gestión del entorno.

4.2 DESARROLLO PRIMER OBJETIVO GENERAL

Con el propósito de dar cumplimiento al primer objetivo específico, se realizó un análisis detallado de la arquitectura actual del portal web del laboratorio, así como de su esquema de despliegue, con el fin de documentar sus componentes, tecnologías utilizadas, forma de interconexión y principales limitaciones en términos de estabilidad, disponibilidad y gestión del entorno.

En primera instancia, se identificó que el portal web se encontraba alojado en un sistema tercerizado, implementado bajo una arquitectura de tipo full stack, compuesta por tres componentes principales: frontend, backend y base de datos. Cada uno de estos elementos

cumplía una función específica dentro del sistema y su correcta interacción resultaba fundamental para el funcionamiento del portal y la atención de las solicitudes de los usuarios.

Como parte del análisis, se realizó una revisión de la estructura interna de los componentes frontend y backend, evaluando la organización de sus carpetas, archivos y dependencias. En el caso del frontend, se analizó la disposición de los archivos relacionados con la interfaz de usuario, estilos, recursos estáticos y lógica de presentación, lo que permitió comprender la forma en que se construye y despliega la capa visual del portal web. De igual manera, se revisaron los archivos de configuración y las dependencias utilizadas para su ejecución.

En cuanto al backend, se examinó la estructura de carpetas correspondiente a la lógica del negocio, la gestión de solicitudes, la conexión con la base de datos y los servicios expuestos por la aplicación. Este análisis permitió identificar cómo se gestionan las operaciones internas del sistema, así como las dependencias necesarias para su funcionamiento. Asimismo, se evaluó la forma en que el backend se comunica tanto con el frontend como con la base de datos.

Durante el análisis del esquema de despliegue, se revisó la manera en que estos componentes estaban alojados y conectados entre sí, identificando una fuerte dependencia entre los servicios. Esta condición generaba dificultades para la administración individual de cada componente y aumentaba el impacto de posibles fallas, ya que cualquier problema en uno de los servicios podía afectar el funcionamiento general del portal web.

Adicionalmente, se analizaron las principales limitaciones del entorno actual en términos de estabilidad, disponibilidad y gestión. Se evidenció la ausencia de mecanismos automatizados para la recuperación ante fallos, la limitada capacidad de escalabilidad y la complejidad para realizar actualizaciones o cambios sin afectar el servicio en producción. Estas condiciones incrementaban los tiempos de indisponibilidad y dificultaban la operación eficiente del portal.

Como resultado de este análisis integral, se obtuvo una visión clara tanto de la arquitectura del sistema como de la organización interna de sus componentes, lo cual permitió identificar debilidades técnicas y oportunidades de mejora. Este diagnóstico sirvió como base para el diseño de la arquitectura objetivo y para la posterior migración del portal web hacia un entorno basado en contenedores, alineado con los objetivos del proyecto.

4.3 DESARROLLO SEGUNDO OBJETIVO GENERAL

Con el fin de dar cumplimiento al segundo objetivo específico, se llevó a cabo un proceso de identificación y análisis de las principales causas de inestabilidad y fallas en la disponibilidad del portal web del laboratorio. Este análisis se realizó a partir de la información obtenida en la evaluación de la arquitectura existente, el esquema de despliegue y el comportamiento del sistema en condiciones reales de operación.

En primer lugar, se identificó que la arquitectura del portal presentaba una alta dependencia entre sus componentes, lo que implicaba que cualquier fallo en uno de los servicios —frontend, backend o base de datos— afectaba directamente la disponibilidad total del sistema. Esta falta de desacoplamiento limitaba la capacidad de aislar errores y aumentaba el impacto de incidentes técnicos sobre el funcionamiento general del portal web.

Asimismo, se evidenció la ausencia de mecanismos automatizados de recuperación ante fallos. En el entorno analizado, cuando un servicio dejaba de responder o se presentaba un error en su ejecución, era necesaria la intervención manual para restablecer su funcionamiento. Esta

condición incrementaba los tiempos de indisponibilidad y afectaba la continuidad del servicio ofrecido a los usuarios.

Otra causa relevante de inestabilidad identificada fue la limitada capacidad de escalabilidad del entorno. La arquitectura existente no permitía ajustar de manera dinámica los recursos asignados a los distintos componentes del sistema, lo que generaba problemas de rendimiento en escenarios de alta demanda o uso concurrente del portal web. Esta situación impactaba negativamente tanto la experiencia del usuario como la estabilidad general del sistema.

Adicionalmente, se identificaron dificultades en la gestión y mantenimiento del entorno tecnológico, especialmente en los procesos de actualización y despliegue de cambios. La falta de un esquema estandarizado para la gestión de versiones y dependencias aumentaba el riesgo de errores durante las actualizaciones, pudiendo generar fallas en la disponibilidad del servicio o comportamientos inesperados en la aplicación.

Finalmente, el análisis permitió establecer que las limitaciones observadas no estaban asociadas únicamente al funcionamiento de los componentes individuales, sino principalmente al modelo de despliegue y gestión del entorno. Estas causas de inestabilidad evidenciaron la necesidad de adoptar una arquitectura más robusta, flexible y automatizada, capaz de mejorar la disponibilidad del servicio y facilitar su administración.

Como resultado de este análisis, se logró identificar de manera clara las principales causas de inestabilidad y fallas en la disponibilidad del portal web, lo cual sirvió como insumo fundamental para el diseño de la arquitectura objetivo basada en contenedores y orquestación con Kubernetes, planteada en los objetivos posteriores del proyecto.

4.4 DESARROLLO TERCER OBJETIVO GENERAL

Con el fin de dar cumplimiento al tercer objetivo específico, se procedió al diseño de una arquitectura objetivo basada en contenedores, orientada a mejorar la gestión, estabilidad y disponibilidad de los servicios que conforman el portal web del laboratorio. Este diseño se fundamentó en los hallazgos obtenidos durante el análisis de la arquitectura existente y en la identificación de las causas de inestabilidad y fallas en la disponibilidad del servicio.

La arquitectura propuesta se estructuró bajo un enfoque modular, en el cual cada componente del portal web —frontend, backend y base de datos— fue definido como un servicio independiente, encapsulado en su respectivo contenedor. Este desacoplamiento permite una administración individual de cada componente, facilitando su despliegue, actualización y mantenimiento sin afectar el funcionamiento global del sistema.

Para la gestión y orquestación de los contenedores se seleccionó Kubernetes, debido a sus capacidades de automatización, escalabilidad y recuperación ante fallos. En el diseño de la arquitectura se definieron los recursos necesarios para cada servicio, así como las políticas de despliegue y comunicación que garantizan la correcta interacción entre los componentes del sistema. Asimismo, se contempló el uso de servicios y configuraciones que permiten el balanceo de carga y la alta disponibilidad de la aplicación.

Adicionalmente, el diseño de la arquitectura objetivo consideró la separación de responsabilidades entre las capas de la aplicación, lo que contribuye a una mejor organización del entorno y a una mayor claridad en la gestión de los servicios. Esta separación facilita la detección y aislamiento de fallas, así como la escalabilidad independiente de cada componente según la demanda del sistema.

La arquitectura propuesta también fue diseñada para ser compatible con un entorno de virtualización empresarial, específicamente la plataforma FusionCompute, lo que permite su despliegue sobre máquinas virtuales y su posterior escalamiento en infraestructuras más robustas. Esta característica asegura la flexibilidad de la solución y su alineación con entornos empresariales reales.

Como resultado de este proceso de diseño, se obtuvo una arquitectura objetivo más robusta, flexible y alineada con las buenas prácticas actuales en el despliegue de aplicaciones web. Esta arquitectura sienta las bases para la migración del portal web hacia un entorno basado en contenedores, permitiendo una gestión más eficiente de los servicios y preparando el sistema para los procesos de implementación y validación desarrollados en las siguientes etapas del proyecto.

4.5 DESARROLLO CUARTO OBJETIVO GENERAL

Con el propósito de cumplir el cuarto objetivo específico, se llevó a cabo la ejecución de la migración del portal web del laboratorio hacia un entorno basado en contenedores, de acuerdo con la arquitectura objetivo previamente diseñada. Este proceso tuvo como finalidad trasladar los componentes del sistema frontend, backend y base de datos a un entorno más eficiente, modular y fácil de gestionar, apoyado en tecnologías de contenedorización y orquestación.

Inicialmente, se realizó la construcción de las imágenes de contenedor correspondientes a cada componente del portal web, asegurando que cada servicio incluyera sus dependencias y configuraciones necesarias para su correcta ejecución. Estas imágenes permitieron estandarizar el entorno de despliegue y garantizar la consistencia entre las distintas etapas del proceso de migración.

Posteriormente, se definieron y configuraron los archivos de despliegue necesarios para la orquestación de los contenedores mediante Kubernetes, incluyendo los manifiestos de tipo Deployment y Service. A través de estas configuraciones, se establecieron las políticas de ejecución, comunicación y exposición de los servicios, garantizando la correcta interacción entre el frontend, el backend y la base de datos dentro del clúster.

Durante la fase de implementación, se contempló inicialmente el despliegue de la solución sobre la plataforma de virtualización empresarial FusionCompute, con el objetivo de ejecutar el clúster de Kubernetes sobre dicha infraestructura. Sin embargo, se presentaron limitaciones técnicas relacionadas con la arquitectura del entorno y con errores en el proceso de carga y despliegue de imágenes de contenedores, lo que impidió completar la migración directamente sobre esta plataforma dentro del alcance del proyecto.

Ante esta situación, se procedió a realizar la implementación del portal web en un entorno local de Kubernetes, lo cual permitió validar de manera efectiva la arquitectura propuesta y asegurar la correcta ejecución de los servicios contenedorizados. En este entorno se logró desplegar satisfactoriamente el frontend, el backend y la base de datos, verificando la comunicación entre los componentes y el correcto funcionamiento del sistema.

Como resultado de esta fase, se completó la migración del portal web hacia un entorno basado en contenedores orquestados con Kubernetes, demostrando la viabilidad técnica de la solución propuesta. Aunque el despliegue final no se realizó directamente sobre la plataforma FusionCompute, la arquitectura implementada se mantiene compatible con entornos de virtualización empresarial, lo que permite su futura implementación una vez superadas las

limitaciones técnicas identificadas.

4.6 DESARROLLO QUINTO OBJETIVO GENERAL

Con el fin de dar cumplimiento al quinto objetivo específico, se realizó una evaluación de los resultados obtenidos tras la migración del portal web del laboratorio hacia una arquitectura basada en contenedores orquestados con Kubernetes. Esta evaluación se orientó a analizar las mejoras alcanzadas en términos de estabilidad, disponibilidad y gestión del entorno, en comparación con la arquitectura previamente utilizada.

En primer lugar, se evaluó la estabilidad del sistema, observándose una mejora significativa en el comportamiento del portal web frente a fallos o interrupciones de los servicios. Gracias a la orquestación proporcionada por Kubernetes, los contenedores pudieron reiniciarse de manera automática ante errores de ejecución, lo que redujo la necesidad de intervención manual y minimizó los tiempos de indisponibilidad del sistema.

En cuanto a la disponibilidad del servicio, se evidenció que la arquitectura basada en contenedores permitió una mayor continuidad operativa del portal web. La capacidad de Kubernetes para gestionar el ciclo de vida de los contenedores y mantener los servicios activos contribuyó a garantizar que la aplicación permaneciera accesible durante las pruebas realizadas, incluso ante fallos parciales de alguno de los componentes.

Asimismo, se evaluaron las mejoras en la gestión del entorno tecnológico. La migración a contenedores facilitó la administración de los servicios, permitiendo una mejor organización del sistema y una mayor claridad en la configuración y despliegue de cada componente. La definición de manifiestos de Kubernetes permitió estandarizar el proceso de despliegue, simplificando tareas como actualizaciones, mantenimiento y ajustes de configuración.

Adicionalmente, la evaluación permitió identificar que la nueva arquitectura ofrece una mayor flexibilidad y escalabilidad frente a la solución anterior. Aunque las pruebas se realizaron en un entorno local, los resultados evidenciaron que la arquitectura diseñada es capaz de adaptarse a entornos más robustos, incluyendo infraestructuras de virtualización empresarial, una vez superadas las limitaciones técnicas identificadas.

Como resultado de esta evaluación, se concluye que la migración del portal web hacia una arquitectura basada en contenedores y orquestada con Kubernetes permitió cumplir los objetivos planteados, logrando mejoras en estabilidad, disponibilidad y gestión del entorno. Estos resultados validan la viabilidad técnica de la solución propuesta y demuestran su pertinencia como una alternativa moderna y eficiente para el despliegue y administración de aplicaciones web en entornos académicos y empresariales.

CAPÍTULO V. ANALISIS

La metodología propuesta para el desarrollo del proyecto permitió abordar de manera estructurada el análisis y la mejora de la infraestructura tecnológica que soporta el portal web del laboratorio. A través de las diferentes fases metodológicas se obtuvieron resultados claros que evidencian tanto el estado real del sistema como las principales problemáticas que justifican la necesidad de migrar hacia una arquitectura más robusta y eficiente.

En la fase inicial de análisis del entorno actual, se llevó a cabo la revisión detallada de la infraestructura existente, identificando la forma en que el portal web se encontraba desplegado y administrado. Durante esta etapa se constató que el sistema operaba sobre un entorno contenedorizado con Docker, gestionado mediante Portainer, en el cual se habían definido únicamente dos contenedores principales. Esta configuración, si bien funcional para escenarios básicos, demostró ser limitada para un entorno que requiere estabilidad, disponibilidad y facilidad de gestión.

Como resultado de este análisis, se evidenció que la arquitectura presentaba una fuerte dependencia entre los contenedores, sin mecanismos de orquestación que permitieran gestionar adecuadamente fallos, reinicios automáticos o escalabilidad. Ante cualquier interrupción de un contenedor, el servicio completo del portal web se veía afectado, generando indisponibilidad para los usuarios del laboratorio. Esta situación puso de manifiesto una de las principales problemáticas del sistema: la falta de tolerancia a fallos y de alta disponibilidad.

Asimismo, durante esta fase se identificaron dificultades en la gestión operativa del entorno. Las tareas de actualización, mantenimiento y despliegue de nuevas versiones del portal se realizaban de forma manual, lo que incrementaba el riesgo de errores y tiempos de indisponibilidad del servicio. La ausencia de un control centralizado del ciclo de vida de la aplicación limitaba la capacidad de respuesta ante fallos y complicaba la administración del sistema en general.

En la fase de evaluación técnica, se analizaron los flujos de comunicación entre el frontend y el backend, así como el consumo de recursos del sistema. Los resultados mostraron que la arquitectura actual no permitía un uso eficiente de los recursos disponibles, debido a que no existían mecanismos de asignación dinámica ni monitoreo continuo del rendimiento. Esto generaba un uso ineficiente de la infraestructura y dificultaba la detección temprana de cuellos de botella o fallos de desempeño.

Otra problemática identificada durante la aplicación de la metodología fue la falta de escalabilidad del entorno. El análisis demostró que la infraestructura no estaba preparada para soportar un aumento en la cantidad de usuarios o servicios sin realizar modificaciones manuales significativas. Esta limitación resulta crítica en un laboratorio tecnológico, donde la demanda de acceso puede variar considerablemente dependiendo de las actividades de capacitación y pruebas técnicas que se estén desarrollando.

En la fase de diseño de la solución, los hallazgos obtenidos durante el análisis del entorno actual sirvieron como base para definir los requerimientos de la arquitectura objetivo. La metodología permitió establecer de manera clara que la solución debía incorporar mecanismos de orquestación, automatización y monitoreo, con el fin de superar las limitaciones identificadas. A partir de este análisis, se concluyó que la adopción de Kubernetes como plataforma de orquestación permitiría resolver de manera efectiva las problemáticas relacionadas con la disponibilidad, escalabilidad y gestión del sistema.

Durante esta fase, también se evidenció la necesidad de migrar el entorno hacia una infraestructura propia basada en virtualización empresarial, lo que permitiría un mayor control sobre los recursos de cómputo, red y almacenamiento. Esta decisión metodológica responde directamente a los problemas detectados en la fase de análisis, especialmente aquellos relacionados con la dependencia de plataformas externas y la limitada capacidad de administración del entorno actual.

En la fase de validación de la propuesta, el enfoque metodológico permitió contrastar los objetivos planteados con los resultados esperados de la nueva arquitectura. El análisis evidenció que la solución propuesta responde de manera coherente a las problemáticas identificadas, ya que incorpora alta disponibilidad, escalabilidad automática, mejor gestión del ciclo de vida de los contenedores y mayor estabilidad del servicio. De esta forma, la metodología facilitó una evaluación crítica de la propuesta, asegurando su viabilidad técnica y su alineación con las necesidades reales del laboratorio tecnológico.

En conclusión, el análisis de la aplicación de la metodología demuestra que el enfoque analítico-sistémico empleado permitió identificar de manera clara las deficiencias del entorno actual y transformarlas en criterios técnicos para el diseño de una solución mejorada. Las problemáticas detectadas en términos de disponibilidad, escalabilidad, gestión y control del sistema evidencian la necesidad de la migración propuesta, validando la metodología como una herramienta adecuada para abordar la problemática planteada en el proyecto.

CAPÍTULO VI. RESULTADOS

La aplicación de la metodología propuesta permitió obtener resultados claros y verificables en relación con el estado del entorno tecnológico del portal web del laboratorio y con la viabilidad de la solución planteada. A lo largo de las diferentes fases del proyecto se lograron identificar, analizar y documentar de manera sistemática las principales condiciones operativas del sistema, así como los beneficios esperados de la arquitectura objetivo definida.

Como resultado de la fase de análisis del entorno actual, se obtuvo una caracterización detallada de la infraestructura existente. Se identificó que el portal web se encontraba desplegado en un entorno contenedorizado básico, administrado mediante Docker y Portainer, con una estructura limitada a dos contenedores principales. Este análisis permitió confirmar que, aunque el sistema cumplía con su función básica de acceso al laboratorio, presentaba deficiencias estructurales que afectaban su estabilidad, disponibilidad y capacidad de crecimiento. La identificación de estas limitaciones constituyó un resultado clave, ya que proporcionó una base técnica sólida para la formulación de la propuesta de mejora.

Durante la valuación técnica del sistema, se obtuvieron resultados que evidenciaron la ausencia de mecanismos de automatización, monitoreo y recuperación ante fallos. Se constató que los reinicios de servicios, las actualizaciones y las tareas de mantenimiento dependían en gran medida de acciones manuales, lo que incrementaba el riesgo de errores operativos y tiempos de indisponibilidad. Este resultado permitió establecer de manera objetiva que la arquitectura actual no era adecuada para un entorno que requiere alta disponibilidad y continuidad del servicio.

En cuanto al análisis de la escalabilidad y gestión de recursos, los resultados demostraron que el sistema no contaba con la capacidad de adaptarse dinámicamente a variaciones en la carga de trabajo. El aumento en el número de usuarios o servicios implicaba modificaciones manuales en la infraestructura, lo cual limitaba la flexibilidad del entorno. Este hallazgo confirmó la necesidad de adoptar una solución que permitiera una gestión más eficiente de los recursos y una escalabilidad controlada.

Como resultado de la fase de diseño de la arquitectura objetivo, se definió una propuesta tecnológica orientada a superar las problemáticas identificadas. La arquitectura diseñada incorpora contenedores orquestados mediante Kubernetes, desplegados sobre una plataforma de virtualización empresarial, lo que permite mejorar significativamente la administración del sistema. Este diseño representa un resultado concreto del proyecto, ya que traduce los hallazgos del análisis en una solución técnica estructurada, alineada con las buenas prácticas de la industria.

Otro resultado relevante fue la definición de una arquitectura lógica basada en capas, que facilita la separación de responsabilidades entre infraestructura, orquestación y aplicación. Esta estructura mejora la mantenibilidad del sistema y permite una mayor claridad en la gestión de los componentes, reduciendo la complejidad operativa observada en el entorno original. Asimismo, se establecieron mecanismos para la gestión del ciclo de vida de las aplicaciones, lo que contribuye a una mayor estabilidad y control del entorno.

En la fase de validación conceptual de la propuesta, los resultados evidenciaron que la arquitectura objetivo cumple con los requerimientos técnicos definidos a partir del análisis inicial. La solución propuesta permite mejorar la disponibilidad del servicio mediante la tolerancia a fallos, optimizar el uso de los recursos a través de la orquestación y facilitar la escalabilidad del sistema de acuerdo con la demanda. Estos resultados confirman la viabilidad técnica de la propuesta y su adecuación a las necesidades del laboratorio tecnológico.

Desde el punto de vista académico, uno de los resultados más relevantes del proyecto fue la aplicación práctica de conocimientos relacionados con virtualización, contenedorización y orquestación de servicios. El desarrollo del proyecto permitió fortalecer habilidades técnicas y analíticas, así como la capacidad de diseñar soluciones tecnológicas fundamentadas en un análisis riguroso del entorno.

En síntesis, los resultados obtenidos demuestran que la metodología aplicada permitió alcanzar los objetivos planteados, al proporcionar un diagnóstico claro del estado actual del sistema y al generar una propuesta de mejora coherente, estructurada y técnicamente viable. La arquitectura objetivo diseñada constituye un resultado significativo del proyecto, ya que ofrece una alternativa sólida para mejorar la estabilidad, disponibilidad y gestión del portal web del laboratorio tecnológico.

CAPÍTULO VII. DISCUSION

La discusión de los resultados obtenidos en el presente proyecto permite analizar de manera crítica los hallazgos derivados de la aplicación de la metodología, así como su relación con los objetivos planteados y con los principios teóricos que sustentan el uso de tecnologías de virtualización, contenedorización y orquestación en infraestructuras modernas.

En primer lugar, los resultados evidencian que la arquitectura inicial del portal web del laboratorio, aunque funcional, no respondía adecuadamente a las necesidades operativas de un entorno académico–tecnológico que demanda alta disponibilidad, estabilidad y capacidad de crecimiento. Esta situación concuerda con lo expuesto en la literatura especializada, donde se establece que los despliegues contenedorizados sin orquestación tienden a presentar limitaciones significativas en escenarios de mayor complejidad o carga operativa. En este sentido, los hallazgos del proyecto confirman que el uso de contenedores por sí solo no garantiza una infraestructura robusta si no se acompaña de mecanismos adecuados de gestión y automatización.

Asimismo, la falta de escalabilidad y tolerancia a fallos identificada durante el análisis refleja una problemática común en entornos donde el crecimiento del sistema no ha sido considerado desde su diseño inicial. La dependencia de procesos manuales para la administración del ciclo de vida de la aplicación no solo incrementa el riesgo de errores, sino que también limita la capacidad de adaptación del sistema frente a nuevas demandas. Estos resultados refuerzan la importancia de adoptar arquitecturas orientadas a la automatización y a la gestión eficiente de recursos, tal como lo plantean las buenas prácticas actuales en infraestructura de TI.

La propuesta de migración hacia una arquitectura basada en Kubernetes surge directamente como respuesta a las problemáticas detectadas. En la discusión de los resultados, se evidencia que la arquitectura objetivo no solo aborda las deficiencias técnicas identificadas, sino que también introduce un cambio conceptual en la forma de administrar el sistema. El paso de una gestión manual y reactiva a un modelo automatizado y proactivo representa una mejora significativa en términos de operación, mantenimiento y continuidad del servicio.

Es importante destacar que, si bien la propuesta diseñada ofrece múltiples ventajas, su implementación también implica nuevos retos. La adopción de tecnologías de orquestación requiere mayores conocimientos técnicos, una correcta planificación del entorno y una curva de aprendizaje asociada a la administración del clúster. No obstante, estos retos se consideran asumibles frente a los beneficios obtenidos, especialmente en un entorno de laboratorio tecnológico donde el aprendizaje y la experimentación forman parte del propósito institucional.

Desde una perspectiva metodológica, la discusión permite afirmar que el enfoque analítico–sistémico resultó adecuado para abordar la problemática planteada. La descomposición del

sistema en sus componentes permitió identificar fallas específicas, mientras que la visión integral del entorno facilitó el diseño de una solución coherente y alineada con las necesidades reales del laboratorio. La relación directa entre análisis, resultados y propuesta demuestra la consistencia interna del proyecto.

En el ámbito académico, los resultados y su discusión evidencian el valor formativo del proyecto, al permitir la aplicación práctica de conceptos teóricos relacionados con infraestructura, virtualización y orquestación de servicios. La experiencia adquirida durante el desarrollo del proyecto contribuye al fortalecimiento de competencias técnicas y analíticas, fundamentales para el desempeño profesional en el área de tecnologías de la información.

En síntesis, la discusión de los resultados confirma que la problemática identificada fue correctamente abordada mediante la metodología aplicada y que la solución propuesta se encuentra técnicamente fundamentada. Los hallazgos obtenidos no solo validan la necesidad de la migración planteada, sino que también aportan una visión crítica sobre la importancia de diseñar infraestructuras tecnológicas escalables, disponibles y alineadas con las demandas actuales del entorno digital.

CAPÍTULO VIII. CONCLUSIONES

- El desarrollo del presente proyecto permitió evidenciar la importancia de aplicar una metodología estructurada para el análisis y mejora de infraestructuras tecnológicas que soportan servicios críticos, como el portal web del laboratorio tecnológico. A través del enfoque analítico–sistémico empleado, fue posible comprender de manera integral el funcionamiento del sistema, identificar sus principales limitaciones y establecer criterios técnicos claros para la formulación de una propuesta de mejora viable y alineada con las necesidades del entorno.
- A partir del análisis del entorno actual, se concluye que la arquitectura existente, basada en un despliegue contenedorizado gestionado de forma básica, presentaba deficiencias significativas en términos de disponibilidad, escalabilidad y gestión operativa. La ausencia de mecanismos de orquestación, automatización y monitoreo centralizado generaba dependencias críticas entre los componentes del sistema, incrementando el riesgo de fallos y afectando la continuidad del servicio ofrecido a los usuarios del laboratorio.
- Asimismo, se evidenció que la gestión manual del ciclo de vida de la aplicación limitaba la capacidad de respuesta ante incidentes, dificultaba los procesos de actualización y aumentaba la probabilidad de errores operativos. Estas problemáticas resultan especialmente críticas en un entorno académico y tecnológico donde se requiere estabilidad, flexibilidad y disponibilidad constante para el desarrollo de actividades de capacitación, pruebas técnicas y entrenamientos especializados.
- La aplicación de la metodología permitió transformar los hallazgos del análisis en requerimientos técnicos concretos, los cuales sirvieron como base para el diseño de la arquitectura objetivo. En este sentido, se concluye que la adopción de una arquitectura basada en contenedores orquestados mediante Kubernetes representa una solución adecuada para superar las limitaciones identificadas, al ofrecer capacidades de alta disponibilidad, escalabilidad automática, mejor administración de los recursos y mayor control sobre el entorno de ejecución.
- De igual manera, la decisión de implementar la solución sobre una plataforma de virtualización empresarial contribuye a fortalecer la gestión de la infraestructura, permitiendo una administración más eficiente de los recursos de cómputo, red y almacenamiento, y reduciendo la dependencia de plataformas externas. Esta elección se alinea con las buenas prácticas de la industria y con las necesidades operativas del laboratorio tecnológico.

- Desde una perspectiva académica, el proyecto permitió aplicar de manera práctica los conocimientos adquiridos durante la formación profesional, fortaleciendo competencias en análisis de arquitecturas, contenedorización, orquestación y gestión de infraestructuras tecnológicas. El desarrollo del proyecto evidenció la relevancia de integrar conceptos teóricos con escenarios reales, favoreciendo la toma de decisiones técnicas fundamentadas y el desarrollo de soluciones orientadas a la mejora continua.
- En conclusión, el proyecto cumple con los objetivos planteados al identificar las problemáticas del entorno actual, analizar sus causas y proponer una solución tecnológica coherente y técnicamente viable. La metodología utilizada demostró ser adecuada para abordar la problemática planteada, y la propuesta de migración constituye una alternativa sólida para mejorar la estabilidad, disponibilidad y gestión del portal web del laboratorio tecnológico, aportando valor tanto a nivel técnico como académico.

CAPÍTULO IX. RECOMENDACIONES

Con base en los resultados obtenidos y el análisis realizado a lo largo del desarrollo del proyecto, se presentan las siguientes recomendaciones orientadas a fortalecer la implementación de la arquitectura propuesta y a garantizar la sostenibilidad del sistema en el tiempo.

Se recomienda avanzar hacia la implementación gradual de la arquitectura objetivo basada en contenedores orquestados, priorizando inicialmente los servicios críticos del portal web. Este enfoque progresivo permitiría reducir riesgos operativos, facilitar la validación de la solución y asegurar una transición controlada desde el entorno actual hacia la nueva infraestructura.

Es recomendable establecer políticas claras de monitoreo y observabilidad del sistema, incorporando herramientas que permitan supervisar el estado de los contenedores, el consumo de recursos y el rendimiento de los servicios en tiempo real. La adopción de mecanismos de monitoreo proactivo contribuirá a la detección temprana de fallos y a la mejora continua de la estabilidad y disponibilidad del portal web.

Asimismo, se sugiere documentar de manera detallada la arquitectura implementada, los procedimientos de despliegue, actualización y recuperación ante fallos. Una documentación técnica adecuada facilitará las labores de administración del sistema, permitirá la transferencia de conocimiento y reducirá la dependencia de personal específico para la operación del entorno.

Se recomienda capacitar al personal técnico y a los responsables del laboratorio en el uso y administración de tecnologías de orquestación y virtualización. El fortalecimiento de competencias técnicas resulta fundamental para garantizar una correcta gestión del clúster, un uso eficiente de los recursos y una adecuada respuesta ante incidentes operativos.

Adicionalmente, se sugiere implementar prácticas de automatización para los procesos de despliegue y actualización de la aplicación, tales como pipelines de integración y entrega continua. Estas prácticas permitirán minimizar errores humanos, reducir tiempos de indisponibilidad y mejorar la calidad del servicio ofrecido por el portal web.

Desde una perspectiva de mejora continua, se recomienda evaluar periódicamente el desempeño de la infraestructura y realizar ajustes en función de la demanda del laboratorio. La revisión constante de métricas de uso, rendimiento y disponibilidad permitirá adaptar la solución a nuevos requerimientos y garantizar su escalabilidad a largo plazo.

Finalmente, se recomienda considerar este proyecto como una base para futuros trabajos de investigación o desarrollo, orientados a profundizar en aspectos como la seguridad del entorno, la alta disponibilidad avanzada o la optimización del consumo de recursos. De esta manera, la solución propuesta podrá evolucionar y mantenerse alineada con las tendencias tecnológicas y las necesidades del entorno académico y empresarial.

REFERENCIAS BIBLIOGRAFICAS

1. SELECTHUB. Hyper-V vs vSphere | Which server virtualization software wins in 2026 Disponible en: SelectHub.
2. TECHWRIX. VMware vSphere vs. Hyper-V: Which virtualization solution should you user 2025. Disponible en: Techwrix.
3. GEEKSFORGEEEKS. How to design a microservices architecture with Docker containers. 2025. Disponible en: GeeksforGeeks.
4. INTELLISOFT. Docker and microservices: The future of scalable and resilient application development. 2024. Disponible en: IntelliSoft.
5. GUDELLI, V. R. Kubernetes-based orchestration for scalable cloud solutions. 2025. Disponible en: IJNRD.
6. IBM. Benefits of Kubernetes. s.f. Disponible en: IBM.
7. DAS, S. Container orchestration and Kubernetes enhancements. 2025. Disponible en: IJSAT.
8. AQASIZADE, H.; ATAIE, E.; BASTAM, M. Experimental assessment of containers running on top of virtual machines. 2024. Disponible en: arXiv.
9. MARTIN, E. Virtualization and containerization: A new concept for data center management. 2021. Disponible en: Università Ca' Foscari Venezia.
10. ROUTECLOUDS. CI/CD pipelines with containers. 2025. Disponible en: Dev.to.
11. CLOUDOPTIMO. Kubernetes for CI/CD: A complete guide. 2025. Disponible en: CloudOptimo.
12. MIRANDA TORREZ. Teorías de la transformación digital. 2023. Disponible en: Repositorio FCA UNAM.
13. SCC EN LA RED. La transformación digital es clave para la optimización de TI. s.f. Disponible en: SCC en la Red.
14. PRICEWATERHOUSECOOPERS (PwC). Computación en la nube. s.f. Disponible en: PwC Colombia.
15. MONROY TENORIO, F.; CRUZ ARROYO, A. Cloud Computing: La computación de la nube. 2022. Disponible en: COINTIC.
16. UNIDAD PARA LAS VÍCTIMAS. Guía de estandarización y migración de bases de datos. s.f. Disponible en: Unidad para las Víctimas.
17. SCIELO COLOMBIA. Artículo científico. 2010. Disponible en: SciELO.
18. SCIENCEDIRECT. Artículo científico sobre gestión tecnológica. 2019. Disponible en: ScienceDirect.
19. NAMASTEDEV. Designing full stack architectures with best practices. s.f. Disponible en: NamasteDev.
20. AMAZON WEB SERVICES (AWS). ¿Qué es la contenedorización? s.f. Disponible en: AWS
21. MIRANTIS. What is Kubernetes orchestration? s.f. Disponible en: Mirantis.
22. SCIELO COLOMBIA. Artículo científico sobre sistemas de información. 2010. Disponible en: SciELO.
23. FUNCIÓN PÚBLICA. Norma colombiana. s.f. Disponible en: Función Pública.
24. FUNCIÓN PÚBLICA. Norma colombiana. s.f. Disponible en: Función Pública.
25. FUNCIÓN PÚBLICA. Norma colombiana. s.f. Disponible en: Función Pública.
26. FUNCIÓN PÚBLICA. Norma colombiana. s.f. Disponible en: Función Pública.

27. FUNCIÓN PÚBLICA. Norma colombiana. s.f. Disponible en: Función Pública.
28. International Organization For Standardization (Iso). Normas Iso. S.F. Disponible En: Iso.
29. Shah, S. (2026). Deploying A Multi-Tier Application In Kubernetes (Frontend, Backend, Database). Medium. Disponible En: <https://Medium.Com>
30. Shah, S. (2026). Deploying A Multi-Tier Application In Kubernetes. Medium. Disponible En: <https://Medium.Com> 2
31. Ibm. ¿Qué Es Kubernetes? Disponible En: <https://Www.Ibm.Com/Es-Es/Think/Topics/Kubernetes> 3
32. Ibm. ¿Qué Es Kubernetes? Disponible En: <https://Www.Ibm.Com/Es-Es/Think/Topics/Kubernetes> 3
33. Kubernetes Documentation. Conceptos De Kubernetes. Disponible En: <https://Kubernetes.io/Es/Docs/Concepts/>
34. Ibm. ¿Qué Es Kubernetes? (Sección Relacionada Con Contenedores Y Virtualización). Disponible En: <https://Www.Ibm.Com/Es-Es/Think/Topics/Kubernetes>
35. Datacamp. Arquitectura De Kubernetes. Disponible En: <https://Www.Datacamp.Com/Es/Blog/Kubernetes-Architecture-Explained> 5
36. Página web Laboratorio Huawei Colombia <https://www.colservicelab.net/>
37. HUAWEI. FusionCompute – Support and Documentation (PID 8576912) [en línea]. Huawei Enterprise Support. Disponible en: <https://support.huawei.com/enterprise/en/scale-out-storage/fusioncompute-pid-8576912>. [Consultado: 23 de junio de 2026]

ANEXO A – Evidencias de la implementación

Figura A1. Estado del clúster de Kubernetes

Se observa la configuración y el estado general del clúster de Kubernetes utilizado en el entorno local. Esta imagen permite verificar la correcta inicialización del clúster y la disponibilidad de los nodos necesarios para el despliegue de los servicios.

Status	Nodes	Total CPU...	Total Mem...	Cluster Version	Created	CPU Usage	Memory Usage	Operation
Normal	3	6	14			13.39%	46.54%	WebCL... Configur...

Figura A2. Creación de imagen de prueba en Docker

La imagen muestra el proceso de construcción de una imagen de prueba utilizando Docker. Esta acción permitió validar el correcto funcionamiento del motor de contenedores antes de proceder con la implementación completa del sistema.

Images

Give feedback

Local

My Hub

406.23 MB / 4.9 GB in use

8 Images

Last refresh: 2 hours ago

Search

	Name	Tag	Image ID	Created	Size	Actions
☐	hola-mundo-node	latest	f12f23ceabd9	4 days ago	229.21 MB	
☐		latest	f12f23ceabd9	4 days ago	229.21 MB	

Figura A3. Creación de pod de prueba

Se evidencia la creación de un pod dentro del clúster de Kubernetes, utilizado como prueba inicial para verificar la capacidad de despliegue de aplicaciones en el entorno configurado.

Nodes		Services		
Name	Status	Name	Cluster IP	Ports
desktop-control-plane	Ready	hola-backend-service	10.96.9.96	3000/TCP
		kubernetes	10.96.0.1	443/TCP
		backend-prod-service	10.96.174.150	3000/TCP
Ingresses		Config maps and secrets		
No ingresses		Name		
		kube-root-ca.crt		

Figura A4. Estructura del proyecto backend

La imagen presenta la organización de carpetas y archivos del backend, lo que permitió identificar componentes clave como controladores, configuraciones y dependencias necesarias para su contenerización.

.git	5/12/2026 2:55 PM	File folder	
ad	5/12/2026 2:55 PM	File folder	
controllers	5/12/2026 2:55 PM	File folder	
db	5/12/2026 2:55 PM	File folder	
middleware	5/12/2026 2:55 PM	File folder	
smtp	5/12/2026 2:55 PM	File folder	
.env	5/12/2026 2:24 PM	ENV File	1 KB
.gitignore	5/12/2026 2:24 PM	Git Ignore Source ...	1 KB
docker-compose.yml	5/12/2026 2:24 PM	YML File	1 KB
Dockerfile	5/12/2026 2:24 PM	File	1 KB
package.json	5/12/2026 2:24 PM	JSON Source File	1 KB
package-lock.json	5/12/2026 2:24 PM	JSON Source File	47 KB
server.js	5/13/2026 11:54 AM	JavaScript Source ...	8 KB

Figura A5. Estructura del proyecto frontend

Se muestra la estructura de directorios del frontend, permitiendo comprender la organización de componentes, archivos estáticos y configuraciones necesarias para su despliegue.

Name	Date modified	Type	Size
.git	5/12/2026 2:55 PM	File folder	
public	5/12/2026 2:55 PM	File folder	
src	5/12/2026 2:55 PM	File folder	
.env	5/13/2026 5:30 PM	ENV File	1 KB
.gitignore	5/12/2026 2:23 PM	Git Ignore Source ...	1 KB
.prettierrc	5/12/2026 2:23 PM	PRETTIERRC File	1 KB
docker-compose.yml	5/12/2026 2:23 PM	YML File	1 KB
Dockerfile	5/13/2026 9:53 AM	File	1 KB
eslint.config.js	5/12/2026 2:23 PM	JavaScript Source ...	1 KB
index.html	5/12/2026 2:23 PM	Microsoft Edge HT...	1 KB
nginx.conf	5/12/2026 2:37 PM	CONF File	1 KB
package.json	5/12/2026 2:23 PM	JSON Source File	2 KB
package-lock.json	5/12/2026 2:23 PM	JSON Source File	131 KB
README.md	5/12/2026 2:23 PM	Markdown Source ...	1 KB
tsconfig.app.json	5/12/2026 2:23 PM	JSON Source File	1 KB
tsconfig.json	5/12/2026 2:23 PM	JSON Source File	1 KB
tsconfig.node.json	5/12/2026 2:23 PM	JSON Source File	1 KB
vite.config.ts	5/12/2026 2:23 PM	TS File	1 KB

Figura A6. Configuración del archivo nginx.conf

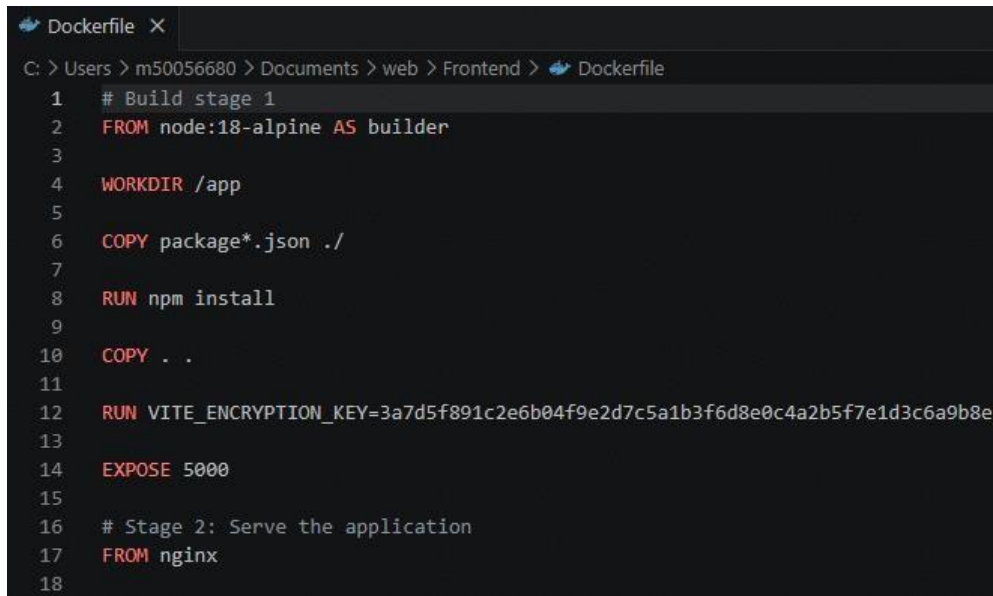
La imagen presenta el archivo de configuración de Nginx utilizado para servir el frontend. Esta configuración permite gestionar rutas y garantizar el correcto funcionamiento de aplicaciones tipo SPA.

```
nginx.conf X
C:\> Users > Documents > web > Frontend > nginx.conf
1  server {
2      listen 80;
3      server_name ColServiceLab;
4
5      location / {
6          root /usr/share/nginx/html;
7          index index.html index.htm;
8          try_files $uri $uri/ /index.html;
9      }
10 }
```

Figura A7. Configuración del Dockerfile del frontend

Se observa el contenido del Dockerfile utilizado para construir la imagen del frontend,

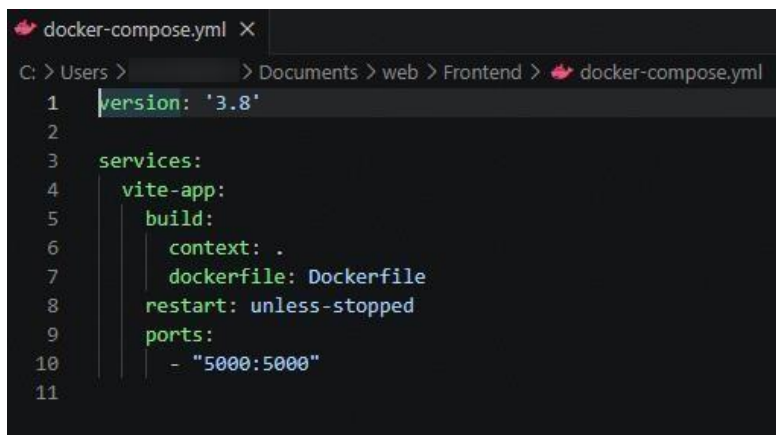
incluyendo la definición de la imagen base, el entorno de trabajo y la instalación de dependencias.



```
Dockerfile X
C: > Users > m50056680 > Documents > web > Frontend > Dockerfile
1 # Build stage 1
2 FROM node:18-alpine AS builder
3
4 WORKDIR /app
5
6 COPY package*.json ./
7
8 RUN npm install
9
10 COPY . .
11
12 RUN VITE_ENCRYPTION_KEY=3a7d5f891c2e6b04f9e2d7c5a1b3f6d8e0c4a2b5f7e1d3c6a9b8e5
13
14 EXPOSE 5000
15
16 # Stage 2: Serve the application
17 FROM nginx
18
```

Figura A8. Configuración de docker-compose

La imagen muestra el archivo docker-compose, el cual permite definir y ejecutar múltiples servicios de forma simultánea, facilitando la integración del frontend y backend en un entorno local.



```
docker-compose.yml X
C: > Users > > Documents > web > Frontend > docker-compose.yml
1 version: '3.8'
2
3 services:
4   vite-app:
5     build:
6       context: .
7       dockerfile: Dockerfile
8     restart: unless-stopped
9     ports:
10      - "5000:5000"
11
```

Figura A9. Configuración del nuevo Dockerfile del backend

Se presenta la versión ajustada del Dockerfile del backend, en la cual se realizan mejoras en la imagen base y configuración del entorno para optimizar el rendimiento y compatibilidad.

```
FROM node:22-alpine

WORKDIR /app

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 3000

CMD ["node", "server.js"]
```

Figura A10. Error de compatibilidad en FusionCompute

La imagen evidencia un error generado al intentar desplegar una imagen en FusionCompute, asociado a incompatibilidades entre arquitecturas (x86 y ARM), lo que afectó la implementación en dicho entorno.

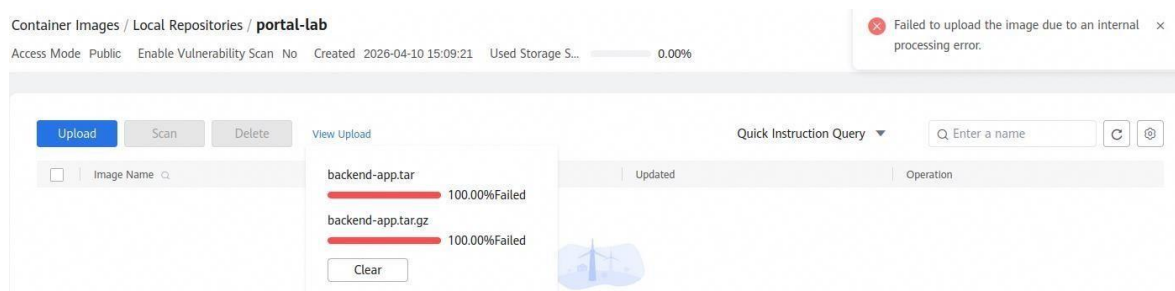


Figura A11. Error en logs de FusionCompute

Se muestran los registros de error obtenidos durante la ejecución en FusionCompute, los cuales permitieron identificar problemas técnicos en la carga y ejecución de contenedores.

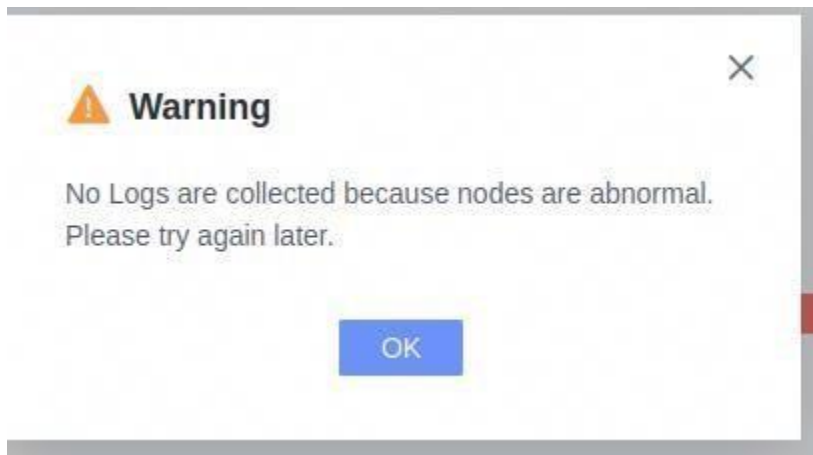


Figura A12. Creación de archivo de configuración YAML

La imagen presenta la definición de recursos en archivos YAML utilizados en Kubernetes, incluyendo configuraciones de despliegue necesarias para la ejecución de los servicios.

```
! backend-deployment.yaml X
C: > Users > m50056680 > Documents > web > k8s > ! backend-deployment.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: backend-deployment
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: backend
10   template:
11     metadata:
12       labels:
13         app: backend
14     spec:
15       containers:
16       - name: backend
17         image: backend-app:1.0
18         imagePullPolicy: IfNotPresent
19         ports:
20         - containerPort: 3000
```

Figura A13. Construcción de imagen Docker (comando build)

Se evidencia la ejecución del comando de construcción de la imagen Docker, confirmando la generación exitosa de la imagen del sistema.

```

PS C:\Users\m50056680\Documents\web> docker build -t backend-app:1.0 ./backend
[*] Building 0.9s (10/10) FINISHED
[+] Building 0.9s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 160B
=> [internal] load metadata for docker.io/library/node:22-alpine
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/node:22-alpine@sha256:8ea2348b068a9544dae7317b4f3aafcdc032df1647bb7d768a05a5cad1
=> => resolve docker.io/library/node:22-alpine@sha256:8ea2348b068a9544dae7317b4f3aafcdc032df1647bb7d768a05a5cad1
=> [internal] load build context
=> => transferring context: 6.38kB
=> CACHED [2/5] WORKDIR /app
=> CACHED [3/5] COPY package*.json ./
=> CACHED [4/5] RUN npm install
=> CACHED [5/5] COPY . .
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:de9fa8e57f1c4b335d715a780d88acc12c3411f5e7f1477c8a58929251ce684
=> => exporting config sha256:dd2e8492a8064bb3b2fb87e72b136c29114a1378b8b7561940e6c1244e6ae991
=> => exporting attestation manifest sha256:20ca46350dc957567e5ed023453903d11a568f91246c6ad61e4f54a39eeffb9d
=> => exporting manifest list sha256:6b7565e300b52b0e3d2d23bf33b82285a12bcbe45c5c035b7037fdac86e75b5a
=> => naming to docker.io/library/backend-app:1.0
=> => unpacking to docker.io/library/backend-app:1.0

```

Figura A14. Verificación de imagen creada.

Se observa la validación de la existencia de la imagen construida en el entorno local, lo que confirma que el proceso de empaquetado fue exitoso.

IMAGE	ID	DISK USAGE	CONTENT SIZE	EXTRA
backend-app:1.0	f9eaf13aaa51	258MB	62MB	
docker/desktop-cloud-provider-kind:v0.5.0	4ad59ce20658	595MB	162MB	U
docker/desktop-containerd-registry-mirror:v0.0.3	179a9ea3ae47	50.3MB	16.1MB	U
frontend-app:1.0	6f6a05584618	239MB	63.3MB	
kindest/node:v1.34.3	08497ee19eac	1.35GB	395MB	U

Figura A15. Creación de pod en Kubernetes

La imagen muestra el proceso de creación de un pod en el clúster, representando la ejecución de un contenedor dentro del entorno orquestado.

```

PS C:\Users\m50056680\Documents\web\k8s> kubectl apply -f frontend-deployment.yaml
deployment.apps/frontend-deployment created

```

Figura A16. Verificación del estado del pod

Se visualiza el estado del pod desplegado, confirmando que se encuentra en ejecución y sin errores, lo que indica un despliegue exitoso.

```

PS C:\Users\m50056680\Documents\web\k8s> kubectl get pods

```

NAME	READY	STATUS	RESTARTS	AGE
backend-deployment-566cfd557f-wqs6w	1/1	Running	0	16h
frontend-deployment-858f76f798-hbtx2	1/1	Running	0	9s

Figura A17. Configuración de servicio en Kubernetes

La imagen presenta la definición de un recurso Service, utilizado para permitir la comunicación interna entre los componentes del sistema dentro del clúster.

```

1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: backend-service
5  spec:
6    selector:
7      app: backend
8    ports:
9      - port: 3000
10       targetPort: 3000
11    type: ClusterIP

```

Figura A18. Creación del service del frontend

Se evidencia la creación del servicio correspondiente al frontend, el cual permite exponer la aplicación para su acceso desde el exterior.

```

1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: frontend-service
5  spec:
6    selector:
7      app: frontend
8    ports:
9      - port: 80
10       targetPort: 80
11       nodePort: 30080
12    type: NodePort

```

Figura A19. Verificación del despliegue del servicio.

La imagen muestra la validación del funcionamiento del servicio configurado, comprobando que la comunicación entre componentes es correcta.

```

PS C:\Users\m50056680\Documents\web\k8s> kubectl get svc
NAME                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
backend-service     ClusterIP     10.96.9.203    <none>          3000/TCP         39s
frontend-service    NodePort      10.96.27.120   <none>          80:30080/TCP     14s
kubernetes           ClusterIP     10.96.0.1      <none>          443/TCP          18h
PS C:\Users\m50056680\Documents\web\k8s> kubectl exec -it frontend-deployment-858f76f798-hbtx2 -- sh
# curl http://backend-service:3000
Backend running OK#

```

Figura A20. Visualización de la aplicación web desplegada

Se observa la interfaz del sistema en ejecución, lo que confirma que el frontend y el backend se encuentran correctamente integrados y funcionando tras el proceso de implementación.

