



Universidad
Norbert Wiener

FACULTAD DE INGENIERÍA Y NEGOCIOS

Escuela Académico Profesional de Ingenierías

Trabajo de Suficiencia Profesional

**Mejoras en los procesos de calidad de software para los
clientes de la empresa Tata Consultancy Services**

Para optar el título profesional de Ingeniero de Sistemas e Informática

Autor:

Bach. Garcés Sáenz, Sonia Lizeth

Identificador orcid:

0000-0001-7782-3708

Asesor :

Mg. Walter Amador Chávez Alvarado

Identificador orcid del asesor:

0000-0001-8614-482X

Lima, Perú

2023

 Universidad Norbert Wiener	DECLARACIÓN JURADA DE AUTORIA Y DE ORIGINALIDAD DEL TRABAJO DE INVESTIGACIÓN	
	CÓDIGO: UPNW-GRA-FOR-033	VERSIÓN: 01 REVISIÓN: 01

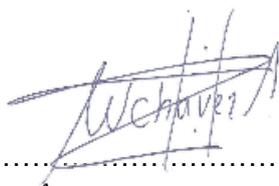
Yo, **Sonia Lizeth Garcés Saenz**, egresado de la Facultad de Ingeniería y Negocios y Escuela Académica Profesional de Ingenierías de la Universidad privada Norbert Wiener declaro que el trabajo académico "**Mejoras en los procesos de calidad de software para los clientes de la empresa Tata Consultancy Services**" Asesorado por el docente: Mg. Walter Amador Chávez Alvarado DNI 09731774 ORCID 0000-0001-8614-482X tiene un índice de similitud de 3 (tres) % con código oid:14912:229156909 verificable en el reporte de originalidad del software Turnitin.

Así mismo:

1. Se ha mencionado todas las fuentes utilizadas, identificando correctamente las citas textuales o paráfrasis provenientes de otras fuentes.
2. No he utilizado ninguna otra fuente distinta de aquella señalada en el trabajo.
3. Se autoriza que el trabajo puede ser revisado en búsqueda de plagios.
4. El porcentaje señalado es el mismo que arrojó al momento de indexar, grabar o hacer el depósito en el turnitin de la universidad y,
5. Asumimos la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión en la información aportada, por lo cual nos sometemos a lo dispuesto en las normas del reglamento vigente de la universidad.



.....
 Firma de autora
 Sonia Lizeth Garcés Sáenz
 DNI: ...76134556.



.....
 Firma Asesor
 Walter Amador Chavez Alvarado
 DNI: 09731774

Lima, 3 de abril de 2023

Resumen

La prueba de calidad es la actividad que va permitir identificar si el software desarrollado, ejecuta los requisitos establecidos por el cliente. El presente trabajo tiene como propósito poder analizar los lineamientos de calidad de software ejecutados en un banco y en base a ello poder mejorar estos procesos de manera que se garantice la calidad del software. La metodología de trabajo aplicado en este informe se encuentra en base a las siguientes teorías: la teoría general de sistemas, teoría de calidad y la teoría de información, lo cual ha permitido desarrollar y dar forma a este trabajo, además este informe se sustenta en base a la experticia laboral obtenida durante el periodo de tiempo de trabajo realizado con el cliente. Tomando como principio al análisis y resultados de este trabajo se pudo concluir que para garantizar la calidad de software no es suficiente verificar si funcionalmente se encuentra alineado a los requisitos del cliente, también es necesario poder conocer el lado no funcional del software, donde se incluye las pruebas y resultados de rendimiento y capacidad que posee el sistema al momento de ejecutar sus tareas para la cual fue desarrollada, de esta forma se podrá minimizar riesgos que involucren a los intereses de la empresa y afecte el beneficio del cliente, por lo tanto la correcta y completa ejecución de lineamientos de pruebas trae como consecuencia poder entregar un software de calidad que va a garantizar la satisfacción a la organización y clientes.

Palabras claves: Calidad, Rendimiento, Funcional, Pruebas, Software

Abstract

The quality test is the activity that will allow us to identify if the software developed meets the requirements established by the client. The purpose of this work is to be able to analyze the software quality guidelines executed in a bank and, based on this, to be able to improve these processes in order to guarantee the quality of the software. The work methodology applied in this report is based on the following theories: general systems theory, quality theory and information theory, which has allowed the development and shaping of this work, in addition this report is based on based on the labor expertise obtained during the period of time of work carried out with the client. Taking the analysis and results of this work as a principle, it was possible to conclude that to guarantee the software quality it is not enough to verify if it is functionally aligned to the client's requirements, it is also necessary to be able to know the non-functional side of the software, where it is included. the tests and results of performance and capacity that the system has at the time of executing its tasks for which it was developed, in this way it will be possible to minimize future risks that involve the interests of the company and the benefit of the client, therefore The correct and complete execution of test guidelines results in being able to deliver quality software to the client that will guarantee the satisfaction of their business needs.

Keywords: Quality, Performance, Functional, Testing, Software

Índice

Caratula.....	i
Resumen.....	ii
Abstract.....	iii
Índice	iv
Introducción	7
Capítulo I: Antecedentes y Descripción General de la Experiencia	9
1.1. Descripción de la Empresa.....	9
1.2. Antecedentes	10
1.3. Problemática.....	12
1.4. Objetivos	13
1.5. Descripción General de la Experiencia.....	14
Capítulo II: Fundamentos Teóricos	16
2.1. Bases Teóricas.....	16
2.2. Bases Conceptuales:	19
2.3. Información de la Empresa:	22
2.4. Descripción del Puesto:.....	22
Capítulo III: Aporte y Desarrollo de la Experiencia	25
3.1. Contextualización:	25
3.2. Descripción Detallada de la Experiencia	26
3.3. Análisis De La Experiencia.....	33
3.4. Aportes.....	36
Capítulo IV: Propuestas	47
Conclusiones	49

Recomendaciones	50
Referencias Bibliográficas	51
Anexo.....	57

Índice de Tablas

Tabla 1 <i>Responsabilidades del equipo</i>	24
Tabla 2 <i>Funciones de los Actores del Diagrama de Proceso</i>	27
Tabla 3 <i>Historia de usuario</i>	29
Tabla 4 <i>Matriz de casos de prueba parte 1</i>	29
Tabla 5 <i>Matriz de casos de prueba parte 2</i>	30
Tabla 6 <i>Documento de Plan de pruebas</i>	31
Tabla 7 <i>Informe de cierre de pruebas de certificación</i>	32
Tabla 8 <i>Matriz de riesgo</i>	39
Tabla 9 <i>Recopilación y análisis de requisitos no funcionales</i>	40
Tabla 10 <i>Planificación de pruebas de rendimiento</i>	41
Tabla 11 <i>Informes y recomendaciones</i>	45

Índice de Figuras

Figura 1 <i>Organigrama Área Transformacion CREW SUPPORT</i>	24
Figura 2 <i>Diagrama de procesos para la ejecucion de pruebas de certificacion</i>	27
Figura 3 <i>Diseño de pruebas de rendimiento en Jmeter</i>	38
Figura 4 <i>Proceso de pruebas no funcionales</i>	39
Figura 5 <i>Diseño de pruebas de rendimiento de Carga</i>	42
Figura 6 <i>Modelado prueba de carga en Jmeter</i>	43
Figura 7 <i>Modelado de prueba de Stress en Jmeter</i>	43

Introducción

Las pruebas de calidad de software son un conjunto de actividades que contemplan herramientas y una serie de procesos, de las cuales muchas organizaciones no le brindan la debida importancia y muy pocos son las que invierten realmente en una adecuada implementación de estas actividades dentro de sus proyectos de software.

Por tal motivo, en este informe de experiencia laboral trataremos acerca de una situación actual de como se viene ejecutando las pruebas de software de un cliente bancario, con el objetivo principal de mejorar los lineamientos de pruebas de calidad de software para garantizar la entrega de un producto de calidad a los usuarios finales, alineado a ello, se tienen como objetivos específicos, capacitar a los *tester* acerca de la actualización de lineamientos de pruebas de calidad de software, automatizar las pruebas funcionales y no funcionales para optimizar el tiempo y esfuerzo en la ejecución de pruebas, y finalmente controlar el cumplimiento de la ejecución de los lineamientos de calidad establecidos para los futuros desarrollos.

Es común que las áreas de calidad de software de las empresas no apuesten por implementar en sus procesos la ejecución de pruebas no funcionales, lo que les ha traído como consecuencia grandes pérdidas y sobre todo insatisfacción de las personas usuarias que interactúan con el software, ya que sucede que al transcurrir el tiempo y por el desconocimiento o falta de información de otros factores como el rendimiento del producto, pues la cantidad de usuarios y peticiones que el software atenderá desde cuando se implementó no será la misma después de 5 años.

La importancia de este trabajo se basa en entregar al cliente un software de calidad, habiendo realizado una adecuada ejecución de pruebas funcionales y no funcionales de software, lo cual va a permitir validar y asegurar no solo su correcto funcionamiento sino también conocer el rendimiento del sistema ante determinados escenarios y de esa forma se tenga un software de calidad con procesamiento de información confiable para la toma de decisiones.

El presente informe se sostiene mediante bases teóricas de calidad, de información y de sistemas, además se toma como referencia la experiencia laboral desarrollada en el área de calidad de sistemas del banco, asimismo, cuenta con 4 capítulos principales, que será descritos a continuación, en el capítulo I se describe de los antecedentes que contienen el mismo objeto de trabajo del cual se está desarrollando y se detalla la experiencia laboral, en el capítulo II se habla de fundamentos teóricos donde se menciona las teóricas aplicadas en este informe además de conceptos teóricos, en el capítulo III se habla de los aportes realizados y el desarrollo de la experiencia laboral y por último en el capítulo IV se precisa sobre las propuestas en base a nuestra experiencia.

El alcance de este proyecto abarca el proceso de calidad de software que son ejecutados por los *tester*, por otro lado, como limitaciones de este trabajo se considera el poco interés del banco en el uso de herramientas de soluciones tecnológicas para ser utilizadas en la ejecución de pruebas.

Capítulo I: Antecedentes y Descripción General de la Experiencia

1.1 Descripción de la Empresa

Tata Consultancy Services conocido como TCS es una empresa multifuncional, que proviene de la India y entre sus principales servicios que ofrecen están relacionados a consultoría de servicios tecnológicos, asimismo, TCS se fundó en el año 1968, inicialmente tenía como nombre Tata Computer Systems, en el cual estuvo enfocado en ofrecer servicios de tarjetas perforadas, luego en 1975 empezó a trabajar para una empresa suiza SIS SegInterSettle, habiendo desarrollado un sistema de comercio electrónico que tenía como nombre SECOM, adicional a ello, también obtuvo un contrato para ofrecer servicios a una empresa canadiense.

En 1980, TCS la sucursal de India incorporó actividades de investigación y desarrollo de software, en 1993 se asoció con una casa de software de Canadá, en el 2005 TCS se incorporó en el mundo de la bioinformática, en el año 2006 la empresa desarrolló un software para la Corporación de Turismo y Catering de Ferrocarriles para India, durante el 2011 TCS decidió ofrecer sus servicios a todo tipo de empresas con servicios tecnológicos implementados en la nube.

En el 2015 se convierte en la empresa más grande de la India, y en el año 2020 logra obtener la máxima rentabilidad en la cual fue reconocida como la mejor consultora de TI del mundo con una capitalización de 144,730 millones de dólares, en consecuencia del crecimiento, se vio en la necesidad de contratar más personal para el desarrollo de soluciones tecnológicas, pues llegó a contratar a más de 40,000 personas.

Actualmente TCS se encuentra brindando sus servicios en los siguientes continentes, América (Argentina, Brasil, Canadá, Chile, Colombia, Ecuador, México, Perú, Uruguay, EE.UU), Asia Pacifico (Australia, Porcelana, Hong Kong, Indonesia, Japón, Malasia, Nueva Zelanda, República de Corea, Singapur, Taiwán, Tailandia), Europa y Reino Unido (Bélgica, Dinamarca, Estonia, Finlandia, Francia, Alemania, Hungría, Irlanda, Italia, Luxemburgo,

Países Bajos, Noruega, Polonia, Portugal, España, Suecia, Suiza, Reino Unido), y también en Oriente Medio y África. Asimismo, TCS abrió su sucursal en Perú con el gerente general Sangram Sahoo hace aproximadamente 10 años atrás, pues apostaron en nuestro país como una oportunidad de crecimiento, ya que consideran, que hay mucho por desarrollar en cuanto a la tecnología.

Las principales actividades de la empresa Tata Consultancy Services, se encuentran enfocadas en brindar consultoría de soluciones tecnológicas para empresas, de las cuales están capacitados para brindar servicios a las siguientes industrias: sector bancario, servicio de comunicaciones, educación, salud, seguro, fabricación, empresas de bienes y consumo, energía, servicios públicos etc., de los cuales dentro de sus servicios los principales productos que brindan son sistemas de información integrados (ERP) en la nube, plataformas, IA, IoT, cibernética, sostenibilidad entre otros. (Tata Consultancy Services,2023).

Tata Consultancy Services Sucursal Del Perú, Con Ruc N° 20535627101, con domicilio en Av. Nicolás de Ayllón N° 2941 Piso 3, distrito de El Agustino, Provincia y Departamento de Lima, TCS se desarrolla bajo el sector de consultoría en TI y en soluciones de negocio, tiene como filosofía de trabajo, enfocarse no solo en los intereses de sus propietarios sino también de sus empleados, clientes, comunidad y el país en general.

Asimismo, Tata Consultancy Services cuenta con un agradable ambiente laboral, además de ser una de las empresas líderes de servicios tecnológicos con mejor renombre en el país. Cabe mencionar que además de ofrecer servicios de consultoría y negocio, TCS invierte en conocimiento y siempre en mantenerse a la vanguardia de la tecnología, ofreciendo siempre nuevas soluciones que capten y ayuden al cliente a mantener y desarrollar su potencial según sea su sector. (Tata Consultancy Services, 2023).

1.2 Antecedentes

A continuación se señalan estudios o investigaciones realizados por otros investigadores relacionados al objeto del presente trabajo.

Antecedentes nacionales

Según Mera (2018), donde su artículo basado en la implementación de software de aplicativo para el ámbito educativo, donde para garantizar la calidad de software, implementó un framework que permitió ejecutar las pruebas de este software basándose en la usabilidad, es decir, si el usuario no brinda su conformidad acerca de la funcionalidad del aplicativo no debe proceder el proyecto con su implementación. Finalmente el autor concluye en que este framework se ha desarrollado de forma exitosa, pues en las pruebas experimentales los resultados fueron óptimos.

Antecedentes internacionales

Según Acosta et al. (2018), presenta el proceso de calidad de software aplicando estándares de calidad. Donde se implementó en este trabajo los estándares de calidad y la validación de pruebas de software de integración para la empresa SYSNET, en el cual se evalúa que el software cumpla con las funciones requeridas por el cliente, sin embargo debido a la fechas límites establecidas para la entrega de software, omiten el proceso de pruebas de calidad, a fin de cumplir en los tiempo de entrega de proyecto, pero a consecuencia de ello, se implementó de forma adecuada los procesos de calidad, donde se concluye que se pudo establecer correctamente el alcance de las pruebas, garantizar la calidad del producto y la reducción de tiempo y costos de entrega.

Según Mascheroni et al. (2018), donde lo complicado que resulta para algunas organizaciones que trabajan bajo la metodología ágil, en realizar las entregas continuas de software en tiempos relativamente reducidos, pues en este escenario se tuvo como reto garantizar la calidad de software donde las pruebas se realicen dentro de ese rango de tiempo, en ese sentido, y en base al análisis realizado se implementó la incorporación de un modelo basado en las pruebas continuas, donde incluyen procedimientos para garantizar la calidad de su productos software, asimismo, se concluye que se logró entregar el software dentro del rango de tiempo establecido, además de cumplir con los lineamientos de calidad de forma adecuada.

Según Jiménez (2020), donde este informe habla acerca de las metodologías y un modelo de calidad que permita la ejecución de pruebas de calidad de software para el sitio web Allison, en el cual se implementó el desarrollo de este modelo que contiene un conjunto

de actividades adecuadamente establecidas, donde se concluye que un 50% de las pruebas cuentan con errores, sin embargo estos errores no son críticos ni van a poner en riesgo la continuidad operativa, ya que también realizando un análisis de los defectos encontrados se pueden corregir en un tiempo corto sin que afecte a los usuarios finales.

1.3. Problemática

El control de calidad del producto software, en cualquier ámbito para cualquier empresa desarrolladora de software, son actividades esenciales para ejecutar, sin embargo antes del siglo XX esta actividad era netamente responsabilidad de la persona encargada de elaborar el producto, al paso de los años el control de calidad del software, viene evolucionando de forma que esta actividad ya no sea realizada por la misma persona que fabricaba el producto, pues ya no es una actividad que le corresponda netamente al programador.

Por tal motivo, se ha incrementado la necesidad de que las empresas desarrolladoras de software busquen independizar e invertir cada vez más en conocimientos para la ejecución de pruebas de calidad de software, con el objetivo que el producto software pueda satisfacer las necesidades de sus clientes, y para poder lograr este objetivo, el software debe pasar por un proceso riguroso de pruebas de calidad, sin embargo, hay empresas que aún no se atreven en invertir en la implementación de procesos de calidad y esta decisión puede generar impacto como por ejemplo, productos software con fallos y defectos que generan insatisfacción de sus clientes, además de afectar de manera directa la productividad, rentabilidad y la imagen de la empresa.

Adicional a ello, también es necesario generar conciencia, de los errores que se cometen al no contar con control de calidad de software bien establecido, es importante también que se defina correctamente los lineamientos de certificación de software, pues de lo contrario estarían presentando muchos problemas con respecto al uso del sistema, y la marca de la empresa se vea afectada perdiendo clientes importantes por la mala calidad de software que entregan al finalizar el desarrollo del producto.

Tata Consultancy Services cómo se mencionó anteriormente, es una empresa que brinda servicios como tercero o proveedor a clientes importantes en el país, en el Perú uno de

sus clientes potenciales se encuentran en el sector de banca y personas, el cual nos enfrentamos a un gran reto como proveedor ya que los bancos tienen altos crecimiento de necesidades en tecnología, asimismo, cabe mencionar que el cliente bancario tienen como enfoque brindar innovaciones tecnológicas para mejorar el servicio a más de 2 millones de clientes.

Las principales actividades del banco, están enfocadas en la prestación de servicios bancarios en el área de banca retail, actualmente se enfoca en banca de personas realizando consumos e hipotecas, lo cual esto representa más del 50 % de su cartera total, adicional de su banca comercial.

Pues para mantener la confianza y permanencia de sus clientes, el banco se preocupa en el desarrollo de proyectos de soluciones tecnológicas, donde el área de sistemas le da mayor enfoque a los desarrollos de software en comparación con la ejecución de las pruebas de calidad, de manera que, los lineamientos existentes para la ejecución de las pruebas de calidad no se encuentran correctamente definidos, por lo que, no se cumplen con los criterios de aceptación por parte del usuario, asimismo, se presenta retrabajo, ya que en el ambiente oficial se ha encontrado fallas que no fueron identificadas en el ambiente de pruebas de aceptación de usuario (UAT) y esta situación genera que el software se retorne al equipo de desarrollo para depurar errores, adicional a ello, hay pérdida de tiempo y costos innecesarios para el banco por los retrabajos.

1.4. Objetivos

Objetivo General

Mejorar los lineamientos de pruebas de calidad de software para garantizar la entrega de un producto de calidad a los usuarios finales.

Objetivos específicos

Capacitar a los *tester* acerca de la actualización de lineamientos de pruebas de calidad de software.

Automatizar las pruebas funcionales y no funcionales para optimizar el tiempo, aplicando el mínimo esfuerzo en la ejecución de pruebas.

Controlar el cumplimiento de la ejecución de los lineamientos de calidad establecidos para los futuros desarrollos.

1.5. Descripción General de la Experiencia

La experiencia profesional se desarrolla en la empresa Tata Consultancy Services, brindando consultoría a un cliente de entidad crediticia, con las funciones de tester QA en el área de sistemas, Squad Escuadrón Virtual, desde el 01 de julio del 2022, trabajando en el proyecto de aplicaciones de Bienes Adjudicados y Garantías, bajo un marco ágil, así mismo, el líder QA del proyecto, se encarga de realizar la supervisión, que los *tester* puedan cumplir con los procesos de calidad establecidos, además de facilitar la gestión de permisos de acceso a rutas compartidas donde se coloca el backup de las certificaciones realizadas y entre otras gestiones.

Las funciones principales son:

- **Análisis de historia de usuario**

Se realiza una revisión del documento donde describe cada uno de los criterios de aceptación determinados por el usuario, pues aquí indica cuáles son las funcionalidades que se está requiriendo que el sistema contemple.

- **Elaboración de matriz de casos de prueba**

Una vez realizado el análisis y revisión de la historia de usuario, se tiene que elaborar la matriz de casos de prueba, en la cual este documento indica una serie de pasos a seguir para la ejecución de pruebas funcionales y no funcionales, adicional a ello, se menciona los resultados esperados por cada caso de prueba.

- **Cargar la matriz de casos de prueba al repositorio**

Actualmente en el área de calidad se trabaja con el repositorio Xray, el cual es una herramienta de Jira que permite administrar los casos de prueba, pues en esta etapa se va a realizar la carga de la matriz de casos de prueba al repositorio para que cuando llegue el

momento de ejecutar las pruebas de calidad, esta herramienta nos permita agilizar y gestionar el proceso de pruebas de forma simplificada.

- **Ejecución de pruebas funcionales y no funcionales de los aplicativos.**

Una vez que las pruebas se encuentren cargadas en el repositorio, se procede a validar las funcionalidades del sistema y comprobar si verdaderamente están ejecutándose con los resultados esperados, así como también se realiza la prueba de performance, donde se va a comprobar si los resultados esperados en cuanto a rendimiento son los que se plasmaron en la matriz de casos de prueba para pruebas no funcionales.

Registrar los defectos encontrados en la ejecución de pruebas e informar al área de desarrollo para su depuración. Si en las pruebas funcionales y/o no funcionales se hallaron fallas, entonces se registran estas fallas con sus respectivas evidencias en la aplicación Xray, y se comunica al área de desarrollo para su respectiva corrección o depuración. De no haber defectos, se elabora el documento de evidencias de pruebas y se envía al usuario para su aprobación.

En caso de no haber encontrado fallas en las pruebas y todo funciona de acuerdo a lo establecido en la matriz de casos, entonces elaborar el documento de evidencias y enviarlo al usuario encargado del proyecto para que pueda revisar el documento y en caso de no haber observaciones, el usuario brinde su aprobación de las pruebas y comunique al líder técnico para su respectivo pase a producción.

- **Elaboración de informe de cierre de pruebas**

Una vez el usuario haya enviado su aprobación, se elabora el informe de cierre de pruebas que consta en detallar los criterios de aceptación de la historia de usuario, con los casos de prueba establecidos, cuantas fallas se hallaron y las horas invertidas en el proceso de certificación.

- **Elaboración de manuales.**

Finalmente, como última actividad, se procede a actualizar el manual del aplicativo con el último desarrollo certificado.

Capítulo II: Fundamentos Teóricos

2.1. Bases Teóricas

Teorías generales de sistema:

Un sistema se define conceptualmente como un grupo de elementos independientemente relacionados, en el cual trabajan en conjunto para lograr un objetivo definido y a su vez este grupo es caracterizado como uno solo o un todo. En su mayoría todo sistema se basa en un modelo conformado por la entrada, el proceso y la salida dentro de un mismo contexto, lo cual este contexto o contorno se encuentra fuera del sistema pero de igual manera afecta al sistema, las entradas vendría a ser la información que ingresa de otro sistema o de su propio entorno, y la salida es la información que proporciona de este sistema hacia otros sistemas o a su entorno, el concepto de sistemas es aplicado tanto para el ámbito humano como tecnológico (Beynon, 2018).

La aproximación de la palabra sistemas abarca un conjunto de componentes relacionados entre sí, que la propiedad de cada uno de ellos se engloba en la unidad, cabe mencionar que la categorización de un sistema se basa en sus características y de cómo se encuentran constituidos, que están categorizado como los sistemas concretos (dispositivos tecnológicos, equipos, etc) y los sistemas abstractos (planes, procesos, categorías, etc), adicional a ello también tenemos los sistemas integrales en el cual se subdivide en sistemas auto dirigidos (son los que tienen subsistemas, divididos por grado en funciones de su funciones, además se caracterizan por tener la capacidad de homeostasis es decir tiene la capacidad de nunca autodestruirse o alterarse por circunstancias externas) y tenemos a los sistemas dirigidos (son los sistemas que no cuenta con subsistemas, por lo tanto son independientes) (Rene, 2018)

La TGS (teoría general de sistemas) representa un método para analizar, estudiar la realidad y desarrollar modelos, mediante los cuales se busca llegar a una aproximación a la percepción del universo, estipulando un modelo no aislado del resto conocido como sistema, por lo tanto, se podría decir que la TGS abarca todos los diferentes campos existentes del conocimiento humano y de esta forma pueda predecir el comportamiento de la realidad de elementos existentes, donde estos elementos tienen su contexto en su respectivo nivel, donde

el grado de relevancia de estos elementos, lo miden lo que estos elementos aportan para los fenómenos generados por otros elementos, para que finalmente se produzca un resultado que pueda demostrar la colaboración de los otros elementos. La formación de los conjuntos de elementos que participan en esta interrelación que procede a un resultado, es lo que conocemos como sistema (Lorenzon, 2020).

Para el desarrollo del presente trabajo, se toma como referencia a las tres bases teóricas de teoría general de sistemas, ya que, cada una de ellas habla acerca de las diferentes formas de ver esta teoría según la óptica de cada uno de los autores, sin embargo, concluyen en la misma idea, el cual es fundamental para el desarrollo de mi trabajo, ya que se enfoca en los sistemas como tal del ámbito tecnológico.

Teoría de calidad:

En primera instancia debemos indicar que cuando se habla de la calidad, básicamente estamos hablando de un concepto abstracto que toda persona puede definir según sus creencias, conocimientos, manera de pensar y ver las cosas, como también si preguntamos en una organización que significa la palabra calidad, nos brindan un concepto personal en base a sus intereses de negocio, según el tipo de industria que sea, como también a los servicios que desarrollen. Por otro lado la palabra calidad también lo podemos definir como calidad, del grado de la medición en cuanto un producto puede satisfacer las necesidades de un cliente, por lo tanto, la calidad es la viabilidad de que un producto pueda alcanzar el nivel de funcionalidad o servicio para la cual fue fabricada y también es importante indicar que la calidad está relacionada con las expectativas que pueda tener el cliente con relación al producto ya que este factor es subjetivo pero es importantísimo pues va a brindar el valor que el cliente le da al producto, muy al margen del cuál sea realmente la funcionalidad del producto (Traba, 2020).

La calidad es un término que tiene múltiples criterios en la humanidad y en la sociedad en general, por lo tanto, no vendría a ser un término total si no referente, ya que este término se basa en el discernimiento subjetivo y se sostiene de diferentes juicios. Por lo tanto la calidad es el conjunto de cualidades y singularidades que posee un producto o servicio, y que tiene la destreza para satisfacer las necesidades del cliente, además de efectuar los criterios por los cuales fue elaborado, teniendo en consideración el precio idóneo que el

cliente desea pagar, adicional a ello también incorporamos a este concepto el uso por el cual el cliente siente que se está satisfaciendo sus necesidades, es decir se acopla a la utilidad por el cual fue fabricado, es decir, la calidad de un producto es la valorización que el cliente le concierne, pues la calidad resalta o engrandece el valor de un producto, pues determine cuál es el grado de perfección que la empresa ha seleccionado para aplicar al producto y satisfacer a sus clientes (Bustamante, 2019).

Por otro lado, tenemos otra percepción que indica, que la Calidad es el perfeccionamiento constante que va permitir brindar mayor satisfacción y agrado al cliente acerca de un determinado producto o servicio, por esa razón en una organización es importante la implementación de un modelo de calidad, pues tiene como objetivo poder cambiar y sumar valor al producto o servicio y este tendría un efecto en la organización así como también el progreso en recursos, incremento del rendimiento y productividad. Por ello, la teoría de la calidad está enfocada en estructurar y normalizar los procesos productivos para conseguir mejores resultados (Carranza, 2013) .

Considero que en base a los autores mencionados para la teoría de calidad, concuerdo con el autor Traba, ya que la calidad es un concepto más abstracto y va depender del tipo de cliente que adquiera el producto, pues el cliente en base a sus necesidades y que tan exigente sea va poder calificar de forma subjetiva el nivel de calidad que considera a un producto, por ello, podemos contar con un cliente que no sea tan exigente y que su nivel de resultado esperado sea lo básico, y que va a considerar la calidad de producto bueno o malo, y para ambos clientes se estaría aplicando el mismo procedimiento para sus productos.

Teoría de la información:

El significado de información es muy amplio pero lo podemos reducir en un conjunto de datos descifrados y relacionados que se encuentran dentro de un contexto específico, por otro lado se precisa que un dato es uno o un grupo de símbolos que son utilizados para representar y querer expresar algo a alguien, por lo tanto, al recepcionar la información, estaríamos asimilando y realizando la interpretación de los datos para poder desarrollar una idea en base a esta información (Beynon, 2018).

La información es un conjunto de datos o signos desarrollados en una determinada circunstancia, un dato simboliza la representación que alguien puede mostrar para dar a

conocer algo, por lo tanto la información es la interpretación de estos datos, estos datos tienen una relación entre sí dentro de una misma circunstancia, por lo tanto la información abarca un conjunto de personas desarrollando la interpretación de datos determinantes. Donde la información es el inicio del conocimiento, que está conformada por determinados acontecimientos, sin embargo, la interpretación de estos conocimientos se convierte en un conocimiento significativo para algo específico. Es por esta razón, que la información y el conocimiento están interrelacionados entre sí, y son vitales para la toma de decisiones en un determinado contexto (Davies, 2018).

Por otro lado la teoría de la información plantea un esquema en el cual detallan como bien se mencionó anteriormente, que la información es un conjunto de datos interpretados por una persona, por tanto, para poder transmitir estas interpretaciones se cuenta con un esquema en el cual figura que la persona, es quien empieza a enviar los datos a través de un mensaje al oyente o receptor, donde se requiere principalmente de un contexto que estén relacionados, tanto la persona que envía el mensaje como la persona quien recibe el mensaje, ambos actores requieren de un código en común para la comprensión entre ambas partes y esta comunicación pueda mantenerse (Castro, 2018).

Concuerdo con los tres autores, ya que la teoría de la información pese a ser un concepto muy utilizado pues existen un sinnúmero de conceptos que atribuyen a esta palabras, pues en este caso los tres autores y cada uno con su punto de vista tienen un concepto acertado de lo que es la información en diferentes aspectos, pues tiene una amplia relación con el contexto, la persona, el lenguaje, etc, por tal motivo se requiere para este trabajo los tres conceptos.

2.2. Bases Conceptuales:

a) Aseguramiento de calidad: Es la seguridad razonable de hacer que la información o funcionamiento de algo en específico, como un producto tenga un procesos y procedimiento creíble para elaborar el producto y brinde confianza en el funcionamiento del producto para la toma de decisiones o para la ejecución de determinadas tareas (Mantilla, 2010)

b) Calidad de software: Es la capacidad inherente al producto software de necesitar comprobar y asegurar de hacer las cosas bien, es decir el ser humano por defecto hace la

ejecución de poner algo en funcionamiento para comprobar si el resultado de estas funciones le favorece para una tarea específica (Flores, 2018).

c) Casos de prueba: Es el proceso donde se evalúa la correcta ejecución de un producto, evaluando si cumple con los criterios en los escenarios establecidos por el cliente o persona que hará uso de las funcionalidades de producto, pues estos casos de prueba van a determinar si el producto es óptimo para ser usado (Aristegui, 2010).

d) Certificación: Es la manera de poder garantizar que una actividad está cumpliendo con ciertos parámetros o criterios solicitados por la persona interesada o el cliente, ya que para certificar un producto, va tener que pasar por un cierto número de procesos (Bien, 2004).

e) Ciclo De Vida: Es un proceso cronológico que tiene un inicio y un final, habiendo cumplido un objetivo en este periodo de tiempo (Rodríguez, 2003).

f) Control de calidad: Es la actividad que pretende asegurar que ciertas tareas de un producto o servicio, sean ejecutadas según lo establecido con el objetivo de que se cumplan con los propósitos solicitados (Rivaguero, 2022).

g) Defecto: El defecto es una consecuencia del error y se encuentra en algún componente del sistema, aunque cabe mencionar que un defecto también puede encontrarse en las diferentes etapas que conlleva el desarrollo de software, como por ejemplo, lo podemos encontrar en un manual de la aplicación dirigido al usuario, en la cual puede indicar una funcionalidad que esta desactualizada en la aplicación, lo cual se podría ver también como un defecto en el componente si lo miramos no solo como el desarrollo sino como todo el conjunto incluyendo la documentación (Board, 2018).

h) Error: Un error o más conocido como *bug* es el punto de partida donde nace y desencadena un fallo en la funcionalidad del sistema, esto es comúnmente producido por acción humana, en este escenario la acción sería producida por la persona que está fabricando el producto y en el contexto de sistemas, el error es producido por la acción humana del desarrollador. Estos errores se presentan generalmente por los siguientes motivos: La lógica de programación, los requerimientos son ambiguos o no están claramente especificados,

presión por cerrar el proyecto a tiempo, poca experiencia en el equipo, poca comunicación en el equipo, tecnologías nuevas y entre otros (Board, 2018).

i) Fallo: Una falla en el sistema es la demostración visible de un defecto o en otras palabras vendría a ser la consecuencia de un defecto. Esto se evidencia cuando se está ejecutando la funcionalidad de la aplicación, en caso el software contiene un defecto, entonces en la ejecución presentará un fallo. (Board, 2018)

j) Prueba de software: Las pruebas de software son el conjunto de procesos que se ejecutan para verificar si el software desarrollado cumple con el rendimiento y realiza las tareas que fueron solicitadas para minimizar o gestionar alguna actividad en específico (Board, 2018).

k) Pruebas funcionales: Las pruebas funcionales son una serie de pasos que van a permitir descubrir discernimientos entre el funcionamiento del sistema y los criterios de aceptación establecidos por el usuario (Morales, 2019).

l) Pruebas no funcionales: Las pruebas no funcionales son un conjunto de procesos que van a permitir evaluar no específicamente las funciones del sistema, sino las cualidades de rendimiento que esta posee (Sommerville, 2006).

m) UAT: (User Acceptance Testing por sus siglas en inglés), es un entorno donde se realizan las pruebas de aceptación de usuario, pues el tester se encarga de realizar las pruebas del desarrollo en este ambiente para comprobar si el software cumple con los criterios de aceptación establecidos (Moreno, 2021).

n) ISO 25000: SQuaRE (System and Software Quality Requirements and Evaluation) indica un grupo de estándares que tiene como propósito la generación de un campo de trabajo general para calificar la calidad de software, de esa forma las compañías que implementan la ISO 25000, garantizan la calidad del software además de reducir costos y tiempo de entrega (Muñoz, 2018).

ñ) ISTQB: ISTQB (International Software Testing Qualifications Board), es una sociedad encargada de realizar la certificación de calidad de software, el cual fue desarrollado

en el 2002 sin fines de lucro en Bulgaria. ISQTB son los delegados de establecer y sostener la actualización del desarrollo de certificación para los analistas de calidad o probadores de software (Flores, 2018).

2.3. Información de la Empresa:

Para poner en marcha las pruebas de calidad de software en la empresa TCS, se utilizan diversas herramientas de gestión de información de los cuales van a permitir gestionar a un mayor nivel de control de los desarrollos de software, que incluyen la ejecución de las pruebas de calidad, además de herramientas que van a permitir visualizar el estado de las pruebas, entre otros. Los sistemas de información utilizados en el área de calidad son los siguientes:

a) Jira: es una herramienta que lo utilizan los equipos que trabajan como parte de la aplicación de la metodología ágil, ya que esta herramienta cuenta con tablero kanban y tableros scrum disponibles, pues esto permite que en los proyectos se puedan monitorear desde el inicio, el proceso y en su etapa final de certificación.

b) Xray: es una extensión de JIRA, es una herramienta muy amplia, que básicamente es utilizado para elaborar el plan de pruebas de software, además al estar integrada con JIRA, el equipo scrum tiene la facilidad de verificar el estado de avance de las pruebas de un determinado desarrollo de software, además, permite verificar los resultados obtenidos en las pruebas.

c) Documentaciones: que incluyen manuales de usuario, matriz de casos de prueba, plan de pruebas, documento de evidencias, informe de cierre de pruebas, etc.

2.4. Descripción del Puesto:

Puesto de trabajo:

El puesto de trabajo TESTER/QA que depende funcionalmente del líder QA del equipo, el cual tiene a cargo a 4 *tester* divididos cada uno en un equipo scrum diferente con sus respectivas aplicaciones de negocio.

Misión de un tester QA:

Garantizar la calidad del software, asegurando que se cumpla las funcionalidades del sistema de acuerdo a las necesidades y lo establecido por el usuario.

Visión de un tester QA:

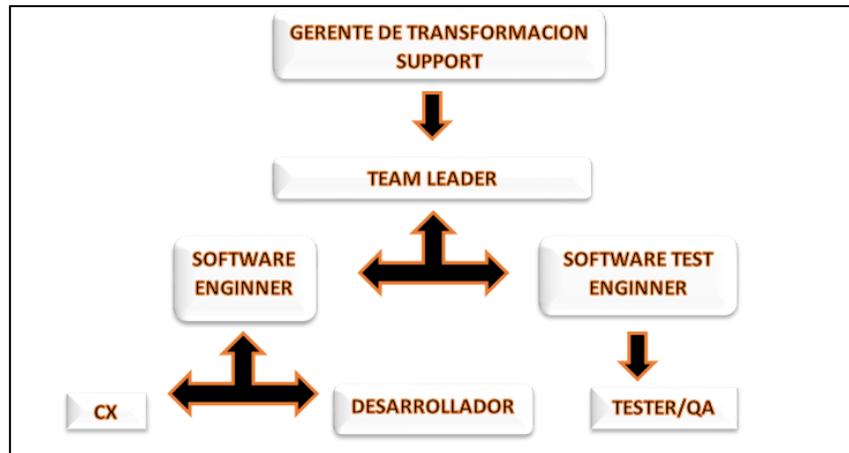
Brindar argumentos razonables de forma precisa y transparente de que el sistema cumple o no cumple con las funciones esperadas.

Funciones de Tester/QA:

- Análisis de historia de usuario.
- Elaboración de matriz de casos donde se evidencien los escenarios de pruebas funcionales y no funcionales.
- Cargar la matriz de casos de prueba al repositorio.
- Ejecución de pruebas funcionales y no funcionales de los aplicativos.
- Registrar los defectos encontrados en la ejecución de pruebas e informar al área de desarrollo para su depuración. De no haber defectos, elaborar el documento de evidencias de pruebas y enviarlo al usuario para su aprobación.
- Elaboración de informe de cierre de pruebas
- Elaboración de manuales.

El área digital de CREW SUPPORT tiene como objetivo crear productos digitales con altos estándares de calidad, utilizando de manera eficiente nuestro conocimiento y las plataformas tecnológicas del banco.

A continuación se muestra el organigrama del área de SUPPORT.

Figura 1*Organigrama Área Transformación CREW SUPPORT*

Nota. Se explica la dependencia funcional del Teste/QA

Detalle de responsabilidades del equipo:**Tabla 1***Responsabilidades del equipo*

Etapa	Formato estándar	Responsable
1. Analizar y diseñar	HU Historia de usuario	CX, Team
	Plan de pruebas	Desarrollador - Tester QA
2. Desarrollar	Resultado de pruebas unitarias (ver CPR)	Desarrollador
	Resultado de pruebas de integración	Desarrollador
3. Realizar pruebas	Resultado de pruebas de certificación	Tester QA
4. Realizar pruebas de certificación	Evidencia de pruebas de certificación + Mail de Conformidad	Tester QA
	IPC Informe de pruebas de certificación	Tester QA

Capítulo III: Aporte y Desarrollo de la Experiencia

3.1. Contextualización:

Al ingresar a laborar a la empresa de consultoría TCS, me asignaron a un cliente del rubro bancario, bajo el puesto de Tester/QA, conformando el equipo tecnológico de SUPPORT, para trabajar en el control de calidad de las aplicaciones de Bienes Adjudicados y Garantías, bajo un marco ágil.

Las pruebas de calidad de software se realizan con el objetivo de certificar que la aplicación se encuentra apta para ser desplegado en el ambiente de producción u operaciones y pueda ser utilizado por el cliente, las pruebas a ejecutar en primera instancia son las funcionales, de las cuales contaban con lineamientos establecidos desde el análisis de requerimientos hasta la etapa final de ejecución de pruebas y aprobación de usuario.

Por otro lado, también se realizan las pruebas no funcionales, de las cuales estas pruebas no tienen un plan definido, sino simplemente se ejecutan pruebas elementales a través de una herramienta de pruebas de rendimiento llamada JMETER, cabe mencionar que tampoco se contaba con un plan de ejecución de estas pruebas, sino simplemente se realiza la grabación del flujo de la aplicación hacia la herramienta JMETER, luego se ejecutaba la herramienta y los resultados se colocaban en un word sin conocer específicamente cual era el significado de los resultados. Y por último, tampoco se realizan pruebas automatizadas, es un plan que aun lo tienen como mejora a mediano plazo, sin embargo, no se cuenta con una persona especializada para poner en ejecución esa actividad.

Por lo tanto se concluye a nivel de contextualización, que el área de calidad de software todavía tiene muchos aspectos por mejorar, ya que no está totalmente consagrada en la ejecución de pruebas de calidad, si bien es cierto siempre hay aspectos que mejorar en cualquier área de sistemas, pero es importante recalcar que un área de calidad que no cuenta con los lineamientos de ejecución de pruebas establecidas, es un área que no garantiza la calidad del software en su totalidad.

3.2. Descripción Detallada de la Experiencia

a) Aplicación de la metodología de pruebas de calidad de software.

Actualmente para el proceso de certificación de software, el área de calidad se enfoca en la ejecución de pruebas funcionales de software, los cuales se tiene un proceso definido para el desarrollo y certificación de estas pruebas, sin embargo se tiene como reto a mediano plazo poder contar con pruebas de performance y pruebas automatizadas, debido a la necesidad del negocio. Líneas abajo se presentará el proceso para el control de calidad que se realiza a las aplicaciones del banco, cabe mencionar que cada área tiene determinadas aplicaciones en desarrollo y operación.

b) Resultados del Q1 – 2023:

Como resultado se tiene un total de 31 historias de usuarios definidos para el Q1 (Enero, febrero, marzo), de los cuales solo se ha desplegado a producción 16 historias de usuario, lo cual es un indicador que se califica como regular ya que no se ha cumplido completamente con la cantidad objetiva.

c) Niveles de prueba:

Tenemos cuatro niveles de prueba de software que son, unitarias (abarca los componentes del software), integración (abarca la relación de la unión de los componentes), sistema (abarca el sistema completo) y aceptación (abarca un todo ejecutando comportamientos reales para lo cual fue desarrollado).

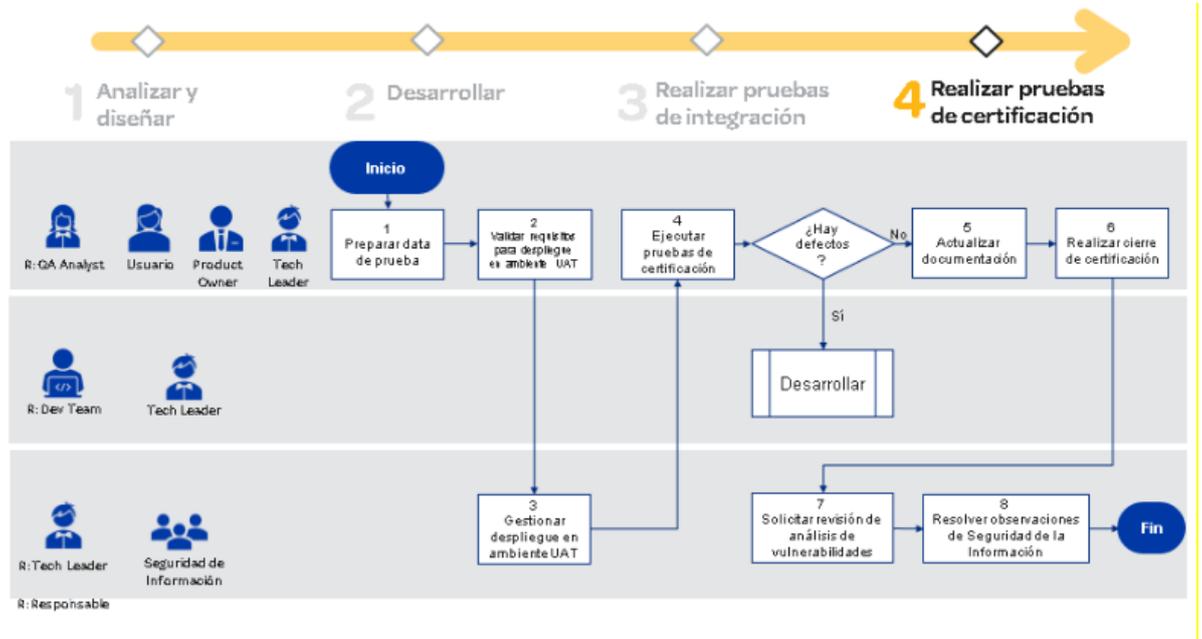
La prueba de aceptación al igual que la prueba de sistema se enfoca en poder probar cual es el comportamiento, respuesta, capacidad, etc, de todo el sistema, su principal objetivo es garantizar la calidad del producto que funcione de acuerdo a lo establecido por el usuario, esta prueba es realizada después de la prueba de sistema, estas pruebas la ejecutan el tester y se incluye al cliente o a las personas interesadas. Cabe mencionar que solo se ejecutan pruebas de tipo funcional (Board, 2018).

Prueba funcional: Son aquellas pruebas de software que van a determinar si el software cumple con las funciones para la cual fue desarrollada. Sus funciones son

determinadas por el cliente, pues es la persona que tendrá una interacción directa con el aplicativo y sus herramientas.

Figura 2

Diagrama de procesos para la ejecución de pruebas de certificación



Fuente: Esto fue elaborado en el equipo de proyectos.

Tabla 2

Funciones de los Actores del Diagrama de Proceso

FUNCIONES DE LOS ACTORES DEL DIAGRAMA DE PROCESO	
ACTORES	FUNCIONES
Product Owner	Es la voz del cliente, tiene como responsabilidad llevar al máximo el valor empresarial para el proyecto, por otro lado participa en las reuniones de sprint planning con el equipo en la cual se le indica al usuario y al product owner cuáles son los proyectos, mejoras o implementaciones que se tienen planificados para el sprint.
Usuario	El usuario es la persona encarga de aprobar las HU planificadas para el sprint, adicional a ello también se encarga de verificar y aprobar las evidencias de las pruebas funcionales enviadas por el analista QA.

Tech Leader	El líder técnico también debe conocer a nivel técnico y funcional cada aplicativo, participa en todas las reuniones scrum, sprint planning, dayli, reuniones de refinamiento si lo hubiese, adicional a ello se encarga de la gestión de pase a producción de los aplicativos una vez certificados por el <i>tester QA</i> .
Dev Team	El equipo de desarrollo, se encarga del desarrollo lógico de cada aplicativo, los programadores participan en todas las reuniones scrum, sprint planning, dayli, reuniones de refinamiento si lo hubiese.
Seguridad De Información	Con el área de Seguridad de la Información, el líder técnico se apoya para poder obtener la aprobación del pase a producción del aplicativo, para esto el equipo de seguridad realiza un análisis de vulnerabilidades, si en caso hubiera una observación en los resultados, estos deberán ser subsanados para que el pase a producción se pueda ejecutar.
QA Analyst	El analista QA se encarga de realizar las pruebas funcionales de los aplicativos planificados para el presente sprint y con miras a ser desplegados al ambiente productivo.

Nota. Los roles participantes en el procesos de pruebas de software

Proceso de certificación

a) Análisis de historia de usuario:

Se realiza una revisión de la Historia de Usuario, entregado por el analista funcional quien ha levanto los requerimientos del usuario en función a los criterios de aceptación del software

Tabla 3*Historia de usuario*

Hu 01: Copia del archivo GACR042M1 a la ruta GACSend	
Yo como	Gestión de portafolio de riesgos de Banca Persona.
Quiero	Que el sistema de GAC pueda copiar el archivo de cartera garantías a una ruta adicional a la que se tiene actualmente de manera automática y diaria.
Para	Para poder utilizar la información en los reportes de seguimiento de la cartera Hipoteca.

Reglas de HU:

[R1] Realizar la copia del archivo GACR042M1 a la ruta GACSend

Criterios de Aceptación:

[R2] El archivo que se copie en la nueva ruta debe ser en el mismo momento en que se genera el archivo actual.

01- CA Verificar que el Job EGAC101 genere una copia del archivo en la nueva ruta

02- CA El archivo debe tener el mismo nombre, estructura y formato que el ya existente.

b) Una vez analizado las Historia de Usuario y Criterios de Aceptación, se elabora la matriz de casos de prueba, en cual va contener los pasos a ejecutar para realizar las pruebas funcionales, adicional de las condiciones, precondiciones, resultados esperados, etc. Y por otro lado también se elabora el documento de plan de pruebas, el cual va contener el

alcance de las pruebas de certificación, que software y hardware se está utilizando, estrategia de pruebas, etc.

Tabla 4

Matriz de casos de prueba parte 1

Funcionalidades HU	Pre Condición	Tipo de Caso de prueba	Tipo de Ejecución	Escenario	App	Descripción
1. Verificar la generación del archivo en la ruta origen.	Ejecución del JOB EGAC101	Medio	Manual	CP001	Gac	Verificar que el archivo txt se encuentre en la ruta origen

Tabla 5

Matriz de casos de prueba parte 2

Descripción	Resultado esperado	Resultado real	Status	Fecha De Ejecución
Ingresar a la ruta de uat y validar que encuentre el archivo txt	En la ruta uat se debe encontrar el archivo txt	En la ruta uat se encuentra el archivo txt	Ok	24/01/23

Tabla 6*Documento Plan de pruebas*

Documento de plan de pruebas	
Ambiente de pruebas:	UAT
Hardware, software y dispositivos externos:	Servidor PANAGON
Data de prueba:	txt
Aplicación:	GAC
Estrategia de prueba:	Verificar que el job EGAC genere una copia de archivos de una ruta origen a una ruta destino
Conformidad a las pruebas:	Rodolfo Pérez
Tipo de pruebas:	Funcional
Técnico/ funcional:	A nivel funcional verificar que el job EGAC genere una copia del archivo de una ruta origen a una ruta destino

c) Carga de matriz de casos de prueba a la herramienta XRAY:

Cuando se tenga lista la matriz de casos de prueba, se procede a cargar el documento al repositorio de pruebas llamado XRAY.

d) Validar registros para desplegar en UAT:

En esta etapa se realiza una validación de los parámetros necesarios para realizar la instalación del aplicativo en ambiente UAT.

e) Ejecución de pruebas de certificación:

Una vez desplegado el desarrollo del software en el ambiente UAT, se procede con la ejecución de pruebas funcionales de las cuales si se encuentran defectos en el funcionamiento de las pruebas se deriva el software con el área de desarrollo para su respectiva depuración, estos defectos encontrados deberán ser cargados en la herramienta Xray, de no encontrarse

defectos en las pruebas, se procede a elaborar el documento de evidencias de pruebas funcionales para enviarlo al usuario, quien se encarga de dar la correspondiente revisión y aprobación.

f) Actualización de documentación:

Cuando las pruebas funcionales de software fueron exitosas, se procede a actualizar los manuales de aplicativo de usuario, con las funcionalidades actualizada.

g) Cierre de certificación:

Finalmente se procede a elaborar el informe de certificación de prueba en el cual se va a detallar un resumen general del proceso de pruebas, además de cuantos casos se probaron y cuantos defectos se hallaron, para que finalmente se pueda tener información de los resultados de las pruebas de calidad.

Tabla 7

Informe de cierre de pruebas de certificación

Informe de pruebas de pruebas de certificación	
SR o Id Jira:	SRT_2023_00335 GAC
Aplicación:	GAC
Historia de usuario:	GAC-Copia de archivos carteras garantía v1
Ambiente de pruebas:	UAT
Ejecutor de pruebas:	Sonia Garcés
	CA El archivo debe tener el mismo nombre, estructura y formato que el ya existente:
Criterios de aceptación	Ruta origen: \\s18rp1\prmcarga\01prm\Windows Nombre del archivo: GACR042M1.TXT Formato: txt

g) Consolidación de certificación en ruta compartida:

Existe una ruta compartida en un determinado servidor que sera utilizado por personas que cuenten con el acceso, en la cual se va a consolidar toda la documentacion desarrollada para las pruebas de calidad de software, en la cual esta carpeta va tener la siguiente estructura.

- Carpeta principal: SRT_2023_XXXX Nombre de la HU
Dentro de esta carpeta principal, se va crear subcapertas con la siguiente estructura:
 - Inicio: se va colocar la HU que se va a desarrollar para el presente sprint, y su respectivo correo de aprobacion de usuario de HU.
 - Preparacion: se va colocar
 - Matriz de casos
 - Plan de pruebas
 - Matriz De Trazabilidad (no esta en uso actualmente)
 - Ejecucion: Se va crear 2 subcarpetas
 - Defectos: los defectos encontrados en las pruebas.
 - Evidencias: las evidencias de las pruebas
Se va a colocar el correo de aprobacion de evidencias, sea el caso que las pruebas fueron exitosas.
- Cierre: se va colocar el informe de pruebas de certificacion

3.3. Análisis de la experiencia

En base a la experiencia obtenida en el área de calidad de software del cliente bancario, se ha podido realizar el siguiente análisis, donde el proceso de calidad que actualmente se viene trabajando alineada al marco ISO 25000 e ISTQB, que tienen un mínimo nivel estándar para ser considerado como un proceso que garantice la calidad del producto software, si bien es cierto desde que se implementó el área de calidad del banco y sus lineamientos, les ha permitido evaluar la funcionalidad del software, sin embargo continuar desarrollando este proceso puede traer consigo riesgos a futuro, ya que estarían descuidando otras cualidades del software que es importante conocer, como por ejemplo su comportamiento en un

determinado contexto. Recordemos que en todo concepto de calidad es de vital importancia la satisfacción del cliente y el cumplimiento de requisitos del producto.

La ISO 25010, a través de un esquema nos indica cuales son las características que tiene que poseer un software para ser considerado software de calidad (ISO25000, 2022), a continuación mencionaremos detalles acerca del modelo de calidad establecida por la ISO 25010.

Adecuación Funcional

Se trata de las capacidades funcionales que debe tener el software para poder cumplir con lo solicitado por el cliente y de esa forma satisfacer sus necesidades de negocio. La adecuación funcional comprende: Completitud funcional (nivel en el que todas las funciones del software abarcan las actividades y propósito del usuario), corrección funcional (capacidad que tiene el software para responder según lo esperado por el cliente), pertinencia funcional (capacidad que tiene el software para proveer las funciones, tarea y objetivos de usuario) (ISO25000, 2022).

Eficiencia de Desempeño:

Se trata de la eficacia del software en base a los recursos utilizados para su ejecución. La eficiencia de desempeño comprende: Comportamiento temporal (tiempo de respuesta, velocidad que se transmiten los datos de un software en el momento que se está ejecutando con relación a un plan de comparativas de rendimiento), utilización de recursos (cantidad de recurso que consume el software cuando está en plena ejecución), capacidad (verificar el comportamiento del software, si cuando se le pone al límite de recursos continúa cumpliendo su función), compatibilidad (capacidad que tiene el software para poner interrelacionarse correctamente entre sus componentes y cumplir con los requisitos especificados), coexistencia (capacidad que tiene el software para convivir con otro software en un mismo entorno con los mismo recursos y ambos cumplir sus funciones independientemente), interoperabilidad (capacidad que tiene uno a más softwares para cambiar información entre sí y poder darle uso a esta información enviada). (ISO25000, 2022)

Usabilidad

Capacidad que posee el software para ser aprendido, intuitivo y utilizado por el usuario. La usabilidad comprende: reconocibilidad de la adecuación (capacidad del software para permitir al usuario poder comprender si el software va acorde a sus necesidades), aprendizabilidad (capacidad del software para ser aprendido por el usuario), operabilidad (capacidad del software que permite al usuario poder controlar el software fácilmente), protección contra errores de usuario (capacidad que tiene el sistema para prevenir que los usuarios tengan errores), estética de interfaz de usuario (capacidad del software para facilitar la interacción de software con usuario), accesibilidad (capacidad que tiene el software para ser utilizados por usuarios con discapacidad.) (ISO 25000, 2022).

Fiabilidad

Capacidad del software de ser considerado como fiable en la ejecución de sus funciones. La fiabilidad comprende: Madurez (capacidad que tiene el software para ser fiable y satisfacer las principales necesidades del usuario), disponibilidad (capacidad del software de encontrarse asequible para cuando el usuario lo requiera), tolerancia a fallos (capacidad del sistema para continuar trabajando ante la presencia de fallos de hardware o software), capacidad de recuperación (capacidad del software para reaccionar adecuadamente ante la presencia de algún fallo repentino) (ISO25000, 2022).

Seguridad

Capacidad del software de resguardar la información a personas no autorizadas, la seguridad comprende: Confidencialidad (capacidad del software que resguarda la información confidencial del sistema, y no permitir el acceso del personas no autorizadas a este nivel de información), integridad (capacidad del software para conservar la información y no sea alterada por personas no autorizadas), no repudio (capacidad del software de poder evidenciar cuales son las acciones que fueron ejecutadas), responsabilidad (capacidad que tiene el software para captar las acciones de una entidad), autenticidad (capacidad que tiene el software para mostrar la autenticidad de una entidad) (ISO25000, 2022).

Mantenibilidad

Capacidad del software que permite poder ser actualizado según la necesidad del negocio. La mantenibilidad comprende: modularidad (capacidad del software que permite realizar cambios en un componente sin que afecte de las funciones de los otros componentes relacionados), reusabilidad (capacidad de un componente de poder ser utilizado para el desarrollo de otro software), analizabilidad (simplicidad de poder analizar el impacto que tendrá el software al realizar algún cambio en parte de ella), capacidad para ser modificado (capacidad que tiene el software de poder ejecutar algún cambio sin que esto genere un impacto en su funcionalidad o rendimiento), capacidad de ser probado (capacidad que tiene el software para poder aplicarse pruebas según una matriz de casos de prueba definido) (ISO25000, 2022).

Portabilidad

Capacidad del software de poder trasladar de forma eficaz sus componentes de un entorno a otro. La portabilidad comprende: adaptabilidad (capacidad que tiene el software para ser adaptado a distintos ambientes de hardware y software), capacidad para ser instalado (simplicidad del software para ser instalado y desinstalado en un determinado entorno), capacidad para ser reemplazado (capacidad del software que puede utilizarse en reemplazo de otro componente que tiene el mismo objetivo de uso) (ISO25000, 2022).

3.4. Aportes

En base a la metodología de pruebas de calidad para las soluciones digitales que viene aplicando el cliente bancario, que se encuentra en proceso de crecimiento y las necesidades de negocio se incrementan a medida que la demanda de clientes aumenta, tienen la necesidad de ampliar el control de calidad y abarcar otros aspectos, pues no solo enfocarse en las pruebas funcionales de software, sino poder tomar en cuenta otros aspectos importantes de la calidad del software y de esa forma minimiza los riesgos que afecta la operación continua del software implementado.

En base a la información y metodología que ya se viene trabajando, se podría implementar el tipo de prueba no funcional con la aplicación JMETER. Cabe mencionar que para iniciar con la implementación de este proceso, los tipos de prueba que se va aplicar serán

carga, estrés y picos, ya que no se tiene la información de los requisitos no funcionales para poder medir los parámetros necesarios, pues se iniciara midiendo y probando cual es la carga que en horas normales el sistema lo soporta.

Objetivos de realizar las pruebas

Validar que se ejecuten las tareas según los requerimientos solicitados, minimizar riesgos, descubrir si el software cuenta con defectos, informar a los interesados acerca de los resultados de las pruebas de calidad para su conocimiento y posterior toma de decisiones.

Jmeter:

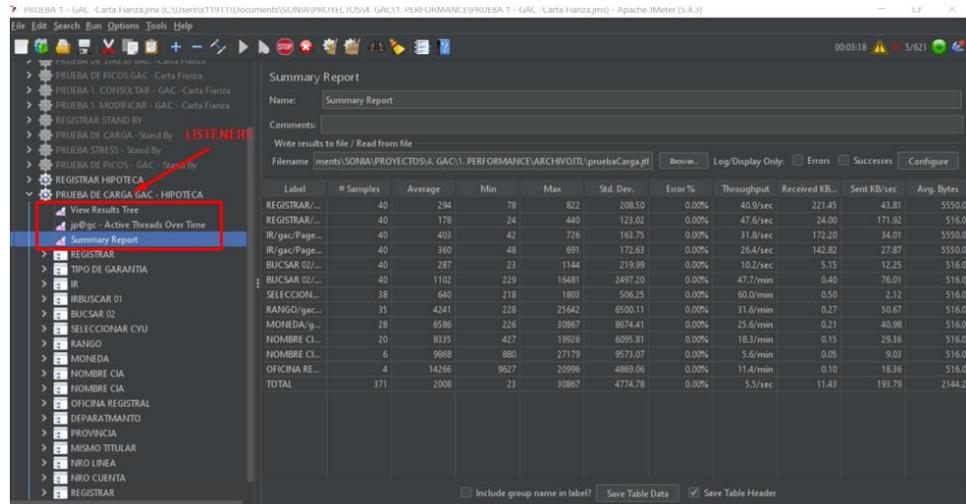
Es una de las más populares herramientas utilizadas por los *tester* para la ejecución de pruebas no funcionales para el rendimiento, de la cual tiene como principales funciones:

- Diseño de un plan de prueba, el cual genera un archivo con extensión .jmx
- Ejecución del plan de prueba
- A través de listeners se podrá evidenciar variedad de resultados del plan de prueba o testplan.

Para ejecutar el diseño de plan de pruebas en la herramienta Jmeter, esta herramienta posee un GUI (interfaz gráfica de usuario), de manera que va facilitar al *tester* la tarea de diseñar y ejecutar el plan de prueba, una vez finalizada la grabación de las acciones del usuario desde el navegador a la herramienta JMETER, este generará un fichero .jmx, al ejecutar este fichero se va a realizar una serie de peticiones al aplicativo que se está probando, pues en cada petición que la herramienta realiza a la aplicación, se guardará información generada en ese instante, pues adicional a ello JMETER va permitir insertar cierta cantidad de usuarios de forma que simule un interacción real de usuarios con el aplicativo en prueba, asimismo, los datos generados en la ejecución del fichero .jmx son producidos también porque internamente este fichero contiene listeners, pues los listeners son herramientas que se encuentran en Jmeter que permite observar los resultados de diferente forma.

Figura 3

Diseño de pruebas de rendimiento en Jmeter



Nota. Pantalla del JMeter

Prueba no funcional:

Las pruebas no funcionales de software, se encarga de probar aspectos no funcionales como la capacidad y desempeño de un software, de las cuales para ejecutar una prueba no funcional de calidad, como aporte se implementará la aplicación de 3 principales pruebas de rendimiento explicadas a continuación:

- **Carga:**

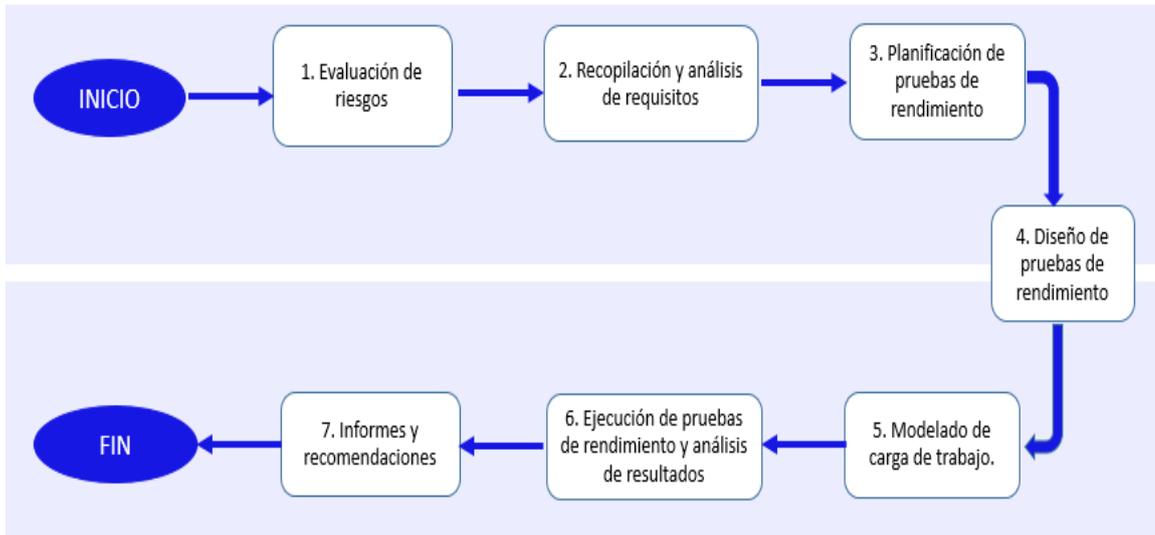
La prueba de carga consiste en poder calcular cual es la cantidad de usuarios que recepciona el sistema en condiciones normales.

- **Stress:**

Esta prueba consiste en enviar al sistema más del límite de usuarios que pueda soportar, y de esa forma verificar el comportamiento del sistema en ese proceso de prueba.

- **Pico:**

Esta prueba consiste en probar el comportamiento del sistema al momento de enviarle repentinamente una excesiva carga de usuarios. Para una adecuada ejecución de pruebas no funcionales, se deberá cumplir con el siguiente diagrama de procesos:

Figura 4*Proceso de pruebas no funcionales***Procesos para pruebas de rendimiento**

- **Evaluación de riesgos**

Tiene como objetivo poder especificar cuáles son los componentes de un software que se requieren realizar las pruebas de forma independiente, en esta matriz de riesgo se califica cual es el componente del software que tiene más criticidad y que requiere realizar las pruebas de rendimiento, además con esta matriz el *tester* puede tener una perspectiva más clara de los escenarios de prueba que se va a implementar.

Tabla 8*Matriz de riesgo*

Nombre Componente	Categoría	Tipo	Vigencia	Disponibilidad	Impacto	justificación
Registrar Carta Fianza	Bienes Adjudicad os	Pag. Web	nuevo	24 x 7	alto	Nueva aplicación, necesita prueba de rendimiento

copilación y análisis de requisitos no funcionales

En esta etapa es donde se recopila y analiza los siguientes datos:

- Número de usuarios esperados por la aplicación
- Tiempo de respuesta
- Número de peticiones procesadas por la aplicación
- Porcentaje de error, etc.

Tabla 9

Recopilación y análisis de requisitos no funcionales

N°	Tipo	Detalle	Impacto	T. respuesta En Segundos	N° de muestras procesadas por la web	% de error
01	Solicitud	El componente debe ser capaz de poder recibir 50 usuarios.	Área: Administración Contabilidad	< a 0.50	<=2500	< a 0.5%
02	Solicitud	El componente debe ser capaz de poder reponerse al recibir un pico de 250 usuarios.	Área: Administración Contabilidad	< 2.150	<= 100	< a 0.5%
03	Solicitud	El componente debe ser capaz de poder reponerse al recibir una carga límite de 350 usuarios.	Área: Administración Contabilidad	< a 30.100	<250	< a 0.5%

- **Planificación de pruebas de rendimiento**

En esta etapa se planifica cuáles serán los escenarios de pruebas de rendimiento a ejecutar en base a lo obtenido en la matriz de riesgos y en la recopilación de requisitos no funcionales, la suma de este análisis se podrá generar un documento llamado plan de pruebas de rendimiento, en este documento se podrá detallar cual es el objetivo de prueba, estrategias de prueba, resultado esperados, etc.

Tabla 10

Planificación de pruebas de rendimiento

Tipo de prueba:	CARGA
Escenario:	Se configura 50 usuarios, para que sean ejecutados cada 1 segundo.
Tiempo r. promedio esperado:	< a 0.560 segundos
Cantidad de usuarios de prueba:	50
Periodo de aceleración (segundos)	0
Duración en segundos:	< 100 segundos
Total de muestras esperadas:	<=2500
% de Error:	< a 0.5%
Tipo de prueba:	PICOS
Escenario:	Se configura 50 usuarios como base, el pico de carga deberá llegar a los 2 minutos y medio, la altura del pico sería una carga de 200 usuarios la duración de la prueba será de 5 minutos
Tiempo r. promedio esperado:	< 2.150 segundos
Cantidad de usuarios de prueba:	250
Periodo de aceleración (segundos)	0
Duración en segundos:	< 350 segundos
Total de muestras esperadas:	<= 100
% de Error:	< a 0.5%
Tipo de prueba:	STRESS
Escenario:	Se configuran 350 usuarios concurrente durante 1 minuto, dividido en 3 cargas, una carga de 50 usuarios, la siguiente de 100 usuarios y finalmente 200 usuarios. El ingreso es cada 6 segundos.
Tiempo promedio esperado:	< a 30.100 segundos

Cantidad de usuarios	350
Periodo de aceleración (segundos)	0
Duración en segundos:	< 650 segundos
Total de muestras esperadas:	<250
% de Error:	< a 0.5%

- **Diseño de pruebas de rendimiento**

En esta etapa a través de la ayuda de la aplicación JMETER, que es una herramienta que va a permitir registrar las acciones que se realiza en un navegador al interactuar con una página web, estas acciones se van a registrar en un script generado por la herramienta de prueba, pues esta generación de script se diseña en base a lo establecido en la planificación de pruebas, pues se va replicar en la página web el flujo a probar, de modo que la herramienta de rendimiento diseñe los script en base a las acciones de prueba que se están ejecutando en el navegador.

Figura 5

Diseño de pruebas de rendimiento de Carga



Nota. Pantalla del JMeter

- **Modelado de carga de trabajo**

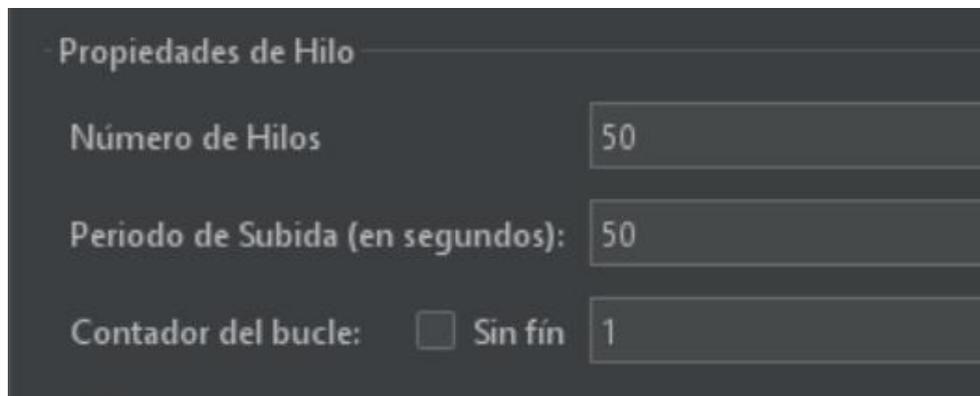
En esta etapa se simula la carga que recibe el software en el ambiente productivo utilizando los scripts diseñados en el paso anterior y lo establecido en el plan de pruebas, por

su puesto esta carga también va depender del tipo de prueba de rendimiento que se va ejecutar, este detalle también está incluido en el plan de pruebas, ya que para cada tipo de prueba va variar la carga que se le aplique. Para una la generación de un correcto modelado de carga de trabajo se requiere de estos dos datos importantes.

- Cantidad de usuarios que espera el sistema
- Cuantas iteraciones requiere un usuario para completar su prueba en un determinado plazo de tiempo.

Figura 6

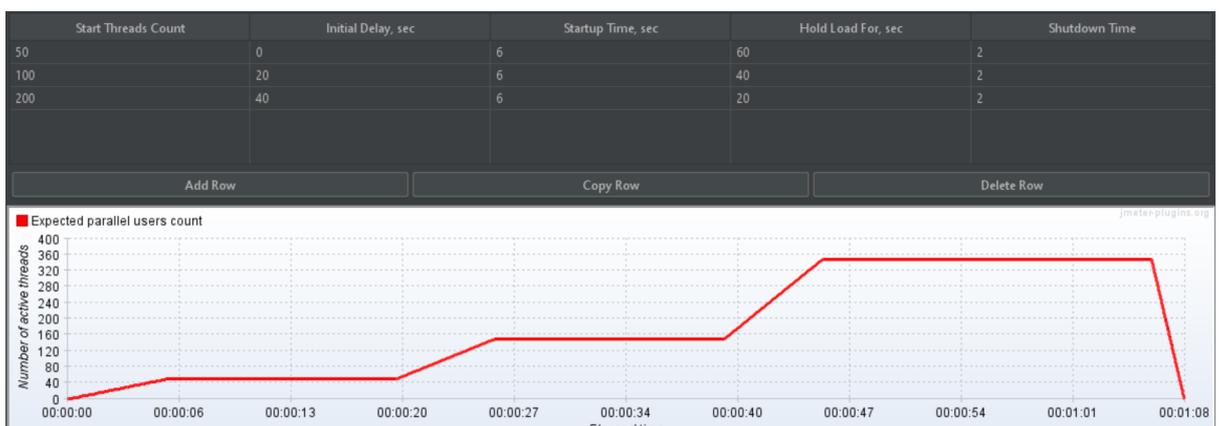
Modelado prueba de carga en Jmeter



Nota. Pantalla del JMeter

Figura 7

Modelado prueba de Stress en Jmeter



Nota. Pantalla del JMeter

- **Ejecución de pruebas de rendimiento y análisis de resultados**

En esta etapa se ejecuta las pruebas diseñadas, utilizando la herramienta de pruebas de rendimiento, en base a los resultados que retorne esta herramienta se genera el análisis de resultados. En este proceso comprende las siguientes actividades:

- Ejecución de pruebas de rendimiento
- Análisis de resultados
- Realizar un match de los resultados contra los requisitos no funcionales
- Elaboración de informe provisional de resultados
- Decidir si en base a los resultados volver a realizar la prueba o completar hasta finalizar.

- **Informes y recomendaciones**

En esta etapa se finaliza el proceso de prueba de performance con un informe oficial de resultado de pruebas, análisis y recomendaciones, este documento finalmente es entregado a las personas interesadas para la toma de decisiones.

Tabla 11*Informes y recomendaciones*

Escenario 1 – Se configura 50 usuarios, para que sean ejecutados cada 1 segundos, los resultados de las pruebas por el componente “Summary Report” son los siguientes.

Cantidad de usuarios	50
Periodo de aceleración (segundos)	0
Duración (segundos)	70
Total de muestras enviadas por la web	2100
Tiempo de respuesta promedio	0.450 segundos

De acuerdo con las pruebas realizadas y a los resultados mostrados por la aplicación, se evidencia que el tiempo promedio para registrar una carta fianza es de 0.541 segundos realizando un total de 2100 requerimientos al servidor, lo cual se encuentra en el rango de tiempo establecido en el documento de requisitos no funcionales.

Escenario 2- Se configura 50 usuarios como base, el pico de carga deberá llegar a los 2 minutos y medio, la altura del pico sería una carga de 200 usuarios la duración de la prueba será de 5 minutos..

Cantidad de usuarios	250
Periodo de aceleración (segundos)	0
Duración (segundos)	650
Total de muestras enviadas por la web	55
Tiempo de respuesta promedio	1.324 segundos

De acuerdo con las pruebas realizadas y a los resultados mostrados por la aplicación, se evidencia que el tiempo promedio para registrar una carta fianza es de 1.324 segundos realizando un total de 55 requerimientos al servidor, y una duración de tiempo de 650 segundos, lo cual no se encuentra en el rango de lo establecido en el documento de requisitos no funcionales

Escenario 3 – Se configura 350 usuarios concurrente durante 1 minuto, dividido en 3 cargas, una carga de 50 usuarios, la siguiente de 100 usuarios y finalmente 200 usuarios. El ingreso es cada 6 segundos.

Cantidad de usuarios	350
Periodo de aceleración (segundos)	0
Duración (segundos)	650

Total de muestras enviadas por la web	184
---------------------------------------	-----

Tiempo de respuesta promedio	33.021 segundos
------------------------------	-----------------

De acuerdo con las pruebas realizadas y a los resultados mostrados por la aplicación, se evidencia que el tiempo promedio para registrar una carta fianza es de 33.021 segundos realizando un total de 184 requerimientos al servidor, lo cual se encuentra en el rango de tiempo establecido en el documento de requisitos no funcionales.

Con respecto a las pruebas realizadas a la aplicación Web con el uso de la herramienta JMETER para el proceso de pruebas de rendimiento, se pudo evidenciar que los tiempos de respuesta en el caso de las pruebas de stress y pico no son las esperadas, pues la herramienta ayudo a realizar una interpretación más clara del comportamiento del sistema, en el cual los resultados al ser entregados a las personas interesadas, realizaron un análisis de los resultados y están evaluando en incorporar mayor espacio en memoria para un mejor rendimiento del sistema.

Cabe mencionar que JMETER es un software que cuenta con múltiples herramientas para la ejecución de pruebas no funcionales, además que tiene una variedad de formas de poder mostrar los resultados obtenidos, es importante que se programen capacitaciones de esta herramienta para poder ampliar los conocimientos acerca de la herramienta y de la variedad de pruebas que se pueden ejecutar, con el fin de que a corto plazo se pueda tener mayor información de rendimiento de las aplicaciones que cuenta el banco y las que estén por desarrollarse.

Capítulo IV: Propuestas

Prueba automatizada:

Con respecto a la ejecución de las pruebas funcionales, es importante también poder responder a los retos que se enfrenta el área de calidad en base a los tiempos que se requiere poder entregar el software al cliente habiendo ya realizado las pruebas funcionales, pues además de enfrentarse a los tiempos límites de entregables, se requiere que en este tiempo limitado, las pruebas sean eficientes, por lo tanto en base a estas necesidades la nueva propuesta es poder realizar la automatización de pruebas funcionales, lo cual va a permitir poder minimizar el tiempo de ejecución de pruebas de un determinado sistema y minimizar también el error humano como se presenta más frecuentemente al realizar las pruebas manuales, por tal motivo, ejecutar las pruebas en menor tiempo y disminuyendo el riesgo de error humano, genera unas pruebas eficientes de software, que en síntesis es lo que se quiere lograr.

Las pruebas automatizadas es un instrumento de ejecución de prueba donde no participa la acción de una persona, estas pruebas abarcan desde pruebas unitarias, de servicio y de interfaz de usuario. (Serna, 2021)

Ventajas

- Rapidez:

Se ejecutan más rápido que las pruebas ejecutadas por una persona.

- Fiabilidad:

Este tipo de pruebas elimina el error humano.

- Repetición:

Se puede probar la respuesta del software de forma repetitiva.

- Reusabilidad:

Se pueden utilizar nuevamente los scripts diseñado para las pruebas automatizadas.

Herramienta de prueba automatizada.

a) Selenium: es un grupo de herramientas de código abierto que permite poder interactuar con el navegador web. Esta herramienta va permitir simular la interacción humana con un navegador, como por ejemplo hacer clics, llenar campos de un formulario, navegar entre páginas, etc, Selenium cuenta con diferentes tipos de herramientas en el cual para la implementación de automatización de pruebas se va a utilizar la herramienta Selenium WebDriver, el cual va interactuar con el navegador sin tener la necesidad de utilizar un servidor intermediario para que las pruebas puedan ser ejecutadas. (Anaya, 2020)

El flujo que utiliza selenium para la ejecución de pruebas es, identificar los elementos web de una página, utilizando localizadores (id, xpath, linktext, ect), luego añadir acciones sobre los elementos web previamente identificados y finalmente ejecutar el test (Coello, 2019).

c) Gherkin: es un lenguaje creado con el objetivo de poder resolver una función específica, particularmente tiene un diseño que va permitir comprender de forma fácil los casos de prueba ya que se encuentra en alto nivel, para que en este caso todo el equipo scrum pueda comprenderlo también, tiene la ventaja de poder implementarse con distintos tipos de herramientas pero en nuestro caso se utilizara las más popular, Cucumber BDD (Ordoñez, 2019).

d) Cucumber: Tiene como objetivo principal poder automatizar pruebas bajo el comportamiento BDD que es un desarrollo guiado por el comportamiento quien se va ocupar de la necesidad del negocio y no de la parte técnica del código, ejecutada desde el punto de vista del usuario, utiliza el lenguaje Gherkin el cual va permitir poder ser comprendido por todas las personas no solo los programadores y QA, ya que también trabaja bajo la metodología Scrum (Sanchez, 2020).

Conclusiones

En base a las pruebas de software desarrolladas en el área de calidad para las aplicaciones del banco, se concluye que para garantizar la calidad de software se requiere la incorporación de lineamientos de pruebas no funcionales, para poder asegurar la entrega de un software con buen nivel de desempeño.

Se capacitó a los tester con los lineamientos de pruebas, incluyendo la ejecución de las pruebas no funcionales para los nuevos desarrollos tecnológicos.

Se automatizaron las pruebas funcionales mediante las herramientas Selenium, utilizando cucumber con el lenguaje Gherkin y las pruebas no funcionales mediante la herramienta JMETER.

Se logró controlar el cumplimiento de la ejecución de las pruebas según lo establecido en los nuevos lineamientos con el apoyo del líder QA, quien se encargó de realizar el seguimiento correspondiente.

Recomendaciones

Se recomienda invertir en capacitaciones de pruebas automatizadas y de rendimiento al personal de QA, de esa forma los conocimientos adquiridos sean aplicados para las pruebas de calidad de software, para evaluar la capacidad de los aplicativos del banco y a su vez minimizar riesgos a futuro, adicional a ello con las pruebas automatizadas va permitir minimizar el tiempo de ejecución de pruebas manuales y error humano.

Se sugiere automatizar las pruebas funcionales, ya que los tiempos de entrega de inicio y cierre de proyecto son reducidos, es por esa razón que los tiempos de pruebas deben agilizarse garantizando que se realizaron correctamente.

Se recomienda que el líder de QA o una persona del equipo realicen el seguimiento correspondiente para la correcta ejecución de los lineamientos de calidad implementados, de esa forma asegurar que se esté ejecutando el proceso de forma adecuada.

En base a los resultados obtenidos en las pruebas no funcionales de software se recomienda poder incorporar más tipos de prueba para determinar otros factores del comportamiento del sistema, y en base a ello tomar decisiones.

Referencias Bibliográficas

- Acosta Solano, L., & Blanquicett, M. (2018). *Prácticas de Pruebas desde la Industria de Software. La Plataforma ASISTO como Caso de Estudio*. Obtenido de https://www.scielo.cl/scielo.php?pid=S0718-07642018000100011&script=sci_arttext&tlng=en
- Anaya, E. (2020). *Inmediatum*. Obtenido de <https://inmediatum.com/blog/piensa-digital/ques-selenium-y-para-que-sirve/>
- Andalucia, J. d. (s.f.). *Introduccion a JMeter*. Obtenido de <https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/388>
- Aristegui, O. J. (2010). *Los Casos de Prueba en la Prueba del Software*. Obtenido de [file:///C:/Users/Sonia%20GS/Downloads/Dialnet-LosCasosDePruebaEnLaPruebaDelSoftware-3399441%20\(1\).pdf](file:///C:/Users/Sonia%20GS/Downloads/Dialnet-LosCasosDePruebaEnLaPruebaDelSoftware-3399441%20(1).pdf)
- Aymara Marin Diaz, Y. T. (2019). *Apuntes para gestionar actividades de calidad en proyectos de desarrollo de software para disminuir los costos de corrección de defectos*. Obtenido de https://www.scielo.cl/scielo.php?pid=S0718-33052019000200319&script=sci_arttext
- Beynon. (2018). *Sistemas de información: Introducción a la informática en las organizaciones*. Obtenido de https://books.google.es/books?hl=es&lr=&id=5jbeDwAAQBAJ&oi=fnd&pg=PR7&dq=teoria+de+sistemas&ots=woKP7IZNMt&sig=R_7z6XzAkw8ngwb2aOpXyFtm2Xk#v=onepage&q&f=false
- Bien, A. (2004). *Una Guia Simple Sobre Certificacion de Turismo Sostenible*. Washington. Obtenido de <https://www.ucipfg.com/Repositorio/MGTS/MGTS14/MGTSV-13/SEMANA1/Guiasimplesobrecertification.pdf>

Board, I. S. (2018). *Programa de Estudio de Nivel Basico*.

Bustamante, M. A. (2019). *Fundamentos de calidad de servicio, el modelo Servqual*.

Obtenido de <https://dialnet.unirioja.es/servlet/articulo?codigo=7361518>

Carranza, J. (2013). *Software de calidad*. Bogota.

Casallas, R. O. (2009). *Gestión de conocimiento para la reutilización de la experiencia*

obtenida en la corrección de defectos de software. Obtenido de

<http://paradigma.uniandes.edu.co/images/sampled/paradigma/ediciones/Edicion2/Numero3/Articulo3/r-rueda-4.pdf>

Castro, M. D. (2018). *El recurso de información y comunicación visual: imagen. Apuntes en*

torno a las Ciencias de la Información y Bibliotecología. Obtenido de

https://www.scielo.sa.cr/scielo.php?pid=S1659-41422018000200102&script=sci_arttext#B14

Céspedes, Guerrero, Sánchez, & Valverde. (2017). *Pruebas de Rendimiento de Software*.

Cifuentes, G. (2018). *Lineamiento Para Investigar y Evaluar Innovaciones Educativas*.

Colombia. Obtenido de

https://books.google.com.pe/books?id=C92iDwAAQBAJ&printsec=frontcover&dq=lineamientos&hl=qu&sa=X&redir_esc=y#v=onepage&q=lineamientos&f=false

Coello, C. (2019). Obtenido de <https://digital55.com/blog/herramientas-testing-introduccion-selenium/>

Davies, P. B. (2018). *Sistemas de información*. Obtenido de

<https://books.google.es/books?hl=es&lr=&id=5jbeDwAAQBAJ&oi=fnd&pg=PR7&dq=teoria+de+la+informacion&ots=woKO5JZNUy&sig=olBRqYSCWuECwCQ4xBVWJO8MXT4#v=onepage&q&f=false>

Flores, D. M. (2018). *Proceso de pruebas del sistema del fondo de empleados*. Obtenido de http://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/11195/Paredes_fd.pdf?sequence=3&isAllowed=y

Inteligo, B. (2017). *Calidad de software*. Colombia.

Interbank. (20 de Febrero de 2023). Obtenido de <https://interbank.pe/nosotros>

Jakobson. (2020). *Rendimiento de Software*.

Lorenzon, E. (2020). *Sistemas y organizaciones*. Obtenido de <http://sedici.unlp.edu.ar/handle/10915/99629>

Mantilla, S. A. (2010). *Aseguramiento; ¿Que Es Eso?* Bogota. Obtenido de https://d1wqtxts1xzle7.cloudfront.net/35220640/SAM-aseguramiento-que-es-eso-libre.pdf?1413895825=&response-content-disposition=inline%3B+filename%3DASEGURAMIENTO_QUE_ES_ESO_Aseguramiento_q.pdf&Expires=1674841119&Signature=d8~GY1twcc7vh6WVgMIbC8IwW3bPitqEl

Mascheroni, M., & Irrazabal, E. (2018). Obtenido de http://sedici.unlp.edu.ar/bitstream/handle/10915/73270/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y

Mejía Cabrera, L. (2019). *Software Libre, Concepto y definición de Software libre, historia y evolucion, características de software libre, software libre y educacion, aplicaciones*. Lima. Obtenido de <https://repositorio.une.edu.pe/bitstream/handle/20.500.14039/5557/MONOGRAF%c3%8da%20-%20MEJIA%20CABRERA%20LEOPOLDO%20-%20FAC.pdf?sequence=1&isAllowed=y>

- Mera, J. (2018). *Diagnóstico de pruebas de calidad en software para ambientes virtuales de aprendizaje sobre dispositivos móviles*. Obtenido de <https://revistas.utp.ac.pa/index.php/memoutp/article/view/1848/2694>
- Miguel, A. S. (2019). *Calidad 3° Edicion*. España. Obtenido de https://books.google.es/books?hl=es&lr=&id=sjqlDwAAQBAJ&oi=fnd&pg=PR5&dq=+calidad&ots=GTS9Jp_dzq&sig=HYLdwu1BxVd9yjnYDGBjCL609sg#v=onepage&q=calidad&f=false
- Morales, E. J. (2019). *Definición e implementación del proceso de pruebas de software basado en NTP-ISO/IEC 12207*.
- Moreno, K. G. (2021). *Práctica empresarial: implementación de desarrollo frontend en sitio web y procesamiento de datos para migración de sistemas de información*. Obtenido de <https://repositorio.utp.edu.co/server/api/core/bitstreams/f9e47a81-ed63-4a2f-aa4a-d8d1c02dcaab/content>
- Muñoz, J. (2018). *Asistente de Software Según la Familia de Normas ISO 25000 Utilizando el Enfoque GQM*. Obtenido de http://sedici.unlp.edu.ar/bitstream/handle/10915/67212/Documento_completo___.pdf-PDFA.pdf?sequence=1&isAllowed=y
- O., J. L. (2010). *Los casos de prueba en la prueba del software*. Obtenido de <file:///C:/Users/Sonia%20GS/Downloads/Dialnet-LosCasosDePruebaEnLaPruebaDelSoftware-3399441.pdf>
- Ordoñez, M. (2019). *OpenWebinars*. Obtenido de <https://openwebinars.net/blog/que-es-gherkin/>
- Organización Internacional de Normalización (2022). *ISO 25000*. Obtenido de <https://iso25000.com/index.php/normas-iso-25000/iso-25010?start=6>

- Oscar, J. (2020). *Pruebas de Calidad Aplicadas al Sitio Web Allison*. Obtenido de <https://dspace.itcolima.edu.mx/handle/123456789/1494>
- Osuna, V. O. (2019). "*Propuestas de mejoras para asegurar la calidad de software*". Obtenido de <http://repositorio.upsin.edu.mx/Fragmentos/tesinas/262016030167OsunaOsunaVanesa6141.pdf>
- Paul, B. (2018). *Introduccion a la Informatica en las Organizaciones*. Obtenido de https://books.google.es/books?hl=es&lr=&id=5jbeDwAAQBAJ&oi=fnd&pg=PR7&dq=teoria+de+sistemas&ots=woKP7IZNMt&sig=R_7z6XzAkw8ngwb2aOpXyFtm2Xk#v=onepage&q&f=false
- Paul, B. (2018). *Sistemas de información: Introducción a la informática en las organizaciones*. Obtenido de https://books.google.es/books?hl=es&lr=&id=5jbeDwAAQBAJ&oi=fnd&pg=PR7&dq=teoria+de+sistemas&ots=woKP7IZNMt&sig=R_7z6XzAkw8ngwb2aOpXyFtm2Xk#v=onepage&q&f=false
- Paul, B. (2018). *Sistemas de información: Introducción a la informática en las organizaciones*. Obtenido de https://books.google.es/books?hl=es&lr=&id=5jbeDwAAQBAJ&oi=fnd&pg=PR7&dq=teoria+de+sistemas&ots=woKP7IZNMt&sig=R_7z6XzAkw8ngwb2aOpXyFtm2Xk#v=onepage&q&f=false
- Ramirez, K. R. (2022). *Automatizacion de Pruebas de Regresion para Optimizar el Proceso de Calidad de Software en la Empresa Orbis Ventures S.A.C*. Obtenido de <https://repositorio.upn.edu.pe/bitstream/handle/11537/32386/Gamarra%20Ramirez%20Karen%20Rosario-Parcial.pdf?sequence=2&isAllowed=y>

- Rene, D. I. (2018). *Algunas reflexiones sobre la teoría general de sistemas y el enfoque sistémico en las investigaciones científicas*. Obtenido de http://scielo.sld.cu/scielo.php?pid=S0257-43142018000200003&script=sci_arttext&tlng=en
- Rivaguero, J. (2022). *¿Control de gestión o gestión de control?* LIMA. Obtenido de <https://www.redalyc.org/pdf/2816/281624914005.pdf>
- Rodríguez, B. I. (2003). *El Análisis del Ciclo de Vida y la Gestion Ambiental*. Obtenido de https://www.ucipfg.com/Repositorio/MAES/MAES-07/BLOQUE-ACADEMICO/Unidad-3/lecturas/ACV_GA.pdf
- San Miguel, P. (2019). *Calidad Fundamentos, Herramientas y Gestion de la Calidad Para Pymes*. ESPAÑA. Obtenido de https://books.google.es/books?hl=es&lr=&id=sjqlDwAAQBAJ&oi=fnd&pg=PR5&dq=QUE+ES+CALIDAD&ots=GTS9Kk_4yy&sig=gK5-91v0VnFkZ03B855IdRig2YI#v=onepage&q=QUE%20ES%20CALIDAD&f=false
- Sanchez, R. (2020). Obtenido de <https://elminimoviable.es/que-es-cucumber-y-como-agilizar-el-testing/>
- Serna, E. (2021). *Una revisión a la realidad de la automatización de las pruebas del software*. Obtenido de https://www.scielo.org.mx/scielo.php?pid=S1405-55462019000100169&script=sci_arttext
- Sommerville, I. (2006). *Ingenieria de Software*. Obtenido de https://books.google.es/books?hl=es&lr=&id=gQWd49zSut4C&oi=fnd&pg=PA1&dq=que+es+un+error+en+un+software&ots=s866srvutf&sig=x4G65_BYHXWvIgowEio8XLs1PtM#v=onepage&q=que%20es%20un%20error%20en%20un%20software&f=false

Traba, L. (2020). *Teoría (y práctica) de las organizaciones : herramientas para la gestión de la calidad*. Obtenido de <https://bibliotecavirtual.unl.edu.ar:8443/handle/11185/5584>

Wang, J. (2019). *Investigación sobre tecnología de prueba de automatización de rendimiento basada en JMeter*. Obtenido de <https://ieeexplore.ieee.org/abstract/document/8806309/authors#authors>

Anexo



Lima, 20 de Febrero del 2023

Señores

Universidad Norbert Wiener

Presente

Por medio del presente se deja constancia que la Srta Sonia Lizeth Garcés Sáenz con documento de identidad 76134556 trabaja en TATA Consultancy Services Sucursal Perú, desde el 01-julio-2022 desempeñando las funciones de Tester QA asignada a proyectos del sector banca y finanzas.

Atentamente

A handwritten signature in black ink, appearing to read 'Alcides Meza', is positioned above the printed name.

Alcides Edwin Meza Alvarez

Delivery Manager

TATA Consultancy Services