



ESCUELA UNIVERSITARIA DE POSGRADO

IMPLEMENTACIÓN DE BUENAS PRÁCTICAS BASADA EN LA METODOLOGÍA
MAC PARA EL ASEGURAMIENTO DE LA CALIDAD DEL DESARROLLO DE
SOFTWARE EN LA EMPRESA CANVIA S.A.

Línea de investigación:
Sistemas de información y optimización

Tesis para optar el grado académico de Doctora en Ingeniería de Sistemas

Autora:

Galindo Francia, Jesica

ORCID: 0000-0002-0801-5923

Asesor:

Gamboa Cruzado, Javier Arturo

ORCID: 0000-0002-0461-4152

Jurado:

Manrique Suárez, Luis Humberto

Rodriguez Rodriguez, Ciro

Quispe Prado, Wilber







Lima - Perú

2024

Document Information

Analyzed document	1A_GALINDO_FRANCIA_JESICA_DOCTORADO_2022.docx (D154375570)
Submitted	2022-12-22 19:27:00 UTC+01:00
Submitted by	Johnny
Submitter email	jastete@unfv.edu.pe
Similarity	0%
Analysis address	jastete.unfv@analysis.arkund.com

Sources included in the report

SA	05 T3_Proyecto_Tesis_Salazar_Jo.docx Document 05 T3_Proyecto_Tesis_Salazar_Jo.docx (D149697896)	 	1
SA	Act2-B2.pdf Document Act2-B2.pdf (D125211044)	 	2
SA	Urkund_TesisVF_CanizaresD_HerreraM.pdf Document Urkund_TesisVF_CanizaresD_HerreraM.pdf (D143614554)	 	1

Entire Document

Gerencia General Soluciones de Tecnología Help Desk ISO Soluciones Soluciones de Negocio QA BPO GSD Item Processing Sección Documental Intermediación Electrónica Administración y Finanzas Desarrollo Organizacional Comercial
ESCUELA UNIVERSITARIA DE POSGRADO
IMPLEMENTACIÓN DE BUENAS PRÁCTICAS BASADA EN LA METODOLOGÍA MAC PARA EL ASEGURAMIENTO DE LA CALIDAD DEL DESARROLLO DE SOFTWARE EN LA EMPRESA CANVIA S.A.
Línea de Investigación: Sistema de Información y Optimización
Tesis para optar el grado académico de Doctora en Ingeniería de sistemas
Autora: Galindo Francia, Jesica
Asesor: Gamboa Cruzado, Javier Arturo (ORCID: 0000-0002-0461-4152)
Jurado: Apellidos, Nombres Apellidos, Nombres Apellidos, Nombres
Lima – Perú 2022
RESUMEN



ESCUELA UNIVERSITARIA DE POSGRADO

IMPLEMENTACIÓN DE BUENAS PRÁCTICAS BASADA EN LA METODOLOGÍA
MAC PARA EL ASEGURAMIENTO DE LA CALIDAD DEL DESARROLLO DE
SOFTWARE EN LA EMPRESA CANVIA S.A.

Línea de Investigación:

Sistema de Información y Optimización

Tesis para optar el grado académico de Doctora en Ingeniería de sistemas

Autora:

Galindo Francia, Jesica

(ORCID: 0000-0002-0801-5923)

Asesor:

Gamboa Cruzado, Javier Arturo

(ORCID: 0000-0002-0461-4152)

Jurado:

Manrique Suárez, Luis Humberto

Rodriguez Rodriguez, Ciro

Quispe Prado, Wilber

Lima – Perú

2024

DEDICATORIA

A Dios creador de toda fuente de sabiduría y conocimiento, a mis padres Bernabe Galindo Huamán y Masedonia Gardenia Francia Vilcapoma que cuyas enseñanzas me ayudan a enfrentar la vida cada día, con sus principios y valores inculcados. A mis hermanos Miguel Ángel Galindo Francia y Luis Alberto Galindo Francia por siempre acompañarme en las buenas y malas situaciones, para ellos mi eterno agradecimiento siempre.

AGRADECIMIENTOS

Debo expresar mi más profundo y sincero agradecimiento a todas aquellas personas que con su ayuda han colaborado en la realización del presente trabajo, en especial al Dr. Javier Gamboa Cruzado por su orientación, seguimiento y supervisión continúa, pero sobre todo por la motivación y el apoyo recibido a lo largo del presente estudio.

ÍNDICE

RESUMEN.....	ix
ABSTRACT.....	x
I. INTRODUCCIÓN.....	1
1.1. Planteamiento del problema.....	2
1.2. Descripción del problema	5
1.3. Formulación del problema	9
<i>1.3.1. Problema general</i>	<i>9</i>
<i>1.3.2. Problemas específicos</i>	<i>9</i>
1.4. Antecedentes	10
1.5. Justificación de la investigación	17
1.6. Limitaciones de la investigación	20
1.7. Objetivos	20
<i>1.7.1. Objetivo general</i>	<i>20</i>
<i>1.7.2. Objetivos específicos</i>	<i>21</i>
1.8. Hipótesis.....	21
<i>1.8.1. Hipótesis general.....</i>	<i>21</i>
<i>1.8.2. Hipótesis Específicas.....</i>	<i>21</i>
II. MARCO TEÓRICO.....	23
2.1. Marco conceptual	23
2.2. Bases Teóricas.....	26
2.3. Marco filosófico	76
2.4. Marco individual	77
III. MÉTODO.....	78
3.1. Tipo de investigación.....	78
3.2. Población y muestra	81
3.3. Operacionalización de variables	82

3.4. Instrumentos	85
3.5. Procedimientos	86
3.6. Análisis de datos	86
IV. RESULTADOS	88
V. DISCUSIÓN DE RESULTADOS	153
VI. CONCLUSIONES	157
VII. RECOMENDACIONES	158
VIII. REFERENCIAS	159
IX. ANEXOS	164

ÍNDICE DE TABLAS

		Pág.
Tabla 1.	Resultados de los diez últimos años de CHAOS report (porcentajes).....	4
Tabla 2.	Datos actuales de los indicadores.....	8
Tabla 3.	Situación Actual vs. Situación Propuesta.....	8
Tabla 4.	Ficha para describir una Buena Práctica – Identificación de la práctica.....	29
Tabla 5.	Objetivos de Calidad de Software, atributos y métricas.....	39
Tabla 6.	Diferencia en indicadores a reporte de defectos y tiempos de respuesta.....	48
Tabla 7.	Seguimiento en tiempo real de las incidencias	48
Tabla 8.	Seguimiento de avance de ejecución en regresiones por sistema operativo....	50
Tabla 9.	Seguimiento de avance de ejecución	50
Tabla 10.	Clasificación en categorías y factores de riesgos - nivel de impacto.....	58
Tabla 11.	Ejemplo de Aplicación de Repertory Grid.....	62
Tabla 12.	Descripción del Sistema de Codificación IR-Kanban.....	63
Tabla 13.	Colección de Datos para ACS Estadístico.....	67
Tabla 14.	Prácticas de calidad y seguridad.....	75
Tabla 15.	Operacionalización de variables.....	83
Tabla 16.	Conceptualización de Variables.....	84
Tabla 17.	Investigación de Campo.....	85
Tabla 18.	Investigación Experimental.....	85
Tabla 19.	Procesamiento y análisis de datos.....	86
Tabla 20.	Resultados de la PosPrueba Gc y PosPrueba Ge para los I1, I2, I3, I4 e I5....	131
Tabla 21.	Promedio de los indicadores de la PosPrueba Gc y PosPrueba Ge.....	132
Tabla 22.	Resultados del Indicador 01 - PosPrueba Ge y PosPrueba del Gc.....	137
Tabla 23.	Indicador 01 - Prueba t para medias de las dos muestras.....	139
Tabla 24.	Resultados del Indicador 02 - PosPrueba Ge y PosPrueba del Gc.....	141
Tabla 25.	Indicador 02 - Prueba t para medias de las dos muestras.....	143
Tabla 26.	Resultados del Indicador 03 - PosPrueba Ge y PosPrueba del Gc.....	144
Tabla 27.	Indicador 03 - Prueba t para medias de las dos muestras.....	146
Tabla 28.	Resultados del Indicador 04 - PosPrueba Ge y PosPrueba del Gc.....	147

Tabla 29. Indicador 04 - Prueba t para medias de las dos muestras.....	149
Tabla 30. Indicador 05 - Opciones de las respuestas.....	150
Tabla 31. Resultados del Indicador 05 - PosPrueba Ge y PosPrueba del Gc.....	150

ÍNDICE DE FIGURAS

		Pág.
Figura 1.	Geografía digital.....	3
Figura 2.	Proceso actual del Aseguramiento de la Calidad (AS – IS).....	6
Figura 3.	Proceso del Aseguramiento de la Calidad de Software.....	36
Figura 4.	Etapas de implementación de metodología Six Sigma.....	44
Figura 5.	Gráfico de seguimiento a incidencias que se encuentra en QA	49
Figura 6.	Progreso consolidado de ciclo de pruebas en el sprint.....	51
Figura 7.	Progreso por barra consolidado de ciclo de pruebas en el sprint.....	52
Figura 8.	Propuesta metodológica de ACS.....	55
Figura 9.	Control de riesgo.....	57
Figura 10.	Historial de cambio.....	58
Figura 11.	Documento de petición de mejoras.....	59
Figura 12.	Estructura del Sistema de Codificación.....	63
Figura 13.	Etapas de Secure-SSDM y principales actividades en cada etapa.	68
Figura 14.	Tipo de diseño.....	79
Figura 15.	Flujograma de la Metodología Six Sigma para el área de Calidad	91
Figura 16.	Flujograma de la Metodología ACS.....	92
Figura 17.	Flujograma de la Metodología de Desarrollo Seguro en solitario (SSDM)...	93
Figura 18.	Diagrama de contexto de la nueva metodología MAC.....	95
Figura 19.	Flujograma tentativo – Nivel Macro.....	96
Figura 20.	Flujograma tentativo – Nivel detallado.....	97
Figura 21.	Artefacto de Plan de pruebas.....	100
Figura 22.	Artefacto de estimación de pruebas.....	103
Figura 23.	Artefacto de casos de pruebas.....	108
Figura 24.	Artefacto de observaciones/incidencias.....	110
Figura 25.	Artefacto de Informe de pruebas.....	113
Figura 26.	Organigrama de la empresa.....	115
Figura 27.	Generación del plan de pruebas general.....	121
Figura 28.	Generación de la estimación de pruebas.....	123

Figura 29.	Generación de los casos de pruebas.....	125
Figura 30.	Registro de las observaciones encontradas.....	127
Figura 31.	Estado de las pruebas.....	128
Figura 32.	Informe de las pruebas.....	130
Figura 33.	Pruebas de normalidad – Indicador 01 (PosPrueba Gc - PosPrueba Ge).....	133
Figura 34.	Pruebas de normalidad – Indicador 02 (PosPrueba Gc – PosPrueba Ge).....	134
Figura 35.	Pruebas de normalidad – Indicador 03 (PosPrueba Gc – PosPrueba Ge).....	135
Figura 36.	Pruebas de normalidad – Indicador 04 (PosPrueba Gc – PosPrueba Ge).....	136
Figura 37.	Indicador 01 - Gráfico para el criterio de decisión.....	139
Figura 38.	Indicador 02 - Gráfico para el criterio de decisión.....	142
Figura 39.	Indicador 03 - Gráfico para el criterio de decisión.....	145
Figura 40.	Indicador 04 - Gráfico para el criterio de decisión.....	148
Figura 41.	Indicador 05: Prueba U de Mann – Whitney para la prueba Gc y Ge.....	152

RESUMEN

La presente tesis tiene como objetivo aplicar buenas prácticas basada en una nueva metodología MAC para el aseguramiento de la calidad del desarrollo de Software en empresas públicas o privadas. Actualmente una empresa desarrolladora de Software mantiene una visión y misión, las buenas prácticas son parte fundamental de un proyecto que está en marcha debido a que estas ayudan a tener un mejor rendimiento a la implementación de sus aplicativos, muchas de estas empresas no cuentan con un proceso formal de revisión de calidad hacia el desarrollo, tanto desde la parte inicial del requerimiento hasta el cierre del proyecto. El método se enfoca en proporcionar la mejora continua en el proceso de calidad para promover pruebas efectivas, proporcionar consejos, técnicas y alternativas, entre otras. La investigación es experimental puro se cuenta con el apoyo de CANVIA S.A, se toma 30 proyectos para poder evaluarlos. Los resultados muestran una mejora en la revisión de sus aplicativos disminuyendo y previniendo las fallas de este, siendo así que el cliente estuvo satisfecho en todo el proceso de pruebas. Por lo tanto, se logró aplicar la metodología MAC (Metodología de aseguramiento de la calidad) en la empresa CANVIA, siendo un proceso con lineamientos de buenas prácticas e institucionaliza para que otros proyectos lo pueden utilizar. El presente trabajo sirve de antecedente para cualquier trabajo involucrado al proceso de calidad y de modelo para su aplicación, en empresas desarrolladoras de Software especialmente en las privadas y públicas que adolecen del tratamiento en el proceso.

Palabras clave: aseguramiento de la calidad, buenas prácticas, metodología, desarrollo de software, proceso.

ABSTRACT

This thesis aims to apply best practices based on a new MAC methodology for quality assurance of software development in public or private companies. Currently a software development company maintains a vision and mission, good practices are a fundamental part of a project that is underway because they help to have a better performance to the implementation of their applications, many of these companies do not have a formal quality review process towards the development, both from the initial part of the requirement to the project closure. The method focuses on providing continuous improvement in the quality process to promote effective testing, provide advice, techniques and alternatives, among others. The research is purely experimental with the support of CANVIA S.A., 30 projects are taken to evaluate them. The results show an improvement in the review of their applications, decreasing and preventing failures of this, so that the customer was satisfied throughout the testing process. Therefore, it was possible to apply the MAC methodology (Quality Assurance Methodology) in the CANVIA company, being a process with guidelines of good practices and institutionalized so that other projects can use it. This work serves as a precedent for any work involved in the quality process and as a model for its application in software development companies, especially in private and public companies that lack the treatment in the process.

Keywords: quality assurance, best practices, methodology, software development, process.

I. INTRODUCCIÓN

En una empresa que desarrolla Software siempre tendrá que mantener una visión y misión a donde desea que este vaya y que desea brindar al cliente, las Buenas Prácticas son partes fundamentales de una empresa o proyecto que está en marcha, debido a que estas si ayudan a tener un mejor rendimiento de implementación de los diferentes Desarrollo de Software y este se complemente con el proceso propuesto. Se implementará una nueva metodología para dicho proceso, que sea clave para asegurar la Calidad del producto al estar integrado con el ciclo de vida de Desarrollo de Software y el mantenimiento. Se requiere realizar un análisis que permita identificar requerimientos de aseguramiento y control de calidad en los procesos actuales y tener alternativas para la gestión efectiva del cumplimiento de los requerimientos que solicitan los clientes.

Las Fábricas de Software dan servicios tanto a empresas públicas como privadas, al tener proyectos estas instituciones tienen un proceso definido en base al área de Calidad de Software, en algunos casos no se definieron en su momento es dependiendo de cada empresa que contrata al Outsourcing para este trabajo, el problema abarcará cuando no hay procedimientos institucionalizados debido a que no se le toma la importancia pertinente a la hora de desarrollar Software, que luego será utilizado por un usuario específico por ende se plantea desarrollar una metodología que soporte este proceso y que contenga la mejora continua al área. Por ultimo contendrá la metodología propuesta para el Aseguramiento de la calidad en base a las Buenas Prácticas al caso de estudio, la implementación de mejoras mediante artefactos que soporten el proceso esto ayuda al óptimo desempeño y mejora continua del proyecto, para ser utilizado por los Analistas de Calidad y ser mostrados en resultados a los interesados del proyecto.

1.1. Planteamiento del problema

La industria del software alcanza una posición relevante en el contexto social actual, por su característica de controlar o hacer accesible los adelantos tecnológicos.

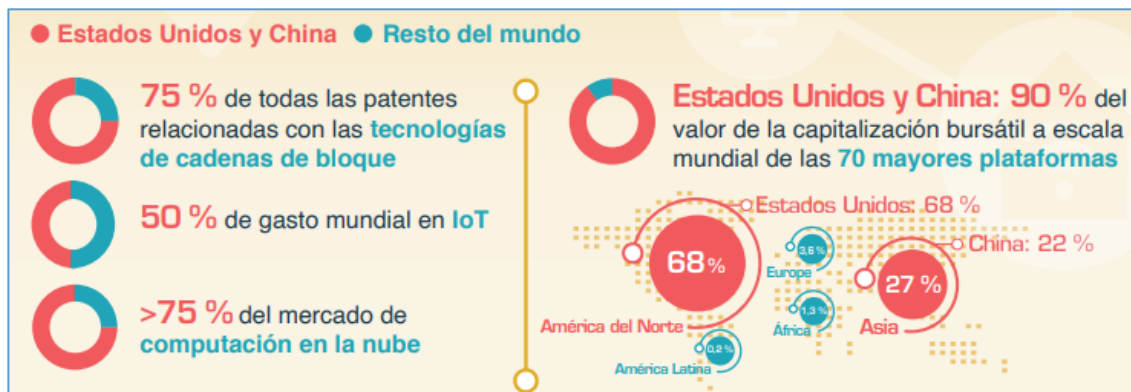
- **Ámbito Internacional**

El software es esencial en la sociedad actual, además, crece cada vez más rápido en complejidad y tamaño:

Por eso es imprescindible asegurar la calidad del programa desarrollado. Un software de calidad no cumple únicamente con la funcionalidad indicada, sino que, además, lo hace de forma eficiente y permite introducir cambios con poca dificultad haciendo que el coste de mantenimiento sea lo más bajo posible. Para garantizar la calidad del software hay que establecer controles, pruebas y métricas, con los que poder establecer de forma objetiva si se cumplen con los estándares. La importancia de estos controles es innegable, pero además éstos han de hacerse de forma continua durante todo el ciclo de vida del producto. Probar, y fallar cuanto antes, tiene un gran impacto económico y de recursos en el desarrollo del software. (López y Fernández, 2020, p. 10).

A futuro, los rubros en los que una proporción mayor de las empresas espera incrementar su dotación de personal son:

Desarrollo de productos, innovación, I+D (25,8%), marketing, investigación de mercados (25,8%) y gestión y aseguramiento de la calidad (19,4%) similar a lo que ocurre en promedio. Los rubros en los que, en cambio, una mayor cantidad de empresas espera una caída del empleo son aquellos vinculados a la gestión de stock (31,0%), mantenimiento (22,6%); la atención al cliente (19,4%) y la relación con proveedores (19,4%) – también similar al promedio. (Dragún et al., 2020, p. 10).

Figura 1*Geografía digital*

- **Ámbito Nacional**

Hoy en día las empresas Peruanas en el ámbito de los proyectos de Desarrollo de Software casi siempre han existido una constante preocupación acerca del posible éxito de estos mismos, y una de las inquietudes más importantes de la Ingeniera de Software es garantizar el éxito:

Uno de los problemas dentro de un proyecto es lo concerniente a un buen sistema de Calidad, cabe precisar que el proceso de Aseguramiento de Calidad en las empresas es primordial debido a que no encontremos errores e estos impacten a la sociedad. En el 2020 el Colegio de Ingenieros del Perú (CIP) analizó el grado de relación entre la gestión de riesgos y la calidad de software en proyectos, en cuanto a los resultados de la evaluación de calidad del software, se identificó que en el 9,2 % de los proyectos, la calidad del software fue regular, mientras que en el 66.7 % fue mala, ya que la capacidad funcional y la percepción de la eficiencia fue baja en el 63,3 % y 64,2 % de los proyectos respectivamente. Ello se refleja en que solo en el 9,2 % de los proyectos se cumplió con todos los requerimientos funcionales y no funcionales propios del sistema. Ello fue producto de que en el 81 % se describió a nivel básico los requerimientos, lo que generó

poca claridad en cuanto al entendimiento de las necesidades del cliente. (Paucar Bernardo et al., 2021, p. 57).

Según los resultados de la tabla 1, en los últimos diez años el porcentaje promedio de los proyectos de TI exitosos es de solo el 30 %, frente al 70 %:

Que corresponde al porcentaje promedio total de proyectos discutidos y cancelados, lo que indudablemente refleja una problemática general en la gestión de proyectos. (Paucar Bernardo et al., 2021, p. 44).

Tabla 1

Resultados de los diez últimos años de CHAOS report (porcentajes)

Proyectos/años	2010	2011	2012	2013	2014	2015	2016	2018
Exitosos	32	29	27	31	28	29	37	33
Discutidos	44	49	56	50	55	52	42	48
Cancelados	24	22	17	19	17	19	21	19

Nota. Los resultados de los diez últimos años CHAOS report ha evidenciado que existe una problemática en la gestión de proyectos. Se puede observar que los proyectos exitosos fueron menos que los proyectos discutidos desde el año 2010 al 2018. Adaptada de *Resultados de los diez últimos años de CHAOS report (porcentajes)*, por Paucar Bernardo et al., 2021, Relación de la gestión de riesgos y calidad de software realizados por los profesionales del Colegio de Ingenieros del Perú del Consejo Departamental de Lima.

Mayormente en un proyecto de desarrollo de Software se tiene tiempos planificados a nivel de las empresas peruanas, trabajan en base a metodologías ágiles y tradicionales tienen diferentes formas de trabajar por ende algunas empresas no tienen un estándar predefinido en sus procesos del día a día.

1.2. Descripción del problema

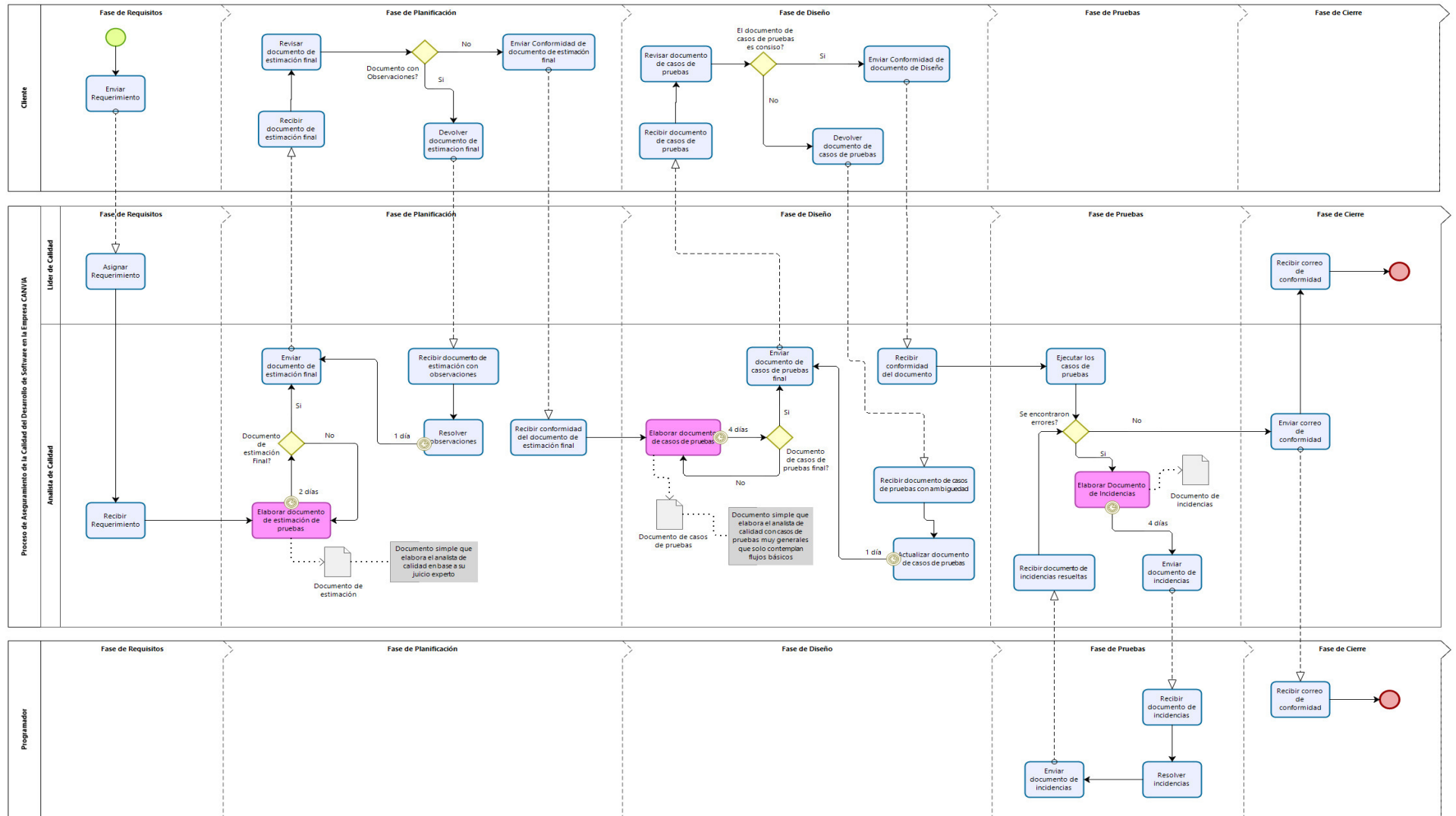
La empresa CANVIA, brinda servicios de tecnologías de la información a empresas privadas o públicas. En el área de calidad se llevan a cabo el proceso de aseguramiento del control de calidad donde utilizan formatos diferentes en cada proyecto de desarrollo de software que se trabaja, como la planificación, el diseño de casos de pruebas, reporte de incidencias y el informe final que se presenta al cliente. Su sede principal está ubicado en Pasaje Acuña 127, Cercado de Lima.

Actualmente esta información se tiene que elaborar según la percepción de cada analista de calidad de cómo desarrollar su formato para que se ingrese la información, esto no es muy factible debido a que no se tiene un estándar para trabajar en base a un proceso que involucra al programador del sistema y al cliente, quien da la conformidad al final de cada requerimiento.

Las planillas que se generan para las fases respectivas en el proceso de aseguramiento de la Calidad mejorado, será capaz de solucionar esta problemática, optimizando este proceso y administrando de una manera más segura y rápida la información necesaria para cumplir con las tareas requeridas. Además se incorporarán procesos para la obtención de datos estadísticos relacionados con las distintas partes que aborda el software en construcción por parte del desarrollador, esto ayudará a trabajar de una forma ordenada y estandarizada para el analista de calidad.

Figura 2

Proceso actual del Aseguramiento de la Calidad (AS – IS)



Sobre el tiempo de entrega, no se tienen los tiempos establecidos para una actividad en el área. “Siendo así plantear los tiempos con cronogramas para poder hacer seguimiento al producto final” (RV, 2021, p. 111).

Sobre el tiempo para la revisión de errores, no se tienen los tiempos establecidos para los errores. “Siendo así plantear documentos que ayuden a este proceso y mejore los tiempos en el levantamiento de observaciones” (Bruchaty y Rastocny, 2020, p. 699).

Sobre el número de errores en pruebas, no se contempla un documento que ayude a las observaciones y las pruebas puedan ser más ágiles en la revisión. “Siendo así plantear documentos detallados de dicha observación y este sea entendible” (Koren y Mani, 2021, p. 189).

Sobre el costo total del proceso, no existe una herramienta que ayude a la gestión del proceso. “Siendo así plantear artefactos detallados que ayuden como herramienta y sirva para todo el ciclo que pasa un desarrollo a calidad” (Hilburn y Towhidnejad, 2021, p. 95).

Sobre la satisfacción del cliente, ultimamente todos los puntajes han sido bajos y hace que el usuario ya no cuente con nuestros servicios. “” (Dudley, 2020, p. 196).

Tabla 2*Datos actuales de los indicadores*

INDICADORES	Datos de Pre – Prueba (Valor promedio)
Tiempo de entrega	48 horas
Tiempo para la revisión de errores	32 horas
Número de errores en pruebas	60 errores
Costo total del Proceso	4000 Soles
Satisfacción del Cliente	Muy malo/ Malo

Para solucionar los problemas descritos anteriormente, se propone la implementación de la nueva metodología MAC, que se obtiene de otras metodologías en común, así se podrá mejorar el proceso de Aseguramiento de la Calidad de Software.

Se presenta a continuación un cuadro comparativo entre la situación actual (AS-IS) y la situación propuesta (TO-BE):

Tabla 3*Situación Actual vs. Situación Propuesta*

Situación Actual (AS – IS)	Situación Propuesta (TO – BE)
Demora en tiempos de entrega (48 horas)	Tiempos de entrega establecidos (32 horas)
Tiempo altos para la revisión de errores (32 horas)	Tiempo bajos para la revisión de errores (24 horas)
Muchos errores en pruebas (60 errores)	Pocos errores en pruebas (30 errores)
Costo total alto del Proceso (4000 soles)	Disminución del Costo total del proceso (2500 soles)
Insatisfacción del Cliente por el servicio brindado (Muy malo)	Satisfacción del Cliente por el servicio brindado (Muy buena)

1.3. Formulación del problema

1.3.1. *Problema general*

¿En qué medida el uso de Buenas Prácticas, basada en la nueva Metodología MAC, mejora el Proceso de Aseguramiento de la Calidad de un proyecto de Desarrollo de Software en la Empresa CANVIA S.A.?

1.3.2. *Problemas específicos*

1. ¿En qué medida el uso de Buenas Prácticas reduce el **tiempo de entrega** del Proceso de Aseguramiento de la Calidad en la empresa CANVIA aplicando la Metodología MAC?.
2. ¿En qué medida el uso de Buenas Prácticas reduce el **tiempo para la revisión de errores** del Proceso de Aseguramiento de la Calidad en la empresa CANVIA aplicando la Metodología MAC?.
3. ¿En qué medida el uso de Buenas Prácticas disminuye el **número de errores en pruebas** del Proceso de Aseguramiento de la Calidad en la empresa CANVIA aplicando la Metodología MAC?.
4. ¿En qué medida el uso de Buenas Prácticas disminuye el **costo total del proceso** de Aseguramiento de la Calidad en la empresa CANVIA aplicando la Metodología MAC?.
5. ¿En qué medida el uso de Buenas Prácticas mejora la **satisfacción del cliente** del Proceso de Aseguramiento de la Calidad en la empresa CANVIA aplicando la Metodología MAC?.

1.4. Antecedentes

Para Aranibar y Méndez (2020) tuvieron como propósito en esta tesis buscar la solución del problema actual que enfrentan las Pequeñas y Medianas Empresas (Pymes), estas son de gran importancia en la economía, representando más del 90 % del total de empresas, generando entre el 60 y 70 por ciento del empleo a nivel mundial (ONU, 2017), y en particular en la industria software. Hoy en día, el uso de software es un recurso indispensable, sin embargo, todavía siguen subsistiendo grandes problemas durante su proceso de desarrollo debido al poco enfoque que se le da a la calidad durante la elaboración de un software. Esto se debe a que los modelos de calidad de procesos en este sector fueron orientadas a grandes empresas, por lo que adaptarlas a las pymes resulta difícil, debido a que conlleva una gran inversión en dinero, tiempo y recursos, que lo hace inaccesible. Como propuesta de solución se propone un modelo de aseguramiento de calidad para ayudar a las pymes a mejorar sus procesos de desarrollo de software, basado en el ciclo de Deming para la mejora continua y en las buenas prácticas del CMMI y la ISO 12207. El modelo propuesto se adapta a cualquier metodología de desarrollo y es fácil de implementar.

Para Navarro (2020) tuvo como propósito que este proyecto surge de una necesidad del departamento de tecnologías de información de la Universidad Autónoma de Bucaramanga de que sus proyectos de desarrollo de software apliquen estándares adoptados por la industria y sigan un marco de buenas prácticas con el fin de reducir los atrasos en los mismos y poder ofrecer calidad en sus productos de software a cada uno de sus clientes. Dentro de los retos estratégicos de la UNAB 2019 – 2024 (Universidad Autónoma de Bucaramanga 2019), se encuentra la transformación digital, debido a esto, el proyecto cobra mucha importancia pues es necesario que los productos que se desarrollen para conseguir este reto tengan la mayor calidad y cumplan con los estándares que se exige actualmente en el mercado.

Para Asencios (2019) tuvo como objetivo que este proyecto de tesis propone la implementación de buenas prácticas para la gestión de servicios de TI para la Zona Registral N° VII – Sede Huaraz ya que la misma, no cuenta con una adecuada gestión de servicios de los sistemas registrales y administrativos en sus oficinas registrales y receptoras, es por ello que, muchas veces el personal de soporte de sistemas que atiende estos eventos, no tiene definido el proceso de escalamiento o los tiempos de atención en que deben ser atendidos según la prioridad del mismo. Muchas veces el servicio de Tecnologías de Información llega a recuperarse, pero no se logra investigar y descubrir las causas raíz de los problemas o peor aún, se tienen incidentes que no son resueltos en realidad. Todo esto repercute en la imagen y la capacidad del personal de la Unidad de Tecnologías de la Información, así como en la continuidad del negocio. Por ello, y tomando en cuenta esta necesidad en la Unidad de Tecnologías de la Información de la Zona Registral N° VII – Sede Huaraz, se presenta el siguiente proyecto de tesis, para poder tener procesos definidos de gestión de servicios de TI con una visión de organización para la atención de estos eventos.

Para Mesones (2021) tuvo como propósito que el presente trabajo de investigación pretende dar respuesta a la problemática creciente sobre la gestión de servicios informáticos en las instituciones del sector salud ya que actualmente las Tecnologías de la Información (TI) se han convertido en un factor importante para la organización y sus objetivos estratégicos. Es de suma importancia comprender la necesidad de visualizar el área de TI como apoyo vital a los procesos de la organización ya que es transversal a las diferentes unidades del negocio. Sin embargo para la mayoría de empresas a nivel mundial la implementación de un Sistema de Gestión de Servicios de Tecnologías de la Información (SGSTI) supone un problema ya que existe un desconocimiento generalizado de los diferentes marcos de referencia y buenas

prácticas para la Gestión de Servicios de TI y la ausencia de personal suficientemente capacitado para su adecuada implementación.

Para Pino et al. (2018) plantearon como propósito que desde hace varios años se viene insistiendo en la “crisis” de la ingeniería del software y en los desastres que los fallos de los productos software pueden llegar a causar en las organizaciones. En la evolución experimentada por la calidad de los sistemas informáticos se ha pasado de un tratamiento centrado fundamentalmente en la inspección y detección de errores en los productos software a una aproximación más sistémica que considera otros componentes organizacionales que afectan a la calidad del software, y especialmente a los procesos del software. En efecto, en la actualidad, la calidad de cualquier producto software no puede ser asegurada simplemente probando solo el producto, es importante considerar que existe una correlación directa entre la calidad del producto obtenido y la calidad del proceso utilizado para su desarrollo. En este sentido, en los últimos años se han publicado diversos estudios y estándares o normas en los que se exponen las buenas prácticas que se deben seguir para la mejora de la calidad de los diferentes componentes de los sistemas informáticos: procesos, proyectos, productos, servicios, personas, etc.

Para Information Systems Audit and Control Association (ISACA, 2019) tuvo como propósito que a la luz de la transformación digital, la información y la tecnología (I&T) se han vuelto cruciales en el apoyo, sostenibilidad y crecimiento de las empresas. Anteriormente, los consejos de administración (juntas de directores) y la alta dirección podría delegar, ignorar o evitar decisiones relacionadas con I&T. En la mayoría de los sectores e industrias, tales actitudes ahora son desaconejadas. La creación de valor para las partes interesadas (es decir, la obtención de beneficios a un costo de recursos óptimo mientras se optimiza el riesgo) es a

menudo impulsada por un alto grado de digitalización en nuevos modelos de negocio, procesos eficientes, innovación exitosa, etc. Las empresas digitalizadas dependen cada vez más de I&T para sobrevivir y crecer.

Para Mohacsi y Ramler (2019) tuvieron como propósito que en esta sección se investiga más a fondo la situación de las pruebas de software y sistemas los problemas observados a las trampas comunes de las pruebas, tal y como las recoge y describe por Firesmith, define las trampas de las pruebas como "cualquier decisión, mentalidad, acción u omisión acción o fallo que innecesariamente y, potencialmente inesperado, hace que las pruebas sean menos eficaz, menos eficiente o más frustrante". Los escollos son conocidos como "antipatrones" o "malas prácticas", que deben evitarse porque sus probables consecuencias negativas pueden ser desastrosas para el proyecto en general. Sin embargo, se ha comprobado que los proyectos Sin embargo, se ha comprobado que los proyectos suelen caer en estas trampas debido a los errores cometidos por los probadores, los gestores, los ingenieros de requisitos y otras partes interesadas.

Para Bhoushika y Roospesh (2019) tuvieron como propósito que en las empresas de software, la calidad del sistema del producto se está convirtiendo en una verdadera preocupación. Hay varias variables que contribuyen a que los productos de software sean malos. Desde principios de la década de 1970, ha surgido el problema del software, ya que los ingenieros de software no han sido capaces de ofrecer un software de alta calidad a tiempo y dentro del presupuesto. Con el tiempo, en sus esfuerzos por producir software de alta calidad y fidelizar a los clientes, las empresas de software se encuentran con numerosos retos. La ineficiencia en la gestión, la administración, el ego de los desarrolladores, calendarios estrictos y tensión, costes adicionales (por ejemplo, la compra de nuevas herramientas), puntos de vista

y valores contradictorios, falta de formación sobre las normas, recursos inadecuados para simplificar el desarrollo de la aplicación, falta de estructura organizativa de gestión de la calidad, escaso conocimiento del proceso. La desaprobación de la alta dirección y de la versión actualizada del sistema, la mala comunicación, la dificultad de codificación y los errores de programación son formas de reducir la planificación de la calidad del software. Los errores deben detectarse y corregirse antes de que el software se entregue al cliente, lo que contribuye a que el producto sea fiable manteniendo la fiabilidad y disminuir los costes. Por lo general, los errores deben detectarse al principio de la fase de pruebas. Además, deben realizarse pruebas de regresión para garantizar la comprobación del componente actualizado del aplicativo.

Para Peláez et al. (2019) tuvieron como propósito que la ingeniería de software, como disciplina, se representa en una serie de subdisciplinas y de buenas prácticas. Una de ellas, el aseguramiento de la calidad del software, y dentro de esta, la calidad de los requisitos. El problema que permite explorar y debatir sobre los requisitos como subdisciplina de la Ingeniería de Software se centra precisamente en la industria del software, como uno de los sectores clave para el desarrollo integral de todo país; esto se debe a que el software impacta diferentes sectores económicos: la educación, el trabajo, la industria, la salud, etc., pero sobre todo contribuye al desarrollo tecnológico e integral de toda sociedad; por ende al ser una industria que ofrece tanto potencial para la sociedad tiene “un gran componente de conocimiento, por lo que requiere un alto desempeño en investigación, desarrollo tecnológico y formación de personas capaces de producir conocimiento y soluciones acordes a las necesidades de todo tipo de organizaciones que lo requieran”.

Para Núñez y González (2020) tuvieron como propósito que el Outsourcing permite a las organizaciones reducir costos, optimizar el uso de recursos, brindar valor agregado en servicios y bienes, y concentrarse en lograr los objetivos estratégicos del negocio. Sin embargo, muchos proyectos de software han fracasado debido al desconocimiento de los factores que intervienen en su planificación. Algunos de estos elementos son una mala comunicación con el proveedor, un seguimiento deficiente del estado de las tareas, la falta de participación activa de los usuarios y las partes interesadas en todas las etapas del proceso y el aseguramiento de la calidad del software. En consecuencia, esta investigación tiene como objetivo ayudar a la gestión del proceso de subcontratación y propone una metodología para su gestión durante el desarrollo del sistema. La metodología consta de 7 fases, cada una con sus entradas, tareas y salidas, que son utilizadas por las siguientes etapas. Esos procesos comparten algunos pasos, pero tratan de manera superflua los aspectos relacionados con las comunicaciones, la gestión de las partes interesadas y el aseguramiento de la calidad. Aun así, varios desafíos requieren atención, incluida la definición de una guía completa para dirigir la administración de la subcontratación.

Para González Obregón et al. (2021) tuvieron como propósito que el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) proporciona el auge de los productos informáticos en diversas esferas de la sociedad. Su presencia, cada vez mayor, en cada uno de los procesos que desarrollan las organizaciones al brindar productos y servicios a sus clientes, requiere que estos evolucionen continuamente y cumplan con altos estándares de calidad. En este ámbito, los productores de software necesitan lograr mayores niveles de competitividad y entregar soluciones capaces de satisfacer las versátiles demandas de la industria. Una de las vías de acercarse a estas metas es la institucionalización de buenas prácticas de ingeniería, gestión (de procesos y de proyectos) y soporte, teniendo como centro

el proceso y como base un enfoque de calidad. En los proyectos de desarrollo de software, es primordial el uso de una metodología que guíe las actividades necesarias para lograr convertir los requerimientos planteados, en un producto con funcionalidades específicas. El éxito de un proyecto se mide en función de tres variables principalmente: costo, tiempo y calidad. Para manejar estas tres variables, los ingenieros de software emplean modelos, procesos y técnicas gerenciales. Las metodologías de desarrollo en conjunto con las mejoras de procesos buscan lograr estas metas.

Para Lei et al. (2021) tuvieron como propósito que las metodologías de mejora de procesos de software apoyan la regulación incesante y el avance del proceso de software para su desarrollo de desempeño. Este desarrollo surge en términos de calidad del producto de software, tiempo, reducción de cambios, etc. El objetivo de la mejora del proceso de software es hacer que el proceso de software sea eficaz y aumentar la calidad del producto de software a través de una evaluación continua. En la práctica, existen varios enfoques y marcos para la mejora de los procesos de software. La toma de decisiones temprana con respecto a la mejora continua del software puede, en última instancia, conducir a desarrollos de software exitosos y al crecimiento de la industria del software y el negocio del software. Teniendo en cuenta la toma de decisiones temprana con respecto a la mejora continua del software, El estudio propuesto ha considerado un sistema de apoyo a la decisión (DSS) para la toma de decisiones basada en multicriterios frente a la gestión de la calidad, el aseguramiento y las métricas. Se consideraron los factores y alternativas necesarios para la mejora continua del software. El proceso del estudio propuesto se llevó a cabo de manera experimental y los resultados fueron lo suficientemente buenos para respaldar el estudio. El estudio ayudará a los profesionales a tomar decisiones tempranas con respecto a la mejora del software basada en la gestión de la calidad.

Para Díaz y Silega (2021) tuvieron como propósito que el número de proyectos de software crecen cada vez más y aumentan su complejidad, al igual que la competitividad entre las empresas de desarrollo de software. Estas empresas buscan lograr que el proceso que guíe el desarrollo de software sea integrado, con el fin de obtener un producto que cumpla con los requisitos definidos, el tiempo, costo y calidad, permitiendo desarrollar un proyecto exitoso. La aplicación temprana de buenas prácticas incrementaría la probabilidad de éxitos en los proyectos de software. Las buenas prácticas incluyen directrices y estándares internacionales, construidos a partir de la experiencia de expertos, con el propósito de mejorar las probabilidades de éxito en el logro de los objetivos de un proyecto. Existe diversidad y heterogeneidad de los estándares disponibles que evalúan los procesos de software, como la familia de normas de ISO 9000, ISO 33000, CMMI, ITIL, COBIT y PMBOK. Sin embargo, la proliferación de los estándares conlleva a que organizaciones se vean abrumadas y confundidas al tomar una decisión sobre la elección y aplicación de un estándar que resuelva sus necesidades y como resultado aumente el costo y tiempo para la evaluación de los mismos.

1.5. Justificación de la investigación

- **Justificación teórica**

Se justifica porque se busca hacer una recolección de lecciones aprendidas existentes en los proyectos de Desarrollo de Software, las cuales actualmente tienen problemas con sus requerimientos en base a una buena revisión de la Calidad de Software. Con la finalidad de cumplir los objetivos planteados de cada uno de los proyectos y no cometer errores comunes que este conlleva al fracaso, se buscará a ayudar a que se analice desde todas las perspectivas del proceso de Aseguramiento de la Calidad, esta metodología está dirigida para las empresas que Desarrollan Software.

- **Justificación práctica**

Se justifica porque permite contar con herramientas y artefactos de apoyo que ayuden a entender y a captar las incidencias que se puedan presentar de forma oportuna, para así poder tener indicadores de mejora continua con un proceso estándar, dirigido para los Analistas de Calidad de Software y todos los involucrados que interfieren en el Proyecto como los Stakeholders quienes van a poder interactuar con estas mejoras al área de Calidad. De este modo, se justifica un proceso de aseguramiento y control de calidad con su respectivo rol y personas encargadas, que le permita a la empresa cumplir con la calidad esperada y puedan generar beneficios para su sostenibilidad.

- **Importancia**

Este proyecto de investigación es importante porque genera una nueva metodología llamada Metodología MAC la cual consolida las buenas prácticas de otras metodologías genéricas acerca del Aseguramiento de la Calidad, además de implementar estándares que soporten el proceso de calidad y se adapten a cualquier organización que desarrolle Software. Esto ayuda a las empresas a tener un procedimiento de mejora continua que con el tiempo puede ser actualizado para que tengas indicadores relevantes.

- **Conveniencia**

Conforme los requerimientos de los proyectos se van mejorando en base a su desarrollo y las revisiones de control de calidad, la necesidad de una nueva metodología que aporte al proceso se hace conveniente, es aquí donde las buenas prácticas son necesarias para el proceso de aseguramiento de la calidad de software se abarque para la entrega de un buen producto.

- **Relevancia social**

Las Buenas Prácticas apoyan a la creación de tener un estándar en el proyecto, esto beneficia a las empresas de trabajar de una forma ordenada tanto en las empresas públicas como privadas, reducirá sus costos, solo hay que saber adecuarlas de acuerdo a una adaptabilidad.

- **Implicaciones prácticas**

La empresa CANVIA S.A. empieza con implementación de un marco de trabajo para poder aplicarlo en su proyecto y esto genera la forma de trabajar en base a nuevos estándares y hace que tenga impacto en la forma de trabajar. Este cambio de una nueva metodología no solo afecta a los procesos de Aseguramiento de Calidad sino al personal encargado de ejecutar este proceso.

- **Valor teórico**

Esta investigación genera discusión tanto sobre el conocimiento existente del área de Aseguramiento de Calidad, dentro del proceso que se busca mejorar ya que de alguna manera confrontan estándares definidas en base a diferentes metodologías existentes para apoyar la implementación.

- **Utilidad metodológica**

Este proyecto de investigación genera una nueva Metodología denominada Metodología MAC para crear nuevos conocimientos válidos y confiables dentro del área de Calidad. También se pone los conocimientos adquiridos durante los años de estudio y trabajo de ingeniería, permitirá sentar las bases teóricas para la problemática en investigación.

1.6. Limitaciones de la investigación

El tiempo es una limitación debido a la poca disponibilidad del personal del área de Calidad, para lo cual se coordina con la gerencia de turno consiguiendo las autorizaciones necesarias para las diversas reuniones de trabajo. Para superar también esta limitación se trabaja con la simulación del proceso en sus diversas etapas haciendo reuniones puntuales con los responsables de área de Calidad para la revisión de los resultados parciales y finales de la investigación.

Las prioridades en cuanto a desarrollo de software son cambiantes, de existir la necesidad de dedicar esfuerzo a otras actividades limitará el desarrollo del proyecto.

Uno de los principales insumos para el desarrollo del proyecto se obtiene de la experiencia del equipo de desarrollo y se cuenta con ellos para ir revisando el avance del proyecto. Sin embargo, si se ellos se encuentran limitados de tiempo para entrevistas o responder encuestas, el proceso de desarrollo del proyecto se puede ver afectado.

1.7. Objetivos

1.7.1. Objetivo general

Implementar las **Buenas Prácticas**, aplicando **Metodología MAC**, para mejorar el **Aseguramiento de la Calidad** de un proyecto de Desarrollo de Software en la Empresa CANVIA S.A.

1.7.2. Objetivos específicos

1. Reducir el tiempo de entrega del Proceso de Aseguramiento de la Calidad en la empresa CANVIA.
2. Reducir el tiempo para revisión de errores del Proceso de Aseguramiento de la Calidad en la empresa CANVIA.
3. Disminuir el número de errores en pruebas del Proceso de Aseguramiento de la Calidad en la empresa CANVIA.
4. Disminuir el costo del Proceso de Aseguramiento de la Calidad en la empresa CANVIA.
5. Mejorar la satisfacción del cliente del Proceso de Aseguramiento de la Calidad en la empresa CANVIA.

1.8. Hipótesis

1.8.1. Hipótesis general

Si se aplica las **buenas prácticas**, mediante la Metodología MAC, entonces mejora el **Aseguramiento de la Calidad** de un proyecto de Desarrollo de Software en la Empresa CANVIA S.A.

1.8.2. Hipótesis Específicas

1. Si se aplica las buenas prácticas, mediante la Metodología MAC, entonces reduce el tiempo de entrega.
2. Si se aplica las buenas prácticas, mediante la Metodología MAC, entonces reduce el tiempo para la revisión de errores.

3. Si se aplica las buenas prácticas, mediante la Metodología MAC, entonces disminuye el número de errores en pruebas.
4. Si se aplica las buenas prácticas, mediante la Metodología MAC, entonces disminuye el costo total del Proceso.
5. Si se aplica las buenas prácticas, mediante la Metodología MAC, entonces mejora la satisfacción del cliente.

II. MARCO TEÓRICO

2.1. Marco conceptual

- **Calidad del Software**

Es la totalidad de la funcionalidad y las características de un producto de software. “Que se relacionan con su capacidad para satisfacer las necesidades declaradas o implícitas” (Graham et al., 2020, p. 248).

- **Aseguramiento de la Calidad**

El aseguramiento de la calidad se asocia con garantizar que las formas estándar de una empresa de realizar diversas tareas se lleven a cabo correctamente:

Dichos procedimientos pueden estar escritos en un manual de calidad que se supone que todos deben seguir. La idea es que si los procesos se llevan a cabo correctamente, los productos producidos serán de mayor calidad. El análisis de la causa raíz y las retrospectivas se utilizan para ayudar a mejorar los procesos y garantizar una calidad más eficaz. Si están siguiendo un estándar de gestión de calidad reconocido, las empresas pueden ser auditadas para asegurarse de que realmente siguen sus procesos prescritos (di lo que haces y haz lo que dices). Aseguramiento de la calidad, parte de la gestión de la calidad enfocada en brindar confianza en que se cumplirán los requisitos de calidad. (Graham et al., 2020, p. 6).

- **Calidad**

Mantener un enfoque en la calidad que produzca entregables que cumplan con los objetivos del proyecto y se alineen con las necesidades, usos y requisitos de aceptación establecidos por los interesados relevantes:

La calidad es el grado en que un conjunto de características inherentes de un producto, servicio o resultado cumple con los requisitos. La calidad incluye la capacidad de satisfacer las necesidades declaradas o implícitas del cliente. El producto, servicio o resultado de un proyecto (mencionado aquí como entregables) es medido para determinar la calidad de la conformidad con los criterios de aceptación y la idoneidad para el uso. (Project Management Institute, 2021, p. 47).

- **Mejora continua**

La mejora continua es el proceso de esfuerzo continuo requerido para mejorar los servicios, productos o procesos:

Este proceso se puede realizar de manera iterativa durante un período. Cómo se hace esto, depende del grado de cambio que buscamos hacer. (Coupland, 2021, p. 140).

- **Metodología**

Una metodología es un sistema de prácticas, técnicas, procedimientos y reglas utilizado por quienes trabajan en una disciplina:

A los equipos del proyecto se les puede requerir que asuman la metodología de la organización matriz. Es decir, el equipo del proyecto adopta un sistema de procesos, gobernanza, métodos y plantillas que proporcionan orientación sobre cómo ejecutar el proyecto. Si bien esto proporciona un grado de coherencia a los proyectos dentro de una organización, la metodología en sí misma todavía puede necesitar personalización para adaptarse a cada proyecto. Las políticas y procedimientos organizacionales prescriben límites autorizados dentro de los cuales el equipo del proyecto puede adaptar. (Project Management Institute, 2021, p. 72).

- **Proceso**

Un conjunto de actividades interrelacionadas o interactivas que transforman entradas en salidas. “Un proceso tomar una o más entradas definidas y las convierte en salidas definidas. Los procesos definen la secuencia de acciones y sus dependencias” (Axelos, 2019, p. 34).

- **Proyecto**

Un proyecto se define como un esfuerzo único, no rutinario, limitado por el tiempo, los recursos y las especificaciones de rendimiento, diseñado para satisfacer las necesidades del cliente:

Una de las características distintivas de la gestión de proyecto es que tiene tanto un principio como un final y normalmente consta de cuatro fases: definición, planificación, ejecución y cierre. La implementación exitosa requiere tantas técnicas y habilidades sociales. Los gerentes de proyecto tienen que planificar y presupuestar proyectos, así como orquestar los aportes de otros. (Larson y Gray, 2021, p. 60).

- **Software de Aplicación**

El software de aplicación es software escrito para, o por, un usuario para realizar un trabajo en particular:

El software de aplicación de propósito general, como una hoja de cálculo electrónica, tiene una amplia variedad de aplicaciones. El software de aplicación de propósito específico, como la gestión de nóminas e inventarios, se usa solo para la aplicación para la que está diseñado. Los programadores de aplicaciones escriben estos programas. (Kaur, 2020, p. 17).

2.2. Bases Teóricas

2.2.1. *Definición de Buenas Prácticas*

Una buena práctica es el proceso de entregar calidad estándar de bienes y servicios en todo momento, de manera eficiente y rentable, utilizando modernos, justos y medios legítimos. Merriam Webster llama a la mejor práctica “un procedimiento que la investigación y la experiencia han demostrado que producen resultados óptimos, y que se establece o propone como un estándar adecuado para generalizar adopción.” Las palabras “resultados óptimos” implican un equilibrio entre los la calidad, el tiempo de principio a fin y el costo.

Wikipedia describe una buena práctica como un método o técnica que ha sido generalmente aceptado como superior a cualquier alternativa porque produce mejores resultados a los logrados por otros medios o se ha convertido en una forma estándar de haciendo cosas:

Simplemente se considera que es la forma más prudente de proceder. Mejor prácticas se utilizan para mantener la calidad como una alternativa a la legislación obligatoria estándares y basarse en la autoevaluación o la evaluación comparativa. Las empresas que no están interesadas en reinventar la rueda pueden ahorrar tiempo y dinero en la investigación y adoptar las mejores prácticas desarrolladas en otros lugares para sus principales actividades y operaciones. Muchos NOCs, IOCs y autoridades reguladoras pueden prescribir condiciones, emitir lineamientos o prácticas recomendadas que forman la base para hacer un trabajo particular de una manera específica en una región. (Pathak, 2022, p. 192).

Una buena práctica es un proceso en constante evolución, se hace más fuerte por la continua investigación para encontrar mejores soluciones que surjan de la mejora del conocimiento base, nueva tecnología o pensamiento innovador:

El esfuerzo y el dinero gastado en la investigación y la prueba y los errores valen los resultados. La documentación y comunicación de las buenas prácticas son esenciales en organizaciones donde los equipos responsables de la entrega de bienes o servicios son sujeto a cambios debido a transferencias, promociones o renuncias de empleados. Si seguidas meticulosamente, tales prácticas ayudan a evitar la gestión de embalses malas prácticas, resultando en pérdida de producción, recuperación de petróleo y gas, e ingresos. El marco de la política de gestión de embalses es extenso. Incluye diversos aspectos de propiedad, desarrollo de recursos, modelos de gobernanza, gestión de riesgos y cumplimiento. Políticas, protocolos, procesos y mejores prácticas están diseñadas en base a la investigación y la experiencia. Estas herramientas ayudan una empresa para hacer frente a situaciones de carácter recurrente y esporádico. Son necesarios estudios e investigaciones específicos para resolver los problemas que surjan de casos excepcionales y complejos. (Pathak, 2022, p. 193).

- **Las Buenas Prácticas como parte de Procesos de Mejora Continua**

La decisión de organizar las actividades y procedimientos institucionales, así como las experiencias, estrategias y métodos de enseñanza y aprendizaje en un contexto permanente de “mejora continua”, expresa la búsqueda de un cambio significativo y pertinente de organización, dinámica y perspectiva, llevado a cabo mediante un proceso sistemático y planificado con la finalidad de mejorar dichos procedimientos y prácticas y con ello, afectar positivamente la formación y aprendizajes que alcanzan los estudiantes.

Lo anterior supone asumir y promover una cultura institucional de mejora continua, es decir, compartir el sentido, los propósitos y fines de este cambio mayor, así como instalar procesos de mejora que han de mantenerse en el tiempo y revisarse permanentemente:

En este camino de cambios, modificaciones y ajustes, van surgiendo también algunas innovaciones que se muestran pertinentes y necesarias para el fortalecimiento de aquellas prácticas y procesos, anteriormente señalados. El cambio, así como la innovación pertinente y con sentido, maduran y se institucionaliza entonces como consecuencia de procesos internos, validados y legitimados por las instituciones y sus comunidades educativas, especialmente por los educadores. Otro aspecto esencial y propio de la Mejora Continua, es la instalación de sistemas y estrategias, rigurosos y planificados de Autoevaluación de la calidad educativa ofrecida. La reflexión y análisis de los resultados de estos procesos diagnósticos, hace posible no sólo la identificación de ámbitos necesarios de afectar de manera de fortalecer y mejorar la calidad referida, sino que ilumina los diseños y la implementación de planes, proyectos, prácticas y acciones de mejoramiento que harán posible lo anterior. La emergencia y validación de buenas prácticas al interior de procesos de mejora, son en definitiva un valioso capital de conocimiento que la comunidad educativa requiere explicitar y socializar al interior de ella, a fin de institucionalizarlas más allá de las personas que estuvieron implicadas en su definición e implementación. (Román, 2019, p. 7).

Algunas consideraciones sobre condiciones y aspectos de la sistematización de Buenas Prácticas:

Siempre será muy importante poder sistematizar las buenas prácticas. Por ello, se sugiere de ser posible, que sea una instancia que se considere desde el inicio de la implementación de proyectos de mejora o de otras experiencias al interior de la mejora.

Esto ayudará a planificarla y durante el desarrollo - a ordenar la información, resguardar instancias de recopilación de opiniones, ideas, percepciones, así como de análisis y reflexión, entre otros. La reflexión sobre la propia práctica, requiere tomar cierta distancia y contar con condiciones tanto institucionales como de las personas que llevarán a cabo la sistematización, que permitan este proceso riguroso de aprendizaje. Antes de iniciar un proceso de sistematización es recomendable definir aspectos claves y sustantivos de ella, de manera de favorecer su desarrollo y aportes buscados. El cuadro a continuación identifica los centrales, en una lógica de preguntas. (Román, 2019, p. 22).

Tabla 4

Ficha para describir una Buena Práctica – Identificación de la práctica

Focos a definir	Descripción
¿Para qué sistematizar?	Explicitar los propósitos de la sistematización; sus objetivos, etc.
¿Qué sistematizar?	Señalar qué de la práctica se sistematizará.
¿Para quién sistematizar?	Público objetivo: ¿Quién usará los productos derivados de la sistematización?
¿Cómo sistematizar?	Definir cuáles serán las estrategias y métodos de investigación que se usarán: revisión de documentación, encuestas, visitas en terreno / campo, grupos focales, talleres y entrevistas, entre otros.

¿Equipo de sistematización? Definir cuál será el equipo que sistematizará la práctica en cuestión. La mirada interna es altamente recomendable: han de ser participantes que conozcan la experiencia, su proceso de implementación.

Nota. Para la identificación de una buena práctica se requiere tener claro el propósito a lo que se mejorará de algo que ya tuvo éxito y rindió como un excelente servicio. Adaptada de *Ficha para describir una Buena Práctica – Identificación de la práctica*, por Román, 2019, Manual de Buenas Prácticas manual en ciclos de mejora continua.

2.2.2. Definición de Aseguramiento de la Calidad de Desarrollo de Software

El software se encuentra cada vez más presente en nuestras vidas y en muchos elementos de nuestro entorno:

Por este motivo, la implicación en el riesgo del software propenso a errores es una cuestión estrechamente ligada a la calidad. El objetivo de Aseguramiento de la calidad es proporcionar a la dirección una perspectiva adecuada del proceso que utiliza el proyecto software y de los productos a crear. El aseguramiento de la calidad software implica la revisión y la inspección de los productos y las actividades del software, con objeto de verificar que se cumplen los procedimientos y estándares aplicables, y de proporcionar a los responsables del proyecto software y a otros directivos los resultados de dichas revisiones e inspecciones. (Jiménez de Parga, 2021, p. 68).

Por lo tanto, la misión del Aseguramiento de la calidad, tiene como referencia conceptual lo siguiente:

Planificar las actividades de aseguramiento de la calidad. Verificar que los productos y el proceso software cumplen los requisitos, estándares y procedimientos establecidos. Informar a la dirección y a las personas relacionadas con el proyecto sobre los resultados del aseguramiento de la calidad. Elevar a la dirección los aspectos que no pueden solucionarse en el contexto del proyecto. (Jiménez de Parga, 2021, p. 68).

Las actividades que recaen sobre el grupo de aseguramiento de la calidad para que se pueda seguir con el proceso:

Realizar el plan de aseguramiento de la calidad en las primeras fases del proyecto. Establecer las actividades del personal de aseguramiento de la calidad respecto al plan de aseguramiento de calidad. Participar en la definición y revisión del plan del proyecto. El grupo de aseguramiento de la calidad asesora en cuanto a la elaboración y revisión de planes, normas y procedimientos de desarrollo de Software. Revisar las actividades de desarrollo de software con el fin de verificar su cumplimiento. Inspeccionar los productos software con el propósito de verificar su corrección (antes de entregar el producto se evalúa en función a estándares, procedimientos y requisitos del contrato). Informar periódicamente al equipo de desarrollo. Documentar las desviaciones de las actividades y productos de software. Revisar periódicamente las propias actividades del grupo de aseguramiento de calidad con el personal de aseguramiento de la calidad del cliente. (Jiménez de Parga, 2021, pp. 68-69).

a). Elementos del aseguramiento de la calidad

El aseguramiento de la calidad del software, “abarca una amplia gama de preocupaciones y actividades que se centran en la gestión de la calidad del software” (Pressman y Maxim, 2020, p. 341).

- **Estándares:** El IEEE, ISO y otras organizaciones de estándares han producido una amplia gama de estándares de ingeniería de software y documentos relacionados. Los estándares pueden ser adoptados voluntariamente por una organización de ingeniería de software o impuestos por el cliente u otras partes interesadas. El trabajo de aseguramiento de la calidad del Software es garantizar que se sigan los estándares que se han adoptado y que todos los productos de trabajo se ajusten a ellos.
- **Revisiones y auditorías:** Las revisiones técnicas son una actividad de control de calidad realizada por ingenieros de software para ingenieros de software. Su intención es descubrir errores. Las auditorías son un tipo de revisión realizada por el personal de Aseguramiento de la Calidad de Software (SQA) con la intención de garantizar que se sigan las pautas de calidad para el trabajo de ingeniería de Software. Por ejemplo, se podría realizar una auditoría del proceso de revisión para garantizar que las revisiones se realicen de una manera que conduzca a la mayor probabilidad de descubrir errores.
- **Pruebas:** Las pruebas de software son una función de control de calidad que tiene un objetivo principal: encontrar errores. El trabajo de aseguramiento de la calidad de software (SQA) es garantizar que las pruebas se planifiquen correctamente y se realicen de manera eficiente para que tengan la mayor probabilidad de lograr su objetivo principal.

- **Recopilación y análisis de errores/defectos:** La única forma de mejorar es medir cómo lo estás haciendo. El aseguramiento de la calidad de software (SQA) recopila y analiza datos de errores y defectos para comprender mejor cómo se introducen los errores y qué actividades de ingeniería de software son las más adecuadas para eliminarlos.
- **Gestión del cambio:** El cambio es uno de los aspectos más disruptivos de cualquier proyecto de software. Si no se gestiona adecuadamente, el cambio puede generar confusión, y la confusión casi siempre conduce a una mala calidad. El aseguramiento de la calidad de software (SQA) garantiza que se hayan instituido prácticas adecuadas de gestión de cambios.
- **Educación:** Cada organización de software quiere mejorar sus prácticas de ingeniería de software. Un contribuyente clave para la mejora es la educación de los ingenieros de software, sus gerentes y otras partes interesadas. La organización del aseguramiento de la calidad de software (SQA) toma la delantera en la mejora de procesos de software y es un proponente y patrocinador clave de programas educativos.
- **Gestión de proveedores.** Se adquieren tres categorías de software de proveedores de software externos: Software empaquetado (por ejemplo: Microsoft Office), un programa a medida que proporciona una estructura esquelética básica que se adapta a las necesidades de un comprador, y software contratado que está diseñado y construido a la medida a partir de las especificaciones proporcionadas por la organización del cliente. El trabajo de la organización del aseguramiento de la calidad de software (SQA) es garantizar que el software de alta calidad resulte sugiriendo prácticas de calidad específicas que el proveedor debe seguir (cuando sea

posible) e incorporando mandatos de calidad como parte de cualquier contrato con un proveedor externo.

- **Gestión de seguridad:** Con el aumento de los delitos cibernéticos y las nuevas regulaciones gubernamentales con respecto a la privacidad, cada organización de software debe instituir políticas que protejan los datos en todos los niveles, establecer una protección de firewall para las aplicaciones móviles y asegurarse de que el software no haya sido manipulado internamente. El aseguramiento de la calidad de software (SQA) garantiza que se utilicen los procesos y la tecnología apropiados para lograr la seguridad del software.
- **La seguridad:** Debido a que el software es casi siempre un componente fundamental de los sistemas calificados por humanos (por ejemplo, aplicaciones de automóviles o aeronaves), el impacto de los defectos ocultos puede ser catastrófico. El aseguramiento de la calidad de software (SQA) puede ser responsable de evaluar el impacto de la falla del software y de iniciar los pasos necesarios para reducir el riesgo.

Gestión de riesgos, aunque el análisis y la mitigación del riesgo es la preocupación de los ingenieros de software:

La organización de aseguramiento de la calidad de software (SQA) asegura que las actividades de gestión del riesgo se lleven a cabo correctamente y que se hayan establecido planes de contingencia relacionados con el riesgo. (Pressman y Maxim, 2020, p. 342).

Además de cada una de estas preocupaciones y actividades, mejorar el proceso y tomar las buenas prácticas:

El aseguramiento de la calidad de software (SQA) trabaja para garantizar que las actividades de soporte de software (por ejemplo, mantenimiento, líneas de ayuda, documentación y manuales) se realicen o produzcan con la calidad como una preocupación dominante. (Pressman y Maxim, 2020, p. 343).

b). Tareas, objetivos y métricas del Aseguramiento de la Calidad de Software (SQA)

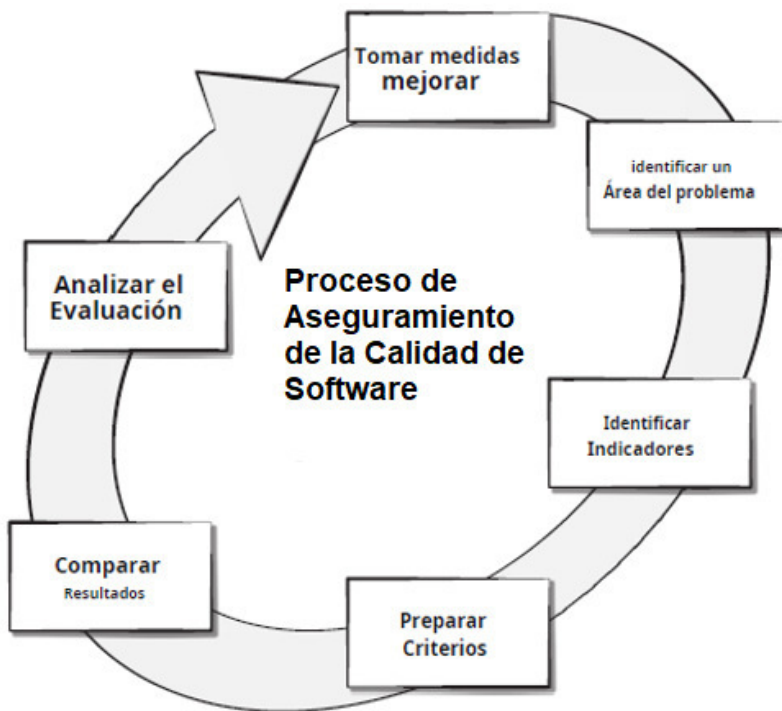
El aseguramiento de la calidad del software se compone de una variedad de tareas asociadas con dos componentes diferentes: los ingenieros de software que realizan el trabajo técnico y un grupo de Aseguramiento de la Calidad (SQA) que tiene la responsabilidad de la planificación, supervisión, mantenimiento de registros, análisis y generación de informes del aseguramiento de la calidad.

El aseguramiento de la calidad del software moderno a menudo se basa en datos, como se muestra en la Figura 3:

Los interesados en el producto definen metas y medidas de calidad, se identifican áreas problemáticas, se miden indicadores y se determina si se necesitan o no cambios en el proceso. Los ingenieros de software abordan la calidad (y realizan actividades de control de calidad) aplicando métodos y medidas técnicas sólidas, realizando revisiones técnicas y realizando pruebas de software bien planificadas. (Pressman y Maxim, 2020, pp. 343-344).

Figura 3

Proceso del Aseguramiento de la Calidad de Software



b1. Tareas del Aseguramiento de la Calidad de Software

La misión del grupo de Aseguramiento de la Calidad de Software (SQA) es ayudar al equipo de software a lograr una alta calidad del producto final. El Software Engineering Institute recomienda un conjunto de actividades de Aseguramiento de la Calidad de Software (SQA) que abordan la planificación, supervisión, mantenimiento de registros, análisis e informes de control de calidad. Estas actividades son realizadas (o facilitadas) por un grupo SQA independiente que:

- **Prepara un plan SQA para un proyecto.** El plan se desarrolla como parte de la planificación del proyecto y es revisado por todas las partes interesadas. Las actividades de garantía de calidad realizadas por el equipo de ingeniería de

software y el grupo de Aseguramiento de la Calidad de Software (SQA) se rigen por el plan. El plan identifica evaluaciones a realizar, auditorías y las revisiones que se realizarán, los estándares aplicables al proyecto, los procedimientos para el informe y seguimiento de errores, los productos de trabajo producidos por el grupo de Aseguramiento de la Calidad de Software (SQA) y los comentarios que se proporcionarán al equipo de software.

- **Participa en el desarrollo de la descripción del proceso de Software del proyecto.** El equipo de Software selecciona un proceso para el trabajo a realizar. El grupo de Aseguramiento de la Calidad de Software (SQA) revisa la descripción del proceso para el cumplimiento de la política de la organización, los estándares de software interno, los estándares impuestos externamente (por ejemplo, ISO-9001) y otras partes del plan del proyecto de software.
- **Revisa las actividades de ingeniería de software para verificar el cumplimiento del proceso de Software definido.** El grupo de Aseguramiento de la Calidad de Software (SQA) identifica, documenta y rastrea las desviaciones del proceso y verifica que se hayan realizado las correcciones.
- **Audita los productos de trabajo de software designados para verificar el cumplimiento de los definidos como parte del proceso de software.** El grupo SQA revisa productos de trabajo seleccionados; identifica, documenta y rastrea las desviaciones; verifica que se han hecho las correcciones; e informa periódicamente los resultados de su trabajo al director del proyecto.
- **Garantiza que las desviaciones en el trabajo de software y los productos de trabajo se documenten y manejen de acuerdo con un procedimiento documentado.** Se pueden encontrar desviaciones en el plan del proyecto, la

descripción del proceso, los estándares aplicables o los productos de trabajo de ingeniería de software.

- **Registra cualquier incumplimiento e informa a la alta dirección.** Los elementos de incumplimiento se rastrean hasta que se resuelvan.

Además de estas actividades, el grupo de Aseguramiento de la Calidad de Software (SQA). “Coordina el control y la gestión del cambio y ayuda a recopilar y analizar las métricas del software” (Pressman y Maxim, 2020, pp. 343-344).

b2. Objetivos, atributos y métricas

Las actividades de Aseguramiento de la Calidad de Software (SQA) descritas en la sección anterior se realizan para lograr un conjunto de objetivos pragmáticos:

- **Requisitos de calidad:** La corrección, integridad y consistencia del modelo de requisitos tendrá una fuerte influencia en la calidad de todos los productos de trabajo que siguen. El Aseguramiento de la Calidad de Software (SQA) debe asegurarse de que el equipo de software haya revisado correctamente el modelo de requisitos para lograr un alto nivel de calidad.
- **Calidad de diseño.** Cada elemento del modelo de diseño debe ser evaluado por el equipo de software para garantizar que muestre alta calidad y que el diseño en sí se ajuste a los requisitos. El Aseguramiento de la Calidad de Software (SQA) busca atributos del diseño que sean indicadores de calidad.
- **Calidad del código.** El código fuente y los productos de trabajo relacionados (por ejemplo, otra información descriptiva) deben cumplir con los estándares de

codificación locales y exhibir características que faciliten el mantenimiento. El Aseguramiento de la Calidad de Software (SQA) debe aislar aquellos atributos que permitan un análisis razonable de la calidad del código.

- **Eficacia del control de calidad.** Un equipo de software debe aplicar recursos limitados de una manera que tenga la mayor probabilidad de lograr un resultado de alta calidad. El Aseguramiento de la Calidad de Software (SQA) analiza la asignación de recursos para revisiones y pruebas para evaluar si se están asignando de la manera más eficaz.

La Tabla 5, identifica los atributos que son indicadores de la existencia de calidad para cada una de las metas discutidas. “También se muestran las métricas que se pueden utilizar para indicar la fuerza relativa de un atributo” (Pressman y Maxim, 2020, pp. 345-346).

Tabla 5

Objetivos de Calidad de Software, atributos y métricas

Meta	Atributo	Métrica
Requisito de calidad	Ambigüedad	– Número de modificadores ambiguos (p. ej., muchos, grandes, aptos para humanos)
	Lo completo	– Número de TBA, TBD
	Comprensibilidad	– Número de secciones/subsecciones
	Volatilidad	– Número de cambios por requisito – Tiempo (por actividad) cuando se solicita el cambio.
	Trazabilidad	– Número de requisitos no trazables al diseño/código

	Claridad del modelo	<ul style="list-style-type: none"> – Número de modelos UML – Número de páginas descriptivas por modelo – Número de errores UML
Calidad de diseño	Integridad arquitectónica	<ul style="list-style-type: none"> – Existencia del modelo arquitectónico
	Compleitud de los componentes	<ul style="list-style-type: none"> – Número de componentes que rastrean el modelo arquitectónico. – Complejidad del diseño procedimental.
	Complejidad de la interfaz	<ul style="list-style-type: none"> – Número promedio de selección para llegar a una función o contenido típico. – Idoneidad del diseño.
	Patrones	<ul style="list-style-type: none"> – Número de patrones utilizados
Calidad de código	Complejidad	<ul style="list-style-type: none"> – Complejidad ciclomática
	Mantenibilidad	<ul style="list-style-type: none"> – Factores de diseño
	Comprensibilidad	<ul style="list-style-type: none"> – Porcentaje de comentarios internos. – Convenciones de nomenclatura de variables – Porcentaje de componentes reutilizados
Eficacia del control de calidad	Asignación de recursos	<ul style="list-style-type: none"> – Porcentaje de horas de personal por actividad
	Tasa de finalización	<ul style="list-style-type: none"> – Actual vs. tiempo de finalización presupuestado
	Revisión de la eficacia	<ul style="list-style-type: none"> – Métricas de revisión
	Eficacia de las pruebas	<ul style="list-style-type: none"> – Número de errores encontrados y criticidad – Esfuerzo requerido para corregir un error – Origen del error

Nota. El proceso de aseguramiento de la calidad está compuesto por métricas ya que es importante para un proyecto de Software con eso puede tomar decisiones para la mejora continua. Adaptada de *Objetivos de Calidad de Software, atributos y métricas*, por Pressman y Maxim, 2020, *Becoming an Agile Software Architect: Strategies, practices, and patterns to help architects design continually evolving solutions.*

2.2.3. Teorías pertinentes de la investigación

a. Metodología Six Sigma para el área de Aseguramiento de la Calidad de Software

La importancia de los procesos en la gestión radica en alcanzar resultados con más eficiencia y eficacia:

Lo cual garantiza aseguramiento de calidad; inicialmente se debe delimitar las jerarquías y poder prestar más atención a los resultados de manera individual con esto, se tiene la posibilidad de estructurar y asignar las tareas, realizar el respectivo seguimiento y control, fortalecimiento de la mejora continua y finalmente un producto o servicio con más alta calidad. Partiendo de lo anterior ¿Es necesario que una organización tenga sus procesos definidos y claros? ¿Cómo la implantación e implementación de un proceso fortalece toda una vertical? Actualmente se tiene como referencia la organización colombiana Rappi, el cual presta servicios de e-commerce convirtiéndose en un sitio virtual que permite a los usuarios obtener productos y servicios desde su dispositivo móvil; una de las verticales (áreas) más representativas es “Restaurantes”, en la cual los usuarios pueden seleccionar el restaurante, menú y precio de su interés, con el fin de generar un pedido que satisfaga sus necesidades o antojos a la hora de comer. Debido a su importancia, esta vertical debe asegurar la calidad de su plataforma digital y de sus procesos internos, de tal manera que se pueda

optimizar los recursos de manera efectiva para mitigar e identificar los defectos que puedan impactar de manera negativa la experiencia del cliente. Para lograr el objetivo de investigación, se toma como base principal la metodología Six Sigma, el cual permite la optimización de procesos y mejoramiento de calidad; esta metodología permite el mejoramiento continuo de la calidad en los procesos existentes, diseño e implementación de indicadores de gestión para la toma de decisiones. (Jimenez y Jimenez, 2020, p. 17).

Actualmente, el sector de las T.I. es uno de los más relevantes en el mercado, por este motivo, debe ser garantizada la calidad de los desarrollos partiendo del cumplimiento de requerimientos establecidos y enfocados en los clientes. El aseguramiento de la calidad permite garantizar el cumplimiento de los procesos del software de manera integral desde la definición del requerimiento hasta el uso por parte del usuario final, para lograr esto es necesario alinear los procesos bajo las buenas prácticas expuestas en el ISTQB.

Rappi es conocido por brindar soluciones oportuna gracias a su portafolio comercial en diferentes países latinoamericanos:

No obstante, tiene oportunidades de mejora en sus procesos que se evidencian en la necesidad de contar con planes estratégicos, procesos documentados e información centralizada, impactando la implementación de indicadores de calidad que repercuten en la toma de decisiones para la mejora continua, gestión de actividades y optimización de recursos de la organización; en consecuencia, su crecimiento y nivel de calidad en sus procesos no era el esperado, aun así, logran superar este desafío y pueden continuar operando, sin dejar de lado la idea de un crecimiento inestable. Es una empresa muy joven que se ha dedicado a crecer, dejando de lado temas como procesos, metodología

y buenas prácticas para la mejora de su calidad interna. Se puede plantear un ejemplo basados en el área de tecnología de Rappi, más específicamente la vertical de “Restaurantes”, donde no se tenían procesos documentado, socializado y aprobado por los involucrados en el proceso de aseguramiento de calidad. Por otra parte, se evidenció ausencia de una jerarquía definida, tareas específicas de acuerdo a los cargos, tiempos definidos para cada tarea, no se tenían herramientas para la ejecución de tareas para personas que ejecutan pruebas, la herramienta para el registro de información no aporta gran información para poder realizar informes generales del área, no se tenía en cuenta la guía de buenas prácticas establecidas en el ISTQB. Partiendo de lo anterior, se puede establecer de entrada un problema de efectividad, donde se estaba perdiendo recursos como tiempos de ejecución, recursos tecnológico y humanos; en pro de definir un proceso y plantear métricas específicas que aportan información consistente y oportuna se optó por la implementación de la metodología Six sigma, permitiendo una mejor planeación, organización, gestión, control, resultados de actividades desarrolladas y que claramente se evidencian con la operatividad de la interfaz para usuarios finales. (Jimenez y Jimenez, 2020, p. 18).

Propuesta de Metodología Six Sigma

1. Materiales y Métodos

La implementación de la metodología Six Sigma para la mejora de sus sistemas de producción y aseguramiento de calidad el área de “QA” en la vertical de “Restaurantes” se soporta en 5 etapas que son:

Figura 4

Etapas de implementación de metodología Six Sigma



a). Etapa de definir

En esta etapa se realiza un diagnóstico de la situación actual de QA en la vertical, por medio del método de observación directa y entrevistas a personas que están directamente relacionadas.

Se realizan actividades como documentar el proceso actual de la vertical, definir las entradas y salidas del proceso, tales como historias de usuario, tareas, defectos reportados e identificar actividades que se pueden mejorar. En el diagnóstico se encuentra lo siguiente: La documentación asociada a procesos,

pruebas, diseño y ejecución de casos de prueba, es escasa y en su defecto obsoleta, pues los tester no cuentan con el tiempo suficiente para estar documentando y actualizando cada cambio que se genera por versión; se tiene registro de información en Jira del último mes (Sistema de información para almacenar y procesar datos), esta data almacenada no es suficiente para extraer y analizar métricas que sustente el trabajo realizado.

b). Etapa de medir

Se recopila la data actual del rendimiento del proceso, usando la información que se tiene almacenada en Jira.

Se realizan actividades como recolección de información adicional al desempeño de la línea base, consideración de información que es importante y nunca se tuvo en cuenta, como el tiempo de ejecución, etc. Se identifican las métricas críticas, como cantidad de defectos reportados por tarea o en regresión. Jira permite la generación de tableros de control y herramientas de gestión no se usaba de manera oportuna, no se documentaba eficientemente, ausencia de data, no se podía montar indicadores, registro total en Jira.

c). Etapa de analizar

Se analiza la información almacenada en el sistema de información, determinando las causas que más impacten en el proceso. Se realizan las siguientes actividades: Establecer las causas verdaderas que no permiten un óptimo proceso y el producto como el re-trabajo, falta de insumos, no hay cronograma, dependencias y centralización de información en personas. Adicional, se identifica que la ausencia de planeación y un proceso para lograr alcanzar los objetivos planeados basados en

buenas prácticas son la fuente de generación de errores, cuello de botella, re-trabajos por parte del QA, como el revisar varias veces una misma incidencia; el número de bugs en producción no identificados en fases tempranas generaban que generaba más costos al corregirlo.

d). Etapa de implementar

De acuerdo a los resultados del análisis, se opta por una solución oportuna y que se ajuste a la problemática evidenciada en fases anteriores. En esta etapa se realiza: Primero, se diseña del proceso, se socializa, evalúa y se acompaña de la TM4J, que es una herramienta para la gestión de pruebas, usada para un mejor control y trazabilidad en los proyectos. El registro y control adecuado de información en la herramienta es más óptima, pues cada persona empieza a registrar información para su control en el Jira; se empieza a registrar y generar métricas de eficiencia, tales como tiempo usado, utilización de recursos y conformidad de la eficiencia y finalmente se estandariza el proceso que se ha definido, de tal manera que se optimice el tiempo y los recursos para cada actividad. A cada tester se le asigna un dispositivo iOS y un Android con el fin de optimizar tiempos y actividades; se realizan capacitaciones de negocio, metodología y técnica, para fortalecer los conocimientos del equipo.

e). Etapa de controlar

Tras validar la implementación, se desarrollan reportes de calidad y efectividad que sirven como control para el aseguramiento de la continuidad del proceso, que este sea optimizado y constantemente actualizado:

Se evidencian los siguientes resultados y actividades: Al implementar la herramienta TM4J en Jira, el control de las pruebas por parte del líderes y managers es más completa, permitiendo una mejor toma de decisiones y liderazgo en el equipo; se define un cronograma específico, donde se plantean fechas de ingreso de tareas a probar, fechas para diseño y documentación de casos de prueba, y tiempos específicos para la ejecución de Release. Cada tester puede informarse y llevar un control de las ejecuciones realizadas y las que tendrá a futuro, para poder alistar los recursos necesarios, mejorando los tiempos de respuesta y ejecución. Se realizan capacitaciones y evaluaciones de control y conocimiento periódicamente, con el objetivo de fortalecer los conocimientos, habilidades y aptitudes de los tester para un óptimo logro de tareas y proyectos. (Jimenez y Jimenez, 2020, p. 21).

2. Resultados

Se evidencia una notable mejoría en el proceso y procedimientos del equipo de QA en la vertical de Restaurantes de Rappi:

Los aspectos que se han implementado y fortalecido son: medición de tiempos de respuesta, medición de incidencias entrantes con su correspondiente responsable, medición de defectos reportados con su respectiva prioridad, cantidad de dispositivos por persona, documentación, etc. Al implementar el TM4J en Jira exitosamente, se evidencia documentación en tiempo real más completa y conveniente para un mejor seguimiento y control de los tester y del proceso en general. Adicionalmente, a cada tester le fueron asignadas

herramientas de manera individual, con el fin de mejorar el rendimiento y reducir dependencias. (Jimenez y Jimenez, 2020, p. 22).

Tabla 6

Diferencia en indicadores a reporte de defectos y tiempos de respuesta

Reportes por versión	Antes	Ahora	Mejora en porcentaje
Bugs reportados	20	5	75%
Bugs reportados	8	2	75%
Bugs de severidad alta	7	1	85%
Tiempo de respuesta	2 días	3 horas	93%

Nota. El proceso de la revisión de calidad tiene indicadores sobre las observaciones reportadas, se puede observar el reporte por revisión del antes y ahora de cómo ha reducido las incidencias y el tiempo de respuesta a la revisión fue menos también. Adaptada de *Ejemplo de la diferencia en indicadores relacionado a reporte de defectos y tiempos de respuesta*, por Jimenez y Jimenez, 2020, Implementación de la metodología Six Sigma para el mejoramiento y optimización de procesos en el equipo aseguramiento de calidad de software de la vertical de Restaurantes de Rappi.

A continuación, se muestran diferentes esquemas de los tableros de seguimiento por sprint:

Tabla 7

Seguimiento en tiempo real de las incidencias

RDR-QA DISPOSITIVOS										
🔍	👤	Tester	Estado	Tipo de incidencia	Resumen	Clave	Prioridad	Responsable	Versión afectada	Versión corregida

	Tester 1	<ul style="list-style-type: none"> • Por empezar pruebas • En proceso de pruebas 	<ul style="list-style-type: none"> • Historia • Tarea • Bug 	Descripción a modo de resumen	ID de la incidencia	<ul style="list-style-type: none"> • Alta • Media • Baja 	Nombre de la persona encargada del desarrollo	Versión donde se presenta el defecto (No aplica para historias y tareas)	Versión donde se presenta la modificación
Android	Anie Jimenez	Testing in progress	Bug	Saturación Modal "No disponible" se manda vacío.	RDR-6464	Baja	Vinicus Pagani	Android_6,35	Android_6,35

Nota. Se puede observar el detalle del seguimiento de la revisión del tester para una aplicación a un sistema operativo para una asignación del proyecto correspondiente. Adaptada de *Seguimiento en tiempo real de las incidencias que se encuentran en QA dispositivos con información para su gestión*, por Jimenez y Jimenez, 2020, Implementación de la metodología Six Sigma para el mejoramiento y optimización de procesos en el equipo aseguramiento de calidad de software de la vertical de Restaurantes de Rappi.

Figura 5

Gráfico de seguimiento a incidencias que se encuentra en QA

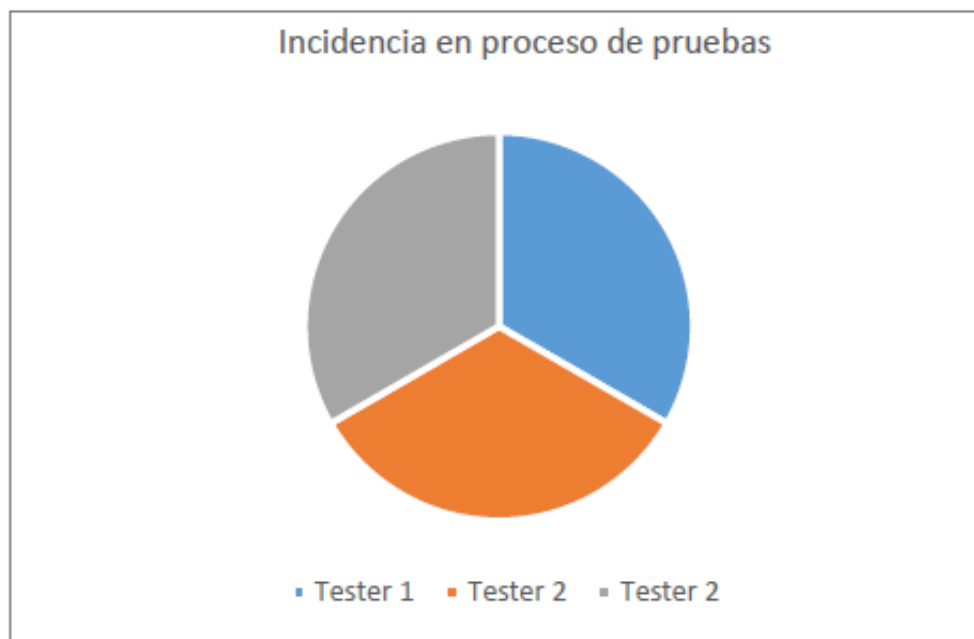


Tabla 8

Seguimiento de avance de ejecución en regresiones por sistema operativo

RDR-QA DISPOSITIVOS (REGRESIÓN)										
Ciclo de prueba	Total de casos a ejecutar	% no ejecutado	% en progreso	% pasados	% fallidos	% bloqueados	% no aplican	Total de casos ejecutados	Total de casos restantes	% total de avance
IDciclo_ Android	23	0 %	4.35 % (1)	56.52 %	26.09 %	13.04 %	0.0 %	22	1	95.65%
IDciclo_i OS_Vers	25	0%	0.0%	72%	24%	4%	0.0%	25	0	100%

Nota. Se puede observar el detalle de las pruebas de regresión por sistema operativo, se identifica el total de casos de pruebas a ejecutar y considerar el avance pertinente. Adaptada de *Seguimiento de avance de ejecución en regresiones por sistema operativo*, por Jimenez y Jimenez, 2020, Implementación de la metodología Six Sigma para el mejoramiento y optimización de procesos en el equipo aseguramiento de calidad de software de la vertical de Restaurantes de Rappi.

Tabla 9

Seguimiento de avance de ejecución

RDR-QA DISPOSITIVOS (REGRESIÓN POR PERSONA)_IOS										
Tester	Total de casos a ejecutar	% no ejecutado	% en progreso	% pasados	% fallidos	% bloqueados	% no aplican	Total de casos ejecutados	Total de casos restantes	% total de avance
Nombre del tester	17	35.29 %	5.88 %	47.06 %	11.76 %	0.0 %	0.0 %	10	7	58.82%

Nombre del tester	16	6.25%	62.5%	62.5%	25%	0.0 %	0.0 %	14	2	87.5 %
-------------------	----	-------	-------	-------	-----	-------	-------	----	---	--------

Nota. Se puede observar el detalle de las pruebas de regresión por persona, se identifica el total de casos de pruebas a ejecutar y considerar el avance pertinente, esto sirve para el seguimiento general de las pruebas. Adaptada de *Seguimiento de avance de ejecución en Regresiones de un sistema operativo por persona*, por Jimenez y Jimenez, 2020, Implementación de la metodología Six Sigma para el mejoramiento y optimización de procesos en el equipo aseguramiento de calidad de software de la vertical de Restaurantes de Rappi.

Información relacionada al sprint en general:

Figura 6

Progreso consolidado de ciclo de pruebas en el sprint

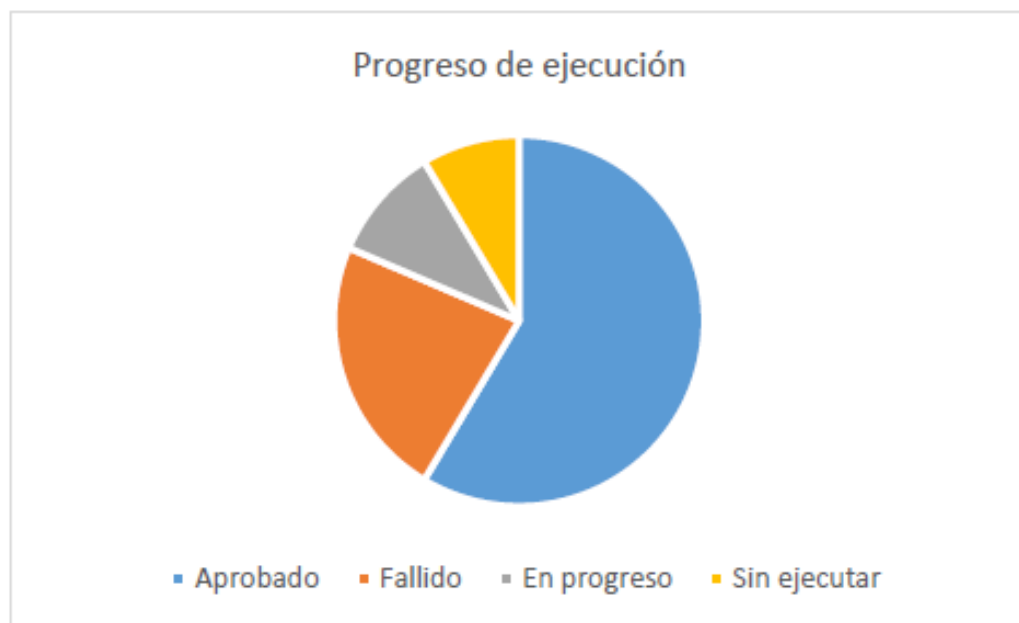
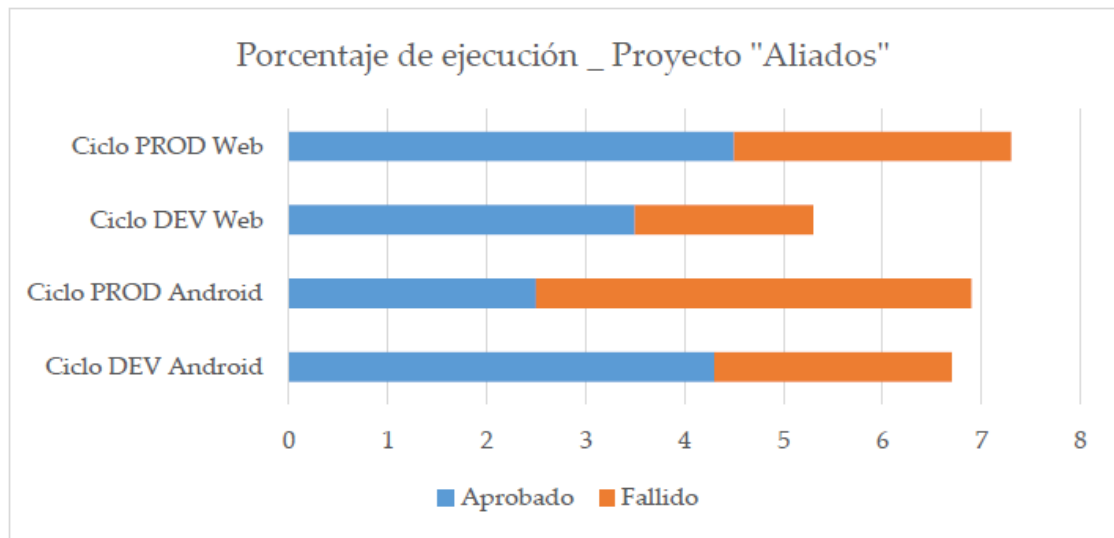


Figura 7

Progreso por barra consolidado de ciclo de pruebas en el sprint



b. Metodología de Aseguramiento de la calidad de Software (ACS)

El aseguramiento de la calidad (ACS), es una actividad que se aplica a todo el proceso del software:

El ACS incluye procedimientos para la aplicación eficaz de métodos y herramientas, supervisa las actividades de control de calidad, tales como revisiones técnicas y las pruebas del software, procedimientos para la administración de cambio y elaboración de reportes. Para llevar a cabo el aseguramiento de la calidad del software de manera adecuada, deben recabarse, evaluarse y divulgarse datos sobre el proceso de ingeniería de software. Los métodos estadísticos aplicados al ACS ayudan a mejorar la calidad del producto y del proceso de software mismo. Uno de los principios fundamentales para lograr la calidad de un producto de software es el Aseguramiento de Calidad, es desde ahí que nace esta propuesta metodológica de Aseguramiento a Calidad, a través de un enfoque práctico. (Carrizo y Alfaro, 2018, p. 118).

- **Propuesta de Aseguramiento de la Calidad**

El método de ACS que se propone en este trabajo, permitirá a cualquier proyecto de desarrollo de software, proporcionar calidad al producto final. Este enfoque práctico está diseñado, para adaptarse o acoplarse a cualquier Metodología de Desarrollo y de esta manera administrar la calidad en desarrollo de un proyecto. Esta propuesta debería incluir un Aseguramiento de la Calidad del software: 1) un proceso de ACS, 2) tareas específicas de aseguramiento de la calidad y control de la calidad, 3) prácticas eficientes de ingeniería de software (métodos y herramientas), 4) control de todos los productos del trabajo de software y de los cambios que sufren, 5) un procedimiento para garantizar el cumplimiento de los estándares del desarrollo de software, y 6) mecanismos de medición y reporte.

Por lo tanto, tomado en consideración lo que se plantea, se desarrolla y propone un método de ACS:

Esta propuesta está compuesta por Esencia y Herramientas el que será medido por Métricas, que buscan no solo una forma de trabajar y administrar la calidad, sino también la posibilidad de mejorar los procesos de desarrollo a través de la recolección y análisis de errores. (Carrizo y Alfaro, 2018, p. 120).

➤ **Método de ACS**

Como se mencionó anteriormente y puede verse en la Figura 8, esta Propuesta Metodológica de Aseguramiento de la Calidad:

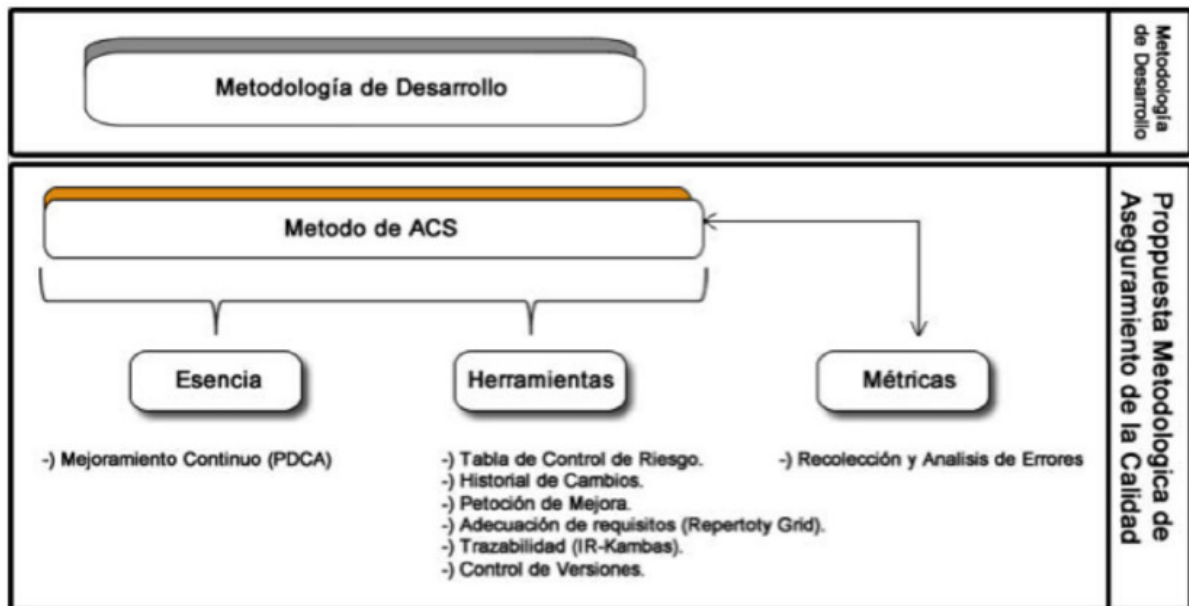
Se puede adaptar o acoplar a cualquier Metodología de Desarrollo que se esté utilizando, sin tener la necesidad de agregar un miembro más o un equipo paralelo de ACS. (Carrizo y Alfaro, 2018, p. 120).

A continuación pasaremos a describir cada una de las tres actividades o tareas que componen la metodología. Es importante destacar que estos tres componentes se relacionan entre sí, entendiendo que una de las herramientas, puede ser una forma de medir o que a su vez, una herramienta tenga el concepto o la filosofía PDCA.

✓ **Esencia**

Esta actividad en el método de ACS, hace referencia a la manera de cómo se debe comprender:

Por el grupo de trabajo, la forma de trabajar, en base a un objetivo que es proporcionar calidad al producto de software. Si todo el equipo de trabajo, desde el líder de equipo, pasando por todos los roles del grupo, trabajaran entendiendo que la calidad es fundamental en el desarrollo de un producto y su objetivo es satisfacer las necesidades del cliente, la administración de la calidad no sería un problema, ya que todos estarían trabajando en función de ese objetivo. (Carrizo y Alfaro, 2018, p. 121).

Figura 8*Propuesta metodológica de ACS*

– Mejoramiento Continuo

La filosofía de mejora continua es una herramienta ideal, para que el equipo de trabajo entienda y busque la calidad en el desarrollo del producto:

El proceso de mejora continua es una constante refinación de las normas para responder de una forma dinámica a las exigencias del cliente y las oportunidades de mejora de los procesos. Para que esto suceda, la administración debe establecer primero el estándar o la base en los procesos o actividades, para que, posteriormente, el ciclo PDCA desempeñe su función reguladora, mejorando los estándares establecidos. De esta manera, el ciclo PDCA permite el aprendizaje organizacional y el logro de mejores estándares. (Carrizo y Alfaro, 2018, p. 121).

- Plan (Planificar o Planear), énfasis a la planificación del proyecto.
- Do (Hacer), corresponde a realizar, fabricar o trabajar el producto planificado.
- Check (Revisar o Comprobar), para confirmar si el cliente está satisfecho o el proceso está según especificaciones.
- Act (Actuar), si se presenta algún reclamo, se actúa sobre el problema sin tener que esperar que finalice el proceso, luego se vuelve a la fase de planificación, volviendo al ciclo PDCA, para obtener un mejoramiento.

✓ **Herramientas**

Las herramientas, en la propuesta metodológica de ACS, ayudan al proceso en hacer el seguimiento pertinente:

Son un conjunto de técnicas y artefactos que ayudarán a mantener el control y a la vez administrar la calidad en el desarrollo desde el inicio hasta el final, de un producto de software. A continuación se describirán y mostrarán las herramientas que propone este método de ACS. (Carrizo y Alfaro, 2018, p. 121).

– **Tabla de Control de Riesgo**

En la figura 9, tiene como objetivo identificar, controlar y eliminar las fuentes de riesgo antes de que empiecen a afectar el cumplimiento de los objetivos del proyecto:

De esta manera se pueden evaluar y estimar el impacto de los riesgos posibles y a su vez establecer un plan de contingencia para mitigarlos

en el caso de que el problema se presente, teniendo como objetivo la pro actividad. Por lo tanto: 1) se inicia antes del trabajo técnico, 2) se categoriza o clasifica según su impacto, 3) identifica los riesgos potenciales, valorando su probabilidad de impacto y 4) establecer un plan. (Carrizo y Alfaro, 2018, p. 121).

– Historial de Cambio

La figura 10, muestra el Documento Historial de Cambio:

Que permite al equipo de trabajo llevar un registro de los cambios o mejoras solicitadas en el Documento de Petición de Mejora, junto con ello también admite llevar un control de los documentos o entregables, tanto sus versiones como alguna modificación en ellos. (Carrizo y Alfaro, 2018, p. 121).

Figura 9

Control de riesgo

Riesgo	Categoría	Probabilidad	Impacto	Plan de Acción

Tabla 10

Clasificación en categorías y factores de riesgos - nivel de impacto

Componentes de Riesgo	Rendimiento
	Coste
	Mantenibilidad
	Planificación
Factores de Riesgo	Despreciable
	Marginal
	Crítico
	Catastrófica

Nota. Se puede observar la clasificación de riesgos por componentes que describen el rendimiento, coste, mantenibilidad y planificación, y los factores de riesgos que describen despreciable, marginal, crítico y catastrófica. Adaptada de *Clasificación en categorías y factores de riesgos - nivel de impacto*, por Carrizo y Alfaro, 2018, Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico.

Figura 10

Historial de cambio

ENTREGABLE/ PETICIÓN DE CAMBIO/ OTROS	VERSION /N° PETICION	FECHA	PRIORIDAD	DESCRIPCIÓN

– Documento de Petición de Mejora

Este documento llamado Petición de Mejora, que se puede observar en la figura 11, tiene como objetivo priorizar y mostrar las modificaciones que proponen o recomiendan los Stakeholders o Clientes:

De esta manera, se podrán analizar dichos cambios o propuestas e implementarlas en el proyecto, controlando de modo ordenado los cambios en el desarrollo del sistema. Cabe destacar que este documento sirve para gestionar cambios o mejoras ya sea en los requisitos del sistema, entregables, artefactos o documentación del proyecto, luego de ser testeada o implementado. (Carrizo y Alfaro, 2018, p. 122).

Figura 11

Documento de petición de mejoras

Petición de Mejora	
Solicitante:	Número de Petición:
Fecha:	Nombre del Proyecto:
Prioridad de Petición:	
Prioridad: A) Alta/Eencial B) Media/Deseada C) Baja/Opcional	
ANTES	DESPUÉS
<hr style="width: 20%; margin: 0 auto;"/> Firma del Solicitante	

– Adecuación de requisitos (Repertory Grid)

Uno de los problemas fundamentales en el desarrollo de software es la mala calidad de la educación de los requisitos del sistema, provocando costos importantes en los proyectos:

Por lo tanto, para evitar los problemas anteriores y desarrollar productos que satisfagan al cliente, se propone utilizar el emparrillado (Repertory Grid) planteada en la Tabla 11 muestra un ejemplo de Repertory Grid, la cual permite determinar el valor de adecuación de dichos requisitos a las necesidades del usuario, a mayor adecuación, mayor calidad. (Carrizo y Alfaro, 2018, p. 123).

Por un lado el polo nominal o de semejanza de los constructos dicotómicos a utilizar, se estructura de la siguiente forma:

- **Correcto:** si, y solo si, cada requisito declarado se encuentra en el producto.
- **Inequívoco:** si, o solo si, cada requisito declarado tiene una sola interpretación. Debe ser inequívoco para aquellos que lo crean y para aquellos que lo usen.
- **Completo:** si, y solo si, están relacionados con la funcionalidad, el desarrollo: las restricciones de diseño, los atributos y las interfaces externas.
- **Consistente:** si, y solo si, ningún subconjunto de requisitos individuales generó conflictos en él.
- **Importante:** si, y solo si, él tiene un identificador para indicar la importancia que lo relaciona al producto.

- **Estable:** si, y solo si, el número de cambios realizados, o que se espera realizar, es mínimo.
- **Comprobable:** si, y solo si, existe algún proceso rentable finito con que una persona o la máquina pueda verificar que el producto reúne el requisito.
- **Modificable:** si, y solo si, su estructura y estilo son tales que pueden hacer cualquier cambio fácilmente, completamente y de forma consistente conservando la estructura y estilo.
- **Identificable:** si, y solo si, su origen está claro y facilita su referencia en el desarrollo futuro o documentación del mismo, debiendo ser tanto identificable dirigido hacia tras, como identificable dirigido hacia a delante.

Por el otro lado, el polo de contraste está compuesto por, Incorrecto, Ambiguo, Incompleto, Débil, Intrascendente, Inestable, No Comprobable, No Cambiable, No Reconocible.

– Trazabilidad (Codificación IR-Kanban)

El no tener un control de trazabilidad, asociado a identificar los diferentes documentos o artefactos en un desarrollo de software. “Podrían causar problemas y por lo tanto perjudicar en la calidad del producto” (Carrizo y Alfaro, 2018, p. 123).

Tabla 11*Ejemplo de Aplicación de Repertory Grid*

		Requisitos funcionales			
		R1	R2	R3	
CONSTRUCTOR	Correcto	3	9	9	Incorrecto
	Inequívoco	9	9	6	Ambiguo
	Completo	9	2	9	Incompleto
	Consistente	7	9	9	Débil
	Importante	9	4	9	Intrascendente
	Estable	9	8	9	Inestable
	Verificable	1	9	9	No comprobable
	Modificable	9	9	9	No cambiable
	Identificable	9	1	9	No reconocible
	Trazable	5	9	2	No trazable

Nota. Se puede observar los requisitos funcionales por estado que está clasificado por puntajes dependiendo de la complejidad y magnitud de la observación. Adaptada de *Ejemplo de Aplicación de Repertory Grid*, por Carrizo y Alfaro, 2018, Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico.

Para esto se propone a IR-Kanban, que es un sistema de codificación del cual se puede obtener un correcto control de la documentación asociada al proyecto como: SRS, Diagramas de flujos, casos de uso, entre otros. Ayudando a identificar diferentes aspectos a través de los sub códigos que lo componen.

En la Figura 12, se observa la estructura de IR-Kanban donde el código resultante, estará conformado por segmentos de información, agrupados de forma similar a las MAC.

Figura 12

Estructura del Sistema de Codificación

$$P_2 P_1 P_0 - H_2 H_1 H_0 - I_2 I_1 I_0 - C_2 C_1 C_0 - \\ V_1 V_0 - E_2 E_1 E_0 - B_2 B_1 B_0 - F_0 - S_2 S_1 S_0 - O_0$$

La Tabla 12, describe la estructura del sistema IR-Kanbas.

Tabla 12

Descripción del Sistema de Codificación IR-Kanban

Sub código	Descripción
P₂P₁P₀	Código identificador de proyecto, por ejemplo “P15”, haciendo referencia al proyecto N° 15.
H₂H₁H₀	Código de 3 dígitos que identifica la naturaleza del artículo que está siendo referenciado, el cual puede ser: Documento de especificación de requisitos (SRS), Caso de Uso (CUF).
I₂I₁I₀	Dígitos para identificar la pertenencia del artículo con algún otro de mayor nivel, por ejemplo si el artículo es identificado como un caso de uso (CUF), esta opción identifica el paquete al cual pertenece.
C₂C₁C₀	Código de 3 dígitos que sirve para identificar el artículo, el cual se incrementa correlativamente.
V₁V₀	Hace referencia a la versión del artículo, por ejemplo “12”, haciendo referencia a la 12° interacción o versión.

E2E1E0	Código de 33 dígitos que identifica el equipo de trabajo “E03”, referenciando al equipo N° 3.
B2B1B0	Identificador del jefe de proyecto o la persona encargada de la validación interna del artículo.
F0	Dígito de Flag que representa el estado de validación interna del artículo, por parte del jefe de proyecto o la persona encargada para ello. Este indicador puede ser: En espera (0), Validado (1), Rechazado (2), Incompleto (3), entre ellos.
S2S1S0	Identificador del stakeholders o la persona encargada de la validación externa del artículo (aprobación).
O0	Dígitos que representa el estado de validación externa del artículo (aprobación), por parte del stakeholders o la persona encargada para ello. Este indicador puede ser: En espera (0), Aprobado (1), Rechazado (2), Incompleto (3), entre otros.

Nota. Se puede observar la descripción de cada sub código del IR-Kanban con esto se puede obtener un correcto control de la documentación asociada al proyecto. Adaptada de *Descripción del Sistema de Codificación IR-Kanban*, por Carrizo y Alfaro, 2018, Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico.

– Control de Versiones

El código fuente es el elemento primordial en un desarrollo de un producto de software y poder administrarlo de forma eficiente, siendo fundamental a la hora de proporcionar calidad al proyecto de software. Una de las herramientas propuestas del método ACS, es el sistema de control de versiones (SCV). Los SCV son una herramienta esencial para manejar

proyectos de software. Proporcionan una serie de funcionalidades claves para el desarrollo de proyectos como es el control de cambios en el código, la reversibilidad de dichos cambios, y la posibilidad de colaborar en el desarrollo del código.

Los principales beneficios de esta herramienta son que ayuden al poder hacer el seguimiento pertinente:

1) cualquier revisión almacenada de un archivo puede ser recuperada para visualizarse o modificarse, 2) pueden desplegarse las diferencias entre distintas versiones, 3) las correcciones pueden ser creadas automáticamente, 4) múltiples desarrolladores pueden trabajar simultáneamente en el mismo proyecto o archivo sin pérdida de datos, 5), los proyectos pueden ser divididos para permitir el desarrollo simultáneo en varias versiones (estas divisiones pueden ser fusionadas para alcanzar el objetivo principal del desarrollo) y 6) el desarrollo distribuido, es soportado a través de las redes de datos con diferentes mecanismos de autenticación. Uno de los más utilizados es Git, inicialmente desarrollado para Linux. Git permite a varios programadores trabajar paralelamente con sus propias copias de trabajo obtenidas de un repositorio, como lo efectúan todos los SCV distribuidos. (Carrizo y Alfaro, 2018, p. 124).

✓ Métricas

La única manera de mejorar es medir como se está haciendo algo. Para ello, este modelo propone una herramienta de ACS estadístico que reúne y clasifica errores

para luego analizarlos y poder mejorar sustancial a la calidad, en el desarrollo de proyectos futuros.

Según Pressman los errores se pueden clasificar en las siguientes causas:

- Especificaciones erróneas o incompletas (EEI).
- Mala interpretación de la comunicación con el cliente (MCC).
- Desviación intencional de las especificaciones (DIE).
- Violación de los estándares de programación (VEP).
- Errores en la representación de los datos (ERD).
- Interfaz componente inconsistente (ICI).
- Error en el diseño lógico (EDL).
- Prueba incompleta o errónea (PIE).
- Documentación inexacta o incompleta (DII).
- Errores en la traducción del lenguaje de programación del diseño (LPD).
- Interfaz humano/computadora ambigua o inconsistente (IHC).
- Varios (V).

Para aplicar el ACS estadísticos se elabora una tabla, como se observa en la Tabla 13:

Que contiene todas las posibles causas de errores, clasificadas en: Serio, Moderado y Menos. De esta manera se puede observar e identificar los errores más críticos o vitales, se puede dar comienzo a sus acciones correctivas. (Carrizo y Alfaro, 2018, p. 124).

Tabla 13*Colección de Datos para ACS Estadístico*

Error	Total		Serio		Moderado		Menor	
	N°	%	N°	%	N°	%	N°	%
EII	186	45	48	50	73	41	65	46
MCC	123	30	13	13	86	49	24	17
PIE	55	13	18	18	5	2	32	23
DII	21	5	14	14	5	2	32	23
LDP	25	6	3	3	6	3	16	11
Total	410	100	96	100	100	100	139	100

Nota. Se puede observar a nivel estadístico el total de errores por tipo sirve para identificar los errores más críticos o vitales, con esto se puede tomar acciones correctivas. Adaptada de *Colección de Datos para ACS Estadístico*, por Carrizo y Alfaro, 2018, Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico.

Es importante destacar que acciones correctivas se centran sobre las pocas causas vitales. “En tanto estas se corrigen, nuevas candidatas se van a la cumbre de la pila” (Carrizo y Alfaro, 2018, p. 125).

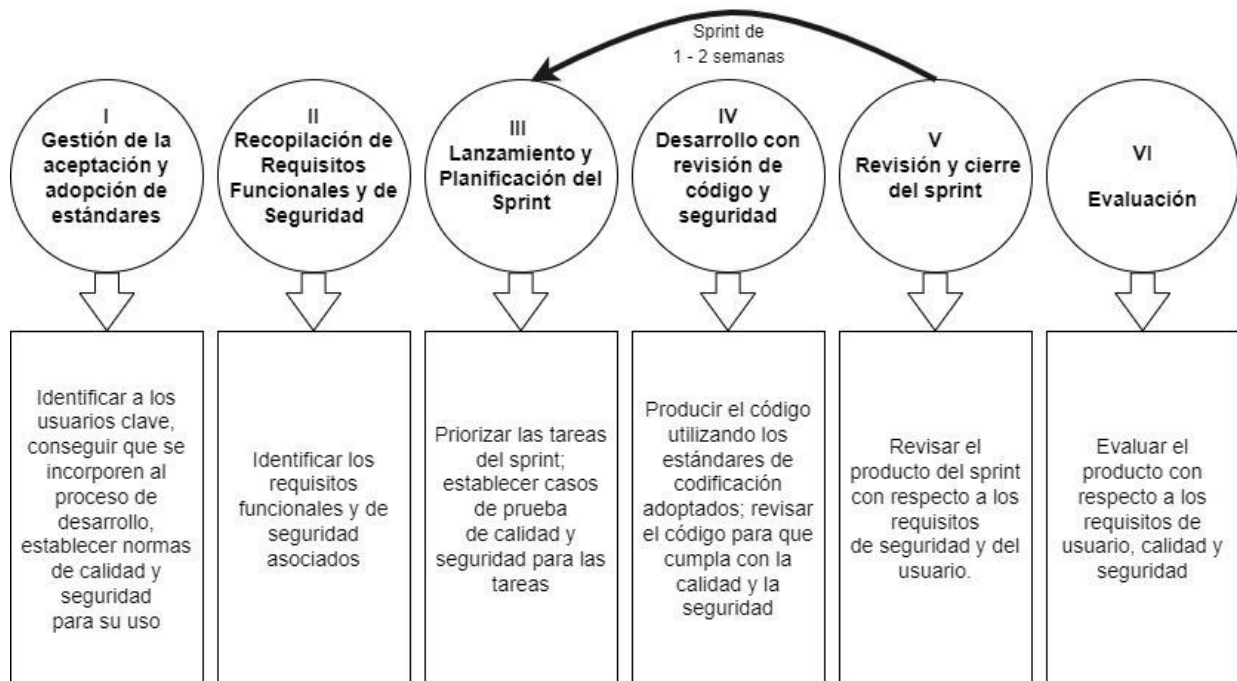
c. Metodología de Desarrollo Seguro en solitario (SSDM)

Es una metodología ágil diseñada a través de la integración de prácticas de calidad y prácticas de seguridad extraídas de la literatura.

Esto le da a la metodología sus características intrínsecas de calidad. La metodología está organizada en seis etapas: Etapa I – Gestión de la aceptación y adopción de estándares, Etapa II – Recopilación de requisitos funcionales y de seguridad, Etapa III – Lanzamiento y planificación del sprint, Etapa IV - Desarrollo y revisión, Etapa V - Revisión del sprint y cierre y Etapa VI - Evaluación. Las etapas se resumen como se muestra en la figura 13. Se resumen las prácticas realizadas en cada una de estas etapas en las siguientes subsecciones. (Mnkandla y Moyo, 2020, p. 33740).

Figura 13

Etapas de Secure-SSDM y principales actividades en cada etapa.



Nota. Las etapas de la Metodología Secure-SSDM está centrada en las buenas prácticas de calidad y seguridad a un nivel de trabajo ágil para el desarrollo de Software. Adaptada de *Etapas de Secure-SSDM y principales actividades en cada etapa*, por Mnkandla y Moyo, 2020, *A Novel Lightweight Solo Software Development Methodology With Optimum Security Practices*.

- **Etapa I: Gestión de la aceptación y adopción de estándares**

Un desarrollador que adopta Secure-SSDM comienza identificando y educando a los usuarios sobre cómo se llevará a cabo el proyecto utilizando la metodología. La educación del usuario en este caso se aplica en los casos en que un proyecto ha identificado a usuarios finales, particularmente en un entorno organizacional. En algunos casos, un producto de software se desarrolla para un conjunto genérico de usuarios. En este caso, el desarrollador utiliza su creatividad para formular requisitos para el software. Si es posible, se puede identificar un conjunto representativo de clientes para que desempeñen el papel de usuarios. La identificación de los usuarios al comienzo de un proyecto de software es un concepto bien establecido de la participación de los usuarios en el desarrollo de software. Además de educar a los usuarios sobre cómo procederá el proceso de desarrollo, el desarrollador concientiza a los usuarios sobre cuestiones de seguridad.

Durante este período, el desarrollador también adopta estándares apropiados (tanto de desarrollo como de seguridad) determinados por el tipo de software en desarrollo para usar en el proyecto:

Se alienta a los desarrolladores a consultar los estándares de calidad y seguridad relevantes para su producto de software para comparar la calidad del producto

en desarrollo. También es importante en este punto llevar a cabo un análisis de seguridad en los roles de usuario identificados. En este punto, el análisis identifica amenazas de seguridad basadas en requisitos de usuario de alto nivel. Aquí se recomienda el uso de un tablero automatizado para capturar y realizar un seguimiento de los requisitos del usuario y sus amenazas asociadas. Trello como herramienta permite al desarrollador organizar y administrar su trabajo para que pueda visualizar fácilmente las tareas en curso, pendientes, realizadas, para revisar, etc. Como herramienta basada en la web, proporciona portabilidad que permite al desarrollador acceder a su tablero desde cualquier lugar. (Mnkandla y Moyo, 2020, p. 33741).

- **Etapa II: Recopilación de Requisitos Funcionales y de Seguridad**

Durante la obtención de requisitos, el desarrollador trabaja con los usuarios para recopilar requisitos funcionales y de seguridad. Esto se puede hacer mediante la recopilación de historias de usuarios que describen las interacciones esperadas de los usuarios con el sistema. Las historias de usuario deben definirse para que sean lo suficientemente pequeñas para que el desarrollador pueda trabajar en ellas de forma independiente. Estos son lo que los usuarios perciben como actividades de valor añadido que se obtendrán del sistema. Estas interacciones esperadas se utilizan para crear diagramas de casos de uso. Recomendamos el uso de diagramas UML en el modelado de casos de uso. Al mismo tiempo, el desarrollador también recopila interacciones inesperadas que pueden llevar a cabo personas no autorizadas en los casos de uso. Estos se identifican mejor con la ayuda del usuario, ya que comprenden mejor el entorno operativo.

El desarrollador debe hacer todo lo posible para que el usuario imagine todas las circunstancias que podrían perturbar el buen flujo de sus procesos:

Una forma de ayudar al usuario a identificar casos de uso indebido es pedirle que imagine qué haría un intruso si accediera al sistema. Se crea un modelo compuesto de casos de uso y casos de mal uso para mostrar las interacciones de los usuarios con el sistema. Los casos de uso y sus casos de uso indebido asociados se utilizan para crear una acumulación de productos priorizada. Una acumulación de productos priorizada enumera los entregables esperados del usuario en cada iteración de desarrollo. La captura de casos de uso indebido en una etapa tan temprana ayuda al desarrollador a priorizar la seguridad, en lugar de tener que pensar en la seguridad en las etapas de implementación y prueba. Para sistemas complejos, el desarrollador puede construir prototipos para facilitar la comunicación entre los usuarios. Esto también ayuda al desarrollador a comprender el sistema. Una vez que el backlog priorizado está en su lugar y acordado por el desarrollador y el cliente, puede comenzar la planificación del sprint. (Mnkandla y Moyo, 2020, p. 33742).

- **Etapa III: Lanzamiento y Planificación del Sprint**

La planificación de versiones y sprints implica identificar la tarea en la parte superior de la cartera de productos priorizada y las tareas que la componen:

Dado que el desarrollador está trabajando por su cuenta, extrayendo elementos de una cartera de productos ya priorizada, puede usar la lógica del sistema para ordenar tareas en un sprint. Las tareas ordenadas en un sprint deben tener estimados de tiempo adjuntos. El desarrollador debe establecer iteraciones cortas de 1 a 2 semanas para mejorar la visibilidad del proceso. Las iteraciones

cortas también motivan al desarrollador al mismo tiempo que promueven la productividad. Se deben hacer diseños apropiados para el entregable del sprint, junto con diseños de seguridad y pruebas de aceptación. Los diagramas de secuencia se pueden utilizar para modelar las interacciones del usuario con el sistema. Otra alternativa es utilizar diagramas de clases. Se recomienda que el desarrollador mantenga los modelos al mínimo posible para mejorar la productividad. Se deben indicar los puntos clave de intercepción de intrusos para permitir que el desarrollador diseñe medios para contrarrestarlos. (Mnkandla y Moyo, 2020, p. 33743).

- **Etapa IV: Desarrollo y revisión**

El desarrollo con revisión construye el producto de software, el desarrollador produce el código para las tareas teniendo cuidado de adherirse a los estándares de codificación y seguridad adoptados al inicio del proyecto:

Recomendamos el uso de herramientas de control de versiones/cambios para realizar un seguimiento de cualquier cambio en el código. Para minimizar los defectos del código, recomendamos que el desarrollador realice revisiones de seguridad a nivel de código y fuente con la ayuda de un socio ficticio. La identificación de errores en el código ayuda a garantizar que solo el código seguro y correcto se integre en el entorno en vivo. Un compañero ficticio es cualquier objeto que el desarrollador puede obtener para explicarle el código. Esto también puede significar practicar el diálogo con uno mismo. Aquí, el desarrollador recorre todas las secciones del código y explica su funcionalidad al maniquí. Esto ayuda a identificar errores en el código. Todos los errores

identificados deben corregirse antes de que el código se integre en la línea de base. (Mnkandla y Moyo, 2020, p. 33743).

- **Etapa V: Revisión del Sprint y Cierre**

Revisar y cerrar el sprint marca el final de un sprint establecido inicialmente para entregar alguna funcionalidad (o algún componente) en el sitio del usuario:

Aquí el desarrollador debe revisar el tiempo que lleva completar el sprint, contra el tiempo inicial establecido para el propósito. Cualquier diferencia debe anotarse y usarse para ajustar las estimaciones para el próximo sprint o proyectos futuros. El desarrollador también revisa la calidad del código frente a los estándares establecidos. Las tareas terminadas se mueven a tareas completadas y las tareas incompletas se trasladan a la siguiente iteración. El desarrollador también realiza integración de código, pruebas de integración y pruebas de seguridad. Se recomienda encarecidamente el uso de un sistema de control de versiones. Estos incluyen herramientas utilizadas para: acceder rápidamente a todos los archivos de código modificados recientemente; corregir la confirmación más reciente; eliminar la confirmación más reciente; y dividir una confirmación en caso de que el desarrollador detecte o sospeche algún conflicto dentro de los componentes del código. Se deben realizar pruebas de seguridad en todo el código antes de la integración. En este punto, el desarrollador también revisa el progreso del proyecto, luego de lo cual planifica el próximo sprint o cierra el proyecto, si no quedan elementos en la cartera de pedidos. (Mnkandla y Moyo, 2020, p. 33743).

- **Etapa VI: Evaluación**

La última etapa evaluación, concluye el proyecto, esto implica la evaluación de los entregables del producto frente a las expectativas del usuario, así como los estándares adoptados al inicio del proyecto:

Este proceso puede realizarse mediante la realización de una prueba de aceptación del sistema. El sistema se prueba tanto para los requisitos del usuario como para el cumplimiento de los requisitos de seguridad. En este punto, el desarrollador debe actualizar su base de conocimientos de desarrollo, así como el repositorio de seguridad. El repositorio de seguridad captura prácticas para incorporar seguridad en el software. Esta gestión del conocimiento promueve el aprendizaje en este entorno. También se recomienda al desarrollador que identifique procesos o tareas para la automatización. Estas son actividades recurrentes dentro del ciclo de vida del proyecto. La automatización de los procesos de desarrollo mejora la productividad del desarrollador. Aunque la mayoría de las etapas de Secure-SSDM se ilustran como secuenciales, se debe tener en cuenta que se trata de un proceso iterativo y que algunas tareas pueden ejecutarse en paralelo según el tamaño y la complejidad de las tareas en cuestión. (Mnkandla y Moyo, 2020, p. 33743).

Tabla 14*Prácticas de calidad y seguridad*

SSDM Prácticas de calidad	Prácticas de calidad
I. Gestión de la aceptación y adopción de estándares	
- Identificación y educación de los usuarios	- Formación en materia de seguridad
- Adopción de normas (calidad y desarrollo)	- Análisis de seguridad de las funciones de los usuarios
II. Obtención de requisitos	
- Creación de historias de usuario	- Creación de casos de uso indebido
- Creación de casos de uso	- Detalle del caso de uso indebido
- Creación y priorización de la cartera de productos	
- Desarrollo de prototipos	
III. Lanzamiento y Planificación del Sprint	
- Priorización de las tareas del sprint y planificación del sprint	- Aplicación de los principios de diseño de seguridad
- Uso de sprints cortos (1-2 semanas)	- Diseño de pruebas de seguridad
- Diseño de la prueba de aceptación	
IV. Desarrollo y revisión	
- Codificación y revisión del código (con maniquí)	- Codificación de la seguridad: cumplimiento de las normas
- Control de versiones y cambios	- Revisión de la seguridad del código fuente
- Refactorización del código	
V. Revisión del Sprint y Cierre	
- Validación de productos/evaluación de resultados	- Pruebas de seguridad
- Integración y pruebas del código	
- Revisión del progreso del sprint y del proyecto	

VI. Evaluación

- | | |
|----------------------------|---|
| - Pruebas del Sistema | - Gestión de la divulgación de la seguridad |
| - Automatización de tareas | - Revisión del repositorio de seguridad |
-

Nota. Las prácticas de calidad y seguridad están clasificadas ya que cada una de ellas tiene sus estándares y que complementan las pruebas de calidad y haya un mejor control según las etapas. Adaptada de *Prácticas de calidad y seguridad*, por Mnkandla y Moyo, 2020, A Novel Lightweight Solo Software Development Methodology With Optimum Security Practices.

2.3. Marco filosófico

En la filosofía Ishikawa es considerado como el padre de la teoría de causa-raíz aplicada a la calidad, básicamente sugiere determinar las causas probables de las ineficiencias presentes en los procesos productivos, y erradicar su ocurrencia.

El ser humano en diversas formas y circunstancias ha demostrado la tendencia hacia la calidad, hacia lo bueno, como algo normal:

En sentido la preocupación por la calidad estaba presente en el hombre mismo, como un elemento fundamental del hombre. Con todo, la concepción de la calidad ha ido evolucionando en cada etapa desarrollada. Esta etapa se da desde la década de 1950, en que se reconoce que la calidad no solamente de los procesos de manufactura, sino que requiere de otros servicios de soportes, tales como el área de producción, y diseño de producto, la ingeniería de proceso, abastecimiento, laboratorio, entre otros. Con ello se requiere el involucramiento de todas las áreas de la organización. (Barrios, 2018, p. 2).

2.4. Marco individual

El conocimiento que se tiene del proceso de Aseguramiento de la Calidad nos ayuda a la implementación de Metodología MAC para así poder obtener los resultados factibles que apoyen el proceso a poder mejorar. En base a la experiencia se puede ver el problema de la realidad de cerca, ya que se reportan incidencias sobre el proceso el día a día y este no es eficiente por esta razón se logra tener indicadores que soporten este proceso y ayude a reducir estos problemas.

III. MÉTODO

3.1. Tipo de investigación

3.1.1. Tipo de investigación

a. Investigación básica: Se aporta conocimiento a través de la creación de una nueva metodología MAC (Metodología de Aseguramiento de la Calidad), a partir de una investigación de diferentes metodologías de buenas prácticas, conforme a la realidad y las necesidades del área de Control de Calidad de la empresa CANVIA S.A.

b. Investigación aplicada: Se realizó la aplicación de la nueva metodología MAC para la implementación de Buenas Practicas, orientado al Proceso de aseguramiento de la Calidad. Se genera plantillas y documentación que deberán ser aplicadas para resolver el problema.

3.1.2. Nivel de investigación

a. Descriptiva: Se describe las características más importantes del Proceso de Aseguramiento de la Calidad de los proyectos de Desarrollo de Software para determinar cómo el proceso puede ser mejorado mediante la nueva Metodología (MAC Metodología de Aseguramiento de la Calidad).

b. Predictiva: También llamada Correlacional – Causal se aplica sobre la variable dependiente para solucionar el proceso, ya que se presenta un nuevo aporte a la Gestión del Aseguramiento de la Calidad de Software para mejorar su proceso.

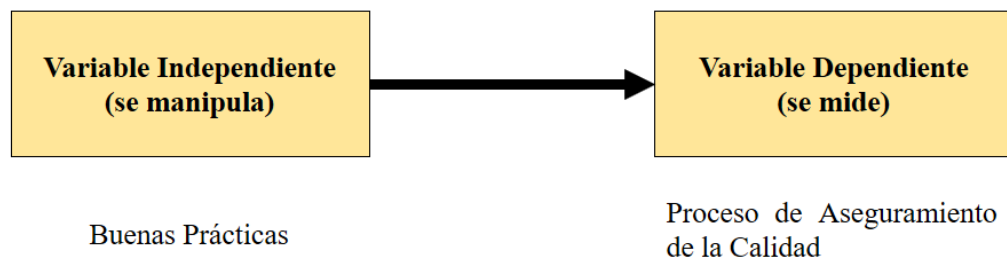
3.1.3. Diseño de la investigación

El tipo de Diseño para esta investigación es **Experimental Puro**.

Un diseño experimental puro es aquel en el que se manipula una o varias variables independientes, en esta investigación la **variable independiente** es las **Buenas Prácticas**, para observar sus efectos sobre una o varias variables dependientes, en este caso la **variable dependiente** es el **Proceso de Aseguramiento de la Calidad**.

Figura 14

Tipo de diseño



Diseño con Pos prueba y Grupo de Control

RG_e	X	O_1
RG_c	$-$	O_2

Donde:

- **R:** Los participantes (procesos) han sido elegidos de forma aleatoria.
- **G_e (Grupo experimental):** Es el grupo al cual se le aplicará el estímulo (Buenas Prácticas).
- **G_c (Grupo de control):** Es el grupo al cual no se le aplicará el estímulo (Buenas Prácticas).
- **O_1 :** Medición de los datos de la variable dependiente **antes** de la aplicación de la variable independiente.

- **O₂**: Medición de los datos de la variable dependiente **después** de la aplicación de la variable independiente.
- **X**: Buenas Prácticas (estímulo o condición experimental).
- **-**: Falta de estímulo o condición experimental.

Explicación:

Se trata de la confrontación de forma intencional de un grupo **Ge**, cuyos elementos han sido escogidos de manera aleatoria (**R**), conformado por los procesos de Aseguramiento de la Calidad, al que le aplico un estímulo de Buenas Prácticas (**X**), luego del cual se le aplica una prueba posterior a dicho tratamiento (**O₁**). A un segundo grupo **Gc**, con elementos también escogidos de manera aleatoria, conformado por pedidos de reserva de citas médicas, no se le administra estímulo alguno, sirviendo sólo como grupo de control; en forma simultánea se le aplica una prueba (**O₂**), se espera que los valores **O₂** sean mejores que los valores **O₁**.

3.2. Población y muestra

3.2.1. *Unidad muestral*

Procesos de Aseguramiento de la Calidad.

Restricciones:

- Empresas que brindan servicios de Desarrollo de Software.
- Micro y pequeñas empresas.
- Empresas latinoamericanas.

3.2.2. *Universo*

Todos los procesos de Aseguramiento de la Calidad en micro y pequeñas Empresas Latinoamericanas que brindan servicios de Desarrollo de Software.

Debido a que no se puede determinar todos los procesos de Aseguramiento de la Calidad del Desarrollo de Software en Latinoamérica, entonces es:

N= Indeterminado

3.2.3. *Muestra*

Los Procesos de Aseguramiento de la Calidad en la empresa CANVIA S.A.

n= 30

3.2.4. *Tipo de muestreo*

Sera de tipo de aleatorio o probabilístico, debido a que los elementos serán escogidos para la muestra aleatoriamente.

3.3. Operacionalización de variables

- **Variable Independiente:** Buenas prácticas.
- **Variable Interveniente:** Metodología MAC.
- **Variable Dependiente:** Aseguramiento de la Calidad de Desarrollo de Software en la empresa CANVIA S.A.

Tabla 15

Operacionalización de variables

Variables	Unidad de Observación	Dimensiones	Indicador (es)	Índices	Unidad de Medida	Fórmula
<u>Variable Independiente:</u> Buenas Prácticas	Registro Manual	-----	Presencia - Ausencia	No, Si	----	-----
<u>Variable Dependiente:</u> Aseguramiento de la Calidad de desarrollo de Software en la Empresa CANVIA S.A.	Cronograma de seguimiento del Proyecto	Tiempo	Tiempo de entrega	[1..40]	Horas	Días de trabajo x 8h
	Cronograma de seguimiento del Proyecto		Tiempo para la revisión de errores.	[1..40]	Horas	Días de trabajo x 8h
	Formato de observaciones	Productividad	Número de errores en pruebas.	[0..20]	Incidencias	$\sum_{i=1}^n ep_i = (ep_1 + ep_2 + ep_3 + \dots + ep_n)$
	Informe de costo de implementación	Costo	Costo total del proceso.	[0..4,000]	Soles	-----
	Informe final del Proyecto	Cliente	Satisfacción del Cliente.	[Muy Malo, Malo, Regular, Bueno, Muy Bueno]	----	-----

Tabla 16*Conceptualización de Variables*

Variable independiente: Buenas prácticas	
Indicador	Conceptualización
Presencia – Ausencia	Se define como la presencia o ausencia de la metodología en la investigación
Variable dependiente: Aseguramiento de la Calidad de desarrollo de Software en la Empresa CANVIA S.A.	
Indicador (es)	Conceptualización
Tiempo de entrega	Son las horas de trabajo que utiliza el Analista de Calidad para la planificación de sus actividades y que estas sean entregadas a tiempo con todo lo requerido.
Tiempo para la revisión de errores	Son las horas de trabajo que utiliza el Analista de Calidad para la revisión del requerimiento asignado y este tome un tiempo determinado.
Numero de errores en pruebas	Son las incidencias que se generan en un entorno de pruebas para un determinado número de sprint (periodo de tiempo), el Analista de Calidad hace la revisión y pueda ser que encuentre un determinado número de errores para dicho requerimiento, este se describe por la sumatoria de todos los errores de pruebas encontrados por sprint, donde es enviado al desarrollo para su solución.
Costo total del proceso	Es el costo de la implementación del proceso en cuanto a servicios de capacitación y cualquier otro gasto relacionado.
Satisfacción del Cliente	Es el cumplimiento de los tiempos planificados con el cliente del requerimiento a revisar, el cliente da su visto bueno al trabajo realizado por la empresa Desarrolladora de Software.

3.4. Instrumentos

a. Técnicas e Instrumentos para investigación de campo

Tabla 17

Investigación de Campo

Técnicas	Instrumentos
1. La Observación directa <ul style="list-style-type: none"> • Sistemática o estructurada • Participante • Individual 	<ul style="list-style-type: none"> • Fichas de observación • Filmaciones • Celular

b. Técnicas e Instrumentos para investigación experimental

Tabla 18

Investigación Experimental

Técnicas	Instrumentos
<ul style="list-style-type: none"> • Seguimiento de comportamiento de los Analistas de Desarrollo de Software. • Seguimiento de las estrategias de la competencia. • Seguimiento de comportamientos de los analistas de Calidad. • Seguimiento de fases intermediarias y resultados del proceso. • Uso de grupos experimentales y de control. 	<ul style="list-style-type: none"> • Hojas estructuradas • Filmaciones • Entrevistas • Internet: Sitios web • Reportes de indicadores

3.5. Procedimientos

Con el objetivo de analizar a los datos obtenidos mediante la aplicación de las técnicas e instrumentos para la investigación se realizó el siguiente procedimiento a fin de contrastar las hipótesis específicas, por ello se realizó por cada hipótesis específica lo siguiente:

Paso 1: Plantear el problema a través de la hipótesis específica nula.

Paso 2: Realizar la prueba de normalidad.

Paso 3: Realizar la prueba estadística con 2 muestras (PosPrueba Gc y PosPrueba Ge) con el software estadístico.

Paso 4: Obtener el p-value a fin de realizar el análisis de resultados.

Paso 5: Aceptas o rechaza la hipótesis específica nula.

3.6. Análisis de datos

Tabla 19

Procesamiento y análisis de datos

Instrumentos de recolección	Instrumentos de análisis
– Cuestionario	– Software estadísticos
– Guía de observación	

Diseño Estadístico

Mediante la recopilación de datos e información de la evaluación del costo involucrado y tiempos integrados para el proceso de Aseguramiento de la Calidad, se aplica los procedimientos de las diferentes metodologías buscadas y analizadas para determinar qué actividades de los proceso en el área de Calidad de la empresa CANVIA S.A. son las que inciden en mayor proporción en el incremento de costos y tiempos, para ello se sigue los pasos

indicados en la nueva metodología MAC (Metodología de Aseguramiento de la Calidad) ejecutando las acciones a corregir para reducir los problemas más relevantes. Para validar los datos estadísticos se hace uso del programa estadístico Minitab 19.

IV. RESULTADOS

4.1. Desarrollo de la Nueva Metodología

Para el desarrollo de la nueva metodología se realiza la revisión de evidencias disponibles de otras investigaciones existentes y relevantes en un tema en común de interés. En la revisión se plantean el análisis de las metodologías existentes para las Buenas Prácticas basada en la Metodología MAC para el Aseguramiento de la Calidad del Desarrollo de Software o similares, con la finalidad de establecer estándares y procesos que puedan disminuir los errores en producción para ser tomadas en una nueva propuesta que sea mejorada a las que ya existen.

4.1.1. *Revisión de las metodologías existentes*

- **Metodología Six Sigma para el área de Aseguramiento de la Calidad de Software**

Es una metodología que hace énfasis a la gestión por procesos que permite identificar, diseñar, formalizar y mejorar los procesos en la organización y generar mayor confianza en el cliente:

Para que esta pueda orientar al éxito a una organización, debe ir de la mano con un buen mecanismo de gestión de información que sirva como apoyo a la toma de decisiones; por medio de esta, también se puede tener un control al cumplimiento de objetivos definidos en la planificación, niveles de satisfacción, indicadores relacionados a productos defectuosos, encontrar puntos clave que generan ineficiencia o procesos obsoletos. Six Sigma aparece como una estrategia enfocada al cliente pues se soporta en datos y estadísticas, en el cual se disminuyen los defectos y minimiza la variabilidad

en los procesos, esto permite efectuar mejoras de desempeño planificadas y aumentar la efectividad; su objetivo principal es obtener un óptimo desempeño libre de defectos, donde “Defecto” se entiende como cualquier no conformidad que ocasione insatisfacción del cliente. Para lograr la satisfacción del cliente es necesario conocer la definición de calidad, el cual se define en la norma ISO 9000 como “el grado en el que un conjunto de características inherentes cumple con los requisitos”. La calidad no debe ofrecerse única y exclusivamente en los productos finales, sino que debe presentarse a lo largo de todo el proceso de producción. Partiendo de lo anterior, se desarrolla la idea del aseguramiento de la calidad, donde se define como principal objetivo cumplir con los requisitos de calidad. (Jimenez y Jimenez, 2020, pp. 17-18).

- **Metodología de Aseguramiento de la calidad de Software (ACS)**

La metodología de Aseguramiento de la Calidad de Software (ACS) considera que la calidad en ingeniería del software es el cumplimiento de los requerimientos contractuales por parte del producto software desarrollado, así como durante el proceso de desarrollo:

La calidad se obtiene mejorando día a día el proceso de producción, mantenimiento y gestión del software. Para optimizar la calidad de los productos y/o servicios es preciso conocer al cliente y sus necesidades, conocer la competencia y poseer un modelo de calidad. Esto último permitirá incrementar la fiabilidad, reducir el mantenimiento, aumentar la satisfacción del cliente, mejorar la dirección del proyecto, detectar errores lo más temprano posible e incrementar el beneficio para el desarrollador. La función de aseguramiento de la calidad del software (ACS) se basa en un planificado y sistemático diseño de acciones y métodos requeridos para garantizar la calidad del mismo. El alcance de la responsabilidad del aseguramiento de la calidad, en el desarrollo de software, abarca muchos constituyentes de una organización, tales como: ingenieros de

software, desarrolladores, líderes de proyecto, clientes y personas encargados se ACS. La calidad es importante en el desarrollo de un producto o servicio y, más aún, en la creación de un producto de software, no solo porque busca cumplir con las expectativas del cliente, sino también por mejorar los procesos internos en la elaboración de un producto, tarea fundamental en el crecimiento y posicionamiento de una empresa. (Carrizo y Alfaro, 2018, p. 115).

- **Prácticas de Calidad de la Metodología de Desarrollo Seguro en Solitario (SSDM)**

La metodología de desarrollo seguro en solitario (SSDM) considera que la difusión del software en todas las áreas de la vida y todas las formas de negocios aumenta la demanda de productos de software seguros y de alta calidad:

Las metodologías de desarrollo de software están diseñadas para mejorar la calidad del software mediante la incorporación de prácticas que promuevan la calidad en el software desarrollado. La seguridad del software es una faceta importante de la calidad del software, particularmente en esta era, donde la mayoría del software se implementa para su uso a través de Internet. Se diseña una metodología que promueve la calidad y la seguridad en los productos de software de los desarrolladores individuales. Integramos prácticas de calidad con prácticas de seguridad ligeras para producir prácticas de desarrollo de software seguras y ágiles. (Mnkandla y Moyo, 2020, p. 33735).

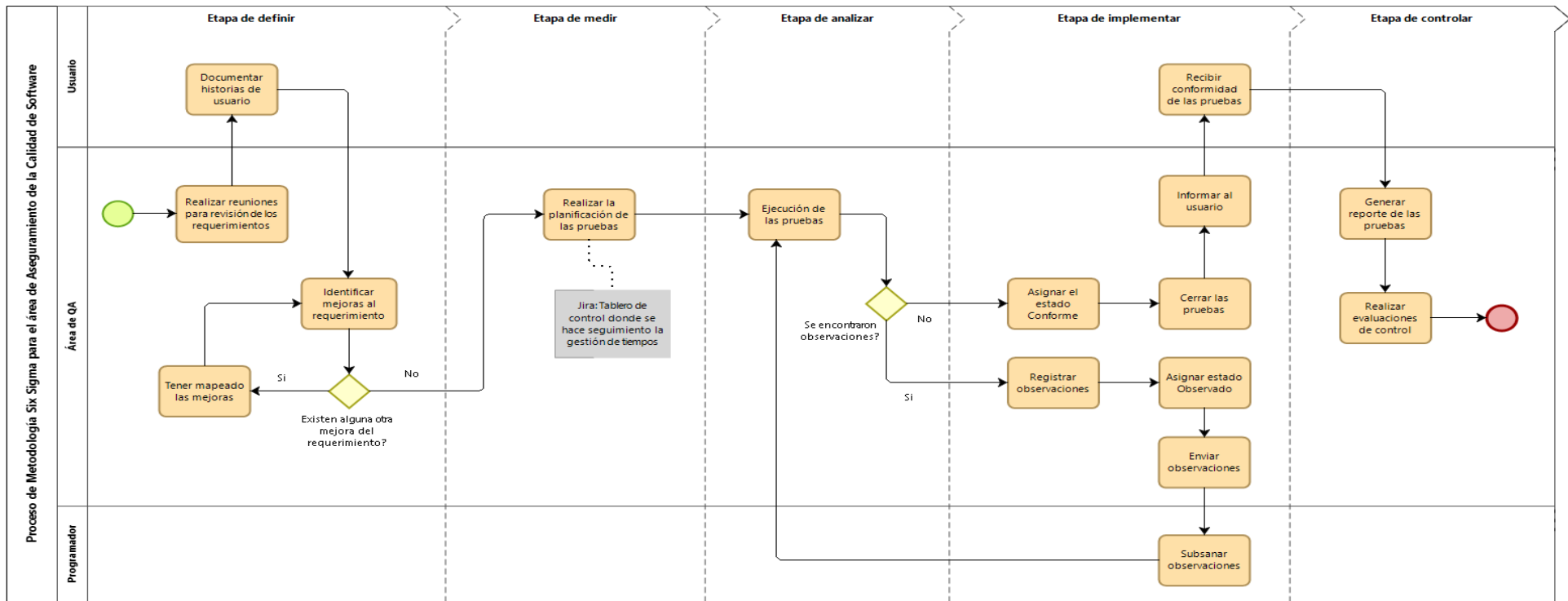
4.1.2. Modelos de Procesos de las Metodologías Existentes

- Metodología Six Sigma para el área de Aseguramiento de la Calidad de Software

En la Figura 15, se detalla el proceso de la Metodología Six Sigma para el aseguramiento de la calidad de Software que alcanza resultados con más eficiencia y eficacia lo cual garantiza procesos más definidos.

Figura 15

Flujograma de la Metodología Six Sigma para el área de Calidad

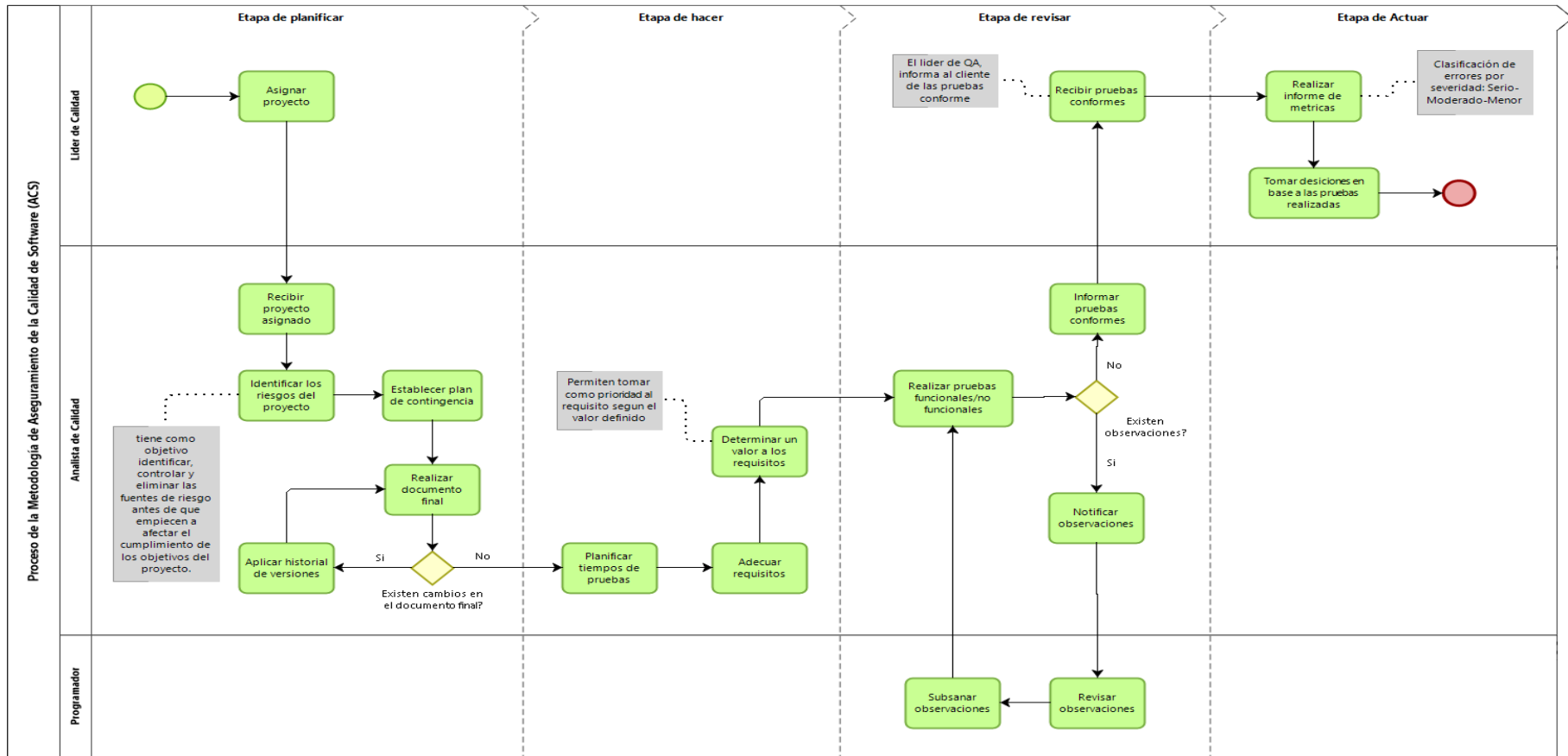


- Metodología de Aseguramiento de la Calidad de Software (ACS)

En la Figura 16, se detalla el proceso para el aseguramiento de la calidad garantizando la mejora continua y abarcando la buena toma de decisiones.

Figura 16

Flujograma de la Metodología ACS

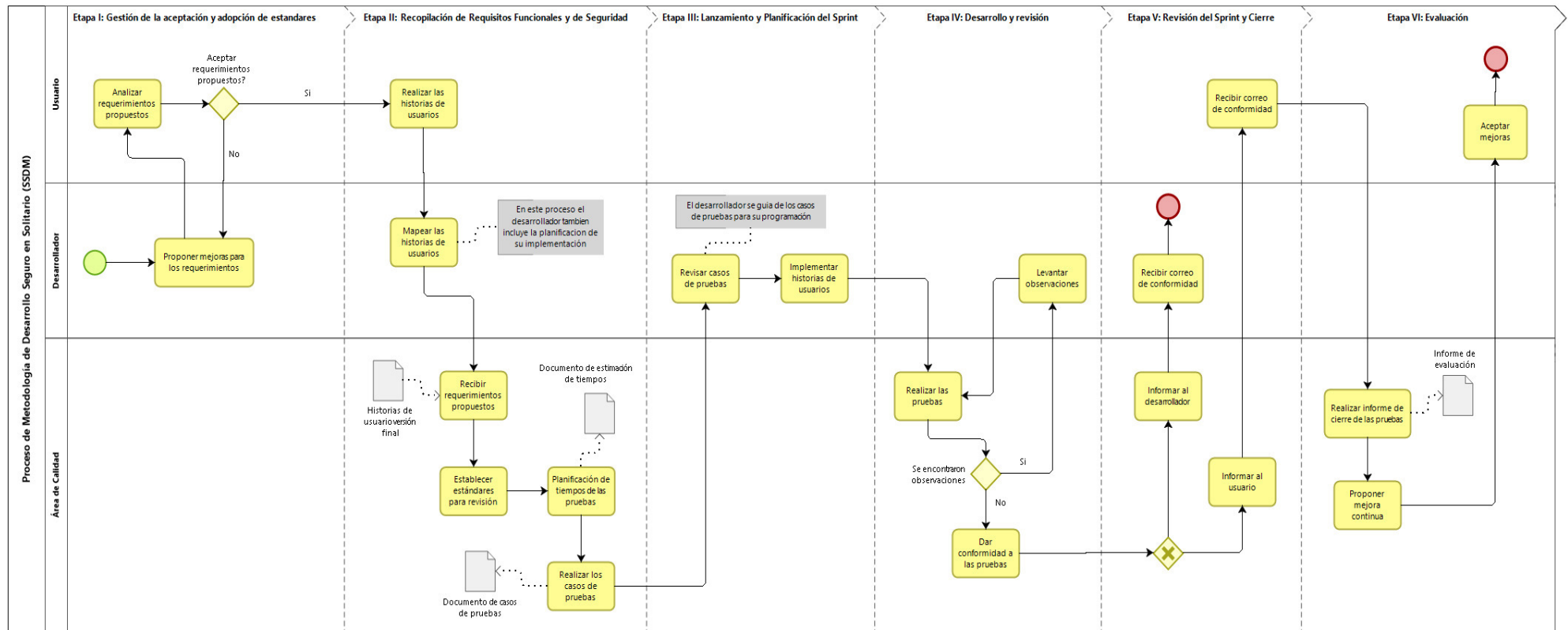


• **Prácticas de Calidad de la Metodología de Desarrollo Seguro en Solitario (SSDM)**

En la Figura 17, se detalla el proceso que permite mejorar la calidad del software, mediante la incorporación de buenas prácticas.

Figura 17

Flujograma de la Metodología de Desarrollo Seguro en solitario (SSDM)



4.1.3. Elaboración de la Nueva Metodología

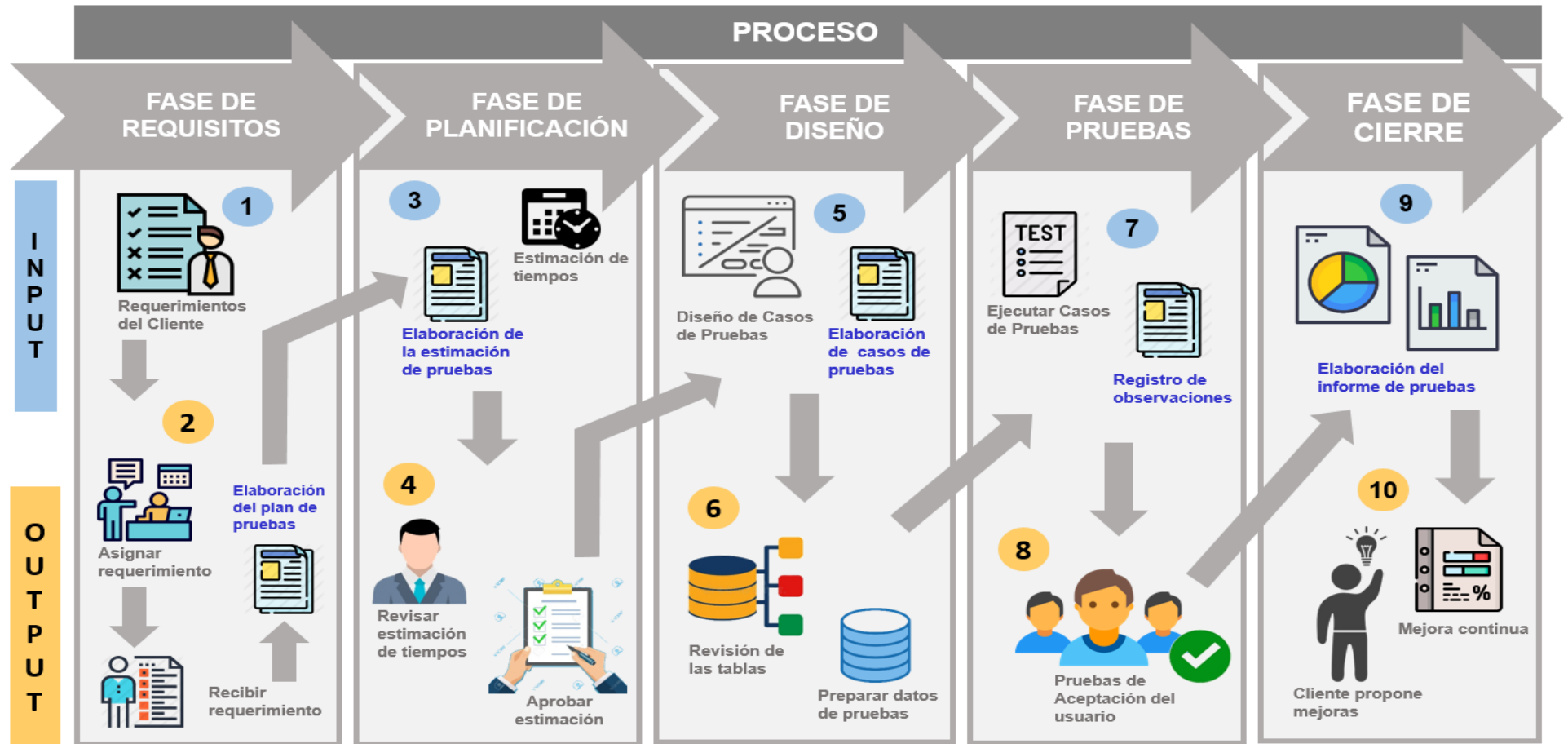
La Nueva Metodología que estructura el ciclo de vida de pruebas de Software para la mejora del proceso de Aseguramiento de la calidad aplicado a Empresas que desarrollan Software, ofrece las buenas prácticas que se implementan. Esta Metodología (Metodología de Aseguramiento de la Calidad - MAC) está dividida en 5 fases que ayudan a presenciar lo siguiente: menos errores en producción, registro de tiempos estimados, análisis de riesgos, mejor toma de decisiones, detalle del registro de las incidencias en el ambiente de pruebas, entre otras. Esto da lugar a generar valor agregado al cliente ya que su empresa también es beneficiada a trabajar con lineamientos y de nuestra parte brindar un servicio de calidad en todas las etapas del ciclo de vida de Software.

La nueva metodología permite trabajar de forma colaborativa con los involucrados pertinentes, en este caso son los programadores, los analistas de calidad, los jefes y/o coordinadores y los Stakeholders o interesados en el proyecto. El seguimiento es muy importante a nivel de requerimientos o proyectos, sirve para gestionar las actividades que se van realizando. La metodología MAC se ha elaborado con la intención de tener un lineamiento en la empresa para con los proyectos de desarrollo de Software, bajo esos estándares podemos tener proyectos exitosos que nos genere valor para otros casos en particular o que se nos presente, para eso se tiene una serie de artefactos que ayuden al proceso por las fases que este pase para su gestión respectivo.

En la Figura 18, a continuación se detalla el proceso de la nueva metodología a nivel de contexto, con sus respectivas fases y las personas involucradas en ella.

Figura 18

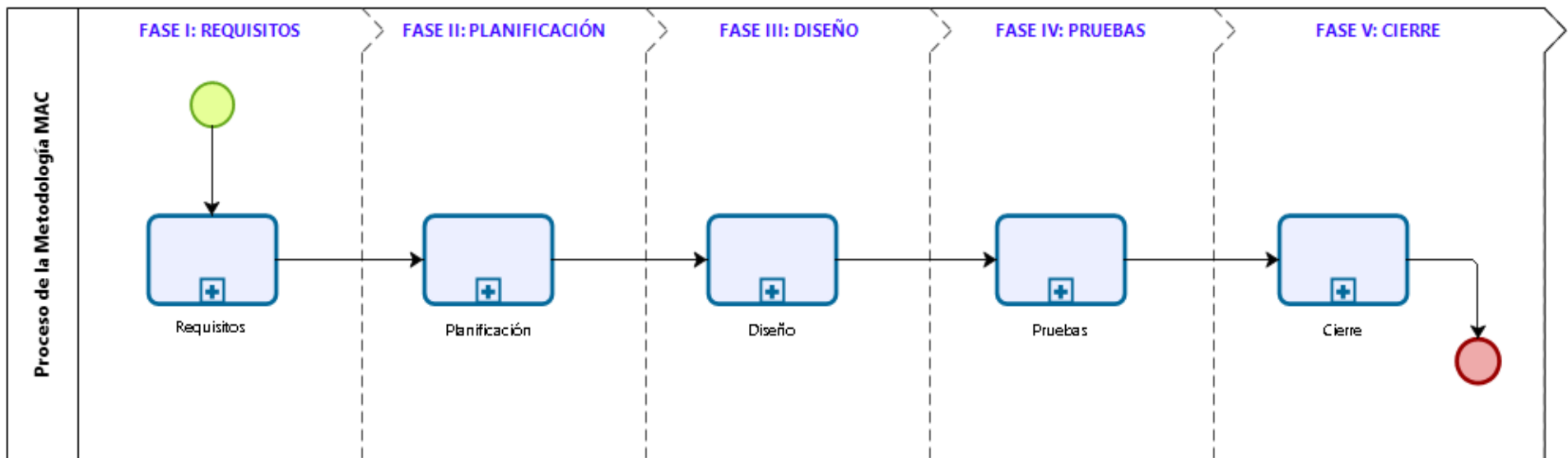
Diagrama de contexto de la nueva metodología MAC



En la Figura 19, a continuación se detalla el proceso de la nueva metodología a nivel macro, esto permite tener un mejor panorama para luego se pueda describir internamente a cada subproceso como fase con las actividades correspondientes en cada una de ellas.

Figura 19

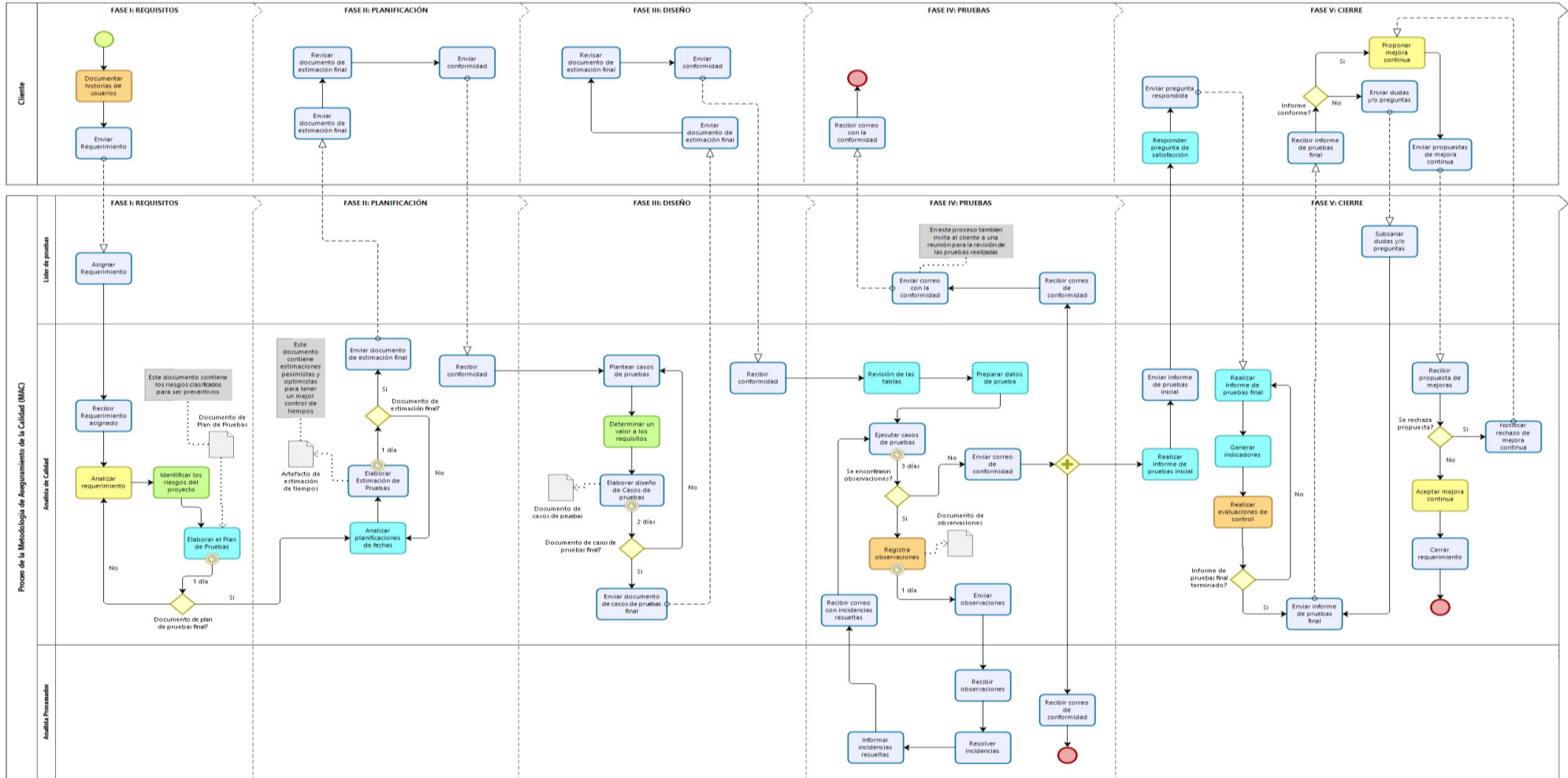
Flujograma tentativo – Nivel Macro



En la Figura 20, a continuación se observa el proceso de la nueva metodología a un nivel detallado.

Figura 20

Flujograma tentativo – Nivel detallado



4.1.3.1. Nueva Metodología – Fases. Para la implementación de la Metodología MAC se elabora cinco (5) fases:

– **Fase I: Requisitos**

La primera fase de la nueva Metodología parte de un enfoque de requerimientos de Software en un proyecto. Se establece a un alto nivel los aspectos desde el punto de vista de pruebas, se hacen necesarios para que apoye la mitigación del riesgo en base a la aparición de fallas en la operación del proceso. A continuación se describe lo que se debe de tomar en cuenta para esta primera fase del requerimiento inicial:

- a). **Revisión de la información de los requerimientos:** Se detalla la información de las necesidades del usuario final, a veces convertido en historias de usuario normalmente este concepto se usa para proyectos ágiles. Esta información está compuesta por requerimientos funcionales y no funcionales, se debe de revisar y analizar esta información debido a que algunos requerimientos están a nivel de negocio y solo el usuario del proceso lo llega a entender claramente. Los requerimientos son inputs para el trabajo del programador y el analista de calidad, para que desarrollen y revisen bien las casuísticas.
- b). **Descripción del Plan de Pruebas:** Para esta nueva metodología de aseguramiento de la calidad tiene como primer artefacto el plan de pruebas se muestra en la figura 17, aquí es donde se describe y detalla el análisis en base al punto de vista de las pruebas en general y el proyecto de software que lo abarca, sirve como guía para iniciar la ejecución de las pruebas. Esta parte es importante que se describa ya que contempla los riesgos que esta implementación podría tener u ocasionar a futuro y esto nos ayuda a prevenir los fallos que pueda tener el software en construcción en diferentes aspectos. A continuación se detalla cada uno de los puntos que este artefacto contempla:

- **Empresa:** Nombre de la empresa, en este caso es CANVIA que tiene clientes en empresas públicas o privadas.
- **Nombre del Proyecto:** Cada proyecto tiene un nombre diferente, dependiendo de la implementación que se quiere realizar.
- **Responsable QA:** Nombre de los responsables a revisar, en cada proyecto se asignan 1 o 2 analistas de calidad para realizar la revisión y ejecución de las pruebas, esto siempre es dependiendo de qué gran magnitud o complejidad sea el proyecto.
- **Objetivo:** Breve descripción del propósito de las pruebas en general.
- **Alcance:** En este punto se contemplan las funcionalidades nuevas que se abarca durante el proyecto en base a los requerimientos que han sido descritos en su momento.
- **Riesgos del proyecto:** Se describen los riesgos que el proyecto pueda contemplar, para los riesgos se toma el impacto por la probabilidad, para el valor de riesgo está definido a nivel general, pero especificado de la siguiente manera: Si esta entre 1.0 y menos a 3.0 es igual a un bajo riesgo (está en color verde), si esta entre 3.0 y menos a 4.5 es igual a un mediano riesgo (está en color amarillo) y si esta entre 4.5 y menos a 9.0 es igual a un alto riesgo (está en color rojo). Por lo tanto el resultado del riesgo es el valor que tome según lo definido sea, bajo, medio o alto. Para poder mitigar estos riesgos se considera una acción de propuesta, como un plan de contingencia a lo que pueda suceder y tomar medidas oportunas.

– Fase II: Planificación

Las actividades en la fase de planificación tienen lugar antes y después de la ejecución de las pruebas de Software. En la figura 18, aquí se tiene que gestionar las pruebas con los tiempos previstos según el proyecto, pero el área de aseguramiento de la calidad hace un análisis del requerimiento antes que sirve para hacer esta planificación de los tiempos y poder dar un estimación optimista y pesimista a la vez. El artefacto que se utiliza para esta metodología refleja la intención de lo que se quiere hacer y a su vez se hace el control de las actividades de la ejecución de las pruebas. Durante la planificación de las pruebas, nos aseguramos de que entendamos las metas y objetivos de los usuarios, de los interesados y del proyecto. La planificación de las pruebas tiene los siguientes puntos relevantes:

- a). **Medir y analizar los resultados de las revisiones y pruebas:** El proceso de una revisión en particular comienza con un requerimiento, el analista de calidad está encargado de estimar sus tiempos en base a las actividades que están definidas en el artefacto.
- b). **Descripción de estimación de pruebas:** Para esta nueva metodología de aseguramiento de la calidad tiene como segundo artefacto la estimación de pruebas figura 22, aquí es donde se describe y detalla los tiempos de las actividades del área de calidad, se consideran los tiempos de esfuerzos optimistas y los tiempos de esfuerzos pesimistas, ya que se tiene que dar un estimado de cuánto tiempo se terminara la ejecución de las pruebas, para que puedan cumplir con el cronograma y esto no impacte en los tiempos del proyecto, ya que se basan en un fecha de entrega al cliente y/o usuario final. A continuación se detalla cada uno de los puntos que este artefacto contempla:
 - **Datos del proyecto:** Nombre de la empresa, en este caso es CANVIA que tiene clientes en empresas públicas o privadas.

- **Metodología:** La metodología puede ser ágil o tradicional, dependiendo del proyecto en el que se está trabajando.
 - **Negocio y/o área impactada:** Nombre del área que se encarga del proceso a implementar los requerimientos.
 - **Aplicativo(s):** En que aplicativo o aplicativos se hará el cambio o modificaciones según el requerimiento.
 - **Código el proyecto:** Es un código o nomenclatura que se le asigna a cada proyecto.
 - **Nombre del proyecto:** Cada proyecto tiene un nombre diferente, dependiendo de la implementación que se quiere realizar.
 - **Número de horas planeadas por día:** Las horas trabajadas por días es de 8 horas jornadas laborables.
-
- **Actividades del Plan de Trabajo:** En este punto se describe las actividades que tiene que realizar el analista o los analistas de calidad, depende de cuantos analistas se trabaje si son 2 o más recursos, se ingresa las horas optimistas y pesimistas estimadas para dicho requerimiento dependiendo de la complejidad, dentro de esta incluido los tiempos de diseño, construcción y ejecución.
 - **Total del tiempo estimado de pruebas:** Son los tiempos totales de las actividades descritas anteriormente con sus respectivas horas estos son calculados automáticamente.

Figura 22*Artefacto de estimación de pruebas*

ESTIMACIÓN DE PRUEBAS			
Datos de Proyecto	Metodología		
	Negocio y/o Área impactada		
	Aplicativo (os)		
	Código del Proyecto		
	Nombre del Proyecto		
	# Horas planeadas por día		
Actividades del Plan de Trabajo	Total Esfuerzo Estimado OPTIMISTA (horas)	Total Esfuerzo Estimado PESIMISTA (horas)	Total Estimado (días)
Ante Proyecto (Requisitos)	0.00	0.00	0.00
Revisar los documentos del proyecto			
Realizar reunión para resolver consultas con QA			
Realizar estimación del proyecto			
Reunión de presentación de la estimación			
Actualizar la estimación y presentación			
Planificación	0.00	0.00	0.00
Cronograma de planificación de la etapa de pruebas			
Actualizar el cronograma (De ser necesario)			
Elaborar Plan de Prueba			
Diseño	0.00	0.00	0.00
Revisar los documentos del proyecto			
Realizar reunión de entendimiento / capacitación			
Diseñar los Casos de Pruebas			
Construcción	0.00	0.00	0.00
Obtener los Datos de pruebas			
Gestionar la instalación del componente en Ambiente QA			
Ejecución	0.00	0.00	0.00
Ejecutar las pruebas de humo			
Ejecutar los casos de pruebas			
Seguimiento de defectos			
Regresión a funcionalidad			
Elaborar el informe final de las pruebas (Cierre)			
Total Tiempo Estimado de Pruebas	0.00	0.00	0.00

– Fase III: Diseño

En esta fase se diseñan los casos de pruebas que permitirán hacer la revisión de los requerimientos según las funcionalidades indicadas por el usuario operativo. Se preparan también los datos de pruebas para ser usadas en la ejecución. Se utiliza el artefacto Diseño de Casos de Pruebas, es el documento que definen los datos de entradas, secuencias y/o pasos, resultados esperados, estado de las pruebas, iteraciones, tipo de errores, entre otros puntos. Los casos de pruebas tienen los siguientes puntos relevantes:

- a). **Especificar casos de pruebas no ambiguos:** Para describir un caso de prueba con sus especificaciones definidas, estos deben de ser claros ya que en un proyecto estos casos se pueden repartir entre analistas de calidad, para ser probados y cada uno de ellos deben de entender el mismo lenguaje o lo que el requerimiento está queriendo decir e implementar, por eso es de mucha importancia leer bien los requerimientos sobre todo entenderlos, que es lo que el usuario necesita para su operación del día a día.
- b). **Descripción de casos de pruebas:** Para esta nueva metodología de aseguramiento de la calidad tiene como tercer artefacto los casos de pruebas se muestra en la figura 23, aquí es donde se describe y detalla los casos de pruebas claros y concisos, este no debe de tener ambigüedad ya que se tiene el paso a paso de ejecución de las pruebas funcionales y no funcionales. Sirve también para hacer el seguimiento a las pruebas ya ejecutadas y las que aún faltan por probar, se tendrá visible el estado en el que se encuentra el caso de prueba, se ejecuta dependiendo de la prioridad para identificar que caso de prueba es relevante para el proyecto. A continuación se detalla cada uno de los puntos que este artefacto contempla:
 - **Nombre del proyecto:** Cada proyecto tiene un nombre diferente, dependiendo de la implementación que se quiere realizar.

- **Fecha de inicio de ejecución:** Fecha en que se inicia las pruebas.
- **Fecha de fin de la ejecución:** Fecha en que se culmina las pruebas.
- **Historia de usuario/requerimiento:** Si se trabaja con proyectos ágiles considerar historias de usuarios y si están con proyectos tradicionales considerar requerimiento, número consecutivo para llevar el seguimiento.
- **Módulo/ aplicación:** Indicar nombre del módulo o aplicación que se describe como caso de prueba, para tenerlo identificado.
- **Sprint (ágil):** Si se trabaja con proyectos ágiles considerar esta columna referente al sprint en que se está trabajando.
- **Escenario:** Es la numeración correlativa que se le asigna al caso de prueba.
- **Número del caso de prueba:** Es la numeración correlativa que se le asigna al caso de prueba.
- **Tipo de prueba:** La prueba puede estar identificada por funcional y no funcional según sea el caso.
- **Flujo:** El caso de prueba puede tener un flujo básico y/o alterno dependiendo de lo que se defina y según el requerimiento.
- **Nombre del caso de prueba:** Incluir un breve nombre del caso de prueba, para que se pueda identificar más rápido.
- **Descripción del caso de prueba:** Incluir una breve descripción del caso de prueba para que se pueda entender que cosa se prueba.
- **Condición:** El caso de prueba puede contener condiciones e ir por diferentes caminos al realizar la ejecución.
- **Datos de entrada:** Son los datos que se prueban y se ingresan al aplicativo para su ejecución.
- **Secuencia:** Se describe los pasos para ejecutar la acción de los casos de pruebas.

- **Resultado esperado:** Se describe el resultado del caso de prueba que sea claro y conciso.
- **Prioridad:** La prioridad la define el usuario, quiere decir cuál es relevante para la ejecución de las pruebas.
- **Estado de la prueba:** Es el estado en el que está el caso de prueba, se contempla por: conforme observado, pendiente, no aplica y bloqueado, según sea el caso de la ejecución.
- **Iteración:** Esto es el re test de las pruebas, ósea cuantas veces se ha probado el mismo caso de prueba, se culminará el re test cuando este todo conforme.
- **Tipo de error:** Esta descrito por los siguientes tipos: error de documentación, a nivel funcional, performance, de data, de ortografía y gramática, de código del software, de la interfaz de usuario gráfica y de usabilidad. Dependiendo de las observaciones que salgan en la ejecución de las pruebas se registra el tipo de error.
- **Severidad del error:** El caso de pruebas puede o no tener error, sino tiene error se le denomina “sin error”, pero si se encuentra alguna incidencia u observación en las pruebas esta se denomina con un valor de la siguiente manera:
 - **Impacto severo:** Igual al valor 1, el sistema falló. No es posible continuar con la certificación de los casos de prueba.
 - **Impacto significativo (medio):** Igual al valor 2, es posible continuar con la certificación de los casos de prueba sin embargo, el defecto impide certificar los casos asociados al mismo porque no existe una solución alternativa a la funcionalidad impactada.
 - **Impacto menor (bajo):** Igual al valor 3, existe una solución alternativa para reemplazar temporalmente la funcionalidad observada, esta deberá ser viable para el

usuario (No impactará en los tiempos de sus operaciones) y se deberá presentar un plan de acción para el cierre del defecto o un seguimiento.

- Igual al valor 4, que son los defectos de cosmética, documentación, acentos, etc.
- **Fecha de ejecución:** Esta fecha se refiere a la ejecución del caso de prueba, el día que se realiza las pruebas.

– Fase IV: Ejecución

En esta fase se ejecutan las pruebas descritas en los casos de pruebas. Se utiliza el artefacto de observaciones, es el documento que define como están descritas las incidencias, para que el programador pueda entenderlas y así puedan corregir el error del sistema. Para esta nueva metodología de aseguramiento de la calidad tiene como cuarto artefacto las observaciones se muestra en la figura 24, aquí es donde se describe y detalla las incidencias presentadas durante las pruebas de calidad, es importante que se trabaje de la mano con los casos de pruebas definidos anteriormente. A continuación se detalla cada uno de los puntos que este artefacto contempla:

- **CP_001:** Esta es una nomenclatura correlativa para identificar que caso de prueba se está observando.
- **Tipo de error:** Esta descrito por los siguientes tipos: error de documentación, a nivel funcional, performance, de data, de ortografía y gramática, de código del software, de la interfaz de usuario gráfica y de usabilidad. Dependiendo de las observaciones que salgan en la ejecución de las pruebas se registra el tipo de error.
- **Descripción de la observación:** Colocar una breve descripción de la observación encontrada que sea clara para que el programador pueda entender lo que le están observando en su aplicativo desarrollado.
- **Evidencias:** En esta sección se colocan las pantallas o imágenes de las observaciones que han salido durante la revisión de las pruebas.

Figura 24*Artefacto de observaciones/incidencias*

Empresa Desarrolladora de Software	Observaciones	FORMATO
		PPG - v0.1
	Revisión QA	22/02/2021
		Página 1 de 2

CP_001	
Tipo de error	
Descripción de la Observación	
Evidencias	

– **Fase V: Cierre**

En esta fase se realiza el informe de pruebas que permite tener un panorama más amplio de las pruebas a nivel gerencial y así se pueda tomar decisiones oportunas y esto ayude a la mejora continua de la empresa y los procesos de sus clientes, tener estos indicadores es relevante para el proyecto en general cuando se tengan reuniones y poder objetar algún indicio de acuerdo a estas cifras que salen por cada proyecto de desarrollo de software en cuanto a la revisión del área de calidad. Los casos de pruebas tienen los siguientes puntos relevantes:

- a). **Reporte con indicadores:** Para describir y realizar un reporte o informe de pruebas se necesitan datos, estos datos para este artefacto salen de los casos de pruebas. Cuando se hacen las pruebas ya se tienen los resultados, eso se logra registrar en el mismo artefacto de casos de pruebas y allí se lleva el control y seguimiento, teniendo todo eso en este artefacto de informe de pruebas podemos medir estos indicadores de resumen de issues (u observaciones) como el tipo de error y severidad, que son datos importantes para las pruebas.
- b). **Descripción del informe de pruebas:** Para esta nueva metodología de aseguramiento de la calidad tiene como quinto artefacto el informe de pruebas figura 25, aquí es donde se describe y detalla las observaciones de las pruebas de calidad que se han hecho anteriormente. Es importante tener un documento de evidencias de las pruebas a nivel general ya que eso ayuda a poder tomar decisiones en base a lo que se está haciendo o no bien, y quizás poder cambiar la manera de trabajar, o tener estándares que ayuden a este proceso. La metodología MAC que se implementa ayuda a tener ese lineamiento, y uno de eso es este artefacto, que está dirigido al jefe de proyecto y al cliente. En este documento también se contempla una pregunta para el usuario de la empresa a quien se le brinda el servicio este satisfecho con lo que se le ofrece como mejora a su proceso y por ende a su

nueva aplicación operacional. A continuación se detalla cada uno de los puntos que este artefacto contempla:

- **Fecha del informe:** Es la fecha en que se elabora el informe de pruebas.
- **Versión del aplicativo:** Es la versión del aplicativo que se ha revisado anteriormente, el aplicativo puede tener varias versiones desplegadas.
- **Aplicativo(s):** Aquí se incluye el nombre del aplicativo, en cada proyecto el aplicativo tiene un nombre diferente.
- **Sprint o iteración – Nombre del proyecto:** Si se trabaja con metodologías ágiles hay que incluir el sprint que duro el proyecto y si se trabaja de la forma tradicional se incluye en iteraciones.
- **Elaborado por:** Se coloca el nombre del analista de calidad a cargo que ejecuto las pruebas y elabora este informe final, para que luego se puedan tomar observaciones oportunas.
- **Estado del proyecto:** Este campo se puede colocar en proceso o terminado. Si está en estado en proceso quiere decir que hay una versión preliminar o inicial del informe y si está en terminado tiene ya una versión final.
- **Resumen de Issues:** En este resumen se presenta las observaciones que hubieron o se han presentado a la hora de hacer las pruebas de calidad, lo hace en gráficos estadísticos para llevar el control de esto, que también sirve como evidencias para futuros proyectos y se tenga evidencias de los proyectos que fueron exitosos o no.
- **Pregunta para el cliente:** Esta pregunta ayuda a tener un indicador de satisfacción del cliente cuando el proyecto y contrato haya terminado, es muy importante para la empresa saber cómo califica el cliente al servicio tercero que se le brinda durante todo el proceso de calidad.

Figura 25

Artefacto de Informe de pruebas

Informe de Pruebas			
---------------------------	--	--	--

Fecha Informe:		Sprint o Iteración - Nombre del Proyecto:	
Versión del Aplicativo:		Elaborado por:	
Aplicativo(s):		Estado del Proyecto	

Resumen de Issues	
--------------------------	--

Tipo de Error	# issues
Documentación	0
Funcional	0
Performance	0
Data	0
Ortografía-Gramática	0
Código	0
Interfaz de Usuario Gráfica	0
Usabilidad	0
Total	0

Usabilidad	0
Interfaz de Usuario Gráfica	0
Código	0
Ortografía-Gramática	0
Data	0
Performance	0
Funcional	0
Documentación	0

Severidad	# issues
Severo (1)	0
Alto (2)	0
Medio (3)	0
Bajo (4)	0
Total Impacto	0

0	0	0	0
Severo (1)	Alto (2)	Medio (3)	Bajo (4)

Pregunta para el cliente:

Responder marcando un "X" de acuerdo a lo que usted considere:

¿Cómo califica el servicio brindado?

Muy buena	<input type="checkbox"/>
Buena	<input type="checkbox"/>
Regular	<input type="checkbox"/>
Malo	<input type="checkbox"/>
Muy Malo	<input type="checkbox"/>

4.2. Implementación de las Buenas Prácticas (nueva Metodología)

Se requiere conocer primero a la empresa donde se aplica la nueva metodología, esta es la empresa CANVIA, a continuación se describe a la empresa.

4.2.1. Descripción de la empresa CANVIA

En CANVIA, impulsa y soporta el proceso de transformación digital de sus clientes, forman parte del portafolio de Advent Internacional y cuentan con 34 años de presencia en el mercado local y más de 2500 colaboradores. Su propósito es hacer más fácil la vida de las personas, innovando e implementando proyectos de transformación digital de nuestros clientes de manera ágil, segura y con resultados de negocio concretos.

Cuenta con un amplio portafolio de soluciones y servicios de transformación digital que se integran con tecnologías de IOT, Big Data, Inteligencia Artificial, Machine Learning y Robótica y que nos permiten plasmar soluciones disruptivas e innovadoras que transforman y optimizan los modelos y procesos de negocios centralizados en las necesidades del usuario.

- **Visión**

Ser la empresa referente en servicios de Transformación Digital en la Región Pacífico.

- **Misión**

Proveer soluciones innovadoras de Transformación Digital que aporten valor al negocio de nuestros clientes en base al conocimiento de industria, reconocidas por nuestra calidad de servicio, excelencia operacional y el talento de nuestros colaboradores.

- **Valores**

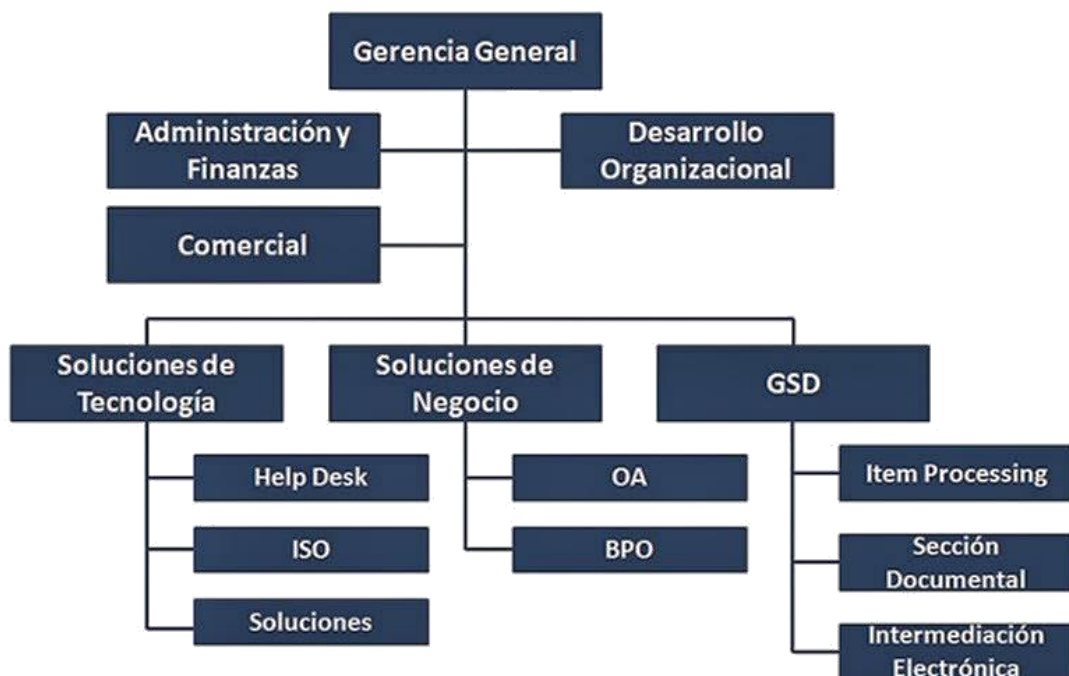
- Cumplimiento
- Eficiencia
- Calidad
- Innovación

- **Organigrama**

Proveer soluciones innovadoras de Transformación Digital que aporten valor al negocio de nuestros clientes en base al conocimiento de industria, reconocidas por nuestra calidad de servicio, excelencia operacional y el talento de nuestros colaboradores.

Figura 26

Organigrama de la empresa



a). Tercerización de Aplicaciones (OA)

El Outsourcing de Aplicación (OA) o tercerización de aplicaciones de CANVIA S.A. agrega valor mediante la optimización de carteras de aplicaciones, mayor rendimiento de la inversión y reducción del costo total de propiedad (TCO). Al externalizar las aplicaciones y sus servicios asociados, el cliente nos puede entregar la responsabilidad del soporte, mantenimiento, operación y administración de sus aplicaciones de negocio, bajo un contrato de servicios.

En CANVIA es posible tener un equipo de profesionales dedicado a administrar sus aplicaciones, o su Fábrica de Software puede trabajar sobre la demanda de acuerdo a las necesidades particulares.

Servicios que brinda son:

- Desarrollo y Mantenimiento
- Testing
- Soporte Funcional
- Procesos operativos y de soporte
- Fabricación de Software o Fábrica de Software

• Fábrica de Software

La solución de la Fábrica de software de CANVIA S.A., es un modelo de servicios que permite ayudar a gestionar el Mantenimiento Correctivo, Evolutivo y Desarrollo de sistemas de información. La propuesta se basa en Acuerdos de Niveles de Servicio (ANS) y Modelos de Estimación Claros, que garantiza la calidad del servicio, en términos de tiempo de respuesta, productividad y eficiencia.

Los pilares de la fábrica de software de CANVIA se basan en una probada estrategia metodológica, una vasta experiencia en el desarrollo de aplicaciones, equipo humano altamente calificado.

Esta Fábrica de software se desarrolla en dos Fases:

- a). Fase I - Implementación de un modelo de Software Factory:** Este modelo involucra la implementación de Metodologías de Gestión, Modelos de Estimación, Métricas, Acuerdos de Niveles de Servicio (ANS), Metodología de Desarrollo de Software y una definición clara de la interacción entre el cliente y CANVIA, para los procesos de Gestión de los Requerimientos, Planeamiento y Control de los Proyectos, Manejo de Cambios y Aceptación de los productos.

- b). Fase II - Servicio de Software Factory:** A través de este modelo de gestión se adaptan las prioridades del negocio definidas por el cliente. Los servicios que incluyen son: Mantenimiento Evolutivo y Mantenimiento Correctivo. Empresas del sector Industria, Comercio, Telecomunicaciones, Gobierno, Banca y Seguros, han confiado en la solución de la Fábrica de Software de CANVIA.

- **Personal involucrado**

- a). Gerente de proyecto**

- Persona cuyo rol se encarga de la responsabilidad total del planeamiento y la ejecución acertados del proyecto.

b). Coordinador de proyecto

Persona cuyo rol se encarga de gestionar, controlar y supervisar el ciclo de vida de un proyecto de software.

c). Líderes Técnicos de proyecto

Persona cuyo rol se encarga de apoyar al jefe de proyecto del área encargada.

d). Análisis de sistemas

- Entrevistar al cliente, ayudándole a identificar sus necesidades.
- Verificar si los requisitos especificados son los correctos.
- Definir una estructura básica del sistema que incluya fuentes de información, módulos de procesamiento de información y resultados esperados.
- Transformar los requerimientos en requisitos de software.

e). Programador

- Interactuar con los analistas de sistemas.
- Manejar lenguajes de programación validos actualmente.
- Manejar herramientas de generación de código.
- Generar prototipos rápidos del sistema.
- Realizar pruebas unitarias.
- Hacer la documentación del código.

f). Analista de Calidad

- Revisar las fases de requerimientos.

- Participar en la revisión de los requisitos del sistema.
- Revisar las políticas de control de cambios, control de errores y control de la configuración.
- Revisar la documentación.
- Realizar los casos de pruebas.

g). Consultores

Los consultores, cumplen el rol de especialistas en alguna determinada tecnología o herramientas, proveen información a consultas, también son asignados a empresas para resolver problemas y proveer soluciones tecnológicas.

4.2.2. Implementación de la Nueva Metodología (MAC)

A continuación se muestra los artefactos generados en un proyecto de Software, para que sean analizados y luego se puedan tomar decisiones preventivas y correctivas.

a). Fase de Requisitos

A continuación en la fase de requisitos se muestra el plan de pruebas general para el proyecto “Intranet virtual”, en la figura 23, donde se describe el objetivo, alcance y riesgos. En la parte de valorización de riesgo se ha puesto unos pesos que determinan si el riesgo es alto, medio o bajo, en este caso se está considerando para el peso alto un valor de entre 4.5 a 9.0, para el peso medio un valor de 3.0 a 4.5 y para el peso bajo un valor de 1.0 a 3.0.

De acuerdo a los riesgos descritos y ponerle el peso considerado a cada uno de ellos, el impacto y la posibilidad de ocurrencia van de la mano para el cálculo del riesgo, por ende se

puede observar que el riesgo promedio obtenido es de 7.3 esto quiere decir que es un riesgo alto porque está entre 4.5 a 9.0, también para este proyecto se considera incluir acciones propuesta, ya que así podemos tomar acciones en un tiempo oportuno a los riesgos altos.

Las acciones propuestas en este proyecto es poner fechas límites para las entregas para que no hallan retrasos, tener reuniones con los involucrados para no ver afectado el alcance del proyecto o de algún requerimiento que sea prioritario para el cliente, etc. Teniendo el plan de pruebas general descrito con todos el ítem's será la primera fase que se debe de tener para continuar con los demás artefactos.

Figura 27

Generación del plan de pruebas general

Plan de Pruebas General				
Empresa	Canvia S.A			
Nombre del Proyecto	Intranet virtual			
Responsable QA	Analista de Calidad 1			
Objetivo				
El propósito del Plan de Pruebas es proveer una referencia a los recursos que se encuentran trabajando en las tareas incluidas dentro del proceso de Pruebas de Calidad del Proyecto Intranet Virtual. En este documento se describe el alcance de la prueba el cual debe cumplir con las expectativas de los usuarios a través de los casos de prueba, identificación de los riesgos que se puedan presentar y el planteamiento de la estrategia a llevar a cabo en el proyecto.				
Alcance				
El alcance contemplará funcionalidades nuevas que serán implementadas en base a los requerimientos descritos en el documento funcional. A continuación los requerimientos:				
<ul style="list-style-type: none"> • Registro de servicios. • Completar información personal. • Seleccionar servicios a ofertarse. 				
Riesgos del Proyecto				
Valorización Riesgo	1.0<=Bajo<3.0		3.0<=Medio<4.5	4.5<=Alto<9.0
Descripción del Riesgo	Impacto	Posibilidad de Ocurrencia	Riesgo	Acción Propuesta
Retraso por incumplimiento en las fechas de entrega de los documentos	3	3	9	Proponer fechas límites de entrega de los documentos y realizar seguimiento para que se cumpla.
Cambio del alcance	3	3	9	Tener una reunión con el Coordinador(a) responsable y todo el equipo, para analizar si esos cambios pueden afectar con lo actual y si es posible realizarlo.
Retraso en entrega de parte del Desarrollador	3	3	9	Comunicar al Coordinador(a) y al equipo de Desarrollo sobre el tema, para buscar soluciones oportunas
Reproceso por cambiar de versión de los documentos de prueba.	3	2	6	Congelar el ambiente de pruebas y realizar un control de cambios
No detectar errores críticos por cambio de versión de los documentos de prueba sin coordinación con el analista de pruebas	3	2	6	Realizar control de cambios
Retraso por no tener acceso al ambiente de pruebas	3	2	6	Alinear el soporte de pruebas de acuerdo con el cronograma o tiempos que tendrá el analista de pruebas.
Cambio en los recursos del Proyecto	3	2	6	Que se capacite al nuevo integrante con documentación y/o diagramas de los flujos, etc.
Promedio			7,3	

b). Fase de Planificación

A continuación en la fase de planificación se muestra la estimación de las pruebas para el proyecto “Intranet virtual”, donde se describe los tiempos de las actividades que se realizan a lo largo de la revisión del aseguramiento y control de la calidad que el área encargada hará. Como se observa el artefacto estimación de pruebas tiene completado los datos del proyecto con los siguientes datos: la metodología, en esta parte para el proyecto mencionado se trabaja de una forma ágil, habrá proyectos que se trabajen de forma tradicional o cascada. Luego se menciona al negocio que va este proyecto, el aplicativo a trabajar es en web, se asigna un código de proyecto (es un número consecutivo de acuerdo a como están ordenados estos proyectos de Desarrollo de Software) y las horas planificadas por día siempre son 8 horas de jornada (o dependiendo del proyecto algunas son 9 horas, puede ser por políticas o normas de la empresa y/o proyectos).

En la figura 28, se muestra que se tiene el llenado del artefacto con las horas estimadas, dividiéndolo en las actividades generales que son el anteproyecto, la planificación de las pruebas, diseño de las pruebas, la construcción de los datos para las pruebas y netamente la ejecución de las pruebas. En el proyecto se está considerando el tiempo total de esfuerzo estimado optimista, que es el tiempo prudente para terminar la actividad y llegar a las fechas de compromiso con el cliente para este tiempo se ha estimado un total de 1163 horas con 10 minutos, también se considera el tiempo total de esfuerzo estimado pesimista, que es el tiempo que el proyecto puede tomar más de lo estimado y hace que atrase algunas actividades con prioridad, para ese tiempo se ha estimado un total de 1403 horas con 69 minutos. Entonces según los cálculos brindados por el artefacto para las horas pesimistas en días dan un total de 25 días aproximadamente y para las horas optimistas dan un total de 23 días aproximadamente.

Figura 28*Generación de la estimación de pruebas*

ESTIMACIÓN DE PRUEBAS			
Datos de Proyecto	Metodología	Ágil	
	Negocio y/o Área impactada	Empleados	
	Aplicativo (os)	Web	
	Código del Proyecto	0001	
	Nombre del Proyecto	Intranet Virtual	
	# Horas planeadas por día	8.00	
Actividades del Plan de Trabajo	Total Esfuerzo Estimado OPTIMISTA (horas)	Total Esfuerzo Estimado PESIMISTA (horas)	Total Estimado (días)
Ante Proyecto	67.50	68.55	8.57
Revisar los documentos del proyecto	21.00	22.05	2.76
Realizar reunión para resolver consultas con QA	2.00	2.00	0.25
Realizar estimación del proyecto	41.50	41.50	5.19
Reunión de presentación de la estimación	1.00	1.00	0.13
Actualizar la estimación y presentación	2.00	2.00	0.25
Planificación	15.00	15.00	1.88
Cronograma de planificación de la etapa de pruebas	8.00	8.00	1.00
Actualizar el cronograma (De ser necesario)	6.00	6.00	0.75
Elaborar Plan de Prueba	1.00	1.00	0.13
Diseño	117.60	145.14	18.14
Revisar los documentos del proyecto	24.00	25.20	3.15
Realizar reunión de entendimiento / capacitación	2.00	2.00	0.25
Diseñar los Casos de Pruebas			
Registrar de servicios	47.20	60.77	7.60
Completar información Personal	38.00	48.93	6.12
Seleccionar servicios a ofertase	6.40	8.24	1.03
Construcción	3.00	3.00	0.38
Obtener los Datos de pruebas	2.00	2.00	0.25
Gestionar la instalación del componente en Ambiente QA	1.00	1.00	0.13
Ejecución	960.00	1172.00	146.50
Ejecutar las pruebas de humo	120.00	133.50	16.69
Ejecutar los casos de pruebas	240.00	315.00	39.38
Seguimiento de defectos	240.00	276.00	34.50
Regresión a funcionalidad	280.00	367.50	45.94
Elaborar el informe final de las pruebas	80.00	80.00	10.00
Total Tiempo Estimado de Pruebas	1163.10	1403.69	175.46
			25 días aprox

c). Fase de Diseño

A continuación en la fase de diseño se muestra los casos de pruebas para el proyecto “Intranet virtual”, donde se describe todos los casos que se han identificado de acuerdo a los requerimientos funcionales y no funcionales que el cliente envía dependiendo de su necesidad. Como se observa en la figura 28 esta descrito por historias de usuarios y se está poniendo cada entregable con un sprint (entregable por cada módulo), se está considerando también flujos básicos para estos requerimientos y se ha identificado que está aplicando pruebas funcionales y no funcionales. La prioridad se definió junto con el usuario operativo, para saber qué casos son relevantes primero para el proceso de negocio, se le considera 1 ya que el usuario menciona que todos los casos son prioridad para este sprint 1.

Netamente para los casos de pruebas se está poniendo el requerimiento inicial, luego de eso como se observa en la figura 29 se describen los pasos o las secuencias correspondientes y el resultado esperado conciso y claro, ya que este artefacto de casos de pruebas también el cliente que se le brinda el servicio lo revisará para una aprobación.

En la parte de la ejecución se realiza en la siguiente fase aún, ya que allí se ejecutará todo lo descrito en este artefacto que se ha ido describiendo. Por ende contendrá los estados de dichas pruebas cuando este se realice.

Figura 29

Generación de los casos de pruebas

Intranet Virtual														
Fecha Inicio de Ejecución		19/09/2021												
Fecha Fin de Ejecución		19/11/2021												
Casos de Prueba														
N°	Historias de Usuario	Módulo/Aplicación	Sprint	Escenario	# Caso de Prueba	Tipo de Prueba	Flujo	Nombre del Caso de Prueba	Descripción del Caso de Prueba	Condición	Datos de Entrada	Secuencia (Pasos)	Resultado Esperado	Prioridad
1	HU1	Intranet Virtual	S1	E1	CP001	Funcional	Básico	Ingresar al Registro de Servicios	Verificar que el usuario pueda ingresar a la página de "Registro de trabajadores"	- Usuario debe estar logueado en el Portal	- Usuario: 43176444 - Contraseña: 123456	Portal Intranet Sección ¿Eres trabajador? 1. Presionar botón Ingresar aquí	El sistema mostrará la pantalla de inicio con un mensaje de bienvenida con la lista de servicios disponibles (Ingeniero, Administrador y Contador) y la opción de registrar nuevo servicio.	1
2	HU2	Intranet Virtual	S1	E1	CP003	Funcional	Básico	Seleccionar servicios a ofertarse	Verificar que el usuario pueda seleccionar un servicio disponible	- Usuario debe estar logueado en el Portal	- Usuario: 43176444 - Contraseña: 123456 - Encuentro servicio?: Si	Directorio de trabajadores 1. ¿Encontró Ud. el/los servicio(s) que brinda?Elegir la opción: Si	El sistema habilitará el botón continuar.	1
3	HU2	Intranet Virtual	S1	E1	CP003	Funcional	Básico	Continuar con el proceso	Verificar que el usuario pueda seleccionar un servicio disponible	- Usuario debe estar logueado en el Portal	- Usuario: 43176444 - Contraseña: 123456	1. Presionar botón continuar	El sistema mostrará la pantalla "Directorio de trabajadores".	1
4	HU3	Intranet Virtual	S1	E1	CP004	Funcional	Básico	Visualizar información Personal disponible	Verificar que el usuario pueda visualizar en modo lectura el tipo documento, número de documento, nombres, apellido paterno, apellido materno, nombres, dirección, celular y poder continuar con lo correspondiente.	- Usuario debe estar logueado en el Portal	- Usuario: 43176444 - Contraseña: 123456 - Habilita dirección actual?: Si - Correo visible?: Si	Sección Información Personal 1. Visualizar Tipo documento 2. Visualizar Número de documento 3. Visualizar Nombres 4. Visualizar Apellido Paterno 5. Visualizar Apellido Materno 6. Visualizar Dirección (RENIEC) 7. Habilitar dirección actual 8. Ingresar dirección actual 9. Seleccionar Ubigeo 10. Visualizar número celular 11. Ingresar correo alterno 12. Desea que esta información sea visible en su perfil?Presionar check Si	El sistema permitirá que el número de celular sea visible en su perfil. El sistema mostrará visible el correo alterno.	1

d). Fase de Ejecución

A continuación en la fase de ejecución se muestra las observaciones encontradas para el proyecto “Intranet virtual”, donde se describe las observaciones a detalle junto a sus evidencias con sus respectivas pantallas (si es que tuviera), en el proyecto se encontraron 2 observaciones en el caso de prueba 002 para el primer sprint, se muestra en la figura 31. Estas observaciones están por tipo de error: data, ya que tiene que ver con la base de datos y los servicios que se consulta para un usuario específico.

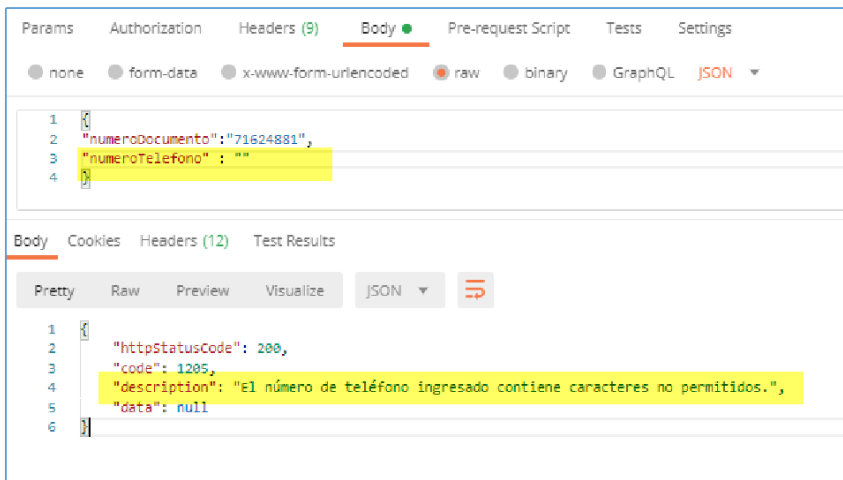
Como se muestra en la siguiente figura 30, para este requerimiento se está utilizando el postman (programa que se utiliza específicamente para este proyecto) para hacer la ejecución de las pruebas a nivel de servicio, quiere decir una prueba no funcional. Esta ejecución se realiza de la mano con los casos de pruebas. Este artefacto será enviado al equipo de desarrollo, que le servirá para subsanar las observaciones encontradas por el equipo de Calidad de Software.

Como podemos ver en la figura 31, en la parte de ejecución ya están descritos los campos referentes a las verificaciones de las pruebas, quiere decir que ya se está haciendo una primera revisión o una primera iteración dependiendo de los escenarios que se tiene. Este dato o llenado de campos, servirá para el informe de las pruebas que es el otro artefacto siguiente.

Figura 30

Registro de las observaciones encontradas

Empresa Desarrolladora de Software	Observaciones	FORMATO
	Revisión QA	PPG - v0.1
		22/02/2021
		Página 1 de 1

CP_002	Consultar Documento - code diferente 1205
Tipo error	Data
Descripción de la Observación	
Cuando se pone el "teléfono" vacío, el servicio muestra el mensaje "El número de teléfono ingresado contiene caracteres no permitidos" que pertenece a el código 1205.	
Evidencias	
	

CP_002	Consultar Documento - code diferente 1204
Tipo error	Data
Descripción de la Observación	
Cuando se pone el "DNI" vacío, el servicio muestra el mensaje "El número de DNI ingresado es incorrecto." Que pertenece a el código 1204.	
Evidencias	
	

Figura 31

Estado de las pruebas

Intranet Virtual

Fecha Inicio de Ejecución	19/09/2021
Fecha Fin de Ejecución	19/11/2021

N°	Historias de Usuario	Módulo/Aplicación	Sprint	Escenario	# Caso de Prueba	Tipo de Prueba	Flujo	Casos de Prueba						Prioridad	Ejecución				
								Nombre del Caso de Prueba	Descripción del Caso de Prueba	Condición	Datos de Entrada	Secuencia (Pasos)	Resultado Esperado		Estado de Prueba	Iteración	Tipo de Error	Severidad de Error	Fecha de Ejecución
1	HU1	Intranet Virtual	S1	E1	CP001	Funcional	Básico	Ingresar al Registro de Servicios	Verificar que el usuario pueda ingresar a la página de "Registro de trabajadores"	- Usuario debe estar logueado en el Portal	- Usuario: 43176444 - Contraseña: 123456	Portal Intranet Sección ¿Eres trabajador? 1. Presionar botón Ingresa aquí	El sistema mostrará la pantalla de inicio con un mensaje de bienvenida con la lista de servicios disponibles (Ingeniero, Administrador y Contador) y la opción de registrar nuevo servicio.	1	Conforme	1		Sin Error	20/09/2021
2	HU2	Intranet Virtual	S1	E1	CP003	Funcional	Básico	Seleccionar servicios a ofertarse	Verificar que el usuario pueda seleccionar un servicio disponible	- Usuario debe estar logueado en el Portal	- Usuario: 43176444 - Contraseña: 123456 - Encuentro servicio?: Si	Directorio de trabajadores 1. ¿Encontró Ud. el/los servicio(s) que brinda? :::::Elegir la opción: Si	El sistema habilitará el botón continuar.	1	Observado	1	Documentación	4	20/09/2021
3	HU2	Intranet Virtual	S1	E1	CP003	Funcional	Básico	Continuar con el proceso	Verificar que el usuario pueda seleccionar un servicio disponible	- Usuario debe estar logueado en el Portal	- Usuario: 43176444 - Contraseña: 123456	1. Presionar botón continuar	El sistema mostrará la pantalla "Directorio de trabajadores".	1	Conforme	1		Sin Error	21/09/2021
4	HU3	Intranet Virtual	S1	E1	CP004	Funcional	Básico	Visualizar información Personal disponible	Verificar que el usuario pueda visualizar en modo lectura el tipo documento, número de documento, nombres, apellido paterno, apellido materno, nombres, dirección, celular y poder continuar con lo correspondiente.	- Usuario debe estar logueado en el Portal	- Usuario: 43176444 - Contraseña: 123456 - Habilita dirección actual?: Si - Correo visible?: Si	Sección Información Personal 1. Visualizar Tipo documento 2. Visualizar Número de documento 3. Visualizar Nombres 4. Visualizar Apellido Paterno 5. Visualizar Apellido Materno 6. Visualizar Dirección (RENEC) 7. Habilitar dirección actual 8. Ingresar dirección actual 9. Seleccionar Ubigeo 10. Visualizar número celular 11. Ingresar correo alterno 12. Desea que esta información sea visible en su perfil? :::::Presionar check Si	El sistema permitirá que el número de celular sea visible en su perfil. El sistema mostrará visible el correo alterno.	1	Observado	2	Data	1	21/09/2020 retest 24/09/2020

e). Fase de Cierre

A continuación en la fase de cierre se muestra el informe general de las pruebas para el proyecto “Intranet virtual”, donde se describe el resumen de pruebas que se hicieron. Para el proyecto se trabaja en 2 sprint es lo que se tiene descrito en la figura 32, este resumen de observaciones o issues se suma el total de observaciones que se encontró en las revisiones del aplicativo como se muestra en la figura 31, aparte también en el resumen se considera la severidad que se puso anteriormente a la observación se muestra con un cuadro gráfico que ayuda a ser más visible el informe.

Este informe será entregado al Scrum Master y Stakeholders para ver los resultados de las pruebas para la toma de decisiones y mejora continua que se tendrán para dicho proyecto cerrado, en caso luego haya otros sprint's. Dentro de este artefacto también se toma la opinión del usuario al que se le brinda el servicio en este caso el usuario responsable respondió que se siente satisfecho con lo brindado e inclusive incluyó unos comentarios adicionales, que será tomado en cuenta para los siguientes entregables.

4.3. Resultados de los indicadores

A continuación se muestra los valores obtenidos por cada indicador tanto en la PosPrueba Gc y PosPrueba Ge.

Tabla 20

Resultados de la PosPrueba Gc y PosPrueba Ge para los 11, 12, 13, 14 e 15

N°	Indicador 01		Indicador 02		Indicador 03		Indicador 04		Indicador 05	
	Tiempo de entrega (horas)		Tiempo para la revisión de errores (horas)		Números de errores en pruebas		Costo total del proceso (Soles)		Satisfacción del Cliente	
	Pos Prueba Gc	Pos Prueba Ge	Pos Prueba Gc	Pos Prueba Ge	Pos Prueba Gc	Pos Prueba Ge	Pos Prueba Gc	Pos Prueba Ge	Pos Prueba Gc	Pos Prueba Ge
1	40	23	30	22	52	35	2300	1900	2	4
2	48	21	20	16	60	50	4000	2800	4	5
3	42	26	32	21	40	32	2500	1800	5	5
4	35	20	27	12	45	15	2100	1100	1	5
5	32	23	26	16	59	30	3200	1900	3	4
6	30	21	32	20	46	10	4000	3200	4	5
7	48	25	29	23	60	40	3500	2500	4	4
8	45	30	24	14	50	35	3600	2100	1	5
9	37	13	28	12	35	21	4000	2000	3	5
10	39	20	30	13	57	40	3700	2100	1	5
11	35	15	32	20	60	50	2500	1800	1	4
12	48	30	31	26	56	50	2900	2200	2	4
13	31	21	22	14	50	40	4000	1500	3	5
14	48	36	25	20	40	25	3100	2300	2	3
15	46	32	32	13	35	12	2800	2200	2	5
16	29	10	30	15	48	32	3200	2500	5	5
17	25	21	28	10	56	40	2700	1100	1	3
18	30	28	27	16	55	43	3200	1500	3	4
19	48	35	25	10	60	52	3100	1300	3	4
20	32	21	32	23	46	35	2200	1800	4	5
21	43	23	29	19	40	30	2900	1600	2	3
22	26	10	26	20	30	10	3600	1800	2	5
23	29	17	23	15	38	15	3700	1100	3	4
24	32	20	32	11	55	20	2600	1800	1	4
25	39	19	23	12	45	22	3900	1300	3	5
26	27	11	20	10	34	19	3700	1200	2	3
27	34	23	26	19	38	15	3500	2500	3	5
28	32	12	30	13	29	14	2200	1150	1	5
29	20	11	32	24	55	12	3900	2500	2	4
30	22	15	25	28	35	15	3800	2600	2	3

Tabla 21*Promedio de los indicadores de la PosPrueba Gc y PosPrueba Ge*

Indicador	Promedios	
	PosPrueba Gc	PosPrueba Ge
I01: Tiempo de entrega	35.73	21.07
I02: Tiempo para la revisión de errores	27.60	16.90
I03: Números de errores en pruebas	40.03	28.63
I04: Costo total del proceso	3213.33	1905.00
I05: Satisfacción del Cliente	2.50	4.33

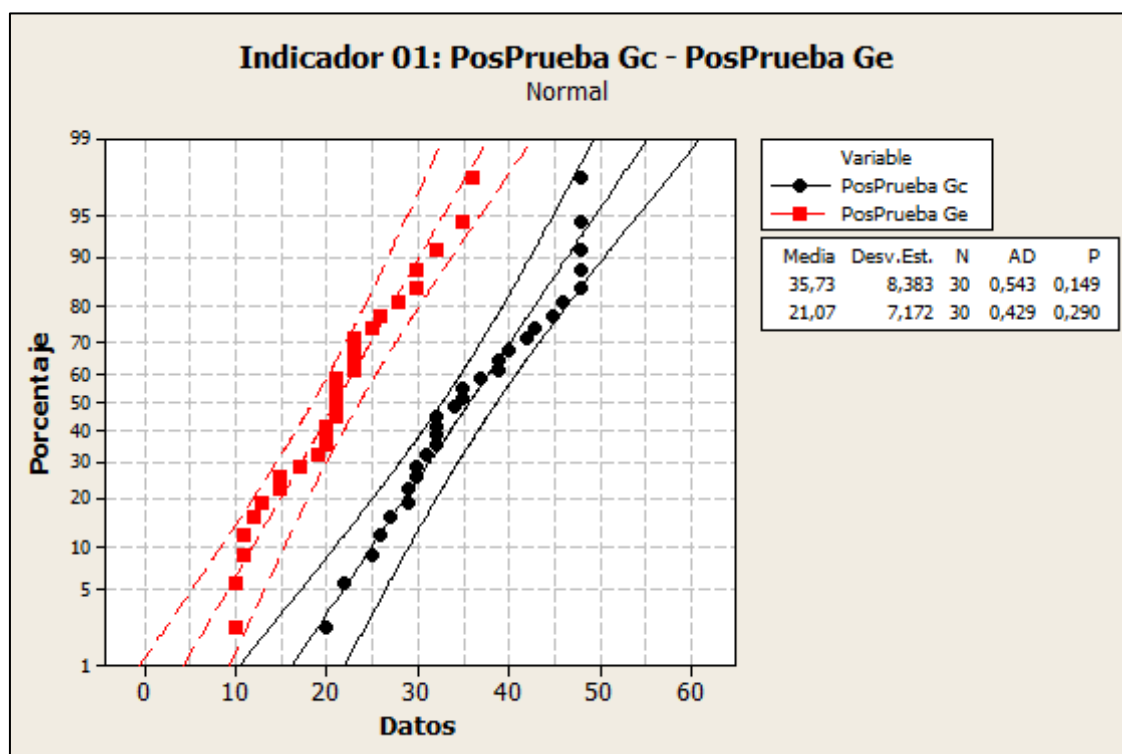
4.4. Pruebas de normalidad

Se realiza la prueba de normalidad en cada uno de los indicadores para ver si los datos tienen comportamiento normal.

a. Indicador 01: Tiempo de entrega (horas)

Figura 33

Pruebas de normalidad – Indicador 01 (PosPrueba Gc - PosPrueba Ge)

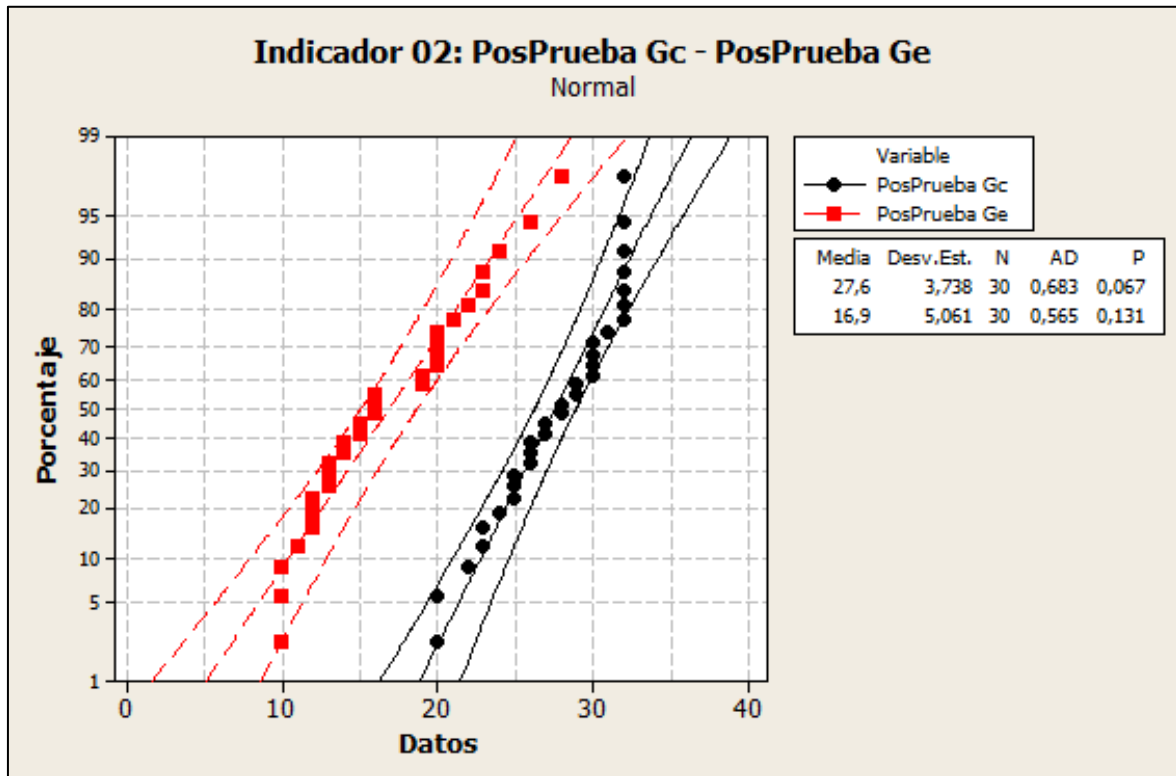


Conclusión: Se observa de la Figura 33, el resultado de la prueba Anderson Darling muestra que el valor p es mayor que 0,05 para el PosPrueba Gc ($p=0,149$) y PosPrueba Ge ($p=0,290$), lo cual confirma que presentan una distribución normal.

b. Indicador 02: Tiempo para la revisión de errores (horas)

Figura 34

Pruebas de normalidad – Indicador 02 (PosPrueba Gc – PosPrueba Ge)

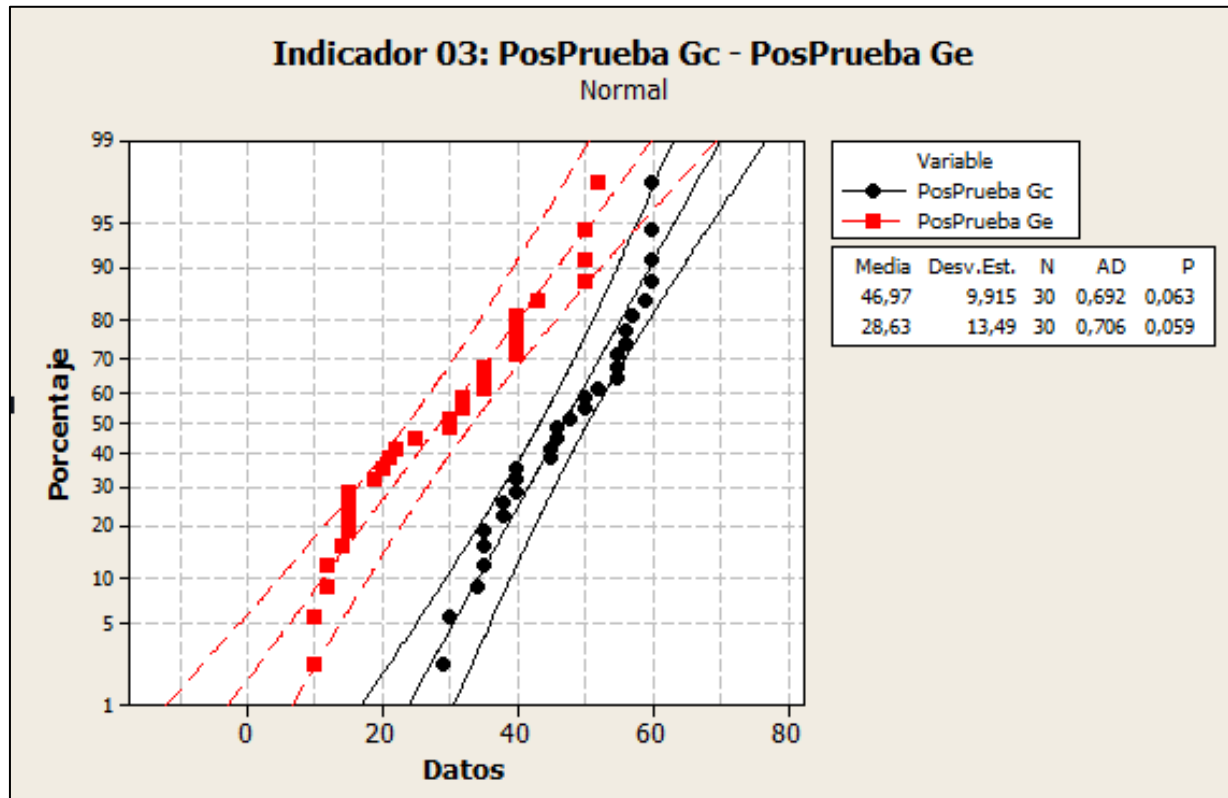


Conclusión: Se observa de la Figura 34, el resultado de la prueba Anderson Darling muestra que el valor p es mayor que 0,05 para el PosPrueba Gc ($p=0,067$) y PosPrueba Ge ($p=0,131$), lo cual confirma que presentan una distribución normal.

c. Indicador 03: Número de errores en calidad

Figura 35

Pruebas de normalidad – Indicador 03 (PosPrueba Gc – PosPrueba Ge)

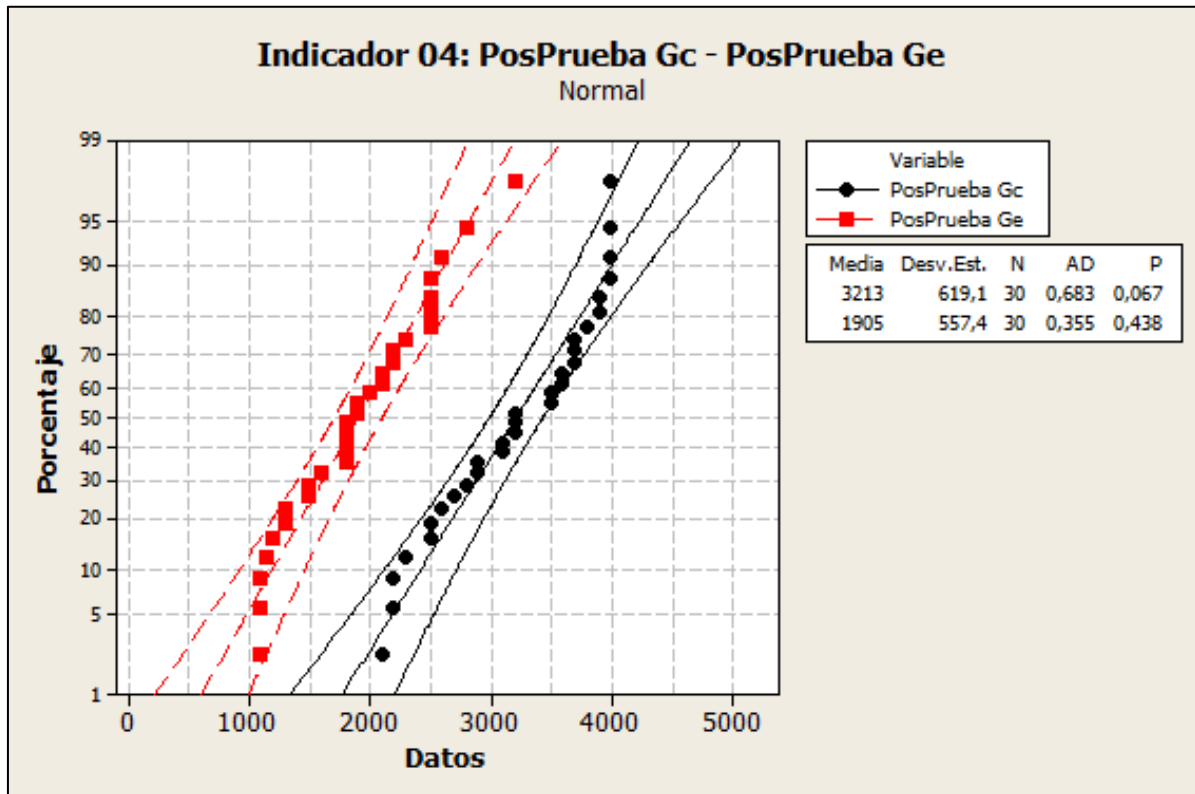


Conclusión: Se observa de la Figura 35, el resultado de la prueba Anderson Darling muestra que el valor p es mayor que 0,05 para el PosPrueba Gc ($p=0,063$) y PosPrueba Ge ($p=0,059$), lo cual confirma que presentan una distribución normal.

d. Indicador 04: Costo total del proceso

Figura 36

Pruebas de normalidad – Indicador 04 (PosPrueba Gc – PosPrueba Ge)



Conclusión: Se observa de la Figura 36, el resultado de la prueba Anderson Darling muestra que el valor p es mayor que 0,05 para el PosPrueba Gc ($p=0,067$) y PosPrueba Ge ($p=0,438$), lo cual confirma que presentan una distribución normal.

4.5. Contrastación de hipótesis

En todos los casos anteriores la prueba de normalidad de datos fue aceptada, por consiguiente se aplica la prueba de t-student para los indicadores paramétricos y Mann-Whitney para los indicadores no paramétricos para la contrastación de las hipótesis para los grupos de control (Gc) y experimental (Ge) en la aplicación de la Metodología MAC.

a. Hipótesis específica 01: Tiempo de entrega

Hi: El uso de las Buenas Prácticas basada en la Metodología MAC **reduce**, el tiempo de entrega para el Aseguramiento de la Calidad del Desarrollo de Software en la Empresa CANVIA S.A. (PosPrueba Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

Se realizó una medición sin el uso de las Buenas Prácticas basada en la Metodología MAC (PosPrueba del Gc) y otra con el uso de las Buenas Prácticas basada en la Metodología MAC (PosPrueba del Ge):

Tabla 22

Resultados del Indicador 01 - PosPrueba Ge y PosPrueba del Gc

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PosPrueba Ge	23	21	26	20	23	21	25	30	13	20	15	30	21	36	32
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	10	21	28	35	21	23	10	17	20	19	11	23	12	11	15
PosPrueba Gc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	40	48	42	35	32	30	48	45	37	39	35	48	31	48	46
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	29	25	30	48	32	43	26	29	32	39	27	34	32	20	22

- **Planteamiento de las hipótesis nula y alterna**

- **H₀**: El uso de las Buenas Prácticas basada en la Metodología MAC **no reduce**, el tiempo de entrega para el Aseguramiento de la Calidad del Desarrollo de Software en la Empresa CANVIA S.A. (PosPrueba Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).
- **H_a**: El uso de las Buenas Prácticas basada en la Metodología MAC **reduce**, el tiempo de entrega para el Aseguramiento de la Calidad del Desarrollo de Software en la Empresa CANVIA S.A. (PosPrueba Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

μ_1 = Media poblacional del **tiempo de entrega** en la PosPrueba del Gc.

μ_2 = Media poblacional del **tiempo de entrega** en la PosPrueba del Ge.

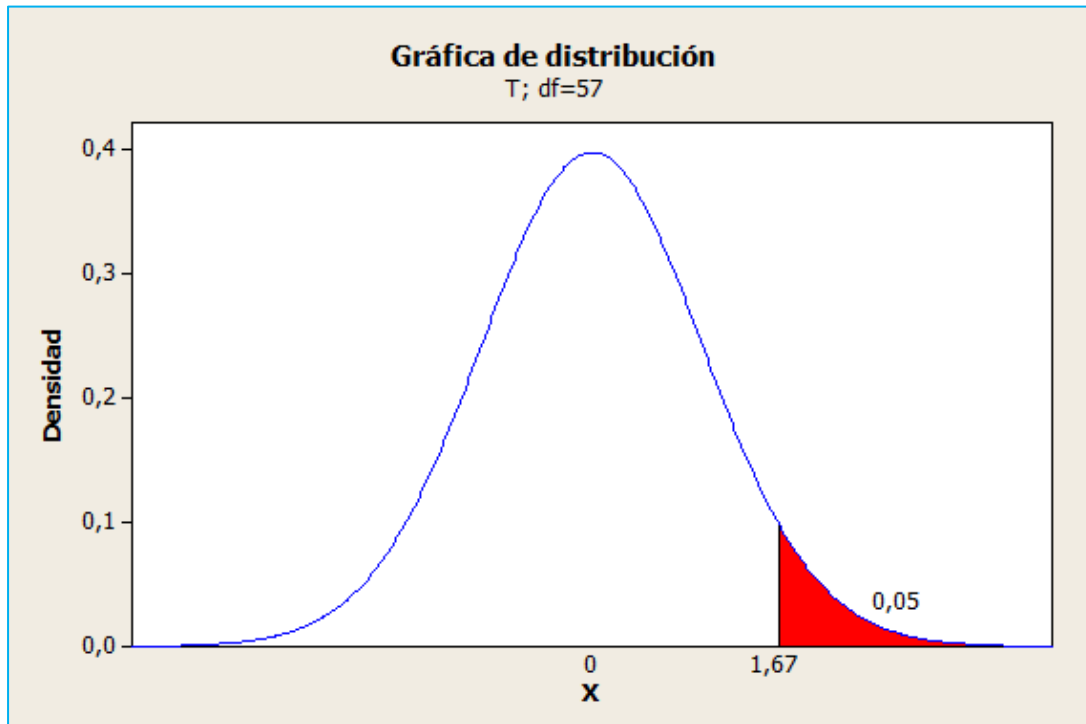
– **H₀**: $\mu_1 \leq \mu_2$

– **H_a**: $\mu_1 > \mu_2$

- Criterios de decisión

Figura 37

Indicador 01 - Gráfico para el criterio de decisión



- Cálculo: Prueba t para medias de las dos muestras

Tabla 23

Indicador 01 - Prueba t para medias de las dos muestras

	PosPrueba Gc	PosPrueba Ge
Media (\bar{X})	35.73	21.07
Desviación Estándar (S)	8.383	7.172
Observaciones (n)	30	30
Diferencia hipotética de las medias		0
Estadístico t		7.28
P(T<=t) una cola		0.000
Valor crítico de t (una cola)		1.67

- **Decisión estadística**

Debido a que el valor- $p = (0.000) < \alpha = (0.05)$, los resultados proporcionan la evidencia necesaria para rechazar la **hipótesis nula (H₀)**, por lo tanto la **hipótesis alterna (H_a)** se acepta. La prueba resultó ser significativa.

b. Hipótesis específica 02: Tiempo para la revisión de errores

Hi: El uso de las Buenas Prácticas basada en la Metodología MAC **reduce**, el tiempo para la revisión de errores para el Aseguramiento de la Calidad del Desarrollo de Software en la Empresa CANVIA S.A. (PosPrueba Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

Se realizó una medición sin el uso de las Buenas Prácticas basada en la Metodología MAC (PosPrueba del Gc) y otra con el uso de las Buenas Prácticas basada en la Metodología MAC (PosPrueba del Ge):

Tabla 24

Resultados del Indicador 02 - PosPrueba Ge y PosPrueba del Gc

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PosPrueba Ge	22	16	21	12	16	20	23	14	12	13	20	26	14	20	13
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	15	10	16	10	23	19	20	15	11	12	10	19	13	24	28
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PosPrueba Gc	30	20	32	27	26	32	29	24	28	30	32	31	22	25	32
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	30	28	27	25	32	29	26	23	32	23	20	26	30	32	25

• **Planteamiento de las hipótesis nula y alterna**

- **Ho:** El uso de las Buenas Prácticas basada en la Metodología MAC **no reduce**, el tiempo para la revisión de errores para el Aseguramiento de la Calidad del Desarrollo de Software en la Empresa CANVIA S.A. (PosPrueba Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).
- **Ha:** El uso de las Buenas Prácticas basada en la Metodología MAC **reduce**, el tiempo para la revisión de errores para el Aseguramiento de la Calidad del Desarrollo de

Software en la Empresa CANVIA S.A. (PosPrueba Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

μ_1 = Media poblacional del **tiempo para la revisión de errores** en la PosPrueba del Gc.

μ_2 = Media poblacional del **tiempo para la revisión de errores** en la PosPrueba del Ge.

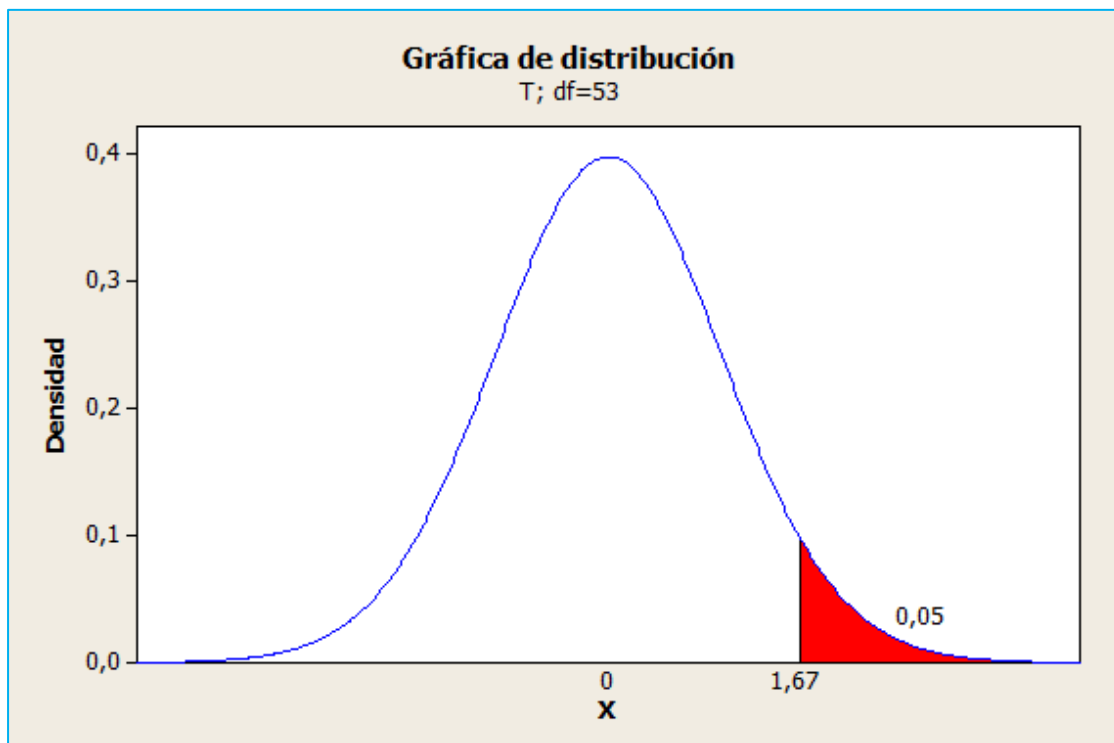
– **H₀**: $\mu_1 \leq \mu_2$

– **H_a**: $\mu_1 > \mu_2$

- **Criterios de decisión**

Figura 38

Indicador 02 - Gráfico para el criterio de decisión



- **Cálculo: Prueba t para medias de las dos muestras**

Tabla 25

Indicador 02 - Prueba t para medias de las dos muestras

	PosPrueba Gc	PosPrueba Ge
Media (\bar{X})	27.60	16.90
Desviación Estándar (S)	3.738	5.061
Observaciones (n)	30	30
Diferencia hipotética de las medias		0
Estadístico t		9.32
P(T<=t) una cola		0.000
Valor crítico de t (una cola)		1.67

- **Decisión estadística**

Debido a que el valor-p = (0.000) < α = (0.05), los resultados proporcionan la evidencia necesaria para rechazar la **hipótesis nula (Ho)**, por lo tanto la **hipótesis alterna (Ha)** se acepta. La prueba resultó ser significativa.

c. Hipótesis específica 03: Numero de errores en pruebas

Hi: El uso de las Buenas Prácticas basada en la Metodología MAC **disminuye**, el número de errores e pruebas para el Aseguramiento de la Calidad del Desarrollo de Software en la Empresa CANVIA S.A. (PosPrueba Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

Se realizó una medición sin el uso de las Buenas Prácticas basada en la Metodología MAC (PosPrueba del Gc) y otra con el uso de las Buenas Prácticas basada en la Metodología MAC (PosPrueba del Ge):

Tabla 26

Resultados del Indicador 03 - PosPrueba Ge y PosPrueba del Gc

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PosPrueba Ge	35	50	32	15	30	10	40	35	21	40	50	50	40	25	12
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	32	40	43	52	35	30	10	15	20	22	19	15	14	12	15
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PosPrueba Gc	52	60	40	45	59	46	60	50	35	57	60	56	50	40	35
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	48	56	55	60	46	40	30	38	55	45	34	38	29	55	35

• **Planteamiento de las hipótesis nula y alterna**

- **Ho:** El uso de las Buenas Prácticas basada en la Metodología MAC **no disminuye**, el número de errores en pruebas para el Aseguramiento de la Calidad del Desarrollo de Software en la Empresa CANVIA S.A. (PosPrueba Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).
- **Ha:** El uso de las Buenas Prácticas basada en la Metodología MAC **disminuye**, el número de errores en pruebas para el Aseguramiento de la Calidad del Desarrollo de

Software en la Empresa CANVIA S.A. (PosPrueba Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

μ_1 = Media poblacional del **número de errores en pruebas** en la PosPrueba del Gc.

μ_2 = Media poblacional del **número de errores en pruebas** en la PosPrueba del Ge.

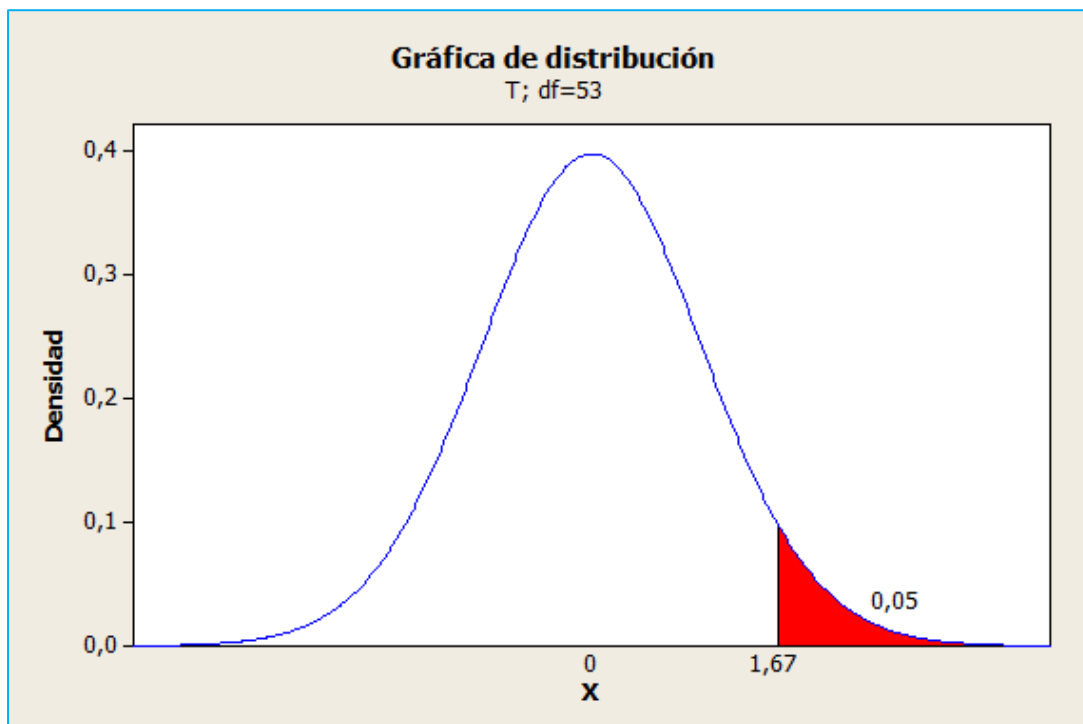
– **H₀**: $\mu_1 \leq \mu_2$

– **H_a**: $\mu_1 > \mu_2$

- **Criterios de decisión**

Figura 39

Indicador 03 - Gráfico para el criterio de decisión



- **Cálculo: Prueba t para medias de las dos muestras**

Tabla 27

Indicador 03 - Prueba t para medias de las dos muestras

	PosPrueba Gc	PosPrueba Ge
Media (\bar{X})	46.97	28.63
Desviación Estándar (S)	9.915	13.49
Observaciones (n)	30	30
Diferencia hipotética de las medias		0
Estadístico t		6.00
P(T \leq t) una cola		0.000
Valor crítico de t (una cola)		1.67

- **Decisión estadística**

Debido a que el valor-p = (0.000) < α = (0.05), los resultados proporcionan la evidencia necesaria para rechazar la **hipótesis nula (Ho)**, por lo tanto la **hipótesis alterna (Ha)** se acepta. La prueba resultó ser significativa.

d. Hipótesis específica 04: Costo total del Proceso

Hi: El uso de las Buenas Prácticas basada en la Metodología MAC **disminuye**, el costo total del proceso para el Aseguramiento de la Calidad del Desarrollo de Software en la Empresa CANVIA S.A. (PosPrueba Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

Se realizó una medición sin el uso de las Buenas Prácticas basada en la Metodología MAC (PosPrueba del Gc) y otra con el uso de las Buenas Prácticas basada en la Metodología MAC (PosPrueba del Ge):

Tabla 28

Resultados del Indicador 04 - PosPrueba Ge y PosPrueba del Gc

	1	2	3	4	5	6	7	8	9	10
	1900	2800	1800	1100	1900	3200	2500	2100	2000	2100
PosPrueba Ge	11	12	13	14	15	16	17	18	19	20
	1800	2200	1500	2300	2200	2500	1100	1500	1300	1800
	21	22	23	24	25	26	27	28	29	30
	1600	1800	1100	1800	1300	1200	2500	1150	2500	2600
	1	2	3	4	5	6	7	8	9	10
	2300	4000	2500	2100	3200	4000	3500	3600	4000	3700
PosPrueba Gc	11	12	13	14	15	16	17	18	19	20
	2500	2900	4000	3100	2800	3200	2700	3200	3100	2200
	21	22	23	24	25	26	27	28	29	30
	2900	3600	3700	2600	3900	3700	3500	2200	3900	3800

- **Planteamiento de las hipótesis nula y alterna**

- **Ho:** El uso de las Buenas Prácticas basada en la Metodología MAC **no disminuye**, el costo total del proceso para el Aseguramiento de la Calidad del Desarrollo de Software

en la Empresa CANVIA S.A. (PosPrueba Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

- **Ha:** El uso de las Buenas Prácticas basada en la Metodología MAC **disminuye**, el costo total del proceso para el Aseguramiento de la Calidad del Desarrollo de Software en la Empresa CANVIA S.A. (PosPrueba Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

μ_1 = Media poblacional del **costo total del proceso** en la PosPrueba del Gc.

μ_2 = Media poblacional del **costo total del proceso** en la PosPrueba del Ge.

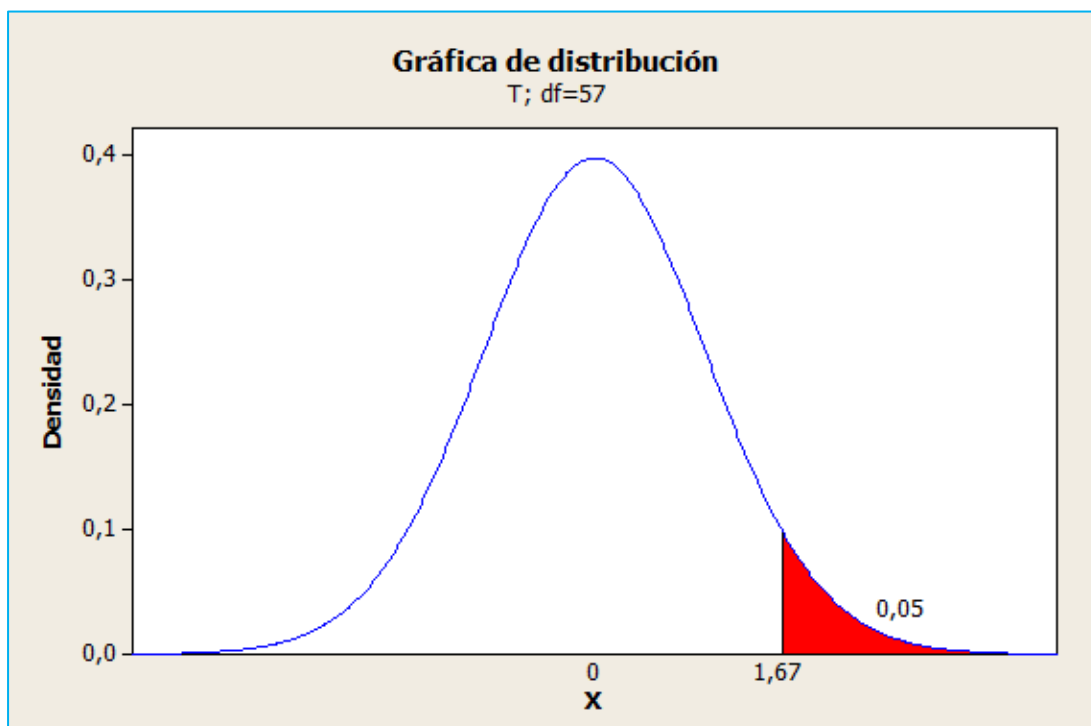
- **Ho:** $\mu_1 \leq \mu_2$

- **Ha:** $\mu_1 > \mu_2$

- **Criterios de decisión**

Figura 40

Indicador 04 - Gráfico para el criterio de decisión



- **Cálculo: Prueba t para medias de las dos muestras**

Tabla 29

Indicador 04 - Prueba t para medias de las dos muestras

	PosPrueba Gc	PosPrueba Ge
Media (\bar{X})	3213.33	1905.00
Desviación Estándar (S)	619.1	557.4
Observaciones (n)	30	30
Diferencia hipotética de las medias	0	
Estadístico t	8.60	
P(T<=t) una cola	0.000	
Valor crítico de t (una cola)	1.67	

- **Decisión estadística**

Debido a que el valor-p = (0.000) < α = (0.05), los resultados proporcionan la evidencia necesaria para rechazar la **hipótesis nula (H₀)**, por lo tanto la **hipótesis alterna (H_a)** se acepta. La prueba resultó ser significativa.

e. Hipótesis específica 05: Satisfacción del cliente

Hi: El uso de las Buenas Prácticas basada en la Metodología MAC **mejora**, la satisfacción del cliente para el proceso del Aseguramiento de la Calidad del Desarrollo de Software en la Empresa CANVIA S.A. (PosPrueba Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

Se realizó una medición sin el uso de las Buenas Prácticas basada en la Metodología MAC (PosPrueba del Gc) y otra con el uso de las Buenas Prácticas basada en la Metodología MAC (PosPrueba del Ge):

Tabla 30

Indicador 05 - Opciones de las respuestas

5	4	3	2	1
Muy Buena	Buena	Regular	Malo	Muy Malo

Tabla 31

Resultados del Indicador 05 - PosPrueba Ge y PosPrueba del Gc

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PosPrueba	M	MM	MB	MM	R	B	B	MM	R	MM	MM	M	R	M	M
Ge	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	MB	MM	R	R	B	M	M	R	MM	R	M	R	MM	M	M
PosPrueba	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	B	MB	MB	MB	B	MB	B	MB	MB	MB	B	B	MB	R	MB
Gc	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	MB	R	B	B	MB	R	MB	B	B	MB	R	MB	MB	B	R

- **Planteamiento de las hipótesis nula y alterna**

- **H₀**: El uso de las Buenas Prácticas basada en la Metodología MAC **no mejora**, la satisfacción del cliente para el proceso del Aseguramiento de la Calidad del Desarrollo de Software en la Empresa CANVIA S.A. (PosPrueba Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).
- **H_a**: El uso de las Buenas Prácticas basada en la Metodología MAC **mejora**, la satisfacción del cliente para el proceso del Aseguramiento de la Calidad del Desarrollo de Software en la Empresa CANVIA S.A. (PosPrueba Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

μ_1 = Media poblacional de la **satisfacción del cliente** en la PosPrueba del Gc.

μ_2 = Media poblacional de la **satisfacción del cliente** en la PosPrueba del Ge.

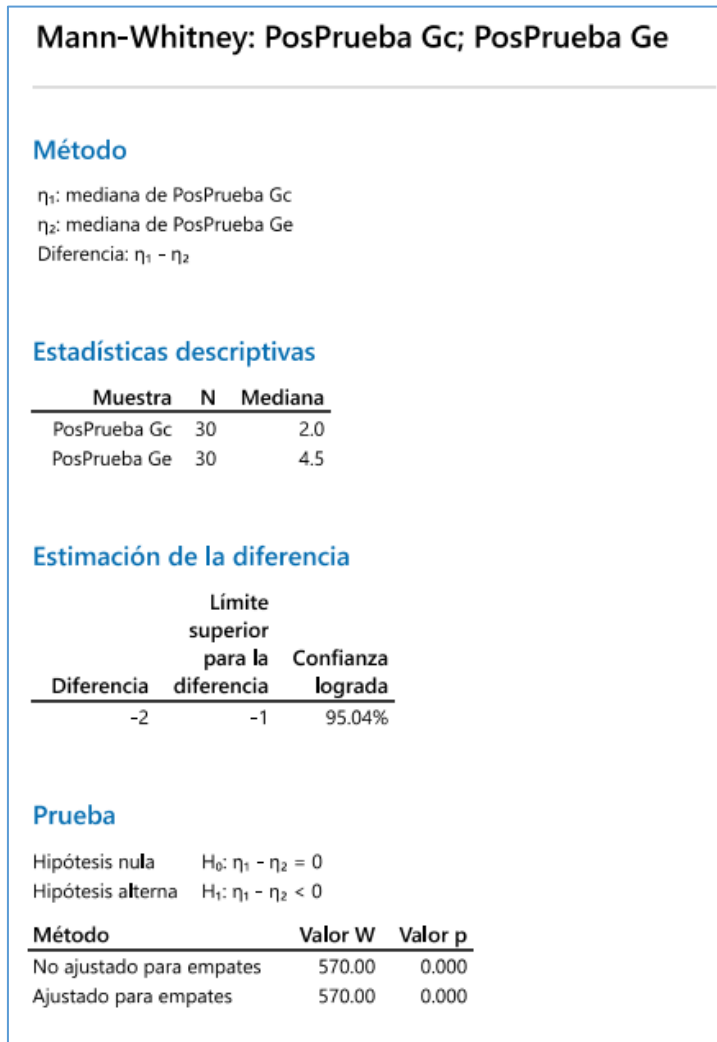
– **H₀**: $\mu_1 \leq \mu_2$

– **H_a**: $\mu_1 > \mu_2$

- **Cálculo: Prueba U de Mann-Whitney para las medias de dos muestras**

Figura 41

Indicador 05: Prueba U de Mann – Whitney para la prueba Gc y Ge



- **Decisión estadística**

Debido a que el valor-p = (0.000) < α = (0.05), los resultados proporcionan la evidencia necesaria para rechazar la **hipótesis nula (H₀)**, por lo tanto la **hipótesis alterna (H_a)** se acepta. La prueba resultó ser significativa.

V. DISCUSIÓN DE RESULTADOS

- a). Para el Indicador 01: Tiempo de entrega para la mejora del proceso del aseguramiento de la calidad del desarrollo de Software en la medición de la *PosPrueba del Gc*, se obtiene sin la aplicación de la nueva metodología MAC un resultado promedio de 36 horas, asimismo en la medición de *PosPrueba del Ge* con la aplicación de la nueva metodología MAC reduce a 21 horas. Los resultados muestran que existe una reducción significativa en el tiempo de entrega para la mejora del proceso del aseguramiento de la calidad del desarrollo de Software en la empresa CANVIA S.A.

De acuerdo a la investigación realizada por Wadood et al. (2022), titulada “Transformaciones ágiles a gran escala para el aseguramiento de la calidad del software: un estudio de caso empírico de Pakistán”, los resultados que se obtiene luego de aplicar una metodología ágil para el área de Calidad, logra reducir un 69% en el tiempo de entrega. Por lo tanto, podemos decir que la utilización de la metodología ágil para el área de calidad permite en ambas investigaciones resultados positivos, basado en la reducción de los tiempos de entrega que se deben considerar en los procesos correspondientes.

- b). Para el Indicador 02: Tiempo para la revisión de errores para la mejora del proceso del aseguramiento de la calidad del desarrollo de Software en la medición de la *PosPrueba del Gc*, se obtuvo sin la aplicación de la nueva metodología MAC un resultado promedio de 27 horas, asimismo en la medición de *PosPrueba del Ge* con la aplicación de la nueva metodología MAC reduce a 16 horas. Los resultados muestran que existe una reducción significativa en el tiempo para la revisión de errores, para la mejora del proceso del aseguramiento de la calidad del desarrollo de Software en la empresa CANVIA S.A.

De acuerdo a la investigación realizada por Carrizo y Alfaro (2018), titulada “Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque

práctico”, los que se obtiene luego de aplicar una metodología para el aseguramiento de la calidad, logra reducir 50% en el tiempo para la revisión de errores.

Por lo tanto, podemos decir que la utilización del enfoque del método de aseguramiento de la calidad permitió en ambas investigaciones resultados positivos, basado en la disminución de los tiempos para la revisión de errores que se deben de considerar en la fase correspondiente para su mejor manejo.

- c). Para el Indicador 03: Números de errores en pruebas para la mejora del proceso del aseguramiento de la calidad del desarrollo de Software en la medición de la *PosPrueba del Gc*, se obtuvo sin la aplicación de la nueva metodología MAC un resultado promedio de 40 errores, asimismo en la medición de *PosPrueba del Ge* con la aplicación de la nueva metodología MAC disminuye a 28 errores. Los resultados muestran que existe una disminución significativa en el número de errores en pruebas para la mejora del proceso del aseguramiento de la calidad del desarrollo de Software en la empresa CANVIA S.A. De acuerdo a la investigación realizada por Jimenez y Jimenez (2020), titulada “Implementación de la metodología Six Sigma para el mejoramiento y optimización de procesos en el equipo aseguramiento de calidad de software de la vertical de Restaurantes de Rappi”, los resultados que se obtiene luego de implementar las buenas prácticas con enfoque de la metodología Six Sigma, se logra disminuir el número de errores en un 60% de los proyectos.

Por lo tanto, podemos decir que la utilización del enfoque para el mejoramiento y optimización de procesos en el equipo aseguramiento de calidad de Software permite en ambas investigaciones resultados positivos, basado en la disminución del número de errores en pruebas que se deben considerar en los procesos correspondientes.

- d). Para el Indicador 04: Costo total del proceso para la mejora del proceso del aseguramiento de la calidad del desarrollo de Software en la medición de la *PosPrueba del Gc*, se obtiene sin la aplicación de la nueva metodología MAC un resultado promedio de S/ 3,213.33 soles, asimismo en la medición de *PosPrueba del Ge* con la aplicación de la nueva metodología MAC disminuye a S/ 1,905.00 soles. Los resultados muestran que existe una disminución significativa en el costo total del proceso en pruebas para la mejora del proceso del aseguramiento de la calidad del desarrollo de Software en la empresa CANVIA S.A.

De acuerdo a la investigación realizada por Saeed et al. (2022), titulada “Optimización de costos de aseguramiento de la calidad del software”, los resultados que se obtiene luego de implementar las buenas prácticas con enfoque de optimización de costo del aseguramiento de la calidad de Software, se logra alinear los procesos teniendo en cuenta el seguimiento del proyecto, disminuyendo los costos a un 57% del presupuesto planificado.

Es importante indicar que para todos los proyectos es primordial brindar un excelente servicio por ello se busca utilizar buenas prácticas, para lograr evitar el incremento de costo de todo el proceso en cada fase, sin perjudicar los tiempos planificados de las actividades.

- e). Para el Indicador 05: Satisfacción del cliente para la mejora del proceso del aseguramiento de la calidad del desarrollo de Software en la medición de la *PosPrueba del Gc*, se obtuvo sin la aplicación de la nueva metodología MAC un resultado promedio de 2.5 (regular), asimismo en la medición de *PosPrueba del Ge* con la aplicación de la nueva metodología MAC mejora a 4.33 (muy buena). Los resultados muestran que existe una mejora significativa en la satisfacción del cliente para la mejora del proceso del aseguramiento de la calidad del desarrollo de Software en la empresa CANVIA S.A.

De acuerdo a la investigación realizada por Tazeen y Waqas (2019), titulado “Ingeniería de requisitos en Agile”, los resultados que se obtiene luego de implementar un marco ágil en aseguramiento de la calidad basado en buenas practicas, logra identificar que el 89% de requerimientos aumentaron la satisfacción del cliente como muy satisfechos y satisfechos tanto en metodologías tradiciones y agiles en el área de calidad.

Cabe resaltar que ambas investigaciones consideraron para el servicio las necesidades del usuario o cliente, implementando las buenas prácticas de calidad que permite para ambos lograr mejorar el grado de satisfacción del usuario.

VI. CONCLUSIONES

- a). Se observa que se ha optimizado el proceso de Aseguramiento de la Calidad a más de lo esperado, el proceso TO-BE tuvo una reducción significativa de tiempos, costos, etc, en consideración a los procesos incluidos a la nueva metodología a diferencia del A-SIS, las cuales no se consideraban.
- b). Se observa que es muy importante capacitar al personal encargado de las revisiones, en este caso a los Analistas de Calidad, quienes son responsables de dar el visto bueno para que un aplicativo sea lanzado a producción.
- c). Se comprueba que, con los nuevos procesos incluidos, en las diferentes fases de la metodología MAC se llegó a tomar mejores decisiones en el proyecto, en cuanto a lo planificado, al levantamiento de información, a las revisiones, entre otras actividades, por parte del Analista de Calidad encargado quien utiliza los artefactos.
- d). Se aprecia que es de gran importancia, tener lineamientos o estándares dentro de un proyecto esto ayuda a que los procesos maduren y se puedan incluir nuevos procesos en un futuro, para que ayuden a definir más cosas según convenga.
- e). Se observa que, en la contrastación de hipótesis, todas las hipótesis alternas fueron aceptadas por lo tanto, los resultados tienen una significancia en cuanto a las evidencias obtenidas de cada uno de los indicadores.
- f). Se comprueba que el haber implementado una nueva metodología de aseguramiento de la calidad generó una retroalimentación de los procesos implementados y existentes, para tener a futuro una mejora continua que ayude a seguir soportando el proceso y sus artefactos proporcionados que se han considerado en cada fase, con esto se desarrolla el uso de las buenas prácticas de los proyectos que se trabajen.

VII. RECOMENDACIONES

- a). Se sugiere tener claro los objetivos para poder implementar una nueva metodología y de conocer bien el proceso actual, para poder proponer los nuevos lineamientos encaso tenga o no tenga dicho proyecto.
- b). Se debe tener reuniones una o dos veces por semana para analizar los resultados en base a las revisiones y poder tomar acciones, y decidir si algo podría faltar incluir a la metodología, ya que este es adaptable y cualquier ítem que se quiera incluir se podrá realizar en cualquier proyecto que se está trabajando.
- c). Se sugiere que el personal del área de calidad, este capacitado para usar la nueva metodología ya que esto contiene fases y en cada una de ellas trabaja con artefacto que apoyan al proceso de la mejora continúa.
- d). Se propone que todos los stakeholders o interesados en el proyecto, se involucren desde el inicio de la implementación de la nueva metodología, ya que serán usados por ellos, para al último tomar decisiones o dar un feedback del proceso actual.
- e). Se sugiere que se institucionalice la metodología en cada proyecto con el cliente debido a que, esto será utilizado de inicio a fin de cada revisión de un aplicativo para que pueda salir a producción sin problemas y este pueda ser utilizado por un público objetivo.
- f). Se propone que el equipo de Calidad incluya sus perspectivas en los artefactos propuestos, en cuanto a las revisiones de las aplicaciones o software, ya que cada uno de ellos como analistas revisa a su experiencia, intuición, capacidad, etc, entonces es recomendable incluirlas por más mínimas que sean.

VIII. REFERENCIAS

- Araníbar, C., y Méndez, L. (2020). *Modelo de aseguramiento de calidad para los procesos de desarrollo de software de las Pymes* [Tesis de Maestría, Universidad Peruana de Ciencias Aplicadas]. Repositorio Institucional UPC. <http://hdl.handle.net/10757/654588>
- Asencios, K. (2019). *Implementación de Buenas Prácticas para la Gestión de Servicios de TI Basado en ITIL V3 para la Unidad de Tecnologías de la Información de la Zona Registral N° VII – Sede Huaraz, 2016* [Tesis de Maestría, Universidad Nacional Santiago Antúnez de Mayolo]. Repositorio Institucional UNASAM. <http://repositorio.unasam.edu.pe/handle/UNASAM/4049>
- Axelos. (2019). *ITIL Foundation*. (4th ed.). The Stationery Office.
- Barrios, A. (2018). *Prospectiva y cambio en las instituciones universitarias y sus relaciones con la sociedad y la empresa*. Director General de los Programas de Postgrado de la UTIC.
- Bayad, J., Pakzad, F., Akoi, S., Aram, A., Shamal, A., Halwest, N., y Anwar, G. (2021). Impact of Service Quality on the Customer Satisfaction: Case study at Online Meeting Platforms. *International journal of Engineering, Business and Management (IJEBM)*, 5(2), 65-77. <https://dx.doi.org/10.22161/ijebm.5.2>
- Bhoushika, D., y Roospesh, K. S. (2019). *Innovations in Smart Cities Applications Volume 5: The Proceedings of the 6th International Conference on Smart City Applications*. (6th ed., vol. 5). Springer.
- Bruchaty, M., y Rastocny, K. (2020). *SOFSEM 2020: Theory and Practice of Computer Science: 46th International Conference on Current Trends in Theory and Practice of*

- Informatics, SOFSEM 2020, Limassol, Cyprus, January 20–24, 2020, Proceedings.*
Springer Nature.
- Carrizo, D., y Alfaro, A. (2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. *Revista chilena de ingeniería (Arica)*, 26(1), 114-129. <http://dx.doi.org/10.4067/S0718-33052018000100114>
- Coupland, M. (2021). *DevOps Adoption Strategies: Principles, Processes, Tools, and Trends.* Packt Publishing.
- Soraluz, A., Valles, M., y Lévano, T. (2021). Desarrollo guiado por comportamiento: buenas prácticas para la calidad de software. *Revista Ingeniería y Desarrollo (Barranquilla)*, 39(1), 190-204. <https://doi.org/10.14482/inde.39.1.005.3>
- Díaz, D., y Silega, N. (2021). Enfoque ontológico para el análisis de estándares de calidad del proceso de software. *Revista Cubana de Ciencias Informáticas*, 15(3), 136-156.
- Dragún, P., Ernst, C., y García, F. (2020). *El futuro del trabajo en el mundo de la industria 4.0.* Oficina de país de la OIT para la Argentina.
- Dudley, J. (2020). *Social Work Evaluation: Enhancing What We Do.* Oxford University Press.
- González, W., Cruz, Y., Parra, A., y Trujillo, Y. (2021). Buenas prácticas en el empleo de CMMI en empresas desarrolladoras de software. *Serie Científica De La Universidad De Las Ciencias Informáticas*, 14(1), 215-225.
- Graham, D., Black, R., y Veenendaal, E. (2020). *Foundations of Software Testing.* Cengage Learning EMEA.
- Hilburn, T., y Towhidnejad, M. (2021). *Software Engineering Practice: A Case Study Approach.* Chapman & Hall.

- ISACA. (2019). *COBIT 2019 Framework: Introduction & Methodology*. ISACA.
- Jiménez de Parga, C. (2021). *UML. Arquitectura de aplicaciones en Java, C++ y Python*. (2da ed.). Ra-Ma.
- Jimenez, A., y Jimenez, A. (2020). Implementación de la metodología Six Sigma para el mejoramiento y optimización de procesos en el equipo aseguramiento de calidad de software de la vertical de Restaurantes de Rappi. *Revista Avenir*, 4(2), 17-24.
- Kaur, D. (2020). *An introduction to System Software*. Alpha Science.
- Koren, I., y Mani, K. (2021). *Fault-Tolerant Systems*. Morgan Kaufmann.
- Larson, E., y Gray, C. (2021). *Project Management: the Managerial Process*. McGraw-Hill Education.
- Lei, S., Nazir, S., y Hussain, A. (2021). Multicriteria Decision Making to Continuous Software. *Scientific Programming*, 2021(1), 1-7. <https://doi.org/10.1155/2021/9953618>
- López, I., y Fernández, L. (2020). *Aplicación de herramientas y análisis de datos para el aseguramiento* [Tesis de Maestría, Universidad de Alcalá]. Repositorio Institucional UAH. <http://hdl.handle.net/10017/43889>
- Mesones, F. (2021). *Modelo de Gestión de incidentes de Tecnologías de la Información basado en marcos de trabajo, estándares y buenas prácticas, para mejorar la calidad de los servicios y agregar valor en las unidades ejecutoras del gobierno regional* [Tesis de Maestría, Universidad Católica Santo Toribio de Mogrovejo]. Repositorio Institucional USAT. <http://hdl.handle.net/20.500.12423/3848>
- Mnkandla, E., y Moyo, S. (2020). A Novel Lightweight Solo Software Development Methodology With Optimum Security Practices. *IEEE Access*, 8(2020), 33735-33747. <https://doi.org/10.1109/ACCESS.2020.2971000>

- Mohacsi, S., y Ramler, R. (2019). *Software Quality The Complexity and Challenges of Software Engineering and Software Quality in the Cloud*. Springer.
- Navarro, H. (2020). *Modelo de Desarrollo e Implementación de Software bajo el marco de Buenas Prácticas del CMMI en el área de Sistemas de Información de la Universidad Autónoma de Bucaramanga* [Tesis de Maestría, Universidad Autónoma de Bucaramanga]. Repositorio Institucional UNAB. <http://hdl.handle.net/20.500.12749/11966>
- Núñez, Y., y González, A. (2020). Gestión de terceros en el desarrollo de software: Propuesta de Metodología. *Enfoque UTE*, 11(2), 71-84. <https://doi.org/10.29019/enfoque.v11n2>
- Pathak, A. (2022). *Petroleum Reservoir Management - Considerations and Practices*. Taylor & Francis Group.
- Paucar, D. V., Acho, P. T., y Peralta, C. S. (2021). Relación de la gestión de riesgos y calidad de software realizados por los profesionales del Colegio de Ingenieros del Perú del Consejo Departamental de Lima. *Interfases*, 14(014), 41-64. <https://doi.org/10.26439/interfases2021.n014.5111>
- Peláez, L., Toro, A., Arias, J., y Rodríguez, D. (2019). Ingeniería de Software: El aseguramiento de la calidad de los requisitos en la industria del software en el Eje Cafetero colombiano. *INGE CUC*, 15(2), 110-122. <https://doi.org/10.17981/ingecuc.15.2.2019.11>
- Pino, F., Rodríguez, M., Piattini, M., Fernández, C., y Delgado, B. (2018). *Modelo de madurez de ingeniería del software Versión 2.0 (MMIS V.2)*. AENOR Internacional.

- Pressman, R., y Maxim, B. (2020). *Becoming an Agile Software Architect: Strategies, practices, and patterns to help architects design continually evolving solutions*. McGraw-Hill Education.
- Project Management Institute. (2021). *Guía de fundamentos para la dirección de proyectos 7ma ed – PMBOK*. (7ma ed.). Project Management Institute.
- Román, M. (2019). *Manual de Buenas Prácticas Manual en Ciclos de Mejora Continua*. Federación Latinoamericana de Colegios de la Compañía de Jesús - FLACSI.
- RV, R. (2021). *Becoming an Agile Software Architect*. Packt Publishing.
- Saeed, S., Jhanjhi, N., Raza, S., y Khan, A. (2022). Cost Optimization of Software Quality Assurance. *Studies in Big Data*, 91(4), 241-255. https://doi.org/10.1007/978-3-030-75855-4_14
- Tazeen, F., y Waqas, M. (2019). Ingeniería de requerimientos en Agile. *International Journal of Education and Management Engineering (IJEME)*, 9(4), 20-33. <https://doi.org/10.5815/ijeme.2019.04.03>
- Wadood, K., Nigar, N., Shahzad, M., Islam, S., Jaleel, A., y Abalo, D. (2022). Large-Scale Agile Transformations for Software Quality Assurance: An Empirical Case Study from Pakistan. *Mathematical Problems in Engineering*, 2022(1), 1-14. <https://doi.org/10.1155/2022/6153744>

IX. ANEXOS

Anexo A. Matriz de consistencia

TÍTULO: IMPLEMENTACIÓN DE BUENAS PRÁCTICAS BASADA EN LA METODOLOGÍA MAC PARA EL ASEGURAMIENTO DE LA CALIDAD DEL DESARROLLO DE SOFTWARE EN LA EMPRESA CANVIA S.A.

PROBLEMA GENERAL	OBJETIVO GENERAL	HIPÓTESIS GENERAL	VARIABLES	INDICADOR (ES)	MÉTODO
¿En qué medida el uso de Buenas Prácticas, basada en la nueva Metodología MAC, mejora el Proceso de Aseguramiento de la Calidad de un proyecto de Desarrollo de Software en la Empresa CANVIA S.A.?	Implementar las Buenas Prácticas, aplicando Metodología MAC, para mejorar el Aseguramiento de la Calidad de un proyecto de Desarrollo de Software en la Empresa CANVIA S.A.	Si se aplica las buenas prácticas, mediante la Metodología MAC, entonces mejora el Aseguramiento de la Calidad de un proyecto de Desarrollo de Software en la Empresa CANVIA S.A.	<u>Variable Independiente:</u> Buenas Prácticas	<ul style="list-style-type: none"> • Presencia - Ausencia 	<p>Tipo de Investigación</p> <ul style="list-style-type: none"> • Básica • Aplicada <p>Nivel de la Investigación</p> <ul style="list-style-type: none"> • Descriptiva • Correlacional-Causal
PROBLEMAS ESPECÍFICOS	OBJETIVO ESPECÍFICOS	HIPÓTESIS ESPECÍFICAS	<u>Variable Dependiente:</u>	<ul style="list-style-type: none"> • Tiempo de entrega. • Tiempo para la revisión de errores. • Número de errores en pruebas. • Costo total del proceso. • Satisfacción del Cliente. 	<p>Universo</p> <p>Todos los procesos de Aseguramiento de la Calidad que brindan servicios de Desarrollo de Software en micro y pequeñas Empresas Latinoamericanas. N= Indeterminado</p> <p>Muestra</p> <p>Los Procesos de Aseguramiento de la Calidad en la empresa CANVIA S.A. n= 30 procesos de Aseguramiento de la Calidad</p>
<ol style="list-style-type: none"> 1. ¿En qué medida el uso de Buenas Prácticas reduce el tiempo de entrega del Proceso de Aseguramiento de la Calidad en la empresa CANVIA aplicando la Metodología MAC? 2. ¿En qué medida el uso de Buenas Prácticas reduce el tiempo para la revisión de errores del Proceso de Aseguramiento de la Calidad en la empresa CANVIA aplicando la Metodología MAC? 3. ¿En qué medida el uso de Buenas Prácticas disminuye el número de errores en pruebas del Proceso de Aseguramiento de la Calidad en la empresa CANVIA aplicando la Metodología MAC? 4. ¿En qué medida el uso de Buenas Prácticas disminuye el costo total del proceso de Aseguramiento de la Calidad en la empresa CANVIA aplicando la Metodología MAC? 5. ¿En qué medida el uso de Buenas Prácticas mejora la satisfacción del cliente del Proceso de Aseguramiento de la Calidad en la empresa CANVIA aplicando la Metodología MAC? 	<ol style="list-style-type: none"> 1. Reducir el tiempo de entrega del Proceso de Aseguramiento de la Calidad en la empresa CANVIA. 2. Reducir el tiempo para revisión de errores del Proceso de Aseguramiento de la Calidad en la empresa CANVIA. 3. Disminuir el número de errores en pruebas del Proceso de Aseguramiento de la Calidad en la empresa CANVIA. 4. Disminuir el costo del Proceso de Aseguramiento de la Calidad en la empresa CANVIA. 5. Mejorar la satisfacción del cliente del Proceso de Aseguramiento de la Calidad en la empresa CANVIA. 	<ol style="list-style-type: none"> 1. Si se aplica las buenas prácticas, mediante la Metodología MAC, entonces reduce el tiempo de entrega. 2. Si se aplica las buenas prácticas, mediante la Metodología MAC, entonces reduce el tiempo para la revisión de errores. 3. Si se aplica las buenas prácticas, mediante la Metodología MAC, entonces disminuye el número de errores en pruebas. 4. Si se aplica las buenas prácticas, mediante la Metodología MAC, entonces disminuye el costo total del Proceso. 5. Si se aplica las buenas prácticas, mediante la Metodología MAC, entonces mejora la satisfacción del cliente. 	Aseguramiento de la Calidad de desarrollo de Software en la Empresa CANVIA S.A.		