



**Framework Inteligente para el desarrollo, integración y despliegue continuo de Sistemas TI de la
Universidad de las Fuerzas Armadas – ESPE basado en la cultura DevOps: “Smart Software
Development – ESPE”**

Guevara Vaca, Luis José y Yépez Castro, Jonathan Andres

Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Trabajo de integración curricular, previo a la obtención del título de Ingeniero en Tecnologías de la
Información

Dr. Fonseca Carrera, Efraín Rodrigo

25 de febrero del 2022



Tesis-Guevara-Yepez-Version-Copyleaks.docx

Scanned on: 13:34 February 24, 2022 UTC



Overall Similarity Score



Results Found



Total Words in Text

Identical Words	351
Words with Minor Changes	94
Paraphrased Words	517
Omitted Words	517



Escaneado a la fecha de creación por:
BERRAIN RODRIGO
FONSECA CARRERA



Website | Education | Businesses



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Certificación

Certifico que el trabajo de titulación, **"Framework Inteligente para el desarrollo, integración y despliegue continuo de Sistemas TI de la Universidad de las Fuerzas Armadas – ESPE basado en la cultura DevOps: 'Smart Software Development – ESPE' "** fue realizado por los señores **Guevara Vaca Luis José y Yépez Castro Jonathan Andres** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas - ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 25 de febrero del 2022

Firma:



Verificado digitalmente por:
Efraín Rodrigo
FONSECA CARRERA

Dr. Efraín R. Fonseca C

DIRECTOR

C. C.: 1710979574



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

RESPONSABILIDAD DE AUTORÍA

Nosotros, **Guevara Vaca Luis José** y **Yépez Castro Jonathan Andres**, con cédulas de ciudadanía N°: 1724865280 y 1723005532 respectivamente, declaramos que el contenido, ideas y criterios del trabajo de titulación: **Framework Inteligente para el desarrollo, integración y despliegue continuo de Sistemas TI de la Universidad de las Fuerzas Armadas – ESPE basado en la cultura DevOps: “Smart Software Development – ESPE”** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas – ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 25 de febrero del 2022

Firma:

Luis José, Guevara Vaca

C. C.: 1724865280

Firma:

Jonathan Andres, Yépez Castro

C. C.: 1723005532



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

AUTORIZACIÓN DE PUBLICACIÓN

Nosotros, Guevara Vaca Luis José y Yépez Castro Jonathan Andres, con cédulas de ciudadanía N°: 1724865280 y 1723005532 respectivamente, autorizamos a la Universidad de las Fuerzas Armadas – ESPE publicar el trabajo de titulación: **Framework Inteligente para el desarrollo, integración y despliegue continuo de Sistemas TI de la Universidad de las Fuerzas Armadas – ESPE basado en la cultura DevOps: "Smart Software Development – ESPE"** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad

Sangolquí, 25 de febrero del 2022

Firma:

Luis José, Guevara Vaca

C. C.: 1724865280

Firma:

Jonathan Andres, Yépez Castro

C. C.: 1723005532

Dedicatoria

Dedico este trabajo de manera especial a mi tía y abuelos, pues ellos han cumplido el rol de padres en mi vida, sin el apoyo de ellos esto no habría sido posible.

Luis José Guevara Vaca

A mi amada madre dedico este trabajo, quien supo defenderse sola contra toda adversidad para sacar a su hijo adelante. Aquí están los frutos de tu trabajo y esfuerzo.

Jonathan Andres Yépez Castro

Agradecimiento

Agradecer primero a Dios, por darme la oportunidad de afrontar el reto de estudiar en una de las mejores instituciones de educación superior. De manera especial agradezco a mi Tía Melva por ser pilar fundamental tanto en mi formación académica como personal, a mis abuelos Juana y Eduardo a quienes considero mis padres, a mi pequeña Sami quien es la mi mayor motivación, gracias por su apoyo y por inculcarme valores que me han servido para crecer como persona.

Agradecer también a la Universidad UFA – ESPE, pues me brindaron la oportunidad de terminar mis estudios superiores, de manera especial a las autoridades del Departamento de Ciencias de la Computación, a los Ingenieros Luis Rocha y Nelly Cevallos quienes nos permitieron desarrollar este proyecto.

A los docentes que fueron parte de mi aprendizaje durante este periodo, de manera especial al Ingeniero Rodrigo Fonseca por guiarnos en el desarrollo de este trabajo con paciencia, dedicación y profesionalismo.

A los amigos que pude encontrar mientras avanzaba en el proceso de formación, a Jonathan, Ariel y Daniel gracias por las experiencias vividas, por su apoyo y confianza; pues hemos formado un grupo donde cada uno de nosotros sabe que puede contar con el apoyo de los otros.

Luis José Guevara Vaca

Agradecimiento

Me gustaría agradecer ante todo a mi familia, abuelos, tíos, primos que de una manera u otra me apoyan y me ven como un ejemplo, a mi madre Rocío Castro por su esfuerzo del día a día por brindar el sustento en casa y brindarme la oportunidad de estudiar.

Agradezco especialmente a Nicolle Pozo, sin tu apoyo nada de esto sería posible, gracias por saber entenderme y estar en los buenos y malos momentos.

Gracias a la Universidad de las Fuerzas Armadas – ESPE por la oportunidad de culminar mis estudios, a los docentes del Departamento de Ciencias de la Computación por su profesionalismo y a la Unidad de Tecnologías de la Información en especial a los ingenieros Nelly Cevallos y Luis Rocha por darnos todas las facilidades en la elaboración de este proyecto, de la misma manera a nuestro tutor Dr. Rodrigo Fonseca.

A mis amigos Luis Guevara, Ariel Altamirano y Daniel Ávila por todas las vivencias, por ser un grupo unido en el que se puede confiar y saber que siempre se puede contar con su apoyo.

Cuando la gratitud es total las palabras sobran.

Jonathan Andres Yépez Castro

Índice de Contenido

Certificación	3
Dedicatoria.....	6
Índice de Contenido	9
Índice de Tablas	12
Índice de Figuras	13
Resumen	14
Abstract.....	15
Capítulo I	16
Introducción.....	16
Antecedentes	16
Problemática	18
Justificación.....	23
Objetivos	24
Objetivo General.....	24
Objetivos Específicos.	24
Alcance	24
Hipótesis	26
Capítulo II	27
Marco Metodológico	27
Estado del Arte.....	27
Planteamiento de la revisión de literatura preliminar.....	27
Criterios de inclusión y exclusión.....	27
<i>Grupo de control.</i>	29
Cadena de búsqueda.....	31
Proceso de selección.....	33
Resúmenes de los Estudios Primarios.....	36
Respuesta a las preguntas de investigación.	40
Resumen general y conclusión del estado del arte.	42
Metodología.....	42

Identificación de la problemática.	43
Definición de los objetivos de la solución.....	43
Diseño y desarrollo.	44
Demostración.....	44
Evaluación.	44
Comunicación.....	44
Marco Teórico.....	45
Red de categorías.....	45
Fundamentación Científica de la Variable Independiente.	46
Diseño de Software.	46
Arquitectura de Software.	47
Esquemas y Patrones de Software.....	47
Framework de desarrollo de software.....	50
Fundamentación Científica de la Variable Dependiente.	50
Tecnologías de la Información.	50
Sistemas de Información.....	51
Desarrollo de Sistemas de Información.	52
Integración, Entrega y despliegue continuos de sistemas de información.	52
Integración continua.....	52
Entrega continua.....	52
Despliegue continuo.	53
Capitulo III.....	55
Construcción de la Solución.....	55
Introducción.....	55
Integración Continua.	58
Despliegue Continuo.....	58
Selección de herramientas.....	59
GitLab.....	60
Jenkins.....	60
Docker.....	61
ArgoCD.....	62

Kubernetes	62
Nginx	63
Implementación	64
Integración Continua	65
Despliegue Continuo	67
Otras herramientas	69
Capitulo IV	72
Validación del Prototipo	72
Capitulo V	77
Conclusiones, Recomendaciones y trabajos futuros	77
Conclusiones	77
Recomendaciones	79
Trabajos Futuros	80
Bibliografía	81
ANEXOS	85

Índice de Tablas

Tabla 1: Matriz de congruencia Metodológica	25
Tabla 2: Artículos del grupo de control	30
Tabla 3: Trazabilidad de la cadena de búsqueda.....	32
Tabla 4: Estudios Primarios.....	34
Tabla 5: Lista de sistemas de información a cargo de la UTIC.....	56
Tabla 6: Comparación tiempos de ejecución	74

Índice de Figuras

Figura 1: Causa 1 y Efecto 1	19
Figura 2: La planificación de los proyectos se ve afectada	20
Figura 3: Usuarios no logran satisfacer sus necesidades.....	20
Figura 4: Sub causa 1 y 2.....	21
Figura 5: Usuarios no pueden acceder a los sistemas de información.....	22
Figura 6: Causa 4 y Efecto 4.....	23
Figura 7: Tendencia temporal de estudios	40
Figura 8: Fases de la metodología Design Science Research.....	43
Figura 9: Red de categorías correspondiente a la variable independiente	45
Figura 10: Red de categorías correspondiente a la variable dependiente	46
Figura 11: Relación entre la integración continua, entrega y despliegue	53
Figura 12: Arquitectura del prototipo de framework.....	58
Figura 13: Arquitectura del prototipo de framework con herramientas	59
Figura 14: Framework propuesto	64
Figura 15: Pipeline propuesto.....	66
Figura 16: Arquitectura Kubernetes	68
Figura 17: DevOps y Herramientas.....	70

Resumen

Las Unidades de Tecnologías de la Información y la Comunicación (UTIC) de las instituciones de educación superior (IES) enfrentan el desafío de incorporar la transformación tecnológica permanente y la forma en que se aplica en su seno para mejorar la calidad de la prestación de servicios de TI. Tras la revisión de literatura preliminar a fin de conocer los problemas más comunes que se presentan los proyectos de desarrollos de sistemas de información en las IES y el análisis exploratorio aplicado a la UTIC de la Universidad de las Fuerzas Armadas – ESPE se dejó en evidencia la gestión inadecuada del desarrollo, integración y despliegue de los sistemas de información. Este proyecto tiene como propósito estructurar un framework de desarrollo de software, para automatizar la integración, entrega y despliegue continuos de los sistemas de información de la Universidad de las Fuerzas Armadas – ESPE, siguiendo los lineamientos de la cultura DevOps. El framework desarrollado incorpora varias mejores prácticas, metodologías y estándares. Como resultado, tras la puesta en marcha del framework se obtuvo considerables mejoras el servicio prestado por la UTIC de la Universidad de las fuerzas Armadas – ESPE en lo concerniente a desarrollo y prestación de los de sistemas de información por lo que se tiende a la eficiencia y calidad.

Palabras clave:

- **FRAMEWORK**
- **INTEGRACIÓN CONTINUA**
- **DESPLIEGUE CONTINUO**
- **INSTITUCIONES DE EDUCACIÓN SUPERIOR**
- **DEVOPS**

Abstract

The Information and Communication Technology Units (UTIC) of Higher Education Institutions (HEI) face the challenge of incorporating the permanent technological transformation and the way it is applied within them to improve the quality of IT service delivery. After the preliminary literature review to know the most common problems that occur in information systems development projects in HEI's and the exploratory analysis applied to the UTIC of the Universidad de las Fuerzas Armadas - ESPE, the inadequate management of the development, integration and deployment of information systems was evidenced. The purpose of this project is to structure a software development framework to automate the continuous integration, delivery, and deployment of the information systems of the Universidad de las Fuerzas Armadas - ESPE, following the guidelines of the DevOps culture. The developed framework incorporates several best practices, methodologies, and standards. As a result, after the implementation of the framework, considerable improvements were obtained in the service provided by the UTIC of the Universidad de las Fuerzas Armadas - ESPE regarding the development and delivery of information systems, which tends to efficiency and quality.

Keywords:

- **FRAMEWORK**
- **CONTINUOUS INTEGRATION**
- **CONTINUOUS DEPLOYMENT**
- **HIGHER EDUCATION INSTITUTIONS**
- **DEVOPS**

Capítulo I

Introducción

Antecedentes

El proceso desarrollo de software tiene la finalidad de crear herramientas que faciliten las labores cotidianas de las personas dentro de un entorno determinado; no obstante, este proceso es complejo e implica la participación de diferentes actores (Martínez, 2015).

En un inicio, los sistemas de información eran básicos y limitados, pero con el pasar del tiempo, la mayor participación de las personas y el avance tecnológico, su complejidad se incrementó, de la mano de diversos problemas identificados en las fases de su proceso de construcción (Martínez, 2015).

Entre los problemas más comunes se encuentran la estimación inadecuada de costos del proyecto (Maida & Julián, 2015), requerimientos inexactos (Álvarez, 2010), carencia de metodologías específicas para el desarrollo (Medina & López, 2015) (Gamboa, 2018) (Esterkin, 2017), entre otros. Esta situación es confirmada en el informe denominado “Chaos 2020” de la firma internacional independiente de asesoría en investigación de TI The Standish Group, donde se establece que los factores principales que inciden para que un proyecto de desarrollo de software falle son: requisitos incompletos, falta de comunicación con el usuario, falta de recursos, falta de planificación y falta de gestión de TI (Henry, 2021).

El estudio de The Standish Group del año 2020 muestra también que del total de proyectos analizados (aproximadamente cincuenta mil entre grandes y pequeños), el 19% fueron calificados como proyectos fallidos, los proyectos considerados como exitosos alcanzan el 31% y el 50% restante son valorados como proyectos desafiantes, es decir, alcanzan un punto medio entre exitoso o fallido (Henry, 2021).

En resumen, lo que se busca es que el sistema de información sea entregado a tiempo, bajo presupuesto, pero, sobre todo, sin defectos ni errores; no obstante, cuando el usuario hace uso de este, la realidad indica todo lo contrario, un sistema plagado de errores; de hecho, uno de cada cuatro proyectos de software falla en la fase de entrega (Martínez, 2015).

Aunque las cifras detalladas no son nada alentadoras, para alcanzar ese nivel ha sido necesario recurrir a herramientas, metodologías o marcos de trabajo. Poco a poco se está alcanzando sistemas de información de mayor calidad y por ende el número de proyecto software considerados exitosos está aumentando de a poco, lo que al parecer indica que la industria de software está en el camino adecuado (Montero, 2015).

Dado que los problemas relacionados con el proceso de desarrollo de sistemas de información son extensibles a todos los ámbitos, este trabajo se enfoca en la problemática en torno al desarrollo y despliegue de los sistemas de información en el contexto de las Instituciones de Educación Superior (IES) del Ecuador, particularmente en las públicas, las cuales apoyadas en el uso de las Tecnologías de la Información y la Comunicación (TIC) buscan mejorar el servicio a su comunidad universitaria y cubrir sus necesidades.

Es común que la mayoría de las IES lleve a cabo el desarrollo de sus sistemas de información; no obstante, dicha actividad se caracteriza por sistemas de información aislados, duplicación de tareas, duplicación de información, baja calidad de los productos desarrollados, entre otros (Veloz, 2019). Estos problemas están relacionados con la carencia de: recursos económicos, recursos humanos, traza en la construcción, definición de procedimientos, aplicación de metodologías, documentación formal, entre otros. A esta situación se suma el hecho de que, las unidades de tecnologías de la información y comunicación de las IES no ven

como importante contar un equipo de desarrollo de sistemas de información bien consolidado (Gonzales, 2019).

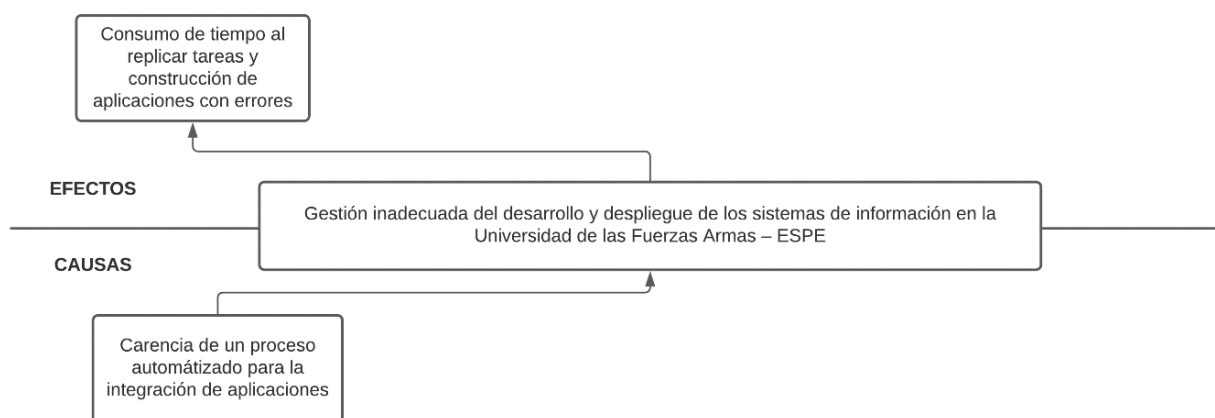
La Universidad de las Fuerzas Armadas - ESPE no es la excepción, ya que se identifica plenamente con la problemática descrita. Al ser la ESPE una universidad reconocida a nivel nacional y regional representa un contexto importante a ser estudiado. Dada la coyuntura de la Unidad de Tecnologías de la Información y la Comunicación (UTIC) de la ESPE, constituyó el caso de estudio ideal para esta investigación.

Problemática

En esta investigación se pretendió abordar la problemática: ***gestión inadecuada del desarrollo y despliegue de los sistemas de información en la Universidad de las Fuerzas Armadas – ESPE***. De aquí en adelante denominaremos a la Universidad de las Fuerzas Armadas – ESPE, Contexto Bajo Estudio (CBE). Para definir la problemática indicada, se realizó un análisis exploratorio mediante una entrevista¹, de la mano de una revisión básica de literatura. La entrevista esta fue aplicada al personal del área de desarrollo de la UTIC.

Lo resultados del estudio indican que una de las causas principales que conlleva a la problemática es la **carencia de un proceso automatizado para la integración de aplicaciones**. Al no contar con un proceso automatizado para el despliegue de aplicaciones, el equipo de desarrollo prepara el ambiente para que el aplicativo funcione; es decir, instalan las herramientas necesarias que garantizan el correcto funcionamiento del sistema (ver figura 1).

¹ Los insumos de la entrevista están accesibles en el Anexo 1

Figura 1***Causa 1 y Efecto 1***

Por ejemplo, para el caso de un proyecto de Backend², este es puesto en producción dependiendo de la tecnología utilizada para su desarrollo. Si se tratase de un proyecto de Frontend³ se genera una versión ejecutable y se lo pasa de manera manual a los servidores para su despliegue. Todas estas actividades representan tiempo lo cual afecta directamente la **planificación de los proyectos de desarrollo de software**.

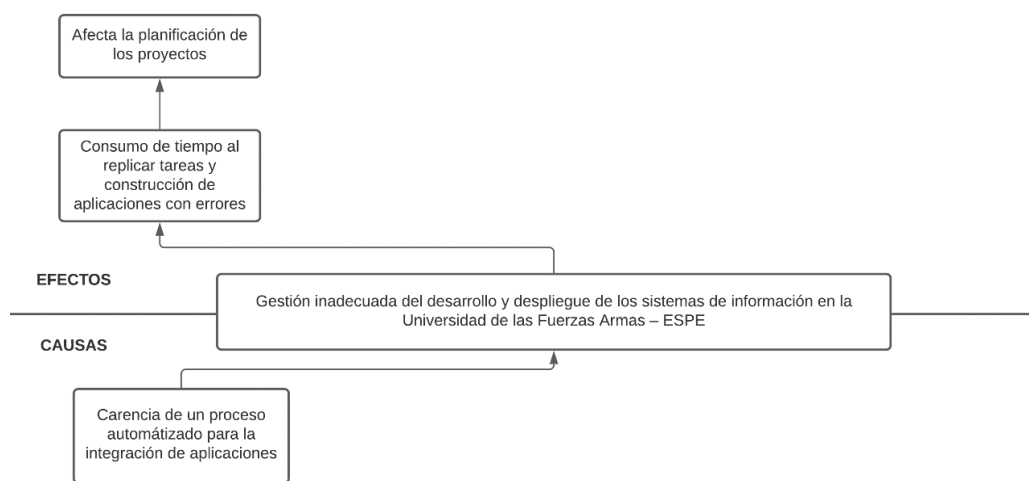
Al ser la integración de las aplicaciones un proceso manual, el personal suele cometer errores al momento de configurar los espacios para el despliegue de los sistemas (ver figura 2), lo cual afecta a gestión de otros proyectos, esto también está relacionado con el hecho de que el personal para el desarrollo es limitado.

² Backend: parte de un Sistema que se encarga de procesar información y manejar la lógica de los procesos.

³ Frontend: parte gráfica de un Sistema, se encarga de mostrar información al usuario de manera ordenada.

Figura 2

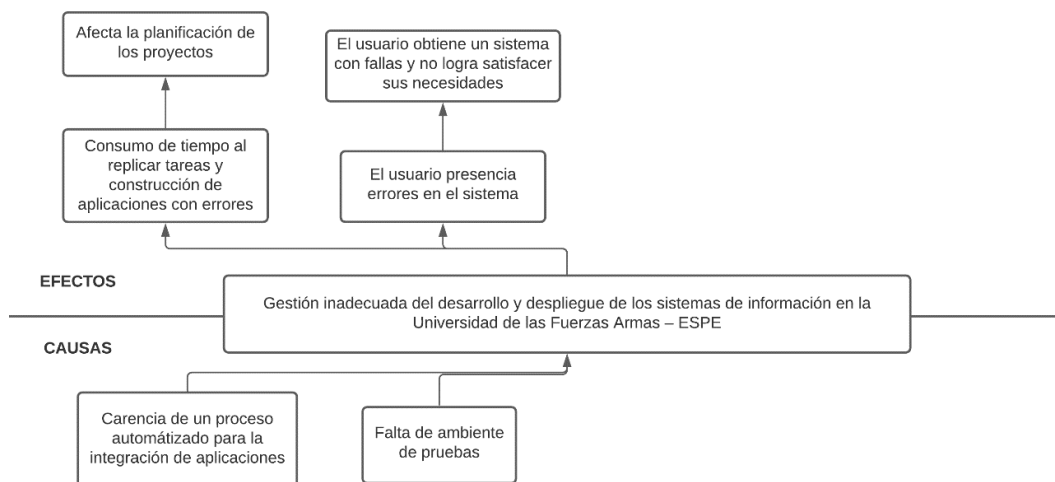
La planificación de los proyectos se ve afectada



La investigación reveló también la **carencia de un ambiente de pruebas** en el área de desarrollo de la UTIC, previo a que los sistemas sean puestos en producción. Actualmente cada desarrollador trabaja en su propio contexto, sin que haya la posibilidad de auditar la realización de pruebas estándar de software, esto ocasiona que **el sistema salga a producción plagado de errores**, lo cual es percibido por el usuario final, quien generalmente se muestra insatisfecho al utilizar un sistema de información que no satisfaga plenamente sus necesidades (ver figura 3).

Figura 3

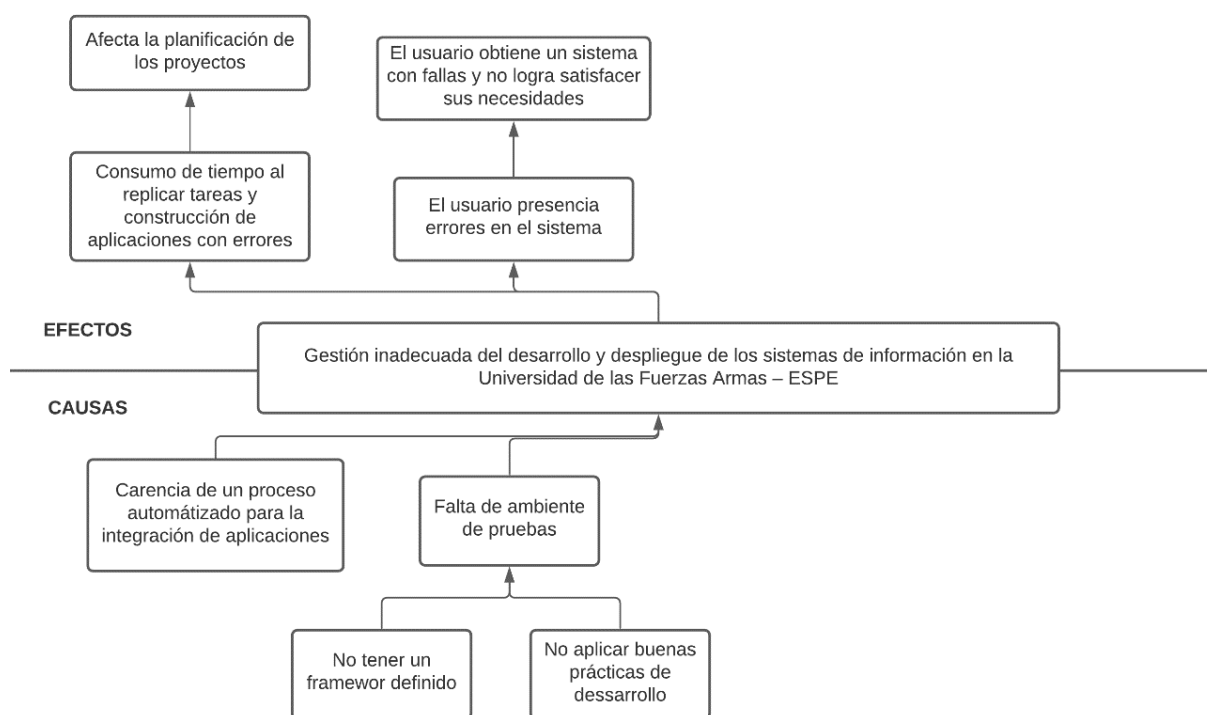
Usuarios no logran satisfacer sus necesidades.



Los miembros del equipo de desarrollo de la UTIC indican que la carencia de un ambiente de pruebas también va de la mano del hecho de que **no se tiene definido un marco de trabajo** (Medina & López, 2015) y a la vez que **no se ha definido un proceso estándar de desarrollo de sistemas de información** (ver figura 4).

Figura 4

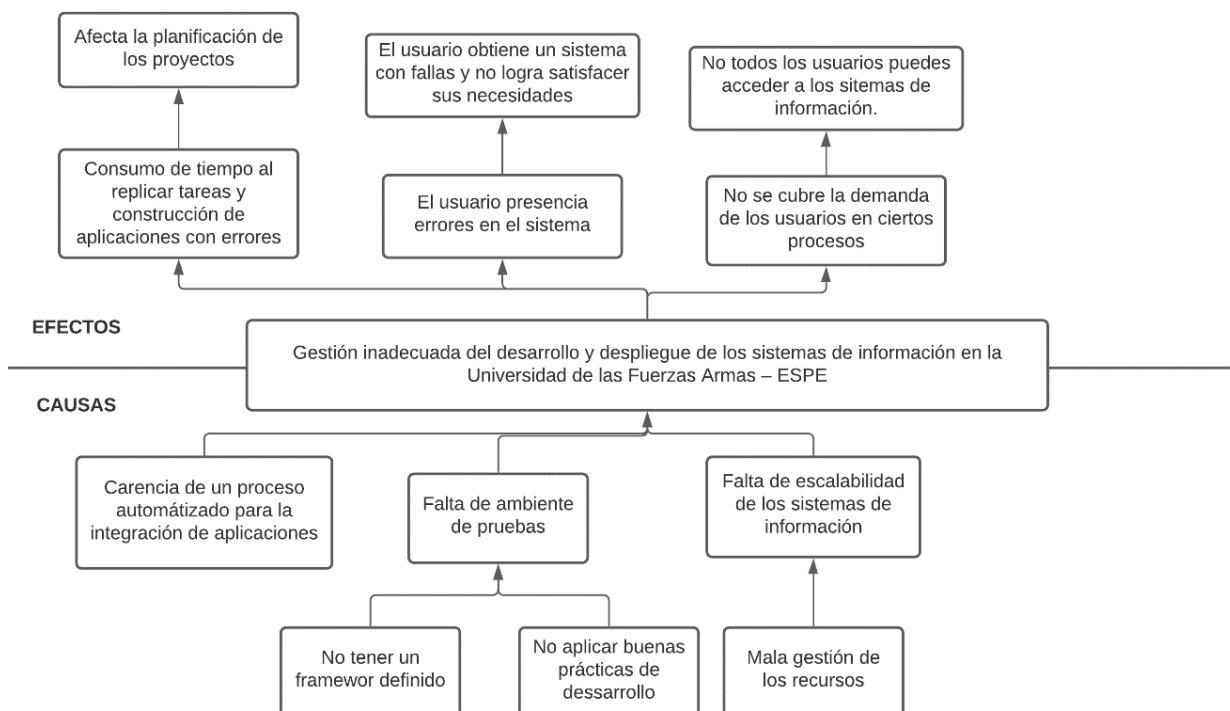
Sub causa 1 y 2



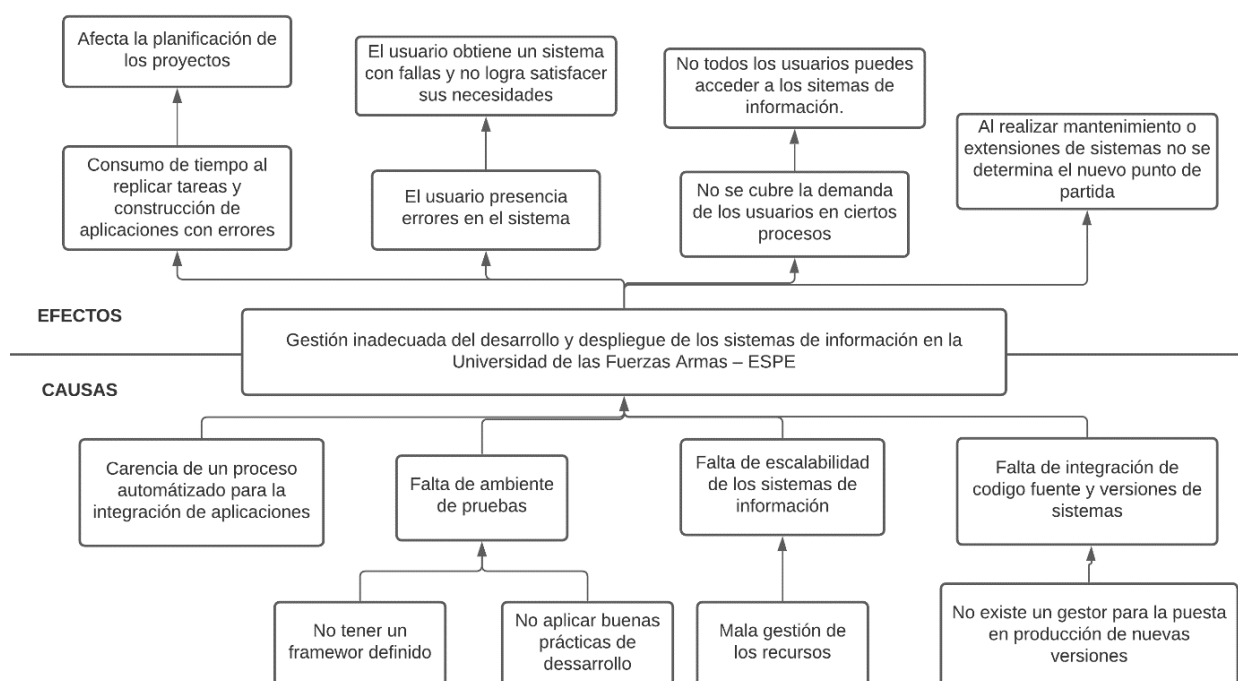
Adicionalmente se identificó que los sistemas de información no son escalables, debido a la gestión inadecuada de los recursos de la infraestructura existente, lo que ocasiona que no se cubra la demanda de los usuarios. Actualmente, el área de infraestructura de la Universidad pone a disponibilidad del área de desarrollo un único servidor donde se despliegan los sistemas. Para satisfacer la demanda, se aumentan los recursos computacionales, que no siempre son suficientes, haciendo que no todos los usuarios puedan acceder a los sistemas de información (ver figura 5).

Figura 5

Usuarios no pueden acceder a los sistemas de información



Otro de los resultados del estudio indica que los problemas comunes relacionados con el desarrollo y despliegue de los sistemas de información son el versionamiento y su integración (Retamal, 2019), es decir, **hay una falta de integración del código fuente y de nuevas versiones de los sistemas**, esto debido a que **no existe un gestor para la puesta en producción de nuevas versiones**, por ende, se desconoce cuáles de estas se encuentran en producción. Como consecuencia, **cuando se requiere realizar el mantenimiento o extensión de un sistema no se puede determinar el nuevo punto de partida** (ver figura 6).

Figura 6**Causa 4 y Efecto 4**

En síntesis, el árbol de problemas mostrado en la figura 6 representa el resultado del estudio de la situación actual.

Justificación

Los sistemas de información cumplen un papel fundamental en las organizaciones (Estupiñán & Orjuela, 2019), ya que aportan al control y seguimiento administrativo de los diferentes procesos que se llevan a cabo en su seno. No obstante, ante lo encontrado en el CBE, queda claro que se precisa de un mecanismo que permita orquestar las actividades inherentes a sus procesos sustantivos y facilite la integración y despliegue de los sistemas de información durante su desarrollo. Esto motivó al presente proyecto, el cual fue guiado por los objetivos que se detallan a continuación.

Objetivos

Objetivo General.

Estructurar un framework de desarrollo de software, para automatizar la integración, entrega y despliegue de las aplicaciones TI de la Universidad de las Fuerzas Armadas – ESPE, siguiendo los lineamientos de la cultura DevOps.

Objetivos Específicos.

- i. Definir la problemática en torno al desarrollo y despliegue de los sistemas de información en la Universidad de las Fuerzas Armadas – ESPE, a través del estudio de la situación actual basado en un análisis exploratorio
- ii. Llevar a cabo un estudio de factibilidad de la sistematización del desarrollo de sistemas TI en una Institución de Educación Superior, por medio de una revisión de literatura preliminar.
- iii. Implementar un framework de desarrollo de aplicaciones TI, siguiendo los lineamientos de la cultura DevOps.
- iv. Validar el framework de desarrollo de aplicaciones TI en la Unidad de Tecnologías de la Información y la Comunicación de la Universidad de las Fuerzas Armadas – ESPE, en base a un prototipo funcional.

Alcance

Este proyecto tiene como alcance estructurar el prototipo de un framework de desarrollo de software para automatizar la integración, entrega y despliegue de los sistemas de información de la Universidad de las Fuerzas Armadas - ESPE, siguiendo lineamientos de la cultura DevOps.

Para este propósito se realizó una revisión de literatura, donde se encontró información relacionada con problemas comunes que se presentan en proyectos de desarrollo de software.

Con la finalidad de tener una guía en el alcance del presente trabajo, se plantea una matriz de congruencia metodológica, la cual nos permite plantear preguntas de investigación relacionadas a cada uno de los objetivos específicos.

Tabla 1

Matriz de congruencia Metodológica

OBJETIVO ESPECÍFICO	PREGUNTAS DE INVESTIGACIÓN
<p>i. Definir la problemática en torno al desarrollo y despliegue de los sistemas de información en la Universidad de las Fuerzas Armadas – ESPE, a través del estudio de la situación actual basado en un análisis exploratorio.</p>	<p>RQ1. ¿El desarrollo de los sistemas de información de la Universidad de las Fuerzas Armadas - ESPE se realiza dentro de un marco de integración de aplicaciones?</p> <p>RQ2. ¿Los sistemas de información desarrollados en la Universidad de las Fuerzas Armadas - ESPE responden adecuadamente a las necesidades de los usuarios?</p> <p>RQ3. ¿Qué elementos son considerados para el aseguramiento de la calidad de los sistemas de información de la Universidad de las Fuerzas Armadas – ESPE?</p>
<p>ii. Llevar a cabo un estudio de factibilidad de la sistematización del desarrollo de sistemas TI en una Institución de Educación Superior, por medio de una revisión de literatura preliminar.</p>	<p>RQ4 ¿Con que frecuencia se presentan estudios relacionados con la sistematización de la integración y despliegue continuos de sistemas de información en las instituciones de educación superior?</p> <p>RQ5 ¿Qué mecanismos (métodos, marcos modelos, herramientas, guías, técnicas) son propuestos para sistematizar la integración y despliegue continuos de sistemas de información en las instituciones de educación superior?</p> <p>RQ6 ¿Cuáles son los retos más comunes relacionados con la sistematización de la integración y despliegue continuos de sistemas de información que deben superar las instituciones de educación superior?</p>

	<p>RQ7 ¿Cuáles son los aspectos más relevantes que se debe tener en cuenta para estructurar un marco de trabajo que permita sistematizar la integración y despliegue de sistemas de información en una institución de educación superior?</p>
<p>iii. Implementar un framework de desarrollo de aplicaciones TI, siguiendo los lineamientos de la cultura DevOps.</p>	
<p>iv. Validar el framework de desarrollo de aplicaciones TI en la Unidad de Tecnologías de la Información y la Comunicación de la Universidad de las Fuerzas Armadas – ESPE, en base a un prototipo funcional.</p>	<p>RQ8: ¿El desarrollo de los sistemas de información de la Universidad de las Fuerzas Armadas - ESPE se realiza dentro de un marco de integración de aplicaciones?</p> <p>RQ9: ¿Los sistemas de información desarrollados en la Universidad de las Fuerzas Armadas - ESPE responden adecuadamente a las necesidades de los usuarios?</p> <p>RQ10: ¿Las herramientas implementadas responden adecuadamente con las necesidades de la Unidad de Tecnologías de la Información de la Universidad de las Fuerza Armadas - Espe?</p>

Hipótesis

Un framework de desarrollo de software permite la automatización de los procesos de integración, entrega y despliegue de sistemas de información.

Capítulo II

Marco Metodológico

En este capítulo se detalla el estudio del estado del arte, la metodología guía del proyecto de titulación y el marco teórico a través de la formulación de la red de categorías de las variables de investigación.

Estado del Arte

El estudio del estado del arte se realizó por medio de una revisión de literatura preliminar (Kitchenham, Budgen, & Brereton, 2015), con el propósito de validar la factibilidad de la propuesta de un marco de trabajo para la integración y despliegue de sistemas de información en una institución de educación superior. Este estudio fue realizado en la base de datos digital SCOPUS.

Planteamiento de la revisión de literatura preliminar.

Una vez definido los antecedentes, la problemática y el tema del presente proyecto, se estableció el objetivo de búsqueda y se plantearon las preguntas de investigación, los cuales corresponden con el objetivo específico 2 y a las preguntas de investigación 4, 5, 6, y 7.

Criterios de inclusión y exclusión.

Parte importante en este capítulo es la definición de los criterios de inclusión y exclusión (CIE), características de interés que se buscan en los artículos dentro del proceso de revisión de literatura preliminar. Dichos criterios determinan si un estudio es considerado (o no), para la investigación. A continuación, se detallan los criterios de inclusión y exclusión.

- *CI1⁴*: Estudios en donde se detalle el proceso y las estrategias empleadas para sistematización de la integración y despliegue continuos de los sistemas de información en las instituciones de educación superior.
- *CI2*: Estudios en donde se especifiquen los campos o áreas considerados en el proceso de sistematización de la integración y despliegue continuos de sistemas de información de las universidades.
- *CI3*: Estudios donde se detallen los mecanismos utilizados para sistematizar la integración y despliegue continuos de sistemas de información en las instituciones de educación superior.
- *CI4*: Estudios en los que se evidencian los retos que deben superar las instituciones de educación superior en torno a la sistematización de la integración y despliegue continuos de sistemas de información.
- *CI5*: Estudios de revisión de literatura donde se mencionen marcos de trabajo y mejores prácticas para la integración y despliegue continuo de sistemas de información en instituciones de educación superior.

Serán excluidos:

- *CE1⁵*: Estudios que presenten soluciones convencionales sin la guía de un paradigma específico.
- *CE2*: Estudios en los cuales se presenten soluciones de integración y despliegue continuo fuera del contexto de las instituciones de educación superior.

⁴ CI: Criterios de inclusión.

⁵ CE: Criterios de exclusión.

- *CE3:* Estudios publicados antes de enero de 2017; o publicados en conferencias o revistas que aparecen indexados en base de datos digitales relevantes como Scopus, pero no en portales accesibles que publican los indicadores científicos de revistas y congresos tales como SCImago Journal & Country Rank (SJR) y Journal Citation Report (JCR).
- *CE4:* Estudios que hagan referencia al uso de herramientas de software propietario; es decir, a herramientas que necesitan de un pago para su uso.

Grupo de control.

El grupo de control (CG) es un conjunto de estudios que responden fielmente a los criterios de inclusión y que son seleccionados en consenso por los investigadores que realizan el estudio del estado del arte. El CG tiene como propósito guiar la revisión de literatura, pero principalmente representa la fuente de los términos que servirán para conformar la cadena de búsqueda.

Para seleccionar los estudios que forman parte del GC cada investigador propone artículos científicos referentes al tema planteado y en un grupo de discusión se define el GC. En la tabla 2 se detallan los estudios considerados para el GC y los términos relevantes que fueron extraídos de cada uno y que sirvieron como insumo para conformar la cadena de búsqueda.

Tabla 2

Artículos del grupo de control

Código	Título	Términos relevantes
CS1	Towards Development a Novel Framework of Web-Based Systems Quality Engineering by the Integration between Information Systems and Software Engineering Theories: Context of Higher Education	Information systems, systems, higher education, education, academic information system, integration, framework, management tools
CS2	Innovation Strategy Services Delivery: An Empirical Case Study of Academic Information Systems in Higher Education Institution	Information systems, software application, higher education, education, academic information system, university, integration
CS3	Pioneering the Automation of Internal Quality Assurance System of Higher Education (IQAS-HE) Using DevOps Approach	
CS4	Research on the Application of DevOps in the Smart Campus of Colleges and Universities	legacy systems, systems, universities, smart campus, integration, framework, continuous deployment, management tools
CS5	DevOps in Practice for Education Management Information System at ECNU	Information systems, systems, management systems, education,

university, integration, framework,
continuous deployment

Los términos relevantes de los estudios del GC fueron identificados mediante el uso de la herramienta de minería de datos NVivo, ya que esta herramienta identificar relaciones que de manera manual no son posibles realizarlas, además, permite la generación de evidencia de los procesos de minería realizados. La presente investigación es de tipo cualitativa, por lo que el uso de NVivo aportó significativamente en el proceso de búsqueda y frecuencia de términos en los estudios de control. Los términos identificados permitieron más adelante la conformación de la cadena de búsqueda.

Cadena de búsqueda.

La cadena de búsqueda recoge varios de los términos relevantes y permite ejecutar consultas en las diferentes bases de datos digitales con la finalidad de encontrar artículos que se relacionen con la problemática de este proyecto.

En base al objetivo de búsqueda y a los términos relevantes mostrados en la tabla 2, se establecieron 4 contextos donde se agrupan los términos y sus sinónimos, los cuales se indican a continuación

- Objeto: A donde va dirigida la acción.
- Contexto: Donde se desarrolla el estudio.
- Propósito: Describe la razón de ser del estudio.
- Mecanismos: Herramientas implementadas.

Para que una cadena de búsqueda sea considerada como ideal, se consideró que cumplan las siguientes características:

- El número de artículos obtenidos es adecuado y manejable para el análisis.
- Los artículos obtenidos se relacionan con la temática y están alineados con los criterios de inclusión y exclusión.
- Dentro de los estudios obtenidos por la cadena, aparecen la mayoría de los estudios del grupo de control.

Una vez identificados y agrupados los términos relevantes, se procedió con el pilotaje de la cadena de búsqueda el cual consiste en un proceso iterativo de ensayo - error - corrección. Este proceso permite probar distintas combinaciones de los términos relevantes de cada uno de los contextos establecidos, hasta definir la cadena de búsqueda ideal que permita identificar los artículos científicos más relevantes para el estudio. En la tabla 3 se particularizan las diferentes cadenas de búsqueda que fueron aplicadas en la base de datos digital SCOPUS como parte del proceso de pilotaje.

Tabla 3

Trazabilidad de la cadena de búsqueda

Cadena de Búsqueda	Número de Artículos Obtenidos
ALL (("information systems") AND ("higher education" OR "universities") AND (" integration" OR "deployment") AND ("framework"))	135737 artículos obtenidos
ALL (("information systems") AND ("higher education" OR "universities") AND (" integration") AND ("framework"))	121252 artículos obtenidos

ALL (("information systems") AND ("higher education" OR "universities")) AND ("continuous integration") AND ("framework")) 410 artículos obtenidos

ALL (("information systems") AND ("higher education" OR "universities")) AND ("continuous integration") AND ("framework" OR "model")) 623 artículos obtenidos

ALL (("information systems" OR "legacy systems") AND ("higher education" OR "universities")) AND ("continuous integration") AND ("framework" OR "model")) 633 artículos obtenidos

ALL (("academic information systems" OR "legacy systems" OR "information systems") AND ("higher education" OR "universities" OR "college") AND ("continuous integration" OR "continuous deployment") AND ("framework" OR "model" OR "architecture")) AND (LIMIT-TO (PUBYEAR , 2022) OR LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017)) 649 artículos obtenidos

Nota: La tabla muestra la trazabilidad de la cadena de búsqueda basándonos la cantidad de artículos obtenidos semejantes a los seleccionados en el grupo de control.

En esta tabla observan los resultados de cada una de las fases del proceso de pilotaje, cada iteración permitió mejorar la calidad de los artículos resultantes. Finalmente se seleccionó la cadena de búsqueda que arrojó 649 estudios que se orientan al objetivo de la investigación, estos estudios pasaron por un proceso de selección y se los utilizó para responder a las preguntas de investigación planteadas.

Proceso de selección.

Este proceso incluye la selección de: Estudios Candidatos, Estudios Relevantes y Estudios Primarios.

Los Estudios Candidatos se obtienen de aplicar la cadena de búsqueda seleccionada en la base digital, en este caso se obtuvieron 649 Estudios Candidatos.

Los estudios seleccionados son aquellos que resultan de la lectura de los estudios candidatos a nivel del título, resumen y palabras clave. Cada investigador realiza esta revisión teniendo en mente los CIE y en consenso se determinan los Estudios Relevantes. Para este caso se encontraron únicamente 4 Estudios Relevantes, lo que indica que hay muy poca investigación en torno a la problemática de la integración y el despliegue de los sistemas de información de las instituciones de educación superior. Por lo tanto, se intentó realizar un proceso de selección de bola de nieve hacia delante y hacia atrás, pero no se obtuvo ningún artículo adicional (Wohlin, 2014).

El proceso de selección de los estudios primarios inicia con la descarga del texto completo de los estudios relevantes. A paso seguido, cada investigador revisa exhaustivamente el texto completo de todos y cada uno de los estudios relevantes. Finalmente, en consenso se seleccionan los estudios primarios. En este caso, los mismos estudios relevantes pasaron a ser estudios primarios, los cuales, sumados a los estudios del grupo de control, dieron un total de 9 Estudios Primarios (ver tabla 4).

Tabla 4

Estudios Primarios

Código	Título	Cita
EP1	s-SCMM: A simplified software configuration and management model for software services in public university	(Mohamad, Yahaya, Deraman, & Mansor, 2020)
EP2	Development and Operations - The Modern Paradigm of the Work of IT Project Teams	(Veres, y otros, 2019)

EP3	An agile-devops reference architecture for teaching enterprise agile	(Ghantous & Gill, 2019)
EP4	Enable cloud DevOps approach for industry and higher education	(Airaj, 2017)
EP5	Towards Development a Novel Framework of Web-Based Systems Quality Engineering by the Integration between Information Systems and Software Engineering Theories: Context of Higher Education	(Ibrahim Fadhel, y otros, 2020)
EP6	Innovation Strategy Services Delivery: An Empirical Case Study of Academic Information Systems in Higher Education Institution	(Purba & Panday, 2017)
EP7	Pioneering the Automation of Internal Quality Assurance System of Higher Education (IQAS-HE) Using DevOps Approach	(Taryana, Setiawan, Fadli, & Murdyantoro, 2017)
EP8	Research on the Application of DevOps in the Smart Campus of Colleges and Universities	(Pingrong, Xiaoquan, & Junqin, 2021)
EP9	DevOps in Practice for Education Management Information System at ECNU	(Yang, y otros, 2020)

Nota: La tabla muestra los artículos seleccionados por los investigadores para formar del grupo de estudios primarios.

En esta tabla se puede observar los títulos de los artículos seleccionados como estudios primarios, estos estudios fueron revisados de manera completa por cada uno de los investigadores con la finalidad de dar respuesta a las preguntas de investigación y concluir el estudio de factibilidad.

Resúmenes de los Estudios Primarios.

EP1: s-SCMM: A simplified software configuration management model for software services in public university.

Este artículo habla sobre la gestión de la configuración de software que se usa en las universidades públicas de Malasia. Luego una serie de entrevistas aplicadas a los profesionales de dichas instituciones se determinó que no se lleva una gestión de configuración de software, debido a que son pocas las herramientas que se adaptan a las necesidades de este proceso. Por otro lado, el artículo propone un modelo de gestión de la configuración de software simplificado denominado: “s-SCMM” el cual consta de tres componentes fundamentales: herramientas, proceso y factor humano. El modelo se apoya en las herramientas y en el control humano para llevar a cabo los procesos de: gestión del cambio, control de versiones, construcción de sistemas y la gestión de puesta en producción. Este modelo promete convertirse en una plataforma que ayuda a la implementación total de la gestión de configuración de software de una manera sencilla.

EP2: Development and Operations - The Modern Paradigm of the Work of IT Project Teams

El artículo se desarrolla en la Universidad Nacional Politécnica de Lviv, se analiza a detalle los sistemas de gestión de instituciones superiores y se estudia el concepto de responsabilidad compartida siendo la cultura de DevOps el paradigma central de los sistemas desarrollados en dicha universidad. Luego del análisis los investigadores resaltan el uso de DevOps para maximizar y optimizar la productividad de los desarrolladores y a su vez aumentar la confiabilidad del software elaborado debido a que en la implantación de DevOps para sus desarrollos se puso en evidencia la fiabilidad, escalabilidad, el trabajo en conjunto optimizado y la seguridad.

EP3: An agile-devops reference architecture for teaching enterprise agile

La problemática de este artículo se basa en cómo simular y establecer un entorno Agile-DevOps ya sea en instituciones superiores o en ambientes empresariales, el artículo se apoya del caso de enseñanza de la Universidad Tecnológica de Sídney en Australia y exponen cinco modelos que hacen uso de prácticas DevOps. La investigación realiza claros ejemplos del papel que desempeña el gerente de proyecto, el analista de negocios, los desarrolladores y el ingeniero de DevOps para que de esta manera se tenga una idea clara de la implementación Agile-DevOps en una institución.

EP4: Enable cloud DevOps approach for industry and higher education

La investigación evidencia la reducción de tiempo en el desarrollo de proyectos y el aumento en la tasa de éxito en la entrega de software haciendo uso de la metodología DevOps propuesta en este artículo, además asegura que la adaptación de nuevas tecnologías tanto en la industria como en el mundo académico es sumamente importante debido a la rápida evolución del software por lo que en varias instituciones de educación superior del Reino Unido e incluso algunas del continente Africano ya han adoptado la implementación de DevOps para sus sistemas de software. La Universidad de Marrakech ofrece cursos de posgrado de computación en la nube y han podido concluir que la metodología DevOps garantiza producir servicios más innovadores a menor precio y de mayor calidad.

EP5: Towards Development a Novel Framework of Web-Based Systems Quality

Engineering by the Integration between Information Systems and Software Engineering

Theories: Context of Higher Education

El artículo resalta las fallas de los sistemas de información, menciona que este tema es de interés en la actualidad. La ingeniería de software es la encargada, de cierto modo, de tratar

disminuir la tasa de usuarios insatisfechos y la tasa de fallas en los sistemas de información, ya que esta tasa de fallas es un indicador que permite determinar si un sistema es exitoso o no en determinados contextos. El estudio se enfoca en el contexto de la Universidad de Yemen, donde se existe la problemática de que los usuarios se encuentran insatisfechos con el sistema de información de actual, por tal razón, los autores, proponen un marco de calidad basado en la norma ISO 25010 con el cual se pueda medir el éxito y la calidad de los sistemas de información basados en la web de la mencionada institución de educación superior.

EP6: Innovation Strategy Services Delivery: An Empirical Case Study of Academic Information Systems in Higher Education Institution

El presente estudio resalta el aporte que ha realizado las tecnologías de la información y comunicación en el contexto de la transformación de la sociedad a nivel mundial. Menciona que las instituciones de educación superior deberían ser las encargadas de emprender este tipo de transformación con la finalidad de que pueden servir a las diferentes partes de la sociedad. En primera instancia se debería iniciar con la transformación de los sistemas de información académica existentes, ya que estos aun no son migrados a las nuevas tecnologías. De manera general el estudio se enfoca en una investigación que permita describir las mejores técnicas para una mejor prestación de servicios en las instituciones de educación superior.

EP7: Pioneering the Automation of Internal Quality Assurance System of Higher Education (IQAS-HE) Using DevOps Approach

El artículo menciona la importancia del desarrollo de software y la gestión de estos para garantizar la calidad. Existen otros procesos dentro del desarrollo de software que son importantes como el reutilizar código y la automatización de los lanzamientos de nuevas versiones. De manera específica, este estudio presenta la automatización del desarrollo

mediante el uso de herramientas de gestión, estas son utilizadas por las instituciones de educación superior con la finalidad de mejorar la calidad, utilizando un enfoque DevOps. Finalmente se expone la colaboración que existe entre diferentes herramientas para mejorar los procesos de integración continua en el contexto expuesto anteriormente.

EP8: Research on the Application of DevOps in the Smart Campus of Colleges and Universities

El estudio menciona que los sistemas antiguos suelen coexistir con los nuevos, por tal razón es importante que los sistemas antiguos puedan ser actualizados lo que a su vez requiere gran cantidad de mano de obra y recursos materiales. Es ahí en donde nace la idea de un campus, con la finalidad de satisfacer las necesidades de los usuarios. El estudio propone una plataforma DevOps donde se pueden utilizar diferentes herramientas y otras tecnologías, esto con la finalidad de proporcionar un canal para que los equipos de desarrollo y operaciones reduzcan los problemas de comunicación y puedan desarrollar actividades de manera ágil y automatizada. Finalmente se menciona que la implementación de dicha plataforma mejora la calidad de la entrega y promueve continuamente el desarrollo de software, permitiendo la construcción de campus inteligentes como se mencionaba en un inicio.

EP9: DevOps in Practice for Education Management Information System at ECNU

Los sistemas educativos de la Universidad Normal del Este de China presentaban varias limitaciones debido a la alta concurrencia y el desarrollo obsoleto. Este artículo establece un nuevo marco de trabajo basado en DevOps, además de sugerir herramientas que faciliten la integración y el despliegue continuo que garantizan la eficiencia del framework. Luego de la implementación de DevOps en los sistemas educativos de dicha universidad pudieron concluir

que se logró mejorar la eficiencia del trabajo y al mismo tiempo asegurar la confiabilidad, disponibilidad y escalabilidad de todos los sistemas software en los que se implementó.

A continuación, se hizo uso de la síntesis de cada uno de los estudios primarios para responder a las preguntas de investigación planteadas para esta revisión sistemática de literatura.

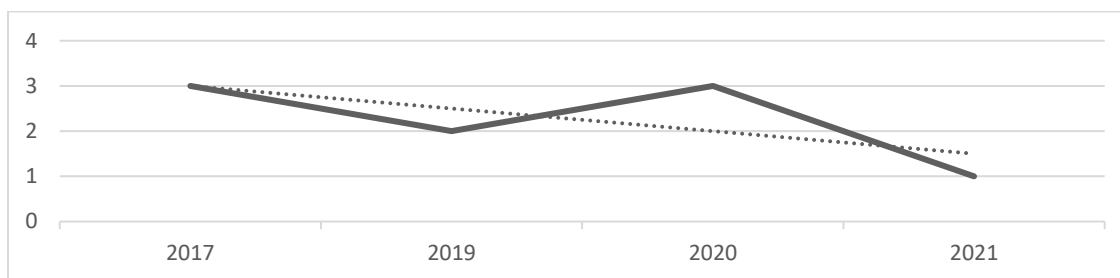
Respuesta a las preguntas de investigación.

¿Con que frecuencia se presentan estudios relacionados con la sistematización de la integración y despliegue continuos de sistemas de información en las instituciones de educación superior?

En el periodo detallado en los criterios de inclusión, es decir, dentro del periodo 2017 – 2021, los estudios primarios se distribuyen de acuerdo a como se muestra en la figura 7.

Figura 7

Tendencia temporal de estudios



La figura resalta la tendencia a la baja de propuestas, sin embargo, esto no quiere decir que la sistematización de la integración y despliegue no sea un tema de interés, si no que al parecer de forma generalizada dentro de las instituciones de educación superior no se le da la relevancia del caso.

¿Qué mecanismos (métodos, marcos modelos, herramientas, guías, técnicas) son propuestos para sistematizar la integración y despliegue continuos de sistemas de información en las instituciones de educación superior?

Existen soluciones que facilitan y mejoran los procesos de integración y despliegue continuos cuando se trata de construir sistemas de información, una de las propuestas se enfoca en la adopción del paradigma DevOps (Veres, y otros, 2019), (Ghantous & Gill, 2019), (Airaj, 2017), (Taryana, Setiawan, Fadli, & Murdyantoro, 2017), (Pingrong, Xiaoquan, & Junqin, 2021), (Yang, y otros, 2020). Esta solución está ligada al uso de herramientas que permiten automatizar los procesos de integración y despliegue de los sistemas de información, dichas herramientas también permiten que los equipos de desarrollo y operaciones trabajen de mejor manera, a su vez esto se ve reflejado en la disminución de tiempos de espera entre la ejecución de actividades de las áreas mencionadas.

Otra solución es la automatización del desarrollo mediante el uso un modelo de gestión de la configuración de software simplificado (Mohamad, Yahaya, Deraman, & Mansor, 2020), este modelo se apoya en el uso de herramientas y el factor humano para llevar a cabo procesos de gestión de cambio, control de versiones, construcción de sistemas y la gestión de la puesta en producción de estos.

¿Cuáles son los retos más comunes relacionados con la sistematización de la integración y despliegue continuos de sistemas de información que deben superar las instituciones de educación superior?

Al momento de poner en marcha las soluciones mencionadas los retos más comunes están relacionados con el conocimiento de los encargados de desarrollo sobre el paradigma DevOps, por tal razón es necesario que exista un proceso de transformación (Purba & Panday,

2017), este proceso debe ser adoptado en primera instancia por las instituciones de educación superior. De esta manera se puede poner en práctica los conocimientos de DevOps al momento de desarrollar sistemas dentro de las instituciones educativas.

Resumen general y conclusión del estado del arte.

A través del estudio del estado del arte se identificaron mecanismos y características comunes en los estudios primarios. Lo que más resaltó en dichos estudios es la utilización de DevOps como paradigma para el mejoramiento de los procesos de integración y despliegue continuos.

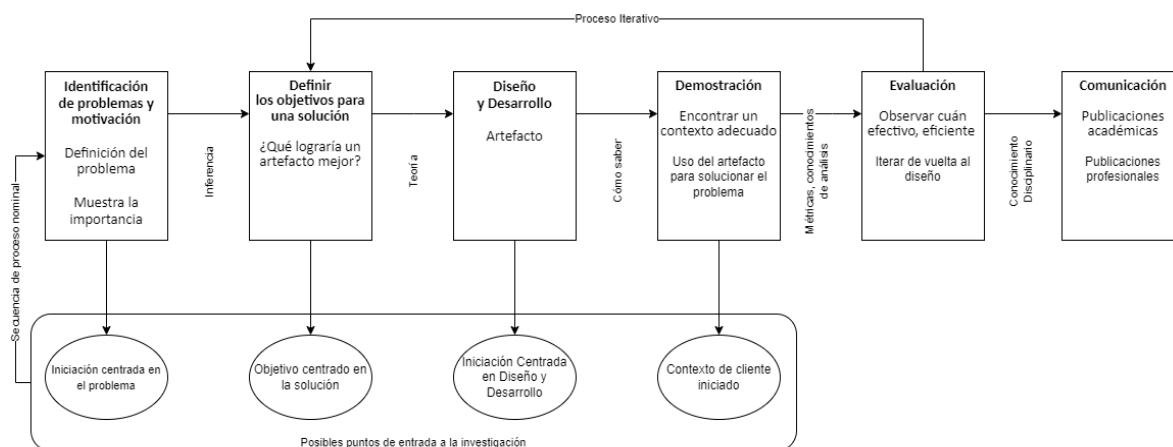
Por lo tanto, se determinó la factibilidad del presente estudio, el cual busca adquirir un mayor conocimiento acerca del uso del paradigma DevOps y con ello estructurar un framework de desarrollo de software que permita automatizar los procesos de integración, entrega y despliegue continuos en el proceso de desarrollo de los sistemas de información de la Universidad de las Fuerzas Armadas – ESPE.

Metodología.

Para el desarrollo del presente proyecto se consideró la aplicación de una metodología, la cual fue elegida teniendo en cuenta el ámbito de la problemática. Al encontrarse el proyecto en el ámbito de las tecnologías de la información, la metodología elegida fue Design Science Research (DSR), la cual permite a los investigadores obtener resultados innovadores y un profundo conocimiento del contexto en el que se lo esté aplicando (Thuan, Drechsler, & Antunes, 2019). Esta metodología es iterativa, es decir, es posible regresar a pasos anteriores si no se cumple con el propósito deseado (Wieringa, 2014). La figura 8 presenta las fases de la metodología Design Science Research.

Figura 8

Fases de la metodología Design Science Research.



Nota: Diagrama tomado de (Wieringa, 2014).

Cada una de las fases de la metodología seleccionada se definen a continuación.

Identificación de la problemática.

Esta fase inicial de la metodología tiene como objetivo definir la problemática a la que se dará solución en el presente proyecto, para ello se llevó a cabo un estudio de situación actual mediante una entrevista como técnica de levantamiento de información y también una revisión de literatura, esto con la finalidad de encontrar relación entre los problemas encontrados en la Universidad de las Fuerzas Armadas – ESPE y otros ámbitos a nivel mundial.

Definición de los objetivos de la solución.

En esta fase se describieron y delimitaron los objetivos específicos, para ellos se realizó una matriz de congruencia metodológica, en ella se describen los lineamientos a seguir para el cumplimiento de cada uno de los objetivos específicos. De manera específica el planteamiento de preguntas de investigación (RQ) ayudan a los investigadores a direccionar el camino a seguir para llegar a un producto final.

Diseño y desarrollo.

En esta fase, mediante el diseño de un prototipo, se propone un modelo conceptual, se estiman recursos tecnológicos y humanos, además de comportamientos esperados. Finalmente, el modelo conceptual será llevado a la práctica y puesto en funcionamiento.

Demostración.

Esta fase tiene como objetivo poner a prueba el funcionamiento del prototipo. En primer lugar, se validaron las funciones básicas, luego se utilizaron métodos de simulación para poner en evidencia el correcto funcionamiento del prototipo final.

Evaluación.

Esta fase tiene como objetivo validar el funcionamiento del prototipo, además, esta etapa permite encontrar fallas y planificar actividades para la corrección de estas. En este momento es donde queda en evidencia que la metodología es un proceso iterativo y permite volver a etapas de diseño o desarrollo según se requiriera.

Comunicación.

Una vez que se hayan logrado los resultados planteados, se presenta el modelo final de prototipo con sus respectivas conclusiones. El uso de esta metodología trae diferentes ventajas en el área de investigación y en el área de desarrollo del proyecto, entre ellas podemos listar las siguientes:

- Posibilidad de generar varias alternativas de diseño en el ciclo iterativo.
- Emplear métodos rigurosos para construir y evaluar el prototipo.
- Detección temprana de fallas y errores.
- Fomenta la investigación para garantizar la innovación del prototipo.

Como se mencionó al inicio de esta sección, cada una de estas fases está relacionada con los objetivos planteados para el desarrollo del proyecto.

Marco Teórico.

Red de categorías.

Con el propósito de encontrar coherencia en el fundamento teórico del presente estudio, se procedió a estructurar una red de categorías tanto para la variable dependiente como para la variable independiente. Para ello, se partió de la hipótesis de trabajo planteada en el capítulo anterior. Dichas variables quedan establecidas de la siguiente manera:

VI⁶: Framework de desarrollo de software

VD⁷: Integración, entrega y despliegue continuo de sistemas de información

Las redes de categorías para las variables independiente y dependiente se muestran en las figuras 9 y 10, respectivamente.

Figura 9

Red de categorías correspondiente a la variable independiente



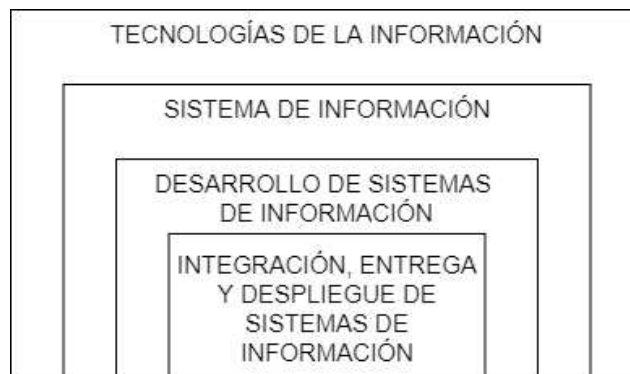
Nota: Esta figura representa cada uno de los contextos a tratar dentro de la variable independiente.

⁶ VI: Variable independiente

⁷ VD: Variable dependiente

Figura 10

Red de categorías correspondiente a la variable dependiente



Nota: Esta figura representa cada uno de los contextos a tratar dentro de la variable dependiente.

Fundamentación Científica de la Variable Independiente.

Diseño de Software.

Para definir el diseño de software se lo suele hacer en dos sentidos distintos, cuando el diseño de software es utilizado como producto y cuando es utilizado como proceso. En el primer sentido, el diseño de software indica el inicio de un sistema de información, es decir, es una proyección del sistema que se quiere obtener. Por lo general este diseño es el resultado de las decisiones del diseñador para formar una abstracción del sistema que es deseado (Durango, 2014).

Otros autores ven al diseño de software como un modelo de instrumentación o implantación que se basa en modelos conceptuales, es decir, esta es una de las primeras etapas del proceso de desarrollo de software (JF, 2017). Generalmente esta fase produce un diseño arquitectónico, de interfaz y procedimental, pero se resalta el arquitectónico ya que este define las relaciones entre los principales elementos, siendo estas herramientas y o servicios adicionales.

Arquitectura de Software.

No existe una definición que sea globalmente aceptada para arquitectura de software, de manera general la arquitectura de software es vista como el trabajo dinámico dentro del proceso de ingeniería, la configuración o topología estática de sistemas de información vista desde un nivel de abstracción elevado o la caracterización de una disciplina que de uno o ambos aspectos mencionados (Reynoso, 2004).

Una de las definiciones más conocidas de la arquitectura de software es la de Clements, pues menciona que esta es una vista de un sistema que incluye varios componentes principales, mostrando la conducta que estos componentes y la forma en como interactúan para alcanzar un objetivo específico (Clements, 1996). Sin embargo, cuando hablamos de la vista arquitectónica de un sistema se habla de una vista abstracta, pues se trata de alcanzar un alto nivel de comprensión de lo que se quiere realizar.

Existen tipos de arquitecturas, entre las más mencionadas se encuentran las distribuidas, no distribuidas y un grupo en donde se engloban a otras que no cumplen con la definición de distribuidas y no distribuidas (Castro, 2015). De manera general, estos tipos de arquitecturas buscan generar un modelo que describa como se organiza un sistema, identificando los principales componentes y las relaciones entre ellos. La arquitectura de software puede favorecer la disponibilidad, el rendimiento, la seguridad y los cambios (Castro, 2015).

Esquemas y Patrones de Software.

Un esquema hace referencia a unidades que implementan varias funcionalidades que son necesarias para uno o varios sistemas de información (Romero, 2006). Cada sistema o aplicación requiere de un esquema que se encargue de implementar cada una de las funcionalidades comunes. El no contar con un esquema definido para el desarrollo de sistemas

de información dificulta la migración de estos, teniendo como inconveniente una pobre respuesta al cambio (Romero, 2006). También se menciona que un esquema funciona como una estrategia de integración de aplicaciones, esto permite que se reduzcan las fallas y los tiempos de stop en la operación de los sistemas.

Por otro lado, los patrones en la práctica de la arquitectura de software constituyen uno de los paradigmas o marcos de diseño arquitectónico (Reynoso, 2004). Existen diversas clases de patrones; de análisis, arquitectura, estructurales, distribuidos, de proceso, entre otros.

Como se mencionó, en la práctica, los patrones de arquitectura de software hacen referencia a paradigmas, en los últimos años el termino DevOps ha tomado auge dentro de la industria de desarrollo de software. DevOps es un esquema en donde se involucran las tareas de los desarrolladores (Dev) y del personal de operaciones (Ops) (Delgado, 2020).

Ops hace referencia a los administradores de sistemas, los cuales tienen la misión de mantener funcionales los sistemas de información, realizar nuevos despliegues y realizar rollbacks de versiones anteriores en caso de ser necesario (Delgado, 2020). Dev se compone por los desarrolladores que trabajan en la lógica y creatividad de los sistemas de información, por lo general estos pasan el mayor de su tiempo codificando soluciones y mejorando aplicaciones con nuevas versiones (Delgado, 2020).

DevOps se convirtió en más que un esquema o patrón a seguir, siendo considerado en la actualidad como un movimiento que aborda las tareas de desarrollo y operaciones, mejorando la colaboración entre estos equipos y agilizando el proceso completo de entrega de software. Esto ha permitido generar un cambio organizacional que acelera el proceso de despliegue de software (Céspedes, 2021) lo cual a su vez influye directamente en la calidad de los productos de software, específicamente, las prácticas asociadas con DevOps influyen directamente en la

confiabilidad y mantenibilidad de los sistemas de información que se desarrollan bajo este paradigma.

Por otra parte, se menciona que DevOps va más allá de una tecnología o conjunto de prácticas, sino que es el resultado de la aplicación de principios que dan un valor a agregado a las tecnologías de la información (Retamal, 2019).

El resultado de esta mezcla de prácticas culturales, conceptos técnicos y arquitectónicos representan a una gestión filosófica que engloba varias actividades como las que se describen a continuación:

- Cultura: en este punto se engloba a personas, procesos y herramientas.
- Automatización: esta parte es importante ya que de esta manera se pretende obtener una rápida retroalimentación.
- Métricas: a través del uso de herramientas se obtiene métricas que permiten determinar la calidad.
- Compartir: esto permite crear una cultura donde los integrantes de los equipos comparten ideas y procesos.

Estas serían las bases para que DevOps sea considerado un paradigma que transmite confianza y da sentido a la comunidad que lo adopte.

Framework de desarrollo de software.

Como tal, un framework puede considerarse a un conjunto de técnicas y conocimientos que permiten la utilización de materiales con la final de sacar provecho y cumplir un objetivo (Degiovannini, 2011).

En otras palabras, se puede definir un framework como una estructura que contiene técnicas que a su vez hacen uso de elementos necesarios, sin embargo, al mencionar framework se considera que este sea capaz de automatizar actividades que son realizadas de manera manual por los usuarios.

Un framework permite agilizar varios procesos, depende de cómo sea construido este, de manera general es un modelo que permite identificar y solventar problemas de manera inmediata.

Fundamentación Científica de la Variable Dependiente.

Tecnologías de la Información.

Las tecnologías de la información se conforman de un gran grupo de herramientas, técnicas y canales desarrollados por las tecnologías que permiten producir, tratar y presentar información a través de varios formatos, sean estos datos, voz o contenido multimedia; con la finalidad de mejorar la calidad de vida del ser humano (Ávila W. , 2013).

Las tecnologías de la información en el contexto de la practica empresarial han avanzado en función del avance tecnológico. Especial importancia en la evolución de las tecnologías de la información tiene el desarrollo de software, pues esto ha permitido que aparezcan nuevas herramientas de gestión con funciones y aplicaciones empresariales, unas de estas son: software de simulación, Workflow, Datamining, Datawarehouse, inteligencia artificial, gestión documental, mapas de conocimiento (Pérez D. , 2007).

Sistemas de Información.

Los sistemas de información comúnmente hacen referencia a un aplicativo que permite realizar determinadas tareas, sin embargo, existen varias definiciones populares sobre este término.

“Los sistemas de información son combinaciones de hardware, software y redes de telecomunicaciones que las personas construyen y usan para recopilar, crear y distribuir datos útiles, generalmente en entornos organizacionales” (Valacich, 2012).

“Los sistemas de información son componentes interrelacionados que trabajan juntos para recopilar, procesar, almacenar y difundir información para respaldar la toma de decisiones, la coordinación, el control, el análisis y la visualización en una organización” (Valacich, 2012).

Como tal estas definiciones se centran en dos formas diferentes de describir a los sistemas de información, por un lado, se resaltan los componentes y por el otro se resalta el papel que desempeñan estos en componentes dentro de una organización.

En cuanto a los componentes de los sistemas de información, los más destacados son los siguientes: hardware, software, datos, personas y procesos. Los primeros tres encajan en la categoría de componentes tecnológicos, mientras que los dos últimos se separan de la parte tecnológica y van a un campo más técnico como la informática (Bourgeois, 2014).

Ahora en cuanto al papel que cumplen los sistemas de información se puede mencionar que uno de los roles de estos sistemas es la recopilación de información, es decir, llevar a cabo el proceso de toma de datos y transformarlos a información para finalmente convertirlos en conocimiento organizacional (Bourgeois, 2014).

Desarrollo de Sistemas de Información.

El desarrollo de sistemas de información es considerado una actividad mediante la cual se facilita una actividad laboral, este proceso a su vez incluye subtarefas de análisis, diseño, desarrollo, implementación y evaluación (Cruz, Miranda, & Gonçalves, 2013).

Depende del punto de vista de donde sea visto este proceso, puede pertenecer a ingeniería de software o a un proceso de adquisición de un producto de software. El objetivo general del desarrollo de sistemas de información es el soporte del proceso de prueba de un sistema, la guía de un proceso de verificación o la base para confirmar la conformidad de los usuarios (Wang & Zhao, 2020).

Integración, Entrega y despliegue continuos de sistemas de información.

Tanto la integración, entrega y despliegue son actividades que se llevan a cabo cuando se construye un sistema para determinados usuarios. A continuación, se presenta una explicación de a que conlleva cada una de estas actividades.

Integración continua.

La integración continua dentro del desarrollo de software es una práctica que ha sido ampliamente establecida en la industria de software. Dentro de esta actividad se integran los miembros de un equipo de codificación de software, con frecuencia estos miembros del equipo actualizan repositorios donde se almacena el código. La integración continua permite a las empresas o grupos de desarrollo mejorar la calidad de software y aumentar la productividad de sus equipos, además de la automatización de pruebas antes de construir una nueva versión de un sistema (SHAHIN, ALI BABAR, & ZHU, 2017).

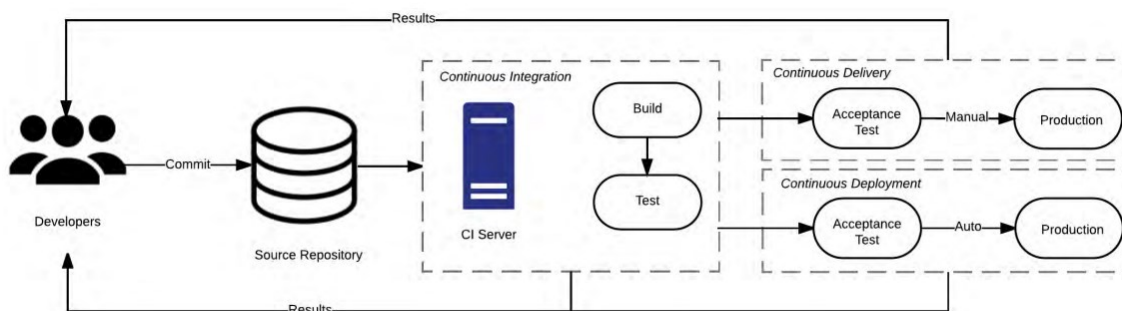
Entrega continua.

La entrega continua tiene como objetivo garantizar que cada una de las aplicaciones este siempre lista para el paso a producción, esto después de haber pasado con éxito por un

proceso de pruebas y controles. La entrega continua emplea un conjunto de prácticas, por ejemplo, la integración continua y la automatización de la implementación a un entorno similar al de producción en donde se realizarán las pruebas necesarias. El hacer uso de este proceso trae grandes beneficios, uno de estos es un menor riesgo de implementación en ambientes de producción, otro beneficio es que los procesos resultan ser menos costosos y más ágiles, esto último permite obtener comentarios de los usuarios de manera pronta (SHAHIN, ALI BABAR, & ZHU, 2017). La figura 11 hace referencia a la conexión que existe entre la entrega y la integración continua, pues la entrega continua requiere una práctica de integración continua.

Figura 11

Relación entre la integración continua, entrega y despliegue



Despliegue continuo.

La práctica de la implementación o despliegue continuo va un poco más allá, pues hace referencia a un proceso automático de implementación y continuo de una aplicación o sistema específico. Lo que diferencia al despliegue continuo de la entrega continua es básicamente el entorno en el que se lo realiza, pues el despliegue se lo hace ya en un ambiente de producción, es decir, un ambiente en donde se el sistema se pone en contacto con usuarios reales.

El objetivo de esta práctica es implementar de manera automática y constante cada cambio en el ambiente de producción, teniendo en cuenta que la práctica del despliegue requiere de la práctica de entrega (SHAHIN, ALI BABAR, & ZHU, 2017).

A manera de conclusión, durante este proceso no deberían existir procesos o tareas manuales, tan pronto como los desarrolladores confirmen un cambio, el cambio se implementa en producción a través de una canalización de entrega y despliegue.

Capítulo III

Construcción de la Solución

Introducción

Con la finalidad de aportar a la resolución de la problemática identificada en el presente proyecto, se desarrolló un prototipo funcional que responda a las necesidades del actual equipo de desarrollo de la Unidad de Tecnologías de la Información y la Comunicación (UTIC) de la Universidad de las Fuerzas Armadas – ESPE referentes a la integración, entrega y despliegue de aplicaciones de TI.

Los procesos de integración, entrega y despliegue de sistemas son actividades que se llevan a cabo periódicamente por parte del personal de desarrollo. La UTIC ha venido trabajando bajo un esquema que requería que su personal lleve a cabo las tareas relacionadas con la integración, entrega y despliegue de sistemas de información de manera manual, es decir, se levanta un ambiente completo con las herramientas necesarias para que el sistema funcione correctamente. Este proceso se lo realizaba por cada uno de los aplicativos que maneja la institución de manera aislada.

En el estudio de la situación actual que se realizó al inicio del proyecto (ver Capítulo 1), se levantó información acerca de los sistemas de información que se encuentran en producción dentro de la universidad, encontrándose un total de 8 sistemas considerados fundamentales para la institución y que se encuentran disponibles para la comunidad universitaria. La tabla 5 muestra los aplicativos y las tecnologías que utilizadas para su desarrollo.

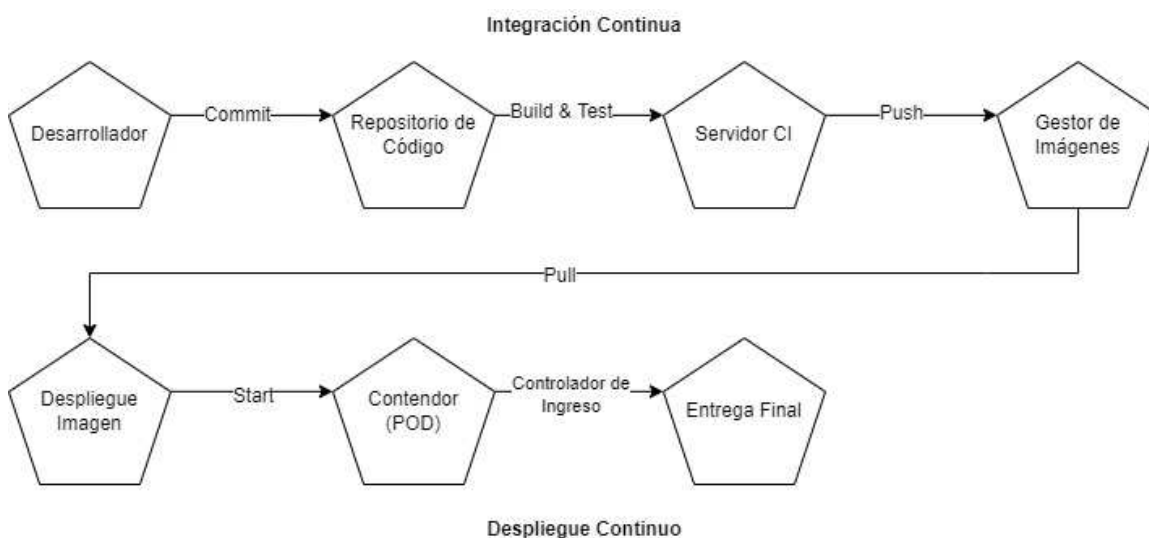
Tabla 5*Lista de sistemas de información a cargo de la UTIC*

#	Nombre	Capa	Tecnología
1	Actividades	Backend	Spring Boot
		Frontend	Angular
2	Calendario Académico	Backend	Spring Boot
		Frontend	Angular
3	Espemático	Backend	Spring Boot
		Frontend	Angular
4	Formularios	Backend	Spring Boot
		Frontend	Angular
5	Pagos	Backend	Spring Boot
		Frontend	Angular
6	Registro	Backend	Spring Boot
		Frontend	Angular
7	Salud	Backend	Spring Boot
		Frontend	Angular
8	Tutorías	Backend	Spring Boot
		Frontend	Angular

Como se puede observar, la UTIC realiza desarrollos utilizando tecnologías específicas. Para poner a disposición estos sistemas, la UTIC generaba un archivo WAR por cada desarrollo, es decir, un paquete comprimido que contiene los componentes web necesarios para que el aplicativo funcione en el servidor, afectando directamente al rendimiento de las aplicaciones ya que todas comparten los mismos recursos. Esto hace referencia a la problemática definida en el capítulo 1, pues existe una gestión inadecuada del desarrollo y despliegue de los sistemas de información.

Por lo expuesto, se planteó un framework de trabajo para automatizar los procesos de integración, entrega y despliegue de los sistemas desarrollados y mantenidos por la UTIC; así como también, para mejorar la gestión de los recursos tecnológicos existentes.

El framework fue concebido siguiendo estrictamente los lineamientos de la cultura DevOps, la cual plantea siete etapas: Planificación, codificación, construcción, pruebas, levantamiento, operatividad y monitoreo. Para operacionalizar el framework, se desarrolló un prototipo que precisó de varias herramientas de código abierto, justamente para automatizar las tareas que eran realizadas de forma manual por los miembros del equipo de desarrollo y operaciones de UTIC. Además, el prototipo fue desarrollado haciendo uso de los recursos tecnológicos que la unidad posee. En la figura 12 se presenta el modelo del framework propuesto para la integración y despliegue continuos de sistemas de información.

Figura 12*Arquitectura del prototipo de framework***Módulos de la arquitectura**

Las características, herramientas y conexiones de cada uno de los módulos se detallan a continuación.

Integración Continua.

Este proceso inicia cuando el desarrollador publica sus cambios en un repositorio de código. A paso seguido, el servidor de Integración Continua (CI de las siglas del inglés Continuous Integration) detecta estos cambios, realiza pruebas y construye una imagen del sistema de información, la cual es almacenada en el gestor de imágenes.

Despliegue Continuo.

Una vez almacenada la última versión de la imagen del sistema de información, se la descarga para su despliegue, es decir, poner en marcha la aplicación dentro de un contenedor. Finalmente, el controlador de ingreso es el encargado realizar las entregas a las peticiones que realiza el usuario mediante el navegador.

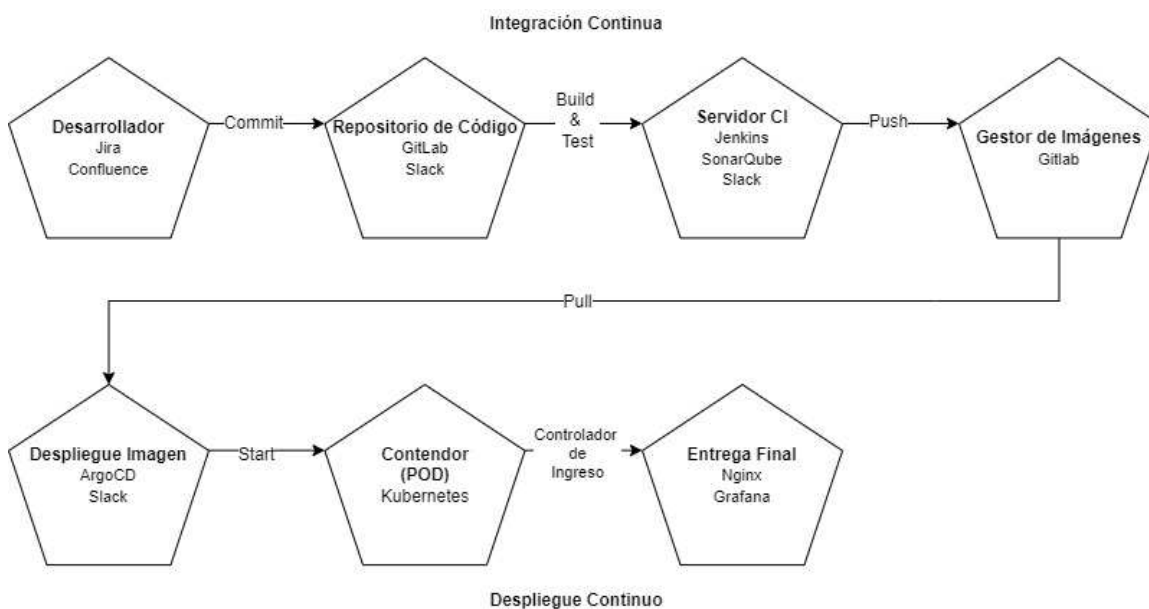
Selección de herramientas

En esta sección se detallan las herramientas utilizadas para estructurar la arquitectura del framework y las razones por las cuales fueron seleccionadas. Es preciso aclarar que las herramientas seleccionadas son Open Source o de software libre, debido a que existe una ley orgánica vigente en el Ecuador en donde se indica que las instituciones públicas en función a sus requerimientos y compatibilidades tecnológicas deben fomentar el uso de sistemas basados en software no propietario o software libre (Republica, 2017). Además, este tipo de herramientas también se eligieron por el gran soporte que tiene, el cual se debe a la cantidad de usuarios que tienen a nivel mundial.

A continuación, en la figura 13 se presenta la arquitectura del prototipo de framework con las herramientas seleccionadas para cada fase.

Figura 13

Arquitectura del prototipo de framework con herramientas



GitLab

GitLab es una herramienta de software libre en su versión Community Edition, para este proyecto se hizo uso de esta instancia en su versión 11.2.3 ya que esta versión es la que se venía utilizando en la UTIC. Esta herramienta permite el control de versiones del código basado en Git. Git es una tecnología que permite el control total de los cambios que se realizan en el código fuente dentro de un marco de trabajo (Tobar, 2018). Los cambios en el código se los realiza de manera local para posteriormente actualizarlos en un servidor centralizado que se encarga de evaluar las modificaciones realizadas en comparación con la última versión.

GitLab dispone de utilidades que ayudan al desarrollador de software a generar un pipeline de integración continua y de entrega continua para agilizar los desarrollos (Tobar, 2018). El pipeline se define como una secuencia de fases que parten desde el código fuente hasta llegar al lanzamiento de una nueva versión de una aplicación, pasando por las fases de construcción, pruebas y despliegue en entornos alternos al de producción. Los pipelines son lanzados de inmediato cuando el desarrollador confirma el commit de los cambios efectuados en el código.

La razón por la cual se utilizó esta herramienta es porque GitLab es una de las herramientas más conocidas y utilizadas (Ávila R. , 2019), además, esta herramienta ya venía siendo utilizada por la UTIC en el proceso de integración de código hacia repositorios.

Jenkins

Jenkins es una herramienta que permite la integración⁸ continua y automática de proyectos de desarrollo, permitiendo detectar fallos de manera inmediata (Burgueño, 2018). Jenkins cuenta actualmente con varias funciones que además de asistir los procesos de

⁸ Integración: Se entiende por integración a la compilación y ejecución de pruebas de todo un proyecto de software

integración continua, asiste también en los procesos de despliegue y a la entrega continua (Brito, 2022).

En este trabajo, la ejecución de trabajos automatizados y su orquestación se lo realiza con Jenkins. Esta herramienta es pieza fundamental sobre la que se sustentan los procesos del paradigma DevOps, los trabajos que vayan a ser ejecutados en repetidas ocasiones se configuran en Jenkins, este servidor o servicio de automatización es gratuito y dispone de varios plugins para poder ampliar su funcionalidad (Olmedo, 2018).

Se seleccionó Jenkins debido a que esta se encuentra dentro del mundo de las herramientas Open Source. Además, esta herramienta es una de las más utilizadas en proyectos que aplican prácticas de la cultura DevOps y permite materializar los flujos de integración y despliegue continuos (Martinez, 2021).

Docker

Docker es una tecnología que permite la construcción y distribución de aplicaciones mediante el uso de contenedores. Esta tecnología dota de una capa de abstracción y automatización (Olmedo, 2018). Docker utiliza características del núcleo del sistema donde se encuentra alojado para que los contenedores sean independientes a pesar de encontrarse dentro de una misma instancia, lo que le permite ser más ligero en comparación con la virtualización clásica.

En este proyecto se hace uso de Docker ya que esta es una plataforma open source y permite ejecutar aplicaciones de manera flexible, sin dejar de mencionar que Docker es la herramienta más utilizada para la contenerización (Martinez, 2021).

Esta herramienta se utilizó para las diferentes actividades que ejecutará el pipeline mencionado anteriormente, con esta herramienta se construirá una imagen mediante un archivo denominado Dockerfile.

ArgoCD

ArgoCD es una herramienta de despliegue continuo que se utiliza especialmente en la aplicación de la metodología GitOps, la cual consiste en una serie de prácticas para gestionar las configuraciones de los sistemas y las infraestructuras mediante Git (RedHat, GitOps, 2021).

De manera específica, ArgoCD es un operador de Kubernetes que define un conjunto de CRDs (Custom Resource Definitions), permitiendo el despliegue de aplicaciones de manera continua en diferentes clústeres y de esta manera centralizar todo el despliegue de una determinada infraestructura. Además, permite implementar la monitorización de los repositorios realizando 'pulling', es decir, cada cierto tiempo lee los repositorios configurados y aplica los cambios cuando exista alguna modificación.

Para el desarrollo de este proyecto se utilizó la versión 2.1.5, la misma que fue instalada en servidor y su configuración se la detalla más adelante.

Kubernetes

Kubernetes es una plataforma desarrollada por Google y establecida de código abierto, esta plataforma permite orquestar la ejecución de aplicaciones que se ejecutan en contenedores, permitiendo ejecutar estas aplicaciones (García, 2018).

La plataforma provee de tecnología de despliegue, mantenimiento y escalado de aplicaciones como una de sus funciones de orquestación de contenedores. Con respecto a los contenedores, Kubernetes, admite varios motores de ejecución entre los cuales se encuentra Docker el mismo que fue elegido para el desarrollo de este proyecto (García, 2018).

Kubernetes realiza la distribución de los contenedores en pods, estos pueden estar en varios nodos formando un clúster como se mencionaba en la definición de ArgoCD. Los contenedores se agrupan en pods, por lo que se suelen agrupar en el mismo pod a contenedores que usan y necesitan los mismos recursos.

En este proyecto se hace uso de Kubernetes ya que al igual que las anteriores herramientas es parte del mundo open source y con la finalidad de formar un clúster que permita orquestar la ejecución de los sistemas de información en diferentes contenedores, donde cada uno de estos se encuentra en una infraestructura lógica independiente y permite aliviar el alojamiento y ejecución de estas.

Nginx

Nginx es un servidor web de código abierto que posee características robustas y funcionalidades diversas, una de ellas es que se lo considera como un servidor de alto rendimiento y modular que se lo puede utilizar como servidor web, servidor de proxy inverso y como equilibrador de carga (RedHat, INSTALACIÓN Y CONFIGURACIÓN DE NGINX, 2019).

Nginx se caracteriza por presentar una arquitectura que está orientada a eventos con la cual maneja las peticiones, presenta un módulo estático y funciona como un servidor proxy simple, donde recibe las solicitudes http del cliente y las reenvía para obtener una respuesta (Pérez N. , 2020).

En este proyecto se utilizó Nginx con la finalidad de que este actúe como controlador de ingreso, es decir, al momento de que el usuario solicita el ingreso a un DNS específico de un aplicativo de los que está encargado UTIC, este se encarga de tomar esta petición y direccionar hacia el POD donde se encuentra desplegada la aplicación. En el caso de que exista una

sobrecarga de peticiones el clúster será el encargado de replicar el POD para satisfacer las peticiones de los usuarios.

Implementación

Para el proceso de implementación se hizo uso de los recursos tecnológicos que posee la UTIC, en este caso se utilizaron dos servidores:

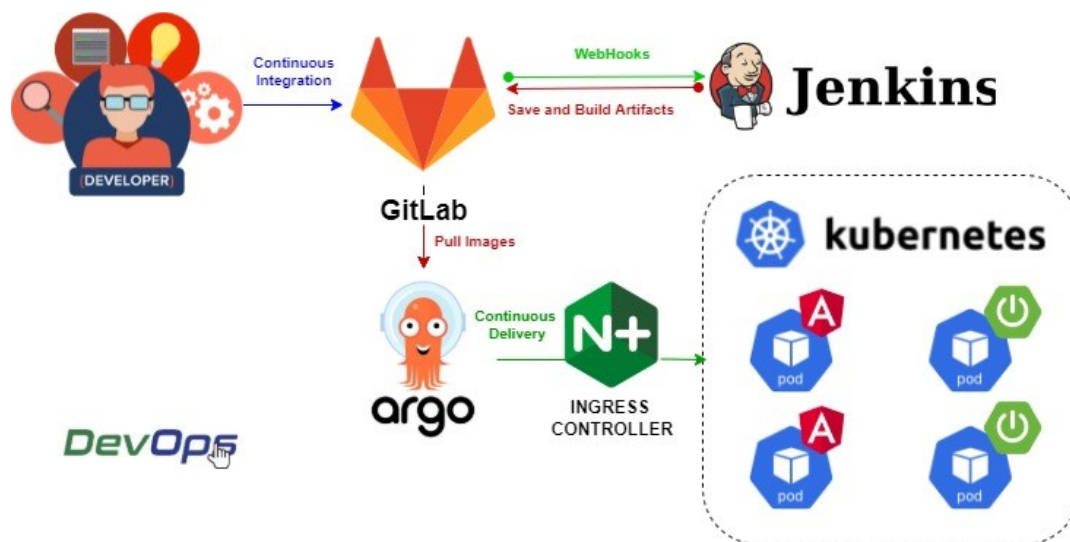
- Servidor 1: 8 CPU, memoria RAM de 16 GB y 80 GB de almacenamiento
- **Servidor 2:** 16 CPU, memoria RAM de 16 GB y 40 GB de almacenamiento.

En el primer servidor se instaló Docker, Kubernetes y ArgoCD como gestor de Kubernetes. En el segundo servidor se instaló Jenkins, Jira, Confluence y SonarQube las cuales ayudan al desarrollo ágil y monitoreo de aplicaciones.

La interacción conjunta de las herramientas mencionadas permite estructurar el framework propuesto, dicha estructura se visualiza en la figura 14.

Figura 14

Framework propuesto



El grafico anterior presenta tanto las herramientas utilizadas como las diferentes etapas que se llevan a cabo en la implementación del paradigma orientado a DevOps.

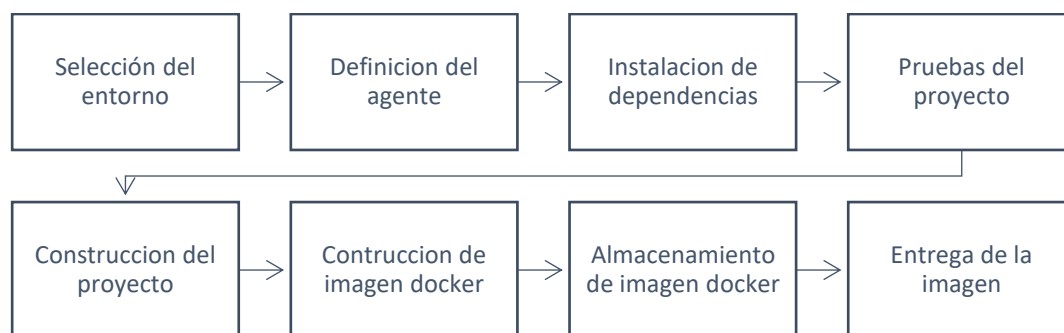
Integración Continua

El ciclo inicia cuando el desarrollador sube los cambios al gestor de versionamiento de código. La UTIC viene utilizando GitLab como repositorio de código, el cual se encarga de almacenar y versionar el código de los sistemas de información que se desarrollan en la unidad.

Dentro de GitLab Se crearon dos repositorios para definir las configuraciones para el despliegue de los sistemas de información. Uno de los repositorios sirve para los proyectos en ambiente de pruebas y el otro para el ambiente de producción.

La integración de GitLab con Jenkins permite que al momento que se realicen cambios en el repositorio se ejecuten una serie de pasos definidos en un pipeline. El pipeline tiene la finalidad de definir el flujo de integración continua y generar una imagen del sistema de información, la cual es almacenada en el propio repositorio de GitLab. Para construir una de estas imágenes, es preciso que cada uno de los proyectos cuente con los archivos Dockerfile y Jenkinsfile. Dockerfile es un archivo de texto que indica los pasos a seguir para construir una imagen dentro de un contenedor de Docker. Jenkinsfile es un archivo de texto en donde se define el pipeline.

Es preciso mencionar que la definición del pipeline obedece a que cada una de sus etapas automatice los procesos de integración y organice el ciclo de vida de las aplicaciones en la UTIC. La figura 15 muestra el orden de las etapas que componen el pipeline propuesto.

Figura 15*Pipeline propuesto*

Cada etapa del pipeline corresponde a una o más tareas que suelen ejecutar los miembros del equipo de desarrollo y operaciones de manera independiente y manual. A continuación, se describe cada una de las tareas que se realizan en cada etapa del pipeline.

Selección del entorno: En esta etapa se especifica el entorno, es decir, el nombre con el que se guardará la imagen, esto se lo hizo de tal manera de que el nombre de la imagen sea el mismo del proyecto de GitLab, para no perder la trazabilidad y no caer en el desorden. Además, se especifica un token de acceso para permitir la integración entre Jenkins y GitLab. Finalmente, se detalla la ruta del directorio en donde se encuentran los archivos necesarios para el despliegue.

Definición del agente: En esta etapa se definen las especificaciones técnicas que tiene el contenedor, es decir, la versión de Docker que se va a utilizar, la tecnología sobre la cual se compila el proyecto y en el caso de que existan volúmenes de persistencia, también deben ser definidos en esta etapa.

Instalación de dependencias: En esta etapa se instalan las de dependencias necesarias para el funcionamiento del sistema de información, de acuerdo con las tecnologías definidas en la etapa anterior.

Pruebas del proyecto: En esta etapa se ejecutan las pruebas del proyecto con la finalidad de constatar que el mismo se encuentre sin fallo alguno, es decir, esta etapa permite validar que la codificación realizada por los desarrolladores no tenga fallas.

Construcción del proyecto: El desarrollo de esta etapa tiene en cuenta el origen de los cambios realizados por el desarrollador. Para esto se manejan dos ramas previamente definidas en GitLab, si se trata de la rama develop la construcción del proyecto se lo hace en ambiente test, mientras que si se trata de la rama master la construcción del proyecto se lo hace en ambiente de producción.

Construcción de imagen Docker: En este punto se hace uso del archivo Dockerfile definido anteriormente, el cual es el encargado de tomar el proyecto construido en la etapa anterior y, a partir de este, generar una imagen “dockerizada” lista para su ejecución.

Almacenamiento de imagen Docker: En esta etapa se almacena la imagen construida, en el repositorio de GitLab; esta imagen, posee como etiqueta el código que se genera cuando se hace commit de la imagen, para no perder de vista la trazabilidad y la identificación de la imagen.

Entrega de la imagen: En esta etapa, al igual que en la etapa de construcción del proyecto, se lo realiza según la rama definida en GitLab. Si se trata de la rama develop se realiza una actualización en el repositorio que contiene las configuraciones para el despliegue en ambiente de pruebas, modificando la etiqueta de la imagen en el directorio definido en la selección del entorno.

Despliegue Continuo

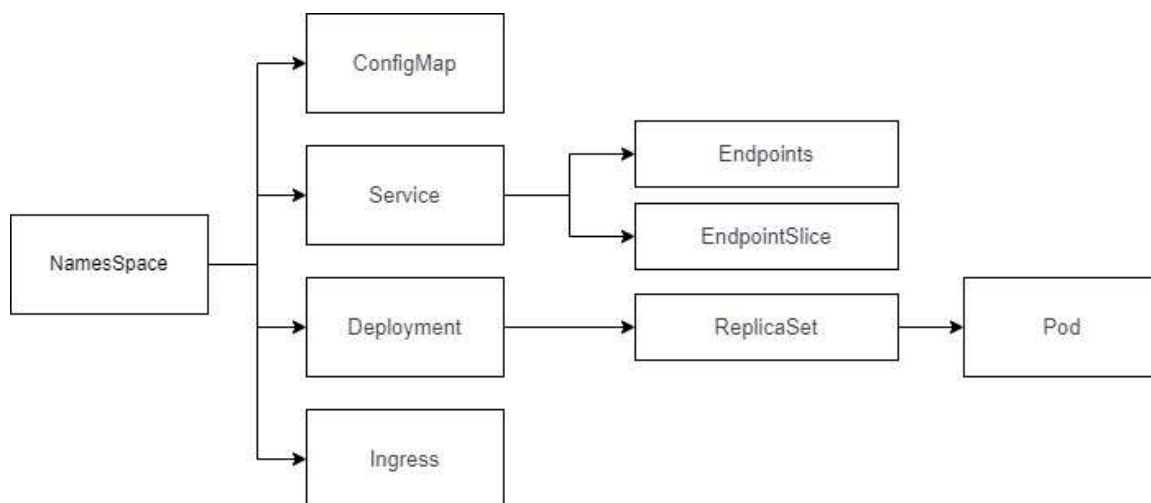
El ciclo continúa con la puesta en funcionamiento de la imagen generada en el proceso anterior. Utilizando Kubernetes se genera un contenedor con todo lo necesario para que el

sistema de información funcione correctamente; de esta manera, se administra adecuadamente los recursos disponibles en la UTIC.

Con el fin de administrar los contenedores de una manera visual se hace uso de ArgoCD que gestiona los proyectos de los sistemas de información haciendo uso de los repositorios mencionados en la fase de CI. De acuerdo con el ambiente que se esté tratando, en cada repositorio de configuración se establecen los archivos necesarios para poner en funcionamiento la imagen, es decir se dan las instrucciones de ejecución del Kubernetes según la arquitectura presentada en la figura 16.

Figura 16

Arquitectura Kubernetes



Cada contenedor de Kubernetes cumple con el esquema propuesto en la figura anterior dónde el namespace es el nombre del sistema de información y hace las veces de separador entre proyectos dentro del Clúster de Kubernetes.

En el configmap se definen las configuraciones del contenedor, ya sean exposición de puertos, definición de volúmenes, zona horaria e integraciones con otras aplicaciones.

Service especifica el servicio a levantar, así como los puertos de entrada y salida del sistema de información; además, crea de forma automática el servicio de endpoints que mapea todas las peticiones del usuario y las ejecuta según sean requeridas. Este espacio crea también los endpointslices pensados para la alta demanda del servicio debido a que permiten distribuir los endpoints en la red a través de múltiples recursos.

La imagen creada y almacenada en el gestor de imágenes definida en el CI es llamada en el Deployment, así como la definición del ambiente donde se va a trabajar y el número de réplicas necesarias. Deployment se encarga también de arrancar el sistema de información tomando en cuenta el número de réplicas definidas para crear los POD's necesarios ayudando de esta manera a la escalabilidad del sistema de información.

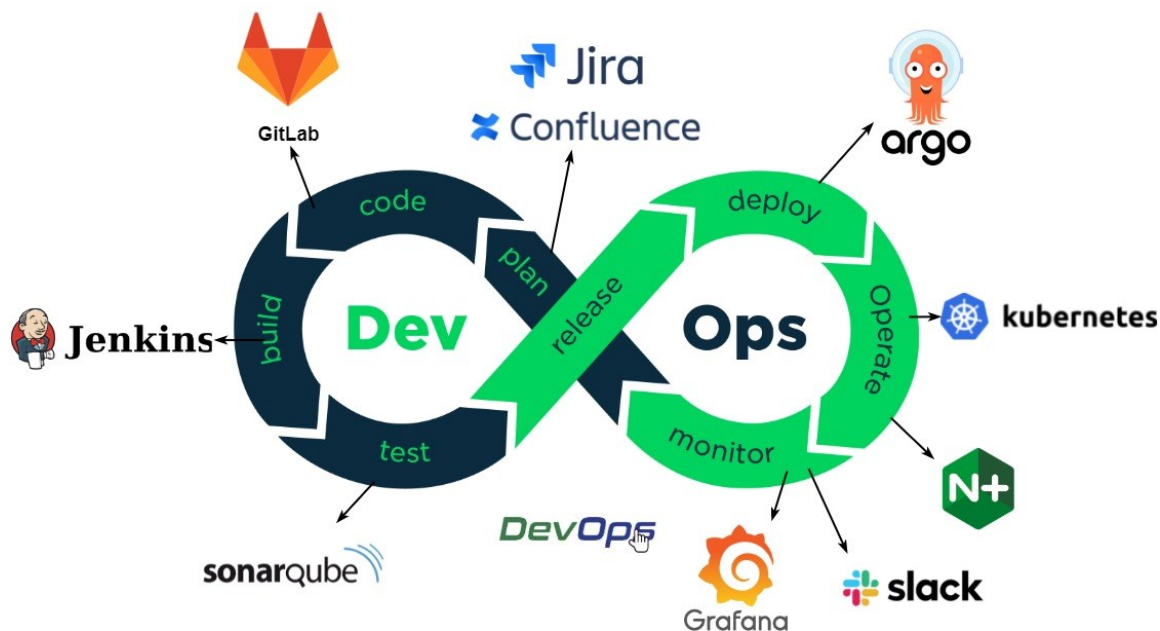
Finalmente, el apartado de Ingress permite al POD salir a la web, aquí se define tanto el DNS a utilizar como el certificado de seguridad del sistema de información, esto se logra haciendo uso de Nginx que sirve como proxy entre el DNS y la aplicación en la web.

Otras herramientas

Con el propósito de cumplir estrictamente con los lineamientos del paradigma de DevOps, además de las funciones de CI/CD fue necesario integrar otras herramientas que ayudan en las diferentes fases de este proceso como se puede ver en la figura 17.

Figura 17

DevOps y Herramientas



En la etapa de planeación se hace uso de Jira y Confluence, herramientas que están relacionadas con metodologías ágiles de desarrollo. Con el Uso de Jira el equipo puede gestionar la planificación creando incidencias a resolver bajo la metodología SCRUM. Por otra parte, Confluence lleva la documentación del proyecto que se esté desarrollando, permitiendo a cualquier desarrollador informarse y entender tanto la lógica del negocio, como la programación del sistema de información.

Como se mencionaba anteriormente la unidad ya venía trabajando con GitLab, sin embargo, se hizo uso de esta herramienta tanto para gestionar las funciones de CI/CD como para la gestión de configuración del código estableciendo normas de uso para las ramas en cada proyecto.

La etapa de pruebas o test de código se la realiza con SonarQube, esta herramienta ayuda a la detección de bugs y código peligroso que pueda alterar la funcionalidad del sistema

de información; además, esta herramienta califica el código indicando el error y cómo mejorarlo.

El deployment de los sistemas de información se lo realiza con ArgoCD gracias a la gestión de Kubernetes realiza, cumpliendo con esto la etapa de despliegue y operación.

Por último, la etapa de monitoreo se la realiza con dos herramientas: Grafana y Slack. Grafana muestra métricas de rendimiento de hardware con diferentes estilos, mediante el uso de un agente llamado Prometheus, que es capaz de monitorear el estado de los servidores y de las aplicaciones en tiempo real. Esto permite actuar de manera inmediata en caso de tener problemas de sobrecarga en el servidor o en los sistemas de información desplegados.

Por otro lado, Slack, además de servir como medio de comunicación para todo el equipo de desarrollo, se integra a GitLab, Jenkins y ArgoCD mostrando notificaciones del proceso en que se encuentra el sistema de información, de esta manera se puede evaluar si existe algún tipo de error en cualquiera de las etapas de DevOps.

Capítulo IV

Validación del Prototipo

Para validar el prototipo de framework de desarrollo, integración y despliegue continuo de sistemas TI de la Universidad de las Fuerzas Armadas – ESPE basado en la cultura DevOps, se realizó un análisis exploratorio mediante una entrevista al personal del área de desarrollo de la UTIC que intervino al inicio de esta investigación.

Con la finalidad de obtener la comparación entre la situación inicial del proceso de desarrollo y despliegue de sistemas de información y la situación actual del framework propuesto que se encuentra implementado, se planteó realizar un estudio que permita validar el funcionamiento del prototipo del framework de desarrollo de sistemas de información en la Universidad de las Fuerzas Armadas – ESPE. Para ello se estructuró una entrevista⁹ que abarca 3 puntos principales.

- Conocer sobre el uso del framework de integración de sistemas de información que se implementó y como estos han permitido mejorar los procesos de integración y despliegue.
- Conocer si los sistemas de información que utilizan el nuevo framework responden adecuadamente a las necesidades de los usuarios.
- Conocer si las herramientas utilizadas responden adecuadamente a las necesidades de la Unidad de Tecnologías de la Información de Universidad de las Fuerzas Armadas – ESPE.

Al momento de realizar la entrevista, a manera de introducción, se realizó una presentación del prototipo con la finalidad de que los miembros del equipo tengan claro el

⁹ Los insumos de la entrevista se encuentran accesibles en el anexo 3

funcionamiento de este y cómo actúan las herramientas utilizadas en cada uno de los procesos.

Acto seguido se procedió con el planteamiento de las preguntas definidas en el diseño de la entrevista de validación ¹⁰ con la finalidad de recopilar información con la finalidad de determinar si el prototipo se encuentra cumpliendo con el objetivo planteado.

Los resultados obtenidos de la entrevista fueron alentadores y se describen a continuación.

En cuanto al primer punto de la entrevista los usuarios expertos que actualmente se encuentran haciendo uso del framework pudieron mencionar que, en efecto, el uso de esta nuevo esquema de trabajo ha permitido mejorar los tiempos en los procesos de integración y despliegue de sistemas de información.

Así también, el uso de los ambientes de test y producción ha sido de gran ayuda al equipo de desarrollo, ya que con esto se tiene claramente definido el entorno en donde se llevan a cabo pruebas, sin embargo, podría existir un ambiente adicional en donde se integre un equipo de QA (Quality Assurance) con la finalidad de realizar pruebas de calidad de software, este ambiente podría ser algo similar al de producción ya que, como se sabe, el ambiente de pruebas no siempre tiene los datos necesarios para realizar este tipo de pruebas.

En lo que respecta al tiempo que tomaba al equipo de desarrollo en poner en producción un sistema de información, todos coinciden en que se ha reducido ampliamente. Por ejemplo, anteriormente se tenía que dejar sin servicio a los usuarios para poder desplegar una nueva versión de un sistema de información; pero ahora, con el uso del framework, los cambios se los puede hacer sin la necesidad de detener el servicio; es decir, el despliegue se realiza de manera transparente para el usuario. Con el propósito de corroborar el criterio de los

¹⁰ Los insumos de la entrevista se encuentran accesibles en el anexo 4.

entrevistados, se procedió a medir la mejora en términos de minutos a partir de los datos que nos presenta Jenkins cada vez que se ejecuta el pipeline implementado, estos a su vez se compararon con los datos entregados por los miembros de equipo en referencia al tiempo que demoraba cada proceso utilizando el método anterior a la puesta en práctica del prototipo. Esta comparación de tiempos se puede observar en la tabla 6.

Tabla 6

Comparación tiempos de ejecución

#	Proceso	Tecnología	Antes del Framework (Tiempo Promedio)	Después del Framework (Tiempo Promedio)
1	Construcción	Spring Boot	5 min	1 min
		Angular	10 min	4 min
2	Test	Spring Boot	3 min	1 min
		Angular	5 min	3 min
3	Despliegue	Spring Boot	10 min	1 min
		Angular	10 min	1 min
4	Publicación	Spring Boot	2 min	30 seg
		Angular	2 min	30 seg

Como se puede observar los tiempos de ejecución en cada tarea se redujeron considerablemente sin afectar el funcionamiento de los demás sistemas de información ni detener el servidor para las actualizaciones.

En cuanto al segundo punto los miembros del equipo de UTIC han logrado identificar que el existe una alta concurrencia de usuarios en los sistemas durante el proceso de matriculación semestral, en ese sentido se realizaron pruebas haciendo uso del framework construido y este supo responder a esta gran concurrencia de usuarios, incluso se realizaron cambios en algunos sistemas de información durante el proceso de matrículas, soportando una concurrencia de aproximadamente ocho mil usuarios. Sin embargo, no se puede negar que este framework se puede encontrar con la limitación de infraestructura tecnológica si en algún momento se requiere otorgar más recursos a los sistemas de información para satisfacer la demanda de los usuarios.

Por último, en referencia al tercer punto, el uso de las herramientas Jira y Confluence es considerado un acierto por parte de los miembros del equipo de UTIC, pues estas herramientas han ayudado en gran medida tanto a la planificación y documentación de los proyectos como al control de trabajo del equipo de desarrollo; etapas que son muy importantes en la construcción de los sistemas de información y antes de la implementación de este framework no se llevaban a cabo, cabe recalcar que estas herramientas son recomendadas por expertos para la gestión de proyectos, incidentes y tareas.

La implementación de DevOps con Jenkins también ha sido considerada un acierto en cuanto al proceso de construcción de los sistemas de información. Se pudo haber seleccionado GitLab para realizar la construcción de los sistemas, sin embargo, Jenkins puede ayudar a futuro en otro tipo de integraciones, como por ejemplo en integraciones con almacenes de proyectos Docker. Además, el hecho de que estas herramientas sean Open Source dan un valor agregado al proyecto ya que no se necesita pagar por su uso y por otro lado se cumple con la normativa gubernamental.

De igual manera, el uso de Kubernetes para producción ha sido considerado un acierto, pues facilita la configuración para el despliegue mediante el método de infraestructura como código, por ende, es notable el cambio que hay en comparación con el método utilizado anteriormente.

El uso de la herramienta Slack ha permitido monitorear de mejor manera el trabajo de los miembros del equipo, pues las notificaciones permiten conocer las diferentes actividades que realizan los desarrolladores y las actividades que realiza el framework. De esta manera se puede dar seguimiento al proceso de desarrollo de sistemas de información desde la fase de codificación hasta la fase de despliegue y entrega de sistemas a los usuarios.

Capítulo V

Conclusiones, Recomendaciones y trabajos futuros

Conclusiones

Ante la presencia constante de problemas en los procesos de desarrollo, integración y despliegue de sistemas de información dentro de la Unidad de Tecnologías de la Información de la Universidad de las Fuerzas Armadas – ESPE, se pudo determinar que existe un manejo inadecuado de los proyectos de desarrollo de software dentro de dicha unidad.

La idea de construir un prototipo de framework para la integración y despliegue de sistemas de información en la UTIC, se validó mediante un estudio de factibilidad. Este estudio se realizó a través de una revisión de literatura preliminar en donde se identificaron la periodicidad de artículos en donde se encuentren involucradas las instituciones de educación superior y estrategias de integración y despliegue continuos de sistemas de información, además, se identificaron mecanismos propuestos para dar solución a la problemática identificada.

Como solución a la problemática identificada existían varias propuestas que se enfocaban en el uso de estándares, marcos de trabajo, pero sobre todo en la adopción del paradigma DevOps. Este paradigma se fundamenta en el uso de herramientas y buenas prácticas que permiten automatizar los procesos de integración y despliegue de los sistemas de información, donde, dichas herramientas permiten que los equipos de desarrollo y operaciones trabajen de mejor manera, a su vez esto se ve reflejado en la disminución de tiempos en la ejecución de las actividades de las áreas mencionadas. Por lo tanto, el resultado del estudio de factibilidad confirmó la construcción de un framework de integración y despliegue continuos basado en el paradigma DevOps.

La construcción del framework orquestó las herramientas descritas en el capítulo III, en donde el pipeline es el principal gestor que permitió realizar de manera automática los procesos de integración de los sistemas de información. De forma paralela, esto permitió gestionar de mejor manera los proyectos de desarrollo de los sistemas de información, ya que estos a partir del uso de este framework se desarrollan dentro de un marco de integración, lo que a su vez permite reducir los tiempos, el despliegue y la recopilación de información para la mejora de los sistemas de información en el caso de que se lo requiera.

Por último, la realización de un estudio exploratorio permitió validar el funcionamiento del prototipo del framework, y de la mano de una entrevista que estaba dirigida al personal del equipo de UTIC, se pudo determinar la conformidad de los miembros de la UTIC con el framework construido. Los comentarios obtenidos fueron clasificados como positivos ya que el framework ahora permite una gestión integral de los proyectos de desarrollo, desde la parte de planificación (con ayuda de las herramientas Jira y Confluence), pasando por la parte de integración y finalmente llegando a la parte del despliegue de los sistemas de información.

Recomendaciones

La problemática identificada podría ser resuelta de varias maneras, sin embargo, el prototipo de solución que se implementó responde a las necesidades de la Unidad de Tecnologías de la Información. Para lograr esto se recomienda realizar un estudio de la situación actual, en donde intervengan los miembros del equipo en este caso de UTIC y de esta manera tener claro el contexto en el que se va a trabajar.

Para construir el planteamiento de una solución de manera sólida se tomaron en cuenta las diferentes prácticas encontradas y descritas en el estado del arte, estas son de utilidad y se pueden acoplar al contexto en el que se está desarrollando el proyecto. Para este caso se tomó como recomendación el uso de herramientas Open Source ya que estas son de libre acceso y además tienen una gran comunidad que da soporte al uso de estas, sin embargo, esto no quiere decir que el uso de herramientas propietarias no sea una buena idea si no que para este contexto las herramientas de libre acceso fueron la mejor opción.

Al momento de poner en práctica el prototipo de solución, se debe realizar un seguimiento constante, ya que esto permite identificar posibles anomalías y la pronta corrección de estas. En cuanto a las pruebas con el prototipo fue fácil realizar este proceso, ya que contamos con dos ambientes, uno para pruebas y otro para producción. Estos ambientes no solo sirvieron para el despliegue de sistemas de información a modo de prueba sino también para probar la interconexión entre las aplicaciones que conforman el framework y para probar el pipeline que se encarga de realizar los procesos de integración y despliegue de sistemas de información.

Finalmente, se debe validar que el prototipo construido se encuentre completamente funcional y de solución a la problemática identificada y para ello no hubo mejor opción que realizar un estudio exploratorio en donde cada miembro del equipo de UTIC, luego de utilizar el

prototipo, emita comentarios y recomendaciones sobre el framework. De esta manera se pudo validar si el prototipo es la propuesta adecuada para solucionar la problemática identificada en un inicio.

Es recomendable que la Universidad de las Fuerzas Armadas - ESPE de prioridad a la definición de un equipo de desarrollo de sistemas de información que se encuentre estructurado con definición de roles, ya que con esto podría repotenciar el tema de la gestión de los sistemas de información.

Trabajos Futuros

Este trabajo representa los primeros pasos dentro de la Unidad de Tecnologías de la Información en cuanto a la gestión de proyectos de desarrollo de sistemas de información se refiere. Sin embargo, queda mucho camino que recorrer para llegar a una adecuada gobernanza de los sistemas de información dentro de la universidad. Para ello se plantean diferentes trabajos que se pueden realizar partiendo de este proyecto.

Como trabajos a futuro se puede alcanzar una madurez en cuanto al proceso de pruebas, es decir, se puede adecuar un ambiente para gestionar de mejor manera la calidad de los sistemas de información. En este ambiente se podría realizar el análisis de código y vulnerabilidades que puedan existir en ese mismo nivel.

Otra etapa podría contemplar la adopción de la cultura SecOps, esto con la finalidad de que se puedan distribuir decisiones de seguridad de manera rápida dentro de un modelo DevOps como el que se implementó en este proyecto.

Bibliografía

- Airaj, M. (2017). Enable cloud DevOps approach for industry and higher education. *Concurrency and Computation: Practice and Experience*, e3937.
- Álvarez, N. (2010). La Teoría de Restricciones aplicada al desarrollo de software (Master's thesis, Universidad Andina Simón Bolívar, Sede Ecuador).
- Ávila, R. (2019). Impacto del uso del sistema de control de versiones GITLAB como herramienta de monitoreo y evaluación académica de trabajos colaborativos en la Facultad de Informática, Electrónica y Comunicación.
- Ávila, W. (2013). Hacia una reflexión historica de las TIC. *Hallazgos*, 213-233.
- Bourgeois, D. (2014). *Information Systems for Business and Beyond*.
- Brito, M. (2022). Integración y entrega continua (CI/CD) con Jenkins.
- Burgueño, L. (2018). DevOps-Herramientas para el desarrollo de software. *Curso de desarrollo de aplicaciones Web usando Java y DevOps*.
- Castro, L. (2015). *Arquitectura del Software*. Cengage Learning Editores.
- Clements, P. (1996). A Survey of Architecture Description Languages. *Proceedings of the International Workshop on Software Specification and Design*.
- Cruz, M., Miranda, I., & Gonçalves, P. (2013). *Handbook of Research on ICTs and Management Systems for Improving Efficiency in Healthcare and Social Care*.
- Degiovannini, M. (2011). Comparativa de Frameworks Web. *JavaHispano.com*, 3.
- Delgado, D. (2020). EL PARADIGMA DEVOPS Y SU IMPLEMENTACIÓN EN EL DESARROLLO DE SOFTWARE. *Universidad & Ciencia*, 134-142.
- Díaz, L., Torruco, U., Martínez, M., & Varela, M. (2013). La entrevista, recurso flexible y dinámico. *Investigación en educación médica*, 162-167.
- Durango, A. (2014). *Diseño de software*. IT Campus Academy.
- Esterkin, V. (2017). Evaluación de calidad en el desarrollo de software dirigido por modelos. *Ingeniare. Revista chilena de ingeniería*, 449-463.
- Estupiñan, A., & Orjuela, D. (2019). Importancia de los Sistemas de Información. *Revista Avenir*, 10-14.
- Gamboa, J. (2018). Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software. *INNOVA Research Journal*, 20-33.
- García, N. (2018). Orquestación de contenedores con Kubernetes. *Tesis de Licenciatura*.

- Ghantous, G. B., & Gill, A. Q. (2019). An agile-devops reference architecture for teaching enterprise agile. *International Journal of Learning, Teaching and Educational Research*, 128-144.
- Gonzales, G. (2019). Study of Quality Practices in Software Development Processes in Universities of Ecuador. *In 2019 International Conference on Information Systems and Software Technologies*, 141-148.
- Henny, P. (2021). *CHAOS 2020: Beyond Infinity*. The Standish Group.
- Henry, P. (2021). *CHAOS 2020: Beyond Infinity*. The Standish Group.
- Ibrahim Fadhel, I. E., Bin Syed Idrus, S. Z., Yeoh Abdullah, M. S., Ibrahim, A. E., Omar, M., & Khred, A. (2020). Towards Development a Novel Framework of Web-Based Systems Quality Engineering by the Integration between Information Systems and Software Engineering Theories: Context of Higher Education. *Journal of Physics: Conference Series*, 1529.
- JF. (2017). *udlap*.
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/fuentes_k_jf/capitulo2.pdf
- Kitchenham, B. A., Budgen, D., & Brereton, P. (2015). Evidence-based software engineering and systematic reviews. *CRC press*.
- Maida, E., & Julián, P. (2015). Metodologías de desarrollo de software.
- Martinez, K. (2021). Análisis de procesos DevOps y migración de aplicación entre nubes.
- Martínez, R. (2015). *El Proceso de Desarrollo de Software*. IT Campus Academy.
- Medina, L., & López, W. (2015). Escoger una metodología para desarrollar software, difícil decisión. *Revista Educación en Ingeniería*, 98-109.
- Mohamad, Z., Yahaya, J., Deraman, A., & Mansor, Z. (2020). s-SCMM: A Simplified Software Configuration Management Model for Software Services in Public University. *International Journal on Advanced Science, Engineering and Information Technology*, 654-661.
- Montero, J. J. (2015). *Fábrica de Software USMP*. Lima: Universidad de San Martín de Porres.
- Olmedo, A. (2018). DevOps y análisis de performance automáticos.
- Pérez, D. (2007). Tecnologías de la información para la gestión del conocimiento. *Intangible Capital*, 31-59.
- Pérez, N. (2020). Eficiencia de los servidores web Apache 2 y Nginx: un estudio de caso. *Serie Científica de la Universidad de las Ciencias Informáticas*, 97-112.
- Pingrong, L., Xiaoquan, S., & Junqin, Y. (2021). Research on the Application of {DevOps} in the Smart Campus of Colleges and Universities. *Journal of Physics: Conference Series*.

- Purba, J. T., & Panday, R. (2017). Innovation Strategy Services Delivery: An Empirical Case Study of Academic Information Systems in Higher Education Institution. *ICSIIIT 2015: Intelligence in the Era of Big Data*, 514-525.
- RedHat. (2019). *INSTALACIÓN Y CONFIGURACIÓN DE NGINX*.
https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html/deploying_different_types_of_servers/setting-up-and-configuring-nginx_deploying-different-types-of-servers
- Mayo *GitOps*. <https://www.redhat.com/es/topics/devops/what-is-gitops>
- Republica, P. d. (2017). *Ley Organica de Empresas Publicas*.
<https://www.telecomunicaciones.gob.ec/wp-content/uploads/2017/05/LEY-ORGANICA-DE-EMPRESAS-PUBLICAS.pdf>
- Retamal, J. (2019). *Plataforma de desarrollo de aplicaciones en el DCC basada en técnicas de DevOps*.
- Reynoso, C. (2004). *Introducción a la Arquitectura de Software*. Universidad de Buenos Aires.
- Romero, P. (2006). Arquitectura de software, esquemas y servicios. *Ingeniería Industrial*.
- SHAHIN, M., ALI BABAR, M., & ZHU, L. (2017). Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices.
- Taryana, A., Setiawan, I., Fadli, A., & Murdyantoro, E. (2017). Pioneering the automation of Internal quality assurance system of higher education (IQAS-HE) using DevOps approach. *International Conference on Sustainable Information Engineering and Technology*, 259-264.
- Thuan, N. H., Drechsler, A., & Antunes, P. (2019). Construction of Design Science Research Questions. *Communications of the Association for Information Systems*, 332-363.
- Tobar, D. (2018). *DevOps para automatización de Gitlab en alta disponibilidad*.
- Valacich, J. (2012). Information systems today: Managing in the digital world. *Prentice Hall*.
- Veloz, G. (2019). Towards a Harmonized Quality Framework for in-house Software Development in Higher Education Institutions. *Ingenieria de Software e Ingenieria del Conocimiento. Jornadas Iberoamericanas, XIV*, 29-42.
- Veres, O., Kunanets, N., Pasichnyk, V., Veretennikova, N., Korz, R., & Leheza, A. (2019). Development and Operations - The Modern Paradigm of the Work of IT Project Teams. *2019 IEEE 14th International Conference on Computer Sciences and Information Technologies (CSIT)*, 103-106.
- Wang, G., & Zhao, W. (2020). Testing and verification of the integrated avionics system. *The Principles of Integrated Technology in Avionics Systems*, 467-532.
- Wieringa, R. (2014). Design Science Methodology for Information Systems and Software Engineering. *Springer Heidelberg New York Dordrecht London*.

- Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. *In Proceedings of the 18th international conference on evaluation and assessment in software engineering*, 1-10.
- Yang, D., Wang, D., Yang, D., Dong, Q., Wang, Y., Zhou, H., & Hong, D. (2020). DevOps in Practice for Education Management Information System at ECNU. *Procedia Computer Science*, 1382-1391.

ANEXOS