



UNIVERSIDAD CÉSAR VALLEJO

ESCUELA DE POSGRADO

**PROGRAMA ACADÉMICO DE MAESTRÍA EN INGENIERÍA
DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA
INFORMACIÓN**

Impacto de la automatización de pruebas en la eficiencia de pruebas
de software en una consultora de TI, Lima 2022

TESIS PARA OBTENER EL GRADO ACADÉMICO DE:
MAESTRA EN INGENIERÍA DE SISTEMAS CON
MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

AUTORA:

Morales Yovera, Wendy Ly ([ORCID: 0000-0003-0272-0637](https://orcid.org/0000-0003-0272-0637))

ASESOR:

Mg. Cardeña Peña Jorge Manuel ([ORCID: 0000-0003-3176-8613](https://orcid.org/0000-0003-3176-8613))

LÍNEA DE INVESTIGACIÓN:

Sistema de Información y Comunicaciones

LÍNEA DE RESPONSABILIDAD SOCIAL UNIVERSITARIA:

Desarrollo económico, empleo y emprendimiento

LIMA - PERÚ

2022

Dedicatoria

Es una dedicatoria para mi creador y todopoderoso Dios, y a mamá Virgen de Guadalupe; por escucharme y sentir su gracia todos los días. A mis ángeles que me cuidan desde el cielo mis abuelos Roberto y Domitila, como también hace poco se unió a ellos mi tía Carmen, dejaron en mi la mejor herencia el amor, humildad y respeto a todos los que me rodea. Sin dejar de lado, también a mis queridos padres María, y Roberto. A Jean B., mi pareja por su tolerancia en esta etapa quien me enseña con amor la disciplina y su preocupación para que pueda equilibrar el trabajo, y los estudios.

Agradecimiento

Para mi familia por entender mis ausencias y contar con su apoyo. También agradezco a nuestro asesor Ing. Cardeña por su apoyo y orientación. Adicionalmente a mis compañeros de maestría que vivimos las experiencias desde el primer curso de la maestría para Alberto, Alithu y Jorge.

Índice de contenidos

| | Pág. |
|--|------|
| Carátula | i |
| Dedicatoria | ii |
| Agradecimiento | iii |
| Índice de contenidos | iv |
| Índice de tablas | v |
| Índice de gráfico y figuras | vi |
| Resumen | vii |
| Abstract | viii |
| I. INTRODUCCIÓN | 1 |
| II. MARCO TEÓRICO | 5 |
| III. METODOLOGÍA | 18 |
| 3.1. Tipo y diseño de Investigación | 18 |
| 3.2. Variables y operacionalización | 19 |
| 3.3. Población, muestra y muestreo | 21 |
| 3.4. Técnicas e instrumentos de recolección de datos | 22 |
| 3.5. Procedimientos | 24 |
| 3.6. Método de análisis de datos | 24 |
| 3.7. Aspectos éticos | 24 |
| IV. RESULTADOS | 26 |
| V. DISCUSIÓN | 35 |
| VI. CONCLUSIONES | 41 |
| VII. RECOMENDACIONES | 42 |
| REFERENCIAS | 43 |
| ANEXOS | 51 |

Índice de tablas

| | Pág. |
|---|------|
| Tabla 1 Matriz de operacionalización de la variable dependiente eficiencia..... | 20 |
| Tabla 2 Población de la investigación | 21 |
| Tabla 3 Ficha técnica del instrumento..... | 23 |
| Tabla 4 Validación del instrumento de recolección de datos | 23 |
| Tabla 5 Medidas descriptivas del tiempo medio por caso de prueba | 26 |
| Tabla 6 Medidas descriptivas del número de caso de prueba ejecutados por hora | 28 |
| Tabla 7 Prueba de normalidad del tiempo medio por caso de prueba..... | 30 |
| Tabla 8 Prueba de normalidad del número de casos de prueba ejecutados | 31 |
| Tabla 9 Resultados de los rangos en la prueba de wilconxon del indicador 1 del tiempo medio por caso de prueba ejecutados por hora | 32 |
| Tabla 10 Resultados estadísticos de contraste en la prueba de Wilcoxon del indicador tiempo medio por caso de prueba | 33 |
| Tabla 11 Prueba de T de Student de muestras emparejadas del indicador número de casos de prueba ejecutados por hora | 34 |

Índice de gráfico y figuras

| | Pág. |
|--|------|
| Figura 1 Comparación de valores medios del indicador tiempo medio por caso de prueba | 27 |
| Figura 2 Comparación de valores medios del indicador número de caso de prueba ejecutados por hora..... | 29 |

Resumen

El objetivo final de la presente investigación fue determinar el impacto de la automatización de pruebas en la eficiencia de las pruebas de software en una consultora de TI en Lima 2022. El desarrollo de la investigación es de tipo aplicada, con un enfoque cuantitativo, bajo el método hipotético deductivo y de diseño pre-experimental. La técnica utilizada fue la observación y el instrumento para la recolección de datos se utilizó la ficha de observación. Así mismo, se aplicó la prueba de Wilcoxon y T de Student, para la contrastación de la hipótesis.

Los resultados demostraron que después de la implementación de la automatización de pruebas lograron una disminución del 74.68% en el tiempo medio por caso de prueba para el primer indicador. El incremento del 220% por el número de casos de pruebas ejecutados para el segundo indicador. Logrando concluir que la automatización de pruebas impacta positivamente en la eficiencia de las pruebas de software en una consultora de TI.

Palabras clave: Automatización de pruebas, pruebas de software, eficiencia, automatización, consultora TI.

Abstract

The final objective of this research was to determine the impact of test automation on the efficiency of software testing in an IT consulting firm in Lima 2022. The research was applied, with a quantitative approach, using the hypothetical deductive method and a pre-experimental design. The technique used was observation and the instrument for data collection was the observation form. The Wilcoxon test and Student's t-test were also used to test the hypothesis.

The results showed that after the implementation of test automation they achieved a 74.68% decrease in the time spent per test case for the first indicator. The increase of 220% for the number of test cases executed for the second indicator. It can be concluded that test automation has a positive impact on the efficiency of software testing in an IT consultancy.

Keywords: Test automation, software testing, efficiency, automation, IT consultancy.

I. INTRODUCCIÓN

En estos últimos años enfrentamos una crisis sanitaria, causando un impacto global e inmediato en el comportamiento de las industrias para cubrir las necesidades de sus consumidores con la ayuda de la tecnología. Muchas empresas no estaban preparadas para estos desafíos, por la falta de capacidad tecnológica e implicó acelerar su transformación digital. En consecuencia, el software ha logrado convertirse en un producto imprescindible para toda empresa y clientes, de cómo el valor e intercambio de información ayuda a tomar mejores decisiones, a su vez se incrementa la complejidad del software al incorporar nuevas funcionalidades (Marín et al., 2020). En virtud de ello, en el presente estamos rodeados de tan excedente de productos de software desde aplicaciones web, móviles, microservicios y otros que crecen día a día.

El software es percibido dependiendo del cumplimiento de cada una de las funcionalidades, entre menos fallas presente mayor será su utilidad y su éxito (Balderas, 2018). Programar es una actividad humana, es propensa a errores y en este sentido no es extraño verse afectado por alguno de ellos, al usar algún software (Duque, 2022). Estos errores pueden incurrir en diferentes magnitudes desde la pérdida de dinero, tiempo, salud e incluso un error menor puede arruinar la experiencia de los usuarios.

De este modo las pruebas de software aportan en alcanzar con gran claridad el funcionamiento de un software y apoya a comprender que los requisitos logren cumplir con las necesidades del cliente. Por eso, las empresas se ven comprometidas en implementar modelos, normas, guías o estándares de calidad reconocidos en la industria para aportar a las pruebas de software (Febles et al., 2022). La investigación realizada en el año 2018 en Europa, participaron diferentes roles en la industria de desarrollo de sistemas de software de 16 países. Demostró que, para la entrega de un producto de software, el 59% representa los esfuerzos dedicados hacia las pruebas para asegurar la calidad del producto (Kuhrman, 2018).

A nivel internacional, en España el Banco BBVA es un caso de éxito en el continente europeo por su innovación digital y trabajo colaborativo al desarrollar su plataforma integral móvil. Para el control y mantenimiento de las aplicaciones móviles en cada país donde se encuentre BBVA. Parte de sus lecciones aprendidas sostiene que la elección correcta de metodologías y herramientas de automatización ayudan a que toda tarea participe en el proceso de pruebas del software sean menos laboriosas. De acuerdo con esto, contribuyó en mejorar la calidad del software y del mismo modo la reducción significativa del tiempo de validación de los despliegues de nuevas versiones de la aplicación. A pesar de las dificultades de la crisis sanitaria, alcanzaron a mantener la disponibilidad del 100% en sus canales digitales web y móvil; no siendo afectada por los continuos despliegues de versiones de sus canales digitales. Logrando ejecutar al mes 3,5 millones de prueba en 17 mil horas de pruebas automatizadas (BBVA, 2019) y (BBVA, 2022).

A nivel nacional se realizó el estudio donde se observa el impacto de una Testing Factory con la implementación de pruebas automatizadas de calidad de software mejoró la efectividad, estabilidad, eficiencia en tiempo, adicionalmente con resultados desarrollados en informes y satisfactorios hacia el cliente (Agurto, 2019).

Mientras tanto a nivel local, para tener el panorama de la problemática de la consultora de TI, fue necesario la elaboración del diagrama árbol de problemas (ver anexo 3). Por temas de seguridad se restringe el nombre de la institución, el propósito de la consultora de TI es ofrecer servicios de informática, consultoría y desarrollos de software a medida. Sus potenciales clientes son parte del rubro de educación, entretenimiento y medio periodísticos. La consultora se encuentra en crecimiento, en estos dos últimos años incorporaron a su equipo desarrollo de software el área para los aseguradores de calidad. Logrando formalizar el proceso de pruebas para sus potenciales clientes, en los desarrollos de aplicaciones web y backend. Con el fin de disminuir las incidencias en producción causadas por las fallas y errores, las cuales no fueron encontrados durante la fase de desarrollo. El equipo de aseguradores de calidad ejerce sus labores dentro de su proceso de pruebas con diferentes tipos de pruebas, desde funcionales, regresión, integrales y

de humo. Siendo una de la actividad más importante del proceso es el diseño y ejecución de casos de pruebas, las cuales son elaboradas de forma manual. En consecuencia, está afectando a la eficiencia de las actividades de las pruebas de software al generar en invertir más tiempo, provocando en incumplir las fechas de entrega del producto, las cuales fueron pactadas en las estimaciones entre la consultora y su cliente. El riesgo es alto hacia el cliente, porque deber cumplir con los objetivos del negocio. Otro de los problemas detectados es la por falta de documentación temprana como experiencia, se deja descuidado criterios y complejidades en los casos de prueba. Afectando la calidad del producto y experiencia de los usuarios e incurre a una necesidad de aumentar la cantidad de casos de prueba para lograr cubrir estas complejidades.

La empresa reconoce que es una amenaza no mejorar estos puntos, porque las empresas de competencia directa están incorporando nuevas estrategias hacia las pruebas de software, por ello ha dispuesto apoyar la presente investigación con el fin de resolver estos problemas.

La investigación reconoce la necesidad de resolver el problema descrito por la consultora, por ello, se formuló el problema general de investigación de la siguiente manera: ¿Cuál es el impacto de la Automatización de Prueba en la Eficiencia de las pruebas de software en una consultora de TI, Lima 2022?, con respecto a los problemas específicos, se formula a continuación las preguntas: ¿Cuál es el impacto de la Automatización de Pruebas en el tiempo medio por caso de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022? y ¿Cuál es el impacto la Automatización de Pruebas en el número de casos de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022?.

La presente investigación se justifica desde diferentes panoramas descritos a continuación: La justificación epistemológica, expresa lo razonable para comprender o explicar la investigación basada en el pensamiento crítico del conocimiento en confrontación y reflexión competente de las teorías científicas, para demostrar la validación correcta de la formulación del problema. Los

resultados obtenidos durante la investigación, demostrará la validación de las hipótesis expuestas, dependiendo de razonabilidad y certeza científica. Desde su justificación teórica, permitirá generar el interés y aprendizaje para fortalecer el conocimiento de las variables de la investigación. En base a las teorías que fortalecieron a las variables de la investigación fueron la teoría general de sistemas, la teoría de la automatización, la teoría de la calidad y la teoría de la eficiencia. En ese mismo contexto la justificación práctica, el desarrollo de la investigación permitió solucionar un problema, al aplicar conocimientos con respecto a la automatización de pruebas útiles para ejecutar próximos proyectos en beneficio a la consultora. Por otra parte, la justificación metodológica, es orientar a la investigación con un diseño experimental, con el fin de obtener resultados fidedignos, utilizando el método de observación para la recolectar los datos obtenidos en cada etapa de la investigación, en este caso del pre-test y de manera similar en el post-test, en donde los instrumentos e indicadores son fiables a partir de las validaciones de los juicios expertos.

Teniendo en cuenta el objetivo general se propuso lo siguiente: Determinar el impacto de la Automatización de Pruebas en la Eficiencia de las pruebas de software en una consultora de TI, Lima 2022. Además, se manifestó los siguientes objetivos específicos: (a) determinar el impacto de la Automatización de Pruebas en el tiempo medio por caso de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022 y (b) determinar el impacto de la Automatización de Pruebas en el número de casos de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022.

La hipótesis general de la presente investigación: La Automatización de Pruebas impacta positivamente en la Eficiencia de las pruebas de software en una consultora de TI, Lima 2022. De modo que las hipótesis específicas se definieron de la siguiente forma: (a) la Automatización de Pruebas reduce el tiempo medio por caso de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022 y (b) la Automatización de Pruebas incrementa el número de casos de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022.

II. MARCO TEÓRICO

Se consideró respaldar la presente investigación con un conjunto de recopilaciones de estudios preliminares, de hechos académicos a nivel nacional e internacional para sostener este estudio:

En cuanto a antecedente nacionales, nos referimos a Gutiérrez (2019), en su investigación, su objetivo principal fue cómo mejorar las pruebas funcionales mediante la implementación de la herramienta Badboy para automatizar las pruebas del aplicativo web. Los objetivos específicos, fueron comprobar sus efectos en los esfuerzo y costos de las pruebas funcionales del software, la investigación se basó en un diseño experimental, como también de tipo de investigación aplicada, adicionalmente sostenida con un nivel de investigación descriptiva y de enfoque cuantitativa. El alcance de la automatización con sus herramientas fue lograr la compatibilidad en las pruebas funcionales del aplicativo web, por eso lograron automatizar un módulo de mayor criticidad perteneciente al proceso de pagos, asimismo, en cada despliegue realizado en el ambiente de desarrollo se ejecutan los casos de pruebas automatizados usando la herramienta, donde se generaron los reportes, adicionalmente si algunos de los casos presentaran errores, automáticamente se guarda la captura de pantalla dentro del reporte. Finalmente, obtuvo resultados satisfactorios al reducir el esfuerzo invertido de horas jornadas de trabajo humano para la elaboración, ejecución y re-ejecutar pruebas un 90% y la reducción del 85% de costos en corregir los errores en la etapa de desarrollo.

Fernández (2018), el objetivo de su investigación fue la mejora en el área de calidad del Banco de Crédito, donde las pruebas de software fueron desarrolladas en la automatización de procesos, por el cual, se identificó dos dimensiones de sus variables de estudios, que son eficiencia y productividad. Presentando como objetivos específicos en determinar el mejoramiento de la automatización, hacia los indicadores correspondientes al tiempo de generación de data y la cantidad de data generada, para que la organización evite continuar con los problemas de demoras, incremento de costos y distribución de los recursos. La naturaleza de la investigación fue aplicada, con enfoque cuantitativo y diseño pre-experimental. Indicando que, para la automatización utilizó la herramienta HP Unified Functional

Testing para destacar su accesibilidad de la creación y mantenimiento de las pruebas en aplicaciones web. Terminando con resultados en la disminución del 56,78% para el tiempo de generación de data y logrando incrementar un 7,71% en productividad de la data generada en beneficio de la empresa.

Medina (2020), en su investigación desarrollada en una compañía de líneas de belleza, su objetivo principal fue de qué manera logra reducir significativamente la cantidad excesiva de pruebas manuales, los tiempos de ejecución de prueba de regresión y mejorar el entendimiento de los requerimientos funcionales del negocio en base a la implementación de la automatización de pruebas. El diseño que planteó para la investigación fue transversal, correlacional-causal, con enfoque cuantitativo, obtuvo como resultado una reducción del 60% de casos de prueba ejecutadas manualmente y la reducción del 50% en el tiempo de ejecución de pruebas manuales, donde se ejecutaron los casos de pruebas automatizadas en función de los registros y consultas de las asesoras de la compañía de belleza en el aplicativo móvil. Para el desarrollo de la herramienta de automatización consideró las buenas prácticas de Behaviour Driven Development, porque mediante el uso del lenguaje Gherkin, le permitió el entendimiento de los criterios de aceptación para los usuarios del negocio y al equipo de desarrollo. Adicionalmente, incorporó el framework de CodeceptJS, con el patrón Page Object Model y generación de reportes.

En la Universidad Tecnológica del Perú, Huamán (2021), en su tesis titulada Modelo de automatización de pruebas para optimizar la gestión de la calidad de software en la empresa Cifin, 2021. Debido a la alta demanda de nuevos requerimientos de sus clientes, a pesar del gran esfuerzo de los analistas de calidad, generaba demora e incumplimientos en el tiempo de entrega por el impacto al realizar las pruebas de manera manual. Su objetivo principal fue mostrar una mejora en la gestión de la calidad del software dentro de Cifin en 2021. La investigación fue aplicada, y tuvo un diseño transversal correlacional-causal, bajo un enfoque cuantitativo. El modelo implementado fue un marco ágil y colaborativo, junto con la automatización de pruebas. La automatización pudo ejecutar 11 438 pruebas, mejorando su eficiencia a 94,38% que las pruebas anteriores realizadas

manualmente. Adicionalmente los casos de pruebas automatizados, los reutilizó en las regresiones, minimizando el tiempo de ejecución y redujo la cantidad de defectos al considerar las pruebas en etapas tempranas del desarrollo. Utilizó como modelo de automatización el Serenity BDD, por su flexibilidad y adaptabilidad. Logrando automatizar 1937 casos de pruebas, pertenecientes a 10 aplicaciones de alta prioridad del negocio.

Chávez (2021), su objetivo general de su investigación fue de qué manera la automatización de los casos de prueba permite mejorar el proceso de pruebas de un sistema de reclutamiento de nombre RPO. Debido a que el sistema RPO de la empresa Proximity, se encontraba presente en 9 compañías y en diferentes idiomas, se debía comprobar que los cambios realizados no impacten a ninguno de sus funcionamientos. Entonces diariamente se realizaban estas pruebas de humo de forma manual, la cual causaba demoras por ser pruebas muy complejas y repetitivas. Con respecto al objetivo de la investigación, se logró crear un proyecto de automatización UI, mediante las herramientas de Selenium y TestNG, permitiendo reproducir las acciones principales de los usuarios mediante la automatización. Finalmente, se encontró la variación en el tiempo de ejecución de pruebas de humo, antes el analista de calidad le tomaba 80 horas al mes ejecutarlas de forma manual y después de la implementación de las pruebas manuales logró reducir a 35 horas al mes, concluyendo que la automatización de pruebas logró una reducción del 56% en el tiempo de ejecución de pruebas de humo para el aplicativo de reclutamiento de la empresa Proximity.

En el ámbito internacional se tiene a Caiza (2019), el objetivo de su investigación fue en implementar métricas al proceso de control de calidad en proyectos de desarrollo de software para la empresa Logiciel en Ecuador. Permitiendo en tomar las mejores decisiones e integrando a las partes interesadas del negocio en la empresa con respecto a los procesos de pruebas y a la calidad del producto final. El estudio fue basado en la metodología de Investigación-Acción, por enfocarse en la exhausta revisión literaria relacionados al tema y la alineación con los objetivos del negocio. Logró considerar en implementar las métricas basadas en la norma ISO 25023 y fundamentos de ISQTB. Finalmente, las

evaluaciones de cada fase del proceso de pruebas involucraron a todos los integrantes. Por consiguiente, los resultados obtenidos lograron formular mejoras al proceso e influyó la calidad del producto final.

Irshad et al. (2021), en su investigación “A Systematic Reuse Process for Automated Acceptance Test: Construction and Elementary Evaluation” en Alemania, el objetivo fue mejorar el desempeño, esfuerzo y condiciones facilitadoras en las pruebas de aceptación con la automatización, logrando realizar un desglose de actividades sistémicas que apoyaron al desarrollo, evaluación de los costos y beneficios de la reutilización de pruebas de aceptación automatizadas. Donde se reunieron a cinco expertos en la industria con altos conocimientos en pruebas de aceptación automatizadas, para realizar una demostración que en 7 de 25 líneas fueron reutilizadas, sin embargo, éstas 7 líneas realizan funcionalidades que requiere mucho tiempo para desarrollar, por lo que el costo del ahorro fue de casi el 75% (1 semana al reutilizar frente a 4 semanas de tiempo de desarrollo desde cero).

Bahaweres (2020), en su investigación titulada “Behavior driven development (BDD) Cucumber Katalon for Automation GUI testing case CURA and Swag Labs” en Indonesia, su objetivo fue diseñar la automatización de pruebas GUI con la implementación de Cucumber BDD usando Katalon Studio. Para ello necesitó crear historias de usuarios con respecto a las dos aplicaciones de comercio electrónico para CURA y SwagLabs, cada historia describe los criterios de aceptación del usuario. Posterior a ello, desplegó Cucumber para definir pasos de prueba en la herramienta de Katalon Studio y crear runners de casos de prueba. La herramienta le permitió generar fácilmente las pruebas automatizadas. Finalmente, logró alcanzar un tiempo de respuesta promedio en el caso de prueba de la web CURA de 1.92 minutos y un tiempo de respuesta promedio en el caso de prueba web de Swag Labs de 2.1 minutos.

Cortés (2020), el objetivo de su investigación fue determinar si al implementar la automatización de pruebas de regresión logre reducir el tiempo de entrega de nuevas versiones de software, en Chile. Encontró tres problemas en el área de

calidad, al no existir una formalización del proceso y sin medición, causaba una gran incertidumbre a la toma de decisiones, el segundo fue que los casos se ejecutaban de forma manual en consecuencia requerían un alto consumo de recursos en horas hombre y por último no existía la documentación de las pruebas. Entonces su objetivo fue definir estándares y metodologías de los procesos de calidad con documentación necesaria, para cada 4 subprocesos y cumplir con los plazos de estimación de la ejecución de pruebas. Siendo este último, la implementación de automatización de pruebas con la herramienta Selenium, permitieron probar cada nueva versión del software, por el cual se obtuvo el resultado de la reducción del 60% a 37,42% en el tiempo de ejecución, logrando permitir a la empresa Surecomp, pueda dar inicio al nuevo servicio de automatización y con capacidad de atender las necesidades de sus múltiples clientes.

En cuanto a la teoría, inicialmente consideramos la teoría general de sistemas, que según Ossa (2016), es una fusión de saberes que se ocupa de la consideración global del fenómeno en estudio, más que de comprender el todo estudiando las partes. Tal como la ciencia tradicional enseña el desarrollo del conocimiento. Domínguez y López (2016) menciona que, cada parte cohesionan entre sí para mantener la estructura, generando su versatilidad con la seguridad plena al producirse de forma correcta su enfoque sistémico desde su visión integral y global, tiene la capacidad de corregir oportunamente ante una amenaza. De igual manera, Arnold y Osorio (1998) lo consideran un mecanismo de integración entre las ciencias naturales y sociales. A su vez, es una herramienta fundamental para la formación y preparación de científicos, con propiedades subdivididas en estrategias desde dos perspectivas de sistemas. Su primera distinción es conceptual, centradas en la relación entre el todo y sus partes. El segundo es centrado en los procesos de frontera. Morote et al. (2018) menciona que, cada sistema desarrolla estrategias para regular la relación entre lo interior y lo externo, las oportunidades y los recursos necesario. La presencia de supervivencia del sistema, según la teoría, se esfuerza en lograr un cierto grado de funcionalidad relacionada con los objetivos compartidos por el mismo sistema. Manteniendo la coherencia como un sistema organizado, efectivo y adaptable a las exigencias de su entorno. Gómez

(2020), cita a Bertalanffy (2000), quien coincide que la teoría de sistemas incluye una gama de teorías y enfoques, incluyendo la teoría de la información, la teoría de los juegos, cibernética, teoría de automatización, teoría en redes y otras. Es correspondiente para el análisis dedicado a los sistemas, contemplando diversos modelos en relación con la naturaleza del caso y particularidades operacionales de diferentes enfoques científicos, siendo posible conservar las relaciones e integración entre las disciplinas.

Así mismo, se tiene la Teoría de la Automatización, argumentada por Chiavenato (2006), que estudia tanto de manera abstracta como simbólica las formas en que un sistema puede procesar la información recibida, pudiendo realizar en cierta medida una serie de operaciones. Similar a los procesos físicos y mentales humanos, logrando gestionar la corrección de los errores que se produzcan durante su actividad, según criterios preestablecidos. Los equipos automatizados pueden proporcionar funciones de seguimiento, almacenamiento y toma de decisión. Según Brunet et al. (2003), las nuevas tecnologías promueven la sustitución de la fuerza de trabajo, al considerar a la automatización la sustitución de personas por máquinas para realiza la misma función. Medina y Guadayol (2010) definieron que, es una disciplina de la ingeniería de gran cobertura para supervisar en tiempo real y controlar operaciones en los procesos industriales.

López (2020), la teoría de eficiencia de Taylorismo, dándole como referencias a sus principales perspectivas donde menciona a Taylor (1911), donde se propuso que el trabajo debe ser dividido en tareas para aumentar la productividad, se conllevó a realizar una producción más audaz, por el cual se mejoró los resultados empresariales en forma eficiente, donde se puede controlar el tiempo de realización de cada tarea por parte del trabajador a la perfección del objetivo fundamental es producir más con menor coste a través del aumento de la productividad de la mano de obra. Asimismo, se pretende estandarizar los procesos de producción logrando analizar la eficiencia al máximo

Según Huertas (2014), nos indica que la teoría de la eficiencia consiste en el criterio que se desarrolla a imagen y semejanza de la idea de eficiencia energética

que surge en el ámbito de la física y mecánica, del siglo XIX, donde se ha explicado la conversión en los ejes del análisis económico del bienestar y gran parte de los análisis tecnológicos que logra los resultados de la creatividad empresarial en sus aspectos predeterminados.

Según Gismano (2012) nos indica que, las teorías de la eficiencia de Taylorismo, es el estudio sistemático de tiempo y movimiento, orientado específicamente al establecimiento del modo más adecuado de realizar la tarea, exige por parte del obrero, donde se desarrolló de un conjunto de habilidades y competencias particulares donde se manejó en los requerimientos de las tareas que cumplen de acuerdo a sus cualidades y diseño de herramientas para lograr la eficiencia del obrero, y principalmente al reducir los movimientos innecesarios.

La teoría de la calidad, en la historia la humanidad le ha dado gran importancia este concepto, para empezar Deming (1989), definió este concepto como ese punto predecible de semejanza y fiabilidad de bajo costo, dado que es el grado de someterse a las exigencias del mercado y la tendencia de mejora continua. En base a su destreza en la industria americana y lecciones aprendidas en Japón, logró mediante 14 pasos explicar cómo transformar la gestión en las empresas. No obstante, para Deming reconoce la gran aportación del ciclo de Stewart, una serie de cuatro procedimientos para perseguir la mejora y descubrir la causa desde una perspectiva estadística aplicada a la calidad. Chacón y Rugel (2018) concluyen que, la calidad y una cultura de mejora continua, logran los objetivos de permitir que los usuarios de un servicio o sistema estén más satisfechos, dentro de sus políticas organizacionales.

En cuanto a los enfoques conceptuales tenemos los siguientes: Con respecto a la variable independiente Automatización de Pruebas. Para Chopra (2018), consiste en que, mediante un conjunto de herramientas logra reproducir aspectos mecánicos de una tarea de prueba y evaluar de manera automática; con el fin de reemplazar o complementar las pruebas manuales por su fácil uso. Para Srivastava (2021), es una técnica de prueba de código controlada por la máquina, para ejecutar un conjunto de pruebas. Medina (2020) indica que, las pruebas

automatizadas se caracterizan por ser rápidas, fiables, repetibles, programables y reusables.

Es más rápida, porque reduce el tiempo de la prueba por ser capaz de ejecutar más pruebas en un lapso menor, al no necesitar intervención humana y permite mejorar la eficiencia de las pruebas (Islam y Quadri, 2020). Mejora la fiabilidad, debido a que al emplear una herramienta de automatización de pruebas hace posible grabar la verificación del conjunto de pruebas y reproducir cuando sea necesario y reduce el riesgo de errores humanos (Srivastava, 2021). Es repetible, porque la automatización de pruebas realiza tareas repetitivas correctamente, asegurando que los resultados se registran en cada interacción (Battina, 2019). Son programables, porque el código de las pruebas de automatización permite establecer datos de entrada del sistema a verificar, comparar resultados esperados y reales, adicionalmente generar informes de verificación. Su objetivo es reducir la cantidad de casos de pruebas que deben ejecutarse manualmente y no eliminar por completo las pruebas manuales. (Srivastava, 2021). Su reusabilidad, permite reutilizar scripts del código de la automatización en función donde sea necesario, en vez de duplicar tiene la característica de rehusar paquetes, clases, objetos de un mismo repositorio (Islam y Quadri, 2020).

Barus (2019) sostiene que, son la mejor manera de aumentar la efectividad, la eficiencia y el alcance de las pruebas de software. Sin embargo, Musliu y Jashari (2021), nos dicen que una implementación incorrecta, las pruebas automatizadas aumentarían los costos y esfuerzos, incluso podrían ser menos efectivas que las pruebas manuales. Para Mahee et al. (2019), la automatización de la prueba es aplicable para diferentes tipos de prueba de caja negra como de caja blanca. El proceso de probar un programa sin consultar el código fuente se llama prueba de caja negra. Las pruebas de caja blanca se ocupan de ejecutar casos especiales, en ramas y rutas específicas en el código fuente.

Blokehead (2016) sostiene que, el ingeniero de pruebas es responsable de mantener el marco de automatización de pruebas. Un marco sólido debe ser capaz de realizar tareas como la limpieza de pruebas, la generación de informes,

recopilación de resultados y la concurrencia mientras se utilizan principios ágiles.

El marco Scrum se usa más comúnmente en metodologías ágiles en proyectos de software. Proporciona un conjunto de mejores prácticas que tienen como objetivo brindar valor a los clientes rápidamente y pueden optimizar los flujos de trabajo, reducir el tiempo de desarrollo y mejorar la transparencia organizacional (Tavares, 2019). Para complementar Galvan et al. (2021) indican que, es un proceso de desarrollo participativo y colaborativo que incluye tres actores comunes son el cliente, el facilitador del proyecto y el equipo de desarrollo; Con un conjunto de prácticas ágiles personalizadas y flexibles que buscan el máximo aprovechamiento de los recursos necesarios, evitando tareas y elementos sin valor añadido. Entregando funcionalidad útil cada 2 a 4 semanas.

Según García (2022) sostiene que, Page Object Model (POM), es el patrón de diseño más popular usado en conjunto con Selenium WebDriver. Arjun y Shahid(2019) indican que, algunas de las ventajas de usar POM es que todos los componentes web se pueden almacenar en un repositorio de objetos que contiene todos los componentes web para cada página web. El POM usa la clase Page Factory para crear objetos. Se archivan todos los elementos web. Todos los métodos y objetos serán seleccionados para cada sitio, permite un mantenimiento de código más fácil

García (2022) define que, Selenium es un proyecto público de código abierto que permite la automatización del navegador web. El componente central de proyecto Selenium es Selenium WebDriver, una librería que le permite controlar los navegadores mediante programación. La automatización depende de las capacidades únicas de cada navegador.

Según Faragó et al. (2020), existen varios enfoques de desarrollo de software ágil, pero la principal es el desarrollo basado en el comportamiento por sus siglas en inglés es BDD, se centra en el comportamiento del usuario para especificar los requisitos del software en construcción. Cada escenario del usuario se considera un único caso de prueba basada en la estructura Dado, Cuando y Entonces.

Cucumber es una herramienta que admite el desarrollo impulsado por el comportamiento (BDD). Lee la especificación de ejecución en texto sin formato y verifica que el programa está haciendo lo que dice esa especificación. La especificación incluye varios ejemplos o escenarios (Musliu y Jashari, 2021).

Todos estos conceptos, se consideraron para construir el marco de automatización de pruebas, bajo la metodología Scrum, POM para reducir el mantenimiento del código por la reutilización del código, Cucumber por la facilidad de comprensión de los criterios de la prueba y Selenium, para controlar los elementos del aplicativo web.

Referente a la variable dependiente Eficiencia. Cequea (2012), es el criterio económico que revela la capacidad administrativa de producir el máximo resultado con el mínimo de recursos, energía y tiempo, por lo tanto, se optimiza utilización de los recursos disponibles para la obtención de resultados deseados. Según Mateus (2016), es la capacidad de disponer de determinados recursos para lograr un objetivo, donde se realiza la mayor cantidad de trabajo con el mínimo esfuerzo y para los que enfrenta trabajos de mayor tamaño requiere de mucho tiempo y dedicación, para conseguir este objetivo es necesario fraccionar las tareas en pequeños módulos o etapas, permitiendo analizar con más detalle cada módulo, rectificar posibles errores y continuar con la siguiente tarea. Asís (2007), sostiene que puede aplicarse tanto a unidades organizativas, como a los diferentes centros de trabajo que forman una organización cuando cumple los objetivos esperados con el mínimo gasto posible de recursos. Asimismo, para González et al. (2018) sostiene que, los sistemas de automatización, big data, ciberseguridad y otros conceptos que forman parte de la industria 4.0 son factores que le permiten adaptarse de manera eficiente a las necesidades del mercado al que se dirige el producto y está encaminando a la optimización global de las industrias, permitiendo mejorar la eficiencia, reducción de costes y el incremento de la calidad.

Por otra parte, Hurtado (2005) recupera que, la importancia de los indicadores debe permitir al dueño del proceso medir la gestión relacionada con la optimización

de los recursos, siempre que exista un indicador de eficiencia, este debe relacionar un recurso con el número de unidades o actividades realizadas, para poder evaluar el consumo de recursos. La redacción del indicador depende del lenguaje que deseen la organización.

La eficiencia de las pruebas de software mide la rentabilidad de las pruebas frente a los recursos de una organización, correspondiendo al esfuerzo total de sus actividades. En tanto, la eficiencia óptima de la prueba es aquella capaz de alcanzar un estándar de calidad de software adecuado a un menor esfuerzo (Mobaraya y Ali, 2019).

Las pruebas de software según Pressman (2005) sostiene que, se centran en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales, es decir la realización de las pruebas para la detección de errores. Sommerville (2011) nos dice que, existen dos principales metas del proceso de pruebas del software es demostrar al desarrollador y al cliente cumple con los requerimientos en base de pruebas. Considerando todas las características del sistema que se incorporarán en la liberación del producto y encontrar situaciones donde el comportamiento del software sea incorrecto, indeseable o no esté de acuerdo con sus especificaciones. Así mismo, ISO/IEC/IEEE (2013), define como un modelo de proceso de prueba general que se puede adaptar a cualquier desarrollo de software y ciclo de vida de la prueba, basándose en tres capas de cobertura: Especificaciones de organización de prueba, Gestión de pruebas y el Proceso dinámico de pruebas.

ISTQB (2018) nos dice que, es un conjunto de actividades para alcanzar los objetivos establecidos en conjunto con las estrategias de pruebas de una organización, constituida con las siguientes principales actividades: la planificación de prueba, el análisis, el diseño y la implementación de prueba, información del avance, los resultados de las pruebas y la evaluación de la calidad de un objeto de prueba. Así tenemos a Mera (2016), manifiesta la importancia de desarrollar un plan de pruebas y una matriz en este proceso. Entonces, al ejecutar los casos de prueba, se puede determinar si el caso está funcionando correctamente y, por lo tanto,

establecer como conforme o de lo contrario un rechazo, exponiéndolo en métricas, ayudando a estimar el tiempo, esfuerzo y asignación; de otra forma determinan también los niveles de riesgo para que el equipo de desarrollo pueda ajustar los flujos de actividades y optimizar así los recursos y el talento de los equipos de desarrollo. Para complementar Kenett et al. (2018) mencionan que, muchas empresas de todos los tamaños abordan las pruebas basadas en riesgos como factor guía en todas las fases del proceso y utilizan la idea directa de centrar las actividades de prueba en los escenarios que desencadenan las situaciones más críticas de un sistema de software (pp.4).

En concordancia a ISO (2013), ISTQB (2018) y Kenett et al. (2018), durante el diseño de la prueba, los objetivos generales de prueba definidos en el plan de prueba se transforman en condiciones de prueba tangibles y casos de pruebas abstractos. La implementación de la prueba comprende en tareas para hacer que los casos de pruebas abstractas sean ejecutables, incluyendo preparar pruebas y datos de prueba, proporcionar registros de soporte o escribir scripts de prueba, que son necesarios para permitir la ejecución automatizada de casos de prueba. En la ejecución de prueba, se ejecutan los casos y se registran y monitorean todos los detalles relevantes de la ejecución. Finalmente, en la fase de evaluación, se evalúan los criterios de salida y los resultados de la prueba se registran en un resumen de informe de prueba.

Retomando lo anterior mencionado y nuestra variable dependiente que es la eficiencia, se ha considerado las siguientes dimensiones:

La primera dimensión es el uso de recursos. Judge et al. (1997) indica que es necesario disponer de una cantidad de recursos inactivos disponibles para cuando una organización necesita adaptarse al cambio. García et al. (2019) reconoce que el uso de recursos es la dimensión de la eficiencia, según se definan sus indicadores desde un punto de vista económico entre la retención de clientes o captación de nuevos, calidad, costos y tiempos de respuesta, contribuye a mejorar de los resultados de un proceso productivo.

Para culminar, la segunda dimensión es la productividad laboral. Gupta (1983), muestra que la productividad laboral se expresa como la relación entre la meta laboral que pretende lograr y la cantidad de recursos gastados. Muestra la eficiencia de recursos humanos en la producción de bienes. Calculando el valor de la producción producida por el trabajador por unidad de tiempo, en un intervalo de hora. Guisado et al. (2016) argumenta que, es importante aumentar la productividad laboral porque afecta directamente a la rentabilidad y continuidad de una empresa. En cuanto García et al. (2019) dice que, la eficiencia es como una combinación específica de factores de producción que puede alcanzar niveles máximos de recursos, mientras incurre en costos mínimos. Al disponer una cantidad de recursos que se utilizan para producir bienes y servicios.

En consideración a los indicadores de la variable dependiente eficiencia, tenemos los siguientes: Tiempo medio por caso de prueba y número de casos de prueba ejecutados por hora.

El primer indicador de acuerdo con la dimensión de uso de recursos es el tiempo medio por caso de prueba. Según Bahaweres(2020) explica que, es el promedio del tiempo de respuesta de la ejecución de un caso de prueba.

Para el segundo indicador en relación con la dimensión de productividad laboral tenemos el número de casos de prueba ejecutados por hora. Para Leños (2018) señala que, el número de casos de prueba puede ser infinito para probar un software. Porque alcanzar a probar un software totalmente es imposible, pero sí tener la intención de aumentar la cobertura de la prueba para cubrir las complejidades críticas. Osherov (2014) destaca que, existen técnicas automáticas para su generación de forma eficiente y logre cumplir con los criterios.

III. METODOLOGÍA

3.1. Tipo y diseño de Investigación

3.1.1. Tipo de investigación

El propósito de la presente investigación fue de tipo aplicada, según Tamayo (2004), este tipo de investigación se domina también cómo activa o dinámica, porque depende de sus descubrimientos y contribuciones teóricas, para confrontar la teoría con la realidad. Por su parte, Muñoz y Cordero (2017), las como fundamental, porque dispone de un espacio para que los objetos de estudio desempeñen sus roles y permitan contribuir de forma activa. Con respecto a Esteban (2018) indica que, la investigación es aquella que estudia problemas concretos en condiciones y características concretas. Orientándose a su aplicación inmediata y no al desarrollo de teorías.

El estudio se trabajó bajo el enfoque cuantitativo, porque se recolectó y analizó datos numéricos para realizar su tratamiento estadístico. Este enfoque logró una selección subjetiva de indicadores para algunos elementos de procesos y hechos de la investigación (Supo y Caverro, 2014). El método que se definió fue el hipotético deductivo, porque plantea hipótesis y deduce las consecuencias; que, a través de los hechos empíricos, operacionalización de las variables e hipótesis permite verificar las hipótesis planteadas y abordarlas en las conclusiones (Hernández y Mendoza, 2018).

3.1.2. Diseño de investigación

Adicionalmente, se consideró un diseño de investigación experimental, según Alan y Cortez (2018) indica que, mediante este diseño el investigador al disponer de una o más variables no verificadas, realiza la manipulación de ellas y las condiciona a un estricto control, con la meta de aclarar el modo y las razones que se ha concretado el fenómeno. En tanto, se planteó de tipo pre- experimental, debido a que implica el análisis de un único grupo. Por la cual se necesitó aplicar una preprueba antes de estímulo, después se implementa el tratamiento y se le realiza una posprueba (Hernández et al., 2014).

3.2. Variables y operacionalización

Variable independiente: Automatización de pruebas

Dado que la variable es cuantitativa, cabe resalta que el alcance del objetivo de la investigación no se considera para ser medida.

Definición conceptual: La variable de Automatización de pruebas según Chopra (2018) consiste en que, mediante un conjunto herramientas logre reproducir los aspectos mecánicos de una tarea de prueba y evaluar de manera automática; con el fin de reemplazar o complementar las pruebas manuales por su facilidad de uso.

Definición operacional: La Automatización de pruebas consiste en que, las herramientas usadas para crear script de pruebas son eficientes y efectivas. Por ser reusable, programable, iterativo y confiable; ayuda a acelerar el proceso en menos tiempo, incrementa la cobertura, reduce el esfuerzo y disminuye los costos.

Variable dependiente: Eficiencia

La variable es de tipo cuantitativa discreta, medido a través de indicadores del estudio con una escala de medición en un intervalo determinado.

Definición conceptual: Eficiencia según Cequea (2012), es el criterio económico que revela la capacidad administrativa de producir el máximo resultado con el mínimo de recursos, energía y tiempo, por lo que es la óptima utilización de los recursos disponibles para la obtención de resultados deseados.

Definición operacional: La eficiencia se basa en alcanzar los mejores resultados deseados con el mínimo recursos.

Indicadores: Con respecto a la variable dependiente, se consideró dos indicadores de medida, en relación con las dimensiones de tiempo y productividad

laboral; el primer indicador es el Tiempo medio por caso de prueba y el segundo indicador se definió como Número de casos de prueba ejecutados por hora.

Escala de medición: Se utilizó la escala de intervalos, por su medición cuantitativa al medir la diferencia entre dos variables. En de acuerdo con los autores Steinberg y Price (2020), se comprende de datos discretos por contener una gran cantidad de números posibles de valores. Con la propiedad de ser conocida y constante, es posible conocer la magnitud de los intervalos. Caracterizada por una unidad de medida.

A continuación, se describe en la tabla 1, la matriz de operacionalización de la variable eficiencia.

Tabla 1

Matriz de operacionalización de la variable dependiente eficiencia

| Dimensión | Indicador | Instrumento | Unidad de medida | Fórmula |
|-----------------------|--|----------------------|------------------|--|
| Uso de recursos | Tiempo medio por caso de prueba | Ficha de observación | Unidad | $X = \text{Tiempo total en ejecutar los casos de prueba} / \text{Total de casos de prueba ejecutados}$ |
| Productividad laboral | Número de casos de prueba ejecutados en una hora | Ficha de observación | Unidad | $X = \text{Número de casos de prueba ejecutados} / \text{Horas-hombre laboradas}$ |

3.3. Población, muestra y muestreo

3.3.1. Población

Hernández y Mendoza (2018) sostienen que, la población es la agrupación de todos los casos de acorde con una serie de especificaciones planteadas por el investigador para delimitar su desarrollo en el proyecto. Según López (2004), es el conjunto de objetos o personas de los que se quiere descubrir algo en una investigación. En tanto, Majid (2018) indica que, la población objetivo del estudio que se pretende estudiar también es conocida como la población de interés. Permitiendo reflexionar sobre las características ideales del participante de la investigación, mediante criterios de elegibilidad, de acuerdo con el entorno del estudio y estrategias optimizan su reclutamiento. Patino y Ferreira (2018) sostienen que, los criterios de elegibilidad determinan si la población está calificada o no para participar en un estudio de investigación.

De esta manera, para la elaboración de la investigación a desarrollar se consideró una población de 32 ejecuciones de pruebas del proceso de pruebas de software de la consultora de TI durante 2 sprint, teniendo en cuenta que cada sprint tiene una duración de 3 semanas.

Criterios de inclusión: Son la ejecución de un conjunto de casos de pruebas de regresión de las pruebas de software, involucradas a las funcionalidades principales del aplicativo web educativo, desarrollado por la consultora de TI.

Criterios de exclusión: No tiene relación a la ejecución de pruebas de nuevas funcionalidades del aplicativo, cómo también a las pruebas de performance y pruebas de humo.

Tabla 2

Población de la investigación

| Población | Cantidad | Indicador |
|---------------------|----------|--------------------------------------|
| Ejecución de Prueba | 32 | Tiempo medio por caso de prueba |
| Ejecución de Prueba | 32 | Número de casos de prueba ejecutados |

3.3.2. Muestra

Una muestra puede identificarse, en el marco de una investigación científica, es parcial o parcialmente representativa de todos los individuos o unidades estudiadas (Ñaupas et al, 2018). Se tomo cómo referencia la fórmula de marco muestral del anexo 4, dando como resultado una muestra de 30 fichas de registro.

3.3.2. Muestreo

Se consideró un muestreo no probabilístico, tal como Hernández et al. (2014) indican que, la selección del subgrupo de la población no se basa en la probabilidad, sino en las características de la investigación. La técnica de muestreo no probabilístico fue de tipo accidental, porque se espera recolectar los elementos necesarios hasta completar el tamaño de la muestra deseada (Otzen y Manterola, 2017). Las muestras fueron recolectadas durante los 2 sprint, se esperó completar una muestra de 30 ejecuciones de prueba.

3.4. Técnicas e instrumentos de recolección de datos

La técnica de recolección de datos utilizada en esta investigación fue la observación. Según Hernández y Mendoza (2018) afirman que, este método de recolección de datos consiste en registrar comportamientos sistemáticos, válidos y confiables de las situaciones observables del objeto en estudio.

Instrumentos de recolección de datos, en este sentido se consideró la ficha de observación cómo instrumento de recolección de datos para la presente investigación. Campos y Lule (2012), el instrumento que permite al observador posicionarse sistemáticamente en la realidad del objeto de estudio; es el recurso de la recolección y obtención de datos de un hecho.

En la tabla 3, visualizamos la tabla de la ficha técnica del instrumento de la presente investigación.

Tabla 3*Ficha técnica del instrumento*

| | |
|--------------------------------------|--|
| Nombre del Instrumento | Guía de observaciones de medición del indicador |
| Autor | Wendy Ly Morales Yovera |
| Año | 2022 |
| Tipo de Instrumento | Ficha de observación |
| Objetivo | Determinar cómo impacta la automatización de pruebas en la eficiencia de pruebas de software en una consultora de TI, Lima 2022. |
| Indicadores | - Tiempo medio por caso de prueba - Número de casos de prueba ejecutados por hora |
| Número de observaciones a recolectar | 30 |
| Aplicación | Directa |

Validez, considerada en la presente investigación es de juicio de expertos, constituida por 3 miembros de la plana docente universitaria, visualizada en la tabla 4 de los participantes de la validación del instrumento. Según Hernández y Baptista (2014), es la autenticidad del instrumento permitiendo el rediseño del instrumento de medición mida la variable de manera correcta, para luego someterlo a la confiabilidad.

Tabla 4*Validación del instrumento de recolección de datos*

| DNI | Grado académico, apellidos y nombres | Institución donde labora | Calificación |
|----------|--------------------------------------|-------------------------------|--------------|
| 17930425 | Mg. Tejada Ruiz, Roberto | Universidad César Vallejo | Aplicable |
| 42097456 | Dr. Acuña Benites, Marlon | Universidad César Vallejo | Aplicable |
| 47190643 | Mg. Tarmeño Juscamaita, Esther | Universidad Privada del Norte | Aplicable |

Confiabilidad, según Ñaupas et al. (2018) sostienen que, la confiabilidad es la no variación significativa de resultados provenientes de la aplicación del instrumento indicado en circunstancias de semejanza o de isomorfismos en la naturaleza propia en donde se desarrolla la investigación. Según Corral (2009) sustenta que, existen instrumentos debido a su naturaleza de recolección de datos no ameritan el cálculo de la confiabilidad, este conjunto está conformado por: entrevistas, escalas de estimación, lista de cotejo, ficha de observación, hoja de registros, inventarios, entre otros. Finalmente, la investigación presentada utiliza ficha de observación de modo que se está considerando la consistencia descrita líneas arriba.

3.5. Procedimientos

El procedimiento de la información para el presente estudio de investigación, se inició la realización de un instrumento de recolección de datos a través de fichas de observación, después se continuó con la validación de tal instrumento mediante juicio de expertos y posteriormente el desarrollo de la aplicación de la investigación.

3.6. Método de análisis de datos

Con respecto al análisis de datos, se aplicará el software IBM SPSS Statistics y Microsoft Excel para el apoyo del pretest y posttest; a fin de un análisis descriptivo se emplearán tablas y figuras para orientar las tendencias centrales en base a la media, con el fin de comprender los resultados del análisis descriptivo de cada indicador. Con respecto al análisis inferencial, se define comprobar la normalidad por cada indicador mediante la prueba Shapiro-Wilk, si se ajusta o no una distribución normal y se empleará el método no paramétrico de Wilcoxon, en caso contrario se utilizaría el método T-Student para realizar el contraste de la hipótesis.

3.7. Aspectos éticos

Este estudio se respetó fielmente a las normas éticas establecidas por la Universidad César Vallejo, de acuerdo con la Resolución de Consejo Universitario N°0722-2021/UCV, velando por la transparencia y veracidad de la información. En la Resolución de Consejo Universitario N°0722-2021/UCV, el artículo 9 de la política anti-plagio, de la Universidad César Vallejo, fomenta la autenticidad de las

investigaciones de la casa de estudio, considerando el plagio como delito, por ello brinda el acceso y disponibilidad del software Turnitin con el objetivo de no superar el índice de similitud del 25%.

Los resultados se obtienen respetando la confidencialidad de los datos y de acuerdo con la ética profesional del investigador.

IV. RESULTADOS

A continuación, se presenta los análisis descriptivos e inferencial, en esta última se desglosa la prueba de normalidad y la prueba de contraste de hipótesis de la presente investigación pre experimental.

4.1 Análisis descriptivos

La presente investigación, examinó los resultados descriptivos, obtenidos del pretest y postest de los indicadores Tiempo medio por caso de prueba y número de casos de prueba ejecutados por hora. Desde la data recuperada en las fichas de observación, luego fueron registradas al software de SPSS Statics para exportar los datos en medidas descriptivas y exponerlas tanto en tablas como en figuras para la consistencia del análisis.

Análisis descriptivos del indicador: Tiempo medio por caso de prueba

Tabla 5

Medidas descriptivas del tiempo medio por caso de prueba

| | | N | Mínimo | Máximo | Media | Desviación Estándar |
|---------|---------------------------------|----|--------|--------|-------|---------------------|
| Pretest | Tiempo medio por caso de prueba | 30 | 12.00 | 17.00 | 14.85 | 1.56512 |
| Postest | Tiempo medio por caso de prueba | 30 | 1.94 | 5.29 | 3.76 | 1.11344 |

Nota: Datos procesados en SPSS Statics 25.

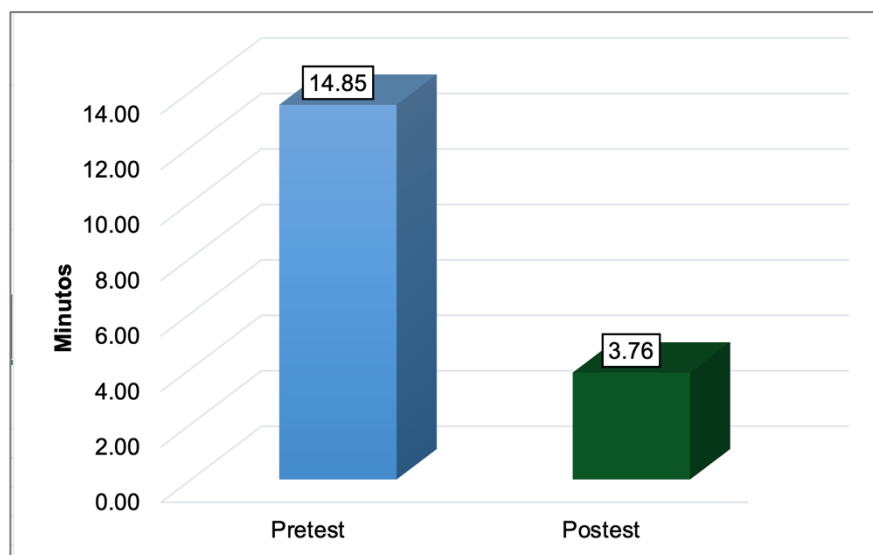
En referencia de nuestro indicador Tiempo medio por caso de prueba, cómo se ve en la tabla 5, tenemos una muestra N igual a 30 ejecuciones de prueba para el pretest y postest. Cómo se detalla en la tabla, tiene una media del pretest es de 14.85 minutos y para la muestra del postest fue de 3.76 minutos, evidenciándose

una reducción del tiempo después de la implementación de la automatización de pruebas de 11.09 minutos.

Adicionalmente, se encontró cambios en los rangos mínimos en las pruebas de pretest y postes, antes le tomaba cómo mínimo 12.00 minutos y después de la implementación logró cómo mínimo 1.94 minutos. De la misma forma para sus rangos máximos, en el pretest de 17.00 minutos y para el postest un máximo de 5.29 minutos.

Figura 1

Comparación de valores medios del indicador tiempo medio por caso de prueba



De igual modo en la figura 1, se aprecia gráficamente los rangos medios del tiempo invertidos por caso de prueba para el pretest y postest. Siendo para el pretest de un 14.85 minutos y para el postest de 3.76 minutos. La diferencia entre el pretest y postes son de 11.09 minutos, representando una reducción significativa del tiempo. equivalente al 74.68%.

Análisis descriptivos del indicador: Número de casos de prueba ejecutados por hora

Tabla 6

Medidas descriptivas del número de caso de prueba ejecutados por hora

| | | N | Mínimo | Máximo | Media | Desv. Desviación |
|---------|---|----|--------|--------|-------|---------------------|
| Pretest | Número de casos de prueba ejecutados por hora | 30 | 3.50 | 5.00 | 4.47 | 0.36292 |
| Postest | Número de casos de prueba ejecutados por hora | 30 | 13.00 | 15.50 | 14.31 | 0.67178 |

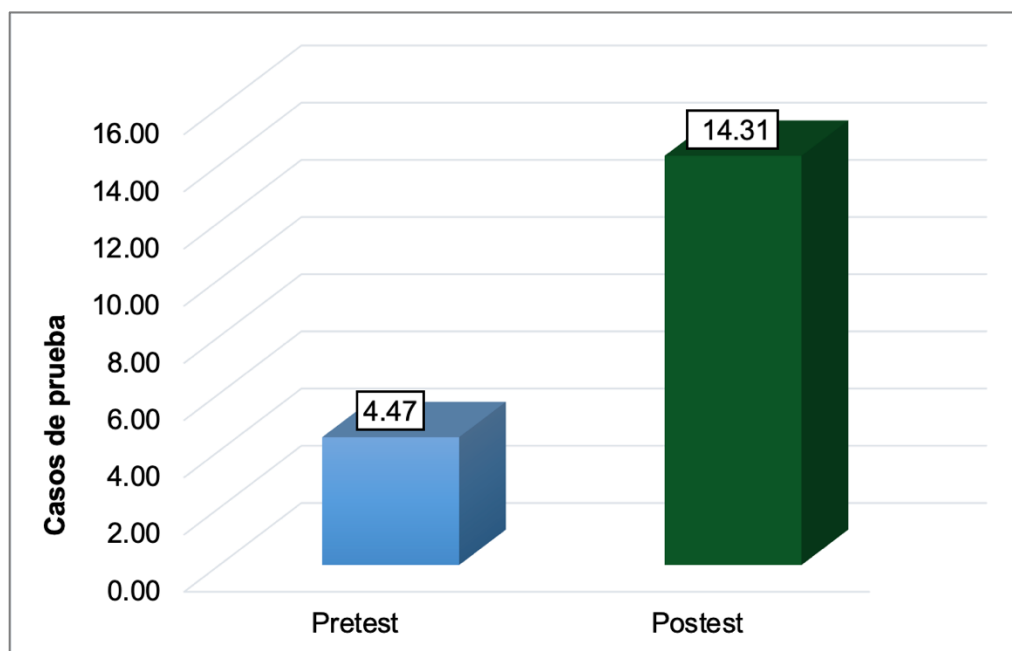
Nota: Datos procesados en el software IBM SPSS Statics 25.

En la tabla 6 se visualiza los valores descriptivos del indicador número de casos de prueba ejecutadas por hora, cómo se detalla en la tabla, en base a una muestra de N igual a 30, siendo ellas representadas por las ejecuciones de prueba. Los resultados evidencian su comportamiento en las pruebas de pretest y postest. El resultado de sus rangos medios, para el pretest fue de 4.47 y en el postest obtuvo 14.31 casos que fueron ejecutados por hora. Evidenciando un incremento de 220%, permitiendo ejecutar más casos de prueba después de la implementación de la automatización de pruebas.

Adicionalmente, se encontró cambios en los rangos mínimos en las pruebas de pretest y postest, en el pretest cómo mínimo se lograba ejecutar 3.50 casos por hora y el postest logró ejecutar 13 casos de prueba por hora. De la misma forma se encontró las diferencias para sus rangos máximos, en el pretest de 5.00 y para el postest un máximo de 15.50 casos de pruebas ejecutados por hora.

Figura 2

Comparación de valores medios del indicador número de caso de prueba ejecutados por hora



De igual modo en la figura 2, se compara los rangos medios del indicador número de casos de prueba ejecutados por hora, evidenciándose que para el pretest se realizaban 4.47 casos de prueba en comparación con los resultados del posttest fue de 14.31 casos de pruebas ejecutados por hora. Concluyendo un incremento significativo para este indicador.

4.2 Prueba de normalidad

La investigación tuvo la oportunidad de elaborar las pruebas de normalidad en sus indicadores Tiempo medio por caso de prueba y el Número de casos de prueba ejecutados por hora. Con el fin de determinar si los indicadores siguen una distribución normal, para ello fueron sometidas a la prueba estadística de Shapiro-Wilk. Siendo una prueba apropiada para probar la normalidad de las 30 fichas de observación y por ser útil con muestras pequeñas $N < 30$ (Bielza y Larrañaga, 2020).

Prueba de normalidad del indicador: Tiempo medio por caso de prueba

Formulación de hipótesis estadística

H₀: Los datos del indicador Tiempo medio por caso de prueba siguen una distribución normal.

H₁: Los datos del indicador Tiempo medio por caso de prueba no siguen una distribución normal.

Tabla 7

Prueba de normalidad del tiempo medio por caso de prueba

| | | Shapiro-Wilk | | |
|---------|---------------------------------|--------------|----|-------|
| | | Estadístico | gl | Sig. |
| Pretest | Tiempo medio por caso de prueba | 0.929 | 30 | 0.046 |
| Postest | Tiempo medio por caso de prueba | 0.909 | 30 | 0.014 |

Nota: Datos procesados en el software IBM SPSS Statics 25.

Los resultados de la prueba de Shapiro-Wilk del indicador Tiempo medio por caso de prueba, demostrada en la tabla 7, se observa que, en el pretest tuvo un valor de significancia de $p = 0.046$. En tal sentido, p es menor a 0.05 permitiendo corroborar que, los datos de la muestra no siguen una distribución normal.

En el caso de los resultados de la prueba de normalidad en el postest, resultó un $p = 0.014$ por lo tanto p es menor 0.05. Nos confirmó que, los datos de las muestras en el postest para el indicador del Tiempo medio por caso de prueba, no tiene una distribución normal. Entonces siendo para ambos resultados inferior a 0.05, se concluye que los datos del indicador del Tiempo medio por caso de prueba no siguen una distribución normal y por tener datos no paramétricos se deberá de plantear la prueba de Wilcoxon para la contrastación de la hipótesis.

Prueba de normalidad del indicador: Número de casos de prueba ejecutados

Formulación de hipótesis estadística

H₀: Los datos del indicador número de casos de prueba ejecutados siguen una distribución normal.

H₁: Los datos del indicador número de casos de prueba ejecutados no siguen una distribución normal.

Tabla 8

Prueba de normalidad del número de casos de prueba ejecutados

| | | Shapiro-Wilk | | |
|----------|----------------------------|--------------|----|-------|
| | | Estadístico | gl | Sig. |
| Pretest | Número de casos de pruebas | 0.951 | 30 | 0.177 |
| Posttest | Número de casos de pruebas | 0.944 | 30 | 0.113 |

Nota: Datos procesados en el software IBM SPSS Statics 25.

Con respecto a los resultados de la prueba de Shapiro-Wilk del segundo indicador, en la tabla 8, se observa que los datos del pretest tuvieron un valor de significancia de $p = 0.177$, sin embargo, $p > 0.05$ corroborando que los datos de la muestra siguen una distribución normal y en el caso del posttest resultó un $p = 0.113$ por lo tanto $p > 0.05$ nos confirma que si tiene una distribución normal. Entonces siendo ambos resultados superiores a 0.05 se concluye que los datos del indicador número de casos de prueba ejecutados siguen una distribución normal y por tener datos paramétricos se planteará la prueba de T Student.

4.3 Análisis inferencial

A pesar de que los resultados de normalidad demostraron que para el indicador tiempo medio por caso de prueba no tiene una distribución normal, se utilizó la prueba de Wilcoxon en cambio el indicador número de casos de prueba se evidenció que, si tiene una distribución normal, entonces se continuó con la prueba de T-Student.

Prueba de hipótesis específica 1: Tiempo medio por caso de prueba

Formulación de hipótesis estadística

H₀: La Automatización de Pruebas no reduce el tiempo medio por caso de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022.

H₁: La Automatización de Pruebas reduce el tiempo medio por caso de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022.

Tabla 9

Resultados de los rangos en la prueba de wilcoxon del indicador 1 del tiempo medio por caso de prueba ejecutados por hora

| | | Rangos | | |
|---|------------------|-----------------|----------------|----------------|
| | | N | Rango promedio | Suma de rangos |
| Postest tiempo medio por caso de prueba – Pretest tiempo medio por caso de prueba | Rangos negativos | 30 ^a | 15.50 | 465.00 |
| | Rangos positivos | 0 ^b | 0.00 | 0.00 |
| | Empates | 0 ^c | | |
| | Total | 30 | | |

a. Postest tiempo medio por caso de prueba < Pretest tiempo medio por caso de prueba

b. Postest tiempo medio por caso de prueba > Pretest tiempo medio por caso de prueba

c. Postest tiempo medio por caso de prueba = Pretest tiempo medio por caso de prueba

Tabla 10

Resultados estadísticos de contraste en la prueba de Wilcoxon del indicador tiempo medio por caso de prueba

| Estadísticos de contraste ^a | Postest tiempo medio por caso de prueba – Pretest tiempo medio por caso de prueba |
|--|---|
| Z | -4.782 ^b |
| Sig. asintótica(bilateral) | 0.000 |

a. Prueba de rangos con signo de Wilcoxon

b. Se basa en rangos positivos.

Nota: Datos procesados en el software IBM SPSS Statics 25.

Contrastación de hipótesis específica 1

De acuerdo con la tabla 10, son los resultados de la prueba de Wilcoxon, permite realizar la contrastación de la hipótesis. Por lo que, el valor del Sig. es de 0.000 y por ser menor a 0.05. Se rechaza la nula y se acepta la alterna con un 95% de confianza.

En las pruebas de rangos de Wilcoxon de tabla 9, las diferencias entre ambas pruebas demostraron para el pretest del tiempo de invertido por caso de prueba fue de 15.50 y postest es de 0.00. Adicionalmente, el valor de $Z=-4.782$ y aplicándole el valor absoluto demuestra que $Z > 1.96$, obteniendo que la hipótesis nula se encontrara en la zona de rechazo. En consecuencia, se logró concluir que la Automatización de Pruebas reduce el tiempo medio por caso de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022.

Prueba de hipótesis específica 2: Número de Casos de Prueba Ejecutados por hora.

Formulación de hipótesis estadística

H₀: La Automatización de Pruebas no incrementa el número de casos de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022.

H₁: La Automatización de Pruebas incrementa el número de casos de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022.

Tabla 11

Prueba de T de Student de muestras emparejadas del indicador número de casos de prueba ejecutados por hora

| | | Diferencias emparejadas | | | | | t | gl | Sig. (bilateral) |
|-----|-----------|-------------------------|---------------------|----------------------------|--|----------|---------|----|---------------------|
| | | Media | Desv. Desviación | Desv. Error promedio | 95% de intervalo de confianza de la diferencia | | | | |
| | | | | | Inferior | Superior | | | |
| Par | Pretest - | -9.84500 | 0.73392 | 0.13399 | -10.119 | -9.57095 | -73.473 | 29 | 0.000 |
| 1 | Postest | | | | | | | | |

Nota: Datos procesados en el software IBM SPSS Statics 25.

Contrastación de hipótesis específica 2

Mediante la elaboración de la prueba de T de Student se logró contrastar la hipótesis, en concordancia con los resultados de la tabla 11, evidenciando el valor del Sig. de 0.000. Siendo el Sig. menor a 0.05 permite confirma el rechazo de la hipótesis nula y se acepta la alterna, con una 95% de confianza. Logrando concluir que, la Automatización de Pruebas incrementa el número de casos de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022.

V. DISCUSIÓN

En los resultados obtenidos en la presente investigación pre experimental, confirman el impacto positivo de la eficiencia de pruebas de software con la implementación de la automatización de pruebas en la consultora de TI, según los resultados de cada indicador de la variable dependiente: tiempo medio por caso de prueba y la cantidad de pruebas ejecutadas por hora, hubo un impacto relevante al comparar los resultados consultadas de otras investigaciones.

Con respecto a los resultados del primer objetivo específico, con el indicador del tiempo medio por caso de prueba, es el tiempo medio que tarda el personal de asegurador de calidad en realizar las actividades necesarias para la ejecución de un caso de prueba siendo su alcance de verificar si cumple con los requisitos definidos del software.

Para el presente indicador se evidenció una reducción del 74.68% después de la implementación de la automatización de prueba, en minutos se redujo 11.09 minutos. Adicionalmente, en el análisis descriptivo se observa la comparación de ambas situaciones, entre las muestras del pretest y posttest; obteniendo una media descriptiva de los datos antes de la implementación un tiempo mínimo de 12.00 minutos a un máximo de 17.00 minutos y después de la implementación un tiempo mínimo de 1.94 minutos a un máximo de 5.29 minutos, confirmando una reducción del tiempo medio por caso de prueba.

Referente al análisis inferencial, fue necesario realizar la prueba de normalidad para validar si los datos de la muestra provienen de una distribución normal, entonces se realizó la prueba de Shapiro Wilk. Los resultados de la prueba de normalidad del indicador tiempo medio por caso de prueba en el pretest es igual a 0.046, así mismo para el posttest fue igual a 0.014, en tal sentido ambos resultados fueron inferiores a 0.05.

Concluyendo, que los datos del primer indicador al no seguir una distribución normal se estableció realizar la prueba no paramétrica de Wilcoxon, para la contrastación de la hipótesis. Conforme a la tabla 9 se tiene los rangos del pretest y postest, después en los resultados de contraste de la tabla 10, encontrando el valor de Z en -4.782^b y para p un valor menor a 0.05 resultando una significancia de 0.000. De este modo, con un 95% de confianza se rechaza la hipótesis nula y se acepta la hipótesis de investigación. Por esta razón, se confirma la aceptación de hipótesis dado que, la automatización de pruebas reduce el tiempo medio por caso de prueba ejecutados durante las pruebas de software en una consultora de TI en Lima.

Los resultados coinciden con las investigaciones de Gutiérrez (2019), logrando mejorar las pruebas de software con respecto a las pruebas funcionales de un sistema web, con la herramienta Badboy. Debido a que es una herramienta práctica, intuitiva para el personal que genere los casos de pruebas automatizados y otro de sus beneficios por ser compatible con sistemas web. Le permitió lograr en reducir el 90 % en el tiempo de ejecutar y re ejecutar pruebas.

Del mismo modo, se concuerda con esta investigación, al tener una herramienta clara y concisa, permite a que cada nuevo integrante en el área de pruebas se acople al equipo en menos tiempo y es por ello por lo que se creó los escenarios de casos de pruebas en lenguaje Gherkin basada en BDD, en consecuencia, le es accesible identificar y estimar los tiempos de ejecución de pruebas.

Referente a la investigación de Fernández (2018), tuvo la oportunidad de implementar en el Banco de Crédito una herramienta de interfaz amigable para automatizar parte de las actividades de pruebas, siendo dos actividades beneficiadas, en colaboración con su equipo lograron generar scripts de pruebas, obteniendo la reducción del 56,78% con respecto al tiempo. Adicional no sólo mejoro el tiempo de respuesta, sino también la productividad laboral para la generación de data, permitiendo a los integrantes del equipo de calidad enfocarse a las pruebas más complejas y mejorar la cobertura de sus data de prueba. A pesar

de que implementó una herramienta de automatización de procesos, se es factible que toda herramienta de automatización logra conseguir sus resultados. De todas maneras, para esta investigación se consideró la implementación de scripts de pruebas para realizar un framework integrando herramientas y librerías, permitiendo tener la independencia del dominio del producto de automatización

Por otro lado, Cortés (2020), tuvo que estandarizar antes 4 subprocesos críticos, porque no seguían una metodología y no se salvaguardaba la información de todos los hechos históricos de las pruebas, posterior a ello implementó un framework de automatización basado en Selenium. La cual, le permitió reducir el tiempo de ejecución de 60% a 37,42%, esta investigación previa nos permitió considerar la herramienta de Selenium, por la certeza de sus beneficios con el tiempo de ejecución.

En tanto, Medina (2020), una de sus estrategias para elaborar la herramienta de automatización, siguió las buenas prácticas del desarrollo dirigido por comportamiento, cómo también el patrón de diseño de Page Object Model con las librería de Selenium, CodeceptJS y Cucumber, logrando alcanzar una reducción del 50% en el tiempo de ejecución de pruebas, describiendo qué antes de la automatización invertía en promedio 22 minutos para ejecutar todos los casos de pruebas pertenecientes a las funcionalidades principales del aplicativo móvil App Crecer y después de la implementación le tomó 11 minutos.

Los resultados de las investigaciones de Chávez (2021), Cortés (2020), Medina (2020) y Gutiérrez (2019), lograron reducir más del 50% en el tiempo dedicado a las pruebas, siendo una importante optimización del tiempo y esfuerzo en las subtarear de la ejecución de pruebas. Logrando un impacto en el ahorro del tiempo, para el beneficio del proceso y a las personas. Confirmando que, las herramientas de automatización reproducen actividades repetitivas de validación mediante scripts de prueba con poca intervención del hombre y evita errores humanos en el proceso.

Para el segundo objetivo específico en relación con el indicador número de casos de prueba, los resultados evidenciaron en el pretest alcanzó 4.47 casos de pruebas ejecutados en una hora y después de la implementación de la automatización de pruebas incremento a 14.31 casos de pruebas ejecutados en una hora. Logrando evidenciar que la implementación de automatización de pruebas logra incrementar un 220%, permitiendo mejorar la cobertura de la prueba por la mayor cantidad de valores de entrada probados.

En el análisis descriptivo se obtuvieron los resultados medios y máximos del pretest y postest; para media descriptiva de los datos obtenidos antes de la implementación mínima de 3.50 a un máximo de 5.00 casos de prueba ejecutados y después de la implementación de la automatización de pruebas se obtuvo un mínimo de 13 a un máximo de 15.50 casos de pruebas ejecutados, confirmando un incremento de los casos de prueba ejecutados.

En el análisis inferencial, la prueba de normalidad del indicador número de casos de prueba ejecutados demostraron en el pretest fue de 0.177 y postest con valor de 0.113, ambos resultados al ser mayores a 0.05, se concluyó que la muestra si siguen una distribución normal.

Finalmente se aplicó la prueba paramétrica de T-Student para la contrastación de la hipótesis, conforme a la tabla 11 de las muestras emparejadas del indicador número de casos de prueba ejecutados, logró un resultado de significancia de 0.00, siendo menor a 0.05.

De este modo, con un 95% de confianza se rechaza la hipótesis nula y se acepta la hipótesis alterna. Por esta razón, la automatización de pruebas incrementa el número de casos de prueba ejecutados durante las pruebas de software en una consultora de TI en Lima.

Los resultados coinciden con las investigaciones antecesoras, como Huamán (2021), después de implementar su modelo de automatización de pruebas logró mejorar su eficiencia el 94.38% por incrementar a 11 438 casos de prueba durante 8 meses, permitiendo mejorar la cobertura de prueba del producto y creando un modelo de automatización con diferentes herramientas cómo Serenety BDD, dos patrones de diseño ScreenPlay y POM, para soportar las pruebas de los aplicativos web y web services. Por su parte Irshad (2021), mediante su sistema de reutilización para la automatización de pruebas de aceptación logró generar más scripts de pruebas permitiendo ejecutar más casos de prueba. Donde 72 casos de prueba en BDD cubrió más de 80 casos de uso.

Medina (2020), alcanzó a escribir 124 casos automatizados para las pruebas de regresión permitiendo reducir el 60% de las pruebas manuales con el objetivo de que los analistas se enfoquen en pruebas más complejas y explorar nuevos escenarios del producto. En tanto Fernández (2018), también se benefició con la automatización por la reutilización de los casos automatizados para incrementar la productividad para la creación de data, favoreciendo a su equipo y a otros canales de servicio en el Banco de Crédito para la continuidad de sus pruebas Antes de la implementación lograba ejecutar 91 casos de prueba para la creación de data, posterior a la adaptación de la automatización por RPA logró obtener 99 creaciones de data, sufriendo un incremento del 7.71%, permitiendo una mejora significativa a su dimensión de productividad de su variable dependiente.

La metodología de investigación utilizada permitió orientar la resolución de un problema basándose en conocimientos previos correspondiente a una investigación aplicada de diseño experimental y tipo pre experimental, con el fin único de controlar un único grupo de experimento. De tal manera, se obtuvieron las mediciones de los indicadores en las pruebas de pretest y posttest, llegando a demostrar el impacto positivo en la eficiencia de las pruebas de software en una consultora de TI con el beneficio de la automatización de pruebas. Del mismo modo, implicó una exhaustiva búsqueda de teorías para profundizar el entendimiento de las variables de investigación.

Por otro lado, sobre el instrumento de recolección de datos se tuvo algunos inconvenientes para recolectar 30 observaciones de ejecuciones de prueba para cada indicador en su respectiva ficha de observación. Debido a que todos los aseguradores de calidad trabajan de forma remota, se tuvo que realizar el acompañamiento por videollamada y observar las actividades desde la opción de compartir pantalla para visualizar la ejecución de pruebas funcionales dentro del aplicativo web y el uso de la herramienta de automatización. Para un mejor control y exactitud de los datos recolectados, para las pruebas de pretest y postest. Posteriormente, la información de las fichas se registró en el software estadístico de SPSS, para la obtención de los análisis estadísticos correspondientes.

El proceso de investigación logra ser replicable para futuras investigaciones para toda entidad correspondiente en consultoría de TI y responsables de las pruebas de software. Por la naturaleza de la investigación, se llevaron actividades secuenciales importantes para lograr sus objetivos, desde la participación del equipo de juicio de expertos en evaluar cada indicador de la dimensión uso de recursos y productividad laboral. Posteriormente el instrumento permitió recolectar los datos en fichas de observación. Los resultados de la estadística descriptiva e inferencial lograron confirmar que la automatización sí impacta positivamente a la eficiencia de las pruebas de software en una consultora de TI, como también de forma específica : (a) la automatización redujo en el tiempo medio por caso de prueba durante las pruebas de software en una consultora de TI, Lima 2022 y (b) la automatización logró incrementar el número de casos de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022.

VI. CONCLUSIONES

Primero : La automatización de pruebas impacta positivamente en la eficiencia de las pruebas de software en una consultora de TI, Lima 2022, por lograr un mejor impacto a sus dos indicadores: tiempo medio por caso de prueba y el número de casos de prueba ejecutados. Conforme a los resultados de la contrastación de las hipótesis. Para la herramienta de automatización se creó un framework, siguiendo la metodología de Scrum y basado en el diseño de patrones de Page Object Model, Selenium y BDD, permitiendo mejorar su confiabilidad, redujo el mantenimiento del código, permitió la reutilización del código, clase, package y page factory como inventario de los locators de la página web.

Segundo : Para el primer indicador tiempo medio por caso de prueba para las pruebas de software en una consultora de TI, se concluye que sin la automatización de prueba tomaba 14.85 minutos y después de la implementación de la automatización de prueba disminuye a 3.76 minutos. Por lo que se refiere la existencia de la reducción del 74.68% del tiempo medio por caso de prueba con un valor de Z de -4.782 y p de 0.00, por tanto, se rechazó la hipótesis nula y se aceptó la hipótesis alternativa. En efecto la automatización de pruebas reduce el tiempo medio por caso de prueba ejecutados durante las pruebas de software en una consultora de TI.

Tercero : Para el segundo indicador número de casos de prueba ejecutados para las pruebas de software en una consultora de TI, se evidencio antes de la implementación de la automatización de pruebas una ejecución de 4.47 casos de prueba y después de la implementación 14.51 casos de pruebas ejecutadas en una hora. Por lo que se refiere el incremento del 220% del número de casos de prueba ejecutados, con valores estadísticos Sig. en 0.000 con un 95% de confianza, por tanto, se rechazó la hipótesis nula y se aceptó la hipótesis alterna. Se concluye que la automatización de pruebas incrementa el número de casos de pruebas ejecutados durante las pruebas de software en una consultora.

VII. RECOMENDACIONES

- Primero : Luego del impacto positivo de la automatización de prueba con los dos indicadores en base a la eficiencia de las pruebas de software, se recomienda a líder de calidad de la consulta en agregar una tarea en cada sprint para el mantenimiento y análisis de nuevos scripts de pruebas para las nuevas funcionalidades e incorporarlas para futuras pruebas de regresión para que la herramienta este actualizada con los nuevos requisitos y evitar deudas de parte de los aseguradores de calidad. Adicionalmente, se recomienda a la consultora y a futuras investigadores evaluar la implementación de machine learning e inteligencia artificial para generar de forma automática casos de prueba y los reportes generados permitan predecir la causa de los errores en un lenguaje técnico para una comunicación fluida con el equipo de desarrollo.
- Segundo : Con respecto al primer indicador tiempo medio por caso de prueba el cual obtuvo una disminución del 74.68%, se recomienda al líder de calidad de la consultora, realizar un análisis para implementar en la automatización de prueba la integración continua para que permita ejecutar los casos en un servidor remoto, debido a que las pruebas se ejecutan en el equipo local del personal en consecuencia disminuye la optimización del equipo del personal, dificultando a realizar otras actividades
- Tercero : Se recomienda a cada personal que integran el área de calidad de la consultora, cuando agregue nuevos casos de prueba en la herramienta, éstas deben estar escritas forma clara y concisa, para que los nuevos talentos de la consultora no tengan complicaciones al interpretar las validaciones de cada caso.

REFERENCIAS

- Agurto, J. (2019). *Automatización de Pruebas de Software en Telco con UFT*. [Tesis de pregrado, Universidad Tecnológica del Perú, Facultad de Ingeniería] Repositorio Institucional. Obtenido de <https://repositorio.utp.edu.pe/handle/20.500.12867/2062>
- Alan, D. y Cortez. (2018) Procesos y Fundamentos de la investigación Científica. ISBN: 978-9942-24-093-4. Ecuador: Editorial UTMACH. Obtenido de <http://repositorio.utmachala.edu.ec/bitstream/48000/12498/1/Procesos-y-FundamentosDeLainvestiagcionCientifica.pdf>
- Arjun R., y Shahid, A (2019). Automation Testing for Order to Cash Process in Microsoft Dynamics 365. *Journal of Software*, 548–558.
- Arnold, M., y Osorio, F. (1998). Introducción a los Conceptos Básicos de la Teoría General de Sistemas. *Cinta de Moebio*, (3), [fecha de Consulta 7 de Junio de 2022]. ISSN: 0717-554X. Obtenido en: <https://www.redalyc.org/articulo.oa?id=10100306>
- Bahaweres, R., Oktaviani, E., Wardhani, L., Hermadi, I., Suroso, A., Solihin, I., y Arkeman, Y. (2020, November). Behavior-driven development (BDD) Cucumber Katalon for Automation GUI testing case CURA and Swag Labs. In *2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)* (pp. 87-92). IEEE.
- Balderas, B. (2018). *Estrategias para las pruebas de software que permiten mejorar la calidad en el desarrollo de proyectos de software*. [Tesis de Pregrado] .Repositorio institucional. Obtenido de <https://hdl.handle.net/20.500.12371/15073>
- Barus, A. (2019, March). The implementation of ATDD and BDD from Testing Perspectives. In *Journal of Physics: Conference Series* (Vol. 1175, No. 1, p. 012112). IOP Publishing. Obtenido de [10.1088/1742-6596/1175/1/012112](https://doi.org/10.1088/1742-6596/1175/1/012112)

- Battina, D. (2019). Artificial Intelligence in Software Test Automation: A Systematic Literature Review. *International Journal of Emerging Technologies and Innovative Research (www.jetir.org| UGC and issn Approved)*, ISSN, 2349-5162.
- BBVA (2019). *El gran dilema del QA: ¿Automatización o testing manual?* Obtenido de <https://www.bbvanexttechnologies.com/blogs/el-gran-dilema-del-qa-automatizacion-o-testing-manual/>
- BBVA (2022). *Los nuevos procesos de desarrollo de 'software' garantizan la calidad de servicios de BBVA durante la crisis sanitaria.* Obtenido de: <https://www.bbva.com/es/los-nuevos-procesos-de-desarrollo-de-software-garantizan-la-calidad-de-servicio-de-bbva-durante-la-crisis-sanitaria/amp/>
- Bertalanffy, L. (2000). Teoría general de los sistemas. Editorial Fondo Cultural Económica, 2º ed. reimpresión, Bogotá, Colombia.
- Bielza, C., y Larrañaga, P. (2020). *Data-driven computational neuroscience: machine learning and statistical models.* Cambridge University Press.
- Blokehead, T. (2016). Scrum - ¡Guía definitiva de prácticas ágiles esenciales de Scrum!
- Brunet, I., Icart, I. B., & Belzunegui, A. (2003). *Flexibilidad y formación: una crítica sociológica al discurso de las competencias* (Vol. 28). Icaria editorial.
- Caiza, G. (2019). *Implementación de métricas para la evaluación del proceso de control de calidad en proyectos de desarrollo de software para la empresa Logiciel.* [Tesis de Maestría, Escuela Politécnica Nacional] en Ecuador. Repositorio Institucional. Obtenido de: <http://bibdigital.epn.edu.ec/handle/15000/20009>
- Campos, G. y Lule, N. (2012). La observación, un método para el estudio de la realidad. *Xihmaj*, ISSN-e 1870-6703, Vol. 7, Nº. 13, 2012, págs. 45-60. Obtenido de <https://dialnet.unirioja.es/servlet/articulo?codigo=3979972>

- Cequea, M., (2012). Modelo multifactorial para optimización de la productividad en el proceso de generación de energía eléctrica: aplicación al caso de las centrales hidroeléctricas venezolanas. *Industriales*.
- Chacón, J. y Rugel, S. (2018). Teorías, modelos y sistemas de gestión de calidad. *Revista Espacios*, 39(50), 14-22.
- Chávez, P. (2021). *Automatización de casos de pruebas usando Selenium y TestNG Framework para mejorar el proceso de pruebas de un sistema que brinda servicios RPO, 2020*. [Tesis de pregrado, Universidad Nacional Mayor de San Marcos] Repositorio institucional. Obtenido de <https://hdl.handle.net/20.500.12672/17023>
- Chopra, R. (2018). *Software quality assurance*. Mercury Learning & Information. ProQuest Ebook Central: <https://www.proquest.com/legacydocview/EBC/5399255?accountid=37408>
- Corral, Y (2009). Validez y confiabilidad de los instrumentos de investigación para la recolección de datos. *Revista Ciencia de la Educación*. Vol. 19. N.º 33.
- Cortés, Á. (2020). Automatización de pruebas de regresión para reducción de tiempo de entrega de nuevas versiones de software. [Tesis Maestría - Universidad de Chile] <https://repositorio.uchile.cl/handle/2250/177640>
- de Asís, F. (2007). Análisis de eficiencia de los Departamentos universitarios. El caso de la Universidad de Sevilla (Vol. 68). Librería-Editorial Dykinson.
- Deming, W. E. (1989). *Quality, productivity and competitiveness: the way out of the crisis. Proc. of Principles for Transforming Management in Western Companies, 1st ed. Madrid, Spain: Editions Diaz de Santos SA, 62-68*.
- Domínguez, V. y López, A. (2016). Teoría General de Sistemas, un enfoque práctico. *TECNOCENCIA Chihuahua*, 10(3), 125-132. Recuperado a partir de <https://148.229.0.27/index.php/tecnociencia/article/view/174>
- Esteban, N. (2018). Tipos de Investigación. Recuperado en: https://alicia.concytec.gob.pe/vufind/Record/UNIS_5b55a9811d9ab27b8e45c193546b0187/Details

- Faragó, D., Friske, M., y Sokenou, D. (2020). Towards a Taxonomy for Applying Behavior-Driven Development (BDD). *Softwaretechnik-Trends*, 40(3), 3-7.
- Febles, D., Trujillo, Y. y Mendosa, A. (2021). Oportunidades de mejora al proceso de aseguramiento de calidad del proceso y el producto. *Revista Cubana de Ciencias Informáticas*, vol. 16, núm. 1, pp. 46-61, 2022. <https://www.redalyc.org/journal/3783/378370413004/>
- Fernández, J. (2018). *Automatización de procesos para mejorar las Pruebas de Software en el área de calidad del Banco de Crédito*. [Tesis de Maestría, Universidad César Vallejo]. Perú. Repositorio institucional. Obtenido de: <https://hdl.handle.net/20.500.12692/23871>
- Galvan, S., Mora, M., Laporte, C. y Duran, H. (2021). Reconciliation of scrum and the project management process of the ISO/IEC 29110 standard-Entry profile—an experimental evaluation through usability measures. *Software Quality Journal*, 29(2), 239–273. <https://doi.org/10.1007/s11219-021-09552-3>
- Garcia, B. (2022). *Hands-On Selenium WebDriver with Java : A Deep Dive into the Development of End-to-End Tests*. O'Reilly.
- García, J., Cazallo, A., Barragan, C., Mercado, M., Olarte, L., y Meza, V. (2019). Indicadores de Eficacia y Eficiencia en la gestión de procura de materiales en empresas del sector construcción del Departamento del Atlántico, Colombia. *Revista Espacios*, 40(22).
- Gismano, Yanina(2012). Dirección Postal: Las Heras 935, Bahía Blanca (8000). (s/f). 10.30.35. Recuperado el 4 de agosto de 2022, de <http://163.10.30.35/congresos/jdsunlp/vii-jornadas-2012/actas/Gismano.pdf>
- Gómez, G. G. (2020). Teoría general de sistemas. Colombia: Ediciones USTA. ISBN: 978-958-631-850-1 Recuperado en: <http://hdl.handle.net/11634/23242>

- González-Filgueira, G. y Permy, F. J. R. (2018). Automatización de una planta industrial de alimentación mediante control distribuido. *Revista Ibérica de Sistemas e Tecnologías de Informação*, (27), 1-17.
- Gupta, V. K. (1983). Labor productivity, establishment size, and scale economies. *Southern Economic Journal*, 853-859.
- Guisado, M., Vila, M., y Guisado-Tatto, M. (2016). Innovación, capacidad productiva, formación en el puesto de trabajo y productividad.
- Gutiérrez, L. (2019). Implementación de la herramienta Badboy para la mejora de pruebas funcionales en aplicaciones web. Universidad Peruana de Ciencias e Informática. Facultad de Ciencias e Ingeniería. Recuperado en: http://repositorio.upci.edu.pe/bitstream/handle/upci/96/T-GUTIERREZ_ZAPATA_KAROLY.pdf?sequence=1&isAllowed=y
- Hernández, R. Baptista, P. Fernández, C. (2014). Metodología de la investigación. (5ta ed.). México D.F., México: Mc Graw – Hill
- Hernández, R. y Mendoza, C. (2018). Metodología de la investigación. Las rutas cuantitativa, cualitativa y mixta. Ed. Mc Graw Hill Education. ISBN: 978-1-4562-6096-5.
- Huamán, E. (2021). Modelo de automatización de pruebas para optimizar la gestión de la calidad de software en la empresa CIFIN. Universidad Tecnológica del Perú, Facultad de Ingeniería. Recuperado en: <https://hdl.handle.net/20.500.12867/4708>
- Huertas, Jesús (2014) Teoría de la eficiencia dinámica, teoría de investigación doi: www://dialnet-LaTeoriaDeLaEficienciaDinamica-1292130.pdf
- Hurtado, F. A. (2005). Gestión y auditoría de la calidad para organizaciones públicas: normas NTCGP 1000: 2004 conforme a la ley 872 de 2003. Universidad de Antioquia.
- Irshad, M., & Petersen, K. (2021). A systematic reuse process for automated acceptance tests: Construction and elementary evaluation. *E-Informatica*

Software Engineering Journal, 15(1), 133–162. <https://doi.org/10.37190/e-Inf210107>

Islam, M. y Quadri, S. (2020). Framework for Automation of Software Testing for Web Application in Cloud-(FASTEST) Web Application. *Solid State Technology*, 63(5), 131-150.

ISO/IEC/IEEE (2013). ISO/IEC/IEEE 29119-2:2013 Software and systems engineering - Software testing – Part 2: Test processes, Ginebra, Suiza: ISO.

Judge, W., Fryxell, G., y Dooley, R. (1997). The New Task of R&D Management: Creating Goal-Directed Communities for Innovation. *California Management Review*, 39(3), 72–85. <https://doi.org/10.2307/41165899>

Kenett, R. S., Ruggeri, F., & Faltin, F. W. (Eds.). (2018). Analytic methods in systems and software testing. John Wiley & Sons.

Kuhrman, M., Diebold, P., Münch, J., Tell, P., Trektore, K. y McCaffery, F., Garousi, V., Linssens, O., Hanser, E. y Prause, C. (2018) Hybrid software development approaches in practice: a European perspective, *IEEE Software*, vol. 36, N.º 4, pp. 20-31. Recuperado en: 10.1109/MS.2018.110161245

Leaños, A. (2018). Mejora en la curva de tiempo de ejecución de proyectos de software con la integración de herramientas para la generación automática de casos de prueba en DUALBIZ.S.R.L. https://www.soe.uagrm.edu.bo/wp-content/uploads/2018/08/Tesis-de-Maestria_AlcidesYohacinLea%C3%B1osRodriguez.pdf

López, J. (2020, abril 14). Taylorismo. Economipedia. <https://economipedia.com/definiciones/taylorismo.html>

Mahee, K, Hameed, K y Asif, A. (2019). Software Test Automation.

Majid, U. (2018). Research fundamentals: Study design, population, and sample size. *Undergraduate research in natural and clinical science and technology journal*, 2, 1-7.

- Marín, A., Trujillo, Y., Buedo, D. (2020). Estrategia de pruebas para organizaciones desarrolladoras de software 2020. ISSN: 2227-1899. Recuperado en: <http://scielo.sld.cu/pdf/rcci/v14n3/2227-1899-rcci-14-03-83.pdf>
- Mateus, M. (2016). *Asistentes y secretarias ¿Profesión de riesgo?*. FC EDITORIAL.
- Medina, J. y Guadayol, C. (2010). La automatización en la industria química (Vol.1). Barcelona, España: Universidad Politécnica de Catalunya ISBN: 978-84-9880. Recuperado en: <http://hdl.handle.net/2099.3/36842>
- Medina, M. (2020). *Automatización de pruebas para proyectos ágiles aplicando el desarrollo dirigido por comportamiento para una compañía de líneas de belleza*. Universidad Tecnológica del Perú, Facultad de Ingeniería. Recuperado en: <https://hdl.handle.net/20.500.12867/3166>
- Mera, J. (2016). Software quality testing process analysis. *Ingeniería Solidaria*, 12(20), 163–176. <https://doi.org/10.16925/in.v12i20.1482>
- Mobaraya, F., & Ali, S. (2019). Technical Analysis of Selenium and Cypress as Functional Automation Framework for Modern Web Application Testing. *Department of Information Technology, AGI Institute, Auckland, New Zealand*. <https://csitcp.com/paper/9/918csit03.pdf>
- Morote, L. et al. (2018). Teorías sistémicas y paradigma de investigación performativa en los estudios superiores de danza. *El artista: revista de investigaciones en música y artes plásticas*, (15), 7. Recuperado en: <https://dialnet.unirioja.es/descarga/articulo/7276872.pdf>
- Muñoz, M., y Cordero, N. (2017). La creación colectiva teatral. Método de acción social y resistencia con el colectivo de personas sin hogar en Sevilla, España. *Estudios Políticos*, (50), 42–61. <https://doi.org/10.17533/udea.espo.n50a03>
- Musliu, A., y Jashari, X. (2021). Software Automated Testing using BDD Approach with Cucumber Framework. *UBT International Conference*.

- Ñaupas, H., Valdivia, M., Palacios, J. y Romero, H. (2018). Metodología de la investigación cuantitativa-cualitativa y redacción de la tesis. Ed. Ediciones de la U. ISBN 978-958-762-876-0.
- Ossa, C. (2016). Teoría general de sistemas: conceptos y aplicaciones. Universidad Tecnológica de Pereira. Recuperado en: <https://doi.org/10.22517/9789587222289>
- Osherove, R. (2014). The art of Unit Testing with examples in C#. Editorial Manning Publications Co.
- Otzen, T. y Manterola, C. (2017). Técnicas de muestreo sobre una población a estudio. *International Journal of Morphology*, 35(1), 227-232. Recuperado en: <https://dx.doi.org/10.4067/S0717-95022017000100037>
- Patino, C., y Ferreira, J. (2018). Inclusion and exclusion criteria in research studies: definitions and why they matter. *Jornal Brasileiro de Pneumologia*, 44, 84-84.
- Pressman, R. (1998). Ingeniería del software: Un enfoque Práctico. Editorial McGraw-Hill, cuarta edición.
- Sommerville, I. (2011). Ingeniería de software (9ª. Ed.). México: Pearson Educación. ISBN: 978-607-32-0603-7.
- Shewhart, W. A. (1931). Control económico de la calidad en manufactura. New York.
- Srivastava, N., Kumar, U., y Singh, P. (2021). Software and performance testing tools. *Journal of Informatics Electrical and Electronics Engineering*, 2(01), 1-12.
- Steinberg, W. y Price, M. (2020). *Statistics alive!* Sage Publications.
- Tamayo, M. (2004). El proceso de la investigación científica. México: Editorial Limusa.
- Tavares, B., da Silva, C. y de Souza, A. (2019). Risk management analysis in Scrum software projects. *International Transactions in Operational Research*, 26(5), 1884–1905. <https://doi.org/10.1111/itor.12401>

ANEXOS

Anexo 1. Matriz de consistencia

| TÍTULO: Impacto de la automatización de pruebas en la eficiencia de pruebas de software en una consultora de TI, Lima 2022 | | | | | |
|---|---|---|--|--------------------------------------|------------------|
| AUTOR: Morales Yovera Wendy | | | | | |
| PROBLEMA | OBJETIVOS | HIPÓTESIS | VARIABLES E INDICADORES | | |
| <p>Problema Principal:</p> <p>¿Cuál es el impacto de la Automatización de Prueba en la Eficiencia de las pruebas de software en una consultora de TI, Lima 2022?</p> <p>Problemas Específicos:</p> <p>PE1: ¿Cuál es el impacto de la Automatización de Pruebas en el tiempo medio por caso de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022?</p> <p>PE2: ¿Cuál es el impacto la Automatización de Pruebas en el número de casos de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022?</p> | <p>Objetivo Principal:</p> <p>Determinar el impacto de la Automatización de Pruebas en la Eficiencia de las pruebas de software en una consultora de TI, Lima 2022.</p> <p>Objetivos específicos:</p> <p>OE1: Determinar el impacto de la Automatización de Pruebas en el tiempo medio por caso de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022.</p> <p>OE2: Determinar el impacto de la Automatización de Pruebas en el número de casos de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022.</p> | <p>Hipótesis Principal:</p> <p>La Automatización de Pruebas impacta positivamente en la Eficiencia de las pruebas de software en una consultora de TI, Lima 2022.</p> <p>Hipótesis específicas:</p> <p>HE1: La Automatización de Pruebas reduce el tiempo medio por caso de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022.</p> <p>HE2: La Automatización de Pruebas incrementa el número de casos de prueba ejecutados durante las pruebas de software en una consultora de TI, Lima 2022.</p> | Variable Independiente: Automatización de Pruebas | | |
| | | | Variable Dependiente: Eficiencia | | |
| | | | Dimensiones | Indicadores | Unidad de Medida |
| | | | Uso de Recursos | Tiempo medio por caso de prueba | Tiempo |
| | | | Productividad laboral | Número de Casos de Prueba Ejecutados | Unidad |

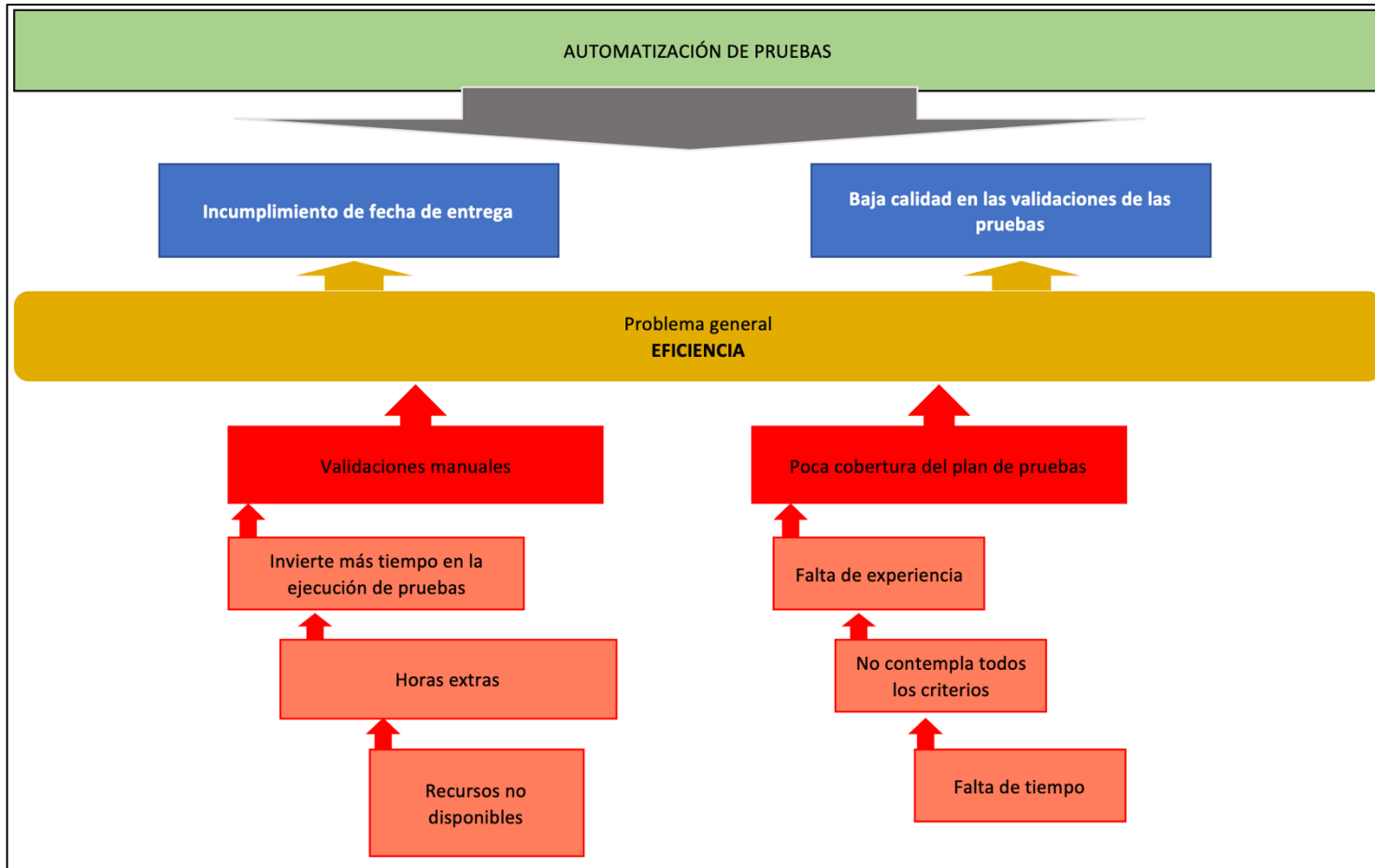
Metodología

| TIPO Y DISEÑO | POBLACIÓN Y MUESTRA | INSTRUMENTO | ESTADÍSTICA |
|---|---|---|--|
| <p><u>Tipo:</u> Aplicada</p> <p><u>Enfoque:</u> Investigación cuantitativa</p> <p><u>Diseño:</u> Pre - Experimental</p> | <p><u>Población:</u> 32 observaciones</p> <p><u>Muestra:</u> 30 observaciones</p> <p><u>Muestreo:</u> No Probabilístico</p> | <p>Técnicas: Observación</p> <p>Instrumento: Ficha de observación</p> | <p>Descriptiva</p> <p>Para realizar el análisis descriptivo, se empleará tablas de contingencia para un análisis bidimensional e histogramas que permitan describir la información respectiva a la muestra.</p> <p>Inferencial</p> <p>Para realizar el análisis inferencial se define la prueba de Shapiro-Wilk para comprobar normalidad por cada indicador si se ajusta o no una distribución normal, se empleará métodos no paramétricos Wilcoxon, caso contrario se aplicará T-Student para la contratación de la hipótesis.</p> |

Anexo 2. Matriz de operacionalización de variables

| TÍTULO: Impacto de la automatización de pruebas en la eficiencia de pruebas de software en una consultora de TI, Lima 2022 | | | | | |
|---|-----------------------|--------------------------------------|----------------------|------------------|--|
| AUTOR: Morales Yovera Wendy | | | | | |
| Variables | Dimensión | Indicador | Instrumento | Unidad de Medida | Fórmula |
| Variable: Eficiencia Según Cequea (2012), es el criterio económico que revela la capacidad administrativa de producir el máximo resultado con el mínimo de recursos, energía y tiempo, por lo que es la óptima utilización de los recursos disponibles para la obtención de resultados deseados. | Uso de recursos | Tiempo medio por caso de prueba | Ficha de observación | Tiempo | $X = \frac{\text{Tiempo total en ejecutar los casos de prueba}}{\text{Total de casos de prueba ejecutados}}$ |
| | Productividad laboral | Número de casos de prueba ejecutados | Ficha de observación | Unidad | $X = \frac{\text{Número de casos de prueba ejecutados}}{\text{Horas-hombre laboradas}}$ |

Anexo 3. Árbol de problemas



Anexo 4. Muestra

| Tamaño muestral para una proporción con marco muestral conocido | | | |
|---|------------------|--------------|----------------------------------|
| $n = \frac{N * Z_{1-\alpha/2}^2 * p * q}{d^2 * (N - 1) + Z_{1-\alpha/2}^2 * p * q}$ | | | |
| Marco muestral | N | 32 | |
| Error Alfa | α | 0.050 | Se acostumbra: 5% |
| Nivel de Confianza | $1-\alpha$ | 0.975 | |
| Z de (1- α) | Z (1- α) | 1.960 | |
| Prevalencia de la Enf. / Prob. | p | 0.500 | Cuando no se tiene colocar: 0.5 |
| Complemento de p | q | 0.500 | |
| Precisión | d | 0.050 | Cuando no se tiene colocar: 0.05 |
| | | | |
| Tamaño de la muestra | n | 29.61 | |

Anexo 5. Instrumento de recolección de datos

Guía de observación N.º 1. Indicador tiempo medio por caso de prueba

| Ficha de observación: Indicador Tiempo medio por caso de prueba / Pretest | | | | |
|---|-------|--|------------------------------------|---|
| Investigador | | Wendy Ly Morales Yovera | | |
| Proceso observado | | Pruebas de software | | |
| Pretest | | | | |
| N.º de Obs. | Fecha | Tiempo total en ejecutar los casos de prueba | Total de casos de prueba ejecutado | Tiempo medio por caso de prueba=Tiempo total en ejecutar los casos de prueba / Total de casos de prueba ejecutado |
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | | | | |
| 10 | | | | |
| 11 | | | | |
| 12 | | | | |
| 13 | | | | |
| 14 | | | | |
| 15 | | | | |
| 16 | | | | |
| 17 | | | | |
| 18 | | | | |
| 19 | | | | |
| 20 | | | | |
| 21 | | | | |
| 22 | | | | |
| 23 | | | | |
| 24 | | | | |
| 25 | | | | |
| 26 | | | | |
| 27 | | | | |
| 28 | | | | |
| 29 | | | | |
| 30 | | | | |

| Ficha de observación: Indicador Tiempo medio por caso de prueba / Post Test | | | | |
|---|-------|--|------------------------------------|---|
| Investigador | | | Wendy Ly Morales Yovera | |
| Proceso observado | | | Pruebas de software | |
| Post Test | | | | |
| N.º de Obs. | Fecha | Tiempo total en ejecutar los casos de prueba | Total de casos de prueba ejecutado | Tiempo medio por caso de prueba=Tiempo total en ejecutar los casos de prueba / Total de casos de prueba ejecutado |
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | | | | |
| 10 | | | | |
| 11 | | | | |
| 12 | | | | |
| 13 | | | | |
| 14 | | | | |
| 15 | | | | |
| 16 | | | | |
| 17 | | | | |
| 18 | | | | |
| 19 | | | | |
| 20 | | | | |
| 21 | | | | |
| 22 | | | | |
| 23 | | | | |
| 24 | | | | |
| 25 | | | | |
| 26 | | | | |
| 27 | | | | |
| 28 | | | | |
| 29 | | | | |
| 30 | | | | |

Guía de observación N.º 2. Indicador número de casos de prueba ejecutados

| Ficha de observación: Indicador número de casos de prueba ejecutados / Pre Test | | | | |
|---|-------|--------------------------------------|--------------------------|--|
| Investigador | | Wendy Ly Morales Yovera | | |
| Proceso observado | | Pruebas de software | | |
| Pre Test | | | | |
| N.º de Obs. | Fecha | Número de Casos de Prueba Ejecutados | Horas – Hombre Laboradas | Número de Casos de Prueba ejecutados en una hora= Número de Casos de Prueba Ejecutados / Horas – Hombre Laboradas |
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | | | | |
| 10 | | | | |
| 11 | | | | |
| 12 | | | | |
| 13 | | | | |
| 14 | | | | |
| 15 | | | | |
| 16 | | | | |
| 17 | | | | |
| 18 | | | | |
| 19 | | | | |
| 20 | | | | |
| 21 | | | | |
| 22 | | | | |
| 23 | | | | |
| 24 | | | | |
| 25 | | | | |
| 26 | | | | |
| 27 | | | | |
| 28 | | | | |
| 29 | | | | |
| 30 | | | | |

| Ficha de observación: Indicador número de casos de prueba ejecutados / Post Test | | | | |
|--|-------|--------------------------------------|--------------------------|--|
| Investigador | | | Wendy Ly Morales Yovera | |
| Proceso observado | | | Pruebas de software | |
| Post Test | | | | |
| N.º de Obs. | Fecha | Número de Casos de Prueba Ejecutados | Horas – Hombre Laboradas | Número de Casos de Prueba ejecutados en una hora= Número de Casos de Prueba Ejecutados / Horas – Hombre Laboradas |
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | | | | |
| 10 | | | | |
| 11 | | | | |
| 12 | | | | |
| 13 | | | | |
| 14 | | | | |
| 15 | | | | |
| 16 | | | | |
| 17 | | | | |
| 18 | | | | |
| 19 | | | | |
| 20 | | | | |
| 21 | | | | |
| 22 | | | | |
| 23 | | | | |
| 24 | | | | |
| 25 | | | | |
| 26 | | | | |
| 27 | | | | |
| 28 | | | | |
| 29 | | | | |
| 30 | | | | |

Anexo 6. Certificado de validación del instrumento de recolección de datos

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO QUE MIDE LA VARIABLE: EFICIENCIA

| N° | INDICADORES | Pertinencia ¹ | | Relevancia ² | | Claridad ³ | | Sugerencias |
|----|--|--------------------------|----|-------------------------|----|-----------------------|----|-------------|
| | | Si | No | Si | No | Si | No | |
| 1 | INDICADOR 1-. Tiempo medio por caso de prueba $\text{Tiempo medio por caso de prueba} = \frac{\text{Tiempo total en ejecutar los casos de prueba}}{\text{Total de casos de prueba ejecutados}}$ | X | | X | | X | | |
| 2 | INDICADOR 2-. Número de casos de prueba ejecutados $\text{Número de casos de prueba ejecutados} = \frac{\text{Número de casos de prueba}}{\text{Horas hombre laboradas}}$ | X | | X | | X | | |

Observaciones (precisar si hay suficiencia): _____

✓ **Opinión de aplicabilidad:** **Aplicable** [x] **Aplicable después de corregir** [] **No aplicable** []

Apellidos y nombres del juez validador: Acuña Benites, Marlon Frank **DNI:** 42097456

Especialidad del validador: Metodólogo

...7.de...Junio del 2022..



Firma del Experto Informante.
Especialidad

¹**Pertinencia:** El ítem corresponde al concepto teórico formulado.
²**Relevancia:** El ítem es apropiado para representar al componente o dimensión específica del constructo
³**Claridad:** Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO QUE MIDE LA VARIABLE: EFICIENCIA

| N° | INDICADORES | Pertinencia ¹ | | Relevancia ² | | Claridad ³ | | Sugerencias |
|----|--|--------------------------|----|-------------------------|----|-----------------------|----|-------------|
| | | Si | No | Si | No | Si | No | |
| 1 | INDICADOR 1-. Tiempo medio por caso de prueba $\text{Tiempo medio por caso de prueba} = \frac{\text{Tiempo total en ejecutar los casos de prueba}}{\text{Total de casos de prueba ejecutados}}$ | X | | X | | X | | |
| 2 | INDICADOR 2-. Número de casos de prueba ejecutados $\text{Número de casos de prueba ejecutados} = \frac{\text{Número de casos de prueba}}{\text{Horas hombre laboradas}}$ | X | | X | | X | | |


Observaciones (precisar si hay suficiencia): _____

✓ **Opinión de aplicabilidad:** **Aplicable** [x] **Aplicable después de corregir** [] **No aplicable** []

Apellidos y nombres del juez validador: ROBERTO JUAN TEJADA RUIZ **DNI:** 17930425

Especialidad del validador: Metodólogo

...6..de...Junio del 2022..



Firma del Experto Informante.
Especialidad

¹**Pertinencia:** El ítem corresponde al concepto teórico formulado.
²**Relevancia:** El ítem es apropiado para representar al componente o dimensión específica del constructo
³**Claridad:** Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO QUE MIDE LA VARIABLE: EFICIENCIA

| N° | INDICADORES | Pertinencia ¹ | | Relevancia ² | | Claridad ³ | | Sugerencias |
|----|--|--------------------------|----|-------------------------|----|-----------------------|----|-------------|
| | | Si | No | Si | No | Si | No | |
| 1 | INDICADOR 1-. Tiempo medio por caso de prueba $\text{Tiempo medio por caso de prueba} = \frac{\text{Tiempo total en ejecutar los casos de prueba}}{\text{Total de casos de prueba ejecutados}}$ | X | | X | | X | | |
| 2 | INDICADOR 2-. Número de casos de prueba ejecutados $\text{Número de casos de prueba ejecutados} = \frac{\text{Número de casos de prueba}}{\text{Horas hombre laboradas}}$ | X | | X | | X | | |

Observaciones (precisar si hay suficiencia): _____

✓ Opinión de aplicabilidad: Aplicable [x] Aplicable después de corregir [] No aplicable []

Apellidos y nombres del juez validador: TARMEÑO JUSCAMAITA ESTHER DNI: 47190643

Especialidad del validador: Docente universitario de la carrera de Ing. De Sistemas

...6..de...Junio del 2022..

¹Pertinencia: El ítem corresponde al concepto teórico formulado.

²Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo

³Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión



Firma del Experto Informante.
Especialidad

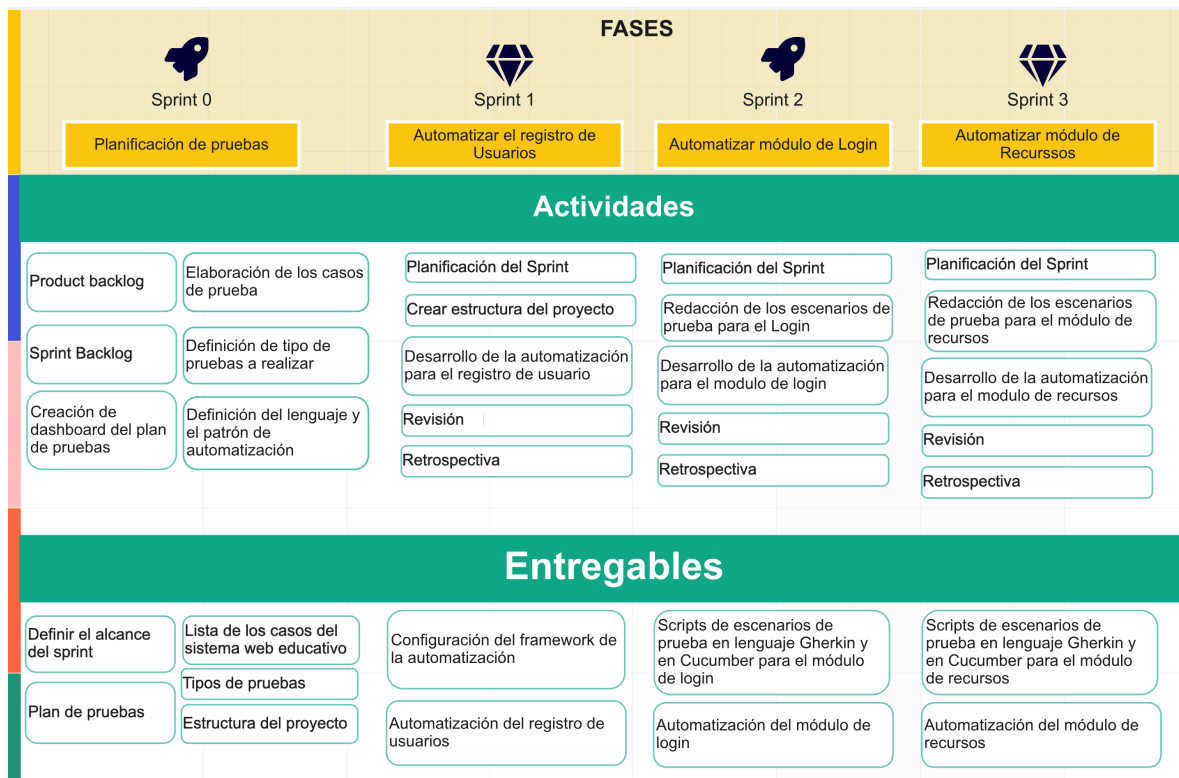
Anexo 7. Base de datos

| | Tiempo invertido por caso de prueba | | Número de casos de prueba ejecutados | |
|----|-------------------------------------|---------|--------------------------------------|---------|
| | Pretest | Postest | Pretest | Postest |
| 1 | 15,42 | 3,56 | 3,50 | 15,00 |
| 2 | 16,30 | 4,88 | 4,48 | 14,50 |
| 3 | 15,40 | 4,31 | 4,92 | 14,50 |
| 4 | 15,00 | 4,22 | 4,00 | 14,50 |
| 5 | 16,00 | 2,27 | 4,43 | 13,50 |
| 6 | 17,00 | 4,90 | 4,92 | 15,50 |
| 7 | 13,00 | 1,94 | 4,46 | 13,50 |
| 8 | 13,50 | 2,44 | 4,48 | 14,00 |
| 9 | 16,04 | 4,10 | 4,95 | 15,43 |
| 10 | 14,43 | 2,20 | 4,47 | 15,00 |
| 11 | 13,00 | 5,29 | 5,00 | 14,00 |
| 12 | 15,00 | 3,38 | 4,80 | 13,50 |
| 13 | 14,50 | 2,06 | 4,44 | 13,00 |
| 14 | 17,00 | 3,70 | 3,95 | 15,00 |
| 15 | 16,00 | 4,81 | 4,95 | 15,03 |
| 16 | 14,43 | 2,12 | 4,47 | 14,00 |
| 17 | 16,40 | 2,44 | 4,95 | 14,50 |
| 18 | 12,00 | 4,80 | 4,74 | 15,50 |
| 19 | 13,00 | 4,00 | 4,14 | 14,30 |
| 20 | 14,62 | 5,19 | 4,15 | 13,50 |
| 21 | 12,00 | 5,13 | 4,60 | 13,50 |
| 22 | 16,00 | 3,70 | 4,27 | 14,00 |
| 23 | 15,00 | 4,90 | 4,62 | 14,50 |
| 24 | 14,39 | 4,56 | 4,48 | 14,50 |
| 25 | 12,00 | 4,94 | 4,27 | 14,00 |
| 26 | 17,00 | 3,50 | 3,96 | 14,00 |
| 27 | 13,00 | 2,83 | 4,15 | 14,00 |
| 28 | 16,00 | 3,13 | 4,63 | 13,50 |
| 29 | 17,00 | 2,62 | 4,11 | 15,00 |
| 30 | 15,00 | 4,80 | 4,62 | 14,50 |

Anexo 9. Fases del desarrollo de la automatización de pruebas basado en metodología SCRUM

Para desarrollar este proyecto, se utilizó la metodología ágil de Scrum. Por ser una metodología flexible y adaptable a cambios. Las ceremonias se realizan con frecuencia, permitiendo mostrar el trabajo realizado.

A continuación, se muestra la siguiente figura las fases de trabajo, se dividió en 4 fases. El Sprint 0 correspondiente a la especificación de la planificación de las pruebas, con la finalidad de la evaluación de las pruebas que serán sometidas a automatizar. Sprint 1, se diseña, modela e implementa la automatización del primer módulo de registro de usuarios. Sprint 2, se diseña, modela e implementa la automatización del primer módulo de login y para terminar el Sprint 3, la automatización del módulo de Recursos.



Duración del proyecto:

Para la elaboración del proyecto de automatización de pruebas, fueron efectuadas desde el 7 de Marzo al 29 de Abril del 2022, correspondientes en Scrum una cantidad de 3 sprint.

| Sprint | Periodo | Semanas |
|-----------------|---------------------|----------------|
| Sprint 0 | 7 marzo – 18 marzo | 2 Semanas |
| Sprint 1 | 21 marzo – 1 abril | 2 Semanas |
| Sprint 2 | 4 abril – 15 abril | 2 Semanas |
| Sprint 3 | 18 abril – 29 abril | 2 Semanas |
| Total | | 8 Semanas |

Consideraciones para la estructura del proyecto

La construcción del proyecto de automatización de pruebas, bajo la metodología Scrum, POM para reducir el mantenimiento del código por la reutilización del código, Cucumber por la facilidad de comprensión de los criterios de la prueba y Selenium, para controlar los elementos del aplicativo web.

Según Garcia (2022) sostiene que, Page Object Model (POM), es el patrón de diseño más popular usado en conjunto con Selenium WebDriver. Arjun y Shahid(2019) indican que, algunas de las ventajas de usar POM es que todos los componentes web se pueden almacenar en un repositorio de objetos que contiene todos los componentes web para cada página web. El POM usa la clase Page Factory para crear objetos. Se archivan todos los elementos web. Todos los métodos y objetos serán seleccionados para cada sitio, permite un mantenimiento de código más fácil

García (2022) define que, Selenium es un proyecto público de código abierto que permite la automatización del navegador web. El componente central de proyecto Selenium es Selenium WebDriver, una librería que le permite controlar los

navegadores mediante programación. La automatización depende de las capacidades únicas de cada navegador.

Según Faragó et al. (2020), existen varios enfoques de desarrollo de software ágil, pero la principal es el desarrollo basado en el comportamiento por sus siglas en inglés es BDD, se centra en el comportamiento del usuario para especificar los requisitos del software en construcción. Cada escenario del usuario se considera un único caso de prueba basada en la estructura Dado, Cuando y Entonces.

Cucumber es una herramienta que admite el desarrollo impulsado por el comportamiento (BDD). Lee la especificación de ejecución en texto sin formato y verifica que el programa está haciendo lo que dice esa especificación. La especificación incluye varios ejemplos o escenarios (Musliu y Jashari, 2021).

Estructura del proyecto

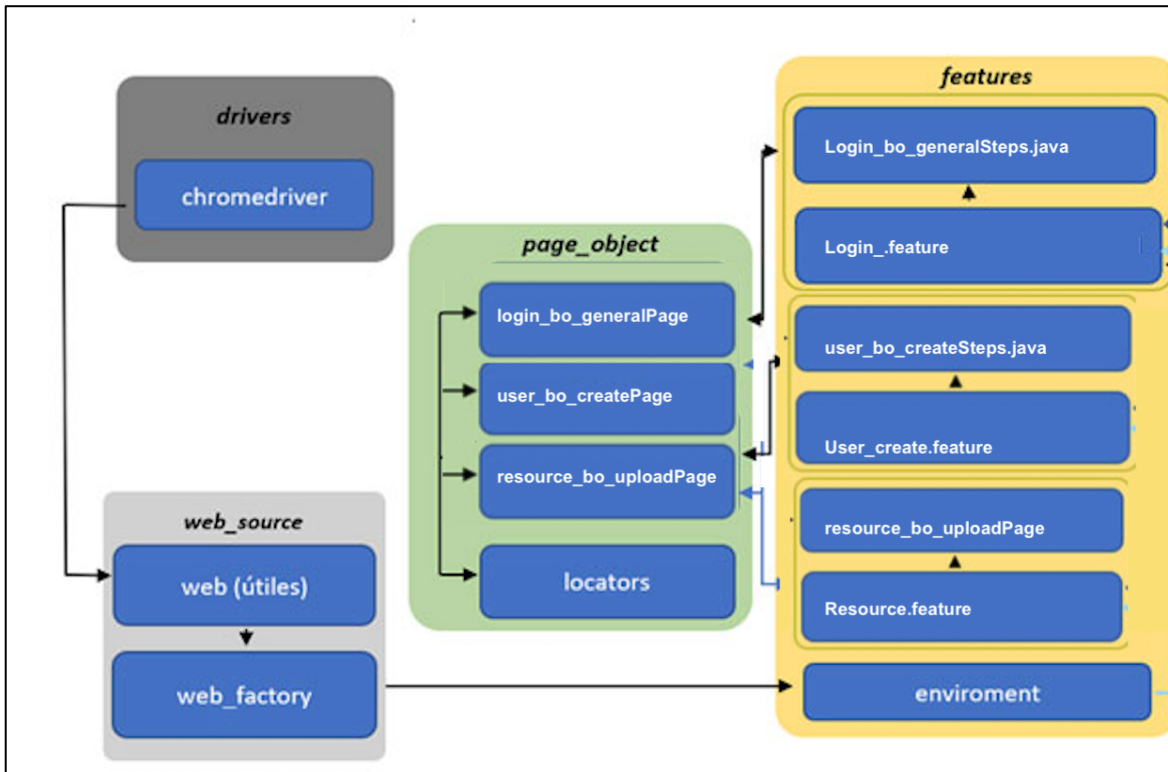
Características

- 01 Herramienta de Entorno de Desarrollo Integrado: **IntelliJ IDEA Community Edition**
- 02 Patron de Desarrollo: **Page Object Model**
- 03 Herramienta automatización: **Selenium Webdriver**
- 04 Lenguaje: **JDK Java version 8**
- 05 **Cucumber, Gherkin y Maven**

El conjunto de esta herramienta permite, crear un framework de automatización, permitiendo crear y verificar el funcionamiento del código elaborado por scripts e integrarlos al caso de prueba automatizado a la suite correspondiente. Para este proyecto, la suite está conformada por las 3 funcionalidades de la web educativa, son: registro de usuarios, login y recursos.

Para construir el framework de automatización de pruebas, se realizó bajo la metodología Scrum, POM para reducir el mantenimiento del código por la reutilización del código, Cucumber por la facilidad de comprensión de los criterios de la prueba y Selenium, para controlar los elementos del aplicativo web.

Patrón de Diseño POM (Page Object Model)



Alcance

El alcance del proyecto de automatización fue realizar 30 casos de pruebas automatizados para 3 módulos del sistema web educativo. El sistema es perteneciente a un cliente del rubro editorial, dedicada a la venta de textos educativos.

| Módulos | Casos automatizados | Tipo de Prueba |
|----------------------|----------------------------------|---------------------|
| Registro de usuarios | 10 casos de pruebas | Prueba de Regresión |
| Login | 10 casos de pruebas | Prueba de Regresión |
| Recursos | 10 casos de pruebas | Prueba de Regresión |
| Total | 30 casos de Prueba Automatizados | |

Anexo 10. Entregable del Sprint 1 – Lista de los casos de prueba a automatizar

Módulo Login

| Módulo | Funcionalidad | Escenario | |
|--------|----------------------------|--------------------|--|
| Login | Ingreso a la red educativa | 01 - Happy Path - | Usuario administrador ingresa correctamente su usuario y clave. |
| | | 02 - Happy Path - | Usuario docente de primaria ingresa correctamente su usuario y clave. |
| | | 03 - Happy Path - | Usuario docente de secundaria ingresa correctamente su usuario y clave. |
| | | 04 - Happy Path - | Usuario supervisor ingresa correctamente su usuario y clave. |
| | | 05 - Happy Path - | Usuario estudiante menor de 5 años ingresa correctamente su usuario y clave. |
| | | 06 - Happy Path - | Usuario estudiante de primaria ingresa correctamente su usuario y clave. |
| | | 07 - Happy Path - | Usuario estudiante de secundaria ingresa correctamente su usuario y clave. |
| | | 08 -UnHappy Path - | Usuario ingresa incorrectamente su usuario y clave. |
| | | 09- UnHappy Path | Usuario no está registrado en el sistema. |
| | | 10- UnHappy Path | Usuario intenta loguearse en el sistema, pero le muestra errores técnicos. |

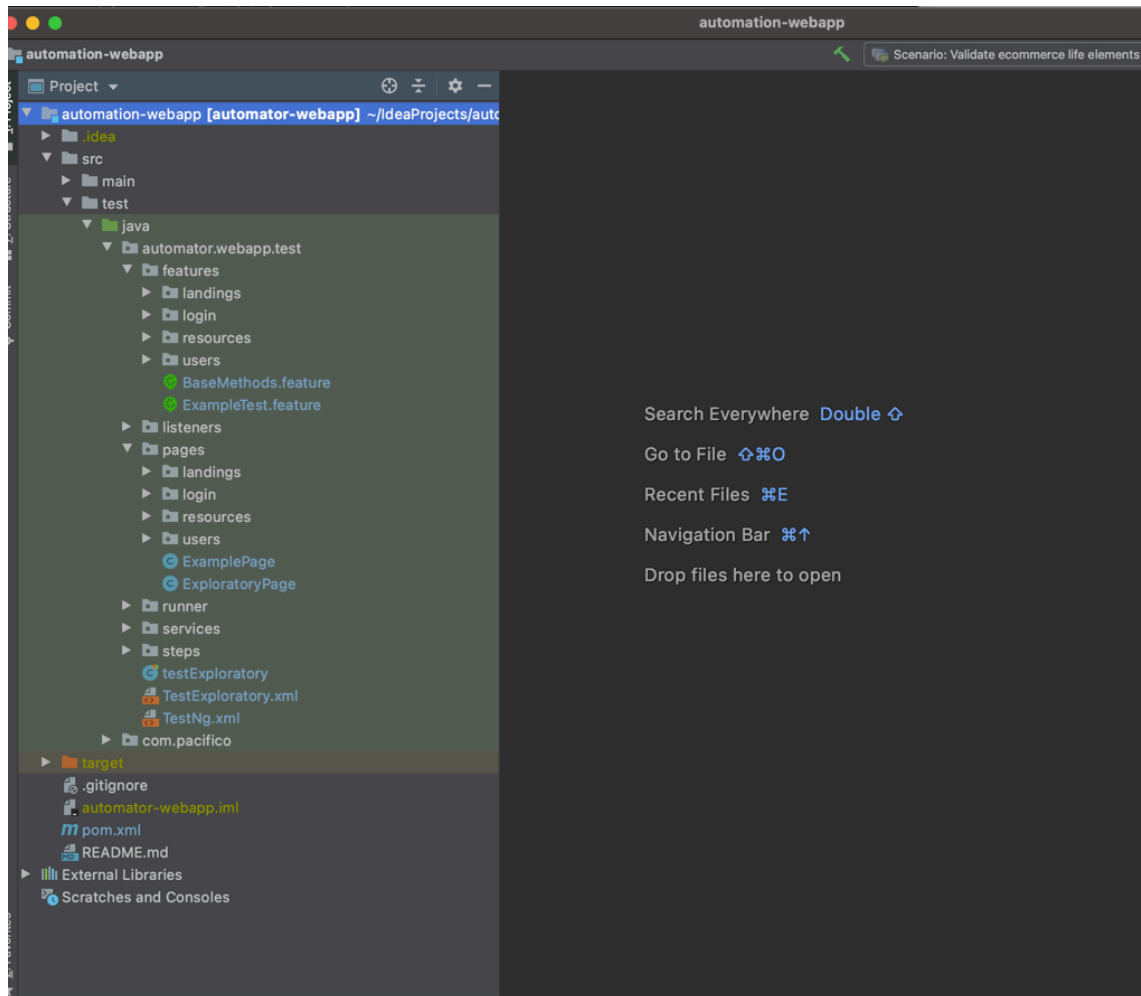
Módulo registro de usuario

| Módulo | Funcionalidad | Escenario | |
|----------------------------|-----------------------------|--------------------|---|
| Registro de usuario | Registro a la red educativa | 01 - Happy Path - | Usuario administrador registra correctamente sus datos personales e institucionales. |
| | | 02 - Happy Path - | Usuario docente de primaria registra correctamente sus datos personales e institucionales. |
| | | 03 - Happy Path - | Usuario docente de secundaria registra correctamente sus datos personales e institucionales. |
| | | 04 - Happy Path - | Usuario supervisor registra correctamente sus datos personales e institucionales. |
| | | 05 - Happy Path - | Usuario estudiante menor de 5 años registra correctamente sus datos personales e institucionales. |
| | | 06 - Happy Path - | Usuario estudiante de primaria registra correctamente sus datos personales e institucionales. |
| | | 07 - Happy Path - | Usuario estudiante de secundaria registra correctamente sus datos personales e institucionales. |
| | | 08 -UnHappy Path - | Usuario ingresa incorrectamente su datos personales e institucionales. |
| | | 09- UnHappy Path | Usuario no registra los datos obligatorios de sus datos personales e institucionales. |
| | | 10- UnHappy Path | Usuario supera el tiempo límite permitido |

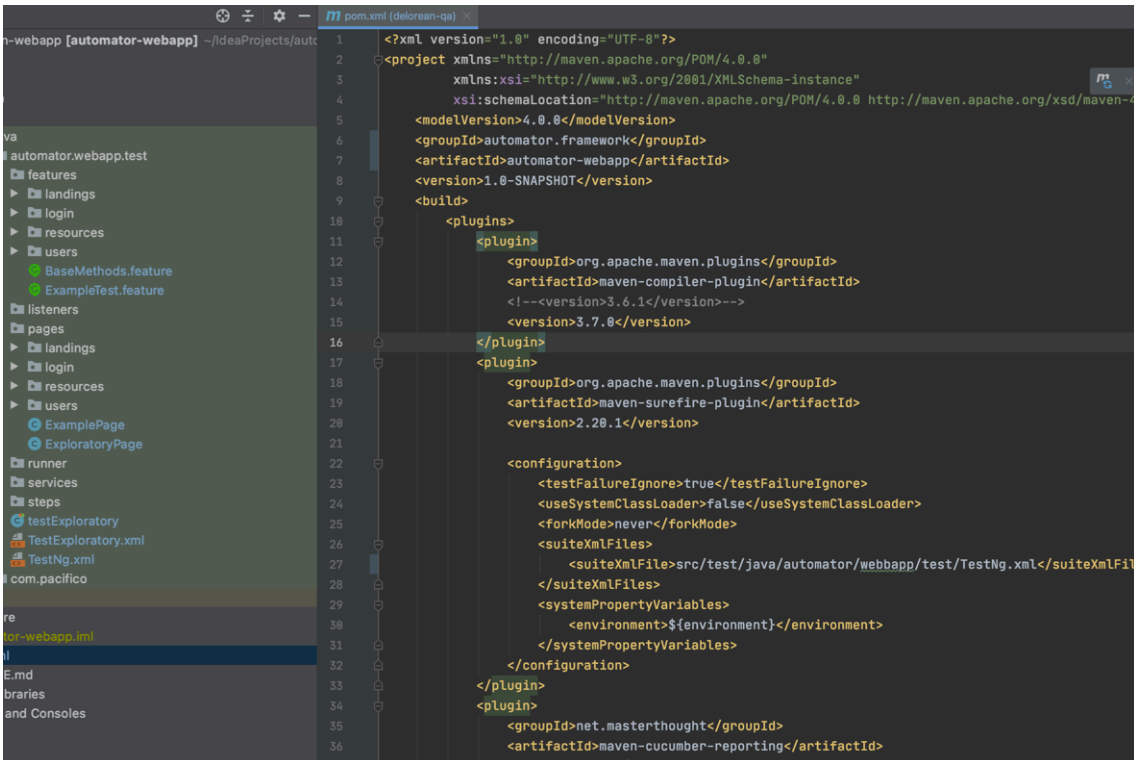
Módulo de recursos

| Módulo | Funcionalidad | Escenario | |
|----------|---|--------------------|--|
| Recursos | Usuarios realizan consulta a los recursos disponibles e interactúan | 01 - Happy Path - | Usuario administrador carga archivos en el módulo de recursos |
| | | 02 - Happy Path - | Usuario administrador realiza consultas por materia y grado estudiantil |
| | | 03 - Happy Path - | Usuario administrador realiza comentarios desde el archivo consultado. |
| | | 04 - Happy Path - | Usuario docente realiza consultas por materia y grado estudiantil |
| | | 05 - Happy Path - | Usuario docente realiza comentarios desde el archivo consultado. |
| | | 06 - Happy Path - | Usuario estudiante realiza consultas por materia y grado estudiantil |
| | | 07 - Happy Path - | Usuario estudiante realiza comentarios desde el archivo consultado. |
| | | 08 -UnHappy Path - | Usuario administrador intenta cargar archivos superiores a 100 MB en el módulo de recursos |
| | | 09- UnHappy Path | Usuario administrador intenta cargar archivos sin los datos obligatorios en el módulo de recursos |
| | | 10- UnHappy Path | Usuario administrador intenta cargar archivos con extensiones no permitidas en el módulo de recursos |

Anexo 11. Entregable del Sprint 1 - Estructura del proyecto de automatización



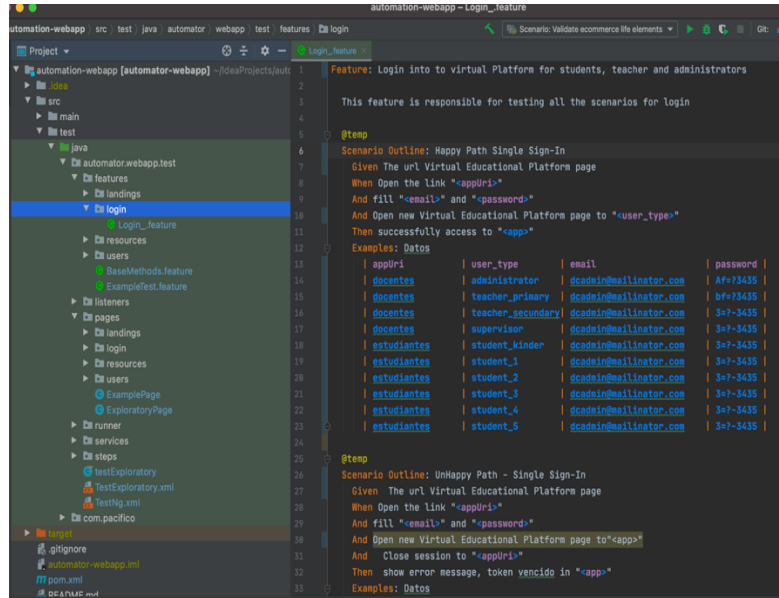
Anexo 12. Entregable del Sprint 2 – Configuración de las dependencias del framework



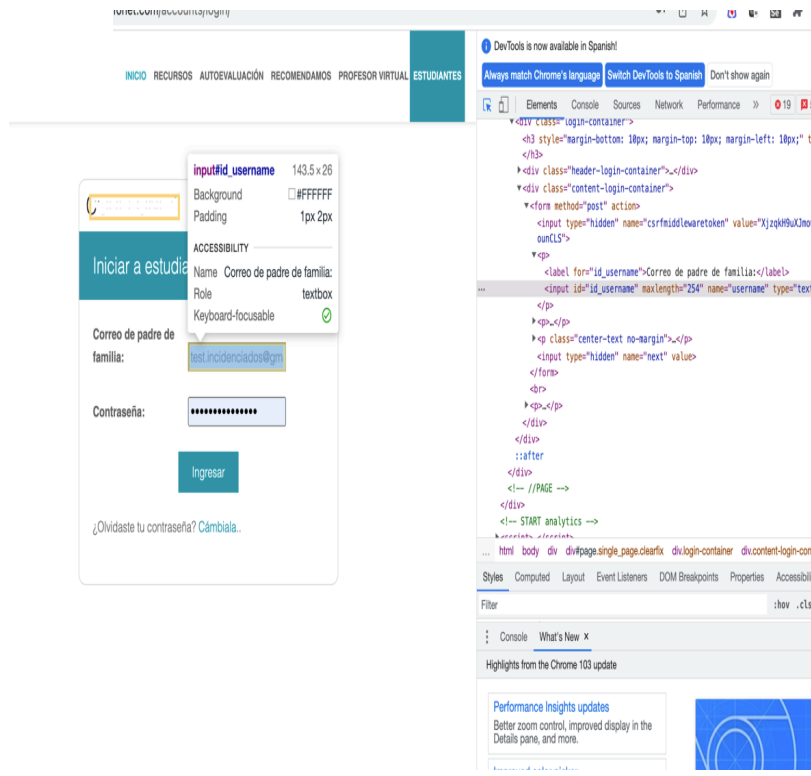
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4
5   <modelVersion>4.0.0</modelVersion>
6   <groupId>automator.framework</groupId>
7   <artifactId>automator-webapp</artifactId>
8   <version>1.0-SNAPSHOT</version>
9   <build>
10     <plugins>
11       <plugin>
12         <groupId>org.apache.maven.plugins</groupId>
13         <artifactId>maven-compiler-plugin</artifactId>
14         <!--<version>3.6.1</version-->
15         <version>3.7.0</version>
16       </plugin>
17       <plugin>
18         <groupId>org.apache.maven.plugins</groupId>
19         <artifactId>maven-surefire-plugin</artifactId>
20         <version>2.20.1</version>
21       </plugin>
22       <configuration>
23         <testFailureIgnore>true</testFailureIgnore>
24         <useSystemClassLoader>false</useSystemClassLoader>
25         <forkMode>never</forkMode>
26         <suiteXmlFiles>
27           <suiteXmlFile>src/test/java/automator/webapp/test/TestNg.xml</suiteXmlFile>
28         </suiteXmlFiles>
29         <systemPropertyVariables>
30           <environment>${environment}</environment>
31         </systemPropertyVariables>
32       </configuration>
33     </plugins>
34     <plugin>
35       <groupId>net.masterthought</groupId>
36       <artifactId>maven-cucumber-reporting</artifactId>
```

Anexo 13. Actividad del Sprint 2 – Redacción y creación de scripts de los escenarios de prueba para login

| Módulo | Escenario |
|--------|---|
| Login | 01 - Usuario administrador ingresa correctamente su usuario y clave. Happy Path - Creación del feature Login |



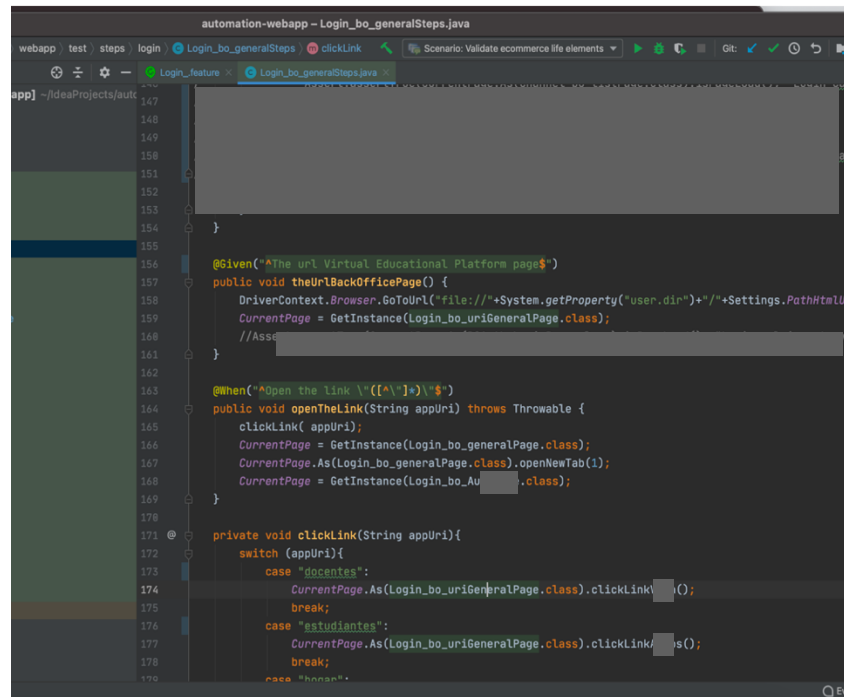
– Mapeo de objetos del módulo Login



-Creación del escenario de prueba

```
Scenario Outline: Happy Path Single Sign-In
Given The url Virtual Educational Platform page
When Open the link "<appUri>"
And fill "<email>" and "<password>"
And Open new Virtual Educational Platform page to "<user_type>"
Then successfully access to "<app>"
Examples: Datos
| appUri | user_type | email | password |
| docentes | administrator | dcadmin@mailinator.com | Af=?3435 |
```

- Creación de los Steps Definitions



```
automation-webapp - Login_bo_generalSteps.java
webapp test steps login Login_bo_generalSteps clickLink Scenario: Validate ecommerce life elements
~/IdeaProjects/auk
147
148
149
150
151
152
153
154
155
156
157 @Given("The url Virtual Educational Platform page$")
158 public void theUrlBackOfficePage() {
159     DriverContext.Browser.GoToUrl("file://" + System.getProperty("user.dir") + "/" + Settings.PathHtmlU
160     CurrentPage = GetInstance(Login_bo_uriGeneralPage.class);
161     //Asser
162 }
163
164 @When("Open the link \"([^\"]+)\")")
165 public void openTheLink(String appUri) throws Throwable {
166     clickLink( appUri);
167     CurrentPage = GetInstance(Login_bo_generalPage.class);
168     CurrentPage.As(Login_bo_generalPage.class).openNewTab(1);
169     CurrentPage = GetInstance(Login_bo_Au
170 }
171
172 private void clickLink(String appUri){
173     switch (appUri){
174         case "docentes":
175             CurrentPage.As(Login_bo_uriGeneralPage.class).clickLink(
176             break;
177         case "estudiantes":
178             CurrentPage.As(Login_bo_uriGeneralPage.class).clickLink(
179             break;
180         case "honor":
```

02 - Usuario docente de primaria ingresa correctamente su usuario y clave.

Happy

Path -

-Creación del escenario de prueba

```
Scenario Outline: Happy Path Single Sign-In
Given The url Virtual Educational Platform page
When Open the link "<appUri>"
And fill "<email>" and "<password>"
And Open new Virtual Educational Platform page to "<user_type>"
Then successfully access to "<app>"
Examples: Datos
| appUri | user_type | email | password |
| docentes | teacher_primary | dcadmin@mailinator.com | bf=?3435 |
```

03 - Usuario docente de secundaria ingresa correctamente su usuario y clave.

Happy

Path - -Creación del escenario de prueba

```
Scenario Outline: Happy Path Single Sign-In
Given The url Virtual Educational Platform page
When Open the link "<appUri>"
And fill "<email>" and "<password>"
And Open new Virtual Educational Platform page to "<user_type>"
Then successfully access to "<app>"
Examples: Datos
| appUri | user_type | email | password |
| docentes | teacher_secondary | dcadmin@mailinator.com | 3=?-3435 |
```

04 - Usuario supervisor ingresa correctamente su usuario y clave.

Happy

Path - -Creación del escenario de prueba

```
Scenario Outline: Happy Path Single Sign-In
Given The url Virtual Educational Platform page
When Open the link "<appUri>"
And fill "<email>" and "<password>"
And Open new Virtual Educational Platform page to "<user_type>"
Then successfully access to "<app>"
Examples: Datos
| appUri | user_type | email | password |
| docentes | supervisor | dcadmin@mailinator.com | 3=?-3435 |
```

05 - Usuario estudiante menor de 5 años ingresa correctamente su usuario y clave.

Happy

Path - -Creación del escenario de prueba

```
Scenario Outline: Happy Path Single Sign-In
Given The url Virtual Educational Platform page
When Open the link "<appUri>"
And fill "<email>" and "<password>"
And Open new Virtual Educational Platform page to "<user_type>"
Then successfully access to "<app>"
Examples: Datos
| appUri | user_type | email | password |
| estudiantes | student_kinder | dcadmin@mailinator.com | 3=?-3435 |
```

06 - Usuario estudiante de primaria ingresa correctamente su usuario y clave.

Happy

Path - -Creación del escenario de prueba

```
Scenario Outline: Happy Path Single Sign-In
Given The url Virtual Educational Platform page
When Open the link "<appUri>"
And fill "<email>" and "<password>"
And Open new Virtual Educational Platform page to "<user_type>"
Then successfully access to "<app>"
Examples: Datos
```

| appUri | user_type | email | password |
|-------------|-----------|------------------------|----------|
| estudiantes | student_1 | dcadmin@mailinator.com | 3=?-3435 |
| estudiantes | student_2 | dcadmin@mailinator.com | 3=?-3435 |
| estudiantes | student_3 | dcadmin@mailinator.com | 3=?-3435 |
| estudiantes | student_4 | dcadmin@mailinator.com | 3=?-3435 |
| estudiantes | student_5 | dcadmin@mailinator.com | 3=?-3435 |

07 - Usuario estudiante de secundaria ingresa correctamente su usuario y clave.

Happy

Path -

-Creación del escenario de prueba

```
Scenario Outline: Happy Path Single Sign-In
Given The url Virtual Educational Platform page
When Open the link "<appUri>"
And fill "<email>" and "<password>"
And Open new Virtual Educational Platform page to "<user_type>"
Then successfully access to "<app>"
```

Examples: Datos

| appUri | user_type | email | password |
|-------------|-----------|------------------------|----------|
| estudiantes | student_1 | dcadmin@mailinator.com | 3=?-3435 |
| estudiantes | student_2 | dcadmin@mailinator.com | 3=?-3435 |
| estudiantes | student_3 | dcadmin@mailinator.com | 3=?-3435 |
| estudiantes | student_4 | dcadmin@mailinator.com | 3=?-3435 |
| estudiantes | student_5 | dcadmin@mailinator.com | 3=?-3435 |

08 - Usuario ingresa incorrectamente su usuario y clave.

UnHappy

Path -

-Creación del escenario

```
Scenario Outline: Login failed, Password incorrect
Given The url Virtual Educational Platform page
When Open the link "<appUri>"
And fill "<email>" and "<password>" to "<user_type>"
Then show error message, password incorrect
Examples: Datos
```

| | | | |
|-------------|-----------|--------------------------|----------|
| estudiantes | student_5 | 44dcadmin@mailinator.com | 1=?-3435 |
|-------------|-----------|--------------------------|----------|

-Creación del step definition para el caso fail

```
@Then("^show error message, password incorrect$")
public void showErrorMessagePasswordIncorrect() {
    Assert.assertEquals(CurrentPage.As(Login_bo_generalPage.class).getLabelPasswordError(), "Su
    cuenta o contraseña no es correcta. Si no recuerda su contraseña, puede restablecerla ahora.");
}
```

09- Usuario no está registrado en el sistema.

UnHappy

Path

-Creación del escenario

```
Scenario Outline: Login failed, user do not exist
Given The url Virtual Educational Platform page
When Open the link "<appUri>"
And fill "<email>" and "<password>" to "<user_type>"
Then show error message, user do not exist
Examples: Datos
```

| | | | |
|-------------|------------|--------------------------|----------|
| estudiantes | student_21 | 411cadmin@mailinator.com | 1=?-3435 |
|-------------|------------|--------------------------|----------|

-Creación del step definition

```
@Then("^show error message, technical fail$")
public void showErrorMessageError() {
    Assert.assertEquals(CurrentPage.As(Login_bo_generalPage.class).getLabelError(), "Presentamos
inconvenientes, vuelva intentarlo en unos minutos.");
}
```

10- Usuario intenta loguearse en el sistema, pero le muestra errores técnicos.

UnHappy -Creación del escenario

Path

```
Scenario Outline: Login failed, technical fail
Given The url Virtual Educational Platform page
When Open the link "<appUri>"
And fill "<email>" and "<password>" to "<user_type>"
Then show error message, technical fail
Examples: Datos
| estudiantes | student_0 | 44dcadmin@mailinator.com | 1=?-3435 |
```

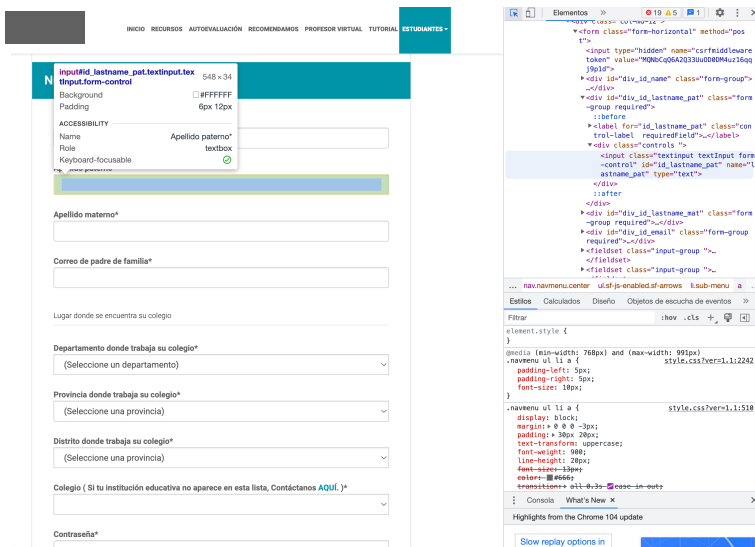
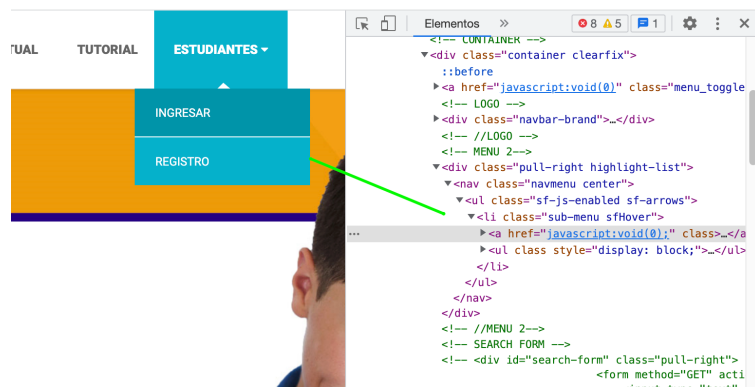
-Creación del step definition

```
@Then("^show error message, technical fail$")
public void showErrorMessageError () {
    Assert.assertEquals(CurrentPage.As(Login_bo_generalPage.class).getLabelError(), "Presentamos
inconvenientes, vuelva intentarlo en unos minutos.");
}
```

Anexo 14. Actividad del sprint 1 – Redacción y creación de scripts de los escenarios para registro de usuario

| Módulo | Escenario |
|---------------------|--|
| Registro de usuario | 01 - Happy Path – Usuario administrador registra correctamente sus datos personales e institucionales. |

-Mapeo de objetos



-Creación del feature

```

Scenario Outline: Happy Path- User create admin to web
Given I want to create user <type_user> into the page
And I enter "< name>", "< lastname_father>" and "< lastname_mother>"
| name | lastname_father | lastname_mother |
| Maria Rosa | Torres | Pearson |
And I enter my personal email "< email>"
| email |
| mtorresp@maillu.com |
And I search my location "<departamento>", "<provincia>" and "<distrit>" in list and click
| departamento | provincia | distrit |
| Lima | Lima | Carabayllo |
And I enter my educational institution "< name_institution>"
| name_institution |
| I.E. Perú -China |
And I create my password "< pwd>"
| pwd |
| P=$D12_90a |
And I enter my password to confirmed "<pwd>"
| pwd |
| P=$D12_90a |
When I go click save button
Then I create successful my user

```

Filtro por tipo de usuario

```

public void filterByStatus(String typeUser) {
    switch (aStatusTest) {
        case "admin":
            [redacted].click();
            break;
        case "teacher":
            [redacted].click();
            break;
        case "supervisor":
            [redacted].med.click();
            break;
        case "student_kid":
            [redacted].ft.click();
            break;
        case "student_prim":
            [redacted].ed.click();
            break;
        case "student_high":
            [redacted].click();
            break;
        default:
            Assert.fail("Don't exist [" + aStatusTest + "]");
    }
    CommonUtil.waitSpinlco();
}

```

Envío del registro

```

public user_bo_createPage pressCreateUser() {
    CommonUtil.scrollIntoViewF(btnCreateAccountUser);
    btnCreateAccountUser.click();
    return GetInstance(user_bo_createPage.class);
}

```

02 - Happy
Path -

Usuario docente de primaria registra correctamente sus
datos personales e institucionales.

Creación del feature

```

Scenario Outline: Happy Path- User create admin to web
Given I want to create user <type_user> into the page
And I enter "<name>", "<lastname_father>" and "<lastname_mother>"
| name | lastname_father | lastname_mother |
| Maria Rosa | Torres | Pearson |
And I enter my personal email "<email>"
| email |
| mtorresp@maillu.com |
And I search my location "<departamento>", "<provincia>" and "<distrit>" in list and click
| departamento | provincia | distrit |
| Lima | Lima | Carabaylo |
And I enter my educational institution "<name_institution>"
| name_institution |
| I.E. Perú -China |
And I create my password "<pwd>"
| pwd |
| P=$D12_90a |
And I enter my password to confirmed "<pwd>"
| pwd |
| P=$D12_90a |
When I go click save button
Then I create successful my user
Examples:
| type_user |
| teacher |

```

Filtro por tipo de usuario

```

public void filterByStatus(String typeUser) {
    switch (aStatusTest) {
        case "admin":
            All.click();
            break;
        case "teacher":
            e.click();
            break;
        case "supervisor":
            ed.click();
            break;
        case "student_kid":
            t.click();
            break;
        case "student_prim":
            .click();
            break;
        case "student_hign":
            ed.click();
            break;
        default:
            Assert.fail("Don't exist [" + aStatusTest + "]");
    }
    CommonUtil.waitSpinlco();
}

```

Envío del registro

```

public user_bo_createPage pressCreateUser() {
    CommonUtil.scrollToViewF(btnCreateAccountUser);
    btnCreateAccountUser.click();
    return GetInstance(user_bo_createPage.class);
}

```

03 - Happy
Path -

Usuario docente de secundaria registra correctamente sus
datos personales e institucionales.

Creación del feature

```

Scenario Outline: Happy Path- User create admin to web
Given I want to create user <type_user> into the page
And I enter "<name>", "<lastname_father>" and "<lastname_mother>"
| name | lastname_father | lastname_mother |
| Maria Rosa | Torres | Pearson |
And I enter my personal email "<email>"
| email |
| mtorresp@maillu.com |
And I search my location "<departamento>", "<provincia>" and "<distrit>" in list and click
| departamento | provincia | distrit |
| Lima | Lima | Carabayllo |
And I enter my educational institution "<name_institution>"
| name_institution |
| I.E. Perú -China |
And I create my password "<pwd>"
| pwd |
| P=$D12_90a |
And I enter my password to confirmed "<pwd>"
| pwd |
| P=$D12_90a |
When I go click save button
Then I create successful my user
Examples:
| type_user |
| supervisor |

```

Filtro por tipo de usuario

```

public void filterByStatus(String typeUser) {
    switch (aStatusTest) {
        case "admin":
            btnAdminAll.click();
            break;
        case "teacher":
            btnTeacherAll.click();
            break;
        case "supervisor":
            btnSupervisorProgramed.click();
            break;
        case "student_kid":
            btnStudentKid.click();
            break;
        case "student_prim":
            btnStudentPrim.click();
            break;
        case "student_hign":
            btnStudentHign.click();
            break;
        default:
            Assert.fail("Don't exist [" + aStatusTest + "]");
    }
    CommonUtil.waitSpinCo();
}

```

Envío del registro

```

public user_bo_createPage pressCreateUser() {
    CommonUtil.scrollIntoViewF(btnCreateAccountUser);
    btnCreateAccountUser.click();
    return GetInstance(user_bo_createPage.class);
}

```

05 - Happy
Path -

Usuario estudiante menor de 5 años registra correctamente sus datos personales e institucionales.

Creación del feature


```

Scenario Outline: Happy Path- User create admin to web
Given I want to create user <type_user> into the page
And I enter "< name>", "< lastname_father>" and "< lastname_mother>"
| name | lastname_father | lastname_mother |
| Maria Rosa | Torres | Pearson |
And I enter my personal email "< email>"
| email |
| mtorresp@maillu.com |
And I search my location "<departamento>", "<provincia>" and "<distrit>" in list and click
| departamento | provincia | distrit |
| Lima | Lima | Carabayllo |
And I enter my educational institution "< name_institution>"
| name_institution |
| I.E. Perú-China |
And I create my password "< pwd>"
| pwd |
| P=${D12}_90a |
And I enter my password to confirmed "<pwd>"
| pwd |
| P=${D12}_90a |
When I go click save button
Then I create successful my user
Examples:
| type_user |
| student_kid |

```

Filtro por tipo de usuario

```

public void filterByStatus(String typeUser) {
    switch (aStatusTest) {
        case "admin":
            // btnAdmin.click();
            break;
        case "teacher":
            // btnTeacher.click();
            break;
        case "supervisor":
            // btnSupervisor.click();
            break;
        case "student_kid":
            // btnStudentKid.click();
            break;
        case "student_prim":
            // btnStudentPrim.click();
            break;
        case "student_hign":
            // btnStudentHign.click();
            break;
        default:
            Assert.fail("Don't exist [" + aStatusTest + "]");
    }
    CommonUtil.waitSpinco();
}

```

Envío del registro

```

public user_bo_createPage pressCreateUser() {
    CommonUtil.scrollToViewF(btnCreateAccountUser);
    btnCreateAccountUser.click();
    return GetInstance(user_bo_createPage.class);
}

```

06 - Happy
Path -

Usuario estudiante de primaria registra correctamente sus
datos personales e institucionales.

Creación del feature

```

Scenario Outline: Happy Path- User create admin to web
Given I want to create user <type_user> into the page
And I enter "<name>", "<lastname_father>" and "<lastname_mother>"
| name      | lastname_father | lastname_mother |
| Maria Rosa | Torres          | Pearson          |
And I enter my personal email "<email>"
| email      |
| mtorresp@maillu.com |
And I search my location "<departamento>", "<provincia>" and "<distrit>" in list and click
| departamento | provincia | distrit |
| Lima         | Lima     | Carabaylo |
And I enter my educational institution "<name_institution>"
| name_institution |
| I.E. Perú -China |
And I create my password "<pwd>"
| pwd      |
| P=$D12_90a |
And I enter my password to confirmed "<pwd>"
| pwd      |
| P=$D12_90a |
When I go click save button
Then I create successful my user
Examples:
| type_user |
| student_prim |

```

Filtro por tipo de usuario

```

public void filterByStatus(String typeUser) {
    switch (aStatusTest) {
        case "admin":
            All.click();
            break;
        case "teacher":
            aStatusTest.click();
            break;
        case "supervisor":
            All.click();
            break;
        case "student_kid":
            aStatusTest.click();
            break;
        case "student_prim":
            aStatusTest.click();
            break;
        case "student_hign":
            aStatusTest.click();
            break;
        default:
            Assert.fail("Don't exist [" + aStatusTest + "]");
    }
    CommonUtil.waitSpinlco();
}

```

Envío del registro

```

public user_bo_createPage pressCreateUser() {
    CommonUtil.scrollToViewF(btnCreateAccountUser);
    btnCreateAccountUser.click();
    return GetInstance(user_bo_createPage.class);
}

```

07 - Happy
Path -

Usuario estudiante de secundaria registra correctamente sus datos personales e institucionales.

Creación del feature

```

Scenario Outline: Happy Path- User create student_high to web
Given I want to create user <type_user> into the page
And I enter "< name>", "< lastname_father>" and "< lastname_mother>"
| name | lastname_father | lastname_mother |
| Maria Rosa | Torres | Pearson |
And I enter my personal email "< email>"
| email |
| mtorresp@maillu.com |
And I search my location "<departamento>", "<provincia>" and "<distrit>" in list and click
| departamento | provincia | distrit |
| Lima | Lima | Carabayllo |
And I enter my educational institution "< name_institution>"
| name_institution |
| I.E. Perú -China |
And I create my password "<pwd>"
| pwd |
| P=$D12_90a |
And I enter my password to confirmed "<pwd>"
| pwd |
| P=$D12_90a |
When I go click save button
Then I create successful my user
Examples:
| type_user |
| student_hign |

```

Filtro por tipo de usuario

```

public void filterByStatus(String typeUser) {
    switch (aStatusTest) {
        case "admin":
            [redacted].click();
            break;
        case "teacher":
            [redacted].click();
            break;
        case "supervisor":
            [redacted].click();
            break;
        case "student_kid":
            [redacted].click();
            break;
        case "student_prim":
            [redacted].click();
            break;
        case "student_hign":
            [redacted].click();
            break;
        default:
            Assert.fail("Don't exist [" + aStatusTest + "]");
    }
    CommonUtil.waitSpinlco();
}

```

Envío del registro

```

public user_bo_createPage pressCreateUser() {
    CommonUtil.scrollToViewF(btnCreateAccountUser);
    btnCreateAccountUser.click();
    return GetInstance(user_bo_createPage.class);
}

```

08 -UnHappy
Path -

Usuario ingresa incorrectamente su datos personales e institucionales.

```

@Then("^show error message, information incorrect$")
public void showErrorMessagePersonInformationIncorrect() {
    Assert.assertEquals(CurrentPage.As(user_bo_createPage.class).getLabelMessageError(), "Ha ingresado datos incorrectos, revise los campo y enviar el formulario nuevamente.");
}

```

09- UnHappy
Path

Usuario no registra los datos obligatorios de sus datos personales e institucionales.

```
@Then("^show error message, labels empty$")
public void showErrorMessageEmptyInformationIncorrect() {
    Assert.assertEquals(CurrentPage.As(user_bo_createPage.class).getLabelMessageError(), "Complete
los datos obligatorios.");
}
```

10- UnHappy
Path

Usuario supera el tiempo límite permitido

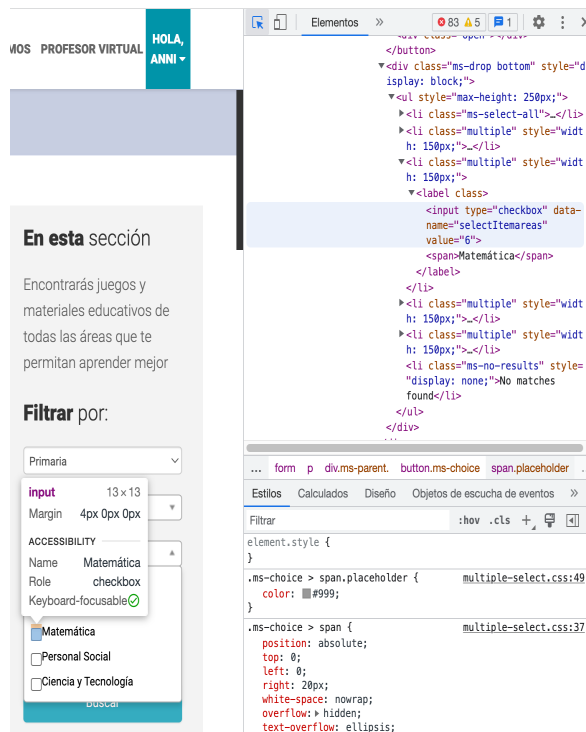
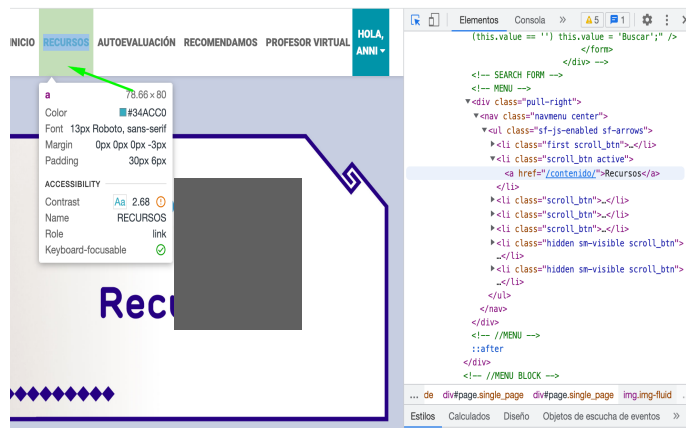
```
@Then("^show error message, token vencido in \\\"([\\\"]*)\\\"$")
public void showErrorMessageTokenVencidoIn(String app) throws Throwable {
    CurrentPage= GetInstance(user_bo_createlPage.class);
    CurrentPage.As(user_bo_createlPage.class).openNewTab(2);
    CommonUtil.waitSpinlco();
    CurrentPage = GetInstance(user_bo_createPage.class);
    Assert.assertEquals("Su sesion ha expirado",
    CurrentPage.As(user_bo_createPage.class).getLabelErrorMessage());
}
```

Anexo 15. Actividad del sprint 3 – Redacción y creación de scripts de los escenarios para recursos

| Módulo | Escenario |
|--------|-----------|
|--------|-----------|

| | |
|----------|---|
| Recursos | 01 - Happy Path - Usuario administrador carga archivos en el módulo de recursos |
|----------|---|

-Mapeo de objetos del módulo de Recursos



-Creación de Escenario

```

Scenario Outline: Happy Path- Upload file in Resources
  Give I search "<type_resource>" with filter in list and click
  And I press create resource button
  And I save are filters successful , enter name information and key words
  | level | grade | materia | fileName | key words |
  | primary | grade_5 | Math | Exercises Sto | exercises,math,5,important |
  And I search "<promotion name>" in life list and click edit link
  And I upload file successful
  | file |
  | imgPng179x282.png |

  When I save "<type_resource>" in resources
  Then I save "<type_resource>" successful
  And I show message successful
  Examples:
  | type_resource |
  | file_book |
  
```

-Creación de step definition

```
@Then("I upload file successful")
public void IValidateUploadResourceFile(DataTable table) {
    CucumberNewUtil.ConvertDataTableToDict(table);
    List<List<String>> data = table.raw();

    CurrentPage.As(IValidateUploadResourceFile)
        .validateLinkLabel("Subir", "Subir", "Subir", "Subir",
        "Subir", "Subir");

    for (int i = 1; i < data.size(); i++) {
        String aFileName = CucumberNewUtil.GetCellValueWithRowIndex("fileName", i);
        String aFileNameLevel = CucumberNewUtil.GetCellValueWithRowIndex("level", i);
        String aFileNameGrade = CucumberNewUtil.GetCellValueWithRowIndex("grade", i);
        String aFileNameMateria = CucumberNewUtil.GetCellValueWithRowIndex("level", i);
        String aFileNameConfirmation =
        CucumberNewUtil.GetCellValueWithRowIndex("fileConfirmation", i);

        CurrentPage.As(resource_bo_Page.class).fillFile(file);
        CurrentPage.As(resource_bo_Page.class).namesImages(aFileName, aFileNameLevel,
        aFileNameConfirmation, aFileNameGrade, aFileNameMaterial);
    }

    CurrentPage.As(IValidateUploadResourceFile)
        .validateLinkLabel("Cambiar nombre", "Cambiar nivel",
        "Cambiar grado", "Cambiar materia", "Cambiar documento");
}
```

02 - Happy Path - Usuario administrador realiza consultas por materia y grado estudiantil

```
public resource_bo_Page pressSearchResources() {
    CommonUtil.scrollToViewF(btnSearchResources);
    btnSearchResources.click();
    return GetInstance(resource_bo_Page.class);
}
```

03 - Happy Path - Usuario administrador realiza comentarios desde el archivo consultado.

```
public resource_bo_Page sendCommentsAdmin() {
    CommonUtil.scrollToViewF(btnSendCommentsAdmin);
    btnSendCommentsAdmin.click();
    return GetInstance(resource_bo_Page.class);
}
```

04 - Happy Path - Usuario docente realiza consultas por materia y grado estudiantil

```
public resource_bo_Page pressSearchResources() {
    CommonUtil.scrollToViewF(btnSearchResources);
    btnSearchResources.click();
    return GetInstance(resource_bo_Page.class);
}
```

05 - Happy Path - Usuario docente realiza comentarios desde el archivo consultado.

```
public resource_bo_Page sendCommentsTeacherToStudent() {
    CommonUtil.scrollToViewF(btnSendCommentsTeacherToStudent);
    btnSendCommentsTeacherToStudent.click();
    return GetInstance(resource_bo_Page.class);
}
```

06 - Happy Path - Usuario estudiante realiza consultas por materia y grado estudiantil

```
public resource_bo_Page sendCommentsStudentToTeacher() {
    CommonUtil.scrollToViewF(btnSendCommentsStudentToTeacher);
    btnSendCommentsStudentToTeacher.click();
    return GetInstance(resource_bo_Page.class);
}
```

07 - Happy Path - Usuario estudiante realiza comentarios desde el archivo consultado.

Anexo 16. Actividad del sprint 1,2 y3 – Generación de reporte con cucumber

Reporte cucumber

Configurar las dependencias de cucumber-report en el archivo pom.xml

```
<plugin>
<groupId>net.masterthought</groupId>
<artifactId>maven-cucumber-reporting</artifactId>
<version>2.8.0</version>
<executions>
  <execution>
    <id>execution</id>
    <phase>verify</phase>
    <goals>
      <goal>generate</goal>
    </goals>
    <configuration>
      <projectName>automator-webapp</projectName>

<outputDirectory>${project.build.directory}</outputDirectory>

<cucumberOutput>${project.build.directory}/cucumber.json</cucumberOutput>
      <skippedFails>>true</skippedFails>
    </configuration>
  </execution>
</executions>
</plugin>
```


Para generar el reporte en la terminal del IDE de IntelliJ IDEA Community

➤ mvn verify

El comando, realizará la ejecución de los 3 Features en cada features se escribieron 10 escenario de prueba, en total resulta de 30 escenarios. A continuación, se visualiza su reporte en versión .html.

