

UNIVERSIDAD NACIONAL
SANTIAGO ANTÚNEZ DE MAYOLO
FACULTAD DE CIENCIAS
ESCUELA PROFESIONAL DE
INGENIERÍA DE SISTEMAS E INFORMÁTICA



QUALITY ASSURANCE ENFOCADO EN AGILE TESTING PARA MEJORAR LA CALIDAD DE SOFTWARE EN LA EMPRESA WASI INNOVATION, LIMA - 2022

TESIS PARA OPTAR EL TITULO PROFESIONAL DE
INGENIERO DE SISTEMAS E INFORMÁTICA

AUTOR:

Bach. COLONIA BALLICO, JEAN FRANKLIN

ASESOR:

MSc. Ing. TREJO FLORES, WILFREDO MANUEL

Huaraz – Perú

2022

Nº Registro: **T116**



DEDICATORIA

A mi Padre y mi Abuela que desde el cielo me dan la fortaleza necesaria para continuar a pesar de las adversidades.

A mi Madre, por el apoyo incondicional en los momentos más difíciles, por sus consejos y su apoyo total hacia mí para superarme tanto como persona y profesional.

A mis hermanos que son la motivación para mejorar cada día como persona y futuro profesional.

A mi pareja, por su comprensión y dedicación en las situaciones complicadas, brindándome el soporte y motivación necesario en el día a día para culminar la investigación.

Y finalmente a mí, por haberme dado una oportunidad de mejorar a pesar de los pocos recursos para lograr ser profesional y por haber realizado este proyecto.

Jean Franklin Colonia Ballico.

AGRADECIMIENTO

A mi Padre y mi Abuela que desde el cielo me dan la fortaleza necesaria para continuar a pesar de las adversidades.

A mi Madre, por el apoyo incondicional en los momentos más difíciles, por sus consejos y su apoyo total hacia mí para superarme tanto como persona y profesional.

A mis hermanos que son la motivación para mejorar cada día como persona y futuro profesional.

A mi pareja, por su comprensión y dedicación en las situaciones complicadas, brindándome el soporte y motivación necesario en el día a día para culminar la investigación.

Y finalmente a mí, por haberme dado una oportunidad de mejorar a pesar de los pocos recursos para lograr ser profesional y por haber realizado este proyecto.

Jean Franklin Colonia Ballico.

RESUMEN

La calidad nació por la necesidad de mejorar los productos y generar mayores ventas en diversos sectores, el software tampoco fue la excepción aunque su inclusión no fue la mejor y por ende se tuvieron casos donde aviones se estrellaron a causa un error de código, falla en frenos de automóviles o la muerte de muchas personas por radiación a causa de una falta de un control de calidad de software.

La presente tesis Quality Assurance enfocado en Agile Testing para mejorar la calidad de software en la empresa Wasi Innovation, lima - 2022, cuyo propósito fue Determinar si la aplicación de Quality Assurance enfocado en Agile Testing mejora la calidad de software en la empresa Wasi Innovation SAC, Lima – 2022,

Para alcanzarlo se aplicó la metodología bajo un enfoque cuantitativo, de nivel descriptivo, diseño pre experimental, donde se tuvo una población de 20 personas tanto el equipo de trabajo del contexto de investigación como representantes por parte del cliente en el producto desarrollado es ese momento, teniendo una muestra probabilística y censal que estuvo conformada por el total de la población, los cuales respondieron cuestionarios pre y post test de manera individual para determinar el nivel de la calidad de software.

Los resultados obtenidos evidenciaron que previo a la aplicación de quality assurance enfocado en agile testing, un 35% estaba poco satisfecho, un 45% moderadamente satisfecho y un 20% satisfecho y después de la aplicación, un 65% moderadamente satisfecho y un 35% satisfecho con la calidad de software.

Por tanto base a la recolección de datos y los resultados obtenidos, se concluye que quality assurance enfocado en agile testing mejora la calidad de software, así lo corrobora la diferencia de medias de pre y post test, y también la prueba estadística T de Student con una significancia $p < 0.05$.

Palabras clave: Quality Assurance, Agile Testing, Scrum, ISTQB, Calidad de Software, ISO/IEC 25010, Pruebas de Software.

ABSTRACT

Quality was born out of the need to improve products and generate higher sales in various sectors, software was not the exception although its inclusion was not the best and therefore there were cases where planes crashed because of a code error, failure in car brakes or the death of many people by radiation due to a lack of software quality control.

The present thesis Quality Assurance focused on Agile Testing to improve software quality in the company Wasi Innovation, Lima - 2022, whose purpose was to determine if the application of Quality Assurance focused on Agile Testing improves software quality in the company Wasi Innovation SAC, Lima - 2022;

To achieve this, the methodology was applied under a quantitative approach, descriptive level, pre-experimental design, with a population of 20 people, both the work team of the research context and representatives of the customer in the product developed at that time, having a probabilistic and census sample that was made up of the total population, which answered questionnaires pre and posttest individually to determine the level of software quality.

The results obtained showed that prior to the application of quality assurance focused on agile testing, 35% were not very satisfied, 45% moderately satisfied and 20% satisfied, and after the application, 65% were moderately satisfied and 35% satisfied with the software quality.

Therefore, based on the data collection and the results obtained, it is concluded that quality assurance focused on agile testing improves software quality, as corroborated by the difference in pre- and post-test means, and also by the Student's t-test with a significance of $p < 0.05$.

Keywords: Quality Assurance, Agile Testing, Scrum, ISTQB, Software Quality, ISO/IEC 25010, Software Testing.

ÍNDICE

DEDICATORIA	ii
AGRADECIMIENTO	iii
RESUMEN	iv
ABSTRACT.....	v
ÍNDICE.....	vi
I. INTRODUCCIÓN	10
1.1 Planteamiento del problema.....	10
1.2 Formulación del problema	13
1.2.1 Problema general.....	13
1.2.2 Problemas específicos	13
1.3 Objetivos de la investigación	14
1.3.1 Objetivo general.....	14
1.3.2 Objetivos específicos	14
1.4 Justificación de la investigación	14
1.4.1 Justificación económica	14
1.4.2 Justificación Operativa.....	15
1.4.3 Justificación Tecnológica.....	15
1.4.4 Justificación legal.....	16
1.4.5 Justificación Social	16
II. MARCO TEÓRICO.....	17
2.1 Antecedentes de la investigación	17
2.1.1 Antecedentes Internacionales.....	17
2.1.2 Antecedentes Nacionales	22
2.1.3 Antecedentes Locales.....	26
2.2 Bases teóricas.....	26
2.2.1 Quality Assurance.....	26
2.2.2 Agile Testing.....	27

2.2.2.1	Agile Manifiesto.....	27
2.2.2.2	Testing.....	27
2.2.3	Calidad	29
2.2.3.1	Calidad de Software	29
2.2.4	Modelos de calidad	30
	Modelos a nivel de proceso:.....	30
	Modelos a nivel de Producto.....	31
2.2.5	Modelos de desarrollo de Software.....	32
2.2.5.1	Proceso de Software.....	32
2.2.6	International Software Testing Qualifications Board (ISTQB)	36
2.2.6.1	Aseguramiento de la calidad y las pruebas	37
2.2.6.2	Los siete principios de pruebas	38
2.2.6.3	Proceso de pruebas.....	39
2.2.6.4	Desarrollo de Software y pruebas de Software	42
2.2.6.5	Niveles de pruebas	43
2.2.6.6	Tipos de Pruebas	46
2.2.7	SCRUM.....	48
2.2.7.1	Roles	49
2.2.7.2	Eventos.....	50
2.2.7.3	Artefactos	51
2.2.7.4	Historias de Usuario.....	52
2.2.8	ISO/IEC 25010.....	53
2.3	Definición de términos.....	56
2.4	Hipótesis	61
2.4.1	Hipótesis general.....	61
2.4.2	Hipótesis específicas	61
2.5	Variables	61
2.5.1	Variable Independiente	61
2.5.2	Variable dependiente.....	62

2.5.3	Operacionalización de variables	63
III.	METODOLOGÍA	64
3.1	Tipo de estudio.....	64
3.2	El diseño de investigación.....	64
3.3	Descripción de la unidad de análisis, población y muestra.....	65
3.3.1	Unidad de análisis	65
3.3.2	Población.....	65
3.3.3	Muestra	65
3.4	Técnicas e instrumentos de recolección de datos.....	66
3.4.1	Técnicas	66
3.4.2	Instrumentos.....	66
3.5	Técnicas de análisis y prueba de hipótesis.....	67
3.5.1	Técnicas de análisis.....	70
3.5.2	Prueba de hipótesis.....	71
IV.	RESULTADOS DE LA INVESTIGACIÓN	72
4.1	Descripción del trabajo de campo	72
4.1.1	ANÁLISIS DE LA SITUACIÓN ACTUAL – PRE TEST	73
4.1.1.1	Identificación de Requerimientos	73
4.1.1.2	Pruebas de software	89
4.1.2	ANÁLISIS DE LA SITUACIÓN ACTUAL – POST TEST.....	104
4.1.2.1	Planificación de pruebas	107
4.1.2.2	Monitoreo y Control de pruebas	115
4.1.2.3	Análisis de pruebas	117
4.1.2.4	Diseño de pruebas	120
4.1.2.5	Implementación de pruebas	132
4.1.2.6	Ejecución de pruebas	143
4.1.2.7	Compleción de pruebas.....	151
4.2	Presentación de resultados y prueba de hipótesis	156
4.2.1	Presentación de resultados	156

4.2.2	Prueba de hipótesis.....	181
4.3	Discusión de resultados.....	184
V.	CONCLUSIONES.....	188
VI.	RECOMENDACIONES.....	190
VII.	REFERENCIAS BIBLIOGRÁFICAS.....	191
VIII.	ANEXOS	198



I. INTRODUCCIÓN

1.1 Planteamiento del problema

La calidad en general se remonta desde hace muchos años atrás con el objetivo de dar realce a los productos que ofrece en contraste con lo ofertado por la competencia y por consecuencia aumentar sus ventas. Algunos casos se dieron durante la segunda guerra mundial donde se sustituyeron procesos manuales por los mecanizados, donde surgen la aparición de los inspectores de calidad, estos puedan evaluar los productos y reportar las fallas o defectos que se pudieran presentar para tomar decisiones y eliminarlos, teniendo en cuenta los requisitos estipulados previamente y las expectativas de los clientes. Para los años 80's la calidad ya tenía un perfil más estratégico y considerado como una ventaja competitiva y en la actualidad se tiene mayor énfasis en el cliente y cuanto está dispuesto a adquirir con respecto a una calidad perceptible a diferencia de otras (Visual México, 2022).

Especificando la calidad en el sector del software, un hito importante se da con la crisis del software que fue mencionada por primera vez en la primera conferencia elaborada por la Organización del Tratado del Atlántico Norte donde Friedrich L. Bauer, el cual se refirió a los problemas que emergieron en los inicios del software y su experimentación dados por errores en planificación, la estimación de los costos, calidad de software, productividad, entre otros. Algunos de los proyectos de software realizados y que fallaron según (Wikipedia, 2022) fueron:

- El año de 1986 un avión de combate F-18 se estrelló debido a un error de código donde no se consideró uno escenario dentro de la condicional “if then” ignorando la expresión “else” porque creían que era innecesario.
- Entre 1985 y 1987, la muerte de personas en varios hospitales de los Estados Unidos de América y Canadá a causa de la maquina Therac-25 que transmitía altos niveles de radiación porque no tenían un debido control de la calidad de software.
- En el año de 1988, Bank of América invirtió alrededor de 23 millones de dólares en un software denominado Masternet para aspectos contables y reportes de fideicomisos pero para que pudiese funcionar se tuvo que añadir 60 millones de dólares más y al final este software fue cancelado.

Otro reciente caso fue el de la empresa Toyota que en el año 2010 admite fallas en el freno del modelo Prius debido a un problema de software donde había un retraso en

el sistema de antibloqueo de los frenos, lo cual generó un coste de 3 billones de dólares (BCC News Mundo, 2010).

El software forma parte de la tecnología ya que es la que está a cargo de controlar los diversos procesos mediante algoritmos y es tan importante porque se ha adherido a nuestra vidas, tanto así como el comercio, cultura, actividades cotidianas, entre otros (Pressman, 2010) . Consecuentemente de su aplicación en diversos sectores, ésta permite atender a los pacientes de diversas clínicas, evitar accidentes de tráfico, manejar aviones, sistemas de transporte, centrales de energía, acueductos, realizar operaciones bancarias, comprar productos y servicios, entre otros realizar cualquier tipo de actividad en su diario vivir (Villadiego, 2020).

Por tal importancia se debe tener un mayor foco en cómo se desarrolla y entrega el producto de software desde una perspectiva de Software Factory Enterprise (sin importar el tamaño de la empresa), y desde la perspectiva del usuario/clientes obtener el mejor producto y/o servicio del software correspondiente que esté construido en base a las especificaciones requeridas y que cumpla al menos un mínimo nivel de calidad de software.

En el Perú existen alrededor de 400 empresas, de las cuales más del 50% son micro empresas, también algunas compañías más grandes que se dedican a la industria del software y servicio de información, las cuales brindan específicamente el servicio de desarrollo de software a medida, algunas de estas soluciones son plataformas e-learning, comercio electrónico, software para realidad aumentada, sistemas para hospitales, farmacéutica, laboratorios, consultorios, seguridad informática, autoconsultoría, inteligencia artificial para comunicación y asistencia a los clientes, bots, asistentes virtuales, entre otros y los principales sectores son las empresas locales de servicios como minería, turismo, salud, retail, financieros y administración pública (Departamento de Inteligencia de Mercados, 2021). En una investigación realizada a 80 micros y pequeñas empresas productoras de software se observa un mayor uso de modelos de calidad en el proceso y menor uso para la calidad de producto, aunque muestran el conocimiento de estos factores (Millones & Amable Ciudad, 2018).

El contexto de estudio de la presente investigación es una Mype desarrolladora de software - WASI INNOVATION, con RUC 20606702141 y dirección en Av. Lima Nro. 1059 Dpto. 201 – distrito de San Miguel – Lima – Perú, cuyo inicio de actividades

se dio en el 2020, la cual está a cargo de su gerente general William Castro Grijalva, actualmente tienen un tipo de trabajo remoto, cabe mencionar que aún no se ha implementado un marco de aseguramiento de la calidad que esté enfocado a las pruebas en los diversos proyectos que han realizado haciendo un contraste con el estado del arte para la mitigación de errores, defectos y fallas en el proceso y control de la calidad ya que es un aspecto que se está implementando en dicha empresa y en la actualidad cuentan con la aplicación de pruebas manuales según las actividades del desarrollo de software. Algunos de los servicios que brindan son:

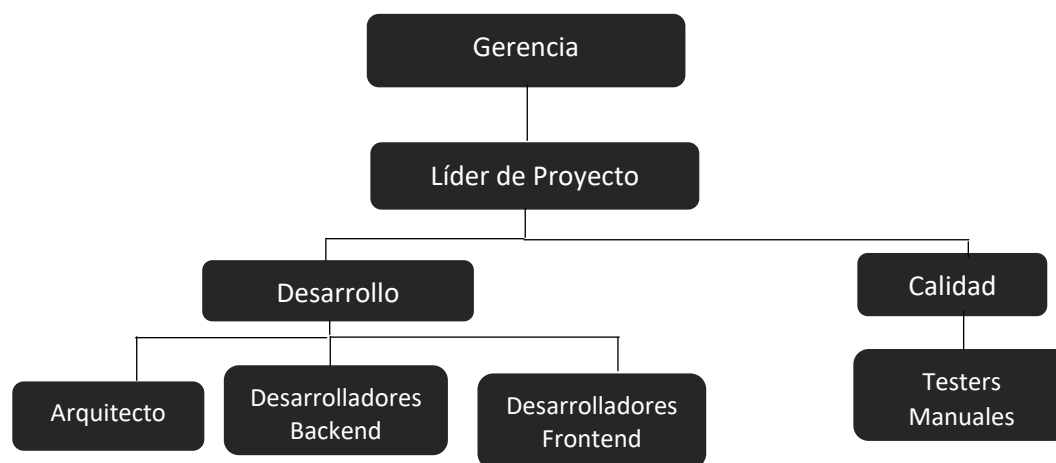
- Desarrollo de Aplicaciones Web y Windows: El diseño y desarrollo de aplicaciones a medida en las plataformas .NET y Java, usando las mejores prácticas y guías estándares. Las herramientas que incluyen pero no están limitadas son MS ASP .NET MVC 4.0, WebAPI, C#, MS SQL Server 2008 R2, Oracle, JSON, OData, JQuery, Javascript, HTML5. CSS3, Angular, KnockOut.js, Java, JavaRichFaces, Spring, Hibernate, Ibatis.
- Desarrollo de aplicaciones Móviles, aplicaciones con un diseño personalizado, exclusivo, intuitivo y amigable para los usuarios de dispositivos móviles como Iphone, Android y Windows Phone.
- Software de propósito especial, soluciones especializadas de minería de datos, es estadística, generación de código, inteligencia artificial o alguna otra de mayor complejidad requerida de investigación especializada o no dispone de recursos inhouse, con el soporte a este tipo de soluciones integrando los datos, análisis predictivo, pronósticos y optimización dentro del contexto de negocio para mejorar y optimizar las decisiones.

Algunos de los proyectos realizados:

- Servicio para la implementación de nuevas funcionalidades en el sistema informático servicio de desarrollo de interfaz para el registro de línea base de monitoreo de los instrumentos de gestión ambiental (IGA), código 01969-2020S y fecha 03/11/2020.
- Servicio de análisis, desarrollo e implementación de sistema en web de las mejoras del sistema de registro de información normativa, código 02159-2020-S y fecha 25/11/2020.

- Diseño e implementación de herramientas informática, servicio para la implementación de requerimientos al módulo de resultados de monitoreo de administrados, código 00658-2021-S y fecha 18/03/2021.
- Servicio para la implementación de nuevas funcionalidades en el sistema informático, código 01160-2021-S y fecha 24/06/2021.
- Servicio de mejoramiento y rediseño de aplicativo informático contratación del servicio de implementación de mejoras al módulo de resultados de monitoreo de administrados, código 01732-2021-S y fecha 14/10/2021.

*Figura 1:
Organigrama Wasi Innovation SAC*



Nota: Organización de la empresa Wasi Innovation SAC en el año 2022, según rol y las áreas correspondientes.

1.2 Formulación del problema

1.2.1 Problema general

¿De qué manera la aplicación de Quality Assurance enfocado en Agile Testing mejora la Calidad de Software en la empresa Wasi Innovation, Lima – 2022?

1.2.2 Problemas específicos

- ¿De qué manera influye la identificación de la calidad de software antes de la aplicación de Quality Assurance enfocado en Agile Testing en la empresa Wasi Innovation SAC, Lima – 2022?
- ¿De qué manera influye la identificación de la calidad de software después de la aplicación de Quality Assurance enfocado en Agile Testing en la empresa Wasi Innovation SAC, Lima – 2022?

1.3 Objetivos de la investigación

1.3.1 Objetivo general

Determinar si la aplicación de Quality Assurance enfocado en Agile Testing mejora la calidad de software en la empresa Wasi Innovation SAC, Lima – 2022.

1.3.2 Objetivos específicos

- Identificar la realidad de la calidad de software antes de la aplicación de Quality Assurance enfocado en Agile Testing en la empresa Wasi Innovation SAC, Lima – 2022.
- Identificar la realidad de la calidad de software después de la aplicación de Quality Assurance enfocado en Agile Testing en la empresa Wasi Innovation SAC, Lima – 2022.

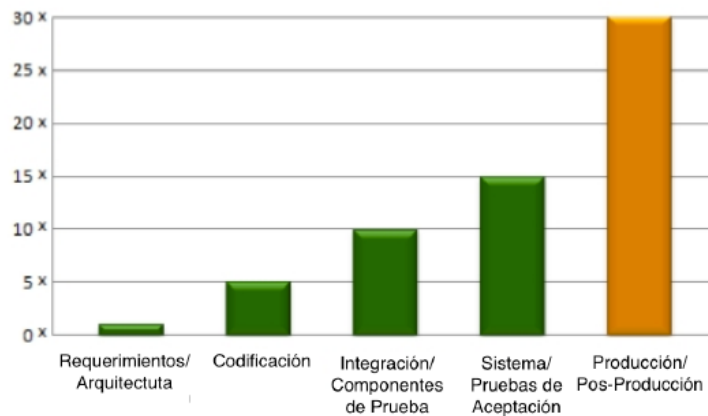
1.4 Justificación de la investigación

1.4.1 Justificación económica

La implementación de quality assurance enfocado en Agile Testing pretende minimizar el costo en el desarrollo y entrega del producto ya que cuanto antes se pueda introducir mediante pruebas u otros genera mayor beneficio porque evita realizar las actividades y procesos nuevamente para reducir o eliminar los errores, defectos y fallas (Hoyos, 2022).

Mientras más tarde se encuentren los errores, defectos o fallas en el desarrollo del software se generará un mayor costo para poder solucionarlo y es en esa medida que el quality assurance (aseguramiento de la calidad / QA) en el proceso y control enfocado en Agile Testing reduce los posibles futuros costos y gastos que hubiesen los cuales se transformarían en cuantiosas pérdidas (Marulanda López, 2014).

*Figura 2:
Costos de corrección de errores en el software*



Nota: En la figura se representa la corrección de los errores detectados en las diversas etapas del software – Desarrollo y producción. Tomado de *Aseguramiento de la calidad en el diseño del software* (p. 25) por Marulanda López, 2014, Universidad EAFIT.

1.4.2 Justificación Operativa

La implementación de Quality Assurance enfocado en Agile Testing otorgará un mayor aporte hacia la calidad del software que se está desarrollando o entregando ya que mediante el proceso de pruebas basado en ISTQB y el marco de trabajo ágil SCRUM se podrá mejorar la calidad y reducir los errores, defectos y fallas que pudieran existir, teniendo una mayor exactitud en los requerimientos establecidos por el cliente y/o usuario ya que al tener un enfoque en el proceso y el control mediante las pruebas en base al ciclo de vida del desarrollo del software (ISTQB, 2018).

1.4.3 Justificación Tecnológica

Desde hace muchos años el desarrollo de software era realizado pero el enfoque y la aplicación de quality assurance (aseguramiento de la calidad) lo cual incluye el proceso del desarrollo y el control de calidad del producto mediante las pruebas es relativamente nuevo y poco aplicado dentro de las empresas desarrolladoras de software dentro del Perú, al igual que nuestro contexto de estudio que es la empresa Wasi Innovation, ISTQB tiene bases del Agile y este tipo de marco de trabajo en el desarrollo y las pruebas se ha implementado dentro de muchas empresas internacionales debido a la flexibilidad al cambio, las mejoras continuas en los entregables del producto desarrollado, el trabajo en equipo y la interacción con los stakeholders internos

y externos, lo que beneficia en gran medida para obtener una mejor calidad de acuerdo a los requerimientos y estándares de calidad ISO, entre otros (ISTQB, 2018).

1.4.4 Justificación legal

Dentro del marco legal del Decreto Supremo N° 066-2011-PCM, el cual garantiza las mejores oportunidades para la apropiación y oportunidades de uso de las Tecnologías de la Información y Comunicaciones, los cuales aseguren la inclusión de todas las personas, el acceso a los servicios sociales que permitan el pleno ejercicio de la ciudadanía.

La Norma Técnica Peruana NTP ISO/IEC 12207:2006 la cual menciona al aseguramiento de calidad dentro de los procesos del ciclo del desarrollo de software como el aseguramiento del producto, proceso, entre otros (Comisión de Reglamentos Técnicos y Comerciales - INDECOPI, 2006).

1.4.5 Justificación Social

Con la implementación de Quality Assurance enfocado a las pruebas las empresas y en específico Wasi Innovation podrá brindar un software con mayores valores agregados aportando a la calidad del software y repercutirá en el usuario final que es la sociedad en general según el sector interesado, obteniendo un mejor producto que tenga un comportamiento según los requerimientos y haya menos posibilidades de errores, defectos y fallas (ISTQB, 2018).

II. MARCO TEÓRICO

2.1 Antecedentes de la investigación

2.1.1 Antecedentes Internacionales

- a) (Kim Nguyen, 2022) en su tesis de maestría *Quality Assurance performance on Medical Technology*, California State Polytechnic University, California – Estados Unidos. Teniendo como **objetivo** proporcionar el impacto significativo en el aseguramiento y control de la calidad en tecnología médica. No se menciona el tipo ni diseño de investigación. Donde aborda sobre la implementación de la tecnología en el campo de la medicina, describiendo la gran importancia que tiene en la vida humana. El aseguramiento y el control de la calidad contribuirá significativamente al éxito del campo de la medicina. Teniendo como **Población** a los profesionales ingenieros y desarrolladores del control de la calidad (no especifica cantidad total) y una **Muestra** aproximada de 40 profesionales. Con los siguientes **Resultados** que se tuvo un 52% de las personas estaba trabajando en el sector médico; la precisión del equipamiento médico tuvo un 23% que si lo eran y se resalta la falta de la garantía de la calidad y la práctica del control en las empresas; un total del 62.5% confirmó que en sus compañías existía el departamento de pruebas de los productos con ingenieros de quality assurance; sobre la técnica y/o metodología usada para el Quality Assurance se tiene que el 69.2% realiza Manual Testing, el 73.1% realiza Pruebas Automatizadas, el 23.1% pruebas de regresión, el 26.9% pruebas unitarias, el 34.6% pruebas de integración, el 19.2% usa un modelo ágil, un 53.8% utiliza el modelo Cascada; Las tecnologías y herramientas más usadas fueron Selenium con 48.1%, Pruebas de front-end con Java, C++ y otros con un 29.6%, Pruebas de Back-end en Assembly, MIPs y otros con un 51.9%, Java y Python con 3.7%; las pruebas más importantes fueron Pruebas manuales con 10.4%, pruebas automatizadas con un 31.3% y Pruebas manuales con Pruebas automatizadas en conjunto con un 58.3%. **Concluyendo** sobre la importancia que tiene garantizar la calidad de la solución en todas las plataformas del mundo, con los impactos de la tecnología informática en el campo médico y la participación del control y garantía de calidad, además

la necesidad de garantía y control de calidad en la tecnología médica actual, aprender y tener el conocimiento de toda la tecnología, así como el método de prueba, ayudará a llevar la escala de precisión de lo posible.

Correlación: La inclusión de las diversas herramientas tecnológicas como lenguajes de programación, otros, y tipos de pruebas de software como base para ejercer una mejor garantía y control de la calidad de software que fueron guía para elaborar el modelo tanto teórico y práctico.

- b) (Lönnroth, 2021) en su tesis de investigación *Improving quality control in automation projects using simulation systems*, KTH Royal Institute of Technology, Estocolmo – Suecia. Donde tuvo como **objetivo** mejorar el control de la calidad para reducir riesgos y tener un mayor aseguramiento para que no se descubran mayores errores en el ciclo de vida, con **metodología de investigación** con enfoque Cualitativo, Explorativo-Inductivo. Trata aspectos del control de la calidad como medida de la evolución de los sistemas automatizados y la complejidad que han adquirido durante el tiempo, combinando la teoría del control de la calidad los métodos de pruebas en el desarrollo de software.. Tuvo como **Población** a todo el personal de la empresa Midro Automation y como **Muestra** al personal que trabaja en desarrollo de sistemas de automatización (no se especifica la cantidad en ningún caso). Con los siguientes **Resultados**, de acuerdo a los posibles tipos errores como Errores de Cuidado, Compatibilidad, Documentos de referencia, problemas de hardware, Malentendidos en la Comunicación, Errores de señal, se tiene que 4 de estos errores pueden sobrevivir durante todo el ciclo de vida del proyecto; también la mayor cantidad de errores puede ocurrir durante la fase de ingeniería y la mayor cantidad de eliminación de errores ocurre durante toda la fase de puesta en marcha y que estos tipos de errores pueden sobrevivir a la puesta en marcha. **Concluyendo** que el aumento de la capacidad de pruebas permitirá mejorar el proceso del control de calidad y que las simulaciones pueden ser herramientas efectivas para poder mitigar los errores que surgen en el desarrollo, especialmente las pruebas funcionales. La combinación de medidas estáticas como dinámicas pueden ser efectivas

para encontrar los errores, si se aplica un mejor control de calidad se ahorrarían altos costos por la reelaboración, en cuanto más viva el error mayor será su costo de corregirlo y el costo del retrabajo de un error crece exponencialmente a medida que avanza un proyecto de desarrollo.

Correlación: La inclusión tanto de pruebas estáticas como dinámicas para obtener un mejor grado de cobertura y control de la calidad de software.

- c) (Abdullahi Wakili et al. 2020) en su investigación *Quality Assurance practices in Agile Methodology*, en el evento International Journal of Advance Research in Science and Engineering, Osmania University, Hyderabad – India. La investigación tuvo como principal **objetivo** analizar las prácticas de aseguramiento de la calidad en metodologías ágiles. No se menciona el tipo ni diseño de investigación. Abordan aspectos de mejora para la calidad de software debido al crecimiento y la complejidad que adquieren y aplicar las métricas de software al desarrollo y a un producto de software es una tarea crítica y lo suficientemente crucial para así aportar conocimiento, llamando a dicha métrica como el aseguramiento de la calidad “quality assurance” como un factor clave para el éxito de cada proyecto de ingeniería de software. También señalan la importancia de la Metodología Ágil como un marco ya que permite el desarrollo con requisitos muy limitados y que admite cambios rápidos en los requisitos y generar un producto rápidamente pero esto no garantizaría calidad sin que se apliquen actividades de software quality assurance al proceso.
- Concluyendo** que aunque algunas metodologías ágiles no son recientes pero se han vuelto muy populares en la industria, los desarrolladores deben saber cómo adaptarse a los nuevos cambios en base a un entorno ágil para poder alcanzar los más altos niveles de calidad que se requieren. La mayoría de las prácticas de métodos ágiles tienen habilidades de quality assurance, algunas dentro de la fase de desarrollo y algunos otros se pueden separar como prácticas de apoyo. Además es difícil comparar la calidad del software resultante del modelo Waterfall (cascada) con metodologías ágiles por las mismas condiciones iniciales de desarrollo, especialmente el costo ya que no se pueden comprar.

Correlación: La inclusión de un marco perteneciente a una metodología ágil por su flexibilidad al cambio y valor agregado de quality assurance para mejorar la calidad de software.

- d) (Thörn, 2020) en su tesis de maestría *Test framework quality assurance: augmenting agile processes with safety standards*, Mälardalen University, Västerås – Suecia. Teniendo como principal **Objetivo** adherir las pruebas en el proceso de desarrollo para mejorar la calidad. La **metodología** de su tesis se basa en el modelo del enfoque de la investigación y de transferencia de la tecnología, lo cual significa el aporte de nuevos conocimientos tanto teóricos y prácticos. Aborda la incrustación de la calidad en los sistemas y que son demostradas mediante las pruebas. Las pruebas automatizadas son importantes en el desarrollo ágil y el marco de trabajo para las pruebas es considerado crítico e importante para el aseguramiento de la calidad. También se aborda aspectos sobre cómo las industrias con procesos ágiles pueden aprender del desarrollo basado en planes para aumentar la calidad del marco de pruebas. La **Población** estuvo conformada por integrantes de la empresa Westermo Network Technologies AB3 (Westermo) y una empresa socia y la **Muestra** estuvo conformada por seis participantes con competencias, antecedentes comunes (un gerente de equipo de pruebas de software, un gerente del equipo de WeOS, tres desarrolladores del equipo de pruebas y un desarrollador del equipo de WeOS). Con los siguientes **Resultados**, se percibieron riesgos relacionados al marco de las pruebas de software como la calidad deficiente en el marco de pruebas del software que puede conducir a enmascarar problemas de detección, la retroalimentación de los casos de prueba fallidos que no podrían llegar a las partes interesadas. **Concluyendo** que las pruebas son parte fundamental del proceso de desarrollo y la actividad más utilizada para el aseguramiento de la calidad, siendo la automatización de pruebas un punto importante en el desarrollo ágil. Agregando que la calidad de los sistemas integrados, tanto en lo que respecta al software como hardware, a menudo se demuestra mediante los resultados de las pruebas producidas y que estas decisiones tomadas durante el desarrollo se basan en la confianza en los resultados de

esas pruebas, lo que impone una misión crítica en cualquier herramienta utilizada, además de la distinción de los errores en el software y las pruebas de software con importantes para la confianza, un error no debe afectar las posteriores pruebas que se realicen.

Correlación: Inclusión de tipos de pruebas, en específico pruebas automatizadas como punto importante dentro del desarrollo ágil y generación de confianza mediante resultados de pruebas.

- e) (Pauld'uro, 2018) en su tesis de maestría *Software development quality assurance process in regional e-commerce company*, Masaryk University, Brno – Republica Checa. Donde tuvo como principales **objetivos** en análisis del estado actual de los procesos de aseguramiento de la calidad, proponer un conjunto de soluciones para mejorar procesos, Resumir las propuestas, validar los resultados y evaluarlos. No se menciona el tipo ni diseño de investigación. Donde aborda aspectos importantes como el aseguramiento de la calidad en el software, específicamente en la plataforma e-commerce Notino, donde se hace un contraste de la situación antes y después de implementar un marco para mejorar la calidad de software dentro de sus procesos con los diversos tipos de software y los diferentes estándares y modelos respecto a este ámbito. Con los siguientes **resultados**, se realizaron un total de 3073 revisiones de código, agregando comentarios y posibles defectos, donde se pudo revelar que agregar revisiones de código fue un paso exitoso porque permitió revelar futuros errores y un impacto crítico en la aplicación en su lanzamiento, se agregaron 71 Pruebas Unitarias al módulo e-shop y 27 a la API, **concluyendo** que la evaluación de los cambios mostró que la propuesta fue exitosa aunque difícil de medir porque aún existía un margen de mejora después de haber realizado todas las actividades para cumplir con los objetivos descritos en su investigación.

Correlación: Inclusión de pruebas de software como revisiones de código, APIs, entre otros como estrategia de evaluación de la calidad de software y mejora.

2.1.2 Antecedentes Nacionales

a) (Condori Fernández, 2022) en su tesis de grado *Pruebas de software en aplicativos de canales digitales de una entidad bancaria usando ISTQB*, Universidad Nacional Mayor de San Marcos, Lima – Perú. Tuvo como principal **objetivo** realizar las pruebas de software en aplicativos de canales digitales de una entidad bancaria usando la metodología de calidad ISTQB. No se menciona el tipo ni diseño de investigación. Aborda temas como la certificación y pruebas de calidad de software en los aplicativos del área de canales digitales, teniendo como finalidad la implementación de los cambios en el tiempo propuesto, llevando a puesta al ambiente de producción para el uso de nuevas funcionalidades como las operaciones en línea y reducir en la medida de lo posible la atención en ventanilla que genera problemas en tema de tiempos y nivel de servicio al cliente. Las pruebas se realizaron en los aplicativos Web-Banca por Internet y App Mobile de la Entidad bancaria. **Concluyendo** que se pudo identificar los posibles riesgos que podrían ocurrir en la etapa de las pruebas, analizar los requerimientos estimando fechas de ejecución, elaborando los diseños de casos de pruebas usando TestLink como herramienta, se ejecutaron las pruebas establecidas en el análisis y diseño donde se culminó con éxito en ejecución tanto en ambiente de pruebas y las de aceptación según el usuario final, donde no se encontraron defectos, al contrario fueron mejoras en el proceso de envío de las operaciones. Al final pasándose a producción donde inicialmente tuvieron una marcha blanca entre empleados obteniendo unos resultados satisfactorios.

Correlación: Inclusión fases de pruebas de software como análisis y ejecución en base a ISTQB, uso de aplicativos web como estrategia de gestión de casos de pruebas.

b) (Villanueva Ajito, 2021) en su tesis de grado *Evaluación de la calidad de uso del software ecommerce de la empresa Mauripro Sailing según la ISO/IEC 25000*, Lima, 2020, Universidad Nacional de Trujillo, Trujillo – Perú. Se tuvo como **Objetivo** principal evaluar la calidad de uso del software ecommerce de la empresa Mauripro Sailing SAC. Con **tipo y**

diseño de investigación cualitativa - descriptiva simple. Teniendo como **Población** a un total de 10 empleados que usan el software ecommerce para la manipulación de los datos dentro de la empresa Mauripro Sailing SAC, con **Muestra** no probabilística donde la cantidad es la misma que la población. Donde se refiere a la medición del software ecommerce de la empresa Mauripro Sailing según la ISO/25000, después de describir la situación de la realidad que se afronta, se realizó el análisis y la presentación de los hallazgos más relevantes donde tuvo en cuenta un método para su evaluación las características y subcaracterísticas de la ISO/IEC 25010 según el instrumento de la autora Calabrese basado en el método objetivo-pregunta-métrica. Los **Resultados** se basaron en el enfoque GQM basado en el modelo de calidad de la Norma ISO/IEC 25001, según la norma ISO/IEC 25000, según la característica de Usabilidad se tuvo un 100% en estética, un 63% en protección frente a errores de usuario, un 90% en aprendizaje, un 33% en accesibilidad, un 33% en inteligibilidad; en la característica de Seguridad se tuvo un 50% en confiabilidad, 100% en integridad, 75% en no repudio, 50% en responsabilidad, 50% en autenticidad; en la subcaracterística de Portabilidad según el dispositivo se tuvo que web responsive tuvo un 100%, aplicación móvil un 0%, adaptación de hardware un 100% y la adaptación del software un 100%. **Concluyendo** que el software ecommerce de la empresa Mauripro Sailing SAC tiene una alta calidad aunque no llegue al 100% satisface las necesidades del cliente, además que cuenta con interfaz agradable que permite a los usuarios la gestión adecuada y brinda todas las funcionalidades que la empresa requiere para su funcionamiento aceptable. También que el nivel de usabilidad y portabilidad son aceptables y el nivel de seguridad supera las expectativas, todo esto en base a la ISO/IEC 25000. **Correlación:** Inclusión estrategia de las características y subcaracterísticas de la ISO/IEC 25010 como referencia de métricas para la evaluación de la calidad de software.

c) (Melgar Velásquez, 2020) en su tesis de grado *Implementación de un modelo de gestión de pruebas de software según ISTQB para mejorar el proceso del área de certificación en tecnologías web de una entidad financiera*, Universidad Tecnológica del Perú, Lima – Perú. La investigación tuvo como principal **Objetivo** la implementación de un modelo de gestión de pruebas de software basado en ISTQB para mejorar el proceso del área de certificación en tecnologías web de una entidad financiera. No se menciona tipo ni diseño de investigación. Se tuvo como **Población** a todas las entidades financieras y como **Muestra** a una de dichas entidades (se reservó el nombre por políticas de la organización). Aborda sobre la implementación de las buenas prácticas de ISTQB para mejorar la calidad de software mediante los diversos tipos de pruebas que se realizaron en el ambiente de pruebas para luego realizarlo en el ambiente de desarrollo y evitar encontrar mayores fallas, el trabajo está enfocado en las pruebas funcionales ya que es ahí donde el usuario final puede realizar dicha manipulación. Teniendo como **Resultados** que se seleccionaron una serie de herramientas usadas en el sector y fueron probadas con los líderes usuarios que gestionan las pruebas en el área, de las cuales se seleccionaron a HP ALM/Quality Center porque proporciona el seguimiento de las tareas relacionadas con el proyecto, IBM Rational ClearQuest ya que permite tener un mayor alcance del requisito enfocado al cambio, IBM Rational ClearCase permite la gestión de las pruebas y Jira para la gestión de las pruebas, especialmente en un proyecto de calidad para la generación de incidentes detectados en la fase de las pruebas donde se puede anexar las evidencias del caso; Se establecieron modelos para el procedimiento de las actividades; para la estimación de los requerimientos; análisis y ratificación de los requerimientos; también se pudieron diseñar los casos de prueba evitando la paradoja del pesticida y se tuvo como herramienta del ALM. **Concluyendo** que al implementar herramientas para la gestión de las pruebas se tiene un mejor control y transparencia en procedimientos en la atención de los requerimientos, además al tener fases de pruebas según ISTQB se obtiene un mejor resultado a partir de los requerimientos solicitados por el usuario de la aplicación web y una estructura base de los

casos de prueba es eficiente para diseñar los escenarios y detectar defectos en el ambiente de pruebas.

Correlación: Inclusión de fases de pruebas según ISTQB para un mejor control y transparencia de los procedimientos, mejores resultados con el uso de gestor de pruebas para el seguimiento de estados.

- d) (Tineo Gamboa & Quispe Cuadros, 2019) en su trabajo de investigación *Testing ágil en aplicaciones móviles android usando espresso*, Universidad Tecnológica del Perú, Lima – Perú. El **Objetivo** de la investigación fue encontrar la consistencia y calidad de la aplicación UNIHHELP PRO mediante el testing ágil con la ayuda de herramientas y la metodología Scrum, con **Tipo de Investigación** descriptiva. Donde describen que realizar testing dentro de un proyecto de desarrollo es esencial, especificando antes y durante el desarrollo, los cuales nos ayudan a reducir la mayor cantidad de errores existentes antes de pasar a un ambiente de producción, los diversos hechos catastróficos confirman las causas de no analizar y testear un producto antes de su lanzamiento y la implementación de las herramientas ágiles son sumamente importantes para garantizar la calidad y consistencia en un producto, también recalcar que el modelo de referencia usado fue Scrum como parte principal del proyecto. En los **Resultados** se describe que el Framework para las pruebas llamado Espresso se pudo implementar sin problemas y por tanto se ejecutaron los casos de prueba relacionados a la validación de la interfaz de usuario y específicamente del módulo de login y la aplicación como todo un sistema. **Concluyendo** que el testing es muy importante antes de lanzar el producto o al momento del desarrollo, también sobre la metodología Scrum que es fundamental como marco de trabajo ágil para la asignación de tareas tanto en el desarrollo y el testing en cada iteración. Finalizando que el testing ágil en los proyectos de software es importante por su aporte a los temas de presupuesto y tiempo.

Correlación: Inclusión del testing ágil como estrategia dentro de un marco de trabajo ágil como Scrum para obtener mejores resultados y aminorar temas de errores en costos, presupuesto y tiempo.

2.1.3 Antecedentes Locales

Después de realizar la revisión bibliográfica correspondiente en los diversos repositorios de las universidades de la localidad, no se encontraron investigaciones relacionadas con el tema y problema de investigación. Por tal motivo no se consiguen investigaciones locales.

2.2 Bases teóricas

2.2.1 Quality Assurance

El nacimiento del Quality Assurance según Carrizo & Alfaro (2018) mencionan que alrededor de 1960 se inicia un movimiento de parte de los consumidores con la necesidad de asegurar los productos que eran comercializados en el mercado para que cumplieran estándares de seguridad según el uso del cliente, a pesar de existir un control de calidad no garantizaba el cumplimiento requerido por parte de los consumidores. A consecuencia de esto nace la innovación del Quality Assurance (Aseguramiento de la calidad), así es como esta filosofía aplicada en las empresas se aplica en el concepto de calidad abarcándola en todas las etapas del ciclo del producto dentro de las organizaciones, algunas de ellas eran el diseño de producto y procesos, la misma producción y las ventas y posventas.

El aseguramiento de la calidad (QA) es una variedad de actividades las cuales se encargan de evaluar y definir las características mínimas y óptimas de los procesos de software generando mayor confianza según los requerimientos establecidos (McGrath, 2014, como se citó en Ibarra Luevano, 2018).

Así mismo Ibarra Luevano (2018) menciona que el aseguramiento de la calidad son prácticas que nos ayudan a evaluar los procesos y productos de software, si estos cumplen con los estándares aplicables y también la adecuación al tipo de producto según los requerimientos que se desea obtener.

Según el Comité Internacional de Certificaciones de Pruebas de Software ISTQB Glossary (2022) son todas las actividades involucradas a brindar una mayor confianza con respecto al cumplimiento de los requisitos de calidad.

2.2.1.1 Software Quality Assurance

Según Carrizo & Alfaro (2018) se refiere a la actividad que se implementa en todo el proceso de software, esto significa desde los

procedimientos, los métodos y las herramientas, las cuales se enfocan en el control de calidad lo que incluye las pruebas de software, aspectos de revisiones técnicas, elaboración de reportes, entre otros, también se recalca el conocimiento del proceso de la ingeniería de software para poder llevar a cabo el aseguramiento de la calidad de software, ya que al conocer estos aspectos será mucho más sencillo mejorar en la calidad.

2.2.2 Agile Testing

Algunas bases para poder entender mejor de lo que es la corriente ágil es el Agile Manifesto, la cual se describe como lo siguiente:

2.2.2.1 Agile Manifesto

Desde sus inicios hasta la actualidad se ha descubierto muchas formas de desarrollar software y se sigue haciendo pero desde la perspectiva del Agile Manifesto nos brinda un mayor enfoque desde la colaboración con el cliente, poder responder al cambio antes de poder seguir lo planificado, se sabe que durante el proceso existen valor hacia la derecha pero también muestra un mayor valor hacia la parte izquierda, donde se encuentran actividades como los requerimientos, el análisis de estos, la interacción con el cliente para poder evaluarlos y continuar con nuestro flujo, entre otros (Crispin & Gregory, 2009).

2.2.2.2 Testing

Es una disciplina dentro de la ingeniería de software, la cual contiene herramientas, procesos y métodos de trabajo con la finalidad de identificar defectos en el software y de esta manera alcanzar cierta estabilidad. También tener en cuenta que el Testing no es una actividad que se realiza al final del desarrollo del software sino que va en paralelo y gracias a este enfoque permite que la construcción vaya correctamente de acuerdo los requerimientos establecidos por el cliente y/o usuario final (Hoyos, 2022).

Proceso que consiste en varias actividades dentro del ciclo de vida de desarrollo del software, tanto en la parte estática como dinámica, tanto en planificar, preparar o evaluar un componente o todo el sistema relacionado al cumplimiento de los requerimientos establecidos y

determinar si son aptos según el propósito correspondiente y detección de defectos (ISTQB, citado en Thörn, 2020).

Objetivo

Uno de los principales objetivos del Testing es aportar calidad al producto dentro del proceso de desarrollo, donde para que un producto llegue a la calidad requerida se debe adherir de acuerdo a sus características según algún estándar como la ISO 25010, que es una normal que está referida a la calidad (Sánchez Peño, 2015).

Según ISTQB (2018) nos menciona que los principales objetivos son:

- La evaluación de los productos, específicamente los requerimientos, si se trabaja en un marco de trabajo ágil las historias de usuario y el código fuente desarrollado o que está en proceso.
- La verificación del cumplimiento de requerimientos establecidos.
- La validación del producto acorde al grado de culminación y funcionalidad esperado por los clientes u otras partes interesadas en el producto.
- Aumentar la confianza en el nivel de la calidad del producto en base a las pruebas.
- La prevención de defectos.
- La detección de los fallos y los defectos.
- Otorgar la mayor información posible a las partes interesadas para la toma de decisiones, específicamente sobre el nivel de la calidad del software el cual es el objeto de prueba.
- Minimizar el nivel de los riesgos como fallas que no han sido detectadas a tiempo y que ocurren en el ambiente de producción.

Entonces ahora con el previo conocimiento sobre qué significa Agile Manifesto y Testing descritos con anterioridad podemos definir Agile Testing que es un enfoque ágil hacia las pruebas, donde el enfoque hacia mejorar continuamente en el trabajo y entregar el mejor producto con la implementación de procesos, métodos y herramientas que nos permitan lograr dicha estabilidad y calidad.

Por tanto la importancia se da en la identificación, prevención y corrección de errores antes que esté en un ambiente de producción, en otras palabras que esté operativo. Así mismo la idea equivocada de que el Testing se realizaba al final, después de tener el código fuente del producto y antes de entregarlo pero en la actualidad el Testing de Software se debe implementar desde el inicio del proceso de desarrollo, y es que por ese motivo nace el ISO 25000 con el motivo de regular los conceptos referentes a los niveles de la calidad de software, dándole importancia al proceso donde cada entregable que forma parte del desarrollo de software y no la entrega final del producto, interviniendo dichos entregables dentro del marco del estándar de calidad ISO 25000, sus características y subcaracterísticas (Hoyos, 2022).

Según Sánchez Peño (2015) los principales propósitos de las pruebas de software son evaluar la calidad y reducir los riesgos de fallas en un determinado software mediante la detección y posterior reparación de dichos defectos en el componente o software que está siendo probado. También se puede incluir la verificación de requerimientos que se hayan especificado y la validación de las funcionalidades, lo que permitirá tomar decisiones según dicha información proveniente de las pruebas para cumplir los requerimientos y estándares reglamentados.

2.2.3 Calidad

Es una serie de características, propiedades inherentes que permiten juzgar el valor de algo en específico, también se describe que es la adecuación de un producto o servicio respecto a los requisitos especificados (Real Academia Española, 2022).

Según Internacional Organization for Standardization (2012) menciona que la calidad es una variedad de características inherentes que cumplen con ciertos requisitos, el termino calidad puede tener adherido ciertos adjetivos tales como pobre, buena o excelente; el termino inherente es la existencia de una característica no temporal.

2.2.3.1 Calidad de Software

De acuerdo a la definición de calidad y con un enfoque más específico hacia el software, la calidad es el nivel de cumplimiento de un producto

de software acorde a los requisitos que se establecieron y el grado de satisfacción de las expectativas de los clientes.

Según Callejas Cuervo et al. (2017) se refiere al grado de desempeño de las características que debe tener un producto durante su desarrollo, donde dichas características garantizan que el usuario final cuente con un sistema con mayor grado de confiabilidad hacia esta, lo cual mejora la satisfacción hacia la eficiencia y funcionalidad del producto realizado.

2.2.4 Modelos de calidad

Los modelos de calidad son la integración de buenas prácticas dentro de uno o varios documentos, donde están dirigidas a varios procesos que van a permitir la medición de dichos avances con respecto a la calidad (Scalone, 2006, citado en Callejas Cuervo et al. 2017).

Según Callejas Cuervo et al. (2017) en el ambiente del desarrollo del software, un modelo de calidad debería permitir la evaluación del sistema correspondiente de forma cualitativa o cuantitativa y consecuentemente con los resultados obtenidos realizar proposición e implementación de estrategias que coadyuven el proceso dentro de las diversas etapas del desarrollo tales como el análisis, diseño, desarrollo y pruebas de software.

Así mismo los autores Callejas Cuervo et al. (2017) clasifican a los modelos en 3 tipos los cuales son Calidad a nivel de proceso, producto y uso.

Modelos a nivel de proceso:

Según Callejas Cuervo et al. (2017) se basa en un enfoque de calidad desde el inicio del proyecto y específicamente en cada etapa un control y seguimiento para reducir riesgos y soporte, sabiendo que si se ignora la verificación de factores en alguna etapa es posible que se presenten riesgos y deficiencias en el proceso y producto final, algunos de los modelos de nivel son:

- **ITIL:** Su objetivo es fortalecer la gestión desde una perspectiva gubernamental teniendo como base 5 elementos: el manejo de aplicaciones e infraestructura, la perspectiva del negocio, la entrega del servicio y el soporte del servicio, todo esto con la finalidad de ofrecer un marco para la prestación de un servicio completo, satisfaciendo necesidades de apoyo,

instalación, redes, hardware, sistemas operativos, hardware y software necesarios (Callejas Cuervo et al. 2017).

- **ISO/IEC 15504:** Enfocado en la evaluación de procesos en pequeñas, medianas empresas mediante sus 6 niveles de madurez los cuales están ordenados en forma ascendente y son: la organización madura, básica, gestionada, establecida, predecible y por último la organización optimizando (Callejas Cuervo et al. 2017).
- **Bootstrap:** Es una metodología la cual permite la mejora de los procesos según 6 actividades: Evaluar la necesidad, inicio del proceso de la mejora, preparar y dirigir la evaluación, analizar los resultados, implantar y finalizar las mejoras (Herrera, 2012, citado en Callejas Cuervo et al. 2017).
- **Dromey:** Evalúa las etapas del proceso dentro del desarrollo como los requisitos, el diseño y la implementación, contenida por ciertas características y subcaracterísticas de calidad (Callejas Cuervo et al. 2017).
- **Personal Software Process (PSP):** Se enfoca en el desarrollo profesional de los ingenieros, teniendo como eje la administración de la calidad de proyectos, reducir los defectos, estimar y planear el trabajo (Vargas, 2010, citado en Callejas Cuervo et al. 2017).
- **Cobit 4.0:** Orientado hacia los negocios y procesos, se basa en controles, teniendo 7 criterios de información que se definen como requerimientos del control del negocio; la efectividad, eficiencia, integridad, efectividad, disponibilidad, cumplimiento y confiabilidad (Guerrero, 2006, citado en Callejas Cuervo et al. 2017).
- **ISO/IEC 20000:** Tiene la finalidad los servicios TI de forma empresarial, estas contando con el nivel de calidad necesario para brindar servicios a los usuarios/clientes. Dividiéndose en especificaciones 20000:2005 y el código de buenas prácticas en la versión 20000-2:2005 (Van Bon, 2008, citado en Callejas Cuervo et al. 2017).

Modelos a nivel de Producto

- **Goal Question Metric:** Permite definir las métricas para medir el avance como el resultado de un proyecto, a medida de la aplicación de preguntas que permiten alcanzar ciertas metas planteadas, teniendo como enfoque las

metas, preguntas y las métricas (Villarroel, 1999, citado en Callejas Cuervo et al. 2017).

- **Boehm:** Modelo tipo incremental, que se divide en tareas y estas en conjuntos de tareas, las cuales se dividen en planear, analizar el riesgo, ingeniería y evaluar (Velazco, 2016, citado en Callejas Cuervo et al. 2017).
- **ISO 9126:** Está basado en el modelo McCall, enfocado para desarrolladores, evaluadores, analistas, otros involucrados en los procesos existentes en el desarrollo del software, tiene 4 segmentos, los cuales son: las métricas tanto internas como externas, el modelo de la calidad, la calidad de métricas, características en torno a la funcionalidad, usabilidad, eficiencia, fiabilidad, portabilidad, mantenibilidad (Ango, 2014, citado en Callejas Cuervo et al. 2017).
- **ISO 25000:** Conocido como SQuaRE, tiene como propósito guiar el desarrollo en base a los requisitos y evaluar los atributos de calidad, como principales características a la adecuación funcional, compatibilidad, fiabilidad, seguridad, portabilidad, mantenibilidad, eficiencia del desempeño (Alfonso, 2012, citado en Callejas Cuervo et al. 2017).

2.2.5 Modelos de desarrollo de Software

Para poder entender mucho mejor un modelo para el desarrollo de software tenemos que tener presente qué es lo que significa un proceso, como se describe en lo siguiente.

2.2.5.1 Proceso de Software

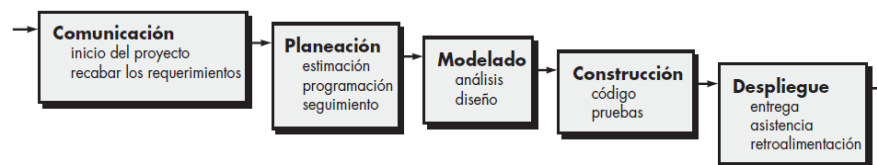
Según Pressman (2010) menciona que es una serie de actividades de trabajo, tareas y acciones las cuales se realiza en un contexto de creación de un determinado producto, donde cada actividad, acción o tarea está regido en base a una estructura o modelo y la relación que tiene con el proceso.

Entonces una estructura general según Pressman (2010) lo define como la comunicación, planeación, modelado, construcción y por último el despliegue y alguno de los modelos según el autor son:

Modelo de la cascada

Según Pressman (2010) el modelo cascado también es conocido por el nombre de “ciclo de vida clásico” que comienza con la especificación de requerimientos de parte de los usuarios/clientes, continuando la planeación, el modelado, la construcción y el despliegue correspondiente. Es el modelo más antiguo referente a la ingeniería de software. En los últimos años, raras veces se han utilizado este tipo de modelos lineales en los proyectos reales, aunque el modelo cascado también acepta repeticiones pero de forma indirecta y lo que causa es confusión en el equipo. También se refiere a algunas ocasiones donde los integrantes de un equipo se sienten confundidos al tener que esperar la finalización de otras etapas.

Figura 3:
Modelo Cascada

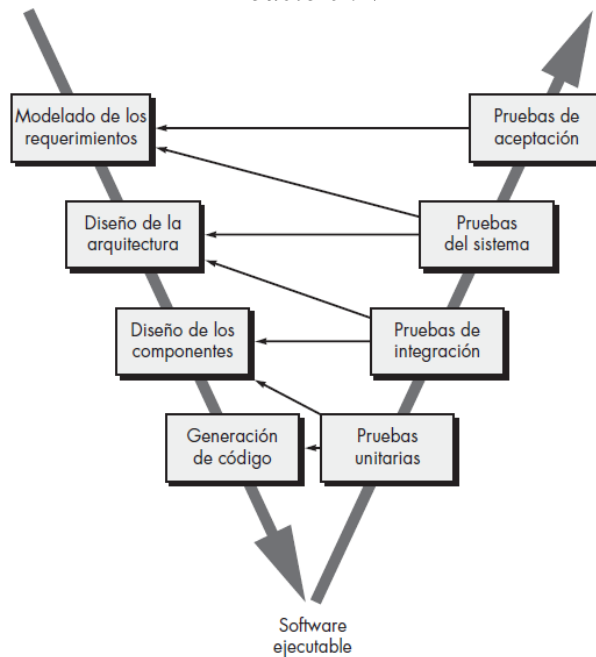


Nota: Modelo Cascada, las fases y los detalles sobre las actividades que se deben realizar en el desarrollo del software. Tomado de *Ingeniería de Software – Un enfoque práctico* (p. 34), por Pressman, 2010, McGraw Hill.

Modelo en V

Surge como una variante al modelo de Cascada, donde cada vez que se avanza hacia abajo ciertas especificaciones como los requerimientos mejoran las representaciones técnicas, por ejemplo en un momento que se haya terminado el código ya se pasa al otro extremo donde se realiza una cantidad de pruebas para poder asegurar la calidad, además es raro su aplicación en proyectos reales y como resultado estos cambios solo causan confusión (Pressman, 2010).

Figura 4:
Modelo en V

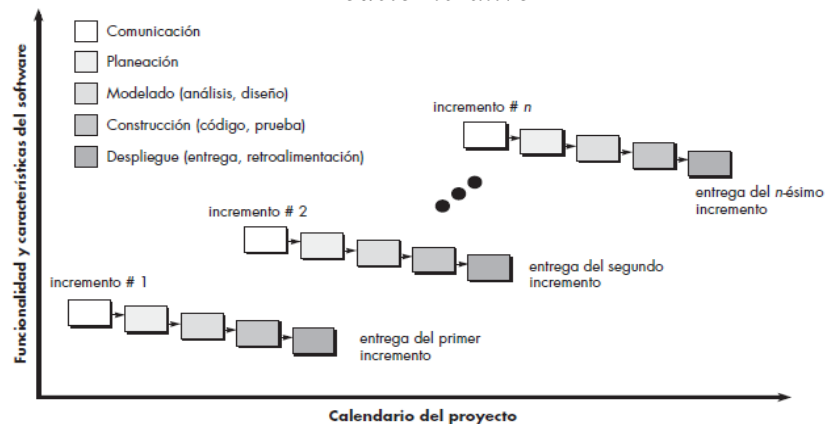


Nota: Modelo en V, las fases y los detalles sobre las actividades que se deben realizar en el desarrollo del software, incluyendo las pruebas con mayor importancia que el Modelo Cascada. Tomado de *Ingeniería de Software – Un enfoque práctico* (p. 35), por Pressman, 2010, McGraw Hill.

Modelos incrementales

Existen circunstancias donde los requerimientos están definidos correctamente pero para el alcance es dificultoso poder establecerlo, entonces si se tuviera una especificación de un requerimiento no habría ningún problema ya que este tipo de modelo es flexible a los cambios y los resultados se van iterando en cada flujo del proceso de software. Cuando se usa este tipo de modelos (Pressman, 2010).

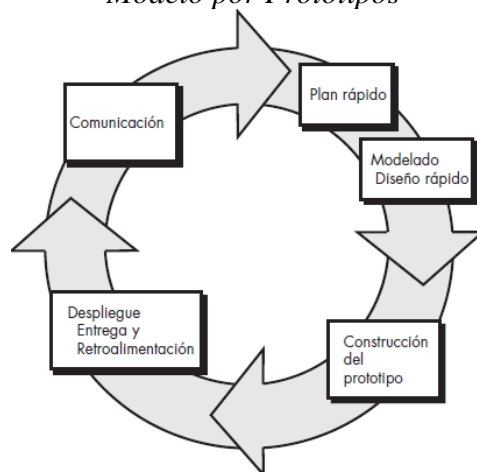
Figura 5:
Modelo Iterativo



Modelo por Prototipos

En circunstancias donde el cliente no tenga claro los requerimientos y el desarrollador tampoco sepa sobre algunos algoritmos y procesos a realizar en base a los requerimientos se tiene en cuenta a este tipo de modelo porque ofrece el mejor enfoque, ya que el paradigma de hacer prototipos ayuda al entendimiento cuando los requerimientos no han sido comprendidos o no están especificados de la mejor manera, siendo un mecanismo para la identificación de requerimientos del producto a desarrollarse (Pressman, 2010).

*Figura 6:
Modelo por Prototipos*

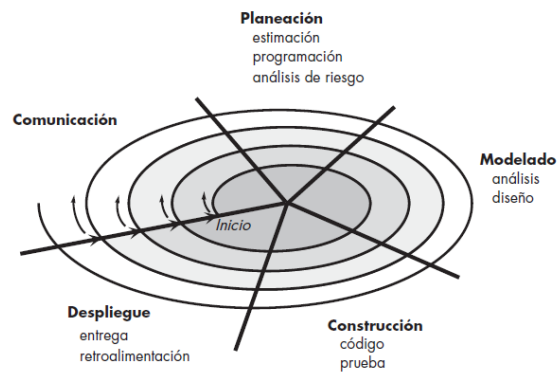


Nota: Modelo por Prototipos con las fases propias del desarrollo de software. Tomado de *Ingeniería de Software – Un enfoque práctico* (p. 37), por Pressman, 2010, McGraw Hill.

Modelo Espiral

Es un modelo enfocado para el desarrollo rápido de versiones cada vez más completas, el cual se caracteriza por ser cíclico abocado a lo incremental y el aseguramiento del compromiso de los stakeholders mediante soluciones óptimas. Este modelo está dividido en base a las fases del desarrollo de software y se recorre en sentido horario partiendo desde el centro (Pressman, 2010).

*Figura 7:
Modelo en Espiral*



Nota: Modelo en Espiral con las sus respectivas fases del desarrollo de software, teniendo un funcionamiento en sentido horario desde el centro. Tomado de *Ingeniería de Software – Un enfoque práctico* (p. 39), por Pressman, 2010, McGraw Hill.

2.2.6 International Software Testing Qualifications Board (ISTQB)

Es el comité Internacional de Certificaciones de Pruebas de Software (ISQTB), organización internacional creada en el año 2002 en Escocia y enfocada en el trabajo voluntario de muchos expertos en el ambiente de las pruebas de software a nivel global. Esta organización suministra un conjunto de programas de estudio y glosarios como base de estándares internacionales por niveles, de acuerdo a estas bases que se tienen como guías para la acreditación y la evaluación con referencia a los profesionales en el ámbito de las pruebas de software (Wikipedia, 2022). Es un esquema de certificación de ámbito global en el campo de las pruebas de software, donde establecer una continua evolución mediante el conocimiento brindado por una gran cantidad de expertos en pruebas de software, lo cual amplia y valida la habilidad en pruebas (ISTQB®, 2022). Como se han mencionado en los conceptos anteriores para el ISQTB el principal foco son las pruebas mediante un marco que presenta para el aseguramiento de la calidad de software desde los requisitos hasta su culminación. Dando importancia a las pruebas y la documentación asociada como aspectos claves para la mitigación de riesgos de fallos que pudieran existir en el ambiente de producción/operación y también las pruebas pueden solucionar algunos errores o imprevistos en base al marco legal o normas específicas sobre el software dentro del país de operación (ISTQB, 2018).

Pruebas como aporte al éxito

Según ISTQB (2018) menciona como las pruebas nos ayudan a llegar al éxito comenzando por la revisión de los requerimientos lo cual podría encontrar el error lo más pronto posible, lo cual reduce la existencia de componentes, funcionalidades incorrectas y que no hayan sido verificadas con anterioridad. El trabajo en conjunto de probadores con los desarrolladores aumenta el entendimiento de los diversos procesos teniendo como base los requerimientos haciendo que los defectos se puedan identificar en una etapa temprana.

2.2.6.1 Aseguramiento de la calidad y las pruebas

El aseguramiento de la calidad se refiere a la observación de cada proceso existente dentro del desarrollo de software lo cual permitirá otorgar mayor confianza hacia los niveles de calidad, ya que cuando estos se llevan a cabo de forma correcta, el producto en base a esos procesos general mayor calidad lo que contribuye a prevenir los defectos. También analizando la causa raíz para detectar y posteriormente eliminar dichos defectos. Y por otra parte las pruebas son parte del control de calidad que concierne a varias actividades y ésta al aseguramiento de la calidad ya que abarca a la ejecución de todo el proceso. Por tanto las pruebas contribuyen al logro de la calidad de diversas maneras (ISTQB, 2018).

Error, defectos y fallas

Según ISTQB (2018) estos conceptos están enlazados y uno repercute en otro, imaginemos un contexto donde una persona comete un error dentro del código y luego se crea un defecto dentro de un componente de software o algún módulo en específico, consecuentemente a este defecto se podría producir una falla dentro del sistema a consecuencia del error en el código y luego la repercusión en un componente o módulo, teniendo en cuenta que los errores pueden ocurrir por diversas situaciones, algunas de estas son:

- Presión del tiempo.
- Falla humana.
- Poca experiencia y/o capacitación.

- Nula o poca comunicación entre los participantes del proyecto desde los requerimientos, el diseño, entre otros.
- La complejidad en los distintos aspectos como el código, infraestructura, arquitectura implantada, entre otros.
- Tecnologías emergentes o poco conocidas por el tiempo de evolución.

Defectos, causas raíz y efectos

La causa raíz es una serie de acciones o condiciones las cuales crearon defectos en ciertos componentes o módulos dentro del ciclo de vida del desarrollo del software, poder identificarlos y eliminarlos ayudará a la mitigación de los defectos existentes y los futuros defectos similares y los efectos que causen estos defectos serán críticos dentro de un módulo o el sistema que se esté elaborando

2.2.6.2 Los siete principios de pruebas

Según ISTQB (2018) menciona que desde el año 1950 han existido diversas pautas pero las más comunes en las pruebas son las siguientes:

- **Las pruebas muestran presencia de defectos, no ausencia**

Este principio se basa en que se puede detectar los defectos existentes pero más no afirmar la ausencia de estos, ya que en caso no se encontraran defectos tampoco significaría que el sistema está libre de defectos.

- **Pruebas Exhaustivas son imposibles**

Realizar la prueba de todas las combinaciones y entradas es imposible y no es factible, exceptuando los casos triviales, en su lugar se debe analizar los riesgos, las técnicas de prueba y la priorización de los esfuerzos de las pruebas.

- **Las pruebas tempranas ahorran tanto tiempo y dinero**

Para poder detectar los defectos lo más pronto posible se realizan las pruebas estáticas y dinámicas, las cuales se inician lo más pronto posible dentro del ciclo de vida del desarrollo del software ya que permiten minimizar los cambios a futuro con estimaciones costosas.

- **Los defectos se agrupan**

Este principio menciona que una reducida cantidad de módulos contiene la gran mayoría de defectos que se han detectado durante las pruebas realizadas previamente a la entrega.

- **Paradoja del pesticida**

Se basa en la repetición de las pruebas varias veces y eso trae como consecuencia no encontrar nuevos defectos, para evitar esto lo que se debería de realizar es modificar acorde a los nuevos datos y que las pruebas varíen para obtener o detectar nuevos defectos.

- **Las pruebas dependen del contexto**

Las pruebas van a variar con respecto al contexto de aplicación, por ejemplo un software industrial no tendrá las mismas especificaciones de pruebas a diferencia de un aplicativo móvil o web.

- **La ausencia de errores es una falacia**

Se hace referencia que al realizar las pruebas se hace una mitigación pero más no eliminar el 100% de los defectos existentes, acorde a los principios anteriormente descrito ya que no se pueden realizar todas las combinaciones de condiciones y que las pruebas detectan los defectos y no su ausencia.

2.2.6.3 Proceso de pruebas

Según ISTQB (2018), los procesos de prueba van a tener una modificación y dependencia acorde a:

- El modelo del ciclo de vida que se esté aplicando en el proyecto.
- Los riesgos del producto y proyecto.
- El dominio empresarial.
- Limitaciones en operaciones como presupuesto, recuerdos, complejidad, plazos de entrega, regulaciones, políticas organizativas, normas internas, externas y otros.

Actividades del proceso de pruebas

- **Planificación de prueba**

Son actividades para la definición de los objetivos de pruebas y cuáles son las limitaciones del contexto, se pueden describir

específicamente técnicas y tareas de las pruebas idóneas, la formulación de un cronograma de fechas para el cumplimiento de las pruebas según fechas límites. También los planes de pruebas pueden ser revisados en función a las actividades en el Monitoreo y Control (ISTQB, 2018).

- ***Monitorización y control de prueba***

Se refiere a la continua comparación del proceso real con lo descrito en el plan de pruebas utilizando métricas de monitoreo de pruebas las cuales se han definido en el plan. Mientras el control de las pruebas se refiere a la toma de acciones requeridas para cumplir los objetivos descritos en el plan, el cual puede actualizarse en el transcurso del tiempo), aquí se pueden agregar las pruebas requeridas según los cambios ya que una cantidad de pruebas están destinadas a un nivel de cobertura (ISTQB, 2018).

- ***Análisis de la prueba***

Según ISTQB (2018) hace referencia a la determinación “qué se va a probar” en criterios de cobertura que sean medibles. Lo cual puede incluir:

- Especificación de requisitos como funcionales, de sistema, historias de usuario, épicas, productos de trabajo, casos de uso, entre otros.
- Información como diagramas, documentos de la arquitectura del sistema o el software, entre otros.
- El código del componente o sistema, metadatos y las consultas a las bases de datos existentes e interfaces.
- El análisis de riesgo lo cual considera aspectos tanto funcionales como no funcionales y de estructura de un componente o todo el sistema.

También la consideración para evaluar las posibles omisiones, inexactitudes, contradicciones, sentencias superfluas y otras como ambigüedades, etc.

La detección de los defectos en esta fase de análisis es un gran beneficio porque no se utiliza otro tipo de herramienta más que la revisión. Realizar algunos tipos de revisiones como el Desarrollo Guiado por Comportamiento (BDD) o el Desarrollo Guiado por las Pruebas de Aceptación (ATDD) que se realizan en base a las condiciones y casos de prueba de los requerimientos, historias de usuario donde se validan y verifican los criterios de aceptación.

- ***Diseño de la prueba***

En esta fase se realiza la pregunta “Cómo realizar la prueba”, para eso se tienen algunos aspectos importantes:

- El diseño y la priorización de los casos de pruebas.
- La identificación de los datos requeridos como condiciones de pruebas.
- Diseñar el inicio de la prueba, también identificar las herramientas y la infraestructura acorde las pruebas por ejecutar.

- ***Implementación de la prueba***

Según ISTQB (2018) esta fase la pregunta que se responde es “¿Tenemos todo listo para la ejecución de las pruebas?”, para lo cual se tiene que tener en cuenta:

- Preparación de datos de prueba.
- Crear entornos de prueba como por ejemplo virtualización de servicios, elementos de infraestructura, verificando que lo previamente configurado esté de forma correcta.

En muchas ocasiones las tareas del diseño y la implementación se combinan en uno solo, también se puede documentar mediante la realización de las pruebas exploratorias en base a los contratos de pruebas elaborados en el análisis de prueba.

- ***Ejecución de prueba***

Según ISTQB (2018) toda la ejecución de pruebas incluye lo siguiente:

- Los identificadores y las versiones de cada elemento, herramientas y el software de la prueba.

- La comparación de resultados (reales y esperados).
 - En análisis de posibles anomalías causadas por defectos en código o sean falsos positivos.
 - El reporte de los defectos teniendo en cuenta las prueba realizadas,
 - El registro de la ejecución de las pruebas (con estados como pass, fail, blocked).
 - Realizar las pruebas como una prueba de confirmación después de haberse corregido según el reporte, también se puede realizar pruebas de regresión).
- **Completación de prueba**

Según ISTQB (2018) en esta fase se hace la recopilación de toda la información obtenida en base a las pruebas realizadas previamente, estos pueden ser entregados según la estructura del proyecto, al final de un proyecto desde un enfoque tradicional o un entregable desde la perspectiva de un marco de trabajo ágil, lo cual puede ayudar mucho en una reunión de Retrospectiva, algunas de las actividades son:

- La creación de un pequeño informe como resumen de las pruebas realizadas previamente.
- Archivar el ambiente, datos, infraestructura de prueba y otros utilizados para su posible reutilización más adelante.
- En análisis de lo aprendido entorno a las pruebas realizadas y los resultados obtenidos para planificar cambios en las futuras iteraciones y/o entregables según sea el caso.
- Mejorar la madurez de las pruebas en base a la información obtenida.

2.2.6.4 Desarrollo de Software y pruebas de Software

Según ISTQB (2018) menciona que para todo proceso de pruebas está hecho según el modelo de ciclo de vida del software que estén aplicando pero se tiene identificado que:

- Por cada tarea o actividad en el desarrollo siempre va a existir una tarea de prueba correspondiente.

- Los encargados de las pruebas tienen participación en las diferentes reuniones donde se define o refina los requerimientos, el diseño, historias de usuario, entre otros tanto como sea posible y estén disponibles.

Independiente del modelo de desarrollo de software que se esté utilizando, las tareas de pruebas comienzan en la fase inicial del ciclo de vida del desarrollo del software, teniendo presente el principio de las pruebas tempranas.

2.2.6.5 Niveles de pruebas

Según ISTQB (2018) son un conjunto de actividades que se organizan entre sí, donde se le considera como una instancia a cada nivel de prueba referente al proceso de pruebas, las cuales guardan relación con otras actividades en el modelo de desarrollo de software, los niveles establecidos son:

- Pruebas de componentes.
- Pruebas de Integración.
- Pruebas del Sistema.
- Pruebas de Aceptación.

Tener en cuenta que para cada prueba se va a requerir un ambiente apropiado, por ejemplo en las pruebas de aceptación de usuario (UAT) es propicio tener un ambiente parecido al de producción/ operación, otro ejemplo el de las pruebas unitarios o por componentes donde se realiza en el ambiente de desarrollo.

Pruebas por Componente

Según ISTQB (2018) principal objetivo es probar de manera individual cada componente, además de incluir la reducción de riesgos, verificar comportamientos tanto funcionales como no funcionales como según se especificaron, la detección de defectos en dicho componente y la prevención de que estos defectos lleguen a niveles más elevados.

En algunos casos donde se tiene marcos de trabajo agiles donde constantemente se va cambiando el código acorde a los requerimientos

es importante realizar dicha prueba de componentes, por ejemplo la corrección de una función que tiene como objetivo realizar ciertos cálculos.

Base de prueba de Componentes

Se tiene como base el diseño a detalles, el código, modelo de datos, las especificaciones de los componentes.

Objetos de prueba de Componentes

Los objetos de prueba más comunes son los componentes como tal, el código y las estructuras de datos, las clases y los módulos de las bases de datos existentes para el sistema.

Defectos y fallos típicos de Componentes

Funcionalidades incorrectas en base a las especificaciones, problemas en los flujos de datos, código y lógica de forma incorrecta.

Pruebas de Integración

Según ISTQB (2018) este tipo de prueba se centra en la interacción entre los componentes y/o el sistema, se centran en las interfaces y sistemas, los paquetes y microservicios, un ejemplo es la prueba hacia un servicio web.

Bases de pruebas de Integración

El diseño del software y del sistema, los diagramas de secuencia, casos de uso, flujos de trabajo, entre otros.

Objetos de prueba de Integración

Las bases de datos, interfaces, APIs, Microservicios e infraestructura.

Defectos y fallos típicos de Integración

Los datos incorrectos, faltantes y codificación incorrecta, la sincronización al llamar las interfaces de manera incorrecta, incongruencias, entre otros, un claro ejemplo es la estructura de mensajes incorrectos entre los sistemas.

Pruebas de Sistema

Según ISTQB (2018) este tipo de pruebas tiene prioridad hacia el comportamiento y las capacidades del sistema o también llamado producto, se conoce a esta prueba como extremo a extremo, estas pruebas son las que otorgan información que las partes interesadas utilizan para tomar las decisiones correspondientes de la entrega.

Base de prueba de Sistema

Las especificaciones de los requerimientos funcionales y no funcionales, casos de uso, historias de usuario, diagrama de estados, manuales de sistema y usuario.

Objeto de pruebas de Sistema

Las aplicaciones, los sistemas de hardware y software, sistemas operativos, entre otros.

Defectos y fallos típicos de Sistema

Los cálculos realizados incorrectamente, los comportamientos tanto funcionales como no funcionales, fallas del sistema en ambientes como producción, fallos del sistema en funcionamiento en base a los manuales de usuario y de sistema.

Pruebas de Aceptación

Según ISTQB (2018) este tipo de pruebas se enfoca en las capacidades y comportamientos de todo el sistema, lo que se desea es generar confianza en la calidad del producto, validar el funcionamiento correcto tal como se espera, verificar que los comportamientos tanto funcionales como no funcionales son los que se habían especificado.

Este tipo de pruebas se subdivide en los siguientes:

Pruebas de aceptación del usuario (PAU)

Se refiere a la aceptación del sistema por parte del cliente/usuarios, los cuales validan el sistema creando confianza y que el sistema pueda satisfacer las necesidades según los requisitos.

Pruebas de aceptación operativa (PAO)

Se realiza por parte del personal que administrará el sistema elaborado, aquí se simula como si estuviera en producción, algunos aspectos que incluyen son la copia de seguridad y restauración, recuperación ante los desastres, la gestión de usuarios, las tareas de mantenimiento, la instalación, desinstalación y la actualización del sistema, pruebas de rendimiento y carga de los datos.

Pruebas de aceptación contractuales y regulatorias

El tipo de pruebas contractuales se realizan según el contrato establecido previamente las cuales son realizadas por el usuario u otro tester y el tipo de prueba regulatorio se refiere a las normas o políticas establecidas en un país o estado las cuales un producto como lo es un sistema deba cumplir al estar en operatividad.

Pruebas Alfa y Beta

Suelen ser usada por desarrolladores de software comercial y obtener los comentarios correspondientes según los resultados. Las pruebas Alfa se realizan dentro el ambiente de desarrollo de la misma organización que está elaborando el sistema, a diferencia del tipo Alfa las pruebas Beta se realizan después de las pruebas Alfa y estas emulan un entorno operativo donde los usuarios pueden realizar las diversas acciones como si el sistema estuviera ya en producción.

2.2.6.6 Tipos de Pruebas

Según ISTQB (2018) los tipos de prueba están avocadas hacia la medición de las características específicas del sistema, las cuales puede ser:

- Calidad funcional como integridad, la exactitud y pertenencia correspondiente
- Calidad no funcional como confiabilidad, eficiencia del rendimiento, la seguridad, usabilidad y compatibilidad correspondiente.
- La arquitectura de un componente o el sistema en su totalidad, si es correcta según las especificaciones establecidas.

Pruebas funcionales

Son las que evalúan el funcionamiento del sistema, se basan en las historias de usuario, especificación funcional, caso de uso. Aquí lo que se pregunta es “qué debe hacer el sistema”. Estas deben realizarse en todos los niveles de las pruebas pero se tiene un mayor enfoque en el comportamiento del sistema, una técnica muy conocida es la prueba de caja negra (ISTQB, 2018).

Pruebas no funcionales

En este tipo de prueba se tiene presente “qué también se comporta el sistema”, se pueden realizar un análisis para definir las condiciones de estrés en las pruebas de rendimiento, también las pruebas de compatibilidad al momento de usar dispositivos e identificar la brecha de cobertura (ISTQB, 2018).

Pruebas de Caja Blanca / Estructurales

Se basa en la estructura interna las cuales incluyen código, flujos de datos contenidos en el sistema, arquitectura. En las pruebas de integración de los componentes, el tipo de pruebas de caja blanca se basa en la arquitectura aplicada, las interfaces entre los componentes, entre otros (ISTQB, 2018).

Pruebas relacionadas con el cambio

En este tipo de prueba se tiene en cuenta los cambios en algún componente, modulo o sistema, en el cual se deben realizar las pruebas para confirmar que esos cambios han sido realizados satisfactoriamente. Se tiene las pruebas de confirmación que se realizan a partir de cambios realizados en el código, etc., otro tipo de pruebas relacionadas al cambio son las pruebas de regresión que es básicamente la revisión si los cambios que han sido realizados han afectado a algún otro componente.

Pruebas de mantenimiento

Este tipo de pruebas se realiza según ciertos aspectos como alguna existencia de modificación de código u otro componente, la migración

de una versión actual a otra donde se use otra versión de plataforma y la jubilación del sistema donde se va a dejar de usar el sistema actual.

2.2.7 SCRUM

Es un marco de trabajo simple con enfoque a la organización de los equipos y realizar las tareas con mayor productividad y calidad, con la capacidad de elegir el trabajo a realizar de acuerdo a lo que el equipo establezca (Sutherland & Schwaber, 2007, citado en (Khalane, 2013). Nace por un artículo realizado por Nonaka y Takeuchi, donde mostraban el cómo las empresas japonesas que innovaban podían realizar las actividades con el fin de producir pero en el menor tiempo, con menores costes y con mayor calidad, también se hace mención del por qué se puso el nombre de Scrum y fue por una jugada en el Rugby (scrum en inglés) y por eso quedó ese nombre, se destaca una serie de buenas prácticas como el trabajo en equipo y la obtención de mejores resultados en un proyecto al igual que en la jugada del Rugby llamada scrum (Sánchez-Oro Portillo, 2017). Según Atlassian (2022) menciona que su mayor uso se da en los equipos de desarrollo de software pero scrum en base a sus principios y lecciones fácilmente se podrían aplicar a cualquier entorno y tipo de trabajo en equipo, esa es una de las tantas razones por las que es popular. Muchas veces se suele confundir scrum con la metodología ágil porque scrum tiene un enfoque a la mejora continua pero scrum es un marco de trabajo, mientras que la metodología ágil es una mentalidad. Scrum es heurístico, se basa en el aprendizaje continuo y también se adapta a factores cambiantes, reconociendo que el equipo no sabe todo cuando se inicia un proyecto pero si se tiene prioridad en la evolución en base a la experiencia, dentro de este marco de trabajo los equipos se puede adaptar a los cambios que existieran con referencia a los requerimientos de usuario, lanzamientos muy cortos.

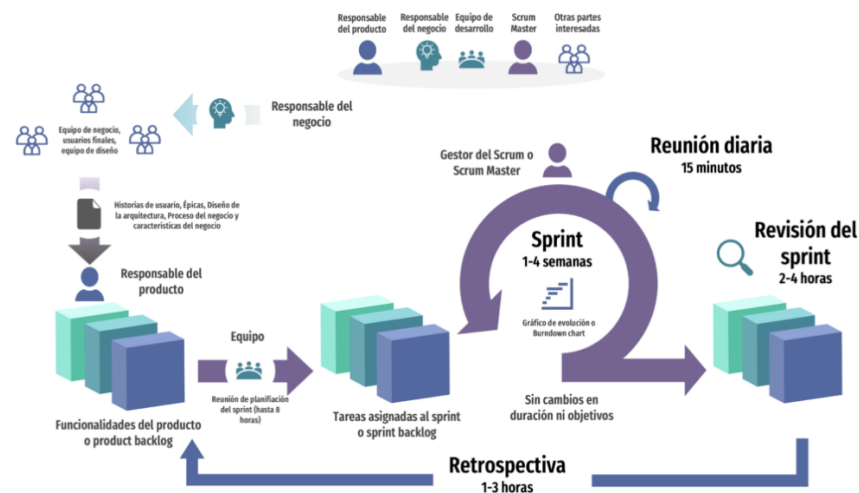
Según Schwaber & Sutherland (2020) Scrum cuenta con 3 importantes pilares los cuales son la transparencia, inspección y adaptación, estos se describen a continuación:

Transparencia: Se basa en que el proceso y el trabajo sean visibles tanto para el equipo encargado como para los clientes.

Inspección: Los artefactos y el progreso direccionados hacia el logro de los objetivos establecidos deben inspeccionarse con cierta frecuencia y diligencia para la posible detección de uno o varios problemas indeseables.

Adaptación: En caso de que el proceso tenga una desviación fuera de lo establecido o el producto resultante sea rechazado, se debe ajustar un poco la perspectiva hacia el logro de los objetivos teniendo cierto grado de adaptación y que se realice lo más pronto posible.

Figura 8:
Ciclo Scrum



Nota: Ciclo del Marco de trabajo Scrum con sus ceremonias y el flujo de trabajo correspondiente. Adaptado de *Implementación de metodología scrum en proyectos* [Fotografía], por Metahabitos, 2022, (<https://www.metahabitos.com/gestion-de-proyectos/metodologia-scrum/>).

2.2.7.1 Roles

Según Schwaber & Sutherland (2020) en el equipo de scrum se tienen los siguientes roles:

- *Developer (DEV)*

Son las personas encargadas de que se tengan resultados para los entregables y consecuentemente un incremento en cada sprint realizado, por lo general son los encargados de realizar un Plan para el Sprint y Sprint Backlog, inculcando la calidad.

- *Product Owner (PO)*

Es el responsable de generar el mayor resultado de trabajo posible del equipo y la manera en que realice puede tener ciertas adaptaciones acorde a la organización y proyecto. Algunos de sus

objetivos es el comunicar explícitamente los objetivos para el producto, ordenar elementos y garantizar la lista dentro del Product Backlog para que sea visible, transparente y comprensible.

- *Scrum Master (SM)*

Es el responsable de que se realicen las actividades acorde al marco de trabajo Scrum, ayudando a comprender teórico-práctico todo sobre scrum dentro de la organización, así mejorando sus prácticas dentro del ambiente.

2.2.7.2 Eventos

Según Schwaber & Sutherland (2020) los eventos dentro de scrum están diseñados para permitir la transparencia, también para crear regularidad y minimizar las posibles necesidades de reuniones no definidas dentro de scrum.

The Sprint

Según Schwaber & Sutherland (2020) es una parte fundamental dentro de scrum (*latidos del corazón como hace referencia Schwaber & Sutherland*), es donde todas las ideas se convierten en valores, tienen una duración fija aproximada de un mes, al término de un sprint comienza otro, todas las actividades para lograr el objetivo como el Sprint Planning, Scrum Daily, Sprint Retrospective, están incluidas en los sprints.

Estos sprints permiten la visibilidad al garantizar la inspeccionar y adaptar el progreso acorde al objetivo del producto y es mejor tener sprints cortos con mayor aprendizaje que los sprints más largos donde pueda adquirir mayor complejidad.

Sprint Planning

Según Schwaber & Sutherland (2020) es donde se realiza el diseño de lo que se realizará dentro del Sprint, se realiza de forma colaborativa para la selección de elementos relacionados a los objetivos del producto donde el objetivo debe finalizar antes del Sprint Planning. También de manera conjunta se seleccionan los elementos del Product Backlog para insertarlos dentro del presente Sprint, posteriormente se puede dar un refinamiento de dichos elementos para mejorar la comprensión y

confianza. Para la realización de los elementos seleccionados se hacen particiones más pequeñas las cuales puedan tenerse en un día o menos y consecuentemente se tengan los incrementos de valor.

Daily Scrum

Según Schwaber & Sutherland (2020) es la inspección del progreso con referencia a alcanzar el objetivo en base al Sprint Backlog, estas reuniones se dan de manera diaria con una duración aproximada de 15 minutos, la cual se realiza en el mismo horario y lugar donde se hablan temas respecto a las actividades que están realizando y la situación en cada una de ellas, generando un enfoque hacia la mejora de autogestión. Identifican los impedimentos, mejoran la comunicación y promueven las decisiones más rápidas eliminando la creación de otras reuniones.

Sprint Review

Según Schwaber & Sutherland (2020) su principal objetivo es inspeccionar los resultados del Sprint y generar futuras adaptaciones, aquí se presentan los resultados de sus trabajos a las partes interesadas para que se analice el progreso orientado al objetivo del producto, se verifica los cambios en el entorno, posteriormente los asistentes conversan sobre lo que se realizará después.

Sprint Retrospective

Según Schwaber & Sutherland (2020) menciona que se tiene como objetivo la planificación en cómo mejorar la calidad y eficacia acorde a los resultados del último Sprint, realizando suposiciones sobre los “por qué” sucedieron tales actividades y generaron tales resultados, analizan cómo les fue en el último Sprint y evalúan si tuvieron conflictos y cómo los resolvieron o no, finalmente se identifican los cambios más óptimos para los casos mencionados previamente.

2.2.7.3 Artefactos

Según Schwaber & Sutherland (2020) los artefactos representan trabajo o valor ya que están diseñados para la transparencia de la información, donde cada artefacto tiene el compromiso para garantizar la mayor información que mejore la transparencia.

Product Backlog

Según Schwaber & Sutherland (2020) se trata de una lista ordenada de lo que se requiere para mejorar el trabajo y se realiza por el equipo scrum, estos elementos posteriormente serán seleccionados para el Sprint Backlog mediante la reunión del Sprint Planning.

Sprint Backlog

Según Schwaber & Sutherland (2020) son elementos que previamente han sido seleccionados del Product Backlog para el Sprint y son almacenados en el Sprint Backlog, el cual es un plan de los desarrolladores y para los desarrolladores donde se tiene visibilidad de qué es lo que se realizará acorde al Sprint para el logro del objetivo.

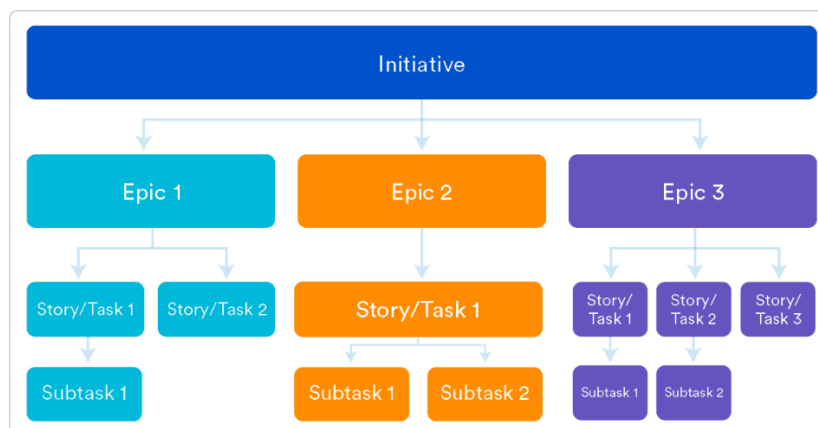
Increment

Según Schwaber & Sutherland (2020) un incremento es un aporte concreto hacia el objetivo del producto que se ha establecido, dicho incremento se suma a otros anteriores y se verifica que todos estos funcionen en conjunto y se pueden crear muchos incrementos dentro de un Sprint.

2.2.7.4 Historias de Usuario

Según Atlassian (2022) las historias de usuario (user stories) son la mínima unidad de trabajo en un marco ágil que se expresa desde la perspectiva del cliente y/o usuario final del software. Es una explicación general e informal de una funcionalidad de software desde el enfoque del usuario, es la forma cómo un elemento de trabajo va a entregar valor hacia el cliente, teniendo un pequeño conjunto de palabras los cuales son: “Como (PERFIL), quiero... para..., un ejemplo claro sería “*como administrador quiero que se pueda realizar la eliminación de los registros de clientes para realizar una mejor gestión del total de registros*”.

Figura 9
Historias de Usuario



Nota: Historias de usuario y la relación con las tareas y épicas dentro de los requerimientos de un proyecto. Adaptado de *Historias de Usuario* [Fotografía], por Atlassian Agile Coach, 2022, (<https://www.atlassian.com/es/agile/project-management/user-stories>).

2.2.8 ISO/IEC 25010

Según la ISO 25000 (2022) se refiere a la piedra angular entorno a un modelo de calidad donde se evalúa la calidad de un producto, en dicho modelo se describen características que la calidad va a tener en cuenta al momento de su evaluación de un software determinado. Dicha calidad se puede referir al nivel de satisfacción de los requerimientos que establecieron los clientes y/o usuarios aportando valor. Los cuales vendrían a ser la funcionalidad, seguridad, rendimiento, la mantenibilidad, entre otros.

Figura 10:
ISO/IEC 25010, características y subcaracterísticas



Características y Subcaracterísticas de la ISO/IEC 25010

- **Adecuación Funcional**

Según la ISO 25000 (2022) es la capacidad del producto de proporcionar las funciones que van a satisfacer las necesidades ya establecidas, subdividida en:

Compleitud Funcional: Trata sobre el nivel en el cual las funcionalidades cubren todos los objetivos del usuario que han sido especificados.

Corrección funcional: La forma de cómo el sistema provee los resultados esperados con la precisión necesaria.

Pertinencia funcional: Es la proporción de funciones apropiadas para los objetivos que el usuario ha establecido.

- **Eficiencia de desempeño**

Se refiere al desempeño y los recursos utilizados bajo ciertas condiciones y se divide en las siguientes subcaracterísticas:

Comportamiento temporal: Tiempos de proceso y respuesta bajo condiciones de comparación con otros productos.

Utilización de recursos: Tipo y cantidad de recursos usados en el funcionamiento bajo ciertas condiciones.

Capacidad: Se refiere a los límites máximos dentro de un sistema de software para el cumplimiento de los requerimientos.

- **Compatibilidad**

Se refiere a la capacidad de la transferencia de información entre componentes o llevar las funciones requeridas usando el mismo hardware o software, se divide en las siguientes subcaracterísticas:

Coexistencia: Hace referencia a la convivencia haciendo uso de un mismo entorno y recursos.

Interoperabilidad: Se refiere a la utilización de información intercambiada entre dos o más sistemas o componentes.

- **Usabilidad**

Es el uso desde la perspectiva del usuario, enfocado al entendimiento, aprendizaje y que resulte agradable y deba ser usado bajo algunas condiciones, se divide en las siguientes subcaracterísticas:

Capacidad para reconocer su adecuación: Se refiere al entendimiento del software por parte del usuario.

Capacidad de aprendizaje: Se refiere al aprendizaje del software por parte del usuario.

Capacidad para ser usado: Se refiere a la usabilidad, operación, control con facilidad del software por parte del usuario.

Protección frente a errores de usuario: La protección ante los errores que pudiese cometer el usuario.

Estética: Si las interfaces son agradables y satisfacen la interacción con los usuarios.

Accesibilidad: Si el producto puede ser usado bajo ciertas discapacidades.

- **Fiabilidad**

Si el sistema puede desempeñarse bajo ciertas condiciones, se divide en las siguientes subcaracterísticas:

Madurez: Fiabilidad en condiciones naturales, normales para satisfacer las necesidades de los usuarios.

Disponibilidad: Es la operatividad y accesibilidad del sistema en base a tiempos, que se pueda usar cuando se requiera.

Tolerancia a fallos: La operación del sistema según lo esperado en base a la presencia de los fallos tanto del hardware como del software.

Capacidad de recuperación: Recuperación y restablecimiento del estado esperado cuando ocurren ciertos fallos o interrupciones.

- **Seguridad**

Está referido a la protección de la información y datos de los usuarios y otros, donde no puedan ser leídos o modificados, se divide en las siguientes subcaracterísticas:

Confidencialidad: Protección al acceso de datos y la información no autorizada.

Integridad: Prevención ante los acceso y las modificaciones que no están autorizadas.

No repudio: Demostrar los eventos acontecidos sin ningún inconveniente.

Responsabilidad: Se refiere a rastrear las acciones realizadas por una entidad.

Autenticidad: Mostrar la identidad de un recurso o un sujeto en específico.

- **Mantenibilidad**

Las modificaciones no causan fallas y el sistema no resulta afectado, se divide en las siguientes subcaracterísticas:

Modularidad: Permite el cambio en un componente y la repercusión de este sea el mínimo en los demás.

Reusabilidad: La reutilización de un activo en uno o más componentes o sistemas.

Analizabilidad: Capacidad con la que se pueda evaluar los cambios realizados y el impacto que tengan en los demás componentes del software.

Capacidad para ser modificado: Que el producto sea modificado sin que cause defectos en el desempeño.

Capacidad para ser probado: Que se pueda llevar a cabo las pruebas para un componente o sistema y determinar si cumplen con lo esperado de acuerdo a los criterios establecidos.

- **Portabilidad**

Trata de la capacidad de transición de un ambiente a otro, ya sea en hardware o software y que el producto o componente siga funcionando de forma eficiente y efectiva, se divide en las siguientes subcaracterísticas:

Adaptabilidad: El cambio de entornos y la respuesta efectiva y eficiente del producto en base a hardware o software.

Capacidad para ser instalado: Facilidad de instalación de un producto en cierto entorno.

Capacidad para ser reemplazado: El producto o sistema puede ser reemplazado por otro con el mismo propósito y en el mismo ambiente.

2.3 Definición de términos

- **QA**

Son las iniciales de Quality Assurance (QA) o en español el aseguramiento de la calidad, que se refiere a una serie de actividades las cuales se encargan de evaluar y definir las características mínimas y óptimas de los procesos de software con el fin de generar mayor confianza según los requerimientos (McGrath, 2014, como se citó en Ibarra Luevano, 2018).

- **Software**

Es un programa de cómputo donde las instrucciones que se ejecutan otorgan funciones, características y desempeños, estructura de datos el cual permite la manipulación de la información, la descripción de las operaciones y uso de programas de la información descriptiva tanto en papel como en aspectos virtuales (Pressman, 2010).
- **Hardware**

Se usa en informática para describir a una serie de componentes que integran la parte material de un ordenador u computador (Real Academia Española, 2022).
- **Software Factory Enterprise**

Se refiere a una empresa como fábrica de software, la cual usa técnicas y principios en la fabricación del software en base a requerimientos definidos por el usuario y definidos a través de un proceso (Wikipedia, 2022).
- **E-learning**

Es un extranjerismo que en español significa educación en línea donde los docentes y estudiantes están bajo un entorno digital de las nuevas tecnologías, redes y computadoras (Wikipedia, 2022).
- **Bots**

Nace como una minimización de la palabra robot, es un programa construido con el fin de realizar tareas repetitivas y están automatizadas (Kaspersky, 2022).
- **Retail**

Conocido la venta minorista o a detalle, la cual abarca a todas las empresas abocadas a la comercialización de productos y servicios al consumidor final (Wikipedia, 2022).
- **Mype**

Son compañías con un promedio de 10 trabajadores, también se referencia a 3 categorías las cuales son pequeñas y medianas empresas (pymes) con una mayor cantidad de trabajadores y facturación (BBVA, 2022).
- **Testing**

Es una disciplina dentro de la ingeniería de software, contenida por herramientas, procesos y métodos, con la finalidad de identificar defectos en el software y de esta manera alcanzar cierta estabilidad (Hoyos, 2022).

- **Scrum**

Es un marco de trabajo simple para la organización de los equipos y realizar las tareas con mayor productividad y calidad (Sutherland & Schwaber, 2007, citado Khalane, 2013).
- **ISTQB**

El Comité Internacional de Certificaciones de Pruebas de Software (International Software Testing Qualifications Board, ISTQB), la cual otorga documentos y glosarios en base a las pruebas de software (Wikipedia, 2022).
- **Stakeholders**

Definidos como el grupo de interés o las partes interesadas dentro de un proyecto, organización, entre otros (ESAN, 2021).
- **Waterfall**

Conocido también como Cascada, es una metodología para el desarrollo de software con un enfoque lineal, el cual es ideal para proyectos donde los objetivos están bien definidos (Forbes, 2022).
- **Testlink**

Es una herramienta para la gestión de las pruebas basadas en sistemas web que optimizan el aseguramiento de la calidad del software, soporta casos de prueba, suites de prueba, planes y proyectos de prueba (Wikipedia, 2022).
- **ISO**

Es una organización internacional de Normación conocida mediante su acrónimo ISO, se enfoca en la creación de estándares internacionales (Wikipedia, 2022).
- **BDD**

Según ISTQB (2018) el Desarrollo Guiado por Comportamiento (BDD). Es un lenguaje común para el entendimiento de todos, se utiliza para el desarrollo y testing (Immune Technology Institute, 2021).
- **ATDD**

Es un esfuerzo de todo el equipo donde se reúnen los desarrolladores con los testers, ingenieros, stakeholders, dando como resultado los requisitos de la aplicación, en un formato previamente establecido y comprensible para todos y que luego se convierten en las pruebas de aceptación automatizadas (Digite, 2022).

- **Pass**

Se considera como una prueba con estado “pass” o pasado si es que el resultado coincide con lo esperado (IBM, 2022).
- **Fail**

Se considera “fail” o falla si es que el resultado de una prueba no coincide con lo esperado (IBM, 2022).
- **Blocked**

Se considera “blocked” o bloqueado cuando no se cumplen las condiciones previas establecidas para la ejecución de la prueba (IBM, 2022).
- **APIs**

Es un mecanismo que permite que dos sistemas o componentes se comuniquen entre sí mediante definiciones y/o protocolos (AWS Amazon, 2022).
- **Atlassian**

Es una empresa de desarrollo de software la cual tiene como principales aportes o productos a Jira, Trello y miles de equipos en todo el mundo, con el objetivo de buenas prácticas, entornos de trabajo, entre otros (Atlassian, 2022).
- **Planificación**

Actividades para definir objetivos de pruebas y limitaciones de contexto.
- **Monitorización y Control**

La monitorización es la comparación del proceso real con lo estipulado en el plan de pruebas. El control es la toma de decisiones para lograr lo descrito en el plan de pruebas.
- **Análisis**

Evaluación de lo que va a probar mediante criterios de cobertura medibles.
- **Diseño**

Se considera la forma en cómo se realizarán las pruebas teniendo en cuenta diseño, priorización de casos de pruebas, entre otros.
- **Implementación**

Se tiene en cuenta la configuración del ambiente para la ejecución de pruebas como por ejemplo servidores, virtualización, infraestructura, entre otros.
- **Ejecución**

Referido a actividades como comparación de resultados, análisis de anomalías, reporte de defectos, entre otros.

- **Compleción**
Recopilación de información de las pruebas realizadas, creación de un informe, análisis del aprendizaje obtenido, mejorar la madurez de las pruebas, entre otros.
- **Completitud funcional**
Nivel de cubrimiento de las funcionalidades especificadas por el usuario (ISO 25000, 2022).
- **Corrección funcional**
Entrega de resultados esperado precisos (ISO 25000, 2022).
- **Pertinencia funcional**
Proporción de funciones apropiadas según los objetivos establecidos por el usuario (ISO 25000, 2022).
- **Comportamiento temporal**
Tiempos de procesamiento y respuesta en comparación con otros productos (ISO 25000, 2022).
- **Utilización de recursos**
Tipo y cantidad de recursos usados en el funcionamiento bajo condiciones (ISO 25000, 2022).
- **Coexistencia**
Convivencia entre el uso del entorno y recursos (ISO 25000, 2022).
- **Interoperabilidad**
Utilización de información intercambiada entre dos o más sistemas o componentes (ISO 25000, 2022).
- **Aprendizaje**
Aprendizaje del software por parte del usuario (ISO 25000, 2022).
- **Accesibilidad**
Que el producto pueda ser usado bajo ciertas discapacidades (ISO 25000, 2022).
- **Protección frente a errores de usuario**
Protección ante los errores que pudiese cometer el usuario (ISO 25000, 2022).
- **Estética**
Que interfaces sean agradables y si satisfacen la interacción con los usuarios (ISO 25000, 2022).

- **Confidencialidad**
Protección al acceso de datos e información no autorizada usuarios (ISO 25000, 2022).
- **Integridad**
Prevención de acceso y las modificaciones no autorizadas (ISO 25000, 2022).
- **No repudio**
Demostrar los eventos acontecidos sin inconveniente (ISO 25000, 2022).

2.4 Hipótesis

2.4.1 Hipótesis general

La aplicación de Quality Assurance enfocado en Agile Testing mejora la calidad de software en la empresa Wasi Innovation SAC, Lima – 2022.

2.4.2 Hipótesis específicas

- Lograr identificar la realidad de la calidad de software antes de la aplicación de Quality Assurance enfocado en Agile Testing en la empresa Wasi Innovation SAC, Lima – 2022.
- Lograr identificar la realidad de la calidad de software después de la aplicación de Quality Assurance enfocado en Agile Testing en la empresa Wasi Innovation SAC, Lima – 2022.

2.5 Variables

2.5.1 Variable Independiente

- **Quality Assurance enfocado en Agile Testing**
Esta variable está conformada por la unión de otras las cuales son Quality Assurance y Agile Testing, las cuales se definen:
 - ❖ **Quality Assurance:** Es una serie de actividades las cuales se encargan de evaluar y definir las características mínimas y optimas de los procesos de software con el fin de generar mayor confianza según los requerimiento (McGrath, 2014, como se citó en Ibarra Luevano, 2018).
 - ❖ **Agile Testing:** Es la implementación de las pruebas de software desde el inicio del proceso de desarrollo, dándole importancia al proceso donde cada entregable que forma parte del desarrollo de software y no la entrega final del producto (Hoyos, 2022).

2.5.2 Variable dependiente

- Calidad de Software

Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan (Pressman, 2010).

2.5.3 Operacionalización de variables

Variables	Definición	Dimensión	Subdimensión	Indicadores	Instrumento
(V.I.) Quality Assurance enfocado en Agile Testing	<p>Quality Assurance: Es una serie de actividades las cuales se encargan de evaluar y definir las características mínimas y optimas de los procesos de software con el fin de generar mayor confianza según los requerimiento (McGrath, 2014, como se citó en Ibarra Luevano, 2018).</p> <p>Agile Testing: Es la implementación de las pruebas de software desde el inicio del proceso de desarrollo, dándole importancia al proceso donde cada entregable que forma parte del desarrollo de software y no la entrega final del producto (Hoyos, 2022).</p>	Framework ISTQB	Modelo del procesos de pruebas	Planificación	Guía de Entrevista
				Monitorización y Control	
				Análisis	
				Diseño	
				Implementación	
				Ejecución	
				Compleción	
(V.D.) Calidad de Software	Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan (Pressman, 2010).	ISO/IEC 25010	Adecuación Funcional	Compleitud funcional	Cuestionario
				Corrección funcional	
				Pertinencia funcional	
			Eficiencia de desempeño	Comportamiento temporal	
				Utilización de recursos	
			Compatibilidad	Coexistencia	
				Interoperabilidad	
			Usabilidad	Aprendizaje	
				Accesibilidad	
				Protección frente a errores de usuario	
			Seguridad	Estética	
				Confidencialidad	
Integridad					
	No repudio				

III. METODOLOGÍA

3.1 Tipo de estudio

La presente investigación se caracteriza por tener un enfoque Cuantitativo, con alcances de acuerdo a la perspectiva del estudio y los objetivos planteados en la investigación de carácter Descriptivo-Correlacional y según el grado de abstracción Aplicada.

Cuantitativa: porque se realiza la recolección y el análisis de los datos para responder a las incógnitas de la investigación y contrastar la hipótesis (Gómez, 2006).

Descriptivo: porque los estudios se enfocan en especificar las características, propiedades y otros aspectos importantes del fenómeno que se está sometiendo al análisis (Gómez, 2006). En nuestra investigación, específicamente nuestra variable Calidad de Software, donde se estudian la realidad como sus características, especificaciones y otros los cuales serán importantes para la identificar la realidad de la calidad de software antes de la aplicación de Quality Assurance enfocado en Agile Testing en la empresa que es contexto de la presente investigación.

Correlacional: evalúa la relación existente entre dos o más conceptos en un contexto particular, además aporta conocimiento de cómo se puede comportar una variable conocimiento el comportamiento de otra variable que está relacionada (Gómez, 2006). En nuestra investigación el aporte correlacional se da a partir de la identificación de la realidad de la calidad de software después de la aplicación de Quality Assurance enfocado en Agile Testing en la empresa que es contexto de la presente investigación.

Aplicada: porque se hace uso de los conocimientos adquiridos previamente para la solución de un problema identificado (Hernández et al. 2014, citado en Valenzuela Salvador, 2020).

3.2 El diseño de investigación

Se tiene como diseño de esta investigación el tipo pre-experimental, en lo cual se tuvo presente la realización de pruebas en dos etapas (pre y pos prueba respectivamente) en un grupo experimental, también está referido a que el investigador puede realizar el contraste de los cambios surgidos en un grupo experimental tras recibir el estímulo (Hernández et al. 2014, citado en Valenzuela Salvador, 2020).

3.3 Descripción de la unidad de análisis, población y muestra

3.3.1 Unidad de análisis

La unidad de análisis en la investigación es el área de desarrollo de la empresa Wasi Innovation, que tiene como objetivo la elaboración de un software y el cliente que está a cargo de la evaluación de dicho proyecto.

3.3.2 Población

En la presente tesis la población está definida a todo el personal del actual proyecto que está siendo desarrollado en la empresa que es el contexto de nuestra investigación y el equipo que verifica el avance del proyecto como representación del cliente. La población con una cantidad total de 20 personas, se detalla en la siguiente tabla:

*Tabla 1 :
Población de la Investigación*

Cargo que desempeña	Cantidad Empleados
Gerente	1
Líder de Proyecto	1
Arquitecto de software	1
Desarrolladores de software	6
Calidad de software	5
Equipo de evaluación (Clientes)	6
TOTAL	20

Nota: No se provee la información específica de los empleados como medida de prevención, establecida dentro de la protección de los datos personales en la Ley N° 29733 (Poder Legislativo del Perú, 2011).

3.3.3 Muestra

Si se tiene una población pequeña, lo preferible es observar a toda esa cantidad de individuos que lo conforman, ya que una mejor alternativa es estudiar a toda una población antes que una pequeña muestra de ella (McGuigan, 1996).

Según Tamayo y Tamayo, citado por (Figueroa Martins & Narea Gutiérrez, 2016), se refiere al grupo de elementos que se consideran con intención de

averiguar las características de la población la cual integran, denominada por Tamayo como muestra censal ya que conforma la cantidad total de elementos adscritos a la investigación.

Se tiene una muestra no probabilística ya que para fines de la presente investigación solo se conforma un total de 20 personas. Por ende este tipo de muestra es censal ya que la población al ser pequeña se puede recolectar la información de todos ellos (Valenzuela Salvador, 2020).

3.4 Técnicas e instrumentos de recolección de datos

3.4.1 Técnicas

Encuesta: La encuesta describe la recolección de los datos mediante el instrumento del cuestionario previamente diseñado para la obtención de la información en base a enunciados cerrados. En esta investigación se tuvo presente el cuestionario para evaluar el nivel de calidad del software antes y después para realizar la contratación de la hipótesis (Malhotra, 2004, citado en Valenzuela Salvador, 2020). Para esta investigación se tendrá en cuenta como instrumento de medición un cuestionario.

Análisis documental: Se refiere a todos los documentos analizados como los términos de referencia y otros documentos especificados por la entidad cliente como base para el modelo del proceso de pruebas. Teniendo como instrumento de esta técnica a la ficha de análisis documental.

Entrevista: Se hace referencia a la entrevista como la obtención de información mediante preguntas abiertas con un estilo de testimonio del sujeto de estudio e investigador (Malhotra, 2004, citado en Valenzuela Salvador, 2020). En esta investigación se tendrá en cuenta una guía de entrevista.

3.4.2 Instrumentos

Cuestionario: Para este instrumento con técnica de encuesta se tendrá en cuenta las preguntas cerradas con alternativas múltiples en base a la satisfacción de la escala de Likert y a las métricas correspondientes de las características y subcaracterísticas de la ISO 25010.

Ficha de análisis documental: Se realizará el análisis de la documentación obtenida desde el inicio del desarrollo del proyecto como base para el modelo de procesos de las pruebas, teniendo mayor énfasis en los términos de referencia entregados al iniciar el actual proyecto en la empresa que es contexto de investigación.

Guía de Entrevista: Para la investigación en específico se realizará una encuesta al gerente general de la empresa Wasi Innovation con el objetivo de analizar la situación actual con respecto a la calidad de software en base a preguntas abiertas sin métricas, las cuales no entrarán dentro de la evaluación del contraste del pre prueba y post prueba.

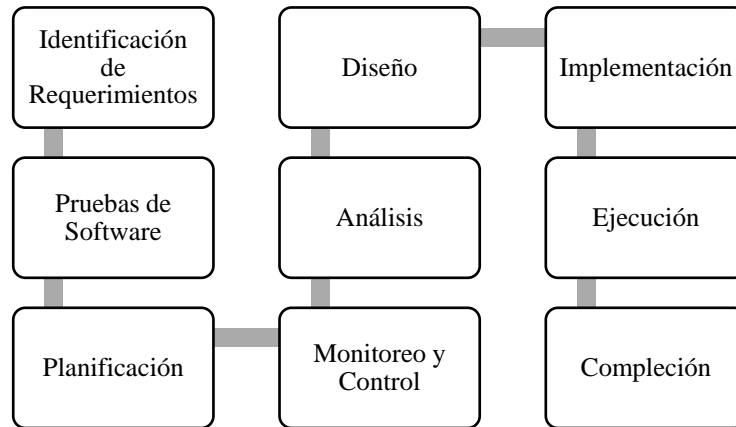
3.5 Técnicas de análisis y prueba de hipótesis

Para el desarrollo de la presente investigación, se tendrá en cuenta el modelo de proceso de pruebas ISTQB dentro de la metodología Scrum, la cual fue la seleccionada previamente por su enfoque y aportes como Agile, que será de ayuda sobre todo para el Agile Testing como medida para mejorar la calidad de software.

Previamente se realizará la recolección de datos para el análisis situacional mediante la entrevista, cuestionario y las pruebas de software respectivas, teniendo en cuenta sus diversos niveles y tipos, todas estas actividades con el propósito de saber la situación actual y el nivel de la calidad de software.

Posteriormente se realizará la implementación del modelo del proceso de pruebas que estará contenida por las fases de la Planificación, Monitoreo y Control, Analisis, Diseño, Implementación, Ejecución y Compleción, todo lo anterior mencionado teniendo como base la metodología Scrum. Teniendo en cuenta los diversos tipos y niveles de pruebas.

Tabla 2:
Fases del Proyecto



Nota: Fases del proyecto, las cuales están contenidas por los objetivos de la presente investigación.

Tabla 3:
Especificación de las fases para el Proyecto

OBJETIVO	ITEM	FASE	DESCRIPCIÓN
Análisis situación Actual - Pre test	Análisis situación	Identificación de Requerimientos	Se realizó la recolección de datos sobre la calidad de software con respecto a las métricas de la ISO 25010 (según las categorías y subcategorías seleccionadas según el vigente proyecto).
		Pruebas de Software	
Planificación			
Monitoreo y Control			
Análisis situación Actual – Post Test	Modelo del proceso de Pruebas	Análisis	
		Diseño	
		Implementación	
		Ejecución	
		Compleción	

- **Identificación de Requerimientos:** En esta fase se prioriza el análisis de la situación de la calidad de software dentro de la empresa antes de la implementación del modelo del proceso de pruebas y acorde a los resultados obtenidos se tomarán las decisiones correspondientes a implementar.
- **Pruebas de Software:** Por otra parte para también identificar la situación actual de la calidad de software pero en el vigente proyecto que están desarrollando, estas pruebas estarán basadas mediante las métricas de la ISO 25010.

Para las siguientes fases se tendrá como base scrum como marco de trabajo frente al desarrollo de software:

- **Planificación:** En esta fase se elaborará un plan de pruebas teniendo como referencia el *Product Backlog* y todo lo que se requiera enfocado a las pruebas y basados en los requerimientos previamente establecidos por el usuario y el product owner u otros que hagan de ese rol.
- **Monitoreo y Control:** En esta fase se realiza un monitoreo con respecto a la situación real y lo previamente establecido en la fase de la planificación (plan de pruebas), y el control correspondiente para la toma de decisiones para lograr lo planificado, esta fase se realizará en todo el ciclo de vida del *proyecto (todos los sprints existentes)*.
- **Análisis:** En esta fase se realiza la especificación de qué es lo que se va a realizar y la cobertura correspondiente para cada caso de prueba, esta fase se realizará en cada **sprint** que pudiera existir en el proyecto hasta su finalización, se tienen en cuenta aspectos como las especificaciones de requisitos funcionales, casos de uso, épicas, historias de usuario y otra documentación que sea necesaria.
- **Diseño:** En esta fase se tiene en cuenta el cómo se va a realizar la prueba, aspectos como los datos requeridos como condiciones de pruebas, diseñar el inicio de las pruebas, priorización de casos, identificar herramientas e infraestructura requerida según sea el caso, esta fase se realizará en cada **sprint**.

- **Implementación:** En esta fase se realiza la preparación de datos de pruebas, entornos de pruebas que fueran necesarios como la virtualización de servicios, infraestructura y toda la configuración requerida para dejar el ambiente de pruebas listo antes de su ejecución, esta fase se realizará en cada **sprint**.
- **Ejecución:** En esta fase se realizan las acciones después de haber realizado la implementación de las pruebas, obteniendo resultado de esta misma, aquí se realiza la comparación de los resultados esperados y los reales, análisis de anomalías existentes, reporte de defectos, registro de ejecución, esta fase se realizará en cada **sprint**.
- **Compleción:** En esta fase se recopila toda la información correspondiente, obtenida previamente de la ejecución de pruebas, la cual sirve como fuente de aprendizaje, mejora e historial de información en una publicación o final del proyecto, esta fase se realizará en cada **sprint**.

3.5.1 Técnicas de análisis

Para esta investigación se va a utilizar el análisis de la situación bajo el instrumento de la guía de encuesta al gerente general y un cuestionario a todo el equipo de desarrollo/calidad y a los usuarios quienes son los clientes, por tanto las respuestas nos ayudarán a entender y evaluar el problema dentro de la presente investigación.

Algunas herramientas que ayudarán al análisis serán una Matriz FODA, diagrama Ishikawa, entre otros, los cuales serán seleccionados según la información obtenida.

- Se usará el instrumento de la guía de entrevista para ver la situación actual de la empresa con respecto a la calidad de software dentro del equipo de desarrollo y calidad, así establecer un punto de partida y los objetivos necesario dentro de la investigación.
- Se usará el instrumento del cuestionario para ver la situación actual del proyecto que están desarrollando, los resultados obtenidos estarán bajo las métricas de la ISO 25010, los cuales son las subcaracterísticas pertenecientes a las características de la ISO mencionada.

Para el procesamiento de la información según la información obtenida de las dos etapas de preprueba y posprueba se tendrá en cuenta el enfoque estadístico descriptivo como los gráficos, frecuencias, media, desviación estándar, entre otros. Estos gráficos son de gran aporte porque representan con mayor facilidad los datos obtenidos, el entendimiento mediante la observación.

Toda la información recolectada será procesada por software con enfoque hacia las estadísticas y gráficos que permitan una mejor perspectiva como MS Excel 2013 e IBM SPSS Statistics.

3.5.2 Prueba de hipótesis

Para la prueba de hipótesis se realizará una prueba antes de realizar algún cambio dentro de nuestro contexto de investigación para obtener la información necesario sobre mediante los instrumentos como la guía de entrevista y el cuestionario con la muestra especificada anteriormente.

Después de realizar la recolección de la información mediante los instrumentos como guía de entrevista y cuestionario en base a los cambios que se realizarán en nuestro contexto de investigación, y se efectuará un contraste de la hipótesis general de nuestra investigación, con estas dos etapas de preprueba y posprueba.

La prueba de Hipótesis será realizada mediante T-Student tipo paramétrico si es que los datos cumplen con los requisitos previstos en caso contrario la prueba no paramétrica de Willcoxon.

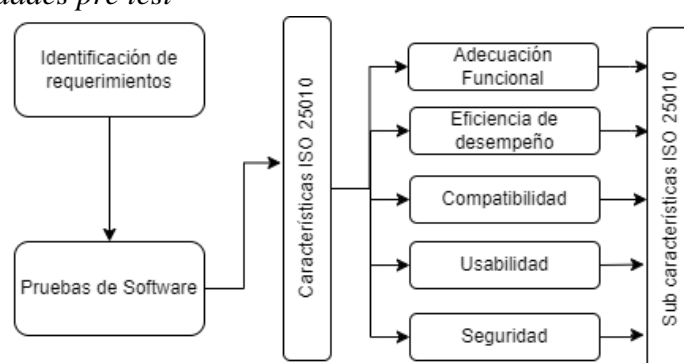
IV. RESULTADOS DE LA INVESTIGACIÓN

4.1 Descripción del trabajo de campo

Dentro de las actividades del trabajo de campo se engloban los 2 objetivos descritos en la presente investigación, los cuales fueron:

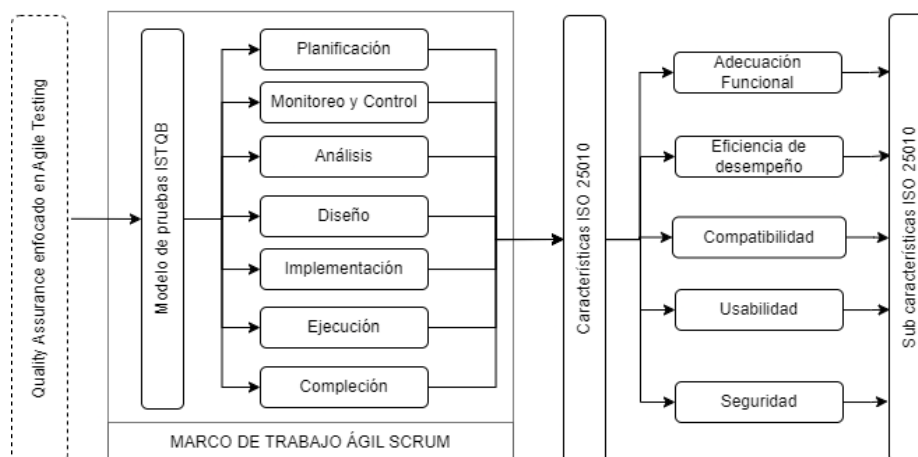
- El análisis de la situación actual (pre test) donde se tiene el enfoque de analizar la situación de la calidad de software antes de la implementación del modelo de pruebas según ISTQB, donde se realizó la identificación de los requerimientos y las pruebas de software acorde a las características de la ISO 25010.

Figura 11:
Actividades pre test



- Por otro lado el otro objetivo fue el análisis de la situación actual (post test) donde se realizó la implementación del modelo del proceso de pruebas ISTQB dentro del marco de trabajo ágil Scrum teniendo en cuenta las diversas fases dentro del modelo y las pruebas de software, las cuales también fueron incluidas del proceso de software y que se realizaron acorde a las características de la ISO 25010.

Figura 12:
Actividades post test



4.1.1 ANÁLISIS DE LA SITUACIÓN ACTUAL – PRE TEST

4.1.1.1 Identificación de Requerimientos

Como estrategia de esta investigación se tuvo la identificación de la situación real dentro de la empresa con respecto a los proyectos en general, el desarrollo y lo más importante para nuestra investigación la calidad de software y su influencia en los proyectos que realizaron, donde se tuvo contacto mediante video llamada con el gerente general, el señor William Castro Grijalva de la empresa que fue contexto de investigación y a partir de esa base de conocimiento brindada por su parte, se identificaron los requerimientos para la inclusión de Quality assurance enfocado en agile testing mediante el modelo de pruebas basado en ISTQB dentro de la empresa y para los presentes y futuros proyectos.

Con respecto a la empresa Wasi Innovation que cuenta relativamente con poco tiempo brindando servicios, específicamente desde el 2020 y su principal foco son los proyectos que requieren de servicios de desarrollo de software por parte del estado peruano.

Dentro de los proyectos realizados se mencionó que se tuvieron problemas con respecto a la calidad de software, los cuales posiblemente fueron causados por no tener una adecuada organización en las pruebas de software, en enfoque de las pruebas desde el inicio del proyecto, la implementación de los diversos tipos y niveles de pruebas, la poca sinergia con los clientes que en su mayoría eran entes del estado que trabajan mediante metodologías clásicas como Waterfall o RUP, donde se tiene un supuesto de todos los requerimientos establecidos y que son otorgados a las empresas desarrolladoras de software pero en el transcurso van apareciendo otros los cuales no fueron estipulados, obligando a las a adecuarse independiente del daño causado en tiempo, desarrollo, costos, entre otros, y por contraparte cuando suscitan cambios por parte de la empresa productora de software, el ente estatal es reactivo a dichos cambios. Algo que resaltar fueron los Términos de referencia (TDR), que son documentos genéricos donde se establecen objetivos y algunos requerimientos no detallados que luego forman base para el desarrollo, teniendo este aspecto como un punto negativo ya que es causa

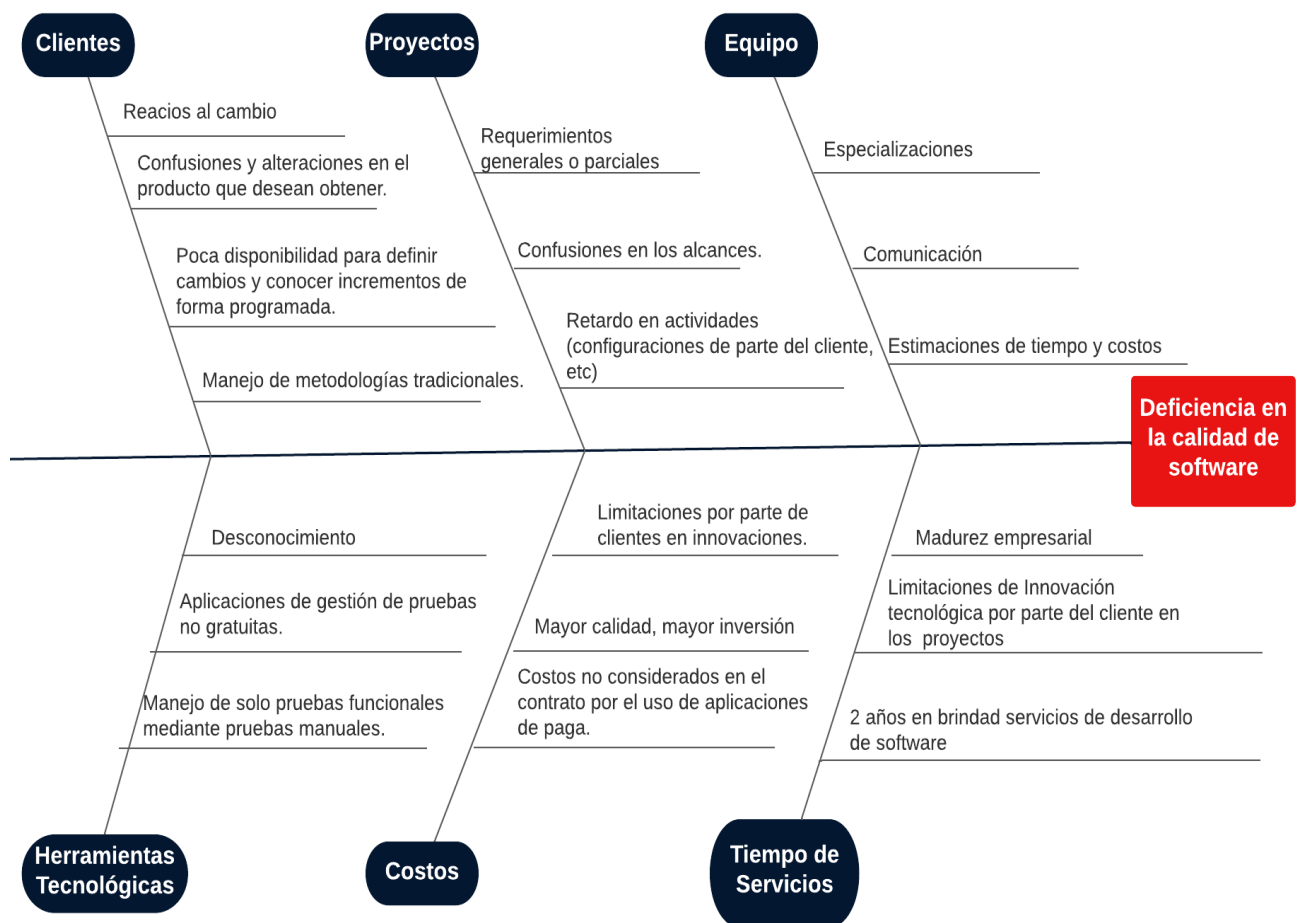
de confusiones y vulneraciones a la calidad de software en la fase de desarrollo y despliegue.

La aplicación que utilizan es el Clickup con un enfoque hacia el desarrollo, esto debido a la adecuación en los proyectos a los que se presentan porque el valor monetario agregado para las aplicaciones extra no son tomadas en cuenta por parte de los clientes y esto genera un mayor gasto por parte de la empresa producto que es Wasi Innovation.

También cuentan con un enfoque ágil y han realizado la inclusión del marco de trabajo Scrum de forma parcial, que fue enfocado en su mayoría al desarrollo de software.

Las principales actividades que realizan para asegurar la calidad de software son las pruebas funcionales, de caja de negra o de interfaz de usuario como se conoce, realizado mediante las pruebas manuales de software.

Figura 13:
Análisis Causa-Efecto



Nota: Diagrama Ishikawa como estrategia de análisis de la deficiencia en la calidad de software

De acuerdo al análisis de la situación actual antes de la implementación del modelo se obtuvieron una serie de requerimientos en base a los cuales se crearon estrategias para mejorar la calidad de software con el modelo de pruebas según ISTQB y todas las actividades extras con el fin de cumplir el objetivo.

Tabla 4:

Requerimientos para mejorar la calidad de software

N°	DESCRIPCIÓN DE REQUERIMIENTOS
REQ_01	Incluir estrategia de mejora en el establecimiento de tareas (requisitos de desarrollo y otros).
REQ_02	Crear plan de pruebas
REQ_03	Crear estrategia de monitoreo y control de pruebas
REQ_04	Creación de estrategia de análisis de pruebas
REQ_05	Crear el diseño de pruebas
REQ_06	Crear una estrategia de pruebas
REQ_07	Implementar las pruebas
REQ_08	Ejecutar las pruebas
REQ_09	Crear el Reporte de pruebas mediante la documentación requerida
REQ_10	Realizar reuniones como medida de capacitación en temas de mejora para la calidad de software, actividades, entre otros.
REQ_11	Realizar diagrama de flujo para las actividades para las pruebas
REQ_12	Implementar pruebas automatizadas
REQ_13	Implementar pruebas de performance
REQ_14	Implementar pruebas de estructura
REQ_15	Implementar BDD
REQ_16	Incluir niveles de pruebas
REQ_17	Incluir el modelo de pruebas con el marco de trabajo ágil Scrum.

Nota: Requerimientos establecidos en base al análisis situacional de la calidad de software en el contexto de estudio como estrategia de mejora.

Descripción general del software en desarrollo en la empresa contexto de la investigación

Para los fines de la investigación se tuvo presente el proyecto que estaba vigente en ese momento, el cual era un proyecto con enfoque e-learning por parte de una entidad estatal con el objetivo de poder brindar educación para los diversos profesionales como becados a dichas capacitaciones.

Tabla 5:

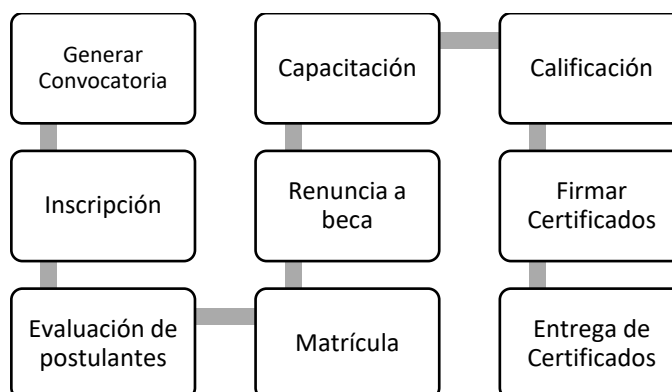
Especificaciones del Término de Referencias del proyecto de software

Términos de Referencia	
Objeto de Contratación	Desarrollar una aplicación web como herramienta para gestionar los procesos de becas de los cursos de capacitación, matrícula, certificados.
	Mediante esta aplicación se mejorarán los servicios que se presta a los profesionales en diversos ámbitos como salud, educación y comunitario.
Finalidad	Gestionar el otorgamiento de las becas para cursos de capacitación, matrícula, reportes y emisión de certificados como parte del programa.
Características Generales	Diseño moderno, fácil de usar por diferentes usuarios internos y externos considerados.
	La aplicación web tendrá colores acordes a la institución. Detalle de características funcionales, arquitectura, entre otros.

Nota: Términos de Referencia como base para el inicio de actividades del proyecto en ese momento vigente.

Figura 14:

Principales procesos del software en desarrollo:



Dentro del proyecto en desarrollo se especificaron todos los apartados (páginas) como parte del producto requerido por el cliente, los cuales fueron:

Tabla 6:

Apartados del Portal en desarrollo

Apartado / página	Descripción
Inicio	Página principal del portal
Nosotros	Información sobre la entidad cliente
Convocatorias	Listado de todas las convocatorias en sus diversas modalidades para la inscripción de los postulantes.
Becados	Listado según filtrado de año, ámbito y el curso de todos los becados en forma de historial.
Reglamento	Información acerca del programa de formación como misión, visión, inscripciones, participantes, modalidad, entre otros.
Eventos	Listado de eventos creados por la entidad cliente.
Certificados	Cursos Convocatorias Tutores

	Firmantes
	Gestores
	Matricula
	Notas
	Certificados
Reportes	Consultas y reportes
	Estadísticas
Configuraciones Generales	Parámetros, Instituciones, Modalidades, Banners, Accesos directos, Logos, Nosotros, Eventos, Documentos y videos.
Mesa de Ayuda y Contacto	Información sobre manuales, preguntas frecuentes, videotutoriales, otros.

Nota: Listado de apartados del proyecto en desarrollo en el cual se realizaron las diversas pruebas.

Los roles y/o tipos de usuario que acceden al software en desarrollo son:

Tabla 7:

Roles/Usuarios del software en desarrollo

Administrador
Firmante
Gestor
Postulante
Tutor

Aspectos técnicos en el desarrollo del software

Tabla 8:

Herramientas Tecnológicas

	Angular 12
	Spring Boot
	Spring JPA
	Maven
Tech	GitLab
Stack/Tools	JBoss
	Oracle 11g
	Java 1.8
	War deploy
	WildFly Server

*Nota: conjunto de elementos tecnológicos mediante
El cual se desarrolla el software.*

Para mayor conocimientos de los apartados/páginas a continuación se muestra todo lo concerniente al portal en desarrollo con el fin de entregar el productor requerido por el cliente.

Interfaces gráficas del software en desarrollo

Figura 15:

Inicio

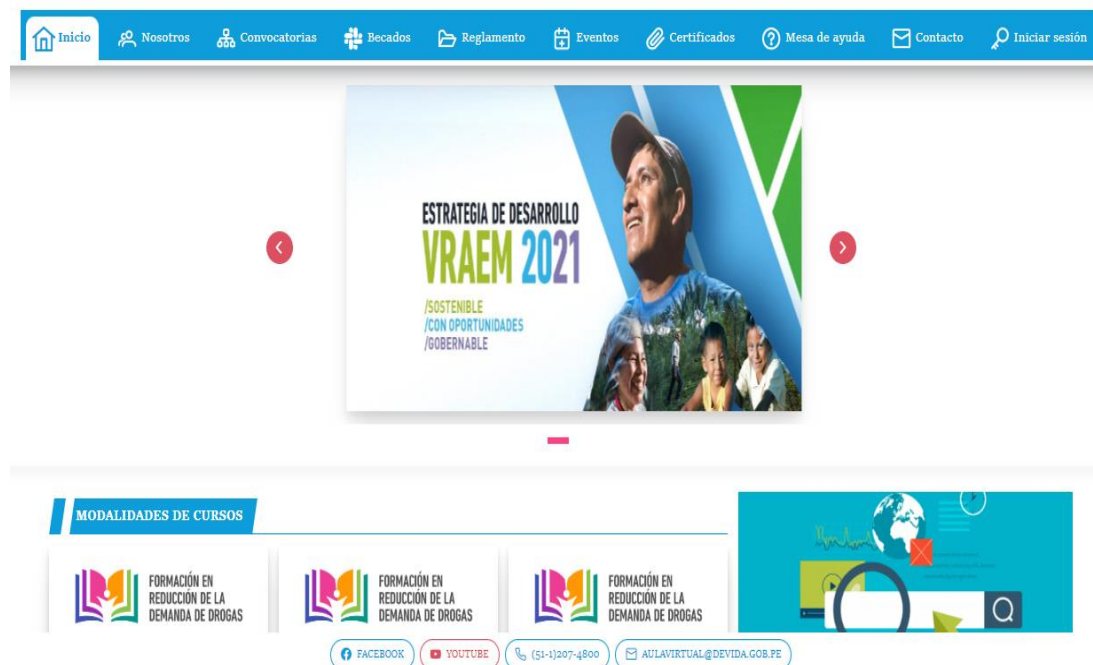


Figura 16:

Nosotros



Figura 17:
Convocatorias

DEVIDA
Comisión Nacional para el Desarrollo y Vida sin Drogas

Inicio Nosotros Convocatorias Becados Reglamento Eventos Certificados Mesa de ayuda Contacto Iniciar sesión

FILTROS

Autoaprendizaje Virtual Semipresencial Presencial Mixta

CONVOCATORIAS ABIERTAS

No se encontraron convocatorias abiertas.

CONVOCATORIAS CERRADAS

DV001 - 1 - 2022 Habilidades Socioemocionales y su Relación con el Consumo de Drogas en el Ámbito Educativo

Inscripción: 21/10/2022 - 22/10/2022

Ámbito: SALUD

Modalidad: Autoaprendizaje

Profesiones u ocupaciones: Administrador Administrativo Almacenero Archivero Arquitecto Autor Bibliotecario Biólogo

Cargos afines: Digitador Coordinador de área Oficinista Apoyo GESTOR DE ÁREA Coordinador Apoyo administrativo ADMINISTRADOR JEFE ENCARGADO DE EQUIPO Asistente administrativo

Etapa
PUBLICACIÓN RESULTADOS
LEER MÁS →

PRÓXIMAS CONVOCATORIAS

FACEBOOK YOUTUBE (51-1)207-4800 AULAVIRTUAL@DEVIDA.GOB.PE

PROGRAMA DE FORMACIÓN VIRTUAL - DEVIDA © 2022

Figura 18:
Becados

DEVIDA
Comisión Nacional para el Desarrollo y Vida sin Drogas

Inicio Nosotros Convocatorias Becados Reglamento Eventos Certificados Mesa de ayuda Contacto Iniciar sesión

Becados

(*) Año 2021 Ambito EDUCACION Curso

FILTRAR LIMPIAR

EXPORTAR 4 ITEMS SELECCIONADOS

Nº	CÓDIGO CURSO	CURSO	APELLIDOS Y NOMBRE	INSTITUCIÓN
1	DV001	DV001 Habilidades Socioemocionales y su Relación con el Consumo de Drogas en el Ámbito Educativo	ACUÑA BUSTAMANTE MARISOL	DIRECCIÓN DE RED DE SALUD LIMA ESTE METROPOLITANA (DRSLEM)
2	DV001	DV001 Habilidades Socioemocionales y su Relación con el Consumo de Drogas en el Ámbito Educativo	ALBINO MENESES JIMMY ANDRÉS	UNIDAD DE GESTION EDUCATIVA LOCAL - BONGARA (UGEL - BONGARA)
3	DV001	DV001 Habilidades Socioemocionales y su Relación con el Consumo de Drogas en el Ámbito Educativo	ALMONACID LOVÓN MÉRILYN JOHANA	Hospital Regional Eleazar Guzmán Barrón (HEGB)
4	DV001	DV001 Habilidades Socioemocionales y su Relación con el Consumo de Drogas en el Ámbito Educativo	ALVARADO SEPULVEDA LUIS MIGUEL	RED DE SALUD HUAYLAS NORTE (RSHN)
5	DV001	DV001 Habilidades Socioemocionales y su Relación con el Consumo de Drogas en el Ámbito Educativo	ANTIQUERA NINA BANDALIA SARAI	UNIDAD DE GESTION EDUCATIVA LOCAL SATIPO (UGEL-S)
6	DV001	DV001 Habilidades Socioemocionales y su Relación con el Consumo de Drogas en el Ámbito Educativo	AROSTEGUI GUTIÉRREZ LUIS ALBERTO	UNIDAD DE GESTIÓN EDUCATIVA LOCAL TARMIA (UGEL TARMIA)

FACEBOOK YOUTUBE (51-1)207-4800 AULAVIRTUAL@DEVIDA.GOB.PE

PROGRAMA DE FORMACIÓN VIRTUAL - DEVIDA © 2022

Figura 19:

Eventos

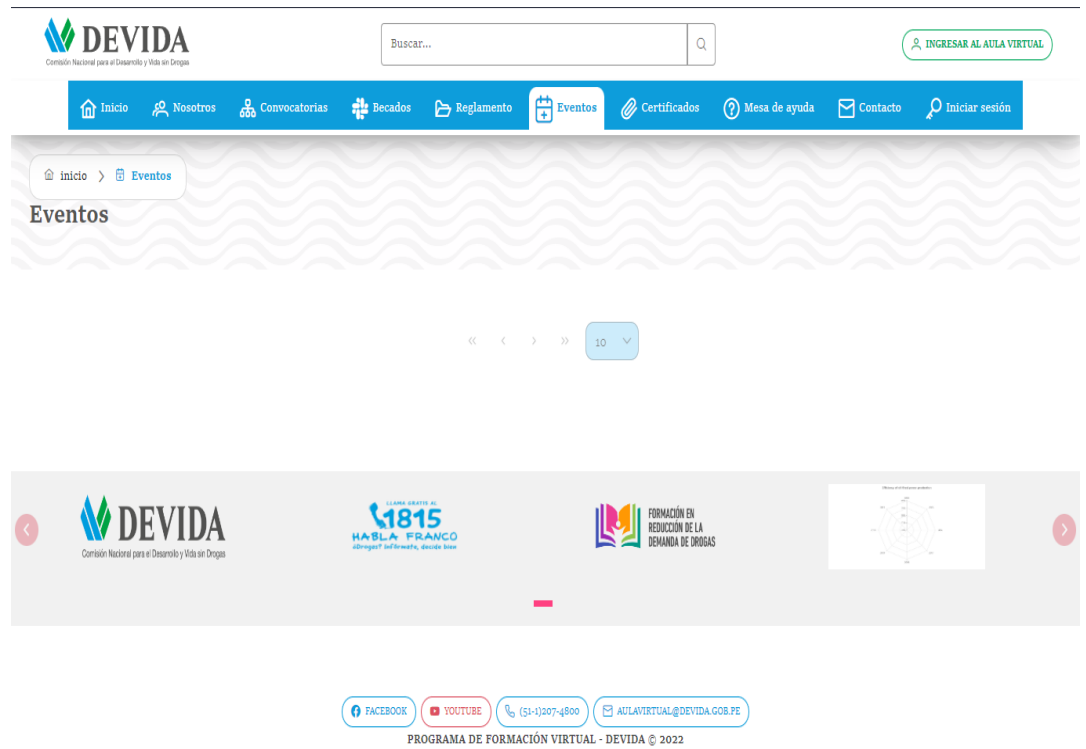


Figura 20:

Certificaciones

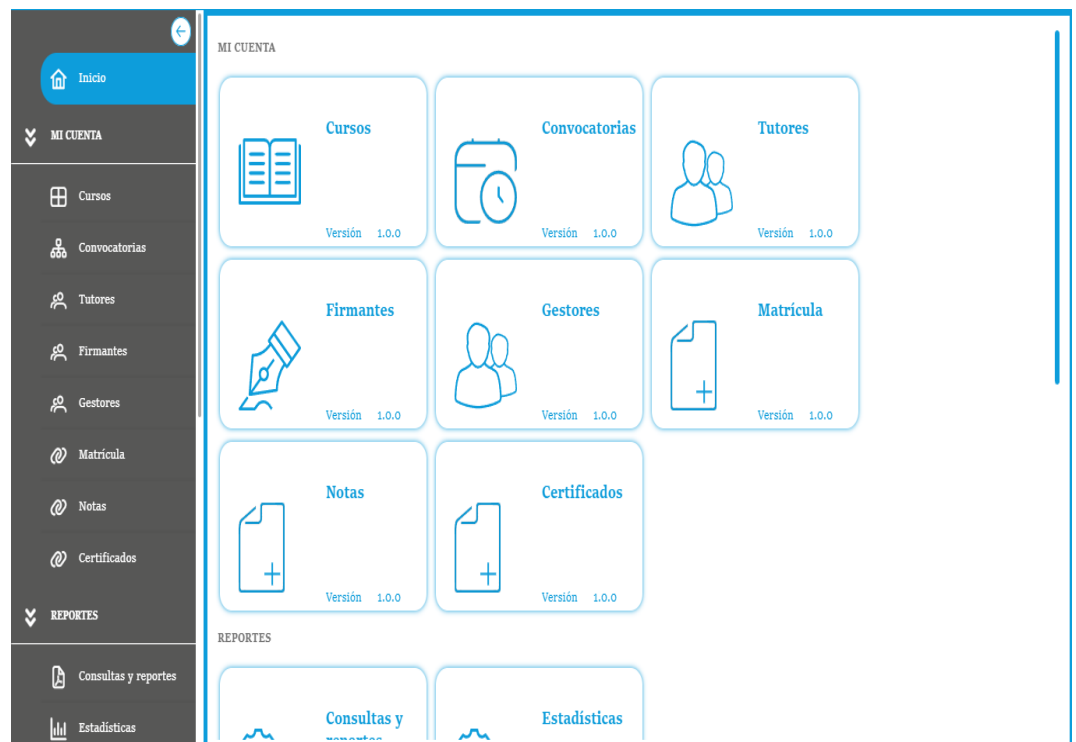


Figura 21:

Cursos

DEVIDA

IR AL PORTAL MALENA JESÚS MT

CRITERIOS DE BÚSQUEDA

Ambito Nombre

FILTRAR LIMPIAR

LISTADO DE CURSOS

NUEVO EXPORTAR 5 ITEMS SELECCIONADOS

Nº	OPCIONES	CÓDIGO	EDICIÓN	NOMBRE	ÁMBITO	FECHA DE CREACIÓN
1		DV004	1	EDICIÓN - DV004 - Documento Técnico de Orientación, Consejería e Intervención Breve para Consumidores de Sustancias Psicoactivas	COMUNITARIO	25/10/2022
2		DV003	2	EDICIÓN - DV003 - Facilitadores educativos para la prevención del consumo de drogas y otros riesgos sociales en el ámbito educativo	EDUCACIÓN	25/10/2022
3		DV003	1	EDICIÓN - DV003 - Facilitadores educativos para la prevención del consumo de drogas y otros riesgos sociales en el ámbito educativo	EDUCACIÓN	25/10/2022
4		DV002	1	EDICIÓN - DV002 - Técnicas y Herramientas para el Facilitador Comunitario en Prevención del Consumo de Drogas	SALUD	24/10/2022
5		DV001	1	EDICIÓN - DV001 - Habilidades Socioemocionales y su Relación con el Consumo de Drogas en el Ámbito Educativo	COMUNITARIO	21/10/2022

Mostrando 1 a 5 de 5 items

Figura 22:

Gestión de Convocatorias

DEVIDA

IR AL PORTAL MALENA JESÚS MT

CRITERIOS DE BÚSQUEDA

Modalidad Ambito Estado Etapa

FILTRAR LIMPIAR

LISTADO DE CONVOCATORIAS

NUEVO EXPORTAR 8 ITEMS SELECCIONADOS

Nº	OPCIONES	ESTADO	ETAPA	AÑO	BLOQUE	DESCRIPCIÓN	MODALIDAD	ÁMBITO	FECHA CREACIÓN
1		CERRADA	EVALUACIÓN	2022	1	DV002 - 1 - 2022 Técnicas y Herramientas para el Facilitador Comunitario en Prevención del Consumo de Drogas	Mixta	EDUCACIÓN	24/10/2022
2		CERRADA	PUBLICACIÓN RESULTADOS	2022	1	DV001 - 1 - 2022 Habilidades Socioemocionales y su Relación con el Consumo de Drogas en el Ámbito Educativo	Autoaprendizaje	SALUD	21/10/2022
3		CERRADA	CERRADO	2021	2	DV003-2-2021 Facilitadores educativos para la prevención del consumo de drogas y otros riesgos sociales en el ámbito educativo	Virtual	COMUNITARIO	

Mostrando 1 a 3 de 3 items

Figura 23:

Tutores

DEVIDA

IR AL PORTAL MALENA JESÚS MT













CRITERIOS DE BÚSQUEDA

Estado

FILTRAR LIMPIAR

LISTADO DE TUTORES

NUEVO EXPORTAR 7 ITEMS SELECCIONADOS

Nº	OPCIONES	TIPO DOC.	Nº DOC.	NOMBRES	APELLIDO PATERNO	APELLIDO MATERNO	CELULAR	EMAIL
1	   	DNI	40497209	FERNANDO	CACERES	GONZALES		tutor_3@yopmail.com
2	   	DNI	40589743	SAMIEL	SALINAS	ESPINOZA		tutor_2@yopmail.com
3	   	DNI	44969413	OMAR	ESPINOZA	FIGEROA		tutor_1@yopmail.com

Mostrando 1 a 3 de 3 items

Figura 24:

Firmantes

DEVIDA

IR AL PORTAL MALENA JESÚS MT













CRITERIOS DE BÚSQUEDA

Estado

FILTRAR LIMPIAR

LISTADO DE FIRMANTES

NUEVO EXPORTAR 7 ITEMS SELECCIONADOS

Nº	OPCIONES	TIPO DOC.	Nº DOC.	NOMBRES	APELLIDO PATERNO	APELLIDO MATERNO	CELULAR	EMAIL
1	   	DNI	12131415	PEDRO ALBERTO	PEREZ	VELEZ		firmante_3@yopmail.com
2	   	DNI	14151617	ERICKA	CANDIO	LÓPEZ		firmante_2@yopmail.com
3	   	DNI	16171819	RICARDO ANTONIO	SOBERÓN	GARRIDO		firmante_1@yopmail.com

Mostrando 1 a 3 de 3 items

Figura 25:

Gestores

DEVIDA
Código de Matrícula y Expediente - 10000000000000000000

IR AL PORTAL MALENA JESÚS MT

CRITERIOS DE BÚSQUEDA
Estado

FILTRAR LIMPIAR

LISTADO DE GESTORES

NUEVO EXPORTAR 7 ITEMS SELECCIONADOS

N°	OPCIONES	TIPO DOC.	N° DOC.	NOMBRES	APELLIDO PATERNO	APELLIDO MATERNO	CELULAR	EMAIL
1		Otros	456464646546546	CARMEN	JARAMILLO	SANCHEZ		gestor01@yopmail.com
2		DNI	09548077	LUIS SANTIAGO	ESPINOZA	PAUL		gestor_2@yopmail.com
3		DNI	45841676	ROXANA	LAIME	VICENTE		gestor_3@yopmail.com
4		DNI	32129444	WALTER	MOSCOL	MARIÑO		gestor_1@yopmail.com

Mostrando 1 a 4 de 4 items

Figura 26:

Matrícula

DEVIDA
Código de Matrícula y Expediente - 10000000000000000000

IR AL PORTAL MALENA JESÚS MT

CRITERIOS DE BÚSQUEDA
(*) Año: 2021 X
(*) Convocatoria: DV003-2-2021 Facilitadores educativos para la prevención del consumo de drogas y otros riesgos sociales en el ámbito... X

FILTRAR LIMPIAR

LISTADO MATRÍCULA

SINCRONIZA MOODLE MATRICULAR NOTIFICAR EXPORTAR 5 ITEMS SELECCIONADOS

N°	OPCIONES	GRUPO SECCIÓN	TIPO DOC.	N° DOC.	APELLIDOS NOMBRES	ESTADO
1		GRUPO SECCIÓN 1	DNI	09490038	ACUÑA BUSTAMANTE MARISOL	ACTIVO
2		GRUPO SECCIÓN 1	DNI	15440603	ALBINO MENESES JIMMY ANDRÉS	ACTIVO
3		GRUPO SECCIÓN 1	DNI	75809035	ALMONACID LOVÓN MERILYN JOHANA	ACTIVO
4		GRUPO SECCIÓN 1	DNI	45560663	ALVARADO SEPULVEDA LUIS MIGUEL	DESERTÓ
5		GRUPO SECCIÓN 1	DNI	00312648	ANTIQUERA NINA BANDALIA SARAI	DESERTÓ
6		GRUPO SECCIÓN 1	DNI	44277965	ACOSTA JULIO CÉSAR	ACTIVO
7		GRUPO SECCIÓN 1	DNI	29672225	ACOSTA ZEA CLAUDIA HANNY	ACTIVO
8		GRUPO SECCIÓN 1	DNI	41611372	AROSTEGUI GUTIÉRREZ LUIS ALBERTO	ACTIVO
9		GRUPO SECCIÓN 1	DNI	09940136	ATALA TICERAN CLELIA ALIONKA	DESERTÓ
10		GRUPO SECCIÓN 1	DNI	10111970	AYVAR BUENDIA JULIO CESAR	DESERTÓ
11		GRUPO SECCIÓN 1	DNI	16631360	BARBOZA ZELADA YMBER ANTONIO	ACTIVO
12		GRUPO SECCIÓN 1	DNI	00486017	BARBIENTOS RIOS NANCY JUANA DE JESUS	ACTIVO
13		GRUPO SECCIÓN 1	DNI	41605286	BEDOYA DOMÍNGUEZ PAOLO	ACTIVO

Figura 27:

Notas

DEVIDA

IR AL PORTAL MALENA JESÚS MT

CRITERIOS DE BÚSQUEDA

(*) Año: 2021 (X) (*) Convocatoria: DV003-2-2021 Facilitadores educativos para la prevención del consumo de drogas y otros riesgos sociales en el ambit... (X)

(*) Grupo Sección: GRUPO SECCION 1 (X) Estado: []

Nota:
*Se recomienda realizar la sincronización de notas al llevar a cabo cualquier modificación en el moodle.
*Considerar que el filtro estado no se toma en cuenta para realizar la sincronización con el moodle.

FILTRAR LIMPIAR

LISTADO DE ALUMNO NOTA

INGRESO MANUAL EVALUACIONES EXPORTAR EXPORTAR 7 ITEMS SELECCIONADOS

Nº	OPCIÓN GRUPO SECCIÓN	TIPO DOC.	Nº DOC.	APELLIDOS NOMBRES	PROMEDIO FINAL	CONDICIÓN	MODO INGRESO NOTA
1	GRUPO SECCIÓN 1	DNI	44277965	ACOSTA JULIO CÉSAR	18	APROBADO	MANUAL
2	GRUPO SECCIÓN 1	DNI	29672225	ACOSTA ZEA CLAUDIA HANNY	17	APROBADO	MANUAL
3	GRUPO SECCIÓN 1	DNI	09490038	ACUÑA BUSTAMANTE MARISOL	19	APROBADO	MANUAL
4	GRUPO SECCIÓN 1	DNI	15440603	ALBINO MENESES JIMMY ANDRÉS	4	DESAPROBADO	MANUAL
5	GRUPO SECCIÓN 1	DNI	75809035	ALMONACID LOVÓN MÉRILYN JOHANA	19	APROBADO	MANUAL
6	GRUPO SECCIÓN 1	DNI	45560663	ALVARADO SEPULVEDA LUIS MIGUEL		DESERTO	MANUAL
7	GRUPO SECCIÓN 1	DNI	00512648	ANTIQUERA NINA BANDALIA SARAI		DESERTO	MANUAL
8	GRUPO SECCIÓN 1	DNI	41611372	AROSTEGUI GUTIÉRREZ LUIS ALBERTO	17	APROBADO	MANUAL
9	GRUPO SECCIÓN 1	DNI	09940136	ATALA TICERAN CLELIA ALIONKA		DESERTO	MANUAL
10	GRUPO SECCIÓN 1	DNI	10111970	AYVAR BUENDIA JULIO CESAR		DESERTO	MANUAL

Figura 28:

Gestión Certificados

DEVIDA

IR AL PORTAL MALENA JESÚS MT

CRITERIOS DE BÚSQUEDA

(*) Año: 2022 (X) (*) Convocatoria: DV001-1-2022 Habilidades Socioemocionales y su Relación con el Consumo de Drogas en el Ambito Ed... (X) Sección: []

FILTRAR LIMPIAR

LISTADO DE CERTIFICADOS

CONFIGURAR PLANTILLA DESCARGAR CERTIFICADOS NOTIFICAR FIRMANTES EXPORTAR 5 ITEMS SELECCIONADOS

Nº	OPCIONES	GRUPO SECCIÓN	Nº DOCUMENTO	APELLIDOS Y NOMBRES	PROMEDIO FINAL	CÓDIGO CERTIFICADO
Mostrando 0 a 0 de 0 ítems						
No se encontraron datos.						

Figura 29:
Consultas y Reportes

DEVIDA
Unidad de Medición y Evaluación de la Calidad Educativa

IR AL PORTAL MALENA JESÚS MT

Tutores
Firmantes
Gestores
Matrícula
Notas
Certificados

REPORTES

Consultas y reportes
Estadísticas

CONFIGURACIONES GENERALES

Parámetros
Instituciones
Modalidades
Banners

CRITERIOS DE BÚSQUEDA

Año: [v] Ambito: [v] curso: [v]
Bloque: [v] Fecha inscripción desde: [] Fecha inscripción hasta: []
Tipo filtro postulante: [v] Postulante: []

FILTRAR LIMPIAR

LISTADO DE INSCRITOS: 85 EXPORTAR EXPORTAR 27 ITEMS SELECCIONADOS

N°	AÑO	AMBITO	CURSO	BLOQUE	APELLIDO PATERNO	APELLIDO MATERNO	NOMBRES	TIPO DOCUME
1	2021	EDUCACIÓN	DV003 Facilitadores educativos para la prevención del consumo de drogas y otros riesgos sociales en el ambito educativo	DV003-2-2021	ACUÑA	BUSTAMANTE	MARISOL	DNI
2	2021	EDUCACIÓN	DV003 Facilitadores educativos para la prevención del consumo de drogas y otros riesgos sociales en el ambito educativo	DV003-2-2021	ALBINO	MENESES	JIMMY ANDRÉS	DNI
3	2021	EDUCACIÓN	DV003 Facilitadores educativos para la prevención del consumo de drogas y otros riesgos sociales en el ambito educativo	DV003-2-2021	ALMONACID	LOVÓN	MÉRILYN JOHANA	DNI
4	2021	EDUCACIÓN	DV003 Facilitadores educativos para la prevención del consumo de drogas y otros riesgos sociales en el ambito educativo	DV003-2-2021	ALVARADO	SEPULVEDA	LUIS MIGUEL	DNI
5	2021	EDUCACIÓN	DV003 Facilitadores educativos para la prevención del consumo de drogas y otros riesgos sociales en el ambito educativo	DV003-2-2021	ANTIQUERA	NINA	BANDALIA SARAI	DNI
6	2021	EDUCACIÓN	DV003 Facilitadores educativos para la prevención del consumo de drogas y otros riesgos sociales en el ambito educativo	DV003-2-2021	AROSTEGUI	GUTIÉRREZ	LUIS ALBERTO	DNI

Figura 30:
Estadísticas

DEVIDA
Unidad de Medición y Evaluación de la Calidad Educativa

IR AL PORTAL MALENA JESÚS MT

Tutores
Firmantes
Gestores
Matrícula
Notas
Certificados

REPORTES

Consultas y reportes
Estadísticas

CONFIGURACIONES GENERALES

Parámetros
Instituciones
Modalidades
Banners

Selección de reporte

E1. GENERAL POR AÑO
E2. GENERAL MENSUAL
E3. CUADRO POR AÑO
E4. CUADRO POR CURSO
E5. APROBADOS POR REGIÓN
E6. APROBADOS POR GÉNERO
E7. APROBADOS POR SECTOR
E8. APROBADOS POR PROFESIÓN
E9. APROBADOS POR CARGO
E10. APROBADOS POR DISCAPACIDAD
E11. APROBADOS POR CURSOS LLEVADOS
E12. APROBADOS POR PROMEDIO

PLANIFICACIÓN DE META

(*) Ambito: EDUCACION (*) Año: 2022

ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	ENERO	TOTAL
2	2	1	1	2	2	1	1	2	2	1	1	18

EXCEL PDF

Ámbitos: COMUNITARIO (*) Años: 2021 2022

AÑO	META FÍSICA	LOGRO DE META	%
2021	252	48	19,05%
2022	58	0	0%

Comparativo de meta y logro del 2021, 2022

AÑO	META FÍSICA	LOGRO DE META
2021	252	48
2022	58	0

Figura 31:
Instituciones

Figura 32:
Modalidades

4.1.1.2 Pruebas de software

Las pruebas de software se realizaron desde un enfoque práctico complementando el análisis situacional de la calidad de software dentro del contexto de investigación, para lo cual se tuvieron en cuenta los diversos tipos de pruebas con la finalidad de poder identificar dichas brechas.

- **Pruebas funcionales**

APARTADO	PROBLEMA	DESCRIPCIÓN
GESTIONAR CONVOCATORIAS	<i>No se logra verificar acción realizada al crear convocatoria.</i>	<i>No se muestra ningún mensaje de información que se ha realizado la acción correspondiente para la sección (obligatoriamente tenemos que presionar el “Botón Cancelar” que es contradictorio a la acción que estamos realizando).</i>
	<i>Sin valores limites en los campos para crear una convocatoria</i>	<i>Al modificar o crear una convocatoria no se está teniendo en cuenta los valores limites lógicos (consultar valores limites según campos).</i>
	<i>No permite eliminar una convocatoria</i>	<i>No está permitiendo eliminar una convocatoria en ningún estado (ya sea registrada, inscripción, publicado, etc).</i>
	<i>No permite editar la convocatoria</i>	<i>No está permitiendo editar una convocatoria en ningún estado.</i>
	<i>No se guarda la imagen de la convocatoria</i>	<i>Se visualiza la imagen al momento de cargar pero no se puede guardar</i>
CURSOS	<i>Error al seleccionar y quitar filtro de campos de la tabla cursos</i>	<i>Filtro para visualización de las columnas con los valores del curso cuando se selecciona uno o varios funciona con normalidad pero cuando se selecciona todo y luego se quita esa selección queda en blanco.</i>
	<i>Funcionalidad Campo Estado de Curso</i>	<i>En el apartado de Cursos se tiene el estado y en la visualización de la tabla también pero al momento de editar la</i>

		<i>información de un curso no se tiene esa opción de desactivar o activar.</i>
	<i>No se puede eliminar cursos</i>	<i>La eliminación de curso no está funcionando</i>
	<i>No se muestra encabezado del reporte en Excel de los cursos</i>	<i>El reporte de Excel de los cursos, se muestra un espacio en blanco arriba de la tabla de resultados.</i>
	<i>Error en límite de valores para la las horas lectivas no definida</i>	<i>Dentro del modal para crear o modificar el curso los valores no están establecidos, p.e. la Nota aprobatoria por lo general tiene una escala tipo 0-20, 0-100, etc. El mismo caso para las horas lectivas y porcentajes de calificación y producto.</i>
	<i>Error en límite de valores en el campo Nota Aprobatoria</i>	
TUTORES	<i>El campo fecha de nacimiento no validado</i>	<i>Modificar información del Tutor, el campo Fecha de nacimiento está permitiendo ingresar valores ilógicos.</i>
	<i>No existen mensajes de confirmación de acciones</i>	<i>Siguiendo el flujo de modificación para el tutor, no se observa un mensaje de información acorde a la acción realizada.</i>
DOCUMENTOS Y VIDEOS	<i>No se visualiza el video</i>	<i>Al momento de visualizar el archivo .mp4, se queda cargando (revisar ruta, ignorar si es por error de infraestructura).</i>
	<i>No se puede eliminar el video</i>	<i>Después de realizar una previa acción de inserción, no se puede eliminar el documento.</i>
	<i>No existen mensajes de confirmación de acciones</i>	<i>Siguiendo el flujo de modificación para el firmante, no se observa un mensaje de información acorde a la acción realizada.</i>
FIRMANTE	<i>Incoherencia de colores para activar y desactivar a los firmantes</i>	<i>Se debe tener en cuenta que el icono no representa el modo activo sino la acción (por el sentido de la flecha) que se realizará para desactivar al firmante.</i>
	<i>Error en la cabecera del reporte en Excel</i>	<i>El reporte no está saliendo acorde al formato especificado (se habló este tema en el daily – viernes 19/08/22)</i>

	<i>Error en edición del género del Firmante</i>	<i>Al modificar la información del Firmante en campo Género no está replicando en la tabla.</i>
	<i>Campo de fecha de nacimiento sin validar</i>	<i>Tener en cuenta la validación que se ha sugerido para la fecha de nacimiento y los valores límites permitidos.</i>
GESTIÓN CERTIFICADOS	<i>No funciona filtro de año</i>	<i>El combo para el filtro por año no está funcionando.</i>
	<i>No funciona filtro de Convocatoria</i>	<i>El combo para el filtro por Convocatoria no está funcionando.</i>
	<i>No funciona filtro de Sección</i>	<i>El combo para el filtro por Sección no está funcionando.</i>
	<i>No funciona botón para exportar como reporte en Excel</i>	<i>La funcionalidad para la exportación como reporte en Excel no realiza ninguna acción/no implementado.</i>
GESTORES	<i>No se visualiza el número del documento de identificación</i>	<i>Al momento de realizar la modificación de información del Gestor, no se visualiza el número de identificación en el modal, tampoco en la tabla de Gestores</i>
	<i>No se visualiza parámetros</i>	<i>No aparece por defecto todos los tipos de parámetros al iniciarse el apartado.</i>
PARÁMETROS	<i>Mensaje Incorrecto al crear un parámetro</i>	<ul style="list-style-type: none"> • <i>Cuando se agrega un parámetro no se muestra en el listado de la tabla (relacionado con el ítem anterior)</i> • <i>Y al realizar el registro muestra un mensaje de “editado de tutor”</i>
	<i>Mensaje incorrecto al eliminar un parámetro</i>	<i>Al momento de eliminar aparece un mensaje correspondiente al tutor “se ha eliminado el tutor”.</i>
	<i>El filtrado por Año no funciona.</i>	
MATRÍCULA	<i>El filtrado por Convocatoria no funciona.</i>	<i>No se muestra el response según el filtrado por Año, tampoco por Convocatoria, ni una combinación de ambos.</i>

MODALIDADES	<i>No se visualiza una nueva modalidad creada</i>	<i>Se crea una modalidad de forma correcta pero para crear una convocatoria se debería tener éste valor dentro del combo de modalidad en la convocatoria.</i>
		<i>Combo para filtrar por Departamento</i>
INSTITUCIÓN	<i>No sirve el filtro de búsqueda en los combos</i>	<i>Combo para filtrar por Provincia</i>
		<i>Combo para filtrar por Distrito</i>
		<i>Combo para filtrar por Nombre</i>
INICIO	<i>No se puede filtrar con el calendario</i>	<i>Al seleccionar un día del mes como filtro no funciona.</i>
	<i>No se puede filtrar mediante la caja de búsqueda</i>	<i>Al ingresar un texto no se está realizando la búsqueda.</i>
	<i>Resposive sizes 770px +</i>	<i>Se genera un doble icono en el header del portal</i>
	<i>Resposive sizes 880px +</i>	<i>Las imágenes se distorsionan en el intervalo de pixeles mencionados.</i>
	<i>Resposive sizes 990px +</i>	<ul style="list-style-type: none"> - <i>Las imágenes se distorsionan.</i> - <i>Los contenedores de Mesa de Ayuda e Iniciar sesión se distorsionan.</i>
	<i>Header de todas las pestañas - size 769px - 960px</i>	<i>Doble icono en header.</i>
	<i>Botón para aula virtual desaparece en 767px</i>	<i>El botón “Ingresar al aula virtual” desaparece desde 767px a menos.</i>
	<i>Botón para aula virtual desaparece en 300px</i>	<i>El botón “Ingresar al aula virtual” desaparece en los 300px a menos.</i>

LOGIN	<i>Deformación de botones</i>	<i>El tamaño de los contenedores de los botones se incrementa mucho más que los demás en el rango de 961px - 1112px (corroborar si es normal)</i>
	<i>Responsive Size 1110px</i>	<i>Los contenedores de Mesa de Ayuda e Iniciar sesión se distorsionan.</i>
GESTIÓN CERTIFICACIONES	<i>Responsive size 399px –</i>	<i>El contenedor de Ir al Portal y la información del usuario queda sobrepuesta por el contenedor de , curso, convocatorias, tutores, firmantes, certificaciones, instituciones, gestores, parámetros, reportes, inscripciones, matrícula, otros más</i>
CONTACTO	<i>No se visualizan los valores del combo Asunto</i>	<i>Al momento de realizar una petición para contactar por cualquier asunto no se está desplegando los valores del combo Asunto.</i>
MESA DE AYUDA	<i>Error al manipular el video de información</i>	<i>No se reproduce el video (automático y manipulado), los controles están desactivados.</i>
TODOS LOS APARTADOS	<i>Error de configuración de grillas de columna a fila de 961px a menor tamaño.</i>	<i>En todos los apartados se muestra que de 961px – (en forma descendente) no se tiene configurado para la visualización de las tablas y el orden cambia a columnas según los ítems de los apartados.</i>

Nota: Listado de problemas encontrados dentro de las pruebas funcionales realizadas.

- **Pruebas no funcionales**

Tabla 9:

Pruebas de Rendimiento - Caso N° 01

CASO 01	RESULTADOS
<i>Cuando se tienen 5 usuario y cada uno va ingresando 20 segundos después</i>	<i>/INICIO: Al ingresar al portal y cuando se realiza la petición de portal/inicio se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 1.993 segundos, un</i>

desde que ingresó el primer usuario y un intervalo de tiempo que nos ayuda a que las peticiones se realicen en un tiempo simulado a la realidad (para este caso un intervalo de 0 a 10 segundos por cada petición).

promedio de 2.899 segundos y un máximo de 5.362 segundos.

CURSO/INICIALIZAR-BUSQUEDA: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.292 segundos, un promedio de 0.811 segundos y un máximo de 1.596 segundos.

CONVOCATORIA/inicializar-

registro?idConvocatoria=85: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.274 segundos, un promedio de 0.962 segundos y un máximo de 1.307 segundos.

CONVOCATORIA/inicializar-

registro?idConvocatoria=84: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.573 segundos, un promedio de 1.103 segundos y un máximo de 1.592 segundos.

CONVOCATORIA/inicializar-

registro?idConvocatoria=83: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.790 segundos, un promedio de 1.120 segundos y un máximo de 1.551 segundos.

CONVOCATORIA/inicializar-

registro?idConvocatoria=81: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.422 segundos, un promedio de 0.795 segundos y un máximo de 1.210 segundos.

REPORTES/grafico/inicializar: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.293

segundos, un promedio de 1.020 segundos y un máximo de 1.615 segundos.

Tabla 10:

Pruebas de Rendimiento - Caso N° 02

CASO 02	RESULTADOS
Quando se tienen 15 usuarios y cada uno va ingresando 20 segundos después desde que ingresó el primer usuario.	<p>/LOGIN: Al ingresar al portal y cuando se realiza el login se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.256 segundos, un promedio de 1.515 segundos y un máximo de 7.312 segundos</p> <hr/> <p>/INICIO: Al ingresar al portal, en inicio se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 1.993 segundos, un promedio de 3.445 segundos y un máximo de 7.913 segundos</p> <hr/> <p>CONVOCATORIA/inicializar-registro?idConvocatoria=85: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.286 segundos, un promedio de 3.057 segundos y un máximo de 19.249 segundos.</p> <hr/> <p>CONVOCATORIA/inicializar-registro?idConvocatoria=84: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.423 segundos, un promedio de 3.195 segundos y un máximo de 20.281 segundos.</p> <hr/> <p>CONVOCATORIA/inicializar-registro?idConvocatoria=81: se observó se realizó una petición por cada usuario y se obtuvo que el</p>

tiempo mínimo de respuesta es 0.391 segundos, un promedio de 3.034 segundos y un máximo de 19.256 segundos.

TUTOR-LISTAR?IDESTADO=0: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.196 segundos, un promedio de 3.080 segundos y un máximo de 22.261 segundos.

TUTOR-INICIALIZAR-

REGISTRO?IDTUTOR=41: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.180 segundos, un promedio de 1.579 segundos y un máximo de 19.254 segundos.

FIRMANTE-INICIALIZAR-

REGISTRO?IDFIRMANTE=3: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.197 segundos, un promedio de 2.830 segundos y un máximo de 20.185 segundos

GESTOR-LISTAR?IDESTADO=0: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.196 segundos, un promedio de 1.495 segundos y un máximo de 19.255 segundos.

Tabla 11:
Pruebas de Rendimiento - Caso N° 03

CASO 03	RESULTADOS
Se tuvo en cuenta a 100 postulantes realizando acciones	/INICIO: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 1.907 segundos, un promedio de 2.880 segundos y

dentro del portal teniendo como ingreso a cada 50 segundos después de que ingresó el primero postulante y un intervalo de tiempo que nos ayuda a que las peticiones se realicen en un tiempo simulado a la realidad (para este caso un intervalo de 0 a 120 segundos por cada petición).	un máximo de 7.002 segundos. Teniendo en cuenta un promedio de Bytes de 2797793.
	/POSTULANTE/INICIALIZAR?IDPOSTULANTE=1: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.704 segundos, un promedio de 1.431 segundos y un máximo de 6.608 segundos. Teniendo en cuenta un promedio de Bytes de 111637.6
	/SOLICITUDBECA/LISTARCONVOCATORIAANIO?ANIO=2022&IDPOSTULANTE=1: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.185 segundos, un promedio de 0.713 segundos y un máximo de 5.703 segundos.
	/REPORTE/REPORTEFICHASHISTORIAL/GENERARFICHA: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.283 segundos, un promedio de 2.127 segundos y un máximo de 4.982 segundos.

Tabla 12:
Pruebas de Rendimiento - Caso N° 04

CASO 04	RESULTADOS
Se tuvo en cuenta a 100 postulantes realizando acciones dentro del portal teniendo como ingreso a cada 1 segundos después de que ingresó el primero postulante y un intervalo de tiempo que nos ayuda a que las peticiones se realicen en un tiempo simulado a la realidad (para este caso un intervalo de 0 a 1 segundos por cada petición).	/LOGIN: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 1.181 segundos, un promedio de 2.181 segundos y un máximo de 3.617 segundos.
	/INICIO: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 3.723 segundos, un promedio de 23.375 segundos y un máximo de 30.931 segundos.
	/REPORTE/REPORTEFICHASHISTORIAL/GENERARFICHA: se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 1.266 segundos, un promedio de 1.892 segundos y un máximo de 3.993 segundos.

Tabla 13:

Pruebas de Rendimiento - Recursos

RECURSOS	RESULTADOS
<p>CASO 1: Cuando 1 usuario ingresa a la ruta porta/inicio y se realizan las peticiones requeridas:</p>	<p>/INICIO: se observó se realizó las peticiones y se obtuvo que el tiempo mínimo de respuesta fue:</p> <ul style="list-style-type: none"> • Primera prueba: un mínimo de 10.525 segundos, un promedio de 10.525 segundos y un máximo de 10.525 segundos, teniendo un promedio total en Bytes de 2797793. • Segunda prueba: un mínimo de 6.148 segundos, un promedio de 6.148 segundos y un máximo de 6.148 segundos, teniendo un promedio total en Bytes de 2797793 • Tercera prueba: un mínimo de 7.521 segundos, un promedio de 7.521 segundos y un máximo de 7.521 segundos, teniendo un promedio total en Bytes de 2797793.
<p>CASO 2: Cuando se tienen 20 usuarios que ingresan al mismo tiempo a la ruta porta/inicio y se realizan las peticiones requeridas:</p>	<p>/INICIO: se observó se realizó las peticiones y se obtuvo que el tiempo mínimo de respuesta fue:</p> <ul style="list-style-type: none"> • Primera prueba: 4.141 segundos, un promedio de 7.581 segundos y un máximo de 11.009 segundos, teniendo un promedio total en Bytes de 2797793. • Segunda prueba: 4.141 segundos, un promedio de 7.581 segundos y un máximo de 11.009 segundos, teniendo un promedio total en Bytes de 2797793. • Tercera prueba: 5.600 segundos, un promedio de 8.387 segundos y un máximo de 11.570 segundos, teniendo un promedio total en Bytes de 2797793. • Cuarta prueba: 5.174 segundos, un promedio de 8.022 segundos y un máximo de 10.877 segundos, teniendo un promedio total en Bytes de 2797793.
<p>CASO 3: Cuando se tienen 50 usuarios que ingresan cada 2 segundos a la ruta porta/inicio y se realizan las peticiones requeridas:</p>	<p>/INICIO: se observó se realizó las peticiones y se obtuvo que el tiempo mínimo de respuesta fue:</p> <ul style="list-style-type: none"> • Primera prueba: 12.327 segundos, un promedio de 20.645 segundos y un máximo de 25.059 segundos, teniendo un promedio total en Bytes de 2797793.

- Segunda prueba: 8.992 segundos, un promedio de 17.450 segundos y un máximo de 22.432 segundos, teniendo un promedio total en Bytes de 2797793.
- Tercera prueba: 6.932 segundos, un promedio de 18.531 segundos y un máximo de 22.748 segundos, teniendo un promedio total en Bytes de 2797793.

CASO 4: Cuando se tienen 100 usuarios que ingresan cada 5 segundos a la ruta porta/inicio y se realizan las peticiones requeridas:

- /INICIO:** se observó se realizó las peticiones y se obtuvo que el tiempo mínimo de respuesta fue:
- Primera prueba: 20.364 segundos, un promedio de 30.893 segundos y un máximo de 41.393 segundos, teniendo un promedio total en Bytes de 2797793.
 - Segunda prueba: 12.027 segundos, un promedio de 32.071 segundos y un máximo de 40.244 segundos, teniendo un promedio total en Bytes de 2797793.
 - Tercera prueba: 23.718 segundos, un promedio de 31.319 segundos y un máximo de 37.433 segundos, teniendo un promedio total en Bytes de 2797793.

- Pruebas de estructura

- Backend

Tabla 14:

Pruebas de estructura - Backend

TIPO	ELEMENTOS	DETALLE
BUGS	Call "existingObject.isPresent()" before accessing the value.	antes de obtener el valor correspondiente se debe validar que existe/que está presente porque si dado el caso se está tomando el valor cuando no existe
	Call "miListaAptoDao.isPresent()" before accessing the value.	antes de obtener el valor correspondiente se debe validar que existe/que está presente porque si dado el caso se está tomando el valor cuando no existe

Use isEmpty() to check whether the collection is empty or not.	Para este tipo de validación se recomienda usar el metodo isEmpty() antes de size()==0
Strings and Boxed types should be compared using "equals()	Casi siempre es un error comparar dos instancias de java.lang.String o tipos en caja como java.lang.Integer usando la igualdad de referencia == o !=, porque no está comparando el valor real sino las ubicaciones en la memoria.
Remove or correct this "removeAll" call.	Algunos métodos requieren que el argumento permanezca sin modificar durante la ejecución.
A "NullPointerException" could be thrown; "miListaConvocatoria" is nullable here.	Nunca se debe desreferenciar/acceder a una referencia a nulo. Si lo hace, se lanzará una NullPointerException
Add "lang" and/or "xml:lang" attributes to this "<html>" element	Las etiquetas / y /<i> tienen exactamente el mismo efecto en la mayoría de los navegadores web, pero existe una diferencia semántico.
'password' detected in this expression, review this potentially hard-coded password	Las contraseñas no deben estar ingresadas de forma estática en el código.
Make sure that enabling CORS is safe here – una cantidad total de (8)	Tener una política de uso compartido de recursos de origen cruzado permisiva es sensible a la seguridad.
	Es un error pensar que un servicio siempre tendrá la misma dirección IP. Cuando cambie, la

<i>Make sure using this hardcoded IP address is safe here.</i>	<i>IP codificada también deberá modificarse</i>
<i>Define a constant instead of duplicating this literal "idImgAsiento" 7 times.</i>	<i>Se debe definir una variable para aquellas cadenas de caracteres usados en más de un bloque de código.</i>
<i>Refactor this method to reduce its Cognitive Complexity from 22 to the 15 allowed</i>	<i>Se debe tratar de minimizar los bloques de código como estrategia de optimización.</i>
<i>Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation</i>	<i>Agregar comentarios explicando del porqué de los métodos vacíos.</i>
<i>Code Commentaries</i>	<i>El código no utilizado debe eliminarse y puede recuperarse del historial de control de código fuente si es necesario.</i>
<i>Remove this unused "applicationContext" private field</i>	<i>Si se declara un campo privado pero no se usa en el programa, puede considerarse código muerto y, por lo tanto, debe eliminarse.</i>
<i>1 duplicated blocks of code must be removed – REGLA DEPRECADA</i>	<i>Cuando se realiza las importaciones, si se hace una importación general con * y después se hace la importación específica que previamente se ha importado como un todo.</i>

➤ **Frontend**

Tabla 15:

Pruebas de estructura - Frontend

TIPO	ELEMENTOS	DETALLE
ISSUES	<i>Unreachable code</i>	<i>Las declaraciones de salto (return, break y continue) y las expresiones throw mueven el flujo de control fuera del bloque de código actual. Entonces, cualquier declaración que venga después de un salto es un código muerto.</i>
	<i>Add a "title" attribute to this <iframe> tag</i>	<i>Las etiquetas <frame> y <iframe> siempre deben tener un atributo de título para los usuarios con o sin discapacidades pueden escanear fácilmente el contenido de todos los marcos a la vez.</i>
	<i>Unexpected missing generic font family</i>	<i>Si ninguno de los nombres de fuente definidos en una declaración de fuente o familia de fuentes está disponible en el navegador del usuario, el navegador mostrará el texto usando su fuente predeterminada.</i>
	<i>Make sure not using rel="noopener" is safe here</i>	<i>El objeto window.opener no es nulo y, por lo tanto, la página abierta puede configurar window.opener.location como un sitio web malicioso).</i>
	<i>Unexpected empty source</i>	<i>Esta regla plantea un problema cuando un archivo CSS está vacío (es decir, contiene solo espacios).</i>
	<i>Remove this commented out code</i>	<i>Los programadores no deben comentar el código, ya que infla los programas y reduce la legibilidad.</i>
	<i>Remove this unused import of 'UtilityFunctions'.</i>	<i>No hay razón para importar módulos que no usas. Hacerlo aumenta innecesariamente la carga.</i>

Variable 'blob' is already defined.

Es posible usar el mismo símbolo varias veces como una variable o una función, pero hacerlo probablemente confunda a los mantenedores

Consider moving declaration of 'key' as it is referenced outside current binding context

Las variables que se declaran dentro de un bloque pero se usan fuera de él (lo cual es posible con una declaración de estilo var) deben declararse fuera del bloque.

Unexpected empty block

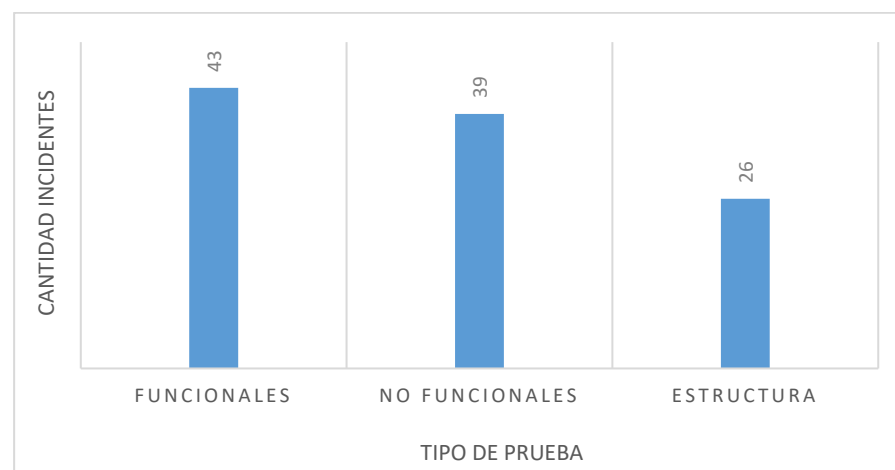
Los bloques vacíos se introducen por error. Son inútiles y evitan la legibilidad del código. Deben eliminarse o completarse con código real.

- **Resumen del resultado de pruebas realizadas**

La cantidad de resultados de los incidentes encontrados en cada tipo de prueba se clasificó según el tipo de error obtenido de manera general con el objetivo de subsanar y mejorar estos aspectos en las diversas fases del software que se estaba desarrollando en ese momento.

Figura 33:

Resumen del total de incidencias encontrados en las pruebas



4.1.2 ANÁLISIS DE LA SITUACIÓN ACTUAL – POST TEST

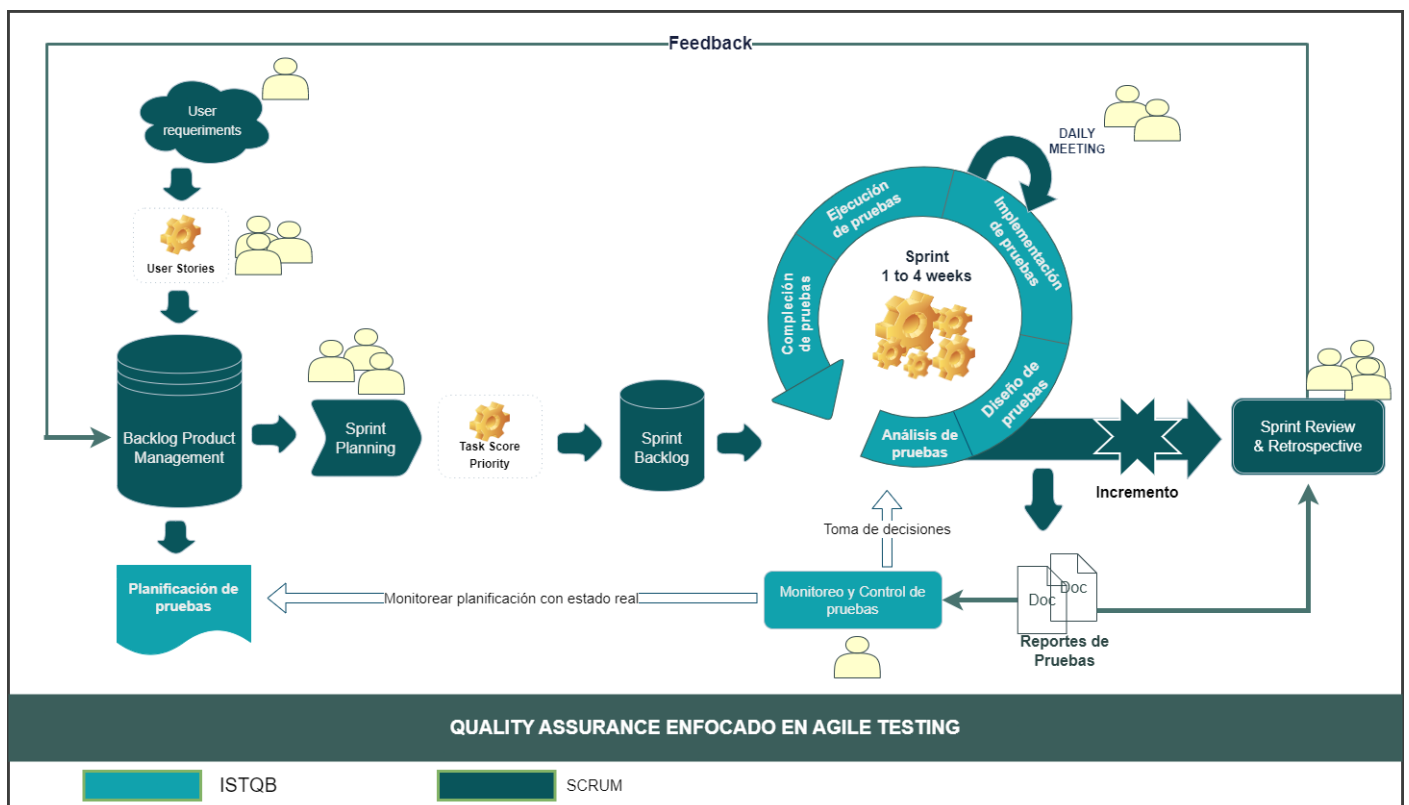
En análisis de la situación actual post test tuvo como objetivo realizar las diversas actividades para poder mejorar la calidad de software en base a los resultados obtenidos en el análisis de la situación actual pre test, para eso se realizó la implementación y adecuación de los requerimientos previamente identificados.

Modelo SCRUM – ISTQB

Después de haber obtenido los requerimientos necesarios para poder implementar el aseguramiento de la calidad mediante las pruebas ágiles teniendo como base el modelo de pruebas según ISTQB, se tuvo como estrategia tener en cuenta los rasgos habituales de la empresa y realizar la adecuación del marco de trabajo ágil Scrum en conjunto con el modelo de pruebas ISTQB.

Figura 34:

Modelo SCRUM – ISTQB



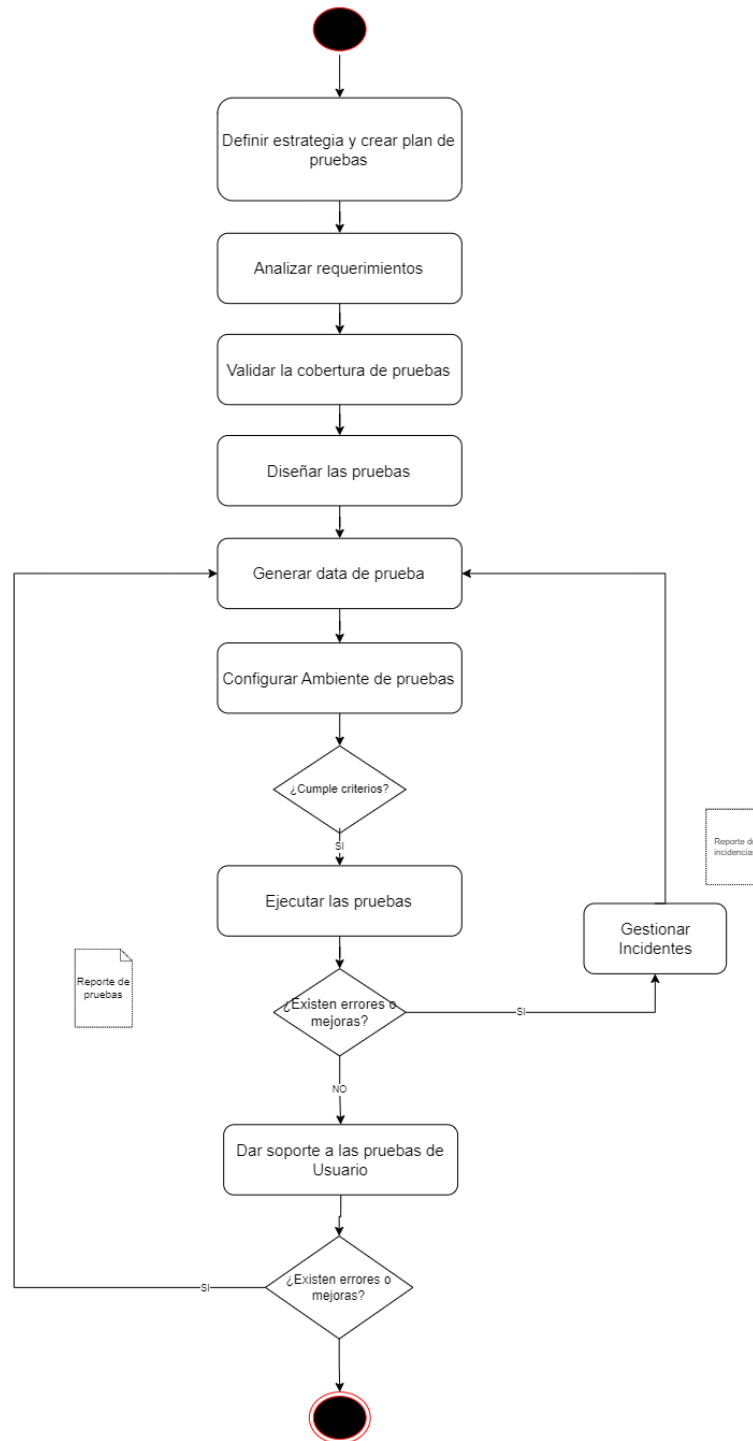
Nota: Unión del marco de trabajo ágil SCRUM con el modelo de pruebas ISTQB como estrategia para mejorar la calidad de software.

DIAGRAMA DE FLUJO DE ACTIVIDADES DE PRUEBAS

El conjunto de actividades para realizar las pruebas es diversos acorde al contexto que lo aplica y por ese motivo se creó un diagrama de flujo de actividades de pruebas como estrategia de apoyo para el área de calidad.

Figura 35:

Diagrama de flujo de actividades de pruebas



HISTORIAS DE USUARIO

En el transcurso de la investigación lo que se pudo observar según las actividades que se realizaban era la dificultad en establecer los requerimientos conforme a lo que se necesitaba, errores por parte del cliente y del equipo encargado del desarrollar el producto. De esta forma se realizó la inclusión de las historias de usuario como estrategia para reducir los errores en el establecimiento de los requerimientos y no tener dificultades en los futuros entregables que hubiese.

Figura 36:

Definición de requerimientos confusos

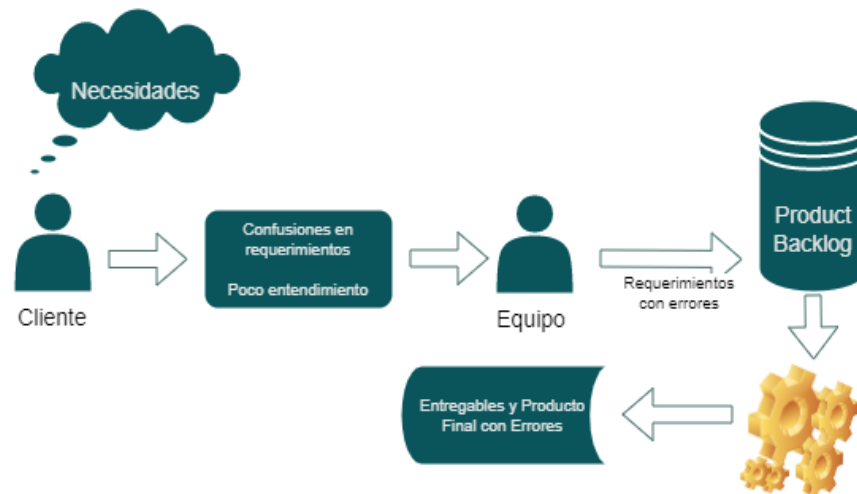


Figura 37:

Requerimientos mediante historias de usuario

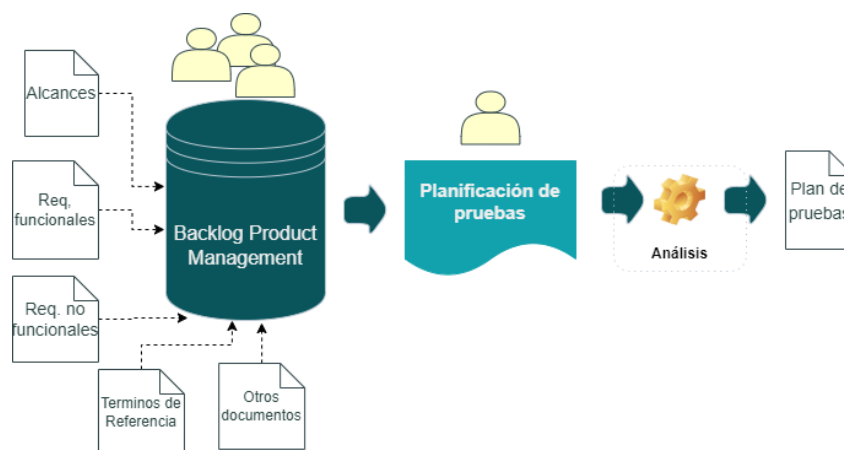


4.1.2.1 Planificación de pruebas

La planificación de pruebas es la primera fase dentro del modelo de pruebas según ISTQB, la cual forma base principal y guía de las siguientes fases.

Figura 38:

Planificación de pruebas



PLAN DE PRUEBAS

El presente plan de pruebas tiene como finalidad establecer lineamientos, objetivos y estrategias de pruebas para el proyecto DEVIDA.

Respecto al proyecto

El proyecto contempla el desarrollo de una aplicación con la finalidad de gestionar el otorgamiento de las becas, convocatorias, cursos, certificaciones, consultas, estadísticas, reportes, entre otros.

Misión de evaluación

La validación del cumplimiento de todos los requerimientos estipulados para la creación del software dentro de lo cual se pueda contar con la información necesaria para poder saber los alcances que se tiene, desviaciones, anomalías, otros, asimismo realizar acciones complementarias como ajustes y otros para el cumplimiento de los objetivos establecidos dentro del proyecto.

Documentos de referencia:

Los principales documentos con los que el cliente y el equipo especifican los requerimientos son:

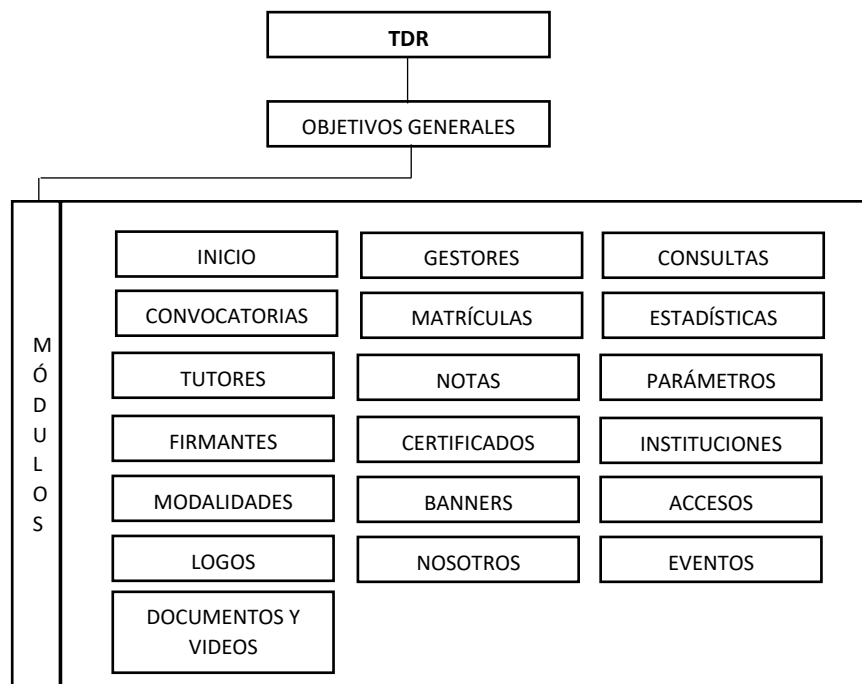
- Términos de Referencia.
- Historias de usuario.
- Requerimientos funcionales.
- Requerimientos no funcionales.
- Otros requerimientos complementarios.
- Documentación de proyectos anteriores.

Alcance:

En relación a los productos a testear están referidos a todos los módulos o apartados contenidos en el desarrollo del proyecto, los cuales son:

Figura 39:

Módulos del software en desarrollo



En relación a los niveles de prueba, se realizarán principalmente las pruebas de sistema, integración y pruebas de Aceptación de Usuario,

En relación a los tipos de prueba, el equipo de calidad podrá enfoque en las pruebas de servicios, pruebas funcionales, rendimiento y otro que sea importante para el cumplimiento de los objetivos.

Restricciones y dependencias

- Historias de usuario bien definidas.
- Accesos a la infraestructura.
- Tener el o los modelos de datos necesarios para el proyecto.
- Documentación funcional.
- Participación del equipo de calidad en las ceremonias del proyecto.
- Configuración del ambiente de pruebas.

Modelos de prueba

- **Ambiente de prueba**

Actualmente se cuenta con un ambiente de despliegue del desarrollo donde se realiza las pruebas de parte de la empresa encargada de la construcción del producto y la entidad cliente en un corto tiempo pondrá un ambiente de pruebas o pre producción donde se realizarán las validaciones correspondientes del software.

A continuación se presenta la lista de ambientes y componentes correspondientemente.

Tabla 16:

Ambientes de pruebas

AMBIENTE	COMPONENTE
DEV/QA	Frontend
	Backend
	Base de datos
	API's de terceros (RENIEC, Moodle, etc).
PRE PRODUCCIÓN (Entidad Cliente)	Frontend
	Backend
	Base de datos

- **Configuraciones/Plataformas**

Las configuraciones para el ambiente de pruebas son una serie de pasos para la conexión mediante VPN a base de datos y otros.

Referentes

Tabla 17:

Referentes

Persona	Rol	Correo
Jean Colonia	Quality Engineer	engshka123@gmail.com
Equipo de desarrollo	Development	-
Equipo de calidad	Quality	-

Estrategia de pruebas funcionales

Las pruebas funcionales incluyen las técnicas de pruebas dinámicas y estáticas donde se revisa la documentación de forma estática.

Para las pruebas dinámicas, se realizarán mediante las pruebas manuales y automatizadas bajo las técnicas de caja negra.

- Se trabajará mediante Gherkin.
- Ejecución, reporte y corrección de incidentes.
- Verificar el alcance de las pruebas.

También se realizarán las pruebas de los servicios de backend.

Estrategia de pruebas de Servicios

El objetivo es probar todos los servicios de parte del backend, simulando a los clientes que ingresan al software.

Para lo cual se realizará la implementación de Karate DSL, POSTMAN y Newman para los reportes.

La finalidad es dar el acompañamiento respectivo al equipo de desarrollo según vayan construyendo los servicios donde se testeará teniendo como mínimo los casos donde los escenarios sean válidos y escenarios donde sean más propensos a ocurrir, dichas pruebas se realizarán de 2 formas, el ambiente local y un ambiente desplegado para las pruebas.

Severidad de los defectos reportados

Los defectos encontrados se tendrán en cuenta durante todo el ciclo de vida y se clasificarán mediante el nivel de severidad y al riesgo que cada uno represente para el software en desarrollo.

- **Bloqueante:** se refiere a cualquier defecto que no permita continuar con las actividades del software y no se pueda realizar las pruebas.
- **Crítico:** defectos que cambien el comportamiento, la funcionalidad del problema o el proceso sea contradictorio con lo esperado según los requerimientos.
- **Mayor:** defectos que no impidan continuar con las pruebas pero que existan pérdidas de las funcionalidades.
- **Menor:** defectos donde existan una menor pérdida de funcionalidades o que se realicen de otra manera.
- **Bajo:** defectos que no tiene impacto directo con el resultado o los requerimientos solicitados por el cliente.

Validación de los casos de pruebas

La validación se realizará mediante pares o en casos especiales en equipo como estrategia de dar feedback sobre la situación o incidentes que se presenten.

Criterios de Aceptación y aprobación del Testing

Los criterios de aceptación están referenciados según los requerimientos/funcionalidades que se tengan previamente y a partir de ello se hayan creado los casos de prueba.

Con referencia a los criterios de aprobación del testing se considera lo siguiente:

- Haber realizado la ejecución del 100% de los casos de prueba previamente diseñados.
- En la fase de pruebas puede existir casos que no apliquen con referencia a los casos de prueba diseñados.
- Tener un porcentaje del 0% de cualquier tipo de severidad.

Entregables

Como mejores prácticas dentro de los entregables se tendrá en consideración los siguientes artefactos de pruebas:

- El plan de pruebas.
- Matriz de casos de pruebas.
- Evidencias de las pruebas.

Planificación

La planificación se realizará de acuerdo a los ambientes, tipos de pruebas y estrategias indicadas previamente.

Tabla 18:

Ciclo de pruebas

ITEM	OBJETIVO	TIPO DE PRUEBA	AMBIENTE	PLATAFORMA
Ciclo de pruebas Estáticos	Verificar toda documentación requerida como estrategia para identificar los escenarios y también diseñar los casos de prueba	Estática	No aplica	No aplica
Ciclo de pruebas Dinámico Funcional	Validar las características funcionales del software con respecto a los requerimientos establecidos previamente para cada uno.	Dinámica / Funcional	QA	Windows 10 Navegadores Web Chrome, Brave, Edge u otro.

Fases de pruebas

Todos los ciclos de pruebas existentes estarán conformados de manera general por las siguientes fases:

- Análisis de historias de usuario / requerimientos.
- Diseño de los casos de prueba.

- Ejecución de los casos de prueba.
- Reporte y seguimiento de defectos.
- Seguimiento del testing.

Riesgos Identificados

Tabla 19:

Riesgos identificados


riesgo	probabilidad de ocurrencia	impacto	acción de mitigación	acción de contingencia
En el inicio de la ejecución de las pruebas y aun se cuenta con omisiones a la especificación de requerimientos.	MEDIA	ALTA	Acordar un nuevo alcance y cobertura de las pruebas respecto a la definición inicial	Realizar el reporte de los defectos previamente detectados en la herramienta donde se gestione este tipo de casos.
En el momento que no se cuenta con la disponibilidad de un ambiente QA para realizar las pruebas necesarias	MEDIA	ALTA	Realizar las acciones necesarias para crear el entorno requerido. Realizar la gestión para contar con dichos entornos.	Acordar con los responsables de dicho servicio por parte del cliente de los riesgos y proceder a ejecutar las posibles pruebas informando el alcance y cobertura respecto al plan.
Cuando no se tenga la disponibilidad del despliegue a testear en tiempo y forma según la planificación realizada	BAJA	ALTA	Re planificar y optimizar del equipo encargado para reducir los desvíos provocados por el desfasaje en el despliegue.	Supervisar el funcionamiento de los distintos procesos relacionados.











Tiempo reducido asignado para las pruebas	MEDIA	MEDIA	Optimizar la coordinación entre el equipo de desarrollo y testing para tener el menor impacto	Re planificar la estimación de tiempo y esfuerzo según el entorno y los recursos disponibles.
---	--------------	--------------	---	---

Herramientas / Repositorios

Figura 40:

Herramientas y Repositorios

PROPÓSITO	HERRAMIENTA	
Gestión de requerimientos / Historias de Usuario	Clickup	
Gestión de Bugs	Clickup	
	Bugzilla	
Gestión de casos de pruebas	Mantis	
	Clickup	
Ejecución de pruebas de rendimiento	JMeter	
	Blazemeter	
Pruebas de Servicios	Karate DSL	
	Postman	
	Newman Reports	
	Maven	
Bases de datos	Oracle DB	

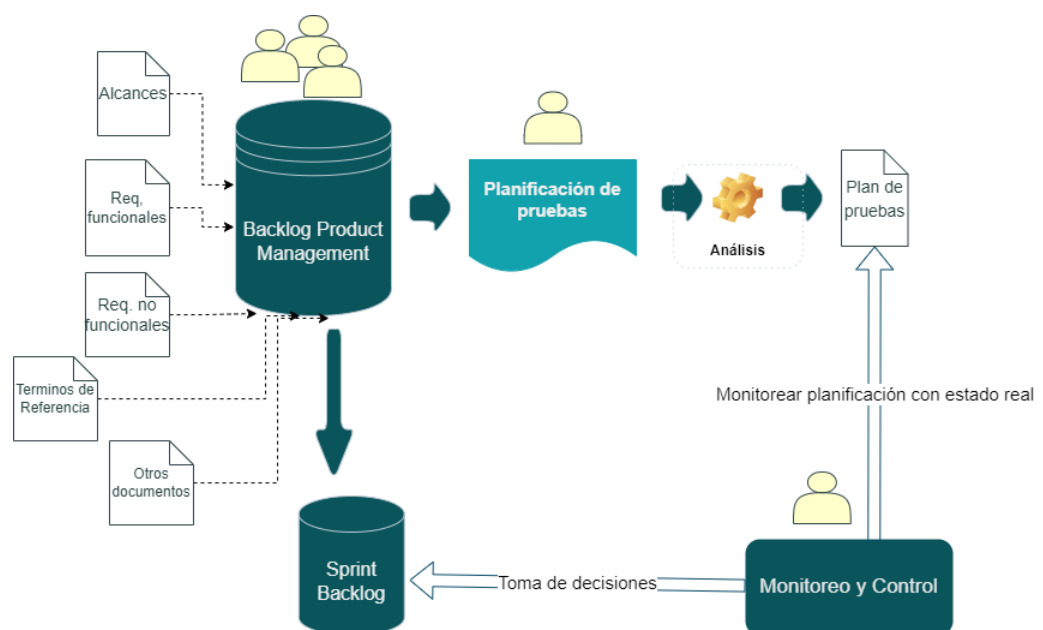
	Gherkin/Cucumber	
Pruebas Funcionales	Selenium	
	WebdriverIO	
	Typescript	
	Java	
	NPM	
	Maven	
Pruebas de estructura	Sonarqube	
Repositorios	GitLab	
	GitHub	

4.1.2.2 Monitoreo y Control de pruebas

En esta fase lo que se realiza es el monitoreo de las actividades que se realizan dentro de las pruebas y un contraste con lo que se ha planificado previamente.

Figura 41:

Monitoreo y Control de pruebas



Monitoreo

Para tener un informe o reporte del progreso actual de las pruebas y contrastar con lo planificado se pueden tener una gran variedad de formatos pero para la presente investigación se tuvieron en cuenta 2 formatos los cuales quedaron como opciones para el área de calidad de la empresa que fue contexto de investigación.

Figura 42:

Modelo de Monitoreo N° 01

RESUMEN	
Total de casos de pruebas	10
Executed / Ejecutados	7
Pass / Aprobados	5
Fail / Fallado	2
No ejecutados	3

Figura 43: Modelo de Monitoreo N° 02

Estado de casos de prueba	Resultados de compilación actual	Resultados de compilación previos
Cantidad total de Casos de prueba	69 ↑	63
Casos de prueba aprobados	42 ↑	31
Casos de prueba fallados	19 ↑	16
Casos de prueba bloqueados	11 ↓	16

Control de pruebas

El control de pruebas se refiere a tomar las acciones correctivas requeridas para conseguir los objetivos previamente establecidos dentro del plan de pruebas u otro documento que haga del mismo.

Un aspecto importante para poder realizar el monitoreo y el control de las pruebas es la cobertura que se tiene en ese momento, la cobertura de pruebas no significa otra cosa que la cuantificación de las pruebas que se han realizado y en total que se tiene.

Tabla 20:

Modelo de cobertura de pruebas

Pruebas del sistema	Cobertura de pruebas	Número de semanas	Avance real
Módulo de cursos	30%	1	15%
Módulo de Convocatorias	30%	1	25%
Módulo de gestión de certificados	40%	2	10%
Total	100%	4	50%

4.1.2.3 Análisis de pruebas

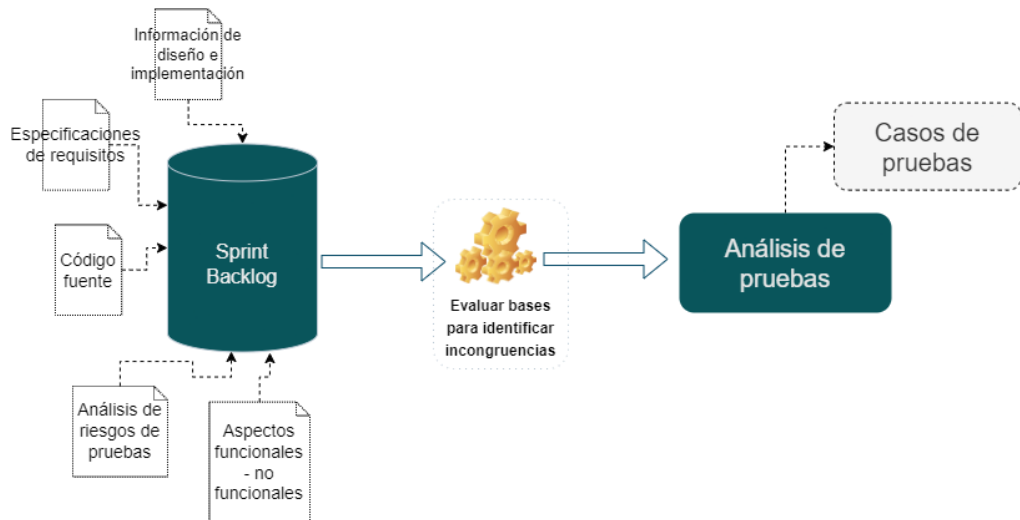
La fase del análisis de las pruebas tiene como objetivo analizar todas las bases de pruebas existentes, específicamente ¿qué es lo que se desea probar? y según el nivel de prueba requerido.

Algunos ejemplos son:

- Las especificaciones de requisitos.
- Información de diseño e implementación (componentes, estructura del sistema, arquitectura, otros).
- Análisis de riesgos (aspectos funcionales y no funcionales).
- Evaluar todas las bases de pruebas para identificar defectos, incongruencias, otros.

Figura 44:

Análisis de pruebas



Como enfoque práctico para la implementación de quality assurance enfocado en agile testing con base en ISTQB se ha sacado un extracto de los requerimientos del desarrollo de software que estaban realizando en ese momento dentro de la empresa contexto de la investigación.

Extracto de requerimientos dentro un de los Sprints realizados dentro del proyecto desarrollado en ese momento.

- Realizar login al sistema según credenciales y roles.
- Los postulantes deberán llenar declaración jurada sobre información real.
- El sistema debe permitir al postulante actualizar su información general consignada.
- El sistema no debe permitir registrar a un postulante en una convocatoria que ya concluyó su etapa de registro de postulantes.
- Mostrar la relación de postulantes según convocatoria con sus respectivos estados de Aptos / No Aptos
- Generación de PDF con toda la información registrada por el postulante (Ficha de Inscripción)
- Reporte de becados y no becados
- Reporte de aprobados, desaprobados, promedio de deserción y retiros obtenidos en el grupo-sección.

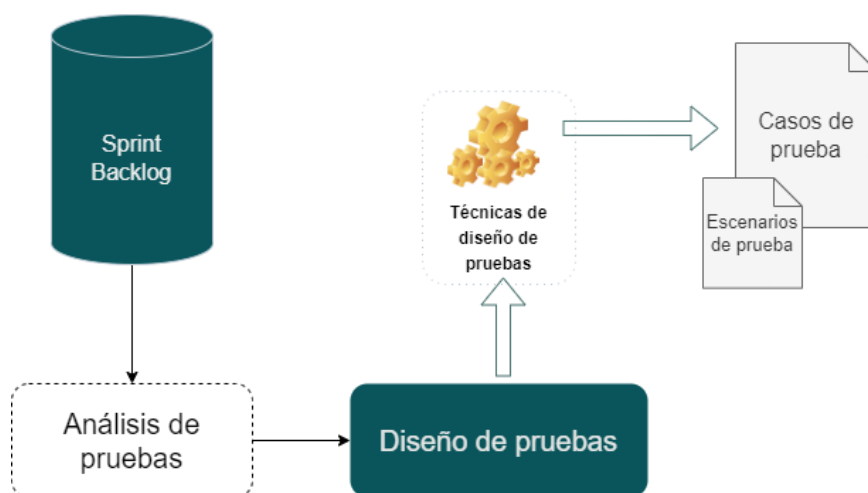
- Crear, subir y editar el modelo de certificado por curso y grupo sección.
- Los certificados digitales deben contar con dos caras.
- Visualizar las notas de los becados según el año, convocatoria, grupo sección y el estado.

4.1.2.4 Diseño de pruebas

Después de haber realizado un previo análisis de lo que se desea probar, en esta fase se tiene como objetivo la priorización del diseño de pruebas según las condiciones y escenarios de prueba, ya antes identificados mediante el análisis.

Figura 45:

Diseño de pruebas



Desde un enfoque práctico se tomó ejemplo un requerimiento ya antes descrito:

- Visualizar las notas de los becados según el año, convocatoria, grupo sección y el estado.
- Realizar login al sistema según credenciales y roles.

Tabla 21:

Identificación de Escenario y caso de prueba

N° REQUERIMIENTO	ESCENARIO	CASO DE PRUEBA
01	Verificar las notas de los becados según una convocatoria	Visualizar las notas de los becados con un determinado estado.
02	Ingresar al sistema	Realizar login al sistema según credenciales y roles

Las estrategias para poder realizar una buena definición de los requerimientos fueron:

- Implementar las historias de usuario en la definición de los requerimientos por parte del cliente.
- Implementar Gherkin-Cucumber y crear features a partir de las historias de usuario con la finalidad de mejorar la comunicación y el trabajo.

Ejemplo de la aplicación:

- **Requerimiento:** Visualizar las notas de los becados según el año, convocatoria, grupo sección y el estado.
- **Historia de Usuario:**

Historia de usuario: Visualización de Notas de becados Como administrador

Requiero ver las notas de los becados que han culminado un curso en una convocatoria según el estado, sea aprobado, desaprobado, deserto o retiro

Para identificar su situación académica, generar sus certificados y los reportes.

Crterios de Aceptación:

- Debe tener privilegios de administrador u otro a excepción del perfil de postulante (becario).
- Debe existir un filtro para buscar según el año.
- Debe existir un filtro para buscar según la convocatoria.
- Debe existir un filtro para buscar según el grupo-sección.
- Para visualizar el listado previamente se debe haber seleccionado como mínimo los filtros de año, convocatoria y grupo-sección.
- También un filtro para buscar según el estado (aprobado, desaprobado, deserto o retiro).
- Si no se ha seleccionado el filtro del estado se debe mostrar el listado con todos los estados existentes según sea el caso.
- No debe aparecer las convocatorias y el grupo-sección que aún no haya culminado.

Implementación de Gherkin con Cucumber para mejorar el trabajo y comunicación entre el equipo de desarrollo y calidad, ya que no solo se pudo identificar inconvenientes en los requerimientos que transmitía el cliente a la empresa encargada de desarrollar el software sino también dentro del equipo de desarrollo y de calidad, donde lo que se debía hacer y las restricciones o características no estaban bien definidas

Sintaxis Gherkin

Característica: tiene como objetivo describir de forma general funciones y agrupar escenarios.

Escenario: contiene las reglas de negocio de algo específico.

Dado / Given: utilizado para describir el contenido inicial del escenario.

Cuando/When: utilizado para describir las acciones a partir del contenido inicial previamente definido.

Entonces/Then: utilizado para describir un resultado dentro del escenario.

Y / And: utilizado para añadir mayores acciones o características después del Dado o Entonces.

Pero / But: utilizado para agregar una condición extra.

Scenario Outline: utilizado para ejecutar varias pruebas mediante diferentes combinaciones de valores.

Figura 46:

BDD con Gherkin-Cucumber

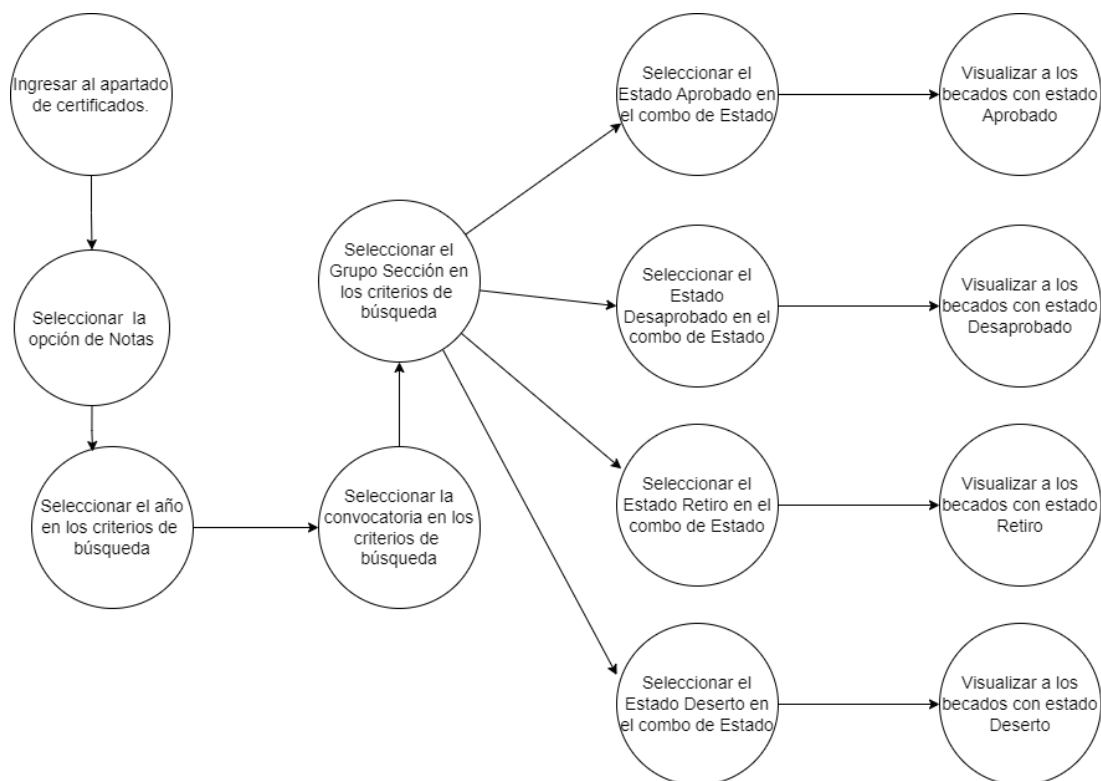
```
Feature: : Probar la Visualización de Notas de becados
  @VerificarNotasDeBecados
  Scenario Outline: Validar las notas de los becados segun filtros y estado
    Given que el usuario se encuentra en la pagina del buscador de duckduckgo
    Given que el usuario ha abierto el navegador e ingresa al sistema con sus credenciales
    And hacer click en el modulo de gestion de certificados
    And se dirige hacia el apartado de las notas
    When selecciona el año en los criterios de busqueda
    And selecciona la convocatoria en los criterios de busqueda
    And selecciona el el grupo seccionen los criterios de busqueda
    And selecciona el estado aprobado en el combo correspondiente
    And hace click en el boton filtrar
    Then se muestra el listado de las notas de los becados segun el estado aprobado

  Examples: e
    | usuario | contraseña |
    | 789641235 | contraseña123 |
```

Como estrategia para los casos de prueba se tomó como referencia la técnica de transición de estados para el requerimiento 01.

Figura 47:

Técnica de transición de estados



Como se puede visualizar se han encontrado 4 casos de pruebas después de haber aplicado la técnica de transición de estados, los cuales son:

Tabla 22:

Casos de prueba identificados

Nº DEL CASO DE PRUEBA	DESCRIPCIÓN
TC_001	Probando el filtrado de las notas de los becarios con estado aprobado
TC_002	Probando el filtrado de las notas de los becarios con estado desaprobado
TC_003	Probando el filtrado de las notas de los becarios con estado retiro.
TC_004	Probando el filtrado de las notas de los becarios con estado desierto

Tabla 23:

Caso de prueba TC_001

Test Case ID	TC_001	Test Case Description	Probando el filtrado de las notas de los becarios aprobados en una convocatoria		
Created By	Jean Colonia	Reviewed By	Jean Colonia	Version	1.0
Tester's Name	Jean	Date Tested	10 de Agosto del 2022	Test Case (Pass/Fail/Not)	
S #	Prerequisites:		S #	Test Data	
1	Haber abierto el navegador		1	Usuario = 46596449	
2	Ingresar al sistema con las credenciales requeridas (admin).		2	Contraseña = *****	
3			3		
4			4		
Test Scenario	Verificar las notas de los becados según una convocatoria				
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Blocked	
1	Ingresar al apartado de certificados.	Visualizar la sección de gestión de certificados.			
2	Seleccionar la opción de Notas	Visualizar la sección de Notas			
3	Seleccionar el año en los criterios de búsqueda	El año seleccionado se muestra como valor en el combo de Año.			
4	Seleccionar la convocatoria en los criterios de búsqueda	La convocatoria seleccionado se muestra como valor en el combo de Año.			
5	Seleccionar el Grupo Sección en los criterios de búsqueda	El grupo sección seleccionado se muestra como valor en el combo de Grupo Sección.			
6	Seleccionar el Estado Aprobado en el combo de Estado	El estado seleccionado se muestra como valor en el combo de Estado.			
7	Hacer clic en el botón filtrar	Se muestra el listado de las notas de los becados según el estado de becados aprobados			

Tabla 24:

Caso de prueba TC_002

Test Case ID	TC_002	Test Case Description	Probando el filtrado de las notas de los becarios desaprobados en una convocatoria		
Created By	Jean Colonia	Reviewed By	Jean Colonia	Version	1.0
Tester's Name	Jean	Date Tested	10 de Agosto del 2022	Test Case (Pass/Fail/Not)	
S #	Prerequisites:		S #	Test Data	
1	Haber abierto el navegador		1	Usuario = 46596449	
2	Ingresar al sistema con las credenciales requeridas (admin).		2	Contraseña = *****	
3			3		
4			4		
Test Scenario Verificar las notas de los becados según una convocatoria					
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Blocked	
1	Ingresar al apartado de certificados.	Visualizar la sección de gestión de certificados.			
2	Seleccionar la opción de Notas	Visualizar la sección de Notas			
3	Seleccionar el año en los criterios de búsqueda	El año seleccionado se muestra como valor en el combo de Año.			
4	Seleccionar la convocatoria en los criterios de búsqueda	La convocatoria seleccionado se muestra como valor en el combo de Año.			
5	Seleccionar el Grupo Sección en los criterios de búsqueda	El grupo sección seleccionado se muestra como valor en el combo de Grupo Sección.			
6	Seleccionar el Estado Desaprobado en el combo de Estado	El estado seleccionado se muestra como valor en el combo de Estado.			
7	Hacer clic en el botón filtrar	Se muestra el listado de las notas de los becados según el estado de becados			

Tabla 25:

Caso de prueba TC_003

Test Case ID	TC_003	Test Case Description	Probando el filtrado de las notas de los becarios con estado retiro en una convocatoria		
Created By	Jean Colonia	Reviewed By	Jean Colonia	Version	1.0
Tester's Name	Jean	Date Tested	10 de Agosto del 2022	Test Case (Pass/Fail/Not)	
S #	Prerequisites:		S #	Test Data	
1	Haber abierto el navegador		1	Usuario = 46596449	
2	Ingresar al sistema con las credenciales requeridas (admin).		2	Contraseña = *****	
3			3		
4			4		
Test Scenario	Verificar las notas de los becados según una convocatoria				
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Blocked	
1	Ingresar al apartado de certificados.	Visualizar la sección de gestión de certificados.			
2	Seleccionar la opción de Notas	Visualizar la sección de Notas			
3	Seleccionar el año en los criterios de búsqueda	El año seleccionado se muestra como valor en el combo de Año.			
4	Seleccionar la convocatoria en los criterios de búsqueda	La convocatoria seleccionado se muestra como valor en el combo de Año.			
5	Seleccionar el Grupo Sección en los criterios de búsqueda	El grupo sección seleccionado se muestra como valor en el combo de Grupo Sección.			
6	Seleccionar el Estado Desaprobado en el combo de Estado	El estado seleccionado se muestra como valor en el combo de Estado.			
7	Hacer clic en el botón filtrar	Se muestra el listado de las notas de los becados según el estado de becados en retiro			

Tabla 26:

Caso de prueba TC_004

Test Case ID	TC_004	Test Case Description	Probando el filtrado de las notas de los becarios con estado deserto en una convocatoria		
Created By	Jean Colonia	Reviewed By	Jean Colonia	Version	1.0
Tester's Name	Jean	Date Tested	10 de Agosto del 2022	Test Case (Pass/Fail/Not)	
S #	Prerequisites:		S #	Test Data	
1	Haber abierto el navegador		1	Usuario = 46596449	
2	Ingresar al sistema con las credenciales requeridas (admin).		2	Contraseña = *****	
3			3		
4			4		
Test Scenario	Verificar las notas de los becados según una convocatoria				
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Blocked	
1	Ingresar al apartado de certificados.	Visualizar la sección de gestión de certificados.			
2	Seleccionar la opción de Notas	Visualizar la sección de Notas			
3	Seleccionar el año en los criterios de búsqueda	El año seleccionado se muestra como valor en el combo de Año.			
4	Seleccionar la convocatoria en los criterios de búsqueda	La convocatoria seleccionado se muestra como valor en el combo de Año.			
5	Seleccionar el Grupo Sección en los criterios de búsqueda	El grupo sección seleccionado se muestra como valor en el combo de Grupo Sección.			
6	Seleccionar el Estado Desaprobado en el combo de Estado	El estado seleccionado se muestra como valor en el combo de Estado.			
7	Hacer clic en el botón filtrar	Se muestra el listado de las notas de los becados según el estado de becados en deserto			

El diseño del caso de prueba para el requerimiento número 02 se realizó mediante la técnica de diseño de pruebas de la Tabla de decisión.

Tabla 27:

Técnica de la tabla de decisión

CONDICIONES	Regla 1	Regla 2	Regla 3	Regla 4
Ingresar Usuario Válido	V	V	F	F
Ingresar Contraseña Válida	V	F	F	V
Resultado	V	F	F	F

Tabla 28:

Caso de prueba TC_005

Test Case ID	TC_005	Test Case Description	Probando el ingreso al sistema con las respectivas credenciales		
Created By	Jean Colonia	Reviewed By	Jean Colonia	Version	1.0
Tester's Name	Jean	Date Tested	10 de Agosto del 2022	Test Case (Pass/Fail/Not)	
S #	Prerequisites:	S #	Test Data		
1	Haber abierto el navegador	1	Usuario = 46596449		
2	Ingresar a la URL del sistema https://tomcat.expertsystems.pe/becas/#/inicio	2	Contraseña = *****		
3		3			
4		4			
Test Scenario	Verificar las notas de los becados según una convocatoria				
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Blocked	
1	Hacer clic en el botón Iniciar Sesión en el	Visualizar la interfaz de Login			
2	Ingresar el usuario	Se visualiza el usuario ingresado en el campo de usuario			
3	Ingresar la contraseña	Se visualiza el usuario ingresado en el campo de la contraseña con el formato * y su respectiva longitud			
4	Hacer click en el botón Iniciar Sesión	Visualizar el sistema con nuestra cuenta ya loggeada previamente y los privilegios correspondientes.			
5					
6					
7					

Tabla 29:
Resumen de casos de prueba

ID ESCENARIO	TEST ESCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS
TC_001	TS_001 Verificar las notas de los becados según una convocatoria	Probando el filtrado de las notas de los becarios con estado aprobado en una convocatoria	1. Haber abierto el navegador. 2. Ingresar al sistema con las credenciales requeridas (admin).	1. Ingresar al apartado de certificados. 2. Seleccionar la opción de Notas. 3. Seleccionar el año en los criterios de búsqueda 4. Seleccionar la convocatoria en los criterios de búsqueda 5. Seleccionar el Grupo Sección en los criterios de búsqueda. 6. Seleccionar el Estado Aprobado en el combo de Estado 7. Hacer clic en el botón filtrar	Usuario válido Contraseña válida	1. Visualizar la sección de gestión de certificados. 2. Visualizar la sección de Notas 3. El año seleccionado se muestra como valor en el combo de Año. 4. La convocatoria seleccionado se muestra como valor en el combo de Año. 5. El grupo sección seleccionado se muestra como valor en el combo de Grupo Sección. 6. El estado seleccionado se muestra como valor en el combo de Estado. 7. Se muestra el listado de las notas de los becados según el estado de becados en aprobado.			
TC_002	TS_001 Verificar las notas de los becados según una convocatoria	Probando el filtrado de las notas de los becarios con estado desaprobado en una convocatoria	1. Haber abierto el navegador. 2. Ingresar al sistema con las credenciales requeridas (admin).	1. Ingresar al apartado de certificados. 2. Seleccionar la opción de Notas. 3. Seleccionar el año en los criterios de búsqueda 4. Seleccionar la convocatoria en los criterios de búsqueda 5. Seleccionar el Grupo Sección en los criterios de búsqueda. 6. Seleccionar el Estado Desaprobado en el combo de Estado 7. Hacer clic en el botón filtrar	Usuario válido Contraseña válida	1. Visualizar la sección de gestión de certificados. 2. Visualizar la sección de Notas 3. El año seleccionado se muestra como valor en el combo de Año. 4. La convocatoria seleccionado se muestra como valor en el combo de Año. 5. El grupo sección seleccionado se muestra como valor en el combo de Grupo Sección. 6. El estado seleccionado se muestra como valor en el combo de Estado. 7. Se muestra el listado de las notas de los becados según el estado de becados en desaprobado..			
TC_003	TS_001 Verificar las notas de los becados según una convocatoria	Probando el filtrado de las notas de los becarios con estado deserto en una convocatoria	1. Haber abierto el navegador. 2. Ingresar al sistema con las credenciales requeridas (admin).	1. Ingresar al apartado de certificados. 2. Seleccionar la opción de Notas. 3. Seleccionar el año en los criterios de búsqueda 4. Seleccionar la convocatoria en los criterios de búsqueda 5. Seleccionar el Grupo Sección en los criterios de búsqueda. 6. Seleccionar el Estado Deserto en el combo de Estado 7. Hacer clic en el botón filtrar	Usuario válido Contraseña válida	1. Visualizar la sección de gestión de certificados. 2. Visualizar la sección de Notas 3. El año seleccionado se muestra como valor en el combo de Año. 4. La convocatoria seleccionado se muestra como valor en el combo de Año. 5. El grupo sección seleccionado se muestra como valor en el combo de Grupo Sección. 6. El estado seleccionado se muestra como valor en el combo de Estado. 7. Se muestra el listado de las notas de los becados según el estado de becados en deserto..			
TC_004	TS_001 Verificar las notas de los becados según una convocatoria	Probando el filtrado de las notas de los becarios con estado retiro en una convocatoria	1. Haber abierto el navegador. 2. Ingresar al sistema con las credenciales requeridas (admin).	1. Ingresar al apartado de certificados. 2. Seleccionar la opción de Notas. 3. Seleccionar el año en los criterios de búsqueda 4. Seleccionar la convocatoria en los criterios de búsqueda 5. Seleccionar el Grupo Sección en los criterios de búsqueda.	Usuario válido Contraseña válida	1. Visualizar la sección de gestión de certificados. 2. Visualizar la sección de Notas 3. El año seleccionado se muestra como valor en el combo de Año. 4. La convocatoria seleccionado se muestra como valor en el combo de Año. 5. El grupo sección seleccionado se muestra como valor en el combo de Grupo Sección.			

				6. Seleccionar el Estado Retiro en el combo de Estado 7. Hacer clic en el botón filtrar		6. El estado seleccionado se muestra como valor en el combo de Estado. 7. Se muestra el listado de las notas de los becados según el estado de becados en retiro.			
TC_005	TS_002 Verificar el acceso al sistema con credenciales de usuario	Probando el ingreso al sistema con las credenciales válidas	1. Haber abierto el navegador. 2. Ingresar a la URL del sistema https://tomcat.expertsystems.pe/becas/#/inicio	1. Hacer clic en el botón Iniciar Sesión en el encabezado 2. Ingresar la contraseña 3. Hacer clic en el botón Iniciar Sesión	Usuario Contraseña	1. Visualizar la interfaz de Login 2. Se visualiza el usuario ingresado en el campo de usuario 3. Se visualiza el usuario ingresado en el campo de la contraseña con el formato * y su respectiva longitud 4. Visualizar el sistema con nuestra cuenta ya loggeada previamente y los privilegios correspondientes.			
TC_006	TS_002 Verificar el acceso al sistema con credenciales de usuario	Probando el ingreso al sistema con la usuario y contraseña inválida	1. Hacer clic en el botón Iniciar Sesión en el encabezado 2. Ingresar la contraseña 3. Hacer clic en el botón Iniciar Sesión	1. Hacer clic en el botón Iniciar Sesión en el encabezado 2. Ingresar la contraseña 3. Hacer clic en el botón Iniciar Sesión	Usuario Contraseña	1. Visualizar la interfaz de Login 2. Se visualiza el usuario ingresado en el campo de usuario 3. Se visualiza el usuario ingresado en el campo de la contraseña con el formato * y su respectiva longitud 4. Visualizar el sistema con nuestra cuenta ya loggeada previamente y los privilegios correspondientes.			
TC_007	TS_002 Verificar el acceso al sistema con credenciales de usuario	Probando el ingreso al sistema con el usuario inválido	1. Hacer clic en el botón Iniciar Sesión en el encabezado 2. Ingresar la contraseña 3. Hacer clic en el botón Iniciar Sesión	1. Hacer clic en el botón Iniciar Sesión en el encabezado 2. Ingresar la contraseña 3. Hacer clic en el botón Iniciar Sesión	Usuario Contraseña	1. Visualizar la interfaz de Login 2. Se visualiza el usuario ingresado en el campo de usuario 3. Se visualiza el usuario ingresado en el campo de la contraseña con el formato * y su respectiva longitud 4. Visualizar el sistema con nuestra cuenta ya loggeada previamente y los privilegios correspondientes.			
TC_008	TS_002 Verificar el acceso al sistema con credenciales de usuario	Probando el ingreso al sistema con la contraseña inválida	1. Hacer clic en el botón Iniciar Sesión en el encabezado 2. Ingresar la contraseña 3. Hacer clic en el botón Iniciar Sesión	1. Hacer clic en el botón Iniciar Sesión en el encabezado 2. Ingresar la contraseña 3. Hacer clic en el botón Iniciar Sesión	Usuario Contraseña	1. Visualizar la interfaz de Login 2. Se visualiza el usuario ingresado en el campo de usuario 3. Se visualiza el usuario ingresado en el campo de la contraseña con el formato * y su respectiva longitud 4. Visualizar el sistema con nuestra cuenta ya loggeada previamente y los privilegios correspondientes.			

Casos de prueba en gestor de tareas

Para la aplicación y gestión de los casos de prueba se trabajó con la aplicación de TestRail como estrategia de gestión y también con la actual aplicación llamada Clickup para contrastar los beneficios y desventajas.

Figura 48:

Caso de prueba con TestRail

The screenshot shows a TestRail interface for a project named 'DEVIDA-PROJECT'. The 'TEST CASES' tab is active, displaying a test case titled 'Probando el filtrado de las notas de los becarios con estado aprobado en una convocatoria'. The test case details include:

- Type:** Functional
- Priority:** Medium
- Estimate:** 2 hours
- References:** UH - Visualizar notas de los becarios
- Automation Type:** Ranorex

Preconditions:

1. Haber abierto el navegador
2. Ingresar al sistema con las credenciales requeridas (admin)

Steps:

Step	Step Description	Expected Result
1	Seleccionar la opción de Notas.	Visualizar la sección de gestión de certificados.
2	Seleccionar la opción de Notas	Visualizar la sección de Notas
3	Seleccionar el año en los criterios de búsqueda	El año seleccionado se muestra como valor en el combo de Año
4	El año seleccionado se muestra como valor en el combo de Año	La convocatoria seleccionada se muestra como valor en el combo de Año.
5	Seleccionar el Grupo Sección en los criterios de búsqueda	El grupo sección seleccionado se muestra como valor en el combo de Grupo Sección.

Actual aplicación para gestión de tareas (requerimientos, bugs, otros)

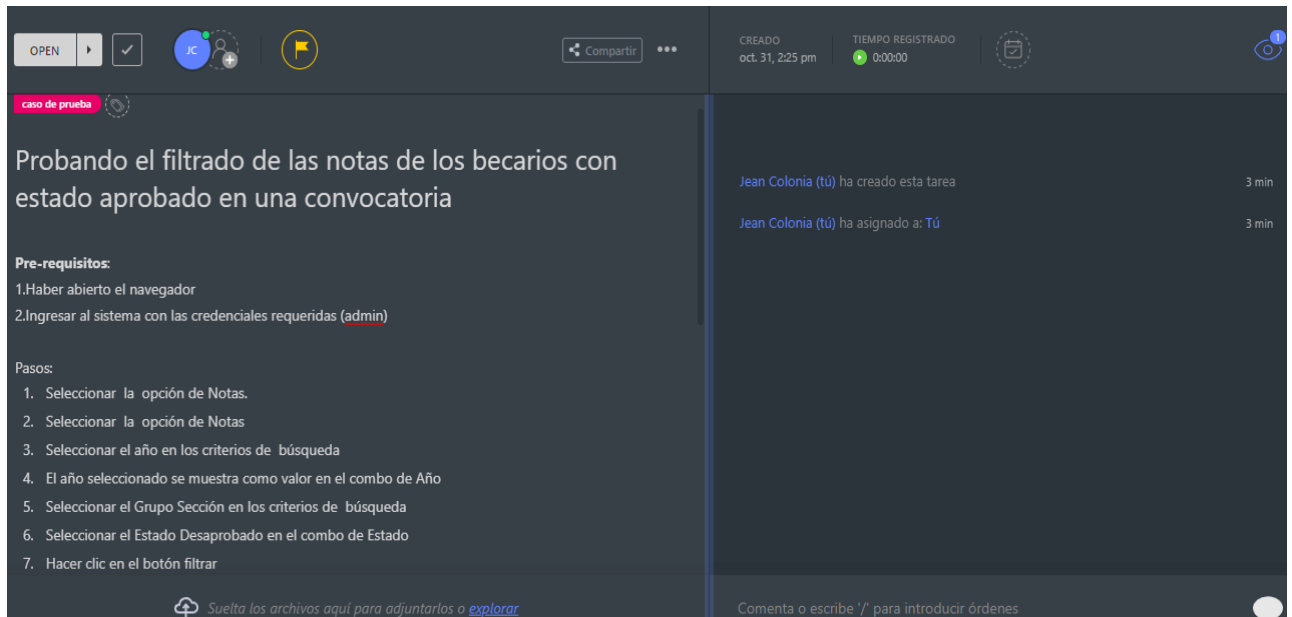
Figura 49:

Visualización de caso de prueba - Clickup

The screenshot shows a Clickup task card for the same test case. The card is titled 'Probando el filtrado de las notas de los becarios con estado aprobado en una convocatoria' and is currently in an 'OPEN' state. The card includes a progress indicator (1 TAREA), a person icon for assignment, a calendar icon for due date, and a flag icon for priority. A '+ Nueva tarea' button is visible at the bottom left of the card.

Figura 50:

Detalle del caso de prueba - Clickup

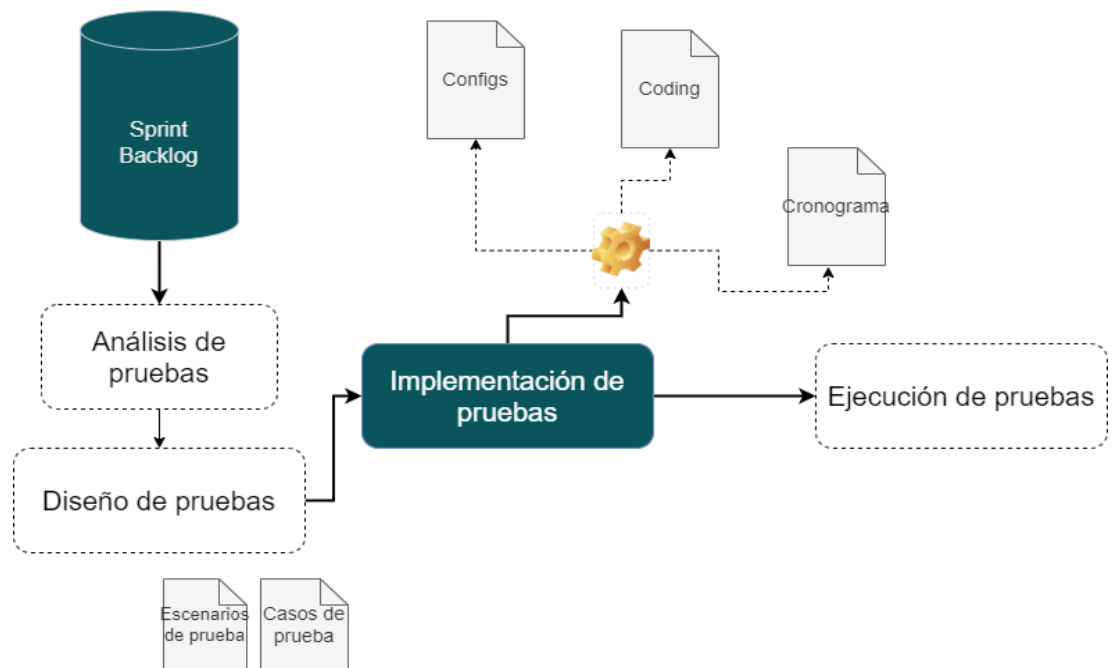


4.1.2.5 Implementación de pruebas

La implementación de pruebas responde a la pregunta de si todo está listo para que se puedan ejecutar las pruebas.

Figura 51:

Implementación de pruebas



Algunas actividades contenidas dentro de esta fase son:

- El desarrollo y la priorización de los procedimientos de prueba (secuencia de acciones para la ejecución de pruebas, scripts de prueba).
- Crear Suites, casos de prueba que se ejecutan en bloque.
- Crear el arnes del entorno de pruebas (configuraciones para ejecutar las pruebas).
- Preparación de los datos de pruebas.

En la fase de la implementación de pruebas se realizó las configuraciones correspondientes y necesarias para cada nivel y tipo de prueba, según lo descrito anteriormente como ejemplo se tomaron algunas pruebas las cuales se realizan de manera funcional, servicios, rendimiento, estructura y otros.

Configuraciones

Figura 52:

Configuraciones de ambiente de pruebas

TIPO	Característica	Descripción
Servicios	Manual	Instalación de POSTMAN y configuración de ambiente, Collections, otros.
	Automatizada	Instalación de dependencias para Karate DSL.
Funcional	Manual	Configuración de navegadores.
	Automatizado	Instalación de dependencias para Selenium. Instalación de dependencias para WebdriverIO
Estructura	Manual	Descarga de dependencias y ejecución de proyectos backend – frontend.
	Automatización	Instalación y configuración de Sonarqube.
Rendimiento	Automatizado	Instalación y configuración de Jmeter y Blazemeter.

Configuración y Código fuente para prueba de servicios – manual: Postman

Figura 53:

Configuración de Enviroment en Postman

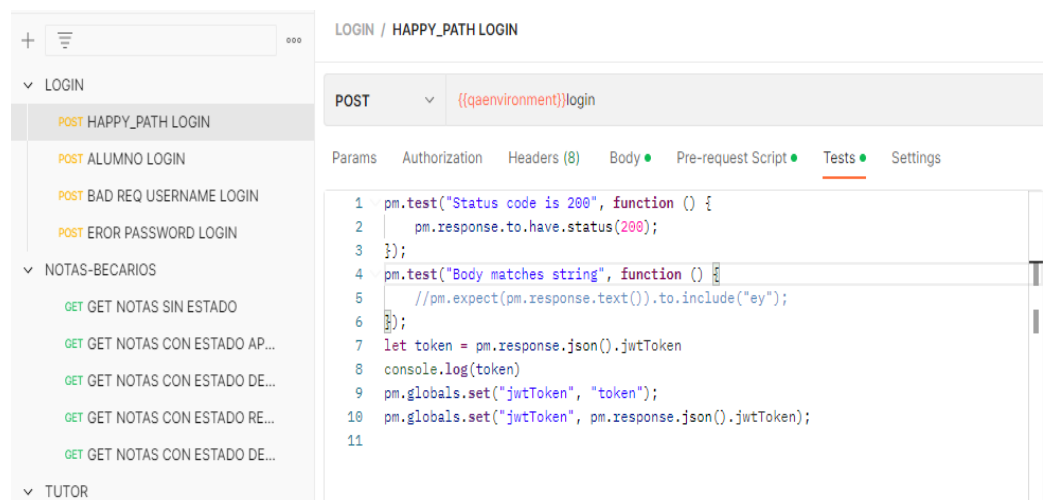
	VARIABLE	TYPE ⓘ	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ	...	Persist All	Reset All
<input checked="" type="checkbox"/>	host	default ▾	localhost:8080/	localhost:8080/			
<input checked="" type="checkbox"/>	login	default ▾	login	login			
<input checked="" type="checkbox"/>	tutor	default ▾	tutor	tutor			
<input checked="" type="checkbox"/>	jwtToken	default ▾	eyJhbGciOiJIUzUxMiJ9.eyJhcG...	eyJhbGciOiJIUzUxMiJ9.eyJhcGVsbGkiOiB...			
<input checked="" type="checkbox"/>	tutorActivo	default ▾	0	0			
<input checked="" type="checkbox"/>	tutorInactivo	default ▾	2	2			
<input checked="" type="checkbox"/>	idTutor	default ▾					
<input checked="" type="checkbox"/>	EliminadoPor	default ▾	jeancolonia	jeancolonia			

La configuración para los casos de prueba TC_005, TC_006, TC_007 y TC_008 que son: Verificar el acceso al sistema con credenciales de usuario.

La implementación de la visualización de los resultados se manejó con dos herramientas para las pruebas manuales de servicios, postman y con Newman reporting templates.

Figura 54:

Configuración extra y manejo de variables en Postman



The screenshot shows the Postman interface for a test named 'LOGIN / HAPPY_PATH LOGIN'. The test is a POST request to the endpoint 'login' using the 'qaenvironment' environment. The 'Tests' tab is active, displaying the following JavaScript code:

```
1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4 pm.test("Body matches string", function () {
5   //pm.expect(pm.response.text()).to.include("ey");
6 });
7 let token = pm.response.json().jwtToken
8 console.log(token)
9 pm.globals.set("jwtToken", "token");
10 pm.globals.set("jwtToken", pm.response.json().jwtToken);
11
```

La configuración para los casos de prueba TC_001, TC_002, TC_003 y TC_004 que son: Verificar el acceso al sistema con credenciales de usuario.

Figura 55:

Asignación de ambiente y token a Notas Becarios - Postman

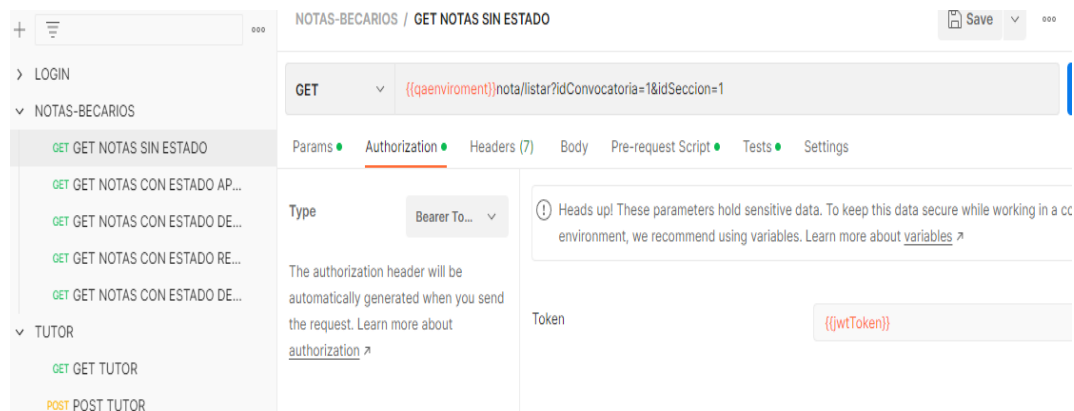


Figura 56:

Test Scripts en Notas de Becarios – Postman



Configuración para prueba de servicios – automatizado: Karate DSL

Para las pruebas automatizadas de los servicios se realizó mediante Karate DSL como un stack tecnológico innovador y sumamente sencillo e importante, para lo cual se realizó la instalación y previa configuración de otros complementos.

Maven

Es una herramienta de gestión y construcción para aplicaciones Java con un modelo de configuración basado en XML.

Figura 57:

Instalación de Maven

```
C:\Users\jcolonia>mvn --version
Apache Maven 3.8.6 (84538c9988a25aec085021c365c560670ad80f63)
Maven home: C:\Program Files\apache-maven-3.8.6
Java version: 11.0.15, vendor: Eclipse Adoptium, runtime: C:\Program Files\Eclipse Adoptium\jdk-11.0.15.10-hotspot
Default locale: en_US, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```

Figura 58:

Creación de proyecto Karate DSL

```
mvn archetype:generate \
-DarchetypeGroupId=com.intuit.karate \
-DarchetypeArtifactId=karate-archetype \
-DarchetypeVersion=1.2.0 \
-DgroupId=com.mycompany \
-DartifactId=myproject
```

Figura 59:

Configuración de paquetes en POM de Maven

```
4
5 <groupId>org.example</groupId>
6 <artifactId>karate-dsl</artifactId>
7 <version>1.0-SNAPSHOT</version>
8 <packaging>jar</packaging>
9
10 <properties>
11   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
12   <java.version>1.8</java.version>
13   <maven.compiler.version>3.8.1</maven.compiler.version>
14   <maven.surefire.version>2.22.2</maven.surefire.version>
15   <karate.version>1.2.0</karate.version>
16 </properties>
17
18 <dependencies>
19   <dependency>
20     <groupId>com.intuit.karate</groupId>
21     <artifactId>karate-junit5</artifactId>
22     <version>1.2.0</version>
23     <scope>test</scope>
```


Figura 60:

Estructura del proyecto de Karate DSL

```
1  src/test/java
2  |
3  +-- karate-config.js
4  +-- logback-test.xml
5  +-- some-reusable.feature
6  +-- some-classpath-function.js
7  +-- some-classpath-payload.json
8  |
9  \-- devida-project-services
10 |
11 |   +-- devidaProjectServicesTest.java
12 |   |
13 |   +-- login
14 |   |   |
15 |   |   +-- login-post.feature
16 |   |   +-- credentialsData.json
17 |   |   \-- LoginRunner.java
18 |   |
19 |   \-- notas
20 |       |
21 |       +-- notas.feature
22 |       +-- notasData.json
23 |       +-- helpers.js
24 |       \-- NotasRunner.java
```

La configuración para los casos de prueba TC_005, TC_006, TC_007 y TC_008 que son: Verificar el acceso al sistema con credenciales de usuario, de los cuales a continuación se muestra un extracto del código.

Figura 61:

Scenario - TC_005: Visualizar las notas de los becarios sin filtros

```
Scenario: visualizar las notas de los becarios sin ningun estado
* def queryW0Parameters = { idConvocatoria:1, idSeccion:1}
Given path '/nota/listar'
And params queryW0Parameters
And header Authorization = jwtToken
When method get
Then status 200
And assert responseStatus = 200
* print response
```

Figura 62: Scenario Outline - TC_005: Visualizar las notas de los becarios sin filtros

```
## using Scenario Outline with multiple data insertion in Example Table
Scenario Outline: visualizar las notas de los becarios con todos los estados
* def queryAprobadoParameters = { idConvocatoria:<idConvocatoriaValue>, idSeccion:<idSeccionValue>, idCondicionFinal:<idCondicionFinalValue>}
Given path '/nota/listar'
And params queryAprobadoParameters
And header Authorization = jwtToken
When method get
Then status 200
And assert responseStatus = 200
* print response
Examples:
| idConvocatoriaValue | idSeccionValue | idCondicionFinalValue |
| 1 | 1 | 1 |
| 1 | 1 | 2 |
| 1 | 1 | 3 |
| 1 | 1 | 4 |
```

La configuración para los casos de prueba TC_001, TC_002, TC_003 y TC_004 que son: Verificar el acceso al sistema con credenciales de usuario donde se muestra un extracto del código fuente de las pruebas de servicios automatizadas realizado.

Figura 63:

Scenario Outline - TC_001: Verificar login con credenciales de usuarios

```
Feature: Testing devida project login with user credentials
Scenario Outline: test user login with multiple data insertion
  * def userCredentials = {username:<username>, password: <password>}
  Given url
  And request userCredentials
  When method POST
  Then status 200
  Examples:
  | username           | password           |
  | "46596449"         | "12345678"         |
  | "312321"           | "12345dasdas312312678" |
  | "46596449"         | "00000"           |
  | "asdas321312000"   | "12345678"         |
```

Figura 64: Scenario - TC_001: Verificar login con credenciales de usuarios

```
Scenario: test user login
  * def userCredentials =
  """
  {
  username:"46596449",
  password: "12345678"
  }
  """
  Given path '/login'
  And request userCredentials
  When method POST
  Then status 200
```

Figura 65:

Main Test Class Runner

```
package devidaProjectServices;

import com.intuit.karate.Results;
import com.intuit.karate.Runner;
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;

class ExamplesTest {

    @Test
    void testParallel() {
        Results results = Runner.path( ..paths: "classpath:devidaProjectServices")
            // .outputCucumberJson(true)
            .parallel( threadCount: 5);
        assertEquals( expected: 0, results.getFailCount(), results.getErrorMessages());
    }
}
```

Reporte en Newman Reporting Templates

Otra opción que se trabajó como estrategia fueron los reportes de las pruebas manuales mediante Postman y Newman, el cual es una extensión que permite una mejor gestión de los resultados obtenidos de las Collections creadas previamente en Postman.

Figura 66:

Instalación de Newman

```
Kanye@DESKTOP-OP8K3P1 MINGW64 ~  
$ npm install -g newman-reporter-htmlextra
```

Figura 67:

Ejecución de Collection para generación del Reporte

```
zyro-image.png  
Kanye@DESKTOP-OP8K3P1 MINGW64 ~/Downloads  
$ newman run postman_collection_2.json -e testEnvNewman.json -r htmlextra
```

Pruebas funcionales

Para el caso de las pruebas manuales solo se tuvo en cuenta la instalación de algunos navegadores como Chrome, Mozilla Firefox, Brave, Edge u otro requerido como soporte para las pruebas manuales.

Pruebas Funcionales Automatizadas

Un aspecto importante que fue soporte de los proyectos para los diversos tipos de pruebas fue Node, donde se realizó la instalación del mismo previamente.

Figura 68:

Node.js

```
Kanye@DESKTOP-OP8K3P1 MINGW64 ~  
$ node --version  
v16.13.0  
Kanye@DESKTOP-OP8K3P1 MINGW64 ~  
$ npm --v  
8.1.0
```

Figura 69:

Creación del proyecto para las pruebas funcionales Automatizadas con WebDriverIO

```
ianye@DESKTOP-0P8K3P1 MINGW64 ~/Desktop/devida-project-webdriverIO
$ npm init wdio .
Need to install the following packages:
  create-wdio
Ok to proceed? (y) y
] / reify:create-wdio: sill audit bulk request {
```

Figura 70:

Creación del proyecto para pruebas funcionales con Selenium

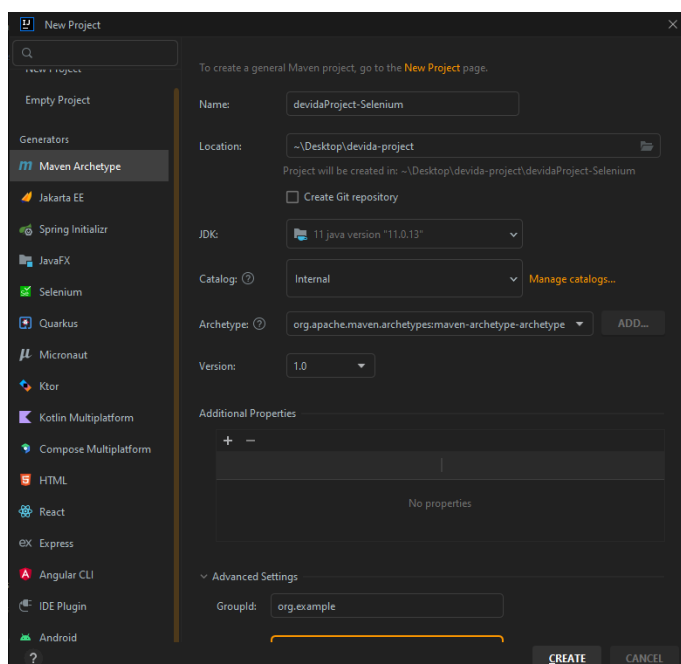
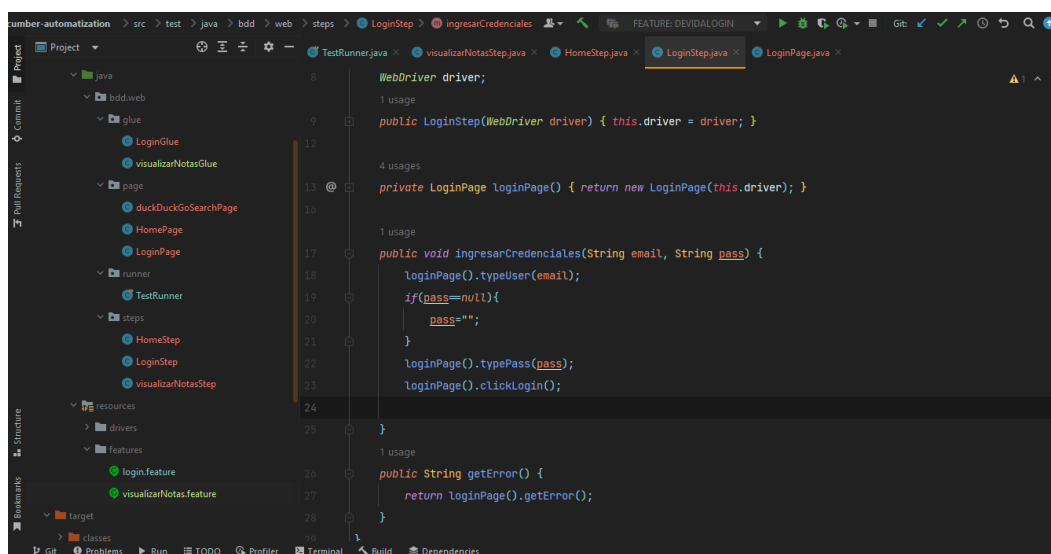


Figura 71:

Visualización de código fuente de pruebas automatizadas con Selenium



Pruebas de Rendimiento

Para este tipo de pruebas se tuvo en cuenta 2 herramientas importantes como Blazemeter y Jmeter con la finalidad de medir el rendimiento del proyecto de software desarrollado por la empresa que fue contexto de investigación.

Blazemeter sirvió como grabador de los scripts cuando se realizaron las acciones requeridas dentro del portal y fueron importadas en jmeter mediante un archivo jmx.

Figura 72:

Blazemeter

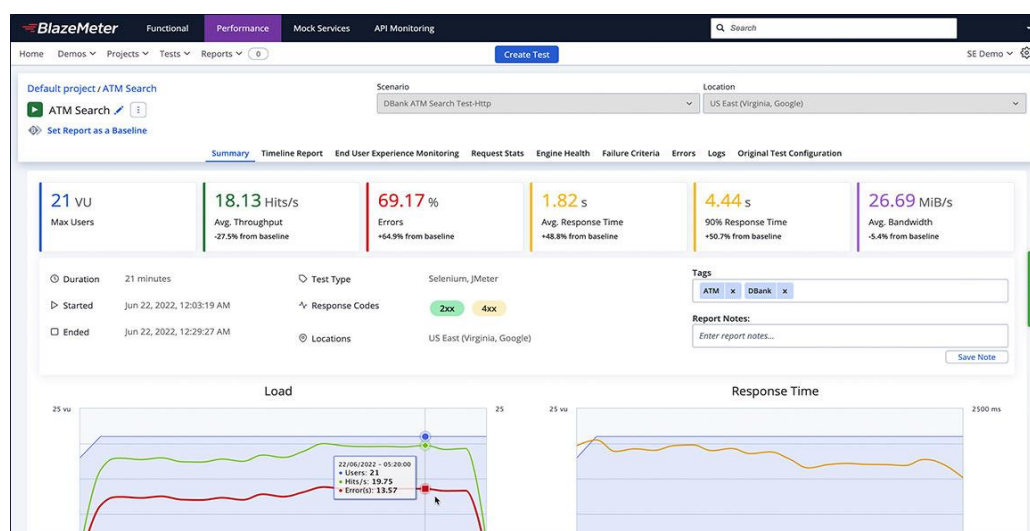
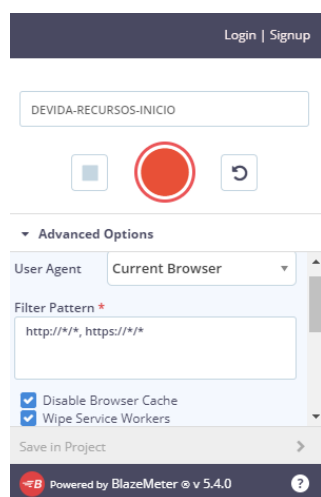


Figura 73:

Blazemeter Recorder



Jmeter como herramienta para la ejecución de los scripts previamente generados por Blazemeter y también para en análisis de los resultados.

Figura 74: Jmeter

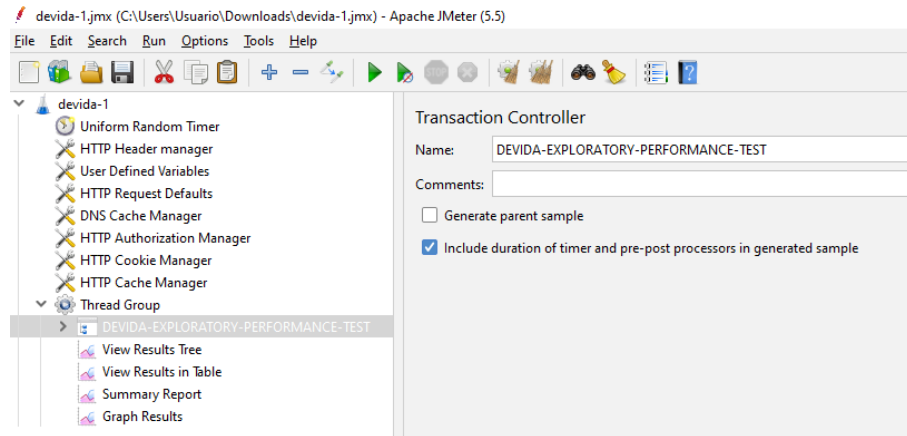


Figura 75: Configuración para las pruebas en Jmeter

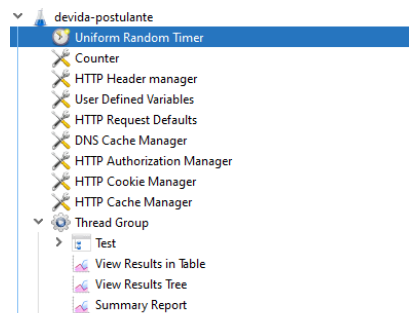
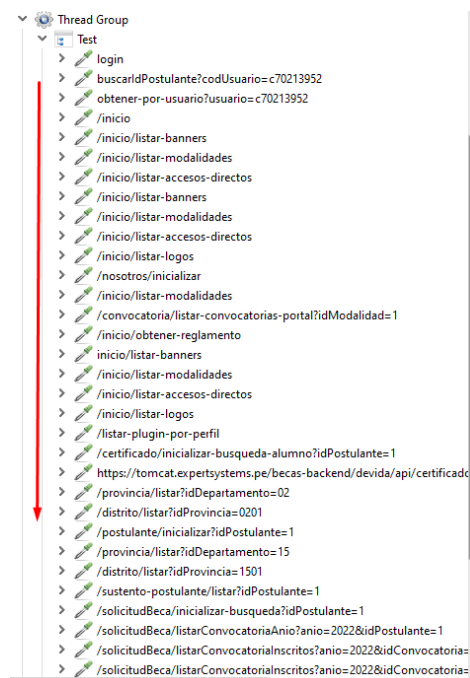


Figura 76: Peticiones en Jmeter

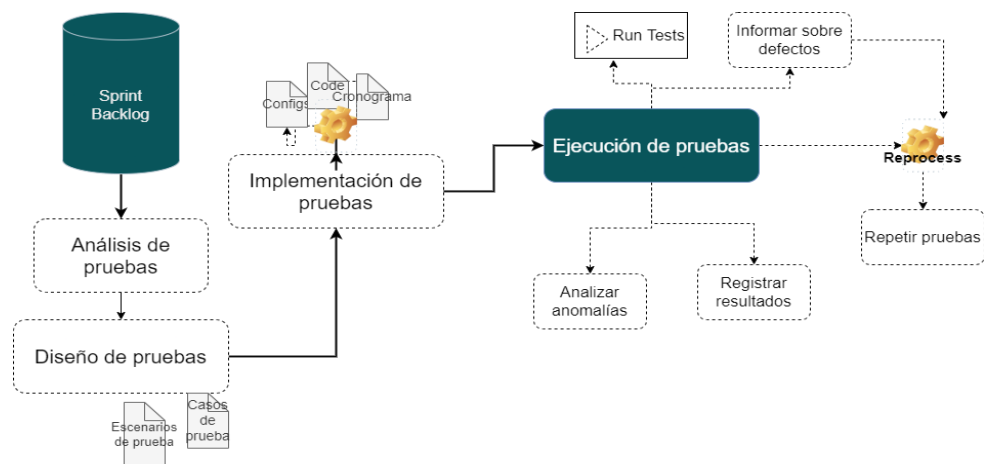


4.1.2.6 Ejecución de pruebas

En la fase de la ejecución de las pruebas se prioriza como tal los casos de prueba, los cuales previamente han sido configurados y se han generado su código fuente correspondientemente para su ejecución, análisis de anomalías, informes sobre defectos existentes y repetir la ejecución de las pruebas si fuera necesario.

Figura 77:

Ejecución de pruebas



La Ejecución de los casos de prueba TC_005, TC_006, TC_007 y TC_008 que fueron: Verificar el acceso al sistema con credenciales de usuario, de los cuales a continuación se muestra un extracto de la ejecución.

Figura 78:

Run Collection Login - Postman Report

RUN SUMMARY			1	2
POST	HAPPY_PATH LOGIN	4 0	✓	✓
	Pass Status code is 200		✓	✓
	Pass Body matches string		✓	✓
POST	ALUMNO LOGIN	4 0	✓	✓
	Pass Status code is 200		✓	✓
	Pass Body matches string		✓	✓
POST	BAD REQ USERNAME LOGIN	0 4	✗	✗
	Fail Status code is 200		✗	✗
	Fail Body matches string		✗	✗
POST	EROR PASSWORD LOGIN	0 4	✗	✗
	Fail Status code is 200		✗	✗
	Fail Status code name has string		✗	✗

La ejecución para los casos de prueba TC_001, TC_002, TC_003 y TC_004 que fueron: Verificar las notas de los becarios que culminaron una convocatoria.

Figura 79:

Run Collection Notas Becarios - Postman Report



Reporte en Newman Reporting Templates

Los resultados de las ejecuciones de los casos de prueba también se realizaron mediante esta herramienta previamente instalada y configurada.

Figura 80:

Ejecución de Newman - Casos de prueba de Login Collection

	executed	failed
iterations	1	0
requests	7	0
test-scripts	4	0
prerequisite-scripts	3	0
assertions	8	4
total run duration: 5.5s		
total data received: 60.1kB (approx)		
average response time: 702ms [min: 212ms, max: 2.5s, s.d.: 754ms]		

Figura 81:

Resumen del resultado - Caso de prueba Login



Figura 82:

Resultado para el caso de prueba TC_005

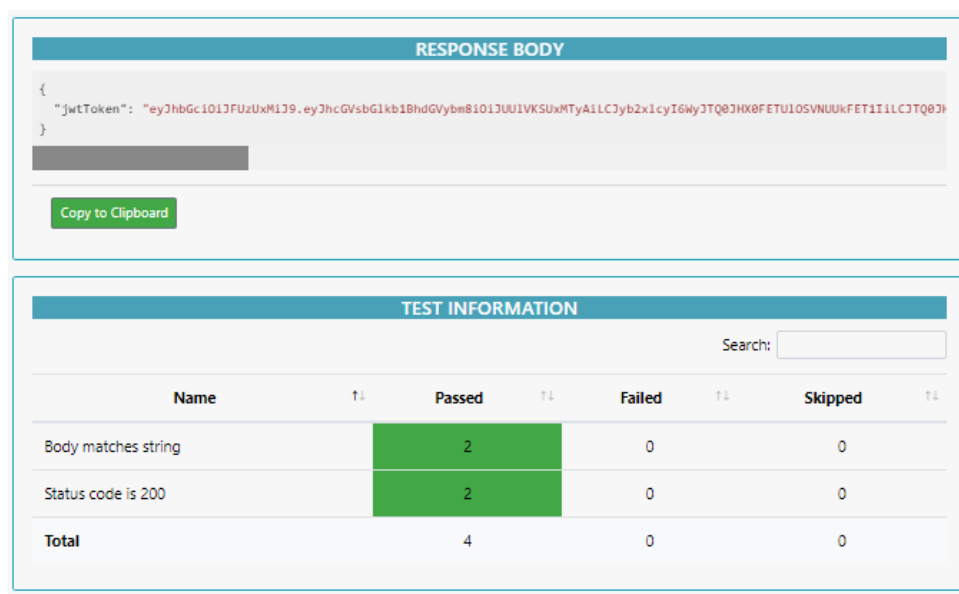


Figura 83: Resultado para el caso de prueba TC_007

RESPONSE BODY

```

{
  "timestamp": "2022-11-03T15:42:44.054+00:00",
  "status": 401,
  "error": "Unauthorized",
  "trace": "org.springframework.web.server.ResponseStatusException: 401 UNAUTHORIZED (\"Falla en la autenticación\"); nested e
  \"message\": \"Falla en la autenticación\",
  \"path\": \"/becas-backend/login\"
}

```

Copy to Clipboard

TEST INFORMATION

Search:

Name	Passed	Failed	Skipped
Body matches string	0	2	0
Status code is 200	0	2	0
Total	0	4	0

TEST FAILURES

Test Name	Assertion Error
Status code is 200	expected response to have status code 200 but got 401
Body matches string	expected '{\"timestamp\": \"2022-11-03T15:42:44.054+00:00\", \"status\": 401, \"error\": \"Unauthorized\", \"tra

Figura 84:

Resumen del resultado - Caso de prueba Notas de becarios

Summary
Total Requests 9
Failed Tests 0
Skipped Tests 0

Newman Run Dashboard

Thursday, 03 November 2022 11:01:42

TOTAL ITERATIONS

1

TOTAL ASSERTIONS

10

TOTAL FAILED TESTS

0

TOTAL SKIPPED TESTS

0

FILE INFORMATION

Collection: NOTAS-BECARIOS

TIMINGS AND DATA

⌚ Total run duration: 6.3s
📄 Total data received: 58.41KB
⌚ Average response time: 651ms

SUMMARY ITEM	TOTAL	FAILED
Requests	9	0
Prerequest Scripts	1	0
Test Scripts	5	0
Assertions	10	0
Skipped Tests	0	-

Ejecución de las pruebas de servicios automatizadas mediante Karate DSL

Figura 85:

Ejecución de caso de prueba Login - Karate DSL

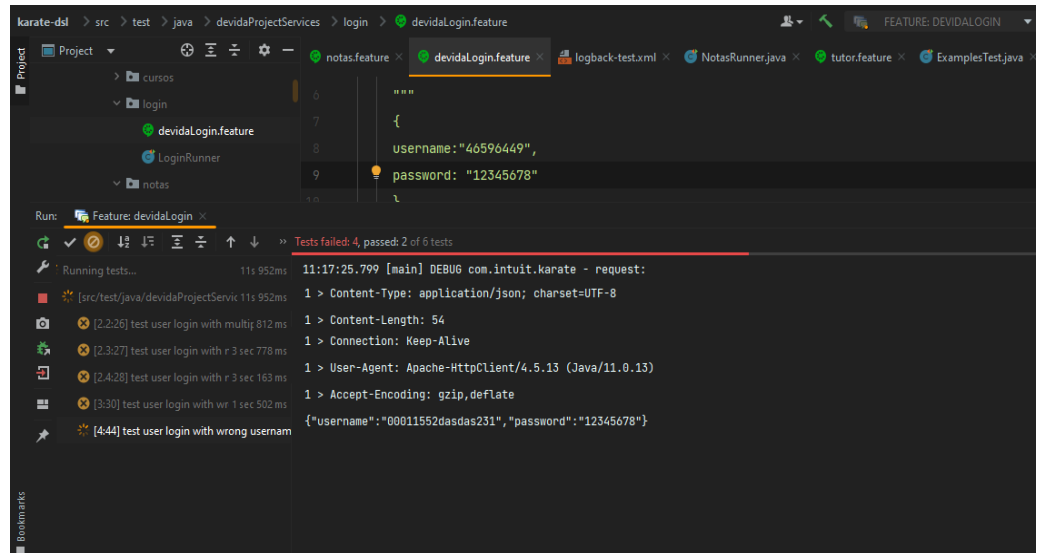


Figura 86:

Resumen de los resultados de la ejecución de caso de prueba Login

Feature	Title	Passed	Failed	Scenarios	Time (ms)
src/test/java/devidaProjectServices/login/devidaLogin.feature	Probando el ingreso al sistema con credenciales de usuario	2	6	8	14312

Figura 87:

Detalle de los resultados de ejecución del caso de prueba Login

Scenario	Time (ms)
[1:4] test user login	2104
[2:1:25] test user login with multiple data insertion	594
[2:2:26] test user login with multiple data insertion	812
[2:3:27] test user login with multiple data insertion	3778
[2:4:28] test user login with multiple data insertion	594
[3:30] test user login with wrong credentials	0
[4:44] test user login with wrong username	811
[5:60] test user login with wrong password	0

Figura 88:

Ejecución de caso de prueba Notas de becarios - Karate DSL

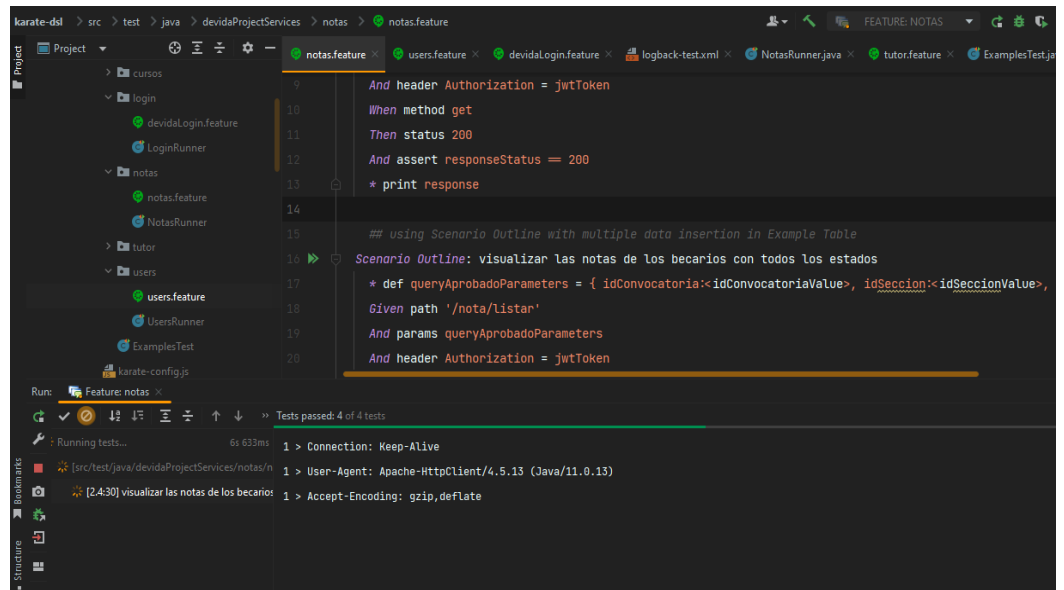


Figura 89:

Resumen de los resultados de la ejecución de caso de prueba Notas de becarios

Feature	Title	Passed	Failed	Scenarios	Time (ms)
src/test/java/devidaProjectServices/notas/notas.feature	Probando el apartado de las notas de los becarios con diversos filtros	9	0	9	16700

Figura 90:

Detalle de los resultados de ejecución del caso de prueba Notas de becarios

Scenario	Time (ms)
Scenario: [1:5] visualizar las notas de los becarios sin ningun estado	ms: 2300
Background:	
6 * def queryWOParameters = { idConvocatoria:1, idSeccion:1}	72
7 Given path '/nota/listar'	1
8 And params queryWOParameters	10
9 And header Authorization = jwtToken	0
10 When method get	2196
11 Then status 200	0
12 And assert responseStatus == 200	2
13 * print response	19
Scenario: [2:1:27] visualizar las notas de los becarios con todos los estados	ms: 648
Background:	
17 * def queryAprobadoParameters = { idConvocatoria:1, idSeccion:1, idCondicionFinal:1}	0
18 Given path '/nota/listar'	0
19 And params queryAprobadoParameters	0
20 And header Authorization = jwtToken	0
21 When method get	838
22 Then status 200	0
23 And assert responseStatus == 200	0
24 * print response	8
Scenario: [2:2:28] visualizar las notas de los becarios con todos los estados	ms: 2092
Background:	
17 * def queryAprobadoParameters = { idConvocatoria:1, idSeccion:1, idCondicionFinal:2}	0
18 Given path '/nota/listar'	0
19 And params queryAprobadoParameters	0

Ejecución de pruebas de rendimiento

Previa instalación, configuración y elaboración de peticiones para los casos de prueba se realizó la ejecución.

Figura 91:

Vista de Árbol de los resultados ejecutados en jmeter

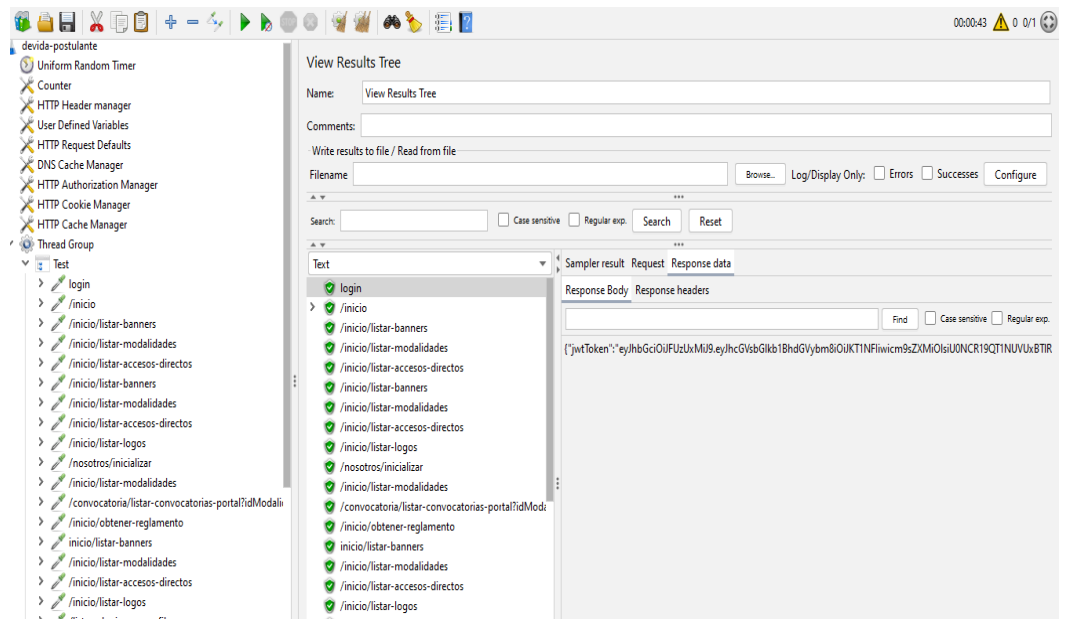


Figura 92:

Vista en Tabla de los resultados ejecutados en jmeter

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time...
1	14:11:02.586	Thread Group ...	login	1593	✓	960	525	1593	1052
2	14:11:07.210	Thread Group ...	/inicio	7780	✓	3139027	3897	503	0
3	14:11:15.212	Thread Group ...	/inicio/listar-b...	919	✓	685	903	919	0
4	14:11:16.717	Thread Group ...	/inicio/listar-b...	196	✓	1542	907	196	0
5	14:11:17.340	Thread Group ...	/inicio/listar-a...	194	✓	748	912	194	0
6	14:11:17.875	Thread Group ...	/inicio/listar-b...	205	✓	685	903	205	0
7	14:11:18.209	Thread Group ...	/inicio/listar-b...	188	✓	1542	907	188	0
8	14:11:18.649	Thread Group ...	/inicio/listar-a...	188	✓	748	912	188	0
9	14:11:19.031	Thread Group ...	/inicio/listar-l...	308	✓	1237	901	308	0
10	14:11:19.906	Thread Group ...	/nosotros/inic...	311	✓	1595	902	311	0
11	14:11:20.456	Thread Group ...	/nosotros/inic...	628	✓	1542	907	628	0
12	14:11:21.182	Thread Group ...	/convocatoria...	444	✓	22790	929	272	0
13	14:11:21.963	Thread Group ...	/inicio/obten...	363	✓	1147	907	363	0
14	14:11:22.562	Thread Group ...	inicio/listar-b...	366	✓	685	903	366	0
15	14:11:23.100	Thread Group ...	/inicio/listar-b...	363	✓	1542	907	363	0
16	14:11:24.167	Thread Group ...	/inicio/listar-a...	363	✓	748	912	363	0
17	14:11:25.255	Thread Group ...	/inicio/listar-l...	366	✓	1237	901	366	0
18	14:11:26.343	Thread Group ...	/listar-plugin-...	364	✗	655	971	364	0
19	14:11:27.076	Thread Group ...	/certificado/i...	287	✗	663	922	287	0
20	14:11:28.341	Thread Group ...	https://tomca...	412	✗	662	928	411	0
21	14:11:29.333	Thread Group ...	/provincia/list...	404	✓	5983	909	404	0
22	14:11:30.484	Thread Group ...	/distrito/listar...	414	✓	4064	907	414	0

Pruebas de estructura

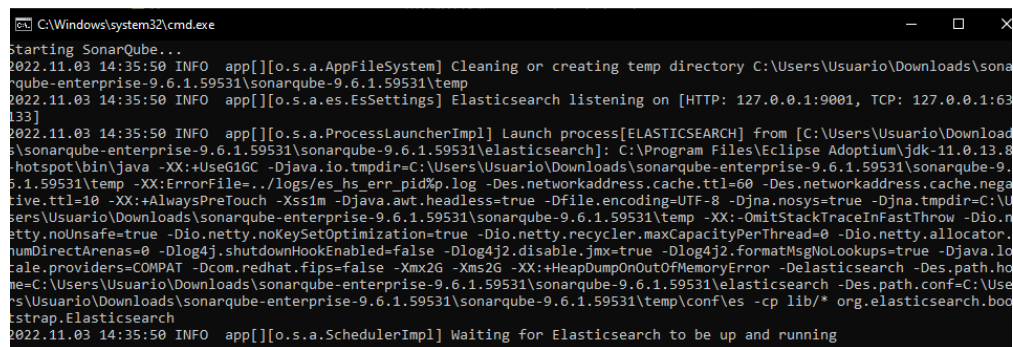
En las pruebas de estructura se realizó de forma manual como automatizada, teniendo en consideración el código fuente del backend y frontend del proyecto que estaban desarrollando en ese momento.

Las pruebas manuales se realizaron con el objetivo de conocer la arquitectura, código en específico y otros como conocimiento general y que se podía mejorar.

Las pruebas automatizadas se realizaron con la herramienta Sonarqube, la cual tuvo como alcance al código fuente tanto del backend y frontend.

Pruebas de estructura automatizadas

Figura 93: Ejecución de Sonarqube



```
C:\Windows\system32\cmd.exe
Starting SonarQube...
2022.11.03 14:35:50 INFO app[[]o.s.a.AppFileSystem] Cleaning or creating temp directory C:\Users\Usuario\Downloads\sonarqube-enterprise-9.6.1.59531\sonarqube-9.6.1.59531\temp
2022.11.03 14:35:50 INFO app[[]o.s.a.es.EsSettings] Elasticsearch listening on [HTTP: 127.0.0.1:9001, TCP: 127.0.0.1:63433]
2022.11.03 14:35:50 INFO app[[]o.s.a.ProcessLauncherImpl] Launch process[ELASTICSEARCH] from [C:\Users\Usuario\Downloads\sonarqube-enterprise-9.6.1.59531\sonarqube-9.6.1.59531\elasticsearch]; C:\Program Files\Eclipse Adoptium\jdk-11.0.13.8-hotspot\bin\java -XX:+UseG1GC -Djava.io.tmpdir=C:\Users\Usuario\Downloads\sonarqube-enterprise-9.6.1.59531\sonarqube-9.6.1.59531\temp -XX:ErrorFile=~/logs/es_hs_err_pid%p.log -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreTouch -Xss1m -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djna.nosys=true -Djna.tmpdir=C:\Users\Usuario\Downloads\sonarqube-enterprise-9.6.1.59531\sonarqube-9.6.1.59531\temp -XX:-OmitStackTraceInFastThrow -Dio.netty.noUnsafe=true -Dio.netty.noKeySetOptimization=true -Dio.netty.recycler.maxCapacityPerThread=0 -Dio.netty allocator.numDirectArenas=0 -Dlog4j.shutdownHookEnabled=false -Dlog4j2.disable.jmx=true -Dlog4j2.formatMsgNoLookups=true -Djava.locale.providers=COMPAT -Dcom.redhat.fips=false -Xmx2G -Xms2G -XX:+HeapDumpOnOutOfMemoryError -Delasticsearch -Des.path.home=C:\Users\Usuario\Downloads\sonarqube-enterprise-9.6.1.59531\sonarqube-9.6.1.59531\elasticsearch -Des.path.conf=C:\Users\Usuario\Downloads\sonarqube-enterprise-9.6.1.59531\sonarqube-9.6.1.59531\temp\conf\es -cp lib/* org.elasticsearch.bootstrap.Elasticsearch
2022.11.03 14:35:50 INFO app[[]o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
```

Figura 94:

Ejecución del Backend con Sonarqube

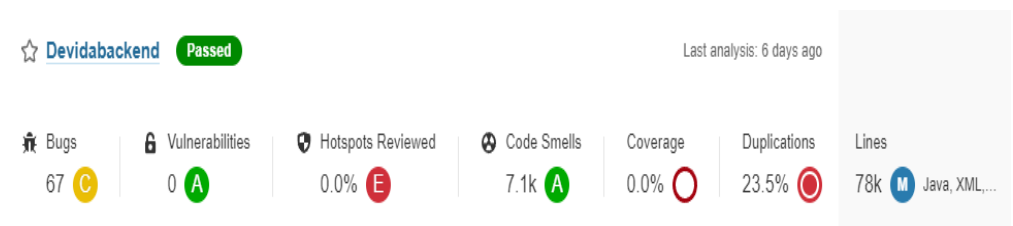
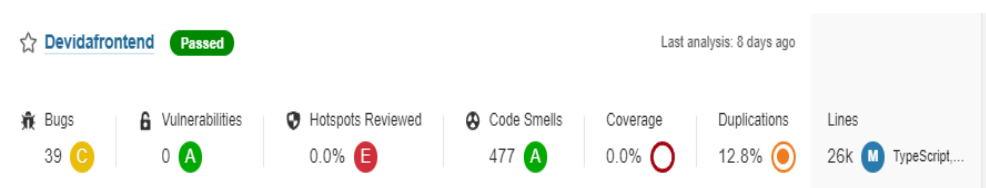


Figura 95:

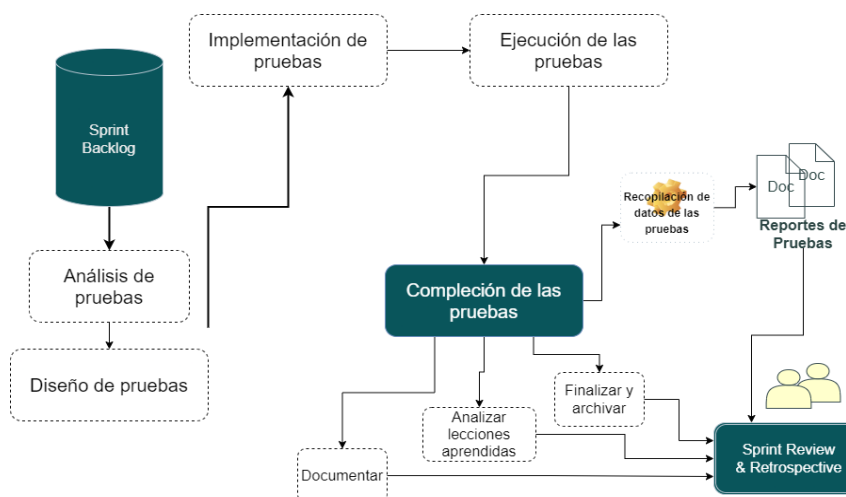
Ejecución del Frontend con Sonarqube



4.1.2.7 Compleción de pruebas

En la fase de completación de las pruebas se realizó la recopilación de los datos al final de cada sprint y se prioriza la comprobación del producto entregado, el cierre de informes existentes, finalizar y archivar, analizar las lecciones aprendidas donde todos estos elementos se mencionan en la reunión del Sprint Review y Retrospective.

Figura 96: Compleción de pruebas



Dentro de la recopilación de los datos de las pruebas realizadas, como ya se mostró con anterioridad en las fases de diseño e implementación se describieron algunos casos los cuales generaron un resultado y en esta fase se recopilan según el estado obtenido.

Figura 97:

Estado del caso de prueba TC_001

Test Case ID	TC_001	Test Case Description	Probando el filtrado de las notas de los becarios aprobados en una convocatoria		
Created By	Jean Colonia	Reviewed By	Jean Colonia	Version	1.0
Tester's Name		Jean	Date Tested		10 de Agosto del 2022
S #		Prerequisites:		S #	
1	Haber abierto el navegador		1	Usuario = 46536443	
2	Ingresar al sistema con las credenciales		2	Contraseña = *****	
3			3		
4			4		
Test Scenario: Verificar las notas de los becados según una convocatoria					
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Blocked	
1	Ingresar al apartado de Notas	Visualizar la sección de gestión de certificados.	Como lo esperado	Pasó	
2	Seleccionar la opción de Notas	Visualizar la sección de Notas	Como lo esperado	Pasó	
3	Seleccionar el año en los criterios de búsqueda	El año seleccionado se muestra como valor en el combo de Año.	Como lo esperado	Pasó	
4	Seleccionar la convocatoria en los criterios de búsqueda	La convocatoria seleccionada se muestra como valor en el combo de Año.	Como lo esperado	Pasó	
5	Seleccionar el Grupo Sección en los criterios de búsqueda	El grupo sección seleccionado se muestra como valor en el combo de Grupo Sección.	Como lo esperado	Pasó	
6	Seleccionar el Estado Aprobado en el combo de Estado	El estado seleccionado se muestra como valor en el combo de Estado.	Como lo esperado	Pasó	
7	Hacer clic en el botón filtrar	Se muestra el listado de las notas de los becados según el estado de becados aprobados	No se muestra ninguna fila de aprobados	Falló	

Figura 98:

Estado del caso de prueba TC_002

Test Case ID	TC_002	Test Case Description	Probando el filtrado de las notas de los becarios aprobados en una convocatoria		
Created By	Jean Colonia	Reviewed By	Jean Colonia	Version	1.0
Tester's Name		Jean	Date Tested		10 de Agosto del 2022
Tester's Name		Jean	Date Tested		10 de Agosto del 2022
S #	Prerequisites:		S #	Test Data	
1	Haber abierto el navegador		1	Usuario = 46536443	
2	Ingresar al sistema con las credenciales		2	Contraseña = *****	
3			3		
4			4		
Test Scenario: Verificar las notas de los becados según una convocatoria					
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Blocked	
1	Ingresar al apartado de	Visualizar la sección de gestión de certificados.	Como lo esperado	Pasó	
2	Seleccionar la opción de Notas	Visualizar la sección de Notas	Como lo esperado	Pasó	
3	Seleccionar el año en los criterios de búsqueda	El año seleccionado se muestra como valor en el combo de Año.	Como lo esperado	Pasó	
4	Seleccionar la convocatoria en los criterios de búsqueda	La convocatoria seleccionado se muestra como valor en el combo de Año.	Como lo esperado	Pasó	
5	Seleccionar el Grupo Sección en los criterios de búsqueda	El grupo sección seleccionado se muestra como valor en el combo de Grupo Sección.	Como lo esperado	Pasó	
6	Seleccionar el Estado Aprobado en el combo de Estado	El estado seleccionado se muestra como valor en el combo de Estado.	Como lo esperado	Pasó	
7	Hacer clic en el botón filtrar	Se muestra el listado de las notas de los becados según el estado de becados aprobados	Como lo esperado	Pasó	

Figura 99:

Estado del caso de prueba TC_003

Test Case ID	TC_003	Test Case Description	Probando el filtrado de las notas de los becarios Desaprobados en una convocatoria		
Created By	Jean Colonia	Reviewed By	Jean Colonia	Version	1.0
Tester's Name		Jean	Date Tested		10 de Agosto del 2022
Tester's Name		Jean	Date Tested		10 de Agosto del 2022
S #	Prerequisites:		S #	Test Data	
1	Haber abierto el navegador		1	Usuario = 46536443	
2	Ingresar al sistema con las credenciales		2	Contraseña = *****	
3			3		
4			4		
Test Scenario: Verificar las notas de los becados según una convocatoria					
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Blocked	
1	Ingresar al apartado de	Visualizar la sección de gestión de certificados.	Como lo esperado	Pasó	
2	Seleccionar la opción de Notas	Visualizar la sección de Notas	Como lo esperado	Pasó	
3	Seleccionar el año en los criterios de búsqueda	El año seleccionado se muestra como valor en el combo de Año.	Como lo esperado	Pasó	
4	Seleccionar la convocatoria en los criterios de búsqueda	La convocatoria seleccionado se muestra como valor en el combo de Año.	Como lo esperado	Pasó	
5	Seleccionar el Grupo Sección en los criterios de búsqueda	El grupo sección seleccionado se muestra como valor en el combo de Grupo Sección.	Como lo esperado	Pasó	
6	Seleccionar el Estado Desaprobado en el combo de Estado	El estado seleccionado se muestra como valor en el combo de Estado.	Como lo esperado	Pasó	
7	Hacer clic en el botón filtrar	Se muestra el listado de las notas de los becados según el estado de becados aprobados	Como lo esperado	Pasó	

Figura 100:

Estado del caso de prueba TC_004

Test Case ID	TC_004	Test Case Description	Probando el filtrado de las notas de los becarios Deserto en una convocatoria		
Created By	Jean Colonia	Reviewed By	Jean Colonia	Version	1.0
Tester's Name		Jean	Date Tested		10 de Agosto del 2022
Tester's Name		Jean	Date Tested		10 de Agosto del 2022
S #	Prerequisites:		S #	Test Data	
1	Haber abierto el navegador		1	Usuario = 46536443	
2	Ingresar al sistema con las credenciales		2	Contraseña = *****	
3			3		
4			4		
Test Scenario: Verificar las notas de los becados según una convocatoria					
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Blocked	
1	Ingresar al apartado de	Visualizar la sección de gestión de certificados.	Como lo esperado	Pasó	
2	Seleccionar la opción de Notas	Visualizar la sección de Notas	Como lo esperado	Pasó	
3	Seleccionar el año en los criterios de búsqueda	El año seleccionado se muestra como valor en el combo de Año.	Como lo esperado	Pasó	
4	Seleccionar la convocatoria en los criterios de búsqueda	La convocatoria seleccionado se muestra como valor en el combo de Año.	Como lo esperado	Pasó	
5	Seleccionar el Grupo Sección en los criterios de búsqueda	El grupo sección seleccionado se muestra como valor en el combo de Grupo Sección.	Como lo esperado	Pasó	
6	Seleccionar el Estado Deserto en el combo de Estado	El estado seleccionado se muestra como valor en el combo de Estado.	Como lo esperado	Pasó	
7	Hacer clic en el botón filtrar	Se muestra el listado de las notas de los becados según el estado de becados aprobados	Como lo esperado	Pasó	

Por otra parte la aplicación de gestión de los casos de prueba y el resultado de su ejecución también fue manejado mediante TestRail y los Bugs mediante Clickup como tareas a realizar con las evidencias del caso.

Figura 101:

Reporte del estado de los casos de prueba en TestRail

ID	Title	Assigned To	Status
T1	Probando el filtrado de las notas de los becarios con estado aprobado en una convocatoria	Jean C.	Passed
T3	Probando el filtrado de las notas de los becarios con estado desaprobado en una convocatoria	Jean C.	Passed
T5	Ingresar al sistema con credenciales válidas	Jean C.	Passed
T7	Ingresar al sistema con credenciales inválidas	Jean C.	Failed
T9	Ingresar al sistema con usuario inválido	Jean C.	Failed
T11	Ingresar al sistema con contraseña inválida	Jean C.	Failed
T13	Probando el filtrado de las notas de los becarios aprobados en una convocatoria	Jean C.	Passed
T15	Probando el filtrado de las notas de los becarios desaprobados en una convocatoria	Jean C.	Passed
T17	Probando el filtrado de las notas de los becarios Deserto en una convocatoria	Jean C.	Passed
T19	Probando el filtrado de las notas de los becarios Retiro en una convocatoria	Jean C.	Passed
T21	Visualizar número de documento - valores límites a fecha de nacimiento	Jean C.	Failed
T23	Probar funcionalidad del filtro de búsqueda en los combos de los certificados	Jean C.	Failed
T25	Visualizar video DOCUMENTOS Y VIDEOS	Jean C.	Failed
T27	Probando la reproducción del video en PÁGINA MESA DE AYUDA	Jean C.	Failed
T29	PARÁMETROS - visualización de listado de parámetros	Jean C.	Blocked
T31	Visualizar mensaje de confirmación de Acción realizada en TUTOR	Jean C.	Retest

Figura 102:

Resumen del estado de los casos de prueba

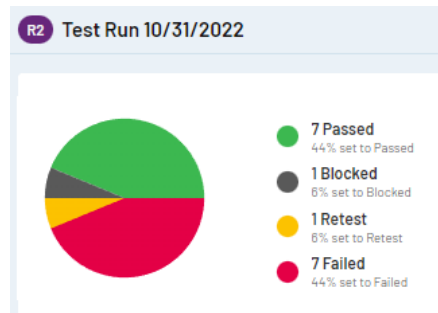


Figura 103:

Actividad de los casos de prueba



Por otro lado se tuvo también en cuenta los reportes manuales realizados con Excel los cuales dan una opción a los gestores de casos de pruebas y reportes que si en un caso no se puedan adquirir, estos resultados sean por gestores o de forma manual tuvieron como objetivo la obtención de información de lo realizado durante todo el sprint y fue corroborado por la fase de Monitoreo y Control de pruebas.

Tabla 30:

Resumen de pruebas en un Sprint como ejemplo

RESUMEN	
Executed / Ejecutados	16
Pass / Aprobados	7
Fail / Fallado	7
Bloqueados	1
Retest	1
No ejecutados	0
Total de casos de pruebas	16

Otro documento importante es la cobertura de pruebas que previamente se había mencionado dentro de la fase de Monitoreo y Control de pruebas que la que nos indica cuanto estamos avanzando con respecto a las pruebas y qué no estaría faltando, este aspecto es sumamente importante porque servirá como feedback para los siguientes sprints o proyectos a realizar en el futuro.

Tabla 31:

Resultado final de la cobertura de pruebas para el sprint

Pruebas del sistema	Cobertura de pruebas	Número de semanas	Avance real
Módulo de cursos	30%	1	30%
Módulo de Convocatorias	30%	1	20%
Módulo de gestión de certificados	40%	2	20%
Total	100%	4	70%

Por tanto desde el ultimo sprint que se tenía una cobertura final del 50% y con el sprint que se realizó se tuvo un incremento del 20% del 100% que es el total del proyecto de software, deduciéndose que se ha llegado a un 70% del total del proyecto con las pruebas realizadas y lógicamente con el desarrollo realizado en paralelo y que fue objeto de pruebas.

4.2 Presentación de resultados y prueba de hipótesis

4.2.1 Presentación de resultados

Los resultados obtenidos en las diversas fases de la investigación fueron agrupados según los objetivos y la medición de la calidad mediante las categorías y subcategorías de la ISO 25010, que también fueron parte de las dimensiones e indicadores.

Tabla 32:

Resumen de resultados obtenidos

RESULTADOS				
Resultados Cualitativos			Resultados Cuantitativos	
Según objetivos de la investigación			Según la medición de la calidad mediante ISO 25010	
Objetivo	Item	Sub Items	Categoría	Subcategoría
Identificar la realidad de la calidad de software antes de la aplicación de Quality Assurance enfocado en Agile Testing	Análisis situacional	Identificar la situación real de la calidad de software	Adecuación Funcional	Compleitud funcional
		Definición de requerimientos		Corrección funcional
		Pruebas de Software		Pertinencia funcional
Identificar la realidad de la calidad de software antes de la aplicación de Quality Assurance enfocado en Agile Testing	Modelo ISTQB	Inclusión del marco de trabajo ágil Scrum	Eficiencia de desempeño	Comportamiento temporal
		Planificación de pruebas		Utilización de recursos
		Monitoreo y Control de pruebas	Compatibilidad	Coexistencia
		Análisis de Pruebas		Interoperabilidad
		Diseño de pruebas	Usabilidad	Aprendizaje
		Implementación de pruebas		Accesibilidad
		Ejecución de pruebas		Protección frente a errores de usuario
		Compleción de pruebas	Seguridad	Estética
				Confidencialidad
				Integridad
		No repudio		

4.2.1.1 Resultados Cualitativos

Objetivo 1: Se logró identificar la realidad de la calidad de software antes de la aplicación de Quality Assurance enfocado en Agile Testing en la empresa Wasi Innovation SAC, Lima – 2022:

- Se logró identificar la situación real de la calidad de software mediante la entrevista realizada al gerente de la empresa que fue contexto de la investigación, donde se pudo evidenciar las dificultades que se tenía con los clientes y su enfoque de trabajo, los recursos, las actividades, conocimientos técnicos y otros que eran base para asegurar la calidad de software en sus proyectos. Por otra parte el testimonio del equipo de calidad complementó la información para poder elaborar una serie de requerimientos.
- Se logró definir de requerimientos necesarios para poder realizar el modelo de calidad de software basado en ISTQB y las actividades complementarias para lograr el objetivo general, se tuvieron un total de 17 requerimientos los cuales fueron desarrollados como parte del modelo para mejorar la calidad de software.
- Se logró realizar las pruebas de software en los diversos tipos como las pruebas de estructura, funcionales, rendimiento y otros donde se incluyeron las pruebas automatizadas y manuales como pruebas de servicios, revisión de código, de rendimiento y las pruebas funcionales.

Objetivo 2: Se logró identificar la realidad de la calidad de software después de la aplicación de Quality Assurance enfocado en Agile Testing en la empresa Wasi Innovation SAC, Lima – 2022:

- Se logró Inclusión del marco de trabajo ágil Scrum según las características que tenía la empresa y se realizó la adecuación correspondiente en las ceremonias, incorporando el Spring Review y el Sprint Retrospective en una sola reunión, en los demás casos como ya se tenía una noción y estaban aplicando Scrum en la medida que se podía, lo cual benefició mucho a la investigación porque ya existía una base o conocimiento previo.

- Se logró realizar la planificación de las pruebas como primera fase dentro del modelo según ISTQB para asegurar la calidad de software en los proyectos en la empresa Wasi Innovation, donde se tuvo en cuenta los requerimientos de software dentro del Product Backlog y el Plan de pruebas donde se detalló algunos aspectos técnicos y de negocios, los alcances, herramientas y otras estrategias para lograr la calidad de software.
- Se logró realizar el Monitoreo y Control de las pruebas como segunda fase del modelo, donde se tuvo en cuenta el monitoreo con respecto a lo que se había planificado en la fase anterior y cuál era la situación real dentro de lo desarrollado y el control de pruebas en cada Sprint para las falencias y la toma de decisiones correspondiente para lograr lo planificado.
- Se logró realizar el Análisis de las pruebas como tercera fase del modelo, dentro del cual se tuvo en cuenta las actividades como el análisis de los documentos base del proyecto como TDR, requerimientos, otros documentos importantes para lograr entregar el producto desarrollado con la mejor calidad posible, también el código fuente del backend, frontend, base de datos y otros necesarios para poder realizar el diseño de los casos de prueba.
- Se logró realizar el Diseño de las pruebas como cuarta fase del modelo, teniendo en cuenta un previo análisis para la elaboración de los escenarios y casos de prueba, estos se hicieron de forma manual en archivos tipo Excel y también usando un gestor de pruebas como TestRail o Clickup que era la aplicación usada por la empresa contexto de esta investigación.
- Se logró realizar la Implementación de las pruebas como quinta fase del modelo, donde se tuvo en cuenta las actividades como la preparación de los ambientes de prueba tipo automatizado o manual y según los niveles y tipos de pruebas, instalación de dependencias, configuraciones, código fuente de las pruebas y otros requeridos para su posterior ejecución.

- Se logró realizar la Ejecución de las pruebas como sexta fase del modelo, las actividades desarrolladas en esta fase fueron la identificación del comportamiento de las ejecuciones de pruebas y en general todos los resultados obtenidos de las pruebas de servicios, estructura, rendimiento, funcionales y otros para su posterior evaluación y mejora en esta misma fase, otra fase o ceremonia Scrum para identificar mejoras.
- Se logró realizar la Compleción de las pruebas como séptima fase del modelo, las actividades desarrolladas en esta fase fueron documentar los resultados obtenidos según la fase de ejecución anterior, realizar la recopilación de dichos resultados de las pruebas mediante los reportes de pruebas y analizar las lecciones aprendidas, finalizar y archivar. Esta fase fue sumamente importante porque genera documentación de todo el trabajo realizado durante un Sprint y que es analizado en la unión de ceremonias Sprint Review y Retrospective.

4.2.1.2 Resultados Cuantitativos

Previo a los resultados según las características y subcaracterísticas de la ISO 25010 para la medición de la calidad de software, se procedió a realizar el análisis de fiabilidad mediante Alfa de Cronbach.

- **Análisis de Fiabilidad:** El análisis de confiabilidad se realizó para corroborar la confiabilidad del instrumento de medición, el cual cumplió un rol sumamente importante que fue la medición de la calidad de software en los proyectos que realiza la empresa que fue contexto de la investigación y en el cual se realizó la recolección de los datos mediante el instrumento de medición que fue un cuestionario para el pretest y postest.

Según Chaves & Rodríguez (2016), la escala de valoración según el coeficiente alfa de Cronbach se clasifica en los siguientes intervalos: Inaceptable, Pobre, Débil, Aceptable, Bueno y Excelente.

Tabla 33:

Escala de valoración de Alfa de Cronbach

Intervalo del coeficiente Alfa de Cronbach	Valoración de la fiabilidad
[0 – 0.49]	Inaceptable
[0.5 – 0.59]	Pobre
[0.6 – 0.69]	Débil
[0.7 – 0.79]	Aceptable
[0.8 – 0.89]	Bueno
[0.9 – 1]	Excelente

El resultado del análisis de confiabilidad para el instrumento de medición según su escala valorativa y sus 14 preguntas provenientes de los indicadores de la operacionalización de variables para la variable Calidad de Software fue la siguiente.

Tabla 34:

Estadísticas de fiabilidad – pre test

Alfa de Cronbach	N de elementos
,944	14

Según la escala valorativa y el resultado obtenido mediante las estadísticas de fiabilidad del Alfa de Cronbach se interpreta que se tiene un valor de 0.944 que se deduce como excelente.

Tabla 35:

Estadísticas de fiabilidad – post test

Alfa de Cronbach	N de elementos
,822	14

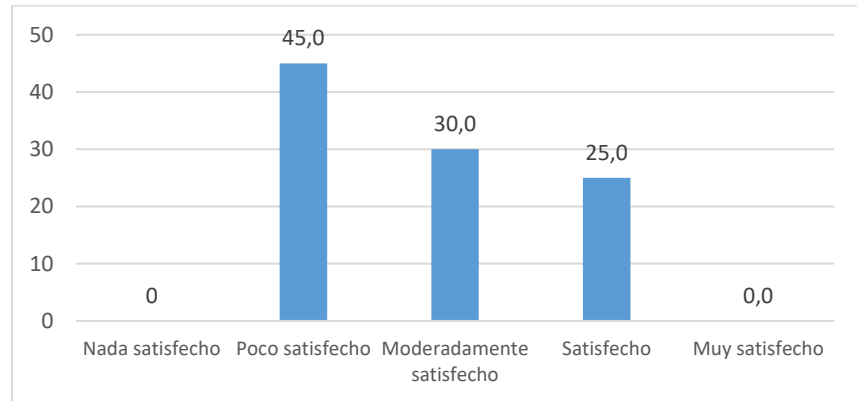
Según la escala valorativa y el resultado obtenido mediante las estadísticas de fiabilidad del Alfa de Cronbach se interpreta que se tiene un valor de 0.822 que se deduce como bueno.

➤ RESULTADOS CUANTITATIVOS – PRE TEST

P1: ¿Está satisfecho con las funcionalidades que brinda el software según los requerimientos establecidos?

Figura 104:

Resultados de Completitud funcional- pre test

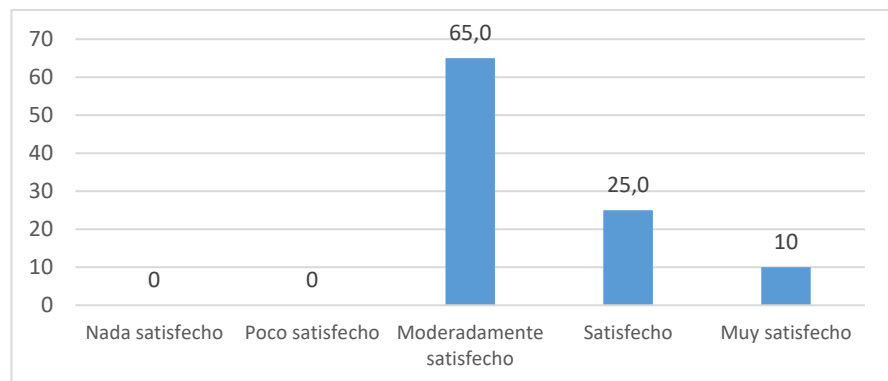


Interpretación: en la figura se observa que un 45% se encuentra poco satisfecho, un 30% se encuentra moderadamente satisfecho y un 25% se encuentra satisfecho con las funcionalidades que brinda el software según los requerimientos establecidos previamente.

P2: ¿Está satisfecho con la precisión de los resultados que brinda el software?

Figura 105:

Resultados de Corrección funcional – pre test

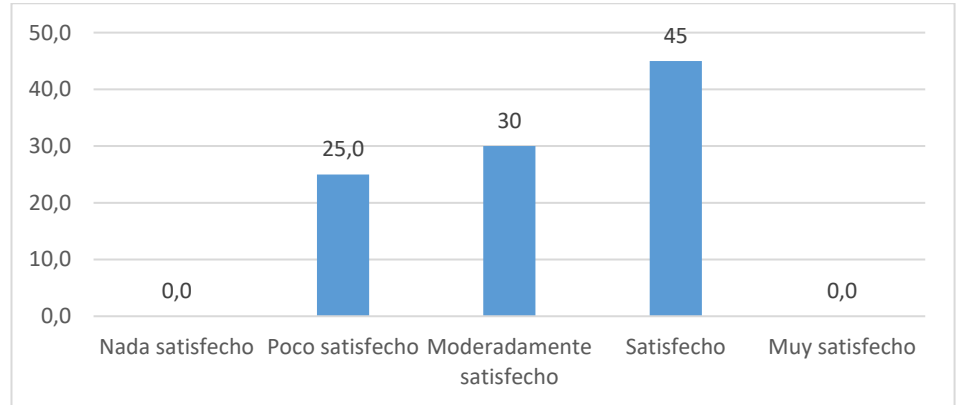


Interpretación: en la figura se observa que un 65% se encuentra moderadamente satisfecho, un 25% se encuentra satisfecho y un 10% se encuentra muy satisfecho con la precisión de los resultados que brinda el software.

P3: ¿En qué grado está satisfecho en relación a las actividades de su día a día que puede realizar con el producto?

Figura 106:

Resultados de Pertinencia funcional – pre test

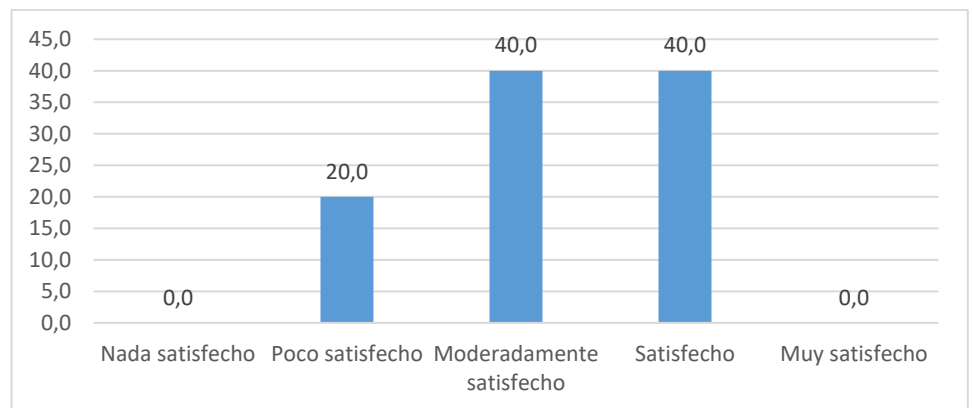


Interpretación: en la figura se observa que un 25% se encuentra poco satisfecho, un 30% se encuentra moderadamente satisfecho y un 45% se encuentra satisfecho con la ayuda que brinda el software para realizar las actividades diarias.

P4: ¿Está satisfecho con el tiempo que demora realizar los procesos como las matriculas, reportes, emisión de certificados y otros?

Figura 107:

Resultados de Comportamiento temporal – pre test

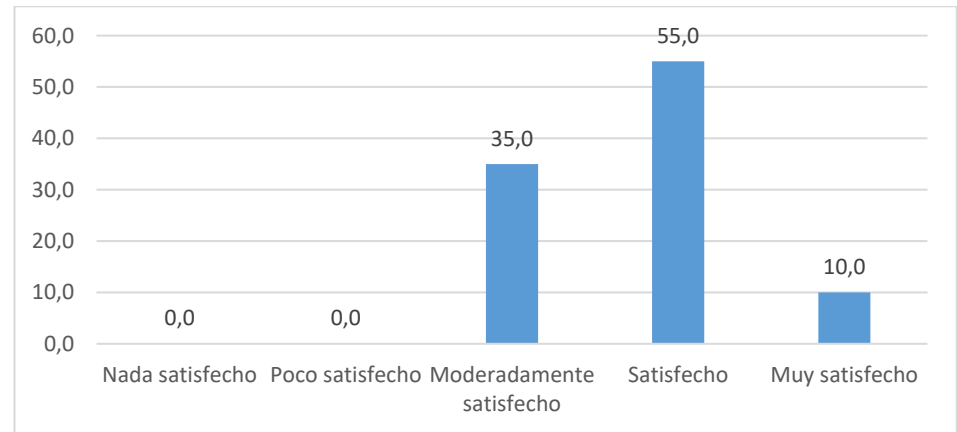


Interpretación: en la figura se observa que un 20% se encuentra poco satisfecho, un 40% se encuentra moderadamente satisfecho y un 40% se encuentra satisfecho con el tiempo que se demora en realizar los procesos diarios mediante el uso del software.

P5: ¿Está satisfecho con el software y los requisitos/recursos que se necesitan para que pueda usarlo?

Figura 108:

Resultados de Utilización de recursos – pre test

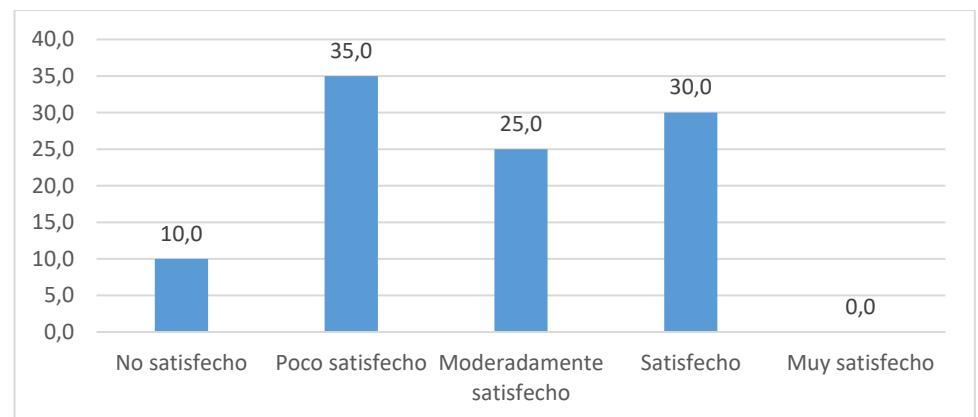


Interpretación: en la figura se observa que un 35% se encuentra moderadamente satisfecho, un 55% se encuentra satisfecho y un 10% se encuentra muy satisfecho con los requisitos y recursos que se necesitan para que puedan usar el software.

P6: ¿Está satisfecho con el funcionamiento del software con otros como el moodle?

Figura 109:

Resultados de Coexistencia - pre test

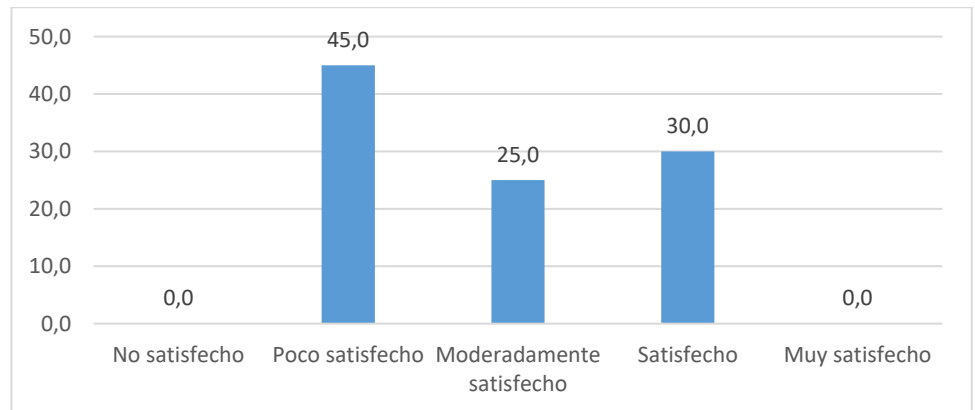


Interpretación: en la figura se observa un 10% no satisfecho, un 35% se encuentra poco satisfecho, un 25% se encuentra moderadamente satisfecho y un 30% se encuentra satisfecho con el funcionamiento del software con otros sistemas como el moodle.

P7: ¿Está satisfecho con el intercambio de información entre software con el moodle y otros?

Figura 110:

Resultados de Interoperabilidad - pre test

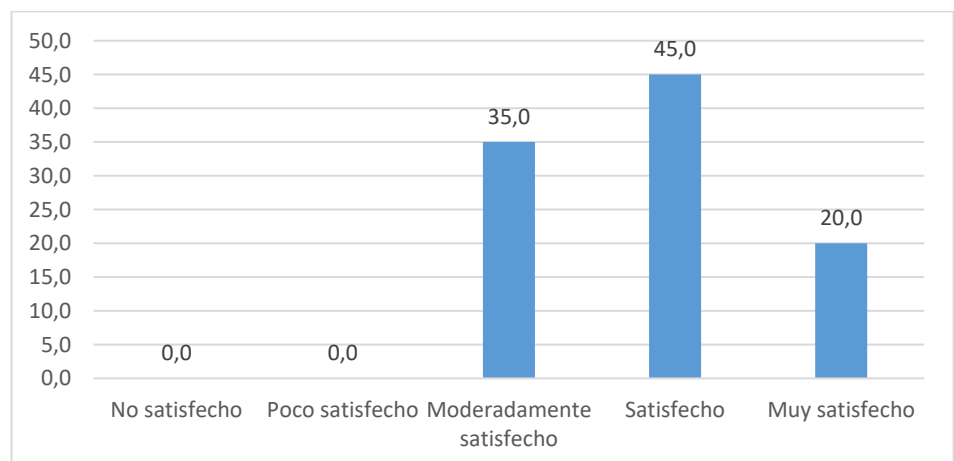


Interpretación: en la figura se observa que un 45% se encuentra poco satisfecho, un 25% se encuentra moderadamente satisfecho y un 30% se encuentra satisfecho con el intercambio de información entre el software y otros sistemas como el moodle.

P8: ¿Está satisfecho con la facilidad de aprendizaje del software?

Figura 111:

Resultados de Aprendizaje - pre test

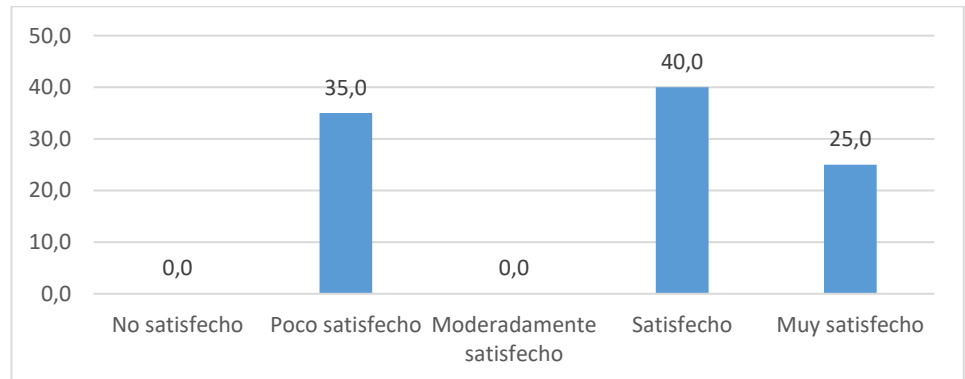


Interpretación: en la figura se observa que un 35% se encuentra moderadamente satisfecho, un 45% se encuentra satisfecho y un 20% se encuentra muy satisfecho con la facilidad de aprendizaje del software.

P9: ¿Está satisfecho con la accesibilidad al software (tiene la capacidad de ser usado por personas con distintas características)?

Figura 112:

Resultados de Accesibilidad - pre test

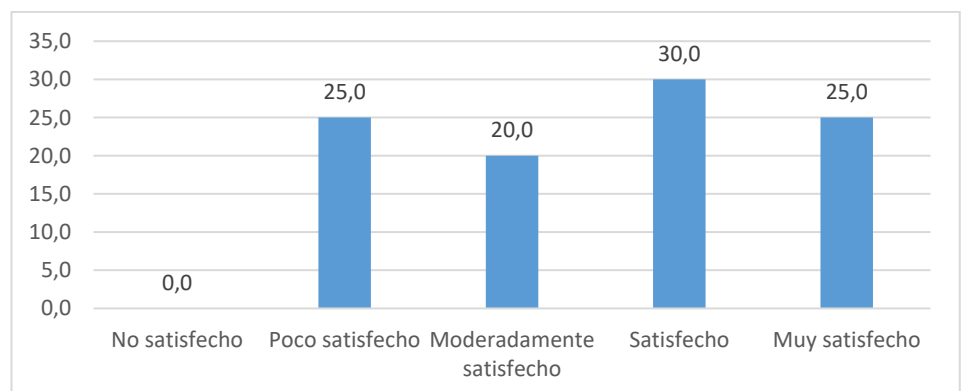


Interpretación: en la figura se observa que un 35% se encuentra poco satisfecho, un 40% se encuentra satisfecho y un 25% se encuentra muy satisfecho con la capacidad de ser usado por personas con distintas características.

P10: ¿Está satisfecho con la prevención de errores del usuario dentro del software (mensajes, alertas u otro que pueda restringir errores)?

Figura 113:

Resultados de Protección frente a errores de usuario - pre test

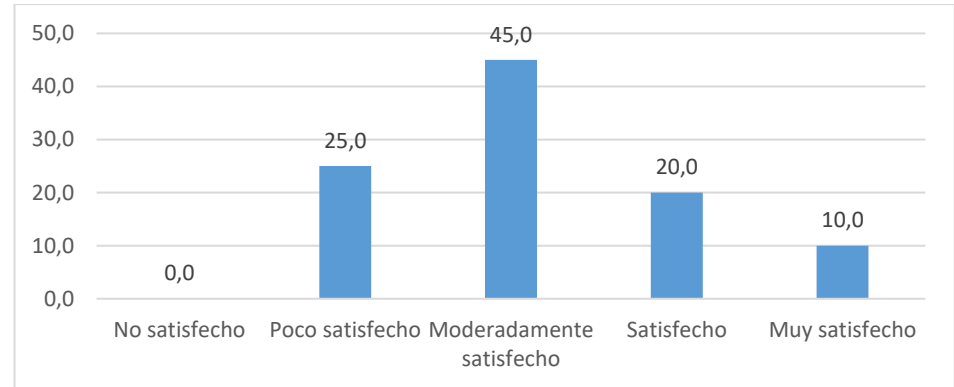


Interpretación: en la figura se observa que un 25% se encuentra poco satisfecho, un 20% se encuentra moderadamente satisfecho, un 30% se encuentra satisfecho y un 25% se encuentra muy satisfecho con la prevención de errores del usuario dentro del software.

P11: ¿Está satisfecho con el diseño, formas y colores del software?

Figura 114:

Resultados de Estética - pre test

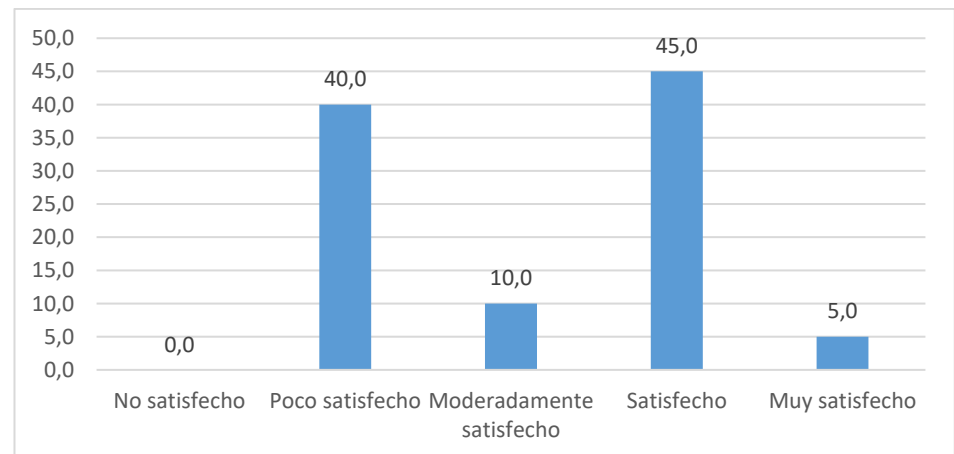


Interpretación: en la figura se observa que un 25% se encuentra poco satisfecho, un 45% se encuentra moderadamente satisfecho, un 20% se encuentra satisfecho y un 10% se encuentra muy satisfecho con el diseño, formas y colores del software.

P12: ¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser accedidos por otros?

Figura 115:

Resultados de Confidencialidad - pre test

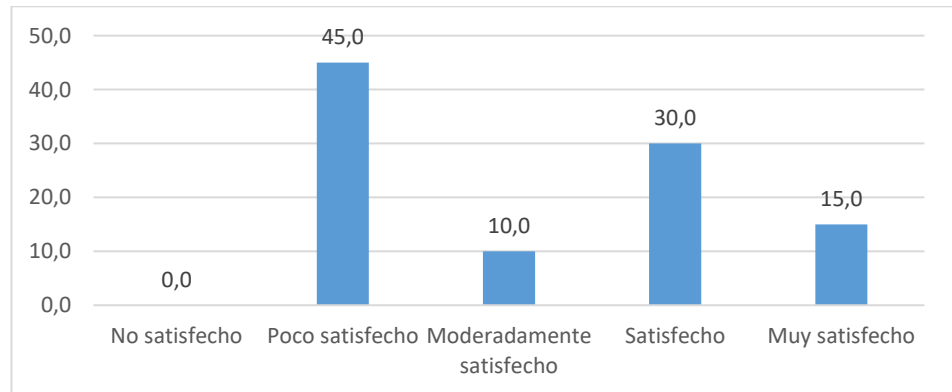


Interpretación: en la figura se muestra que un 40% se encuentra poco satisfecho, un 10% se encuentra moderadamente satisfecho, un 45% se encuentra satisfecho y un 5% muy satisfecho con la seguridad de los datos para que no puedan ser accedidos por otros.

P13: ¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser modificados por otro?

Figura 116:

Resultados de Integridad - pre test

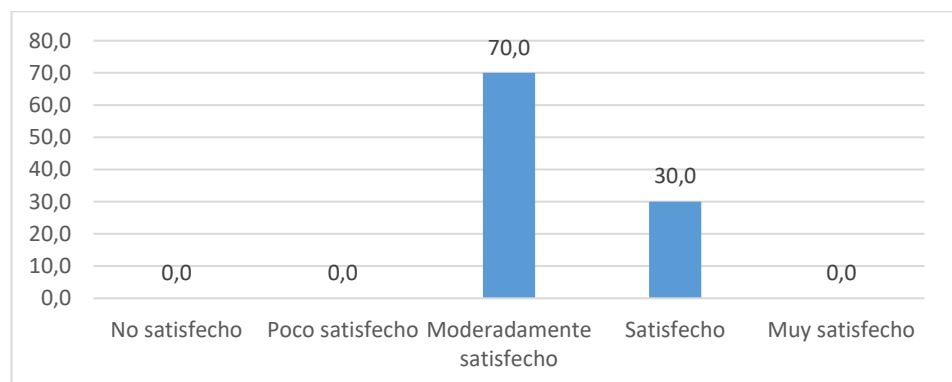


Interpretación: en la figura se observa que un 45% se encuentra poco satisfecho, un 10% se encuentra moderadamente satisfecho, un 30% se encuentra satisfecho y un 15% se encuentra muy satisfecho con la seguridad de los datos que brinda el software para que no puedan ser modificados por otros.

P14: ¿Está satisfecho con la capacidad del software para registrar acciones indebidas u otras anomalías (auditorías)?

Figura 117:

Resultados de No repudio – pre test

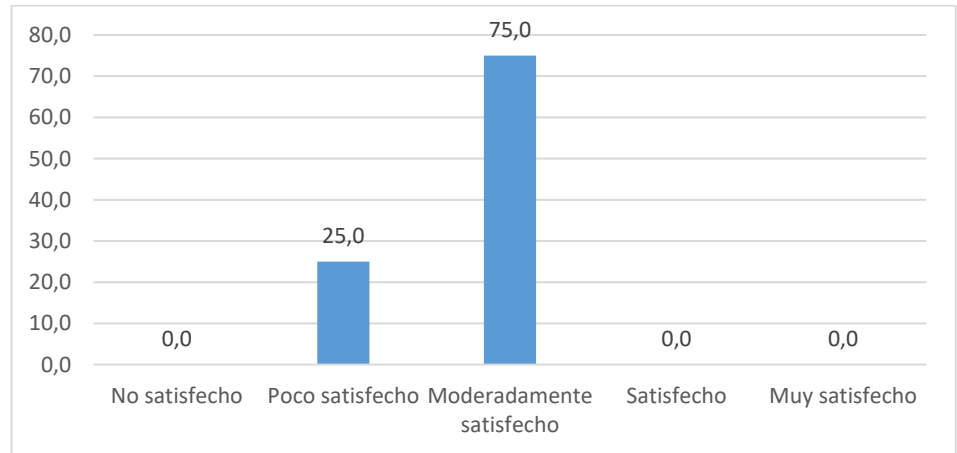


Interpretación: en la figura se observa que un 70% se encuentra moderadamente satisfecho y un 30% se encuentra satisfecho con la capacidad del software para poder registrar acciones indebidas y otras anomalías.

Dimensión: Adecuación Funcional

Figura 118:

Resultados de la dimensión Adecuación Funcional - pre test

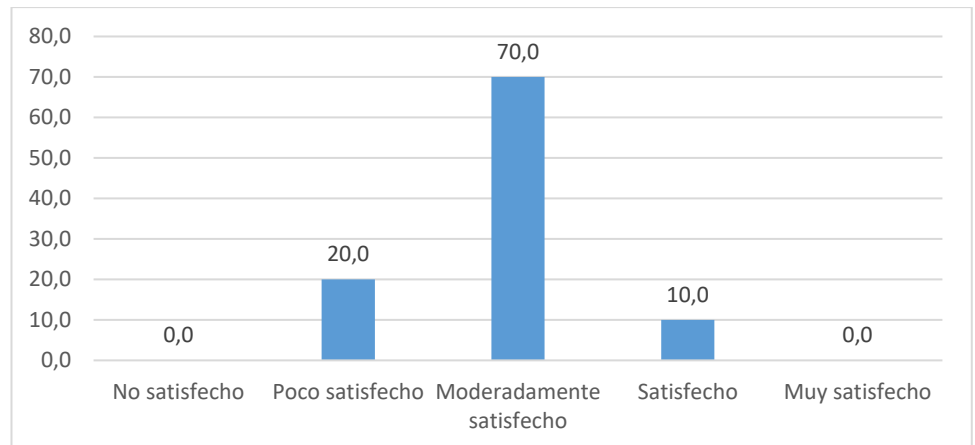


Interpretación: en la figura se observa que un 25% se encuentra poco satisfecho y un 75% se encuentra moderadamente satisfecho con respecto a la adecuación funcional del software.

Dimensión: Eficiencia de desempeño

Figura 119:

Resultados de la dimensión Eficiencia de desempeño - pre test

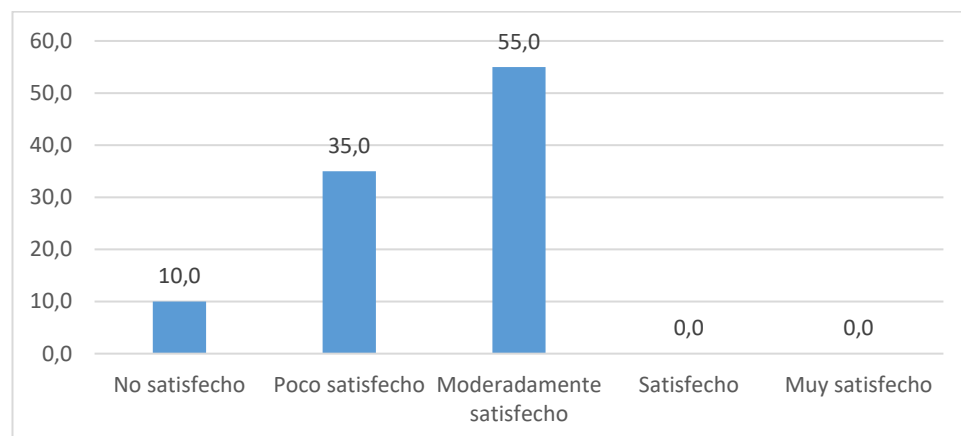


Interpretación: en la figura se observa que un 20% se encuentra poco satisfecho, un 70% se encuentra moderadamente satisfecho y un 10% se encuentra satisfecho con la eficiencia de desempeño del software.

Dimensión: Compatibilidad

Figura 120:

Resultados de la dimensión Compatibilidad - pre test

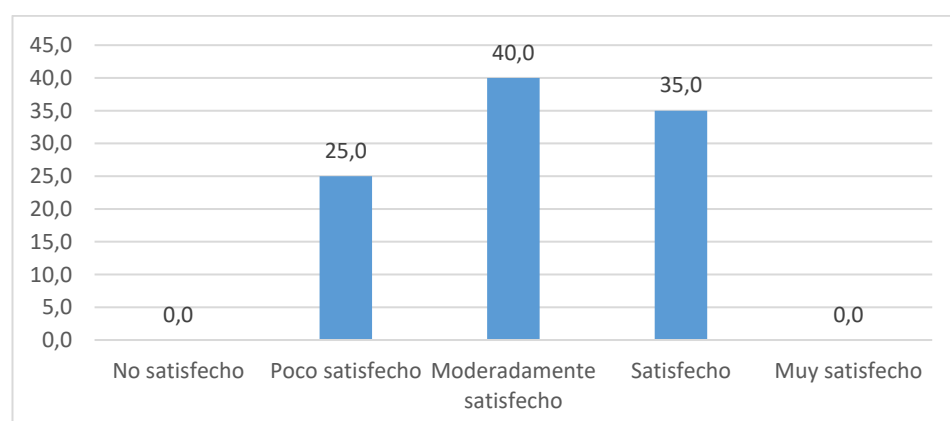


Interpretación: en la figura se observa que un 10% no se encuentra satisfecho, un 35% se encuentra poco satisfecho y un 55% se encuentra moderadamente satisfecho con la compatibilidad del software.

Dimensión: Usabilidad

Figura 121:

Resultados de la dimensión Usabilidad - pre test

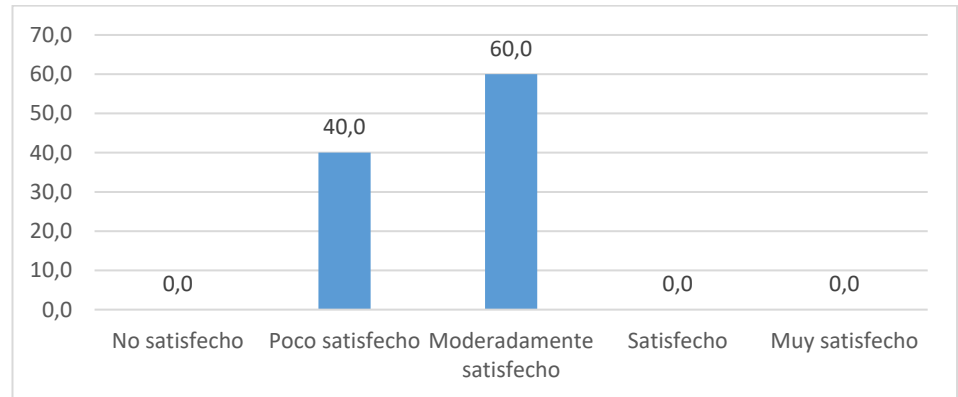


Interpretación: en la figura se observa que un 25% se encuentra poco satisfecho, un 40% se encuentra moderadamente satisfecho y un 35% se encuentra satisfecho con la usabilidad del software.

Dimensión: Seguridad

Figura 122:

Resultados de la dimensión Seguridad - pre test

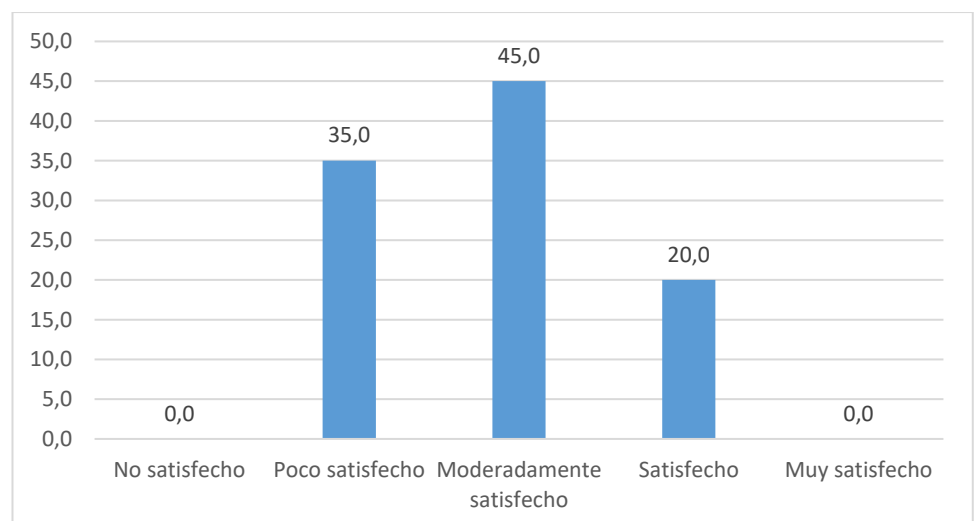


Interpretación: en la figura se observa que un 40% se encuentra poco satisfecho y un 60% se encuentra moderadamente satisfecho con la seguridad que brinda el software.

Variable: Calidad de Software

Figura 123:

Resultados para la variable Calidad de Software - pre test



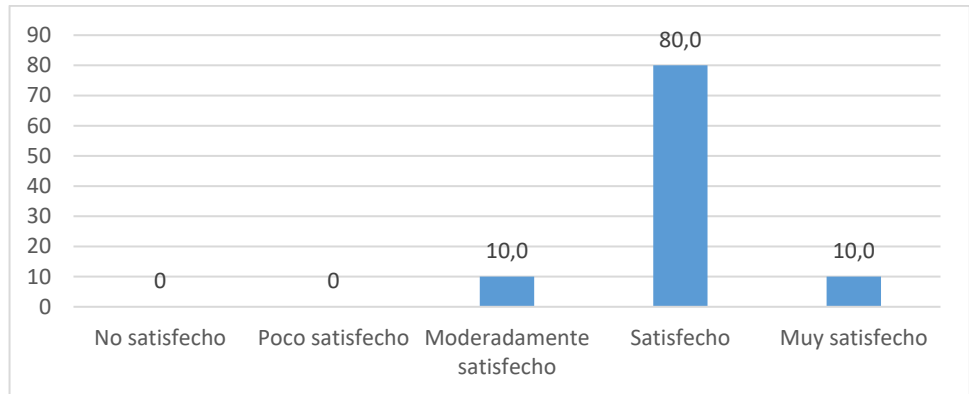
Interpretación: en la figura se observa que un 35% se encuentra poco satisfecho, un 45% se encuentra moderadamente satisfecho y un 20% se encuentra satisfecho con la calidad del software.

➤ RESULTADOS CUANTITATIVOS – POST TEST

P1: ¿Está satisfecho con las funcionalidades que brinda el software según los requerimientos establecidos?

Figura 124:

Resultados de Completitud funcional- post test

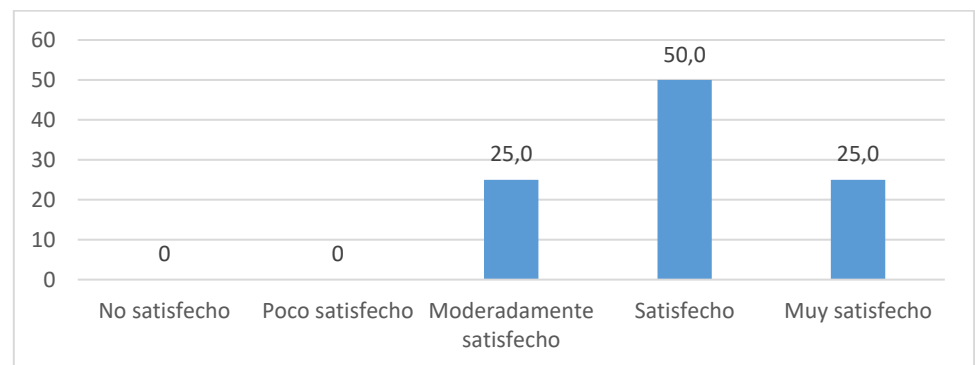


Interpretación: en la figura se observa que un 10% se encuentra moderadamente satisfecho, un 80% se encuentra satisfecho y un 10% se encuentra muy satisfecho con las funcionalidades que brinda el software según los requerimientos establecidos previamente.

P2: ¿Está satisfecho con la precisión de los resultados que brinda el software?

Figura 124:

Resultados de Corrección funcional – post test

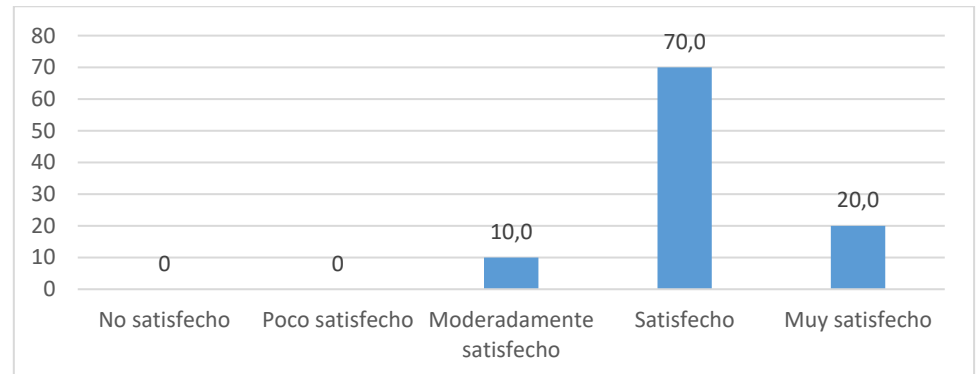


Interpretación: en la figura se observa que un 25% se encuentra moderadamente satisfecho, un 50% se encuentra satisfecho y un 25% se encuentra muy satisfecho con la precisión de los resultados que brinda el software.

P3: ¿En qué grado está satisfecho en relación a las actividades de su día a día que puede realizar con el producto?

Figura 125:

Resultados de Pertinencia funcional – post test

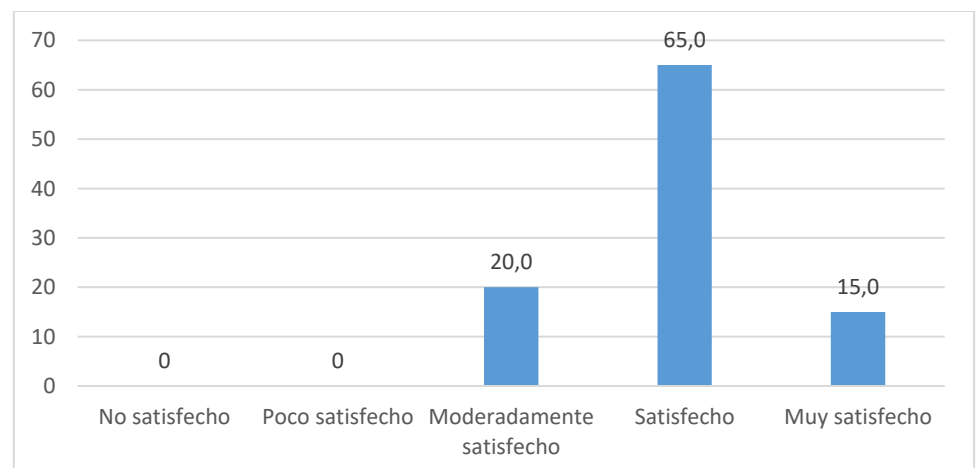


Interpretación: en la figura se observa que un 10% se encuentra moderadamente satisfecho, un 70% se encuentra satisfecho y un 20% se encuentra muy satisfecho con la ayuda que brinda el software para realizar las actividades diarias.

P4: ¿Está satisfecho con el tiempo que demora realizar los procesos como las matriculas, reportes, emisión de certificados y otros?

Figura 126:

Resultados de Comportamiento temporal – post test

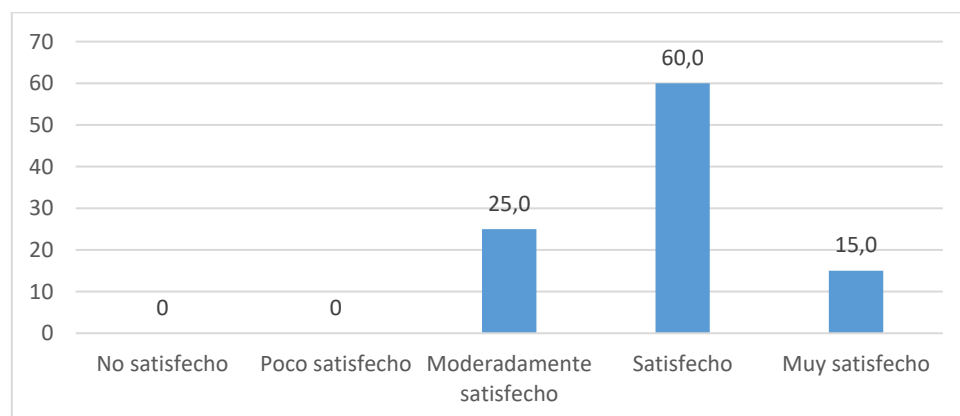


Interpretación: en la figura se observa que un 20% se encuentra moderadamente satisfecho, un 65% se encuentra satisfecho y un 15% se encuentra muy satisfecho con el tiempo que se demora en realizar los procesos diarios mediante el uso del software.

P5: ¿Está satisfecho con el software y los requisitos/recursos que se necesitan para que pueda usarlo?

Figura 127:

Resultados de Utilización de recursos – post test

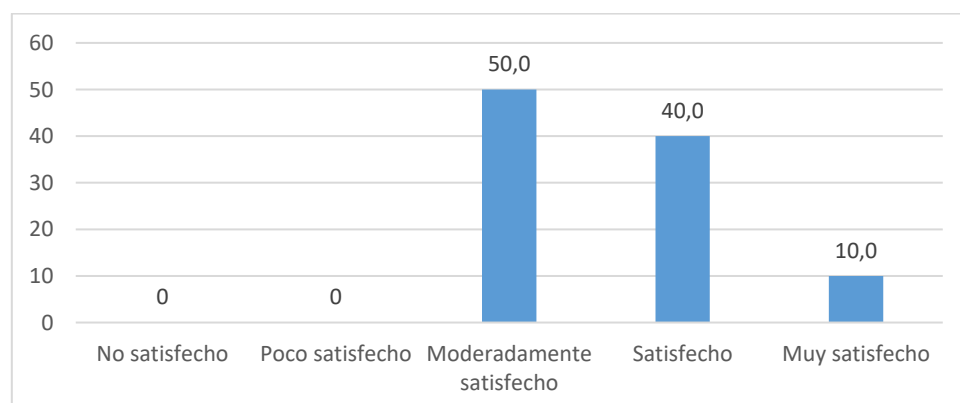


Interpretación: en la figura se observa que un 25% se encuentra moderadamente satisfecho, un 60% se encuentra satisfecho y un 15% se encuentra muy satisfecho con los requisitos y recursos que se necesitan para que puedan usar el software.

P6: ¿Está satisfecho con el funcionamiento del software con otros como el moodle?

Figura 128:

Resultados de Coexistencia - post test

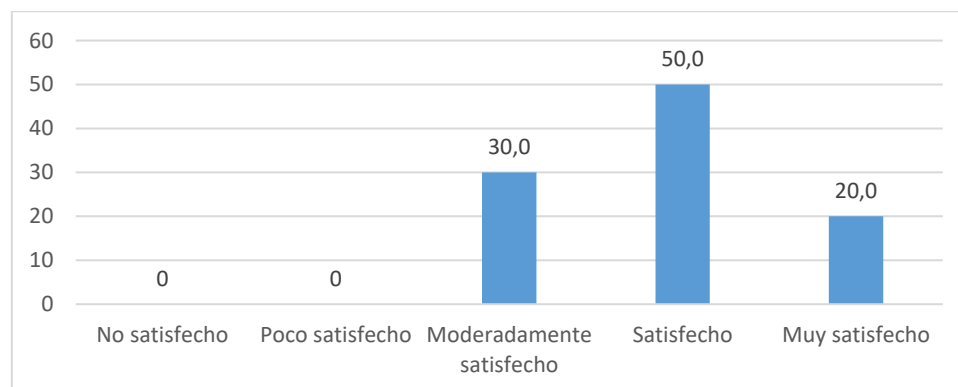


Interpretación: en la figura se observa que un 50% se encuentra moderadamente satisfecho, un 40% se encuentra satisfecho y un 10% se encuentra muy satisfecho con el funcionamiento del software con otros sistemas como el moodle.

P7: ¿Está satisfecho con el intercambio de información entre software con el moodle y otros?

Figura 129:

Resultados de Interoperabilidad - post test

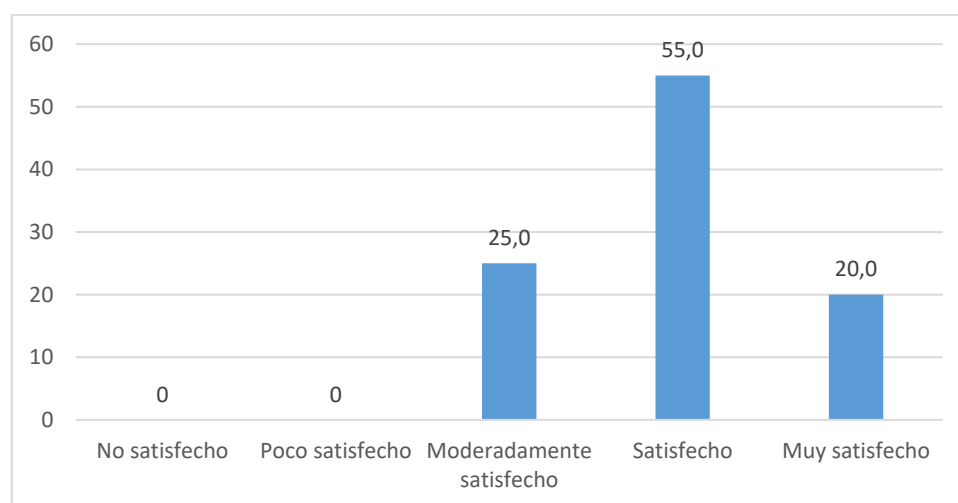


Interpretación: en la figura se observa que 30% se encuentra moderadamente satisfecho, un 50% se encuentra satisfecho y un 20% se encuentra muy satisfecho con el intercambio de información entre el software y otros sistemas como el moodle.

P8: ¿Está satisfecho con la facilidad de aprendizaje del software?

Figura 130:

Resultados de Aprendizaje - post test

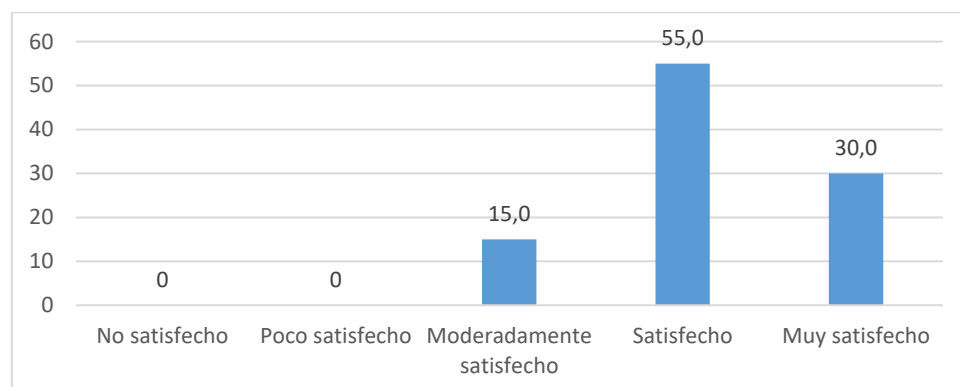


Interpretación: en la figura se observa que un 25% se encuentra moderadamente satisfecho, un 55% se encuentra satisfecho y un 20% se encuentra muy satisfecho con la facilidad de aprendizaje del software.

P9: ¿Está satisfecho con la accesibilidad al software (tiene la capacidad de ser usado por personas con distintas características)?

Figura 131:

Resultados de Accesibilidad - post test

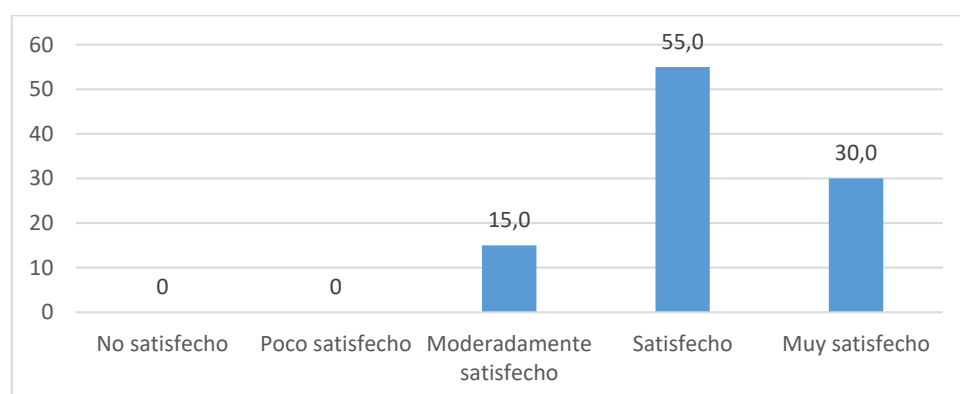


Interpretación: en la figura se observa que un 15% se encuentra moderadamente satisfecho, un 55% se encuentra satisfecho y un 30% se encuentra muy satisfecho con la capacidad de ser usado por personas con distintas características.

P10: ¿Está satisfecho con la prevención de errores del usuario dentro del software (mensajes, alertas u otro que pueda restringir errores)?

Figura 132:

Resultados de Protección frente a errores de usuario - post test

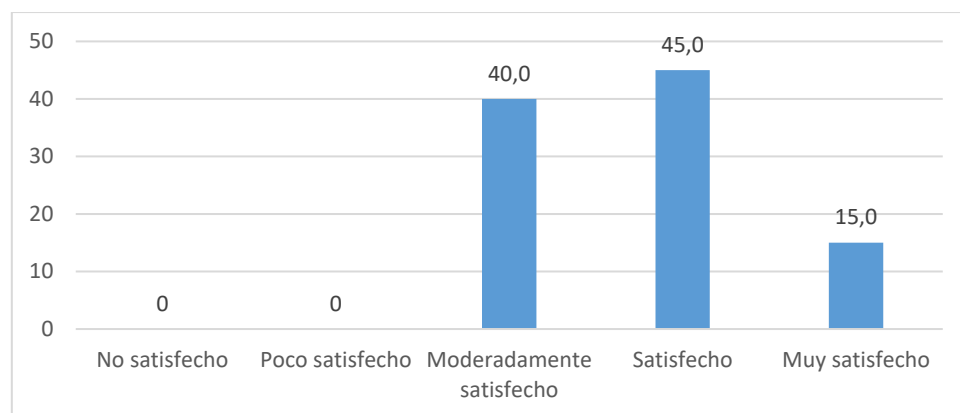


Interpretación: en la figura se observa que un 15% se encuentra moderadamente satisfecho, un 55% se encuentra satisfecho y un 30% se encuentra muy satisfecho con la prevención de errores del usuario dentro del software.

P11: ¿Está satisfecho con el diseño, formas y colores del software?

Figura 133:

Resultados de Estética - post test

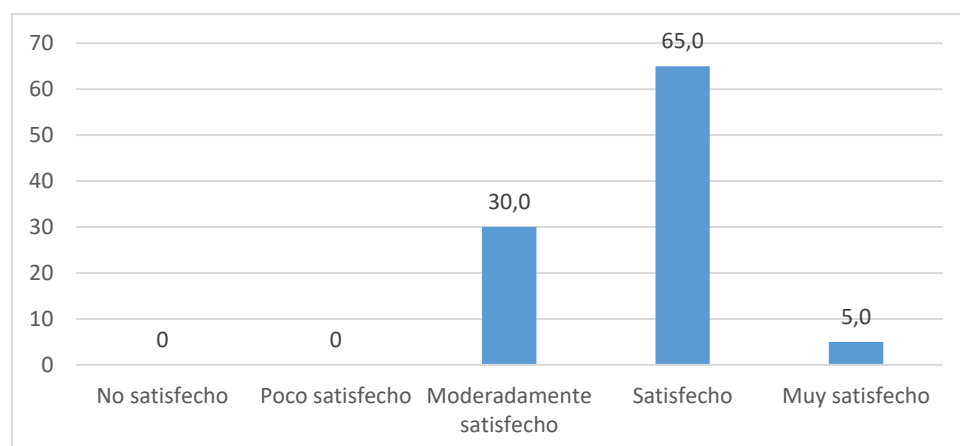


Interpretación: en la figura se observa que un 40% se encuentra moderadamente satisfecho, un 45% se encuentra satisfecho y un 15% se encuentra muy satisfecho con el diseño, formas y colores del software.

P12: ¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser accedidos por otros?

Figura 134:

Resultados de Confidencialidad - post test

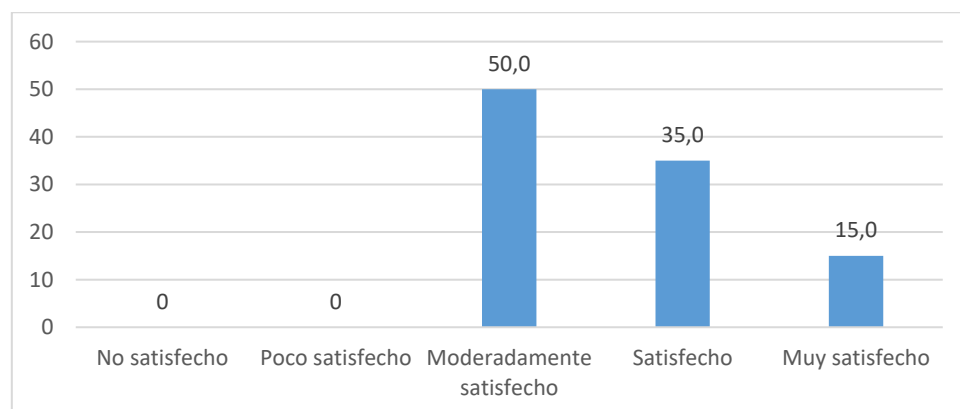


Interpretación: en la figura se muestra que un 30% se encuentra moderadamente satisfecho, un 65% se encuentra satisfecho y un 5% muy satisfecho con la seguridad de los datos para que no puedan ser accedidos por otros.

P13: ¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser modificados por otro?

Figura 135:

Resultados de Integridad - post test

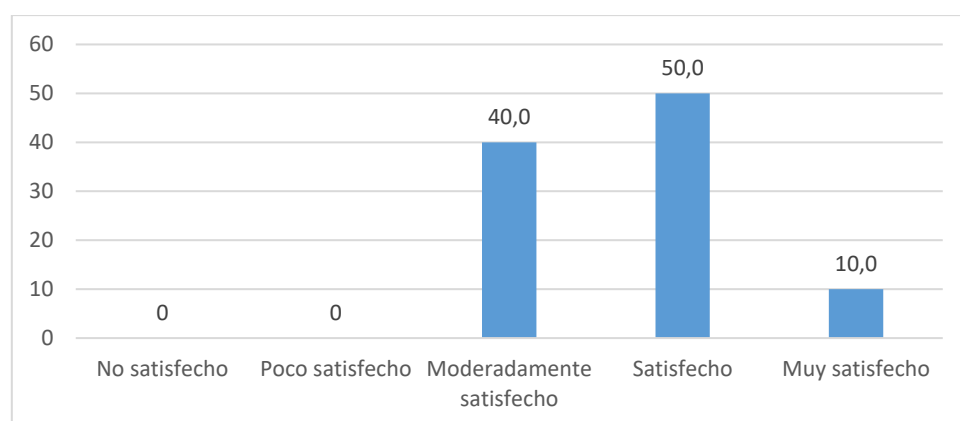


Interpretación: en la figura se observa que un 50% se encuentra moderadamente satisfecho, un 35% se encuentra satisfecho y un 15% se encuentra muy satisfecho con la seguridad de los datos que brinda el software para que no puedan ser modificados por otros.

P14: ¿Está satisfecho con la capacidad del software para registrar acciones indebidas u otras anomalías (auditorías)?

Figura 136:

Resultados de No repudio – post test

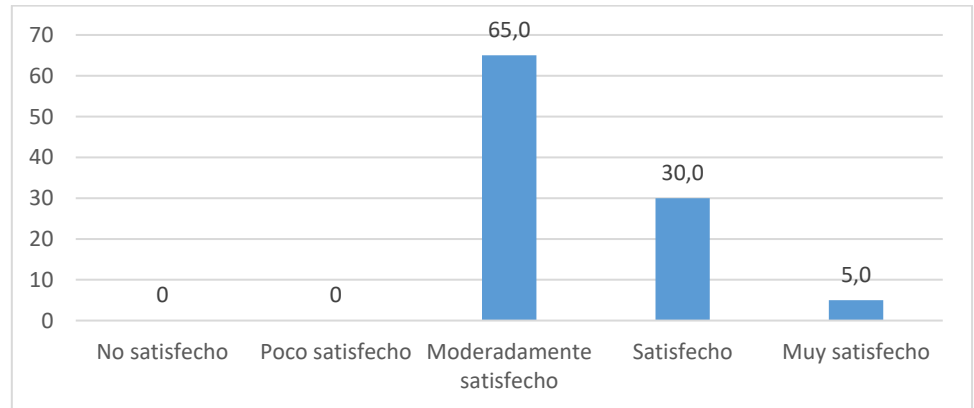


Interpretación: en la figura se observa que un 70% se encuentra moderadamente satisfecho y un 30% se encuentra satisfecho con la capacidad del software para poder registrar acciones indebidas y otras anomalías.

Dimensión: Adecuación Funcional

Figura 137:

Resultados de la dimensión Adecuación Funcional - post test

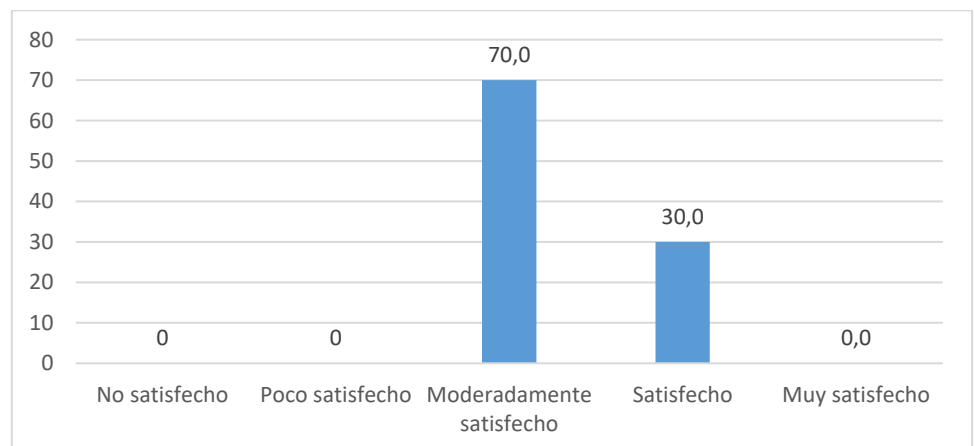


Interpretación: en la figura se observa que un 65% se encuentra moderadamente satisfecho, un 30% se encuentra satisfecho y un 5% se encuentra muy satisfecho con respecto a la adecuación funcional del software.

Dimensión: Eficiencia de desempeño

Figura 138:

Resultados de la dimensión Eficiencia de desempeño - post test

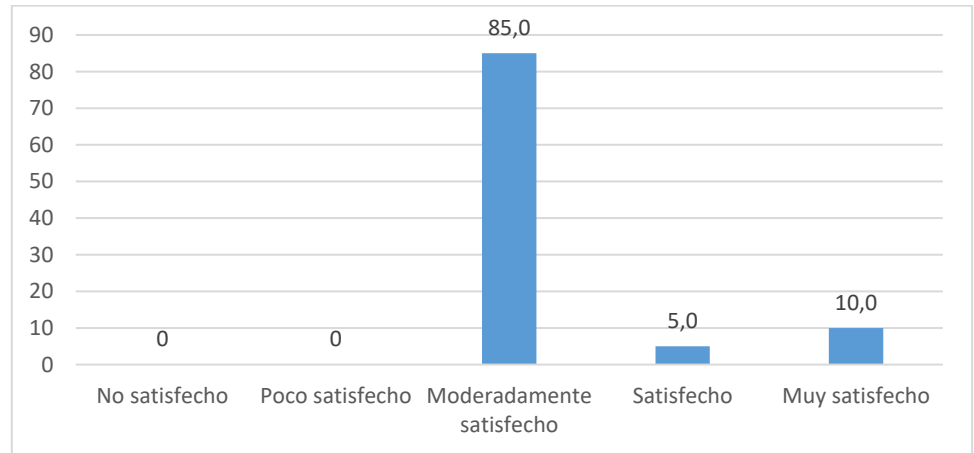


Interpretación: en la figura se observa que un 70% se encuentra moderadamente satisfecho y un 30% se encuentra satisfecho con la eficiencia de desempeño del software.

Dimensión: Compatibilidad

Figura 139:

Resultados de la dimensión Compatibilidad - post test

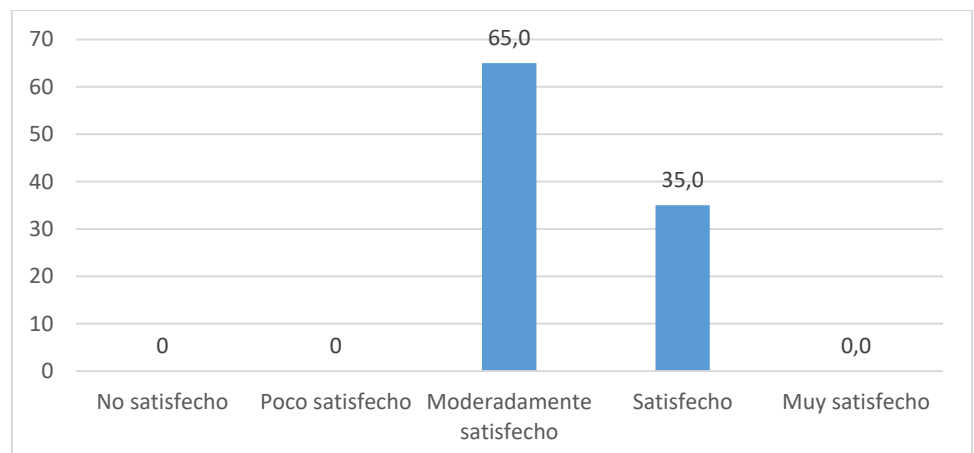


Interpretación: en la figura se observa que un 85% se encuentra moderadamente satisfecho, un 5% se encuentra satisfecho y un 10% se encuentra muy satisfecho con la compatibilidad del software.

Dimensión: Usabilidad

Figura 140:

Resultados de la dimensión Usabilidad - post test

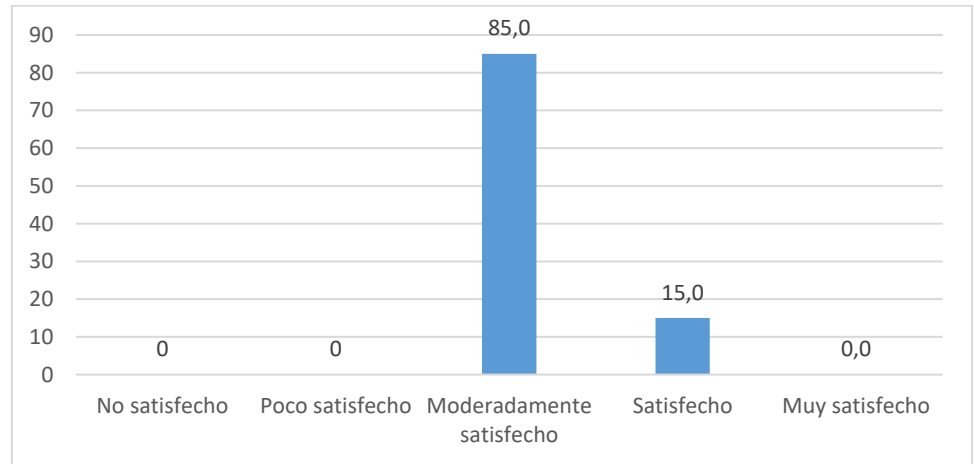


Interpretación: en la figura se observa que un 65% se encuentra moderadamente satisfecho y un 35% se encuentra satisfecho con la usabilidad del software.

Dimensión: Seguridad

Figura 141:

Resultados de la dimensión Seguridad - post test

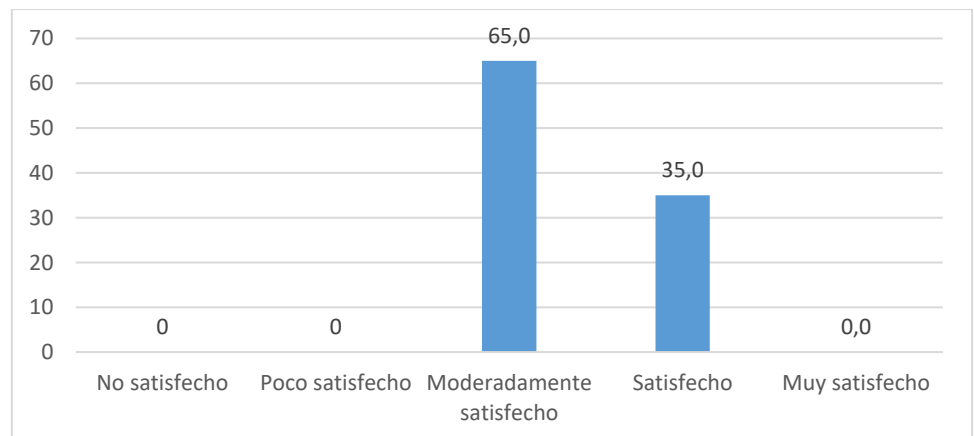


Interpretación: en la figura se observa que un 85% se encuentra moderadamente satisfecho y un 15% se encuentra satisfecho con la seguridad que brinda el software.

Variable: Calidad de Software

Figura 142:

Resultados para la variable Calidad de Software - post test



Interpretación: en la figura se observa que un 65% se encuentra moderadamente satisfecho y un 35% se encuentra satisfecho con la calidad del software.

Teniendo los datos previamente obtenidos del pre test y post test, se procedió a realizar la aplicación de estadísticos, los cuales ayudaron para el contraste de la hipótesis planteada dentro de la investigación, los resultados obtenidos fueron:

Tabla 36:

Análisis de la calidad de software antes y después

	Media	N	Desv. Desviación	Desv. Error promedio
Calidad de Software Pre test	3.2643	20	0.69256	0.15486
Calidad de Software Post test	3.9063	20	0.36899	0.08251

Como se observa en el contraste de la variable dependiente de la calidad de software, se ha tenido un incremento en la media, teniendo antes de la aplicación del quality assurance enfocado en agile testing un valor en la medida de 3.2643 y después un valor en la media de 3.9063, viéndose un incremento en la calidad de software.

Tabla 37:

Correlaciones de muestras emparejadas

	N	Correlación	Sig.
Calidad de Software Pre test & Calidad de Software Post test	20	0.955	0.000

4.2.2 Prueba de hipótesis

Para la prueba de resultados, previamente se empleó el análisis de fiabilidad mediante el Coeficiente Alfa de Cronbach, después el análisis para los datos, contrastando la variable dependiente antes y después de la aplicación de quality assurance enfocado en agile testing.

Con la hipótesis de la investigación se pasó a establecer las hipótesis correspondientes, tanto nula como alterna.

H₀: La aplicación de Quality Assurance enfocado en Agile Testing no mejora la calidad de software en la empresa Wasi Innovation SAC, Lima – 2022.

H₁: La aplicación de Quality Assurance enfocado en Agile Testing mejora la calidad de software en la empresa Wasi Innovation SAC, Lima – 2022.

Para poder contrastar la hipótesis general, se realizó el análisis para ver si los datos correspondían a un comportamiento paramétrico o no paramétrico, en tal caso se tuvo en cuenta la muestra menor a 30, por tanto se aplicó Shapiro Wilk con la siguiente regla de decisión:

Si $p_{valor} \leq 0.05$ los datos tienen un comportamiento no paramétrico.

Si $p_{valor} \geq 0.05$ los datos tienen un comportamiento paramétrico.

Además se pudo obtener el valor sacando mediante las diferencias de la calidad de software en pre test y post test.

Tabla 38:

Prueba de normalidad de la calidad de software

Diferencia de la Calidad de software pre test y post test	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
	0.179	20	0.093	0.930	20	0.155

En la tabla anterior se puede verificar que la significancia de la calidad de software es un valor de 0.155, mayor a 0.05, por tanto teniendo presente la regla de decisión antes detallada, se concluye que en la prueba de normalidad realizada se tiene como resultado un comportamiento paramétrico, por tanto se prosigue a realizar la prueba T Student.

Tabla 39:

Planteamiento para contrastar la hipótesis

O1	X	O2
Pre test	Aplicación de Quality Assurance enfocado en Agile Testing	Post test

Planteamiento

H₀: $O1 = O2 \rightarrow$ Se refiere a que después de la aplicación de quality assurance enfocado en agile testing no se logra tener ningún cambio o mejora.

H₁: $O1 < O2 \rightarrow$ Se refiere a que después de la aplicación de quality assurance enfocado en agile testing se logra tener una mejora en la calidad de software, obteniendo valores más altos que en el pre test.

H₀: La aplicación de Quality Assurance enfocado en Agile Testing no mejora la calidad de software en la empresa Wasi Innovation SAC, Lima – 2022.

H_a: La aplicación de Quality Assurance enfocado en Agile Testing mejora la calidad de software en la empresa Wasi Innovation SAC, Lima – 2022.

Para esto se realizó un análisis de la forma previamente planteada para contrastar la hipótesis según el enfoque estadístico con las siguientes reglas que se adecuaban a lo descrito anteriormente, por ende se seleccionó la forma de la Cola Inferior Izquierda.

Tabla 40:

Resultados de la prueba T Student

	Diferencias emparejadas						t	gl	Sig. (bilateral)
	Media	Desv. Desviación	Desv. Error promedio	Inferior	Superior				
Calidad de Software Pre test & Calidad de Software Post test	-0.64200	0.35731	0.07990	-0.80923	-0.47477	-8.035	19	0.000	

De la anterior tabla se puede observar que la significancia 0.000 tiene un valor menor a 0.05 y por tanto se re afirma la aceptación de la hipótesis alterna y se niega la hipótesis nula, por tanto quedando demostrado estadísticamente que la aplicación de Quality Assurance enfocado en Agile Testing mejora la calidad de software en la empresa Wasi Innovation SAC, Lima – 2022.

4.3 Discusión de resultados

Con respecto al objetivo general Determinar si la aplicación de Quality Assurance enfocado en Agile Testing mejora la calidad de software en la empresa Wasi Innovation SAC, Lima – 2022, se obtuvo como resultado que la aplicación de quality assurance enfocado en agile testing mejora la calidad de software en el contexto de investigación, lo cual es confirmado por la diferencia de medias en el pre test y post test, de igual forma en la tabla 40 que $p < 0.05$ como resultado en la prueba de hipótesis.

Según (Abdullahi Wakili et al. 2020) en su investigación Quality Assurance practices in Agile Methodology, en el evento International Journal of Advance Research in Science and Engineering, Osmania University, Hyderabad – India, realzan a la metodología ágil por su capacidad de flexibilidad en el desarrollo con pocos requisitos, admitiendo futuros cambios y en base a ello responder con un producto rápidamente pero esto no implica el respaldo de la calidad sin la aplicación de actividades, además menciona que las prácticas de métodos ágiles tienen habilidades de quality assurance. De la misma manera Tineo Gamboa & Quispe Cuadros (2019) en su trabajo de investigación Testing ágil en aplicaciones móviles android usando espresso, Universidad Tecnológica del Perú, Lima – Perú, mencionan que el testing es fundamental antes de lanzar un producto y en el desarrollo, también que la metodología Scrum es importante como marco de trabajo ágil para asignar tareas en el desarrollo y el testing en cada iteración existente, finalizando que el testing ágil es en los proyectos de software es crucial para el aporte en aspectos como presupuesto y tiempo, lo cual coincide con la presente investigación porque en base a ese enfoque se pudo seleccionar la metodología ágil, en específico el marco de trabajo Scrum para la creación del modelo de quality assurance enfocado en agile testing en conjunto con el enfoque del framework ISTQB debido a su flexibilidad a los cambios y los beneficios para implementar fases y actividades como planificación, monitoreo y control, análisis, diseño, implementación, ejecución y compleción de pruebas, también los diversos tipos de pruebas de software en los niveles existentes, y por otra parte las actividades, ceremonias, artefactos y otros del mismo marco de trabajo, de los cuales se pudo identificar la sinergia esperada debido a que manejaban el mismo enfoque ágil. Por consecuencia se pudo lograr un mejor manejo de las actividades de desarrollo y aseguramiento de la calidad en el producto desarrollado en ese momento.

Según (Thörn, 2020) en su tesis de maestría *Test framework quality assurance: augmenting agile processes with safety standards*, Mälardalen University, Västerås – Suecia, resalta que las pruebas de software son imprescindibles para el desarrollo y aun mucho más para el aseguramiento de la calidad dentro del desarrollo ágil, y que todo software o hardware se demuestra en base a las pruebas y una posterior toma de decisiones que se basan en la confianza de los resultados obtenidos en esas pruebas y que por consecuencia las pruebas de software son importantes para la confianza, lo cual coincide con la presente investigación ya que debido a los resultados obtenidos se demostró que la aplicación de quality assurance enfocado en agile testing, teniendo en cuenta la parte teórica y sobre todo práctica donde las pruebas de software fueron imprescindibles para mostrar el nivel de calidad del software mediante la ejecución de dichas pruebas y posterior análisis de los reportes como sustento tanto para el equipo como para el cliente como estrategia para la generación de confianza y aceptación.

Acorde a esto Lönnroth (2021) en su tesis de investigación *Improving quality control in automation projects using simulation systems*, KTH Royal Institute of Technology, Estocolmo – Suecia, resalta que a mayor capacidad de pruebas se podrá mejorar el proceso del control de la calidad y las pruebas funcionales pueden ser efectivas para poder mitigar errores que surjan en el desarrollo, tanto las pruebas estáticas como las dinámicas pueden ser efectivas para encontrar errores y si se aplica un mejor control de calidad se ahorrarían altos costos por la reelaboración, mientras más viva el error su crecimiento será exponencial tanto como los costos y retrabajo que este generen, así como también Paulduro (2018) en su tesis de maestría *Software development quality assurance process in regional e-commerce company*, Masaryk University, Brno – Republica Checa, tuvo un enfoque hacia las pruebas estáticas donde resalta la revisión de código, comentarios y posibles defectos, lo cual fue un paso exitoso porque ayudó a relevar futuros errores y un impacto severo en la aplicación en su lanzamiento, así como las pruebas unitarias y pruebas de servicios, finalizando que después la propuesta fue un éxito aunque difícil de medir porque aún se tenía un margen de mejora después de haber realizado todas las actividades establecidas, también Kim Nguyen (2022) en su tesis de maestría *Quality Assurance performance on Medical Technology*, California State Polytechnic University, California – Estados Unidos. Menciona los diversos tipos de pruebas tanto manuales como automatizadas y las herramientas tecnológicas son importantes para garantizar la calidad de software en todas las plataformas del mundo,

tanto aprender y tener conocimiento de toda tecnología así como el método de prueba ayudará a llevar la escala de precisión de lo posible, todo lo anteriormente descrito por dichos autores coincide con la investigación ya que varios de los resultados fueron ejecutados mediante las pruebas tanto estáticas como dinámicas ya que al abarcar un mayor número de tipos de pruebas se pudo tener una mayor cobertura en las pruebas tanto para su prevención y detección, así como para el reporte de los errores pero a pesar de lo realizado y que la calidad de software haya mejorado en base a quality assurance enfocado en agile testing, aún se tiene un margen de mejora debido al tiempo de madurez, también a la etapa en la que se encontraba el proyecto que era desarrollado en ese momento y las diversas pruebas de software que pudieron aplicarse. Y desde la perspectiva de la gestión de las pruebas Melgar Velásquez (2020) en su tesis de grado Implementación de un modelo de gestión de pruebas de software según ISTQB para mejorar el proceso del área de certificación en tecnologías web de una entidad financiera, Universidad Tecnológica del Perú, Lima – Perú, menciona que con el uso de aplicaciones como ALM para el diseño de los casos de prueba y Jira donde se puede tener un mejor control y transparencia en la atención de esos requerimientos, también que las fases de pruebas en base a ISTQB generaron un mejor resultado a partir de los requerimientos solicitados por el usuario y casos de prueba eficientes para diseñar escenarios y detectar defectos, así mismo Condori Fernández (2022) en su tesis de grado *Pruebas de software en aplicativos de canales digitales de una entidad bancaria usando ISTQB*, Universidad Nacional Mayor de San Marcos, Lima – Perú, menciona que mediante la gestión de casos de prueba con aplicaciones como TestLink se pudieron elaborar los diseños de casos de prueba, su ejecución teniendo un previo análisis u diseño, así como la identificación de los posibles riesgos, análisis de requerimientos y las estimaciones de las fechas de ejecución. Desde la perspectiva de ejecución y resultados, Villanueva Ajito (2021) en su tesis de grado Evaluación de la calidad de uso del software ecommerce de la empresa Mauripro Sailing según la ISO/IEC 25000, Lima, 2020, Universidad Nacional de Trujillo, Trujillo – Perú, realizó la evaluación de un software ecommerce en base a las características y subcaracterísticas de la ISO/IEC 25000 mediante la herramienta GQM (objetivo-pregunta-métrica), donde tuvo una gran variedad de resultados, finalizando que el software evaluado tiene una alta calidad aunque no llegue al 100% satisface las necesidades del cliente, este resultado coincide con la presente investigación ya que debido al tiempo y nivel de madurez no se pudo

obtener el 100% de calidad de software en un proyecto desarrollado por el contexto de investigación debido a la etapa de aplicación de quality assurance enfocado en agile testing y también la madurez que irá adquiriendo dependiendo de su adecuación tanto en el equipo y la reacción de los clientes.

Teniendo como base los antecedentes, marco teórico y los resultados hallados se puede declarar que el quality assurance enfocado en agile testing según la adecuación y madurez que vaya obteniendo, tanto teorica como práctica, aplicando los diversos tipos de pruebas, las fases de las pruebas, la sinergia con el marco de trabajo SCRUM y la realización de las ceremonias y gestión de artefactos, las herramientas como lenguajes de programación, aplicaciones para gestión de casos de prueba y otros requeridos, se podrá tener mejores resultados que los actuales, pudiendo pasar de un nivel satisfecho a muy satisfecho según nuestra escala valorativa presentada con anterioridad.

V. CONCLUSIONES

- La aplicación de Quality Assurance enfocado en Agile Testing mejora la calidad de software en la empresa Wasi Innovation SAC, Lima – 2022 porque se construyó un modelo teórico-práctico en base al estado del arte y necesidades del sector tecnológico, teniendo en cuenta la identificación de requerimientos y su adecuación dentro del contexto de la investigación según sus limitaciones, fortalezas, oportunidades y otros. Por otra parte el respaldo estadístico del resultado obtenido de la calidad de software antes y después, en el que se pudo visualizar un mayor incremento, también se pudo corroborar por la prueba estadística T de Student donde se obtuvo una significancia $p < 0,05$.
- Se realizó la identificación de requerimientos como estrategia para la elaboración y aplicación del modelo de quality assurance enfocado en agile testing acorde a los resultados obtenidos de parte del equipo de desarrollo, calidad, gerente de la empresa y otras personas involucradas.
- Se realizaron las pruebas de software, tanto funcionales, de estructura, servicios, de rendimiento y otros, como estrategia para evaluar el nivel de la calidad de software en el producto que estaba siendo desarrollado en ese momento y así tomar en cuenta las falencias como antecedentes e implementar mejoras con la aplicación de quality assurance enfocado en agile testing.
- Se realizó la creación y aplicación del modelo de quality assurance enfocado en agile testing teniendo como base el marco de trabajo SCRUM e ISTQB, donde se tuvo como resultado una sinergia esperada y requerida que mejoró la calidad de software.
- Se realizó la planificación de las pruebas dentro del modelo para poder gestionar los alcances a desarrollarse en un proyecto desde la perspectiva de pruebas, y se tuvo como resultado una mejor visión tanto para el equipo como para el cliente de que se realizará y que aspectos se emplearán.
- Se realizó el monitoreo y control de pruebas dentro del donde se tuvo como resultado un mejor manejo tanto del avance real y de lo establecido, y una mejor precisión en la toma de decisiones en los momentos requeridos para lograr los objetivos.
- Se realizó el análisis de pruebas dentro del modelo donde se tuvo como resultado la identificación de los requerimientos que contenían errores e incoherencias que fueron reportados y modificados.

- Se realizó el diseño de pruebas dentro del modelo donde se tuvo como resultado la elaboración de casos de prueba aplicando técnicas de diseño de pruebas de software y los cuales fueron sustento al finalizar cada sprint.
- Se realizó la implementación de pruebas dentro del modelo donde se tuvo como resultado la mejora en la cobertura de los tipos de pruebas, configuraciones y generación de código.
- Se realizó la Ejecución de pruebas dentro del modelo donde se obtuvieron los resultados de las pruebas, los cuales fueron analizados, registrados y se gestionaron los defectos.
- Se realizó la Compleción de pruebas dentro del modelo donde se tuvo como resultado la documentación requerida para la gestión de actividades y análisis de la situación real.

VI. RECOMENDACIONES

- Emplear la metodología ágil dentro del desarrollo y gestión de la calidad de software permite flexibilidad dentro de los procesos existentes y el proyecto en general porque coexiste con los cambios que surgen en el transcurso porque el cliente no tiene claro todos los requerimientos al iniciarse el proyecto.
- Quality assurance enfocado en agile testing teniendo como base Scrum e ISTQB comparten la misma visión del Agile Manifesto y por consecuencia una excelente sinergia del cual se puede sacar el máximo provecho en los proyectos de software.
- Tener en cuenta el tipo de cliente, ya que muchas entidades están acostumbradas a trabajar con metodologías tradicionales como cascada y para eso se debe realizar ciertas modificaciones en el flujo de trabajo del proyecto donde se aplicará.
- El modelo generado puede ser empleado por cualquier tipo de empresa pero se sugiere realizar un previo análisis sobre su implementación y adecuación, ya que al ser un modelo ágil en cada contexto de aplicación tiene ciertas características diferentes.
- Para el equipo de desarrollo y calidad de software del contexto donde se realizó la aplicación de quality assurance enfocado en agile testing, se sugiere tener en cuenta el modelo tanto teórico y práctico ya que cada fase, actividad u otro mantiene un orden y sucesión.
- Las pruebas de software no van a eliminar los errores existentes pero si se tiene una mayor cobertura de ellas se obtendrá un software con menores errores y mayor confianza, por ello se debe tener en cuenta los distintos niveles y tipos de pruebas de software.
- Las pruebas automatizadas tanto de servicios, funcionales, estructura u otro son importantes porque aminoran el tiempo en las pruebas de software pero se debe evaluar específicamente la cobertura de las pruebas funcionales automatizadas porque si los casos de prueba son cortos o no son repetitivos, es mejor no realizarlos porque para ello se debe generar un ambiente, codificación y otros los cuales pueden tener un tiempo estimado mayor en contraste a una prueba funcional manual.

VII. REFERENCIAS BIBLIOGRÁFICAS

- Abdullahi Wakili, A., Nasiru Alhasan, L. y Hamisu Kamagata, A. (2020). *Quality assurance practices in agile methodology*. International Journal of Advance Research in Science and Engineering, Hyderabad, India.
- Amazon Web Services. (01 de Agosto de 2022). *¿Qué es una API?*. <https://aws.amazon.com/es/what-is/api/>
- Atlassian Agile Coach. (28 de julio de 2022). *Historias de usuario*. <https://www.atlassian.com/es/agile/project-management/user-stories>
- Atlassian. (20 de julio de 2022). *¿Qué es scrum?*. <https://www.atlassian.com/es/agile/scrum>
- Atlassian. (20 de julio de 2022). *About Atlassian*. <https://www.atlassian.com/es>
- BBVA. (24 de julio de 2022). *Pymes y Mypes*. <https://www.bbva.com/es/pymes-y-mypes-en-que-se-diferencian/>
- BCC News Mundo. (19 de julio de 2022). *Toyota admite fallas en freno del Prius*. https://www.bbc.com/mundo/economia/2010/02/100204_toyota_vehiculos_prius_frenos_ganancias_prius_jp
- Callejas Cuervo, M., Alarcón Aldana, A. C. y Álvarez Carreño, A. M. (2017). Modelos de calidad del software, un estado del arte. *Entramado*. <http://www.scielo.org.co/pdf/entra/v13n1/1900-3803-entra-13-01-00236.pdf>
- Carrizo, D., & Alfaro, A. (2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. *Ingeniare Revista Chilena de Ingeniería*, 26(1), 16. <https://www.scielo.cl/pdf/ingeniare/v26n1/0718-3305-ingeniare-26-01-00114.pdf>
- Chaves, B. E. y Rodriguez M, L. (2016). Análisis de confiabilidad y validez en un cuestionario sobre los entornos personales de aprendizaje. *Revista Ensayos Pedagógicos*, <https://dialnet.unirioja.es/descarga/articulo/7038088.pdf>

Comisión de Reglamentos Técnicos y Comerciales - INDECOPI. (2006). *Norma Técnica Peruana NTP-ISO/IEC 12207:2006*. INDECOPI. https://www.academia.edu/16281496/NTP_ISO_IEC_12207

Condori Fernández, E. E. (2022). *Pruebas de software en aplicativos de canales digitales de una entidad bancaria usando ISTQB*. [Tesis de grado, Universidad Nacional Mayor de San Marcos]. Repositorio Institucional - Universidad Nacional Mayor de San Marcos. https://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/17876/Condori_fe.pdf?sequence=1&isAllowed=y

Crisis del software. (10 de julio de 2022). En *Wikipedia*. https://es.wikipedia.org/wiki/Crisis_del_software

Crispin, L., & Gregory, J. (2009). *Agile Testing*. Addison Wesley.

Departamento de Inteligencia de Mercados. (2021). Panorama de la Industria de Software y Servicios de Informática. *Prom Perú*. https://boletines.exportemos.pe/recursos/boletin/2255806_Informe_especializado_Industria_del_Software_25.06.21.pdf

Digite. (07 de agosto de 2022). *ATDD Colaboración centrada en el usuario*. <https://www.digite.com/es/agile/el-desarrollo-orientado-a-pruebas-de-aceptacion-atdd/>

Educación en línea (21 de julio de 2022). En

Wikipedia. https://es.wikipedia.org/wiki/Educaci%C3%B3n_en_l%C3%ADnea

ESAN. (04 de agosto de 2021). *Diferenciar Stakeholders*. <https://www.esan.edu.pe/conexion-esan/como-identificar-a-los-stakeholders-de-tu-organizacion>

Figueroa Martins, J. J. y Narea Gutiérrez, A. G. (2016). *Competencias del modelo ucab desarrolladas mediante las actividades ejecutadas en la séptima edición del moit*. [Tesis de grado - Universidad Católica Andrés Bello]. Repositorio Institucional -

Universidad Católica Andrés Bello.
<http://biblioteca2.ucab.edu.ve/anexos/biblioteca/marc/texto/AAT5194.pdf>

Forbes. (27 de julio de 2022). *What is Waterfall Methodology?*
<https://www.forbes.com/advisor/business/what-is-waterfall-methodology/>

Gómez, M. (2006). *Introducción a la Metodología de la Investigación Científica*. Editorial Brujas.

Hoyos, P. (11 de 08 de 2022). *Testing de software y su importancia*. <https://pacifitic.org/que-es-el-testing-de-software-y-por-que-es-tan-importante-en-el-desarrollo-de-software/>

Ibarra Luevano, S. A. (2018). *Desarrollo de una herramienta de soporte para el aseguramiento de la calidad en productos de software*. [Tesis de maestría, Centro de Investigación de Matemáticas, Guanajuato - México]. Repositorio Institucional - Centro de Investigación de Matemáticas.
<https://cimat.repositorioinstitucional.mx/jspui/bitstream/1008/1041/1/ZAC%20TE%20066.pdf>

IBM. (10 de agosto de 2022). *Test Execution States*.
<https://www.ibm.com/docs/en/elm/7.0.0?topic=application-test-execution-states>

Immune Technology Institute. (08 de agosto de 2021). *BDD Definición e Importancia en el desarrollo*. <https://immune.institute/blog/bdd-que-es-y-por-que-es-importante/>

Internacional Organization for Standardization. (2012). *Normal internacional ISO 9000:2000*. Ginebra, Suiza: ISO.

International Software Testing Qualifications Board. (14 de julio de 2022). En *Wikipedia*.
https://es.wikipedia.org/wiki/International_Software_Testing_Qualifications_Board

ISO 25000. (12 de agosto de 2022). *ISO/IEC 25010*. <https://iso25000.com/index.php/normas-iso-25000/iso-25010>

- ISTQB ®. (06 de agosto de 2022). *What ISTQB do*. <https://www.istqb.org/about-us/what-we-do>
- ISTQB. (01 de agosto de 2022). *ISTQB Glossary*. <https://glossary.istqb.org/en/search/quality%20assurance>
- ISTQB. (28 de julio de 2018). *Certified Tester Foundation level (CTFL) Syllabus*. https://istqb-main-web-prod.s3.amazonaws.com/media/documents/ISTQB-CTFL_Syllabus_2018_v3.1.1.pdf
- Kaspersky. (23 de julio de 2022). *Bots: una definición*. <https://latam.kaspersky.com/resource-center/definitions/what-are-bots>
- Khalane, T. (2013). *Software Quality Assurance in Scrum*. [Tesis de maestría, University of Cape Town]. Repositorio Institucional - University of Cape Town, Sudafrica. https://open.uct.ac.za/bitstream/item/5604/thesis_com_2013_khalane_t.pdf?sequence=1
- Kim Nguyen, N. (2022). *Quality assurance performance on medical technology*. [Tesis de maestría, California State Polytechnic University]. Repositorio Institucional - California State Polytechnic University. <https://scholarworks.calstate.edu/downloads/td96k846m>
- Lönnroth, V. (2021). *Improving quality control in automation project using simulation systems*. [tesis de nivel avanzado, KTH Royal Institute of Technology]. Repositorio Institucional - KTH Royal Institute of Technology, Suecia. <https://www.diva-portal.org/smash/get/diva2:1552546/FULLTEXT01.pdf>
- Marulanda López, J. E. (2014). *Aseguramiento de la calidad en el diseño del software*. [Tesis de grado, Universidad EAFIT]. Repositorio Institucional - Universidad EAFIT, Medellin. https://repository.eafit.edu.co/bitstream/handle/10784/5270/Jaime_MarulandaLopez_2014.pdf?sequence=2&isAllowed=y

- McGuigan, F. (1996). *Psicología Experimental - Métodos de Investigación*. Prentice Hall Hispanoamérica S.A.
- Melgar Velásquez, R. (2020). *Implementación de un modelo de gestión de pruebas de software según ISTQB para mejorar el proceso el área de certificación en tecnología web de una entidad financiera*. [Tesis de grado, Universidad Tecnológica del Perú]. Repositorio Institucional - Universidad Tecnológica del Perú, Perú. https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/3290/Roberth%20Melgar_Trabajo%20de%20Suficiencia%20Profesional_Titulo%20Profesional_2020.pdf?sequence=1&isAllowed=y
- Metahabitos. (2022). *Implementación de metodología scrum en proyectos*. <https://www.metahabitos.com/gestion-de-proyectos/metodologia-scrum/>
- Millones, R. R. y Amable Ciudad, M. (2018). Uso de modelos de calidad en las mypes productoras de software de Lima. *Revistas Universidad de Lima*, 1(1), 20. https://revistas.ulima.edu.pe/index.php/Ingenieria_industrial/article/view/4543/4488
- Organización Internacional de Normalización (26 de julio de 2022). En *Wikipedia*. https://es.wikipedia.org/wiki/Organizaci%C3%B3n_Internacional_de_Normalizaci%C3%B3n
- Paulduro, T. (2018). *Software development quality assurance process in regional e-commerce company*. [Tesis de maestría, Masaryk University]. Repositorio Institucional - Masaryk University, Republica Checa. <https://is.muni.cz/th/llz4x/thesis-paulduro.pdf>
- Poder Legislativo del Perú. (17 de agosto de 2011). *Ley N° 29733 - Ley de protección de los datos personales*. <https://www.leyes.congreso.gob.pe/Documentos/Leyes/29733.pdf>
- Pressman, R. (2010). *Ingeniería de Software*. Mc Graw Hill.
- Real Academia Española. (03 de agosto de 2022). *Hardware*. <https://www.rae.es/dpd/hardware>
- Real Academia Española. (05 de agosto de 2022). *Calidad*. <https://dle.rae.es/calidad?m=form>

Retail. (01 de Agosto de 2022). En *Wikipedia*. <https://es.wikipedia.org/wiki/Retail>

Sánchez Peño, J. M. (2015). *Pruebas de software. Fundamentos y Técnicas*. [tesis de grado, Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicaciones]. Repositorio Institucional - Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicaciones, Madrid, España. https://oa.upm.es/40012/1/PFC_JOSE_MANUEL_SANCHEZ_PENO_3.pdf

Sánchez-Oro Portillo, A. F. (2017). *Agiles Testing - Estado del arte - Su aplicación en empresas TIC de Extremadura*. [Tesis de grado, Universidad de Extremadura]. Repositorio Institucional - Universidad de Extremadura, España. https://dehesa.unex.es/bitstream/10662/6617/1/TFGUEX_2017_Sanchez-Oro_Portillo.pdf

Schwaber, K., & Sutherland, J. (2022). *Scrum Guides*.

Software Factory (19 de julio de 2022). En *Wikipedia*. https://en.wikipedia.org/wiki/Software_factory

Testlink. (10 de Agosto de 2022). En *Wikipedia*. <https://en.wikipedia.org/wiki/TestLink>

Thörn, J. (2020). *Test framework quality assurance: augmenting agile processes with safety standards*. [Tesis de maestría, Mälardalen University, Västerås]. Repositorio Institucional - Mälardalen University, Västerås, Suecia. <https://www.diva-portal.org/smash/get/diva2:1436123/FULLTEXT01.pdf>

Tineo Gamboa, B., & Quispe Cuadros, N. (2019). *Testing Ágil en aplicaciones móviles android usando espresso*. [Tesis de bachillerato, Universidad Tecnológica del Perú]. Repositorio Institucional - Universidad Tecnológica del Perú, Perú. https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/2037/Nicanor%20Quispe_Breizer%20Tineo_Trabajo%20de%20Investigacion_Bachiller_2019.pdf?sequence=1&isAllowed=y

- Valenzuela Salvador, J. E. (2020). *Sistema Web mediante indicadores de resultados para la mejora de la Gestión Administrativa en la Universidad Nacional Santiago Antúnez de mayolo - Huaraz 2017*. [Tesis de grado, Universidad Nacional Santiago Antúnez de Mayolo]. Repositorio Institucional - Universidad Nacional Santiago Antúnez de Mayolo, Perú. <http://repositorio.unasam.edu.pe/handle/UNASAM/5009>
- Villadiego, R. (23 de julio de 2022). *¿Por qué el Software conquistó al mundo?* <https://forbes.co/2020/08/27/red-forbes/por-que-el-software-conquistó-al-mundo/>
- Villanueva Ajito, C. N. (2021). *Evaluación de la calidad del uso del software ecommerce de la empresa Mauripro Sailing según la ISO/IEC 25000, Lima, 2020*. [Tesis de grado, Universidad Nacional de Trujillo]. Repositorio Institucional - Universidad Nacional de Trujillo. <https://dspace.unitru.edu.pe/bitstream/handle/UNITRU/16714/Villanueva%20Ajito.pdf?sequence=1&isAllowed=y>
- Visual México. (16 de julio de 2022). *Evolución de la calidad*. <https://visualmexico.com.mx/evolucion-de-la-calidad/>

VIII. ANEXOS

- Matriz de consistencia de la investigación.
- Instrumento de recolección de datos – Cuestionario.
- Carta de presentación para Juicio de Expertos.
- Escala para validación de matriz de Juicio de Expertos.
- Matriz de validación de Juicio de Expertos.
- Reporte de pruebas de software.

MATRIZ DE CONSISTENCIA DE LA INVESTIGACIÓN

TÍTULO	PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES	DIMENSIONES E INDICADORES	METODOLOGÍA
<p style="text-align: center;">QUALITY ASSURANCE ENFOCADO EN AGILE TESTING PARA MEJORAR LA CALIDAD DE SOFTWARE EN LA EMPRESA WASI INNOVATION, LIMA - 2022</p>	<p>¿De qué manera la aplicación de Quality Assurance enfocado en Agile Testing mejora la Calidad de Software en la empresa Wasi Innovation, Lima – 2022?</p>	<p>OBJETIVO GENERAL Determinar si la aplicación de Quality Assurance enfocado en Agile Testing mejora la calidad de software en la empresa Wasi Innovation SAC, Lima – 2022.</p> <p>OBJETIVOS ESPECÍFICOS OE1: Identificar la realidad de la calidad de software antes de la aplicación de Quality Assurance enfocado en Agile Testing en la empresa Wasi Innovation SAC, Lima – 2022</p> <p>OE2: Identificar la realidad de la calidad de software después de la aplicación de Quality Assurance enfocado en Agile Testing en la empresa Wasi Innovation SAC, Lima – 2022.</p>	<p>HIPÓTESIS GENERAL La aplicación de Quality Assurance enfocado en Agile Testing mejora la calidad de software en la empresa Wasi Innovation SAC, Lima – 2022.</p> <p>HIPÓTESIS ESPECÍFICAS HE1: Lograr identificar la realidad de la calidad de software antes de la aplicación de Quality Assurance enfocado en Agile Testing en la empresa Wasi Innovation SAC, Lima – 2022.</p> <p>HE2: Lograr identificar la realidad de la calidad de software después de la aplicación de Quality Assurance enfocado en Agile Testing en la empresa Wasi Innovation SAC, Lima – 2022.</p>	<p>V.I.</p> <p>Quality Assurance enfocado en Agile Testing</p>	<p>FRAMEWORK ISTQB</p> <ul style="list-style-type: none"> • Planificación • Monitorización y Control • Análisis • Diseño • Implementación • Ejecución • Compleción 	<p style="text-align: center;">TIPO DE INVESTIGACIÓN</p> <p>La presente investigación se caracteriza por tener un enfoque cuantitativo, descriptivo-correlacional y por grado de abstracción aplicada.</p> <p>Cuantitativa: porque se realiza la recolección y el análisis de los datos para responder a las incógnitas de la investigación y contrastar la hipótesis (Gómez, 2006).</p> <p>Descriptivo-correlacional: descriptivo porque los estudios se enfocan en especificar las características y otros aspectos importantes del fenómeno que se está sometiendo al análisis y correlacional por la relación existente entre dos o más conceptos en un contexto particular (Gómez, 2006).</p> <p>Aplicada: porque se hace uso de los conocimientos adquiridos previamente para la solución de un problema identificado (Hernández et al. 2014, citado en Valenzuela Salvador, 2020).</p> <p>DISEÑO DE LA INVESTIGACIÓN</p> <ul style="list-style-type: none"> • Pre-experimental en la cual se tuvo en cuenta la realización de las pruebas en dos etapas (Hernández et al. 2014). <p style="text-align: center;">POBLACIÓN Y MUESTRA</p> <p>La población conforma un total de 20 integrantes. Por ende la muestra es censal ya que la población al ser pequeña se puede recolectar la información de todos ellos (Valenzuela Salvador, 2020).</p>
				<p>V.D.</p> <p>Calidad de Software</p>	<p>ISO/IEC 25010</p> <ul style="list-style-type: none"> • Adecuación Funcional • Eficiencia de desempeño • Compatibilidad • Usabilidad • Seguridad 	

Instrumento de recolección de datos

Cuestionario

Recolección de datos de la calidad de software (pre test)

Fecha: 31/08/2022 **Rol:** Gerente general

Nombres y Apellidos: Wiliam Castro

El presente cuestionario correspondiente a la investigación denominada *Quality Assurance enfocado en Agile Testing para mejorar la Calidad de Software en la empresa Wasi Innovation, Lima – 2022*, por lo cual se requiere su participación con el fin de determinar el grado de satisfacción con respecto a la calidad del software en los proyectos que se realizan dentro de la empresa donde labora.

Instrucciones: Acorde a las preguntas que se presentan deberá marcar con una equis (X) una alternativa según corresponda.

Escala valorativa:

No satisfecho	Poco satisfecho	Moderadamente satisfecho	Satisfecho	Muy Satisfecho
1	2	3	4	5

N°	ITEMS	ESCALA				
		1	2	3	4	5
1	¿Está satisfecho con funcionalidades que brinda el software según los requerimientos establecidos?		X			
2	¿Está satisfecho con la precisión de los resultados que brinda el software?			X		
3	¿En qué grado está satisfecho en relación a las actividades de su día a día que puede realizar con el producto?				X	
4	¿Está satisfecho con el tiempo que demora realizar los procesos como las matriculas, reportes, emisión de certificados y otros?			X		
5	¿Está satisfecho con el software y los requisitos/recursos que se necesitan para que pueda usarlo?				X	
6	¿Está satisfecho con el funcionamiento del software con otros como el moodle?	X				
7	¿Está satisfecho con el intercambio de información entre software con el moodle y otros?		X			
8	¿Está satisfecho con la facilidad de aprendizaje del software?				X	
9	¿Está satisfecho con la accesibilidad al software (tiene la capacidad de ser usado por personas con distintas características)?				X	

10	¿Está satisfecho con la prevención de errores del usuario dentro del software?			X		
11	¿Está satisfecho con el diseño, formas y colores del software?			X		
12	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser accedidos por otro?			X		
13	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser modificados por otro?			X		
14	¿Está satisfecho con cómo el sistema registra las acciones realizadas por los usuarios de forma tal que no puedan negar sus acciones indebidas u otras anomalías?			X		

Cuestionario

Recolección de datos de la calidad de software (pre test)

Fecha: 22/08/2022 **Rol en el proyecto:** Desarrollador

Nombres y Apellidos: Rick Moreno

El presente cuestionario correspondiente a la investigación denominada *Quality Assurance enfocado en Agile Testing para mejorar la Calidad de Software en la empresa Wasi Innovation, Lima – 2022*, por lo cual se requiere su participación con el fin de determinar el grado de satisfacción con respecto a la calidad del software en los proyectos que se realizan dentro de la empresa donde labora.

Instrucciones: Acorde a las preguntas que se presentan deberá marcar con una equis (X) una alternativa según corresponda

Escala valorativa:

No satisfecho	Poco satisfecho	Moderadamente satisfecho	Satisfecho	Muy Satisfecho
1	2	3	4	5

N°	ITEMS	ESCALA				
		1	2	3	4	5
1	¿Está satisfecho con funcionalidades que brinda el software según los requerimientos establecidos?				X	
2	¿Está satisfecho con la precisión de los resultados que brinda el software?				X	
3	¿En qué grado está satisfecho en relación a las actividades de su día a día que puede realizar con el producto?				X	
4	¿Está satisfecho con el tiempo que demora realizar los procesos como las matriculas, reportes, emisión de certificados y otros?				X	
5	¿Está satisfecho con el software y los requisitos/recursos que se necesitan para que pueda usarlo?				X	
6	¿Está satisfecho con el funcionamiento del software con otros como el moodle?			X		
7	¿Está satisfecho con el intercambio de información entre software con el moodle y otros?			X		
8	¿Está satisfecho con la facilidad de aprendizaje del software?				X	
9	¿Está satisfecho con la accesibilidad al software (tiene la capacidad de ser usado por personas con distintas características)?					X

10	¿Está satisfecho con la prevención de errores del usuario dentro del software?					X
11	¿Está satisfecho con el diseño, formas y colores del software?			X		
12	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser accedidos por otro?				X	
13	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser modificados por otro?					X
14	¿Está satisfecho con cómo el sistema registra las acciones realizadas por los usuarios de forma tal que no puedan negar sus acciones indebidas u otras anomalías?			X		

Cuestionario

Recolección de datos de la calidad de software (pre test)

Fecha: 22/08/2022 **Rol en el proyecto:** Jefe de Proyecto

Nombres y Apellidos: Stephanie Carrión

El presente cuestionario correspondiente a la investigación denominada *Quality Assurance enfocado en Agile Testing para mejorar la Calidad de Software en la empresa Wasi Innovation, Lima – 2022*, por lo cual se requiere su participación con el fin de determinar el grado de satisfacción con respecto a la calidad del software en los proyectos que se realizan dentro de la empresa donde labora.

Instrucciones: Acorde a las preguntas que se presentan deberá marcar con una equis (X) una alternativa según corresponda.

Escala valorativa:

No satisfecho	Poco satisfecho	Moderadamente satisfecho	Satisfecho	Muy Satisfecho
1	2	3	4	5

N°	ITEMS	ESCALA				
		1	2	3	4	5
1	¿Está satisfecho con funcionalidades que brinda el software según los requerimientos establecidos?			X		
2	¿Está satisfecho con la precisión de los resultados que brinda el software?			X		
3	¿En qué grado está satisfecho en relación a las actividades de su día a día que puede realizar con el producto?			X		
4	¿Está satisfecho con el tiempo que demora realizar los procesos como las matriculas, reportes, emisión de certificados y otros?			X		
5	¿Está satisfecho con el software y los requisitos/recursos que se necesitan para que pueda usarlo?			X		
6	¿Está satisfecho con el funcionamiento del software con otros como el moodle?				X	
7	¿Está satisfecho con el intercambio de información entre software con el moodle y otros?			X		
8	¿Está satisfecho con la facilidad de aprendizaje del software?			X		
9	¿Está satisfecho con la accesibilidad al software (tiene la capacidad de ser usado por personas con distintas características)?				X	

10	¿Está satisfecho con la prevención de errores del usuario dentro del software?				X	
11	¿Está satisfecho con el diseño, formas y colores del software?			X		
12	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser accedidos por otro?				X	
13	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser modificados por otro?				X	
14	¿Está satisfecho con cómo el sistema registra las acciones realizadas por los usuarios de forma tal que no puedan negar sus acciones indebidas u otras anomalías?				X	

Cuestionario

Recolección de datos de la calidad de software (pre test)

Fecha: 23/08/2022 **Rol en el proyecto:** QA

Nombres y Apellidos: Ibette Pamela

El presente cuestionario correspondiente a la investigación denominada *Quality Assurance enfocado en Agile Testing para mejorar la Calidad de Software en la empresa Wasi Innovation, Lima – 2022*, por lo cual se requiere su participación con el fin de determinar el grado de satisfacción con respecto a la calidad del software en los proyectos que se realizan dentro de la empresa donde labora.

Instrucciones: Acorde a las preguntas que se presentan deberá marcar con una equis (X) una alternativa según corresponda.

Escala valorativa:

No satisfecho	Poco satisfecho	Moderadamente satisfecho	Satisfecho	Muy Satisfecho
1	2	3	4	5

N°	ITEMS	ESCALA				
		1	2	3	4	5
1	¿Está satisfecho con funcionalidades que brinda el software según los requerimientos establecidos?			X		
2	¿Está satisfecho con la precisión de los resultados que brinda el software?			X		
3	¿En qué grado está satisfecho en relación a las actividades de su día a día que puede realizar con el producto?			X		
4	¿Está satisfecho con el tiempo que demora realizar los procesos como las matriculas, reportes, emisión de certificados y otros?			X		
5	¿Está satisfecho con el software y los requisitos/recursos que se necesitan para que pueda usarlo?			X		
6	¿Está satisfecho con el funcionamiento del software con otros como el moodle?		X			
7	¿Está satisfecho con el intercambio de información entre software con el moodle y otros?		X			
8	¿Está satisfecho con la facilidad de aprendizaje del software?		X			
9	¿Está satisfecho con la accesibilidad al software (tiene la capacidad de ser usado por personas con distintas características)?			X		

10	¿Está satisfecho con la prevención de errores del usuario dentro del software?			X		
11	¿Está satisfecho con el diseño, formas y colores del software?			X		
12	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser accedidos por otro?		X			
13	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser modificados por otro?		X			
14	¿Está satisfecho con cómo el sistema registra las acciones realizadas por los usuarios de forma tal que no puedan negar sus acciones indebidas u otras anomalías?			X		

Cuestionario

Recolección de datos de la calidad de software (pre test)

Fecha: 23/08/2022 **Rol en el proyecto:** Gestor de Calidad

Nombres y Apellidos: Emily Salcedo

El presente cuestionario correspondiente a la investigación denominada *Quality Assurance enfocado en Agile Testing para mejorar la Calidad de Software en la empresa Wasi Innovation, Lima – 2022*, por lo cual se requiere su participación con el fin de determinar el grado de satisfacción con respecto a la calidad del software en los proyectos que se realizan dentro de la empresa donde labora.

Instrucciones: Acorde a las preguntas que se presentan deberá marcar con una equis (X) una alternativa según corresponda.

Escala valorativa:

No satisfecho	Poco satisfecho	Moderadamente satisfecho	Satisfecho	Muy Satisfecho
1	2	3	4	5

N°	ITEMS	ESCALA				
		1	2	3	4	5
1	¿Está satisfecho con funcionalidades que brinda el software según los requerimientos establecidos?		X			
2	¿Está satisfecho con la precisión de los resultados que brinda el software?			X		
3	¿En qué grado está satisfecho en relación a las actividades de su día a día que puede realizar con el producto?				X	
4	¿Está satisfecho con el tiempo que demora realizar los procesos como las matriculas, reportes, emisión de certificados y otros?			X		
5	¿Está satisfecho con el software y los requisitos/recursos que se necesitan para que pueda usarlo?				X	
6	¿Está satisfecho con el funcionamiento del software con otros como el moodle?	X				
7	¿Está satisfecho con el intercambio de información entre software con el moodle y otros?		X			
8	¿Está satisfecho con la facilidad de aprendizaje del software?				X	
9	¿Está satisfecho con la accesibilidad al software (tiene la capacidad de ser usado por personas con distintas características)?				X	

10	¿Está satisfecho con la prevención de errores del usuario dentro del software?			X		
11	¿Está satisfecho con el diseño, formas y colores del software?			X		
12	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser accedidos por otro?			X		
13	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser modificados por otro?			X		
14	¿Está satisfecho con cómo el sistema registra las acciones realizadas por los usuarios de forma tal que no puedan negar sus acciones indebidas u otras anomalías?			X		

Cuestionario

Recolección de datos de la calidad de software (pre test)

Fecha: 26/08/2022 **Rol en el proyecto:** Arquitecto de Software

Nombres y Apellidos: Arley Henostroza

El presente cuestionario correspondiente a la investigación denominada *Quality Assurance enfocado en Agile Testing para mejorar la Calidad de Software en la empresa Wasi Innovation, Lima – 2022*, por lo cual se requiere su participación con el fin de determinar el grado de satisfacción con respecto a la calidad del software en los proyectos que se realizan dentro de la empresa donde labora.

Instrucciones: Acorde a las preguntas que se presentan deberá marcar con una equis (X) una alternativa según corresponda.

Escala valorativa:

No satisfecho	Poco satisfecho	Moderadamente satisfecho	Satisfecho	Muy Satisfecho
1	2	3	4	5

N°	ITEMS	ESCALA				
		1	2	3	4	5
1	¿Está satisfecho con funcionalidades que brinda el software según los requerimientos establecidos?		X			
2	¿Está satisfecho con la precisión de los resultados que brinda el software?					X
3	¿En qué grado está satisfecho en relación a las actividades de su día a día que puede realizar con el producto?				X	
4	¿Está satisfecho con el tiempo que demora realizar los procesos como las matriculas, reportes, emisión de certificados y otros?				X	
5	¿Está satisfecho con el software y los requisitos/recursos que se necesitan para que pueda usarlo?				X	X
6	¿Está satisfecho con el funcionamiento del software con otros como el moodle?				X	
7	¿Está satisfecho con el intercambio de información entre software con el moodle y otros?				X	
8	¿Está satisfecho con la facilidad de aprendizaje del software?					X
9	¿Está satisfecho con la accesibilidad al software (tiene la capacidad de ser usado por personas con distintas características)?				X	

10	¿Está satisfecho con la prevención de errores del usuario dentro del software?					X
11	¿Está satisfecho con el diseño, formas y colores del software?				X	
12	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser accedidos por otro?				X	
13	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser modificados por otro?				X	
14	¿Está satisfecho con cómo el sistema registra las acciones realizadas por los usuarios de forma tal que no puedan negar sus acciones indebidas u otras anomalías?				X	

Cuestionario

Recolección de datos de la calidad de software (pre test)

Fecha: 28/08/2022 **Rol en el proyecto:** Desarrollo

Nombres y Apellidos: Osterling

El presente cuestionario correspondiente a la investigación denominada *Quality Assurance enfocado en Agile Testing para mejorar la Calidad de Software en la empresa Wasi Innovation, Lima – 2022*, por lo cual se requiere su participación con el fin de determinar el grado de satisfacción con respecto a la calidad del software en los proyectos que se realizan dentro de la empresa donde labora.

Instrucciones: Acorde a las preguntas que se presentan deberá marcar con una equis (X) una alternativa según corresponda.

Escala valorativa:

No satisfecho	Poco satisfecho	Moderadamente satisfecho	Satisfecho	Muy Satisfecho
1	2	3	4	5

N°	ITEMS	ESCALA				
		1	2	3	4	5
1	¿Está satisfecho con funcionalidades que brinda el software según los requerimientos establecidos?				X	
2	¿Está satisfecho con la precisión de los resultados que brinda el software?			X		
3	¿En qué grado está satisfecho en relación a las actividades de su día a día que puede realizar con el producto?			X		
4	¿Está satisfecho con el tiempo que demora realizar los procesos como las matriculas, reportes, emisión de certificados y otros?			X		
5	¿Está satisfecho con el software y los requisitos/recursos que se necesitan para que pueda usarlo?				X	
6	¿Está satisfecho con el funcionamiento del software con otros como el moodle?				X	
7	¿Está satisfecho con el intercambio de información entre software con el moodle y otros?				X	
8	¿Está satisfecho con la facilidad de aprendizaje del software?				X	
9	¿Está satisfecho con la accesibilidad al software (tiene la capacidad de ser usado por personas con distintas características)?		X			

10	¿Está satisfecho con la prevención de errores del usuario dentro del software?				X	
11	¿Está satisfecho con el diseño, formas y colores del software?				X	
12	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser accedidos por otro?		X			
13	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser modificados por otro?		X			
14	¿Está satisfecho con cómo el sistema registra las acciones realizadas por los usuarios de forma tal que no puedan negar sus acciones indebidas u otras anomalías?			X		

Cuestionario

Recolección de datos de la calidad de software (pre test)

Fecha: 31/08/2022 **Rol en el proyecto:** Programador

Nombres y Apellidos: Alvaro Castro

El presente cuestionario correspondiente a la investigación denominada *Quality Assurance enfocado en Agile Testing para mejorar la Calidad de Software en la empresa Wasi Innovation, Lima – 2022*, por lo cual se requiere su participación con el fin de determinar el grado de satisfacción con respecto a la calidad del software en los proyectos que se realizan dentro de la empresa donde labora.

Instrucciones: Acorde a las preguntas que se presentan deberá marcar con una equis (X) una alternativa según corresponda.

Escala valorativa:

No satisfecho	Poco satisfecho	Moderadamente satisfecho	Satisfecho	Muy Satisfecho
1	2	3	4	5

N°	ITEMS	ESCALA				
		1	2	3	4	5
1	¿Está satisfecho con funcionalidades que brinda el software según los requerimientos establecidos?			X		
2	¿Está satisfecho con la precisión de los resultados que brinda el software?				X	
3	¿En qué grado está satisfecho en relación a las actividades de su día a día que puede realizar con el producto?				X	
4	¿Está satisfecho con el tiempo que demora realizar los procesos como las matriculas, reportes, emisión de certificados y otros?			X		
5	¿Está satisfecho con el software y los requisitos/recursos que se necesitan para que pueda usarlo?				X	
6	¿Está satisfecho con el funcionamiento del software con otros como el moodle?				X	
7	¿Está satisfecho con el intercambio de información entre software con el moodle y otros?				X	
8	¿Está satisfecho con la facilidad de aprendizaje del software?					X
9	¿Está satisfecho con la accesibilidad al software (tiene la capacidad de ser usado por personas con distintas características)?					X

10	¿Está satisfecho con la prevención de errores del usuario dentro del software?				X	
11	¿Está satisfecho con el diseño, formas y colores del software?					X
12	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser accedidos por otro?				X	
13	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser modificados por otro?				X	
14	¿Está satisfecho con cómo el sistema registra las acciones realizadas por los usuarios de forma tal que no puedan negar sus acciones indebidas u otras anomalías?				X	

Cuestionario

Recolección de datos de la calidad de software (pre test)

Fecha: 31/08/2022 **Rol en el proyecto:** Administrador de Base de datos

Nombres y Apellidos: Junior Condor

El presente cuestionario correspondiente a la investigación denominada *Quality Assurance enfocado en Agile Testing para mejorar la Calidad de Software en la empresa Wasi Innovation, Lima – 2022*, por lo cual se requiere su participación con el fin de determinar el grado de satisfacción con respecto a la calidad del software en los proyectos que se realizan dentro de la empresa donde labora.

Instrucciones: Acorde a las preguntas que se presentan deberá marcar con una equis (X) una alternativa según corresponda.

Escala valorativa:

No satisfecho	Poco satisfecho	Moderadamente satisfecho	Satisfecho	Muy Satisfecho
1	2	3	4	5

N°	ITEMS	ESCALA				
		1	2	3	4	5
1	¿Está satisfecho con funcionalidades que brinda el software según los requerimientos establecidos?		X			
2	¿Está satisfecho con la precisión de los resultados que brinda el software?			X		
3	¿En qué grado está satisfecho en relación a las actividades de su día a día que puede realizar con el producto?		X			
4	¿Está satisfecho con el tiempo que demora realizar los procesos como las matriculas, reportes, emisión de certificados y otros?		X			
5	¿Está satisfecho con el software y los requisitos/recursos que se necesitan para que pueda usarlo?			X		
6	¿Está satisfecho con el funcionamiento del software con otros como el moodle?		X			
7	¿Está satisfecho con el intercambio de información entre software con el moodle y otros?		X			
8	¿Está satisfecho con la facilidad de aprendizaje del software?			X		
9	¿Está satisfecho con la accesibilidad al software (tiene la capacidad de ser usado por personas con distintas características)?		X			

10	¿Está satisfecho con la prevención de errores del usuario dentro del software?		X			
11	¿Está satisfecho con el diseño, formas y colores del software?		X			
12	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser accedidos por otro?		X			
13	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser modificados por otro?		X			
14	¿Está satisfecho con cómo el sistema registra las acciones realizadas por los usuarios de forma tal que no puedan negar sus acciones indebidas u otras anomalías?			X		

Cuestionario

Recolección de datos de la calidad de software (post test)

Fecha: 28/10/2022 **Rol:** Gerente general

Nombres y Apellidos: Wiliam Castro

El presente cuestionario correspondiente a la investigación denominada *Quality Assurance enfocado en Agile Testing para mejorar la Calidad de Software en la empresa Wasi Innovation, Lima – 2022*, por lo cual se requiere su participación con el fin de determinar el grado de satisfacción con respecto a la calidad del software en los proyectos que se realizan dentro de la empresa donde labora.

Instrucciones: Acorde a las preguntas que se presentan deberá marcar con una equis (X) una alternativa según corresponda.

Escala valorativa:

No satisfecho	Poco satisfecho	Moderadamente satisfecho	Satisfecho	Muy Satisfecho
1	2	3	4	5

N°	ITEMS	ESCALA				
		1	2	3	4	5
1	¿Está satisfecho con funcionalidades que brinda el software según los requerimientos establecidos?			X		
2	¿Está satisfecho con la precisión de los resultados que brinda el software?			X		
3	¿En qué grado está satisfecho en relación a las actividades de su día a día que puede realizar con el producto?				X	
4	¿Está satisfecho con el tiempo que demora realizar los procesos como las matriculas, reportes, emisión de certificados y otros?			X		
5	¿Está satisfecho con el software y los requisitos/recursos que se necesitan para que pueda usarlo?			X		
6	¿Está satisfecho con el funcionamiento del software con otros como el moodle?			X		
7	¿Está satisfecho con el intercambio de información entre software con el moodle y otros?				X	
8	¿Está satisfecho con la facilidad de aprendizaje del software?			X		
9	¿Está satisfecho con la accesibilidad al software (tiene la capacidad de ser usado por personas con distintas características)?				X	

10	¿Está satisfecho con la prevención de errores del usuario dentro del software?				X	
11	¿Está satisfecho con el diseño, formas y colores del software?			X		
12	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser accedidos por otro?			X		
13	¿Está satisfecho con la seguridad de los datos que brinda el software para que no puedan ser modificados por otro?			X		
14	¿Está satisfecho con cómo el sistema registra las acciones realizadas por los usuarios de forma tal que no puedan negar sus acciones indebidas u otras anomalías?			X		

CARTA DE PRESENTACIÓN – JUICIO DE EXPERTOS

Huaraz, 26 de Setiembre del 2022

SEÑOR(A): _____

Yo, Bach. Jean Franklin Colonia Ballico, identificado con DNI N° 71848963, en mi estado de egresado de la Escuela Profesional de Ingeniería de Sistemas e Informática de la Universidad Nacional Santiago Antúnez de Mayolo, departamento de Ancash, Republica del Perú.

Respetado Juez, me dirijo a Ud. con la finalidad de solicitarle la validación del contenido de los ítems correspondientes al instrumento que utilizaré para la obtención de información que es requerida dentro de la investigación titulada: “QUALITY ASSURANCE ENFOCADO EN AGILE TESTING PARA MEJORAR LA CALIDAD DE SOFTWARE EN LA EMPRESA WASI INNOVATION, LIMA – 2022”, por tanto adjunto la documentación pertinente:

- Matriz de Operacionalización de Variables
- Matriz de Consistencia.
- Instrumento de medición documental (Entrevista y Cuestionario)
- Glosario de términos.

Debido a su amplia experiencia y los méritos académicos obtenidos, le ruego su apoyo en la validación del instrumento de medición documental en base a la matriz de juicio de experto.

Agradeciendo con anticipación su valiosa colaboración.

Atentamente.



Bach. Jean Franklin Colonia Ballico
DNI N° 71848963

VALIDACIÓN DE MATRIZ DE JUICIO DE EXPERTOS

Para la validación se tuvo en cuenta a los expertos en el área de la calidad de software, que mediante su experiencia y conocimiento obtenido nos ayudaron con la validación y observaciones sobre el instrumento de recolección de datos.

La validación se realizará mediante 4 categorías, las cuales cada una cuentan con una calificación en escala de 4 ítems y sus correspondientes descripciones. De acuerdo a lo mencionado anteriormente califique cada uno de los ítems según corresponda.

CATEGORÍA	CALIFICACIÓN	INDICADOR
<u>SUFICIENCIA</u> Los ítems que pertenecen a una misma dimensión bastan para obtener la medición de ésta.	1. No cumple con el criterio	Los ítems no son suficientes para medir la dimensión
	2. Bajo nivel	Los ítems miden algún aspecto de la dimensión pero no corresponden con la dimensión total
	3. Moderado Nivel	Se deben incrementar algunos ítems para poder evaluar la dimensión completamente.
	4. Alto Nivel	Los ítems son suficientes
<u>CLARIDAD</u> El ítem se comprende fácilmente, es decir, su sintáctica y semántica son adecuadas.	1. No cumple con el criterio	El ítem no es claro
	2. Bajo nivel	El ítem requiere bastantes modificaciones o una modificación muy grande en el uso de las palabras de acuerdo con su significado o por la ordenación de las mismas
	3. Moderado Nivel	Se requiere una modificación muy específica de algunos de los términos del ítem
	4. Alto Nivel	El ítem es claro, tiene semántica y sintaxis adecuada
<u>COHERENCIA</u> El ítem tiene relación lógica con la dimensión o indicador que está midiendo.	1. No cumple con el criterio	El ítem no tiene relación lógica con la dimensión
	2. Bajo nivel	El ítem tiene una relación tangencial con la dimensión
	3. Moderado Nivel	El ítem tiene una relación moderada con la dimensión que está midiendo
	4. Alto Nivel	El ítem se encuentra completamente relacionado con la dimensión que está midiendo
<u>RELEVANCIA</u> El ítem es esencial o importante, es decir debe ser incluido.	1. No cumple con el criterio	El ítem puede ser eliminado sin que se vea afectada la medición de la dimensión
	2. Bajo nivel	El ítem tiene alguna relevancia, pero otro ítem puede estar incluyendo lo que mide éste.
	3. Moderado Nivel	El ítem es relativamente importante.
	4. Alto Nivel	El ítem es muy relevante y debe ser incluido.



DATOS DEL JUEZ EXPERTO

Nombres y Apellidos	Cecilia Beatriz Briozzo
Formación Académica	Analista de Ciencias de la Computación. Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires, Argentina.
Áreas de Experiencia Laboral	Calidad de Software
Tiempo (total de años ejerciendo)	+20 años
Cargo Actual	QA Account Architect del clúster de servicios a Estados Unidos (USA).
Institución	Baufest
Firma	



MATRIZ DE VALORACIÓN – JUICIO EXPERTO

Este aporte será sumamente importante para la investigación denominada *Quality Assurance enfocado en Agile Tester para mejorar la Calidad de Software en la empresa Wasi Innovation, Lima – 2022:*

CUESTIONARIO

DIMENSIÓN	SUB DIMENSIÓN	INDICADOR	SUFICIENCIA				COHERENCIA				RELEVANCIA				CLARIDAD				OBSERVACIONES
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
ISO/IEC 25010	Adecuación Funcional	Compleitud funcional				X				X				X				X	
		Corrección funcional				X				X				X				X	
		Pertinencia funcional				X				X				X				X	Se puede mejorar la formulación: En qué grado está satisfecho en relación a las actividades de su día a día que puede realizar con el producto?
	Eficiencia de desempeño	Comportamiento temporal				X				X				X				X	
		Utilización de recursos				X				X				X				X	
	Compatibilidad	Coexistencia				X				X				X				X	
		Interoperabilidad				X				X				X				X	
	Usabilidad	Aprendizaje				X				X				X				X	Se puede reformular, por ejemplo: ¿Está satisfecho con la facilidad de aprendizaje del software?
		Accesibilidad			X					X				X				X	
		Protección frente a errores de usuario				X				X				X				X	
		Estética				X				X				X				X	
	Seguridad	Confidencialidad				X				X				X				X	
		Integridad				X				X				X				X	
		No repudio				X			X					X					Sugiero reformular la pregunta: por ejemplo: ¿Está satisfecho con cómo el sistema registra las acciones realizadas por los usuarios de forma tal que no puedan negar sus acciones?





DATOS DEL JUEZ EXPERTO

Nombres y Apellidos	Rosa Mercedes Julca Pérez
Formación Académica	Lic. Investigación Operativa- Ingeniería de Sistemas e Informática
Áreas de Experiencia Laboral	Calidad de Software
Tiempo (total de años ejerciendo)	+10 años
Cargo Actual	QA&Tester P3
Institución	Baufest
Firma	



MATRIZ DE VALORACIÓN – JUICIO EXPERTO

Este aporte será sumamente importante para la investigación denominada *Quality Assurance enfocado en Agile Tester para mejorar la Calidad de Software en la empresa Wasi Innovation, Lima – 2022:*


CUESTIONARIO

DIMENSIÓN	SUB DIMENSIÓN	INDICADOR	SUFICIENCIA				COHERENCIA				RELEVANCIA				CLARIDAD				OBSERVACIONES
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
ISO/IEC 25010	Adecuación Funcional	Complejidad funcional				X				X				X				X	
		Corrección funcional				X				X				X				X	
		Pertinencia funcional				X				X				X				X	
	Eficiencia de desempeño	Comportamiento temporal				X				X				X				X	
		Utilización de recursos				X				X				X				X	
	Compatibilidad	Coexistencia				X				X				X				X	
		Interoperabilidad				X				X				X				X	
	Usabilidad	Aprendizaje				X				X				X				X	
		Accesibilidad				X				X				X				X	
		Protección frente a errores de usuario				X				X				X				X	
		Estética				X				X				X				X	
	Seguridad	Confidencialidad				X				X			X				X		
		Integridad				X				X			X				X		
		No repudio				X				X			X				X		





DATOS DEL JUEZ EXPERTO

Nombres y Apellidos	Lidia Zapata Tixi
Formación Académica	Ing. Sistemas e Informática
Áreas de Experiencia Laboral	Calidad de software, Metodología Scrum
Tiempo (total de años ejerciendo)	6 años de experiencia
Cargo Actual	Analista QA-Accenture Quality
Institución	Empresa Baufest
Firma	



MATRIZ DE VALORACIÓN – JUICIO EXPERTO

Este aporte será sumamente importante para la investigación denominada *Quality Assurance enfocado en Agile Tester para mejorar la Calidad de Software en la empresa Wasi Innovation, Lima – 2022:*

CUESTIONARIO

DIMENSIÓN	SUB DIMENSIÓN	INDICADOR	SUFICIENCIA				COHERENCIA				RELEVANCIA				CLARIDAD				OBSERVACIONES
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
ISO/IEC 25010	Adecuación Funcional	Compleitud funcional				X				X				X				X	
		Corrección funcional				X				X				X				X	
		Pertinencia funcional				X				X				X				X	
	Eficiencia de desempeño	Comportamiento temporal				X				X				X				X	
		Utilización de recursos				X				X				X				X	
	Compatibilidad	Coexistencia				X				X				X				X	
		Interoperabilidad				X				X				X				X	
	Usabilidad	Aprendizaje				X				X				X				X	
		Accesibilidad				X				X				X				X	
		Protección frente a errores de usuario				X				X				X				X	
		Estética				X				X				X				X	
	Seguridad	Confidencialidad				X				X				X				X	
		Integridad				X				X				X				X	
No repudio					X				X				X				X		





DATOS DEL JUEZ EXPERTO

Nombres y Apellidos	Lucero Carrera Real
Formación Académica	Ing. en Sistemas Computacionales
Áreas de Experiencia Laboral	Business Analyst Quality assurance Liderazgo de equipo
Tiempo (total de años ejerciendo)	+10 años
Cargo Actual	QA Architect
Institución	Baufest
Firma	



UNIVERSIDAD NACIONAL SANTIAGO ANTÚNEZ DE MAYOLO
FACULTAD DE CIENCIAS
Escuela Profesional de Ingeniería de Sistemas e Informática



MATRIZ DE VALORACIÓN – JUICIO EXPERTO

Este aporte será sumamente importante para la investigación denominada *Quality Assurance enfocado en Agile Tester para mejorar la Calidad de Software en la empresa Wasi Innovation, Lima – 2022:*

DIMENSIÓN	SUB DIMENSIÓN	INDICADOR	SUFICIENCIA				COHERENCIA				RELEVANCIA				CLARIDAD				OBSERVACIONES
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
ISO/IEC 25010	Adecuación Funcional	Compleitud funcional				X				X				X				X	
		Corrección funcional				X				X				X				X	
		Pertinencia funcional				X				X				X				X	
	Eficiencia de desempeño	Comportamiento temporal				X				X				X				X	
		Utilización de recursos				X				X				X				X	
	Compatibilidad	Coexistencia				X				X				X				X	
		Interoperabilidad				X				X				X				X	
	Usabilidad	Aprendizaje				X				X				X				X	
		Accesibilidad				X				X				X				X	
		Protección frente a errores de usuario				X				X				X				X	
		Estética				X				X				X				X	
	Seguridad	Confidencialidad				X				X				X				X	
		Integridad				X				X				X				X	
No repudio					X				X				X				X		



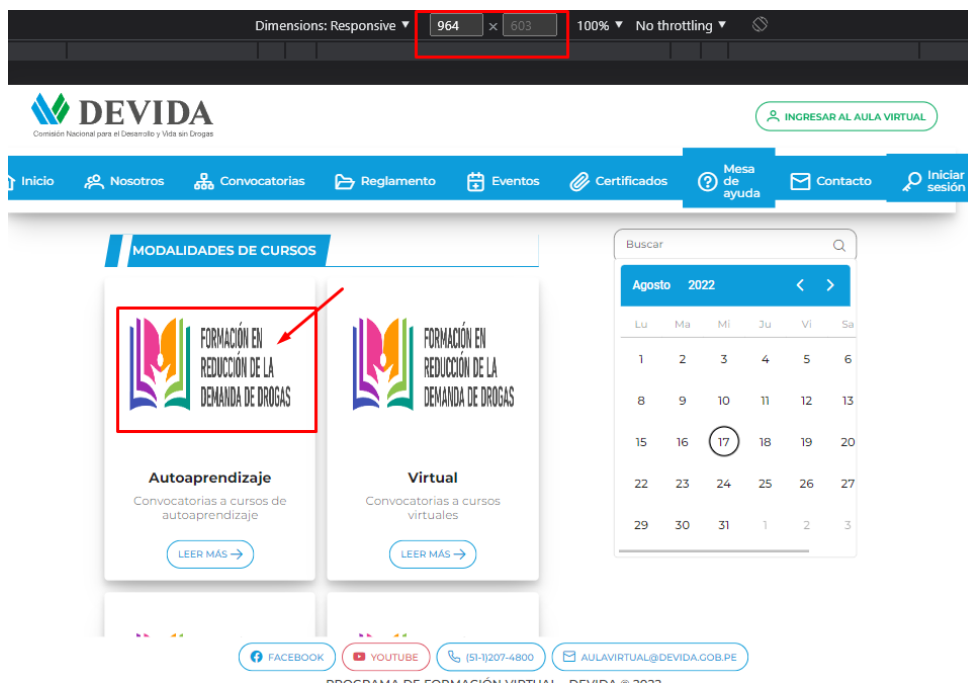
PAGES

INICIO

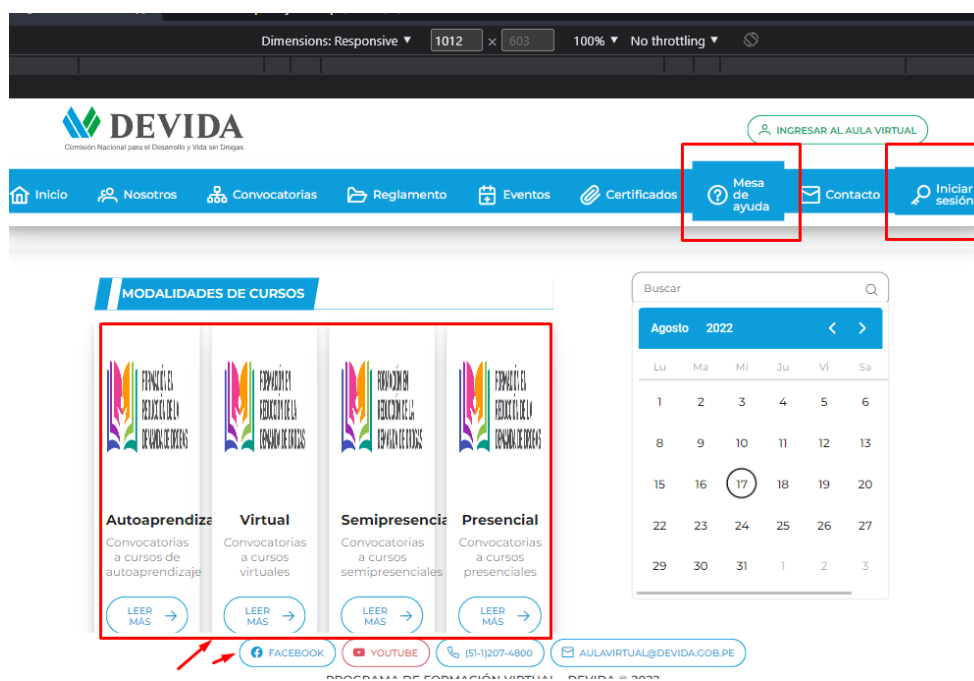
Responsive sizes 770px +



Responsive sizes 880px +



Responsive sizes 990px +



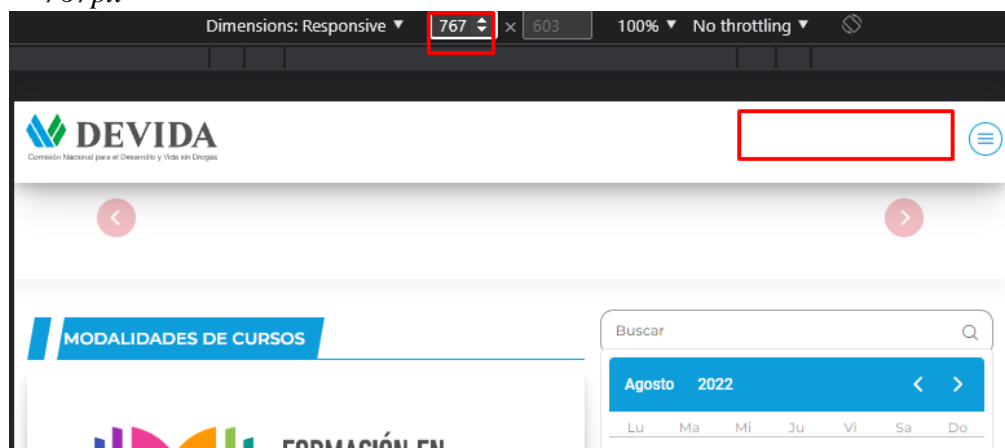
Header de todas las pestañas - size 769px - 960px



Nota: el botón "Ingresar al aula virtual desaparece desde 767px - (de forma descendente)



767px -



300px -



LOGIN

El tamaño de los contenedores de los botones se incrementa mucho más que los demás en el rango de 961px - 1112px (corroborar si es normal)

Responsive Size 961px



Responsive Size 1110px



CERTIFICACIONES

Home

Responsive size 399px –

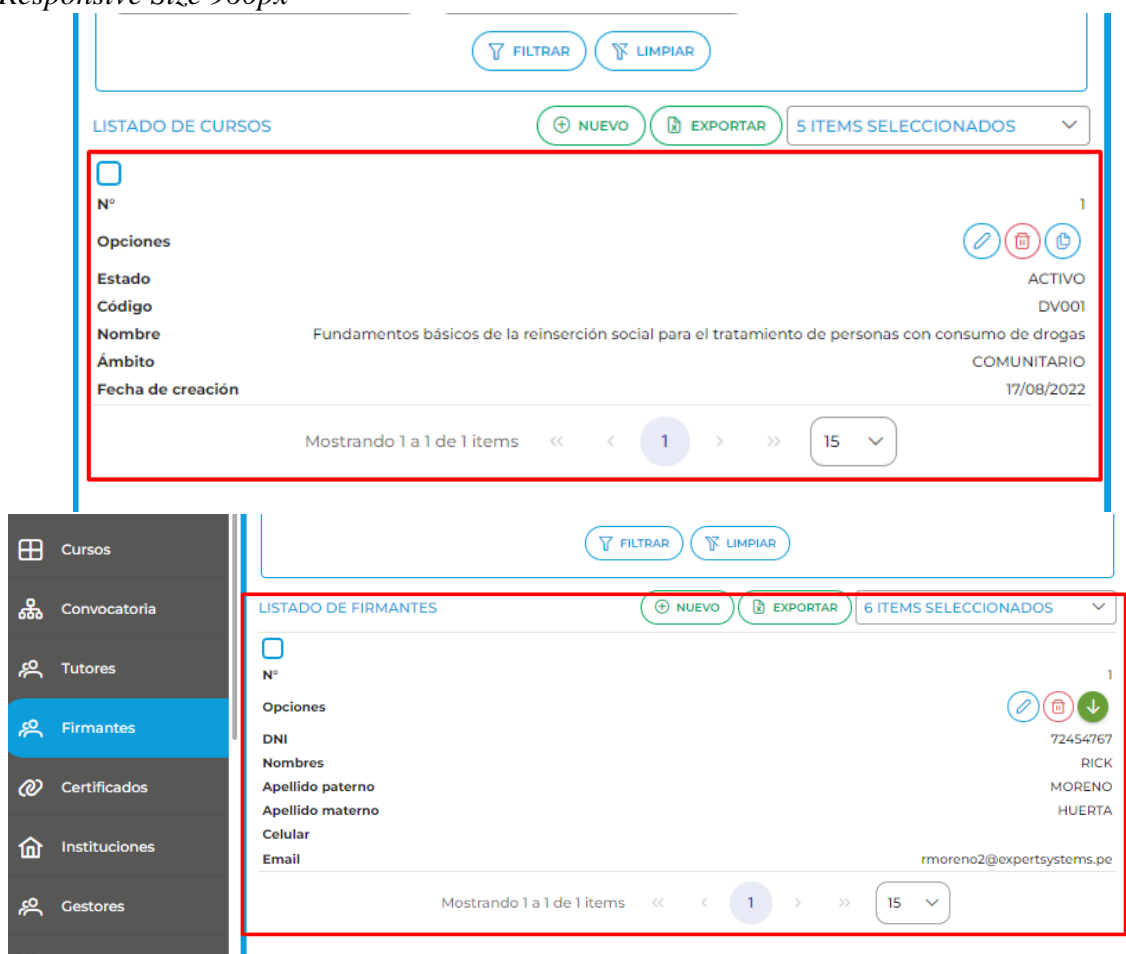
Omitir el error si es que no se tiene como requerimiento entrada con vista mobile (para todos los apartados como home, curso, convocatorias, tutores, firmantes, certificaciones, instituciones, gestores, parámetros, reportes, inscripciones, matrícula, otros más).



En todos los apartados se muestra que de 961px – (en forma descendente) no se tiene configurado para la visualización de las tablas y el orden cambia a columnas según los ítems de los apartados.



Responsive Size 960px



- Inconvenientes al usar los filtros de buscar y día en el calendario en apartado de Home:

MODALIDADES DE CURSOS

Autoaprendizaje
Convocatorias a cursos de autoaprendizaje

LEER MÁS →

Virtual
Convocatorias a cursos virtuales

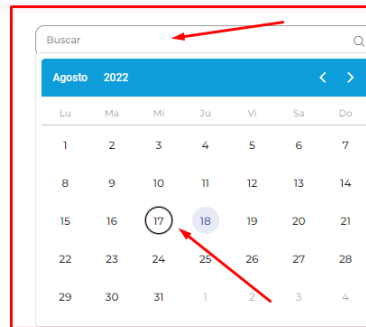
LEER MÁS →

Semipresencial
Convocatorias a cursos semipresenciales

LEER MÁS →

Presencial
Convocatorias a cursos presenciales

LEER MÁS →



Convocatorias

- Al seleccionar las aulas de la convocatoria, hay un botón para crear una nueva aula no es intuitivo al momento de hacer la selección de una sección:

Selecciono nuevo:

ESTADO DE CONVOCATORIAS [NUEVO] [EXPORTAR] 6 ITEMS SELECCIONADOS

AULAS CONVOCATORIA

AULAS Y TUTORES

LISTADO DE TUTORES ASIGNADOS (10 SECCIONES)

[NUEVO] 5 ITEMS SELECCIONADOS

N°	OPCIONES	SECCIÓN	DNI	NOMBRE COMPLETO	CELULAR	EMAIL
Mostrando 0 a 0 de 0 items						

[CANCELAR]

Selecciono Tutor

SELECCIONAR TUTOR

LISTADO DE TUTORES [EXPORTAR] 6 ITEMS SELECCIONADOS

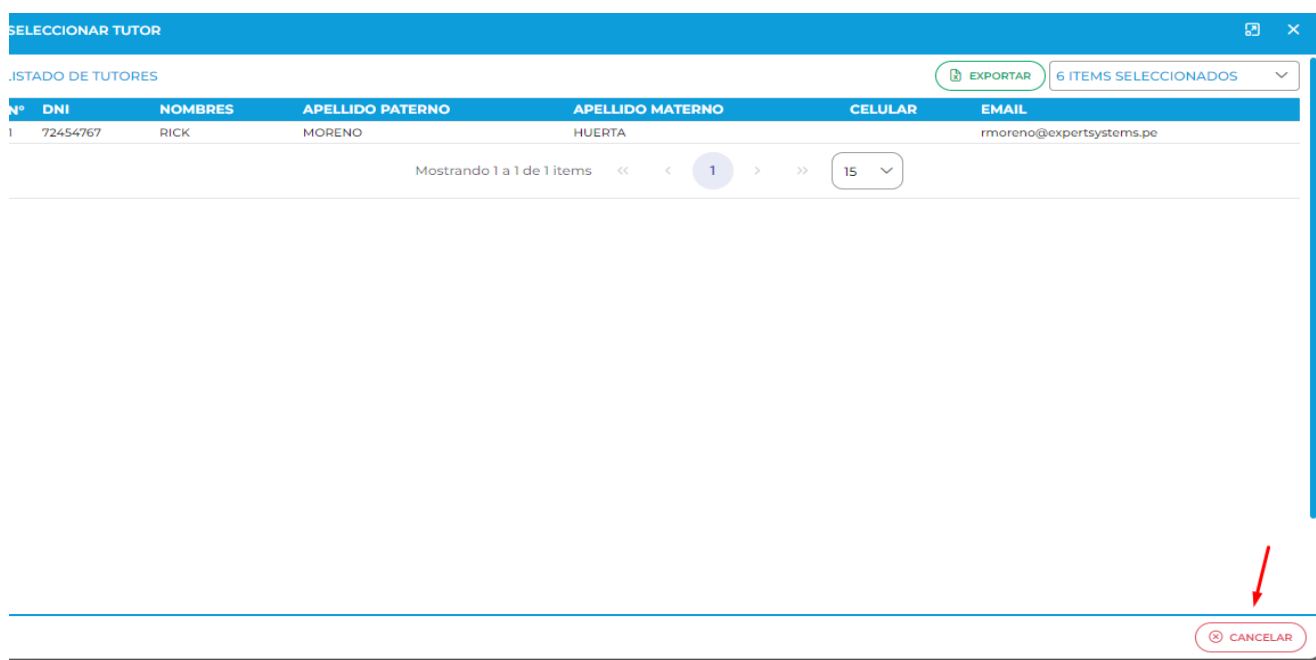
N°	DNI	NOMBRES	APELLIDO PATERNO	APELLIDO MATERNO	CELULAR	EMAIL
1	72454767	RICK	MORENO	HUERTA		rmoreno@expertsystems.pe

Mostrando 1 a 1 de 1 items

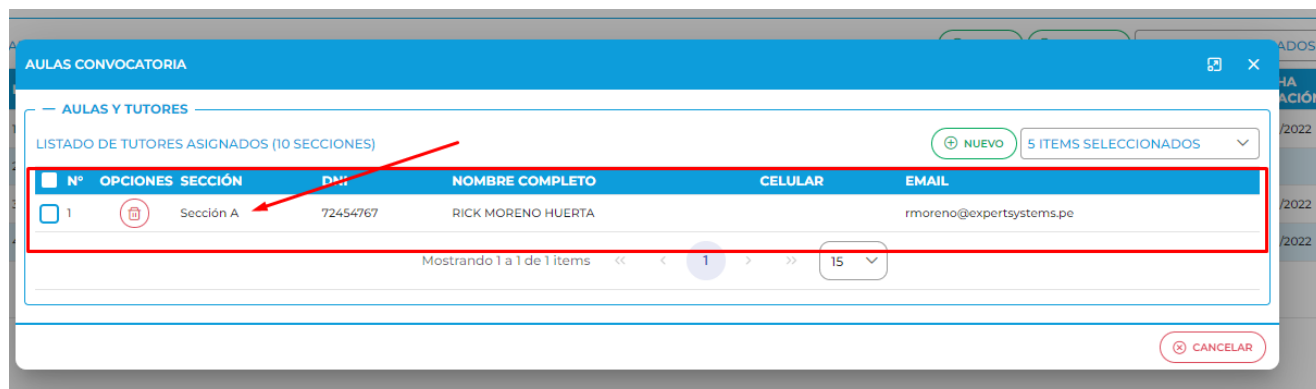
Selecciono Sección



No se muestra ningún mensaje de información que se ha realizado la acción correspondiente para la sección (obligatoriamente tenemos que presionar el “Botón Cancelar” que es contradictorio a la acción que estamos realizando).



Después de poner “Cancelar” muestra el modal pero ya con la sección creada y tutor asignado:



- Al modificar o crear una convocatoria no se está teniendo en cuenta los valores límites lógicos (consultar valores límites según campos).

MODIFICAR CONVOCATORIA

INFORMACIÓN GENERAL

ASIGNAR CURSO

(*) Código curso: DV001 (*) Nombre curso: Fundamentos básicos de la reinserción social para el tratamiento de personas con consu (*) Ámbito curso: COMUNITARIO

(*) Número aulas: 26 (*) Número vacantes por aula: 2,032,132,132 (*) Número sesiones: 999,999,999,999 (*) Modalidad: Virtual

(*) Año: 2022 (*) Bloque: (*) Gestor asignado: RICARDO MENDOZA JIMENEZ (*) Código: DV001 - 5 - 2022

(*) Descripción: DV001 - 5 - 2022 Fundamentos básicos de la reinserción social para el tratamiento de personas con consumo de drogas

GENERAR ENLACE PARA COMPARTIR

CRONOGRAMA

Asignar cronograma automáticamente [VER CONFIGURACIÓN](#)

(*) Fecha publicación convocatoria: 19/08/2022 (*) Fecha inicio inscripción: 26/08/2022 (*) Fecha fin inscripción: 25/09/2022 (*) Fecha inicio evaluación: 26/09/2022

- No está permitiendo eliminar una convocatoria en ningún estado (ya sea registrada, inscripción, publicado, etc).

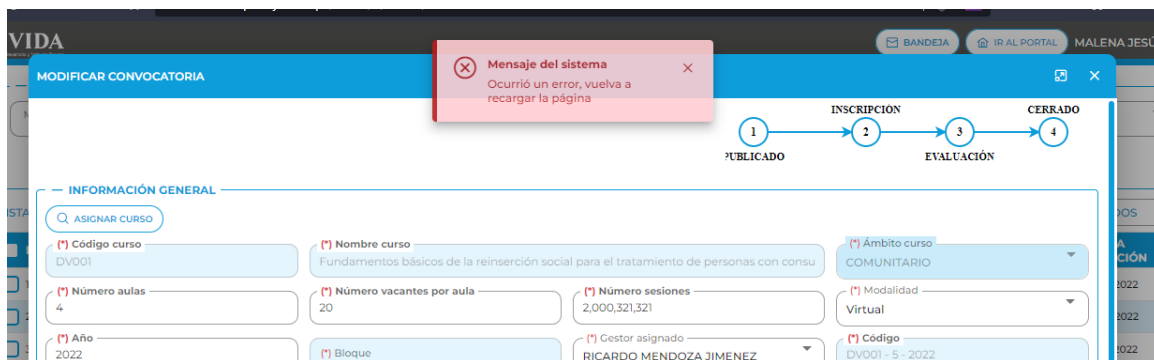
ESTADO DE CONVOCATORIAS NUEVO EXPORTAR 6 ITEMS SELECCIONADOS

Nº	OPCIONES	ESTADO	ETAPA	DESCRIPCIÓN	MODALIDAD	ÁMBITO	FECHA CREACIÓN
1		ABIERTA	REGISTRADO	DV001 - 5 - 2022 Fundamentos básicos de la reinserción social para el tratamiento de personas con consumo de drogas	Virtual	COMUNITARIO	19/08/2022
2		ABIERTA	INSCRIPCIÓN	DV011 - 1 - 2022 Técnicas y Herramientas para el Facilitador Comunitario en Prevención del Consumo de Drogas	Virtual	SALUD	18/08/2022
3		ABIERTA	INSCRIPCIÓN	DV009 - - 2022 Líderes Comunitarios para la Prevención del Consumo de Drogas	Presencial	EDUCACIÓN	17/08/2022
4		ABIERTA	PUBLICADO	DV008 - 1 - 2022 Fundamentos para la Prevención Selectiva en la Reducción de la Demanda de Drogas	Autoaprendizaje	SALUD	17/08/2022
5		ABIERTA	INSCRIPCIÓN	DV007 - 2 - 2022 Habilidades Socioemocionales y su Relación con el Consumo de Drogas en el Ámbito Educativo	Semipresencial	SALUD	17/08/2022
6		ABIERTA	INSCRIPCIÓN	DV007 - 1 - 2022 Habilidades Socioemocionales y su Relación con el Consumo de Drogas en el Ámbito Educativo	Virtual	SALUD	17/08/2022

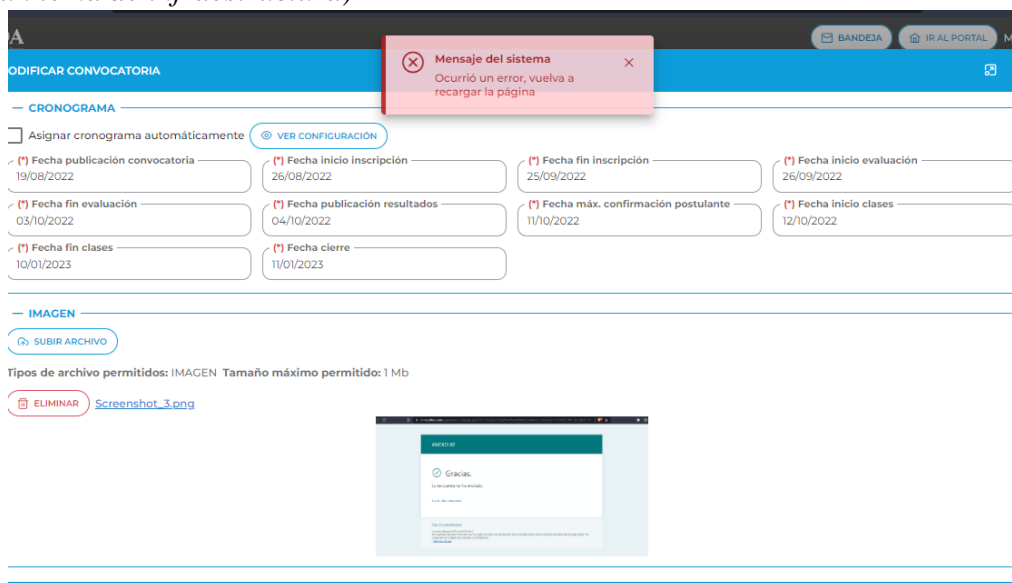
LISTADO DE CONVOCATORIAS NUEVO EXPORTAR 6 ITEMS SELECCIONADOS

Nº	OPCIONES	ESTADO	ETAPA	DESCRIPCIÓN	MODALIDAD	ÁMBITO	FECHA CREACIÓN
1		ABIERTA	REGISTRADO	DV001 - 5 - 2022 Fundamentos básicos de la reinserción social para el tratamiento de personas con consumo de drogas	Virtual	COMUNITARIO	19/08/2022
2		ABIERTA	INSCRIPCIÓN	DV011 - 1 - 2022 Técnicas y Herramientas para el Facilitador Comunitario en Prevención del Consumo de Drogas	Virtual	SALUD	18/08/2022
3		ABIERTA	INSCRIPCIÓN	DV009 - - 2022 Líderes Comunitarios para la Prevención del Consumo de Drogas	Presencial	EDUCACIÓN	17/08/2022
4		ABIERTA	PUBLICADO	DV008 - 1 - 2022 Fundamentos para la Prevención Selectiva en la Reducción de la Demanda de Drogas	Autoaprendizaje	SALUD	17/08/2022
5		ABIERTA	INSCRIPCIÓN	DV007 - 2 - 2022 Habilidades Socioemocionales y su Relación con el Consumo de Drogas en el Ámbito Educativo	Semipresencial	SALUD	17/08/2022
6		ABIERTA	INSCRIPCIÓN	DV007 - 1 - 2022 Habilidades Socioemocionales y su Relación con el Consumo de Drogas en el Ámbito Educativo	Virtual	SALUD	17/08/2022

- No está permitiendo editar la convocatoria (Ignorar si es un tema de infraestructura).



- *Se visualiza la imagen al momento de cargar pero no se puede guardar (ignorar si es un tema de infraestructura)*



Tutores

- *Modificar información del Tutor, el campo Fecha de nacimiento está permitiendo ingresar valores ilógicos.*

Acorde a la complejidad de la validación de este tipo de valores, tenerlo en cuenta.

MODIFICAR TUTOR

INFORMACIÓN GENERAL

(*) Tipo de documento: Carnet de extranjería
 (*) Número de documento: []
 (*) Nombres: CARMEN
 (*) Apellido paterno: LOPEZ
 (*) Apellido materno: ALEJO
 Dirección: abc 123
 (*) Email: tettiquimmeice-1521@yopmail.com

INFORMACIÓN ADICIONAL

RUC: [] Género: [] Fecha de nacimiento: 19/08/2022 Celular: []
 Teléfono fijo: []

GUARDAR CANCELAR

13:08 18/08/2022

Siguiendo el flujo de modificación para el tutor, no se observa un mensaje de información acorde a la acción realizada.

CRITERIOS DE BÚSQUEDA

Estado: []

FILTRAR LIMPIAR

Sin mensaje de información después de realizar la modificación para el tutor

NUEVO EXPORTAR 7 ITEMS SELECCIONADOS

LISTADO DE TUTORES

N°	OPCIONES	TIPO DOC.	N° DOC.	NOMBRES	APELLIDO PATERNO	APELLIDO MATERNO	CELULAR	EMAIL
1	[] [] [] []	Carnet de extranjería		CARMEN	LOPEZ	ALEJO		tettiquimmeice-1521@yopmail.com
2	[] [] [] []	Carnet de extranjería	46789875456	ERNESTO	DODINO	FLORENTINO		zawutrukepra-5564@yopmail.com
3	[] [] [] []	DNI		RICK	MORENO	HUERTA		rmoreno@expertsystems.pe

Mostrando 1 a 3 de 3 items

(El mismo caso de validación se percibe para los Firmantes)

- Observación sobre los colores, se está trabajando con el color verde porque está activo pero "se debe tener en cuenta que el icono no representa el modo activo sino la acción (por el sentido de la flecha) que se realizará para desactivar al firmante". Se puede trabajar en sentido contrario con respecto a colores, si deseo desactivar con un color rojo y si se desea activar un color verde (sugerencia).

LISTADO DE FIRMANTES

N°	OPCIONES	TIPO DOC.	N° DOC.	NOMBRES
1	[] [] [] []	Carnet de extranjería		PEDRO
2	[] [] [] []	Carnet de extranjería	546353435363	OMAR
3	[] [] [] []	DNI	72454767	RICK

Mostrando 1 a 3

LISTADO DE FIRMANTES

N°	OPCIONES	TIPO DOC.	N° DOC.	NOMBRES
1	[] [] [] []	Carnet de extranjería		PEDRO
2	[] [] [] []	Listado de firmantes	546353435363	OMAR
3	[] [] [] []	DNI	72454767	RICK

Mostrando 1 a 3

Gestores

- Al momento de realizar la modificación de información del Gestor, no se visualiza el número de identificación en el modal, tampoco en la tabla de Gestores

STADO DE GESTORES

NUEVO EXPORTAR 7 ITEMS SELECCIONADOS

N°	OPCIONES	TIPO DOC.	N° DOC.	NOMBRES	APELLIDO PATERNO	APELLIDO MATERNO	CELULAR	EMAIL
1		DNI		RICK	MORENO	HUERTA		rmoreno@expertsystems.pe

Mostrando 1 a 1 de 1 items << < 1 > >> 15

Instituciones

- filtro en los demás apartados está funcionando de la forma en que si solo seleccionas un valor dentro del combo te filtra, independientemente si se esté llenando o no valores en los demás combos pero en este apartado aún no se ha implementado.

CRITERIOS DE BÚSQUEDA

Departamento Provincia Distrito Nombre

FILTRAR LIMPIAR

LISTADO DE INSTITUCIONES

NUEVO EXPORTAR 5 ITEMS SELECCIONADOS

N°	OPCIONES	DEPARTAMENTO	PROVINCIA	DISTRITO	NOMBRE	DIRECCIÓN
1		Huánuco	Huánuco	Pillo Marca	DEMUNA	S/N
2		Puno	Huancané	Huancane	Poder Judicial	S/N
3		La Libertad	Trujillo	Trujillo	Ciro Alegría	S/N
4		Lima	Lima	Lima	Instituto Arzobispo Loayza	S/N
5		Ayacucho	Huanta	Canayre	DEVIDA	S/N
6		La Libertad	Chepén	Pacanga	DIRESA	S/N
7		Lima	Lima	Lima	ESSALUD	S/N

Mostrando 1 a 7 de 7 items << < 1 > >> 15

Usando un combo del filtro:

CRITERIOS DE BÚSQUEDA

Departamento: Provincia: Distrito: Nombre:

LISTADO DE INSTITUCIONES 5 ITEMS SELECCIONADOS

N°	OPCIONES	DEPARTAMENTO	PROVINCIA	DISTRITO	NOMBRE	DIRECCIÓN
Mostrando 0 a 0 de 0 items << < > >> 15						

Usando dos combos a más del filtro:

CRITERIOS DE BÚSQUEDA

Departamento: Provincia: Distrito: Nombre:

LISTADO DE INSTITUCIONES 5 ITEMS SELECCIONADOS

N°	OPCIONES	DEPARTAMENTO	PROVINCIA	DISTRITO	NOMBRE	DIRECCIÓN
Mostrando 0 a 0 de 0 items << < > >> 15						

CRITERIOS DE BÚSQUEDA

Departamento: Provincia: Distrito: Nombre:

LISTADO DE INSTITUCIONES 5 ITEMS SELECCIONADOS

N°	OPCIONES	DEPARTAMENTO	PROVINCIA	DISTRITO	NOMBRE	DIRECCIÓN
Mostrando 0 a 0 de 0 items << < > >> 15						

Firmantes

- El reporte no está saliendo acorde al formato especificado (se habló este tema en el daily – viernes 19/08/22)

ARCHIVO INICIO INSERTAR DISEÑO DE PÁGINA FÓRMULAS DATOS REVISAR VISTA ACROBAT Cuenta Microsoft

VISTA PROTEGIDA Cuidado—los archivos de Internet pueden contener virus. Si no tiene que editarlo, es mejor que siga en Vista protegida.

TIPO DOC.	N° DOC.	NOMBRE: APELLIDO PATERNO	APELLIDO MATERNO	CELULAR	EMAIL
Carnet de extra		PEDRO	MAGUIÑA	BUEN	987654321 xauroizutreumo-2968+f@yopmail.com
Carnet de extra	546353435363	OMAR	TIBURCIO	GOMEZ	zagrocaukaubroi-3810+f@yopmail.com
DNI	72454767	RICK	MORENO	HUERTA	rmoreno+f@expertsystems.pe

- Al modificar la información del Firmante en campo Género no está replicando en la tabla, también tener en cuenta la validación que se ha sugerido para la fecha de nacimiento y los valores límites permitidos.

MODIFICAR FIRMANTE

INFORMACIÓN GENERAL

(*) Tipo de documento: DNI
 (*) Número de documento: []
 (*) Nombres: RICK
 (*) Apellido paterno: MORENO

(*) Apellido materno: HUERTA
 Dirección: ANCASH/HUARI/HUARI - HUARI
 (*) Email: rmoreno@expertsystems.pe

INFORMACIÓN ADICIONAL

RUC: []
 Género: MASCULINO
 Fecha de nacimiento: 19/08/2022
 Celular: 987456987

Teléfono fijo: []
 Cargo: TEST

GUARDAR CANCELAR

Resultado después de la modificación (no replican datos como el género):

LISTADO DE FIRMANTES

NUEVO EXPORTAR 11 ITEMS SELECCIONADOS

N°	OPCIONES	TIPO DOC.	N° DOC.	NOMBRES	APELLIDO PATERNO	APELLIDO MATERNO	DIRECCIÓN	UBIGEO	GÉNERO	FECHA NACIMIENTO	CELULAR	EMAIL
1	[] [] [] []	Carnet de extranjería	546353435363	OMAR	TIBURCIO	GOMEZ						zagrocaukaubroi-3810@yopmail.com
2	[] [] [] []	DNI		RICK	MORENO	HUERTA	ANCASH/HUARI/HUARI - HUARI			19/08/2022	987456987	rmoreno@expertsystems.pe

Mostrando 1 a 2 de 2 items

Firmantes

- El filtro de búsqueda no está funcionando acorde a los valores que se ingresan en los combos.

CRITERIOS DE BÚSQUEDA

(*) Año: [] (*) Convocatoria: []
 Sección: []

FILTRAR LIMPIAR

VER MODELO CERTIFICADO GENERAR CERTIFICADOS GENERAR CON PLANTILLA

LISTADO DE CERTIFICADO

EXPORTAR SECCIÓN, DNI, APELLIDOS ...

N°	OPCIONES	SECCIÓN	DNI	APELLIDOS NOMBRES
1	[] [] [] [] [] []	AULA 1	12345678	FACTOR HUERTA OSTERLING
2	[] [] [] [] [] []	AULA 1	12345678	FACTOR HUERTA OSTERLING

Mostrando 1 a 2 de 2 items

Ingreso de valores a combos:

CRITERIOS DE BÚSQUEDA

(*) Año: 2022

(*) Convocatoria: DV011 - 1 - 2022 Técnicas y Herramientas para el Facilitador Comunitario en Prevención del Consumo de Drogas

Sección:

FILTRAR LIMPIAR

VER MODELO CERTIFICADO GENERAR CERTIFICADOS GENERAR CON PLANTILLA

LISTADO DE CERTIFICADO

EXPORTAR SECCIÓN, DNI, APELLIDOS ...

N°	OPCIONES	SECCIÓN	DNI	APELLIDOS NOMBRES
Mostrando 0 a 0 de 0 items				

Después de realizar el filtro limpias los valores del combo ya no se muestra nada en la tabla:

CRITERIOS DE BÚSQUEDA

(*) Año

(*) Convocatoria

Sección

FILTRAR LIMPIAR

VER MODELO CERTIFICADO GENERAR CERTIFICADOS GENERAR CON PLANTILLA

LISTADO DE CERTIFICADO

EXPORTAR SECCIÓN, DNI, APELLIDOS ...

N°	OPCIONES	SECCIÓN	DNI	APELLIDOS NOMBRES
Mostrando 0 a 0 de 0 items				

No se encontraron datos.

- Exportar no implementado

FILTRAR LIMPIAR

VER MODELO CERTIFICADO GENERAR CERTIFICADOS GENERAR CON PLANTILLA

LISTADO DE CERTIFICADO

EXPORTAR SECCIÓN, DNI, APELLIDOS ...

N°	OPCIONES	SECCIÓN	DNI	APELLIDOS NOMBRES
1		AULA 1	12345678	FACTOR HUERTA OSTERLING
2		AULA 1	12345678	FACTOR HUERTA OSTERLING

Mostrando 1 a 2 de 2 items

Cursos

Observación ante error de usuario

- Filtro para visualización de las columnas con los valores del curso cuando se selecciona uno o vario funciona con normalidad pero cuando se selecciona todo y luego se quita esa selección queda en blanco. Evaluar la funcionalidad de quitar la selección de todos los ítems cuando se tiene un response de gran longitud de data.

Filtrado con todos los items

9 ITEMS SELECCIONADOS

	ÁMBITO	HORAS LECTIVAS	NORA APROBATORIA	PORCENTAJE CALIFICACIÓN CONTÍNUA	<input checked="" type="checkbox"/>
Modelos de Participación Comunitaria para la Prevención del Consumo de Drogas	COMUNITARIO	230	14	60	<input checked="" type="checkbox"/>
Técnicas y Herramientas para el Facilitador Comunitario en Prevención del Consumo de Drogas	SALUD	230	14	60	<input checked="" type="checkbox"/>
Formación pública local para la prevención de drogas legales e ilegales en el ámbito educativo	EDUCACIÓN	230	14	60	<input checked="" type="checkbox"/>
Modelos para la Prevención del Consumo de Drogas	EDUCACIÓN	230	14	60	<input checked="" type="checkbox"/>
Prevención Selectiva en la Prevención de Drogas	SALUD	230	14	60	<input checked="" type="checkbox"/>
Modelos de Participación Comunitaria y su Relación con el Consumo de Drogas en el Ámbito Educativo	SALUD	230	14	60	<input checked="" type="checkbox"/>

ESTADO
 CÓDIGO
 NOMBRE
 HORAS LECTIVAS

Filtrado cuando se quita todos los items

LISTADO DE CURSOS

<input type="checkbox"/>	N°	OPCIONES	<input type="checkbox"/>
<input type="checkbox"/>	1		<input type="checkbox"/>
<input type="checkbox"/>	2		<input type="checkbox"/>
<input type="checkbox"/>	3		<input type="checkbox"/>
<input type="checkbox"/>	4		<input type="checkbox"/>
<input type="checkbox"/>	5		<input type="checkbox"/>
<input type="checkbox"/>	6		<input type="checkbox"/>
<input type="checkbox"/>	7		<input type="checkbox"/>
<input type="checkbox"/>	8		<input type="checkbox"/>
<input type="checkbox"/>	9		<input type="checkbox"/>
<input type="checkbox"/>	10		<input type="checkbox"/>
<input type="checkbox"/>	11		<input type="checkbox"/>

ESTADO
 CÓDIGO
 NOMBRE
 ÁMBITO
 HORAS LECTIVAS

- **Consulta**, si en el apartado de Cursos se tiene el estado y en la visualización de la tabla también, por qué al momento de editar la información de un curso no se tiene esa opción de desactivar o activar...

CRITERIOS DE BÚSQUEDA

<input type="checkbox"/>	N°	OPCIONES	ESTADO	CÓDIGO	NOMBRE	ÁMBITO	FECHA DE CREACIÓN
<input type="checkbox"/>	1		ACTIVO	DV012	Modelos de Participación Comunitaria para la Prevención del Consumo de Drogas	COMUNITARIO	17/08/2022
<input type="checkbox"/>	2		ACTIVO	DV011	Técnicas y Herramientas para el Facilitador Comunitario en Prevención del Consumo de Drogas	SALUD	17/08/2022

Modal de modificación del curso (sin la observación descrita).

MODIFICAR CURSO

INFORMACIÓN GENERAL

Código: DV012

Nombre: Modelos de Participación Comunitaria para la Prevención del Consumo de Drogas

Ámbito: COMUNITARIO

Descripción: La Política Nacional contra las Drogas al 2030, plantea como Objetivo Prioritario 3 (OP3): Reducir el consumo de drogas en poblaciones en situación de vulnerabilidad. Este objetivo está orientado a disminuir el consumo de drogas a través de la prevención del consumo de drogas a toda la población, con especial atención a las poblaciones en situación de vulnerabilidad que considera en principio, a la población en edad escolar. Además, del incremento de la oferta de servicios de tratamiento de adicción a drogas en toda la población en general, con especial atención a poblaciones en situación de vulnerabilidad

Contenido: Crea estrategias eficaces para promover la participación en los programas de prevención del consumo de drogas en entornos comunitarios, a través del uso de técnicas y herramientas pertinentes. Contenido: Unidad 01: Fundamentos y principios del facilitador Unidad 02: Aprendizaje y técnicas para incentivar la participación Unidad 03: Técnicas y estrategias comunitarias para impulsar la participación grupal Unidad 04: Aprendizaje cooperativo en comunidad

- La eliminación de curso no está funcionando (omitir si es un error de infraestructura), por lo general debería eliminarse o actualizar el flag lógico de dicha eliminación, ya que no se tiene dependencia (este curso no se ha seleccionado en ninguna de las convocatorias existentes y debería realizarse la acción con normalidad).
SE REALIZÓ LA PRUEBA MEDIANTE UN CURSO COPIADO Y UNO NUEVO.

N°	OPCIONES	ESTADO	CÓDIGO	NOMBRE	ÁMBITO	FECHA DE CREACIÓN
1		ACTIVO	DV012	Modelos de Participación Comunitaria para la Prevención del Consumo de Drogas	COMUNITARIO	17/08/2022
2		ACTIVO	DV011	Técnicas y Herramientas para el Facilitador Comunitario en Prevención del Consumo de Drogas	SALUD	17/08/2022
3		ACTIVO	DV010	Estrategias de gestión pública local para la prevención del consumo de drogas legales e ilegales en el ámbito comunitario	EDUCACIÓN	17/08/2022
4		ACTIVO	DV009	Lideres Comunitarios para la Prevención del Consumo de Drogas	EDUCACIÓN	17/08/2022
5		ACTIVO	DV008	Fundamentos para la Prevención Selectiva en la Reducción de la Demanda de Drogas	SALUD	17/08/2022

DV012 eliminado sigue visualizándose en la tabla

N°	OPCIONES	ESTADO	CÓDIGO	NOMBRE	ÁMBITO	FECHA DE CREACIÓN
1		ACTIVO	DV012	Modelos de Participación Comunitaria para la Prevención del Consumo de Drogas	COMUNITARIO	17/08/2022
2		ACTIVO	DV011	Técnicas y Herramientas para el Facilitador Comunitario en Prevención del Consumo de Drogas	SALUD	17/08/2022
3		ACTIVO	DV010	Estrategias de gestión pública local para la prevención del consumo de drogas legales e ilegales en el ámbito comunitario	EDUCACIÓN	17/08/2022
4		ACTIVO	DV009	Lideres Comunitarios para la Prevención del Consumo de Drogas	EDUCACIÓN	17/08/2022

- El reporte de Excel de los cursos, se muestra un espacio en blanco arriba de la tabla de resultados.

B11

ESTADO	CÓDIGO	NOMBRE	ÁMBITO	FECHA DE CREACIÓN
ACTIVO	DV013	Comunitaria para la Prevención del Consumo de Drogas	EDUCACIÓN	19/08/2022
ACTIVO	DV012	Modelos de Participación Comunitaria para la Prevención del Consumo de Drogas	COMUNITARIO	17/08/2022
ACTIVO	DV011	Técnicas y Herramientas para el Facilitador Comunitario en Prevención del Consumo de Drogas	SALUD	17/08/2022
ACTIVO	DV010	Estrategias de gestión pública local para la prevención del consumo de drogas legales e ilegales en el ámbito comunitario	EDUCACIÓN	17/08/2022
ACTIVO	DV009	Lideres Comunitarios para la Prevención del Consumo de Drogas	EDUCACIÓN	17/08/2022
ACTIVO	DV008	Fundamentos para la Prevención Selectiva en la Reducción de la Demanda de Drogas	SALUD	17/08/2022
ACTIVO	DV007	Habilidades Socioemocionales y su Relación con el Consumo de Drogas en el Ámbito Educativo	SALUD	17/08/2022
ACTIVO	DV006	Acoso Escolar (Bullying) y su Relación con el Consumo de Drogas	EDUCACIÓN	17/08/2022
ACTIVO	DV005	Facilitadores Educativos para la Prevención del Consumo de Drogas y otros Riesgos Sociales.	COMUNITARIO	17/08/2022
ACTIVO	DV004	Habilidades Socioemocionales para Fomentar la Adherencia al Tratamiento del Abuso de Drogas	SALUD	17/08/2022
ACTIVO	DV003	Motivación al Cambio para Desarrollar la Adherencia al Tratamiento por Consumo de Drogas	COMUNITARIO	17/08/2022
ACTIVO	DV002	reinserción social para el tratamiento de personas con consumo	EDUCACIÓN	17/08/2022
ACTIVO	DV001	Fundamentos básicos de la reinserción social para el tratamiento de personas con consumo de drogas	COMUNITARIO	17/08/2022

espacio en blanco del formato (se mencionó en el Daily del viernes)

- *Dentro del modal para crear o modificar el curso los valores no están establecidos, p.e. la Nota aprobatoria por lo general tiene una escala tipo 0-20, 0-100, etc. El mismo caso para las horas lectivas y porcentajes de calificación y producto. (consultar escala de valores limites).*

COPIAR CURSO

Los participantes que aprueben el curso (con una nota igual o mayor a 14) recibirán un certificado de la Comisión Nacional para el Desarrollo y Vida sin Drogas (DEVIDA) por un total de 40 horas lectivas. La institución no entrega CONSTANCIA de participación.

Profesionales o actores que laboren en instituciones públicas de gobiernos locales, regionales, ONG, equipos técnicos o coordinadores de programas sociales. El curso requiere contar con internet y con tiempo para tres 3 horas presenciales durante 4 semanas (video conferencias) y 7 horas a la semana para las actividades de aprendizaje virtual.

(*) Profesiones asociadas
Arqueólogo ☒ Cartógrafo ☒

EVALUACION

(*) Horas lectivas: 23,013,232,313,213,230,000,00d

(*) Nota aprobatoria: 2,111,111

(*) % calificación continua: 100

(*) % producto final: 100

PREREQUISITOS

Parámetros

- **Consulta:** Antes de realizar el filtrado todos los tipos de parámetros no se muestran en la tabla... ya que solo se muestran con el filtrado (puede llegar a ser confuso).

CRITERIOS DE BÚSQUEDA

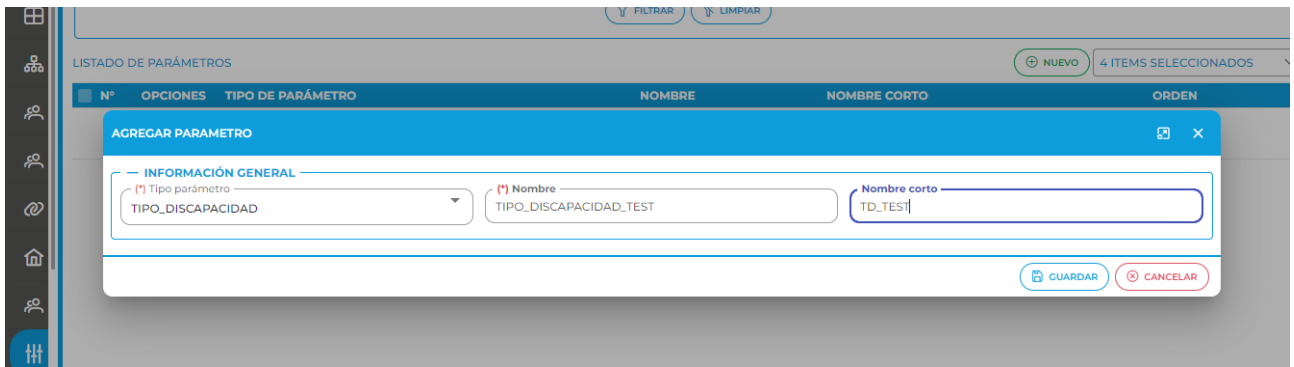
Tipo parámetro

FILTRAR LIMPIAR

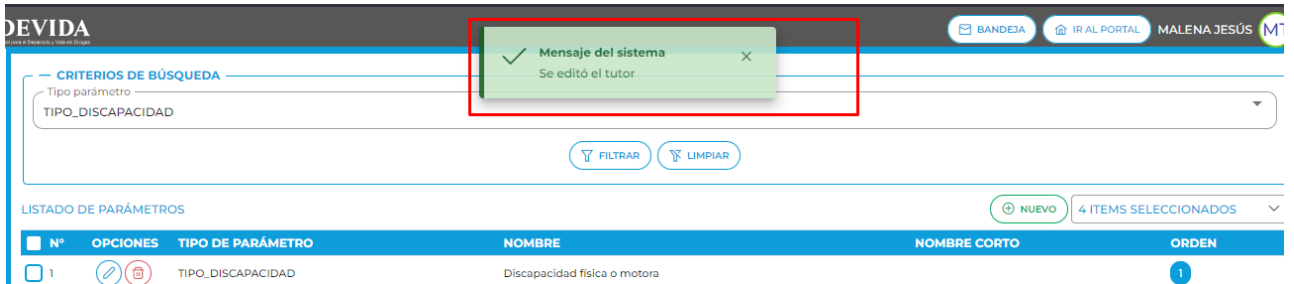
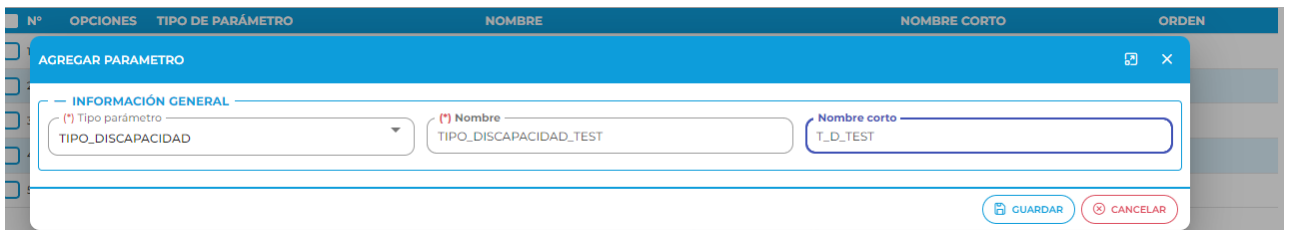
LISTADO DE PARÁMETROS NUEVO 4 ÍTEM SELECCIONADOS

N°	OPCIONES	TIPO DE PARÁMETRO	NOMBRE	NOMBRE CORTO	ORDEN
Mostrando 0 a 0 de 0 ítems					

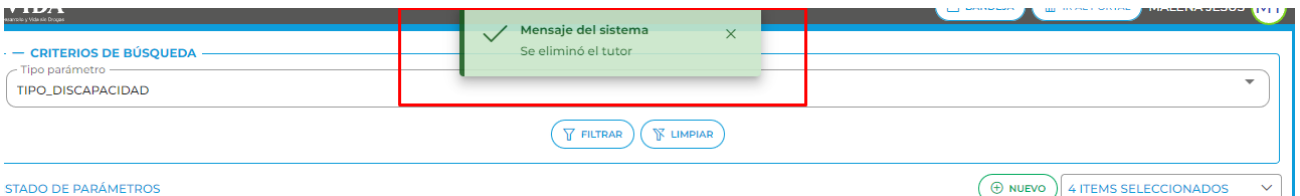
- Cuando se agrega un parámetro no se muestra en el listado de la tabla (relacionado con el ítem anterior)



Y al realizar el registro muestra un mensaje de “editado de tutor”

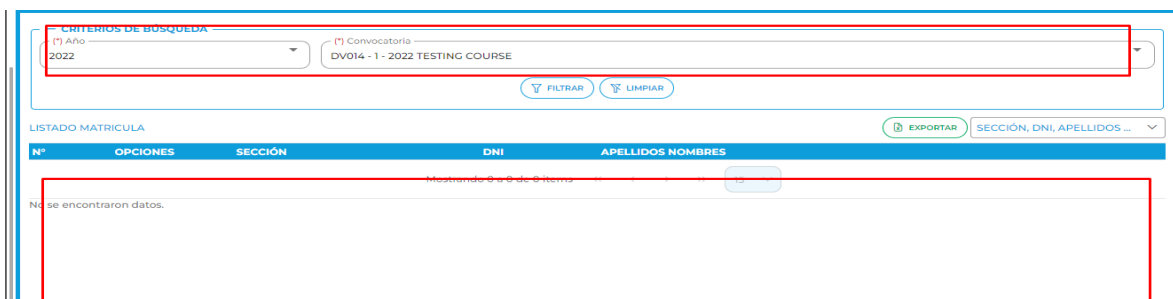


- Al momento de eliminar aparece un mensaje correspondiente a “tutor”



Matrícula

No se muestra filtrado.



Modalidades

- Se crea una modalidad de forma correcta pero para crear una convocatoria se debería tener éste valor dentro del combo de modalidad en la *convocatoria*.

N°	OPCIONES	ORDEN	TÍTULO	DESCRIPCIÓN	FECHA DE CREACIÓN	IMAGEN
1		1	Autoaprendizaje	Convocatorias a cursos de autoaprendizaje	17/08/2022	
2		2	Virtual	Convocatorias a cursos virtuales	17/08/2022	
3		3	Semipresencial	Convocatorias a cursos semipresenciales	17/08/2022	
4		4	Presencial	Convocatorias a cursos presenciales	17/08/2022	
5		5	MIXTA	MODALIDAD MIXTA	19/08/2022	

En el apartado de convocatoria no se visualiza la nueva modalidad creada.

Se había creado una modalidad mixta antes y no se visualiza

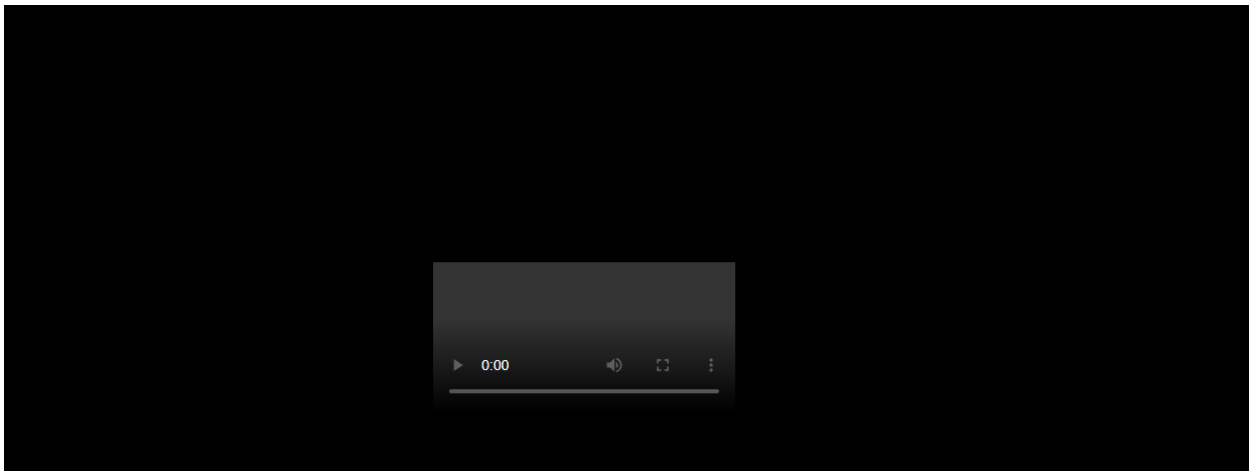
Modalidad:

- Autoaprendizaje
- Virtual
- Semipresencial
- Presencial

Documentos y videos

Al momento de visualizar el archivo .mp4, se queda cargando (revisar ruta, ignorar si es por error de infraestructura).

N°	OPCIONES	NOMBRE	TAMAÑO
1		programa-de-formacion-virtual-devida-5_EocVNBfJ.mp4	149582



- No se puede eliminar archivo

RECLAMACIÓN

SUBIR ARCHIVO







Tipos de archivo permitidos: PDF Tamaño máximo permitido: 1 Mb

ELIMINAR 2L_PDFprueba.pdf

MANUALES DE USUARIO

SUBIR ARCHIVOS

Tipos de archivo permitidos: PDF Tamaño máximo permitido: 1 Mb




N°	OPCIONES	NOMBRE	TAMAÑO
1	  	cronograma actividades.pdf	528389
2	  	2L_PDFprueba.pdf	11627

Mostrando 0 a 0 de 0 items << < > >> 15

VIDEOS

SUBIR ARCHIVOS

Tipos de archivo permitidos: VIDEO Tamaño máximo permitido: 10 Mb

N°	OPCIONES	NOMBRE	TAMAÑO
1	  	programa-de-formacion-virtual-devida-5_FocVNBFI.mn4	149582

Inscripciones

No implementado.

Notas

No implementado.

Reportes

No implementado.

CONTACTO

Los valores del combo no se despliegan, no se pudo realizar más pruebas por esta observación, por tanto no se puede enviar ningún mensaje como contacto a la entidad.

Envíanos un mensaje

(*) Nombre completo

(*) Email

(*) Asunto

(*) Mensaje

ENVIAR

MESA DE AYUDA

- No se reproduce el video (automático y manipulado), los controles están desactivados.

DEVIDA
Comisión Nacional para el Desarrollo y Vida sin Drogas

INGRESAR AL AULA VIRTUAL

Inicio Nosotros Convocatorias Reglamento Eventos Certificados **Mesa de ayuda** Contacto Iniciar sesión

MANUALES DE USUARIO

[cronograma actividades.pdf](#)

[21_PDForueba.pdf](#)

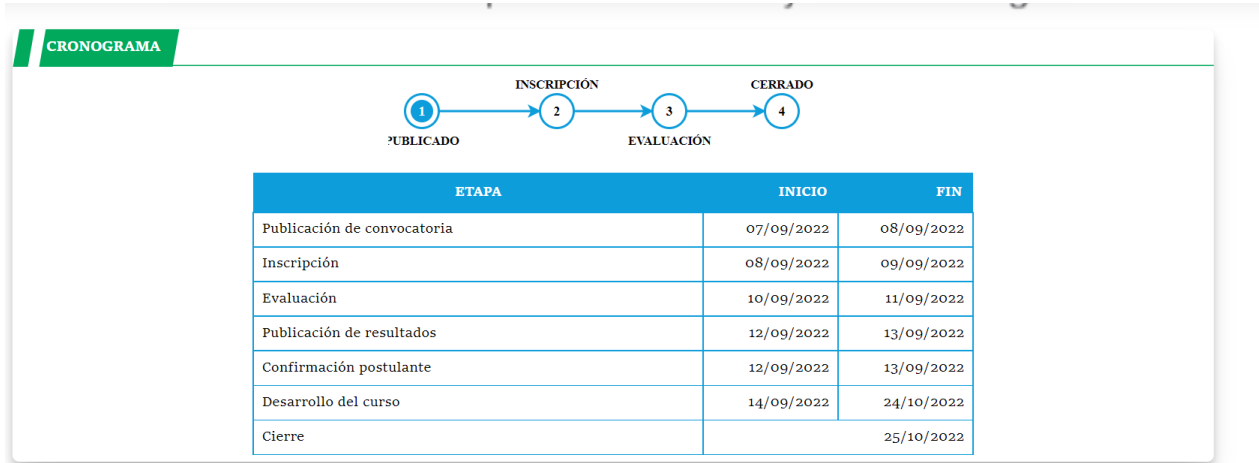
VIDEO TUTORIALES

CINTA

0:00

REVISIÓN ANTES DEL SEGUNDO ENTREGABLE:

- Se tenía como dificultad la organización correspondiente de una convocatoria según la configuración de los días y se procedió a manejarlo de diferente manera por motivos de horarios con el servidor



- Refactoring de almacenamiento de ID_POSTULANTE en el Local Storage, ocurrían errores al cerrar sesión, se podían realizar varias acciones como un usuario loggeado con privilegios. (Era por un tema de código – ya resuelto).

DEVIDA
Misión Nacional para el Desarrollo y Vida en Dignidad

CRONOGRAMA

ETAPA	INICIO	FIN
Publicación de convocatoria	07/09/2022	08/09/2022
Inscripción	08/09/2022	09/09/2022
Evaluación	10/09/2022	11/09/2022
Publicación de resultados	12/09/2022	13/09/2022
Confirmación postulante	12/09/2022	13/09/2022
Desarrollo del curso	14/09/2022	24/10/2022
Cierre		25/10/2022

Application
Manifest
Service Workers
Storage

Storage
Local Storage
https://tomcat.expertsys

Session Storage
IndexedDB
Web SQL
Cookies
Trust Tokens
Interest Groups

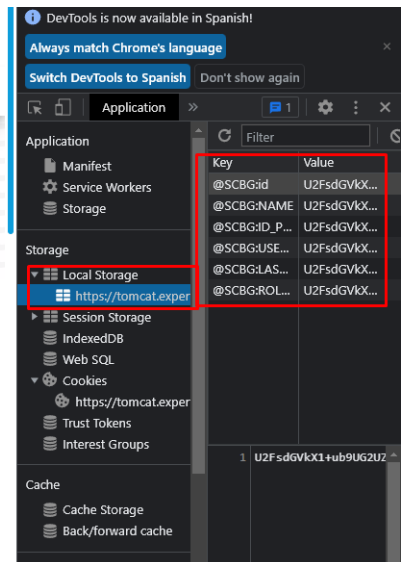
Cache
Cache Storage
Back/forward cache

Background Services
Background Fetch
Background Sync
Notifications
Payment Handler

Key Value

Select a value to preview

Al día de hoy 08/09/22 se ha vuelto a presentar este error de storage de credenciales cuando no estoy loggeado:



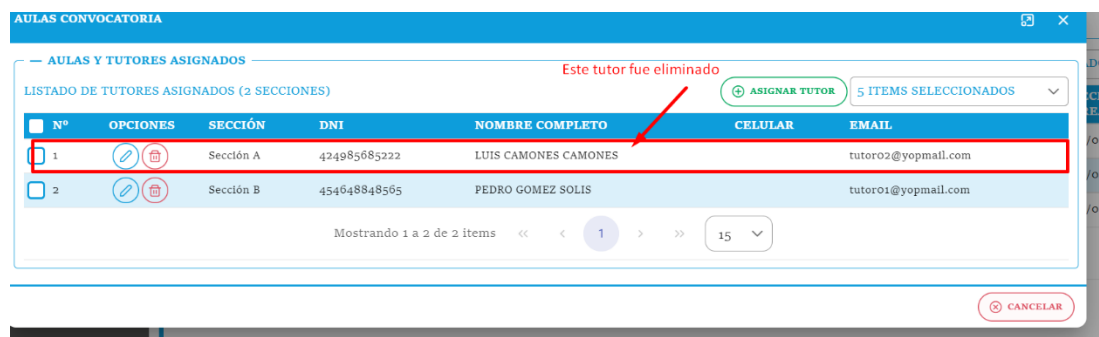
NOTA: Este error se da porque se abre 2 pestañas y se inicia sesión en una y la otra pestaña queda como si no se hiciera login pero tiene privilegios como loggeado (Se tiene que realizar un refresh de la pagina de manera forzada).

- Cuando se ha asignado un tutor a una sección X y posterior se realizó la eliminación de dicho tutor, sigue mostrándose en el listado de tutores asignados a las secciones
- Se debe poder realizar una validación para dicha acción:

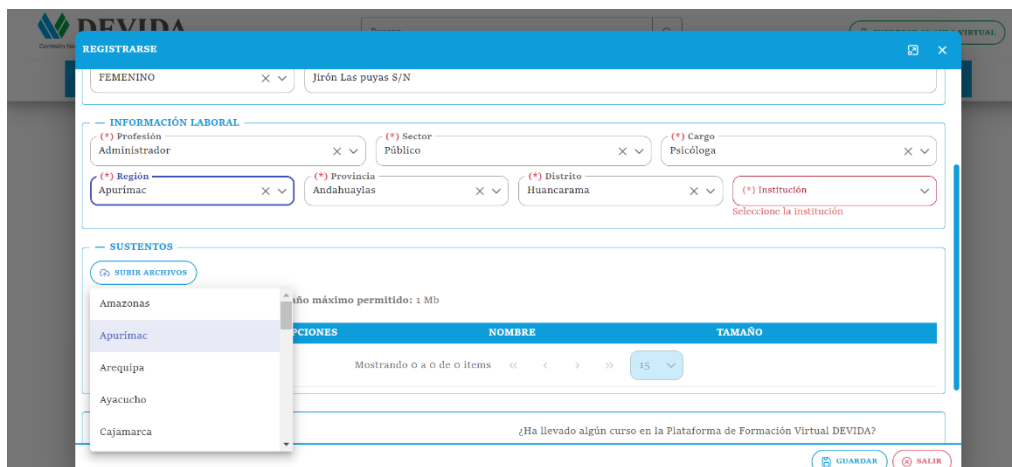
Ya se ha realizado la eliminación del tutor:



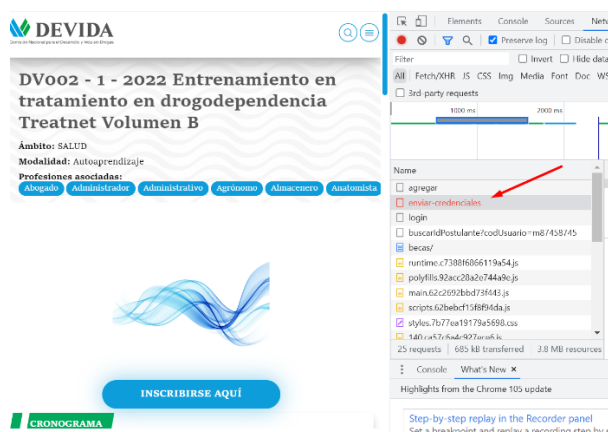
Pero sigue apareciendo en el listado de asignación de tutores en la convocatoria:



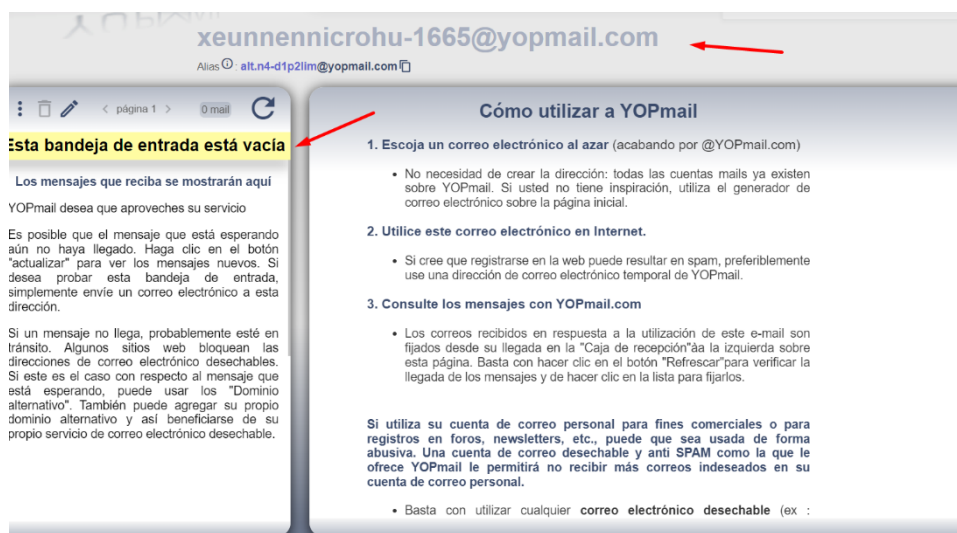
* Al momento de usar el scroll con el mouse se está deformando el combo de opciones en el modal de registro de postulante:



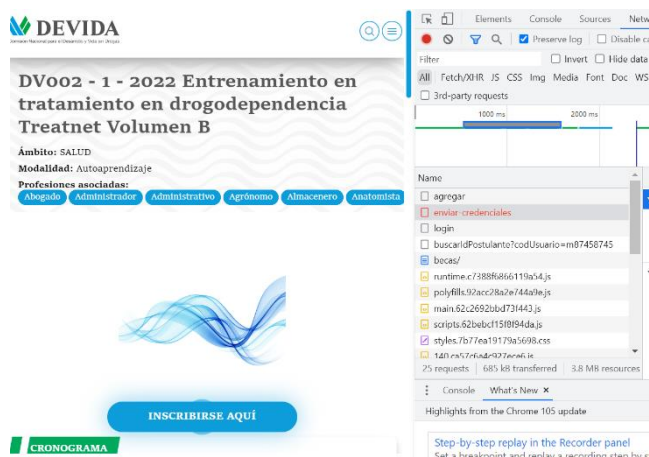
- El servicio para el envío de correos para los postulantes que se están registrando no está llegando ningún mensaje con las credenciales de su cuenta (Se ha realizado varias pruebas y todas han fallado – Fecha 07/09/2022)



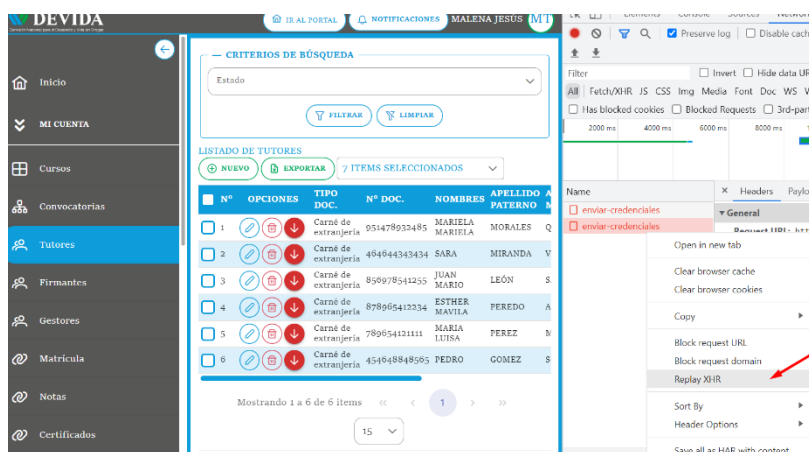
Mail :



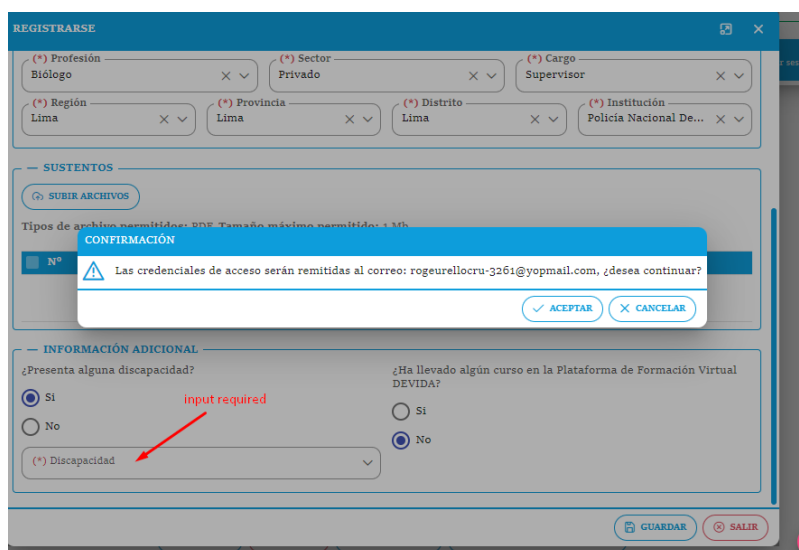
Otro ejemplo en el caso de creación de un Gestor:



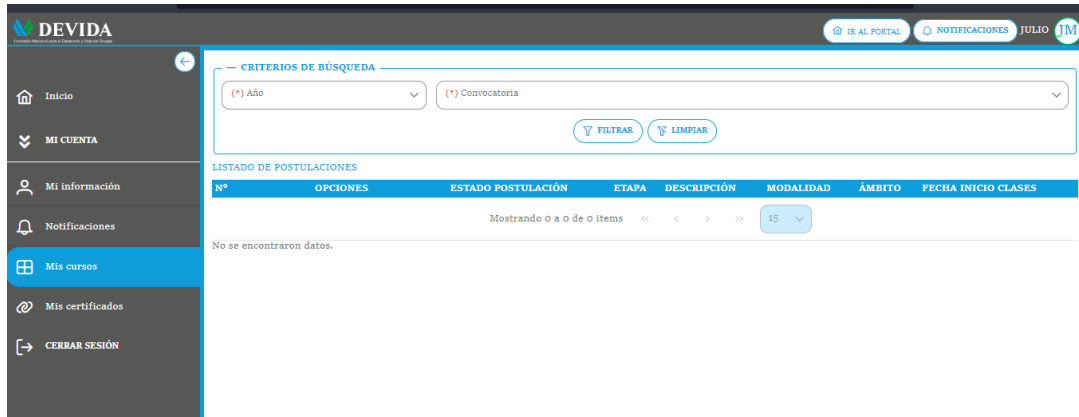
Otro ejemplo con Tutor:



- CAMPO REQUERIDO PERO PERMITE INGRESO EN BLANCO



- Con el perfil de postulante se ingresa a ver sus cursos y no se despliega por default el listado sino debe filtrarse (EVALUAR):

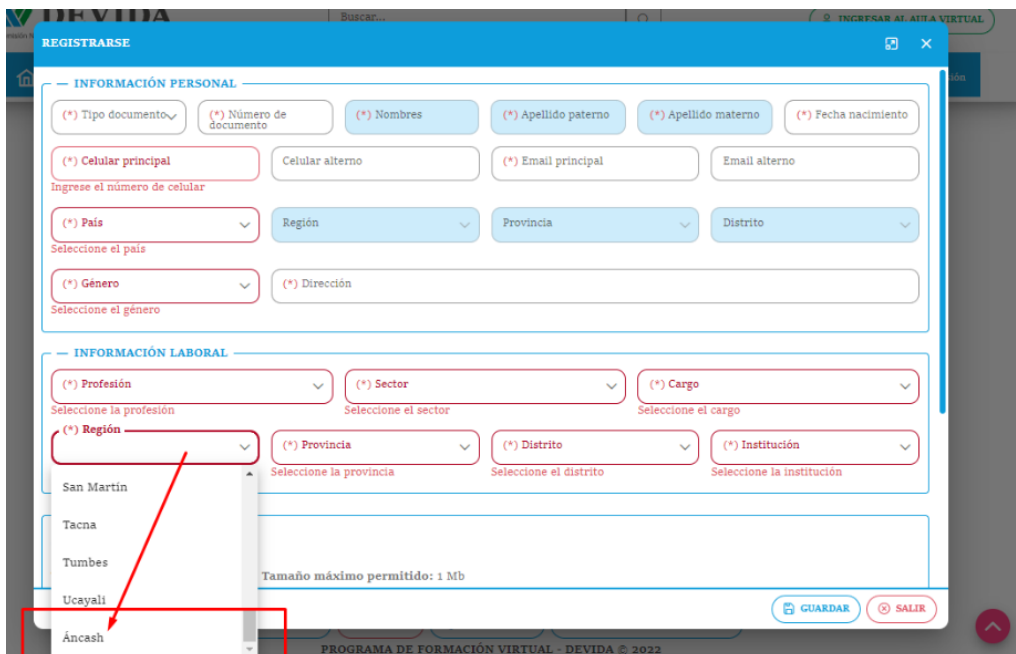


12/09 A HORAS DE ENTREGAR EL 2DO REALEASE:

- Validación de extensión de firma digital.
- Modificación de cambio de estado "RETIRADO" a "RETIRO DEL CURSO"
- Los Responses en los combos de los certificados no funciona porque está listando a todos los cursos aunque el postulante, firmante u otro no esté vinculado con este curso.
- OBS: Overflow del Combo y Orden de valores en Región

Se ha notado que se tiene un overflow en los valores dentro del combo Región.

- Además del ordenamiento alfabético que "Áncash" al tener tilde se está posicionando al final (consultar porque puede llegar a ser confuso su posición).



STRUCTURE TEST

Se realizaron este tipo de pruebas teniendo como objetivo principal la mejora a futuro con respecto al código fuente elaborado en el proyecto hasta el momento. Solo se está resumiendo los tipos de issues que se presentan debido a su similitud.

Para este ítem se tuvo en cuenta el código fuente del backend teniendo como resultado en modo resumen lo siguiente:

☆ **Devidabackend** Passed Last analysis: 6 days ago

🐛 Bugs	🔒 Vulnerabilities	🔍 Hotspots Reviewed	👤 Code Smells	Coverage	Duplications	Lines
67 C	0 A	0.0% E	7.1k A	0.0% C	23.5% C	78k M Java, XML,...

En este documento solo se tuvo en cuenta los aspectos más importantes.

Bugs

Algunos de los más importantes son los siguientes:

- src/.../drivenAdapters/repository/AccesoDirectoRepositoryImpl.java
Call "existingObject.isPresent()" before accessing the value. 1 month ago ▾ L47 🔗 🔽
🐛 Bug ▾ 🚨 Major ▾ 🔵 Open ▾ Not assigned ▾ 10min effort Comment 🔗 cwe ▾
- src/.../drivenAdapters/repository/AlumnoCertificadoRepositoryImpl.java
Call "existingObject.isPresent()" before accessing the value. 2 months ago ▾ L50 🔗 🔽
🐛 Bug ▾ 🚨 Major ▾ 🔵 Open ▾ Not assigned ▾ 10min effort Comment 🔗 cwe ▾
- src/.../drivenAdapters/repository/AlumnoRepositoryImpl.java
Call "existingObject.isPresent()" before accessing the value. 2 months ago ▾ L72 🔗 🔽
🐛 Bug ▾ 🚨 Major ▾ 🔵 Open ▾ Not assigned ▾ 10min effort Comment 🔗 cwe ▾
- Strings and Boxed types should be compared using "equals()". 18 days ago ▾ L130 🔗 🔽
🐛 Bug ▾ 🚨 Major ▾ 🔵 Open ▾ Not assigned ▾ 5min effort Comment 🔗 cert, cwe ▾
- Call "miListaAptoDao.isPresent()" before accessing the value. 21 days ago ▾ L175 🔗 🔽
🐛 Bug ▾ 🚨 Major ▾ 🔵 Open ▾ Not assigned ▾ 10min effort Comment 🔗 cwe ▾
- Remove or correct this "removeAll" call. 21 days ago ▾ L179 🔗 🔽
🐛 Bug ▾ 🚨 Major ▾ 🔵 Open ▾ Not assigned ▾ 15min effort Comment 🔗 No tags ▾

🔗 Call "existingObject.isPresent()" before accessing the value.

Devidabackend src/.../infrastructure/drivenAdapters/repository/AccesoDirectoRepositoryImpl.java See all issues in this file 🔗

```

42 rmore... public AccesoDirecto obtener(Long idAccesoDirecto) {
43         Optional<AccesoDirectoDao> existingObject = accesoDirectoJpa.findById((idAccesoDirecto));
44 rmore...         if (!existingObject.isPresent()) {
45 rmore...             //throw new
46         }
47         return mvcConversionService.convert(existingObject.get(), AccesoDirecto.class);
48     }
    
```

🐛 Call "existingObject.isPresent()" before accessing the value.

¿Por qué es un issue?

Se ha considerado como un Bug ya que antes de obtener el valor correspondiente se debe validar que existe/que está presente porque si dado el caso se está tomando el valor cuando no existe, esto nos daría un error.

Solución:

```
Optional value can hold either a value or not. The value held in the Optional can be accessed using the get() method, but it will throw a NoSuchElementException if there is no value present. To avoid the exception, calling the isPresent() or ! isEmpty() method should always be done before any call to get().

Alternatively, note that other methods such as orElse(...), orElseGet(...), or orElseThrow(...) can be used to specify what to do with an empty Optional.
```

Noncompliant Code Example

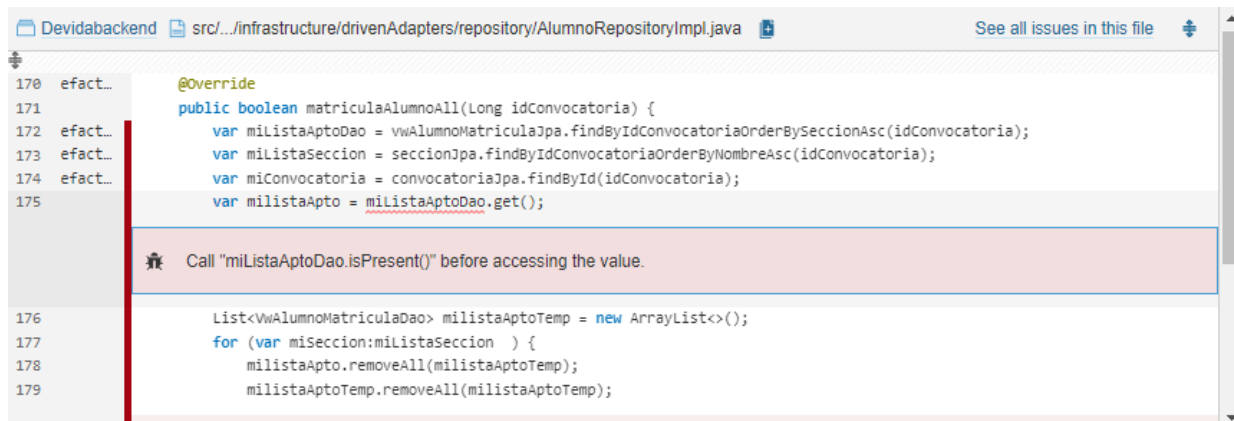
```
Optional<String> value = this.getOptionalValue();
```

Algunos de los archivos donde se encuentra este issue:

- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/AccesoDirectoRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/AlumnoCertificadoRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/AlumnoRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/ArchivoRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/BannerRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/ConvocatoriaRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/CursoRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/EventoRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/FirmanteRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/GestorRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/InstitucionRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/LogoRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/ModalidadRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/PaisRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/ParametroRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/PostulanteRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/PreguntaFrecuenteRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/ResponsableInstitucionRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/SeccionRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/SolicitudBecaRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/TblSegUsuarioRepositoryImpl.java
- src/main/java/pe/gob/devida/becas/infraestructura/drivenAdapters/repository/TutorRepositoryImpl.java

Otros parecidos:

✚ Call "miListaAptoDao.isPresent()" before accessing the value.



The screenshot shows an IDE window titled "Devidabackend" with the file path "src/.../infrastructure/drivenAdapters/repository/AlumnoRepositoryImpl.java". The code is as follows:

```
170 efact... @Override
171           public boolean matriculaAlumnoAll(Long idConvocatoria) {
172 efact...         var miListaAptoDao = vwAlumnoMatriculaJpa.findByIdConvocatoriaOrderBySeccionAsc(idConvocatoria);
173 efact...         var miListaSeccion = seccionJpa.findByIdConvocatoriaOrderByNombreAsc(idConvocatoria);
174 efact...         var miConvocatoria = convocatoriaJpa.findById(idConvocatoria);
175           var miListaApto = miListaAptoDao.get();

176           List<VwAlumnoMatriculaDao> miListaAptoTemp = new ArrayList<>();
177           for (var miSeccion:miListaSeccion ) {
178               miListaApto.removeAll(miListaAptoTemp);
179               miListaAptoTemp.removeAll(miListaAptoTemp);
```

An IDE error message is displayed below the code, stating: "Call 'miListaAptoDao.isPresent()' before accessing the value." The error points to line 175 where `miListaAptoDao.get()` is called.

Solución:

```
this.getOptionalValue().ifPresent(stringValue ->
    // Do something with stringValue
);
```

or

```
Optional<String> value = this.getOptionalValue();

// ...

if (value.isPresent()) {
    String stringValue = value.get();
}
```

Más:

```

Devidabackend src/.../infraestructura/drivenAdapters/repository/NotaRepositoryImpl.java See all issues in this file
182 efect...     var miIdAlumnoMoodle= buscarUsuario("username",miAlumno.getUsuarioMoodle());
183             if(miIdAlumnoMoodle>0){
184                 var idUsuario=miIdAlumnoMoodle;
185 efect...     //MIS NOTAS REGISTRADOS
186             var misNotasSincronizados=notaJpa.findByIdAlumno(miAlumno.getIdAlumno());
187             var miListaNotaSincronizado=miNotasSincronizados.get();

Call "misNotasSincronizados.isPresent()" before accessing the value.

188 efect...     var misEvaluacionesNotas=obtenerAlumnoNotas(idUsuario,misNotasAll);
189
190             for ( var miEvaluacionNota: misEvaluacionesNotas) {
191                 if(miEvaluacionNota.getItemmodule()!=null){
192                     //SI ES UNA NOTA NO ES CALCULADO

```

```

Devidabackend src/.../infraestructura/drivenAdapters/repository/NotaRepositoryImpl.java See all issues in this file
104             if(miIdAlumnoMoodle>0){
105                 var idUsuario=miIdAlumnoMoodle;
106 efect...     //MIS NOTAS REGISTRADOS
107             var misNotasSincronizados=notaJpa.findByIdAlumno(miAlumno.getIdAlumno());
108             var miListaNotaSincronizado=miNotasSincronizados.get();

Call "misNotasSincronizados.isPresent()" before accessing the value.

109 efect...     var misEvaluacionesNotas=obtenerAlumnoNotas(idUsuario,misNotasAll);
110
111             for ( var miEvaluacionNota: misEvaluacionesNotas) {
112                 if(miEvaluacionNota.getItemmodule()!=null){
113                     //SI ES UNA NOTA NO ES CALCULADO
114                 var esNotaSincronizado= buscarNotaRegistrosSincronizado( idUsuario,miEvaluacionNota
,misListaNotaSincronizado);

```

```

Devidabackend src/.../infraestructura/drivenAdapters/repository/PaisRepositoryImpl.java See all issues in this file
34 acast...     public String obtenerNombrePorId(Long idPais) {
35                 Optional<PaisDao> existingObject = paisJpa.findById(idPais);
36 rmore...     if (!existingObject.isPresent()) {
37 acast...     //throw new
38             }
39             return existingObject.get().getNombre();

Call "existingObject.isPresent()" before accessing the value.

40         }
41 acast...     }
42

```

✚ Use isEmpty() to check whether the collection is empty or not.

Para este tipo de validación se recomienda usar el metodo isEmpty() antes de size()==0

Ejemplo del archivo:

src/main/java/pe/gob/devida/becas/infraestructure/drivenAdapters/repository/AccesoDirectoRepositoryImpl.java

```
31
32     @Override
33     public List<AccesoDirectoGridRow> listar(String nombre) {
34         List<AccesoDirectoGridRowDao> existingObject = accesoDirectoJpa.filtrarAccesosDirectos(nombre.toUpperCase());
35         if (existingObject.size() == 0) {
36             // throw new
37         }
38         return existingObject.stream().map(p -> mvcConversionService.convert(p,
```

Solución:

Using `Collection.size()` to test for emptiness works, but using `Collection.isEmpty()` makes the code more readable and can be more performant. The time complexity of any `isEmpty()` method implementation should be $O(1)$ whereas some implementations of `size()` can be $O(n)$.

Noncompliant Code Example

```
if (myCollection.size() == 0) { // Noncompliant
    /* ... */
}
```

Compliant Solution

```
if (myCollection.isEmpty()) {
    /* ... */
}
```

🚦 Strings and Boxed types should be compared using "equals()

Casi siempre es un error comparar dos instancias de `java.lang.String` o tipos en caja como `java.lang.Integer` usando la igualdad de referencia `==` o `!=`, porque no está comparando el valor real sino las ubicaciones en la memoria.

```

125  efact...      }
126  efact...      var miConvocatoria = convocatoriaJpa.findById(alumnoMatricula.getAlumnoMatricula().getIdConvocatoria());
127  efact...      var cantMatriculados= alumnoJpa.countByIdSeccion(alumnoMatricula.getIdSeccion());
128  efact...      if(miAlumnoMatricula.get().getIdAlumno()!=null){
129  efact...          //MODIFICAMOS SECCION
130  efact...          if(alumnoMatricula.getIdSeccion()!=miAlumnoMatricula.get().getIdSeccion()){

131  efact...          if(cantMatriculados<miConvocatoria.get().getNumeroVacantesPorAula()){
132  efact...          var miAlumnoOptional= alumnoJpa.findById(miAlumnoMatricula.get().getIdAlumno());
133  efact...          var miAlumnoEdit=miAlumnoOptional.get();
134  efact...          miAlumnoEdit.setIdSeccion(alumnoMatricula.getIdSeccion());
135  efact...          alumnoJpa.save(miAlumnoEdit);

```

Solución:

```

String firstName = getFirstName();
String lastName = getLastName();

if (firstName != null && firstName.equals(lastName)) { ... };

```

✚ Remove or correct this "removeAll" call.

Algunos métodos requieren que el argumento permanezca sin modificar durante la ejecución, pasar una colección a sí misma puede resultar en un comportamiento indefinido.

```

176  efact...      List<VwAlumnoMatriculaDao> milistaAptoTemp = new ArrayList<>();
177  efact...      for (var miSeccion:miListaSeccion ) {
178  efact...          milistaApto.removeAll(miListaAptoTemp);
179  efact...          milistaAptoTemp.removeAll(miListaAptoTemp);

180  efact...      var cantMatriculados = alumnoJpa.countByIdSeccion(miSeccion.getId());
181  efact...      for (var miAlumno:miListaApto ) {
182  efact...          var miAlumnoMatricula = vwAlumnoMatriculaJpa.findByIdSolicitudBeca(miAlumno.getIdSolicitudBeca());
183  efact...          if (miAlumnoMatricula.get().getIdAlumno() == null) {

```

Solución:

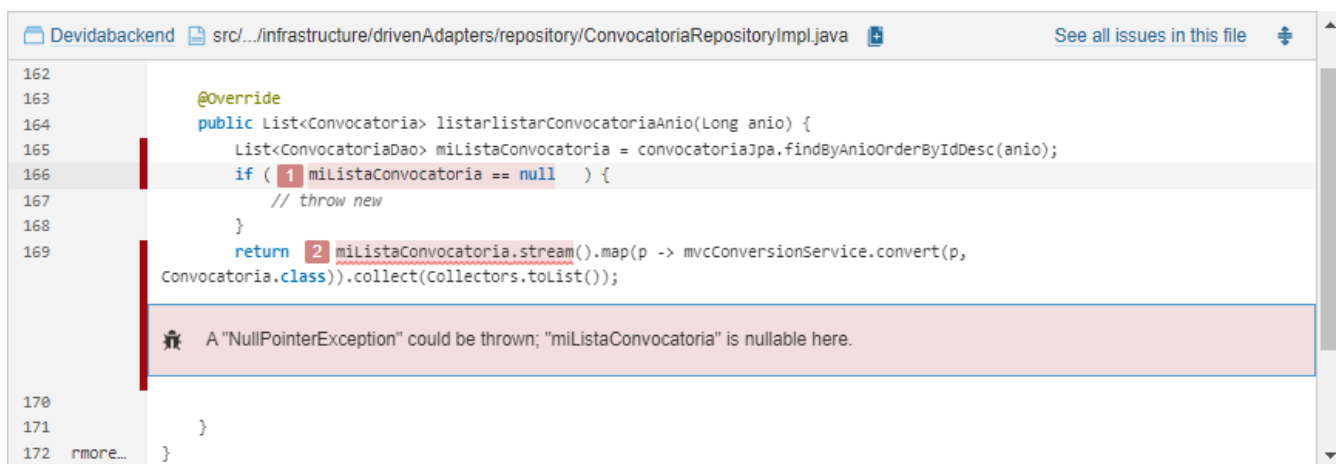
Noncompliant Code Example

```
List<Object> objs = new ArrayList<Object>();
objs.add("Hello");

objs.add(objs); // Noncompliant; StackOverflowException if objs.hashCode() called
objs.addAll(objs); // Noncompliant; behavior undefined
objs.containsAll(objs); // Noncompliant; always true
objs.removeAll(objs); // Noncompliant; confusing. Use clear() instead
objs.retainAll(objs); // Noncompliant; NOOP
```

✚ A "NullPointerException" could be thrown; "miListaConvocatoria" is nullable here.

Nunca se debe desreferenciar/acceder a una referencia a nulo. Si lo hace, se lanzará una NullPointerException. En el mejor de los casos, tal excepción provocará la terminación abrupta del programa. En el peor de los casos, podría exponer información de depuración que sería útil para un atacante, o podría permitir que un atacante eluda las medidas de seguridad.



The screenshot shows an IDE window titled "Devidabackend" with the file "src/.../infrastructure/drivenAdapters/repository/ConvocatoriaRepositoryImpl.java". The code snippet is as follows:

```
162
163     @Override
164     public List<Convocatoria> listarlistarConvocatoriaAnio(Long anio) {
165         List<ConvocatoriaDao> miListaConvocatoria = convocatoriaJpa.findByIdDesc(anio);
166         if ( 1 miListaConvocatoria == null ) {
167             // throw new
168         }
169         return 2 miListaConvocatoria.stream().map(p -> mvcConversionService.convert(p,
Convocatoria.class)).collect(Collectors.toList());
170
171     }
172     rmore... }
```

A warning message is displayed below the code: "A 'NullPointerException' could be thrown; 'miListaConvocatoria' is nullable here."

Otros similares:



The screenshot shows an IDE window titled "Devidabackend" with the file "src/.../infrastructure/drivenAdapters/repository/DepartamentoRepositoryImpl.java". The code snippet is as follows:

```
40 acast... }
41
42 acast... @Override
43 rmore... public String obtenerNombrePorId(String idDepartamento) {
44     DepartamentoDao existingObject = departamentoJpa.findByIdDepartamento(idDepartamento);
45     if ( 1 existingObject == null ) {
46         //throw new
47     }
48     return 2 existingObject.getNombre();
49 acast... }
50
51 rmore... }
```

A warning message is displayed below the code: "A 'NullPointerException' could be thrown; 'existingObject' is nullable here."


```

Devidabackend src/.../infrastructure/drivenAdapters/repository/DistritoRepositoryImpl.java See all issues in this file
40 acast... @Override
41 rmore... public String obtenerNombrePorId(String idDistrito) {
42         DistritoDao existingObject = distritoJpa.findByIdDistrito(idDistrito);
43 acast...         if ( 1 existingObject == null) {
44             //throw new
45         }
46 rmore...         return 2 existingObject.getNombre();
47 acast...     }
48
49 rmore... }
50

```

✖ A "NullPointerException" could be thrown; "existingObject" is nullable here.

✚ **Insert a <!DOCTYPE> declaration to before this <HTML> tag**

La declaración <!DOCTYPE> le dice al navegador web qué versión (X)HTML se está utilizando en la página y, por lo tanto, cómo interpretar los diversos elementos. Los validadores también confían en él para saber qué reglas aplicar. Siempre debe preceder a la etiqueta <html>.

```

Devidabackend src/main/resources/templates/SmtpeMailTemplate.html See all issues in this file
1 aheno... <HTML>
2
3         <HEAD>
4             <TITLE>Correo Devida</TITLE>
5         </HEAD>
6         <BODY>
7             [ ${MensajeBody} ]
8         <P><B>Enviado por Sistema De Seguridad y Configuracion</B></P>
9

```

✖ Insert a <!DOCTYPE> declaration to before this <HTML> tag.

✖ Add "lang" and/or "xml:lang" attributes to this "<html>" element

✖ Replace this tag by .

✚ **Add "lang" and/or "xml:lang" attributes to this "<html>" element**

Las etiquetas / y /<i> tienen exactamente el mismo efecto en la mayoría de los navegadores web, pero existe una diferencia fundamental entre ellas: y tienen un significado semántico, mientras que y <i> solo transmiten información de estilo como CSS.

```

1  ahen... <HTML><HEAD><TITLE>Correo Devida</TITLE></HEAD><BODY>[{$MensajeBody}]<P><B>Enviado por Sistema De Seguridad y
Configuracion</B></P><P><I>Esta es una notificaci&oacute;n autom&aacute;tica, por favor no la responda.</I></P></BODY>
</HTML>

✖ Insert a <!DOCTYPE> declaration to before this <HTML> tag.

✖ Add "lang" and/or "xml:lang" attributes to this "<html>" element

✖ Replace this <B> tag by <strong>.

✖ Replace this <I> tag by <em>.

```

Security Hotspots

✚ 'password' detected in this expression, review this potentially hard-coded password

Debido a que es fácil extraer cadenas del código fuente o binario de una aplicación, las contraseñas no deben codificarse de forma rígida. Esto es particularmente cierto para las aplicaciones que se distribuyen o que son de código abierto.

```

24  @service
25  public class FirmaDigital {
26      public void firmarArchivo(InputStream pdfParaFirmar, OutputStream pdfFirmado, String razon, String contacto, String
localizacion, Rectangle rectanguloFirma, Image imagen, Integer paginaFirma) throws KeyStoreException, IOException,
UnrecoverableKeyException, NoSuchAlgorithmException, CertificateException {
27          String nombreFirmaDigital = "my_private_key.pfx";
28          String passwordFirmaDigital = "12345678";

'password' detected in this expression, review this potentially hard-coded password.

29          KeyStore ks = KeyStore.getInstance("pkcs12");
30          Resource resource = new ClassPathResource(nombreFirmaDigital);
31          ks.load(resource.getInputStream(), passwordFirmaDigital.toCharArray());

```

Solución:

Recommended Secure Coding Practices

- Store the credentials in a configuration file that is not pushed to the code repository.
- Store the credentials in a database.
- Use your cloud provider's service for managing secrets.
- If a password has been disclosed through the source code: change it.

Compliant Solution

```
Connection conn = null;
try {
    String uname = getEncryptedUser();
    String password = getEncryptedPass();
    conn = DriverManager.getConnection("jdbc:mysql://localhost/test?" +
        "user=" + uname + "&password=" + password);
```

Make sure that enabling CORS is safe here – una cantidad total de (8)



The screenshot shows a code editor window for a file named `AlumnoCertificadoController.java`. The code includes imports for `AlumnoCertificadoInicializarDto` and `java.util.List`, and annotations for `@RestController`, `@RequestMapping("/devida/api/alumno-certificado")`, and `@CrossOrigin(origins = {"*"}, allowedHeaders = "*")`. A red warning box is overlaid on the `@CrossOrigin` annotation with the text "Make sure that enabling CORS is safe here." and a "Comment" button.

Tener una política de uso compartido de recursos de origen cruzado permisiva es sensible a la seguridad.

Spring MVC framework:

- [CrossOrigin](#)

```
@CrossOrigin("trustedwebsite.com") // Compliant
@RequestMapping("")
public class TestController {
    public String home(ModelMap model) {
        model.addAttribute("message", "ok ");
        return "view";
    }
}
```

✚ **Make sure using this hardcoded IP address is safe here.**

Los servicios actuales tienen una arquitectura en constante cambio debido a sus necesidades de escalabilidad y redundancia. Es un error pensar que un servicio siempre tendrá la misma dirección IP. Cuando cambie, la IP codificada también deberá modificarse. Esto tendrá un impacto en el desarrollo, la entrega y la implementación del producto:



```
src/main/java/pe/gob/devida/becas/applications/SecurityConfig.java
53  @Bean
54  CorsConfigurationSource corsConfigurationSource() {
55      CorsConfiguration configuration = new CorsConfiguration();
56      configuration.setAllowedOrigins(Arrays.asList("http://localhost:4200", "http://127.0.0.1:4200",
57      "https://wildfly.expertsystems.pe", "https://sistemas.devda.gob.pe", "https://tomcat.expertsystems.pe",
58      "http://192.168.1.216:8080"));
59
60      configuration.addAllowedMethod("*");
61      configuration.addAllowedHeader("*");
62      UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();
63      source.registerCorsConfiguration("/*" configuration);
```

Solución:

Compliant Solution

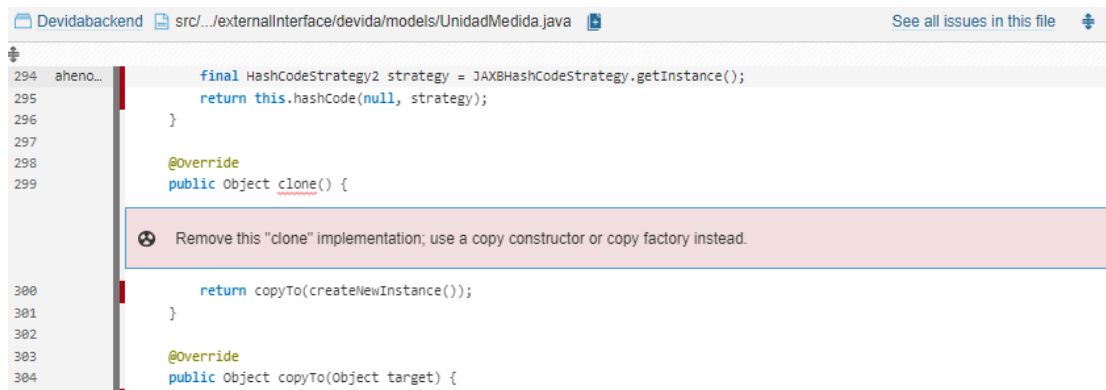
```
String ip = System.getenv("IP_ADDRESS"); // Compliant
Socket socket = new Socket(ip, 6667);
```

Code Smell

Bloqueante

✚ **Remove this "clone" implementation; use a copy constructor or copy factory instead (con un resultado de 147 veces realizado)**

El clon generalmente comparte estado con el objeto que se está clonando. Si ese estado es mutable, no tienes dos objetos independientes. Si modificas uno, el otro también cambia. Y de repente, obtienes un comportamiento aleatorio.



```

294 aheno...   final HashCodeStrategy2 strategy = JAXBHashCodeStrategy.getInstance();
295           return this.hashCode(null, strategy);
296       }
297
298       @Override
299       public Object clone() {
300
301           return copyTo(createNewInstance());
302       }
303
304       @Override
305       public Object copyTo(Object target) {

```

Error:

Noncompliant Code Example

```

public class MyClass {
    // ...

    public Object clone() { // Noncompliant
        //...
    }
}

```

Solución:

```

public class MyClass {
    // ...

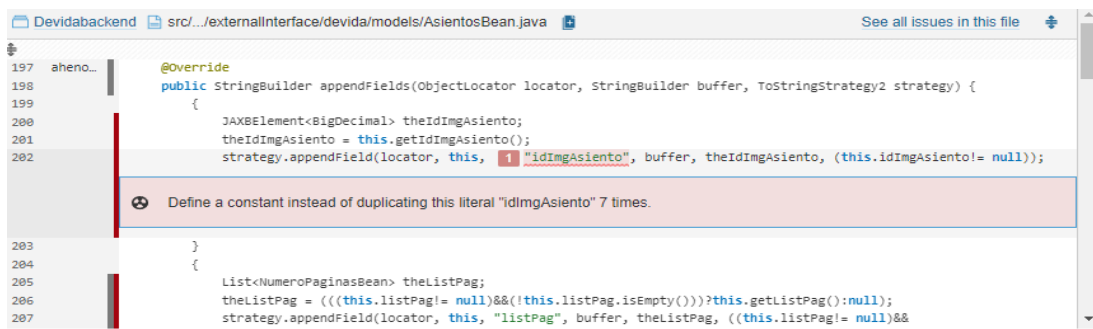
    MyClass (MyClass source) {
        //...
    }
}

```

Code Smell

*Critico
Total: 932*

 **Define a constant instead of duplicating this literal "idImgAsiento" 7 times.**



```

197 aheno...   @Override
198           public StringBuilder appendFields(ObjectLocator locator, StringBuilder buffer, ToStringStrategy2 strategy) {
199               {
200                   JAXBElement<BigDecimal> theIdImgAsiento;
201                   theIdImgAsiento = this.getIdImgAsiento();
202                   strategy.appendField(locator, this, "idImgAsiento", buffer, theIdImgAsiento, (this.idImgAsiento != null));
203               }
204           }
205           {
206               List<NumeroPaginasBean> theListPag;
207               theListPag = (((this.listPag != null) && (!this.listPag.isEmpty())) ? this.getListPag() : null);
208               strategy.appendField(locator, this, "listPag", buffer, theListPag, ((this.listPag != null) &&

```

```

Devidabackend sro/.../externalInterface/devida/models/AsientosBean.java See all issues in this file
232 JAXBElement<BigDecimal> lhsIdImgAsiento;
233 lhsIdImgAsiento = this.getIdImgAsiento();
234 JAXBElement<BigDecimal> rhsIdImgAsiento;
235 rhsIdImgAsiento = that.getIdImgAsiento();
236 if (!strategy.equals(LocatorUtils.property(thisLocator, 2 "idImgAsiento", lhsIdImgAsiento),
LocatorUtils.property(thatLocator, 3 "idImgAsiento", rhsIdImgAsiento), lhsIdImgAsiento, rhsIdImgAsiento,
(this.idImgAsiento!= null), (that.idImgAsiento!= null))) {
237     return false;
238 }
239 }
240 {
241     List<NumeroPaginasBean> lhsListPag;

aheno... public int hashCode(ObjectLocator locator, HashCodeStrategy2 strategy) {
278     int currentHashCode = 1;
279     {
280         JAXBElement<BigDecimal> theIdImgAsiento;
281         theIdImgAsiento = this.getIdImgAsiento();
282         currentHashCode = strategy.hashCode(LocatorUtils.property(locator, 4 "idImgAsiento", theIdImgAsiento),
currentHashCode, theIdImgAsiento, (this.idImgAsiento!= null));

```

Solución:

```

private static final String ACTION_1 = "action1"; // Compliant

public void run() {
    prepare(ACTION_1); // Compliant
    execute(ACTION_1);
    release(ACTION_1);
}

```

Se ha encontrado este tipo de code smell en varios archivos como por ejemplo:

```

Devidabackend sro/.../entryPoints/controller/Public/ReportePublicController.java See all issues in this file
227 headers.add(HttpHeaders.CONTENT_DISPOSITION, "attachment; filename=Convocatorias.xlsx");
228
229 return ResponseEntity.ok()
230     .contentType(
231         MediaType.valueOf(1 "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet"))
232     .headers(headers)
233     .body(new InputStreamResource(ins));
234
235 } catch (Exception e) {
236     log.error("Error al generar el reporte", e);
237     throw new ClientException(e);
238 }

```

Define a constant instead of duplicating this literal "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet" 8 times.

Bulk Change		211 / 932 issues	42d effort
<input type="checkbox"/>	Define a constant instead of duplicating this literal "ddpDoble" 7 times. Code Smell Critical Open Not assigned 16min effort Comment	2 months ago L1304	design
<input type="checkbox"/>	Define a constant instead of duplicating this literal "ddpEstado" 7 times. Code Smell Critical Open Not assigned 16min effort Comment	2 months ago L1309	design
<input type="checkbox"/>	Define a constant instead of duplicating this literal "ddpFecact" 7 times. Code Smell Critical Open Not assigned 16min effort Comment	2 months ago L1314	design
<input type="checkbox"/>	Define a constant instead of duplicating this literal "ddpFecal" 7 times. Code Smell Critical Open Not assigned 16min effort Comment	2 months ago L1319	design
<input type="checkbox"/>	Define a constant instead of duplicating this literal "ddpFecbaj" 7 times. Code Smell Critical Open Not assigned 16min effort Comment	2 months ago L1324	design
<input type="checkbox"/>	Define a constant instead of duplicating this literal "ddpFlag22" 7 times. Code Smell Critical Open Not assigned 16min effort Comment	2 months ago L1329	design
<input type="checkbox"/>	Define a constant instead of duplicating this literal "ddpIdenti" 7 times. Code Smell Critical Open Not assigned 16min effort Comment	2 months ago L1334	design
<input type="checkbox"/>	Define a constant instead of duplicating this literal "ddpInter1" 7 times. Code Smell Critical Open Not assigned 16min effort Comment	2 months ago L1339	design
<input type="checkbox"/>	Define a constant instead of duplicating this literal "ddpLlitt" 7 times. Code Smell Critical Open Not assigned 16min effort Comment	2 months ago L1344	design
<input type="checkbox"/>	Define a constant instead of duplicating this literal "ddpMolase" 7 times. Code Smell Critical Open Not assigned 16min effort Comment	2 months ago L1349	design
<input type="checkbox"/>	Define a constant instead of duplicating this literal "ddpNombre" 7 times. Code Smell Critical Open Not assigned 16min effort Comment	2 months ago L1354	design
<input type="checkbox"/>	Define a constant instead of duplicating this literal "ddpNomvia" 7 times. Code Smell Critical Open Not assigned 16min effort Comment	2 months ago L1359	design

✚ Refactor this method to reduce its Cognitive Complexity from 22 to the 15 allowed

```

Devidabackend src/.../externalInterface/devida/models/PODERJUDICIALVerificarAntecedentesPenalesResponse.java See all issues in this file
Refactor this method to reduce its Cognitive Complexity from 22 to the 15 allowed.

217 1 if (right instanceof PODERJUDICIALVerificarAntecedentesPenalesResponse) {
218     final PODERJUDICIALVerificarAntecedentesPenalesResponse target = this;
219     final PODERJUDICIALVerificarAntecedentesPenalesResponse leftObject =
220     ((PODERJUDICIALVerificarAntecedentesPenalesResponse) left);
221     final PODERJUDICIALVerificarAntecedentesPenalesResponse rightObject =
222     ((PODERJUDICIALVerificarAntecedentesPenalesResponse) right);
223     {
224         Boolean _returnShouldBeMergedAndSet = strategy.shouldBeMergedAndSet(leftLocator, rightLocator,
225         ((leftObject._return!= null) 2 &&!leftObject._return.isEmpty()), ((rightObject._return!= null) 3 &&
226         (!rightObject._return.isEmpty())));
227         4 if (_returnShouldBeMergedAndSet == Boolean.TRUE) {
228             List<String> lhsReturn;
229             lhsReturn = (((leftObject._return!= null) 5 &&!leftObject._return.isEmpty()) 6 ?
230             leftObject.getReturn():null);
231             List<String> rhsReturn;
232             rhsReturn = (((rightObject._return!= null) 7 &&!rightObject._return.isEmpty()) 8 ?
233             rightObject.getReturn():null);
234             List<String> mergedReturn = ((List<String> ) strategy.merge(LocatorUtils.property(leftLocator,

```

Otros ejemplos:

Refactor this method to reduce its Cognitive Complexity from 38 to the 15 allowed.

```

Devidabackend src/.../externalInterface/devida/models/Proyecto.java See all issues in this file
Refactor this method to reduce its Cognitive Complexity from 38 to the 15 allowed.

405     final Object draftCopy = ((target == null) 1 ?createNewInstance():target);
406     2 if (draftCopy instanceof Proyecto) {
407         final Proyecto copy = ((Proyecto) draftCopy);
408         {
409             Boolean anoEjeShouldBeCopiedAndSet = strategy.shouldBeCopiedAndSet(locator, (this.anoEje!= null));
410             3 if (anoEjeShouldBeCopiedAndSet == Boolean.TRUE) {
411                 Integer sourceAnoEje;
412                 sourceAnoEje = this.getAnoEje();
413                 Integer copyAnoEje = ((Integer) strategy.copy(LocatorUtils.property(locator, "anoEje", sourceAnoEje,
414                 sourceAnoEje, (this.anoEje!= null)));
415                 copy.setAnoEje(copyAnoEje);
416             } 4 else {
417                 5 if (anoEjeShouldBeCopiedAndSet == Boolean.FALSE) {

```

Refactor this method to reduce its Cognitive Complexity from 38 to the 15 allowed.

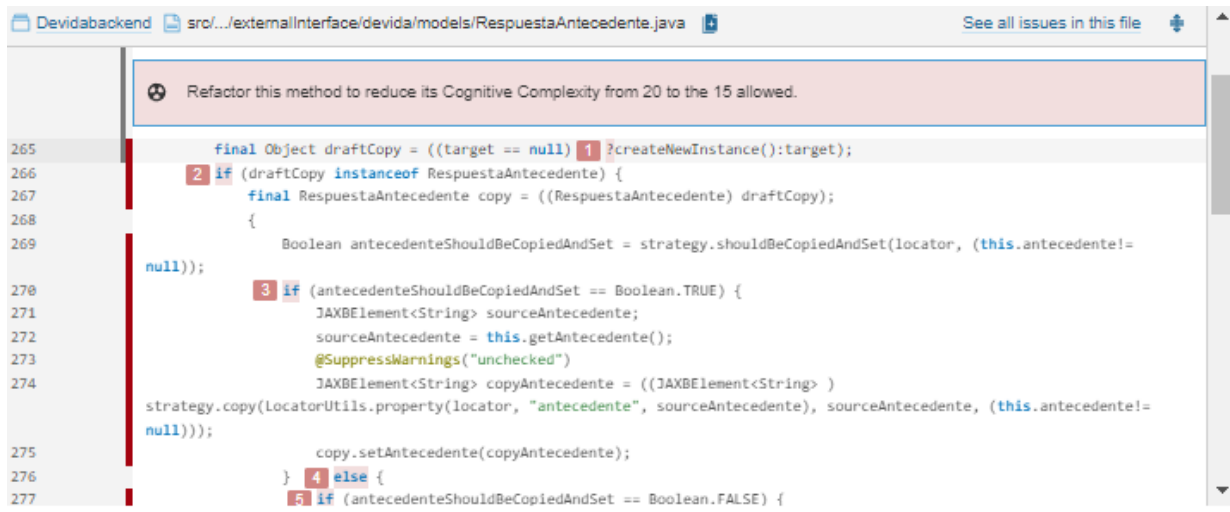
```

Devidabackend src/.../externalInterface/devida/models/ProyectoLocalizacion.java See all issues in this file
Refactor this method to reduce its Cognitive Complexity from 38 to the 15 allowed.

407     final Object draftCopy = ((target == null) 1 ?createNewInstance():target);
408     2 if (draftCopy instanceof ProyectoLocalizacion) {
409         final ProyectoLocalizacion copy = ((ProyectoLocalizacion) draftCopy);
410         {
411             Boolean centroPobladoShouldBeCopiedAndSet = strategy.shouldBeCopiedAndSet(locator, (this.centroPoblado!=
412             null));
413             3 if (centroPobladoShouldBeCopiedAndSet == Boolean.TRUE) {
414                 JAXBElement<String> sourceCentroPoblado;
415                 sourceCentroPoblado = this.getCentroPoblado();
416                 @SuppressWarnings("unchecked")
417                 JAXBElement<String> copyCentroPoblado = ((JAXBElement<String> )
418                 strategy.copy(LocatorUtils.property(locator, "centroPoblado", sourceCentroPoblado, sourceCentroPoblado,
419                 (this.centroPoblado!= null)));
420                 copy.setCentroPoblado(copyCentroPoblado);
421             } 4 else {
422                 5 if (centroPobladoShouldBeCopiedAndSet == Boolean.FALSE) {

```


Refactor this method to reduce its Cognitive Complexity from 20 to the 15 allowed.



```
Devidabackend src/.../externalInterface/devida/models/RespuestaAntecedente.java See all issues in this file

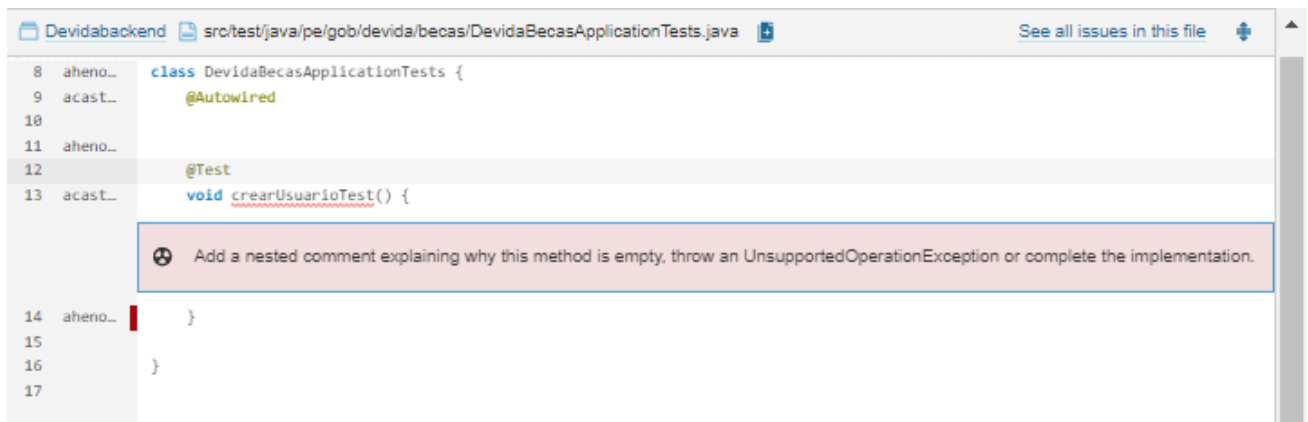
⊕ Refactor this method to reduce its Cognitive Complexity from 20 to the 15 allowed.

265 final Object draftCopy = ((target == null) ? createNewInstance():target);
266 if (draftCopy instanceof RespuestaAntecedente) {
267     final RespuestaAntecedente copy = ((RespuestaAntecedente) draftCopy);
268     {
269         Boolean antecedenteShouldBeCopiedAndSet = strategy.shouldBeCopiedAndSet(locator, (this.antecedente!=
270 null));
271         if (antecedenteShouldBeCopiedAndSet == Boolean.TRUE) {
272             JAXBElement<String> sourceAntecedente;
273             sourceAntecedente = this.getAntecedente();
274             @SuppressWarnings("unchecked")
275             JAXBElement<String> copyAntecedente = ((JAXBElement<String> )
276 strategy.copy(LocatorUtils.property(locator, "antecedente", sourceAntecedente), sourceAntecedente, (this.antecedente!=
277 null)));
278             copy.setAntecedente(copyAntecedente);
279         } else {
280             if (antecedenteShouldBeCopiedAndSet == Boolean.FALSE) {
```

Recommend file by sonarqube:

<https://www.sonarsource.com/docs/CognitiveComplexity.pdf>

✚ Add a nested comment explaining why this method is empty, throw an `UnsupportedOperationException` or complete the implementation



```
Devidabackend src/test/java/pe/gob/devida/becas/DevidaBecasApplicationTests.java See all issues in this file

8 aheno... class DevidaBecasApplicationTests {
9 acast...     @Autowired
10
11 aheno...
12 acast...     @Test
13 aheno...     void crearUsuarioTest() {
14
15
16
17

⊕ Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation.
```

Solución:

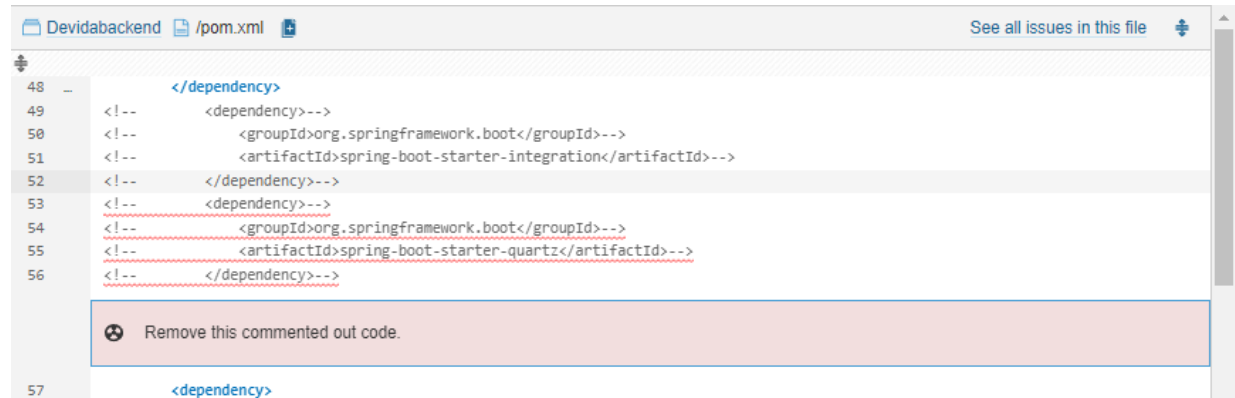
```
@Override
public void doSomething() {
    // Do nothing because of X and Y.
}

@Override
public void doSomethingElse() {
    throw new UnsupportedOperationException();
}
```

Code Smell

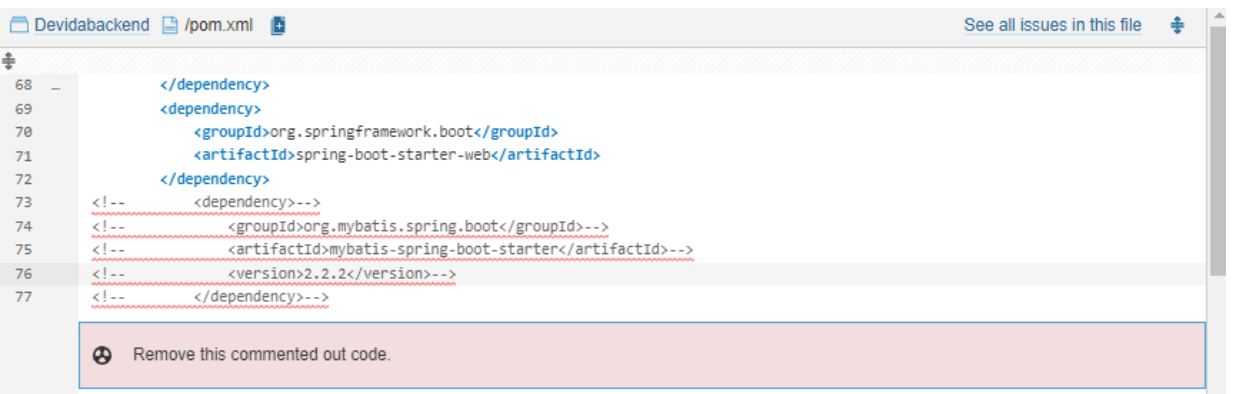
Comentarios de Código

Los programadores no deben comentar el código, ya que infla los programas y reduce la legibilidad. El código no utilizado debe eliminarse y puede recuperarse del historial de control de código fuente si es necesario.



```
48 -      </dependency>
49     <!--      <dependency>-->
50     <!--          <groupId>org.springframework.boot</groupId>-->
51     <!--          <artifactId>spring-boot-starter-integration</artifactId>-->
52     <!--      </dependency>-->
53     <!--      <dependency>-->
54     <!--          <groupId>org.springframework.boot</groupId>-->
55     <!--          <artifactId>spring-boot-starter-quartz</artifactId>-->
56     <!--      </dependency>-->
```

Remove this commented out code.



```
68 -      </dependency>
69     <dependency>
70         <groupId>org.springframework.boot</groupId>
71         <artifactId>spring-boot-starter-web</artifactId>
72     </dependency>
73     <!--      <dependency>-->
74     <!--          <groupId>org.mybatis.spring.boot</groupId>-->
75     <!--          <artifactId>mybatis-spring-boot-starter</artifactId>-->
76     <!--          <version>2.2.2</version>-->
77     <!--      </dependency>-->
```

Remove this commented out code.



```
139 -     <!--      <dependency>-->
140     <!--          <groupId>org.springframework.batch</groupId>-->
141     <!--          <artifactId>spring-batch-test</artifactId>-->
142     <!--          <scope>test</scope>-->
143     <!--      </dependency>-->
144     <!--      <dependency>-->
145     <!--          <groupId>org.springframework.integration</groupId>-->
146     <!--          <artifactId>spring-integration-test</artifactId>-->
147     <!--          <scope>test</scope>-->
148     <!--      </dependency>-->
```

Remove this commented out code.

Remove this unused "applicationContext" private field

Si se declara un campo privado pero no se usa en el programa, puede considerarse código muerto y, por lo tanto, debe eliminarse. Esto mejorará la capacidad de mantenimiento porque los desarrolladores no se preguntarán para qué se usa la variable.

```

Devidabackend src/main/java/pe/gob/devida/becas/applications/ThymeleafConfig.java
4 aheno... import org.springframework.context.annotation.Configuration;
5
6 @Configuration
7 public class ThymeleafConfig {
8
9     private final ApplicationContext applicationContext;

    Remove this unused "applicationContext" private field.

10
11     public ThymeleafConfig(ApplicationContext applicationContext) {
12         this.applicationContext = applicationContext;
13     }
14 }

```

1 duplicated blocks of code must be removed – REGLA DEPRECADA

```

Devidabackend src/.../pe/gob/devida/becas/domain/model/ConvocatoriaGridRow.java
1 acast... package pe.gob.devida.becas.domain.model;
2
3 rmore... import java.util.List;
4
5 acast... import lombok.*;
6         import lombok.experimental.FieldDefaults;
7
8     @Data
9     @Builder

```

Merge this if statement with the enclosing one

```

Devidabackend src/.../devida/becas/domain/usecase/postulante/PostulanteRegistrar.java
47 rmore...     if(idProvincia != null) {
48 rmore...         if(postulante.getIdProvincia().equals("0")) {
49 rmore...             postulante.setIdProvincia(null);
50 rmore...         }
51 rmore...     }
52 rmore...     if(idDistrito != null) {
53 rmore...         if(postulante.getIdDistrito().equals("0")) {
54 rmore...             postulante.setIdDistrito(null);

```

Solución:

Noncompliant Code Example

```
if (file != null) {
    if (file.isFile() || file.isDirectory()) {
        /* ... */
    }
}
```

Compliant Solution

```
if (file != null && isFileOrDirectory(file)) {
    /* ... */
}

private static boolean isFileOrDirectory(File file) {
    return file.isFile() || file.isDirectory();
}
```

🔧 Complete the task associated to this TODO comment

Las etiquetas TODO se usan comúnmente para marcar lugares donde se requiere más código, pero que el desarrollador desea implementar más adelante. A veces, el desarrollador no tendrá tiempo o simplemente se olvidará de volver a esa etiqueta.



The screenshot shows a code editor with the following code:

```
256 efact... byte[] white = null;
257 try {
258     primary = Hex.decodeHex("0d9ddb");
259     white = Hex.decodeHex("ffffff");
260 } catch (Exception e) {
261     // TODO: handle exception
262 }
263 XSSFCOLOR primaryColor = new XSSFCOLOR(primary, null);
264 XSSFCOLOR whiteColor = new XSSFCOLOR(white, null);
265 XSSFFONT headerCellFont = workbook.createFont();
266 headerCellFont.setColor(whiteColor);
```

A task popup is displayed over the TODO comment, stating: "Complete the task associated to this TODO comment."

Frontend Source

☆ **Devidafrontend** Passed Last analysis: 8 days ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
39 C	0 A	0.0% E	477 A	0.0% 	12.8% C	26k M TypeScript...

🔧 Unreachable code.

Las declaraciones de salto (return, break y continue) y las expresiones throw mueven el flujo de control fuera del bloque de código actual. Entonces, cualquier declaración que venga después de un salto es un código muerto.

```
197 acast... public esModificarMasivo(): boolean {
198     switch (this.opcion) {
199       case 'MODIFICAR_CERTIFICADO':
200         return true;
201         break;
202     case 'MODIFICAR_CERTIFICADO_MASIVO':
203       return true;
204       break;
205     }
206   }
207 }
```

Solución:

```
Noncompliant Code Example

function fun(a) {
  var i = 10;
  return i + a;
  i++;          // Noncompliant; this is never executed
}

Compliant Solution

function fun(int a) {
  var i = 10;
  return i + a;
}
```

🚦 Add a "title" attribute to this <iframe> tag

Los marcos permiten unir diferentes páginas web en un mismo espacio visual. Los usuarios sin discapacidades pueden escanear fácilmente el contenido de todos los marcos a la vez. Sin embargo, los usuarios con discapacidad visual que usan lectores de pantalla escuchan el contenido de la página de forma lineal

El atributo de title se utiliza para enumerar todos los marcos de la página, lo que permite a los usuarios navegar fácilmente entre ellos. Por lo tanto, las etiquetas <frame> y <iframe> siempre deben tener un atributo de título.

```

Devidafontend src/.../views/xs-portal/xs-contacto/xs-contacto-plugin/xs-contacto-plugin.component.html See all issues in this file
29 <section class="container" >
30 <h2<i class="pi pi-map-marker"></i> Av. Benavides 2199-8 Lima 18 - Perú</h2>
    ✖ Replace this <i> tag by <em>.
31 <iframe
32   src=
  "https://www.google.com/maps/embed?pb=!1m14!1m8!1m3!1d15603.111524213093!2d-77.0097213!3d-
  12.1273446!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x0%3A08xeb21314c29e7cc3a12sDevida!5e0!3m2!1ses!2spe!4v1654592292481!5m2!1ses!2spe"
33   height="450" style="border:0;" allowfullscreen="" loading="lazy"
34   refererpolicy="no-referrer-when-downgrade"></iframe>
    ✖ Add a "title" attribute to this <iframe> tag.
35 </section>

```

Solución:

Noncompliant Code Example

```

<frame src="index.php?p=menu"> <-- Non-Compliant -->
<frame src="index.php?p=home" name="contents"> <-- Non-Compliant -->

```

Compliant Solution

```

<frame src="index.php?p=menu" title="Navigation menu"> <-- Compliant -->
<frame src="index.php?p=home" title="Main content" name="contents"> <-- Compliant -->

```

Unexpected missing generic font family

Si ninguno de los nombres de fuente definidos en una declaración de fuente o familia de fuentes está disponible en el navegador del usuario, el navegador mostrará el texto usando su fuente predeterminada. Recomendación: Serif, Sans-serif, cursiva, fantasía, Monospace

```

Devidafontend src/.../views/xs-portal/xs-postulante/xs-postulante-registrar/xs-postulante-registrar.component.scss See all issues in this file
1 acast...
2 #textoAutorizacion{
3   font-size: 18px;
4   font-family: 'Sitka-Small';
    ✖ Unexpected missing generic font family
5
6   display: flex;
7   justify-content: center;
8   margin-bottom: 80px;
9 }
10 .options{
11   display: flex;
12   justify-content: center;

```

Solución:

Noncompliant Code Example

```
a {
  font-family: Helvetica, Arial, Verdana, Tahoma; /* Noncompliant; there is no generic font family in the list */
}
```

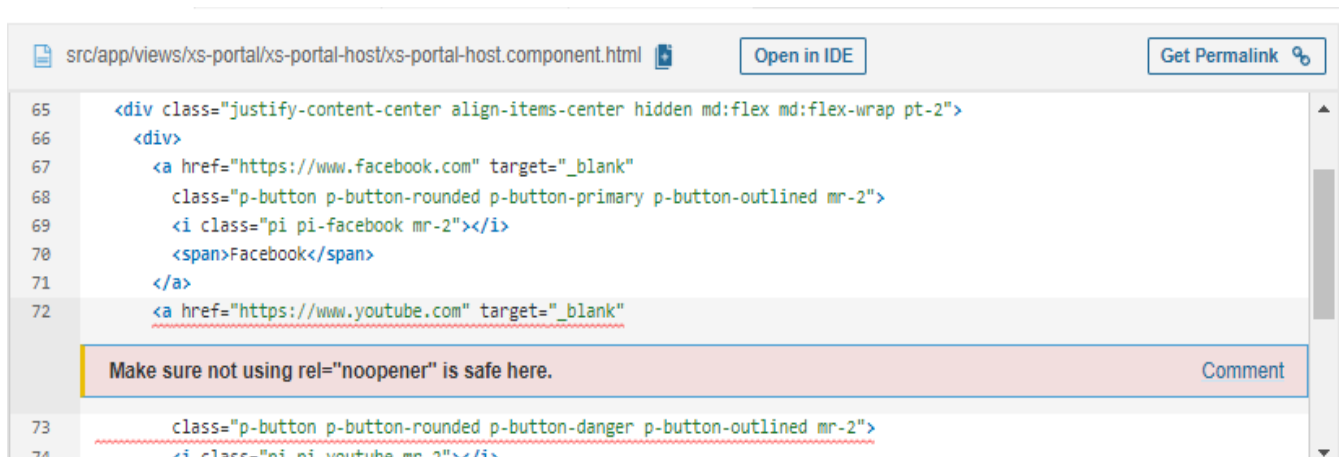
Compliant Solution

```
a {
  font-family: Helvetica, Arial, Verdana, Tahoma, sans-serif;
}
```

Make sure not using rel="noopener" is safe here

Una ventana recién abierta que tenga acceso a la ventana de origen podría permitir ataques de phishing básicos (el objeto window.opener no es nulo y, por lo tanto, la página abierta puede configurar window.opener.location como un sitio web malicioso).

** En Chrome 88+, Firefox 79+ o Safari 12.1+ target=_blank en las anclas implica rel=noopener que activa la protección de forma predeterminada*



```
src/app/views/xs-portal/xs-portal-host/xs-portal-host.component.html Open in IDE Get Permalink
65 <div class="justify-content-center align-items-center hidden md:flex md:flex-wrap pt-2">
66   <div>
67     <a href="https://www.facebook.com" target="_blank"
68       class="p-button p-button-rounded p-button-primary p-button-outlined mr-2">
69       <i class="pi pi-facebook mr-2"></i>
70       <span>Facebook</span>
71     </a>
72     <a href="https://www.youtube.com" target="_blank"
73       class="p-button p-button-rounded p-button-danger p-button-outlined mr-2">
74     <i class="ni ni-youtube mr-2"></i>
```

Make sure not using rel="noopener" is safe here. Comment

Solución:

Para evitar que las páginas abusen de window.opener, use rel=noopener en para forzar que su valor sea nulo en las páginas abiertas.

```
<a href="http://petsocialnetwork.io" target="_blank" rel="noopener"> <!-- Compliant -->
```

Unexpected empty source

Esta regla plantea un problema cuando un archivo CSS está vacío (es decir, contiene solo espacios).

Devidafontend src/app/app.component.scss See all issues in this file

```
1 -
```

Unexpected empty source

Devidafontend src/app/shared/xs-prime-components/xs-prime-checkbox/xs-prime-checkbox.component.scss See all issues in this file

```
1 -
```

Unexpected empty source

Devidafontend src/app/shared/xs-prime-components/xs-prime-input-password/xs-prime-input-password.component.scss See all issues in this file

```
1 -
```

Unexpected empty source

Remove this commented out code

Los programadores no deben comentar el código, ya que infla los programas y reduce la legibilidad. El código no utilizado debe eliminarse y puede recuperarse del historial de control de código fuente si es necesario.

Devidafontend src/app/driven-adapters/api/acceso-directo.service.ts See all issues in this file

```
54 return lastValueFrom(this.httpClient.post<boolean>(  
55   this.endpoint + 'acceso-directo/eliminar',  
56   { params: <{ idAccesoDirecto?: number, auditoriaEliminadoPor?: string }>parametro }  
57 ));  
58 }  
59  
60 eliminarMultiple(parametro: AccesoDirectoGridRowModel[]): Promise<boolean> {  
61   parametro.forEach((item: AccesoDirectoGridRowModel) => {  
62     // item.auditoriaEliminadoPor = this.currentUser;  
63   });  
64   return lastValueFrom(this.httpClient.post<boolean>(  
65     this.endpoint + 'acceso-directo/eliminar-multiple',  
66     parametro
```

Remove this commented out code.


```
Devidafrontend src/app/driven-adapters/api/curso.service.ts See all issues in this file
80 rmore...   });
81           }
82
83 rmore...   eliminarMultiple(parametro: CursoGridRowModel[]): Promise<boolean> {
84           parametro.forEach((item: CursoGridRowModel) => {
85             // item.auditoriaEliminadoPor = this.currentUser;
86
87           });
88           return lastValueFrom(this.httpClient.post<boolean>(
89             this.endpoint + 'curso/eliminar-multiple',
90             parametro
91           ));

```

```
Devidafrontend src/app/driven-adapters/api/gestor.service.ts See all issues in this file
66           }
67
68 rmore...   eliminarMultiple(parametro: GestorGridRowModel[]): Promise<boolean> {
69           parametro.forEach((item: GestorGridRowModel) => {
70             // item.auditoriaEliminadoPor = this.currentUser;
71
72           });
73           return lastValueFrom(this.httpClient.post<boolean>(
74             this.endpoint + 'gestor/eliminar-multiple',
75             parametro
76           ));
77           }

```

```
Devidafrontend src/app/driven-adapters/api/video.service.ts See all issues in this file
43           }
44
45           eliminarMultiple(parametro: VideoModel[]): Promise<boolean> {
46           parametro.forEach((item: VideoModel) => {
47             // item.auditoriaEliminadoPor = this.currentUser;
48
49           });
50           return lastValueFrom(this.httpClient.post<boolean>(
51             this.endpoint + 'video/eliminar-multiple',
52             parametro
53           ));
54           }

```

 Remove this unused import of 'UtilityFunctions'.

No hay razón para importar módulos que no usas. Hacerlo aumenta innecesariamente la carga.

```
Devidafontend src/app/driven-adapters/api/evaluacion.service.ts See all issues in this file
```

```
2 import { Injectable } from '@angular/core';
3 import { lastValueFrom } from 'rxjs';
4 import { EvaluacionModel } from 'src/app/models/EvaluacionModel';
5 import { environment } from 'src/app/environments/environment';
6 import { UtilityFunctions } from 'src/app/shared/shared.utility';
```

Remove this unused import of 'UtilityFunctions'.

```
7
8 @Injectable({
9   providedIn: 'root'
10 })
11 export class EvaluacionService {
12   endpoint = environment.API_URL;
13 }
```

```
Devidafontend src/app/driven-adapters/api/matricula.service.ts See all issues in this file
```

```
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3 import { lastValueFrom } from 'rxjs';
```

Remove this unused import of 'lastValueFrom'.

```
4 import { MatriculaInicializarBusquedaDto } from 'src/app/models/dto/MatriculaInicializarBusquedaDto';
```

Remove this unused import of 'MatriculaInicializarBusquedaDto'.

```
5 import { environment } from 'src/app/environments/environment';
6 import { UtilityFunctions } from 'src/app/shared/shared.utility';
7
8
9 @Injectable({
```

```
Devidafontend src/app/driven-adapters/api/matricula.service.ts See all issues in this file
```

```
3 import { lastValueFrom } from 'rxjs';
```

Remove this unused import of 'lastValueFrom'.

```
4 import { MatriculaInicializarBusquedaDto } from 'src/app/models/dto/MatriculaInicializarBusquedaDto';
```

Remove this unused import of 'MatriculaInicializarBusquedaDto'.

```
5 import { environment } from 'src/app/environments/environment';
6 import { UtilityFunctions } from 'src/app/shared/shared.utility';
7
8
9 @Injectable({
10   providedIn: 'root'
11 })
```

```

Devidafrontend src/app/driven-adapters/api/public/solicitud-beca-public.service.ts
2 import { Injectable } from '@angular/core';
3 import { lastValueFrom } from 'rxjs';
4 import { InscribirPostulanteDto } from 'src/app/models/dto/InscribirPostulanteDto';
5 import { SolicitudBecaDtoInicializarDto } from 'src/app/models/dto/SolicitudBecaDtoInicializarDto';
6 import { SolicitudBecaPostulantePostDto } from 'src/app/models/dto/SolicitudBecaPostulantePostDto';

Remove this unused import of 'SolicitudBecaPostulantePostDto'.

7 rmore... import { FichaPostulanteGenerarModel } from 'src/app/models/FichaPostulanteGenerarModel';

Remove this unused import of 'FichaPostulanteGenerarModel'.

```

Unexpected empty constructor

```

Devidafrontend src/app/driven-adapters/session-storage.service.ts
8 ahen... providedIn: 'root',
9 })
10 export class SessionStorageService {
11   private encryptStorage = new EncryptStorage(environment.SECRET_KEY, {storageType: "sessionStorage", encAlgorithm:
    "Rabbit", prefix: "@SCBG"});
12
13   constructor() {
14     ...
15
Unexpected empty constructor.

```

Expected:

```

function foo() {
  // This is intentional
}

var foo = () => {
  do_something();
};

```

Unexpected var, use let or const instead

La distinción entre los tipos de variables creados por var y let es significativa, y un cambio a let ayudará a aliviar muchos de los problemas de alcance de variables

```

Devidafrontend src/app/shared/shared.utility.ts
135 }
136
137 rmore... downloadBlobFile(blob: Blob, type: string) {
138   var _blob = new Blob([blob], { type: type });

Unexpected var, use let or const instead.

139   var link = document.createElement('a');


Unexpected var, use let or const instead.

140   link.href = window.URL.createObjectURL(_blob);
141   link.target = '_blank';
142   // Link.download = 'test.pdf';

```

🚩 'blob' is already defined.

Esta regla verifica que una declaración no use un nombre que ya está en uso. De hecho, es posible usar el mismo símbolo varias veces como una variable o una función, pero hacerlo probablemente confunda a los mantenedores. Además, es posible que dichas reasignaciones se realicen por error y que el desarrollador no se dé cuenta de que la nueva asignación sobrescribe el valor de la variable.

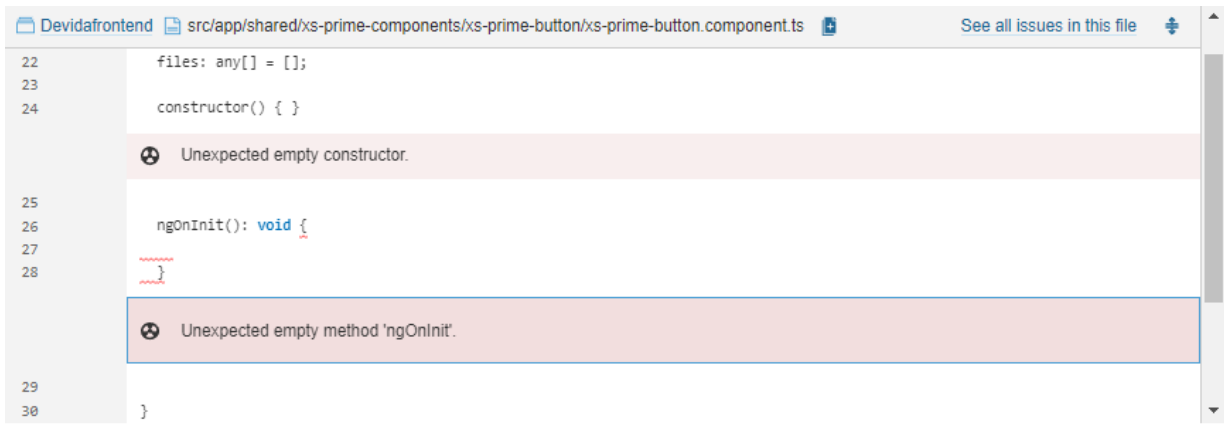


```
127 rmore... downloadExcelBlob(blob: Blob, filename: string) {
128           var blob = new Blob([blob], { type: 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet' });
129
130           var link = document.createElement('a');
```

The screenshot shows a code editor with a TypeScript file. A function `downloadExcelBlob` is defined. On line 129, the variable `blob` is redeclared with `var blob = new Blob(...)`. A red arrow points to this line, and another red arrow points to the error message: `'blob' is already defined.` Other error messages visible are `Unexpected var, use let or const instead.`

🚩 Unexpected empty method 'ngOnInit'.

Una omisión involuntaria y debe corregirse para evitar un comportamiento inesperado.



```
22 files: any[] = [];
23
24 constructor() { }
25
26 ngOnInit(): void {
27
28 }
29
30 }
```

The screenshot shows a code editor with a TypeScript file. A class has a constructor and a `ngOnInit` method. The `ngOnInit` method is empty. A red arrow points to the `ngOnInit` method signature, and another red arrow points to the error message: `Unexpected empty method 'ngOnInit'.`

🚩 Consider moving declaration of 'key' as it is referenced outside current binding context

Las variables que se declaran dentro de un bloque pero se usan fuera de él (lo cual es posible con una declaración de estilo `var`) deben declararse fuera del bloque.

```

32 rmore...   onPress(e: any) {
33             if (this.onlyLetters) {
34                 var key = String.fromCharCode(e.keyCode);
35
36                 if (/[a-zA-ZæiούΆΈΙΟϞΝ ]/.test(key)) {
37                     return true;
38                 } else {
39                     e.preventDefault();
40                     return false;
41                 }
42 rmore...   }
43             if (this.onlyNumbers) {
44                 var key = String.fromCharCode(e.keyCode);
45
46                 if (/[0-9]/.test(key)) {
47                     return true;
48                 } else {
49                     e.preventDefault();
50                     return false;
51                 }
52             }
53         }
54     }
55 }

```

Expected:

```

function doSomething(a, b) {
    var x;

    if (a > b) {
        x = a - b;
    }

    if (a > 4) {
        console.log(x);
    }

    for (let i = 0; i < m; i++) {
    }
}

```

🛠️ Replace this union type with a type alias

Los tipos de unión e intersección son convenientes pero pueden dificultar la lectura y el mantenimiento del código. Por lo tanto, si se usa una unión o intersección en particular en varios lugares, se recomienda el uso de un alias de tipo.

```

6 rmore...   styleUrls: [ './xs-section-title.component.scss' ]
7   })
8   export class XsSectionTitleComponent implements OnInit {
9
10 rmore...   @Input() title: string = '';
11   @Input() borderLeftStyle: 'primary' | 'success' | 'white' = 'primary';
12
13   @Input() borderBottomStyle: 'primary' | 'success' | 'white' = 'primary';
14   @Input() backgroundStyle: 'primary' | 'success' | 'white' = 'primary';
15 rmore...   constructor() { }

```

🛠️ Unexpected empty block

Los bloques vacíos sobrantes generalmente se introducen por error. Son inútiles y evitan la legibilidad del código. Deben eliminarse o completarse con código real.

```
Devidafontend src/app/views/xs-admin/xs-admin-host/xs-admin-host.component.scss See all issues in this file
234 rmore...
235 @media screen and (min-width: 769px) and (max-width: 1024px) {}
Unexpected empty block
236
237 @media screen and (min-width: 1025px) and (max-width: 1200px) {}
Unexpected empty block
238
239 @media screen and (min-width: 1201px) {}
Unexpected empty block
```

Unnecessary semicolon

```
Devidafontend src/app/views/xs-admin/xs-firmante/xs-firmante-registrar/xs-firmante-registrar.component.ts See all issues in this file
239 you@e... if (this.dni?.value) {
240 rmore... switch (this.idTipoDocumento?.value) {
241 you@e... case this.ID_TIPO_DOCUMENTO_DNI:
242 if (this.dni?.value.toString().length != 8) {
243 return { required: this.MENSAJES.VALIDACION.DNI_8_DIGITOS }
}; break;
Unexpected semicolon.
244 rmore... case this.ID_TIPO_DOCUMENTO_CARNE:
245 you@e... if (this.dni?.value.toString().length != 12) {
246 return { required: this.MENSAJES.VALIDACION.CARNET_12_DIGITOS }
}; break;
Unexpected semicolon.
```

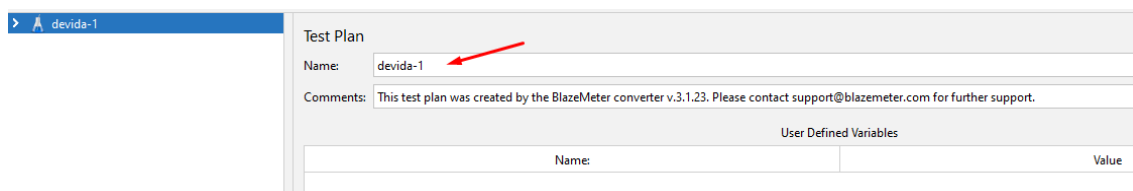
PERFORMANCE

PERFIL: ADMIN/FIRMANTE/OTRO ROL DEVIDA

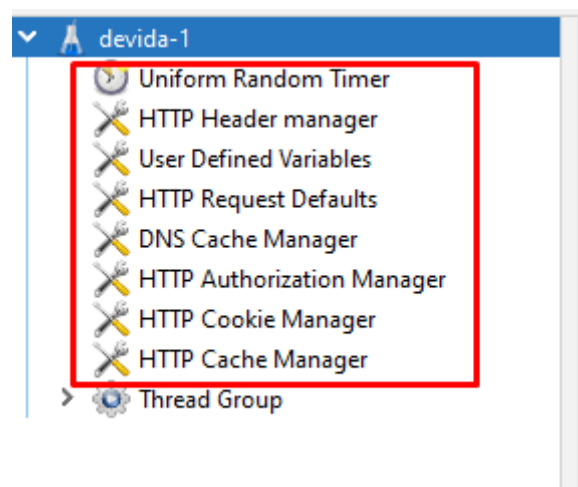
TEST SUITE

La suite para este caso fue un recorrido por todo el portal con el perfil de ADMIN, teniendo como objetivo medir el grado de respuesta del portal con respecto a performance.

Nombre Test Plan:



Config Elements:



Para la configuración se tuvo en cuenta con:

- Random Timer que nos ayuda a que las peticiones se realicen en un tiempo simulado a la realidad (para este caso un intervalo de 0 a 10 segundos por cada petición).
- HTTP Config: Elementos como los headers, request default, authorization, cookie y cache, los cuales permiten realizar las peticiones de forma correcta mediante el protocolo mencionado. Algunos ejemplos son:

Header Managers:

HTTP Header Manager

Name:

Comments:

Headers Stored in the Header Manager

Name:	Value
Accept	application/json, text/plain, */*
Content-Type	application/json

HTTP Header Manager

Name:

Comments:

Headers Stored in the Header Manager

Name:	Value
sec-ch-ua	"Google Chrome";v="105", "Not)A;Brand";v="8", "Chromium";v="105"
sec-ch-ua-mobile	?0
sec-ch-ua-platform	"Windows"
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36

User defined variables:

User Defined Variables

Name:

Comments:

User Defined Variables

Name:	Value
BASE_URL_1	tomcat.expertsystems.pe

DNS Cache Manager

DNS Cache Manager

Name:

Comments:

Options

Clear cache each iteration

Use system DNS resolver Use custom DNS resolver

DNS Servers

HTTP Cache manager

HTTP Cache Manager

Name:

Comments:

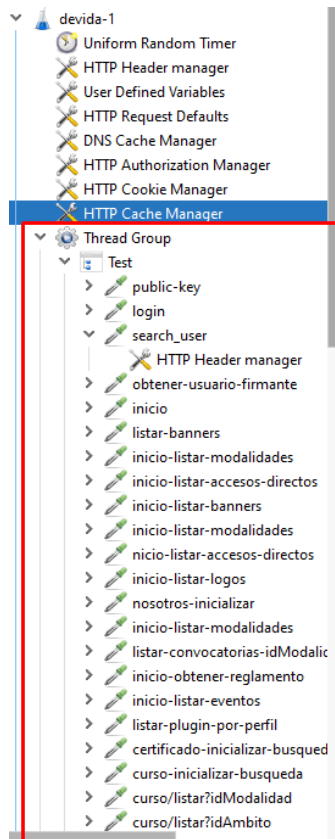
Clear cache each iteration?

Use Thread Group configuration to control cache clearing

Use Cache-Control/Expires header when processing GET requests

Max Number of elements in cache

Las peticiones se realizaron en un Grupo de Hilos:



Configuración del Thread Group

Thread Group

Name:

Comments:

Action to be taken after a Sampler error

Continue
 Start Next Thread Loop
 Stop Thread
 Stop Test
 Stop Test Now

Thread Properties

Number of Threads (users):

Ramp-up period (seconds):

Loop Count: Infinite

Same user on each iteration
 Delay Thread creation until needed
 Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

PERFIL: POSTULANTE

Label	# Sampl...	Average	Min	Max	Std. Dev.	Error %	Throughput	Received ...	Sent KB/sec	Avg. Bytes
/inicio/listar-modalidades	400	630	186	2500	399.85	0.00%	18.8/min	0.54	0.28	1770.3
/inicio/listar-accesos-directos	300	686	185	2350	433.28	0.00%	14.4/min	0.18	0.21	755.0
/inicio/listar-banners	200	578	189	1886	361.61	0.00%	20.1/min	0.22	0.30	685.0
/inicio/listar-logos	200	673	188	2524	441.47	0.00%	11.0/min	0.19	0.16	1077.0
/certificado/inicializar-busqueda-alumno?idPostula...	200	681	157	2439	429.77	0.50%	7.0/min	0.07	0.10	569.1
/solicitudBeca/listarConvocatoriascritos?anio=20...	200	699	190	3774	494.61	0.50%	11.7/min	0.48	0.18	2542.5
login	100	1374	727	3060	510.69	0.00%	42.6/min	0.67	0.34	965.0
buscarIdPostulante?codUsuario=c70213952	100	533	191	1553	296.35	0.00%	27.0/min	0.21	0.41	484.0
obtener-por-usuario?usuario=c70213952	100	550	190	1599	314.40	0.00%	21.3/min	0.15	0.32	446.0
/inicio	100	2880	1907	7002	1070.94	0.00%	18.7/min	851.80	1.13	2797793.0
/nosotros/inicializar	100	692	188	2870	478.17	0.00%	10.4/min	0.27	0.15	1595.1
/convocatoria/listar-convocatorias-portal?idModali...	100	688	208	1737	354.56	0.00%	9.2/min	0.59	0.14	3931.1
/inicio/obtener-reglamento	100	652	200	2960	463.57	0.00%	9.9/min	0.19	0.15	1147.0
/inicio/listar-banners	100	599	188	2022	397.97	0.00%	9.9/min	0.11	0.14	685.0
/inicio/plugin-por-perfil	100	673	203	3758	518.09	0.00%	7.5/min	0.21	0.12	1754.3
https://tomcat.expertsystems.pe/becas-backend/d/...	100	744	187	2260	483.09	0.00%	7.1/min	0.06	0.11	485.0
/provincia/listar?idDepartamento=02	100	731	195	3749	543.40	0.00%	7.0/min	0.68	0.10	5994.1
/distrito/listar?idProvincia=0201	100	710	187	2360	429.35	1.00%	7.4/min	0.49	0.11	4054.0
/postulante/inicializar?idPostulante=1	100	1431	704	6608	709.90	0.00%	7.2/min	13.02	0.11	111637.6
/provincia/listar?idDepartamento=15	100	679	186	2331	449.65	0.00%	6.9/min	0.36	0.10	3198.9
/distrito/listar?idProvincia=1501	100	887	204	2735	477.66	1.00%	6.5/min	1.43	0.10	13441.0
/sustento-postulante/listar?idPostulante=1	100	671	195	2024	412.38	0.00%	7.0/min	0.13	0.10	1168.0
/solicitudBeca/inicializar-busqueda?idPostulante=1	100	648	192	1935	432.18	1.00%	6.9/min	0.06	0.10	554.4

- devida-postulante
 - Uniform Random Timer
 - Counter
 - HTTP Header manager
 - User Defined Variables
 - HTTP Request Defaults
 - DNS Cache Manager
 - HTTP Authorization Manager
 - HTTP Cookie Manager
 - HTTP Cache Manager
 - Thread Group
 - Test
 - login
 - buscarIdPostulante?codUsuario=c70213952
 - obtener-por-usuario?usuario=c70213952
 - /inicio
 - /inicio/listar-banners
 - /inicio/listar-modalidades
 - /inicio/listar-accesos-directos
 - /inicio/listar-banners
 - /inicio/listar-modalidades
 - /inicio/listar-accesos-directos
 - /inicio/listar-logos
 - /nosotros/inicializar
 - /inicio/listar-modalidades
 - /convocatoria/listar-convocatorias-portal?idModalidad=1
 - /inicio/obtener-reglamento
 - inicio/listar-banners
 - /inicio/listar-modalidades
 - /inicio/listar-accesos-directos
 - /inicio/listar-logos
 - /listar-plugin-por-perfil
 - /certificado/inicializar-busqueda-alumno?idPostulante=1
 - https://tomcat.expertsystems.pe/becas-backend/d/...

CASO 1

- Cuando se tienen 5 usuarios y cada uno va ingresando 20 segundos después desde que ingresó el primer usuario y un intervalo de tiempo que nos ayuda a que las peticiones se realicen en un tiempo simulado a la realidad (para este caso un intervalo de 0 a 10 segundos por cada petición).

Resultados:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
public-key	5	2106	1231	3333	832.91	0.00%	0.38932	0.31	0.15	810
login	5	494	300	666	149.09	0.00%	0.49885	0.48	0.24	994
search_user	5	541	292	861	190.13	0.00%	0.55922	0.24	0.53	446
obtener-usuario-firmante	5	557	209	950	297.92	0.00%	0.80502	0.35	0.75	446
inicio	5	2899	1993	5362	1241.61	0.00%	0.35956	982.37	1.3	2797731
listar-banners	5	219	202	230	11.31	0.00%	1.0513	0.7	0.97	685
inicio-listar-modalidades	15	342	206	556	95.63	0.00%	0.26159	0.45	0.24	1770
inicio-listar-accesos-directos	5	327	275	374	34.36	0.00%	0.41639	0.31	0.39	755
inicio-listar-banners	5	372	267	625	128.97	0.00%	0.3916	0.26	0.36	685
inicio-listar-accesos-directos	5	417	282	614	117.3	0.00%	0.201	0.15	0.19	755
inicio-listar-logos	5	450	319	606	92.97	0.00%	0.17765	0.19	0.16	1077
nosotros-inicializar	5	447	310	621	138.59	0.00%	0.18707	0.29	0.17	1596.2
listar-convocatorias-idModalidad=1	5	754	442	1199	306.13	0.00%	0.15193	0.58	0.14	3931.4
inicio-obtener-reglamento	5	586	214	1298	370.87	0.00%	0.12441	0.14	0.12	1147
inicio-listar-eventos	5	495	243	776	208.08	0.00%	0.11944	0.06	0.11	485
listar-plugin-por-perfil	5	498	365	760	142.83	0.00%	0.11406	0.55	0.12	4926.4
certificado-inicializar-busqueda	20	625	219	2170	489.61	0.00%	0.09574	0.06	0.09	598
curso-inicializar-busqueda	5	811	292	1596	474.38	0.00%	0.13347	0.27	0.12	2075
curso/listar?idModalidad	5	502	342	718	133.71	0.00%	0.14021	0.85	0.13	6196
curso/listar?idAmbito	5	556	315	764	158.14	0.00%	0.13726	0.2	0.13	1498
curso/inicializar-registro?idCurso	5	843	713	1170	168.31	0.00%	0.1188	6.19	0.11	53348
convocatoria-inicializar-busqueda	5	632	340	1192	313.84	0.00%	0.10221	0.63	0.09	6326.6
convocatoria-listar?idModalidad=0&idAmbito=0&idEs	5	993	784	1415	219.08	0.00%	0.09612	7.44	0.09	79256
convocatoria-inicializar-registro?idConvocatoria=85	5	962	274	1307	378.44	0.00%	0.09565	1.95	0.09	20833.6
convocatoria-inicializar-registro?idConvocatoria=84	5	1103	573	1592	401.95	0.00%	0.10125	2.07	0.1	20910.2
convocatoria-inicializar-registro?idConvocatoria=83	5	1120	790	1551	290.97	0.00%	0.09328	1.5	0.09	16516.8
convocatoria-inicializar-registro?idConvocatoria=81	5	795	422	1210	265.7	0.00%	0.09979	2.21	0.09	22633

- **/INICIO:** Al ingresar al portal y cuando se realiza la petición de portal/inicio se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 1.993 segundos, un promedio de 2.899 segundos y un máximo de 5.362 segundos.
- **CURSO/INICIALIZAR-BUSQUEDA:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.292 segundos, un promedio de 0.811 segundos y un máximo de 1.596 segundos.
- **CONVOCATORIA/inicializar-registro?idConvocatoria=85:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.274 segundos, un promedio de 0.962 segundos y un máximo de 1.307 segundos.
- **CONVOCATORIA/inicializar-registro?idConvocatoria=84:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.573 segundos, un promedio de 1.103 segundos y un máximo de 1.592 segundos.

- **CONVOCATORIA/inicializar-registro?idConvocatoria=83:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.790 segundos, un promedio de 1.120 segundos y un máximo de 1.551 segundos.
- **CONVOCATORIA/inicializar-registro?idConvocatoria=81:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.422 segundos, un promedio de 0.795 segundos y un máximo de 1.210 segundos.
- **REPORTES/grafico/inicializar:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.293 segundos, un promedio de 1.020 segundos y un máximo de 1.615 segundos

CASO 2:

Cuando se tienen 15 usuarios y cada uno va ingresando 20 segundos después desde que ingresó el primer usuario.

Resultados:

Label	# Sample	Average	Min	Max	Std. Dev	Error %	Throughpt	Received KB/s	Sent KB/s	Avg. Byte
login	15	1515	256	7312	2140.29	0.00%	0.43858	0.43	0.21	994
search_user	15	2082	202	9245	2776.75	0.00%	0.37968	0.17	0.36	446
obtener-usuario-firmante	15	482	183	4064	959.03	0.00%	0.35765	0.16	0.33	446
inicio	15	3445	1993	7914	1924.52	0.00%	0.32997	901.54	1.19	2797731
listar-banners	15	255	187	768	137.48	0.00%	0.34415	0.23	0.32	685
inicio-listar-modalidades	45	275	190	835	151.58	0.00%	0.50515	0.87	0.47	1770.3
inicio-listar-accesos-directos	15	208	186	232	14.13	0.00%	0.29296	0.22	0.27	755
inicio-listar-banners	15	700	187	7614	1821.27	0.00%	0.28864	0.19	0.27	685
inicio-listar-accesos-directos	15	244	189	509	84.36	0.00%	0.37908	0.28	0.35	755
inicio-listar-logos	15	225	191	455	62.32	0.00%	0.35368	0.37	0.33	1077
nosotros-inicializar	15	213	192	235	11.18	0.00%	0.33929	0.53	0.31	1595
listar-convocatorias-idModalidad=1	15	261	215	556	79.79	0.00%	0.35095	1.35	0.33	3930.5
inicio-obtener-reglamento	15	333	192	612	167.33	0.00%	0.33966	0.38	0.32	1147
inicio-listar-eventos	15	325	192	814	187.7	0.00%	0.32575	0.15	0.3	485
listar-plugin-por-perfil	15	272	211	796	141.66	0.00%	0.30503	1.47	0.31	4925
certificado-inicializar-busqueda	60	281	188	1632	205.49	0.00%	0.24895	0.15	0.23	598
curso-inicializar-busqueda	15	242	189	565	89.16	0.00%	0.31264	0.63	0.29	2075
curso/listar?idModalidad	15	229	192	448	60.04	0.00%	0.26275	1.71	0.27	6197.6
curso/listar?idAmbito	15	254	188	564	94.24	0.00%	0.24786	0.36	0.23	1498.3
curso/inicializar-registro?idCurso	15	657	525	944	130.28	0.00%	0.25241	13.15	0.23	53335.3
convocatoria-inicializar-busqueda	15	268	186	832	152.24	0.00%	0.26355	1.63	0.24	6318.5
convocatoria-listar?idModalidad=0&idAmbito=0&idEstado=	15	721	639	987	106.04	0.00%	0.2815	21.78	0.27	79240.2
convocatoria-inicializar-registro?idConvocatoria=85	15	3057	286	19245	6352.61	13.33%	0.18986	3.42	0.16	18470.7
convocatoria-inicializar-registro?idConvocatoria=84	15	3195	423	20291	6511.97	13.33%	0.18226	3.3	0.15	18537.9
convocatoria-inicializar-registro?idConvocatoria=83	15	579	178	1819	389.17	6.67%	0.19071	2.91	0.17	15609.3
convocatoria-inicializar-registro?idConvocatoria=81	15	3034	391	19256	6363.79	13.33%	0.19251	3.77	0.16	20028.1
tutor-inicializar-busqueda	30	996	180	19245	3399.18	6.67%	0.25816	0.31	0.22	1248.4
tutor-listar?idEstado=0	15	3080	196	22261	6967.01	6.67%	0.18887	0.67	0.16	3616.7
tutor-inicializar-registro?idTutor=41	15	1579	180	19254	4725.93	13.33%	0.1855	0.56	0.15	3109.6
firmante-inicializar-busqueda	15	2869	198	19251	6429.71	13.33%	0.18035	0.25	0.14	1392.3
firmante-listar?idEstado=0	15	354	196	575	153.75	0.00%	0.17146	0.28	0.16	1695.8

- **/LOGIN:** Al ingresar al portal y cuando se realiza el login se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.256 segundos, un promedio de 1.515 segundos y un máximo de 7.312 segundos.

- **/INICIO:** Al ingresar al portal, en inicio se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 1.993 segundos, un promedio de 3.445 segundos y un máximo de 7.913 segundos.
- **CONVOCATORIA/inicializar-registro?idConvocatoria=85:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.286 segundos, un promedio de 3.057 segundos y un máximo de 19.249 segundos.
- **CONVOCATORIA/inicializar-registro?idConvocatoria=84:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.423 segundos, un promedio de 3.195 segundos y un máximo de 20.281 segundos.
- **CONVOCATORIA/inicializar-registro?idConvocatoria=81:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.391 segundos, un promedio de 3.034 segundos y un máximo de 19.256 segundos.
- **TUTOR-LISTAR?IDESTADO=0:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.196 segundos, un promedio de 3.080 segundos y un máximo de 22.261 segundos.
- **TUTOR-INICIALIZAR-REGISTRO?IDTUTOR=41:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.180 segundos, un promedio de 1.579 segundos y un máximo de 19.254 segundos.
- **FIRMANTE-INICIALIZAR-REGISTRO?IDFIRMANTE=3:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.197 segundos, un promedio de 2.830 segundos y un máximo de 20.185 segundos.
- **GESTOR-LISTAR?IDESTADO=0:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.196 segundos, un promedio de 1.495 segundos y un máximo de 19.255 segundos.

PERFIL: POSTULANTE

CASO 1

Se tuvo en cuenta a 100 postulantes realizando acciones dentro del portal teniendo como ingreso a cada 50 segundos después de que ingresó el primero postulante y un intervalo de

tiempo que nos ayuda a que las peticiones se realicen en un tiempo simulado a la realidad (para este caso un intervalo de 0 a 120 segundos por cada petición).

Resultados:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
login	100	1374	727	3060	510.69	0.00%	0.71074	0.67	0.34	965
buscarIdPostulante?codUsuario=c70213952	100	533	191	1553	298.35	0.00%	0.45038	0.21	0.41	484
obtener-por-usuario?usuario=c70213952	100	550	190	1599	314.4	0.00%	0.35483	0.15	0.32	446
/inicio	100	2880	1907	7002	1070.94	0.00%	0.31176	851.8	1.13	2797793
/inicio/listar-banners	200	578	189	1886	361.61	0.00%	0.33519	0.22	0.3	685
/inicio/listar-modalidades	400	630	186	2500	399.85	0.00%	0.31343	0.54	0.28	1770.3
/inicio/listar-accesos-directos	300	686	185	2350	433.28	0.00%	0.24042	0.18	0.21	755
/inicio/listar-logos	200	673	188	2524	441.47	0.00%	0.18311	0.19	0.16	1077
/nosotros/inicializar	100	692	188	2870	478.17	0.00%	0.17349	0.27	0.15	1595.1
/convocatoria/listar-convocatorias-portal?idModalidad=1	100	688	208	1737	354.56	0.00%	0.15401	0.59	0.14	3931.1
/inicio/obtener-reglamento	100	652	200	2960	463.57	0.00%	0.16576	0.19	0.15	1147
inicio/listar-banners	100	599	188	2022	397.97	0.00%	0.16426	0.11	0.14	685
/listar-plugin-por-perfil	100	673	203	3768	518.09	0.00%	0.12424	0.21	0.12	1754.3
/certificado/inicializar-busqueda-alumno?idPostulante=1	200	681	157	2439	429.77	0.50%	0.11705	0.07	0.1	569.1
https://tomcat.expertsystems.pe/becas-backend/devida/api/certificado/lis	100	744	187	2260	483.09	0.00%	0.11777	0.06	0.11	485
/provincia/listar?idDepartamento=02	100	731	195	3749	543.4	0.00%	0.11655	0.68	0.1	5994.1
/distrito/listar?idProvincia=0201	100	710	187	2360	429.35	1.00%	0.1227	0.49	0.11	4054
/postulante/inicializar?idPostulante=1	100	1431	704	6608	709.9	0.00%	0.11944	13.02	0.11	111637.6
/provincia/listar?idDepartamento=15	100	679	186	2331	449.65	0.00%	0.11496	0.36	0.1	3198.9
/distrito/listar?idProvincia=1501	100	887	204	2735	477.66	1.00%	0.10882	1.43	0.1	13441
/sustento-postulante/listar?idPostulante=1	100	671	195	2024	412.38	0.00%	0.1167	0.13	0.1	1168
/solicitudBeca/inicializar-busqueda?idPostulante=1	100	648	192	1935	432.18	1.00%	0.11435	0.06	0.1	554.4
/solicitudBeca/listarConvocatoriaAnio?anio=2022&idPostulante=1	100	713	185	5703	661.96	0.00%	0.1089	0.3	0.1	2809.3
/solicitudBeca/listarConvocatorialnscritos?anio=2022&idConvocatoria=6&	200	699	190	3774	494.61	0.50%	0.19513	0.48	0.18	2542.5

- **/INICIO:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 1.907 segundos, un promedio de 2.880 segundos y un máximo de 7.002 segundos. Teniendo en cuenta un promedio de Bytes de 2797793.
 - **/POSTULANTE/INICIALIZAR?IDPOSTULANTE=1:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.704 segundos, un promedio de 1.431 segundos y un máximo de 6.608 segundos. Teniendo en cuenta un promedio de Bytes de 111637.6
 - **/SOLICITUDBECA/LISTARCONVOCATORIAANIO?ANIO=2022&IDPOSTULANTE=1:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.185 segundos, un promedio de 0.713 segundos y un máximo de 5.703 segundos.
- /REPORTE/REPORTEFICHASHISTORIAL/GENERARFICHA:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 0.283 segundos, un promedio de 2.127 segundos y un máximo de 4.982 segundos.

CASO 2:

Se tuvo en cuenta a 100 postulantes realizando acciones dentro del portal teniendo como ingreso a cada 1 segundos después de que ingresó el primero postulante y un intervalo de tiempo que nos ayuda a que las peticiones se realicen en un tiempo simulado a la realidad (para este caso un intervalo de 0 a 1 segundos por cada petición).

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
login	100	2181	1181	3617	524.92	0.00%	23.21802	21.88	11.11	965
buscarIdPostulante?codUsuario=c70213952	100	710	190	1748	399.81	0.00%	33.73819	15.95	30.34	484
obtener-por-usuario?usuario=c70213952	100	261	185	708	97.93	0.00%	40.84967	17.79	36.46	446
/inicio	100	23375	3723	30931	4594.78	0.00%	2.77778	7589.5	10.06	2797793
/inicio/listar-banners	200	270	181	543	64.98	0.00%	5.72639	3.83	5.05	685
/inicio/listar-modalidades	400	274	182	752	75.71	0.00%	9.75634	16.87	8.64	1770.3
/inicio/listar-accesos-directos	300	266	185	669	73.5	0.00%	7.31261	5.39	6.51	755
/inicio/listar-logos	200	264	181	552	65	0.00%	5.22931	5.5	4.6	1077
/nosotros/inicializar	100	262	183	671	81.23	0.00%	3.11643	4.86	2.75	1595.3
/convocatoria/listar-convocatorias-portal?IdModalidad=1	100	329	211	784	111.18	0.00%	3.15766	12.12	2.86	3931
/inicio/obtener-reglamento	100	279	191	753	93.9	0.00%	3.15338	3.53	2.79	1147
inicio/listar-banners	100	276	192	822	98.68	0.00%	3.15268	2.11	2.78	685
/listar-plugin-por-perfil	100	306	203	606	85.03	0.00%	3.22341	5.52	3.06	1754.3
/certificado/inicializar-busqueda-alumno?IdPostulante=1	200	280	185	645	74.07	0.00%	4.77441	2.6	4.3	557
https://tomcat.expertsystems.pe/becas-backend/devida/api/certificad	100	271	185	578	77.33	0.00%	3.15467	1.49	2.86	485
/provincia/listar?IdDepartamento=02	100	286	181	719	107.24	0.00%	3.19509	18.7	2.84	5993.8
/distrito/listar?IdProvincia=0201	100	273	185	601	79.1	0.00%	3.22904	12.82	2.86	4065
/postulante/inicializar?IdPostulante=1	100	980	720	3898	391.14	0.00%	3.13303	341.54	2.79	111628
/provincia/listar?IdDepartamento=15	100	271	185	499	70.2	0.00%	3.6249	11.32	3.22	3198.5
/distrito/listar?IdProvincia=1501	100	363	196	877	116.23	0.00%	3.52734	46.66	3.12	13546.3
/sustento-postulante/listar?IdPostulante=1	100	278	196	486	70.23	0.00%	3.61311	4.12	3.23	1168
/solicitudBeca/inicializar-busqueda?IdPostulante=1	100	283	192	814	97.97	0.00%	3.71996	1.93	3.33	530
/solicitudBeca/listarConvocatoriaAnio?anio=2022&IdPostulante=1	100	262	193	430	56.15	0.00%	3.70055	10.15	3.36	2809.3
/solicitudBeca/listarConvocatorialnsritos?anio=2022&IdConvocatoria=	200	306	189	678	82.29	0.00%	6.93842	17.21	6.44	2540.3

- **/LOGIN:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 1.181 segundos, un promedio de 2.181 segundos y un máximo de 3.617 segundos.
- **/INICIO:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 3.723 segundos, un promedio de 23.375 segundos y un máximo de 30.931 segundos.
- **/REPORTE/REPORTEFICHASHISTORIAL/GENERARFICHA:** se observó se realizó una petición por cada usuario y se obtuvo que el tiempo mínimo de respuesta es 1.266 segundos, un promedio de 1.892 segundos y un máximo de 3.993 segundos.

RECURSOS

INICIO

Resultados de las pruebas en la ruta portal/inicio y la relación con los recursos requeridos:

CASO 1: Cuando 1 usuario ingresa a la ruta portal/inicio y se realizan las peticiones requeridas:

- **/INICIO:** se observó se realizó las peticiones y se obtuvo que el tiempo mínimo de respuesta es:

Primera prueba: 10.525 segundos, un promedio de 10.525 segundos y un máximo de 10.525 segundos, teniendo un promedio total en Bytes de 2797793.

Segunda prueba: 6.148 segundos, un promedio de 6.148 segundos y un máximo de 6.148 segundos, teniendo un promedio total en Bytes de 2797793

Tercera prueba: 7.521 segundos, un promedio de 7.521 segundos y un máximo de 7.521 segundos, teniendo un promedio total en Bytes de 2797793

CASO 2: Cuando se tienen 20 usuarios que ingresan al mismo tiempo a la ruta portal/inicio y se realizan las peticiones requeridas:

Primera prueba: 4.141 segundos, un promedio de 7.581 segundos y un máximo de 11.009 segundos, teniendo un promedio total en Bytes de 2797793.

Segunda prueba: 4.141 segundos, un promedio de 7.581 segundos y un máximo de 11.009 segundos, teniendo un promedio total en Bytes de 2797793.

Tercera prueba: 5.600 segundos, un promedio de 8.387 segundos y un máximo de 11.570 segundos, teniendo un promedio total en Bytes de 2797793.

Cuarta prueba: 5.174 segundos, un promedio de 8.022 segundos y un máximo de 10.877 segundos, teniendo un promedio total en Bytes de 2797793.

CASO 3: Cuando se tienen 50 usuarios que ingresan cada 2 segundos a la ruta porta/inicio y se realizan las peticiones requeridas:

Primera prueba: 12.327 segundos, un promedio de 20.645 segundos y un máximo de 25.059 segundos, teniendo un promedio total en Bytes de 2797793.

Segunda prueba: 8.992 segundos, un promedio de 17.450 segundos y un máximo de 22.432 segundos, teniendo un promedio total en Bytes de 2797793.

Tercera prueba: 6.932 segundos, un promedio de 18.531 segundos y un máximo de 22.748 segundos, teniendo un promedio total en Bytes de 2797793.

CASO 4: Cuando se tienen 100 usuarios que ingresan cada 5 segundos a la ruta porta/inicio y se realizan las peticiones requeridas:

Primera prueba: 20.364 segundos, un promedio de 30.893 segundos y un máximo de 41.393 segundos, teniendo un promedio total en Bytes de 2797793.

Segunda prueba: 12.027 segundos, un promedio de 32.071 segundos y un máximo de 40.244 segundos, teniendo un promedio total en Bytes de 2797793.

Tercera prueba: 23.718 segundos, un promedio de 31.319 segundos y un máximo de 37.433 segundos, teniendo un promedio total en Bytes de 2797793.