



**Diseño de un sistema para automatizar el proceso de registro de titulación y emisión de expedientes de estudiantes de la Universidad de las Fuerzas Armadas ESPE.**

Gómez García, Kevin Romeo

Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

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

Ing. Diaz Zuñiga, Magi Paul

27 de marzo de 2023

## TrabajodeTitulacionFinal\_Gomez\_Ke...

### Scan details

Scan time:  
August 23th, 2023 at 20:2 UTC

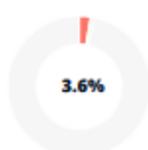
Total Pages:  
43

Total Words:  
10588

MAGI  
PAUL DIAZ  
ZUNIGA

Firmado digitalmente por  
MAGI PAUL DIAZ  
ZUNIGA  
Fecha: 2023.11.16  
11:30:02 -05'00'

### Plagiarism Detection



Types of plagiarism		Words
Identical	2.6%	277
Minor Changes	0.3%	30
Paraphrased	0.7%	70
Omitted Words	0%	0

### AI Content Detection



Text coverage

- AI text
- Human text

### Plagiarism Results: (44)

**Browsing Lima Norte by Author "Villavicencio Meza..."** 0.5%

<https://repositorio.ucv.edu.pe/handle/20.500.12692/42042/b...>

Politics Statistics English español English Login Toggle navigation Toggle navigation ...

**Browsing Maestría en Ingeniería de Sistemas con m...** 0.5%

<https://repositorio.ucv.edu.pe/handle/20.500.12692/42039/b...>

Politics Statistics English español English Login Toggle navigation Toggle navigation ...

**Listar Lima Norte por tema "Universidad Nacional ..."** 0.5%

<https://repositorio.ucv.edu.pe/handle/20.500.12692/42042/b...>

Política Estadísticas español español English Acceder Cambiar navegación Cambiar navegación ...



**Departamento de Ciencias de la Computación**

**Carrera de Tecnologías de la Información**

### **Certificación**

Certifico que el trabajo de titulación: **“Diseño de un sistema para automatizar el proceso de registro de titulación y emisión de expedientes de estudiantes de la Universidad de las Fuerzas Armadas ESPE.”** fue realizado por el señor Gómez García, Kevin Romeo; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

**Sangolquí, 15 de noviembre del 2023**

.....  
Ing. Paul Diaz Zuñiga  
C.C.: 1707249072



**Departamento de Ciencias de la Computación**

**Carrera de Tecnologías de la Información**

**Responsabilidad de Autoría**

Yo, Gómez García, Kevin Romeo, con cédula de ciudadanía n° 1725211104 declaro que el contenido, ideas y criterios del trabajo de titulación: **"Diseño de un sistema para automatizar el proceso de registro de titulación y emisión de expedientes de estudiantes de la Universidad de las Fuerzas Armadas ESPE"** es de mi 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, 15 de noviembre del 2023**

.....  
Gómez García, Kevin Romeo

C.C.: 1725211104



Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

#### Autorización de Publicación

Yo, Gómez García, Kevin Romeo, con cedula de ciudadanía n° 1725211104 autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **"Diseño de un sistema para automatizar el proceso de registro de titulación y emisión de expedientes de estudiantes de la Universidad de las Fuerzas Armadas ESPE"** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 15 de noviembre del 2023

.....  
Gómez García, Kevin Romeo

C.C.: 1725211104

## Dedicatoria

Dedico este trabajo de titulación con profundo cariño y gratitud a aquellos que han sido pilares fundamentales en mi trayectoria académica y personal.

A mi hermana, cuyo amor incondicional y aliento constante me han impulsado a superar desafíos y alcanzar metas que parecían inalcanzables.

A mi madre y a mi padre, cuyo inquebrantable apoyo y sabias palabras han sido mi guía a lo largo de esta travesía educativa.

A Jessika, quien iluminó las noches de desarrollo con su apoyo incansable, recordándome que puedo lograrlo incluso cuando las dificultades parecían abrumadoras.

A mis amigos Javier, Luis y David, cuya amistad y aliento me han dado fuerzas para perseverar, recordándome que cada paso hacia adelante es un logro valioso.

No puedo dejar de mencionar a Joel y Violeta, quienes estuvieron presentes desde el inicio de esta etapa y cuya amistad me ha nutrido en múltiples formas a lo largo de este viaje.

A mi familia en su conjunto, por ser mi red de seguridad y mi fuente constante de motivación. Sus palabras de aliento y apoyo incondicional han sido el viento bajo mis alas.

Este trabajo es el resultado de muchas manos extendidas y corazones generosos. A todos ustedes, mi más sincero agradecimiento. Sin su presencia en mi vida, este logro no sería posible.

## **Agradecimiento**

En el proceso de culminar este trabajo de titulación, hubo individuos y grupos que deseo expresar mi más sincero agradecimiento por su contribución invaluable. Sus esfuerzos y apoyo han dejado una marca imborrable en este logro.

En primer lugar, deseo expresar mi profundo agradecimiento al Ingeniero Paul Diaz. Su dedicación y firme compromiso en proporcionarme las herramientas y oportunidades necesarias para completar este trabajo han sido fundamentales. Su mentoría y guía han sido una fuente constante de inspiración y motivación a lo largo de este trayecto.

Agradezco también a la Ingeniera Mafla y al Departamento de Unidad de Admisión y Registro por brindarme la valiosa oportunidad de presentar una mejora que contribuirá al crecimiento y desarrollo de nuestra universidad. Su apertura a nuevas ideas y su respaldo en esta iniciativa son ejemplos inspiradores de liderazgo y visión.

Un agradecimiento especial va dirigido a los ingenieros del programa de Ingeniería en Tecnologías de la Información (ITIN). Su compromiso en impartir conocimiento y habilidades de manera clara y efectiva ha sido esencial para el desarrollo exitoso de este proyecto. Las bases que me han proporcionado no solo han enriquecido mi formación, sino que también han sido la piedra angular de este trabajo.

Este trabajo no habría sido posible sin la contribución directa e indirecta de cada uno de ustedes. A todos, les estoy profundamente agradecido.

## índice de contenidos

Dedicatoria .....	6
Agradecimiento.....	7
<b>índice de contenidos</b> .....	8
<b>índice de Tabla</b> .....	10
<b>índice de figuras</b> .....	11
<b>Capítulo I</b> .....	15
<b>Introducción</b> .....	15
<b>Antecedentes</b> .....	15
<b>Planteamiento del problema</b> .....	16
<b>Objetivos</b> .....	17
<b>Objetivo general</b> .....	17
<b>Objetivos específicos</b> .....	17
<b>Alcance</b> .....	18
<b>Hipótesis</b> .....	19
<b>Capítulo II</b> .....	20
<b>Marco Teórico</b> .....	20
<b>Fundamento Teórico</b> .....	23
<b>Estado del arte</b> .....	25
<b>Estrategia de búsqueda</b> .....	25
<b>Construcción y afinación de la cadena de búsqueda</b> .....	25
<b>Selección de estudios</b> .....	25
<b>Capítulo III</b> .....	31
<b>Metodología</b> .....	31
<b>Herramientas y Tecnologías</b> .....	31
<b>Metodología de Desarrollo de Software: Kanban</b> .....	32
<b>Capítulo IV</b> .....	34
<b>Desarrollo</b> .....	34
<b>Introducción</b> .....	34
<b>Arquitectura del Sistema</b> .....	35
<b>Desarrollo del código</b> .....	39
<b>Backend (Servidor)</b> .....	39
<b>Frontend (Cliente)</b> .....	67

<b>Capítulo V</b> .....	75
<b>Análisis de Resultados</b> .....	75
<b>Introducción</b> .....	75
<b>Análisis Comparativo: Reducción del Tiempo de Procesamiento</b> .....	75
<b>Proceso Manual</b> .....	75
<b>Proceso con el Sistema Workflow vía Web</b> .....	76
<b>Resultados y Beneficios</b> .....	76
<b>Conclusión</b> .....	77
<b>Análisis de Eficiencia Administrativa</b> .....	77
<b>Proceso Manual y Desafíos Administrativos</b> .....	77
<b>Automatización de Flujos de Trabajo y Simplificación</b> .....	77
<b>Resultados y Beneficios</b> .....	78
<b>Conclusión</b> .....	78
<b>Capítulo VI</b> .....	79
<b>Conclusiones</b> .....	79
<b>Recomendaciones</b> .....	80
<b>Referencias</b> .....	83
<b>Apéndices</b> .....	84

**índice de Tabla**

Tabla 1 Estudios relacionados con el tema propuesto .....	26
---	----

**índice de figuras**

<b>Figura 1</b> IMPLEMENTACION DE LA ARQUITECTURA .....	37
<b>Figura 2</b> Grades Model.....	41
<b>Figura 3</b> Student Model .....	42
<b>Figura 4</b> Record Model Parte 1 .....	43
<b>Figura 5</b> Record Model Parte 2 .....	44
<b>Figura 6</b> Record Model Parte 3 .....	45
<b>Figura 7</b> Grades DTO.....	46
<b>Figura 8</b> Student DTO .....	47
<b>Figura 9</b> Record DTO .....	48
<b>Figura 10</b> Grades DAO.....	49
<b>Figura 11</b> Student DAO .....	50
<b>Figura 12</b> Record DAO Parte 1 .....	51
<b>Figura 13</b> Record DAO Parte 2 .....	52
<b>Figura 14</b> Grades Service .....	53
<b>Figura 15</b> StudentService Parte 1.....	54
<b>Figura 16</b> StudentService Parte 2.....	55
<b>Figura 17</b> StudentService Parte 3.....	56
<b>Figura 18</b> RecordService Parte 1 .....	56
<b>Figura 19</b> RecordService Parte 2 .....	57
<b>Figura 20</b> RecordService Parte 3 .....	58
<b>Figura 21</b> RecordGeneratorService.....	59
<b>Figura 22</b> gradesLoader Controller.....	60
<b>Figura 23</b> studentRequest Controller Parte 1 .....	61
<b>Figura 24</b> studentRequest Controller Parte 2 .....	62
<b>Figura 25</b> studentRequest Controller Parte 3 .....	63

<b>Figura 26</b> directorResponse Controller Parte 1.....	64
<b>Figura 27</b> directorResponse Controller Parte 2.....	65
<b>Figura 28</b> directorResponse Controller Parte 3.....	66
<b>Figura 29</b> main.module.....	68
<b>Figura 30</b> Main-routing-module .....	69
<b>Figura 31</b> Estructura.....	70
<b>Figura 32</b> student-request .....	71
<b>Figura 33</b> director-response .....	72
<b>Figura 34</b> record-generator.....	73
<b>Figura 35</b> workflow .....	74
<b>Figura 36</b> main view .....	74

## Resumen

Este proyecto tuvo como objetivo el diseño de un sistema tipo Workflow vía web para regular y agilizar los procesos desde el cumplimiento de la malla curricular hasta la titulación de los estudiantes de tercer nivel en la Universidad de las Fuerzas Armadas ESPE. El sistema se aplicó a todas las modalidades de estudio, incluyendo la modalidad presencial, en línea y dual, con un enfoque especial en las Carreras que contemplan la Unidad de Integración Curricular en su malla. El sistema mejoró la eficiencia administrativa, proporcionó una guía clara para los estudiantes durante su proceso de titulación y aseguró adecuado desarrollo del proceso académico en referencia.

Para lograr estos objetivos, se ha realizado un análisis exhaustivo de los flujos de trabajo existentes, identificando áreas de mejora y posibles ineficiencias en el proceso de titulación. Con base en esta información, se ha diseñado una arquitectura de sistema web que permita automatizar y agilizar el registro de cumplimiento de requisitos académicos, la generación de informes y certificados, así como la gestión de documentos relacionados con el proceso de titulación.

El sistema se fundamenta en conceptos teóricos de Workflow y tecnologías de automatización de procesos, lo que garantiza una implementación eficiente y efectiva.

Además, se ha considerado la fundamentación legal para asegurar el cumplimiento de las regulaciones y normativas pertinentes al proceso de titulación en la universidad.

La implementación de este sistema tipo Workflow vía web promete optimizar los procedimientos, reducir errores y proporcionar una experiencia más ágil para los estudiantes durante su proceso de titulación. Esta solución contribuyó significativamente a mejorar la eficiencia administrativa y a fortalecer la calidad del sistema educativo en la Universidad de las Fuerzas Armadas ESPE.

*Palabras clave:* Workflow, titulación, automatización, proceso académico.

### **Abstract**

This project aimed to design a web-based Workflow system to regulate and streamline the processes from the fulfillment of the curriculum to the graduation of third-level students at the Universidad de las Fuerzas Armadas ESPE. The system was applied to all modes of study, including in-person, online, and dual modalities, with a special focus on programs that include the Curricular Integration Unit in their curriculum. The system improved administrative efficiency, provided clear guidance for students during their graduation process, and ensured the proper development of the academic process in reference. To achieve these objectives, a comprehensive analysis of existing workflows was conducted, identifying areas for improvement and possible inefficiencies in the graduation process. Based on this information, a web-based system architecture was designed to automate and expedite the registration of academic requirements, the generation of reports and certificates, as well as the management of documents related to the graduation process. The system is based on theoretical concepts of Workflow and process automation technologies, ensuring efficient and effective implementation. Furthermore, legal foundations have been considered to ensure compliance with regulations and relevant norms related to the graduation process at the university. The implementation of this web-based Workflow system promises to optimize procedures, reduce errors, and provide a more streamlined experience for students during their graduation process. This solution significantly contributed to improving administrative efficiency and strengthening the quality of the educational system at the Universidad de las Fuerzas Armadas ESPE

*Keywords:* Workflow, degree, automation, academic process

## Capítulo I

### Introducción

#### Antecedentes

En el entorno académico, la gestión de los procesos de titulación y emisión de expedientes de estudiantes de tercer nivel ha sido tradicionalmente un proceso manual y complejo en muchas instituciones educativas. La Universidad de las Fuerzas Armadas ESPE no ha sido una excepción, enfrentando desafíos relacionados con la eficiencia administrativa y la experiencia estudiantil en este ámbito.

La automatización de procesos mediante sistemas tipo Workflow vía web ha surgido como una solución innovadora para abordar estos desafíos. Investigaciones previas han demostrado que la implementación de sistemas Workflow en el ámbito académico puede agilizar los procedimientos, reducir errores y mejorar la experiencia estudiantil durante el proceso de titulación.

Además, proyectos similares de automatización de procesos en otras instituciones educativas han evidenciado beneficios significativos, incluyendo una mayor transparencia, una gestión más eficiente de la información, y una reducción en los tiempos de respuesta para la emisión de documentos y certificados.

Con base en estos antecedentes y la creciente necesidad de optimizar los procesos académicos en la Universidad de las Fuerzas Armadas ESPE, surge la relevancia de este proyecto de diseño de un sistema tipo Workflow vía web. Esta propuesta busca abordar las limitaciones actuales, mejorar la eficiencia administrativa y brindar una experiencia más ágil y satisfactoria a los estudiantes durante su proceso de titulación.

## Planteamiento del problema

El proceso de titulación y emisión de expedientes de estudiantes de tercer nivel en la Universidad de las Fuerzas Armadas ESPE enfrenta diversos desafíos que afectan tanto a los estudiantes como al personal administrativo. Actualmente, la gestión de estos procesos puede ser manual y propensa a errores, lo que conlleva a retrasos y dificultades en la entrega de documentos y certificados. Además, la falta de un sistema automatizado y estandarizado dificulta el seguimiento adecuado del progreso académico de los estudiantes, lo que podría afectar su experiencia educativa.

## Justificación e importancia

El diseño de un sistema tipo Workflow vía web para automatizar y regular los procesos entre el cumplimiento de la malla curricular y la titulación de los estudiantes de la Universidad de las Fuerzas Armadas ESPE es una necesidad imperante. La implementación de este sistema ofrecerá numerosos beneficios, incluyendo:

- **Optimización y agilización de procesos:** La automatización de tareas y flujos de trabajo reducirá significativamente los tiempos de respuesta y minimizará la carga administrativa, permitiendo que el personal se enfoque en tareas más estratégicas.
- **Gestión eficiente de la información:** El sistema permitirá centralizar y organizar la información relacionada con el cumplimiento de la malla curricular y la titulación, facilitando el acceso y actualización en tiempo real, lo que mejorará la toma de decisiones basadas en datos precisos.
- **Reducción de errores y garantía de precisión:** La automatización de procesos reducirá los errores manuales en la generación de informes y documentos, asegurando la precisión y confiabilidad de los expedientes y certificados de los estudiantes.

- **Facilitar el cumplimiento normativo:** El sistema establecerá lineamientos claros y estandarizados que aseguren el cumplimiento de las regulaciones y normativas pertinentes al proceso de titulación.
- **Mejora en la experiencia estudiantil:** Los estudiantes se beneficiarán de un proceso de titulación más ágil y transparente, lo que les permitirá concentrarse en su formación académica y en la preparación para su titulación.

## Objetivos

### Objetivo general

El objetivo de este proyecto es diseñar y desarrollar un sistema tipo Workflow vía web para automatizar y optimizar los procesos de cumplimiento de la malla curricular y titulación de estudiantes de tercer nivel en la Universidad de las Fuerzas Armadas ESPE. El sistema proporcionará una gestión eficiente de la información académica, facilitará la comunicación entre estudiantes y personal administrativo, y contribuirá a mejorar la experiencia estudiantil y la eficiencia administrativa en cada período académico ordinario.

### Objetivos específicos

- Realizar un levantamiento de información de los procesos y las instancias por las que atraviesan los documentos que deben ser legalizados y para titulación de estudiantes.
- Diseñar un sistema web tipo workflow, que permita automatizar y mejorar los procesos de titulación.

## Alcance

El alcance del proyecto consiste en diseñar un sistema tipo Workflow vía web para regular y automatizar los procesos desde el cumplimiento de la malla curricular hasta la titulación de estudiantes de tercer nivel en todas las modalidades de estudio de la institución educativa. El proyecto abarca diversas etapas clave:

- **Análisis de procesos:** Se llevará a cabo un minucioso análisis de los flujos de trabajo actuales relacionados con el cumplimiento de la malla curricular y titulación. Esto implica identificar y documentar detalladamente los pasos, tareas y responsabilidades involucradas en cada etapa, con el objetivo de encontrar posibles oportunidades de mejora, puntos críticos y cuellos de botella.
- **Diseño del sistema:** Se procederá a diseñar la arquitectura del sistema tipo Workflow vía web, tomando en cuenta las mejores prácticas de desarrollo de software y las necesidades específicas de la institución. Esto incluirá definir los roles y permisos de los usuarios, los flujos de trabajo y las interacciones entre los diferentes actores involucrados en el proceso.
- **Desarrollo de funcionalidades:** Se implementarán las funcionalidades necesarias para automatizar y regular los procesos que conectan el cumplimiento de la malla curricular y la titulación. Esto engloba aspectos como el registro del cumplimiento de requisitos académicos, la generación de informes y certificados, la gestión de documentos y la facilitación de la comunicación entre los usuarios del sistema.
- **Integración con sistemas existentes:** En caso de que la institución cuente con sistemas o plataformas ya establecidas, se trabajará en la integración del sistema tipo Workflow con estas soluciones, garantizando una interoperabilidad y transferencia de datos eficiente.

- **Pruebas y validación:** Se llevarán a cabo pruebas exhaustivas del sistema para asegurar su correcto funcionamiento, estabilidad y capacidad de respuesta. Los resultados obtenidos se validaron y se realizaron ajustes y correcciones si era necesario antes de su implementación definitiva.
- **Implementación y capacitación:** Una vez que el sistema esté listo, se procederá a su implementación en la institución educativa. Se brindará capacitación al personal administrativo y a los usuarios involucrados para asegurar un uso adecuado y una adopción exitosa por parte de todos los actores.

Es esencial tener en cuenta que el alcance del proyecto podría estar sujeto a cambios según los recursos disponibles, los plazos establecidos y las prioridades definidas. Por tanto, se recomienda mantener una comunicación fluida con los interesados y una gestión efectiva del alcance para garantizar la exitosa entrega del sistema dentro de los límites establecidos.

## **Hipótesis**

La hipótesis planteada sostiene que la implementación del sistema tipo Workflow vía web en la Universidad de las Fuerzas Armadas ESPE mejorará la eficiencia y agilidad del proceso de titulación al reducir los tiempos de respuesta y optimizar la gestión administrativa. Además, se anticipa que el sistema brindará a los estudiantes una experiencia más satisfactoria y transparente durante su proceso de titulación.

## Capítulo II

### Marco Teórico

La automatización en las instituciones educativas tiene una gran importancia debido a que puede aportar una serie de beneficios significativos que mejoran la eficiencia y la precisión en el manejo de registros y expedientes de los estudiantes. Algunas de las ventajas más relevantes son las siguientes:

**Mayor rapidez y eficiencia:** La automatización permite realizar tareas repetitivas y administrativas de manera más rápida y sin errores humanos. Los procesos manuales pueden ser lentos y propensos a errores, mientras que con la automatización se aceleran los flujos de trabajo y se optimizan los recursos.

**Reducción de errores:** Al eliminar la intervención manual en ciertas etapas del proceso, se minimiza la posibilidad de cometer errores en la recopilación, procesamiento y almacenamiento de datos. Esto conduce a una mayor precisión en los registros y expedientes de los estudiantes.

**Mejora en la gestión de información:** La automatización permite centralizar y gestionar la información de manera más eficiente, lo que facilita el acceso a los datos cuando sea necesario. Esto resulta en una toma de decisiones más informada y una mejor planificación institucional.

**Acceso en línea para estudiantes y personal:** Al implementar sistemas automatizados, los estudiantes y el personal pueden acceder a la información relevante en línea en cualquier momento y lugar. Esto fomenta la transparencia y la colaboración, además de reducir la carga administrativa para el personal encargado.

**Integración de sistemas:** La automatización puede permitir la integración de diferentes sistemas y plataformas utilizados en la institución educativa, como sistemas de información

estudiantil, plataformas de aprendizaje en línea, sistemas de biblioteca, entre otros. Esto mejora la interoperabilidad y la eficiencia global de los procesos.

**Facilita el seguimiento del progreso estudiantil:** Los sistemas automatizados pueden ayudar a realizar un seguimiento más efectivo del progreso académico de los estudiantes, sus calificaciones, asistencia y otros aspectos relevantes para su titulación.

**Cumplimiento normativo y regulatorio:** Al automatizar los procesos de registro y expedición de documentos, se facilita el cumplimiento de las regulaciones y normativas establecidas por las autoridades educativas y gubernamentales.

En última instancia, la automatización conlleva ahorro de tiempo y recursos, ya que al reducir la carga administrativa y eliminar procesos manuales, la institución puede destinar esos recursos a otras áreas clave de mejora y desarrollo. En resumen, la automatización es una herramienta esencial para mejorar la eficiencia, precisión y calidad en la gestión de registros y expedientes en las instituciones educativas.

### **Herramientas y Tecnologías utilizadas**

- **Java 17:** Java 17 incluye una serie de nuevas características y mejoras, como:
  - ❖ Un nuevo compilador y optimizador que mejoran el rendimiento de las aplicaciones Java.
  - ❖ Nuevas API para trabajar con archivos y sistemas de archivos.
  - ❖ Un nuevo sistema de seguridad que hace que las aplicaciones Java sean más seguras.
  - ❖ Una serie de correcciones de errores y mejoras de estabilidad.
- **Groovy:** Groovy es un lenguaje de programación dinámico diseñado para ser compatible con Java y aprovechar su ecosistema mientras agrega características y sintaxis más concisas y expresivas. Groovy se ejecuta en la Plataforma Java (Java Virtual Machine, JVM) y se utiliza comúnmente como un lenguaje de scripting,

aunque también es capaz de ser utilizado para el desarrollo de aplicaciones más complejas.

- **Oracle 12c:** Oracle 12c es una versión de la base de datos relacional Oracle Database, desarrollada por Oracle Corporation. El "c" en 12c hace referencia a "Cloud", ya que esta versión se enfoca en ofrecer capacidades y características orientadas a la nube, aunque también es adecuada para implementaciones on-premise (en las instalaciones del cliente).
- **Docker:** Docker es una plataforma de virtualización a nivel de sistema operativo que se utiliza para desarrollar, implementar y ejecutar aplicaciones en contenedores. Los contenedores son entornos de ejecución ligeros y autónomos que encapsulan una aplicación y todas sus dependencias, incluidas bibliotecas y configuraciones, en una unidad única y portátil.
- **Jenkins:** Jenkins es una plataforma que ayuda a los equipos de desarrollo a automatizar y agilizar el proceso de construcción, prueba y despliegue de software, lo que contribuye a mejorar la calidad del software y la eficiencia del desarrollo.
- **Gradle:** es una herramienta de construcción y automatización que proporciona un enfoque flexible y basado en el lenguaje para la gestión de proyectos y tareas relacionadas con el desarrollo de software. Su capacidad para manejar proyectos de diferentes tamaños y su enfoque en la eficiencia y la personalización lo hacen una opción popular entre los desarrolladores y equipos de desarrollo.
- **GitLab:** es una plataforma integral de desarrollo de software que proporciona herramientas para la gestión del ciclo de vida del desarrollo, la colaboración en equipos y la implementación continua. Su característica más prominente es el control de versiones basado en Git, que permite a los equipos de desarrollo rastrear y gestionar cambios en el código fuente de manera efectiva. Además del control de versiones, GitLab ofrece una variedad de características para respaldar el desarrollo colaborativo y la entrega continua.

- **IntelliJ IDEA:** diseñado específicamente para programadores y desarrolladores de software, y proporciona un conjunto completo de herramientas para escribir, depurar y mantener aplicaciones en diferentes lenguajes de programación, con un enfoque particular en lenguajes como Java, Kotlin, Groovy y otros.
- **Git:** El concepto principal detrás de Git es el de "repositorio", que es una base de datos que almacena todas las versiones de los archivos de un proyecto y registra los cambios realizados en ellos a lo largo del tiempo

### Fundamento Teórico

El fundamento teórico de este proyecto se fundamenta en conceptos clave que respaldan y enriquecen el diseño del sistema tipo Workflow vía web para automatizar el proceso de registro de titulación en la Universidad de las Fuerzas Armadas ESPE. A continuación, se describen los conceptos fundamentales:

- **Flujo de trabajo (Workflow):** Se refiere a la secuencia de pasos y actividades ordenadas de forma lógica que se llevan a cabo para completar una tarea o proceso específico. Los flujos de trabajo pueden ser manuales o automatizados y definen la ruta que deben seguir los datos o documentos, las acciones a realizar y los roles involucrados en cada etapa del proceso. La adopción de flujos de trabajo bien definidos en el sistema permitirá una gestión estructurada y consistente del proceso de registro de titulaciones.
- **Automatización de procesos académicos:** Hace referencia al uso de tecnologías y herramientas informáticas para reemplazar actividades manuales y repetitivas en los procesos académicos. La automatización agiliza las tareas administrativas, minimiza los errores y garantiza un flujo de trabajo más eficiente. En el contexto del proyecto, la automatización de procesos académicos será fundamental para optimizar el proceso de titulación y agilizar la emisión de expedientes.

- **Gestión de documentos electrónicos:** Consiste en la administración y control de documentos en formato electrónico a lo largo de su ciclo de vida. La gestión electrónica de documentos facilita el acceso, búsqueda, almacenamiento y preservación de información de manera organizada y segura. La implementación de un sistema de gestión de documentos será esencial para asegurar la trazabilidad y disponibilidad de los expedientes de titulación de los estudiantes.
- **Interoperabilidad:** Hace referencia a la capacidad de diferentes sistemas o plataformas para intercambiar y utilizar información de manera efectiva y sin restricciones. En el contexto del proyecto, la interoperabilidad será crucial para integrar el sistema tipo Workflow con otras herramientas o bases de datos existentes en la universidad, permitiendo el flujo de datos de manera transparente y eficiente.
- **Experiencia del usuario (UX):** Se refiere a la interacción entre los usuarios y el sistema, y busca asegurar que los usuarios puedan utilizar el sistema de manera fácil, eficiente y satisfactoria. El diseño centrado en la experiencia del usuario será esencial para garantizar la adopción exitosa del sistema por parte del personal administrativo y los estudiantes.

La comprensión y aplicación de estos conceptos en el diseño del sistema permitirá mejorar la eficiencia administrativa, reducir los tiempos de respuesta, brindar una experiencia más satisfactoria a los estudiantes y asegurar una adecuada gestión de los procesos de registro de titulación en la Universidad de las Fuerzas Armadas ESPE.

## **Estado del arte**

### **Estrategia de búsqueda**

Asegurarnos de tener una comprensión clara y específica del tema, identificar palabras clave y conceptos principales relacionados con el tema, con esto logre obtener los artículos y repositorios más relevantes.

### **Construcción y afinación de la cadena de búsqueda**

Para obtener el resultado de la búsqueda use las siguientes palabras clave: (workflow OR flujo de trabajo) AND (gestión trámites académicos OR administración trámites académicos) AND unidad integración curricular

### **Selección de estudios**

En la cadena de búsqueda proporcionado dio 9.050 resultados relacionados con el tema, como fue un gran número de resultados use un filtro para trabajar con un número manejable de artículos quedándonos 30

Filtros para tomar en cuenta para los artículos seleccionados

- Publicados desde el 2015
- Hacen referencia a la gestión de trámites académicos
- Artículos que trabajan con workflow

**Tabla 1***Estudios relacionados con el tema propuesto*

Código	Título	Cita
RE1	Sistema web integrado para la gestión de trámites académico-administrativos en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial	(Aldás Flores, Clay Barcia Gallardo, Guillermo Alexander, 2023)
RE2	Transformación digital en los procesos de la Unidad de Integración Curricular de la Facultad de Ciencias Administrativas de la Universidad Técnica de Ambato	(Moreno Lalama, Lizeth Guadalupe & Carrasco Morales, Cynthia Nicole, 2023)
RE3	Aplicación web para el servicio de trámites académicos de la UNACH usando una arquitectura basada en microservicios	(Alvarado Zambrano, Erick Alexis, 2023)
RE4	Las tecnologías de información y comunicación (Tic's) y su aporte a la actividad secretarial de la Unidad de Estudios a Distancia de la Universidad Técnica Estatal de Quevedo	(Aguayo Álvarez, Lucia Alejandra, 2015)
RE5	Aplicación de Sistemas Workflow en la Gestión Académica	(Suquitana Fajardo, Byron Fabian, 2018)

Código	Título	Cita
RE6	Diseño de un programa de mejora de la gestión académica en la Universidad Nacional Federico Villarreal	(Villavicencio Meza, Jorge Luis, 2020)

**Nota:**

Esta tabla muestra algunos artículos y tesis relacionados al tema de estudio.

- **RE1 (Aldás Flores, Clay Barcia Gallardo, Guillermo Alexander, 2023) Sistema web integrado para la gestión de trámites académico administrativos en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial**

Este proyecto de investigación surge con el propósito de mejorar la gestión de la documentación académico-administrativa en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial (FISEI), satisfaciendo las necesidades y expectativas de los usuarios. La propuesta consiste en implementar un sistema web integrado para la gestión de trámites, utilizando la metodología Extreme Programming (XP), que se divide en cuatro fases: exploración, planificación, iteraciones y pruebas

El sistema web desarrollado se basa en el patrón de arquitectura en capas, empleando React Js para el cliente y Laravel para el servidor, con una base de datos MySQL y la API de Google Drive. El resultado es una plataforma que integra las áreas de Unidad de Titulación e Integración Curricular, Consejo Académico y Secretaría General de Facultad, con el fin de agilizar y asegurar la eficiencia de los trámites

La implementación de este sistema proporciona una valiosa ayuda en la gestión de documentos para las áreas mencionadas. Facilita la visualización del estado de los

documentos, mejorando la organización tanto a nivel individual como colectivo para el personal involucrado. Además, optimiza el tiempo y los recursos de estudiantes y docentes al realizar los trámites requeridos de manera más eficiente y segura.

- **RE2 (Moreno Lalama, Lizeth Guadalupe & Carrasco Morales, Cynthia Nicole, 2023) Transformación digital en los procesos de la Unidad de Integración Curricular de la Facultad de Ciencias Administrativas de la Universidad Técnica de Ambato**

Esta investigación busca mejorar la gestión del proceso de titulación mediante la automatización de actividades utilizando tecnologías BPM para la transformación digital de la Unidad de Integración Curricular en la Facultad de Ciencias Administrativas de la Universidad Técnica de Ambato. Se proponen plataformas digitales basadas en tecnologías BPM para la automatización de procesos de titulación, lo que mejoraría el desempeño y la productividad de la institución. Los hallazgos indican que la unidad no utiliza un sistema de seguimiento de estudiantes, por lo que la implementación de un sistema sería una ventaja competitiva que brinde una mejor atención y acceso a la información académica.

- **RE3 (Alvarado Zambrano, Erick Alexis, 2023) Aplicación web para el servicio de trámites académicos de la UNACH usando una arquitectura basada en microservicios**

En la Universidad Nacional de Chimborazo, la constante necesidad de actualizar los procesos de las secretarías de las carreras ha llevado a la identificación de la urgente creación de un aplicativo que agilice la obtención de certificados académicos y reduzca los tiempos de atención a los estudiantes. Para abordar esta necesidad, la investigación se enfocó en desarrollar un aplicativo web para los trámites académicos de la UNACH, utilizando la metodología SCRUM y una arquitectura basada en microservicios.

Durante la fase de planificación, se consideraron los requerimientos funcionales del módulo y se asignaron las tareas mediante el Product Backlog. Para evaluar el rendimiento del aplicativo, se utilizó la herramienta Jmeter, y para validar sus características de calidad, se empleó el modelo de calidad FURPS. Los resultados mostraron una eficacia del 95%, un tiempo de respuesta de 5 segundos y un uso de recursos del 25%. Estos resultados confirman que el aplicativo cumple con los criterios de calidad establecidos

- **RE4 (Aguayo Álvarez, Lucia Alejandra, 2015) Las tecnologías de información y comunicación (Tic's) y su aporte a la actividad secretarial**

Este trabajo se llevó a cabo en la Unidad de Estudios a Distancia de la Universidad Técnica Estatal de Quevedo, ubicada en la Provincia de Los Ríos, cantón Quevedo. Su objetivo fue determinar el impacto de las Tecnologías de la Información y Comunicación (TIC's) en la actividad secretarial de la Unidad.

Se empleó el método inductivo-deductivo, basándose en contenidos de libros sobre TIC's, administración y telecomunicaciones para diseñar la propuesta. Además, se analizaron factores a través de encuestas aplicadas a estudiantes de la UED. Se recomendó la implementación de una herramienta de trámites, denominada "Diseño de la estructura metodológica del Workflow", para agilizar los procesos académico-administrativos de los estudiantes.

- **RE5 (Suquitana Fajardo, Byron Fabian, 2018) Aplicación de Sistemas Workflow en la Gestión Académica**

Este proyecto muestra la utilidad de una herramienta Workflow para el modelamiento y automatización de procesos administrativos en la gestión académica. La herramienta ha sido implementada y modernizada para adaptarse a las exigencias del mercado globalizado y en constante cambio, donde las nuevas tendencias tecnológicas impulsan cambios en los procesos organizacionales.

En la gestión académica, se han identificado varios procesos administrativos con inconsistencias, trabas y retrasos en la ejecución y entrega de trámites, lo que afecta a los usuarios y la reputación de la Institución. La herramienta de aplicación Workflow propone agilizar estos trámites mediante la automatización, complementando el manejo de datos estructurados. Esto permite mejorar actividades como revisión, aprobación y traslado de documentos, optimizando los procesos y su relación con los sistemas internos.

En general, el artículo presenta una solución integral basada en tecnologías y sistemas de información que aumenta la productividad y facilita el flujo oportuno y ágil de información entre los departamentos participantes en el Workflow. Distribuye elementos esenciales como datos, documentos, tareas y más, que permiten a los participantes alcanzar sus objetivos de manera eficiente.

- **RE6 (Villavicencio Meza, Jorge Luis, 2020) Diseño de un programa de mejora de la gestión académica**

El propósito de esta investigación fue diseñar un programa de mejora para la gestión académica de la Universidad Nacional Federico Villarreal, enfocándose en mejorar la eficacia y rapidez de los procesos. Se utilizó un enfoque cualitativo y la modalidad investigación-acción, empleando técnicas de entrevista, observación y análisis documental.

El diagnóstico reveló desconocimiento de los procesos para acceder a la información, generando demoras y sobrecostos. Se identificaron áreas de mejora en la matrícula, el aprendizaje y los trámites, donde la definición clara de los procesos y la implementación de tecnologías como la firma digital podrían agilizar la gestión.

El diseño del programa de mejora se centró en definir el flujo de procesos para la matrícula virtual, el aprendizaje virtual y los trámites académicos, lo que facilitaría el acceso rápido a la información. Se concluyó que el programa de mejora permitiría reducir tiempo y recursos en la gestión académica.

Se recomienda a la universidad implementar el programa de mejora para optimizar la gestión de la matrícula virtual, el aprendizaje virtual y los trámites, definiendo roles claros para cada unidad orgánica.

## Capítulo III

### Metodología

#### Herramientas y Tecnologías

En el desarrollo del sistema para automatizar el proceso de registro de titulación en la Universidad de las Fuerzas Armadas ESPE, se emplearon diversas herramientas y tecnologías. A continuación, se proporciona una descripción detallada de las principales herramientas utilizadas:

- **Java 17:** Se seleccionó Java 17 como el lenguaje de programación principal para el desarrollo del sistema. Java ofrece una plataforma robusta y confiable que permite la creación de aplicaciones web escalables y seguras.
- **Groovy:** Groovy se utilizó como un lenguaje de scripting para implementar tareas automatizadas y simplificar el desarrollo de ciertos componentes del sistema.
- **Oracle 12c:** Se empleó Oracle 12c como el sistema de gestión de bases de datos para almacenar y gestionar la información relacionada con los procesos de titulación de los estudiantes.
- **Docker:** Docker se utilizó para crear y desplegar contenedores virtuales, lo que facilitó el desarrollo, prueba y despliegue del sistema en diferentes entornos.
- **Jenkins:** Jenkins se utilizó como servidor de integración continua para automatizar el proceso de compilación, pruebas y despliegue del sistema, lo que garantiza una entrega continua y una mayor eficiencia en el ciclo de desarrollo.
- **Gradle:** Se empleó Gradle como una herramienta de construcción que simplifica la gestión de dependencias y la compilación del proyecto.

- **GitLab:** GitLab se utilizó como plataforma de gestión de código fuente y colaboración entre los miembros del equipo de desarrollo.
- **IntelliJ IDEA:** Se utilizó IntelliJ IDEA como el entorno de desarrollo integrado (IDE) principal para facilitar la codificación, depuración y pruebas del sistema.
- **Git:** Git se empleó como el sistema de control de versiones para rastrear y administrar los cambios en el código fuente del proyecto.

La elección y aplicación de estas herramientas y tecnologías fueron fundamentales para el desarrollo exitoso del sistema, permitiendo una implementación eficiente y garantizando la calidad y funcionalidad del mismo. Cada una de estas herramientas desempeñó un papel importante en la automatización del proceso de registro de titulación y en la mejora de la eficiencia administrativa en la Universidad de las Fuerzas Armadas ESPE.

### **Metodología de Desarrollo de Software: Kanban**

Para el desarrollo del sistema tipo Workflow vía web para automatizar el proceso de registro de titulación en la Universidad de las Fuerzas Armadas ESPE, se adoptó la metodología ágil Kanban. Kanban es un enfoque de gestión de proyectos que se basa en la visualización del flujo de trabajo y en la optimización del proceso de entrega continua. A continuación, se describen los principios y prácticas clave de Kanban aplicados en el desarrollo de este proyecto:

- **Visualización del Flujo de Trabajo:** Se creó un tablero Kanban para representar visualmente todas las tareas y actividades involucradas en el desarrollo del sistema. El tablero se dividió en columnas que representaban las diferentes etapas del proceso, como "Por hacer", "En progreso", "En revisión" y "Terminado". Esto permitió al equipo tener una visión clara del estado de cada tarea y facilitó la gestión de las actividades en curso.

- **Limitación del Trabajo en Progreso:** Se establecieron límites en el número de tareas que podían estar en progreso al mismo tiempo en cada columna del tablero Kanban. Esto ayudó a evitar la sobrecarga del equipo y a mantener un flujo de trabajo más equilibrado, evitando que se acumulen tareas sin terminar.
- **Enfoque en la Entrega Continua:** Kanban se centró en la entrega continua de incrementos de funcionalidades. Cada vez que se completaba una tarea, se agregaba una nueva tarea desde la columna "Por hacer" al tablero, lo que permitía un desarrollo más ágil y una mayor adaptabilidad a los cambios y requerimientos del proyecto.
- **Colaboración y Comunicación:** Kanban fomentó la colaboración entre los miembros del equipo y la comunicación continua sobre el estado de las tareas y los posibles obstáculos. Esto permitió resolver problemas rápidamente y mantener una dinámica de trabajo eficiente.
- **Feedback y Mejora Continua:** Se realizaron reuniones periódicas para revisar el progreso del proyecto y obtener feedback del equipo y los stakeholders. Esta retroalimentación fue fundamental para identificar áreas de mejora y realizar ajustes en el proceso de desarrollo.

La metodología Kanban proporcionó una estructura flexible y adaptativa para el desarrollo del sistema, lo que permitió al equipo abordar eficientemente las tareas y mantener un flujo de trabajo constante y equilibrado. Gracias a la visualización del proceso y a la entrega continua de incrementos funcionales, se logró una mayor eficiencia en el desarrollo y se respondió de manera efectiva a los cambios y requerimientos del proyecto. La metodología Kanban demostró ser un enfoque efectivo para gestionar el desarrollo del sistema tipo Workflow vía web en la Universidad de las Fuerzas Armadas ESPE y contribuyó al éxito del proyecto.

## Capítulo IV

### Desarrollo

#### Introducción

En este capítulo, se presenta una descripción detallada del proceso de desarrollo del sistema tipo Workflow vía web para automatizar el proceso de registro de titulación en la Universidad de las Fuerzas Armadas ESPE. El objetivo de este capítulo es proporcionar una visión completa de cómo se diseñó, implementó y puso en funcionamiento el sistema, abordando los aspectos más relevantes y clave del proyecto.

A lo largo de este capítulo, se abordarán los siguientes aspectos específicos del sistema:

1. **Arquitectura del Sistema:** Se describe la estructura y organización del sistema, presentando los componentes principales y su interacción. La arquitectura proporciona una visión general de cómo se diseñó el sistema para asegurar su eficiencia y escalabilidad.
2. **Diseño de la Interfaz de Usuario:** Se expondrá el diseño de la interfaz de usuario del sistema, destacando las consideraciones estéticas y de usabilidad que se tuvieron en cuenta para garantizar una experiencia amigable y efectiva para los usuarios.
3. **Implementación de Funcionalidades Clave:** Se detalla la implementación de las funcionalidades clave que permiten automatizar el proceso de registro de titulación. Estas funcionalidades son fundamentales para lograr los objetivos propuestos en el proyecto.
4. **Integración de Herramientas y Tecnologías:** Se explicará cómo se integraron las diversas herramientas y tecnologías utilizadas en el desarrollo del sistema. La

interoperabilidad y la correcta sincronización entre las diferentes herramientas son cruciales para el funcionamiento global del sistema.

5. **Pruebas y Validación del Sistema:** Se presentarán las pruebas realizadas para asegurar la calidad y el correcto funcionamiento del sistema. Se abordarán los escenarios de prueba y los resultados obtenidos.

Con esta introducción, se pretende brindar una visión general de los temas que se abordarán en este capítulo y contextualizar al lector sobre el desarrollo del sistema tipo Workflow vía web en la Universidad de las Fuerzas Armadas ESPE.

## **Arquitectura del Sistema**

La arquitectura diseñada para el sistema de inteligencia de negocios se configura como la solución requerida por la Universidad de las Fuerzas Armadas ESPE para la creación de aplicaciones que impulsen la toma de decisiones informadas en el ámbito académico. Esta arquitectura ha sido concebida específicamente para satisfacer las necesidades y requisitos de la universidad, aprovechando las herramientas ya disponibles y licenciadas por la institución.

Al ser una solución alineada con las expectativas de la Universidad, la arquitectura se estructura en cuatro áreas fundamentales para garantizar una gestión eficiente de los datos académicos:

1. **Fuente de Datos:** Se ha tenido en cuenta la utilización de herramientas ya presentes en la universidad, como Google Spreadsheets, que es utilizada por la Unidad de Gestión de la Investigación para mantener los formatos relacionados con los datos académicos. Esta integración asegura una fuente de datos confiable y coherente para el proceso de análisis.

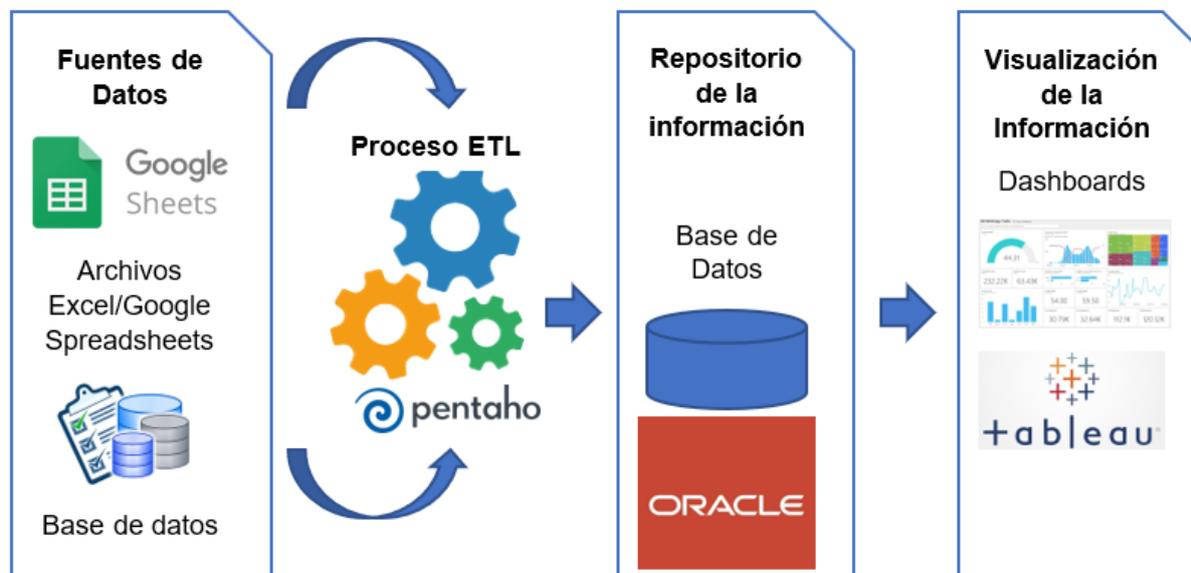
2. **Proceso ETL:** La utilización de Pentaho Data Integration como herramienta para la extracción, transformación y carga de datos es un componente clave de la arquitectura. Su vinculación con el Pentaho Server CE en uso en la universidad asegura una transición eficiente y segura de los datos hacia el siguiente paso del proceso.
3. **Repositorio de Información (Base de Datos):** La elección de Oracle 12c como el sistema de gestión de bases de datos relacional está basada en el licenciamiento existente en la universidad. Esto garantiza que el almacenamiento y consulta de la información académica se realice con una plataforma de probada eficiencia y escalabilidad.
4. **Visualización de Datos:** La integración de Tableau como herramienta de análisis visual es una elección estratégica que permite que los usuarios de la universidad gestionen y presenten los hallazgos en los datos almacenados en el repositorio. Tableau facilita la creación de visualizaciones interactivas y reportes que impulsan una toma de decisiones informada y oportuna.

Con esta arquitectura, la Universidad de las Fuerzas Armadas ESPE se posiciona para crear aplicaciones de inteligencia de negocios que promuevan una gestión más efectiva y una mejor comprensión de los datos académicos. Al optimizar el proceso de toma de decisiones mediante una arquitectura bien definida y ajustada a las herramientas que la universidad ya posee, se asegura un desarrollo ágil y exitoso de aplicaciones que contribuyan al fortalecimiento del sistema educativo en general.

La arquitectura se implementa como se define en la imagen a continuación:

Figura 1

## Implementación de la Arquitectura



Además, como parte del diseño de la arquitectura del sistema de inteligencia de negocios, se incorporaron las siguientes características y enfoques:

- **Microservicios:** Con el objetivo de garantizar la flexibilidad y escalabilidad del sistema, se adoptó una arquitectura basada en microservicios. Cada funcionalidad del sistema se desarrolló como un microservicio independiente, lo que permite realizar cambios y actualizaciones de manera ágil y eficiente. Esta modularidad facilita la gestión y evolución de cada componente de forma individual, sin afectar al resto del sistema.
- **Base de Datos Oracle 12c:** El sistema se apoya en una robusta base de datos Oracle 12c para el almacenamiento de datos académicos. La elección de Oracle 12c se basó en su comprobada robustez, confiabilidad y capacidad para manejar grandes volúmenes de información. Esta base de datos proporciona una sólida infraestructura para gestionar los datos académicos de manera eficiente y segura.

- **APIs (Interfaz de Programación de Aplicaciones):** Se diseñaron APIs para facilitar la comunicación y la integración entre los diferentes componentes del sistema. Estas APIs permiten una interacción efectiva entre los microservicios y otras aplicaciones o sistemas de la universidad, asegurando un flujo de datos transparente y consistente.
- **Seguridad y Acceso Controlado:** La protección de la información confidencial de los estudiantes es una prioridad. Por ello, se implementaron medidas de seguridad robustas en el sistema. Se establecieron mecanismos de autenticación y autorización para garantizar un acceso controlado a las funcionalidades y datos del sistema, asegurando la privacidad y confidencialidad de la información académica.
- **Escalabilidad y Tolerancia a Fallos:** La arquitectura del sistema está diseñada para ser altamente escalable y tolerante a fallos. Los microservicios permiten escalar cada componente de forma independiente según las necesidades del sistema, garantizando un óptimo rendimiento y adaptabilidad ante cambios en la demanda. Asimismo, se implementaron mecanismos de recuperación ante posibles fallos para asegurar la disponibilidad continua del sistema, manteniendo la continuidad en el acceso a la información académica.

Estas características y enfoques fueron cuidadosamente considerados y aplicados en el diseño de la arquitectura del sistema, cumpliendo con los requerimientos y expectativas de la Universidad de las Fuerzas Armadas ESPE. Con una arquitectura sólida y adecuadamente planificada, se sientan las bases para el desarrollo de aplicaciones de inteligencia de negocios que impulsen una gestión más eficiente y una mejor toma de decisiones en el ámbito académico.

## Desarrollo del código

### Backend (Servidor)

El Backend del sistema Workflow vía web constituye el motor detrás de las operaciones y la lógica del sistema. Esta parte es esencial para la gestión eficiente de los flujos de trabajo, la manipulación de datos y la coordinación de tareas en todo el proceso de titulación. Aunque su funcionamiento está principalmente oculto para los usuarios finales, es la columna vertebral que garantiza la funcionalidad fluida y confiable del sistema.

El Backend presenta las siguientes características clave:

**Gestión de Flujos de Trabajo:** El Backend controla los flujos de trabajo automatizados, asignando tareas a las partes apropiadas en función de reglas predefinidas. Coordinando las aprobaciones y notificaciones, garantiza que las solicitudes avancen sin problemas a través del proceso de titulación.

**Base de Datos Centralizada:** Almacena y administra los datos relevantes para el proceso de titulación, incluidos los registros de solicitudes, documentos generados y estados de aprobación. Proporciona acceso seguro y eficiente a la información para su uso en la interfaz de usuario.

**Integración de Sistemas:** El Backend facilita la integración con otras aplicaciones y sistemas utilizados en la universidad, como sistemas de gestión académica y bases de datos de estudiantes. Esto asegura una transferencia fluida de datos y la coherencia de la información.

**Automatización de Tareas:** Implementa la automatización de tareas repetitivas, lo que reduce la carga de trabajo del personal administrativo y minimiza la posibilidad de errores humanos. Esto incluye la generación automatizada de informes, certificados y otros documentos.

**Seguridad y Autenticación:** El Backend garantiza la seguridad de los datos y la autenticación de usuarios, protegiendo la información sensible y controlando el acceso a las funciones y datos relevantes.

**Escalabilidad y Rendimiento:** Está diseñado para manejar cargas de trabajo variables y garantizar un rendimiento constante, incluso cuando se procesan múltiples solicitudes simultáneamente.

En conjunto, el Backend del sistema Workflow vía web trabaja en segundo plano para garantizar la correcta ejecución de los procesos de titulación. Su capacidad para gestionar flujos de trabajo, administrar datos y automatizar tareas es esencial para la eficiencia del sistema en su conjunto. Aunque los usuarios finales pueden no interactuar directamente con el Backend, su contribución es crucial para lograr los objetivos de optimización y agilización de la titulación en la Universidad de las Fuerzas Armadas ESPE.

## MODELOS

Se refiere a una representación de datos o conceptos que refleja una estructura o entidad del mundo real. Los modelos se utilizan para organizar, manipular y procesar datos de manera eficiente y coherente en una aplicación. En este caso definimos los modelos para el uso de toda la aplicación, se lo hace de una manera similar a la base de datos que se va a usar, incluyendo los tipos de datos y nombres de las variables, de igual manera se referencia la tabla usada para el desarrollo.

**Figura 2***Grades Model*

```
1 package ec.edu.espe.stella.model;
2
3 import jakarta.persistence.*;
4 import lombok.Data;
5
6 5 usages Kevin Romeo Gomez Garcia
7 @Data
8 @Entity
9 @Table(name = "GRADES")
10 public class Grades {
11     @Id
12     @Column(name = "ID")
13     private Long id;
14
15     @Column(name = "IDCARD")
16     private String idCard;
17
18     @Column(name = "GPA")
19     private Float gpa;
20
21     @Column(name = "FINALEXAM")
22     private Float finalExam;
23
24     @Column(name = "BASECOMPONENT")
25     private Float baseComponent;
26
27 }
```

Figura 3

*Student Model*

```
1 package ec.edu.espe.stella.model;
2
3 import jakarta.persistence.*;
4 import lombok.Data;
5
6 /* usages: ± krgomez +1 */
7 @Data
8 @Entity
9 @Table(name = "STUDENT")
10 public class Student {
11     @Id
12     @Column(name = "ID")
13     private Long id;
14
15     @Column(name = "IDCARD")
16     private String idCard;
17
18     @Column(name = "NAME")
19     private String name;
20
21     @Column(name = "LASTNAME")
22     private String lastName;
23
24     @Column(name = "TYPE")
25     private String type;
26
27     @Column(name = "STATUS")
28     private String status;
29
30     @Column(name = "STEP")
31     private Integer step;
32 }
```

**Figura 4***Record Model Parte 1*

```
1 package ec.edu.espe.stella.model;
2
3 > import ...
4
5
6 > @{...}
7
8
9 public class Record {
10
11     @Id
12     @GeneratedValue(strategy = GenerationType.IDENTITY)
13     @Column(name = "ID")
14     private Long id;
15
16     @Column(name = "IDCARD")
17     private String idCard;
18
19     @Column(name = "PERSONALINFO")
20     private String personalInfo;
21
22     @Column(name = "DOCALUMNI")
23     private String docAlumni;
24
25     @Column(name = "DNI")
26     private String dni;
27
28     @Column(name = "CERTDEGREE")
29     private String certDegree;
30
31     @Column(name = "CERTPERIOD")
32     private String certPeriod;
33
34     @Column(name = "DOCREC")
35     private String docRec;
36
```

Figura 5

## Record Model Parte 2

```
7      @Column(name = "DOCGRADES")
8      private String docGrades;
9
10     @Column(name = "DOCADEQUACY")
11     private String docAdequacy;
12
13     @Column(name = "DOCLIBRARY")
14     private String docLibrary;
15
16     @Column(name = "DOCREs")
17     private String docRes;
18
19     @Column(name = "DOCCURRICULUM")
20     private String docCurriculum;
21
22     @Column(name = "DOCSYSCURRICULUM")
23     private String docSysCurriculum;
24
25     @Column(name = "DOCPAS")
26     private String docPas;
27
28     @Column(name = "DOCVIN")
29     private String docVin;
30
31     @Column(name = "CERCURRICULUM")
32     private String cerCurriculum;
33
34     @Column(name = "PHOTO")
35     private String photo;
```

**Figura 6***Record Model Parte 3*

```
@Column(name = "DOCNODUE")  
private String docNoDue;  
  
@Column(name = "REPIMP")  
private String repImp;  
  
@Column(name = "PAYMENT")  
private String payment;  
  
@Column(name = "BACHELORDEGREE")  
private String bachelorDegree;
```

## DTO

Es un patrón de diseño utilizado para encapsular un conjunto de datos y transmitirlos entre diferentes componentes de un sistema, como capas de una aplicación o entre sistemas distribuidos.

### Figura 7

#### *Grades DTO*

```
1 package ec.edu.espe.stella.dto;
2
3 import lombok.Data;
4
5 no usages Kevin Romeo Gomez Garcia
6 @Data
7 public class GradesDto {
8     private Long id;
9     private String idCard;
10    private Double gpa;
11    private Double finalExam;
12    private Double baseComponent;
13
14 }
15
```

**Figura 8***Student DTO*

```
no usages  👤 krgomez +1  
@Data  
public class StudentDto {  
    private Long id;  
    private String idCard;  
    private String name;  
    private String lastName;  
    private String type;  
    private String status;  
    private Integer step;  
}
```

**Figura 9***Record DTO*

```
1 package ec.edu.espe.stella.dto;
2
3 import lombok.Data;
4
5 no usages ↗ Kevin Romeo Gomez Garcia *
6 @Data
7 public class RecordDto {
8     private Long id;
9     private String idCard;
10    private String personalInfo;
11    private String docAlumni;
12    private String dni;
13    private String certDegree;
14    private String certPeriod;
15    private String docRec;
16    private String docGrades;
17    private String docAdequacy;
18    private String docLibrary;
19    private String docRes;
20    private String docCurriculum;
21    private String docSysCurriculum;
22    private String docPas;
23    private String docVin;
24    private String cerCurriculum;
25    private String photo;
26    private String docNoDue;
27    private String repImp;
28    private String payment;
29    private String bachelorDegree;
30
31 }
```

## DAO

Es un patrón de diseño que se utiliza para separar la lógica de acceso a datos de la lógica de negocio en una aplicación. En este apartado se ha creado consultas personalizadas a través del repositorio, de manera que haga una consulta específica a la base de datos.

### Figura 10

#### *Grades DAO*

```
1 package ec.edu.espe.stella.dao;
2
3 import ec.edu.espe.stella.model.Grades;
4 import org.springframework.data.jpa.repository.JpaRepository;
5 import org.springframework.stereotype.Repository;
6
7 import java.util.Optional;
8
9 3 usages Kevin Romeo Gomez Garcia
10 @Repository
11 public interface GradesDao extends JpaRepository<Grades, Long> {
12     1 usage Kevin Romeo Gomez Garcia
13     Optional<Grades> findByIdCard(String idCard);
14 }
```

Figura 11

*Student DAO*

```
package ec.edu.espe.stella.dao;

import java.util.List;
import java.util.Optional;

import ec.edu.espe.stella.model.Student;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

3 usages  👤 krgomez +1 *
@Repository
public interface StudentDao extends JpaRepository<Student, Long> {
    5 usages  👤 krgomez
    Optional<Student> findByIdCard(String idCard);
    1 usage  👤 Kevin Romeo Gomez Garcia
    List<Student> findAllByStatus(String status);
}
```

Figura 12

## Record DAO Parte 1

```

package ec.edu.espe.stella.dao;

import ec.edu.espe.stella.model.Record;
import jakarta.transaction.Transactional;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Modifying;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

import java.util.Optional;

6 usages  ↕ Kevin Romeo Gomez Garcia *
@Repository
public interface RecordDao extends JpaRepository<Record, Long> {
    5 usages  ↕ Kevin Romeo Gomez Garcia
    Optional<Record> findByIdCard(String idCard);
    1 usage  ↕ Kevin Romeo Gomez Garcia
    ⚠ @Transactional
    @Modifying
    @Query("UPDATE Record r SET " +
        "r.docPas = CASE WHEN ?1 = 'docPasantias' THEN ?2 ELSE r.docPas END, " +
        "r.docVin = CASE WHEN ?1 = 'docVinculacion' THEN ?2 ELSE r.docVin END, " +
        "r.docAdequacy = CASE WHEN ?1 = 'docSuficiencia' THEN ?2 ELSE r.docAdequacy END, " +
        "r.docNoDue = CASE WHEN ?1 = 'docNoAdeudar' THEN ?2 ELSE r.docNoDue END, " +
        "r.docRes = CASE WHEN ?1 = 'docResolucion' THEN ?2 ELSE r.docRes END, " +
        "r.docRec = CASE WHEN ?1 = 'docRecord' THEN ?2 ELSE r.docRec END " +
        "WHERE r.idCard = ?3")
    void updateDocumentPath(String docType, String docPath, String idCard);

```

Figura 13

## Record DAO Parte 2

```

@Transactional
@Modifying
@Query("UPDATE Record r SET " +
    "r.certDegree = CASE WHEN ?1 = 'certDegree' THEN ?2 ELSE r.certDegree END, " +
    "r.certPeriod = CASE WHEN ?1 = 'certPeriod' THEN ?2 ELSE r.certPeriod END, " +
    "r.cerCurriculum = CASE WHEN ?1 = 'cerCurriculum' THEN ?2 ELSE r.cerCurriculum END " +
    "WHERE r.idCard = ?3")
void updateCertificatePath(String certType, String certPath, String idCard);

no usages new *
@Transactional
@Modifying
@Query("UPDATE Record r SET " +
    "r.personalInfo = CASE WHEN ?1 = 'personalInfo' THEN ?2 ELSE r.personalInfo END, " +
    "r.dni = CASE WHEN ?1 = 'dni' THEN ?2 ELSE r.dni END, " +
    "r.photo = CASE WHEN ?1 = 'photo' THEN ?2 ELSE r.photo END, " +
    "r.repImp = CASE WHEN ?1 = 'repImp' THEN ?2 ELSE r.repImp END, " +
    "r.payment = CASE WHEN ?1 = 'payment' THEN ?2 ELSE r.payment END, " +
    "r.bachelorDegree = CASE WHEN ?1 = 'bachelorDegree' THEN ?2 ELSE r.bachelorDegree END " +
    "WHERE r.idCard = ?3")
void updateOtherFields(String field, String value, String idCard);

no usages new *
@Transactional
@Modifying
@Query("UPDATE Record r SET " +
    "r.docAlumni = CASE WHEN ?1 = 'docAlumni' THEN ?2 ELSE r.docAlumni END, " +
    "r.docGrades = CASE WHEN ?1 = 'docGrades' THEN ?2 ELSE r.docGrades END, " +
    "r.docLibrary = CASE WHEN ?1 = 'docLibrary' THEN ?2 ELSE r.docLibrary END, " +
    "r.docCurriculum = CASE WHEN ?1 = 'docCurriculum' THEN ?2 ELSE r.docCurriculum END, " +
    "r.docSysCurriculum = CASE WHEN ?1 = 'docSysCurriculum' THEN ?2 ELSE r.docSysCurriculum END " +
    "WHERE r.idCard = ?3")
void updateAdditionalDocuments(String docType, String docPath, String idCard);

```

## SERVICIOS

Es un componente o módulo que encapsula una funcionalidad específica y se utiliza para realizar tareas o acciones dentro de una aplicación. Ayuda a modularizar el código, promover la reutilización y separar responsabilidades en el diseño de software. Aquí definimos la lógica del componente, está dividido en diferentes capas de manera que se forma una arquitectura independiente por cada servicio.

### Figura 14

#### *Grades Service*

```
@Service
public class GradesService {

    2 usages
    private final GradesDao gradesDao;

    1 Kevin Romeo Gomez Garcia
    @Autowired
    public GradesService(GradesDao gradesDao) { this.gradesDao = gradesDao; }

    1 usage 1 Kevin Romeo Gomez Garcia
    public Double calculateFinalGrade(String idCard) {

        Grades grades = gradesDao.findByIdCard(idCard).orElse( other: null);

        if (grades == null) {
            throw new IllegalArgumentException("No se encontró el registro para el ID de tarjeta proporcionado.");
        }

        Double weightedGPA = grades.getGpa() * 0.875;
        Double weightedFinalExam = grades.getFinalExam() * 0.125;
        Double weightedBaseComponent = grades.getBaseComponent() * 1.0;

        Double finalGrade = (weightedGPA + weightedFinalExam) * 0.8 + weightedBaseComponent * 0.2;

        return finalGrade;
    }
}
```

Figura 15

## StudentService Parte 1

```
@Service
public class StudentService {

    10 usages
    private final StudentDao studentDao;
    3 usages
    private final RecordDao recordDao;

    ± Kevin Romeo Gomez Garcia +1
    @Autowired
    public StudentService(StudentDao studentDao, RecordDao recordDao) {
        this.studentDao = studentDao;
        this.recordDao = recordDao;
    }

    1 usage ± Kevin Romeo Gomez Garcia
    public Optional<Integer> getStudentStepByIdCard(String idCard) {
        Optional<Student> studentOptional = studentDao.findByIdCard(idCard);
        if (!recordDao.findByIdCard(idCard).isPresent()) {
            Record newRecord = new Record();
            newRecord.setIdCard(idCard);
            recordDao.save(newRecord);
        }
        return studentOptional.map(Student::getStep);
    }

    1 usage ± Kevin Romeo Gomez Garcia
    public Optional<String> getStudentStatusByIdCard(String idCard) {
        Optional<Student> studentOptional = studentDao.findByIdCard(idCard);
        return studentOptional.map(Student::getStatus);
    }
}
```

Figura 16

*StudentService Parte 2*

```
1 usage  ▲ Kevin Romeo Gomez Garcia
| public List<Student> getAllStudentsWithStatusP() { return studentDao.findAllByStatus("P"); }

1 usage  ▲ Kevin Romeo Gomez Garcia
public void updateStudentStatusAndType(String idCard, String newStatus, String newType) {
    Optional<Student> studentOptional = studentDao.findByIdCard(idCard);
    if (studentOptional.isPresent()) {
        Student student = studentOptional.get();
        student.setStatus(newStatus);
        student.setType(newType);
        studentDao.save(student);
    } else {
        throw new IllegalArgumentException("No se encontró el estudiante con el ID proporcionado");
    }
}

1 usage  ▲ Kevin Romeo Gomez Garcia
public void changeStudentStatusToP(String idCard) {
    Optional<Student> studentOptional = studentDao.findByIdCard(idCard);
    if (studentOptional.isPresent()) {
        Student student = studentOptional.get();
        if ("N".equals(student.getStatus())) {
            student.setStatus("P");
            studentDao.save(student);
        } else {
            throw new IllegalStateException("El estado del estudiante no es 'N'");
        }
    } else {
        throw new IllegalArgumentException("No se encontró el estudiante con el ID proporcionado");
    }
}
```

Figura 17

*StudentService Parte 3*

```

1 usage  ↕ Kevin Romeo Gomez Garcia
public void changeStudentStep(String idCard, int newStep) {
    Optional<Student> studentOptional = studentDao.findByIdCard(idCard);
    if (studentOptional.isPresent()) {
        Student student = studentOptional.get();
        student.setStep(newStep);
        studentDao.save(student);
    } else {
        throw new IllegalArgumentException("No se encontró el estudiante con el ID proporcionado");
    }
}

```

Figura 18

*RecordService Parte 1*

```

package ec.edu.espe.stella.service;

> import ...

9 usages  ↕ Kevin Romeo Gomez Garcia *
@Service
public class RecordService {

    7 usages
    private final RecordDao recordDao;

    @Value("../records")
    private String recordFolderPath;

    ↕ Kevin Romeo Gomez Garcia
    @Autowired
    > public RecordService(RecordDao recordDao) { this.recordDao = recordDao; }

    2 usages  ↕ Kevin Romeo Gomez Garcia
    > public List<String> getNullDocuments(String idCard) {...}

    1 usage  ↕ Kevin Romeo Gomez Garcia
    > public boolean hasAllRequiredDocuments(String idCard) {...}

    2 usages  ↕ Kevin Romeo Gomez Garcia
    > private String getExistingDocument(Record record, String docPropertyName) {...}

    2 usages  ↕ Kevin Romeo Gomez Garcia
    > private String getDocPropertyName(String documentType) {...}

```

**Figura 19***RecordService Parte 2*

```
1 usage  🡕 Kevin Romeo Gomez Garcia
public byte[] downloadFile(String idCard, String documentType) throws IOException {...}

1 usage  🡕 Kevin Romeo Gomez Garcia *
public void uploadDocument(String idCard, String documentType, MultipartFile file) throws IOException {...}

1 usage  🡕 Kevin Romeo Gomez Garcia
private void setDocumentProperty(Record record, String docPropertyName, String filename) {...}

1 usage  🡕 Kevin Romeo Gomez Garcia
public void saveFileToFolder(MultipartFile file, Path folderPath, String fileName) throws IOException {...}

1 usage  🡕 Kevin Romeo Gomez Garcia
public void createStudentFolder(String idCard) {...}

1 usage  🡕 Kevin Romeo Gomez Garcia
private String getFileExtension(String fileName) {...}

1 usage  🡕 Kevin Romeo Gomez Garcia
public List<String> getListOfDocuments(String idCard) {...}
```

**Figura 20***RecordService Parte 3*

```
public void uploadDocument(String idCard, String documentType, MultipartFile file) throws IOException {
    String folderName = recordFolderPath + "/ID_" + idCard;

    createStudentFolder(idCard);

    Optional<Record> recordOptional = recordDao.findByIdCard(idCard);
    if (recordOptional.isPresent()) {
        Record record = recordOptional.get();

        String docPropertyName = getDocPropertyName(documentType);

        if (docPropertyName != null) {
            String existingDocument = getExistingDocument(record, docPropertyName);

            if (existingDocument != null) {...}

            String fileExtension = getFileExtension(file.getOriginalFilename());
            String fileName = idCard + "_" + documentType + "." + fileExtension;
            String documentPath = folderName + "/" + fileName;

            setDocumentProperty(record, docPropertyName, documentType);
            recordDao.save(record);

            recordDao.updateDocumentPath(documentType, documentPath, idCard);
            saveFileToFolder(file, Path.of(folderName), fileName);
        } else {
            throw new IllegalArgumentException("Tipo de documento no válido.");
        }
    }
}
```

Figura 21

*RecordGeneratorService*

```
3 usages Kevin Romeo Gomez Garcia *
16 @Service
17 public class RecordGeneratorService {
18
19     6 usages
20     private final RecordDao recordDao;
21
22     @Value("./records")
23     private String recordFolderPath;
24
25     Kevin Romeo Gomez Garcia
26     @Autowired
27     public RecordGeneratorService(RecordDao recordDao) { this.recordDao = recordDao; }
28
29     1 usage Kevin Romeo Gomez Garcia
30     public void uploadDocument(String idCard, String documentType, MultipartFile file) throws IOException {...}
31
32     1 usage Kevin Romeo Gomez Garcia
33     private String getDocPropertyName(String documentType) {...}
34
35     1 usage Kevin Romeo Gomez Garcia
36     private String getExistingDocument(Record record, String docPropertyName) {...}
37
38     1 usage Kevin Romeo Gomez Garcia *
39     private String getFileExtension(String fileName) {...}
40
41     1 usage Kevin Romeo Gomez Garcia
42     public void saveFileToFolder(MultipartFile file, Path folderPath, String fileName) throws IOException {...}
43
44     1 usage Kevin Romeo Gomez Garcia *
45     public List<String> getListOfRecordDocuments(String idCard) {...}
46
```

## CONTROLADORES

Esta es la capa final del backend, aquí definimos las rutas de las funcionalidades que se van a consumir en cada microservicio, al no ser una arquitectura monolítica, se pueden consumir los diferentes componentes de manera individual y mostrar diferentes resultados a través de API's.

### Figura 22

#### *gradesLoader Controller*

```
package ec.edu.espe.stella.controller;

import ...

@KevinRomeoGomezGarcia +1
@RestController
public class gradesLoader {

    2 usages
    private final GradesService gradesService;

    @KevinRomeoGomezGarcia
    @Autowired
    > public gradesLoader(GradesService gradesService) { this.gradesService = gradesService; }

    @KevinRomeoGomezGarcia
    @GetMapping("/calculateFinalGrade")
    > public Double calculateFinalGrade(@RequestParam String idCard) { return gradesService.calculateFinalGrade(idCard); }
}
```

Figura 23

*studentRequest Controller Parte 1*

```
@RestController
public class studentRequest {

    5 usages
    private final StudentService studentService;
    3 usages
    private final RecordService recordService;

    ⚡ Kevin Romeo Gomez Garcia +1
    @Autowired
    public studentRequest(StudentService studentService, RecordService recordService) {
        this.studentService = studentService;
        this.recordService = recordService;
    }

    ⚡ Kevin Romeo Gomez Garcia +1
    @PostMapping("/verifyDocuments")
    public ResponseEntity<List<String>> validateId(@RequestParam String idCard) {
        List<String> nullDocuments = recordService.getNullDocuments(idCard);

        if (nullDocuments.isEmpty() && recordService.hasAllRequiredDocuments(idCard)) {
            return ResponseEntity.ok( body: null);
        } else {
            return ResponseEntity.ok(nullDocuments);
        }
    }
}
```

Figura 24

*studentRequest Controller Parte 2*

```
± Kevin Romeo Gomez Garcia
@GetMapping("/{getStudentStep}")
public ResponseEntity<Integer> getStudentStep(@RequestParam String idCard) {
    Optional<Integer> stepOptional = studentService.getStudentStepByIdCard(idCard);

    if (stepOptional.isPresent()) {
        return ResponseEntity.ok(stepOptional.get());
    } else {
        return ResponseEntity.notFound().build();
    }
}

± Kevin Romeo Gomez Garcia
@GetMapping("/{getStudentStatus}")
public ResponseEntity<Map<String, String>> getStudentStatus(@RequestParam String idCard) {
    Optional<String> statusOptional = studentService.getStudentStatusByIdCard(idCard);

    if (statusOptional.isPresent()) {
        Map<String, String> response = new HashMap<>();
        response.put("status", statusOptional.get());
        return ResponseEntity.ok(response);
    } else {
        return ResponseEntity.notFound().build();
    }
}
```

Figura 25

*studentRequest Controller Parte 3*

```
Kevin Romeo Gomez Garcia
@PostMapping("/changeStatusPending")
public ResponseEntity<Map<String, String>> changeStudentStatus(@RequestParam String idCard) {
    try {
        studentService.changeStudentStatusToP(idCard);
        Map<String, String> response = new HashMap<>();
        response.put("message", "OK");
        return ResponseEntity.ok(response);
    } catch (IllegalArgumentException e) {
        return ResponseEntity.notFound().build();
    } catch (IllegalStateException e) {
        return ResponseEntity.badRequest().build();
    }
}

Kevin Romeo Gomez Garcia
@PostMapping("/changeStudentStep")
public ResponseEntity<Void> changeStudentStep(@RequestParam String idCard, @RequestParam int newStep) {
    try {
        studentService.changeStudentStep(idCard, newStep);
        return ResponseEntity.ok().build(); // Estado 200 OK
    } catch (IllegalArgumentException e) {
        return ResponseEntity.notFound().build(); // Estado 404 Not Found
    }
}
```

Figura 26

*directorResponse Controller Parte 1*

```
┆ Kevin Romeo Gomez Garcia +1 *  
@RestController  
public class directorResponse {  
  
    3 usages  
    private final StudentService studentService;  
  
    3 usages  
    private final RecordService recordService;  
  
    ┆ Kevin Romeo Gomez Garcia  
    @Autowired  
    public directorResponse(StudentService studentService, RecordService recordService) {  
        this.studentService = studentService;  
        this.recordService = recordService;  
    }  
  
    ┆ Kevin Romeo Gomez Garcia  
    @GetMapping("/getAllStudents")  
    public ResponseEntity<List<Student>> getAllStudents() {  
        List<Student> allStudents = studentService.getAllStudentsWithStatusP();  
  
        if (!allStudents.isEmpty()) {  
            return ResponseEntity.ok(allStudents);  
        } else {  
            return ResponseEntity.noContent().build();  
        }  
    }  
}
```

**Figura 27***directorResponse Controller Parte 2*

```
Kevin Romeo Gomez Garcia
@GetMapping("/download")
public ResponseEntity<ByteArrayResource> downloadFile(
    @RequestParam String idCard,
    @RequestParam String documentType) throws IOException {

    byte[] fileBytes = recordService.downloadFile(idCard, documentType);

    if (fileBytes != null) {
        HttpHeaders headers = new HttpHeaders();
        headers.setContentType(MediaType.APPLICATION_OCTET_STREAM);
        headers.setContentDispositionFormData( name: "attachment", filename: idCard + "_" + documentType + ".txt");

        ByteArrayResource resource = new ByteArrayResource(fileBytes);

        return ResponseEntity.ok()
            .headers(headers)
            .contentType(MediaType.APPLICATION_OCTET_STREAM)
            .body(resource);
    } else {
        return ResponseEntity.notFound().build();
    }
}
```

Figura 28

*directorResponse Controller Parte 3*

```
± Kevin Romeo Gomez Garcia +1
@PostMapping("/updateTypeAndStatus")
public ResponseEntity<String> updateStudentStatusAndType(
    @RequestParam String idCard,
    @RequestParam String newStatus,
    @RequestParam String newType) {
    try {
        studentService.updateStudentStatusAndType(idCard, newStatus, newType);
        return ResponseEntity.ok(body: "Estado y tipo de estudiante actualizados correctamente.");
    } catch (IllegalArgumentException e) {
        return ResponseEntity.badRequest().body(e.getMessage());
    }
}

± Kevin Romeo Gomez Garcia
@GetMapping("/documents")
public ResponseEntity<List<String>> getListOfDocuments(
    @RequestParam String idCard) {
    List<String> documents = recordService.getListOfDocuments(idCard);
    if (!documents.isEmpty()) {
        return new ResponseEntity<>(documents, HttpStatus.OK);
    } else {
        return ResponseEntity.notFound().build();
    }
}
```

## Frontend (Cliente)

El Front-end del sistema Workflow vía web se refiere a la interfaz de usuario que interactúa directamente con los estudiantes, profesores y personal administrativo. Esta parte del sistema se ha diseñado con un enfoque en la usabilidad, la accesibilidad y la experiencia del usuario. La interfaz ofrece una plataforma intuitiva y fácil de usar que permite a los usuarios realizar acciones clave relacionadas con el proceso de titulación.

La interfaz de Front-end ofrece las siguientes características destacadas:

**Diseño Responsivo:** La interfaz se adapta de manera fluida a diferentes dispositivos, como computadoras de escritorio, tabletas y dispositivos móviles, asegurando una experiencia coherente y accesible para todos los usuarios.

**Navegación Sencilla:** Se ha priorizado una estructura de navegación clara y lógica, lo que permite a los usuarios moverse fácilmente entre las diferentes secciones del proceso de titulación, presentación de documentos y seguimiento de requisitos.

**Formularios Intuitivos:** Los formularios se han diseñado de manera intuitiva, guiando a los usuarios a través del proceso de ingreso de información necesaria de manera clara y eficiente. Se han incorporado validaciones en tiempo real para minimizar errores.

En conjunto, el Front-end del sistema Workflow vía web despliega una interfaz de usuario moderna y amigable que mejora la experiencia de los usuarios involucrados en el proceso de titulación. La atención cuidadosa al diseño, la usabilidad y la accesibilidad contribuyen a una interacción eficiente y efectiva, lo que a su vez respalda los objetivos de agilizar y mejorar la titulación en la Universidad de las Fuerzas Armadas ESPE.

## Modulo Principal

Figura 29

*main.module*

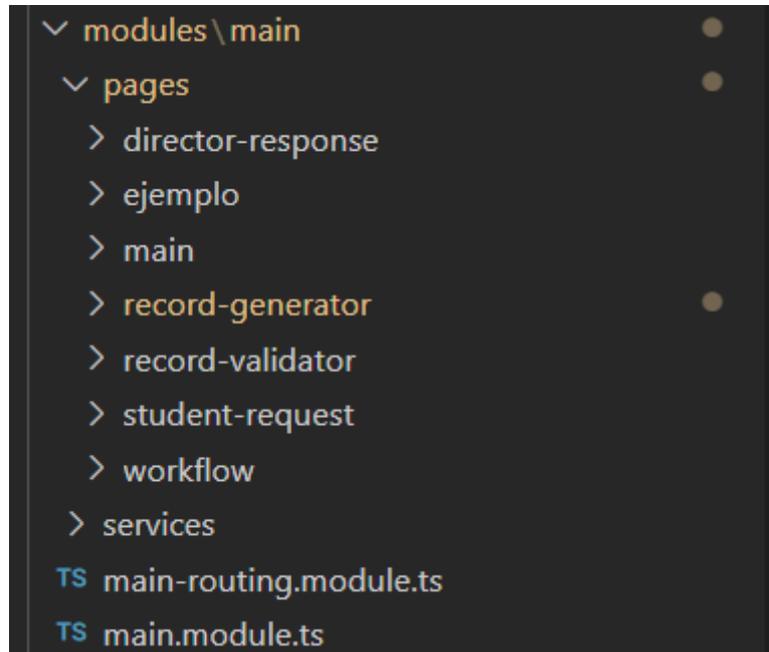
```
prototipo-ng-master > src > app > modules > main > TS main.module.ts > ...
 1  import { NgModule } from '@angular/core';
 2  import { CommonModule } from '@angular/common';
 3
 4  import { MainRoutingModule } from './main-routing.module';
 5  import { MainComponent } from './pages/main/main.component';
 6  import { EjemploComponent } from './pages/ejemplo/ejemplo.component';
 7  import { WorkflowComponent } from './pages/workflow/workflow.component';
 8  import { StudentRequestComponent } from './pages/student-request/student-request.component';
 9  import { DirectorResponseComponent } from './pages/director-response/director-response.component';
10  import { RecordGeneratorComponent } from './pages/record-generator/record-generator.component';
11  import { RecordValidatorComponent } from './pages/record-validator/record-validator.component';
12
13
14  @NgModule({
15    declarations: [
16      MainComponent,
17      EjemploComponent,
18      WorkflowComponent,
19      StudentRequestComponent,
20      DirectorResponseComponent,
21      RecordGeneratorComponent,
22      RecordValidatorComponent
23    ],
24    imports: [
25      CommonModule,
26      MainRoutingModule
27    ]
28  })
29  export class MainModule { }
```

Figura 30

*Main-routing-module*

```
prototipo-ng-master > src > app > modules > main > TS main-routing.module.ts > ...
 1  import { NgModule } from '@angular/core';
 2  import { RouterModule, Routes } from '@angular/router';
 3  import { MainComponent } from './pages/main/main.component';
 4  import { EjemploComponent } from './pages/ejemplo/ejemplo.component';
 5  import { StudentRequestComponent } from './pages/student-request/student-request.component';
 6  import { DirectorResponseComponent } from './pages/director-response/director-response.component';
 7  import { RecordGeneratorComponent } from './pages/record-generator/record-generator.component';
 8  import { RecordValidatorComponent } from './pages/record-validator/record-validator.component';
 9  import { WorkflowComponent } from './pages/workflow/workflow.component';
10
11  const routes: Routes = [
12    { path: 'principal', component: MainComponent },
13    { path: 'ejemplo', component: EjemploComponent },
14    { path: 'workflow', component: WorkflowComponent },
15    { path: 'studentRequest', component: StudentRequestComponent },
16    { path: 'directorResponse', component: DirectorResponseComponent },
17    { path: 'recordGenerator', component: RecordGeneratorComponent },
18    { path: 'recordValidator', component: RecordValidatorComponent },
19  ];
20
21  @NgModule({
22    imports: [RouterModule.forChild(routes)],
23    exports: [RouterModule]
24  })
25  export class MainRoutingModule { }
26
```

Figura 31

*Estructura*

```

  ✓ modules \ main
  ✓ pages
    > director-response
    > ejemplo
    > main
    > record-generator
    > record-validator
    > student-request
    > workflow
  > services
  TS main-routing.module.ts
  TS main.module.ts

```

## COMPONENTES

Figura 32

*student-request*

```
> import { Component, OnInit, ViewChild, ElementRef } from '@angular/core'; ...
> @Component({ ...
})
export class StudentRequestComponent implements OnInit {
  idCard: string;
  studentStatus: string;
  missingDocuments: string[] = [];

  @ViewChild('fileInputRef') fileInputRef: ElementRef;

  constructor(
    private route: ActivatedRoute,
    private router: Router,
    private apiService: ApiService
  ) {}

  ngOnInit() {
    this.route.queryParams.subscribe((queryParams) => {
      this.idCard = queryParams.id;
      this.getStudentStatus();
      this.verifyDocuments();
    });
  }

  getStudentStatus() {
    if (this.idCard) {
      this.apiService.getStudentStatus(this.idCard).subscribe(
        status => {
          this.studentStatus = status;
        },
        error => {
          console.error('Error al obtener el estado del estudiante:', error);
        }
      );
    }
  }
}
```

Figura 33

*director-response*

```
1 > import { Component, OnInit } from '@angular/core'; ...
4
5 @component({
6   selector: 'app-director-response',
7   templateUrl: './director-response.component.html',
8   styleUrls: ['./director-response.component.scss']
9 })
10 export class DirectorResponseComponent implements OnInit {
11   idCard: string;
12   documents: string[] = [];
13
14   constructor(
15     private route: ActivatedRoute,
16     private apiService: ApiService
17   ) {}
18
19   ngOnInit() {
20     this.route.queryParams.subscribe(params => {
21       this.idCard = params['id'];
22       this.loadListOfDocuments();
23     });
24   }
25
26   loadListOfDocuments() {
27     this.apiService.getListOfDocuments(this.idCard).subscribe(
28       (documents) => {
29         this.documents = documents;
30       },
31       (error) => {
32         console.error('Error al obtener la lista de documentos:', error);
33       }
34     );
35   }
}
```

Figura 34

*record-generator*

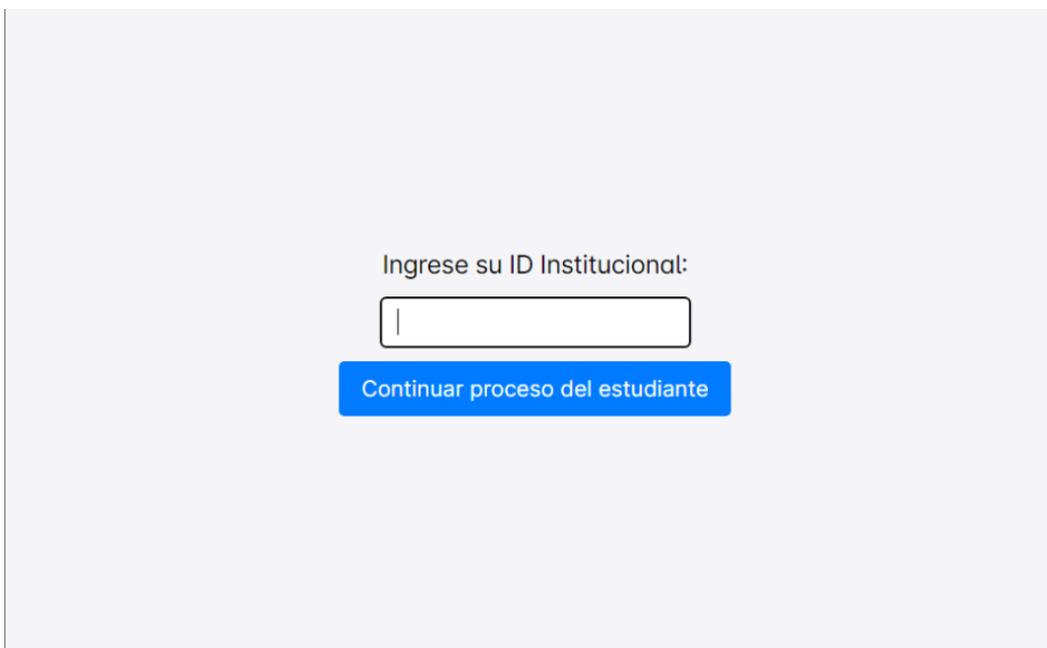
```
10 export class RecordGeneratorComponent implements OnInit {
11     idCard: string;
12     missingDocuments: string[] = [];
13
14     @ViewChild('fileInputRef') fileInputRef: ElementRef;
15
16     constructor(
17         private route: ActivatedRoute,
18         private router: Router,
19         private apiService: ApiService
20     ) {}
21
22     ngOnInit() {
23         this.route.queryParams.subscribe((queryParams) => {
24             this.idCard = queryParams.id;
25             this.verifyDocuments();
26         });
27     }
28
29     verifyDocuments() {
30         if (this.idCard) {
31             this.apiService.verifyDocuments(this.idCard).subscribe(
32                 (response: string[]) => {
33                     this.missingDocuments = response;
34                 },
35                 (error) => {
36                     console.error('Error al verificar documentos:', error);
37                 }
38             );
39         }
40     }
41 }
```

Figura 35

*workflow*

```
14 constructor(  
15     private route: ActivatedRoute,  
16     private router: Router,  
17     private apiService: ApiService  
18 ) {}  
19  
20 ngOnInit() {}  
21  
22 onIdInput(event: Event) {  
23     this.idCard = (event.target as HTMLInputElement).value;  
24 }  
25  
26 getStudentStep() {  
27     if (this.idCard) {  
28         this.apiService.getStudentStep(this.idCard).subscribe(  
29             (step) => {  
30                 this.apiResponse = step;  
31                 if (this.apiResponse >= 1 && this.apiResponse <= 4) {  
32                     const routeNames = ['studentRequest', 'directorResponse', 'recordGenerator', 'recordValidator'];  
33                     const routeName = routeNames[this.apiResponse - 1];  
34                     this.router.navigate(['main', routeName], { queryParams: { id: this.idCard, step: this.apiResponse } });  
35                 } else {  
36                     console.log('Valor de step inválido:', this.apiResponse);  
37                 }  
38             },  
39             (error) => {  
40                 console.error('Error al obtener el paso del estudiante:', error);  
41                 this.apiResponse = -1;  
42             }  
43         );  
44     } else {  
45         this.apiResponse = -1;  
46     }  
47 }  
48
```

Figura 36

*main view*

Ingrese su ID Institucional:

Continuar proceso del estudiante

## Capítulo V

### **Análisis de Resultados**

#### **Introducción**

En este capítulo, se presentarán y analizarán los resultados obtenidos durante la implementación y prueba del sistema Workflow vía web para la regulación y agilización de los procesos de titulación en la Universidad de las Fuerzas Armadas ESPE. Se examinarán los efectos del sistema en términos de eficiencia administrativa, guía para los estudiantes y mejora del proceso académico.

#### **Análisis Comparativo: Reducción del Tiempo de Procesamiento**

En este análisis comparativo, se examinó el impacto de la implementación del sistema Workflow vía web en el tiempo de procesamiento del proceso de titulación en comparación con el proceso manual anterior. El objetivo principal era determinar si la automatización y agilización de los flujos de trabajo contribuyeron a una disminución significativa en el tiempo requerido para completar cada etapa del proceso.

#### **Proceso Manual**

En el proceso manual previo a la implementación del sistema Workflow vía web, el proceso de titulación involucraba una serie de pasos administrativos que requerían una interacción constante entre estudiantes, profesores y personal administrativo. La documentación necesaria se presentaba físicamente en oficinas, y la comunicación se realizaba principalmente mediante correo electrónico y reuniones presenciales. Esto a menudo resultaba en demoras debido a la necesidad de esperar respuestas, programar reuniones y coordinar acciones manuales.

### **Proceso con el Sistema Workflow vía Web**

Tras la implementación del sistema Workflow vía web, se observó una transformación significativa en la forma en que se llevaba a cabo el proceso de titulación. El sistema permitió a los estudiantes y al personal administrativo interactuar en línea a través de una plataforma centralizada. Los flujos de trabajo se automatizaron, lo que eliminó la necesidad de múltiples interacciones manuales y permitió un seguimiento más fluido del progreso de cada solicitud de titulación.

### **Resultados y Beneficios**

Los resultados de este análisis comparativo revelaron una reducción sustancial en el tiempo de procesamiento de las solicitudes de titulación con la implementación del sistema Workflow vía web. Los principales hallazgos incluyen:

**Reducción de Tiempo en Aprobaciones:** Las aprobaciones necesarias para avanzar en el proceso se lograron en un tiempo considerablemente menor. Los flujos de trabajo automatizados notificaron a los responsables de las aprobaciones de manera instantánea, lo que resultó en una respuesta más rápida y en la reducción del tiempo de espera.

**Eliminación de Repeticiones:** La necesidad de repetir pasos y proporcionar información redundante se redujo drásticamente. La información requerida se ingresaba en el sistema una vez y se compartía automáticamente entre las diferentes etapas del proceso.

**Mayor Transparencia:** Tanto los estudiantes como el personal administrativo tenían acceso en tiempo real al estado de las solicitudes. Esto redujo la necesidad de consultas repetitivas y proporcionó claridad en cuanto al progreso.

## **Conclusión**

El análisis comparativo demuestra de manera concluyente que la implementación del sistema Workflow vía web ha tenido un impacto significativo en la reducción del tiempo de procesamiento en el proceso de titulación. La automatización de flujos de trabajo, la eliminación de pasos manuales y la mayor eficiencia en las comunicaciones han resultado en una mejora notable en la velocidad y agilidad del proceso. El sistema ha demostrado ser un recurso valioso para la optimización de los procedimientos y la mejora de la eficiencia administrativa en la Universidad de las Fuerzas Armadas ESPE.

## **Análisis de Eficiencia Administrativa**

En este análisis, se evaluó el impacto del sistema Workflow vía web en la eficiencia administrativa del proceso de titulación en la Universidad de las Fuerzas Armadas ESPE. Se examinaron los cambios en la gestión de tareas, el manejo de documentos y la comunicación interna para determinar cómo la automatización y la centralización de procesos han contribuido a mejorar la eficiencia en la administración.

## **Proceso Manual y Desafíos Administrativos**

Antes de la implementación del sistema, el proceso de titulación se basaba en métodos manuales que requerían una gran cantidad de recursos y tiempo. Los empleados administrativos tenían que manejar múltiples solicitudes, verificar requisitos y asegurarse de que los documentos estuvieran completos y correctos. Además, la comunicación entre diferentes departamentos y con los estudiantes a menudo se realizaba a través de correos electrónicos y reuniones presenciales, lo que generaba demoras y la posibilidad de malentendidos.

## **Automatización de Flujos de Trabajo y Simplificación**

Con la introducción del sistema Workflow vía web, se observó una transformación significativa en la forma en que se manejaban las tareas administrativas relacionadas con la

titulación. Los flujos de trabajo se automatizaron, lo que permitió la asignación automática de tareas, recordatorios y notificaciones de progreso. Los empleados administrativos ahora podían acceder a un tablero centralizado que mostraba el estado en tiempo real de cada solicitud, lo que facilitó la gestión y el seguimiento.

### **Resultados y Beneficios**

Los resultados de este análisis indican una mejora sustancial en la eficiencia administrativa tras la implementación del sistema Workflow vía web:

**Reducción de Carga de Trabajo:** La automatización de tareas liberó a los empleados administrativos de actividades repetitivas y les permitió enfocarse en tareas más estratégicas y de mayor valor.

**Agilización de Procesos:** La asignación automática de tareas y la eliminación de pasos manuales redujeron significativamente el tiempo necesario para procesar cada solicitud de titulación.

**Mejora en la Comunicación:** La plataforma centralizada de comunicación permitió un intercambio más rápido y claro de información entre diferentes departamentos y con los estudiantes, reduciendo la necesidad de reuniones presenciales y correos electrónicos.

**Seguimiento Preciso:** El tablero de control proporcionó una visión en tiempo real del estado de cada solicitud, lo que permitió a los administradores identificar cuellos de botella y tomar medidas correctivas de manera oportuna.

### **Conclusión**

El análisis de eficiencia administrativa demuestra de manera concluyente que la implementación del sistema Workflow vía web ha tenido un impacto positivo en la mejora de la eficiencia administrativa en el proceso de titulación. La automatización de flujos de trabajo, la centralización de información y la simplificación de tareas han resultado en una

reducción significativa de la carga de trabajo, una agilización de procesos y una mejora en la comunicación interna. En última instancia, el sistema ha contribuido a una administración más eficiente y efectiva en la Universidad de las Fuerzas Armadas ESPE.

## Capítulo VI

### Conclusiones

La implementación del sistema Workflow vía web ha representado un avance significativo en la optimización de los procesos de titulación en la Universidad de las Fuerzas Armadas ESPE. A través de un análisis exhaustivo de los resultados, se ha confirmado que esta solución tecnológica ha tenido un impacto positivo y sustancial en varios aspectos clave, contribuyendo a la eficiencia administrativa, la guía para los estudiantes y la mejora general del proceso académico. Las conclusiones fundamentadas en los datos recopilados y el análisis realizado son las siguientes:

**Eficiencia Administrativa Mejorada:** El sistema Workflow vía web ha demostrado su capacidad para optimizar la gestión administrativa relacionada con el proceso de titulación. La automatización de flujos de trabajo ha reducido la carga de trabajo manual, permitiendo a los empleados administrativos enfocarse en tareas más estratégicas y de mayor valor. La asignación automática de tareas y los recordatorios han acelerado el procesamiento de solicitudes, lo que se traduce en una reducción significativa en los tiempos de respuesta y aprobación.

**Agilización del Proceso de Titulación:** La implementación del sistema ha tenido un impacto positivo en la velocidad y fluidez del proceso de titulación. La eliminación de pasos manuales y la centralización de la información han agilizado la obtención de aprobaciones, la verificación de requisitos y la generación de documentos necesarios. Esto ha conducido a una reducción sustancial en el tiempo total requerido para completar el proceso de

titulación, mejorando la experiencia tanto para los estudiantes como para el personal administrativo.

**Mejora en la Experiencia Estudiantil:** Los resultados demuestran que el sistema Workflow vía web ha proporcionado a los estudiantes una guía clara y un acceso más rápido a información relevante durante su proceso de titulación. La plataforma centralizada ha facilitado la presentación de documentos, el seguimiento de requisitos y la comunicación con las autoridades académicas. Esto ha contribuido a una experiencia más fluida y transparente para los estudiantes, reduciendo la incertidumbre y los obstáculos en su camino hacia la titulación.

**Cumplimiento de Regulaciones y Normativas:** El análisis ha confirmado que el sistema ha fortalecido el cumplimiento de las regulaciones y normativas relacionadas con el proceso de titulación en la universidad. La automatización de flujos de trabajo ha asegurado que los requisitos y procedimientos establecidos sean seguidos de manera consistente en todas las solicitudes de titulación, minimizando el riesgo de omisiones o errores.

En resumen, la implementación exitosa del sistema Workflow vía web ha transformado positivamente el proceso de titulación en la Universidad de las Fuerzas Armadas ESPE. Los resultados obtenidos reflejan una mejora evidente en la eficiencia administrativa, una experiencia más ágil para los estudiantes y una mayor conformidad con las regulaciones. Esta solución tecnológica no solo ha optimizado los procedimientos de titulación, sino que también ha sentado las bases para una administración más eficiente y efectiva en otros ámbitos académicos.

### **Recomendaciones**

Basadas en el análisis exhaustivo de los resultados y la experiencia adquirida durante la implementación del sistema Workflow vía web, se presentan las siguientes recomendaciones para optimizar aún más la eficiencia y el impacto del sistema en el proceso de titulación:

**Capacitación Continua:** Establecer programas de capacitación periódica para estudiantes, profesores y personal administrativo sobre el uso efectivo del sistema. Esto asegurará que todos los usuarios estén plenamente informados y aprovechen al máximo las capacidades del sistema.

**Mejoras Incrementales:** Continuar con el monitoreo constante de la interacción de los usuarios con el sistema y recopilar sus comentarios para identificar áreas de mejora. Implementar actualizaciones incrementales para abordar desafíos específicos y optimizar aún más la experiencia del usuario.

**Personalización:** Brindar opciones para personalizar la interfaz de usuario según las preferencias individuales de los usuarios, permitiendo una experiencia más adaptada a sus necesidades y preferencias.

**Integración Adicional:** Explorar oportunidades para una mayor integración con otros sistemas utilizados en la universidad, como sistemas de gestión de cursos y bibliotecas digitales. Esto podría facilitar aún más la recopilación y el intercambio de información relevante.

**Monitoreo y Análisis de Datos:** Establecer un sistema de monitoreo continuo para rastrear métricas clave de desempeño, como tiempos de procesamiento, tasas de error y tiempos de respuesta. Utilizar estos datos para identificar tendencias y áreas de mejora.

**Feedback de Usuarios:** Fomentar una cultura de retroalimentación entre los usuarios, brindando canales para que los estudiantes, profesores y personal administrativo compartan sus opiniones y sugerencias. Esta información puede ser inestimable para la evolución futura del sistema.

**Exploración de Nuevas Funcionalidades:** Investigar la posibilidad de agregar nuevas funcionalidades al sistema, como la integración de herramientas de colaboración en línea, que puedan enriquecer aún más la experiencia de los usuarios y fomentar la colaboración.

**Seguridad y Actualizaciones:** Mantener un enfoque continuo en la seguridad de los datos y la protección de la privacidad de los usuarios. Realizar actualizaciones regulares del

sistema y parches de seguridad para mantenerlo al día con las últimas medidas de seguridad.

**Evaluación de Satisfacción:** Realizar encuestas periódicas de satisfacción de los usuarios para evaluar su percepción del sistema y su experiencia general. Utilizar los resultados para implementar mejoras específicas y medir el progreso con el tiempo.

**Planificación a Largo Plazo:** Desarrollar una estrategia a largo plazo para la evolución y mejora continua del sistema, asegurando que esté alineado con las necesidades cambiantes de la universidad y los avances tecnológicos.

Estas recomendaciones buscan fortalecer la efectividad y el impacto del sistema Workflow vía web en la Universidad de las Fuerzas Armadas ESPE. Al implementar estas sugerencias, la universidad puede continuar mejorando la eficiencia administrativa, la experiencia estudiantil y la calidad general del proceso de titulación.

## Referencias

- A. B. (Marzo de 2023). *Sistema web integrado para la gestión de trámites académico administrativos*. Obtenido de Sistema web integrado para la gestión de trámites académico administrativos en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial: <https://repositorio.uta.edu.ec/jspui/handle/123456789/37732>
- Aguayo Álvarez, Lucia Alejandra. (2015). *Las tecnologías de información y comunicación (Tic's) y su aporte a la actividad secretarial de la Unidad de Estudios a Distancia de la Universidad Técnica Estatal de Quevedo, del cantón Quevedo, provincia de Los Ríos, año 2015*. Obtenido de Las tecnologías de información y comunicación (Tic's) y su aporte a la actividad secretarial de la Unidad de Estudios a Distancia de la Universidad Técnica Estatal de Quevedo, del cantón Quevedo, provincia de Los Ríos, año 2015: <http://repositorio.uteq.edu.ec/handle/43000/5101>
- Alvarado Zambrano, Erick Alexis, B. (10 de Enero de 2023). *Aplicación web para el servicio de trámites académicos de la UNACH usando una arquitectura basada en microservicios*. Obtenido de Aplicación web para el servicio de trámites académicos de la UNACH usando una arquitectura basada en microservicios: <http://dspace.unach.edu.ec/handle/51000/10247>
- Moreno Lalama, Lizeth Guadalupe, M., & Carrasco Morales, Cynthia Nicole. (Marzo de 2023). *Transformación digital en los procesos de la Unidad de Integración Curricular*. Obtenido de Transformación digital en los procesos de la Unidad de Integración Curricular de la Facultad de Ciencias Administrativas de la Universidad Técnica de Ambato: <https://repositorio.uta.edu.ec/jspui/handle/123456789/37749>
- Suquitana Fajardo, Byron Fabian, M. (Mayo de 2018). *Aplicación de Sistemas Workflow en la Gestión Académica*. Obtenido de Aplicación de Sistemas Workflow en la Gestión Académica: <http://repositorio.unemi.edu.ec/xmlui/handle/123456789/4091>

Villavicencio Meza, Jorge Luis. (2020). *Diseño de un programa de mejora de la gestión académica en la Universidad Nacional Federico Villarreal, San Miguel*. Obtenido de Diseño de un programa de mejora de la gestión académica en la Universidad Nacional Federico Villarreal, San Miguel: <https://hdl.handle.net/20.500.12692/55680>

## **Apéndices**