# UNIVERSIDAD NACIONAL DE CAÑETE

## FACULTAD DE INGENIERÍA

### ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

# Marco de pruebas automatizadas para el software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete

## TESIS PARA OPTAR EL TITULO PROFESIONAL DE INGENIERO DE SISTEMAS


**AUTOR(ES):**
**EDUARDO ZOSIMO ROMAN SANCHEZ**
**orcid.org/0000-0001-8414-8560**

**JOSE ADRIAN ANCAJIMA RODRIGUEZ**
**orcid.org/0009-0001-0400-096X**

**ASESOR Y/O COASESOR:**
**MAG. AMANDA DURAN CARHUAMACA**
**orcid.org/0000-0001-8183-5891**
**MAG. LEONIDAS ASTO HUAMAN**
**orcid.org/0000-0003-2003-1798**


FIRMA DIGITAL
Firmado digitalmente por: DURAN CARHUAMACA AMANDA FIR 20114878 hard
Motivo: En señal de conformidad
Fecha: 13/07/2024 21:38:36-0500

**LINEA DE INVESTIGACION**
**PLATAFORMA TIC**

**San Vicente de Cañete, Perú, 2024**

FIRMA DIGITAL
Firmado digitalmente por: ASTO HUAMAN Leonidas FAU 20491363402 soft
Motivo: Doy V° B°
Fecha: 14/07/2024 21:12:31-0500

## Firma de los jurados

FIRMA DIGITAL

Firmado digitalmente por: OSEDA GAGO DULIO
Motivo: Soy el autor del documento
Fecha: 07/08/2024 09:36:34-0500

**Dr. Dulio Oseda Gago**

**Secretario**

**Jurado Evaluador**

Facultad de Ingeniería



**Mg. Eddye Arturo Sanchez Castillo**

**Presidente**

**Jurado Evaluador**

Facultad de Ingeniería

FIRMA DIGITAL

Firmado digitalmente por: ANGOMA ASTUCURI MIRIAM FIR 20044924 hard
Motivo: Soy el autor del documento
Fecha: 07/08/2024 12:39:41-0500

**Dra. Miriam Angoma Astucuri**

**Vocal**

**Jurado Evaluador**

Facultad de Ingeniería

## Acta de sustentación

UNIVERSIDAD NACIONAL DE CAÑETE

| Código: F-M01.03.04-GT-003<br>Fecha de Aprobación: 16-10-2023 | Versión: 04 |
|---|---|

**FACULTAD DE INGENIERIA**

**ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS**

En la ciudad de Cañete, en la sala de sustentaciones N° 03 de la Facultad de Ingeniería, siendo las 09:00 horas, del día 21 de julio del 2024, se reunió el Jurado Evaluador y el asesor, designados por DECRETO 014-2024-UI-FI/UNDC.
Presidente(a): Mg. Eddye Arturo Sanchez Castillo
Secretario (a): Ph.D. Dulio Oseda Gago
Vocal : Dra. Miriam Angoma Astucuri

Asesor : Mg. Amanda Durán Carhuamaca

Para el acto de sustentación de la tesis titulada: **Marco de pruebas automatizadas para el software de gestión digital de procesos de investigación de la Universidad Nacional de Cañete**; presentada por el Bachiller (es) **Roman Sanchez Eduardo Zosimo y Ancajima Rodriguez Jose Adrian** para optar al Título Profesional de Ingeniero de Sistemas.

Después de concluido el acto de sustentación y luego de que el sustentante dio respuesta a las preguntas respectivas, el Jurado Evaluador calificó la sustentación o defensa de la investigación según los siguientes criterios, y promediando las notas del jurado se obtuvo:

| **Tesista: Roman Sanchez Eduardo Zosimo** | | | | | | |
|---|---|---|---|---|---|---|
| **Criterio** | **Desaprobado (menor de 14)** | **Regular (14 -15)** | **Bueno (16-17)** | **Muy bueno (18-19)** | **Excelente (20)** | **Puntaje** |
| Claridad de la exposición | | | 17 | | | 17 |
| Dominio del tema | | | 17 | | | 17 |
| Visión holística del trabajo | | | 16 | | | 16 |
| **Promedio** | | | | | | **17** |

| **Tesista: Ancajima Rodriguez Jose Adrian** | | | | | | |
|---|---|---|---|---|---|---|
| **Criterio** | **Desaprobado (menor de 14)** | **Regular (14 -15)** | **Bueno (16-17)** | **Muy bueno (18-19)** | **Excelente (20)** | **Puntaje** |
| Claridad de la exposición | | | 17 | | | 17 |
| Dominio del tema | | | 17 | | | 17 |
| Visión holística del trabajo | | | 16 | | | 16 |
| **Promedio** | | | | | | **17** |

F-M01.03.04-GT-003Toda copia de este documento, sea del entorno virtual o del documento original en físico es considerada "copia no controlada"

Para obtener la nota final de todo el proceso de investigación, conforme al artículo 100° del Reglamento General de Investigación de la Universidad Nacional de Cañete aprobado mediante Resolución Nº 180-2023-UNDC/CO, se procedió a calcular la nota final:

| Tesista | Nota del Informe (valor 40%) | Nota de Sustentación (valor 60%) | Nota Final |
|---|---|---|---|
| **Roman Sanchez Eduardo Zosimo** | 15 | 17 | 16 |
| **Ancajima Rodriguez Jose Adrian** | 15 | 17 | 16 |

Y, establece como calificación FINAL: 16 (Diesiseis)

- Aprobado, con nota Regular ( ), Bueno (X), Muy Bueno (...), Excelente (...)
- Desaprobado (...)

Por tanto, el graduado se encuentra expedito(s) ( X ), impedido(s) ( ) para realizar los trámites que corresponden a la obtención del Título Profesional de Ingeniero de Sistemas.

Siendo las 09:30 Horas se dio por terminado el acto de sustentación.

| | | | |
|---|---|---|---|
| Ph.D. Dulio Oseda Gago<br>**Secretario** | Mg. Eddye Arturo Sanchez Castillo<br>**Presidente** | Dra. Miriam Angoma Astucuri<br>**Vocal** | Mg. Amanda Durán Carhuamaca<br>**Asesor** |

F-M01.03.04-GT-003Toda copia de este documento, sea del entorno virtual o del documento original en físico es considerada "copia no controlada"

## Dedicatoria

Dedico esta tesis a las personas que han sido mi inspiración, apoyo y amor incondicional a lo largo de este emocionante viaje académico.

A mi familia, por su apoyo constante y su confianza en mí en momentos en que las incertidumbres oscurecían mi camino. Cada logro que he logrado refleja su amor y sacrificio.

## Agradecimientos

Quiero expresar mi más sincera gratitud a todas las personas que han dado un gran aporte a la realización de esta tesis. Sin su apoyo inquebrantable, orientación y aliento constante, este logro no habría sido posible.

Quiero expresar mi gratitud a Dios por su presencia constante con nosotros, así como a los asesores de nuestra Tesis por su dedicación, habilidad y paciencia durante este proceso. Su valiosa recomendación y guía fueron esenciales para la creación de esta investigación.

# INDICE

Portada ..... I
Firma de los jurados ..... II
Acta de sustentación ..... III
Dedicatoria ..... V
Agradecimientos ..... VI
Índice de tablas ..... X
Índice de Figuras ..... XI
Índice de Ilustración ..... XII
Resumen ..... XIII
Abstract ..... XIV
Capítulo I: Introducción ..... 1
1.1 Planteamiento del problema ..... 1
1.2 Formulación del problema ..... 3
1.2.1 Problema general ..... 3
1.2.2 Problemas específicos ..... 3
1.3 Objetivos ..... 3
1.3.1 Objetivo General ..... 3
1.3.1 Objetivos Específicos ..... 3
1.4 Justificación ..... 3
1.4.1 Justificación Teórica ..... 3
1.4.2 Justificación Práctica ..... 4
1.4.3 Justificación Tecnológica ..... 4
1.4.4 Justificación Metodológica ..... 5
1.5 Delimitación de la investigación ..... 5
1.5.1 Delimitación teórica o conceptual ..... 5
1.5.2 Delimitación temporal ..... 6
1.5.3 Delimitación espacial ..... 6
Capítulo II: Marco Teórico ..... 7
2.1. Antecedentes de investigación ..... 7
2.1.1. Artículos científicos ..... 7
2.1.2. Tesis nacionales e internacionales ..... 9
2.2. Bases teóricas ..... 13
2.2.1 Variable Independiente: Pruebas Automatizadas ..... 13
2.2.2 Variable Dependiente: Eficiencia en el Proceso de Pruebas ..... 17
2.3. Definición de términos básicos ..... 24

2.3.1 BDD ..... 24
2.3.2 Page Object Model ..... 24
2.3.3 Pruebas Funcionales ..... 25
2.3.4 Pruebas no Funcionales ..... 25
2.3.5 Selenium Web Driver ..... 25
2.3.6 Cucumber ..... 25
2.3.7 Gherkin ..... 26
2.3.8 Junit ..... 26
2.3.9 Java ..... 26
2.3.10 Scrum ..... 26
Capítulo III: Hipótesis y Variables ..... 28
3.1. Hipótesis ..... 28
3.1.1. Hipótesis general ..... 28
3.1.2. Hipótesis específicas ..... 28
3.2. Definición conceptual de las variables ..... 28
3.2.1. Variable Independiente: Pruebas Automatizadas ..... 28
3.2.2. Variable Dependiente: Eficiencia en el Proceso de Pruebas ..... 28
3.3. Operacionalización de las variables (Definición operacional) ..... 29
3.3.1. Variable Independiente: Pruebas Automatizadas ..... 29
3.3.2. Variable Dependiente: Eficiencia en el Proceso de Pruebas ..... 30
Capítulo IV: Metodología ..... 31
4.1. Enfoque de la investigación ..... 31
4.2. Tipo de investigación ..... 31
4.3. Nivel de investigación ..... 31
4.4. Método de investigación ..... 32
4.5. Diseño de investigación ..... 32
4.6. Población, muestra y muestreo ..... 32
4.6.1. Población ..... 32
4.6.2. Muestra ..... 33
4.6.3. Muestreo ..... 33
4.7. Técnicas e instrumentos de recolección de datos ..... 33
4.7.1. Técnica ..... 33
4.7.2. Instrumento ..... 33
4.8. Técnicas estadísticas de análisis de datos ..... 34
4.9. Consideraciones éticas ..... 34
Capítulo V: Resultados ..... 35

5.1. Resultados Descriptivos ..... 35
5.1.1. Evaluación descriptiva del indicador 1: Número de errores detectados (NED) ..... 35
5.1.2. Evaluación descriptiva del indicador 2: Tiempo de ejecución de las pruebas (TEP). ..... 36
5.2. Análisis inferencial ..... 37
5.2.1. Pruebas de normalidad ..... 37
5.3. Prueba de Hipótesis ..... 40
Capítulo VI: Discusión ..... 42
Capítulo VII: Conclusiones ..... 44
Capítulo VIII: Recomendaciones ..... 45
Referencias ..... 46
Anexos ..... 49
Anexo 1: Matriz de consistencia ..... 49
Anexo 2: Instrumento de recolección de datos ..... 50
Anexo 3: Certificado de validez de contenido (Juicio de expertos) ..... 51
Anexo 4: Constancia de grados y títulos de los validadores (SUNEDU) ..... 54
Anexo 5: Base de datos de los indicadores (recolección de datos) ..... 57
Anexo 6: Autorización de la institución para realizar la investigación ..... 63
Anexo 7: Autorización de la inclusión en el repositorio de la biblioteca ..... 65
Anexo 8: Autorización de publicación en el repositorio institucional de la UNDC ..... 66
Anexo 9: Constancia de Similitud ..... 68
Anexo 10: Metodología de desarrollo de la solución tecnológica ..... 69
Anexo 11: Artículo científico ..... 111
6.1. Aspectos Administrativos Cronograma de actividades ..... 125
6.2. Presupuesto ..... 127
Presupuesto total del proyecto ..... 127
5.3. Financiamiento ..... 127

## Índice de tablas

Tabla I - Evaluación descriptiva del indicador NED ..... 35
Tabla II -Evaluación descriptiva del indicador TEP ..... 36
Tabla III - Test de normalidad del indicador NED ..... 38
Tabla IV - Prueba de Confiabilidad del indicador NED ..... 38
Tabla V - Test de normalidad del indicador TEP ..... 39
Tabla VI - Combinación lineal - Test de normalidad del indicador TEP ..... 39
Tabla VII - Prueba de Confiabilidad del indicador TEP ..... 39
Tabla VIII - Prueba de U de Mann-Whitney del indicador NED ..... 40
Tabla IX - Prueba de U de Mann-Whitney del indicador TEP ..... 41
Tabla X - Cuadro de roles del Proyecto ..... 69
Tabla XI - Tabla general matriz de casos ..... 72
Tabla XII - Matriz de casos login y perfil de usuario ..... 73
Tabla XIII - Matriz de casos modulo Iniciar un nuevo proyecto ..... 75
Tabla XIV - Matriz de casos modulo Divisiones ..... 78
Tabla XV - Matriz de casos modulo Investigador ..... 79
Tabla XVI - Matriz de casos modulo Proyecto ..... 80
Tabla XVII - Matriz de casos de Carga de documento ..... 82

# Índice de Figuras

Figura 1 - Diagrama de Barras del indicador NED ..... 35
Figura 2 - Diagrama de Caja de Bigotes del indicador TEP ..... 37
Figura 3 - Backlog del proyecto ..... 69
Figura 4 - Diagrama Gedipi ..... 71
Figura 5 - Tablero Sprint 1 ..... 72
Figura 6 - Tablero Sprint 2 ..... 85
Figura 7 - Iniciar Sesión ..... 85
Figura 8 - Editar perfil ..... 86
Figura 9 - Perfil editado ..... 86
Figura 10 - Editar perfil - cambiar contraseña ..... 87
Figura 11 – Modalidad ..... 87
Figura 12 - Ingresar datos del proyecto ..... 88
Figura 13 - Agregar autores, asesor, coasesor ..... 89
Figura 14 - Registro correctamente del proyecto ..... 89
Figura 15 - Búsqueda por divisiones (área, línea, sublinea) ..... 90
Figura 16 - Búsqueda por investigador ..... 90
Figura 17 - Búsqueda por proyecto ..... 91
Figura 18 - Carga de documento ..... 91
Figura 19 - Tablero Sprint 3 ..... 92
Figura 20 - Estructura del proyecto ..... 93
Figura 21 - Feature Login ..... 94
Figura 22 - Pages login ..... 94
Figura 23 - Definitions Login ..... 95
Figura 24 - Reportes tiempo de ejecución del login ..... 96
Figura 25 - Feature edición perfil ..... 97
Figura 26 - Pages edición perfil ..... 97
Figura 27 - Definitions edición perfil ..... 98
Figura 28 - Tiempo de ejecución prueba perfil ..... 99
Figura 29 - Tiempo de ejecución de cada paso Perfil ..... 99
Figura 30 - Features busqueda ..... 100
Figura 31 - Pages búsqueda ..... 101
Figura 32 - Definitions búsqueda ..... 102
Figura 33 - Tablero Sprint 4 ..... 103
Figura 34 - Feature crear nuevo proyecto Pre grado ..... 103
Figura 35 - Feature crear nuevo proyecto “Postgrado” ..... 104
Figura 36 – Pages crear nuevo proyecto ..... 104
Figura 37 – Definitions crear nuevo proyecto ..... 105
Figura 38 - Tiempo de ejecución de la prueba crear nuevo proyecto ..... 106
Figura 39 - Tiempo de ejecución de la prueba crear nuevo proyecto paso a paso ..... 107
Figura 40 - Features carga de documento ..... 108
Figura 41 - Pages carga de documento ..... 108
Figura 42 - Definitions subir documento ..... 109
Figura 43 - Tabla de detección de errores ..... 110
Figura 44 - Grafica de número de errores detectados ..... 110

## Índice de Ilustración

Ilustración 1 - Lenguaje Gherkin ..... 16
Ilustración 2 - Metodología scrum - Sprint ..... 19
Ilustración 3 - Estudio de metodologías ágiles. Aplicación SCRUM ..... 20

## Resumen

La Universidad Nacional de Cañete enfrenta importantes desafíos en el proceso de pruebas del software de Gestión Digital de Procesos de Investigación (GEDIPI), debido a la creciente complejidad del software utilizado. A través de la implementación de un Marco de Pruebas Automatizadas, este estudio busca abordar estos problemas y mejorar la eficacia del software de GEDIPI. Los métodos utilizados para esta automatización son framework como Selenium WebDriver, lenguaje de programación Java, Gherkin ,BDD y el patrón de diseño POM, bajo la metodología ágil Scrum, la automatización de pruebas es crucial para mantener la calidad del software y acelerar el ciclo de desarrollo de GEDIPI.

El problema principal es la ineficiencia del proceso de pruebas actual, que se caracteriza por tiempos prolongados, costos elevados y baja cobertura de pruebas. Este estudio examina cómo la automatización de las pruebas puede asegurar la calidad del software mediante pruebas automatizadas integradas de manera eficiente en el ciclo de desarrollo ágil y aumentar la productividad del proceso de pruebas, asi como tambien detectar cada errores ya sea funcionales y no funcionales. Se cree que el Marco de Pruebas Automatizadas mejorará significativamente estos aspectos y brindará una solución útil para la Universidad Nacional de Cañete.



**Palabras claves-** automatización web, scrum, pom, bdd, selenium, gherkin, java

**Abstract**

The National University of Cañete faces important challenges in the testing process of the Digital Research Process Management Software (GEDIPI), due to the increasing complexity of the software used. Through the implementation of an Automated Testing Framework, this study seeks to address these problems and improve the effectiveness of GEDIPI software. The methods used for this automation are frameworks such as Selenium WebDriver, Java programming language, Gherkin, BDD and the POM design pattern, under the agile Scrum methodology, testing automation is crucial toining software quality and accelerating the GEDIPI development cycle.

The main problem is the inefficiency of the current test process, which is characterized by prolonged time, high costs and low test coverage. This study examines how testing automation can ensure the quality of software through automated testing efficiently integrated into the agile development cycle and increase the productivity of the testing process, as well as detecting each functional and non-functional errors. It is believed that the Automated Testing Framework will significantly improve these aspects and provide a useful solution for the National University of Cañete.

**Keywords**- web automation, scrum, pom, bdd, selenium, gherkin, java

# Capítulo I: Introducción

## 1.1 Planteamiento del problema

El software y la tecnología, ya sea a través de páginas web o aplicaciones móviles, son indudablemente valiosos y facilitan la vida de las personas. Sin embargo, la calidad del software puede verse afectada porque la programación es humana y, por lo tanto, propensa a errores. Estos errores son indeseables y es mejor identificarlos en las etapas tempranas del desarrollo que en los ambientes de producción.

La Automatización web es un practica garantizada hoy en día para la calidad, fiabilidad, eficiencia y la velocidad en un software, donde utilizamos herramientas, frameworks y scripts. Estas pruebas ayudan proporcionando de una manera sistemática y confiable para verificar que las aplicaciones funcionen según lo esperado.

A nivel internacional, según [1] comenta que la automatización de pruebas es un paso fundamental para mejorar la calidad del software, ya que las aplicaciones se vuelven más complejas e incorporan funcionalidades adicionales. Esto conduce a una disminución en la confiablidad de la calidad final del software desarrollado. En estos casos la automatización de pruebas se vuelve muy importante a la hora de abordar el proceso de ejecución de las pruebas. Pueden ser aplicadas en diferentes niveles de casos de pruebas, desde pruebas unitarias, hasta pruebas de aceptación. En la actualidad, hay una variedad de herramientas disponibles para crear y ejecutar scripts, que son pequeños programas que el usuario puede interactuar con una aplicación específica y simular las acciones que podrían realizar de manera real. Dentro del testeo de aplicaciones con interfaz web, una de las herramientas de automatización de pruebas funcionales que más importancia está cobrando es Selenium. Selenium es un framework de desarrollo opensource con una gran comunidad que permite a los usuarios finales navegar a través de una aplicación web. Un script que se procesa por la herramienta indica los pasos que se deben seguir durante la ejecución de la prueba, como abrir el navegador, acceder a una URL específica, analizar el texto que se muestra en pantalla, acceder a un enlace específico, introducir datos, etc. Selenium ha avanzado y ahora ofrece una API completa para varios lenguajes de programación. Con las clases y métodos que ofrece, se pueden crear scripts que representan cada prueba automatizada. Actualmente la herramienta soporta los principales navegadores web como Chrome, Firefox, Internet Explorer, Safari, etc.

A nivel nacional, según [2] comenta que las pruebas de software automatizadas aumentan la eficacia y la eficiencia. Las enormes ventajas de estas pruebas incluyen una reducción del tiempo de prueba y la capacidad de cubrir más escenarios. reducir en gran medida los errores manuales reduce significativamente las fallas manuales y utiliza pocos recursos en pruebas. Sin embargo, las pruebas automatizadas no garantizan la falta de fallas en la producción en un 100%. Los escenarios de pruebas en las plataformas son inmensos, por lo que se deben realizar pruebas constantemente para reducir la cantidad de fallos.

Esta investigación se examina el software de gestión digital de procesos de investigación que facilita la búsqueda de investigaciones, la creación de nuevos proyectos y la aprobación de investigaciones, entre otras cosas. Sin embargo, se encontró que durante el proceso de desarrollo de este sistema no se realizaron pruebas manuales o funcionales adecuadas. Se reconoce la necesidad de implementar pruebas automatizadas en el futuro a medida que el sistema y su complejidad aumentan. Este método no solo promete mejorar la eficiencia y consistencia de las pruebas, sino que también permitirá detectar errores de manera más rápida y precisa, lo que liberará recursos humanos para tareas más estratégicas durante el proceso de desarrollo de software. Como resultado, el objetivo de esta investigación es evaluar la calidad del software de manera automatizada utilizando la herramienta Selenium WebDriver y el lenguaje Gherkin, que proporciona una descripción clara y concisa de cada escenario de prueba.

## 1.2 Formulación del problema

### 1.2.1 Problema general

¿De qué manera la implementación de un Marco de Pruebas Automatizadas puede mejorar la eficacia del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete?

### 1.2.2 Problemas específicos

**PE1:** ¿De qué manera la implementación de un Marco de Pruebas Automatizadas puede mejorar la calidad del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete?

**PE2:** ¿De qué manera la implementación de un Marco de Pruebas Automatizadas puede incrementar la productividad en el proceso de pruebas del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete?

## 1.3 Objetivos

### 1.3.1 Objetivo General

Evaluar de qué manera la implementación de un Marco de Pruebas Automatizadas puede mejorar la eficacia del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

### 1.3.1 Objetivos Específicos

**OE1:** Evaluar de qué manera la implementación de un Marco de Pruebas Automatizadas puede mejorar la calidad del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

**OE2:** Evaluar de qué manera la implementación de un Marco de Pruebas Automatizadas puede incrementar la productividad en el proceso de pruebas del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

## 1.4 Justificación

### 1.4.1 Justificación Teórica

[3] definen la justificación teórica como la explicación de las razones científicas y teóricas que respaldan la realización de una investigación, destacando cómo el estudio contribuirá al desarrollo del conocimiento en un campo específico a través de la resolución

de problemas teóricos y la generación de nuevas hipótesis.

En el actual contexto del desarrollo de software, la calidad aborda un papel fundamental en la satisfacción del usuario y el éxito de un proyecto, un marco de pruebas automatizadas surge como un plan estratégico que busca mejorar el proceso de ejecución de pruebas de software en el sistema de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete. Esta investigación busca abordar la necesidad de optimizar el proceso de verificación y validación en el desarrollo de software. Al utilizar herramientas y métodos de automatización, se espera reducir los costos en comparación con las pruebas manuales, así como aumentar la eficiencia, la robustez y la confiabilidad del sistema. Esto permite un entregable más rápido sin comprometer la estabilidad del portal y las nuevas funcionalidades que se desarrollen.

### 1.4.2 Justificación Práctica

[3] definen la justificación práctica como la explicación de los beneficios tangibles y aplicables que un estudio puede proporcionar, destacando la utilidad de los resultados para resolver problemas específicos, mejorar prácticas o contribuir a la toma de decisiones en contextos particulares.

Al implementar un marco de pruebas automatizadas en el software de Gestión Digital de Procesos de Investigación, se busca agilizar la detección y corrección de errores, optimizando la experiencia del usuario para garantizar el funcionamiento libre de errores. Además, se busca eliminar ciertas actividades repetitivas de recursos humanos, realizando otras actividades en desarrollo de software, para promover una cultura de adaptabilidad y mejora continua en un entorno tecnológico.

### 1.4.3 Justificación Tecnológica

Según [4] ,la justificación tecnológica debe evaluar la innovación que la tecnología introduce en el área de estudio, su disponibilidad y accesibilidad, así como su capacidad de aplicarse y adaptarse al contexto específico del proyecto.

De esta manera, justificación tecnológica para un marco de pruebas automatizadas radica en la creciente complejidad de las aplicaciones web y la necesidad de mantener altos estándares de calidad en un entorno que cambia constantemente. Utilizando frameworks de automatización, es posible analizar exhaustivamente las funcionalidades del software de Gestión Digital de Procesos de Investigación, incluyendo las funciones de los usuarios,

el rendimiento del sistema y la integridad de los datos. Esto permite detectar y corregir errores de manera temprana, y adaptarse a los cambios en el código y la implementación de nuevos requisitos, asegurando la fiabilidad y estabilidad del software.

### 1.4.4 Justificación Metodológica

[3] definen la justificación metodológica como la explicación de por qué el enfoque, los métodos y las técnicas elegidos para la investigación son apropiados para alcanzar los objetivos del estudio, destacando cómo estos elementos metodológicos contribuyen a la validez y confiabilidad de los resultados obtenidos.

Un marco de pruebas automatizadas se justifica metodológicamente, en la necesidad de adoptar un enfoque estructurado para evaluar la calidad de software de manera continua. Mediante el uso de metodologías agiles como Scrum y junto con las técnicas de automatización de pruebas como BDD (Behavior Driven Development) donde se pueden integrar la verificación y validación de las funcionalidades en todas las etapas del ciclo de desarrollo, lo que permite la detección temprana de errores, una mejora continua en la calidad de software y reducir los tiempos en entregar nuevas funcionalidades.

## 1.5 Delimitación de la investigación

### 1.5.1 Delimitación teórica o conceptual

Esta tesis se enfoca en implementar un marco de pruebas automatizadas para el software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete, utilizando dos enfoques clave: el modelo Page Object Model (POM) y el desarrollo guiado por comportamiento (Behavior Driven Development, BDD). El POM es un patrón de diseño que facilita la creación de clases que representan páginas web, haciendo que el código de prueba sea más fácil de mantener y leer. Por su parte, BDD es una metodología ágil que fomenta la colaboración entre desarrolladores, testers y partes interesadas no técnicas, mediante la creación de casos de prueba en lenguaje natural que describen el comportamiento esperado del sistema. Estos enfoques se utilizan para evaluar la eficiencia del proceso de pruebas, centrándose en la funcionalidad, fiabilidad, usabilidad y rendimiento, garantizando así una cobertura completa y precisa del sistema bajo prueba.

### 1.5.2 Delimitación temporal

La investigación actual se llevará a cabo entre abril y julio de 2024. El diseño y desarrollo del marco de pruebas automatizadas son parte de este marco temporal. También incluye la implementación y validación del marco en un entorno simulado que imita las funciones del sistema de gestión digital de procesos de investigación (GEDIPI). La revisión de la literatura, la creación de estándares de prueba, la realización de pruebas automatizadas y el análisis de los resultados son actividades específicas. Además, este período incluye la redacción y presentación final de la tesis, lo que garantiza que todas las etapas del proyecto se completen minuciosamente y dentro del plazo establecido.

### 1.5.3 Delimitación espacial

Esta tesis se enfoca en la Universidad Nacional de Cañete, en particular en el ámbito del software de Gestión Digital de Procesos de Investigación (GEDIPI) empleado en dicha institución. El propósito de esta investigación es desarrollar y validar un marco de pruebas automatizadas que sea compatible con este sistema. El objetivo es mejorar la eficiencia y eficacia de las pruebas de software en el entorno académico y administrativo de la Universidad. Esta limitación garantiza que los hallazgos y conclusiones de la investigación sean relevantes y aplicables a las necesidades particulares de la institución.

# Capítulo II: Marco Teórico

## 2.1. Antecedentes de investigación

### 2.1.1. Artículos científicos

Según [5] en su artículo, nos explica que el objetivo de todas las empresas es que todas las versiones de software, páginas web entren a producción sin fallas, tampoco defectos. Hoy en día hay demasiados lanzamientos de software en muy poco tiempo donde suelen pasarse fallas. Si ignoramos estos problemas, a futuro necesitaremos aumentar recursos para mantener la calidad de los productos existentes mientras agregan nuevas funciones día a día. Indica también que para esta investigación se mostrará la importancia de las pruebas automatizadas, implementando el framework de Cucumber, usando Selenium Web Driver y Gherkin, con el lenguaje de programación Java usando escenarios de BDD, en conclusión, se verán resultados que mostraran cada caso de prueba, si paso correctamente o no, además mostrará el tiempo necesario para la ejecución de cada caso de prueba y la ejecución general de las pruebas.

Según [6] en su artículo, nos explica que una de las etapas del desarrollo de software es hacer también pruebas, que tiene como objetivo asegurarse de que el software este construido para cumplir con las especificaciones, por eso cuando ejecutamos casos de pruebas, existe alta probabilidad de encontrar errores, por eso el objetivo de esta investigación es realizar pruebas utilizando métodos ATDD (Desarrollo impulsados por pruebas de aceptación) y BDD (Desarrollo impulsado por comportamiento). Este proyecto se centra en la aplicación de los métodos ATDD y BDD para probar los tres proyectos finales de los estudiantes de IT Del evaluados utilizando Robot Framework y Cucumber Framework para descubrir si estos dos métodos son efectivos y pueden mejorar el proceso de desarrollo de software. Los resultados obtenidos son que la aplicación de los métodos ATDD y BDD son efectivos y ayudan a eliminar el error lo antes posible. No dice que Tanto TDD como BDD son enfoques que se utilizan para involucrar más a todos los clientes en todo el ciclo de vida del software. Tambien que Robot Framework y Cucumber Framework son herramientas de prueba de automatización que pueden aplicar ATDD y BDD correspondientemente.

Según [7] en su artículo, nos dice que el objetivo fue analizar y comparar diferentes herramientas de automatización de pruebas con Selenium. El estudio comenzó investigando las herramientas que los equipos de control de calidad preferían. Además, Para obtener una comprensión práctica de cada una de las herramientas seleccionadas, se seleccionó un entorno web y se creó un plan de pruebas para la verificación y validación de un conjunto de funcionalidades actuales. El estudio comenzó con Selenium, luego Cypress, Robot Framework y finalmente Cucumber. Compararon cada uno con Selenium para encontrar sus fortalezas y debilidades en varios aspectos. En conclusión, se obtuvo que el software perfecto no existe, la elección de una herramienta de automatización de pruebas de software dependerá mucho de la empresa, el propósito del proyecto, los conocimientos de la persona que lo utilizará, la distribución y comunicación del equipo, el ámbito en el que se desarrolla, entre otros factores. Sin embargo, según el análisis y comparación de los resultados, Selenium no es una herramienta insuperable; existen herramientas alternativas en el mercado que se destacan en algún aspecto por encima de Selenium.

[8] en su artículo, nos dice que el objetivo fue realizar pruebas de automatización con Selenium web driver y Cucumber de un software, una vez ya este listo el software. El equipo de Garantía de Calidad creará los escenarios para probar el software. Las pruebas de software se realizan de manera metódica para encontrar errores y garantizar que los resultados sean los que esperábamos. Para aumentar las tasas de éxito, se utilizan pruebas de automatización en lugar de pruebas manuales utilizando Selenium Web Driver y Cucumber con la metodología de trabajo BDD. Se concluyo que el investigador utilizó BDD Framework y Selenium Web Driver para mejorar la comunicación y comprensión entre los participantes. Además, utiliza un lenguaje común que todos pueden comprender, lo que aumenta la visibilidad de todos los participantes. Esto permite a los desarrolladores reducir el costo de mantener el código y producir productos de alta calidad que satisfacen perfectamente las necesidades de los usuarios.

### 2.1.2. Tesis nacionales e internacionales

**Tesis nacionales**

[9]en su tesis señaló que su objetivo era descubrir cómo la automatización de pruebas afecta la eficiencia de las pruebas de software en una consultora de TI en Lima en 2022. La metodología fue de tipo aplicada, enfoque cuantitativo, método hipotético deductivo con diseño preexperimental, la técnica empleada fue la observación y el instrumento fue la ficha de observación, aplicando la prueba de la hipótesis de Student T y Wilcoxon. Los resultados mostraron que el primer indicador experimentó una disminución del 74,68 % en el tiempo medio por caso de prueba después de la automatización de pruebas, mientras que el segundo indicador experimentó un aumento del 22,0 % en el número de casos de prueba ejecutados. Se descubrió que la automatización de las pruebas mejora la eficiencia de las pruebas de software en una consultora de TI.

[10] afirmaron en su tesis que su objetivo era implementar automatización en las pruebas de software de la empresa Entrega creativa, ya que esto podría mejorar la eficiencia de las pruebas, el nivel de calidad y reducir las fallas que pueden ocurrir cuando se realizan manualmente. La metodología que se utilizó un enfoque cuantitativo y un diseño experimental-preexperimental porque solo se aplica la solución a un grupo. Se utilizó una metodología híbrida entre RUP y SCRUM, que consta de cuatro fases: inicio, elaboración, construcción y transición. Se descubrió que la automatización de pruebas de regresión mejora el manejo de recursos de la organización, reduciendo el tiempo de ejecución en un 47 %. Por lo tanto, se llegó a la conclusión de que la automatización de pruebas aumenta la eficacia, mejora la calidad y, sobre todo, reduce la cantidad de fallos durante la implementación.

Como argumenta [11] en su tesis, su objetivo era automatizar los casos de prueba para mejorar el proceso de pruebas de un sistema que proporciona servicios RPO utilizando Selenium y el TestNG Framework. Los analistas de calidad realizaron pruebas complicadas, repetitivas y que requerirían mucho tiempo durante la investigación. Resolver esos problemas fue el objetivo del desarrollo de

este proyecto. Los resultados de las pruebas se obtuvieron utilizando el proyecto UI Automatización. Las pruebas incluyen un número cada vez mayor de instancias de clientes: todas las instancias de clientes ahora se ejecutan pruebas de humo. La prueba dura menos de dos horas y no requiere ayuda humana. Se descubrió que el proyecto de interfaz de usuario automatizada pudo probar con éxito a 100 clientes en pruebas de humo y habilitó ocho recursos para incluir varias pruebas de aplicaciones más utilizables con más profundidad. A través de la integración con Jenkins, las pruebas automatizadas se ejecutan diariamente, generando informes de prueba e informando los resultados de la ejecución.

De la misma manera [12] en su tesis que tuvo como objetivo mejorar el proceso de prueba del módulo del sistema que desarrolla la empresa Experts Systems mediante la automatización de las pruebas de software, lo que permite mejorar las pruebas de software para el personal de alta calidad. Debido a que las pruebas se realizan manualmente, se demoran las evidencias de incidencia en el flujo para reportar al personal de desarrollo para su levantamiento, lo que provoca retrasos en las entregas de sprint. La metodología fue de nivel descriptivo, diseño experimental- preexperimental, se emplearon varias herramientas Open Source, incluido Selenium, un framework para la automatización del módulo del sistema que permitió cuantificar los esfuerzos realizados en cada recurso. La metodología ágil utilizada fue el marco de trabajo SCRUM, que consta de cinco sprints. Se obtuvo como resultados que a automatización ayudó a mejorar la imagen y los servicios de la empresa, teniendo la seguridad de la calidad de sus sistemas, mejorando el ambiente de calidad, para reducir el riesgo. Se concluyó que la automatización de las pruebas en el módulo del sistema mejoró la cobertura de los escenarios de prueba y la automatización de los tiempos de prueba.

**Tesis internacionales**

En España [13] indica que en su tesis el objetivo principal del trabajo es crear un entorno de trabajo con herramientas (Cucumber, Gherkin, Java) para la ejecución de pruebas automáticas en una aplicación web. En particular, este entorno de trabajo realizaría pruebas funcionales de aplicaciones web, demostrando que las

pruebas automáticas son más efectivas que las pruebas manuales. La metodología fue de nivel descriptivo y de diseño preexperimental. La metodología ágil fue el marco de trabajo SCRUM, que consta de cinco sprints. Las pruebas automáticas concluyeron que no se necesitan automatizar todos los procesos de una aplicación, sino los más rápidos, bajos en costos de mantenimiento y utilizados.

En España, según indica [14], en su investigación cuyo objetivo o es crear una herramienta que pueda generar pruebas de forma automática mediante la especificación de varios escenarios y su código correspondiente en lenguaje SOIL y comparar metodologías entre BDD Y DGC, para que podamos implementarlo en USE, Se ha descubierto gracias al desarrollo de esta herramienta que DGC también es una forma de desarrollo muy útil que se puede aplicar sobre modelos, y es algo que debe estudiarse más en profundidad porque puede ser un gran avance para el futuro.

En Costa Rica, según [15], en su investigación, cuyo objetivo fue crear un conjunto de prácticas y experiencias para la automatización de pruebas funcionales de integración en C# mediante experimentación en laboratorio, que sirvan como referente del tema para los profesionales del área de la calidad en software. La metodología que uso cuantitativa, diseño experimental, que es un método que permite identificar las causas por las que se producen resultados específicos, y la metodología ágil Kanban. En conclusión, si se analizan todas las variables y factores creados y ejecutados durante el experimento, se puede recomendar a todos los profesionales del campo de calidad el uso framework del Test Library Architecture. Si bien el marco Record and Playback tiene características ventajosas, es fundamental invertir en la creación de scripts automatizados sólidos que puedan garantizar por completo la calidad de los productos de software y permitir la reducción de costos y tiempos a largo plazo.

En Chile, según [16] cuyo objetivo fue automatizar las pruebas de regresión para reducir el esfuerzo necesario para realizar estas pruebas manualmente. La metodología empleada para el desarrollo del proyecto fue el análisis cuantitativo y cualitativo en base a la metodología de automatización de pruebas, se realizaron 07

actividades, incluyendo la referencia y el estado del arte, la definición del proceso de calidad, la selección de casos de prueba, la gestión de los casos de prueba, la automatización de casos, la ejecución de casos automatizados y la toma de métricas. Finalmente, se evaluó la solución. Se obtuvo como resultados que las pruebas automatizadas cubrían el 3,75% de las 400.000 líneas de código JAVA totales del sistema allTRA y se realizaron pruebas en 11 de las 23 funcionalidades principales del sistema. Finalmente, es importante destacar que la mayoría de las líneas de código son implementaciones para la integración y comunicación con otros sistemas.

En Ecuador, como argumenta [17],en su tesis su objetivo era crear un framework gráfico que permitiera la implementación de pruebas automatizadas con el fin de reducir el tiempo y los costos durante el proceso de desarrollo de software, especialmente en el campo de las pruebas. La metodología consistió en siete etapas distintas: documentación, análisis, diseño, desarrollo, implementación, evaluación y pruebas. Estas etapas se llevaron a cabo en un orden secuencial porque se basa en el modelo de desarrollo de software en cascada que evalúa el comportamiento de una aplicación. Se utilizaron los criterios de aceptación para reducir el tiempo de ejecución e implementación, y el primer paso es la documentación. Se descubrió que después de implementar el marco, se observó una reducción del 90 % en el tiempo de ejecución en comparación con el manual y un ahorro del 66 % en el tiempo de implementación. Esto se debió a una variedad de factores, como habilidad, conocimiento, trabajo y esfuerzo, entre otros.

## 2.2. Bases teóricas

### 2.2.1 Variable Independiente*:* Pruebas Automatizadas

#### A. Definición Pruebas Automatizadas

Las pruebas automatizadas son procesos de verificación y validación del software, como lo describe [18] es el proceso donde se utiliza herramientas, frameworks, para ejecutar pruebas determinadas en una aplicación o sistema de software. La automatización puede ser útil para automatizar tareas repetitivas y ahorrar tiempo y esfuerzo.

Se comparan los resultados reales con los esperados y se reportan los resultados automáticamente Estas pruebas permiten identificar de manera rápida errores, defectos y problemas de rendimiento durante el ciclo de desarrollo de software, lo que ayuda a mejorar la calidad del producto final y reduce los costos asociados con la detección y corrección de errores.

#### B. Dimensiones

**Cobertura de pruebas:**

[19] define la cobertura de pruebas como la medida en que las pruebas han cubierto el conjunto de funcionalidades de un sistema, proporcionando una indicación de qué tan bien se han verificado todas las partes del software.

**Tasa de errores detectados**

[20] afirma que la tasa de errores detectados es un indicador que refleja la cantidad de defectos o fallas identificados durante el proceso de pruebas en relación con el número total de defectos presentes en el software.

**Tiempo de ejecución de las pruebas**

El tiempo de ejecución de las pruebas se define como el tiempo que se toma para ejecutar un conjunto de pruebas sobre un sistema, lo cual es crucial para evaluar la eficiencia y rendimiento de las pruebas automatizadas, según [21].

## C. Indicadores

### Proporción de funcionalidades verificadas por las pruebas automatizadas:

[22] afirman que la proporción de funcionalidades verificadas por pruebas automatizadas es la medida en que las pruebas automatizadas cubren las funcionalidades del software, evaluando la efectividad de las pruebas en validar el comportamiento esperado del sistema.

### Número de errores identificados por las pruebas automatizadas.

De acuerdo con [23], el número de errores identificados por las pruebas automatizadas es una métrica crucial que indica la cantidad de fallos detectados por las pruebas automatizadas, lo cual es fundamental para evaluar la calidad del software y la efectividad del proceso de pruebas.

### Duración total para ejecutar todas las pruebas automatizadas.

El tiempo total requerido para completar todas las pruebas automatizadas de un sistema, lo cual es una métrica importante para evaluar la eficiencia del proceso de pruebas, afirma [24]

## D. Teorías

### Itellij Idea

Según [25], es un entorno de desarrollo integrado (IDE) diseñado para facilitar el desarrollo de sistemas informáticos. Esta herramienta se creó con el objetivo de aumentar la productividad del desarrollador. Su diseño ergonómico y su asistencia de codificación inteligente hacen que el desarrollo de software sea productivo y agradable. Este IDE está diseñado para trabajar con Java, pero puede trabajar con otros lenguajes de forma nativa o con varios plugins. Además, nos permite integrar varios marcos que podrían ser útiles.

### Selenium

Según [26], es una herramienta de código abierto destinada a automatizar aplicaciones basadas en navegadores web. Selenium es un conjunto de programas

de software que ayuda con la automatización de pruebas. Las pruebas se pueden crear o codificarlas en una variedad de lenguajes de programación populares como java, Python, groovy, etc, y la mayoría de los navegadores web actuales las pueden ejecutar.

Selenium funciona con muchos sistemas operativos para aplicaciones móviles, como iOS, Windows Mobile y Android, así como con Windows, Linux y Macintosh.

**Selenium Web Driver**

Además [26] indica que Selenium WebDriver es la herramienta más popular de la suite Selenium. Proporciona una interfaz de programación que permite interactuar con los elementos de una página web y simular las acciones del usuario, como hacer clic en botones, enviar formularios y navegar por páginas. Varios lenguajes de programación, como Java, Python, C# y Ruby, están soportados por WebDriver.

**Selenium IDE**

Según [26], Selenium IDE es una herramienta de código abierto para automatizar el desarrollo de pruebas que permite grabar, editar y depurar pruebas visualmente en un navegador web. Está disponible como una extensión para los navegadores Chrome y Firefox. Aunque no es tan adaptable como WebDriver, es útil para pruebas exploratorias y rápidas.

**Gherkin**

Según [27] este lenguaje es diseñado para hacer que la descripción del comportamiento del software sea fácil de entender tanto para desarrolladores como para personas no técnicas. Es utilizado principalmente en BDD (Desarrollo impulsado por el comportamiento) para escribir escenarios de prueba y especificaciones de comportamiento de manera estructurada y comprensible.

Las características principales de Gherkin incluyen:

**Sintaxis simple y legible**: Gherkin permite a los usuarios describir el

comportamiento del software de una manera comprensible y sin ambigüedades gracias a su sintaxis simple y basada en lenguaje natural.

**Estructura basada en palabras clave**: Gherkin se basa en palabras clave predefinidas, como "Feature", "Scenario", "Given", "When", "Then" y "And", que ayudan a estructurar los escenarios de prueba de manera coherente y fácilmente comprensible.

**Escenarios de prueba basados en ejemplos**: Gherkin permite la definición de escenarios de prueba utilizando ejemplos concretos que ilustran el comportamiento esperado del software en diferentes situaciones.

**Soporte para internacionalización**: Gherkin funciona con múltiples idiomas, permite que equipos distribuidores y multiculturales colaboren en la especificación y ejecución de pruebas.

Lenguaje Gherkin – ejemplo

```
# Comment
@tag
Feature: Eating too many cucumbers may not be good for you

  Eating too much of anything may not be good for you.

  Scenario: Eating a few is no problem
    Given Alice is hungry
    When she eats 3 cucumbers
    Then she will be full
```

**Ilustración 1 -** Lenguaje Gherkin

### 2.2.2 Variable Dependiente: Eficiencia en el Proceso de Pruebas

#### A. Definición

**Eficiencia en el Proceso de Pruebas:** *[28]*afirman que la eficiencia del proceso de pruebas se refiere a la capacidad de realizar pruebas de software de manera efectiva y con el mínimo uso de recursos, incluyendo tiempo, esfuerzo y costos. Una prueba eficiente no solo detecta un gran número de defectos, sino que también optimiza el uso de recursos, asegurando una alta calidad del software con el menor costo posible.

El proceso de pruebas automatizadas debe ser eficiente para garantizar que las pruebas sean efectivas, rápidas y rentables. Las siguientes prácticas pueden aumentar la eficiencia de este proceso:

**Planificación y Estrategia**: Comienza por establecer una estrategia de pruebas automatizadas clara y bien definida. Identifica qué pruebas deben automatizarse y cuáles pueden ser manuales. Prioriza las pruebas basadas en su criticidad y frecuencia de ejecución.

**Selección de Herramientas:** Elije las herramientas de automatización de pruebas adecuadas para tu proyecto y requisitos específicos. Asegúrate de que las herramientas seleccionadas sean compatibles con las tecnologías y plataformas que utilizas y que puedan integrarse fácilmente con tus herramientas de desarrollo y CI/CD.

**Diseño de Casos de Prueba:** Diseña casos de prueba claros, concisos y modulares que sean fáciles de mantener y ejecutar. Utiliza técnicas como el Page Object Model para estructurar tus pruebas de manera eficiente y promover la reutilización del código.

**Automatización Continua:** Integra la automatización de pruebas en tu proceso de integración continua (CI) para ejecutar pruebas automáticamente cada vez que se realizan cambios en el código. Esto ayuda a identificar problemas de manera temprana y garantiza una retroalimentación rápida sobre la calidad del software.

**Gestión de Ambientes de Pruebas:** Configura y gestiona ambientes de pruebas estables y reproducibles que reflejen con precisión los ambientes de producción, utiliza herramientas de virtualización y contenedores para facilitar la configuración y despliegue de ambientes de pruebas bajo demanda.

**Monitoreo y Reportes:** Implementa un sistema de monitoreo automatizado para supervisar la ejecución de pruebas y generar informes de resultados. Utiliza métricas como la cobertura de código y la tasa de éxito de las pruebas para evaluar la efectividad de tus pruebas automatizadas y guiar la mejora continua.

**Mantenimiento y Refactorización:** Dedica tiempo regular al mantenimiento y refactorización de tus pruebas automatizadas para asegurarte de que sigan siendo efectivas y relevantes a medida que evoluciona el software, elimina pruebas obsoletas o duplicadas y actualiza las pruebas existentes para reflejar cambios en los requisitos o en la aplicación.

**B. Dimensiones**

**Calidad del Software:** La calidad del software se refiere a la medida en que un software cumple con los requisitos funcionales y no funcionales especificados por el cliente, afirma [29].Esto incluye características como confiabilidad, mantenimiento, eficiencia y usabilidad del software.

**Productividad en el Proceso de Pruebas:** La productividad del proceso de pruebas se mide por la cantidad de pruebas que se pueden realizar en un período de tiempo determinado, así como por la eficacia con la que estas pruebas detectan defectos en el software, afirma [19].Un proceso de pruebas eficiente no solo encuentra muchos defectos, sino que también lo hace de manera eficiente, ahorrando tiempo y recursos.

**C. Indicadores**

**Número de errores detectados:** Según [28]**, e**s el número total de errores o fallas en el software detectados durante un ciclo de pruebas. La evaluación de la efectividad del proceso de pruebas y la calidad del software depende de este indicador.

**Tiempo de ejecución de las pruebas:** Según [24]**,** es la duración total que se requiere para completar un conjunto de pruebas. Para evaluar la eficiencia del proceso de pruebas y el impacto de la automatización en la reducción de tiempos, este indicador es esencial.

**Tiempo de respuesta del sistema:** Según [19]**,** es la duración promedio que toma el sistema para responder a las solicitudes de los usuarios. Este indicador es fundamental para evaluar la funcionalidad y la usabilidad del software.

## D. Teorías

### Metodología Scrum

La metodología Scrum es un método iterativo e incremental para la gestión y el desarrollo de proyectos. Es un pequeño marco que ayuda a las personas, grupos y empresas a generar valor a través de soluciones adaptables para problemas complejos. El equipo Scrum trabaja juntos y cada tarea se asigna a requisitos y tecnologías para entregar productos funcionando de manera incremental.

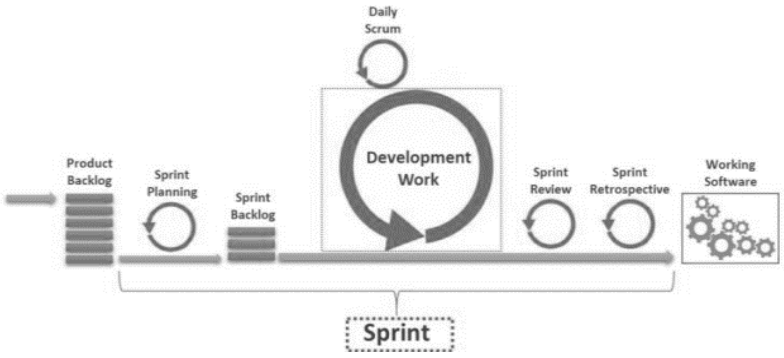



**Ilustración 2 -** Metodología scrum - Sprint

### Equipo Scrum

Según [31], Un equipo Scrum es la unidad fundamental de Scrum. El Scrum Master, el propietario del producto (Product Owner) y los desarrolladores conforman el equipo Scrum. Un equipo Scrum no tiene jerarquías ni subequipos. Es un grupo colaborativo de profesionales que se concentran en un solo objetivo, el objetivo del producto.

Roles de scrum

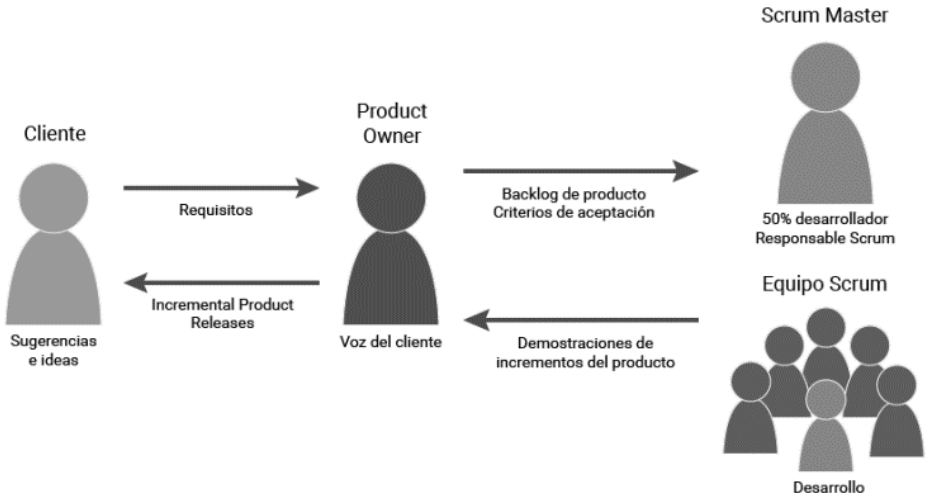



**Ilustración 3 -** Estudio de metodologías ágiles. Aplicación SCRUM

**Scrum Master**

Según [31], La responsabilidad de establecer Scrum de acuerdo con las especificaciones de la Guía Scrum recae en el Scrum Master. Ayuda a que todos en el equipo y en toda la organización comprendan la teoría y la práctica de Scrum.

El Scrum Master sirve al equipo de Scrum de varias maneras, incluyendo:

- Los miembros del equipo deben recibir capacitación en autogestión y multifuncionalidad;
- Ayudando al equipo Scrum a centrarse en la creación de incrementos de alto valor que cumplan con la definición de hecho;
- Ayudar a los equipos de Scrum a superar los obstáculos;
- Asegurarse de que todos los eventos de Scrum se lleven a cabo de manera positiva, productiva y respetando el cronograma.
- Liderar, capacitar y mentorizar a la organización mientras adopta Scrum;
- Asesorar y planificar la implementación de Scrum en la organización;
- Ayudar a las personas y a las partes interesadas a comprender y promulgar un enfoque empírico para el trabajo complejo; y eliminar obstáculos entre

las partes interesadas y los equipos de Scrum.

**Product Owner**

Aquí [31] afirma que está a cargo de maximizar el valor del producto que crea el equipo Scrum. La forma en que esto se hace puede variar según las organizaciones, los equipos Scrum y las personas.

El Propietario del Producto también es responsable de la gestión eficaz de la pila del producto (Product Backlog), que incluye:

- Desarrollar y comunicar el propósito del producto;
- Crear y comunicar de manera clara los elementos de trabajo pendientes; solicitar los elementos de trabajo pendientes;
- Garantizar que el trabajo pendiente sea visible, comprensible y transparente.

**Desarrolladores**

Según [31], desarrolladores forman parte del equipo Scrum y se encargan de desarrollar cualquier componente de un Incremento útil (funcional) en cada Sprint.
Según el dominio del trabajo, los desarrolladores suelen requerir una variedad de habilidades específicas. Sin embargo, los desarrolladores siempre están a cargo de:

- Crear un plan para el Sprint, el Sprint Backlog;
- Inculcar la calidad adhiriéndose a una definición de Hecho;
- Adaptar su plan cada día hacia el Objetivo Sprint;
- Responsabilizarse mutuamente como profesionales.

**Ceremonias de scrum**

**Sprint**

Es un período de tiempo predeterminado durante el cual se realizan esfuerzos concentrados y se produce un aumento potencialmente entregable del

producto. Los sprints suelen durar una a cuatro semanas, y al final de cada sprint, el equipo debe presentar un producto o mejora funcional al cliente o a los interesados.

Según [31] durante el sprint:

- No se hacen cambios que pongan en peligro el Objetivo Sprint;
- La calidad no disminuye;
- El trabajo pendiente del producto se refina según sea necesario;
- El alcance se puede clarificar y renegociar con el Propietario del Producto a medida que se aprende más.

**Planificación del Sprint**

Según [31], la Reunión de Planificación de Sprint de cada mes no puede durar más de ocho horas. Para sprints más cortos, el evento generalmente es más corto. El Scrum Master se encarga de asegurarse de que el evento se lleve a cabo y que los participantes entiendan su propósito. El Scrum Master enseña al equipo Scrum cómo mantenerse al día con las técnicas.

Paso a paso para la planificación de un sprint:

- Preparación
- Explicación del objetivo del sprint
- Selección de elementos del Product Backlog
- Estimación del trabajo
- Planificación del sprint:
- Definición de la definición de "Listo":
- Confirmación del compromiso del equipo:

**Dayli Scrum**

Según [31], El propósito del Daily Scrum es revisar el progreso hacia el objetivo del Sprint y ajustar el backlog del Sprint según sea necesario, ajustando el

próximo trabajo planeado.

El Daily Scrum es un evento para desarrolladores del equipo Scrum que dura 15 minutos o más. Se realiza al mismo tiempo y en el mismo lugar todos los días laborables del Sprint para reducir la complejidad. Si el Scrum Master o el propietario del producto están participando activamente en los elementos de trabajo pendientes de Sprint, participante como desarrolladores.

**Sprint Review**

El objetivo de la revisión de Sprint es inspeccionar el resultado de Sprint para determinar adaptaciones futuras. El equipo Scrum presenta a las partes interesadas principales los resultados de su trabajo y discute el progreso hacia el objetivo de producto. Se realiza una revisión del sprint.

**Sprint retrospective**

Según [32], Se realiza un balance general de la iteración alrededor de tres horas después de revisar cada Sprint. El equipo de desarrollo y el Scrum Máster analizan los resultados y sugieren mejoras para los próximos Sprint, asegurándose de que la implementación de Scrum sea lo más exitosa posible.

Con el Scrum Máster, el equipo de desarrollo se sentirá motivado y motivado para mejorar las tareas y los procesos de desarrollo para garantizar una mayor eficiencia y calidad de producción en los próximos Sprints.

**Artefactos**

Los artefactos de Scrum representan valor o trabajo. Están diseñados para brindar la mayor transparencia posible de los datos cruciales. En consecuencia, la base de adaptación es uniforme para todos los inspectores.

**Product Backlog**

Segun [31], dice que se lista con prioridad constante de todas las funcionalidades, mejoras y requisitos del producto. Es responsabilidad del

propietario del producto mantener el inventario de productos actualizado y priorizado en función del valor que aporta al cliente.

**Sprint Backlog**

Segun [31], afirma que lista de todas las historias de usuario y tareas seleccionadas del inventario de productos para completar durante el sprint actual. El equipo Scrum se compromete a completar el trabajo de inventario durante el sprint.

**Incremento**

Según [31] dentro de un Sprint, se pueden crear múltiples incrementos. La suma de los incrementos, que apoya el empirismo, se presenta en la Revisión Sprint. Esta suma incluye todas las historias de usuario completadas y cualquier trabajo adicional realizado durante el sprint actual. El aumento del producto debe ser potencialmente entregable al final de cada sprint y estar listo para su revisión.

## 2.3. Definición de términos básicos

### 2.3.1 BDD

Según [33], es el proceso de BDD mejora la comunicación entre los desarrolladores y los clientes y garantiza que todos los proyectos se centren en la entrega de lo que realmente necesita el cliente, cumpliendo con todos los requisitos, las pruebas automatizadas desempeñan un papel crucial. Se integran en escenarios de pruebas automatizadas, herramientas de pruebas automatizadas, ejecución de pruebas automatizadas, retroalimentación continua, son esenciales para BDD porque permiten la verificación automatizada del comportamiento del sistema descrito en los escenarios de prueba. Esto acelera el proceso de desarrollo, mejora la calidad del software y promueve una colaboración más efectiva entre los miembros del equipo.

### 2.3.2 Page Object Model

Según [34], este patrón representa cada una de las pantallas que componen el sitio web o la aplicación, también conocido como POM. Se utiliza con Selenium WebDriver y otras tecnologías para mejorar el mantenimiento de pruebas y reducir la duplicación de

código, utilizado en el desarrollo de pruebas automatizadas para estructurar y mantener el código de manera eficiente, fomentar la reutilización del código, separar la lógica de prueba de la lógica de interfaz de usuario y facilitar la mantenibilidad de las pruebas.

### 2.3.3 Pruebas Funcionales

Según [35], las pruebas funcionales se basan en evaluar si un componente o software satisface los requisitos funcionales especificados y que funcionen según lo esperado por los clientes, existen diversas pruebas como, test de humo, test de interfaces, test de regresión, test de aceptación, test de funcionalidad, test de conformidad.

### 2.3.4 Pruebas no Funcionales

Según [35], la prueba no funcional está basado a aspectos que no están relacionados directamente por el comportamiento funcional del software, sino más bien como características como la usabilidad, escalabilidad, la eficiencia del desempeño o la seguridad, en resumen, pruebas de rendimiento, pruebas de fiabilidad, pruebas de seguridad, pruebas de compatibilidad.

### 2.3.5 Selenium Web Driver

Según [36], es un framework de código abierto utilizado para automatizar aplicaciones web, permite a los equipos de prueba y desarrollo mejorar la eficiencia y la calidad de sus procesos. Su amplio soporte para varios lenguajes de programación y su capacidad para interactuar directamente con el navegador lo convierten en una opción popular para la automatización de pruebas web, se prueba en varios lenguajes más populares incluidos como java, C#, Groovy, Perl, PHP, Python, Rubí y Scala.

### 2.3.6 Cucumber

Según [37], es una herramienta poderosa para escribir y ejecutar pruebas automatizadas en un formato legible por humanos, facilita la colaboración entre desarrolladores, probadores y partes interesadas del proyecto. Su enfoque en el BDD ayuda a asegurarse de que las pruebas estén alineadas con los requisitos del negocio y el comportamiento esperado del sistema. El lenguaje común con el que Cucumber especifica las pruebas se denomina Gherkin.

### 2.3.7 Gherkin

Según [27], es un lenguaje flexible y fácil de entender que ayuda a definir y comunicar el comportamiento del software de manera efectiva y colaborativa. Esto lo convierte en una herramienta útil para la práctica de BDD.

### 2.3.8 Junit

Según [38], es un framework basado en pruebas y utiliza anotaciones para identificar pruebas, que están escritas como métodos. Es parte de la familia de pruebas unitarias xUnit. Es un framework básico para la creación y ejecución de pruebas automatizadas. Celebra que los programadores prueben su propio software como una medida política. JUnit es ampliamente utilizado para crear una colección de pruebas unitarias que pueden ejecutarse automáticamente cuando se modifica el software, los podemos utilizar en pruebas unitarias con Junit, Integración con herramientas de automatización de prueba, Ejecución de pruebas automatizadas, Generación de informes y retroalimentación. Esencial para la automatización de pruebas en el ecosistema Java, y se puede utilizar en conjunto con otras herramientas y marcos de trabajo para automatizar pruebas en varios niveles, desde pruebas unitarias hasta pruebas de integración y pruebas de sistema.

### 2.3.9 Java

Lenguaje de programación orientado a objetos de propósito general con dependencias de implementación reducidas. La portabilidad de Java significa que el código puede ejecutarse en cualquier máquina virtual. Según [39] es uno de los lenguajes de programación más importantes de hoy en día donde todo programa Java debe compilarse y una máquina virtual debe interpretar el código generado. Independientemente de la plataforma en la que se haya realizado el proceso de compilación, el código compilado se puede ejecutar en máquinas virtuales. Es un sistema capaz de procesar cantidades significativas de datos. Varios dispositivos reciben datos, órdenes y programas como entrada y producen un resultado como salida.

### 2.3.10 Scrum

"Scrum es un marco ágil que permite a los equipos trabajar de manera colaborativa

para desarrollar productos complejos", [40] afirma este marco se basa en iteraciones cortas y permite una mejora continua a través de inspecciones y ajustes constantes.

# Capítulo III: Hipótesis y Variables

## 3.1. Hipótesis

### 3.1.1. Hipótesis general

La implementación de un Marco de Pruebas Automatizadas mejorará significativamente la eficacia del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

### 3.1.2. Hipótesis específicas

**HE1:** La implementación de un Marco de Pruebas Automatizadas mejorará significativamente la calidad del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

**HE2:** La implementación de un Marco de Pruebas Automatizadas incrementará significativamente la productividad en el proceso de pruebas del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

## 3.2. Definición conceptual de las variables

### 3.2.1. Variable Independiente: Pruebas Automatizadas

Según [41] las pruebas automatizadas son un conjunto de procedimientos y técnicas que utilizan herramientas y scripts para ejecutar automáticamente casos de prueba en un software o sistema, sin intervención humana constante, en lugar de realizar pruebas manuales, que son propensas a errores y consumen mucho tiempo, las pruebas automatizadas permiten verificar la funcionalidad, rendimiento y seguridad de un sistema de manera más eficiente y repetible, siendo esenciales para asegurar la calidad y confiabilidad del software, especialmente en proyectos complejos o de gran escala.

### 3.2.2. Variable Dependiente: Eficiencia en el Proceso de Pruebas

La eficiencia en las pruebas se refiere a la capacidad de realizar pruebas de software efectiva y con el uso óptimo de recursos, incluyendo tiempo, esfuerzo y costos, una prueba eficiente maximiza la cobertura de casos de pruebas relevantes y minimiza el tiempo y los recursos necesarios para ejecutarlas [42].

## 3.3. Operacionalización de las variables (Definición operacional)

### 3.3.1. Variable Independiente: Pruebas Automatizadas

| Variable | Dimensiones | Indicador | Definición | Instrumento | Escala | Fórmula / Método de Cálculo |
|---|---|---|---|---|---|---|
| Pruebas Automatizadas | Cobertura de pruebas [19] | Proporción de funcionalidades verificadas por las pruebas automatizadas [22] | Es la medida en que las pruebas automatizadas cubren las funcionalidades del software, evaluando la efectividad de las pruebas en validar el comportamiento esperado del sistema. [22] | Herramientas de automatización (Selenium, Cucumber) | De razón | $CP = \frac{Fverificadas}{Ftotales} \times 100$<br>Donde:<br>**Fverificadas:** Número de funcionalidades verificadas por las pruebas automatizadas.<br>**Ftotales:** Número total de funcionalidades del sistema.<br>**CP:** Porcentaje de cobertura de pruebas, indicando qué proporción de las funcionalidades del sistema han sido verificadas por las pruebas automatizadas. |
| | Tasa de errores detectados [20] | Número de errores identificados por las pruebas automatizadas [23] | Es una métrica crucial que indica la cantidad de fallos detectados por las pruebas automatizadas, lo cual es fundamental para evaluar la calidad del software y la efectividad del proceso de pruebas. [23] | Herramientas de reporte de pruebas | De razón | $TE = \frac{E\,\mathrm{det}\,e\,ctados}{Etotales} \times 100$<br>Donde:<br>**Edetectados:** Número de errores identificados por las pruebas automatizadas.<br>**Etotales:** Número total de errores presentes en el sistema.<br>**TE:** Porcentaje de errores detectados, mostrando la eficacia de las pruebas automatizadas para encontrar errores en el sistema. |
| | Tiempo de ejecución de las pruebas [21] | Duración total para ejecutar todas las pruebas automatizadas [24] | Tiempo total en horas o minutos para ejecutar todas las pruebas automatizadas [24] | Herramientas de medición de tiempo | De razón | Tiempo total en horas o minutos |

### 3.3.2. Variable Dependiente: Eficiencia en el Proceso de Pruebas

| Variable | Dimensiones | Indicador | Definición | Instrumento | Escala | Fórmula / Método de Cálculo |
|---|---|---|---|---|---|---|
| Eficiencia en el Proceso de Pruebas | Calidad del Software [29] | Número de errores detectados [28] | Se refiere al número total de errores o fallas en el software detectados durante un ciclo de pruebas. [28] | Ficha de registro | De razón | Suma de todos los errores encontrados durante las pruebas. |
| | Productividad en el Proceso de Pruebas [19] | Tiempo de ejecución de las pruebas [24] | Es la duración total que se requiere para completar un conjunto de pruebas. [24] | Ficha de registro | De razón | Tiempo total transcurrido desde el inicio hasta el final de la ejecución de todas las pruebas. |
| | Rendimiento del Software [30] | Tiempo de respuesta del sistema [19] | Es la duración promedio que toma el sistema para responder a las solicitudes de los usuarios. [19] | Ficha de registro | De razón | Promedio de los tiempos de respuesta para una serie de solicitudes. |

# Capítulo IV: Metodología

## 4.1. Enfoque de la investigación

El proyecto se llevará a cabo empleando un enfoque cuantitativo que implica la recolección y análisis de datos numéricos para su procesamiento estadístico [43]. Empleando un enfoque cuantitativo, ya que implica la implementación y el análisis de un marco de pruebas automatizadas basado en la metodología (Behavior Driven Development) y la evaluación de su efectividad mediante los datos numéricos obtenidos a través de pruebas y mediciones en las que se llevará a cabo una comparación entre pruebas manuales y automatizadas.

## 4.2. Tipo de investigación

La investigación será de tipo aplicada, [44] afirma que a este tipo de estudio se le conoce también como activa o dinámica, ya que depende de sus hallazgos y contribuciones teóricas para comparar la teórica con la realidad; asimismo, se enfoca en problemas específicos en condiciones y características concretas, centrándose en su aplicación inmediata. El proyecto será aplicado porque se enfoca en la implementación práctica de un marco de pruebas automatizadas para el software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete, este tipo de estudio busca resolver problemas específicos y mejorar procesos concretos en el entorno real, utilizando el Page Object Model y Behavior Driven Development, buscando lograr resultados aplicables y útiles para la investigación.

## 4.3. Nivel de investigación

[44] afirma que, en el nivel aplicativo, es necesario evaluar el éxito de la solución del problema, intervienen la población o unidades de estudio para obtener resultados positivos y mejorarlo. Nuestro proyecto presenta un nivel aplicativo debido a que se centra en la implementación práctica de un marco de pruebas automatizadas para el sofware de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete, en este nivel se lleva a cabo la intervención directa en el proceso de investigación, empleando los conocimientos teóricos y metodológicos de manera práctica buscando resolver problemas concretos en el entorno; además se busca evaluar el éxito de las pruebas, mediante la eficacia y eficiencia del framework, para lograr mejorar tangibles y transformaciones positivas en la gestión de los procesos de investigación en la Universidad Nacional de

Cañete.

### 4.4. Método de investigación

El método que se elegirá será el hipotético- deductivo, el cual se caracteriza por la formulación de hipótesis seguida de la deducción de sus consecuencias, además se apoya de la observación de hechos empíricas, operacionalización de variables y evolución de hipótesis para verificar su validez, lo que facilita abordar en las conclusiones de la investigación [45]. Nuestro trabajo presentará el método hipotético- deductivo, ya que nos va a permitir formular hipótesis sobre la eficacia y eficiencia del diseño y metodología ágil, para luego deducir consecuencias a partir de la observación de los datos empíricos recopilados durante la implementación, de esta manera, el método nos va a permitir verificar las hipótesis planteadas y fundamentar las conclusiones en base a la evidencia concreta.

### 4.5. Diseño de investigación

Según [43], un diseño preexperimental se distingue por no tener un control estricto sobre las variables externas y no realizar una manipulación exhaustiva de la variable independiente. Las etapas exploratorias de la investigación, donde el objetivo es obtener una primera aproximación a la relación entre variables, utilizan principalmente este tipo de diseño. El proyecto será de diseño preexperimental será de diseño preexperimental porque se centrará en evaluar el impacto de la implementación de pruebas automatizadas en la eficiencia del proceso de pruebas, utilizando un solo grupo de estudio con mediciones antes y después de la intervención, sin control riguroso sobre otras variables, lo que permite obtener una primera aproximación al efecto de las pruebas automatizadas, aunque con limitaciones en la generalización de los resultados.

### 4.6. Población, muestra y muestreo

#### 4.6.1. Población

[46] afirma que la población se refiere al conjunto completo de elementos o individuos que poseen las características que se estudian, puede ser cualquier grupo definido, ya sea personas, objetos, eventos, etc., que se considere relevante para el estudio. En el presente proyecto, la población estará conformada por todas las funcionalidades (registros, etiquetas, descargar de archivos, creación de proyectos, etc.) del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional

de Cañete.

### 4.6.2. Muestra

La muestra es un subconjunto de la población seleccionado para participar en un estudio con el propósito de hacer inferencias sobre la población total a partir de los resultados obtenidos en el análisis de este subconjunto de la población, según [46]. Para el presente trabajo, se considerará treinta (30) funcionalidades críticas seleccionadas, que están presentes en el módulo de registro de investigaciones, el módulo de seguimiento de proyectos y el módulo de generación de reportes, entre otros.

### 4.6.3. Muestreo

[47] definen el muestreo como el proceso mediante el cual se selecciona un subconjunto de personas o elementos de una población más grande para hacer inferencias sobre esa población. Para asegurarse de que los hallazgos del estudio sean representativos y generalizables, este proceso es crucial. En la presente investigación, el muestreo será no probabilístico por criterio, eligiendo aquellas funcionalidades que son esenciales para el funcionamiento del sistema y que representan adecuadamente la diversidad de tareas que realiza el software.

## 4.7. Técnicas e instrumentos de recolección de datos

### 4.7.1. Técnica

[48] definen una técnica de investigación como un conjunto de procedimientos y métodos específicos que se utilizan para la recolección y análisis de datos en una investigación, que permiten al investigador obtener la información necesaria para resolver las preguntas de investigación. La técnica a utilizar en este proyecto será la observación estructurada.

### 4.7.2. Instrumento

[49] definen un instrumento de investigación como un recurso que el investigador utiliza para recopilar datos, tales como encuestas, cuestionarios, entrevistas y observaciones estructuradas, diseñados específicamente para medir las variables de interés del estudio. El instrumento que se utilizará en la presente investigación serán las fichas de registros.

## 4.8. Técnicas estadísticas de análisis de datos

Para llevar a cabo el análisis de datos, se utilizará fichas de registros para recopilar información, que se organizará en hojas de cálculo de Excel 2021. A continuación, para obtener una visión general de los resultados, se generarán estadísticas descriptivas utilizando los datos organizados. La prueba de hipótesis se llevará a cabo posteriormente con los datos de las pruebas previas y posteriores utilizando el programa estadístico SPSS IBM 24. Además, se utilizarán estadísticas descriptivas para analizar los datos y los procedimientos. Finalmente, se prepararán los resultados inferenciales utilizando el sistema estadístico y las hojas de Excel.

## 4.9. Consideraciones éticas

Es esencial tener en cuenta varias consideraciones éticas al crear un marco de pruebas automatizadas para el software de Gestión Digital de Procesos de Investigación en la Universidad Nacional de Cañete. Proteger la confidencialidad de los datos utilizados en las pruebas, tanto la información confidencial de los proyectos de investigación como la información personal de los investigadores, es crucial. Además, es importante que las pruebas automatizadas no interfieran con el uso cotidiano del sistema por parte de los usuarios, minimizando cualquier interrupción en su trabajo. A su vez, es necesario obtener el consentimiento informado de todos los participantes en las pruebas y explicarles claramente los objetivos, procedimientos y posibles efectos del estudio. Finalmente, cumplir con todas las normas y políticas de la universidad, así como con las leyes y reglamentos aplicables sobre protección de datos y privacidad.

# Capítulo V: Resultados

## 5.1. Resultados Descriptivos

Las Tablas 1, 2, 3 y las Figuras 1, 2 y 3 muestran los datos de este estudio.

### 5.1.1. Evaluación descriptiva del indicador 1: Número de errores detectados (NED)

**Tabla I - Evaluación descriptiva del indicador NED**

| Estadísticos descriptivos | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **N** | **Mínimo** | **Máximo** | **Media** | **Desviación estándar** | **Varianza** | **Coeficiente de Variabilidad (DE/M)*100** |
| **Errores detectados Pre test** | 30 | 1 | 2 | 1,33 | 0,479 | 0,230 | 63,70% |
| **Errores detectados Post test** | 30 | 1 | 3 | 2,67 | 0,547 | 0,299 | 20.48% |
| **N válido (por lista)** | 30 | | | | | | |

Esta tabla I, dado su media, desviación estándar, varianza, valor mínimo, valor máximo y coeficiente de variabilidad, muestra el análisis descriptivo del indicador NED en el pre test y post test.

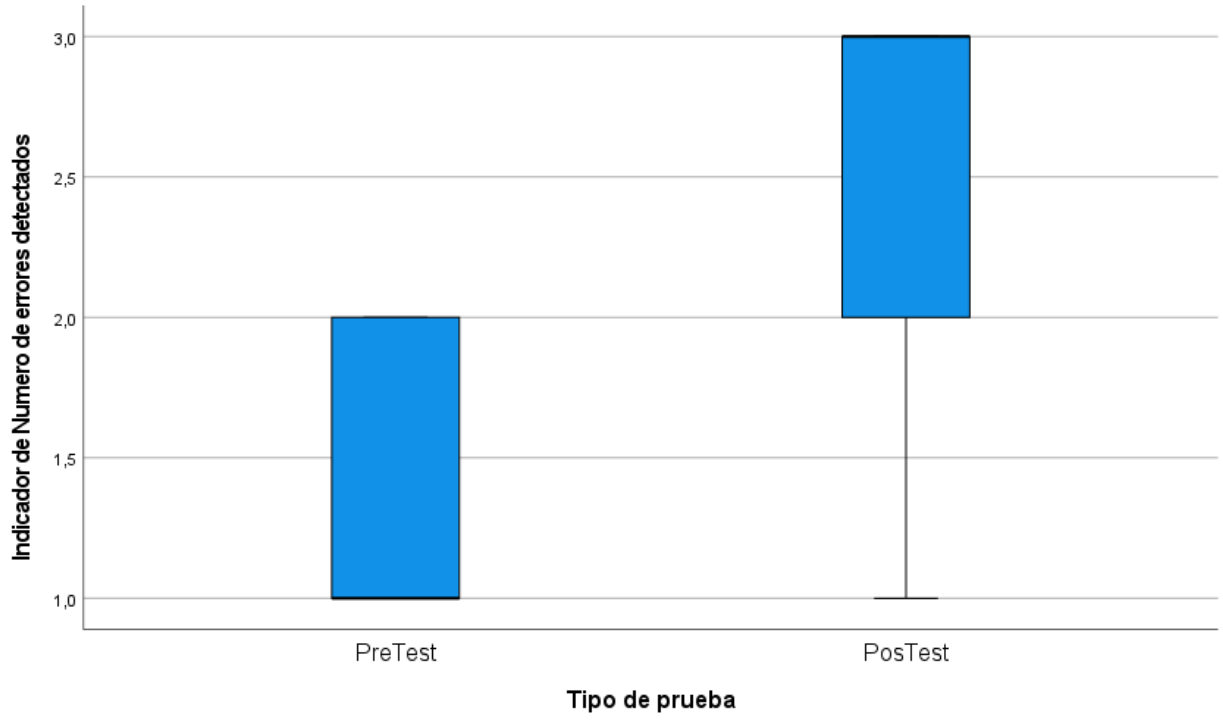



**Figura 1 -** Diagrama de Barras del indicador NED

La tabla I muestra que el Pre-Test del indicador NED obtuvo una media del 1,33%, mientras que el Post-Test obtuvo un porcentaje del 2,67%.

Por otra parte, en la figura 1, el diagrama de barras se muestra que existe diferencias entre el Pre-Test y Post-Test para el indicador Número de errores detectados, indicando que con la automatización (Post Test) se detecta más Número de errores que la ejecución de pruebas manuales (Pre Test). Esto demuestra una gran diferencia entre el tiempo previo y posterior a la implementación de la automatización.

### 5.1.2. Evaluación descriptiva del indicador 2: Tiempo de ejecución de las pruebas (TEP).

**Tabla II -Evaluación descriptiva del indicador TEP**

| Estadísticos descriptivos | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **N** | **Mínimo** | **Máximo** | **Media** | **Desviación estándar** | **Varianza** | **Coeficiente de Variabilidad (DE/M)*100** |
| **Tiempo de ejecución de pruebas Pre test** | 30 | 1,37 | 4,28 | 2,39 | 0,75635 | 0,572 | 31,65% |
| **Tiempo de ejecución de pruebas Post test** | 30 | 0,30 | 3,21 | 1,21 | 0,75705 | 0,573 | 62,61% |
| **N válido (por lista)** | 30 | | | | | | |

Esta tabla II, dado su media, desviación estándar, varianza, valor mínimo, valor máximo y coeficiente de variabilidad, muestra el análisis descriptivo del indicador TEP en el pre test y post test.

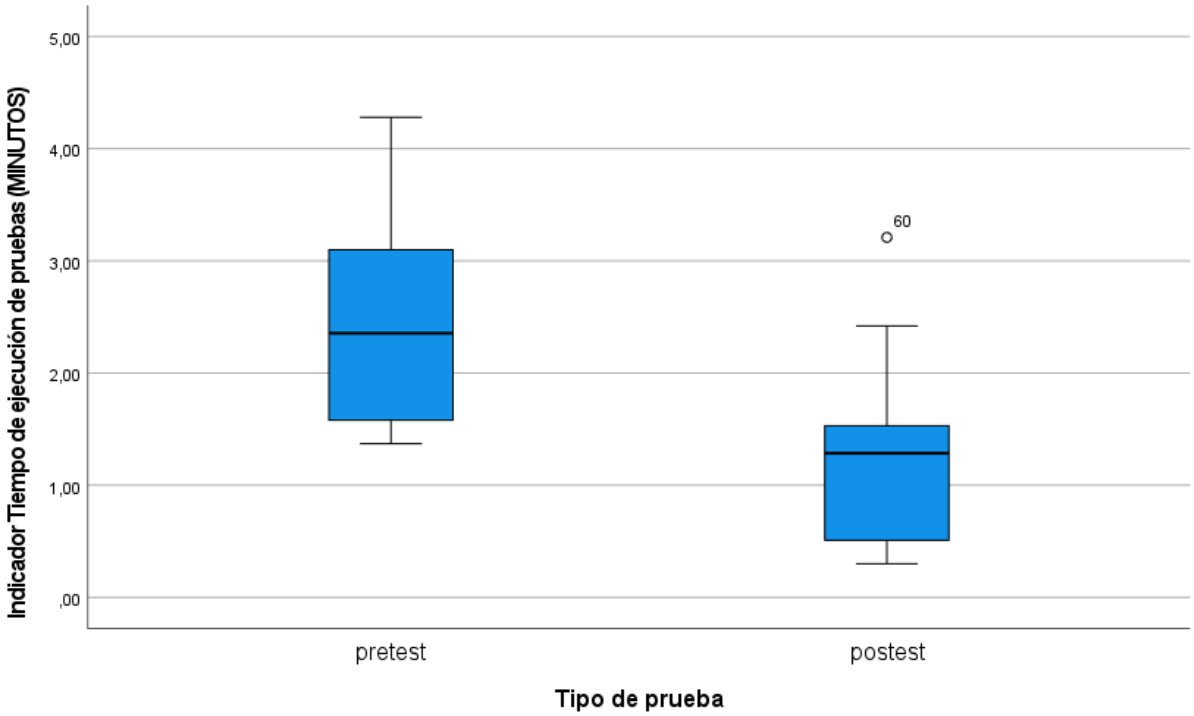



**Figura 2 -** Diagrama de Caja de Bigotes del indicador TEP

Tal como se observa en la tabla II, muestra que el Pre-Test del indicador TEP obtuvo una media del 2,39 %, mientras que el Post-Test obtuvo un porcentaje del 1,21%.

Por otra parte, en la figura 2 se observa, el diagrama de caja de bigotes, se muestra la diferencia entre el Pre-Test y Post-Test, indicando que con la automatización (Post Test), el tiempo de ejecución de la prueba es más rápido que las pruebas manuales (Pre Test). Esto demuestra una gran diferencia entre el tiempo previo y posterior a la implementación de la automatización.

## 5.2. Análisis inferencial

### 5.2.1. Pruebas de normalidad

Para determinar si un conjunto de datos sigue una distribución normal, se utiliza una prueba de normalidad. [50] afirma que la prueba de normalidad permite evaluar la adecuación de una muestra a la distribución normal, lo cual es esencial para la aplicación adecuada de varios métodos estadísticos que asumen esta distribución. En términos prácticos, estas pruebas comparan la distribución de los datos observados con una distribución normal teórica y evalúan las discrepancias entre las dos. Esto proporciona un marco para determinar si se pueden realizar inferencias estadísticas basadas en la

normalidad de los datos.

Para las evaluaciones posteriores, se estableció un nivel de significancia de 0.05 y una confianza previamente establecida del 95%. Los resultados de la significancia calculados en la columna Sig tanto del análisis Pre-Test como del análisis Post-Test se tomarán en cuenta para la toma de decisiones. Para los criterios de decisión se considerará que si Sig. < 0.05 se rechaza la Ho y se acepta la H1, y si Sig >= 0.05 se acepta la Ho y se rechaza la H1.

**Prueba de normalidad del indicador 1: Número de errores detectados (NED).**

- Ho: El indicador NED tiene datos con distribución normal.
- H1: El indicador NED no tiene datos con distribución normal.

**Tabla III - Test de normalidad del indicador NED**

| | Tipo de prueba | Shapiro-Wilk | | |
|---|---|---|---|---|
| | | Estadístico | gl | Sig. |
| Indicador de Numero de errores detectados | Pre Test | ,597 | 30 | ,001 |
| | Post Test | ,623 | 30 | ,001 |

En la tabla III, se observa que el indicador Número de errores detectados para el Pre-Test tiene una Sig=0,001<0,05 y en el Post-Test tiene una Sig=0,001<0,05, por lo tanto, la distribución es no normal. Entonces se rechaza la hipótesis nula y se acepta la hipótesis alternante. Por lo tanto, se va a utilizar procedimientos de la estadística no paramétricas con U de Mann-Whitney.

**Tabla IV - Prueba de Confiabilidad del indicador NED**

| Correlaciones | | | | |
|---|---|---|---|---|
| | | | VAR00001 | VAR00002 |
| Rho de Spearman | VAR00001 | Coeficiente de correlación | 1,000 | ,913* |
| | | Sig. (bilateral) | . | ,030 |
| | | N | 5 | 5 |
| | VAR00002 | Coeficiente de correlación | ,913* | 1,000 |
| | | Sig. (bilateral) | ,030 | . |
| | | N | 5 | 5 |

En la tabla IV, de acuerdo con la prueba de test y retest, utilizando la correlación de rho de Spearman, que arrogo un valor de 0,913 es una correlación positiva muy fuerte [51].

**Prueba de normalidad del indicador 2: Tiempo de ejecución de las pruebas (TEP) (Minutos).**

- Ho: El indicador TEP tiene datos con distribución normal.
- H1: El indicador TEP no tiene datos con distribución normal.

**Tabla V - Test de normalidad del indicador TEP**

| | Tipo de prueba | Shapiro-Wilk | | |
|---|---|---|---|---|
| | | Estadístico | gl | Sig. |
| Indicador Tiempo de ejecución de pruebas (MINUTOS) | Pre Test | ,939 | 30 | ,085 |
| | Post Test | ,903 | 30 | ,010 |

En la tabla V, se observa que el indicador Tiempo de ejecución de las pruebas para el Pre-Test tiene una Sig=0,85>0,05 por lo tanto tiene una distribución normal y en el Post-Test el Sig=0,010<0,05 por lo tanto tiene una distribución no normal. Para poder definir qué tipo de distribución prevalece, se va a realizar una combinación lineal de ambas, a través de una diferencia.

**Tabla VI - Combinación lineal - Test de normalidad del indicador TEP**

| | Shapiro-Wilk | | |
|---|---|---|---|
| | Estadístico | gl | Sig. |
| Diferencia | ,561 | 30 | ,001 |

En la tabla VI, la combinación lineal de la diferencia para ambas distribuciones resulto el Sig=0,001<0,05, por lo tanto, la distribución es no normal. Entonces se rechaza la hipótesis nula y se acepta la hipótesis alternante. Por lo tanto, se va a utilizar procedimientos de la estadística no paramétricas con U de Mann-Whitney.

**Tabla VII - Prueba de Confiabilidad del indicador TEP**

| Correlaciones | | | | |
|---|---|---|---|---|
| | | | PreTestD2 | PostTestD2 |
| Rho de Spearman | PreTestD2 | Coeficiente de correlación | 1,000 | ,997** |
| | | Sig. (bilateral) | . | ,000 |
| | | N | 30 | 30 |
| | PostTestD2 | Coeficiente de correlación | ,997** | 1,000 |
| | | Sig. (bilateral) | ,000 | . |
| | | N | 30 | 30 |

En la tabla VII, de acuerdo con la prueba de test y retest, utilizando la correlación de rho de Spearman, que arrogo un valor de 0,997 es una correlación positiva muy fuerte [51].

### 5.3. Prueba de Hipótesis

La prueba de hipótesis es un método estadístico para determinar una población a partir de una muestra. La prueba de hipótesis implica formular una hipótesis nula y una alternativa, luego usar datos muestrales para determinar si la hipótesis nula se rechaza o no, según [51].Este método permite evaluar la evidencia de los datos y hacer inferencias sobre la población, ayudando a determinar si existe una diferencia significativa o efecto en un contexto específico.

**Prueba de hipótesis con el indicador 1: Número de errores detectados (NED).**

- Ho: La implementación de un Marco de Pruebas Automatizadas no mejorará significativamente la calidad del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.
- H1: La implementación de un Marco de Pruebas Automatizadas mejorará significativamente la calidad del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

Nivel significación =0.05

Criterio de prueba = si el Sig<0.05 se rechaza la hipótesis nula y se acepta la alternante.

**Tabla VIII - Prueba de U de Mann-Whitney del indicador NED**

| Estadísticos de prueba[a] | |
|---|---|
| | Indicador de Numero de errores detectados |
| U de Mann-Whitney | 60,000 |
| W de Wilcoxon | 525,000 |
| Z | -6,120 |
| Sig. asin. (bilateral) | ,000 |
| a. Variable de agrupación: Tipo de prueba | |

En la tabla VIII, muestra que se rechaza la hipótesis nula y se acepta la hipótesis alternante, ya que se utilizó la prueba de U de Mann-Whitney donde hubo una Sig=0,000<0,05. Se

concluye que la implementación de un Marco de Pruebas Automatizadas mejorará significativamente la calidad del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

**Prueba de hipótesis con el indicador 2: Tiempo de ejecución de las pruebas (TEP)**

- Ho: La implementación de un Marco de Pruebas Automatizadas no incrementará significativamente la productividad en el proceso de pruebas del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.
- H1: La implementación de un Marco de Pruebas Automatizadas incrementará significativamente la productividad en el proceso de pruebas del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

Nivel significación =0.05

Criterio de prueba = si el Sig<0.05 se rechaza la hipótesis nula y se acepta la alternante.

**Tabla IX - Prueba de U de Mann-Whitney del indicador TEP**

| **Estadísticos de prueba[a]** | |
|---|---|
| | Indicador Tiempo de ejecución de pruebas (MINUTOS) |
| U de Mann-Whitney | 127,000 |
| W de Wilcoxon | 592,000 |
| Z | -4,778 |
| Sig. asin. (bilateral) | ,000 |
| a. Variable de agrupación: Tipo de prueba | |

En la tabla IX, muestra que se rechaza la hipótesis nula y se acepta la hipótesis alternante, ya que se utilizó la prueba de U de Mann-Whitney donde hubo una Sig=0,000<0,05. Se concluye que la implementación de un Marco de Pruebas Automatizadas incrementará significativamente la productividad en el proceso de pruebas del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

## Capítulo VI: Discusión

Este proyecto de automatización web se centró en implementar pruebas automatizadas para mejorar la eficiencia y la calidad del proceso de pruebas en el software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete. Durante el estudio, se compararon los resultados antes y después de la implementación de la automatización en dos indicadores clave: NED (Número de Errores Detectados) y TEP (Tiempo de Ejecución de Pruebas)

**Respecto al indicador 1: NED.**

De acuerdo con los hallazgos destacados de este estudio, en la Pre-Test del indicador NED, se obtuvo un promedio de 1.33%, mientras que los resultados después de la implementación de la automatización web fueron promedio de 2.67%. Estos resultados indican que la automatización web puede encontrar mayor número de errores con un promedio de 1,34% en el indicador NED.

Estos resultados obtenidos demostraron el aumento del 1.34% sugiere que la automatización web ha sido efectiva en la detección de un mayor número de errores en comparación con las pruebas manuales previas. Este hallazgo resalta la capacidad de la automatización para mejorar la calidad del proceso de pruebas al identificar más fallos potenciales, lo cual es crucial para garantizar la fiabilidad y la integridad del sistema evaluado.

Por otro lado, en el apartado de análisis inferencial del indicador NED, de acuerdo con las pruebas de normalidad Shapiro-Wilk, se concluyó que este indicador lleva una distribución no normal, por esta razón se realizó la prueba U de Mann-Whitney para la contratación de hipótesis **(Tabla VIII)**, en ella se obtuvo un de Sig=0,000<0,05, por ende se rechaza la hipótesis nula y se acepta la hipótesis alternante, dejando en evidencia que la implementación de un Marco de Pruebas Automatizadas mejorará significativamente la calidad del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

**Respecto al indicador 2: TEP.**

Por otro lado, antes de la automatización web, el indicador TEP mostraba un promedio del 2.39% durante el Pre-Test. Posteriormente, con la automatización implementada, este promedio disminuyó significativamente a 1.21%. Esta reducción del 1.18% indica que la automatización web ha permitido ejecutar los casos de prueba con mayor rapidez y eficiencia en comparación con los métodos manuales previos. La capacidad de la automatización para acelerar los procesos de

prueba (TEP) resalta su eficacia para optimizar el tiempo y los recursos invertidos en la verificación del sistema.

Por otro lado, en el apartado de análisis inferencial del indicador TEP, de acuerdo con las pruebas de normalidad Shapiro-Wilk, se utilizó la combinación lineal de la diferencia para ambas distribuciones resulto el Sig=0,001<0,05, por lo tanto, la distribución es no normal, por esta razón se realizó la prueba U de Mann-Whitney para la contratación de hipótesis **(Tabla IX)**, en ella se obtuvo una Sig=0,000<0,05, por ende se rechaza la hipótesis nula y se acepta la hipótesis alternante, dejando en evidencia que la implementación de un Marco de Pruebas Automatizadas incrementará significativamente la productividad en el proceso de pruebas del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

## Capítulo VII: Conclusiones

Se concluye que la automatización web ha demostrado ser un método efectivo para mejorar varios aspectos del software de gestión digital de procesos de investigación.

Primero, los resultados mostraron un aumento significativo en la eficiencia en la detección de errores (indicador NED). Antes de la automatización, el NED promedio era del 1,33%, sin embargo, el promedio aumentó a 2.67% después de que se implementó la automatización web. Esto indica que se detectó un mayor número de errores gracias a la automatización, con un aumento promedio del 1,34%.

En segundo lugar, los resultados también fueron positivos con relación al tiempo de ejecución de pruebas (indicador TEP). En la prueba previa, el TEP promedio fue de 2.39%, pero después de la automatización, se redujo a 1.21%. La automatización web acelera la ejecución de casos de prueba, como lo demuestra esta disminución del 1,18%.

En resumen, los hallazgos de esta investigación indican que el uso del lenguaje Gherkin y Selenium WebDriver en la automatización web mejora la eficiencia en la detección de errores y la rapidez en la ejecución de pruebas, estos resultados respaldan el uso de técnicas de automatización para mejorar la eficiencia y la calidad de los procesos de prueba de software.

## Capítulo VIII: Recomendaciones

En primer lugar, se recomienda que las organizaciones y equipos de desarrollo de software utilicen herramientas como Selenium WebDriver para implementar la automatización web. Los resultados han demostrado que esta práctica mejora significativamente la detección de errores, lo cual es fundamental para mantener la calidad del software.

En segundo lugar, es recomendable que los desarrolladores y los testers reciban capacitación sobre el uso del lenguaje Gherkin. Esta herramienta facilita la comunicación entre los diferentes miembros del equipo y mejora la eficacia de las pruebas automatizadas al proporcionar una descripción clara y concisa de los escenarios de prueba.

Se recomienda también la implementación de pruebas automatizadas desde los primeros pasos del desarrollo del software. Esta práctica puede reducir costos y tiempo de desarrollo al ayudar a identificar y corregir errores tempranamente. Por lo tanto, se aconseja que la automatización se integre en el ciclo de desarrollo continuo.

Es esencial establecer un sistema de seguimiento continuo para evaluar el rendimiento de las pruebas automatizadas. Regularmente revisar y modificar los scripts de prueba ayudará a mantener y mejorar la eficacia de la detección de errores y la rapidez de ejecución.

Ampliar el alcance de las pruebas automatizadas para incluir más casos y escenarios posibles también es recomendable. Esto incluye pruebas de rendimiento, seguridad y usabilidad, que brindan una evaluación completa del software.

Por último, pero no menos importante, es esencial estar al día con las nuevas tecnologías y herramientas de automatización. Se puede mejorar aún más la calidad y eficiencia del proceso de pruebas evaluando periódicamente las nuevas opciones disponibles en el mercado. Las instituciones y equipos de desarrollo de software pueden garantizar un software de mayor calidad y rendimiento implementando estas recomendaciones.

# Referencias


[1] R. Cubas Montenegro, «Testing y Calidad de Software. Automatización de pruebas con Selenium WebDriver,» Madrid, 2019.

[2] E. Liza Silva, «Implementación de automatización aplicada a pruebas de software del área de calidad de la empresa Entrega Creativa,» Lima, 2022.

[3] H.-S. y. Mendoza, «Metodología de la investigación. Las rutas cuantitativa, cualitativa y mixta 6th ed,» McGraw-Hill Education, Ciudad de México, 2018.

[4] R. F. y. P. B. J. Hernández, «Metodología de la investigación 6th ed,» McGraw-Hill, Ciudad de México, 2014.

[5] A. Musliu y X. Jashari, «Software Automated Testing using BDD,» pristina, Kosovo, 2021.

[6] A. C. Barus, «The implementation of ATDD and BDD from,» 2019.

[7] R. Segovia Gonzalez, D. K. Urribarri y M. L. Larrea, «Automation Tools for Web Testing. Beyond,» ASSE, Argentine Symposium on Software Engineering, Argentina, 2022.

[8] N. Mittal, «Automation Testing Using Cucumber and Selenium WebDriver,» BioGecko, Nagpur, Maharastra, 2023.

[9] W. Morales, «Impacto de la automatización de pruebas en la eficiencia de pruebas de software en una consultora de TI,» UCV, Peru, 2022.

[10] E. V. R. Liza, «Implementación de automatización aplicada a pruebas de software del área de calidad de la empresa Entrega Creativa,» Universidad Tecnológica del Perú, Perú, 2022.

[11] P. Chávez, «Automatización de casos de pruebas usando Selenium y TewstNG Framework para mejorar el proceso de pruebas de un sistema que brinda servicios RPO,2020,» Universidad Nacional Mayor de San Marcos, Perú, 2020.

[12] I. García, «Automatización de prueba de software para mejorar el proceso de prueba del módulo del sistema que desarrolla empresa Experts Systems S.A.C,» Universidad Nacional Santiago Antúnez de Mayolo, Lima, 2022.

[13] J. Palomino Barrasa, «Automatizacion de plan de pruebas funcional: Plataforma web de la universidad politecnica de madrid,» universidad politecnica de madrid, Madrid, 2020.

[14] I. A. G. Calero, «Creación de pruebas automatizadas para modelos ejecutables de software,» UNIVERSIDAD DE MÁLAGA, 2021, 2021.

[15] F. Salas Villalobos y H. Jiménez Chévez, «Prácticas y experiencias en automatización de pruebas funcionales de integración en C#,» UNIVERSIDAD NACIONAL COSTA RICA, COSTA RICA, 2021.

[16] A. Cortés, «Automatización de pruebas de regresión para reducción de tiempo de entrega de nuevas versiones de sofware,» Universidad de Chile, Chile, 2020.

[17] D. Spindola, «Desarrollo de un framework de pruebas automatizadas para la verificaicón de aplicaciones web,» Universidad Autpnoma de Querétaro, Ecuador, 2021.

[18] J. R. Páez Maldonado, «Implementación de pruebas automatizadas para la integración de los procesos deesarrollo de software de una POC (Tienda virtual,» Bogota, 2023.

[19] R. Pressman, «Ingeniería de software: Un enfoque práctico, 7th ed.,» McGraw-Hill, Ciudad de México, 2010.

[20] I. Sommerville, «Ingeniería de software 9th ed.,» Addison-Wesley, Madrid, 2011.

[21] S. y. B. Myers, «The Art of Software Testing 3rd ed,» Hoboken, NJ: John Wiley & Sons, 2011.

[22] F. y. Highsmith, «Refactoring: Improving the Design of Existing Code 2nd ed,» Addison-Wesley, 2020.

[23] C. y. Gregory, «Agile Testing: A Practical Guide for Testers and Agile Teams 2nd ed.,» Addison-Wesley, 2021.

[24] G. Adzic, «Specification by Example: How Successful Teams Deliver the Right Software2nd ed,» Manning Publications, 2019.

[25] I. Idea, «Intellij Idea,» Ijtellij Idea, 2020.

[26] N. Garg, «Selenium WebDriver Step By Step Guide,» AdactIn Group Pty Ltd., 2014.

[27] Cucumber.io, «Cucumber.io,» Cucumber.io, 2024. [En línea]. Available: https://cucumber.io/. [Último acceso: 21 mayo 2024].

[28] B. y. P. Kaner, «Lessons Learned in Software Testing: A Context-Driven Approach 2nd ed.,» Wiley, 2020.

[29] Sommerville, «Software Engineering 10th ed.,» Pearson, 2015.

[30] J. Lewis, «Software Testing and Continuous Quality Improvement 4th ed,» Auerbach Publications, 2019.

[31] K. Schwaber y J. Sutherland, «La Guía Scrum,» Creative Commons, 2020.

[32] M. Torrado Nevado, «Estudio de metodologías ágiles. Aplicación SCRUM,» Universidad de sevilla, sevilla, 2019.

[33] L. J. Delgado Taday, «METODOLOGÍA BDD APLICADA AL DESARROLLO DEL SISTEMA WEB PARA EVALUAR LA CALIDAD DE LOS SERVICIOS TURÍSTICOS COMUNITARIOS RURALES DEL CANTÓN RIOBAMBA,» Riobamba, 2022.

[34] S. España, «Patrones de diseño en automatización:PageObjects,» 2014.

[35] ISQTB, Probador Certificado de ISQTB, 2018 ed., Belgica, 2018.

[36] D. Spindola Hernandez, «Desarrollo de un framework de pruebas automatizadas para la verificacion de aplicaciones web,» Santiago de Queretaro, 2021.

[37] E. Parra Valverde, «Estudio del control de calidad en procesos de desarrollo de software. Aplicación práctica de pruebas automatizadas a un sitio web de comercio electrónico,» Valladolid, 2020.

[38] R. D C y M. A E, «A Comprehensive Study on Automation Testing using JUnit,» IRJET, India, 2020.

[39] W. Rojas y M. Silva, Introduccion a java: guia de actividades practicas, Colombia: Universidad El Bosque, 2016.

[40] Rubin, «Essential Scrum: A Practical Guide to the Most Popular Agile Process 2nd ed,» Addison-Wesley, 2020.

[41] F. Toledo, «Introducción a las pruebas de sistemas de información,» Abstracta, Uruguay, 2014.

[42] D. Esteve, «implantación de un proceso de automatización de pruebas para una aplicación de software,» Universidad Politécnica de Valencia, Valencia, España, 2015.

[43] C. Bernal, «Metodología de la investigación (Tercera edición),» Pearson Educación de Colombia Ltda, Colombia, 2010.

[44] M. Tamayo, «El proceso de la investigación científica,» Editorial Limusa, Mexico, 2004.

[45] R. y. M. C. Hernández, «Metodología de la investigación. Las rutas cuantitativa, cualitativa y mixta,» Ed. Mc Graw Hill Education, 2018.

[46] G. Baena, «todología de la investigación. Grupo Editorial Patria,» Grupo Editorial Patria, Mexico, 2017.

[47] C. y. Creswell, «Research Design: Qualitative, Quantitative, and Mixed Methods Approaches 5th

ed.,» Thousand Oaks, CA: SAGE Publications, 2020.

[48] F. y. B. Hernández, «Metodología de la investigación 6th ed.,» McGraw-Hill, Ciudad de México, 2014.

[49] C. y. L. Sampieri, «Metodología de la investigación 6th ed.,» McGraw-Hill, Ciudad de México, 2014.

[50] D. M. L. y. D. F. Stephan, «Statistics for Managers Using Microsoft Excel, 8ª ed,» Pearson, Boston, USA, 2017.

[51] D. C. M. y. G. C. Runger, «Applied Statistics and Probability for Engineers, 6ª ed,» Wiley, NJ, USA, 2014.

[52] M. K. y. J. Kim, «Software Testing: Principles and Practices 1st ed.,» Springer, 2021.

# Anexos

## Anexo 1: Matriz de consistencia

<table>
<tr><td colspan="6">Título: Marco de Pruebas Automatizadas para el software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete</td></tr>
<tr><td colspan="6">Autor: Román Sánchez Eduardo Zósimo – Ancajima Rodríguez, José Adrián</td></tr>
<tr><td>Problema</td><td>Objetivos</td><td>Hipótesis</td><td colspan="3">Variables e Indicadores</td></tr>
<tr><td rowspan="3">PG: ¿De qué manera la implementación de un Marco de Pruebas Automatizadas puede mejorar la eficacia del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete?</td><td rowspan="3">OG: Evaluar de qué manera la implementación de un Marco de Pruebas Automatizadas puede mejorar la eficacia del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.</td><td rowspan="3">HG: La implementación de un Marco de Pruebas Automatizadas mejorará significativamente la eficacia del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.</td><td colspan="3">Variable Independiente: Pruebas Automatizadas</td></tr>
<tr><td colspan="3">Variable Dependiente: Eficiencia en el Proceso de Pruebas</td></tr>
<tr><td>Dimensiones</td><td>Indicadores</td><td>Escala</td></tr>
<tr><td rowspan="2">PE1: ¿De qué manera la implementación de un Marco de Pruebas Automatizadas puede mejorar la calidad del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete?</td><td rowspan="2">OE1: Evaluar de qué manera la implementación de un Marco de Pruebas Automatizadas puede mejorar la calidad del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.</td><td rowspan="2">HE1: La implementación de un Marco de Pruebas Automatizadas mejorará significativamente la calidad del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.</td><td>Calidad de Software</td><td>Número de errores detectados</td><td>De razón</td></tr>
<tr><td>Productividad en el Proceso de Pruebas</td><td>Tiempo de ejecución de las pruebas</td><td>De razón</td></tr>
<tr><td rowspan="3">PE2: ¿De qué manera la implementación de un Marco de Pruebas Automatizadas puede incrementar la productividad en el proceso de pruebas del software de Gestión</td><td rowspan="3">OE2: Evaluar de qué manera la implementación de un Marco de Pruebas Automatizadas puede incrementar la productividad en el proceso de pruebas del</td><td rowspan="3">HE2: La implementación de un Marco de Pruebas Automatizadas incrementará significativamente la productividad en el proceso de pruebas del software de Gestión Digital de Procesos de Investigación de la</td><td colspan="3">Metodología</td></tr>
<tr><td colspan="2">Enfoque: Cuantitativo</td><td>Población: Todas las funcionalidades del sistema de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete</td><td>Técnica: Observación estructurada</td></tr>
<tr><td>Tipo: Aplicada</td><td>Muestra: Treinta (30) funcionalidades críticas seleccionadas</td><td colspan="2">Instrumento: Fichas de registro</td></tr>
<tr><td colspan="2">Método: Hipotético-Deductivo</td><td>Muestreo: no probabilístico por criterio</td><td></td></tr>
</table>

| | | | | | |
|---|---|---|---|---|---|
| Digital de Procesos de Investigación de la Universidad Nacional de Cañete? | software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete. | Universidad Nacional de Cañete. | **Diseño:** Experimental de carácter Pre-experimental | | |
| | | | **Nivel:** Aplicativo | | |

**Anexo 2: Instrumento de recolección de datos**

## Ficha de Registro 1: Número de Errores Detectados

| **Ficha de registro del indicador 1: Número de Errores Detectados (NED)** | |
|---|---|
| **Investigador:** Román Sánchez Eduardo Zósimo - Ancajima Rodríguez Adrián | |
| **Universidad:** Universidad Nacional de Cañete | |
| **Fase:** Pre-Test | |
| **Proceso Observado:** Ejecución de Pruebas Automatizadas | |
| **Indicador:** Número de Errores Detectados | |
| **Medida:** Cantidad | |
| **Fórmula:** Total de errores detectados | |
| **NED:** Número de Errores Detectados | |
| **Ítem** | **NED** |
| 1 | |
| 2 | |
| 3 | |
| ... | |
| 30 | |

## Ficha de Registro 2: Tiempo de Ejecución de las Pruebas

| **Ficha de registro del indicador 3: Tiempo de Ejecución de las Pruebas (TEP)** | |
|---|---|
| **Investigador:** Román Sánchez Eduardo Zósimo - Ancajima Rodríguez Adrián | |
| **Universidad:** Universidad Nacional de Cañete | |
| **Fase:** Pre-Test | |
| **Proceso Observado:** Ejecución de Pruebas Automatizadas | |
| **Indicador:** Tiempo de Ejecución de las Pruebas | |
| **Medida:** Tiempo (minutos) | |
| **Fórmula:** Tiempo total de ejecución | |
| **TEP:** Tiempo de Ejecución de las Pruebas | |
| **Ítem** | **TEP (min)** |
| 1 | |
| 2 | |
| 3 | |
| ... | |
| 30 | |

## Anexo 3: Certificado de validez de contenido (Juicio de expertos)

### Validación del Experto 1

UNIVERSIDAD NACIONAL DE CAÑETE

# VALIDACIÓN DEL INSTRUMENTO DE INVESTIGACIÓN POR CRITERIO DE JUECES

**I. DATOS GENERALES**

1.1 Apellidos y nombres del juez : Mg. Alex Abelardo, Pacheco Pumaleque
1.2 Cargo e institución donde labora : Docente investigador de la Universidad Nacional de Cañete
1.3 Nombre del instrumento evaluado: Fichas de registros para la evaluación de la eficiencia del software de Gestión Digital de Procesos de Investigación.
1.4. Autor (es) del instrumento : Bach. Eduardo Zósimo Román Sánchez, Bach. Adrián Ancajima Rodríguez.

**II. ASPECTO DE LA VALIDACIÓN**

| INDICADORES | CRITERIOS | Deficiente 1 | Baja 2 | Regular 3 | Buena 4 | Muy buena 5 |
|---|---|---|---|---|---|---|
| 1.CLARIDAD | Está formulado con lenguaje apropiado y comprensible | | | | | X |
| 2.OBJETIVIDAD | Permite medir hechos observables. | | | | | X |
| 3.ACTUALIDAD | Adecuado al avance de la ciencia y tecnología. | | | | | X |
| 4.ORGANIZACIÓN | Presentación sistemática y ordenada. | | | | | X |
| 5.SUFICIENCIA | Comprende aspectos de las variables en cantidad y calidad suficiente. | | | | | X |
| 6.PERTINENCIA | Permite conseguir datos de acuerdo a los objetivos planteados. | | | | | X |
| 7.CONSISTENCIA | Pretende conseguir datos basados en teorías o modelos teóricos. | | | | | X |
| 8.COHERENCIA | Entre variables, indicadores y los ítems. | | | | | X |
| 9.METODOLOGÍA | La metodología responde al propósito de la investigación. | | | | | X |
| 10.APLICACIÓN | Los datos permiten un tratamiento estadístico pertinente y adecuado. | | | | | X |
| | | ↓ | ↓ | ↓ | ↓ | ↓ |
| CONTEO TOTAL DE MARCAS (realice el conteo en cada una de las categorías de la escala) | | 0 A | 0 B | 0 C | 0 D | 50 E |

$$Coeficiente\ de\ Validez = \frac{A+B+C+D+E}{50} = \frac{50}{50} = 1$$

**III. CALIFICACIÓN GLOBAL** (Ubique el coeficiente de validez obtenido en el intervalo respectivo y marque con un aspa en el círculo asociado)

| Niveles | Condición | Intervalos |
|---|---|---|
| Desaprobado | ( ) | [0,00 – 0,50] |
| Observado | ( ) | [0,56 – 0,75] |
| Aprobado | (X) | [0,76 – 1,00] |

**IV. OPINIÓN DE APLICABILIDAD**

.....

.....

Lugar y fecha: San Vicente de Cañete, 20 de mayo de 2024

Firmado digitalmente por PACHECO PUMALEQUE ALEX ABELARDO
Motivo: Doy V° B°
Fecha: 24.05.2024 17:32:27 -05:00

**Firma del juez**

**Validación del Experto 2**

UNIVERSIDAD NACIONAL DE CAÑETE

# VALIDACIÓN DEL INSTRUMENTO DE INVESTIGACIÓN POR CRITERIO DE JUECES

## I. DATOS GENERALES

1.1 Apellidos y nombres del juez : Mg. Ricardo Carlos, Inquilla Quispe
1.2 Cargo e institución donde labora : Jefe de la Oficina de Sistemas de Información y Tecnologías de Información de la Universidad Nacional de Cañete
1.3 Nombre del instrumento evaluado: Fichas de registros para la evaluación de la eficiencia del software de Gestión Digital de Procesos de Investigación.
1.4. Autor (es) del instrumento : Bach. Eduardo Zósimo Román Sánchez, Bach. Adrián Ancajima Rodríguez.

## II. ASPECTO DE LA VALIDACIÓN

| INDICADORES | CRITERIOS | Deficiente 1 | Baja 2 | Regular 3 | Buena 4 | Muy buena 5 |
|---|---|---|---|---|---|---|
| 1.CLARIDAD | Está formulado con lenguaje apropiado y comprensible | | | | | X |
| 2.OBJETIVIDAD | Permite medir hechos observables. | | | | X | |
| 3.ACTUALIDAD | Adecuado al avance de la ciencia y tecnología. | | | | | X |
| 4.ORGANIZACIÓN | Presentación sistemática y ordenada. | | | | | X |
| 5.SUFICIENCIA | Comprende aspectos de las variables en cantidad y calidad suficiente. | | | | X | |
| 6.PERTINENCIA | Permite conseguir datos de acuerdo a los objetivos planteados. | | | | | X |
| 7.CONSISTENCIA | Pretende conseguir datos basados en teorías o modelos teóricos. | | | | | X |
| 8.COHERENCIA | Entre variables, indicadores y los ítems. | | | | | X |
| 9.METODOLOGÍA | La metodología responde al propósito de la investigación. | | | | | X |
| 10.APLICACIÓN | Los datos permiten un tratamiento estadístico pertinente y adecuado. | | | | | X |
| | | ↓ | ↓ | ↓ | ↓ | ↓ |
| CONTEO TOTAL DE MARCAS (realice el conteo en cada una de las categorías de la escala) | | 0 A | 0 B | 0 C | 8 D | 40 E |

$$Coeficiente\ de\ Validez = \frac{A+B+C+D+E}{50} = \frac{48}{50} = 0.96$$

## III. CALIFICACIÓN GLOBAL (Ubique el coeficiente de validez obtenido en el intervalo respectivo y marque con un aspa en el círculo asociado)

| Niveles | Condición | Intervalos |
|---|---|---|
| Desaprobado | ( ) | [0,00 – 0,50] |
| Observado | ( ) | [0,56 – 0,75] |
| Aprobado | (X) | [0,76 – 1,00] |

## IV. OPINIÓN DE APLICABILIDAD

APLICABLE

Lugar y fecha: San Vicente de Cañete, 24 de mayo de 2024

**Firma del juez**

## Validación del Experto 3

UNIVERSIDAD NACIONAL DE CAÑETE

### VALIDACIÓN DEL INSTRUMENTO DE INVESTIGACIÓN POR CRITERIO DE JUECES

**I. DATOS GENERALES**

1.1 Apellidos y nombres del juez : Mag. Hussein Anthony, Palomino Quispe
1.2 Cargo e institución donde labora : Docente en la Universidad Nacional de Cañete
1.3 Nombre del instrumento evaluado: Fichas de registros para la evaluación de la eficiencia del software de Gestión Digital de Procesos de Investigación.
1.4. Autor (es) del instrumento : Bach. Eduardo Zósimo Román Sánchez, Bach. Adrián Ancajima Rodríguez.

**II. ASPECTO DE LA VALIDACIÓN**

| INDICADORES | CRITERIOS | Deficiente 1 | Baja 2 | Regular 3 | Buena 4 | Muy buena 5 |
|---|---|---|---|---|---|---|
| 1.CLARIDAD | Está formulado con lenguaje apropiado y comprensible | | | | | x |
| 2.OBJETIVIDAD | Permite medir hechos observables. | | | | | x |
| 3.ACTUALIDAD | Adecuado al avance de la ciencia y tecnología. | | | | x | |
| 4.ORGANIZACIÓN | Presentación sistemática y ordenada. | | | | | x |
| 5.SUFICIENCIA | Comprende aspectos de las variables en cantidad y calidad suficiente. | | | | | x |
| 6.PERTINENCIA | Permite conseguir datos de acuerdo a los objetivos planteados. | | | | | x |
| 7.CONSISTENCIA | Pretende conseguir datos basados en teorías o modelos teóricos. | | | | | x |
| 8.COHERENCIA | Entre variables, indicadores y los ítems. | | | | | x |
| 9.METODOLOGÍA | La metodología responde al propósito de la investigación. | | | | | x |
| 10.APLICACIÓN | Los datos permiten un tratamiento estadístico pertinente y adecuado. | | | | | x |
| | | ↓ | ↓ | ↓ | ↓ | ↓ |
| CONTEO TOTAL DE MARCAS (realice el conteo en cada una de las categorías de la escala) | | 0 A | 0 B | 0 C | 4 D | 45 E |

$$Coeficiente\ de\ Validez = \frac{A+B+C+D+E}{50} = \frac{49}{50} = 0.98$$

**III. CALIFICACIÓN GLOBAL** (Ubique el coeficiente de validez obtenido en el intervalo respectivo y marque con un aspa en el círculo asociado)

| Niveles | Condición | Intervalos |
|---|---|---|
| Desaprobado | ( ) | [0,00 – 0,50] |
| Observado | ( ) | [0,56 – 0,75] |
| Aprobado | (x) | [0,76 – 1,00] |

**IV. OPINIÓN DE APLICABILIDAD**

Aplicable.

Lugar y fecha: San Vicente de Cañete, 7 de julio de 2024

Firmado digitalmente por: PALOMINO QUISPE HUSSEIN ANTHONY

**Firma del juez**

## Anexo 4: Constancia de grados y títulos de los validadores (SUNEDU)

### Experto 1

| Graduado | Grado o Título | Institución |
|---|---|---|
| PACHECO PUMALEQUE, ALEX ABELARDO<br>**DNI 41651279** | **INGENIERO DE SISTEMAS**<br><br>**Fecha de diploma: 31/08/2011**<br>Modalidad de estudios: - | UNIVERSIDAD PRIVADA TELESUP S.A.C.<br>***PERU*** |
| PACHECO PUMALEQUE, ALEX ABELARDO<br>**DNI 41651279** | **LICENCIADO EN EDUCACION ESPECIALIDAD EN INGLES**<br><br>**Fecha de diploma: 08/09/2010**<br>Modalidad de estudios: - | UNIVERSIDAD NACIONAL FEDERICO VILLARREAL<br>***PERU*** |
| PACHECO PUMALEQUE, ALEX ABELARDO<br>**DNI 41651279** | **BACHILLER EN EDUCACION**<br><br>**Fecha de diploma: 27/10/2008**<br>Modalidad de estudios: -<br><br>Fecha matrícula: Sin información (***)<br>Fecha egreso: Sin información (***) | UNIVERSIDAD NACIONAL FEDERICO VILLARREAL<br>***PERU*** |
| PACHECO PUMALEQUE, ALEX ABELARDO<br>**DNI 41651279** | **BACHILLER EN INGENIERIA DE SISTEMAS**<br><br>**Fecha de diploma: 22/02/2010**<br>Modalidad de estudios: -<br><br>Fecha matrícula: Sin información (***)<br>Fecha egreso: Sin información (***) | UNIVERSIDAD PRIVADA TELESUP S.A.C.<br>***PERU*** |
| PACHECO PUMALEQUE, ALEX ABELARDO<br>**DNI 41651279** | **MASTER EN DIRECCION ESTRATEGICA EN TECNOLOGIAS DE LA INFORMACION**<br><br>**Fecha de diploma: 30/11/15**<br>Modalidad de estudios: A DISTANCIA<br><br>Fecha matrícula: Sin información (***)<br>Fecha egreso: Sin información (***) | UNIVERSIDAD DE PIURA<br>***PERU*** |
| PACHECO PUMALEQUE, ALEX ABELARDO<br>**DNI 41651279** | **MAESTRO EN INGENIERÍA INDUSTRIAL CON MENCIÓN EN GERENCIA DE LA CALIDAD Y PRODUCTIVIDAD**<br><br>**Fecha de diploma: 23/08/19**<br>Modalidad de estudios: PRESENCIAL<br><br>Fecha matrícula: 21/07/2011<br>Fecha egreso: 02/10/2018 | UNIVERSIDAD NACIONAL DEL CALLAO<br>***PERU*** |

**Experto 2**

**REGISTRO NACIONAL DE GRADOS ACADÉMICOS Y TÍTULOS PROFESIONALES**

| Graduado | Grado o Título | Institución |
|---|---|---|
| INQUILLA QUISPE, RICARDO CARLOS<br>**DNI 00515158** | **INGENIERO EN INFORMATICA Y SISTEMAS**<br><br>**Fecha de diploma: 01/09/2003**<br>Modalidad de estudios: - | UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN<br>***PERU*** |
| INQUILLA QUISPE, RICARDO CARLOS<br>**DNI 00515158** | **BACHILLER EN CIENCIAS INFORMATICA Y SISTEMAS**<br><br>**Fecha de diploma: 02/04/2002**<br>Modalidad de estudios: -<br><br>Fecha matrícula: Sin información (***)<br>Fecha egreso: Sin información (***) | UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN<br>***PERU*** |
| INQUILLA QUISPE, RICARDO CARLOS<br>**DNI 00515158** | **BACHILLER EN CIENCIAS CON MENCION EN INFORMATICA Y SISTEMAS**<br><br>**Fecha de diploma: 02/04/2002**<br>Modalidad de estudios: -<br><br>Fecha matrícula: Sin información (***)<br>Fecha egreso: Sin información (***) | UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN<br>***PERU*** |
| INQUILLA QUISPE, RICARDO CARLOS<br>**DNI 00515158** | **MAESTRO EN INGENIERIA DE SISTEMAS CON MENCIÓN EN INGENIERIA DE SOFTWARE**<br><br>**Fecha de diploma: 13/11/19**<br>Modalidad de estudios: PRESENCIAL<br><br>Fecha matrícula: 15/01/2013<br>Fecha egreso: 31/12/2015 | UNIVERSIDAD NACIONAL FEDERICO VILLARREAL<br>***PERU*** |
| INQUILLA QUISPE, RICARDO CARLOS<br>**DNI 00515158** | **MAESTRO EN ADMINISTRACIÓN DE NEGOCIOS - MBA**<br><br>**Fecha de diploma: 11/04/22**<br>Modalidad de estudios: SEMIPRESENCIAL<br><br>Fecha matrícula: 06/04/2020<br>Fecha egreso: 31/01/2022 | UNIVERSIDAD CÈSAR VALLEJO S.A.C.<br>***PERU*** |

**Experto 3**



PERÚ | Ministerio de Educación | Superintendencia Nacional de Educación Superior Universitaria | Dirección de Documentación e Información Universitaria y Registro de Grados y Títulos

## REGISTRO NACIONAL DE GRADOS ACADÉMICOS Y TÍTULOS PROFESIONALES

| Graduado | Grado o Título | Institución |
|---|---|---|
| PALOMINO QUISPE, HUSSEIN ANTHONY<br>**DNI 73623097** | **BACHILLER EN INGENIERÍA DE SISTEMAS**<br>**Fecha de diploma: 26/06/19**<br>Modalidad de estudios: PRESENCIAL<br>Fecha matrícula: 22/04/2014<br>Fecha egreso: 04/01/2019 | UNIVERSIDAD NACIONAL DE CAÑETE<br>***PERU*** |
| PALOMINO QUISPE, HUSSEIN ANTHONY<br>**DNI 73623097** | **INGENIERO DE SISTEMAS**<br>**Fecha de diploma: 26/11/21**<br>Modalidad de estudios: PRESENCIAL | UNIVERSIDAD NACIONAL DE CAÑETE<br>***PERU*** |
| PALOMINO QUISPE, HUSSEIN ANTHONY<br>**DNI 73623097** | **MAESTRO EN INGENIERÍA DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN**<br>**Fecha de diploma: 13/11/23**<br>Modalidad de estudios: PRESENCIAL<br>Fecha matrícula: 04/04/2022<br>Fecha egreso: 17/08/2023 | UNIVERSIDAD CÉSAR VALLEJO S.A.C.<br>***PERU*** |

## Anexo 5: Base de datos de los indicadores (recolección de datos)

**Indicador 1**

| Ficha de registro del indicador 1: Número de Errores Detectados (NED) | |
|---|---|
| **Investigador: Román Sánchez Eduardo Zósimo - Ancajima Rodríguez Adrián** | |
| **Universidad: Universidad Nacional de Cañete** | |
| **Fase: Pre-Test** | |
| **Proceso Observado: Ejecución de Pruebas Automatizadas** | |
| **Indicador: Número de Errores Detectados** | |
| **Medida: Cantidad** | |
| **Fórmula: Total de errores detectados** | |
| **NED: Número de Errores Detectados** | |
| **Ítem** | **NED** |
| **1** | 1 |
| **2** | 2 |
| **3** | 0 |
| **4** | 2 |
| **5** | 1 |
| **6** | 2 |
| **7** | 1 |
| **8** | 0 |
| **9** | 0 |
| **10** | 0 |
| **11** | 1 |
| **12** | 1 |
| **13** | 2 |
| **14** | 1 |
| **15** | 2 |
| **16** | 2 |
| **17** | 1 |
| **18** | 1 |
| **19** | 1 |
| **20** | 2 |
| **21** | 2 |
| **22** | 0 |
| **23** | 2 |
| **24** | 0 |
| **25** | 0 |
| **26** | 0 |
| **27** | 0 |
| **28** | 2 |
| **29** | 1 |
| **30** | 1 |

| Ficha de registro del indicador 1: Número de Errores Detectados (NED) | |
|---|---|
| **Investigador: Román Sánchez Eduardo Zósimo - Ancajima Rodríguez Adrián** | |
| **Universidad: Universidad Nacional de Cañete** | |
| **Fase: Post - Test** | |
| **Proceso Observado: Ejecución de Pruebas Automatizadas** | |
| **Indicador: Número de Errores Detectados** | |
| **Medida: Cantidad** | |
| **Fórmula: Total de errores detectados** | |
| **NED: Número de Errores Detectados** | |
| **Ítem** | **NED** |
| **1** | 4 |
| **2** | 5 |
| **3** | 4 |
| **4** | 5 |
| **5** | 4 |
| **6** | 5 |
| **7** | 4 |
| **8** | 5 |
| **9** | 5 |
| **10** | 4 |
| **11** | 5 |
| **12** | 1 |
| **13** | 2 |
| **14** | 2 |
| **15** | 4 |
| **16** | 2 |
| **17** | 3 |
| **18** | 5 |
| **19** | 5 |
| **20** | 4 |
| **21** | 4 |
| **22** | 3 |
| **23** | 2 |
| **24** | 3 |
| **25** | 4 |
| **26** | 5 |
| **27** | 4 |
| **28** | 2 |
| **29** | 5 |
| **30** | 5 |

**Indicador 2**

| Ficha de registro del indicador 2: Tiempo de Ejecución de las Pruebas (TEP) | |
|---|---|
| **Investigador: Román Sánchez Eduardo Zósimo - Ancajima Rodríguez Adrián** | |
| **Universidad: Universidad Nacional de Cañete** | |
| **Fase: Pre-Test** | |
| **Proceso Observado: Ejecución de Pruebas Automatizadas** | |
| **Indicador: Tiempo de Ejecución de las Pruebas** | |
| **Medida: Tiempo (minutos)** | |
| **Fórmula: Tiempo total de ejecución** | |
| **TEP: Tiempo de Ejecución de las Pruebas** | |
| **Ítem** | **TEP (min)** |
| **1** | 1,40 |
| **2** | 1,45 |
| **3** | 1,58 |
| **4** | 1,37 |
| **5** | 2,10 |
| **6** | 2,06 |
| **7** | 2,23 |
| **8** | 2,41 |
| **9** | 2,58 |
| **10** | 2,54 |
| **11** | 2,30 |
| **12** | 3,10 |
| **13** | 3,28 |
| **14** | 3,45 |
| **15** | 3,49 |
| **16** | 2,54 |
| **17** | 2,30 |
| **18** | 3,10 |
| **19** | 3,28 |

| | |
|---|---|
| **20** | 1,40 |
| **21** | 1,45 |
| **22** | 1,58 |
| **23** | 1,37 |
| **24** | 2,10 |
| **25** | 2,41 |
| **26** | 2,58 |
| **27** | 2,54 |
| **28** | 2,30 |
| **29** | 3,10 |
| **30** | 4,28 |

| Ficha de registro del indicador 3: Tiempo de Ejecución de las Pruebas (TEP) | |
|---|---|
| **Investigador: Román Sánchez Eduardo Zósimo - Ancajima Rodríguez Adrián** | |
| **Universidad: Universidad Nacional de Cañete** | |
| **Fase: Pre-Test** | |
| **Proceso Observado: Ejecución de Pruebas Automatizadas** | |
| **Indicador: Tiempo de Ejecución de las Pruebas** | |
| **Medida: Tiempo (minutos)** | |
| **Fórmula: Tiempo total de ejecución** | |
| **TEP: Tiempo de Ejecución de las Pruebas** | |
| **Ítem** | **TEP (min)** |
| **1** | 0,33 |
| **2** | 0,38 |
| **3** | 0,51 |
| **4** | 0,30 |
| **5** | 0,53 |
| **6** | 0,59 |
| **7** | 0,56 |
| **8** | 1,34 |
| **9** | 1,51 |
| **10** | 1,47 |
| **11** | 1,23 |
| **12** | 1,53 |
| **13** | 2,21 |
| **14** | 2,38 |
| **15** | 2,42 |
| **16** | 1,47 |
| **17** | 1,23 |
| **18** | 1,53 |
| **19** | 2,21 |
| **20** | 0,33 |
| **21** | 0,38 |
| **22** | 0,51 |
| **23** | 0,30 |

| | |
|---|---|
| **24** | 0,73 |
| **25** | 1,34 |
| **26** | 1,51 |
| **27** | 1,47 |
| **28** | 1,23 |
| **29** | 1,53 |
| **30** | 3,21 |

## Anexo 6: Autorización de la institución para realizar la investigación

UNIVERSIDAD NACIONAL
DE CAÑETE
Ley de Creación N° 29488

*"Año del Bicentenario, de la consolidación de nuestra Independencia, y de la conmemoración de las heroicas batallas de Junín y Ayacucho"*

**VICEPRESIDENCIA DE INVESTIGACIÓN**

San Vicente, 25 de octubre de 2024

**OFICIO N° 970-2024/UNDC/CO/VPIN**

**Señora**
**Dra. Amanda Durán Carhuamaca**
**Directora (e) de la Unidad de Investigación de la Facultad de Ingeniería**
**Universidad Nacional de Cañete**
**Presente. -**

**ASUNTO** : **RESPUESTA A SOLICITUD DE AUTORIZACIÓN PARA EL USO DEL SISTEMA GEDIPI EN INVESTIGACIÓN**

**REF.** : **OFICIO N° 142-2024/UNDC/FI-UIIS/ADC**

De mi especial consideración:

Me dirijo a usted para saludarla, al mismo tiempo en en respuesta al documento de la referencia donde solicita emisión del documento de autorización para el uso del sistema GEDIPI en el marco de la investigación (Tesis) titulada "Marco de Pruebas Automatizadas para el Sistema de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete", me permito comunicar lo siguiente:

- Tras revisar el historial del proyecto, es importante destacar que, si bien se cuenta con un informe técnico favorable emitido por la Oficina de Tecnologías de la Información (OTI) y la ratificación de la Comisión Organizadora, las pruebas automatizadas se llevaron a cabo sin la autorización formal previa de esta Vicepresidencia de Investigación, tal como lo requiere el reglamento vigente. Este procedimiento es fundamental para garantizar el adecuado uso de los sistemas institucionales y asegurar la debida coordinación entre las áreas involucradas.

En ese sentido, no podemos emitir una autorización retroactiva para las actividades ya realizadas sin la debida aprobación. Sin embargo, considerando el avance de la investigación y con la finalidad de regularizar la situación a partir de este punto, **emito el presente documento formal autorizando el uso del sistema GEDIPI desde la recepción del documento de la referencia**, deslindando responsabilidades sobre las pruebas realizadas sin la correspondiente autorización.

Dicho documento será emitido bajo los términos mencionados, con el propósito de dar continuidad a los procedimientos institucionales que permitan la validación y reconocimiento de los resultados de la investigación, sin perjuicio de las acciones futuras para asegurar que todos los proyectos cumplan con los requisitos establecidos por la normativa interna.

Sin otro particular me despido de usted.

Atentamente,

**Firmado digitalmente**
Dra. Janett Karina Vásquez Pérez
Vicepresidente de Investigación

JKVP/fmyv

FIRMA DIGITAL
Firmado digitalmente por: VASQUEZ PEREZ JANETT KARINA FIR 16789943 hard
Motivo: Doy V° B°
Fecha: 25/10/2024 10:40:38-0500

## Anexo 7: Autorización de la inclusión en el repositorio de la biblioteca



### AUTORIZACIÓN DE INCLUSIÓN EN EL REPOSITORIO DE LA BIBLIOTECA

Yo, **ANCAJIMA RODRIGUEZ JOSE ADRIAN, ROMAN SANCHEZ EDUARDO ZOSIMO,** titular del documento de identidad número **71436255, 71215241**,estudiantes de la carrera **INGENIERA DE SISTEMAS** de la Universidad Nacional de Cañete (UNDC), por medio de la presente, en calidad de autor y actuando en representación de mis co-autores, autorizo a la Biblioteca Central de la UNDC a incluir y difundir la tesis titulada **"Marco de pruebas automatizadas para el software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete"**, que he presentado para obtener el grado de **Título Profesional en Ingeniería de Sistemas**, en su repositorio digital y otros medios electrónicos que la universidad considere pertinentes.

A través de esta autorización:

1. Reconozco que los derechos de autor de la tesis me pertenecen y que cualquier uso indebido por parte de terceros es responsabilidad exclusiva de quienes lo realicen.
2. Permito que la tesis pueda ser consultada por usuarios internos y externos de la Universidad y que se realicen copias de seguridad, migraciones de formato y otras acciones necesarias para su preservación y acceso.
3. Acepto que esta autorización no implica la renuncia a ningún derecho que me corresponda como autor de la tesis, incluidos los derechos de reproducción, distribución y comunicación pública.
4. Entiendo que la Universidad se compromete a utilizar la tesis exclusivamente con fines académicos y no comerciales.
5. Declaro que la información proporcionada en este documento es veraz y completa. Me hago responsable por cualquier inexactitud o falsedad en la misma.

Fecha: 12/07/2024

Atentamente,

Firma:







Av. Mariscal Benavides 1370, Sede Académica: Casa de la Cultura.
Código Postal 15701. San Vicente, Cañete, Lima, Perú | www.undc.edu.pe

## Anexo 8: Autorización de publicación en el repositorio institucional de la UNDC

| UNIVERSIDAD NACIONAL DE CAÑETE | POLÍTICAS Y REGLAMENTO DEL REPOSITORIO INSTITUCIONAL DEACCESO ABIERTO | | |
|---|---|---|---|
| | Código: R-M02-VRI-004 | Revisión: 01 | Fecha: Marzo-2023 |

**AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL DE LA UNDC**

**I. INFORMACIÓN DEL DOCUMENTO**

**1.1 Título del documento:**

Marco de pruebas automatizadas para el software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete

**1.2 Descripción (Seleccionar)**

| | | | |
|---|---|---|---|
| Tesis de Pregrado | X | Libro | |
| Tesis de Maestría | | Capítulo de libro | |
| Tesis de Doctorado | | Monografía | |
| Trabajo de Investigación | | Artículo | |
| Tesis de Segunda Especialidad | | Conferencia | |
| Trabajo de Suficiencia Profesional | | Programa informativo | |
| Trabajo académico | | Datos | |

**1.3 Autores del documento**

| Apellidos | Nombres | Correo | DNI | ORCID |
|---|---|---|---|---|
| Ancajima Rodriguez | José Adrian | 1671436255@undc.edu.pe | 15413791 | orcid.org/0009-0001-0400-096X |
| Román Sánchez | Eduardo Zosimo | 1571215241@undc.edu.pe | 71215241 | orcid.org/0000-0001-8414-8560 |

**1.4 Asesor (es)**

| Apellidos | Nombres | Correo | DNI | ORCID |
|---|---|---|---|---|
| Duran Carhuamaca | Amanda | aduran@undc.edu.pe | 20114878 | orcid.org/0000-0001-8183-5891 |
| Asto Huaman | Leonidas | lasto@undc.edu.pe | 41774206 | orcid.org/0000-0003-2003-1798 |

**1.5 Centro de Investigación**

| | |
|---|---|
| Facultad | De Ingeniería |
| Escuela Profesional | De Ingeniería de Sistemas |

**II. ORIGINALIDAD DEL TRABAJO PRESENTADO**

Con la presentación de esta ficha, el(los) autor(es) señalan expresamente que la obra es original, ya que sus contenidos son producto de su directa contribución intelectual. Se reconoce también que todos los datos y las referencias a materiales ya publicados están debidamente identificados con su respectivo crédito e incluidos en las notas bibliográficas y en las citas que se destacan como tal.

**III. AUTORIZACIÓN DEL TRABAJO PRESENTADO**

Por medio del presente documento, afirmo y garantizo(amos) ser el (los) legítimo (s), titular(es) de todos los derechos de propiedad intelectual sobre los entregables, las obras, los contenidos, los productos y/o las creaciones en general (en adelante, los "Contenidos" que serán incluidos en el Repositorio Institucional de la Universidad Nacional de Cañete (en adelante, la "Universidad").

Autorizo a la Universidad a publicar los Contenidos en el Repositorio Institucional de la Universidad y, en consecuencia, en el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto, sobre la base de lo establecido en la Ley N° 30035, sus normas reglamentarias, modificatorias, sustitutorias y conexas, y de acuerdo con las políticas de acceso abierto que la Universidad aplique en relación con su Repositorio Institucional. Autorizo expresamente toda consulta y uso de los contenidos, por parte de cualquier persona, por el tiempo de duración de los derechos patrimoniales de autor y derechos conexos, a título gratuito y a nivel mundial.

En consecuencia, la Universidad tendrá la posibilidad de divulgar y difundir los contenidos, de manera total o parcial, sin limitación alguna y sin derecho a pago de contraprestación, remuneración ni regalía alguna a favor mío; en los medios, canales y plataformas que la Universidad y/o el Estado de la República del Perú determinen, a nivel mundial, sin restricción geográfica alguna y de manera indefinida, pudiendo crear y/o extraer los metadatos sobre los Contenidos, e incluir los contenidos en los índices y buscadores que estimen necesarios para promover su difusión.

**IV: PUBLICACIÓN DIFERIDA**

Este ítem solo es de interés para los autores que han decidido la publicación diferida de sus documentos. Caso contrario, obvia este apartado.

| Autorización | Motivo (marcar) | | Firma |
|---|---|---|---|
| Solicito la publicación diferida del documento depositado en el repositorio, por 12 meses. (indicar el tiempo que solicita) | Exclusividad de revista, editor | | |
| | Por patente | | |
| | Secreto o seguridad nacional | | |
| | Por otras razones | | |

Cualquier motivo utilizado, el solicitante debe acreditar con documento su pedido (subir junto con la autorización la evidencia que sustenta su solicitud).

Yo, **ANCAJIMA RODRIGUEZ JOSE ADRIAN** , **ROMAN SANCHEZ EDUARDO ZOSIMO** con DNI N° **71436255**, **71215241** en mi calidad de autor y actuando en representación de mis co-autores, autorizo la publicación del documento indicado en el punto 1, bajo las condiciones indicadas en el punto 2 y 3, dejando constancia que el archivo digital que estoy entregando a la Universidad Nacional de Cañete, como parte del proceso de obtención del título profesional o grado académico, contiene la versión final del documento sustentado y aprobado por el jurado.



Firma

**12/07/2024**

Fecha

## Anexo 9: Constancia de Similitud

**UNIVERSIDAD NACIONAL DE CAÑETE**
Ley de Creación Nº 29488
**FACULTAD DE INGENIERÍA**
**UNIDAD DE INVESTIGACION DE LA FACULTAD DE INGENIERIA**
UNDC "Año del Bicentenario, de la Consolidación de nuestra Independencia
Y de la conmemoración de las heroicas batallas de Junín y Ayacucho"

San Vicente de Cañete, 20 de julio del 2024

**CONSTANCIA DE SIMILITUD DE INFORME FINAL DE TESIS N° 015-2024**

**Título del Trabajo de Investigación:** "Marco de pruebas automatizadas para el software de gestión digital de procesos de investigación de la Universidad Nacional de Cañete"

**Autor(es):** Ancajima Rodriguez Jose Adrian
Román Sánchez Eduardo Zósimo


EDUARDO ZOSIMO ROMAN SANCHEZ - JOSE ADRIAN ANCAJIMA RODRIGUEZ

| 13% | 12% | 1% | 4% |
|---|---|---|---|
| ÍNDICE DE SIMILITUD | FUENTES DE INTERNET | PUBLICACIONES | TRABAJOS DEL ESTUDIANTE |

| | | |
|---|---|---|
| 1 | repositorio.undc.edu.pe | 6% |
| 2 | hdl.handle.net | 3% |
| 3 | repositorio.unasam.edu.pe | 1% |
| 4 | oa.upm.es | 1% |
| 5 | Submitted to Universidad Nacional de Cañete | 1% |
| 6 | repositorio.unp.edu.pe | 1% |
| 7 | documentop.com | 1% |
| 8 | Submitted to Universidad Popular del Cesar,UPC | 1% |

Por lo tanto, se otorga esta constancia a solicitud del interesado(a) para los fines que estime conveniente.

**Mg. Amanda Durán Carhuamaca**
(e) Directora de la Unidad de Investigación
Facultad de Ingeniería – UNDC

FIRMA DIGITAL
Firmado digitalmente por: DURAN CARHUAMACA AMANDA FIR 20114878 hard
Motivo: En señal de conformidad
Fecha: 20/07/2024 13:52:11-0500

## Anexo 10: Metodología de desarrollo de la solución tecnológica

### 1. Desarrollo de la solución

Para esta investigación, se seleccionó la metodología ágil del marco de trabajo Scrum, es la que se seleccionó en el capítulo II, que especifica y detalla los Sprint fundamentales del desarrollo. Por lo tanto, la solución se divide en la aplicación de la metodología propuesta para la solución planteada.

### 2. Fase de Iniciación

**Asignación de roles:** Se decidió asignar roles a los participantes principales del proyecto actual de acuerdo con el marco de desarrollo ágil SCRUM para una toma de decisiones eficiente.

**Tabla X - Cuadro de roles del Proyecto**

| **Roles** | **Persona Asignada** |
|---|---|
| Scrum Máster | Eduardo Román Sánchez |
| Scrum Team | Adrian Ancajima Rodriguez<br>Eduardo Román Sánchez |

### 2.1 Backlog de Jira

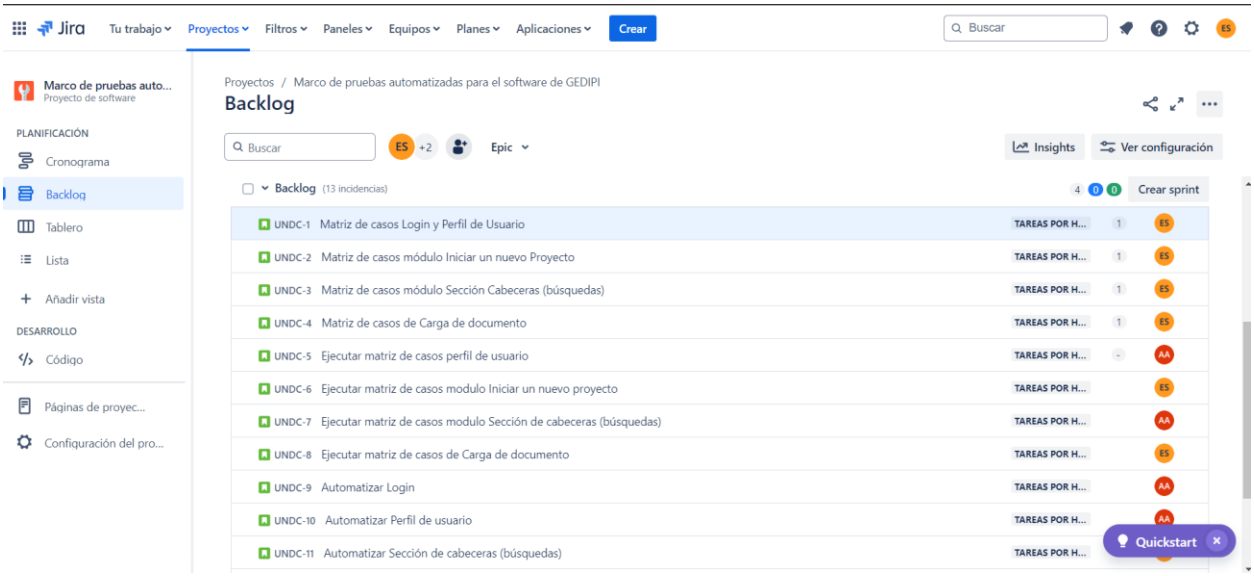



**Figura 3 -** Backlog del proyecto

La Figura 3, muestra el backlog del proyecto "Marco de pruebas automatizadas para el software de GEDIPI" gestionado mediante JIRA. A continuación, se detallan las tareas

listadas en el backlog:

- **UNDC-1: Matriz de casos Login y Perfil de Usuario**: Esta tarea se centra en la creación de una matriz de pruebas para las funcionalidades de inicio de sesión y perfil de usuario.
- **UNDC-2: Matriz de casos módulo Iniciar un nuevo Proyecto**: Desarrollo de casos de prueba para la funcionalidad de iniciar un nuevo proyecto.
- **UNDC-3: Matriz de casos módulo Sección Cabeceras (búsquedas)**: Pruebas relacionadas con la búsqueda en la sección de cabeceras.
- **UNDC-4: Matriz de casos de Carga de documento**: Pruebas para la funcionalidad de carga de documentos.
- **UNDC-5: Ejecutar matriz de casos perfil de usuario**: Ejecución de pruebas para el perfil de usuario.
- **UNDC-6: Ejecutar matriz de casos módulo Iniciar un nuevo proyecto**: Ejecución de pruebas para la creación de nuevos proyectos.
- **UNDC-7: Ejecutar matriz de casos módulo Sección de cabeceras (búsquedas)**: Ejecución de pruebas para la sección de cabeceras.
- **UNDC-8: Ejecutar matriz de casos de Carga de documento**: Ejecución de pruebas para la carga de documentos.
- **UNDC-9: Automatizar Login**: Automatización de pruebas para el inicio de sesión.
- **UNDC-10: Automatizar Perfil de usuario**: Automatización de pruebas para el perfil de usuario.
- **UNDC-11: Automatizar Sección de cabeceras (búsquedas)**: Automatización de pruebas para la sección de cabeceras.

### 2.2. Gestión del Backlog

El backlog es una herramienta esencial para la planificación y el seguimiento de las tareas del proyecto. Cada tarea en el backlog se asigna a un miembro del equipo y se monitorea con frecuencia para garantizar que se cumplan los plazos y se mantenga la calidad del trabajo. Se puede ver el progreso de cada tarea, encontrar bloqueos y reprioritizar las tareas según las necesidades del proyecto con JIRA.

## 3. Fase de planificación e implementación

Para crear los Sprints, se creó un cronograma de actividades que reflejaba la implementación de los cuatro Sprints, que se llevaron a cabo durante una semana cada uno.

### a. Sprint 1: Realización de matriz de casos de pruebas

El Sprint 1 se desarrolló durante una semana, que duro cinco días laborales, en el que se debe presentar la lista de casos de prueba. Para que el sistema web GEDIPI funcione correctamente, es necesario prestar atención al detalle y comprender claramente los requisitos y funcionalidades del software. Estos pasos pueden mejorar la calidad del software y reducir la probabilidad de que haya defectos en el producto final.

Para crear la lista de casos de prueba, utilizaremos las características del sistema web y desarrollaremos una estrategia de prueba que se adapte al negocio del producto.

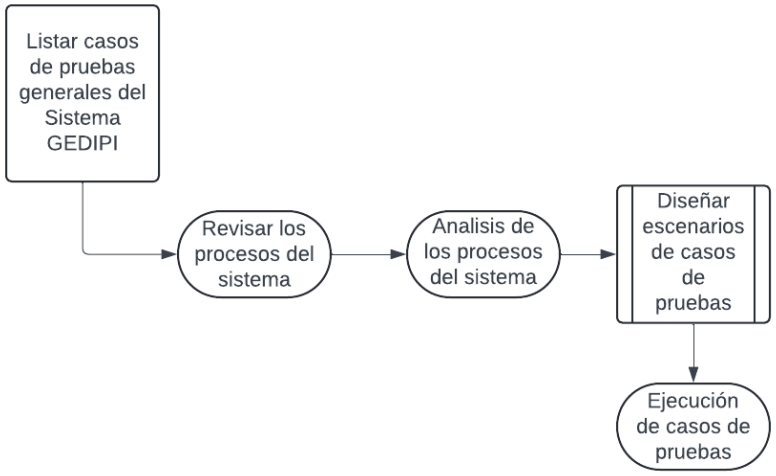



**Figura 4 -** Diagrama Gedipi

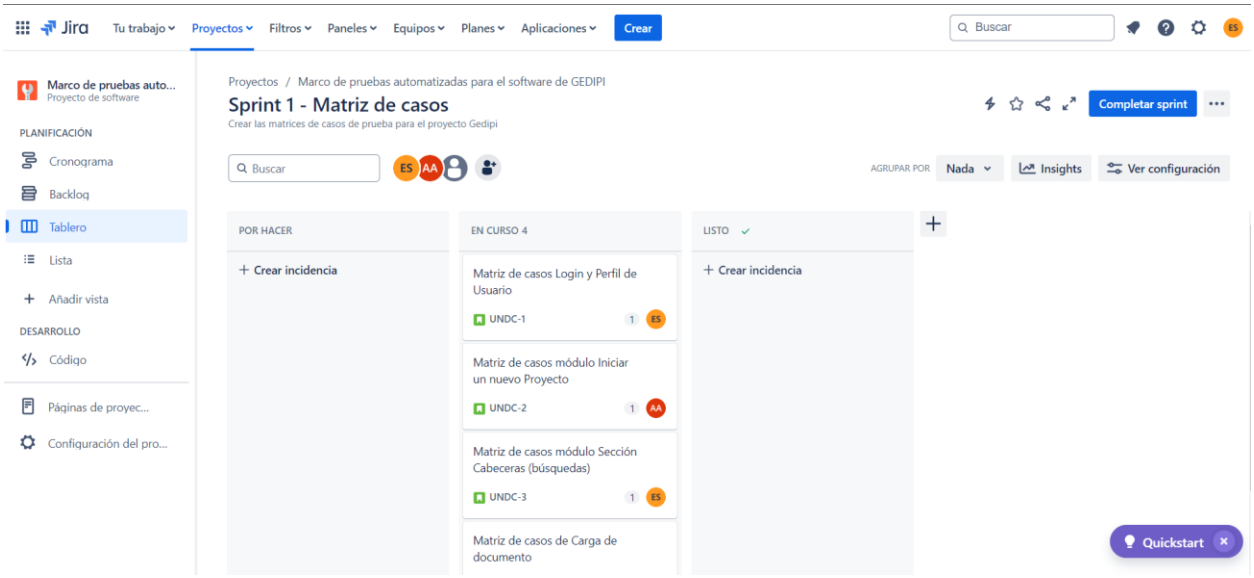



**Figura 5 -** Tablero Sprint 1

## Tabla XI - Tabla general matriz de casos

| Módulos | N° | Descripción de casos generales | Validación |
|---|---|---|---|
| Login | 1 | Ingresar al sistema | Este módulo es el acceso al sistema web |
| Iniciar un proyecto | 2 | Crear un proyecto Pregrado | Este módulo es para iniciar un proyecto |
| | 3 | Crear un proyecto Postgrado | |
| Mis proyectos – Documentos | 4 | Añadir documento | Este módulo es para añadir, generar y |
| | 5 | Generar formato N°1 | |

| | | | |
|---|---|---|---|
| | 6 | Descargar fut | descargar documentos |
| Búsqueda | 7 | Buscar por divisiones o categorías | En este módulo se realizan las búsquedas, por categorías, investigador, proyectos |
| | 8 | Buscar por investigador | |
| | 9 | Buscar por proyecto | |
| Perfil de usuario | 10 | Editar información del usuario | En este módulo se edita información y cierra sesión del usuario |
| | 11 | Cambiar contraseña del usuario | |
| | 12 | Cerrar sesión | |

### i. Matriz de casos login y perfil de usuario

**Tabla XII - Matriz de casos login y perfil de usuario**

| Módulos | N° | Descripción de casos generales | Descripción de los escenarios |
|---|---|---|---|
| Login | 1 | Ingresar al sistema | Ingresar al sistema con usuario y contraseña correcta |
| | 2 | | Ingresar al sistema con usuario o contraseña incorrecta |
| Perfil de usuario | 3 | | |

| | | | |
|---|---|---|---|
| | | | |
| | | | |

### ii. Matriz de casos modulo Iniciar un nuevo proyecto

**Tabla XIII - Matriz de casos modulo Iniciar un nuevo proyecto**

| Módulos | Descripción de casos generales | N° | Descripción de los escenarios | Datos principales | Registro |
|---|---|---|---|---|---|
| Iniciar un proyecto | Crear un proyecto Pregrado | 1 | Iniciar modalidad - Bachiller | -Seleccionar Área de Investigación<br>-Seleccionar Línea de Investigación<br>-Seleccionar Sublínea de Investigación<br>-Seleccionar Tipo de Diseño<br>-Seleccionar Tipo de Investigación<br>-Ingresar Titulo de la Investigación<br>-Ingresar Resumen<br>-Ingresar Objetivo General<br>-Ingresar Objetivos Específicos<br>-Ingresar Duración en meses<br>-Ingresar Código de publicación | -Ingresar Autores<br>-Ingresar Asesor<br>-Ingresa Co Asesor |
| | | 2 | Iniciar modalidad - Título Profesional | Registrar información por defecto y continuar | - |
| | | 3 | Iniciar modalidad - Semillero de Investigación | -Seleccionar Área de Investigación<br>-Seleccionar Línea de Investigación<br>-Seleccionar Sublínea de Investigación<br>-Seleccionar Tipo de Diseño<br>-Seleccionar Tipo de Investigación | - |

| | | | | | |
|---|---|---|---|---|---|
| | | | | -Ingresar Titulo de la Investigación<br>-Ingresar Resumen<br>-Ingresar Objetivo General<br>-Ingresar Objetivos Específicos | |
| | Crear un proyecto Postgrado | 4 | Iniciar modalidad - Segunda Especialidad | -Seleccionar Área de Investigación<br>-Seleccionar Línea de Investigación<br>-Seleccionar Sublínea de Investigación<br>-Seleccionar Tipo de Diseño<br>-Seleccionar Tipo de Investigación<br>-Ingresar Titulo de la Investigación<br>-Ingresar Resumen<br>-Ingresar Objetivo General<br>-Ingresar Objetivos Específicos<br>-Ingresar Duración en meses | -Ingresar Autores<br>-Ingresar Asesor<br>-Ingresa Co Asesor |
| | | 5 | Iniciar modalidad - Maestría | -Seleccionar Área de Investigación<br>-Seleccionar Línea de Investigación<br>-Seleccionar Sublínea de Investigación<br>-Seleccionar Tipo de Diseño<br>-Seleccionar Tipo de Investigación<br>-Ingresar Titulo de la Investigación<br>-Ingresar Resumen<br>-Ingresar Duración en meses | |

| | | | | | |
|---|---|---|---|---|---|
| | | 6 | Iniciar modalidad - Doctorado | Registrar información por defecto y continuar | |
| | | 7 | Iniciar modalidad - Diplomado | -Seleccionar Área de Investigación<br>-Seleccionar Línea de Investigación<br>-Seleccionar Sublínea de Investigación<br>-Seleccionar Tipo de Diseño<br>-Seleccionar Tipo de Investigación<br>-Ingresar Titulo de la Investigación<br>-Ingresar Resumen<br>-Ingresar Objetivo General<br>-Ingresar Objetivos Específicos<br>-Ingresar Duración en meses | -Ingresar Autores<br>-Ingresa Co Asesor |

### iii. Matriz de casos modulo Sección de cabeceras (búsquedas)

**Tabla XIV - Matriz de casos modulo Divisiones**

| Módulos | Descripción de casos generales | N° | Descripción de los escenarios | Validación – Área | Validación – Líneas | Validación – SubLíneas | Búsqueda de Docentes Especializados |
|---|---|---|---|---|---|---|---|
| Búsqueda | Buscar por divisiones o categorías | 1 | Búsqueda por área | -Ingresar nombre<br>-Seleccionar facultad<br>-Seleccionar botón (buscar)<br>-Seleccionar botón (Ver Todas) | -Seleccionar botón (Ver Todas) | -Seleccionar botón (Ver Todas) | -Seleccionar botón (ver) |
| | | 2 | Búsqueda por línea | | -Seleccionar facultad<br>-Seleccionar botón (buscar) | -No tenga SubLíneas | - |
| | | 3 | Búsqueda por sublínea | | -Ingresar nombre completo | - | - |

**Tabla XV - Matriz de casos modulo Investigador**

| Módulos | Descripción de casos generales | N° | Descripción de los escenarios | Validación | Detalles del investigador |
|---|---|---|---|---|---|
| Búsqueda | Buscar por investigador | 4 | Búsqueda por Investigador | -Ingresar Nombre<br>-Seleccionar tipo de investigador (alumno)<br>-Seleccionar el Centro de Investigación<br>-Seleccionar botón (buscar)<br>-Seleccionar alumno - botón (Ver Más) | -Validación de los datos personales |
| | | 5 | Búsqueda por Investigador | -Ingresar Nombre<br>-Seleccionar tipo de investigador (docente)<br>-Seleccionar el Centro de Investigación<br>-Seleccionar botón (buscar)<br>-Seleccionar docente - botón (Ver Más) | -Validación de los datos personales |
| | | 6 | Búsqueda por Investigador | -Ingresar Nombre completo<br>-Seleccionar tipo de investigador (todos)<br>-Seleccionar el Centro de Investigación(todos)<br>-Seleccionar botón (buscar) | - |

**Tabla XVI - Matriz de casos modulo Proyecto**

| Módulos | Descripción de casos generales | N° | Descripción de los escenarios | Validación | Validación del proyecto | Detalles del proyecto |
|---|---|---|---|---|---|---|
| Búsqueda | Buscar por proyecto | 7 | Búsqueda por proyecto | -Ingresar Titulo o código<br>-Seleccionar Tipo de investigación<br>-Seleccionar Unidad de Investigación<br>-Seleccionar Área de Investigación<br>-Seleccionar Línea de Investigación<br>-Seleccionar Sublínea de Investigación<br>-Seleccionar botón (buscar) | -N°<br>-Título<br>-Centro de Investigación<br>-Fecha<br>-Código<br>-Detalle, seleccionar botón | -Validación detalles del proyecto |
| | | 8 | Búsqueda por proyecto | -Seleccionar Tipo de investigación(todos)<br>-Seleccionar Unidad de Investigación(todos) | -N°<br>-Título<br>-Centro de Investigación<br>-Fecha<br>-Código<br>-Detalle, seleccionar botón | -Validación detalles del proyecto |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | 9 | Búsqueda por proyecto | -Ingresar Titulo o código completo<br>-Seleccionar Tipo de investigación<br>-Seleccionar Unidad de Investigación<br>-Seleccionar Área de Investigación<br>-Seleccionar Línea de Investigación | -N°<br>-Título<br>-Centro de Investigación<br>-Fecha<br>-Código<br>-Detalle, seleccionar botón | -Validación detalles del proyecto |

### iv. Matriz de casos de Carga de documento

**Tabla XVII - Matriz de casos de Carga de documento**

| Módulos | Descripción de casos generales | N° | Descripción de los escenarios | Validación | Detalles del proyecto | Detalles de la fase | Añadir documento |
|---|---|---|---|---|---|---|---|
| Mis proyectos – Documentos | Añadir documento | 1 | -Ingresar al proyecto correspondiente | -Validación detalles mis proyecto<br>-Seleccionar proyecto correspondiente<br>-Seleccionar botón | -Validación detalles del proyecto<br>-Validación datos del tramite<br>-Validación autores del proyecto<br>-Validación asesores del proyecto<br>-Validación co asesores del proyecto<br>-Validación fases del proyecto<br>-Seleccionar botón (Detalles) | -Seleccionar botón (añadir documento) | -Ingresar titulo<br>-Ingresar número del documento<br>-Subir documento<br>-Guardar documento |
| | Añadir documento | 2 | -Ingresar al proyecto | -Validación detalles mis | -Validación detalles del proyecto | -Seleccionar botón | -Subir documento<br>-Guardar |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | correspondiente | proyecto<br>-Seleccionar proyecto correspondiente<br>-Seleccionar botón | -Validación datos del tramite<br>-Validación autores del proyecto<br>-Validación asesores del proyecto<br>-Validación co asesores del proyecto<br>-Validación fases del proyecto<br>-Seleccionar botón (Detalles) | (añadir documento) | documento |
| | Generar formato N°1 | 3 | -Ingresar al proyecto correspondiente | -Validación detalles mis proyecto | -Validación detalles del proyecto<br>-Validación datos del tramite<br>-Validación autores del proyecto<br>-Validación asesores del proyecto<br>-Validación co asesores del proyecto | -Seleccionar botón (Generar formato n° 1) | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | -Validación fases del proyecto<br>-Seleccionar botón (Detalles) | | |
| | Descargar fut | 4 | -Ingresar al proyecto correspondiente | -Seleccionar proyecto correspondiente | -Validación detalles del proyecto<br>-Validación datos del tramite<br>-Validación autores del proyecto<br>-Validación asesores del proyecto<br>-Validación co asesores del proyecto<br>-Validación fases del proyecto<br>-Seleccionar botón (Detalles) | -Seleccionar botón (Descargar fut) | |

## b. Sprint 2: Ejecución de pruebas manuales y detección de errores (bugs)

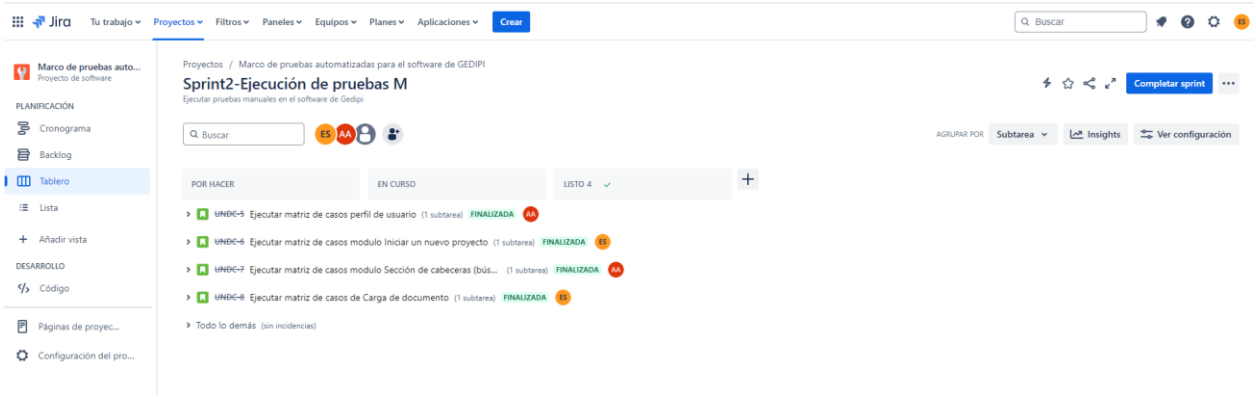



**Figura 6** - Tablero Sprint 2

### i. Ejecutar matriz de casos perfil de usuario

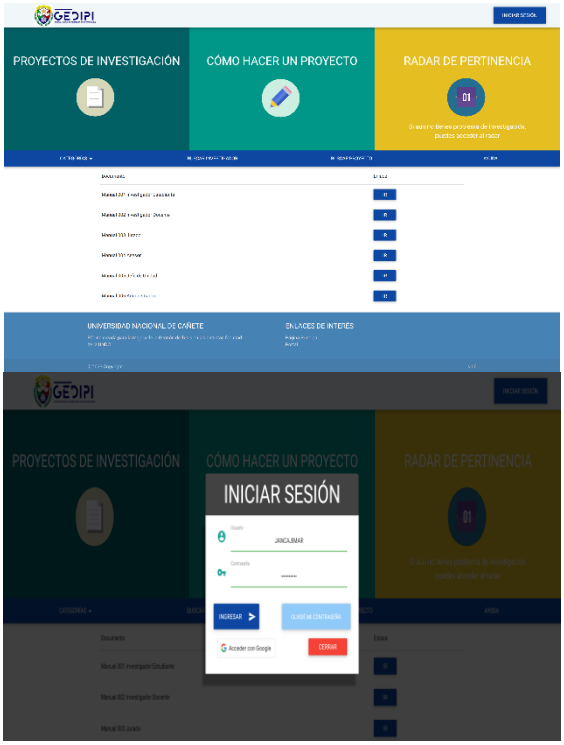



**Figura 7** - Iniciar Sesión

En la figura 7, se muestra la pantalla de inicio de sesión de la página web de GEDIPI (Gestión Digital de Procesos de Investigación). Se va a automatizar el módulo de inicio de sesión de la página web de GEDIPI, donde los usuarios ingresan sus credenciales para acceder al sistema.




**Figura 8 -** Editar perfil

En la figura 8, se muestra la pantalla de "Mi Perfil" de la página web de GEDIPI. Se va a automatizar el módulo de "Mi Perfil" de la página web de GEDIPI, donde los usuarios pueden ver y editar sus datos personales.




**Figura 9 -** Perfil editado

Figura 9, para automatizar el formulario de "Mi Perfil" mostrado en la figura de la página web de GEDIPI, seguiremos un proceso estructurado para verificar que todas las funciones y campos del formulario operen correctamente.




**Figura 10 -** Editar perfil - cambiar contraseña

En la figura 10, se muestra la pantalla de "Cambio de Contraseña" de la página web de GEDIPI. Se va a automatizar el módulo de "Cambio de Contraseña" de la página web de GEDIPI, donde los usuarios pueden actualizar su contraseña.

### ii. Ejecutar matriz de casos modulo Iniciar un nuevo proyecto

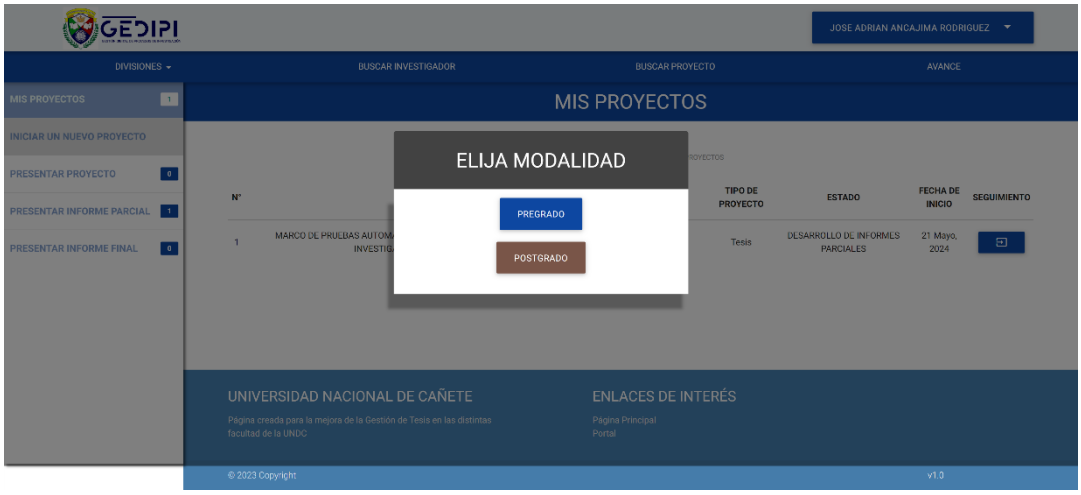



**Figura 11 –** Modalidad

En la figura 11, se muestra la pantalla de "Iniciar un Nuevo Proyecto" de la página web de GEDIPI. Se va a automatizar el módulo de "Iniciar un Nuevo Proyecto" de la página web de GEDIPI, donde los usuarios pueden seleccionar la modalidad de su proyecto de investigación.

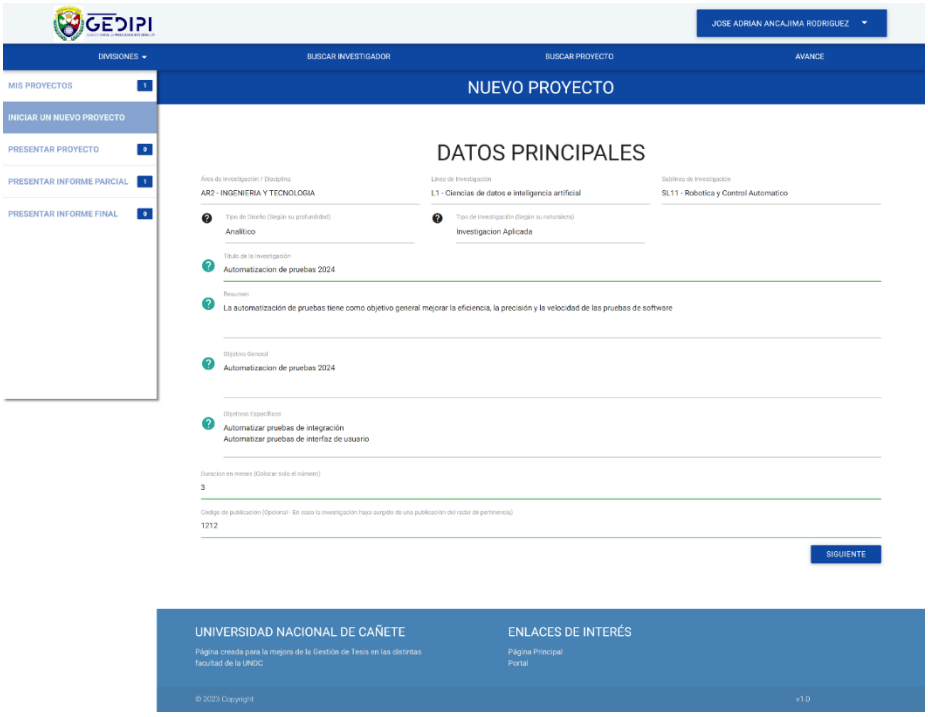



**Figura 12 -** Ingresar datos del proyecto

En la figura 12, se muestra el formulario para "Nuevo Proyecto" de la página web de GEDIPI (Gestión Digital de Procesos de Investigación). Se va a automatizar el módulo de "Nuevo Proyecto" de la página web de GEDIPI, donde los usuarios pueden ingresar los datos principales de su nuevo proyecto de investigación. Muestra la pantalla del formulario para "Nuevo Proyecto" de la página web de GEDIPI. Se va a automatizar el módulo de "Nuevo Proyecto" de la página web de GEDIPI, donde los usuarios pueden ingresar los datos principales de su nuevo proyecto de investigación

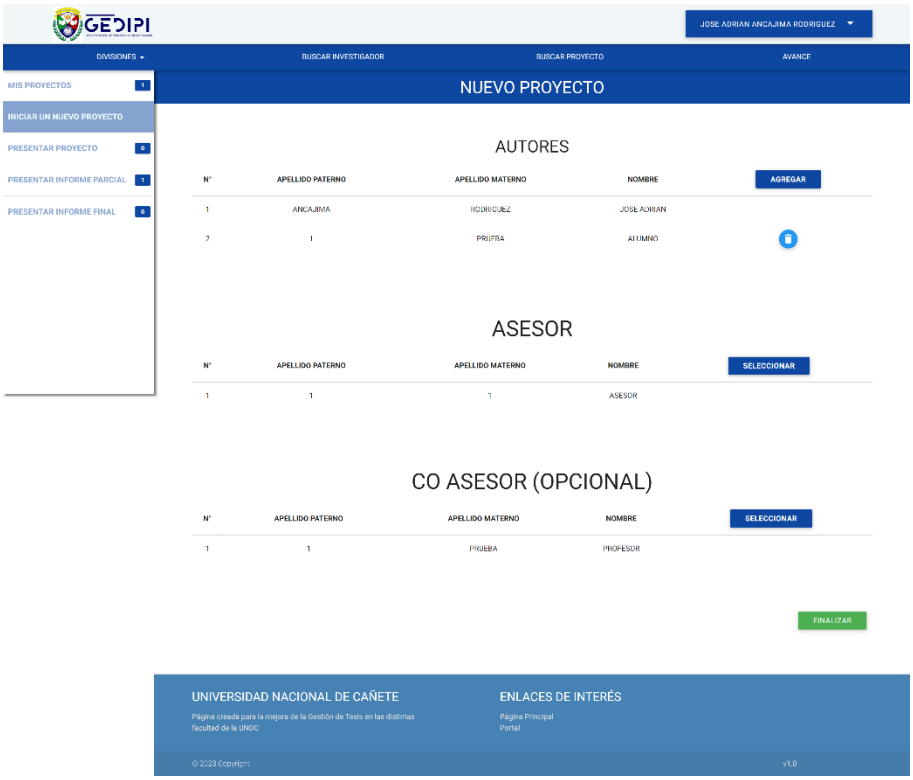



**Figura 13 -** Agregar autores, asesor, coasesor

En la figura 13 se muestra la sección extendida de "Autores, Asesor y Co Asesor (Opcional)" del formulario para "Nuevo Proyecto" de la página web de GEDIPI. Se va a automatizar el módulo de "Autores, Asesor y Co Asesor" de la página web de GEDIPI, donde los usuarios pueden agregar autores al proyecto, seleccionar un asesor y un co-asesor opcional.




**Figura 14 -** Registro correctamente del proyecto

En la figura 14, se muestra la pantalla de confirmación de "Registro Completado" de la página web de GEDIPI. Se va a automatizar el módulo de "Registro Completado" de la página web de GEDIPI, donde se notifica al usuario que su nuevo proyecto ha sido registrado correctamente.

### iii. Ejecutar matriz de casos modulo Sección de cabeceras (búsquedas)




**Figura 15 -** Búsqueda por divisiones (área, línea, sublinea)

En la figura 15, se muestra la pantalla de búsqueda por divisiones de la página web de GEDIPI. Se va a automatizar el módulo de "Búsqueda por Divisiones" de la página web de GEDIPI, donde los usuarios pueden buscar proyectos o información por áreas, líneas y sublíneas de investigación.

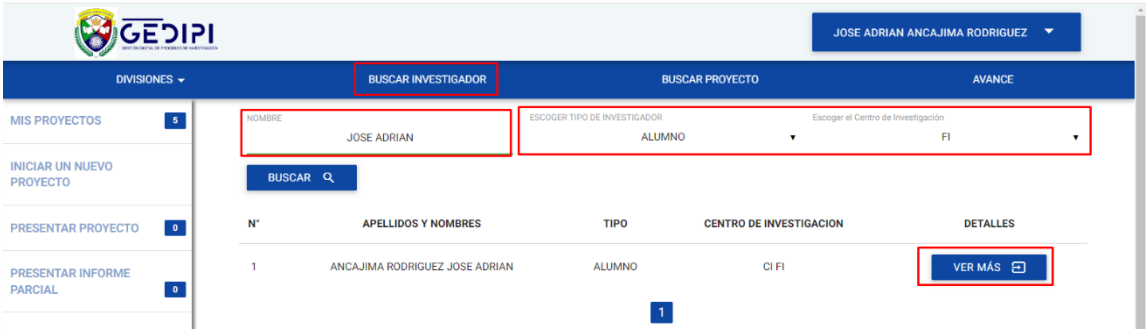



**Figura 16 -** Búsqueda por investigador

En la figura 16, se muestra la pantalla de búsqueda por investigador de la página web de GEDIPI. Se va a automatizar el módulo de "Búsqueda por Investigador" de la página web de GEDIPI, donde los usuarios pueden buscar investigadores específicos en el sistema.




**Figura 17 -** Búsqueda por proyecto

En la figura 17, se muestra la pantalla de búsqueda por proyecto de la página web de GEDIPI. Se va a automatizar el módulo de "Buscar Proyecto" de la página web de GEDIPI, donde los usuarios pueden buscar proyectos específicos en el sistema.

### iv. Ejecutar matriz de casos de Carga de documento

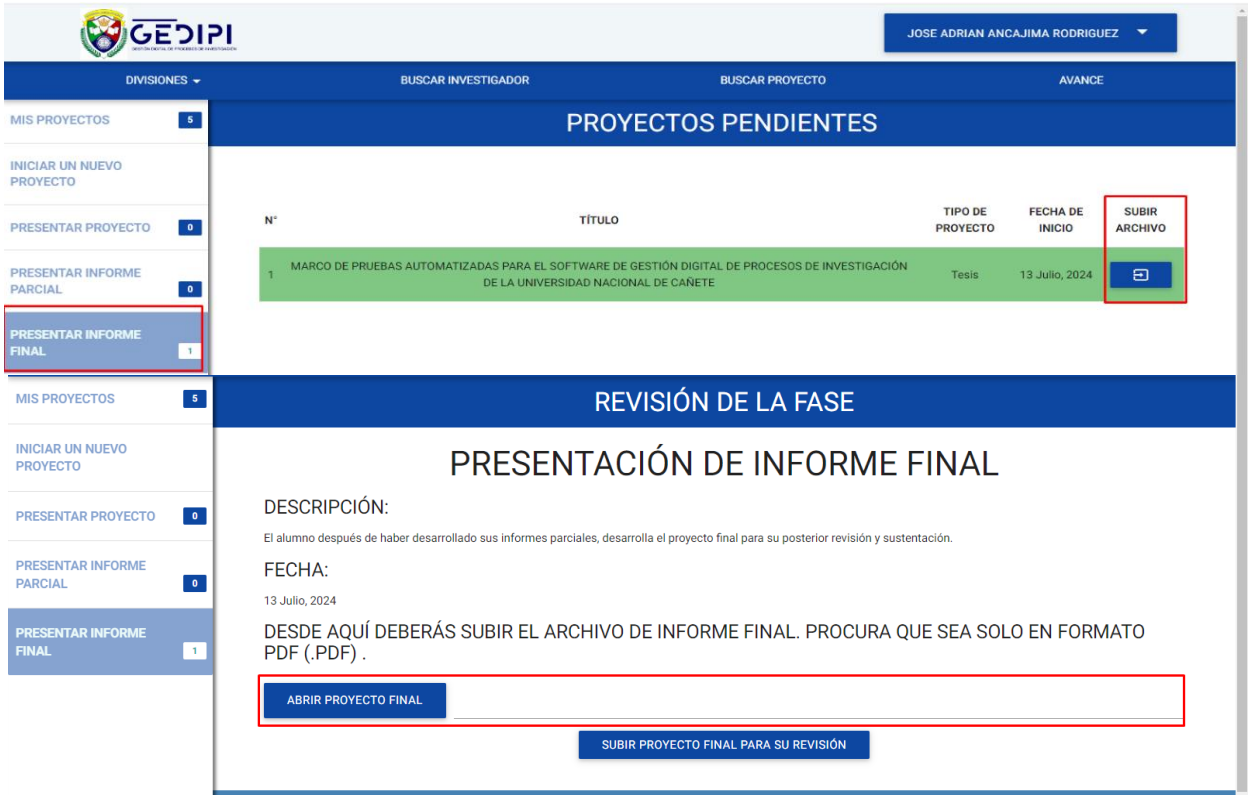



**Figura 18 -** Carga de documento

En la figura 18, se muestra la pantalla de carga de documentos en la sección de "Presentar Informe Final" de la página web de GEDIPI. Se va a automatizar el módulo de "Carga de Documento" para la presentación del informe final de la página web de GEDIPI, donde los usuarios pueden subir sus informes finales para revisión.

### c. Sprint 3: Pruebas automatizadas primera parte

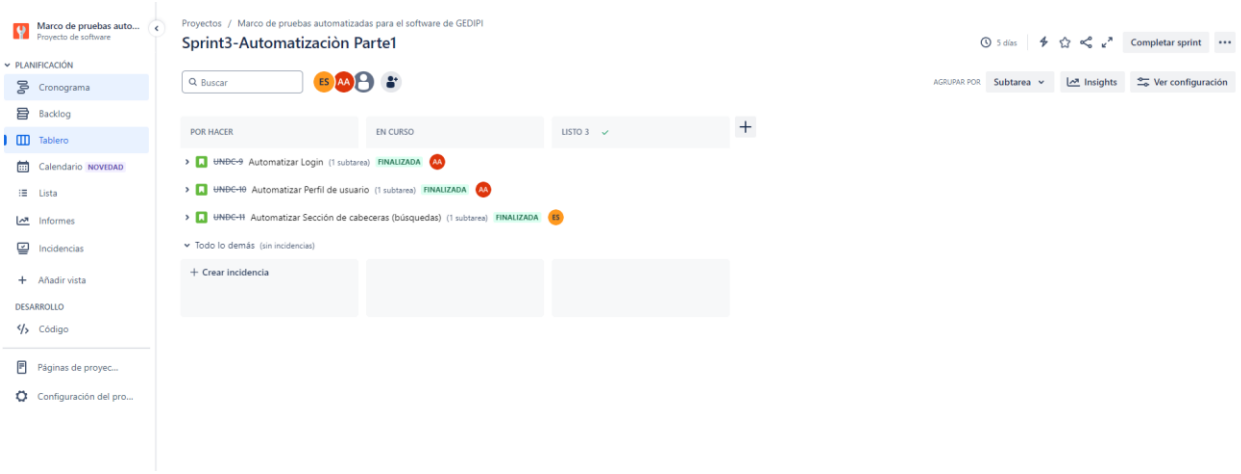



**Figura 19 -** Tablero Sprint 3

#### i. Estructura del proyecto

Para la estructura de este proyecto esta basando en:

- Crear el Feature
- Redactar los casos de pruebas en lenguaje Gherkin dentro del feature
- Definir StepDefinitions
- Definir Pages
- Definir Util, Hooks, Runner

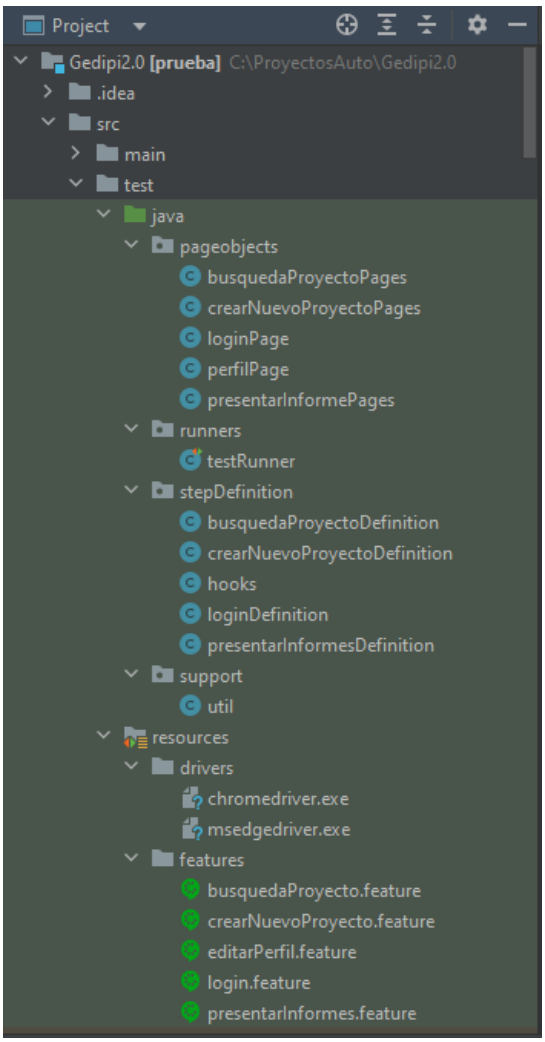



**Figura 20 -** Estructura del proyecto

En la figura 20, se muestra la estructura que se utilizara para la automatización de los módulos principales de GEDIPI, dividiéndose por carpetas y clases con el modelo de patrón page object model, Selenium webdriver, gherkin.

### ii. Automatizar Login

Para desarrollar la automatización del módulo Login, es necesario seguir los pasos siguientes.

- Crear el Feature login
- Redactar los casos de pruebas en lenguaje Gherkin dentro del feature
- Definir StepDefinitions
- Definir Pages

- Definir Util, Hooks, Runner

```
# language: es
@LOGINGEDIPI
Característica: Funcionalidad del login gedipi
  Escenario: Ingresar login
    Dado que la web este operativa
    Cuando inicia sesion
    E ingresa el usuario "JANCAJIMAR"
    E ingresa la contrasenia "Faqfaq456*"
    E ingresa usuario
```

**Figura 21 -** Feature Login

En la figura 21, se muestra un archivo de características escrito en el lenguaje Gherkin para la automatización de pruebas utilizando el marco de trabajo Cucumber. El escenario presentado se enfoca en la funcionalidad de inicio de sesión (login) del sistema GEDIPI.

loginPage.java | loginDefinition.java

```
package pageobjects;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.ExpectedConditions;
import support.util;
3 usages
public class loginPage extends util {
    2 usages
    @FindBy(xpath = "//a[@class='waves-effect waves-light btn-large']") protected WebElement lnksignin;
    2 usages
    @FindBy(xpath = "//input[@id='icon_prefix']") protected WebElement txtusuario;
    2 usages
    @FindBy(xpath = "//input[@id='icon_telephone']") protected WebElement txtcontrasenia;
    1 usage
    @FindBy(xpath = "//button[@class='btn-large waves-effect waves-light center col m4 s4']") protected WebElement btningresar;
    1 usage
    public loginPage() {
        PageFactory.initElements(driver, page: this);
    }

    1 usage
    public void clickSignIn(){
        wait.until(ExpectedConditions.elementToBeClickable(lnksignin));
        lnksignin.click();
    }
    1 usage
    public void IngresarUsuario(String texto){
        wait.until(ExpectedConditions.visibilityOf(txtusuario));
```

**Figura 22 -** Pages login

En la figura 22, se muestra una clase Java utilizada para automatizar la funcionalidad de inicio de sesión (login) de la página web de GEDIPI utilizando el marco de trabajo

Selenium. Esta clase es parte del patrón de diseño Page Object Model (POM), que es una práctica común para estructurar y organizar pruebas automatizadas de manera eficiente.

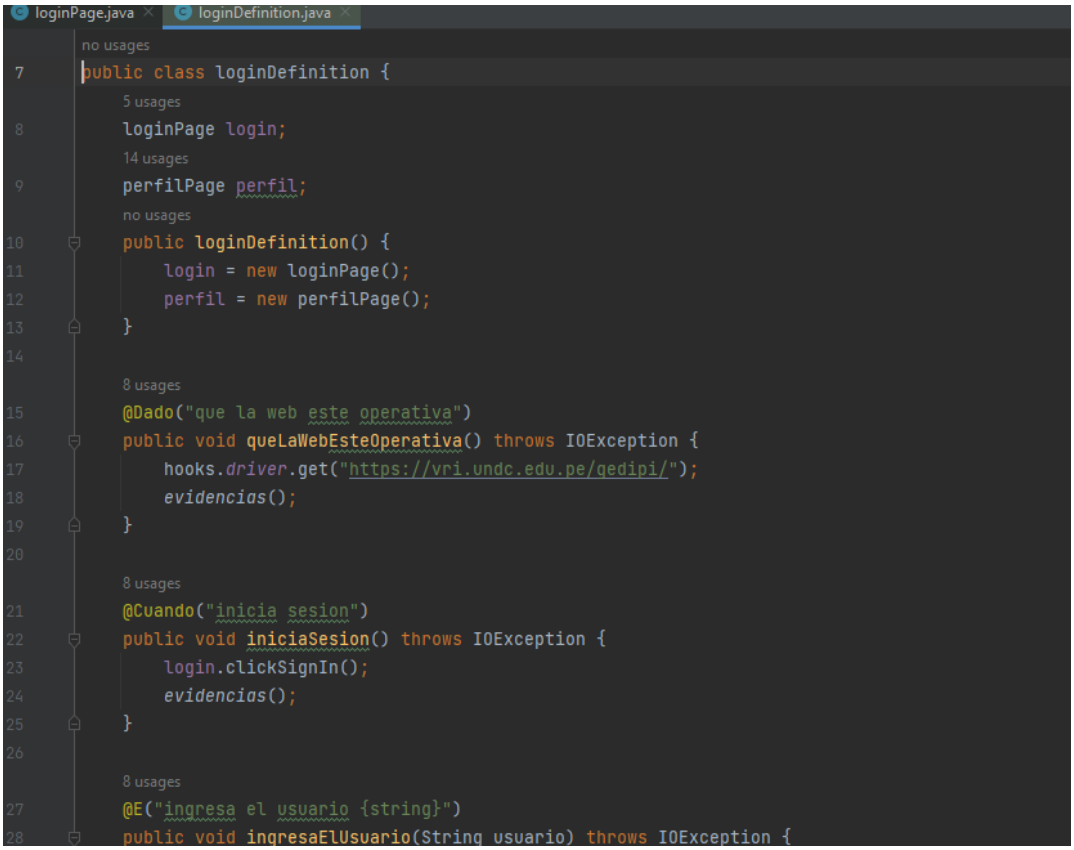



**Figura 23 -** Definitions Login

En la figura 23, se muestra una clase Java llamada loginDefinition utilizada para definir los pasos de las pruebas automatizadas de la funcionalidad de inicio de sesión (login) de la página web de GEDIPI utilizando el marco de trabajo Cucumber. Esta clase contiene la implementación de los pasos definidos en el archivo de características Gherkin, que fueron descritos anteriormente.

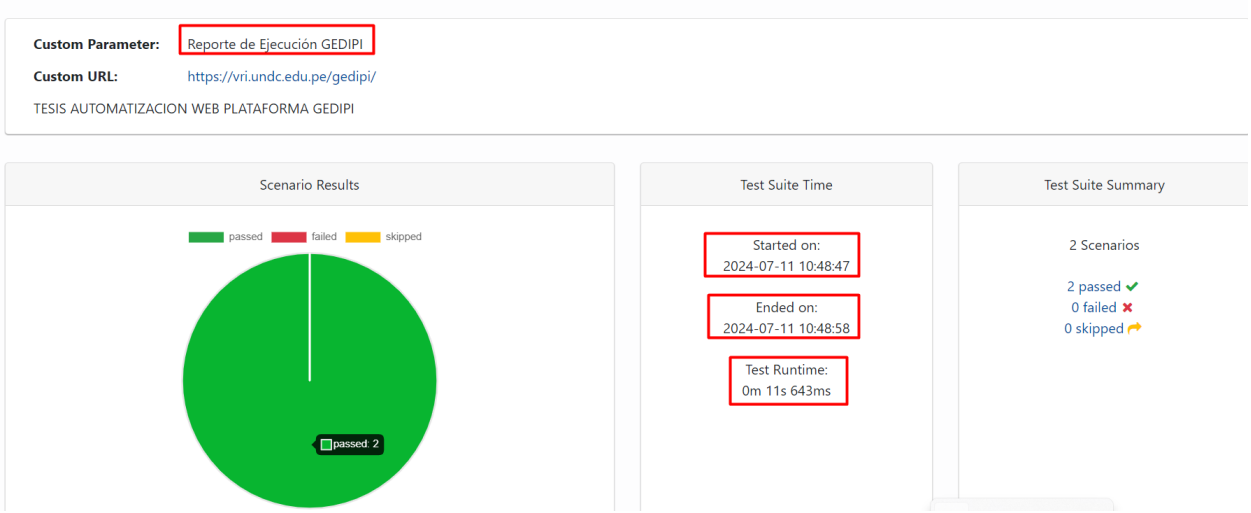



**Figura 24 -** Reportes tiempo de ejecución del login

En la figura 24, se muestra un reporte de ejecución de pruebas automatizadas para la plataforma web GEDIPI. El reporte proporciona una visión general del tiempo de ejecución de las pruebas y sus resultados.

### iii. Automatizar edición de perfil, contraseña

Para desarrollar la automatización del módulo editar perfil y cambiar contraseña, es necesario seguir los pasos siguientes.

- Crear el Feature editar perfil y cambiar contraseña
- Redactar los casos de pruebas en lenguaje Gherkin dentro del feature
- Definir StepDefinitions
- Definir Pages
- Definir Util, Hooks, Runner

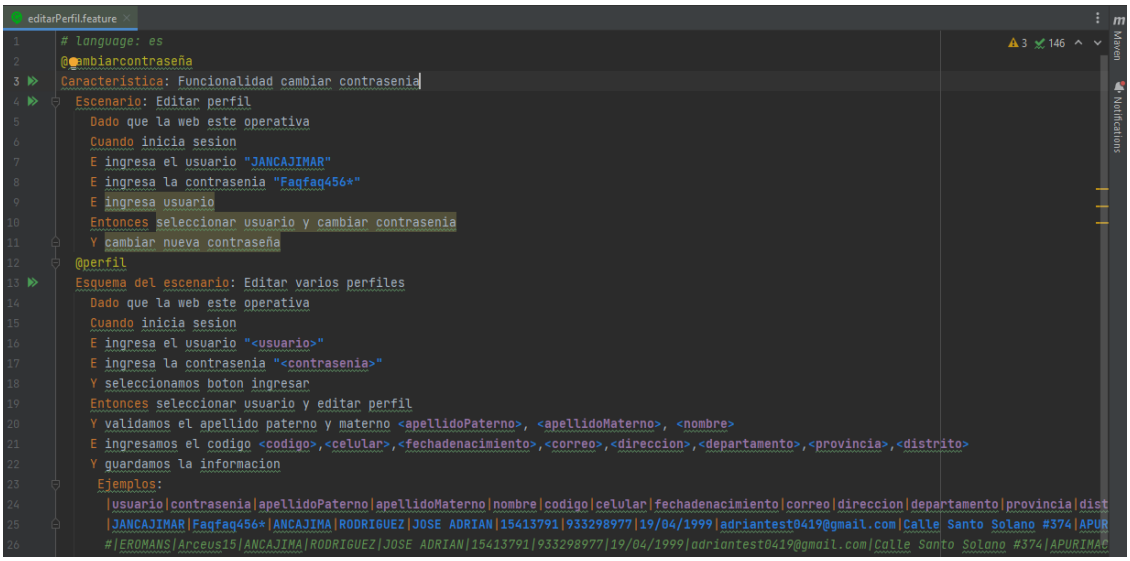



**Figura 25 -** Feature edición perfil

En la figura 25, se muestra un archivo de características escrito en el lenguaje Gherkin para la automatización de pruebas utilizando el marco de trabajo Cucumber. Este archivo se enfoca en la funcionalidad de cambiar la contraseña y la edición del perfil de usuario en la plataforma GEDIPI.

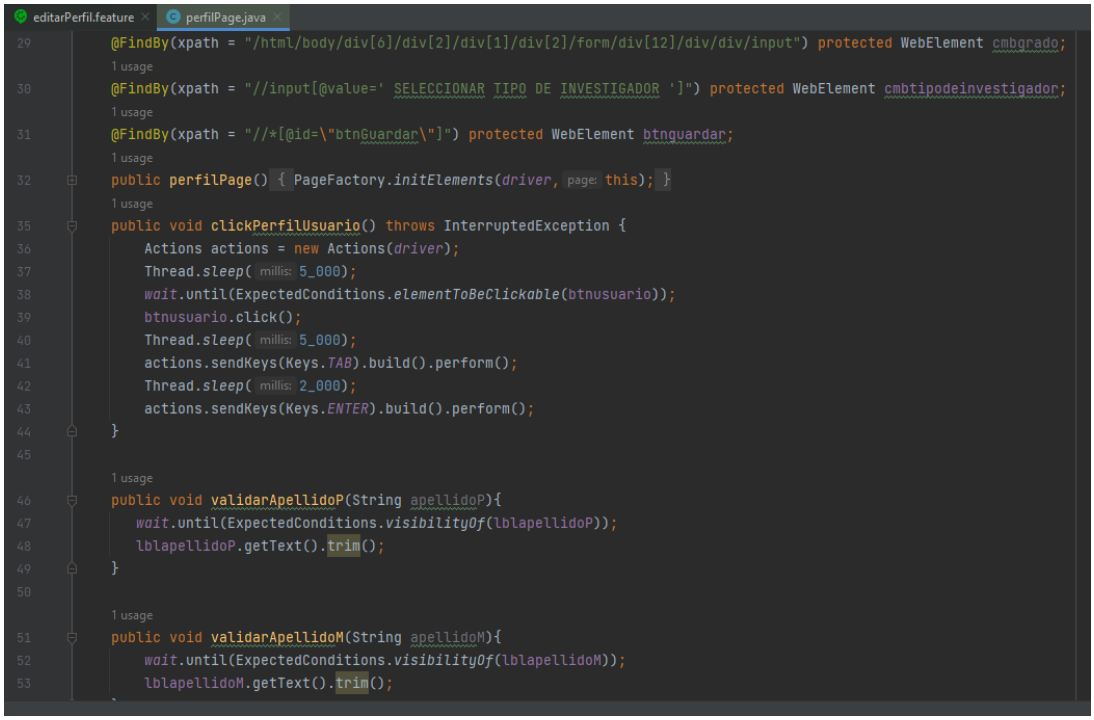



**Figura 26 -** Pages edición perfil

En la figura 26, se muestra una clase Java llamada perfilPage utilizada para automatizar la

funcionalidad de edición de perfil en la página web de GEDIPI utilizando el marco de trabajo Selenium. Esta clase es parte del patrón de diseño Page Object Model (POM), que es una práctica común para estructurar y organizar pruebas automatizadas de manera eficiente.

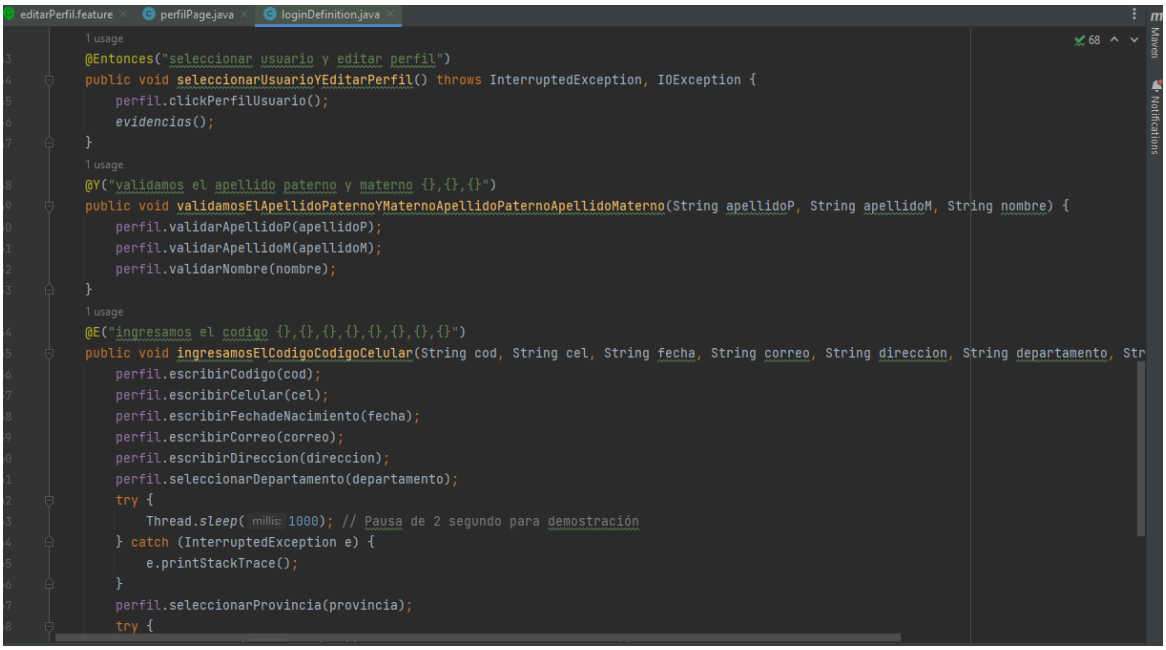



**Figura 27 -** Definitions edición perfil

En la figura 27, se muestra una clase Java utilizada para definir los pasos de las pruebas automatizadas de la funcionalidad de edición de perfil en la plataforma GEDIPI utilizando el marco de trabajo Cucumber. Esta clase contiene la implementación de los pasos definidos en el archivo de características Gherkin.

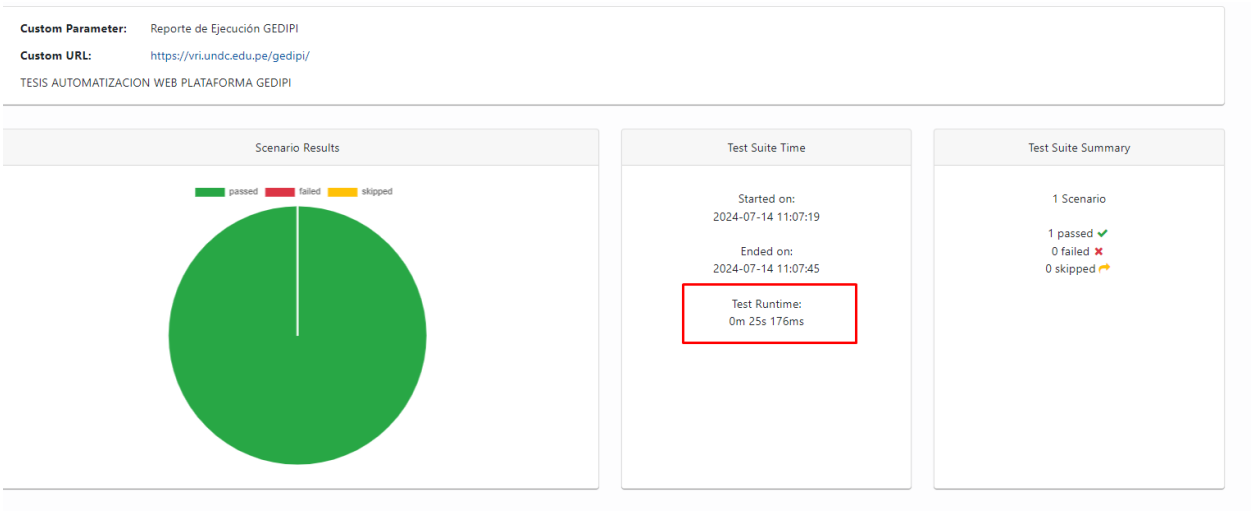



**Figura 28 -** Tiempo de ejecución prueba perfil

En la figura 28, se muestra un reporte de ejecución de pruebas automatizadas para la funcionalidad de edición de perfil en la plataforma web GEDIPI. El reporte proporciona una visión general del tiempo de ejecución de las pruebas y sus resultados.

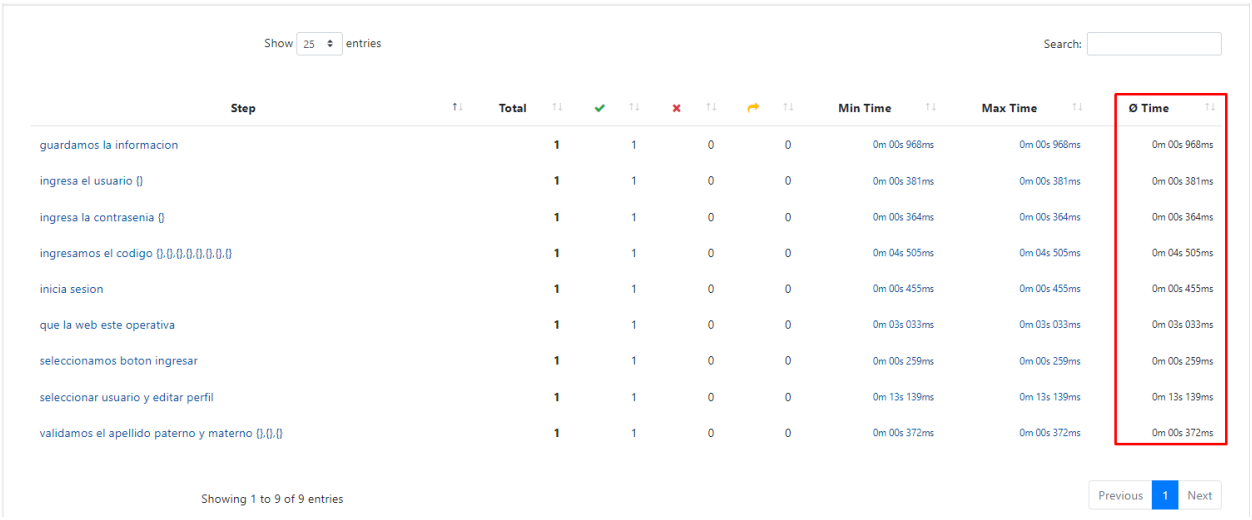



**Figura 29 -** Tiempo de ejecución de cada paso Perfil

En la figura 29, se muestra un reporte detallado del tiempo de ejecución de cada paso de la prueba automatizada de la funcionalidad de edición de perfil en la plataforma web de GEDIPI. Este reporte proporciona una visión granular de la duración de cada paso individual dentro del escenario de prueba.

### iv. Automatizar búsqueda de proyectos

Para desarrollar la automatización del módulo de búsqueda de divisiones, investigadores, proyectos ,es necesario seguir los pasos siguientes.

- Crear el Feature de búsqueda de las cabeceras
- Redactar los casos de pruebas en lenguaje Gherkin dentro del feature
- Definir StepDefinitions
- Definir Pages
- Definir Util, Hooks, Runner

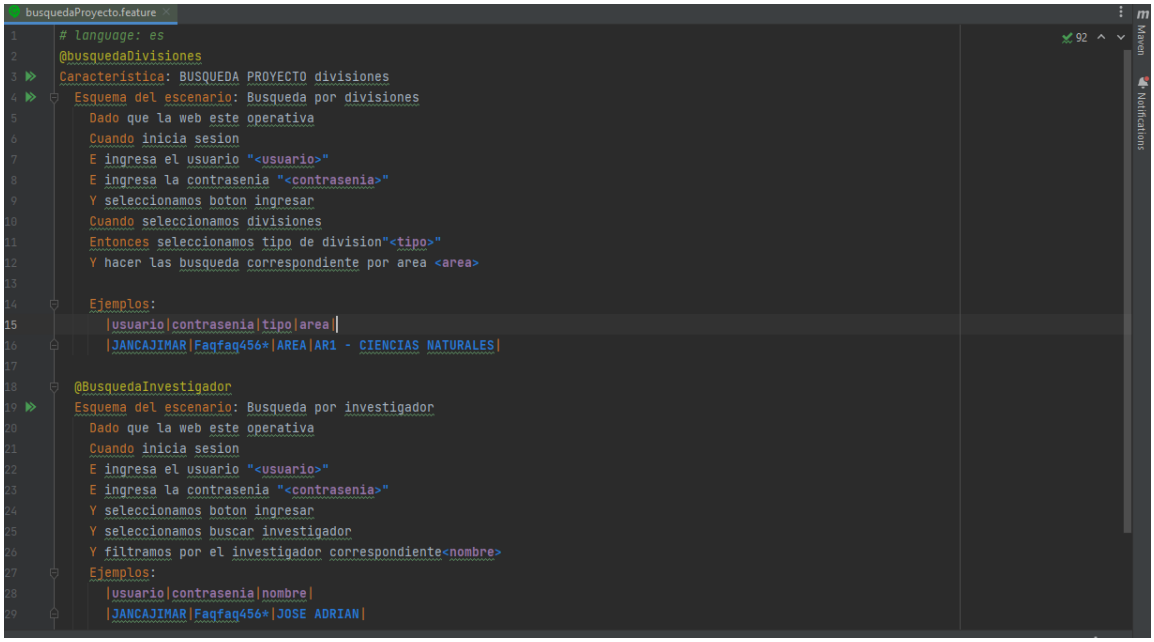



**Figura 30 -** Features busqueda

En la figura 30, se muestra un archivo de características escrito en Gherkin para la automatización de pruebas utilizando el marco de trabajo Cucumber. Este archivo se enfoca en la funcionalidad de búsqueda de proyectos en la plataforma GEDIPI, específicamente la búsqueda por divisiones y por investigador.

busquedaProyecto.feature busquedaProyectoPages.java

```
@FindBy(css = "#dropdownmenu") protected WebElement cmbtipodivsion;
1 usage
@FindBy(css= "#busqueda") protected WebElement txtingresarnombreareа;
4 usages
@FindBy(xpath= "/html/body/div[5]/div/div[3]") protected WebElement btnbuscarinvestigador;
1 usage
@FindBy(xpath= "//*[@id=\"busqueda\"]") protected WebElement txtingresarnombreinvestigador;
1 usage
@FindBy(xpath= "//*[@id=\"btnBuscar\"]") protected WebElement btnbuscarinvestigadorproyecto;
1 usage
@FindBy(xpath= "//*[@id=\"bodyInvestigadores\"]/tr/td[5]/a") protected WebElement btnbuscarinvestigadorproyectovermas;

1 usage
public busquedaProyectoPages() { PageFactory.initElements(driver, page: this); }
1 usage
public void clickDivisiones(){
    wait.until(ExpectedConditions.elementToBeClickable(btndivison));
    btndivison.click();
}
1 usage
public void seleccionarTipoDivison(String tipo){
    //ul[@id='dropdownmenu']//a[contains(text(), '%s')]
    try {
        // Hacer clic en el input para mostrar las opciones
        cmbtipodivsion.click();

        // Esperar a que las opciones estén visibles
        WebElement opcionArea = wait.until(ExpectedConditions.visibilityOfElementLocated(
                By.cssSelector("#dropdownmenu a[href*='tipo=" + tipo + "']")
```

**Figura 31 -** Pages búsqueda

En la figura 31, se muestra una clase Java llamada busquedaProyectoPages utilizada para automatizar la funcionalidad de búsqueda de proyectos en la página web de GEDIPI utilizando el marco de trabajo Selenium. Esta clase es parte del patrón de diseño Page Object Model (POM), que es una práctica común para estructurar y organizar pruebas automatizadas de manera eficiente.

busquedaProyecto.feature | busquedaProyectoDefinition.java | busquedaProyectoPages.java

```
25      @Y("hacer las busqueda correspondiente por area {}")
26      public void hacerLasBusquedaCorrespondientePorAreaArea(String area) {
27          busqueda.escribirNombreArea(area);
28      }
29
        1 usage
30      @Y("seleccionamos buscar investigador")
31      public void seleccionamosBuscarInvestigador() {
32          busqueda.clickinvestigador();
33      }
34
        1 usage
35      @Y("filtramos por el investigador correspondiente{}")
36      public void filtramosPorElInvestigadorCorrespondiente(String nombreInvestigador) {
37          busqueda.escribirNombreInvestigador(nombreInvestigador);
38          try {
39              Thread.sleep( millis: 3000); // Pausa de 1 segundo para demostración
40          } catch (InterruptedException e) {
41              e.printStackTrace();
42          }
43          busqueda.clickbuscarInvestigador();
44          try {
45              Thread.sleep( millis: 3000); // Pausa de 1 segundo para demostración
46          } catch (InterruptedException e) {
47              e.printStackTrace();
48          }
49          busqueda.clickbuscarInvestigadorVerMas();
50      }
51  }
```

**Figura 32 -** Definitions búsqueda

En la figura 32, se muestra una clase Java utilizada para definir los pasos de las pruebas automatizadas de la funcionalidad de búsqueda de proyectos en la plataforma GEDIPI utilizando el marco de trabajo Cucumber. Esta clase contiene la implementación de los pasos definidos en el archivo de características Gherkin.

### d. Sprint 4: Pruebas automatizadas segunda parte

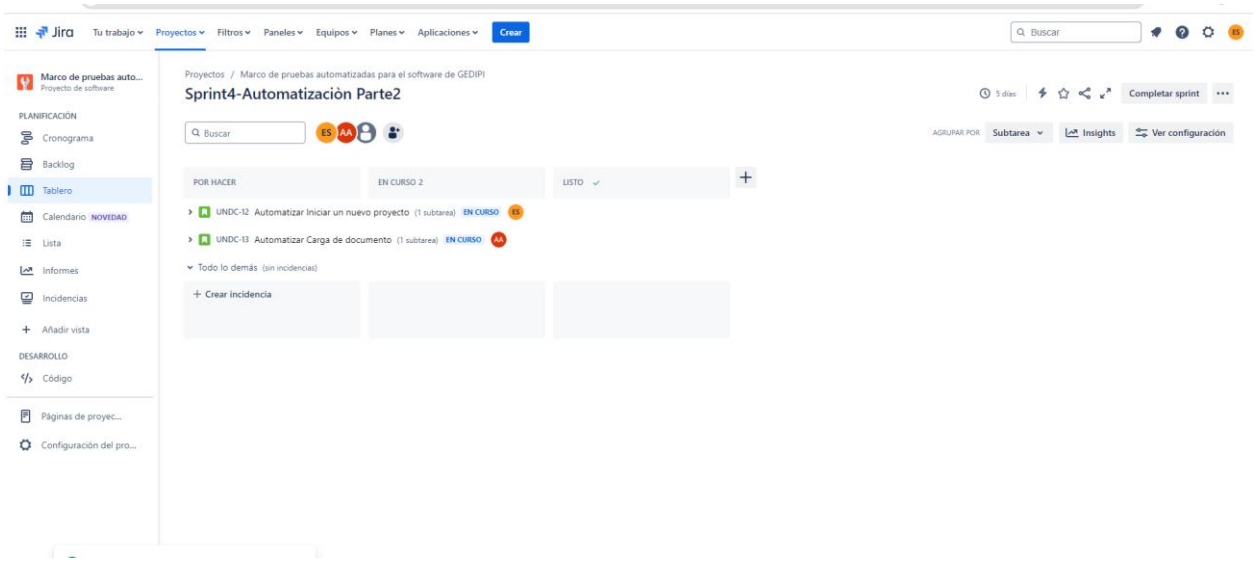


**Figura 33 -** Tablero Sprint 4

En la figura 33, se muestra un tablero de trabajo del Sprint 4 del proyecto de automatización del software GEDIPI, utilizando la herramienta Jira. Este tablero es una herramienta de gestión visual que permite hacer un seguimiento del progreso de las tareas del sprint.

### i. Automatizar Iniciar un nuevo proyecto

Para desarrollar la automatización del módulo Iniciar un proyecto, es necesario seguir los pasos siguientes.

- Crear el Feature Iniciar nuevo proyecto
- Redactar los casos de pruebas en lenguaje Gherkin
- Definir StepDefinitions
- Definir Pages
- Definir Util, Hooks, Runner

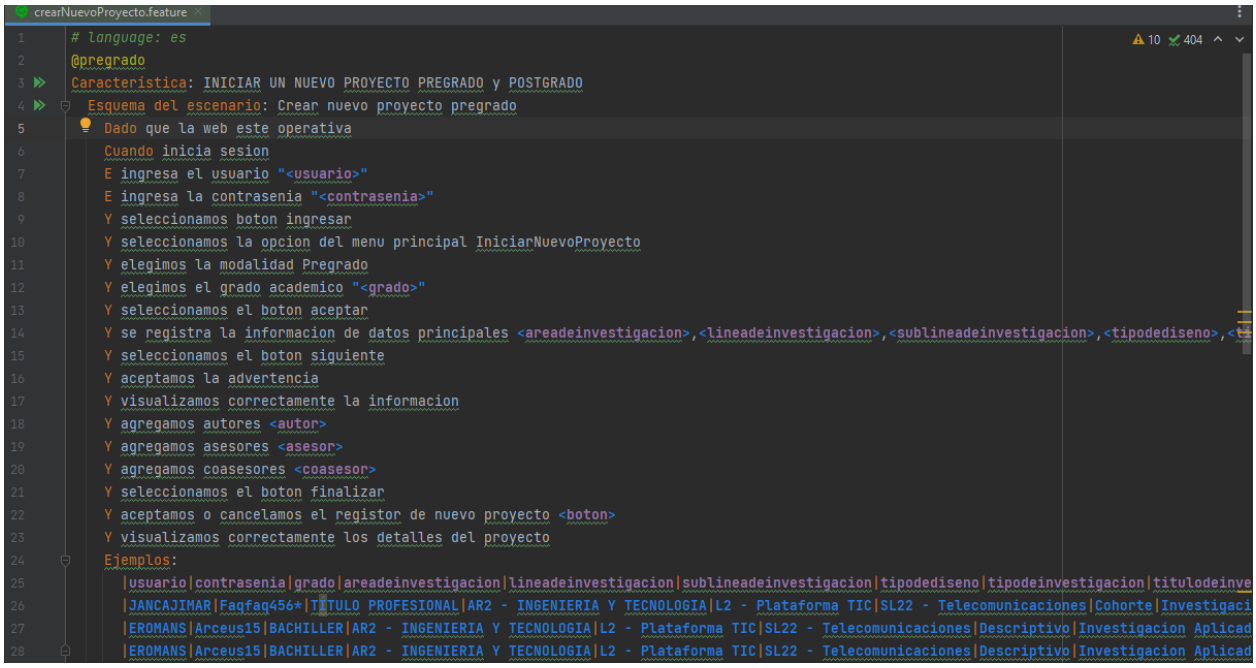



**Figura 34 -** Feature crear nuevo proyecto Pre grado

En la figura 34, se muestra un archivo de características escrito en Gherkin para la automatización de pruebas utilizando el marco de trabajo Cucumber. Este archivo se enfoca en la funcionalidad de crear un nuevo proyecto de pregrado en la plataforma web GEDIPI.

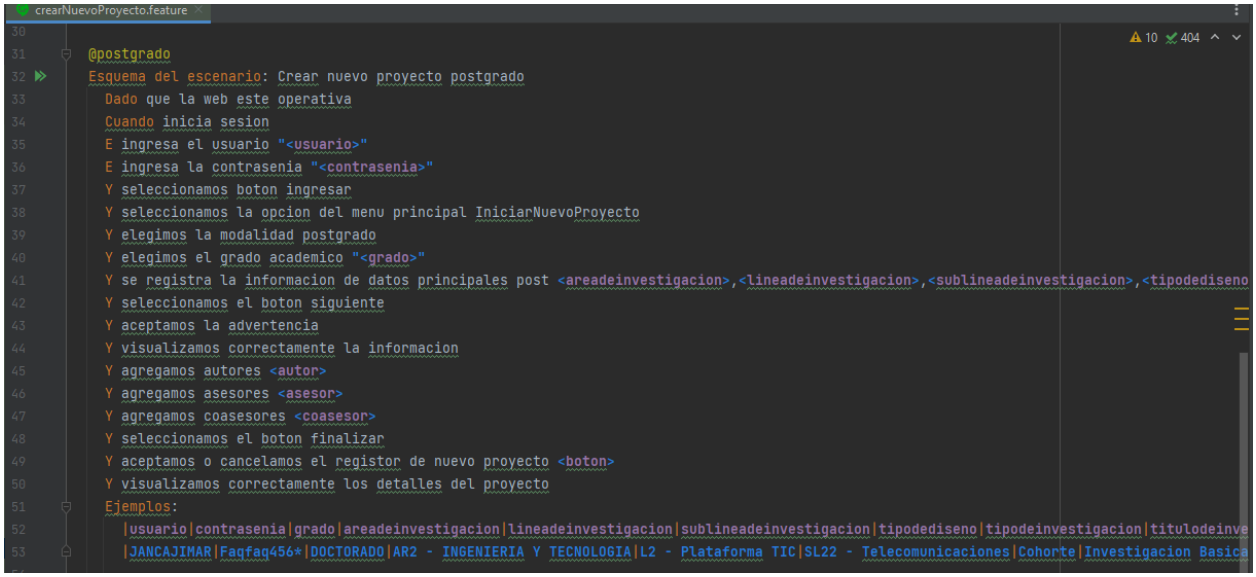



**Figura 35 -** Feature crear nuevo proyecto “Postgrado”

En la figura 35, se muestra un archivo de características escrito en Gherkin para la automatización de pruebas utilizando el marco de trabajo Cucumber. Este archivo se enfoca en la funcionalidad de crear un nuevo proyecto de postgrado en la plataforma web GEDIPI.

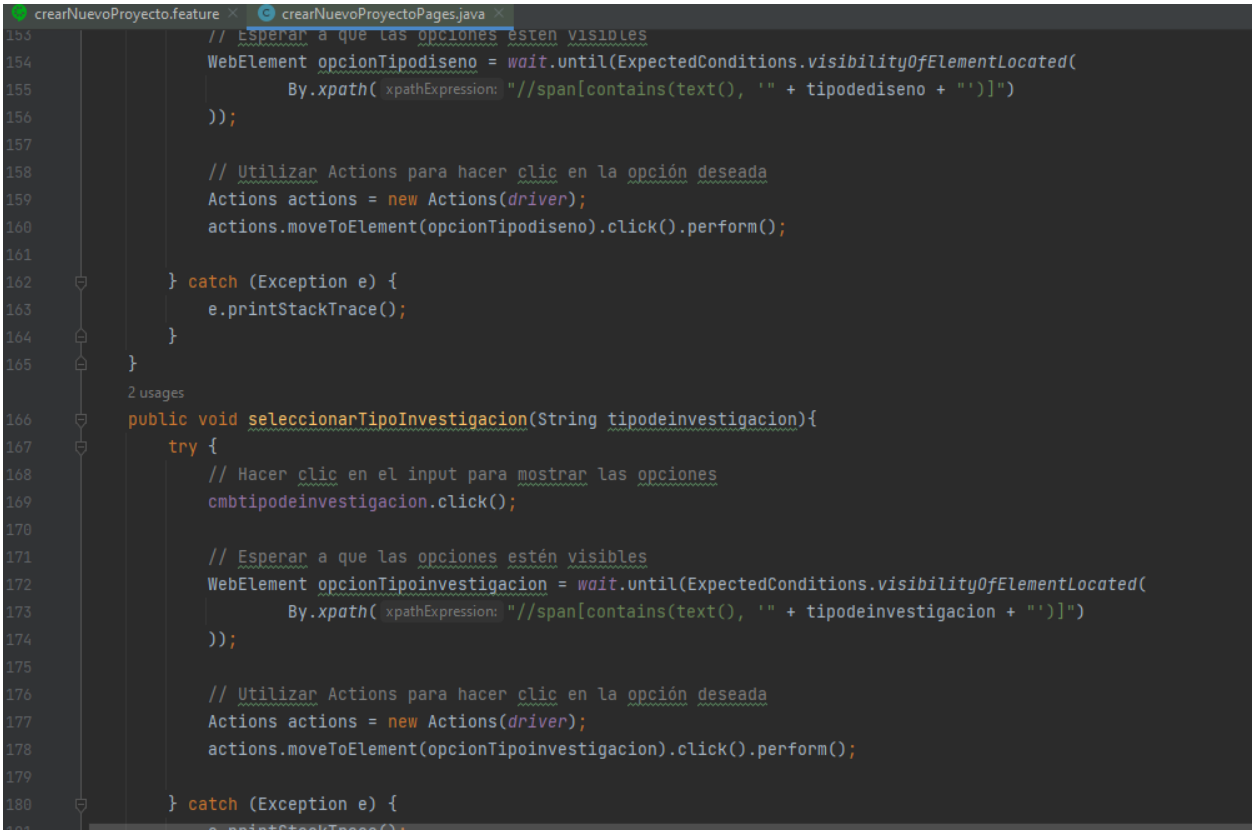



**Figura 36 –** Pages crear nuevo proyecto

En la figura 36, se muestra una clase Java que implementa el patrón Page Object Model (POM) para automatizar la funcionalidad de creación de un nuevo proyecto en la plataforma GEDIPI. Esta clase define métodos para interactuar con los elementos de la página web relacionados con la selección del tipo de diseño y tipo de investigación en el formulario de creación de un nuevo proyecto.

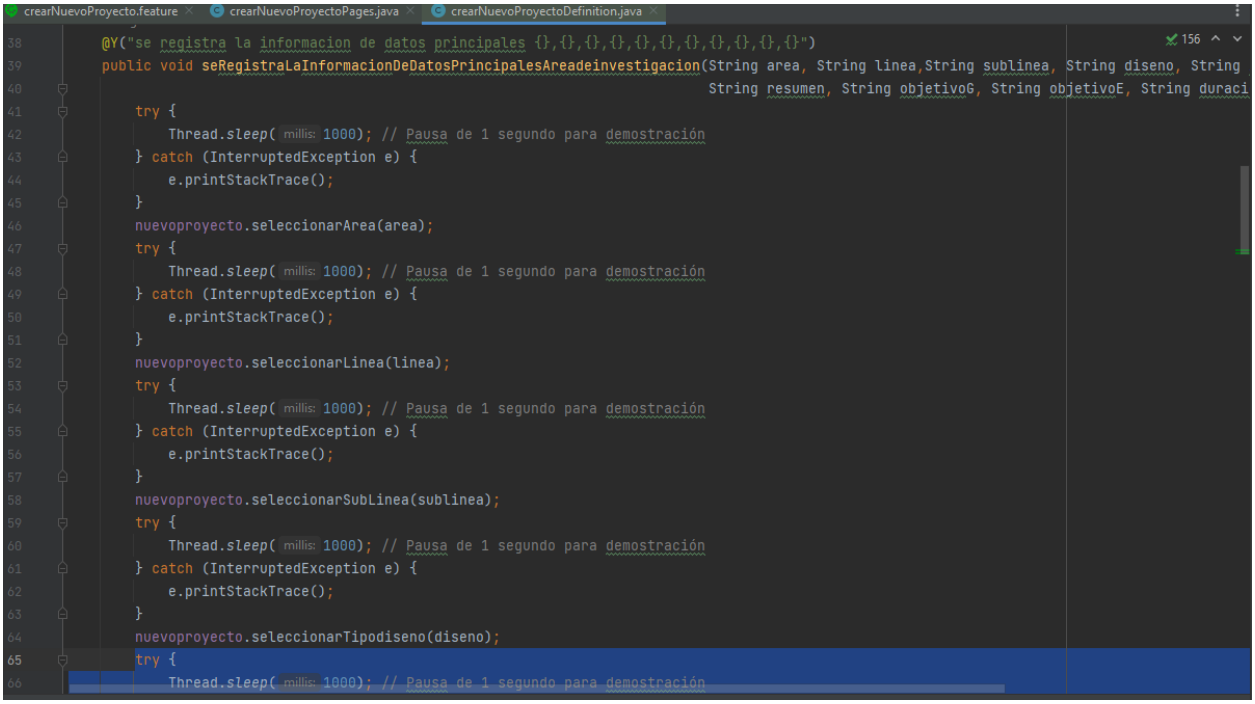



**Figura 37 –** Definitions crear nuevo proyecto

En la figura 37, se muestra una definición de pasos en Java para automatizar la creación de un nuevo proyecto en la plataforma GEDIPI utilizando el framework Cucumber. Este archivo está asociado a un archivo de características Gherkin que define el escenario de prueba.

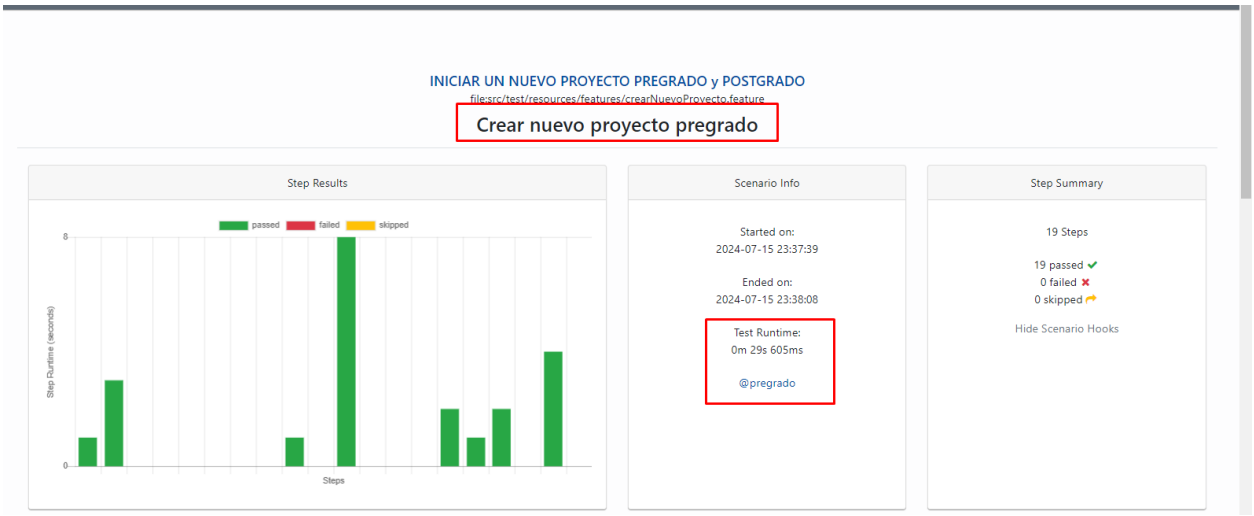
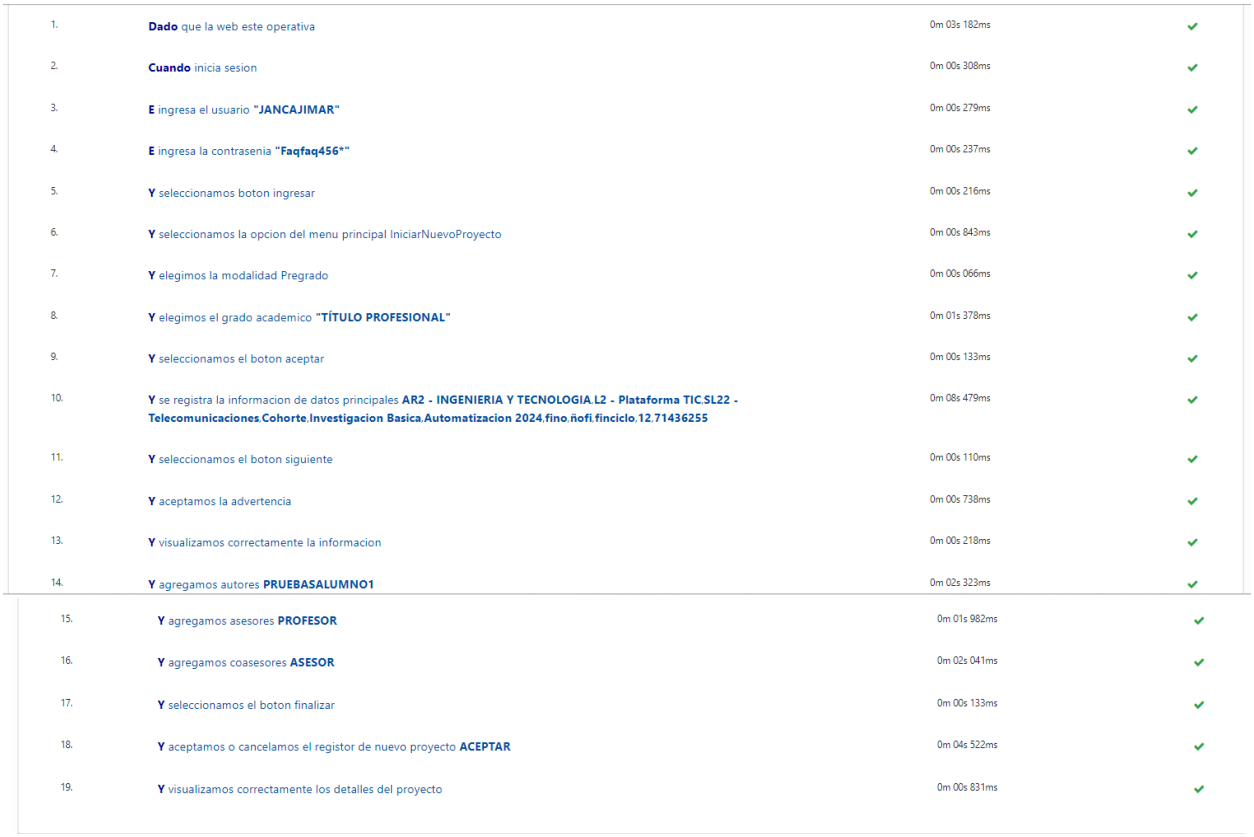



**Figura 38 -** Tiempo de ejecución de la prueba crear nuevo proyecto

En la figura 38. se presenta un informe de ejecución de una prueba automatizada para la funcionalidad de "Crear nuevo proyecto de pregrado" en la plataforma GEDIPI. El informe muestra información sobre el tiempo de ejecución de la prueba, los resultados de cada paso y un resumen del escenario.



| | | | |
|---|---|---|---|
| 1. | **Dado** que la web este operativa | 0m 03s 182ms | ✔ |
| 2. | **Cuando** inicia sesion | 0m 00s 308ms | ✔ |
| 3. | **E** ingresa el usuario **"JANCAJIMAR"** | 0m 00s 279ms | ✔ |
| 4. | **E** ingresa la contrasenia **"Faqfaq456*"** | 0m 00s 237ms | ✔ |
| 5. | **Y** seleccionamos boton ingresar | 0m 00s 216ms | ✔ |
| 6. | **Y** seleccionamos la opcion del menu principal IniciarNuevoProyecto | 0m 00s 843ms | ✔ |
| 7. | **Y** elegimos la modalidad Pregrado | 0m 00s 066ms | ✔ |
| 8. | **Y** elegimos el grado academico **"TÍTULO PROFESIONAL"** | 0m 01s 378ms | ✔ |
| 9. | **Y** seleccionamos el boton aceptar | 0m 00s 133ms | ✔ |
| 10. | **Y** se registra la informacion de datos principales **AR2 - INGENIERIA Y TECNOLOGIA,L2 - Plataforma TIC,SL22 - Telecomunicaciones,Cohorte,Investigacion Basica,Automatizacion 2024,fino,ñofi,finciclo,12,71436255** | 0m 08s 479ms | ✔ |
| 11. | **Y** seleccionamos el boton siguiente | 0m 00s 110ms | ✔ |
| 12. | **Y** aceptamos la advertencia | 0m 00s 738ms | ✔ |
| 13. | **Y** visualizamos correctamente la informacion | 0m 00s 218ms | ✔ |
| 14. | **Y** agregamos autores **PRUEBASALUMNO1** | 0m 02s 323ms | ✔ |
| 15. | **Y** agregamos asesores **PROFESOR** | 0m 01s 982ms | ✔ |
| 16. | **Y** agregamos coasesores **ASESOR** | 0m 02s 041ms | ✔ |
| 17. | **Y** seleccionamos el boton finalizar | 0m 00s 133ms | ✔ |
| 18. | **Y** aceptamos o cancelamos el registor de nuevo proyecto **ACEPTAR** | 0m 04s 522ms | ✔ |
| 19. | **Y** visualizamos correctamente los detalles del proyecto | 0m 00s 831ms | ✔ |

**Figura 39 -** Tiempo de ejecución de la prueba crear nuevo proyecto paso a paso

En la figura 39, se muestra un informe detallado del tiempo de ejecución de cada paso de la prueba automatizada para la funcionalidad de "Crear nuevo proyecto de pregrado" en la plataforma GEDIPI. Este informe desglosa el tiempo que toma cada paso, desde la inicialización de la prueba hasta la finalización del registro del nuevo proyecto.

### ii. Automatizar Carga de documento

Para desarrollar la automatización del módulo carga de documento, es necesario seguir los pasos siguientes.

- Crear el Feature cargar documentos
- Redactar los casos de pruebas en lenguaje Gherkin
- Definir StepDefinitions
- Definir Pages
- Definir Util, Hooks, Runner

```
presentarInformes.feature
1   # language: es
2   @subirinformefinal
3   Característica: PRESENTAR INFORME FINAL - SUBIR DOCUMENTO
4     Esquema del escenario: Subir proyecto final
5       Dado que la web este operativa
6       Cuando inicia sesion
7       E ingresa el usuario "<usuario>"
8       E ingresa la contrasenia "<contrasenia>"
9       Y seleccionamos boton ingresar
10      Y seleccionamos presentar informe final
11      Y subimos el archivo "<rutaArchivo>"
12      Ejemplos:
13        | usuario    | contrasenia  | rutaArchivo                                |
14        | JANCAJIMAR | Faqfaq456*   | C:/Users/el_pi/Downloads/archivoPrueba.jpeg |
```

**Figura 40 -** Features carga de documento

En la figura 40, se presenta el código de una característica (feature) en Gherkin para automatizar la funcionalidad de subir el informe final en la plataforma GEDIPI. El código detalla el escenario "Subir proyecto final", que especifica los pasos necesarios para realizar esta acción.

```
presentarInformes.feature   presentarInformePages.java
}

1 usage
public void clickSubirArchivo(){
    wait.until(ExpectedConditions.elementToBeClickable(btnsubirarchivofinal));
    btnsubirarchivofinal.click();
}

1 usage
public void clickAceptarInformacion(){
    wait.until(ExpectedConditions.elementToBeClickable(btnaceptararchivoInformacion));
    btnaceptararchivoInformacion.click();
}

1 usage
public void subirArchivo(String rutaArchivo) {
    try {
        // Utiliza el WebDriverWait definido en la clase util
        WebElement inputArchivo = wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath( xpathExpression: "//input[@name='archivo']"));
        inputArchivo.sendKeys(rutaArchivo);
    } catch (TimeoutException e) {
        System.err.println("Timeout al intentar encontrar y interactuar con el elemento 'inputArchivo'.");
        e.printStackTrace();
        // Puedes agregar más manejo de excepciones según sea necesario
    }
```

**Figura 41 -** Pages carga de documento

En la figura 41, se muestra el código de una clase de Java que implementa las acciones

necesarias para automatizar la funcionalidad de carga de documentos en la plataforma GEDIPI. Esta clase define métodos que interactúan con los elementos de la interfaz de usuario para completar el proceso de subir un archivo.

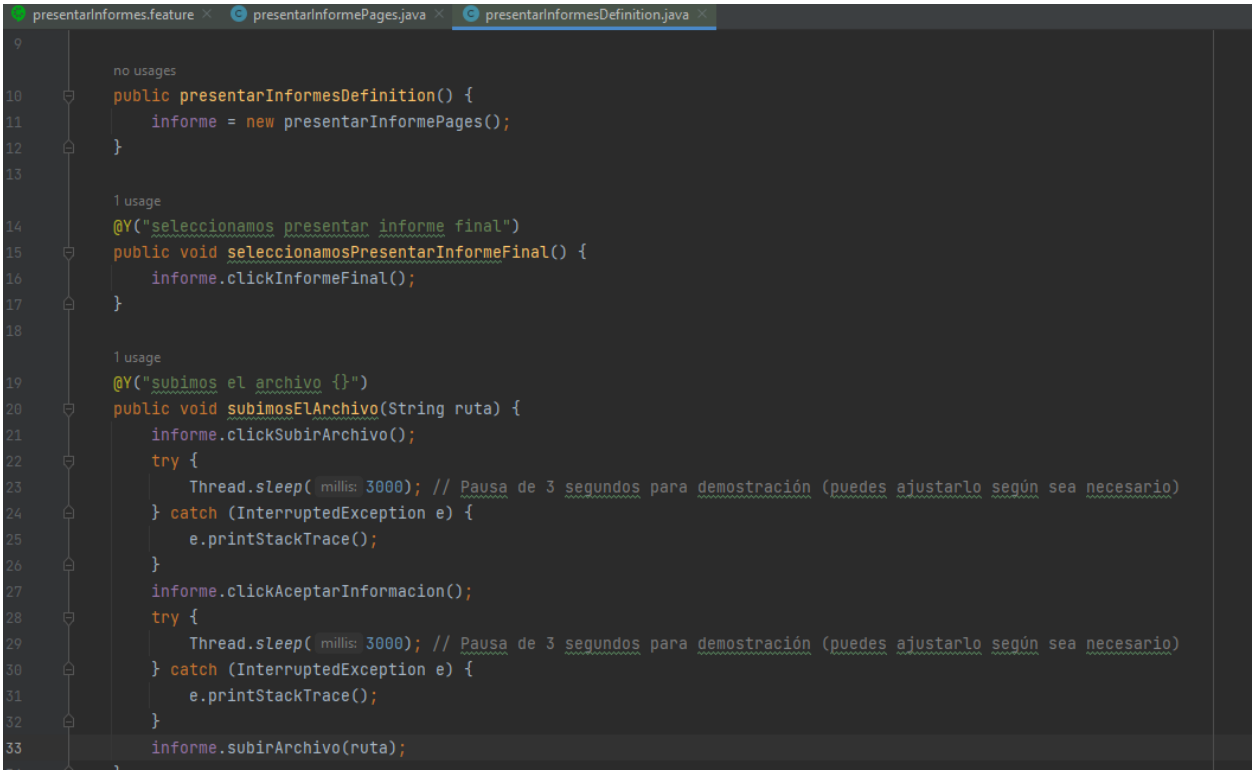



**Figura 42 -** Definitions subir documento

En la figura 42, se muestra el código de la clase presentarInformesDefinition en Java, que define los pasos necesarios para automatizar la funcionalidad de subir un documento como informe final en la plataforma GEDIPI. La clase utiliza anotaciones de Cucumber para enlazar los pasos definidos en Gherkin con los métodos de Java.

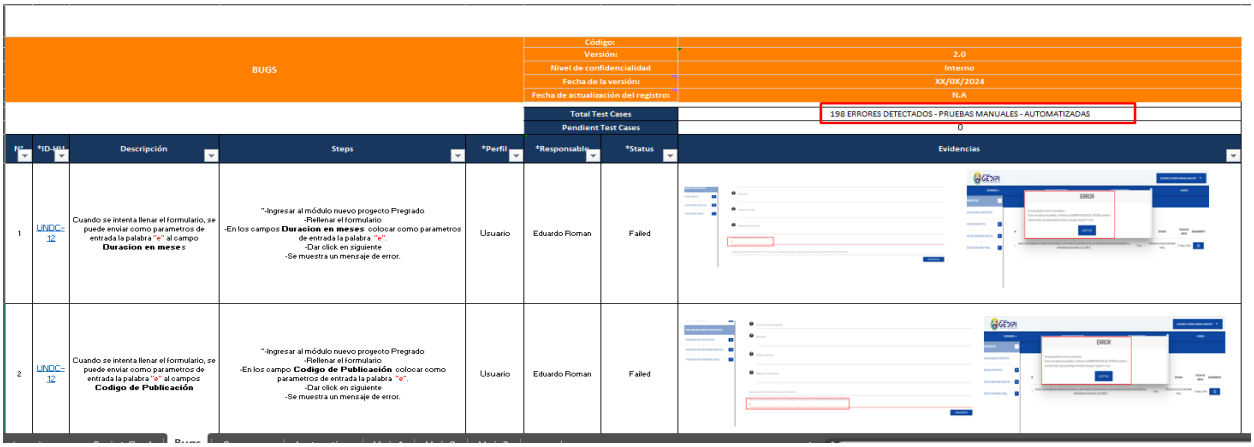


**Figura 43 -** Tabla de detección de errores

La figura 43, muestra una tabla de detección de errores utilizada para registrar y gestionar los errores encontrados durante las pruebas del sistema GEDIPI. En esta tabla se detallan varios aspectos relacionados con cada error detectado, incluyendo la descripción, los pasos para reproducir el error, el perfil del usuario que encontró el error, el estado del error y las evidencias correspondientes.

SUMMARY

| | Status | | | |
|---|---|---|---|---|
| | Failed | | Total Cantidad | Total % |
| N° HU | Cantidad | % | | |
| UNDC-12 | 19800% | 100% | 198 | 100% |
| Total | 19800% | 100% | 198 | 100% |

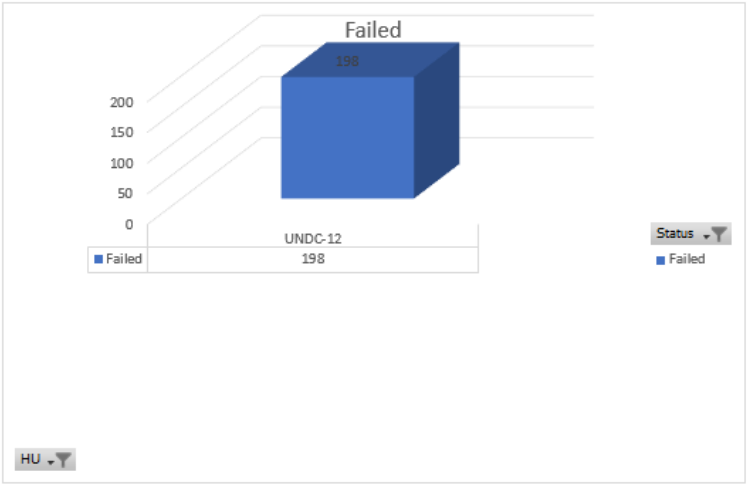



**Figura 44 -** Grafica de número de errores detectados

La figura 44, muestra una tabla y una gráfica que resumen el número de errores detectados durante las pruebas del sistema GEDIPI. La tabla enumera los errores por identificación de HU (Historial de Usuario), el estado de los errores, la cantidad de errores y el porcentaje del total de errores detectados. La gráfica visualiza estos datos de manera más clara y efectiva.

**Anexo 11: Artículo científico**

# Marco de pruebas automatizadas para el software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete

***ROMAN, Eduardo. ANCAJIMA, Adrian.***
1571436255@undc.edu.pe, 1771436255@undc.edu.pe
*Programa de ingeniería de sistemas, Facultad de ingeniería, Universidad Nacional de Cañete*

**Abstract -** The National University of Cañete faces important challenges in the testing process of the Digital Management of Research Processes (GEDIPI) software, due to the increasing complexity of the software used. Through the implementation of an Automated Testing Framework, this study seeks to address these issues and improve the effectiveness of the GEDIPI software. Modern methods such as SCRUM, Page Object Model (POM), and Behavior Driven Development (BDD) are expected to reduce testing time, increase the number of errors detected, and improve quality in the software testing execution process.The main problem is the inefficiency of the current testing process, which is characterized by long times, high costs and low test coverage. This study examines how test automation can improve software quality and increase the productivity of the testing process. It is believed that the Automated Testing Framework will significantly improve these aspects and provide a useful solution for the National University of Cañete.

**Keywords-** web automation, scrum, POM, BDD

**Resumen -** La Universidad Nacional de Cañete enfrenta importantes desafíos en el proceso de pruebas del sofware de Gestión Digital de Procesos de Investigación (GEDIPI), debido a la creciente complejidad del software utilizado. A través de la implementación de un Marco de Pruebas Automatizadas, este estudio busca abordar estos problemas y mejorar la eficacia del software de GEDIPI. Se espera que los métodos modernos como SCRUM, el Page Object Model (POM) y el Behavior Driven Development (BDD) reduzcan el tiempo de las pruebas, aumenten el número de errores detectados y mejoren la calidad en el proceso de ejecución de pruebas del software. El problema principal es la ineficiencia del proceso de pruebas actual, que se caracteriza por tiempos prolongados, costos elevados y baja cobertura de pruebas. Este estudio examina cómo la automatización de las pruebas puede mejorar la calidad del software y aumentar la productividad del proceso de pruebas. Se cree que el Marco de Pruebas Automatizadas mejorará significativamente estos aspectos y brindará una solución útil para la Universidad Nacional de Cañete.

**Palabras claves-** automatización web, scrum, POM, BDD

## I.INTRODUCCIÓN

La Universidad Nacional de Cañete enfrenta importantes desafíos en el proceso de pruebas del sofware de Gestión Digital de Procesos de Investigación (GEDIPI), debido a la creciente complejidad del software utilizado. A través de la implementación de un Marco de Pruebas Automatizadas, este estudio busca abordar estos problemas y mejorar la eficacia del software de GEDIPI. Se espera que los métodos modernos como el Page Object Model (POM) y el Behavior Driven Development (BDD) reduzcan el tiempo y el costo de las pruebas, aumenten la cobertura y mejoren el rendimiento y la calidad del software.

El problema principal es la ineficiencia del proceso de pruebas actual, que se caracteriza por tiempos prolongados, costos elevados y baja cobertura de pruebas. Este estudio examina cómo la automatización de las pruebas puede mejorar la calidad del software, aumentar la productividad del proceso de pruebas y optimizar el rendimiento del software. Se cree que el Marco de Pruebas Automatizadas mejorará significativamente estos aspectos y brindará una solución útil para la Universidad Nacional de Cañete.

### A. Planteamiento del problema

A nivel nacional, según [2] comenta que las pruebas de software automatizadas aumentan la eficacia y la eficiencia. Las enormes ventajas de

estas pruebas incluyen una reducción del tiempo de prueba y la capacidad de cubrir más escenarios. reducir en gran medida los errores manuales reduce significativamente las fallas manuales y utiliza pocos recursos en pruebas. Sin embargo, las pruebas automatizadas no garantizan la falta de fallas en la producción en un 100%. Los escenarios de pruebas en las plataformas son inmensos, por lo que se deben realizar pruebas constantemente para reducir la cantidad de fallos.

A nivel internacional, según [1] comenta que la automatización de pruebas es un paso fundamental para mejorar la calidad del software, ya que las aplicaciones se vuelven más complejas e incorporan funcionalidades adicionales. Esto conduce a una disminución en la confiablidad de la calidad final del software desarrollado. En estos casos la automatización de pruebas se vuelve muy importante a la hora de abordar el proceso de ejecución de las pruebas. Pueden ser aplicadas en diferentes niveles de casos de pruebas, desde pruebas unitarias, hasta pruebas de aceptación. En la actualidad, hay una variedad de herramientas disponibles para crear y ejecutar scripts, que son pequeños programas que el usuario puede interactuar con una aplicación específica y simular las acciones que podrían realizar de manera real. Dentro del testeo de aplicaciones con interfaz web, una de las herramientas de automatización de pruebas funcionales que más importancia está cobrando es Selenium. Selenium es un framework de desarrollo opensource con una gran comunidad que permite a los usuarios finales navegar a través de una aplicación web. Un script que se procesa por la herramienta indica los pasos que se deben seguir durante la ejecución de la prueba, como abrir el navegador, acceder a una URL específica, analizar el texto que se muestra en pantalla, acceder a un enlace específico, introducir datos, etc. Selenium ha avanzado y ahora ofrece una API completa para varios lenguajes de programación. Con las clases y métodos que ofrece, se pueden crear scripts que representan cada prueba automatizada. Actualmente la herramienta soporta los principales navegadores web como Chrome, Firefox, Internet Explorer, Safari, etc.

Esta investigación se examina el software de gestión digital de procesos de investigación que facilita la búsqueda de investigaciones, la creación de nuevos proyectos y la aprobación de investigaciones, entre otras cosas. Sin embargo, se encontró que durante el proceso de desarrollo de este sistema no se realizaron pruebas manuales o funcionales adecuadas. Como resultado, el objetivo de esta investigación es evaluar la calidad del software de manera automatizada utilizando la herramienta Selenium WebDriver y el lenguaje Gherkin, que proporciona una descripción clara y concisa de cada escenario de prueba.

**B. Problema General**

¿De qué manera la implementación de un Marco de Pruebas Automatizadas puede mejorar la eficacia del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete?

**C. Problema específicos**

- PE1: ¿De qué manera la implementación de un Marco de Pruebas Automatizadas puede mejorar la calidad del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete?
- PE2: ¿De qué manera la implementación de un Marco de Pruebas Automatizadas puede incrementar la productividad en el proceso de pruebas del software de

Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete?

### D. Objetivo general

Evaluar de qué manera la implementación de un Marco de Pruebas Automatizadas puede mejorar la eficacia del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

### E. Objetivos específicos

- OE1: Evaluar de qué manera la implementación de un Marco de Pruebas Automatizadas puede mejorar la calidad del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.
- OE2: Evaluar de qué manera la implementación de un Marco de Pruebas Automatizadas puede incrementar la productividad en el proceso de pruebas del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

### F. Marco conceptual

**Pruebas Automatizadas**. son procesos de verificación y validación del software, como lo describe [18] es el proceso donde se utiliza herramientas, frameworks, para ejecutar pruebas determinadas en una aplicación o sistema de software. La automatización puede ser útil para automatizar tareas repetitivas y ahorrar tiempo y esfuerzo. Se comparan los resultados reales con los esperados y se reportan los resultados automáticamente Estas pruebas permiten identificar de manera rápida errores, defectos y problemas de rendimiento durante el ciclo de desarrollo de software, lo que ayuda a mejorar la calidad del producto final y reduce los costos asociados con la detección y corrección de errores.

### G. Dimensiones

**Cobertura de Prueba.** [19] define la cobertura de pruebas como la medida en que las pruebas han cubierto el conjunto de funcionalidades de un sistema, proporcionando una indicación de qué tan bien se han verificado todas las partes del software.

**Tasa de errores detectado**. [20] afirma que la tasa de errores detectados es un indicador que refleja la cantidad de defectos o fallas identificados durante el proceso de pruebas en relación con el número total de defectos presentes en el software.

**Tiempo de ejecución de las pruebas**. El tiempo de ejecución de las pruebas se define como el tiempo que se toma para ejecutar un conjunto de pruebas sobre un sistema, lo cual es crucial para evaluar la eficiencia y rendimiento de las pruebas automatizadas, según [21].

### H. Indicadores

**Proporción de funcionalidades verificadas por las pruebas automatizadas.** [22] afirman que la proporción de funcionalidades verificadas por pruebas automatizadas es la medida en que las pruebas automatizadas cubren las funcionalidades del software, evaluando la efectividad de las pruebas en validar el comportamiento esperado del sistema.

**Número de errores identificados por las pruebas automatizadas.** De acuerdo con [23], el número de errores identificados por las pruebas automatizadas es una métrica crucial que indica la cantidad de fallos detectados por las pruebas automatizadas, lo cual es fundamental para evaluar la calidad del software y la efectividad del proceso de pruebas.

**Duración total para ejecutar todas las pruebas automatizadas.** El tiempo total requerido para completar todas las pruebas automatizadas de un sistema, lo cual es una métrica importante para evaluar la eficiencia del proceso de pruebas, afirma [24]

## II. TRABAJOS RELACIONADOS

[10] afirmaron en su tesis que su objetivo era implementar automatización en las pruebas de software de la empresa Entrega creativa, ya que esto podría mejorar la eficiencia de las pruebas, el nivel de calidad y reducir las fallas que pueden ocurrir cuando se realizan manualmente. La metodología que se utilizó un enfoque cuantitativo y un diseño experimental-preexperimental porque solo se aplica la solución a un grupo. Se utilizó una metodología híbrida entre RUP y SCRUM.

[11] en su tesis, su objetivo era automatizar los casos de prueba para mejorar el proceso de pruebas de un sistema que proporciona servicios RPO utilizando Selenium y el TestNG Framework. Los analistas de calidad realizaron pruebas complicadas, repetitivas y que requerirían mucho tiempo durante la investigación. Resolver esos problemas fue el objetivo del desarrollo de este proyecto. Los resultados de las pruebas se obtuvieron utilizando el proyecto UI Automatización. Las pruebas incluyen un número cada vez mayor de instancias de clientes: todas las instancias de clientes ahora se ejecutan pruebas de humo. La prueba dura menos de dos horas y no requiere ayuda humana.

## III. METODOLOGÍA

### A. Enfoque de la investigación

El proyecto se llevará a cabo empleando un enfoque cuantitativo que implica la recolección y análisis de datos numéricos para su procesamiento estadístico [43]. Empleando un enfoque cuantitativo, ya que implica la implementación y el análisis de un marco de pruebas automatizadas basado en la metodología (Behavior Driven Development) y la evaluación de su efectividad mediante los datos numéricos obtenidos a través de pruebas y mediciones en las que se llevará a cabo una comparación entre pruebas manuales y automatizadas.

### B. Tipo de investigación

La investigación será de tipo aplicada, [44] afirma que a este tipo de estudio se le conoce también como activa o dinámica, ya que depende de sus hallazgos y contribuciones teóricas para comparar la teórica con la realidad; asimismo, se enfoca en problemas específicos en condiciones y características concretas, centrándose en su aplicación inmediata. El proyecto será aplicado porque se enfoca en la implementación práctica de un marco de pruebas automatizadas para el software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete, este tipo de estudio busca resolver problemas específicos y mejorar procesos concretos en el entorno real, utilizando el Page Object Model y Behavior Driven Development, buscando lograr resultados aplicables y útiles para la investigación.

### C. Nivel de investigación

[44] afirma que, en el nivel aplicativo, es necesario evaluar el éxito de la solución del problema, intervienen la población o unidades de estudio para obtener resultados positivos y mejorarlo. Nuestro proyecto presenta un nivel aplicativo debido a que se centra en la implementación práctica de un marco de pruebas automatizadas para el sofware de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete, en este nivel se lleva a cabo la intervención directa en el proceso de investigación, empleando los conocimientos teóricos y metodológicos de manera práctica buscando resolver problemas

concretos en el entorno; además se busca evaluar el éxito de las pruebas, mediante la eficacia y eficiencia del framework, para lograr mejorar tangibles y transformaciones positivas en la gestión de los procesos de investigación en la Universidad Nacional de Cañete.

### D. Metodo de investigación

El método que se elegirá será el hipotético-deductivo, el cual se caracteriza por la formulación de hipótesis seguida de la deducción de sus consecuencias, además se apoya de la observación de hechos empíricas, operacionalización de variables y evolución de hipótesis para verificar su validez, lo que facilita abordar en las conclusiones de la investigación [45]. Nuestro trabajo presentará el método hipotético- deductivo, ya que nos va a permitir formular hipótesis sobre la eficacia y eficiencia del diseño y metodología ágil, para luego deducir consecuencias a partir de la observación de los datos empíricos recopilados durante la implementación, de esta manera, el método nos va a permitir verificar las hipótesis planteadas y fundamentar las conclusiones en base a la evidencia concreta.

### E. Diseño de investigación

Según [43], un diseño preexperimental se distingue por no tener un control estricto sobre las variables externas y no realizar una manipulación exhaustiva de la variable independiente. Las etapas exploratorias de la investigación, donde el objetivo es obtener una primera aproximación a la relación entre variables, utilizan principalmente este tipo de diseño. El proyecto será de diseño preexperimental será de diseño preexperimental porque se centrará en evaluar el impacto de la implementación de pruebas automatizadas en la eficiencia del proceso de pruebas, utilizando un solo grupo de estudio con mediciones antes y después de la intervención, sin control riguroso sobre otras variables, lo que permite obtener una primera aproximación al efecto de las pruebas automatizadas, aunque con limitaciones en la generalización de los resultados.

### F. Población, Muestra, Muestreo

[46] afirma que la población se refiere al conjunto completo de elementos o individuos que poseen las características que se estudian, puede ser cualquier grupo definido, ya sea personas, objetos, eventos, etc., que se considere relevante para el estudio. En el presente proyecto, la población estará conformada por todas las funcionalidades del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

La muestra es un subconjunto de la población seleccionado para participar en un estudio con el propósito de hacer inferencias sobre la población total a partir de los resultados obtenidos en el análisis de este subconjunto de la población, según [46]. Para el presente trabajo, se considerará treinta (30) funcionalidades críticas seleccionadas, que estan presentes en el módulo de registro de investigaciones, el módulo de seguimiento de proyectos y el módulo de generación de reportes, entre otros.

[47] definen el muestreo como el proceso mediante el cual se selecciona un subconjunto de personas o elementos de una población más grande para hacer inferencias sobre esa población. Para asegurarse de que los hallazgos del estudio sean representativos y generalizables, este proceso es crucial. En la presente investigación, el muestreo será no probabilístico por criterio, eligiendo aquellas funcionalidades que son esenciales para el funcionamiento del sistema y que representan adecuadamente la diversidad de tareas que realiza el software.

### G. Técnica, Instrumento

[48] definen una técnica de investigación como un conjunto de procedimientos y métodos específicos que se utilizan para la recolección y análisis de datos en una investigación, que permiten al investigador obtener la información necesaria para resolver las preguntas de investigación. La técnica a utilizar en este proyecto será la observación estructurada.

[49] definen un instrumento de investigación como un recurso que el investigador utiliza para recopilar datos, tales como encuestas, cuestionarios, entrevistas y observaciones estructuradas, diseñados específicamente para medir las variables de interés del estudio. El instrumento que se utilizará en la presente investigación serán las fichas de registros.

### H. Técnicas estadística de análisis de datos

Para llevar a cabo el análisis de datos, se utilizará fichas de registros para recopilar información, que se organizará en hojas de cálculo de Excel 2021. A continuación, para obtener una visión general de los resultados, se generarán estadísticas descriptivas utilizando los datos organizados. La prueba de hipótesis se llevará a cabo posteriormente con los datos de las pruebas previas y posteriores utilizando el programa estadístico SPSS IBM 24. Además, se utilizarán estadísticas descriptivas para analizar los datos y los procedimientos. Finalmente, se prepararán los resultados inferenciales utilizando el sistema estadístico y las hojas de Excel.

## IV. RESULTADOS OBTENIDOS

### A. Análisis descriptivo

Las Tablas 1, 2, 3 y las Figuras 1, 2 y 3 muestran los datos de este estudio.

**Evaluación descriptiva del indicador 1:** Número de errores detectados (NED)

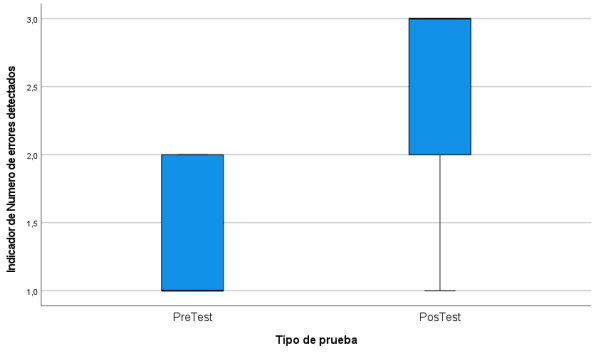
TABLA I

EVALUACIÓN DESCRIPTIVA DEL INDICADOR NED

| Estadísticos descriptivos | | | | | | | |
|---|---|---|---|---|---|---|---|
| | N | Mínimo | Máximo | Media | Desviación estándar | Varianza | Coeficiente de Variabilidad (DE/M)*100 |
| Errores detectados Pre test | 30 | 1 | 2 | 1,33 | 0,479 | 0,230 | 63,70% |
| Errores detectados Post test | 30 | 1 | 3 | 2,67 | 0,547 | 0,299 | 20.48% |
| N válido (por lista) | 30 | | | | | | |

Esta tabla I, dado su media, desviación estándar, varianza, valor mínimo, valor máximo y coeficiente de variabilidad, muestra el análisis descriptivo del indicador NED en el pre test y post test.

**Figura 1** - Diagrama de Caja de Bigotes del indicador NED




La tabla I muestra que el Pre-Test del indicador NED obtuvo una media del 1,33%, mientras que el Post-Test obtuvo un porcentaje del 2,67%.

Por otra parte, en la figura 1, el diagrama de barras se muestra que existe diferencias entre el Pre-Test y Post-Test para el indicador Número de errores detectados, indicando que con la automatización (Post Test) se detecta más Número de errores que la ejecución de pruebas

manuales (Pre Test). Esto demuestra una gran diferencia entre el tiempo previo y posterior a la implementación de la automatización.

**Evaluación descriptiva del indicador 2:** Tiempo de ejecución de las pruebas (TEP)

TABLA II

EVALUACIÓN DESCRIPTIVA DEL INDICADOR TEP

| Estadísticos descriptivos | | | | | | | |
|---|---|---|---|---|---|---|---|
| | N | Mínimo | Máximo | Media | Desviación estándar | Varianza | Coeficiente de Variabilidad (DE/M)*100 |
| Tiempo de ejecución de pruebas Pre test | 30 | 1,37 | 4,28 | 2,39 | 0,75635 | 0,572 | 31,65% |
| Tiempo de ejecución de pruebas Post test | 30 | 0,30 | 3,21 | 1,21 | 0,75705 | 0,573 | 62,61% |
| N válido (por lista) | 30 | | | | | | |

Esta tabla II, dado su media, desviación estándar, varianza, valor mínimo, valor máximo y coeficiente de variabilidad, muestra el análisis descriptivo del indicador TEP en el pre test y post test.

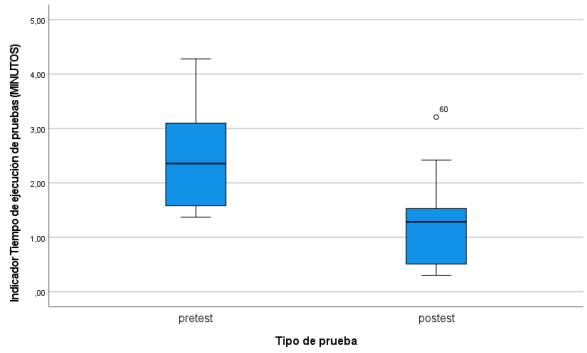



**Figura 2 -** Diagrama de Caja de Bigotes del indicador TEP

Tal como se observa en la tabla II, muestra que el Pre-Test del indicador TEP obtuvo una media del 2,39 %, mientras que el Post-Test obtuvo un porcentaje del 1,21%.

Por otra parte, en la figura 2 se observa, el diagrama de caja de bigotes, se muestra la diferencia entre el Pre-Test y Post-Test, indicando que con la automatización (Post Test), el tiempo de ejecución de la prueba es más rápido que las pruebas manuales (Pre Test). Esto demuestra una gran diferencia entre el tiempo previo y posterior a la implementación de la automatización.

### B. Análisis Inferencial

**Pruebas de normalidad.** [50] afirma que la prueba de normalidad permite evaluar la adecuación de una muestra a la distribución normal, lo cual es esencial para la aplicación adecuada de varios métodos estadísticos que asumen esta distribución. En términos prácticos, estas pruebas comparan la distribución de los datos observados con una distribución normal teórica y evalúan las discrepancias entre las dos. Esto proporciona un marco para determinar si se pueden realizar inferencias estadísticas basadas en la normalidad de los datos.

Para las evaluaciones posteriores, se estableció un nivel de significancia de 0.05 y una confianza previamente establecida del 95%. Los resultados de la significancia calculados en la columna Sig tanto del análisis Pre-Test como del análisis Post-Test se tomarán en cuenta para la toma de decisiones. Para los criterios de decisión se considerará que si Sig. < 0.05 se rechaza la Ho y se acepta la H1, y si Sig >= 0.05 se acepta la Ho y se rechaza la H1.

**Prueba de normalidad del indicador 1:** Número de errores detectados (NED).

- Ho: El indicador NED tiene datos con distribución normal.
- H1: El indicador NED no tiene datos con distribución normal.

TABLA III

TEST DE NORMALIDAD DEL INDICADOR NED

| | Tipo de prueba | Shapiro-Wilk | | |
|---|---|---|---|---|
| | | Estadístico | gl | Sig. |
| Indicador de Numero de errores detectados | Pre Test | ,597 | 30 | ,001 |
| | Post Test | ,623 | 30 | ,001 |

En la tabla III, se observa que el indicador Número de errores detectados para el Pre-Test tiene una Sig=0,001<0,05 y en el Post-Test tiene una Sig=0,001<0,05, por lo tanto, la distribución es no normal. Entonces se rechaza la hipótesis nula y se acepta la hipótesis alternante. Por lo tanto, se va a utilizar procedimientos de la estadística no paramétricas con U de Mann-Whitney.

TABLA IV

PRUEBA DE CONFIABILIDAD DEL INDICADOR NED

| Correlaciones | | | | |
|---|---|---|---|---|
| | | | VAR00001 | VAR00002 |
| Rho de Spearman | VAR00001 | Coeficiente de correlación | 1,000 | ,913* |
| | | Sig. (bilateral) | . | ,030 |
| | | N | 5 | 5 |
| | VAR00002 | Coeficiente de correlación | ,913* | 1,000 |
| | | Sig. (bilateral) | ,030 | . |
| | | N | 5 | 5 |

En la tabla IV, de acuerdo con la prueba de test y retest, utilizando la correlación de rho de Spearman, que arrogo un valor de 0,913 es una correlación positiva muy fuerte [51].

**Prueba de normalidad del indicador 2:** Tiempo de ejecución de las pruebas (TEP) (Minutos).

- Ho: El indicador TEP tiene datos con distribución normal.
- H1: El indicador TEP no tiene datos con distribución normal.

TABLA V

TEST DE NORMALIDAD DEL INDICADOR TEP

| | Tipo de prueba | Shapiro-Wilk | | |
|---|---|---|---|---|
| | | Estadístico | gl | Sig. |
| Indicador Tiempo de ejecución de pruebas (MINUTOS) | Pre Test | ,939 | 30 | ,085 |
| | Post Test | ,903 | 30 | ,010 |

En la tabla V, se observa que el indicador Tiempo de ejecución de las pruebas para el Pre-Test tiene una Sig=0,85>0,05 por lo tanto tiene una distribución normal y en el Post-Test el Sig=0,010<0,05 por lo tanto tiene una distribución no normal. Para poder definir qué tipo de distribución prevalece, se va a realizar una combinación lineal de ambas, a través de una diferencia.

TABLA VI

COMBINACIÓN LINEAL - TEST DE NORMALIDAD DEL INDICADOR TEP

| | Shapiro-Wilk | | |
|---|---|---|---|
| | Estadístico | gl | Sig. |
| Diferencia | ,561 | 30 | ,001 |

En la tabla VI, la combinación lineal de la diferencia para ambas distribuciones resulto el Sig=0,001<0,05, por lo tanto, la distribución es no normal. Entonces se rechaza la hipótesis nula y se acepta la hipótesis alternante. Por lo tanto, se va a utilizar procedimientos de la estadística no paramétricas con U de Mann-Whitney.

TABLA VII

PRUEBA DE CONFIABILIDAD DEL INDICADOR TEP

| Correlaciones | | | | |
|---|---|---|---|---|
| | | | Pre Test D2 | Post Test D2 |
| Rho de Spearman | PreTestD2 | Coeficiente de correlación | 1,000 | ,997** |
| | | Sig. (bilateral) | . | ,000 |
| | | N | 30 | 30 |
| | Post Test D2 | Coeficiente de correlación | ,997** | 1,000 |
| | | Sig. (bilateral) | ,000 | . |
| | | N | 30 | 30 |

En la tabla VII, de acuerdo con la prueba de test y retest, utilizando la correlación de rho de Spearman, que arrogo un valor de 0,997 es una correlación positiva muy fuerte [51].

### C. Prueba de hipótesis

La prueba de hipótesis es un método estadístico para determinar una población a partir de una muestra. La prueba de hipótesis implica formular una hipótesis nula y una alternativa, luego usar datos muestrales para determinar si la hipótesis nula se rechaza o no, según [51].Este método permite evaluar la evidencia de los datos y hacer inferencias sobre la población, ayudando a determinar si existe una diferencia significativa o efecto en un contexto específico.

**Prueba de hipótesis con el indicador 1:** Número de errores detectados (NED).

- Ho: La implementación de un Marco de Pruebas Automatizadas no mejorará significativamente la calidad del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.
- H1: La implementación de un Marco de Pruebas Automatizadas mejorará significativamente la calidad del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

Nivel significación =0.05
Criterio de prueba = si el Sig<0.05 se rechaza la hipótesis nula y se acepta la alternante.

TABLA VIII

PRUEBA DE U DE MANN-WHITNEY DEL INDICADOR NED

| Estadísticos de prueba[a] | |
|---|---|
| | Indicador de Numero de errores detectados |
| U de Mann-Whitney | 60,000 |
| W de Wilcoxon | 525,000 |
| Z | -6,120 |
| Sig. asin. (bilateral) | ,000 |
| a. Variable de agrupación: Tipo de prueba | |

En la tabla VIII, muestra que se rechaza la hipótesis nula y se acepta la hipótesis alternante, ya que se utilizó la prueba de U de Mann-Whitney donde hubo una Sig=0,000<0,05. Se concluye que la implementación de un Marco de Pruebas Automatizadas mejorará significativamente la calidad del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

**Prueba de hipótesis con el indicador 2:** Tiempo de ejecución de las pruebas (TEP)

- Ho: La implementación de un Marco de Pruebas Automatizadas no incrementará significativamente la productividad en el proceso de pruebas del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.
- H1: La implementación de un Marco de Pruebas Automatizadas incrementará significativamente la productividad en el proceso de pruebas del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

Nivel significación =0.05

Criterio de prueba = si el Sig<0.05 se rechaza la hipótesis nula y se acepta la alternante.

TABLA IX

PRUEBA DE U DE MANN-WHITNEY DEL INDICADOR TEP

| Estadísticos de prueba[a] | |
|---|---|
| | Indicador Tiempo de ejecución de pruebas (MINUTOS) |
| U de Mann-Whitney | 127,000 |
| W de Wilcoxon | 592,000 |
| Z | -4,778 |
| Sig. asin. (bilateral) | ,000 |
| a. Variable de agrupación: Tipo de prueba | |

En la tabla XI, muestra que se rechaza la hipótesis nula y se acepta la hipótesis alternante, ya que se utilizó la prueba de U de Mann-Whitney donde hubo una Sig=0,000<0,05. Se concluye que la implementación de un Marco de Pruebas Automatizadas incrementará significativamente la productividad en el proceso de pruebas del software de Gestión Digital de Procesos de Investigación de la Universidad Nacional de Cañete.

## V. CONCLUSIONES

Se concluye que la automatización web ha demostrado ser un método efectivo para mejorar varios aspectos del software de gestión digital de procesos de investigación.

Primero, los resultados mostraron un aumento significativo en la eficiencia en la detección de errores (indicador NED). Antes de la

automatización, el NED promedio era del 1,33%, sin embargo, el promedio aumentó a 2.67% después de que se implementó la automatización web. Esto indica que se detectó un mayor número de errores gracias a la automatización, con un aumento promedio del 1,34%.

En segundo lugar, los resultados también fueron positivos con relación al tiempo de ejecución de pruebas (indicador TEP). En la prueba previa, el TEP promedio fue de 2.39%, pero después de la automatización, se redujo a 1.21%. La automatización web acelera la ejecución de casos de prueba, como lo demuestra esta disminución del 1,18%.

En resumen, los hallazgos de esta investigación indican que el uso del lenguaje Gherkin y Selenium WebDriver en la automatización web mejora la eficiencia en la detección de errores y la rapidez en la ejecución de pruebas, estos resultados respaldan el uso de técnicas de automatización para mejorar la eficiencia y la calidad de los procesos de prueba de software.

## VI. REFERENCIAS


[1] R. Cubas Montenegro, «Testing y Calidad de Software. Automatización de pruebas con Selenium WebDriver,» Madrid, 2019.

[2] E. Liza Silva, «Implementación de automatización aplicada a pruebas de software del área de calidad de la empresa Entrega Creativa,» Lima, 2022.

[3] H.-S. y. Mendoza, «Metodología de la investigación. Las rutas cuantitativa, cualitativa y mixta 6th ed,» McGraw-Hill Education, Ciudad de México, 2018.

[4] R. F. y. P. B. J. Hernández, «Metodología de la investigación 6th ed,» McGraw-Hill, Ciudad de México, 2014.

[5] A. Musliu y X. Jashari, «Software Automated Testing using BDD,» pristina, Kosovo, 2021.

[6] A. C. Barus, «The implementation of ATDD and BDD from,» 2019.

[7] R. Segovia Gonzalez, D. K. Urribarri y M. L. Larrea, «Automation Tools for Web Testing. Beyond,» ASSE, Argentine Symposium on Software Engineering, Argentina, 2022.

[8] N. Mittal, «Automation Testing Using Cucumber and Selenium WebDriver,» BioGecko, Nagpur, Maharastra, 2023.

[9] W. Morales, «Impacto de la automatización de pruebas en la eficiencia de pruebas de software en una consultora de TI,» UCV, Peru, 2022.

[10] E. V. R. Liza, «Implementación de automatización aplicada a pruebas de software del área de calidad de la empresa Entrega Creativa,» Universidad Tecnológica del Perú, Perú, 2022.

[11] P. Chávez, «Automatización de casos de pruebas usando Selenium y TewstNG Framework para mejorar el proceso de pruebas de un sistema que brinda servicios RPO,2020,» Universidad Nacional Mayor de San Marcos, Perú, 2020.

[12] I. García, «Automatización de prueba de software para mejorar el proceso de prueba del módulo del sistema que desarrolla empresa Experts Systems S.A.C,» Universidad Nacional Santiago Antúnez de Mayolo, Lima, 2022.

[13] J. Palomino Barrasa, «Automatizacion de plan de pruebas funcional: Plataforma web de la universidad politecnica de madrid,» universidad politecnica de madrid, Madrid, 2020.

[14] I. A. G. Calero, «Creación de pruebas automatizadas para modelos ejecutables de software,» UNIVERSIDAD DE MÁLAGA, 2021, 2021.

[15] F. Salas Villalobos y H. Jiménez Chévez, «Prácticas y experiencias en automatización de pruebas funcionales de integración en C#,» UNIVERSIDAD NACIONAL COSTA RICA, COSTA RICA, 2021.

[16] A. Cortés, «Automatización de pruebas de regresión para reducción de tiempo de entrega de nuevas versiones de sofware,» Universidad de Chile, Chile, 2020.

[17] D. Spindola, «Desarrollo de un framework de pruebas automatizadas para la verificaicón de aplicaciones web,» Universidad Autpnoma de Querétaro, Ecuador, 2021.

[18] J. R. Páez Maldonado, «Implementación de

pruebas automatizadas para la integración de los procesos deesarrollo de software de una POC (Tienda virtual,» Bogota, 2023.

[19] R. Pressman, «Ingeniería de software: Un enfoque práctico, 7th ed.,» McGraw-Hill, Ciudad de México, 2010.

[20] I. Sommerville, «Ingeniería de software 9th ed.,» Addison-Wesley, Madrid, 2011.

[21] S. y. B. Myers, «The Art of Software Testing 3rd ed,» Hoboken, NJ: John Wiley & Sons, 2011.

[22] F. y. Highsmith, «Refactoring: Improving the Design of Existing Code 2nd ed,» Addison-Wesley, 2020.

[23] C. y. Gregory, «Agile Testing: A Practical Guide for Testers and Agile Teams 2nd ed.,» Addison-Wesley, 2021.

[24] G. Adzic, «Specification by Example: How Successful Teams Deliver the Right Software2nd ed,» Manning Publications, 2019.

[25] I. Idea, «Intellij Idea,» Ijtellij Idea, 2020.

[26] N. Garg, «Selenium WebDriver Step By Step Guide,» AdactIn Group Pty Ltd., 2014.

[27] Cucumber.io, «Cucumber.io,» Cucumber.io, 2024. [En línea]. Available: https://cucumber.io/. [Último acceso: 21 mayo 2024].

[28] B. y. P. Kaner, «Lessons Learned in Software Testing: A Context-Driven Approach 2nd ed.,» Wiley, 2020.

[29] Sommerville, «Software Engineering 10th ed.,» Pearson, 2015.

[30] J. Lewis, «Software Testing and Continuous Quality Improvement 4th ed,» Auerbach Publications, 2019.

[31] K. Schwaber y J. Sutherland, «La Guía Scrum,» Creative Commons, 2020.

[32] M. Torrado Nevado, «Estudio de metodologías ágiles. Aplicación SCRUM,» Universidad de sevilla, sevilla, 2019.

[33] L. J. Delgado Taday, «METODOLOGÍA BDD APLICADA AL DESARROLLO DEL SISTEMA WEB PARA EVALUAR LA CALIDAD DE LOS SERVICIOS TURÍSTICOS COMUNITARIOS RURALES DEL CANTÓN RIOBAMBA,» Riobamba, 2022.

[34] S. España, «Patrones de diseño en automatización:PageObjects,» 2014.

[35] ISQTB, Probador Certificado de ISQTB, 2018 ed., Belgica, 2018.

[36] D. Spindola Hernandez, «Desarrollo de un framework de pruebas automatizadas para la verificacion de aplicaciones web,» Santiago de Queretaro, 2021.

[37] E. Parra Valverde, «Estudio del control de calidad en procesos de desarrollo de software. Aplicación práctica de pruebas automatizadas a un sitio web de comercio electrónico,» Valladolid, 2020.

[38] R. D C y M. A E, «A Comprehensive Study on Automation Testing using JUnit,» IRJET, India, 2020.

[39] W. Rojas y M. Silva, Introduccion a java: guia de actividades practicas, Colombia: Universidad El Bosque, 2016.

[40] Rubin, «Essential Scrum: A Practical Guide to the Most Popular Agile Process 2nd ed,» Addison-Wesley, 2020.

[41] F. Toledo, «Introducción a las pruebas de sistemas de información,» Abstracta, Uruguay, 2014.

[42] D. Esteve, «implantación de un proceso de automatización de pruebas para una aplicación de software,» Universidad Politécnica de Valencia, Valencia, España, 2015.

[43] C. Bernal, «Metodología de la investigación (Tercera edición),» Pearson Educación de Colombia Ltda, Colombia, 2010.

[44] M. Tamayo, «El proceso de la investigación científica,» Editorial Limusa, Mexico, 2004.

[45] R. y. M. C. Hernández, «Metodología de la investigación. Las rutas cuantitativa, cualitativa y mixta,» Ed. Mc Graw Hill Education, 2018.

[46] G. Baena, «todología de la investigación. Grupo Editorial Patria,» Grupo Editorial Patria, Mexico, 2017.

[47] C. y. Creswell, «Research Design: Qualitative, Quantitative, and Mixed Methods Approaches 5th ed.,» Thousand Oaks, CA: SAGE Publications, 2020.

[48] F. y. B. Hernández, «Metodología de la investigación 6th ed.,» McGraw-Hill, Ciudad de México, 2014.

[49] C. y. L. Sampieri, «Metodología de la investigación 6th ed.,» McGraw-Hill, Ciudad de México, 2014.

[50] D. M. L. y. D. F. Stephan, «Statistics for Managers Using Microsoft Excel, 8ª ed,» Pearson, Boston, USA, 2017.

[51] D. C. M. y. G. C. Runger, «Applied Statistics and Probability for Engineers, 6ª ed,» Wiley, NJ, USA, 2014.

[52] M. K. y. J. Kim, «Software Testing: Principles and Practices 1st ed.,» Springer, 2021.

## 6.1. Aspectos Administrativos Cronograma de actividades

| N° | Actividad | Entregable | Responsable | Abril | Mayo | Junio | Julio |
|---|---|---|---|---|---|---|---|
| 1 | Revisión de literatura sobre el tema de pruebas automatizadas | Archivos encontrados | Equipo de investigación | X | | | |
| 2 | Planteamiento del problema y objetivos | Documento del problema y objetivos | Equipo de investigación | X | | | |
| 3 | Diseño del marco teórico | Marco teórico preliminar | Equipo de investigación | X | | | |
| 4 | Identificación de herramientas de pruebas automatizadas | Listado de herramientas | Equipo de investigación | | X | | |
| 5 | Elaboración del marco conceptual | Documento del marco conceptual | Equipo de investigación | | X | | |
| 6 | Selección de herramientas para la implementación | Herramientas seleccionadas | Equipo de investigación | | X | | |
| 7 | Desarrollo del plan de pruebas | Plan de pruebas documentado | Equipo de investigación | | X | | |
| 8 | Diseño de casos de prueba | Casos de prueba diseñados | Equipo de investigación | | X | | |
| 9 | Implementación de entorno de pruebas | Entorno de pruebas configurado | Equipo de investigación | | | X | |
| 10 | Desarrollo de scripts de pruebas | Scripts de prueba desarrollados | Equipo de investigación | | | X | |
| 11 | Ejecución de pruebas iniciales | Resultados de pruebas iniciales | Equipo de investigación | | | X | |
| 12 | Análisis de resultados de pruebas iniciales | Informe de análisis inicial | Equipo de investigación | | | X | |
| 13 | Ajustes y optimización de scripts de prueba | Scripts optimizados | Equipo de investigación | | | | X |
| 14 | Ejecución de pruebas finales | Resultados de pruebas finales | Equipo de investigación | | | | X |
| 15 | Análisis de resultados finales | Informe de resultados finales | Equipo de investigación | | | | X |
| 16 | Documentación del proceso de pruebas | Documento del proceso | Equipo de investigación | | | | X |
| 17 | Redacción del capítulo de metodología | Capítulo de metodología redactado | Equipo de investigación | | | | X |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 18 | Redacción del capítulo de resultados | Capítulo de resultados redactado | Equipo de investigación | | | | X |
| 19 | Redacción del capítulo de discusión | Capítulo de discusión redactado | Equipo de investigación | | | | X |
| 20 | Redacción del capítulo de conclusiones | Capítulo de conclusiones redactado | Equipo de investigación | | | | X |
| 21 | Revisión y corrección de la tesis | Tesis revisada y corregida | Equipo de investigación | | | | X |
| 22 | Preparación de la defensa de la tesis | Presentación preparada | Equipo de investigación | | | | X |
| 23 | Simulación de la defensa | Simulación realizada | Equipo de investigación | | | | X |
| 24 | Presentación de la tesis para revisión final | Tesis presentada para revisión | Equipo de investigación | | | | X |
| 25 | Incorporación de sugerencias finales | Tesis ajustada con sugerencias | Equipo de investigación | | | | X |
| 26 | Preparación de la documentación final | Documentación final preparada | Equipo de investigación | | | | X |
| 27 | Presentación final de la tesis | Tesis presentada | Equipo de investigación | | | | X |
| 28 | Publicación de la tesis en el repositorio institucional | Tesis publicada | Equipo de investigación | | | | X |
| 29 | Difusión de resultados | Resultados difundidos | Equipo de investigación | | | | X |
| 30 | Evaluación y cierre del proyecto | Informe de cierre | Equipo de investigación | | | | X |

## 6.2. Presupuesto

### Tabla 1: Aporte no monetario del proyecto de investigación

| Clasificador de gastos | Concepto | Monto S/. |
|---|---|---|
| 163.34.45.68 | Uso de instalaciones y equipos de la Universidad Nacional de Cañete | 5,000 |
| | Acceso a bases de datos y bibliotecas | 3,000 |
| | Asesoría de profesores y expertos | 4,000 |
| | Licencias de software | 2,000 |
| | Recursos tecnológicos (servidores, etc.) | 6,000 |
| **Total** | | **20,000** |

### Tabla 2: Aporte monetario del proyecto de investigación

| Clasificador de gastos | Concepto | Monto S/. |
|---|---|---|
| 234.56.78.90 | Materiales de oficina | 200 |
| 234.56.78.91 | Publicaciones y material impreso | 200 |
| 234.56.78.92 | Gastos de transporte | 300 |
| 234.56.78.93 | Inscripción en taller de tesis | 6,200 |
| 234.56.78.95 | Incentivos para participantes en pruebas | 100 |
| **Total** | | **7,000** |

### Presupuesto total del proyecto

El presupuesto total del proyecto será: valor presupuesto monetario + valor presupuesto no monetario.

- **Aporte no monetario:** S/. 20,000
- **Aporte monetario:** S/. 7,000

**Presupuesto total del proyecto:** S/. 27,000

## 5.3. Financiamiento

El financiamiento del proyecto de investigación será cubierto de la siguiente manera:

| Entidad financiadora | Monto S/. | Porcentaje |
|---|---|---|
| Universidad Nacional de Cañete (UNDC) | 10,000 | 37.04% |
| Tesista: Eduardo Román | 3,500 | 12.96% |
| Tesista: Adrián Ancajima | 3,500 | 12.96% |
| Ministerio de Educación | 10,000 | 37.04% |
| | | |