



Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y Tecnología

Máster Universitario en Desarrollo y Operaciones (DevOps)

**Automatización del proceso de
implementación de infraestructura TI para
soportar servicios de misión crítica**

Trabajo fin de estudio presentado por:	Antonio Rafael Gálvez Horna
Tipo de trabajo:	TFM
Director/a:	Javier Cubo Villalba
Fecha:	20/07/2023

Resumen

Este trabajo de investigación se enfoca en la automatización del proceso de implementación de infraestructura TI para soportar servicios de misión crítica, con el objetivo de abordar la complejidad de los procesos manuales, los errores humanos, los altos costos y los tiempos prolongados que actualmente se experimentan. Para lograr esto, se requiere una revisión exhaustiva del proceso actual y realizar los ajustes necesarios para mejorar la capacidad de respuesta y la confiabilidad de la infraestructura TI. Además, se propone el uso de herramientas de automatización de código abierto, como Ansible y Terraform, junto con herramientas de orquestación como AWX, Github y Jenkins.

Los resultados obtenidos demuestran mejoras en los indicadores clave del proceso, lo cual respalda la aceptación de la hipótesis general, por lo tanto; esta investigación ofrece a cualquier organización mejorar las rutinas tradicionales y transformarlas en eficientes y eficaces, capaces de simplificar y agilizar la creación y configuración de la infraestructura necesaria. Esto implica automatizar tareas como la provisión de servidores, la instalación de software, la configuración de redes y otros elementos clave. Al hacerlo, se vuelven más consistentes y confiables, ya que se basan en configuraciones estandarizadas y repetibles, proporcionando un mayor control sobre los flujos de trabajo.

Palabras clave: Automatización, Infraestructura TI, misión crítica, Open Source.

Abstract

This research work focuses on the automation of the IT infrastructure implementation process to support critical services, with the aim of addressing the complexity of manual processes, human errors, high costs and long times that are currently experienced. Achieving this requires a comprehensive review of the current process and making necessary adjustments to improve the responsiveness and reliability of the IT infrastructure. In addition, the use of open source automation tools, such as Ansible and Terraform, is proposed, along with orchestration tools such as AWX, Github, and Jenkins.

The results obtained show improvements in the key indicators of the process, which supports the acceptance of the general hypothesis, therefore; This research offers any organization to improve traditional routines and transform them into efficient and effective ones, capable of simplifying and speeding up the creation and configuration of the necessary infrastructure. This involves automating tasks such as server provisioning, software installation, network configuration, and other key elements. By doing so, they become more consistent and reliable, as they are based on standardized and repeatable configurations, providing greater control over workflows.

Keywords: Automation, IT Infrastructure, Mission-critical, Open Source.

Índice de contenidos

1. Introducción	7
1.1. Justificación del trabajo	8
1.2. Planteamiento del problema	9
1.3. Estructura de la memoria	11
2. Contexto y estado del arte	13
2.1. Contextualización y antecedentes	14
2.2. Trabajos relacionados	22
2.3. Conclusiones del estado del arte	28
3. Objetivos y metodología de trabajo	31
3.1. Objetivo general	31
3.2. Objetivos específicos	31
3.3. Metodología del trabajo	32
4. Desarrollo específico de la contribución	41
4.1. Análisis del estado actual	41
4.2. Análisis y selección de herramientas	46
4.3. Diseño e implementación de la solución	49
4.4. Evaluación y difusión de resultados	58
4.5. Aplicación de la propuesta	59
5. Conclusiones y trabajo futuro	74
5.1. Conclusiones	74
5.2. Líneas de trabajo futuro	75
Referencias bibliográficas	76
Anexo A. Entorno de automatización	79
Anexo B. Entorno de repositorio de código	80

Anexo C. Manifiesto Ansible.....	81
Anexo D. Bash Scripts	82
Anexo E. PS Scripts	84

Índice de figuras

Figura 1. The Conference Board.....	8
Figura 2. DevOps Tools Periodic Table	15
Figura 3. Modelos de servicio.	19
Figura 4. Proceso solicitud de despliegue de VMs.....	42
Figura 5. Proceso de despliegue de VMs	43
Figura 6. Proceso de configuración de backup de VMs	43
Figura 7. Proceso configuración de monitoreo de VMs.....	44
Figura 8. Proceso de validaciones post implementación del despliegue	45
Figura 9. Proceso documentación del nuevo activo de TI	45
Figura 10. Diagrama de entorno e integración	49
Figura 11. Diagrama de flujo del proceso de implementación	52
Figura 12. Diagrama de despliegue y entrega continua	54
Figura 13. Diagrama del proceso de implementación	56

Índice de tablas

Tabla 1. Comparación de indicadores en el proceso de implementación.....	10
Tabla 2. Aportes a la investigación.....	28
Tabla 3. Visualización del flujo de trabajo.....	32
Tabla 4. Solicitud de implementación	33
Tabla 5. En proceso	34

Tabla 5. Pruebas	34
Tabla 6. Despliegado	35
Tabla 7. Gestión de la capacidad.....	36
Tabla 8. Automatización de tareas.....	37
Tabla 9. Seguimiento.....	38
Tabla 10. Adaptación a cambios	39
Tabla 11. Cuadro comparativo del potencial de las herramientas.	47
Tabla 12. Cuadro comparativo del potencial de las herramientas.	57
Tabla 13. Tabulación de datos.....	59
Tabla 14. Prueba de normalidad del indicador complejidad	62
Tabla 15. Prueba de normalidad del indicador tiempo	62
Tabla 16. Prueba de normalidad del indicador costo	63
Tabla 17. Prueba de normalidad del indicador productividad	63
Tabla 18. Prueba de normalidad del indicador errores en la implementación	64
Tabla 19. Prueba estadística para indicador de tiempo	65
Tabla 20. Prueba estadística para indicador de complejidad	65
Tabla 21. Prueba estadística para indicador de costos.....	66
Tabla 22. Prueba estadística para indicador de errores	66
Tabla 23. Prueba t student del antes y después del indicador de productividad.....	67
Tabla 24. Resultados del indicador de tiempo según objetivo	67
Tabla 25. Resultados del indicador complejidad según objetivo.....	68
Tabla 26. Resultados del indicador costo según objetivo	70
Tabla 27. Resultados del indicador de errores según objetivo.....	71
Tabla 28. Resultados del indicador de productividad según objetivo	72

1. Introducción

En la actualidad, la automatización del proceso de implementación de infraestructura TI se ha convertido en un factor crucial para respaldar los servicios de misión crítica en el ámbito de la tecnología de la información (TI) (Sánchez, 2018). La utilización de herramientas ha permitido agilizar y optimizar estos procesos; ya que proporcionan soluciones eficientes para describir, controlar y orquestar la infraestructura como código (Atlassian, Infrastructure as code, 2023), lo que facilita la replicabilidad, escalabilidad y gestión de flujos de trabajo, revolucionando la forma en que se implementa la infraestructura TI, reduciendo el tiempo requerido y mejorando la eficiencia en la gestión.

El proceso manual de implementación de la infraestructura de TI suele ser costoso debido a que requiere de la intervención de expertos altamente calificados, aumenta el riesgo de errores y puede llevar a retrasos significativos en el tiempo de implementación (Cameron, 2022); en consecuencia, la automatización de estos procesos es esencial para el negocio, implicando con ello consistencia, eficiencia, rapidez, escalabilidad, reducción de errores y mayor control en los procesos de implementación y gestión de la infraestructura TI de la organización. Por esta razón, Infraestructura como Código (IaC), “integración continua y entrega continua” (CI/CD), favorecen una implementación eficiente y escalable de la infraestructura TI necesaria.

La propuesta se enfoca en optimizar las rutinas tradicionales de implementación de infraestructura TI, mediante una evaluación de diversas soluciones de automatización, con el objetivo de seleccionar la más adecuada, para posteriormente diseñar, implementar y documentar un ciclo de mejora continua. Aplicando la contrastación de la hipótesis planteada en el estudio, el objetivo principal es validar el impacto positivo de los indicadores que conlleva el proceso de automatización propuesto.

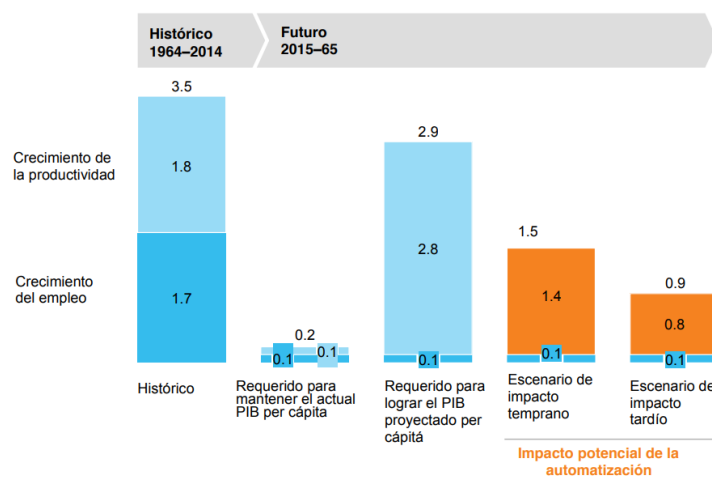
Por consiguiente, la contribución del presente estudio permitirá a las organizaciones adoptar la automatización, proporcionando la oportunidad de mejorar sus prácticas tradicionales y transformarlas en procesos más eficientes, efectivos, consistentes, confiables, bajo un mejor control de los flujos de trabajo; estandarizados y repetibles.

1.1. Justificación del trabajo

La automatización es una solución tecnológica clave para reducir errores humanos y aumentar la eficiencia de los centros de datos. Se trata de que los flujos de trabajo y procesos de rutina se autogestionen y ejecuten sin intervención humana, por lo tanto, el objetivo es que las aplicaciones y/o servicios atiendan eficazmente, sin errores humanos y en el menor tiempo posible. El problema que se quiere solucionar con este trabajo de investigación es la complejidad y el tiempo que conlleva la implementación de infraestructura TI necesaria para soportar servicios de misión crítica en las organizaciones. Es importante porque las empresas y organizaciones de todo tipo dependen cada vez más de la tecnología para llevar a cabo sus operaciones diarias. La implementación manual de plataformas tecnológicas requiere de personal calificado y puede ser costosa y susceptible a errores. Además, el tiempo para llevar a cabo la implementación puede retrasar el inicio de las operaciones, lo que puede afectar negativamente la rentabilidad de la organización.

En respuesta a la creciente necesidad de automatizar el proceso de implementación de infraestructura, según (Manyika, y otros, 2017), encontró que la automatización podría aumentar el crecimiento de la productividad a nivel mundial (entre un 0,8 y un 1,4 por ciento anual) a medida que se reduce el crecimiento del empleo anual. Como se muestra en la figura a continuación, el impacto potencial de la automatización proyectada desde el 2015 hasta el año 2065 puede ayudar a cerrar la brecha del crecimiento del PIB proveniente de la disminución de la población en edad de trabajar a nivel mundial.

Figura 1. The Conference Board



Fuente: (Manyika, y otros, 2017)

Adicionalmente, es importante contar con infraestructura robusta y confiable para garantizar la continuidad del negocio y la satisfacción de los clientes, según (ITIC, 2022), indica que un tiempo de inactividad no planificado de solo cinco minutos puede resultar en pérdidas financieras significativas para las organizaciones.

Esta investigación va en concordancia con tendencias como las mencionadas anteriormente y tiene como objetivo desarrollar una solución para automatizar el proceso de implementación de infraestructura TI que permita a las organizaciones reducir tiempo, costos y mejorar la calidad de los servicios de TI para cumplir de manera eficiente con sus objetivos de negocio, a través de scripts desarrollados con herramientas como Ansible, Terraform, Bash Scripting y Python.

La automatización de procesos de implementación de infraestructura TI, (RedHat, ¿Qué es la automatización? , 2023) tiene una alta relevancia en la transformación digital. Los entornos digitales modernos y dinámicos necesitan adaptarse más rápido que nunca, y la automatización es indispensable para lograrlo, debido a que desempeña un papel clave al ahorrar tiempo y complejidades, y garantizar una configuración consistente y segura de los sistemas.

1.2. Planteamiento del problema

La necesidad de tener una infraestructura robusta, escalable y altamente disponible para asegurar que los sistemas críticos de la empresa estén siempre en funcionamiento y disponibles para su uso, se convierten en un desafío, debido a la complejidad y el tiempo que conlleva la implementación manual de una infraestructura de TI.

El presente trabajo de investigación propone una solución que contribuya a reducir el tiempo y la complejidad de la configuración necesaria para desplegar infraestructura de TI que soporte servicios de misión crítica en las organizaciones.

Para demostrar la hipótesis, se deberá desarrollar el proceso de automatización del despliegue de máquinas virtuales sobre un entorno VMware y realizar pruebas y evaluaciones para medir su efectividad. Además, se deberá comparar los resultados obtenidos de una implementación tradicional frente a los resultados obtenidos de un proceso automatizado,

para analizar el ahorro de tiempo y recursos que se logra con la solución propuesta. A continuación, se presenta la matriz de indicadores que va a formar parte de este análisis:

Tabla 1. Comparación de indicadores en el proceso de implementación.

Indicadores	Proceso	
	Tradicional	Automatizado
Complejidad	34	28
Tiempo	36	30
Costo	2300	2000
Productividad	120	135
Errores	13	10

Fuente: Elaboración propia

Como planteamiento del presente trabajo, se pretende llegar a la siguiente conjetura de investigación:

- La implementación del proceso de automatización para el despliegue de máquinas virtuales sobre un entorno de virtualización puede mejorar significativamente la eficiencia y la confiabilidad del proceso, reducir errores humanos, costos y aumentar la productividad, en comparación con los procesos tradicionales.

Para demostrar la hipótesis general, se deberán demostrar estadísticamente las siguientes hipótesis específicas para evidenciar que si generan mejora:

- Mayor flexibilidad en la implementación: se espera que la automatización permita una mayor flexibilidad en la implementación de la infraestructura, lo que significa que se puedan adaptar mejor a cambios complejos y requerimientos del negocio y del entorno tecnológico.
- Reducción del tiempo de implementación de la infraestructura TI: se espera que la automatización permita reducir significativamente el tiempo que se tarda en implementar la infraestructura TI necesaria para soportar los servicios de misión crítica del negocio.
- Reducción de costos asociados a procesos de automatización: se espera que los costos involucrados en la adquisición de herramientas de automatización, el tiempo dedicado al

desarrollo de scripts o configuraciones automatizadas, y los costos indirectos relacionados con la reducción de errores y tiempos de inactividad; incurra en una reducción significativa de los costos totales, costos por unidad de tiempo y costo promedio por máquina virtual desplegada.

- Aumento de la eficiencia del equipo de operaciones: se espera que la automatización permita al equipo de operaciones trabajar de manera más eficiente y con mayor precisión, lo que se traducirá en una mayor productividad y una reducción en el número de horas hombre necesarias para implementar la infraestructura de TI.
- Mejora en la calidad de la implementación: se espera que la automatización garantice una mayor calidad en la implementación de la infraestructura de TI, minimizando los errores y reduciendo la necesidad de realizar correcciones posteriores.

1.3. Estructura de la memoria

El Capítulo 1, titulado "Introducción", se centra en establecer la justificación de la relevancia del trabajo, presentar el problema o necesidad que se abordará, plantear los objetivos generales y establecer la hipótesis a demostrar. Se expone la justificación de la relevancia, se presenta el problema o necesidad a abordar, se plantean los objetivos generales del trabajo y se establece la hipótesis a demostrar. La justificación destaca la importancia del tema, el problema identifica la situación que requiere atención, los objetivos describen las metas amplias del trabajo y la hipótesis plantea una afirmación que puede ser verificada o refutada. Este capítulo establece las bases para el desarrollo de la investigación y proporciona un marco conceptual y teórico para comprender el contexto y los objetivos del estudio.

Capítulo 2: Contexto y estado del arte, se centra en establecer el contexto y estado del arte del tema de investigación, se realiza una revisión bibliográfica exhaustiva y se presenta los conceptos fundamentales relacionados al tema. Este capítulo ayuda a situar el trabajo en el contexto más amplio de la investigación existente y proporciona las bases teóricas necesarias para abordar el tema de manera informada y rigurosa.

El Capítulo 3 se centra en describir el enfoque y diseño de la investigación, la población y muestra seleccionada, así como las técnicas de recolección y análisis de datos utilizadas. Se explica el enfoque y diseño de la investigación, detallando las estrategias y procedimientos utilizados para recopilar y analizar los datos. Se aborda la población y muestra de estudio,

describiendo la población objetivo y cómo se selecciona una muestra representativa. Se presentan las técnicas de recolección de datos utilizadas, como entrevistas, cuestionarios u observación, y se explican los procedimientos y protocolos utilizados. Además, se describen las técnicas de análisis de datos empleadas, como análisis estadístico o técnicas de codificación cualitativa. Este capítulo proporciona los detalles metodológicos esenciales para llevar a cabo la investigación de manera rigurosa y efectiva.

Capítulo 4: Desarrollo de la propuesta, se centra en describir la propuesta o solución desarrollada en el trabajo, su implementación y puesta en marcha, el análisis de los resultados obtenidos y la discusión y comparación de estos resultados con otros trabajos relacionados. Se proporciona una descripción detallada de la propuesta, explicando los conceptos y técnicas utilizados. Se describe la implementación práctica de la solución, detallando los pasos, recursos y protocolos utilizados. Se realiza un análisis de los resultados, presentando los datos recopilados y los hallazgos principales, utilizando técnicas de análisis de datos y elementos visuales. Se lleva a cabo una discusión de los resultados y se comparan con otros trabajos relacionados, identificando las contribuciones y limitaciones de la propuesta. Se plantean posibles direcciones para futuras investigaciones. Este capítulo brinda una visión integral del desarrollo y evaluación de la propuesta, destacando su impacto y su contexto en relación con otros trabajos en el campo.

Capítulo 5: Conclusiones y trabajo futuro, se centra en presentar las conclusiones principales obtenidas a partir de la investigación, identifica las limitaciones del estudio y propone posibles áreas de trabajo futuro. Este capítulo cierra el trabajo al resumir los logros alcanzados, reconocer las limitaciones y sugerir posibles caminos para la continuación de la investigación en el tema abordado.

2. Contexto y estado del arte

Actualmente, la automatización de procesos se ha convertido en un elemento fundamental para mejorar la eficiencia y eficacia en las organizaciones, por lo que el uso de tecnologías emergentes como la inteligencia artificial, el aprendizaje automático y la automatización robótica de procesos (RPA) se ha vuelto cada vez más común.

Es importante mencionar que, en el contexto de la infraestructura de TI, la automatización del proceso de implementación puede abarcar desde la configuración de hardware hasta la creación de entornos virtuales y la gestión de la nube. En este sentido, existen herramientas y tecnologías que permiten automatizar estas tareas, como herramientas de orquestación, plataformas de automatización de la nube y soluciones de infraestructura como código (IaC).

Así mismo cabe mencionar, que el movimiento liderado por Patrick Debois y Andrew Shafter ha derribado la barrera existente entre los departamentos de desarrollo y operaciones, fomentando una colaboración conjunta y dinámica que beneficia a ambos proyectos. Esta separación se basaba en objetivos antagónicos sostenidos por cada departamento. Mientras que el equipo de desarrollo se enfocaba en la velocidad del proyecto, el equipo de operaciones priorizaba la estabilidad. Estas diferencias incentivaban la oposición entre ellos.

Con la implementación de la metodología DevOps, tanto la estabilidad como la velocidad de entrega se mantienen sin comprometerse, ya que ambos departamentos comparten los mismos objetivos, integrando el progreso y el avance de manera conjunta. Algunos de los puntos clave son:

- **Fast Time To Market (TTM):** Busca minimizar el tiempo requerido para desarrollar el código y presentar el trabajo ante el cliente.
- **Minimizar las fallas en producción:** Trabaja en la elaboración de soluciones que reduzcan las fallas en la producción, ya que estas afectan la estabilidad del proyecto. Es fundamental asegurarse de que el desarrollo se realice de manera estable sin comprometer el TTM.
- **Recuperación inmediata de fallas:** Se reconoce que en la vida real ocurren fallas, se cometen errores y las cosas pueden romperse. Es importante aceptar esta realidad y estar preparados para actuar rápidamente frente a estas situaciones.

Basado en el presente contexto, se seguirá una metodología Agile, donde se propone un enfoque basado en la colaboración, la iteración y la adaptabilidad en estrecha colaboración

con los stakeholders. Se establecerán iteraciones cortas, donde se priorizarán los requisitos y se desarrollarán las funcionalidades en incrementos iterativos. Se utilizarán herramientas de infraestructura como código (IaC) para definir y gestionar la configuración de la infraestructura de manera automatizada. Se realizarán pruebas continuas y se recopilará el feedback de los usuarios y los stakeholders para iterar y mejorar el proceso de implementación. La comunicación constante y la adaptabilidad serán clave para asegurar la eficiencia y la calidad en la implementación de la infraestructura tecnológica.

2.1. Contextualización y antecedentes

Los procesos tradicionales de implementación de infraestructura tecnológica han evolucionado a lo largo de los años, pero todavía se caracterizan por ser enfoques manuales y repetitivos. Estos procesos implican una intervención humana extensa, lo que conlleva a una serie de desafíos y limitaciones. Estos procesos requieren un mayor tiempo de implementación. Cada paso del proceso debe ser realizado manualmente, lo que implica un mayor esfuerzo y consumo de tiempo. Además, la intervención humana aumenta el riesgo de errores cometidos, lo que puede afectar la calidad y confiabilidad de la implementación. Incluso pequeños errores pueden tener un impacto significativo en el rendimiento y la estabilidad de la infraestructura tecnológica.

Otra limitación de los procesos tradicionales es la falta de estandarización y documentación adecuada. Cada implementación puede ser única y personalizada, lo que dificulta la repetibilidad y la escalabilidad de los procesos. Además, la falta de documentación adecuada dificulta la transferencia de conocimientos y la colaboración entre los miembros del equipo, lo que puede resultar en una falta de coherencia y consistencia en las implementaciones. Se puede decir que la falta de automatización es otro aspecto importante que limita la eficiencia y adaptabilidad de los procesos tradicionales. La intervención humana en cada etapa del proceso dificulta la capacidad de respuesta y adaptación a los cambios. Los requisitos y condiciones pueden cambiar rápidamente, y los procesos manuales no pueden mantener el ritmo necesario para realizar ajustes rápidos y eficientes. Esto puede llevar a retrasos en la implementación, problemas de rendimiento y falta de agilidad en la gestión de la infraestructura.

Una de las principales bases para fomentar una cultura orientada al desarrollo con enfoque en la velocidad de entrega y la estabilidad sostenida radica en el respaldo proporcionado por herramientas específicamente diseñadas para facilitar estas prácticas. Es fundamental poder identificar las herramientas necesarias y utilizar aquellas que sean más adecuadas para cada situación. Existe una amplia oferta de herramientas disponibles para cada tarea requerida. Para explorar estas opciones, se recomienda consultar la tabla periódica de herramientas DevOps proporcionada por XebiaLabs¹. Esta tabla brinda una visión general de las diversas herramientas disponibles en el mercado y facilita la selección de las más apropiadas para cada contexto.

Figura 2. DevOps Tools Periodic Table



Fuente: <https://digital.ai/learn/devops-periodic-table/>

En este sentido, vamos a mencionar las tecnologías y conceptos más relevantes en el campo de la automatización:

2.1.1. Git

Es un sistema de control de versiones, (Git, 2023) distribuido ampliamente utilizado en el desarrollo de software. Permite gestionar y rastrear los cambios en los archivos de un proyecto, proporcionando un historial completo de modificaciones. Los desarrolladores pueden trabajar de manera colaborativa, revertir a versiones anteriores y fusionar contribuciones. Los archivos se almacenan en un repositorio, donde se registra la diferencia

¹ DevOps Tools Periodic Table: <https://digital.ai/learn/devops-periodic-table/>

entre versiones. Git es eficiente en términos de espacio y tiempo, y utiliza un enfoque distribuido, permitiendo que cada desarrollador tenga una copia completa del repositorio. También destaca por su capacidad para manejar ramas de desarrollo, facilitando el trabajo paralelo y colaborativo. Además, ofrece características como la gestión de conflictos, etiquetado de versiones y colaboración a través de plataformas de alojamiento. Git se ha convertido en una herramienta fundamental en el desarrollo de software.

2.1.2. Jira

Desarrollado por Atlassian, (Atlassian, 2023) es una potente herramienta de gestión de proyectos y seguimiento de problemas utilizada ampliamente en el desarrollo de software y otros sectores industriales. Permite registrar, organizar y dar seguimiento a tareas, problemas, errores y mejoras a lo largo del ciclo de vida del proyecto. Jira es altamente personalizable, con estructuras modulares que permiten la creación de proyectos y la adaptación de tipos de problemas, flujos de trabajo y esquemas de notificación según las necesidades del equipo. Cuenta con características adicionales y complementos que amplían su funcionalidad, como la integración con sistemas de control de versiones y herramientas de seguimiento de tiempo. También brinda soporte para la gestión de proyectos ágiles a través de su módulo "Jira Software", permitiendo el uso de metodologías como Scrum y Kanban, y proporcionando seguimiento de historias de usuario, sprints y tableros visuales de trabajo.

2.1.3. Jenkins

Es una herramienta de integración continua (CI) de código abierto, (Jenkins, 2023) que automatiza el proceso de compilación, prueba y despliegue de aplicaciones en el desarrollo de software. Supervisa los repositorios de código fuente, detecta cambios y realiza acciones predefinidas como compilación, pruebas y despliegue. Una característica clave de Jenkins es la creación y gestión de pipelines de entrega continua (CD) que definen el flujo de trabajo completo. Es altamente personalizable, con una amplia gama de complementos y una interfaz web intuitiva. Jenkins es escalable y se integra con otras herramientas como Git y JIRA, lo que lo hace adecuado para proyectos de cualquier tamaño y múltiples equipos de desarrollo.

2.1.4. Ansible, (RedHat, 2022)

Desarrollado por Red Hat, es una poderosa herramienta de automatización de TI utilizada para administrar y configurar sistemas informáticos de manera eficiente y escalable. Se basa en la

infraestructura como código (IaC), lo que permite describir y gestionar la configuración de forma programática utilizando archivos simples y legibles. Ansible se enfoca en la automatización de tareas repetitivas y la gestión de la configuración de servidores y dispositivos de red. Utiliza SSH para establecer conexiones seguras y ejecutar comandos de forma remota, lo que simplifica el proceso de implementación y administración sin requerir agentes adicionales en los sistemas. Con su lenguaje declarativo y su capacidad de escalabilidad y gestión centralizada, Ansible proporciona una solución flexible y eficiente. Además, se integra con herramientas y servicios existentes, lo que permite su adaptación a diferentes entornos y necesidades específicas de los administradores de sistemas.

2.1.5. Virtualización

Es una tecnología que permite la creación de entornos virtuales, conocidos como máquinas virtuales (VM), en un servidor físico. Estas máquinas virtuales funcionan de manera independiente y ejecutan sistemas operativos y aplicaciones en un entorno aislado. Hay varios tipos de virtualización, como la de servidor, escritorio, red y almacenamiento. La virtualización ofrece beneficios significativos, como la consolidación de servidores, mayor eficiencia y flexibilidad, aislamiento y seguridad, ahorro de costos y escalabilidad, lo que ha transformado la forma en que se implementan los sistemas informáticos.

Gracias a la virtualización, (AWS, Virtualizacion, ¿Qué es?, 2023) es posible ejecutar múltiples máquinas virtuales en un solo servidor físico, lo que permite una mejor utilización de los recursos y una reducción en el número de servidores requeridos. Además, la virtualización ofrece una mayor eficiencia y flexibilidad al permitir la rápida asignación y liberación de recursos según sea necesario. También garantiza el aislamiento y la seguridad de las máquinas virtuales, ya que funcionan de forma independiente, lo que evita que los problemas en una máquina virtual afecten a otras. Además, la virtualización facilita la copia de seguridad, restauración y pruebas de recuperación, lo que contribuye a una gestión más eficiente. En general, la virtualización ha revolucionado la forma en que se implementan los sistemas informáticos al ofrecer una mayor agilidad, ahorro de costos y escalabilidad en entornos empresariales.

Actualmente el mercado trae diversas soluciones de virtualización, podemos mencionar KVM (Kernel-based Virtual Machine), es una tecnología de virtualización de código abierto que permite ejecutar múltiples sistemas operativos como máquinas virtuales (VM) en un único

servidor físico. Se basa en el hipervisor integrado en el kernel de Linux, por otro lado, también tenemos a Red Hat Virtualization (RHV), es una solución de virtualización empresarial desarrollada por Red Hat, una de las compañías líderes en el campo del software de código abierto. RHV se basa en el hipervisor de código abierto Kernel-based Virtual Machine (KVM) y proporciona una plataforma robusta y escalable para la virtualización de servidores y escritorios.

2.1.6. VMware vSphere (VMware, 2023)

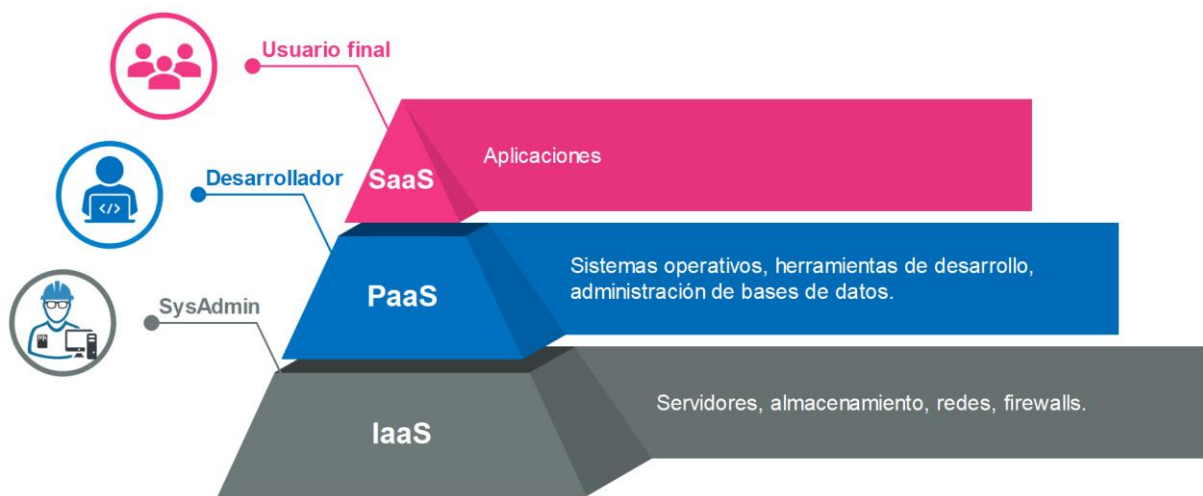
VMware es una empresa líder en el campo de la virtualización de servidores y software de infraestructura en la nube. VMware desarrolla y proporciona una amplia gama de productos y soluciones de virtualización que permiten a las organizaciones crear entornos virtuales en los que se pueden ejecutar múltiples sistemas operativos y aplicaciones en un solo servidor físico. En este sentido, se cuenta con **VMware vSphere**, la cual es una plataforma de virtualización de servidores. Proporciona una infraestructura de virtualización completa que permite a las organizaciones crear y administrar entornos virtuales altamente eficientes y escalables. vSphere permite la creación de máquinas virtuales en servidores físicos, lo que permite consolidar múltiples sistemas operativos y aplicaciones en un solo hardware. Esta consolidación optimiza el uso de recursos, reduce los costos de hardware y simplifica la administración de infraestructuras de TI, incluye una amplia gama de características y herramientas avanzadas que mejoran la disponibilidad, el rendimiento y la seguridad de los entornos virtualizados. Además de ello, ofrece capacidades de gestión centralizada, migración en vivo de máquinas virtuales, balanceo de carga, recuperación ante desastres y capacidades de alta disponibilidad.

2.1.7. El Cloud Computing, (AWS, Cloud Computing, 2023)

La computación en la nube, es un modelo de entrega de servicios informáticos a través de Internet. En lugar de tener que gestionar y mantener la infraestructura localmente, los usuarios pueden acceder a recursos informáticos a través de proveedores de servicios en la nube. Esto les permite utilizar y pagar por los servicios según su demanda, evitando la necesidad de invertir en infraestructura costosa. Los principales modelos de servicio en la nube incluyen:

- **Software como Servicio (SaaS):** Permite a los usuarios acceder y utilizar aplicaciones y software a través de la nube, sin necesidad de instalar o mantenerlos localmente.
- **Plataforma como Servicio (PaaS):** Proporciona una plataforma de desarrollo y ejecución de aplicaciones en la nube, permitiendo a los desarrolladores centrarse en la creación de software sin preocuparse por la infraestructura subyacente.
- **Infraestructura como Servicio (IaaS):** Ofrece recursos de infraestructura, como servidores virtuales, almacenamiento y redes, a través de la nube, permitiendo a los usuarios gestionar y controlar su propia infraestructura de manera flexible.

Figura 3. Modelos de servicio.



Fuente: (OpenWebinars, 2021)

El Cloud Computing proporciona ventajas como la escalabilidad, el acceso global, la eficiencia y el ahorro de costos, las actualizaciones automáticas y una mayor capacidad de almacenamiento. Permite escalar recursos según la demanda, acceder a servicios desde cualquier lugar, reducir costos de infraestructura, recibir actualizaciones de software automáticamente y almacenar grandes cantidades de datos sin limitaciones de espacio.

2.1.8. Los modelos de nube (OpenWebinars, 2021)

Se refieren a los diferentes enfoques y arquitecturas utilizados para implementar y ofrecer servicios en la nube. Estos modelos definen la forma en que los recursos informáticos, como servidores, almacenamiento y redes, se organizan y entregan a los usuarios y clientes. Existen tres modelos principales de nube:

- **Nube pública:** Los servicios y recursos informáticos son proporcionados por proveedores en la nube y están disponibles para el público a través de Internet. Los usuarios pagan por lo que consumen y evitan los costos de adquisición y mantenimiento de infraestructura.
- **Nube privada:** Los servicios y recursos informáticos se implementan y gestionan internamente dentro de una organización. Proporciona mayor seguridad y control sobre los datos y aplicaciones, siendo accesible solo para la organización y sus usuarios autorizados.
- **Nube híbrida:** Combina elementos de la nube pública y privada, permitiendo a las organizaciones utilizar una combinación de servicios y recursos de ambas nubes. Facilita la integración de aplicaciones y datos, brindando flexibilidad y opciones adicionales.

Además, existe el concepto de nube comunitaria, que se refiere a un entorno compartido por múltiples organizaciones con intereses o requisitos comunes.

La elección del modelo de nube depende de los requisitos, cargas de trabajo, seguridad y cumplimiento de una organización, considerando aspectos como escalabilidad, disponibilidad, flexibilidad y costos asociados.

2.1.9. AWS (AWS, Cloud Computing, 2023)

Es una plataforma líder en servicios en la nube proporcionada por Amazon. Ofrece una amplia gama de servicios en categorías como:

- **Cómputo:** Amazon EC2 permite ejecutar aplicaciones en máquinas virtuales escalables y flexibles.
- **Almacenamiento y bases de datos:** Amazon S3 ofrece almacenamiento duradero y seguro, mientras que Amazon RDS y Amazon DynamoDB son servicios de bases de datos administradas.
- **Redes y entrega de contenido:** Amazon VPC permite crear redes privadas virtuales, y Amazon CloudFront acelera la entrega de contenido a nivel mundial.
- **Herramientas de desarrollo y gestión:** AWS Elastic Beanstalk facilita la implementación y administración de aplicaciones web, y AWS CloudFormation automatiza la creación y gestión de recursos.

- **Inteligencia artificial y análisis:** Amazon Rekognition ofrece capacidades de análisis de imágenes y reconocimiento facial, y Amazon SageMaker es un entorno de aprendizaje automático administrado.

Estos son solo algunos ejemplos de los muchos servicios que ofrece AWS. La plataforma también cuenta con una infraestructura global robusta que garantiza la disponibilidad y la tolerancia a fallos.

2.1.10. La infraestructura como código (IaC) (RedHat, 2022)

Es un enfoque para administrar y provisionar la infraestructura de TI mediante archivos de configuración en lugar de cambios manuales. Con IaC, la infraestructura se define y gestiona a través de código, lo que automatiza y estandariza la implementación y configuración de los recursos tecnológicos. Los archivos de configuración describen los recursos necesarios, sus características y su configuración. Hay varias herramientas como Terraform, AWS CloudFormation, Azure Resource Manager y Google Cloud Deployment Manager que permiten implementar IaC de manera eficiente y reproducible. Los beneficios incluyen una gestión ágil y eficiente, cambios rápidos y consistentes, estandarización y consistencia, colaboración mejorada, trazabilidad y gobernabilidad.

2.1.11. Terraform, (Hashicorp, 2023)

Es una herramienta de infraestructura como código, desarrollada por HashiCorp, permite definir y gestionar la infraestructura de forma declarativa a través de un archivo de configuración. Utiliza un lenguaje de configuración simple y legible y es compatible con una amplia gama de proveedores de infraestructura. Terraform planifica y ejecuta cambios de manera segura y controlada, manteniendo un estado de la infraestructura que registra los recursos y su configuración. Es altamente modular y extensible, permitiendo la creación de infraestructura consistente y escalable. La comunidad de usuarios contribuye con módulos y complementos adicionales para ampliar las capacidades de Terraform.

2.1.12. La metodología Agile

Es un enfoque colaborativo, iterativo e incremental para el desarrollo de proyectos. Se centra en la flexibilidad, la adaptabilidad y la entrega temprana de valor a los clientes. **(Medina, 2021)** “Es importante recalcar que estas metodologías no se limitan a proyectos tecnológicos, sino que se pueden aplicar en cualquier ámbito o sector donde se produzcan cambios de forma

frecuente". Se rige por principios como la entrega continua, la colaboración estrecha con el cliente y la capacidad de respuesta a los cambios. Existen diferentes metodologías y marcos de trabajo que se basan en los principios y valores de Agile, como Scrum, Kanban, Lean, XP y DSDM. Todos ellos comparten la filosofía de colaboración, entrega temprana de valor y adaptabilidad.

La metodología Agile se basa en cuatro valores fundamentales, que son:

- **Individuos e interacciones sobre procesos y herramientas:** Prioriza la comunicación y la colaboración efectiva entre los miembros del equipo.
- **Software funcionando sobre documentación extensiva:** Enfoca en la entrega de software funcional y utilizable en lugar de enfocarse en documentación excesiva.
- **Colaboración con el cliente sobre negociación contractual:** Busca una relación cercana con el cliente, trabajando en colaboración para comprender y satisfacer sus necesidades.
- **Responder al cambio sobre seguir un plan:** Acepta que los requisitos y las circunstancias pueden cambiar durante el desarrollo del proyecto, promoviendo la adaptabilidad y la capacidad de respuesta rápida.

2.1.13. El enfoque DevOps

Integra las actividades de desarrollo y operaciones en un ciclo de trabajo continuo y colaborativo para mejorar la calidad y eficiencia en la entrega de software. Se basa en la colaboración y comunicación efectiva entre los equipos, la automatización de tareas repetitivas, el uso de infraestructura como código, el enfoque en la calidad y retroalimentación continua, y la entrega frecuente y en pequeños incrementos. Para implementarlo, se utilizan diversas herramientas y tecnologías como Git, Jenkins, Ansible, KVM y vSphere, entre otras.

La automatización de procesos de infraestructura es un campo en constante evolución con múltiples herramientas y tecnologías disponibles. Su adopción puede proporcionar una ventaja competitiva al mejorar la eficiencia y calidad de los procesos de infraestructura.

2.2. Trabajos relacionados

En el campo de las tecnologías de la información, la automatización y la innovación desempeñan un papel clave para mejorar la eficiencia, agilidad, implementación y gestión. Diversos investigadores han abordado esta temática desde diferentes perspectivas, con el

objetivo de optimizar los procesos, reducir costos y acelerar los despliegues de infraestructuras TI. En esta investigación, se analizarán y examinarán varios trabajos que ofrecen aportes significativos en áreas como la implementación de infraestructuras en la nube mediante código, la utilización de virtualización, el provisionamiento automático de configuraciones, la evaluación de herramientas como Terraform en el ciclo de vida del DevOps, el desarrollo de modelos de migración a la nube, la implementación de metodologías como GitOps, la aplicación de buenas prácticas en infraestructura como código (IaC) y la mejora de la automatización en las operaciones de los centros de datos. A través de este estudio, se espera obtener una visión más completa y enriquecedora sobre la automatización e innovación en infraestructuras de TI, y su impacto en la eficiencia y agilidad de los servicios tecnológicos.

Según la investigación de (Choez, 2022), el objetivo de examinar las distintas soluciones en la nube para automatizar infraestructuras, dentro del contexto del software de código abierto, con el fin de desarrollar un prototipo. Esto permitió evaluar la disponibilidad de los servicios utilizando la escala de Kuder Richardson, obteniendo un alto nivel de confiabilidad y validez. Se realizaron pruebas utilizando herramientas incluyendo Terraform para la definición de la infraestructura, AWS como proveedor de servicios en la nube, GitHub para el control de versiones, Packer para la creación de imágenes y Ansible para la automatización de la infraestructura. Como resultado, se logró desarrollar una infraestructura en la nube que se adapta a cualquier requerimiento. A través de esta investigación, se concluye que las infraestructuras definidas mediante código aceleran la creación de servicios, ya que toda la configuración y la información se encuentran alojadas en la nube, ofreciendo confiabilidad, seguridad y, sobre todo, costos más bajos en comparación con las infraestructuras tradicionales.

Para (Sánchez, 2018) la referencia de investigación menciona que las empresas de desarrollo de software están optando por virtualizar sus sistemas para reducir costos de hardware sin comprometer el rendimiento. También se busca reducir los tiempos de despliegue de grandes sistemas mediante el uso de tecnologías y herramientas que permitan la virtualización y el despliegue eficiente de infraestructuras basadas en software.

En particular, (Sánchez, 2018) se destaca el creciente peso de las infraestructuras basadas en microservicios en las decisiones de desarrollo debido a su capacidad para el despliegue

automatizado, escalable, rápido y robusto, así como la reducción de costos al virtualizar partes específicas del hardware. Esto permite el despliegue distribuido de lo que solía ser una sola máquina que requería grandes recursos.

Para la gestión de un clúster de microservicios en el trabajo de fin de máster, (Sánchez, 2018) utiliza la herramienta Kubernetes. Esta herramienta permite el despliegue, control, escalado y monitoreo del entorno de microservicios.

En el trabajo de (Rázuri, 2019), describe la implementación de un sistema de gestión de continuidad del negocio (SGCN) basado en la norma ISO 22301 en una entidad financiera. El objetivo es demostrar que este sistema permitirá a la organización estar preparada para hacer frente a situaciones adversas que puedan poner en riesgo la consecución de sus objetivos.

Se enfatiza (Rázuri, 2019) la importancia y la necesidad actual de contar con un SGCN, y se describen los objetivos de la investigación. En el capítulo II se proporcionan antecedentes sobre la continuidad del negocio y cómo ha evolucionado con el tiempo, y se presentan conceptos clave como análisis de impacto, gestión de riesgos y auditoría, que son necesarios para la implementación del sistema.

(Rázuri, 2019) Consiste en realizar un análisis previo a la implementación para identificar las brechas en la organización en relación con la continuidad del negocio, llevar a cabo la implementación propiamente dicha, que implica analizar el impacto, identificar los principales productos de la organización, los procesos que los respaldan, las amenazas que podrían afectar las operaciones y los planes para mitigar dichas amenazas. Además de ello, enfocarse en la mejora continua, mediante evaluaciones periódicas como auditorías, para identificar debilidades en el sistema y corregirlas oportunamente.

La referencia de investigación (Yunga, 2018) se enfoca en la implementación del provisionamiento automático de configuraciones en infraestructuras multivendedor utilizando Ansible. Para llevar a cabo esta implementación, se utilizó la herramienta de emulación GNS3 y su máquina virtual a través de VM Ware. Se seleccionaron plataformas de networking como CISCO, ARISTA y JUNIPER, basándose en una comparación de características (Benchmarking).

Durante la simulación, (Yunga, 2018) se emplearon diversas topologías con configuraciones de BGP, VLANS y MPLS en dispositivos de red como routers, switches y firewalls. El nodo central, utilizando Ansible, se encargó de aplicar todas las configuraciones a los dispositivos

de red mediante directorios, archivos, playbooks escritos en YAML y plantillas escritas en JINJA2. Cada uno de estos elementos desempeña un papel específico en la aplicación de tareas de configuración al nodo secundario.

Además, (Yunga, 2018) concluye que Ansible es una herramienta de automatización "sin agentes" que no utiliza bases de datos ni demonios externos, lo que evita vulnerabilidades en los equipos. A diferencia de otras herramientas que requieren agentes externos como Chef y Puppet, Ansible se considera más seguro. Para administrar la implementación de Ansible en diversas topologías de red con miles de dispositivos, se recomienda mantener una documentación estructurada y ordenada de los playbooks y plantillas. Sublime Text 3, un editor de texto y código fuente compatible con varios lenguajes de programación, se menciona como una herramienta útil para esta tarea.

La referencia de investigación (Huerlo, 2021) se enfoca en evaluar la factibilidad técnica de la herramienta Terraform para su adopción en el ciclo de vida del DevOps, específicamente en la creación y aprovisionamiento de instancias en la nube. El objetivo es mejorar la administración de la infraestructura, beneficiando a departamentos de TI, programadores independientes y empresas de diferentes tamaños.

Para llevar a cabo el estudio, (Huerlo, 2021) se utilizaron tres proveedores de servicios en la nube con el fin de observar posibles limitaciones al utilizar diferentes proveedores. Se empleó una metodología experimental que implicó probar la herramienta y ejecutar scripts para crear una infraestructura con recursos. Además, se utilizó una ficha de evaluación que medía tres características: usabilidad, eficiencia de desempeño y compatibilidad (integración).

En comparación con otros estudios realizados por otros autores, (Huerlo, 2021) coinciden con estudios que se centraban en la implementación de Terraform, mientras que en otros casos difieren debido al uso de diferentes herramientas o métricas que no eran adecuadas para Terraform. Sin embargo, uno de los logros más destacados fue descubrir que Terraform puede ser incluido en entornos novedosos como CI/CD (Integración Continua y Despliegue Continuo).

La referencia de investigación (Loo & Rojas, 2018) se enfoca en desarrollar un modelo de migración a la nube para los servidores de un centro de datos de pequeñas y medianas empresas (pymes). El objetivo principal es implementar este modelo basado en buenas prácticas y tecnologías de plataforma en la nube, considerando las necesidades específicas del

negocio. La propuesta se fundamenta en las recomendaciones de proveedores, la literatura y los marcos de trabajo Togaf y CCRA v4.

El proyecto (Loo & Rojas, 2018) consta de tres partes: entrada (input), solución (solution) y salida (output). La etapa de entrada implica recopilar información, identificar los principios, objetivos y impulsores del negocio, comprender los activos y la arquitectura de TI, y definir los requisitos para la migración. Con base en esta información, se evalúa la viabilidad de la migración y se analizan los proveedores de plataformas en la nube. Se determina si es conveniente llevar a cabo la migración y se seleccionan los componentes clave, como el repositorio en la nube, el servicio de importación de la plataforma, los scripts de comandos para invocar servicios, las instancias desplegadas en el entorno de la nube y las herramientas de administración y monitoreo de esas instancias.

La referencia de investigación, (Payá, 2021) se enfoca en desarrollar un modelo de migración a la nube para los servidores de un centro de datos de pequeñas y medianas empresas (pymes). El objetivo principal es implementar este modelo basado en buenas prácticas y tecnologías de plataforma en la nube, considerando las necesidades específicas del negocio. La propuesta se fundamenta en las recomendaciones de proveedores, la literatura y los marcos de trabajo Togaf y CCRA v4.

El proyecto, (Payá, 2021) consta de tres partes: entrada (input), solución (solution) y salida (output). La etapa de entrada implica recopilar información, identificar los principios, objetivos y impulsores del negocio, comprender los activos y la arquitectura de TI, y definir los requisitos para la migración. Con base en esta información, se evalúa la viabilidad de la migración y se analizan los proveedores de plataformas en la nube. Se determina si es conveniente llevar a cabo la migración y se seleccionan los componentes clave, como el repositorio en la nube, el servicio de importación de la plataforma, los scripts de comandos para invocar servicios, las instancias desplegadas en el entorno de la nube y las herramientas de administración y monitoreo de esas instancias.

La referencia de investigación (Chacón, 2022) describe la implementación de una arquitectura GitOps utilizando Git y Kubernetes, junto con la infraestructura como código (IaC), para lograr el aprovisionamiento automatizado de tecnologías y aplicaciones en una empresa. El objetivo es agilizar el proceso de puesta en producción, mejorar la resolución de problemas, la

recuperación ante contingencias, la eficiencia y reducir los errores humanos. La solución se implementó en tres proveedores de nube pública: Amazon, Azure y Google Cloud.

El enfoque de la metodología, (Chacón, 2022) consta de cuatro fases principales. En la primera fase se construyen los repositorios de control de versiones y se configuran los WebHooks. En la segunda fase se prepara la herramienta de integración y despliegue continuo. En la tercera fase se utiliza Terraform y Jenkins para implementar la infraestructura como código. Por último, en la cuarta fase se realiza el despliegue continuo de la aplicación Odoos mediante Jenkins.

Los resultados obtenidos (Chacón, 2022) demuestran que, a través del aprovisionamiento automatizado, se reducen los tiempos en las pruebas y entregas del producto final, así como la tasa de fallos en el despliegue e integración. Se asegura que los cambios puedan funcionar correctamente y se acelera el desarrollo de aplicaciones sin comprometer la seguridad. La implementación de las técnicas de GitOps y las buenas prácticas facilitaron los cambios en la infraestructura y la incorporación de nuevos miembros al equipo de manera automatizada.

La investigación, (Moreno, 2022) tiene como objetivo aplicar las buenas prácticas del desarrollo de software a la infraestructura como código (IaC). Se busca que el código de la infraestructura esté almacenado en un repositorio accesible por todo el equipo, que antes de realizar modificaciones en la infraestructura, pueda ser revisado por un compañero, se analice la calidad del código y se realicen pruebas para garantizar el comportamiento esperado de la infraestructura. Además, se busca automatizar el despliegue de nuevos cambios y la capacidad de realizar rollback a versiones anteriores de la infraestructura, aplicando buenas prácticas del desarrollo de software a la IaC en un entorno empresarial colaborativo.

Para la siguiente investigación, (Reyes, 2022) se aborda la necesidad de una metodología de implementación de tecnologías de automatización para mejorar los procesos dentro de un centro de datos. Con base en esta necesidad, se propone una metodología llamada ASA (Automation Systems Applied) que se desarrolla a partir de la consolidación de buenas prácticas de la industria, estándares internacionales, procesos sugeridos por los fabricantes y la experiencia del investigador.

Mediante la metodología ASA (Reyes, 2022), se analizan los indicadores clave del proceso y se evidencian mejoras significativas en cada uno de ellos. Estas mejoras sustentan cinco hipótesis

específicas que serán demostradas para validar la aceptación de la hipótesis general, especificando los porcentajes de mejora alcanzados. Al finalizar el estudio, se logra el objetivo de mejorar el proceso de automatización en las operaciones del centro de datos, lo cual justifica la relevancia de la investigación en el ámbito de las tecnologías de automatización.

Los resultados de este trabajo, (Reyes, 2022) proporcionan información valiosa sobre las herramientas y técnicas utilizadas en la automatización de procesos de infraestructura. Además, ofrecen una perspectiva útil sobre los desafíos comunes en este ámbito y las soluciones propuestas. En conjunto, esta investigación contribuye al avance y conocimiento en el campo de la automatización en centros de datos.

2.3. Conclusiones del estado del arte

Los trabajos vistos anteriormente abordan diferentes aspectos de la automatización de la infraestructura TI y proporcionan información valiosa sobre las herramientas, metodologías y mejores prácticas utilizadas en este campo. A continuación, se presenta un resumen de cada uno de ellos y se define la contribución al aporte propio.

Tabla 2. Aportes a la investigación.

Trabajo de Investigación	Aporte	Propuesta de Aporte Propio
Choez, 2022	Desarrollo de una infraestructura en la nube mediante código, reduciendo costos y acelerando la creación de servicios.	Proponer un procedimiento automatizado para la implementación de infraestructuras TI, considerando las mejores prácticas y herramientas disponibles.
Sánchez (2018)	Uso de virtualización y microservicios para reducir costos y acelerar el despliegue de infraestructuras basadas en software.	Diseñar un modelo de despliegue automatizado optimizando el uso de recursos y garantizando la reducción de costos.

Yunga (2018)	Implementación del provisionamiento automático de configuraciones en infraestructuras multivendor utilizando Ansible.	Investigar y desarrollar una solución para el proceso automatizado de la configuración de infraestructuras TI, integrando herramientas como Ansible, GIT y Terraform.
Huerlo (2021)	Evaluación de la factibilidad técnica de Terraform en el ciclo de vida del DevOps, mejorando la administración de la infraestructura.	Proponer una estrategia de implementación de Terraform en el proceso de CI/CD, integrando la automatización de la infraestructura con las pruebas y despliegue continuo.
Loo & Rojas (2018)	Desarrollo de un modelo de migración a la nube para pymes, basado en buenas prácticas y tecnologías de plataforma en la nube.	Diseñar un marco de trabajo basado en buenas prácticas, considerando la complejidad de las aplicaciones y la infraestructura existente.
Chacón (2022)	Implementación de una arquitectura GitOps para el aprovisionamiento automatizado de tecnologías y aplicaciones en una empresa.	Aplicación de metodología para la gestión de cambios y versionado de la infraestructura como código, aplicando principios de GitOps y CI/CD en entornos empresariales complejos.
Moreno (2022)	Aplicación de buenas prácticas del desarrollo de software a la infraestructura como código (IaC), permitiendo revisión, pruebas y despliegue automatizado.	Investigar y proponer técnicas de aseguramiento de calidad para el código de infraestructura, como la implementación de pruebas

		automatizadas y análisis estático del código.
Reyes (2022)	Desarrollo de una metodología (ASA) para mejorar la automatización en las operaciones de un centro de datos.	Diseñar un framework de automatización para la gestión integral de la infraestructura TI, considerando aspectos como la seguridad y la escalabilidad.

Fuente: Elaboración propia

La automatización de procesos de implementación de infraestructura TI para soportar servicios de misión crítica muestra la importancia y necesidad de esta tarea en el ámbito empresarial, permite mejorar la eficiencia y eficacia en la gestión de la infraestructura, minimizar errores humanos y reducir los tiempos de implementación, lo que se traduce en una mayor agilidad y capacidad de respuesta frente a los cambios del entorno. Aunque existen diversas herramientas y soluciones de automatización en el mercado, cada organización debe adaptarlas a sus necesidades específicas y contar con un plan de gestión de cambios adecuado para asegurar el éxito de la implementación. Por lo tanto, la identificación y propuesta de metodologías y buenas prácticas para la automatización de procesos sigue siendo un tema de interés en el campo de las tecnologías de la información. Además, la tendencia hacia la nube y la virtualización está impulsando la necesidad de automatizar los procesos de implementación y gestión de la infraestructura TI en este tipo de entornos.

3. Objetivos y metodología de trabajo

Los objetivos de investigación en este estudio se alinean con la definición de la problemática y la hipótesis que busca responder la pregunta de investigación planteada. Para lograr estos objetivos, es fundamental seguir una metodología de trabajo y el análisis de indicadores, los cuales serán detallados y explorados en profundidad a lo largo del estudio. Estos indicadores se consideran fundamentales para abordar el problema planteado y proporcionar respuestas a la pregunta de investigación.

3.1. Objetivo general

Desarrollar una solución que permita automatizar el proceso de implementación de la infraestructura TI necesaria para soportar los servicios de misión crítica de una organización, logrando reducir costos, tiempos, errores y complejidad innecesaria. Esto incluye la revisión y optimización de procesos, así como la selección y configuración de las herramientas necesarias.

3.2. Objetivos específicos

A continuación, se presentan objetivos específicos que forman parte del trabajo de automatización del proceso de implementación de infraestructura TI para soportar servicios de misión crítica:

- Analizar los procesos actuales de implementación de infraestructura TI en la organización para identificar oportunidades de mejora y optimización.
- Identificar los requerimientos de infraestructura TI necesarios para soportar servicios de misión crítica del negocio.
- Evaluar diferentes herramientas y tecnologías disponibles para la automatización del proceso de implementación de infraestructura TI y seleccionar las más adecuadas para la organización.
- Optimizar el proceso de implementación de infraestructura TI, definir roles y responsabilidades, construir flujos de trabajo y replantear procesos de validación y aprobación.
- Capacitar al personal de TI en el uso de la nueva solución de automatización y proporcionar documentación para su uso y mantenimiento.

- Analizar resultados obtenidos de las pruebas Pre y Post de los indicadores de costos, tiempo, complejidad y errores.

3.3. Metodología del trabajo

La metodología Kanban se presenta como una elección estratégica y efectiva para la gestión de proyectos en el presente estudio. Su enfoque visual y su capacidad para optimizar el flujo de trabajo ofrecen ventajas significativas en términos de transparencia y visibilidad de tareas, lo que permite identificar cuellos de botella y tomar decisiones fundamentadas. Al establecer límites de trabajo en progreso, Kanban también promueve una gestión más eficiente de la capacidad y la demanda, evitando la sobrecarga del equipo y mejorando la ejecución de tareas. Asimismo, su flexibilidad inherente permite una mayor adaptabilidad a los cambios y requisitos emergentes, favoreciendo una implementación ágil y exitosa de los proyectos. De esta manera, la metodología del trabajo se dividirá en los siguientes pasos:

3.3.1. Visualización del flujo de trabajo

En esta etapa, se establece un tablero Kanban que representará de manera visual el flujo de trabajo de implementación de infraestructura TI, este tablero propiamente dicho, nos permite visualizar el estado de las tareas y facilita la gestión del proceso.

Tabla 3. Visualización del flujo de trabajo

<i>Solicitud de implementación</i>	<i>En proceso</i>	<i>Pruebas</i>	<i>Desplegado</i>
<i>Tarjeta 1</i>	Tarjeta 3		
<i>Tarjeta 2</i>	Tarjeta 4		
	Tarjeta 5		

Fuente: Elaboración propia

Se han utilizado cuatro columnas para representar las etapas del flujo de trabajo: "Solicitud de implementación", "En proceso", "Pruebas" y "Desplegado". Cada columna representa el estado actual de las solicitudes de implementación. Las tarjetas numeradas representan las solicitudes específicas que se están procesando en cada etapa. Esto permite a todo el equipo involucrado en el proceso tener una visión global y actualizada del estado de las

implementaciones en cualquier momento. Además, el uso de tarjetas individuales para cada solicitud de implementación permite un seguimiento preciso de cada tarea y facilita la asignación de recursos y la planificación de plazos.

3.3.2. Definición de políticas de trabajo

Se establecen políticas claras que rigen el flujo de trabajo en cada etapa del proceso de implementación de infraestructura TI. Estas políticas son guías y criterios específicos que determinan cuándo y cómo se debe mover una tarjeta de una columna a otra en el tablero Kanban. Estas políticas son fundamentales para garantizar la consistencia y la calidad en el proceso de implementación.

A continuación, se define las políticas de trabajo para cada etapa del flujo de trabajo en el tablero:

Etapas: Solicitud de implementación

En esta etapa, se definen las políticas para el movimiento de tarjetas desde la columna "Solicitud de implementación" a la siguiente etapa.

Tabla 4. Solicitud de implementación

<i>Política</i>	<i>Descripción</i>
<i>Requerimientos completos</i>	La tarjeta debe tener todos los requerimientos necesarios para la implementación especificados de manera clara y detallada.
<i>Priorización adecuada</i>	Se debe asignar una prioridad a cada tarjeta para determinar el orden de implementación.
<i>Recursos asignados</i>	Debe haber recursos disponibles y asignados para trabajar en la implementación de la tarjeta.
<i>Aprobación</i>	La tarjeta debe ser revisada y aprobada por el responsable antes de avanzar a la siguiente etapa.

Fuente: Elaboración propia

Etapa: En proceso

En esta etapa, se definen las políticas para el movimiento de tarjetas desde la columna "En proceso" a la siguiente etapa.

Tabla 5. En proceso

<i>Política</i>	<i>Descripción</i>
<i>Tareas asignadas</i>	Cada tarjeta en esta etapa debe tener tareas específicas asignadas a los miembros del equipo.
<i>Avance actualizado</i>	El progreso de cada tarjeta en esta etapa debe ser actualizado regularmente en el tablero Kanban.
<i>Plazos respetados</i>	Las tareas asignadas deben cumplir con los plazos establecidos para mantener el flujo de trabajo.
<i>Revisión de avance</i>	Antes de pasar a la siguiente etapa, se debe realizar una revisión del avance y asegurar que todas las tareas estén completas.

Fuente: Elaboración propia

Etapa: Pruebas

En esta etapa, se definen las políticas para el movimiento de tarjetas desde la columna "Pruebas" a la siguiente etapa.

Tabla 5. Pruebas

<i>Política</i>	<i>Descripción</i>
<i>Tareas de prueba definidas</i>	Se deben definir las tareas específicas de prueba que deben ser completadas para cada tarjeta.
<i>Resultados documentados</i>	Los resultados de las pruebas deben ser documentados de manera clara y detallada.

<i>Corrección de errores</i>	Si se identifican errores durante las pruebas, deben ser corregidos antes de pasar a la siguiente etapa.
<i>Aprobación de pruebas</i>	Las pruebas deben ser aprobadas por el responsable o equipo encargado antes de avanzar a la siguiente etapa.

Fuente: Elaboración propia

Etapa: Desplegado

En esta etapa, se definen las políticas para el movimiento de tarjetas desde la columna "Desplegado" a la siguiente etapa.

Tabla 6. Desplegado

<i>Política</i>	<i>Descripción</i>
<i>Plan de despliegue establecido</i>	Se debe contar con un plan de despliegue establecido y documentado para cada tarjeta.
<i>Aprobación de despliegue</i>	El plan de despliegue debe ser aprobado antes de proceder con la implementación.
<i>Registro de detalles</i>	Se deben registrar los detalles del despliegue, como la fecha, la versión implementada y cualquier información relevante.
<i>Verificación de éxito</i>	Después del despliegue, se debe verificar que la implementación haya sido exitosa y el servicio esté en funcionamiento.

Fuente: Elaboración propia

Estas políticas de trabajo definidas para cada etapa del flujo de trabajo en el tablero Kanban son fundamentales para garantizar la consistencia y la calidad en el proceso de implementación de infraestructura TI. Establecen criterios claros y objetivos para el movimiento de tarjetas, como la finalización de pruebas o la aprobación de implementación, lo que contribuye a un proceso estructurado y controlado.

3.3.3. Gestión de la capacidad y la demanda

Se establece un límite de trabajo en progreso (WIP, por sus siglas en inglés) para controlar la cantidad de solicitudes de implementación que se pueden manejar simultáneamente en el proceso de implementación de infraestructura TI. Esto garantiza una capacidad adecuada para el equipo y evita la sobrecarga, lo que a su vez contribuye a una implementación eficiente y de calidad, a continuación, se presenta una tabla que muestra el límite de trabajo en progreso para cada etapa del flujo de trabajo:

Tabla 7. Gestión de la capacidad

<i>Etapa</i>	<i>Límite de trabajo en progreso (WIP)</i>
<i>Solicitud de implementación</i>	5 solicitudes
<i>En proceso</i>	3 solicitudes
<i>Pruebas</i>	2 solicitudes
<i>Desplegado</i>	1 solicitudes

Fuente: Elaboración propia

Establecer límites de trabajo en progreso ayuda a evitar la acumulación excesiva de tareas en cada etapa y garantiza que el equipo pueda manejar eficientemente las solicitudes de implementación. Estos límites pueden ajustarse según las capacidades y recursos disponibles del equipo, así como la complejidad de las implementaciones, además se debe mencionar que es importante monitorear y revisar regularmente el flujo de trabajo y los límites de WIP para adaptarlos a medida que se obtenga experiencia y se identifiquen áreas de mejora.

3.3.4. Automatización de tareas

Se identifican las tareas repetitivas y propensas a errores en el proceso de implementación de infraestructura TI, y se busca automatizarlas para agilizar el proceso y minimizar los errores humanos. Para lograr esto, se utilizan herramientas y scripts que permiten la ejecución automática de las tareas, brindando eficiencia y consistencia en la implementación. A continuación, se presenta una tabla que muestra las tareas comunes que pueden automatizarse en el proceso de implementación de infraestructura TI, junto con las herramientas o scripts que se pueden utilizar para su automatización:

Tabla 8. Automatización de tareas

Tarea	Herramienta/Script de automatización
<i>Solicitud de despliegue de VMs</i>	Sistema de gestión de solicitudes
<i>Despliegue de VMs</i>	Herramienta de automatización de infraestructura como Terraform y Ansible
<i>Configuración de backup de VMs</i>	Herramienta de backup
<i>Configuración de monitoreo de VMs</i>	Herramienta de monitoreo
<i>Validaciones post implementación del despliegue</i>	Scripts personalizados para realizar pruebas y verificaciones
<i>Documentación del nuevo activo de TI</i>	Script para generar evidencias de documentación

Fuente: Elaboración propia

La automatización de estas tareas permite reducir el tiempo y los recursos necesarios para llevar a cabo la implementación de infraestructura TI, ya que elimina la necesidad de realizarlas manualmente. Además, al utilizar herramientas y scripts, se minimizan los errores humanos, lo que garantiza una implementación más confiable y consistente. Por lo tanto, es importante destacar que la identificación de las tareas adecuadas para la automatización debe realizarse de manera cuidadosa, enfocándose en aquellas que son repetitivas, consumen tiempo y son propensas a errores.

Una vez que se ha implementado la automatización de tareas, es importante monitorear su funcionamiento y realizar actualizaciones y mejoras periódicas para mantener la eficiencia y la confiabilidad del proceso de implementación. Esto incluye la revisión y optimización de los scripts y herramientas utilizados, así como la incorporación de nuevas tareas en el flujo de automatización según sea necesario.

3.3.5. Seguimiento y mejora continua

Se monitorea constantemente el flujo de trabajo en el tablero Kanban para identificar cuellos de botella, tiempos de espera o cualquier otro problema en el proceso. Se realizan reuniones

periódicas para analizar los datos y buscar formas de mejorar la eficiencia y la calidad de la implementación.

Tabla 9. Seguimiento

Aspecto	Descripción
<i>Monitoreo del flujo de trabajo</i>	Se realiza un seguimiento constante del tablero Kanban para identificar posibles cuellos de botella, tiempos de espera prolongados o cualquier otro problema en el proceso de implementación. Esto permite visualizar el estado actual de las solicitudes de implementación y tomar medidas para mantener un flujo de trabajo fluido.
<i>Reuniones periódicas de revisión</i>	Se llevan a cabo reuniones periódicas, como las reuniones de revisión de Kanban, donde se analizan los datos recopilados del tablero Kanban y se discuten los desafíos y oportunidades de mejora. Estas reuniones involucran a los miembros del equipo de implementación, así como a los responsables de la gestión del proyecto, con el objetivo de identificar áreas de mejora y tomar acciones correctivas o preventivas.
<i>Análisis de datos y métricas</i>	Se recopilan y analizan datos sobre el proceso de implementación, como los tiempos de ciclo, los tiempos de espera, las tasas de éxito, los errores o cualquier otra métrica relevante. Estos datos se utilizan para identificar patrones, tendencias y áreas problemáticas que requieren atención. La recopilación y el análisis de datos ayudan a fundamentar las decisiones de mejora y evaluar el impacto de las acciones tomadas.
<i>Acciones de mejora</i>	Con base en el análisis de datos y las discusiones durante las reuniones de revisión, se proponen y ejecutan acciones de mejora. Estas acciones pueden incluir la optimización del flujo de trabajo, la eliminación de cuellos de botella, la reducción de tiempos de espera,

la implementación de mejores prácticas o la actualización de políticas y procedimientos. Se busca constantemente mejorar la eficiencia y la calidad de la implementación.

Fuente: Elaboración propia

El seguimiento y la mejora continua son componentes esenciales de la metodología Kanban. A través del monitoreo constante, las reuniones periódicas, el análisis de datos y la implementación de acciones de mejora, se busca optimizar el proceso de implementación de infraestructura TI para garantizar una mayor eficiencia, reducir los tiempos de espera y mejorar la calidad de los servicios de misión crítica.

3.3.6. Adaptación a cambios y requisitos emergentes

Permite una adaptación flexible a los cambios en los requisitos o prioridades. Si surge una nueva solicitud o se modifica una existente, se puede agregar al tablero Kanban y priorizar en función de las políticas establecidas. Esto garantiza la capacidad de respuesta a los requisitos emergentes y optimiza el proceso de implementación."

Tabla 10. Adaptación a cambios

<i>Aspecto</i>	<i>Descripción</i>
<i>Recepción de cambios y nuevas solicitudes</i>	Recibir cambios y nuevas solicitudes de implementación de manera ágil. Cuando surge un cambio en los requisitos o se presenta una nueva solicitud, se agrega al tablero Kanban como una nueva tarjeta. Esto garantiza la visibilidad y seguimiento de los cambios emergentes en el flujo de trabajo.
<i>Priorización basada en políticas</i>	Estas políticas establecen los criterios y reglas para determinar la prioridad de una tarjeta en relación con otras. Se considera factores como la criticidad del servicio, la fecha límite, los recursos disponibles, entre otros. Esto permite asignar recursos y esfuerzos de manera adecuada en función de las necesidades emergentes.

<i>Reasignación de tareas y recursos</i>	cuando surge un cambio o una nueva solicitud prioritaria, se puede reasignar tareas y recursos en consecuencia. Esto implica ajustar el trabajo en progreso (WIP) y redistribuir las responsabilidades para acomodar los cambios. El equipo puede colaborar para garantizar que los recursos estén adecuadamente asignados a las nuevas solicitudes y que los plazos se cumplan.
<i>Adopción de enfoque incremental</i>	En lugar de intentar implementar todos los cambios a la vez, se priorizan y se implementan de manera secuencial. Esto evita la sobrecarga y permite una adaptación gradual y controlada a medida que surgen nuevas solicitudes o se modifican los requisitos existentes.
<i>Mejora continua en función de los cambios</i>	Los cambios y requisitos emergentes ofrecen oportunidades de mejora en el proceso de implementación. A medida que se enfrentan nuevos desafíos, se pueden identificar áreas de mejora y aplicar acciones correctivas o preventivas. El equipo puede analizar cómo los cambios afectan el flujo de trabajo y buscar formas de optimizarlo para futuras implementaciones.

Fuente: Elaboración propia

Al permitir la recepción ágil de cambios, la priorización basada en políticas, la reasignación de tareas y recursos, el enfoque incremental y la mejora continua, Kanban garantiza una mayor capacidad de respuesta y flexibilidad en el proceso de implementación de infraestructura TI para soportar servicios de misión crítica.

4. Desarrollo específico de la contribución

4.1. Análisis del estado actual

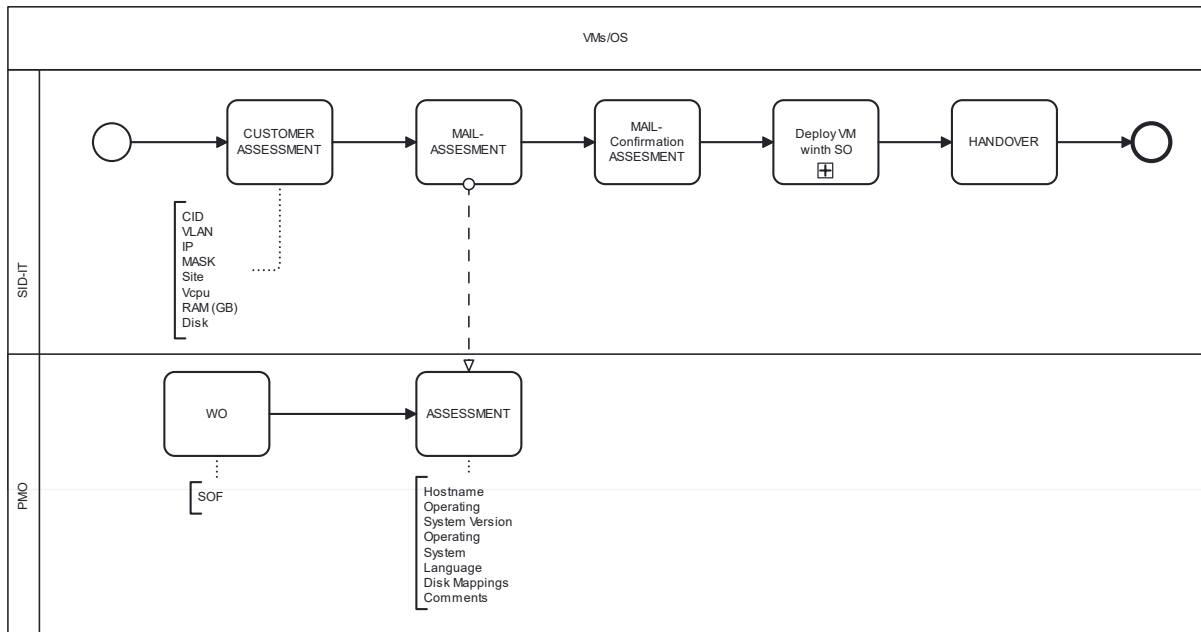
Se llevará a cabo una exhaustiva revisión del proceso de implementación de infraestructura TI actualmente en uso, con el propósito de detectar y analizar a fondo los problemas y desafíos más relevantes que se enfrenta. El objetivo primordial de esta revisión es obtener una comprensión completa de los obstáculos y deficiencias presentes en el proceso actual, para así poder abordarlos de manera efectiva y buscar soluciones que optimicen la implementación de infraestructura TI.

El proceso de implementación de una máquina virtual (VM) a partir de la recepción de la documentación de requerimientos se puede describir utilizando la metodología de gestión de procesos de negocio (BPM, por sus siglas en inglés). A continuación, se detalla el proceso paso a paso:

4.1.1. Documentación de requerimientos:

En esta etapa, se recibe la documentación que describe los requerimientos específicos para la implementación de la VM. Esto puede incluir detalles como el sistema operativo, la capacidad de almacenamiento, los recursos de CPU y memoria necesarios, entre otros reflejados en un documento denominado "assessment". Aquí se lleva a cabo un análisis exhaustivo de los requerimientos recibidos. Esto implica revisar y comprender los detalles proporcionados, identificar posibles inconsistencias o brechas, y realizar consultas adicionales si es necesario, con el objetivo de obtener una comprensión clara de los requisitos.

Figura 4. Proceso solicitud de despliegue de VMs



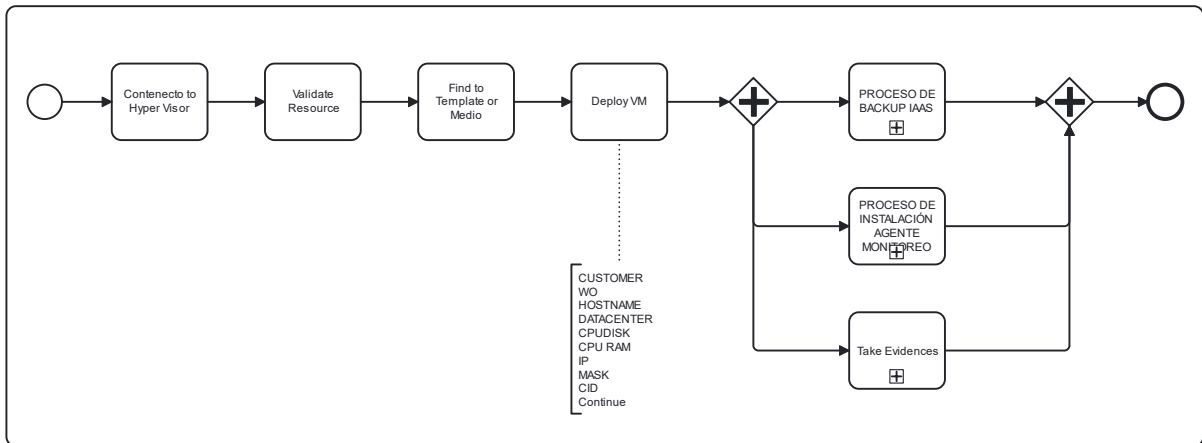
Fuente: Elaboración propia

4.1.2. Diseño de la configuración de la VM:

En esta fase, se elabora un diseño detallado de la configuración de la máquina virtual. Esto implica determinar la cantidad de recursos necesarios, seleccionar el sistema operativo adecuado, establecer las políticas de seguridad, definir la conectividad de red y realizar cualquier otra configuración relevante. Una vez que se ha completado el diseño de la configuración de la VM, se procede a obtener la aprobación de los responsables o partes interesadas involucradas en el proceso. Esto garantiza que el diseño cumple con los requisitos establecidos y está alineado con las políticas y estándares de la organización. Una vez aprobado el diseño, se inicia la implementación de la máquina virtual. Esto implica configurar los recursos de hardware y software necesarios, instalar el sistema operativo, establecer la conectividad de red, aplicar las políticas de seguridad y llevar a cabo cualquier otra acción requerida para poner en funcionamiento la VM.

Además del proceso de implementación de una máquina virtual, es importante considerar los procesos de implementación de agentes de monitoreo y backup para asegurar la supervisión y la protección adecuada de los sistemas y datos

Figura 5. Proceso de despliegue de VMs

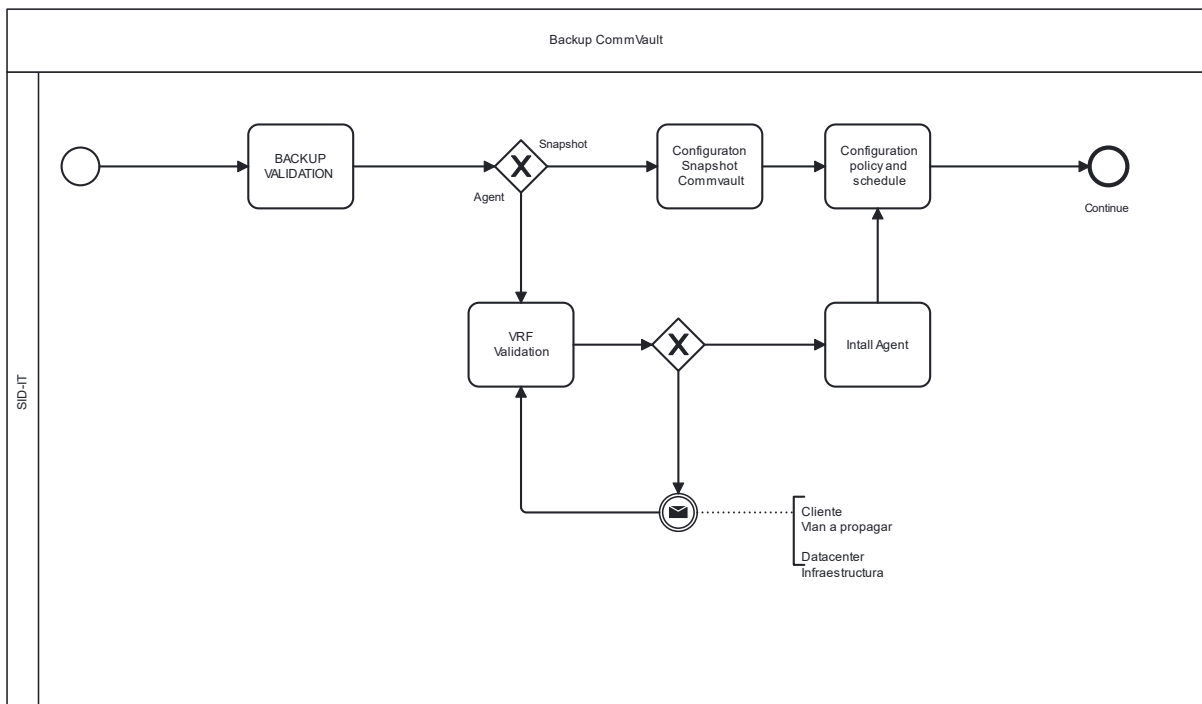


Fuente: Elaboración propia

4.1.3. Configuración de Backup

Considerando los sistemas y datos críticos que deben ser respaldados. Se definen criterios como la frecuencia de los backups, la retención de datos, los tiempos de recuperación y los mecanismos de almacenamiento, de esta manera se establecen políticas de backup que determinan cómo, cuándo y qué datos se respaldarán. Esto incluye la definición de los horarios de backup, los destinos de almacenamiento, los tipos de backup (completo, incremental, diferencial), y las exclusiones o inclusiones específicas de archivos o directorios.

Figura 6. Proceso de configuración de backup de VMs

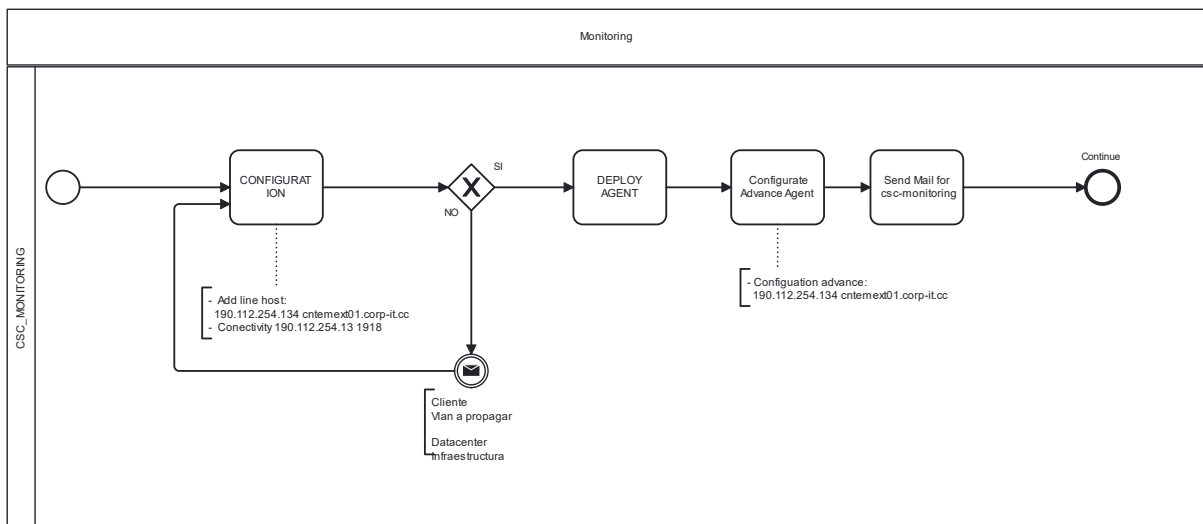


Fuente: Elaboración propia

4.1.4. Configuración de monitoreo

Así mismo se evalúan los requisitos específicos de monitoreo de la infraestructura y los sistemas. Se identifican los parámetros clave a monitorear, como el rendimiento del sistema, la utilización de recursos, las métricas de red, los eventos críticos, entre otros y se procede a instalar los agentes de monitoreo en los sistemas y dispositivos que se desean supervisar, estos ayudan a identificar y notificar proactivamente cualquier evento o comportamiento anormal que pueda indicar problemas o fallas en los sistemas, así mismo se definen los destinatarios de las alertas y se establece un plan de acción para abordarlas.

Figura 7. Proceso configuración de monitoreo de VMs

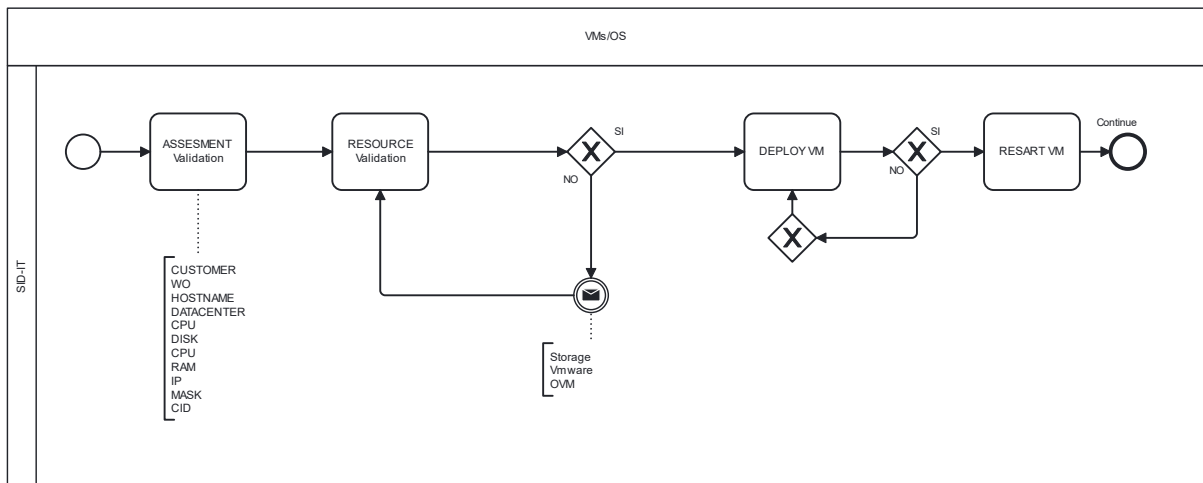


Fuente: Elaboración propia

4.1.5. Validaciones post implementación

Una vez implementada la VM, se llevan a cabo pruebas exhaustivas para verificar su correcto funcionamiento. Esto puede incluir pruebas de conectividad de red, pruebas de carga, pruebas de seguridad y cualquier otra prueba relevante para garantizar que la VM cumple con los requisitos y está preparada para su uso en producción. Después de que las pruebas y verificaciones hayan sido exitosas, se solicita la aprobación final para poner la VM en producción. Una vez obtenida la aprobación, la máquina virtual se considera lista para su uso y se integra en el entorno de producción.

Figura 8. Proceso de validaciones post implementación del despliegue

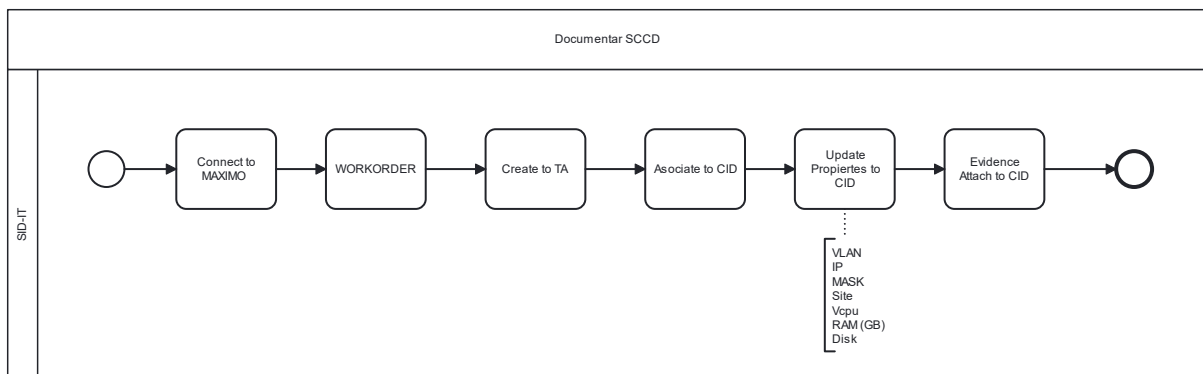


Fuente: Elaboración propia

4.1.6. Documentación del activo de TI

A lo largo de todo el proceso, se debe mantener una documentación completa y actualizada de cada etapa. Esto incluye registros de los requerimientos recibidos, el diseño de la configuración de la VM, los resultados de las pruebas y cualquier otro detalle relevante.

Figura 9. Proceso documentación del nuevo activo de TI



Fuente: Elaboración propia

Es importante tener en cuenta que los procesos descritos pueden variar según las políticas y prácticas específicas de cada organización. Sin embargo, esta descripción proporciona una visión general de las etapas principales involucradas en la implementación de una máquina virtual a partir de la recepción de la documentación de requerimientos, utilizando la metodología de gestión de procesos de negocio (BPM).

4.2. Análisis y selección de herramientas

Una vez realizado el análisis del estado actual, se analizarán las herramientas disponibles en el mercado para la automatización de la implementación de infraestructura TI y se seleccionará la más adecuada en función de las necesidades de la organización. Para ello, es importante considerar los siguientes aspectos:

4.2.1. Requisitos específicos

Se consideran aspectos como la capacidad de gestión y configuración de infraestructuras TI, la capacidad de despliegue automatizado, la integración con otras herramientas y sistemas, y la capacidad de adaptación a entornos de misión crítica.

4.2.2. Compatibilidad y sinergia

Se evalúa la compatibilidad y la sinergia entre las herramientas disponibles, también se busca herramientas que sean compatibles y que se complementen entre sí para lograr una integración fluida en el ciclo DevOps. Por ejemplo, Git se utiliza para gestionar el control de versiones del código, Ansible para la automatización de la configuración, Terraform para la implementación de infraestructuras y VMware vSphere para la virtualización y administración de recursos.

4.2.3. Capacidad de escalabilidad y seguridad

Es importante tener en cuenta la capacidad de las herramientas para escalar y adaptarse a entornos de infraestructura de gran tamaño. Además, la seguridad es un factor crítico en los entornos de misión crítica, por lo que es fundamental evaluar las características de seguridad que ofrecen las herramientas, como el cifrado de datos, la autenticación y la gestión de accesos.

4.2.4. Experiencia y comunidad de usuarios

Se investiga la reputación y la experiencia de las herramientas en la comunidad de usuarios. Revisa casos de estudio, testimonios y opiniones de expertos para tener una idea de cómo las herramientas han sido utilizadas en entornos similares.

4.2.5. Capacidad de personalización y extensibilidad

Considerar si las herramientas permiten la personalización y extensión según las necesidades específicas. Esto es especialmente importante para adaptarse a los requisitos únicos de la

infraestructura TI y servicios de misión crítica. es recomendable realizar pruebas y prototipos para evaluar su funcionalidad y eficacia, esto permitirá tomar una decisión informada y elegir las herramientas que mejor se ajusten a los objetivos y requisitos.

A continuación, se muestra un cuadro comparativo del potencial de las herramientas de automatización de la infraestructura de TI para cada fase del ciclo DevOps:

Tabla 11. Cuadro comparativo del potencial de las herramientas.

Ciclo DevOps	Herramienta	Ventajas	Limitaciones
Planificación	Jira	Facilita la colaboración entre los miembros del equipo. Permite una mejor planificación del trabajo.	Puede requerir tiempo y esfuerzo para configurar y personalizar según las necesidades específicas del proyecto.
Desarrollo	Git	Facilita la gestión de versiones y la colaboración en el desarrollo de software.	Puede haber una curva de aprendizaje inicial para aquellos que no están familiarizados con los conceptos de control de versiones
Construcción	Jenkins	Amplia comunidad de usuarios y soporte. Permite la configuración flexible de pipelines de construcción y prueba.	Puede requerir una curva de aprendizaje inicial. La configuración puede ser compleja para casos de uso avanzados
Provisionamiento	Ansible	Configuración sencilla y fácil de entender con su enfoque declarativo. Puede ser utilizado para automatizar la configuración	No es adecuado para casos de uso que requieren una configuración altamente personalizada.

		de una amplia gama de sistemas y aplicaciones	
Despliegue	Terraform	Permite una configuración y gestión eficiente de la infraestructura a través de archivos de configuración. Puede ser utilizado en múltiples proveedores de nube y entornos locales.	Puede haber una curva de aprendizaje inicial para aquellos que no están familiarizados con el enfoque de infraestructura como código.
Testing	Bash scripting PowerShell Python	Amplia variedad de bibliotecas y marcos de prueba disponibles. Fácil integración con otras herramientas y sistemas.	Requiere conocimientos de programación para crear y mantener las pruebas automatizadas.
Entrega	Packer	Permite la creación de imágenes consistentes y reproducibles. Integración con diferentes herramientas de infraestructura.	Requiere conocimientos de configuración y scripting para crear y mantener las configuraciones de imagen.
Despliegue	vSphere	Amplia funcionalidad y capacidad de gestión de recursos virtuales. Ampliamente utilizado en entornos empresariales.	Requiere infraestructura física y licencias de vSphere. Puede requerir conocimientos específicos de la plataforma.

Fuente: Elaboración propia

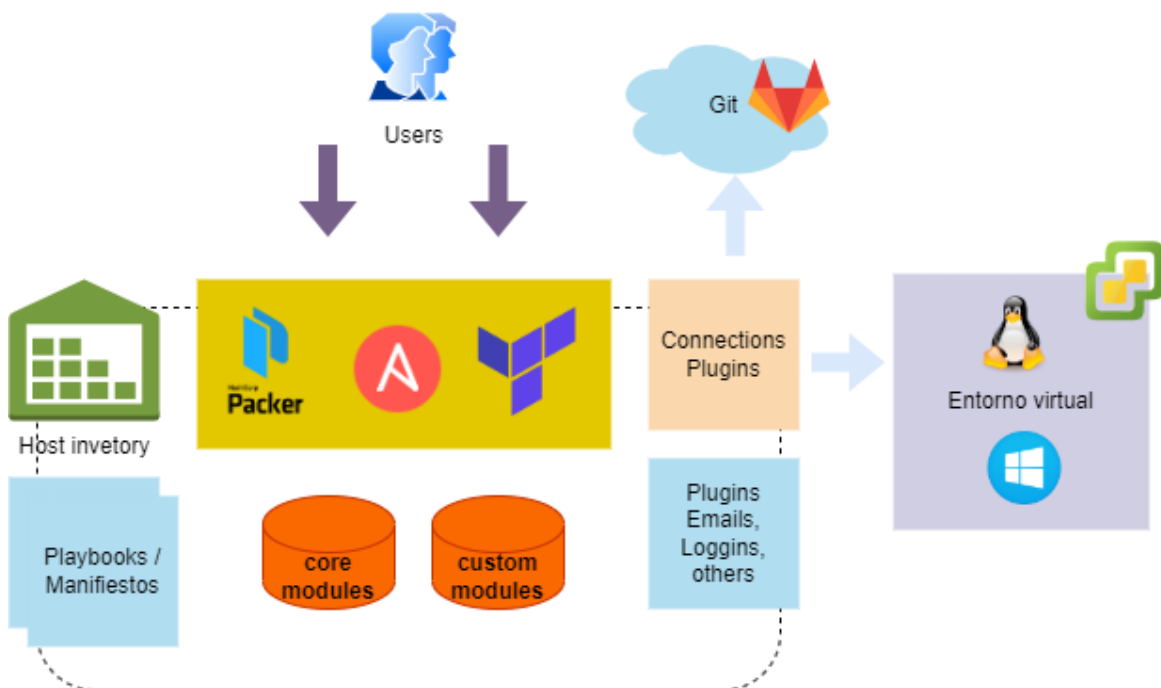
4.3. Diseño e implementación de la solución

Tomando en consideración los requisitos específicos de la organización, las metodologías aplicables y las mejores prácticas de la industria, se diseñó un diagrama de flujo el cual representa de manera visual y secuencial los pasos involucrados en el proceso de automatización. Este a su vez, nos proporciona una representación gráfica, clara y concisa las actividades, decisiones y relaciones entre diferentes elementos.

4.3.1. Implementación e integración de componentes

Para iniciar el proceso de implementación de infraestructura TI en el entorno de GitLab y VSphere, se dará paso a la creación de los manifiestos mediante la herramienta Ansible (ver anexo C). Estos "playbooks" constituyen un conjunto de instrucciones esenciales que permiten llevar a cabo diversas llamadas y ejecutar el código previamente escrito en Packer y Terraform. La sinergia entre estas tecnologías posibilita un despliegue efectivo y preciso de los requisitos específicos de cada área cliente. Es importante destacar que estos manifiestos serán confeccionados por equipos especializados y almacenados de manera organizada en un repositorio de GitLab (ver anexo B). De esta forma, se garantiza la centralización de las configuraciones y el control sobre las diferentes versiones de los manifiestos.

Figura 10. Diagrama de entorno e integración



Fuente: Elaboración propia

La etapa de creación de los manifiestos con Ansible representa el primer paso crucial en el proceso de automatización. Al aprovechar la flexibilidad y capacidad declarativa de Ansible, se logra una descripción clara y concisa de la infraestructura deseada. Estos playbooks se convierten en la base para la ejecución y despliegue de la infraestructura en el entorno de VSphere.

A continuación, el proceso avanza a la utilización de Packer y Terraform, donde se lleva a cabo la creación de imágenes de máquinas virtuales y el provisionamiento automático de recursos en el entorno de VSphere. Packer se encarga de crear las imágenes de manera reproducible, garantizando su consistencia y portabilidad entre diferentes plataformas. Por otro lado, Terraform, al operar como Infraestructura como Código (IaC), simplifica y agiliza el despliegue y la gestión de recursos en la nube, permitiendo una mayor flexibilidad y escalabilidad.

La sinergia entre Ansible, Packer y Terraform es esencial para asegurar una implementación automatizada y coherente en el entorno de GitLab y VSphere. Estas herramientas trabajan de manera conjunta, permitiendo que el proceso de implementación sea más eficiente y fiable. El resultado final es una infraestructura TI que se adapta a las necesidades específicas de cada área cliente, impulsando la eficiencia, la consistencia y el control en los flujos de trabajo.

La creación de manifiestos con Ansible, el uso de Packer para la generación de imágenes y el aprovisionamiento automatizado con Terraform representan el cimiento de un proceso de implementación automatizado, permitiendo a las organizaciones mejorar sus rutinas tradicionales y transformarlas en prácticas eficientes y efectivas. Esta automatización simplifica y agiliza la creación y configuración de la infraestructura requerida, respaldando así los servicios de misión crítica y brindando un mayor control sobre todo el proceso. Con esta metodología de implementación, se garantiza una mayor consistencia, confiabilidad y eficacia en la gestión de la infraestructura TI.

4.3.2. Gestión de código

La gestión y actualización de los manifiestos constituye una fase crucial en el proceso de automatización de la infraestructura TI. Con el objetivo de mantener un entorno controlado y organizado, solo el integrador y el responsable del código tendrán la autoridad para revisar y realizar modificaciones en los manifiestos. Esta medida garantiza la integridad de las configuraciones y evita posibles conflictos o errores causados por múltiples intervenciones. Si

durante el proceso de revisión se identifican errores o mejoras potenciales, se comunicarán de manera efectiva al integrador para que proceda con las correcciones necesarias antes de reintegrar el código actualizado en el repositorio de GitLab. Este enfoque de gestión colaborativa asegura que el contenido de los manifiestos sea confiable, actualizado y bien documentado, promoviendo así la eficiencia y la calidad en el proceso de implementación.

Una vez que los manifiestos han sido revisados y se han realizado las correcciones pertinentes, se procederá con la ejecución de las tareas de despliegue. Esta etapa del proceso implica una evaluación exhaustiva de los resultados obtenidos a partir de la ejecución automatizada. Se llevará a cabo una medición detallada de los tiempos de ejecución para evaluar la eficiencia y rapidez con la que se implementa la infraestructura TI. Además, se realizará una minuciosa verificación de posibles errores o inconsistencias en el despliegue. La finalidad de esta evaluación es asegurar que los resultados obtenidos sean coherentes con las expectativas y requisitos definidos previamente.

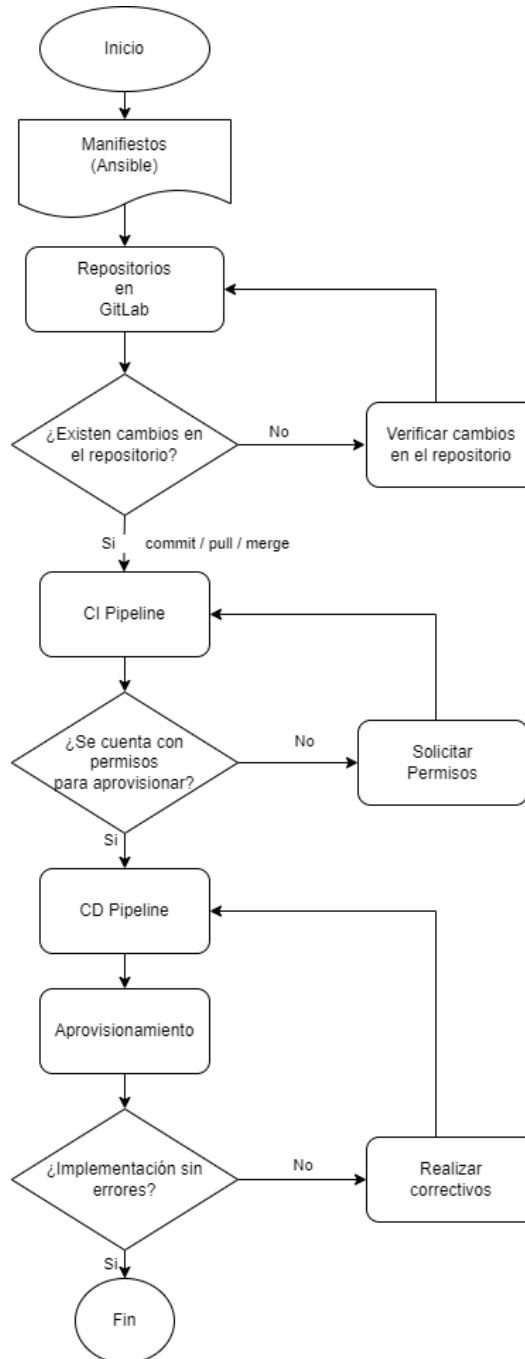
En el caso de que los resultados obtenidos durante la ejecución no alcancen los estándares establecidos, se llevará a cabo una revisión adicional del código y de los manifiestos implicados en el proceso. Esta revisión se realizará con el objetivo de identificar y corregir cualquier posible error o desviación que haya surgido. Una vez que se hayan realizado las actualizaciones necesarias en los manifiestos y en el repositorio de GitLab, se procederá a volver a ejecutar el proceso de despliegue. Este enfoque iterativo y de mejora continua asegura que los despliegues se realicen de manera efectiva y consistente, proporcionando una infraestructura TI confiable y ajustada a las necesidades de cada área cliente.

La aplicación de la metodología propuesta garantiza una implementación automatizada de la infraestructura TI para soportar servicios de misión crítica, lo que se traduce en beneficios significativos para las organizaciones. La automatización de los procesos manuales permite agilizar y optimizar la creación y configuración de la infraestructura necesaria, simplificando tareas como la provisión de servidores, la instalación de software y la configuración de redes. Con esto, se logra una mayor consistencia y confiabilidad en los flujos de trabajo, reduciendo el riesgo de errores humanos y minimizando los tiempos de implementación.

Asimismo, la integración de Ansible, Packer y Terraform posibilita una descripción clara y concisa de la infraestructura requerida, mejorando la capacidad de respuesta y la escalabilidad de los servicios de misión crítica. Al automatizar y estandarizar estos procesos, se facilita la

replicabilidad de la infraestructura en diferentes entornos, lo que resulta fundamental para respaldar los servicios de misión crítica en un entorno empresarial dinámico y en constante evolución.

Figura 11. Diagrama de flujo del proceso de implementación



Fuente: Elaboración propia

4.3.3. Despliegue y pruebas

Una vez que los manifiestos han sido creados y refinados en detalle utilizando la herramienta Ansible, y han sido almacenados de manera organizada en el repositorio de GitLab, se dará paso a la etapa de ejecución de las tareas de despliegue. Esta fase es de vital importancia, ya que representa el momento en el cual las configuraciones definidas en los manifiestos se aplicarán efectivamente para implementar la infraestructura TI requerida.

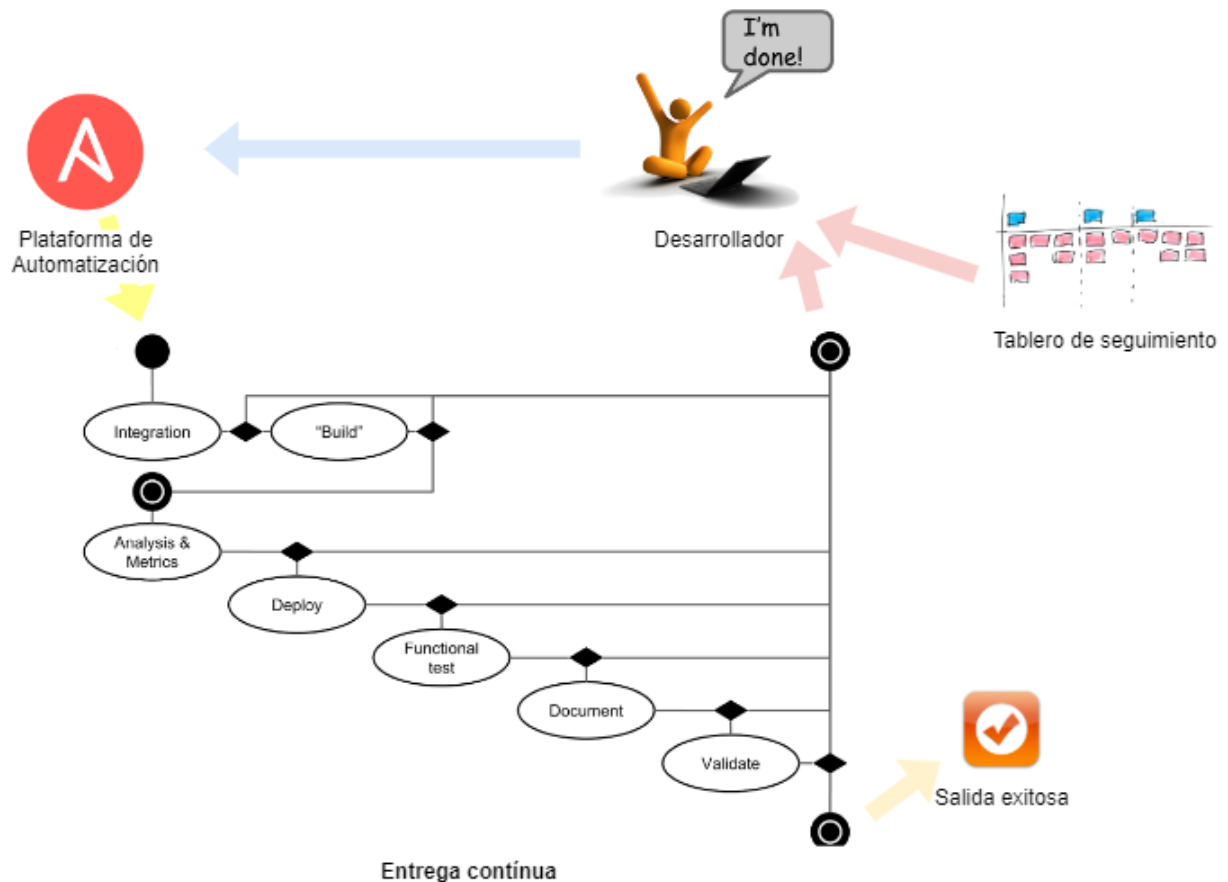
Durante la ejecución de las tareas de despliegue, se llevará a cabo una evaluación exhaustiva de los resultados obtenidos. Esta evaluación comprenderá diversos aspectos críticos, incluyendo la medición de los tiempos de ejecución de cada tarea, el rendimiento de las operaciones y la verificación detallada de posibles errores o inconsistencias en el proceso de despliegue.

Para medir los tiempos de ejecución, se registrarán meticulosamente los intervalos de tiempo desde el inicio hasta la finalización de cada tarea. Esta información permitirá evaluar la eficiencia del proceso de automatización, identificando posibles cuellos de botella o áreas de mejora que puedan afectar la velocidad y agilidad de la implementación.

Asimismo, se realizará una verificación detallada de posibles errores en la ejecución de las tareas. Se analizarán cuidadosamente los registros y reportes generados durante el despliegue, con el fin de identificar cualquier error o problema que haya surgido. En caso de que se detecten inconvenientes, se procederá a realizar una revisión adicional del código y de los manifiestos involucrados en el proceso.

La revisión adicional del código permitirá identificar posibles errores de sintaxis o lógica que puedan estar afectando el funcionamiento adecuado de las tareas de despliegue. De igual manera, se verificará la integridad de los manifiestos y su coherencia con los requisitos y especificaciones establecidos previamente.

Figura 12. Diagrama de despliegue y entrega continua



Fuente: Elaboración propia

En caso de que se detecten errores o áreas de mejora, se procederá a realizar las correcciones necesarias en el código y se actualizarán los repositorios correspondientes en GitLab. Estas actualizaciones asegurarán que los manifiestos estén en su versión más actual y que reflejen las modificaciones realizadas para corregir cualquier inconveniente.

Una vez realizadas las correcciones, se volverá a ejecutar el proceso de despliegue para comprobar que las mejoras implementadas han tenido el efecto deseado y que los resultados obtenidos cumplen con las expectativas establecidas previamente. Este enfoque iterativo y de mejora continua garantiza que los despliegues se realicen de manera efectiva y que la infraestructura TI se ajuste de manera óptima a las necesidades de cada área cliente.

4.3.4. Documentación y mantenimiento

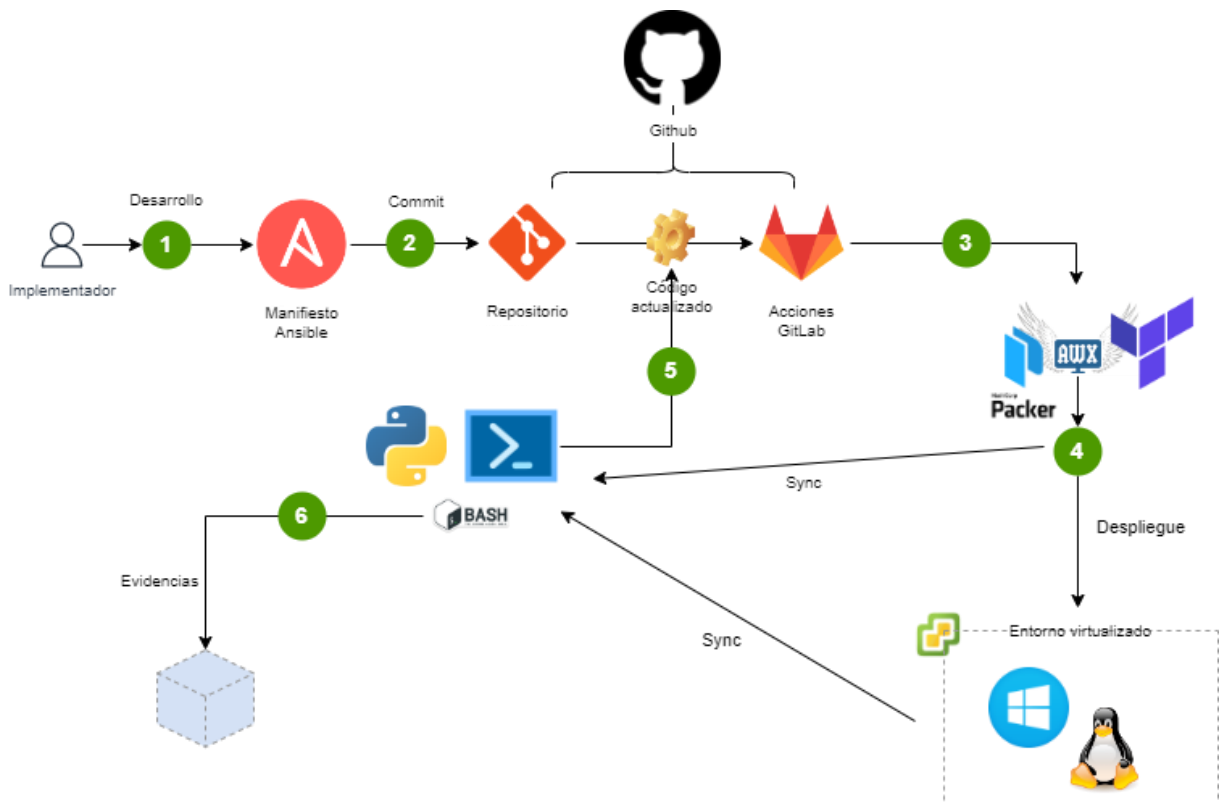
La fase final del proceso marca un hito significativo en el proyecto. Una vez que se han alcanzado los resultados deseados y se ha cumplido con todos los requisitos de implementación, se considerará que la implementación ha sido completada exitosamente.

Esta etapa de cierre es de vital importancia, ya que implica asegurar que todos los cambios y mejoras se hayan aplicado correctamente y que el nuevo entorno de infraestructura TI esté listo para respaldar de manera eficiente y confiable los servicios de misión crítica de la organización.

En esta fase final, se llevarán a cabo las siguientes actividades clave:

- **Documentación del proceso:** Es fundamental documentar todo el proceso realizado durante la automatización de la implementación de infraestructura TI. Se registrarán los pasos específicos, las decisiones tomadas y cualquier detalle relevante que pueda ser útil para futuras referencias y mantenimiento. La documentación proporcionará un valioso recurso para comprender y replicar el proceso de implementación en el futuro, asegurando la consistencia y el correcto seguimiento de las mejores prácticas establecidas.
- **Validación de resultados:** Se realizará una revisión exhaustiva de los resultados obtenidos durante todo el proceso de automatización. Se verificará que todas las tareas de despliegue se hayan ejecutado de manera exitosa y que los resultados se alineen con los objetivos y requisitos establecidos previamente. Cualquier desviación o inconsistencia será revisada y corregida antes de dar por finalizada la implementación.
- **Evaluación de la efectividad:** Se analizará la eficacia y eficiencia del proceso de automatización implementado. Se compararán los indicadores clave antes y después de la automatización para medir el impacto positivo que ha tenido en la infraestructura TI y en los servicios de misión crítica. Esta evaluación permitirá validar la contribución de la automatización en la mejora de la capacidad de respuesta, la fiabilidad y la calidad de los servicios.
- **Capacitación y transferencia de conocimiento:** Se brindará capacitación al equipo encargado de la operación y mantenimiento de la infraestructura TI automatizada. La transferencia de conocimiento garantizará que el equipo esté plenamente capacitado para gestionar y mantener la infraestructura de manera adecuada y aprovechar al máximo las capacidades de la automatización.
- **Cierre:** Con la implementación exitosa y la documentación completa, se procederá al cierre formal del proyecto de automatización. Se revisarán y documentarán los logros alcanzados, los desafíos superados y las lecciones aprendidas durante todo el proceso.

Figura 13. Diagrama del proceso de implementación



Fuente: Elaboración propia

Como se ha descrito anteriormente, el proceso de automatización de la implementación de infraestructura de TI sigue un enfoque estructurado y eficiente como se describió en el diagrama de flujo (Figura 11) y se plasma en el contexto real (Figura 13), haciendo uso de orquestadores como AWX (anexo A), y validaciones exhaustivas para garantizar su correcta configuración con el uso de scripts escritos en Python, Bash y PowerShell (anexos D y E), según sea necesario.

A continuación, se presenta un resumen de los pasos del proceso de automatización de la implementación de infraestructura TI:

Tabla 12. Cuadro comparativo del potencial de las herramientas.

Paso	Descripción
1	Creación de manifiestos con Ansible: Desarrollar "playbooks" que contengan las instrucciones para realizar llamadas y ejecutar código en Packer y Terraform. Estos manifiestos se redactarán por equipos especializados.
2	Almacenamiento en el repositorio de GitLab: Los manifiestos creados se guardarán y mantendrán de manera organizada en un repositorio de GitLab. Solo el integrador y responsable del código podrán revisar y realizar modificaciones
3	Ejecución de las tareas de despliegue: Se procederá a ejecutar las tareas definidas en los manifiestos para implementar la infraestructura TI requerida.
4	Evaluación de resultados: Se realizará una evaluación exhaustiva de los resultados obtenidos, incluyendo la medición de tiempos de ejecución y la verificación de posibles errores.
5	Revisión y actualización: En caso de que los resultados no cumplan con las expectativas, se realizará una revisión adicional del código y se actualizarán los repositorios correspondientes antes de volver a ejecutar el proceso de despliegue.
6	Documentación del proceso: Una vez que los despliegues sean exitosos y los resultados cumplan con los requisitos establecidos, se documentará el proceso de automatización para futuras referencias y mejoras.

Fuente: Elaboración propia

4.4. Evaluación y difusión de resultados

Se evaluará la solución implementada, comparando los resultados obtenidos con los objetivos específicos definidos en la metodología y analizando su impacto en la misión crítica del negocio, así mismo se documentará todo el proceso de implementación y evaluación de la solución, para que pueda ser utilizado por otros profesionales del sector. Además, se difundirán los resultados obtenidos a través de publicaciones y presentaciones en congresos y eventos relevantes.

Para la realización de estos pasos, se utilizarán diversas herramientas y metodologías, como la metodología Agile para el diseño e implementación de la solución y herramientas como Ansible, Terraform y Kubernetes para la automatización de la infraestructura. La evaluación se llevará a cabo mediante el análisis de métricas relevantes, como el tiempo de implementación y la tasa de éxito en la implementación de infraestructura crítica.

La evaluación debería cubrir al menos una mínima evaluación de la usabilidad de la herramienta, así como de su aplicabilidad para resolver el problema propuesto. Estas evaluaciones suelen realizarse con usuarios expertos, de ser posible, o con pruebas de usabilidad más básicas.

En cualquier caso, sería de especial relevancia y valor añadido el incluir también pruebas unitarias, de integración, según lo que corresponda, validando así la propuesta en la medida de lo posible. Esto es algo crucial para que queden más clarificadas las contribuciones que aporta el desarrollo.

4.5. Aplicación de la propuesta

La propuesta de automatización del proceso de implementación de infraestructura TI para soportar servicios de misión crítica ha sido sometida a una evaluación exhaustiva con el objetivo de analizar y comparar los resultados obtenidos antes y después de la implementación de la automatización. Esta evaluación se realizó mediante un pre y post test, que permitió medir de manera objetiva los cambios y mejoras generados por la automatización.

En el pretest, se realizó una evaluación inicial del proceso de implementación de infraestructura TI antes de aplicar la propuesta de automatización. Durante esta fase, se recopilaron datos relacionados con el tiempo de implementación, la cantidad de errores encontrados, el costo asociado, la complejidad del proceso y la capacidad de respuesta. Estos indicadores proporcionaron una base sólida para evaluar el impacto de la automatización. Posteriormente, se procedió a realizar el post test, en el cual se evaluaron nuevamente los indicadores clave del proceso de implementación. Se registraron y compararon los datos pre y post test.

Los resultados obtenidos en el post test demostraron mejoras significativas en todos los indicadores evaluados:

Tabla 13. Tabulación de datos

	<i>Tiempo</i>		<i>Complejidad</i>		<i>Costo</i>		<i>Errores</i>		<i>Productividad</i>	
	<i>(horas)</i>		<i>(índice)</i>		<i>US \$</i>		<i>(porcentajes)</i>		<i>(índice)</i>	
	Pre	Post	Pre	Post	Pre	Post	Pre	Post	Pre	Post
1	40	32	35	29	2500	2400	15	11	150	188
2	33	31	31	27	2350	2100	12	9	90	99
3	36	32	33	31	2400	2400	14	11	70	101
4	37	35	29	29	2100	1900	11	8	180	189

5	35	31	32	28	2300	1800	13	9	115	123
6	42	40	34	27	2400	2350	15	12	119	134
7	34	30	35	35	2350	2100	12	9	150	167
8	33	29	32	29	2100	2000	14	14	120	145
9	39	42	31	28	2400	2100	14	13	98	101
10	38	31	34	31	2350	1900	13	13	145	156
11	32	28	33	27	2200	2100	10	9	156	167
12	36	31	35	32	2300	2200	12	11	134	156
13	37	31	31	29	2450	2400	11	10	137	189
14	42	28	34	29	2200	2100	13	13	98	120
15	45	40	35	31	2400	2300	14	12	102	132
16	36	31	33	29	2350	2300	15	14	110	134
17	38	31	32	28	2400	2300	13	12	138	156
18	39	38	34	32	2300	2100	12	11	88	189
19	37	35	33	29	2100	1800	15	14	110	131
20	35	31	31	28	2350	2350	14	14	105	115
21	36	32	35	31	2450	2100	13	9	120	128
22	39	32	34	32	2600	2300	11	8	99	109
23	35	31	33	28	2500	2400	13	7	130	136
24	37	35	32	28	2300	2100	12	11	145	178
25	35	35	31	28	2200	2300	11	8	178	198
26	36	32	35	31	2300	1800	14	12	104	114

27	37	32	28	28	2100	2000	15	13	87	95
28	38	32	33	27	2400	2100	12	9	95	110
29	35	31	32	30	2300	2200	15	13	112	132
30	38	32	35	30	2400	2300	12	11	134	154
31	39	29	34	31	2350	2300	9	9	112	122
32	41	43	29	28	2400	2100	10	7	145	156
33	43	41	31	39	2100	2000	12	10	178	198
34	44	42	32	30	2400	2350	13	12	120	122
35	46	42	33	31	2100	2000	14	11	134	139

Fuente: Elaboración propia

Según los resultados de los indicadores para el proceso de implementación de la infraestructura TI se procedió a calcular la normalidad de los datos. Donde se considera lo siguiente:

H. nula: siguen una distribución normal($p>0.05$)

H. alterna: no siguen una distribución normal($p<0.05$)

Tabla 14. Prueba de normalidad del indicador complejidad

Pruebas de normalidad

	Kolmogorov-Smirnov ^a			Shapiro-Wilk(n<50)		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Complejidad_pre	,139	35	,087	,919	35	,013
Complejidad_post	,187	35	,003	,823	35	,0001

Fuente: Elaboración propia

Para el indicador de la complejidad en el pretest, se determina que con la prueba Shapiro Wilk el sig. bilateral fue de 0.013($p<0.05$) y que dicho indicador no sigue una distribución normal. Después de la implementación del proceso de automatización de la infraestructura TI, con lo que respecta al indicador de la complejidad en el post test fue de 0.0001($p<0.05$), determinando que los datos no siguen una distribución normal.

Tabla 15. Prueba de normalidad del indicador tiempo

Pruebas de normalidad

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Tiempo_pre	,136	35	,101	,948	35	,101
Tiempo_post	,303	35	,000	,832	35	,000

Fuente: Elaboración propia

Para el indicador tiempo de implementación en el pretest, se determina que con la prueba Shapiro Wilk su sig bilateral fue de 0.101($p>0.05$), se interpreta que el tiempo en horas sigue una distribución normal.

Después de la implementación del proceso de automatización de la infraestructura TI, con lo que respecta al indicador de tiempo en el post test, su sig. bilateral fue de 0.000 ($p<0.05$) mostrando que los datos no siguen una distribución normal.

Tabla 16. Prueba de normalidad del indicador costo

Pruebas de normalidad

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
<i>Costo_pre</i>	,182	35	,005	,914	35	,010
<i>Costo_post</i>	,189	35	,003	,916	35	,011

Fuente: Elaboración propia

Para el indicador de costos en el pretest, se determina que con la prueba Shapiro Wilk su valor fue de 0.01($p<0.05$) demostrando que el valor de las muestras no sigue una distribución normal.

Con lo que respecta al indicador de costos en el post test, su sig. bilateral fue de 0.011 mostrando que los datos no siguen una distribución normal.

Tabla 17. Prueba de normalidad del indicador productividad

Pruebas de normalidad

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
<i>Productividad_pre</i>	,117	35	,200*	,967	35	,375
<i>Productividad_post</i>	,125	35	,181	,942	35	,064

Fuente: Elaboración propia

Para el indicador de productividad en el pretest, se determina que con la prueba Shapiro Wilk el sig. bilateral fue de 0.375($p>0.05$) y que dicho indicador sigue una distribución normal.

Con lo que respecta al indicador de productividad en el post test, su sig. bilateral fue de 0.064 mostrando que los datos siguen una distribución normal.

Tabla 18. Prueba de normalidad del indicador errores en la implementación

	Pruebas de normalidad					
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
<i>Errores_pre</i>	,144	35	,064	,937	35	,044
<i>Errores_post</i>	,152	35	,040	,940	35	,057

Fuente: Elaboración propia

Para el indicador de índice de errores en el pretest, se determina que el sig. bilateral fue de 0.044($p<0.05$) y que dicho indicador no sigue una distribución normal.

Con lo que respecta al indicador de índices de errores en el post test, su sig. bilateral fue de 0.057 mostrando que los datos siguen una distribución normal.

4.5.1. Análisis de la mejora implementada

Una vez calculada la normalidad de los indicadores, es necesario comprobar la efectividad de la implementación. Para su verificación se realiza la prueba t- student en muestras pareadas según los indicadores que siguen una distribución normal, en caso de no seguir una distribución normal deberá ser realizada a través de la prueba Wilcoxon, donde se considera lo siguiente:

Ho (Hipótesis nula): No es efectivo ($p>0.05$)

Ha (Hipótesis alterna): Es efectivo y significativo ($p<0.05$)

4.5.1.1. Análisis del tiempo de implementación

Al analizar que el indicador no sigue una distribución normal entonces se vio pertinente usar la prueba Wilcoxon. Por lo tanto, en la prueba estadística se encontró un valor de sig bilateral 0.0003($p < 0,05$), al rechazarse la hipótesis nula, entonces el tiempo de implementación fue efectivo y se redujo.

Tabla 19. Prueba estadística para indicador de tiempo

<i>Estadísticos de prueba^a</i>	
	Tiempo_post - Tiempo_pre
Z	-4,850 ^b
Sig. asintótica (bilateral)	,0003

Fuente: Elaboración propia

a. Prueba de Wilcoxon de los rangos con signo

b. Se basa en rangos positivos.

4.5.1.2. Análisis de la complejidad después de la implementación

Al analizar que el indicador, no sigue una distribución normal entonces se vio pertinente usar la prueba Wilcoxon. Por lo tanto, en la prueba estadística se encontró un valor de sig bilateral 0.0002($p < 0,05$), al rechazarse la hipótesis nula, entonces se determina que el índice de complejidad se redujo.

Tabla 20. Prueba estadística para indicador de complejidad

<i>Estadísticos de prueba^a</i>	
	Complejidad_post - Complejidad_pre
Z	-4,361 ^b
Sig. asintótica (bilateral)	,0002

Fuente: Elaboración propia

a. Prueba de Wilcoxon de los rangos con signo

b. Se basa en rangos positivos.

4.5.2. Análisis del costo después de la implementación

Al analizar que el indicador, no sigue una distribución normal entonces se vio pertinente usar la prueba Wilcoxon. Por lo tanto, en la prueba estadística se encontró un valor de sig bilateral 0.00001($p < 0,05$), al rechazarse la hipótesis nula, entonces se determina que el costo de implementación se redujo.

Tabla 21. Prueba estadística para indicador de costos

<i>Estadísticos de prueba^a</i>	
	Costo_post - Costo_pre
Z	-4,837 ^b
<i>Sig. asintótica (bilateral)</i>	.000001

Fuente: Elaboración propia

a. Prueba de Wilcoxon de los rangos con signo

b. Se basa en rangos positivos.

4.5.3. Análisis de la cantidad de errores después de la implementación

Al analizar que el indicador no sigue una distribución normal entonces se vio pertinente usar la prueba Wilcoxon. Por lo tanto, en la prueba estadística se encontró un valor de sig bilateral 0.00001($p < 0,05$) al rechazarse la hipótesis nula, entonces se determina que el índice de errores se redujo.

Tabla 22. Prueba estadística para indicador de errores

<i>Estadísticos de prueba^a</i>	
	Errores_post - Errores_pre
Z	-4,835 ^b
<i>Sig. asintótica (bilateral)</i>	.000001

Fuente: Elaboración propia

a. Prueba de Wilcoxon de los rangos con signo

b. Se basa en rangos positivos

4.5.4. Análisis de la productividad después de la implementación

Al analizar que el indicador de productividad sigue la distribución normal es necesario usar la prueba t -student para muestras relacionadas. En la prueba estadística se encontró un valor de sig bilateral 0.001($p < 0,05$), al rechazarse la hipótesis nula, entonces se comprobó que la productividad se incrementó.

Tabla 23. Prueba t student del antes y después del indicador de productividad

	Diferencias emparejadas				t	gl	Sig. (bilateral)	
	Media	Desviación estándar	Media de error estándar	95% de intervalo de confianza de la diferencia				
				Inferior				Superior
Productividad Pre - Post	-19.29	17.78	3.00	-25.39	-13.18	-6.42	34	.0000002

Fuente: Elaboración propia

H nula: $p > 0.05$ -no hay diferencia entre el antes y el después en la productividad

H alterna: $p < 0.05$ -haya diferencia entre el antes y después en la productividad

Tabla 24. Resultados del indicador de tiempo según objetivo

Muestras	Tiempo		Reduce el tiempo	Cumple el Objetivo
	Pre	Post		
1	40	32	Si	Si
2	33	31	Si	Si
3	36	32	Si	Si
4	37	35	Si	Si
5	35	31	Si	Si
6	42	40	Si	Si
7	34	30	Si	Si
8	33	29	Si	Si
9	39	42	Si	Si
10	38	31	Si	Si
11	32	28	Si	Si
12	36	31	Si	Si

13	37	31	Si	Si
14	42	28	Si	Si
15	45	40	Si	Si
16	36	31	Si	Si
17	38	31	Si	Si
18	39	38	Si	Si
19	37	35	Si	Si
20	35	31	Si	Si
21	36	32	Si	Si
22	39	32	Si	Si
23	35	31	Si	Si
24	37	35	Si	Si
25	35	35	Si	Si
26	36	32	Si	Si
27	37	32	Si	Si
28	38	32	Si	Si
29	35	31	Si	Si
30	38	32	Si	Si
31	39	29	Si	Si
32	41	43	Si	Si
33	43	41	Si	Si
34	44	42	Si	Si
35	46	42	Si	Si

Fuente: Elaboración propia

En la tabla se muestra que se cumplió el objetivo, demostrando que el tiempo se redujo.

Tabla 25. Resultados del indicador complejidad según objetivo

<i>Muestras</i>	<i>Complejidad</i>		<i>Disminución de la complejidad</i>	<i>Cumple el Objetivo</i>
	<i>Pre</i>	<i>Post</i>		
1	35	29	Si	Si
2	31	27	Si	Si
3	33	31	Si	Si
4	29	29	Si	Si

Automatización del proceso de implementación de infraestructura TI para soportar servicios de misión crítica.

5	32	28	Si	Si
6	34	27	Si	Si
7	35	35	Si	Si
8	32	29	Si	Si
9	31	28	Si	Si
10	34	31	Si	Si
11	33	27	Si	Si
12	35	32	Si	Si
13	31	29	Si	Si
14	34	29	Si	Si
15	35	31	Si	Si
16	33	29	Si	Si
17	32	28	Si	Si
18	34	32	Si	Si
19	33	29	Si	Si
20	31	28	Si	Si
21	35	31	Si	Si
22	34	32	Si	Si
23	33	28	Si	Si
24	32	28	Si	Si
25	31	28	Si	Si
26	35	31	Si	Si
27	28	28	Si	Si
28	33	27	Si	Si
29	32	30	Si	Si
30	35	30	Si	Si
31	34	31	Si	Si
32	29	28	Si	Si
33	31	39	Si	Si
34	32	30	Si	Si
35	33	31	Si	Si

Fuente: Elaboración propia

En la tabla se indica que se cumplió el objetivo, demostrando de la automatización del proceso de implementación reduce la complejidad de la implementación.

Tabla 26. Resultados del indicador costo según objetivo

<i>Muestras</i>	<i>Costo</i>		<i>Reduce el Costo</i>	<i>Cumple el Objetivo</i>
	<i>Pre</i>	<i>Post</i>		
1	2500	2400	Si	Si
2	2350	2100	Si	Si
3	2400	2400	Si	Si
4	2100	1900	Si	Si
5	2300	1800	Si	Si
6	2400	2350	Si	Si
7	2350	2100	Si	Si
8	2100	2000	Si	Si
9	2400	2100	Si	Si
10	2350	1900	Si	Si
11	2200	2100	Si	Si
12	2300	2200	Si	Si
13	2450	2400	Si	Si
14	2200	2100	Si	Si
15	2400	2300	Si	Si
16	2350	2300	Si	Si
17	2400	2300	Si	Si
18	2300	2100	Si	Si
19	2100	1800	Si	Si
20	2350	2350	Si	Si
21	2450	2100	Si	Si
22	2600	2300	Si	Si
23	2500	2400	Si	Si
24	2300	2100	Si	Si
25	2200	2300	Si	Si
26	2300	1800	Si	Si
27	2100	2000	Si	Si
28	2400	2100	Si	Si
29	2300	2200	Si	Si
30	2400	2300	Si	Si

31	2350	2300	Si	Si
32	2400	2100	Si	Si
33	2100	2000	Si	Si
34	2400	2350	Si	Si
35	2100	2000	Si	Si

Fuente: Elaboración propia

En la tabla se indica que se ha cumplido el objetivo, demostrando que el costo de la implementación se redujo.

Tabla 27. Resultados del indicador de errores según objetivo

<i>Muestras</i>	<i>Errores</i>		<i>Reduce los errores</i>	<i>Cumple el Objetivo</i>
	<i>Pre</i>	<i>Post</i>		
1	15	11	Si	Si
2	12	9	Si	Si
3	14	11	Si	Si
4	11	8	Si	Si
5	13	9	Si	Si
6	15	12	Si	Si
7	12	9	Si	Si
8	14	14	Si	Si
9	14	13	Si	Si
10	13	13	Si	Si
11	10	9	Si	Si
12	12	11	Si	Si
13	11	10	Si	Si
14	13	13	Si	Si
15	14	12	Si	Si
16	15	14	Si	Si
17	13	12	Si	Si
18	12	11	Si	Si
19	15	14	Si	Si
20	14	14	Si	Si

21	13	9	Si	Si
22	11	8	Si	Si
23	13	7	Si	Si
24	12	11	Si	Si
25	11	8	Si	Si
26	14	12	Si	Si
27	15	13	Si	Si
28	12	9	Si	Si
29	15	13	Si	Si
30	12	11	Si	Si
31	9	9	Si	Si
32	10	7	Si	Si
33	12	10	Si	Si
34	13	12	Si	Si
35	14	11	Si	Si

Fuente: Elaboración propia

En la tabla se indica que se ha cumplido el objetivo, demostrando que con la implementación la cantidad del índice de errores se redujeron.

Tabla 28. Resultados del indicador de productividad según objetivo

<i>Muestras</i>	<i>Productividad</i>		<i>Incrementa la productividad</i>	<i>Cumple el Objetivo</i>
	<i>Pre</i>	<i>Post</i>		
1	150	188	Si	Si
2	90	99	Si	Si
3	70	101	Si	Si
4	180	189	Si	Si
5	115	123	Si	Si
6	119	134	Si	Si
7	150	167	Si	Si
8	120	145	Si	Si
9	98	101	Si	Si
10	145	156	Si	Si
11	156	167	Si	Si

Automatización del proceso de implementación de infraestructura TI para soportar servicios de misión crítica.

12	134	156	Si	Si
13	137	189	Si	Si
14	98	120	Si	Si
15	102	132	Si	Si
16	110	134	Si	Si
17	138	156	Si	Si
18	88	189	Si	Si
19	110	131	Si	Si
20	105	115	Si	Si
21	120	128	Si	Si
22	99	109	Si	Si
23	130	136	Si	Si
24	145	178	Si	Si
25	178	198	Si	Si
26	104	114	Si	Si
27	87	95	Si	Si
28	95	110	Si	Si
29	112	132	Si	Si
30	134	154	Si	Si
31	112	122	Si	Si
32	145	156	Si	Si
33	178	198	Si	Si
34	120	122	Si	Si
35	134	139	Si	Si

Fuente: Elaboración propia

En la tabla se indica que se cumplió el objetivo, demostrando de que con la implementación el índice de la productividad se mejoró.

5. Conclusiones y trabajo futuro

En esta sección se presentan las conclusiones derivadas del trabajo realizado, así como los beneficios tangibles obtenidos, una visión clara de los resultados y su relevancia en relación con los objetivos planteados. Se resalta el impacto positivo que la implementación de la automatización del proceso de implementación de infraestructura TI para soportar servicios de misión crítica, destacando aspectos como la reducción de la complejidad, tiempos de implementación, costos asociados, errores humanos y aumento de la productividad. Además, se destacan beneficios como la agilidad a través de adopción de metodologías Agile.

5.1. Conclusiones

El presente estudio da como resultado un proceso automatizado para la implementación de infraestructura TI, abordando desde la problemática existente, ofreciendo un marco teórico y conceptual sólido e incluyendo la ejecución eficiente de tareas, para con ello lograr reducir los tiempos de implementación y minimizar la probabilidad de errores humanos causados por tareas repetitivas y propensas a errores.

A través de la integración de diversas tecnologías de automatización, como Jira, Gitlab, Jenkins, Ansible, Terraform, Bash scripting, PowerShell, Python, Packer y vSphere, se logró mejorar la eficiencia y la calidad de despliegue para nuevos requerimientos de infraestructura TI.

La implementación automatizada propuesta ha demostrado beneficios significativos. Se logró una reducción del 10.95% en el tiempo del proceso de automatización de operaciones, lo que ha permitido una mayor agilidad en la entrega de servicios de TI.

Se alcanzó una reducción del 7.2% en los costos asociados a la automatización de operaciones, lo que representa un ahorro significativo para la organización.

Se observó una disminución del 9.09% en la complejidad del proceso de automatización de operaciones, lo que ha facilitado su gestión y mantenimiento.

La efectividad del proceso se incrementó en un 13.54%, lo que se reflejó en una mayor calidad de los servicios ofrecidos por el equipo de implementación.

Por último, se registró una reducción del 15.4% en los errores relacionados, lo que ha contribuido a mejorar la confiabilidad y estabilidad de las nuevas implementaciones.

Estos resultados indican que la automatización del proceso de implementación de infraestructura TI, ha sido exitosa en la consecución de sus objetivos. ha generado mejoras significativas en todos los indicadores de medición, demostrando su efectividad y su potencial para ser implementada en otras organizaciones.

5.2. Líneas de trabajo futuro

En el proceso de automatización, es crucial realizar un análisis exhaustivo de los procesos manuales, tanto nuevos como existentes, en la implementación de infraestructura de TI. Esto nos permitirá identificar áreas propensas a errores, tareas que consumen mucho tiempo y altos costos, con el objetivo de proponer soluciones automatizadas que mejoren la eficiencia y precisión del proceso. Es fundamental garantizar la continuidad en este análisis detallado de los procesos manuales, para aprovechar todas las oportunidades de optimización disponibles.

Para mejorar y enriquecer el proceso de automatización, es necesario ampliar la investigación y explorar nuevas herramientas que puedan complementar las ya mencionadas. Esto implica evaluar herramientas de orquestación, gestión de configuración y CI/CD adicionales, con el fin de encontrar aquellas que se adapten de manera específica a las necesidades de la implementación de infraestructura de TI. Este enfoque de investigación permitirá estar al tanto de las últimas innovaciones y garantizar que se cuente con un conjunto completo de herramientas que optimicen el proceso de automatización.

Considerando que se menciona Jira en el presente estudio, se puede explorar aún más la integración, seguimiento y gestión de proyectos, esto permitirá una mayor visibilidad, coordinación y colaboración entre los diferentes equipos involucrados en el proceso.

Dado el uso de scripts y manifiestos en el proceso de automatización, se debe realizar una revisión continua y mejorar la calidad del código, optimizar su rendimiento y mantener una estructura coherente. Además, se puede considerar la creación de bibliotecas o módulos reutilizables para agilizar el desarrollo y la implementación de infraestructura de TI.

Es importante establecer métricas y realizar un seguimiento de los indicadores clave de rendimiento para evaluar la efectividad de la automatización en el proceso de implementación de infraestructura de TI. Esto implica establecer criterios de éxito, definir métricas relevantes, así como la utilización de herramientas de monitorización para recopilar datos y generar informes periódicos.

Referencias bibliográficas

- Atlassian. (27 de 03 de 2023). *Infrastructure as code*. Obtenido de <https://www.atlassian.com/es/microservices/cloud-computing/infrastructure-as-code>
- Atlassian. (13 de 05 de 2023). *Jira*. Obtenido de <https://www.atlassian.com/es/software/jira>
- AWS. (07 de 05 de 2023). *Cloud Computing*. Obtenido de <https://aws.amazon.com/es/what-is-cloud-computing/>
- AWS. (07 de 05 de 2023). *Virtualización, ¿Qué es?* Obtenido de <https://aws.amazon.com/es/what-is/virtualization/>
- Cameron, P. (30 de 10 de 2022). Automatización de procesos: beneficios para el mundo empresarial. Obtenido de <https://www.fortra.com/es/recursos/guias/automatizacion-de-procesos-5-principales-beneficios-en-empresas>
- Chacón, L. (2022). Diseño, desarrollo e implementación de una arquitectura GitOps (Git y Kubernetes) que integre infraestructura basada en código desplegadas en plataformas IaaS (Infraestructura como Servicio). *[Trabajo de Titulación]*. Universidad Politécnica Salesiana, Cuenca. Obtenido de <https://dspace.ups.edu.ec/bitstream/123456789/22397/1/UPS-CT009712.pdf>
- Choez, C. (2022). Automatización de infraestructura basada en software de libre distribución. *[Trabajo Fin de Máster]*. Pontificia Universidad Católica del Ecuador, Quito.
- Git. (20 de 05 de 2023). *Git*. Obtenido de <https://git-scm.com/>
- Hashicorp. (30 de 03 de 2023). *Terraform*. Obtenido de <https://developer.hashicorp.com/terraform/docs>
- Huerlo, J. (2021). Terraform Como Herramienta Para Automatizar La Creación De Infraestructuras Siguiendo El Concepto "Infraestructura Como Código". *[Trabajo de titulación]*. PUCESE - Escuela de Sistemas y Computación, Esmeraldas. Obtenido de <https://repositorio.pucese.edu.ec/bitstream/123456789/2602/1/Huerlo%20Quintero%20Jurgen%20Ronaldo.pdf>

- ITIC. (2022). *Global server hardware, server OS reliability report*. IBM. Obtenido de <http://www.ibm.com/downloads/cas/VQ5B65YZ>
- Jenkins. (14 de 05 de 2023). *Jenkins*. Obtenido de <https://www.jenkins.io/>
- Loo, F., & Rojas, C. (2018). Modelo de migración a la nube de los servidores de un data center. *[Trabajo de titulación]*. Universidad Peruana de Ciencias Aplicadas (UPC), Lima, Lima.
- Manyika, J., Chui, M., Miremadi, M., Bughin, J., George, K., Willmott, P., & Dewhurst, M. (2017). *Un futuro que funciona: Automatización, empleo y productividad*. McKinsey Global Institute. Obtenido de <https://www.mckinsey.com/~media/mckinsey/featured%20insights/digital%20disruption/harnessing%20automation%20for%20a%20future%20that%20works/a-future-that-works-executive-summary-spanish-mgi-march-24-2017.pdf>
- Medina, D. (06 de 07 de 2021). Metodologías ágiles. (R. Unir, Entrevistador) Obtenido de Metodologías ágiles.
- Moreno, J. (2022). CI/CD en infraestructura como código (IaC). Caso práctico en Amazon Web Services (AWS). *[Trabajo de titulación]*. Universidad Oberta de Catalunya, Catalunya. Obtenido de https://openaccess.uoc.edu/bitstream/10609/145707/8/moreno_juanTFG0622memoria.pdf
- OpenWebinars. (22 de 03 de 2021). *Cloud Computing*. Obtenido de <https://openwebinars.net/blog/tipos-de-cloud-computing/>
- Payá, A. (2021). Egida: Sistemas para el despliegue automatizado de configuraciones de seguridad con detección temprana de errores. *[Proyecto de investigación]*. Universidad de Oviedo, Oviedo, España. Obtenido de https://digibuo.uniovi.es/dspace/bitstream/handle/10651/60335/TFM_AntonioPay%c3%a1Gonz%c3%a1lez.pdf
- Rázuri, A. (2019). Desarrollo de un Sistema de Gestión de Continuidad de Negocio en una entidad financiera, basado en la ISO 22301. *[Trabajo de titulación]*. Universidad Nacional Mayor de San Marcos, Lima. Obtenido de <https://hdl.handle.net/20.500.12672/10726>

RedHat. (24 de 05 de 2022). *Ansible*. Obtenido de <https://www.ansible.com>

RedHat. (11 de 05 de 2022). *RedHat, IaC*. Obtenido de <https://www.redhat.com/es/topics/automation/what-is-infrastructure-as-code-iac>

RedHat. (14 de 02 de 2023). *¿Qué es la automatización?* . Obtenido de <https://www.redhat.com/es/topics/automation/whats-it-automation>

Reyes, A. (2022). Metodología de operaciones para la automatización de procesos del centro de datos. *[Trabajo de fin de master]*. UNIR, Lima. Obtenido de <https://reunir.unir.net/bitstream/handle/123456789/13730/Reyes%20Vargas%2c%20Andy.pdf>

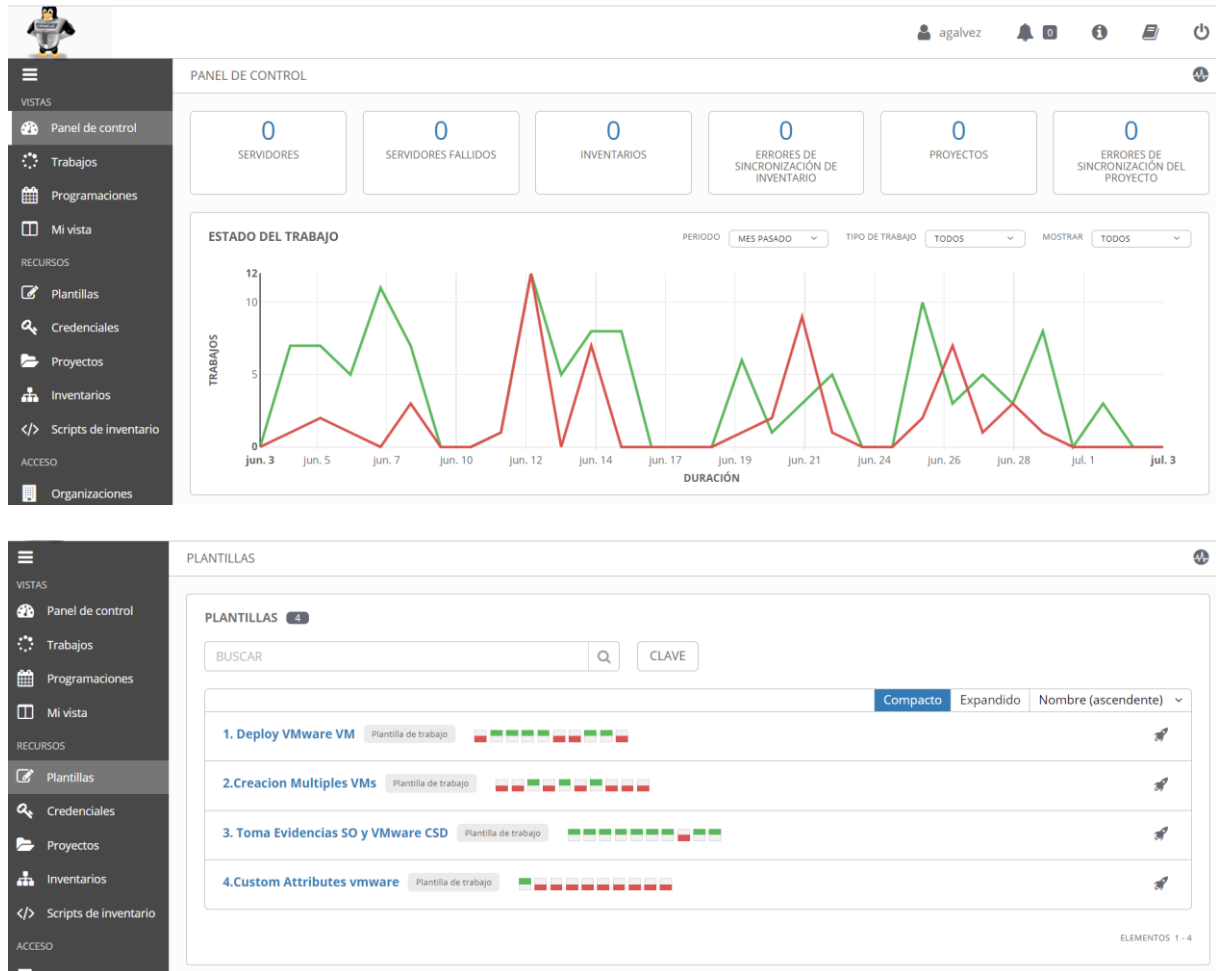
Sánchez, M. (2018). Despliegue automatizado de infraestructura de Microservicios y prueba de concepto de su funcionamiento. *[Trabajo de fin de master]*. Universidad de Granada, Granada.

VMware. (18 de 02 de 2023). *vSphere*. Obtenido de <https://www.vmware.com/es/products/vsphere.html>

Yunga, A. (2018). Implementación del provisionamiento automático de configuraciones (Network Automation) en infraestructuras multivendor con Ansible. *[Trabajo de titulación]*. Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador.

Anexo A. Entorno de automatización

Dashboard - Oracle Linux Automation Manager



Anexo B. Entorno de repositorio de código

Plataforma GitLab

The screenshot shows the GitLab interface for a repository named 'B2B'. The left sidebar contains navigation options: Project information, Repository, Files, Commits, Branches, Tags, Contributors, Graph, Compare, Issues (0), Merge requests (0), CI/CD, and Deployments. The main content area shows the repository path 'olam > B2B > Repository' and a dropdown menu for the 'main' branch. A recent commit is displayed: 'Update SuseEvidencias.j2' by Administrator, authored 1 week ago. Below this is a table of files and their last commit details.

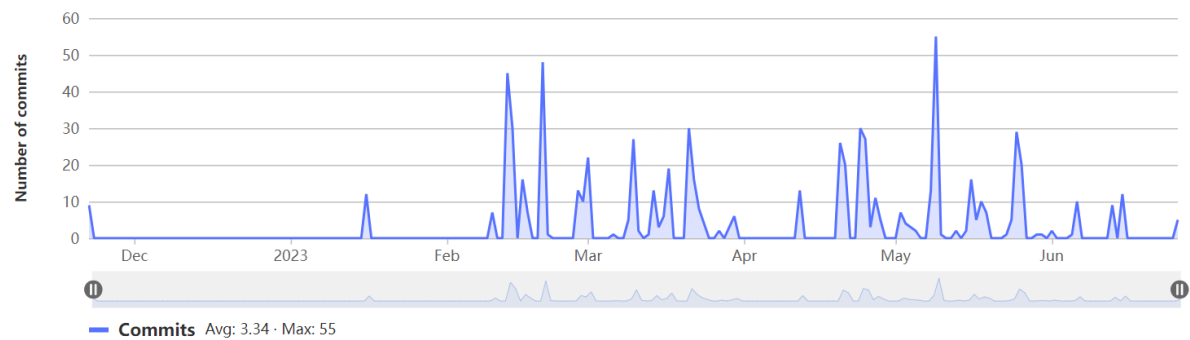
Name	Last commit	Last update
FILES	Add new directory	3 months ago
PLAYBOOK	Update cumstumer.yml	1 month ago
PWS	Update WindowsDelivery_V5.ps1	1 month ago
Roles	Update SuseEvidencias.j2	1 week ago
SHELL	Update UbuntuEvidencias.sh	1 month ago
VARS	Update DataCenter.yml	2 weeks ago

olam > B2B > Contributors

The contributors section shows the 'main' branch selected and a 'History' button. The area below is currently empty, indicating no contributors are listed for this branch.

Commits to main

Excluding merge commits. Limited to 6,000 commits.



Anexo C. Manifiesto Ansible

Código de despliegue

```

1 ---
2 - hosts: localhost
3   connection: local
4   gather_facts: false
5   vars_files:
6     - "/VARS/DataCenter.yml" #Vars File For DataCenter Info
7   tasks:
8     - name: Set DC Variables
9       set_fact:
10        VCENTER: "{{ item.IP }}"
11        DATACENTER: "{{ item.DCNAME }}"
12        CLUSTER: "{{ item.CLNAME }}"
13        loop: "{{ CWBDC }}"
14        when: DC_NAME == "{{ item.SITE }}"
15     - vmware_guest:
16        hostname: "{{ VCENTER }}" # IP del vCenter
17        username: 'correo@domain.com' # Usuario de vCenter
18        password: 'password' # Password de vCenter
19        validate_certs: False
20        name: "{{ HOSTNAME }}" # Nombre de VM y de como se visualizará en el vCenter
21        #content_library: 'ContentLibrary'
22        template: '{{ SO }}' # Nombre de la Plantilla en el vCenter
23        datacenter: "{{ DATACENTER }}" # Nombre del Datacenter donde se va a desplegar
24        cluster: "{{ CLUSTER }}" # Nombre del Cluster donde se va a desplegar
25        folder: '{{ FOLDER }}' # Nombre del folder donde se va a desplegar
26        annotation: 'Servidor desplegado desde Ansible' # Nota que se quiera colocar en el Servidor desplegado
27        disk:
28        - size_gb: '{{ DISK }}' # Tamaño a asignar al disco del Sistema Operativo
29          type: thin # Formato del disco thin, thick o eagerzeroedthick
30          datastore: 'DS_SERVICE_ANSIBLE_SIDIT' # Nombre del Datastore donde se va a desplegar
31        hardware:
32        memory_mb: '{{ RAM | int * 1024 }}' # Cantidad de Memoria RAM a asignar
33        num_cpus: '{{ CPU }}' # Cantidad de CPUs a asignar
34        scsi: paravirtual # Formato de conector de disco atavez de VMware
35        networks:
36        - name: '{{ portgroup }}' # Nombre de la Red a asignar
37          ip: '{{ IP }}' # Dirección IP a asignar
38          netmask: '{{ MASK }}' # Mascara de Red a asignar
39          gateway: '{{ GW }}' # Gateway a asignar
40          dns_servers: '{{ DNS }}' # Dirección (es) IP de los DNS a asignar
41          device_type: vmxnet3 #Tipo de Interfaz de Red a asignar
42          start_connected: True
43          state: poweredon # Estado del Servidor luego del despliegue
44          wait_for_ip_address: yes
45 ..

```

Generación de evidencias

```

1 ---
2 #####EVIDENCIAS DE VMWARE 3 ARCHIVOS #####
3 # Elaborado por: AGH - Implementacion #
4 # Fecha: 09 Marzo de 2023 #
5 # Compañía: #
6 # Plataforma: VMWare #
7 #####
8 # Creacion de evidencia a nivel de Vmware #
9 #####
10 - hosts: localhost
11   tasks:
12     - name: Creacion carpeta
13       file:
14         path: "/evidences/{{ CLIENTE }}"
15         state: directory
16     - name: Creacion carpeta
17       file:
18         path: "/evidences/{{ CLIENTE }}/{{ WO }}"
19         state: directory
20     - name: Ejecutar Evidencias de Vmware
21       shell:
22         args:
23         executable: /usr/bin/pwsh
24         cmd: PWS/Evidences2.ps1 "{{ VM }}" "{{ CLIENTE }}/{{ WO }}" "{{ VCENTER }}"
25         # "{{ VCUSER }}" "{{ VCPWD }}"
26         #command: ../SCRIPT/PWS/Evidences2.ps1 "{{ VM }}"
27       register: result
28     - debug: var=result.stdout_lines

```

Anexo D. Bash Scripts

Script de evidencias

```

1  #!/bin/bash
2  cat /dev/null > /tmp/Evidencias.txt
3  echo "****" >> /tmp/Evidencias.txt
4  echo "===== " >> /tmp/Evidencias.txt
5  echo "===== ARCHIVO EVIDENCIAS LINUX ===== " >> /tmp/Evidencias.txt
6  echo "===== " >> /tmp/Evidencias.txt
7  echo "****" >> /tmp/Evidencias.txt
8  echo "" >> /tmp/Evidencias.txt
9  echo "$(date)" >> /tmp/Evidencias.txt
10 echo "" >> /tmp/Evidencias.txt
11 echo "===== " >> /tmp/Evidencias.txt
12 echo "===== HOSTNAME ===== " >> /tmp/Evidencias.txt
13 echo "===== " >> /tmp/Evidencias.txt
14 echo "" >> /tmp/Evidencias.txt
15 echo "$(hostname)" >> /tmp/Evidencias.txt
16 echo "" >> /tmp/Evidencias.txt
17 echo "===== " >> /tmp/Evidencias.txt
18 echo "===== EVIDENCIAS DISCOS ===== " >> /tmp/Evidencias.txt
19 echo "===== " >> /tmp/Evidencias.txt
20 echo "" >> /tmp/Evidencias.txt
21 echo "$(lsblk | grep disk)" >> /tmp/Evidencias.txt
22 echo "" >> /tmp/Evidencias.txt
23 echo "$(lvsdisplay -v)" >> /tmp/Evidencias.txt
24 echo "" >> /tmp/Evidencias.txt
25 echo "===== " >> /tmp/Evidencias.txt
26 echo "===== USO DE DISCO ===== " >> /tmp/Evidencias.txt
27 echo "===== " >> /tmp/Evidencias.txt
28 echo "" >> /tmp/Evidencias.txt
29 echo "$(df -hTP)" >> /tmp/Evidencias.txt
30 echo "" >> /tmp/Evidencias.txt
31 echo "===== " >> /tmp/Evidencias.txt
32 echo "===== VERSION SO ===== " >> /tmp/Evidencias.txt
33 echo "===== " >> /tmp/Evidencias.txt
34 echo "" >> /tmp/Evidencias.txt
35 echo "$(uname -a)" >> /tmp/Evidencias.txt
36 echo "" >> /tmp/Evidencias.txt
37 echo "$(cat /etc/system-release)" >> /tmp/Evidencias.txt
38 echo "" >> /tmp/Evidencias.txt
39 echo "===== " >> /tmp/Evidencias.txt
40 echo "===== PROCESADOR ===== " >> /tmp/Evidencias.txt
41 echo "===== " >> /tmp/Evidencias.txt
42 echo "" >> /tmp/Evidencias.txt
43 echo "$(lscpu | grep ^"CPU(s)")" >> /tmp/Evidencias.txt
44 echo "" >> /tmp/Evidencias.txt

```

```

45 echo "===== " >> /tmp/Evidencias.txt
46 echo "===== MEMORIA ===== " >> /tmp/Evidencias.txt
47 echo "===== " >> /tmp/Evidencias.txt
48 echo "" >> /tmp/Evidencias.txt
49 echo "$(free -m)" >> /tmp/Evidencias.txt
50 echo "" >> /tmp/Evidencias.txt
51 echo "$(swapon -s)" >> /tmp/Evidencias.txt
52 echo "" >> /tmp/Evidencias.txt
53 echo "$(swapon -s)" >> /tmp/Evidencias.txt
54 echo "" >> /tmp/Evidencias.txt
55 echo "===== " >> /tmp/Evidencias.txt
56 echo "===== UPTIME ===== " >> /tmp/Evidencias.txt
57 echo "===== " >> /tmp/Evidencias.txt
58 echo "" >> /tmp/Evidencias.txt
59 echo "$(uptime)" >> /tmp/Evidencias.txt
60 echo "" >> /tmp/Evidencias.txt
61 echo "$(last | grep reboo )" >> /tmp/Evidencias.txt
62 echo "" >> /tmp/Evidencias.txt
63 echo "===== " >> /tmp/Evidencias.txt
64 echo "===== TIMEDATECTL ===== " >> /tmp/Evidencias.txt
65 echo "===== " >> /tmp/Evidencias.txt
66 echo "" >> /tmp/Evidencias.txt
67 echo "$(timedactl)" >> /tmp/Evidencias.txt
68 echo "" >> /tmp/Evidencias.txt
69 echo "===== " >> /tmp/Evidencias.txt
70 echo "===== REGISTRO ===== " >> /tmp/Evidencias.txt
71 echo "===== " >> /tmp/Evidencias.txt
72 echo "" >> /tmp/Evidencias.txt
73 echo "$(dnf repolist | grep receiving)" >> /tmp/Evidencias.txt
74 echo "$(echo "password" | sudo -S -k subscription-manager list --consumed | grep Status)" >> /tmp/Evidencias.txt
75 echo "" >> /tmp/Evidencias.txt
76 echo "===== " >> /tmp/Evidencias.txt
77 echo "===== ENTORNO GRAFICO ===== " >> /tmp/Evidencias.txt
78 echo "===== " >> /tmp/Evidencias.txt
79 echo "" >> /tmp/Evidencias.txt

```

```

79 echo "" >> /tmp/Evidencias.txt
80 echo "$(ps -edf |grep gnome)" >> /tmp/Evidencias.txt
81 echo "" >> /tmp/Evidencias.txt
82 echo "===== " >> /tmp/Evidencias.txt
83 echo "===== LAST ===== " >> /tmp/Evidencias.txt
84 echo "===== " >> /tmp/Evidencias.txt
85 echo "" >> /tmp/Evidencias.txt
86 echo "$(last)" >> /tmp/Evidencias.txt
87 echo "" >> /tmp/Evidencias.txt
88 echo "===== " >> /tmp/Evidencias.txt
89 echo "===== NTP ===== " >> /tmp/Evidencias.txt
90 echo "===== " >> /tmp/Evidencias.txt
91 echo "" >> /tmp/Evidencias.txt
92 echo "$(chronyc tracking; chronyc sources)" >> /tmp/Evidencias.txt
93 echo "" >> /tmp/Evidencias.txt
94 echo "===== " >> /tmp/Evidencias.txt
95 echo "===== SYSTEMCTL ===== " >> /tmp/Evidencias.txt
96 echo "===== " >> /tmp/Evidencias.txt
97 echo "" >> /tmp/Evidencias.txt
98 echo "$(systemctl | grep running)" >> /tmp/Evidencias.txt
99 echo "" >> /tmp/Evidencias.txt
100 echo "===== " >> /tmp/Evidencias.txt
101 echo "===== YUM HISTORY ===== " >> /tmp/Evidencias.txt
102 echo "===== " >> /tmp/Evidencias.txt
103 echo "" >> /tmp/Evidencias.txt
104 echo "$(yum history)" >> /tmp/Evidencias.txt
105 echo "" >> /tmp/Evidencias.txt
106 echo "===== " >> /tmp/Evidencias.txt
107 echo "===== MESSAGES ===== " >> /tmp/Evidencias.txt
108 echo "===== " >> /tmp/Evidencias.txt
109 echo "" >> /tmp/Evidencias.txt
110 echo "$(echo "password" | sudo -S -k sudo tail -19 /var/log/messages)" >> /tmp/Evidencias.txt
111 echo "" >> /tmp/Evidencias.txt
112 echo "===== " >> /tmp/Evidencias.txt
113 echo "===== DIRECCION IP ===== " >> /tmp/Evidencias.txt
114 echo "===== " >> /tmp/Evidencias.txt
115 echo "" >> /tmp/Evidencias.txt
116 echo "$(hostname -I)" >> /tmp/Evidencias.txt
117 echo "" >> /tmp/Evidencias.txt
118 echo "$(ifconfig)" >> /tmp/Evidencias.txt
119 echo "" >> /tmp/Evidencias.txt

```

```

120 echo "$(echo "password" | sudo -S -k nmcli device status)" >> /tmp/Evidencias.txt
121 echo "" >> /tmp/Evidencias.txt
122 echo "$(netstat -r)" >> /tmp/Evidencias.txt
123 echo "" >> /tmp/Evidencias.txt
124 echo "===== " >> /tmp/Evidencias.txt
125 echo "===== TOP ===== " >> /tmp/Evidencias.txt
126 echo "===== " >> /tmp/Evidencias.txt
127 echo "" >> /tmp/Evidencias.txt
128 echo "$(top -b -n 1)" >> /tmp/Evidencias.txt
129 echo "" >> /tmp/Evidencias.txt
130 echo "=====*****" >> /tmp/Evidencias.txt
131 echo "=====***** CLOUD SERVICE DEPLOYMENT *****" >> /tmp/Evidencias.txt
132 echo "=====*****" >> /tmp/Evidencias.txt
133 echo "" >> /tmp/Evidencias.txt
134 echo "" >> /tmp/Evidencias.txt
135 echo "Cloud Service Deployment CSD@1la.com" >> /tmp/Evidencias.txt
136 echo "" >> /tmp/Evidencias.txt
137

```

Anexo E. PS Scripts

Evidencias Windows Delivery

```

1  $computer = $env:COMPUTERNAME
2
3  Mkdir C:\Tools\ -Force
4  date > "C:\Tools\$computer.txt"
5  Add-Content C:\Tools\$computer.txt "===== HOSTNAME ====="
6  $env:COMPUTERNAME >> "C:\Tools\$computer.txt"
7  Add-Content C:\Tools\$computer.txt "===== HOTFIX ID ====="
8  wmic qfe get HotFixID,"InstalledBy","InstalledOn >> "C:\Tools\$computer.txt"
9  Get-Hotfix | sort installedon >> "C:\Tools\$computer.txt" -ErrorAction SilentlyContinue
10 Add-Content C:\Tools\$computer.txt "===== SYSTEMINFO ====="
11 systeminfo >> "C:\Tools\$computer.txt"
12 Add-Content C:\Tools\$computer.txt "===== DIRECCION IP ====="
13 IPConfig /ALL >> "C:\Tools\$computer.txt"
14 Add-Content C:\Tools\$computer.txt "===== RUTAS ====="
15 route print >> "C:\Tools\$computer.txt"
16 Add-Content C:\Tools\$computer.txt "===== FIREWALL ====="
17 netsh advfirewall show allprofiles >> "C:\Tools\$computer.txt"
18 Add-Content C:\Tools\$computer.txt "===== POLITICAS LOCALES ====="
19 net accounts >> "C:\Tools\$computer.txt"
20 Add-Content C:\Tools\$computer.txt "===== DISCOS ====="
21 Get-WmiObject -Class Win32_LogicalDisk | Select-Object -Property DeviceID, VolumeName, @{Label="FreeSpace (Gb)";
22 expression={$_.FreeSpace/1GB}.ToString('F2')}, @{Label="Total (Gb)"; expression={$_.Size/1GB}.ToString('F2')},
23 @{label="FreePercent"; expression=[Math]::Round((($_.freespace / $_.size) * 100, 2)}}|ft >> "C:\Tools\$computer.txt"
24 Add-Content C:\Tools\$computer.txt "===== ARCHIVO DE PAGINACION ====="
25 Get-CimInstance -ClassName Win32_PageFileUsage | Select-Object -Property Name, AllocatedBaseSize >> "C:\Tools\$computer.txt"
26 Add-Content C:\Tools\$computer.txt "===== SOFTWARE INSTALADO ====="
27 wmic product get Description,"InstallDate","InstallLocation","Version >> "C:\Tools\$computer.txt"
28 Add-Content C:\Tools\$computer.txt "===== CARACTERISTICAS INSTALADAS ====="
29 Get-WindowsFeature | where{$_.InstallState -eq "Installed"} >> "C:\Tools\$computer.txt"
30 Add-Content C:\Tools\$computer.txt "===== CUENTAS & GRUPOS SO ====="
31 $listofgroups = [ADSI]"WinNT://$env:computername,computer"
32 $listofgroups.psbase.children | where { $_.psbase.schemaClassName -eq "group" } | select name >> "C:\Tools\$computer.txt"
33 wmic useraccount where localaccount='TRUE' get Name,"Fullname","Description","Disabled","AccountType","PasswordExpires",
34 "SID","Status >> "C:\Tools\$computer.txt"net localgroup $administratorgrouplocalname >> "C:\Tools\$computer.txt"
35 Add-Content C:\Tools\$computer.txt "===== ACTIVACION PRODUCTO ====="
36 cscript c:\Windows\system32\slmgr.vbs -dlv >> "C:\Tools\$computer.txt"
37 Add-Content C:\Tools\$computer.txt "===== "
38 Add-Content C:\Tools\$computer.txt "===== CLOUD SERVICE DEPLOYMENT ====="
39 Add-Content C:\Tools\$computer.txt "===== "
40 Add-Content C:\Tools\$computer.txt " "
41 Add-Content C:\Tools\$computer.txt " " Cloud Service Deployment | "
42 Add-Content C:\Tools\$computer.txt " "

```