



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Dirección General de Estudios de Posgrado
Facultad de Ingeniería de Sistemas e Informática
Unidad de Posgrado

**Modelo de procesos para un servicio de testing de
software y la calidad del producto final en una
empresa de telecomunicaciones**

TESIS

Para optar el Grado Académico de Magíster en Ingeniería de
Sistemas e Informática con mención en Gestión de Tecnología de
Información y Comunicaciones

AUTOR

Arturo Alejandro BARTRA MORE

ASESOR

Dr. Cayo Víctor LEÓN FERNÁNDEZ

Lima, Perú

2025



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Bartra, A. (2025). *Modelo de procesos para un servicio de testing de software y la calidad del producto final en una empresa de telecomunicaciones*. [Tesis de maestría, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Unidad de Posgrado]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Arturo Alejandro Bartra More
Tipo de documento de identidad	DNI
Número de documento de identidad	40233946
URL de ORCID	https://orcid.org/0000-0002-3411-7456
Datos de asesor	
Nombres y apellidos	Cayo Víctor León Fernández
Tipo de documento de identidad	DNI
Número de documento de identidad	07001405
URL de ORCID	https://orcid.org/0000-0003-1704-8214
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Luis Soto Soto
Tipo de documento	DNI
Número de documento de identidad	22409532
Miembro del jurado 1	
Nombres y apellidos	César Augusto Angulo Calderón
Tipo de documento	DNI
Número de documento de identidad	32907109
Miembro del jurado 2	
Nombres y apellidos	Fany Yexenia Sobero Rodríguez
Tipo de documento	DNI
Número de documento de identidad	20120467
Miembro del jurado 3	
Nombres y apellidos	Nilo Eloy Carrasco Ore
Tipo de documento	DNI

Número de documento de identidad	09342780
Datos de investigación	
Línea de investigación	C.0.3.20. Gestión de los Sistemas informáticos y de información
Grupo de investigación	No aplica
Agencia de financiamiento	Sin financiamiento
Ubicación geográfica de la investigación	Edificio: Universidad Nacional Mayor de San Marcos País: Perú Departamento: Lima Provincia: Lima Distrito: Cercado de Lima Calle: Av. Carlos Germán Amezaga #375 Latitud: -12.05611 Longitud: -77.08444
Año o rango de años en que se realizó la investigación	2022 – 2025
URL de disciplinas OCDE	Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
Universidad del Perú, Decana de América
Facultad de Ingeniería de Sistemas e Informática
Vicedecanato de Investigación y Posgrado
Unidad de Posgrado

**ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR EL GRADO ACADÉMICO DE
MAGÍSTER EN INGENIERÍA DE SISTEMAS E INFORMÁTICA CON MENCIÓN EN
GESTIÓN DE TECNOLOGÍA DE INFORMACIÓN Y COMUNICACIONES**

A los quince (15) días del mes de diciembre de 2025, siendo las 11:00 am., se reunieron en el Auditorio Alfredo Celso Alva Bravo, el Jurado de Tesis conformado por los siguientes docentes:

Dr. Luis Soto Soto (Presidente)
Mg. César Augusto Angulo Calderón (Miembro)
Dra. Fany Yexenia Sobero Rodríguez (Miembro)
Mg. Nilo Eloy Carrasco Ore (Miembro)
Dr. Cayo Víctor León Fernández (Miembro Asesor)

Se inició la Sustentación invitando al candidato a Magíster **ARTURO ALEJANDRO BARTRA MORE**, para que realice la exposición oral de la tesis para optar el Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Gestión de Tecnología de Información y Comunicaciones, siendo la Tesis intitulada:

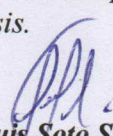
**“MODELO DE PROCESOS PARA UN SERVICIO DE TESTING DE SOFTWARE Y LA
CALIDAD DEL PRODUCTO FINAL EN UNA EMPRESA DE TELECOMUNICACIONES”**

Concluida la exposición, los miembros del Jurado de Tesis procedieron a formular sus preguntas que fueron absueltas por el graduando; acto seguido se procedió a la evaluación correspondiente, habiendo obtenido la siguiente calificación:

Dieciocho (18) Muy Bueno

Por tanto, el presidente del Jurado, de acuerdo con el Reglamento General de Estudios de Posgrado, otorga al Bachiller **ARTURO ALEJANDRO BARTRA MORE** el Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Gestión de Tecnología de Información y Comunicaciones.

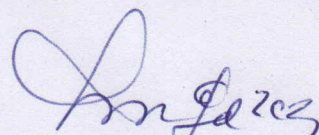
Siendo las 12:10 horas, el presidente del Jurado de Tesis, da por concluido el acto académico de Sustentación de Tesis.


Dr. Luis Soto Soto
(Presidente)


Mg. César Augusto Angulo Calderón
(Miembro)


Dra. Fany Yexenia Sobero Rodríguez
(Miembro)


Mg. Nilo Eloy Carrasco Ore
(Miembro)


Dr. Cayo Víctor León Fernández
(Miembro Asesor)

CERTIFICADO DE SIMILITUD

Yo, Cayo León Fernández, en mi condición de asesor acreditado con Dictamen N° 000384-2025-UPG-VDIP-FISI/UNMSM de la tesis, cuyo título es: **Modelo de Procesos para un Servicio de Testing de Software y la calidad del producto final en una empresa de telecomunicaciones**, presentado por el bachiller: Arturo Alejandro Bartra More, para optar por el grado académico de Magíster en Ingeniería de Sistemas e Informática con mención en Gestión de Tecnología de Información y Comunicaciones.

CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante la plataforma PIDS, el documento evaluado cuenta con el 18% de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional**.

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del grado académico de Magíster en Ingeniería de Sistemas e Informática con mención en Gestión de Tecnología de Información y Comunicaciones.

Lima, 06 de Setiembre del 2025

Firma del Asesor



Nombres y Apellidos del asesor: Cayo León Fernández

DNI 07001405

Dedicatoria

Dedico este trabajo a mi familia, y especialmente a mis pequeños hijos Gonzalo y Gerardo, que son mi fortaleza y los impulsores de mi vida y mejora continua.

AGRADECIMIENTOS

A mi familia entera.

A mis amigos.

A mi red profesional.

A mis compañeros de la vida laboral.

A mi asesor, por su invaluable guía y apoyo.

ÍNDICE

ÍNDICE	3
RESUMEN	6
ABSTRACT	7
CAPÍTULO 1: INTRODUCCIÓN	8
1.1. Situación Problemática	8
1.2. Formulación del Problema.....	10
1.2.1. Problema general	10
1.2.2. Problemas específicos.....	10
1.3. Justificación Teórica.....	11
1.4. Justificación Práctica	12
1.5. Objetivos de la Investigación.....	13
1.5.1. Objetivo General	13
1.5.2. Objetivos Específicos	13
CAPÍTULO 2: MARCO TEÓRICO	14
2.1. Marco Epistemológico de la Investigación	14
2.2. Antecedentes de la Investigación.....	14
2.2.1. Estudios sobre Testing de Aplicaciones.....	15
2.2.2. Estudios sobre gestión de servicios.....	17
2.2.3. Estudios sobre outsourcing de TI.....	18
2.3. Bases Teóricas	20
2.3.1. Gestión del servicio de TI.....	20
2.3.2. Gobierno de Tecnología de Información	24
2.3.3. Mejora de Proceso de Desarrollo de Software	27
2.3.4. Mejora de Proceso de Pruebas de Software	29
2.3.5. Estándares de Calidad.....	30
2.3.5.3. Gestión de proyectos	33
2.3.6. Testing de aplicaciones.....	34
2.3.7. Outsourcing	40
CAPÍTULO 3: METODOLOGÍA	43
3.1. Alcances y limitaciones de la investigación.....	43
3.1.1. Alcances de la Investigación.....	43
3.1.2. Limitaciones de la Investigación	43
3.2. Metodología de Investigación.....	43
3.2.1. Proceso de Investigación	43
3.2.2. Desarrollo del Método	45

3.2.3. Método de Revisión de la Literatura.....	47
CAPÍTULO 4: MODELO DE PROCESOS.....	49
4.1. Diseño del Modelo	49
4.2. Construcción del Modelo.....	51
4.2.1. Gestión de la ejecución de pruebas:	51
4.2.2. Gestión del servicio de testing:	53
4.2.3. Gestión de mejora continua:	54
4.2.4. Gestión de infraestructura y testware:.....	54
4.2.5. Control de la ejecución de pruebas:	54
4.2.6. Gestión de competencias profesionales:	55
4.3. Hipótesis de Modelo Propuesto	56
4.3.1. Hipótesis General	56
4.3.2. Hipótesis Específicas	56
4.4. Identificación de Variables	57
4.4.1. Variable Dependiente	57
4.4.2. Variable Independiente.....	57
4.5. Operacionalización de Variables	58
4.5.1. Operacionalización de la variable modelo de procesos para un servicio de testing de software.....	58
4.5.2. Operacionalización de la variable Calidad del producto final.....	59
CAPÍTULO 5: ESTUDIO DE CASO	60
5.1. Diseño de la Investigación (Protocolo).....	60
5.2. Unidad de Análisis	61
5.3. Población de Estudio	61
5.4. Tamaño de la Muestra	62
5.5. Selección de la Muestra.....	62
5.6. Técnicas de Recolección de Datos.....	62
5.7. Análisis e Interpretación de la Información	63
5.8. Resultados	64
CAPÍTULO 6: RESULTADOS.....	65
6.1. Resultados descriptivos	65
6.1.1. Resultados descriptivos de la variable modelo de procesos para un servicio de testing	65
6.1.2. Resultados descriptivos de la variable calidad del producto final	83
6.2. Resultados inferenciales	96
6.2.1. Selección de prueba de normalidad	96
6.2.2. Regresión logística ordinal	97

6.2.3. Decisión estadística	97
6.2.4. Prueba de hipótesis general.....	97
6.2.5. Prueba de hipótesis específica 1	99
6.2.6. Prueba de hipótesis específica 2	100
6.2.7. Prueba de hipótesis específica 3	101
6.2.8. Prueba de hipótesis específica 4	102
6.2.9. Prueba de hipótesis específica 5	103
6.2.10. Prueba de hipótesis específica 6	104
6.3. Discusiones.....	105
CONCLUSIONES	107
RECOMENDACIONES	109
REFERENCIAS BIBLIOGRÁFICAS	111
ANEXOS	117
ANEXO 1 - Cuestionario “Modelo de Procesos para un Servicio de Testing”	117
ANEXO 2 - Cuestionario “Calidad del Producto Final”	119
ANEXO 3 - Flujo y caracterización del Proceso de Gestión de la Ejecución de Pruebas	121

RESUMEN

La presente investigación tuvo como objetivo establecer cómo influye un modelo de procesos para un servicio de testing de software en la calidad del producto final en una empresa de telecomunicaciones. La justificación radica en la ausencia de un modelo unificador que estandarice la gestión del testing bajo outsourcing, pese a su creciente relevancia en el aseguramiento de calidad de software crítico en diversos sectores. Para ello, se diseñó y aplicó un método basado en el enfoque de la ciencia del diseño, que combinó revisión de literatura, elaboración de un modelo sustentado en estándares internacionales como CMMI, TMMI e ITIL, y su validación mediante un estudio de caso en una consultora peruana especializada en servicios de testing. Entre los resultados, se evidenció una mejora significativa en la eficiencia de tiempos de entrega y una reducción en el número y criticidad de las incidencias reportadas, así como un aumento en la satisfacción de los clientes. Se concluye que la adopción del modelo propuesto contribuye de manera positiva al desempeño de los servicios de testing, facilitando procesos más controlados, repetibles y alineados con las mejores prácticas internacionales.

Palabras clave: testing de software, calidad, outsourcing, modelo de procesos, CMMI.

ABSTRACT

This research aimed to establish how a process model for a software testing service influences the quality of the final product of a telecommunications company. The justification lies in the absence of a unifying model that standardizes the management of testing under outsourcing, despite its growing relevance in ensuring the quality of critical software across various sectors. To this end, a method based on the Design Science approach was developed, combining literature review, the creation of a model supported by international standards such as CMMI, TMIMI, and ITIL, and its validation through a case study in a Peruvian consultancy specialized in testing services. Among the results, there was a significant improvement in delivery time efficiency and a reduction in the number and severity of reported incidents, as well as an increase in customer satisfaction. It is concluded that the adoption of the proposed model contributes positively to the performance of testing services, enabling more controlled, repeatable processes aligned with international best practices.

Keywords: software testing, quality, outsourcing, process model, CMMI.

CAPÍTULO 1: INTRODUCCIÓN

1.1. Situación Problemática

El software es un elemento fundamental en la vida de las organizaciones. En los últimos años el software está a cargo de tareas críticas en instituciones de todos los sectores como: salud, telecomunicaciones, banca, seguros, energía, minería, etc.

El ciclo vida de software, así como las diversas actividades que lo componen pueden ser realizados tanto dentro de la misma organización que usará el software como fuera de esta. Diversos autores y organizaciones (Pressman & Maxim, 2020; Sommerville, 2016; ISACA, 2022) definen este ciclo, coincidiendo con las etapas de Análisis, Diseño, Desarrollo y Pruebas, (esta última conocida también como la fase de Testing)

La fase de Testing (Pressman & Maxim, 2020; Sommerville, 2016; SWEBOK (IEEE Computer Society [IEEE-CS], 2024) es la fase que verifica y valida diversos aspectos del software como: el cumplimiento de los requisitos funcionales, de usabilidad, de rendimiento, entre otros (NTP-ISO-IEC 12207, 2006; ISO/IEC 12207, 2016). Es en esta fase que se verifica si el producto se construyó con la calidad esperada, lo que implica cumplir con los requerimientos solicitados y esperados por el usuario o cliente.

El outsourcing, es la actividad mediante la cual una empresa u organización delega una o varias actividades propias a un tercero, teniendo en cuenta aspectos como costes, riesgos, seguridad y capacidades del tercero (Hanafizadeh y Zareravasan, 2020; Barthelemy, 2003). Este escenario es beneficioso cuando está enfocado en áreas del negocio en los que no se tiene la suficiente fortaleza o experiencia para efectuar una gestión exitosa (Francheschini, 2003). El testing de software es una actividad que por su naturaleza encaja en este escenario; por tal motivo es la fase del proceso de desarrollo de software que es más propensa y recomendable de externalizar en la actualidad.

Diversos estudios realizados referentes al testing de software han sido desarrollados (Baresi, L., 2006), muchos de ellos de orientación técnica, en búsqueda de las mejores estrategias para su ejecución e inclusive su automatización (Catelani et al., 2011). Pocos han

orientado sus investigaciones para el modelado del testing de software como un servicio externo a la organización.

Los resultados obtenidos de estos estudios, entre otros, han definido nuevas estrategias de pruebas (Kruse et al., 2010) o algoritmos para hallar los mejores casos de prueba; y hasta se han desarrollado modelos, como el modelo de madurez de capacidades del Testing como TMM y TMMi (Illinois Institute of Technology. (1998); TMMi Foundation, (2022) para describir las etapas por las que atraviesa una organización de pruebas y las practicas relacionadas con cada fase. Pero ninguno de ellos ha tratado acerca de la manera en que se debe organizar este tipo de entidades cuando brinda servicios de testing de software a clientes externos y bajo demanda, es decir, cuando se rige bajo la práctica del outsourcing.

La carencia de un modelo estándar que reúna las características apropiadas para la implementación eficiente de un servicio de testing de software hace de la creación de un modelo de procesos para este tipo de servicios una necesidad en el campo de la calidad del software (Pressman & Maxim, 2020). El beneficio de esta implementación incurriría en toda aquella organización cuyas actividades se rigen plenamente por el uso y buen funcionamiento de sus aplicaciones.

En el Perú, debido al crecimiento económico sostenido de los últimos años, las empresas dedican un presupuesto mayor orientado a brindar calidad al software que producen (interna o externamente), o compran. Debido a esto las actividades de aseguramiento de la calidad del producto software y sobre todo la actividad de testing de software se ha convertido en uno de los servicios más solicitados por las organizaciones (Dávila et al. 2017).

Por ello se estima que la creación de este modelo asegurará la puesta en producción de software de calidad, en los tiempos estimados y bajo el presupuesto establecido. Asimismo, se reducirán costos en la fase de mantenimiento, que es la etapa de mayor coste del ciclo de vida del software (Chen et al., 2009), incrementará la calidad del software desarrollado y hará efectiva su puesta en producción.

La presente investigación define un modelo de procesos para un servicio de testing de software con el objetivo de entregar productos de calidad al término del ciclo de vida de desarrollo del software. Se definen un modelo de procesos que engloba todo el ciclo vida de un Servicio de Testing de Aplicaciones.

Los modelos constitutivos del modelo propuesto son: un modelo de pruebas (basado en CMMI, TMMI y en la revisión de la literatura) que enmarca las actividades de las pruebas de las aplicaciones, definiendo los roles actividades y flujos de trabajo; un modelo de gestión de servicio (basado en los principios de gestión de servicios de ITIL 4) que se ocupa de la demanda y capacidades de los equipos para asegurar la continuidad de las operaciones; un modelo para la planificación y control de ejecución de las pruebas (basado en los modelos de gestión de proyectos planteados en PMBOK, (Project Management Institute [PMI], 2021) que gestiona la definición de métricas e indicadores para la evaluación de los proyectos de Testing y la medición de los acuerdos de niveles de servicio.

Marcos de trabajo y estándares como el CMMI, TMMI, ITIL, PMBOK, ISO 2000, ISO 9126, ISO 25000 fueron analizados y sus fundamentos y principios son la base para la elaboración del modelo.

La puesta en marcha de un estudio de caso verifica el aporte de la implementación del modelo de procesos en una empresa de software del rubro de calidad y dedicada al testing de software cuya forma de trabajo y estructura orgánica difiere del modelo propuesto en esta investigación. Se utilizan, para verificar el aporte del modelo, los registros históricos de calidad y mediciones previas de la organización para contrastar los resultados obtenidos de la implantación del modelo propuesto.

Se finaliza el trabajo de investigación con las conclusiones y recomendaciones, producto de la observación y análisis de los resultados obtenidos. Así también se trazan rutas para futuras investigaciones.

1.2. Formulación del Problema

1.2.1. Problema general

¿Cómo influye un modelo de procesos para un servicio de testing de software en la calidad del producto final en una empresa de telecomunicaciones?

1.2.2. Problemas específicos

- ¿Cómo influye la ejecución de pruebas en la calidad del producto final en una empresa de telecomunicaciones?

- ¿Cómo influye el servicio de testing en la calidad del producto final en una empresa de telecomunicaciones?
- ¿Cómo influye la mejora continua en la calidad del producto final en una empresa de telecomunicaciones?
- ¿Cómo influye la infraestructura y testware en la calidad del producto final en una empresa de telecomunicaciones?
- ¿Cómo influye el control de ejecución de pruebas en la calidad del producto final en una empresa de telecomunicaciones?
- ¿Cómo influye las competencias profesionales en la calidad del producto final en una empresa de telecomunicaciones?

1.3. Justificación Teórica

Existen en la actualidad estándares o marcos de trabajo, que conforman las denominadas buenas prácticas: CMMI V2.2 (ISACA, 2022), TMMI 1.3 (TMMi Foundation, 2022), PMBOK (Project Management Institute, 2021), ITIL 4 (AXELOS, 2019), las cuales han sido ejercitadas y cuentan con el consenso de la comunidad profesional y científica, tanto por los resultados obtenidos, tanto por su condición de repetitividad y de funcionalidad.

Son numerosas las empresas consultoras de tecnología de información, entre ellas las dedicadas al rubro de testing de software, que subutilizan estos modelos o no han considerado su uso o adopción, por varios factores (Cater-Steel, 2006): el desconocimiento de las normas o frameworks, no tener asignada la función a ningún miembro u oficina de la organización, la inexistencia de un área de control de la calidad, la incertidumbre de cual práctica es mejor o como empezar a abordarlas, y en el extremo, no considerarlas importantes o necesarias para una mejor entrega de servicios TI.

El presente trabajo de investigación aborda los principales componentes, áreas de proceso y elementos de los marcos de trabajo mencionados en la construcción de un modelo de procesos para un servicio de testing de aplicaciones. El producto obtenido y su novedad de implementación justifican su estudio y desarrollo.

1.4. Justificación Práctica

En una investigación llevada a cabo por el AMR Research, acerca del outsourcing del Testing, comenta que este ahorra a las empresas hasta un 75% del costo de hacerlo ellas mismas y además asegura la calidad del producto. La misma investigación revela que el costo de hallar y reparar defectos toma aproximadamente el 50% del costo total del software producido (Deloitte, 2008).

Diversos estudios recientes coinciden en que el mercado global del outsourcing de testing de software experimenta un crecimiento sostenido y acelerado impulsado por la digitalización, la adopción de metodologías ágiles y DevOps, y el uso de automatización e inteligencia artificial en los procesos de aseguramiento de la calidad. Según Market Research Future (2025), se estima que este mercado pasará de USD 30.75 mil millones en 2025 a USD 99.37 mil millones en 2034, con una tasa de crecimiento anual compuesta (CAGR) del 14.2 %.

The Business Research Company reporta que el mercado creció de USD 52.92 mil millones en 2024 a USD 61.13 mil millones en 2025, con un CAGR del 15.5 %, mientras que Global Growth Insights proyecta un crecimiento anual cercano al 13.9 % entre 2025 y 2033. Estas cifras confirman que la contratación de servicios externos de testing es cada vez más prioritaria para las organizaciones, dado que contribuye a la mejora de la calidad, la estandarización de procesos y la reducción del time to market de los productos de software.

Con este escenario y con el potencial de mercado en la región, la investigación tiene por objeto construir un modelo de procesos para un servicio de testing de software que permita la entrega oportuna y de calidad de los servicios brindados, mediante la utilización de marcos de trabajo y buenas prácticas establecidas.

Temas tales como los componentes a utilizar de los marcos de trabajo, las características o procesos a implantar de estos, la simultaneidad de adopción y otros con el fin de crear un modelo de procesos para el servicio de testing de aplicaciones serán tratados.

1.5. Objetivos de la Investigación

1.5.1. Objetivo General

Establecer cómo influye un modelo de procesos para un servicio de testing de software en la calidad del producto final en una empresa de telecomunicaciones.

1.5.2. Objetivos Específicos

- Establecer cómo influye la ejecución de pruebas en la calidad del producto final en una empresa de telecomunicaciones.
- Establecer cómo influye el servicio de testing en la calidad del producto final en una empresa de telecomunicaciones.
- Establecer cómo influye la mejora continua en la calidad del producto final en una empresa de telecomunicaciones.
- Establecer cómo influye la infraestructura y testware en la calidad del producto final en una empresa de telecomunicaciones.
- Establecer cómo influye el control de ejecución de pruebas en la calidad del producto final en una empresa de telecomunicaciones.
- Establecer cómo influye las competencias profesionales en la calidad del producto final en una empresa de telecomunicaciones.

CAPÍTULO 2: MARCO TEÓRICO

2.1. Marco Epistemológico de la Investigación

La industria de TI necesita optimizar la gestión de calidad y la fiabilidad en los negocios, así como dar respuesta a un número creciente de exigencias contractuales y regulatorias; por eso se incrementa la implementación de mejores prácticas de TI.

Cada compañía necesita adaptar el uso de prácticas y estándares a sus propias necesidades. En esta línea, los estándares, las mejores prácticas y los marcos de trabajo juegan un rol significativo en el rendimiento organizacional.

COBIT (ISACA, 2019) contribuye a determinar lo que se debe hacer, al introducir el concepto de gobierno de las tecnologías de la información y priorizar su alineación con los objetivos empresariales. ITIL 4 (AXELOS, 2019) brinda un método para gestionar los servicios, mientras que ISO/IEC 9000 presenta la idea de un sistema de gestión de calidad (SGC). Por su parte, ISO/IEC 9000-3 es la implementación del SGC en el desarrollo relacionado con la ingeniería del software; ISO/IEC 20000 (ISO/IEC 20000-1, 2005), que fomenta la adopción de un modelo integrado de procesos para optimizar la eficacia en el suministro de servicios tecnológicos; ISO/IEC 25000, enfocado en la calidad del software; CMMI (ISACA, 2022) que proporciona un modelo de medición tanto de madurez como de capacidad y las mejores prácticas para el desarrollo del software y la prestación de servicios tecnológicos; y TMMI (TMMi Foundation, 2022), creado por un grupo independiente y basado en CMMI, orientado exclusivamente al proceso de pruebas de software.

2.2. Antecedentes de la Investigación

Abordaremos aquí trabajos previos relacionados al tema de investigación de la tesis, en los campos de Testing de Software, mejora de proceso, calidad de sistemas de información, entrega de servicios, que han tratado de resolver el problema identificado, y que influyen en la calidad, eficiencia y capacidad de los servicios entregados al cliente. Se analizarán estudios sobre la adopción de los marcos de trabajo utilizados, con el objetivo de mejorar sus procesos y gestionar sus servicios.

2.2.1. Estudios sobre Testing de Aplicaciones

Se sabe que la validación, la verificación y las pruebas son procesos costosos para crear y mantener un producto de software. El precio puede llegar a ser entre el 30% y el 50% del coste total de un producto, sin incluir los gastos por inconformidad que se producen en la organización que emplea dicho producto, según diversas investigaciones (Deloitte, 2008).

Sin embargo, las demoras en los proyectos y la inesperada evolución de estos hacen que con frecuencia el esfuerzo planificado para realizar las pruebas disminuya, lo cual resulta en la liberación de productos con un número de fallas más alto de lo razonable. Además de razones coyunturales como las que se han expuesto, no siempre se valora lo esencial que es invertir en una planificación, gestión y ejecución apropiadas de las pruebas, tanto para los desarrolladores como para los gerentes y los usuarios finales.

Investigaciones en el área de testing de aplicaciones, entrega de servicios de TI mediante el escenario del outsourcing, adopción de estándares y marcos de trabajo han seguido varios caminos.

La intención del Testing es aumentar la certeza sobre el correcto desempeño del software. El SWEBOK (IEEE Computer Society [IEEE-CS], 2024) indica que las pruebas de software implican la validación dinámica del comportamiento de un programa en un número limitado de casos de prueba, elegidos apropiadamente desde el dominio de las ejecuciones, que suele ser infinito, en comparación con el comportamiento especificado esperado.

Los primeros estudios en este campo se preocuparon por el desarrollo de algoritmos para la elaboración y ejecución de casos de prueba adecuados con la finalidad de detectar la mayor cantidad posible de errores con el menor esfuerzo posible (Kaner et al., 1999). Otros trabajos tomaron una orientación más técnica tratando de crear oráculos de prueba y buscar la manera de automatizar las pruebas. Otros autores como (Kshirasagar, et al., 2008), fueron más allá, yendo de la vista de ejecución hacia la organización en sí de los equipos que ejecutan las pruebas.

Se desarrollaron modelos para la ejecución del proceso de Testing como el modelo V (Baresi, L., 2006) que especifica, para cada fase del proceso de desarrollo, una fase paralela que la valide y verifique los entregables resultantes. (ver figura 9). Este modelo especifica la

ejecución de determinado tipo de pruebas en cada una de las etapas del ciclo de desarrollo (Myers, 2004)

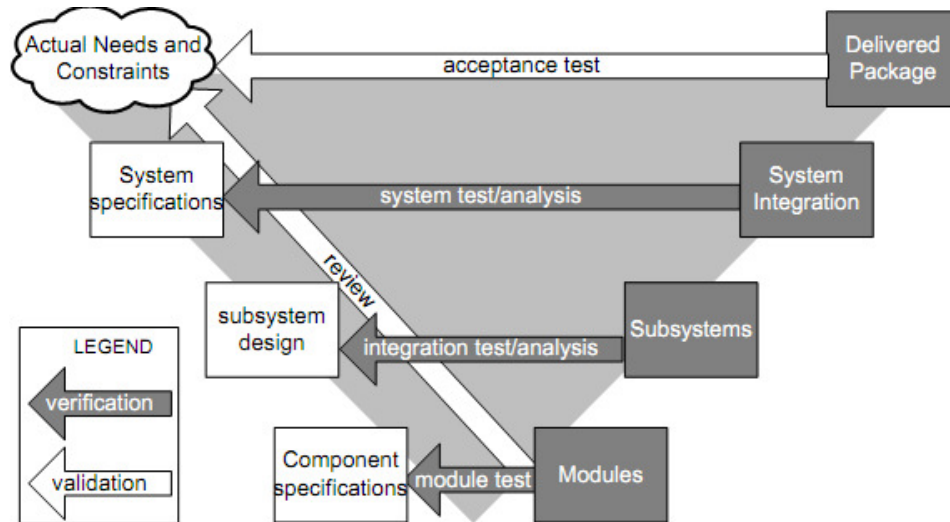


Figura 1. Modelo V de Pruebas Software. Tomado de Baresi (2006)

De la misma forma la fase de Testing es delineada en las diferentes metodologías de desarrollo como el Proceso Unificado de Desarrollo, RUP (Rational Unified Process). En esta metodología el Testing va en paralelo a todo el proceso, desde la captura de requerimientos hasta la fase de Testing propiamente dicha.

También, investigadores reunidos alrededor del Illinois Institute of Technology desarrollaron un símil del modelo de madurez de software, pero orientado a las pruebas software (TMMi Foundation, 2022). Este modelo es el TMMI, el cual indica los estadios por los que debe pasar una organización de Pruebas y las prácticas y artefactos relacionados según el nivel de madurez alcanzado.

Otros estudios como el de Gillenson (2011) convienen en la incorporación de las actividades de Testing, entre ellas las de verificación y validación durante todo el ciclo de desarrollo, haciendo así al tester parte del equipo, con la labor de aportar calidad, por el conocimiento adquirido sobre el producto durante todo el desarrollo del proyecto software. Y otros, como el de Onita, (2007) que buscan alinear los ciclos de Testing y desarrollo utilizando conocidos modelos de alineamiento estratégico como el desarrollado por Venkatraman (1990)

Una perspectiva diferente de la fase de Testing la plantea Yang (2008). Su preocupación por la conceptualización y el papel del Testing en los ciclos de desarrollo, limitándolo a solo observar el código fue su mayor preocupación. Esto aunado con el paradigma de los Servicios de TI, es que considera al Testing de Software como un servicio, para el uso interno, dentro del ciclo de desarrollo, como externo, en la validación del producto final con el usuario, y al equipo de Testing como los proveedores. Argumenta además que el Testing en este nuevo paradigma debe abordar temas como: la usabilidad del software, la satisfacción del usuario, el mapeo de requerimientos a las necesidades de negocio, la coherencia a los sistemas de información. En este estudio adapta el uso de una herramienta de medición de servicios, SERVQUAL, al contexto del Testing, dándole un enfoque de orientación de servicios y a las ISO de calidad de software como el ISO 9126.

2.2.2. Estudios sobre gestión de servicios

La gestión de servicios ha sido una extensión o continuidad del desarrollo a lo largo del tiempo; sin embargo, en años recientes se está modificando esta realidad. Según investigaciones realizadas por Gartner Group, entre el 70 y el 80 % de los gastos durante la vida útil de los sistemas de información corresponden a la etapa de explotación (Casson, 2005). Esta circunstancia se constata al observar situaciones en las que el 60% del tiempo de los desarrolladores se emplea en labores de mantenimiento, o cuando las tareas cotidianas del personal de TI parecen enfocarse en actividades de gestión.

Los modelos de gestión de servicios utilizan ITIL 4 (AXELOS, 2019) como fundamento para llevar a cabo un desarrollo que amplíe el modelo, estableciendo procesos que no están descritos en ITIL.

La gestión de los servicios se fundamenta hoy en día en procesos, que son una serie de actividades lógicas unidas con el fin de alcanzar un objetivo específico (Cappiello et al., 2009). Los beneficios de los procesos son:

- Se especifican las metas y el modo de alcanzarlas en un proceso.
- Para cada proceso, determinamos un input y el output requerido para llegar a nuestras metas, así como lo que otros procesos requieren de nosotros para lograr las suyas. En un proceso se detallan los objetivos y cómo se van a lograr.

- Podemos monitorear esos procesos uno por uno cuando describimos una organización completa en diversos procesos. Se supervisa por secciones en vez de hacerlo de manera integral.
- Hacemos que las personas sean responsables de su eficacia, eficiencia y del resultado de su proceso. Por eso, no solo supervisamos sino también regulamos nuestra organización.
- Si no hemos logrado el objetivo, podemos debatir cómo optimizar las actividades en un proceso.
- Para prevenir conflictos de interés, se distribuyen las responsabilidades.

Por otro lado, tenemos al Modelo de Madurez de Capacidad del Servicio de TI (CMM IT Service) que es un modelo de madurez destinado a organizaciones que proveen servicios de TI, persigue un doble objetivo: evaluar la madurez actual de servicios de TI y proporcionar la dirección para la mejora de los procesos de servicio de TI de la organización. Los propietarios de los derechos de autor son Vrije Universiteit de Amsterdam y CIBIT.

2.2.3. Estudios sobre outsourcing de TI

Trabajos relacionados al outsourcing de procesos de TI tratan sobre lo conveniente de llevar a este modelo de trabajo áreas de proceso de la compañía en la cual no se tenga el dominio necesario para llevar a cabo o cuya delegación a una empresa externa abarate los costos de su ejecución (Barthelemy, 2003), aquí refiere en su estudio denominado “Los siete pecados capitales del outsourcing” entre otras cosas, qué debería y qué no ser llevado al modelo del outsourcing.

Entre estas actividades, refiriéndose al proceso de desarrollo de software, la fase de testing es la candidata ideal para tercerizar (Hanafizadeh y Zareravasan, 2020; Barthelemy, 2003), por abaratar los costos y aportar a la calidad del producto final al ser entregadas, en el papel, a proveedores de servicios eficientes y especializados. Myers (Myers, 2004) opina que la organización de las pruebas software debe ser removida de la organización y llevada fuera de esta en términos organizacionales; es decir, que deje de pertenecer por completo a la compañía (Khan, 2010).

Otros estudios analizan el impacto del rol del componente humano de TI en la organización y lo catalogan como fundamental en la transferencia de conocimiento entre la

empresa que contrata y la empresa contratada (Park, 2011). Este estudio, como el de Barthelemy (2003) y Hanafizadeh y Zareravasan (2020), consideran muy importante el factor humano y la gestión del conocimiento para mantener el control de la gestión de un servicio en un escenario de outsourcing; lo que implica que la firma cliente debería identificar y conservar al personal que tiene las capacidades y conocimientos para responder de manera efectiva ante cambios tecnológicos y organizacionales en el despliegue del servicio. Y Hodosi (2024) encuentra críticos para el éxito de este tipo de contrataciones el contrato, compromiso, comunicación, colaboración y confianza en el proveedor contratado asegurando, junto al componente técnico, la buena ejecución del outsourcing.

Refiriéndose específicamente sobre el tema del outsourcing del testing de aplicaciones Poston, (2007) en su estudio sobre la gestión de este tipo de servicio efectúa un estudio de caso a catorce (14) stakeholders de una compañía (catalogada dentro de las Fortune 500 companies) en el que identifica los principales intereses de estos entre los que se puede mencionar: El contrato con los proveedores, las estrategias mutuas para abordar el outsourcing del testing, Las expectativas del cliente acerca del rendimiento del servicio, la gestión del servicio del proveedor, el manejo de problemas con el proveedor, la infraestructura tecnológica del proveedor, los niveles de servicio acordados con el proveedor, las estrategias de pruebas a adoptar. Se puede observar, en el aspecto de gestión del servicio, que el control sobre la ejecución del ciclo completo, es decir, la gestión de todo como un servicio es una de las principales preocupaciones de los responsables, de ambos lados, de un contrato de outsourcing,

2.3. Bases Teóricas

2.3.1. Gestión del servicio de TI

Según Hochstein (2005), la gestión de servicios de TI se fundamenta en una perspectiva organizativa que tiene como bases el manejo de procesos y la gestión de calidad a lo largo del ciclo vital de los servicios.

Se trata de una disciplina orientada a procesos que busca alinear los servicios de TI ofrecidos con lo que las compañías requieren, como señalan Hochstein et al. (2005), destacando los beneficios que puede recibir el cliente final.

Normalmente, la administración de servicios de TI implica el empleo de outsourcing, insourcing y servicios compartidos. Para que estas prácticas tengan éxito, es fundamental conservar un extenso fundamento de conocimiento en la organización (Park, Im, Kim, 2011).

Los propósitos de la gestión de servicios TI son:

- Brindar una administración apropiada de la calidad.
- Mejorar la eficacia de las operaciones.
- Alinear la infraestructura de TI y los procedimientos comerciales.
- Disminuir los peligros vinculados a los servicios de TI.
- Crear negocios.

Los beneficios y barreras que la gestión de servicios de TI según Serrano (2021), como un mejor alineamiento y percepción de la calidad de los servicios recibidos; así como vencer la resistencia al cambio y adaptación a la cultura de las organizaciones resulta clave para una adopción exitosa de estos marcos de gestión.

2.3.1.1. ITIL v4

En los últimos años de 1980 el UK's Central Computer and Telecommunications Agency (CCTA) desarrollo el framework de ITIL para reducir costos y mejorar la entrega de servicios TI, ITIL 4 (AXELOS, 2019). El enfoque de ITIL es proveer un comprehensivo y cohesivo conjunto de plantillas y mejores prácticas para los procesos operacionales principales

de TI. Sus principales áreas de aplicación son el soporte y la entrega del servicio TI, que define procesos clave que las organizaciones de TI deben poseer para la provisión de servicios de calidad para sus usuarios.

El área de soporte del servicio tiene que ver con la provisión de servicios TI asociados a los procesos de mantenimiento y soporte del día a día (Bittinger, 2005). Dentro de esta área se encuentra la función de servicio de escritorio, la cual fue diseñada para ser el principal contacto entre el usuario y la organización de TI. El segmento de entrega de servicio cubre los procesos requeridos para la planificación y entrega de servicios TI de calidad y vela a largo plazo por los procesos asociados con la mejora de la calidad del servicio TI entregado.

ITIL ha sido muy adoptado en Europa, especialmente en el sector gobierno, y su implementación está creciendo en Australia y Norteamérica. EXIN International, es la empresa líder en entrenamiento y certificación ITIL.

El framework ITIL es gestionado en la actualidad por el UK Office of Government Commerce y sus procesos de mejores prácticas están soportados por el British Standards Institute's BS 15000, para la gestión de servicios TI.

ITIL es un marco de gestión de servicios de TI que define las mejores prácticas de forma integrada y coherente para gestionar el ciclo de vida de los servicios, adaptándose a las necesidades específicas de cada organización. En su versión más reciente, ITIL 4, introduce el Sistema de Valor del Servicio (Service Value System), que integra principios guía, gobernanza, prácticas y mejora continua orientadas a la creación de valor conjunto entre proveedores y clientes (AXELOS, 2019).

Las organizaciones dependen cada vez más de las tecnologías de la información y este es un elemento esencial para alcanzar sus metas corporativas y satisfacer las necesidades del negocio. Con frecuencia, un departamento de TI no toma en cuenta los objetivos de la empresa como si fueran los suyos propios, sino que se ve a sí mismo simplemente como un proveedor. ITIL sostiene que la TI tiene que ser percibida como una parte esencial de la compañía, alineada de cerca con los objetivos corporativos. La tecnología de la información (TI) tiene que colaborar con el negocio principal, convertirse en un socio comercial y establecer esa tan crucial relación comercial. Los clientes exigen hoy en día mucho más de los servicios de TI y se está buscando

ofrecer servicios, no productos como hace unos años. Esto supone un desafío para la organización de TI.

2.3.1.2. *SLM*

La filosofía de gestión de los EE. UU. llamada service level management (SLM) o gestión del nivel de servicio, surgió a mediados de los años 90's, y tiene un enfoque de definición y entrega operacional de servicios de TI (Cater-Steel, 2006). Parte del ímpetu de su surgimiento fue el incremento del outsourcing y la necesidad de celebrar contratos con proveedores de servicios externos. Una definición explica: "SLM es un proceso que permite a los proveedores de centro de datos la entrega de servicios acorde a un SLA (service level agreement, o acuerdo de nivel de servicio) que describe el rendimiento esperado de este proveedor de servicios". En este caso el SLM no es solo un subproceso de un gran framework, sino que permite definir ampliamente la entrega del servicio y su soporte.

SLM incluye los procesos necesarios para asegurar que los acuerdos de servicios sean cumplidos. Estos procesos incluyen: monitoreo de sistemas, redes y aplicaciones; el cálculo de los actuales niveles de servicio comparándolos con los acuerdos de nivel de servicios contratados; y el reporte de rendimiento del servicio.

2.3.1.3. *ISO 20000*

El 14 de diciembre de 2005, las entidades IEC (International Electrotechnical Commission) e ISO (International Organization for Standardization) publicaron la serie ISO/IEC 20000, que se encarga de la gestión de servicios TI (Tecnologías de la Información). Este estándar ha sido reconocido a nivel internacional. La serie 20000 es el resultado de la implementación de la serie BS 15000, creada por la British Standards Institution (BSI), una organización británica dedicada a establecer normas.

Este estándar establece las pautas para una gestión de servicios tecnológicos de calidad (ver figura 2) y fomenta la implementación de un modelo integrado de procesos con el fin de aumentar la efectividad en la entrega de los servicios tecnológicos. La ISO 20000, al ser publicada, evidencia que las tecnologías de la información han alcanzado un nivel de madurez que requiere a la mayor parte de las instituciones adoptarlas para garantizar su supervivencia.

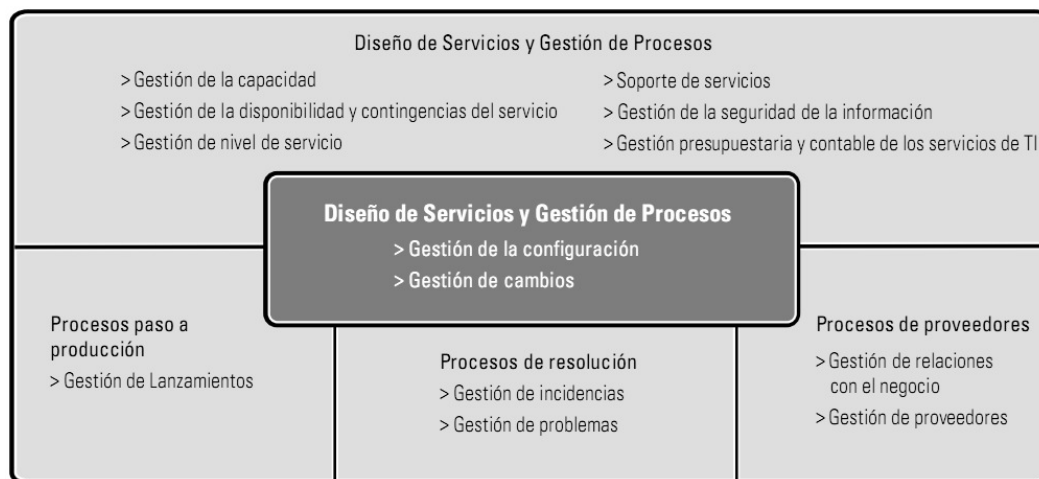


Figura 2. Procesos de gestión de servicios de la norma ISO 20000

La ISO 20000 es un estándar que brinda las bases para evaluar y confirmar si una entidad ha tenido éxito en la implementación de las buenas prácticas definidas por ITIL.

Sustituye la norma BS 15000 y ofrece una fórmula estándar para comprobar que las organizaciones han seguido las mejores prácticas de gestión de servicios de TI, tal como lo ha determinado el modelo ITIL, que ha funcionado como un estándar de facto en la gestión de servicios durante cerca de dos décadas.

La meta final de la nueva norma es:

- Disminuir la exposición a los riesgos que provienen de las operaciones
- Cumplir con los compromisos contractuales
- Mostrar la calidad del servicio

Los documentos de la BS 15000 que sirven como base para el contenido de la norma ISO 20000 son los siguientes:

- Parte 1: engloba un grupo de requisitos mínimos y fomenta el uso de un modelo de procesos integrados para llevar a cabo una administración efectiva de los servicios que satisfaga las necesidades de las compañías y sus consumidores
- Parte 2: incluye un código de prácticas para la administración de servicios ("Code of Practice for Service Management") que examina los componentes esenciales de las prácticas definidas por el modelo ITIL. El objetivo de este documento es asistir a las organizaciones en la creación de procesos que satisfagan las metas establecidas en la Parte 1.

2.3.2. Gobierno de Tecnología de Información

Weill y Ross (2004) definen el Gobierno TI como el conjunto de acciones que lleva a cabo la unidad de TI en colaboración con la alta dirección, con el fin de movilizar sus recursos de la manera más eficiente para responder a las demandas regulatorias, operativas o comerciales.

El Gobierno de TI forma parte fundamental del gobierno de la compañía en su totalidad y reúne la estructura directiva y organizativa requerida para garantizar que TI respalda y propicia el progreso de los objetivos estratégicos establecidos.

Un gobierno de TI asegura que, mediante el alineamiento entre la estrategia de información y el Gobierno de TI de potencie el desempeño organizacional mediante Wang, F., et al. (2022):

- La tecnología de la información (TI) está en sintonía con la estrategia empresarial.
- Los servicios y funciones de TI se ofrecen con el valor más alto que sea posible o de la manera más eficaz.
- Se conocen y gestionan todos los riesgos asociados a TI, y los recursos de TI están protegidos.

Se puede caracterizar al Gobierno de TI como un sistema de relaciones y procedimientos para liderar y supervisar la compañía, con el propósito de lograr las metas del negocio y agregar valor a la vez que se equilibran los riesgos y el rendimiento sobre TI y sus procesos.

Los dos deberes fundamentales del núcleo de TI son proporcionar valor al negocio y reducir los riesgos asociados con TI. La dirección de la entidad requiere extender sus responsabilidades gubernamentales a TI y proporcionar estructuras y procedimientos que garanticen que las Tecnologías de Información pueden respaldar las estrategias y metas de la organización.

La implementación de la gobernanza de TI se realiza en diferentes circunstancias y condiciones, que son definidas por elementos como:

- Cultura y ética de la empresa y del sector.

- Normas, regulaciones y directrices en vigor, ya sean internas o externas.
- Los valores, la misión y la visión de la organización.
- La organización de sus funciones y obligaciones.
- Objetivos estratégicos y tácticos de la entidad.

2.3.2.1. *COBIT: Gobierno y gestión de tecnologías de información*

El marco COBIT (Control Objectives for Information and Related Technologies) es un conjunto de buenas prácticas orientadas a la gobernanza y gestión de las Tecnologías de la Información (TI) en las organizaciones. Originalmente desarrollado por ISACA en la década de 1990, COBIT surgió con el propósito de ayudar a las empresas a integrar los objetivos de control de TI con los requerimientos de cumplimiento normativo y las necesidades del negocio. Desde su primera versión, ha evolucionado significativamente hasta su versión más reciente, COBIT 2019, que se caracteriza por un enfoque flexible y adaptable a distintos contextos y tamaños de organización (ISACA, 2019).

COBIT se basa en la premisa de que el éxito empresarial depende de que los procesos de TI se alineen con las estrategias corporativas. Para ello, establece principios, objetivos de gobierno y prácticas de gestión que permiten que las organizaciones maximicen el valor de la información y la tecnología. Este enfoque holístico considera aspectos como la calidad de la información, la optimización de riesgos y la gestión eficiente de los recursos tecnológicos (ISACA, 2012). Su estructura se sustenta en componentes como procesos, estructuras organizativas, principios, políticas, información, cultura, habilidades y servicios (De Haes et al., 2009)

COBIT incluye un modelo de objetivos de gestión y gobierno que se estructura en dominios. COBIT 2019 establece cinco dominios que contienen en total 40 objetivos de gobierno y gestión: Evaluar, dirigir y monitorear (EDM); alinear, planificar y organizar (APO); construir, adquirir e implementar (BAI); ofrecer, prestar servicio y apoyar (DSS); y supervisar, evaluar y valorar (MEV). Cada uno de estos dominios abarca actividades y prácticas particulares que las organizaciones pueden ajustar a su nivel de madurez y requerimientos de control (ISACA, 2019).

Uno de los aportes fundamentales de COBIT radica en su capacidad de servir como puente entre otros marcos y estándares, tales como ITIL, ISO/IEC 27001, ISO/IEC 38500 y COSO. Su enfoque integrador permite alinear los controles internos de TI con marcos de referencia de gestión de riesgos y de auditoría. De hecho, COBIT ha sido reconocido como una herramienta de referencia clave por auditores internos y externos al momento de evaluar el control y gobierno de TI en las organizaciones (Hardy, 2006).

En relación con la gestión de riesgos, COBIT proporciona lineamientos que permiten identificar, evaluar y tratar los riesgos tecnológicos de forma estructurada y en consonancia con el apetito de riesgo de la organización. En COBIT 2019, la relación con la gestión de riesgos se articula mediante componentes de diseño como factores de diseño, objetivos de gobierno y prácticas específicas, lo que facilita su integración con marcos como ISO 31000 y NIST SP 800-37 (ISACA, 2019). Esta visión facilita que los responsables de TI puedan justificar inversiones, priorizar iniciativas y demostrar cumplimiento ante reguladores y accionistas.

Otro aspecto relevante es el enfoque en la medición del desempeño de los procesos de TI. COBIT propone un sistema de evaluación que permite determinar el grado de capacidad de cada proceso mediante niveles que van desde el nivel 0 (proceso incompleto) hasta el nivel 5 (proceso optimizado). Este enfoque, conocido como Process Capability Model, facilita identificar brechas, planificar mejoras y monitorear la evolución del gobierno y gestión de TI en el tiempo (ISACA, 2012).

COBIT también ha evolucionado en su aplicabilidad a organizaciones de diferentes tamaños y niveles de madurez. Mientras las versiones previas estaban principalmente orientadas a grandes corporaciones, COBIT 2019 se diseñó con un enfoque modular que permite adaptar los componentes de gobierno y gestión en entornos medianos y pequeños. Además, el concepto de factores de diseño permite personalizar el marco según prioridades estratégicas, perfiles de riesgo, complejidad y cultura organizacional (ISACA, 2019).

Finalmente, el uso de COBIT en la actualidad no solo contribuye al fortalecimiento del control interno y la gestión de riesgos, sino que también aporta valor en términos de alineamiento estratégico, optimización de recursos y fomento de una cultura orientada al cumplimiento y la mejora continua. Estudios recientes destacan que su adopción efectiva depende en gran medida del compromiso de la alta dirección y de la capacidad de las

organizaciones para integrar las prácticas de COBIT con otros marcos de referencia relevantes (De Haes, et al., 2013).

2.3.3. Mejora de Proceso de Desarrollo de Software

Las compañías han notado que la calidad de los procesos empleados para desarrollar y mantener sus productos de software está estrechamente relacionada con la calidad de estos mismos productos. Para lograrlo, las compañías certifican sus procedimientos con modelos que la industria del software considera válidos, tales como ISO/IEC 15504 o CMMI.

2.3.3.1. CMMI V2.2

El Capability Maturity Model Integration (CMMI) es un modelo de maduración que mejora los procesos de desarrollo y productos y servicios (ISACA, 2022). Se creó en el Software Engineering Institute (SEI) de la Universidad Carnegie Mellon, con la colaboración de las industrias y el gobierno de Estados Unidos. Este modelo incluye las mejores prácticas para dirigir las actividades de mantenimiento y desarrollo a lo largo del ciclo vital del producto, desde su concepción hasta su entrega y mantenimiento. El propósito del CMMI es proveer una guía para mejorar los procesos organizacionales y su capacidad para manejar el desarrollo la adquisición y el mantenimiento de productos y servicios.

El marco CMMI (Cater-Steel, 2006) es la estructura fundamental que organiza los elementos comunes y los modelos actuales de CMMI, además de las reglas y técnicas para crear modelos, sus métodos de valoración y su material de formación.

Las constelaciones de CMMI han sido generadas para dar soporte a específicas áreas de interés. Actualmente tres constelaciones del framework CMMI se encuentran disponibles: CMMI Development, CMMI Services y CMMI Supplier Management. Todas fundamentan su estructura en niveles de madurez que van desde un nivel inicial hasta un nivel optimizado que busca en esencia la mejora continua de sus procesos, se pueden observar en la figura 3.



Figura 3: Niveles de madurez CMMI V2.0. Tomado de ISACA

Un modelo CMMI está básicamente compuesto de una dimensión por área de práctica que logra una dimensión de madurez, según el grado de experiencia y buena ejecución. En un modelo CMMI un área de práctica posee un conjunto de prácticas relativas (ver figura 4) que dependen de un grupo de práctica y su contexto específico.

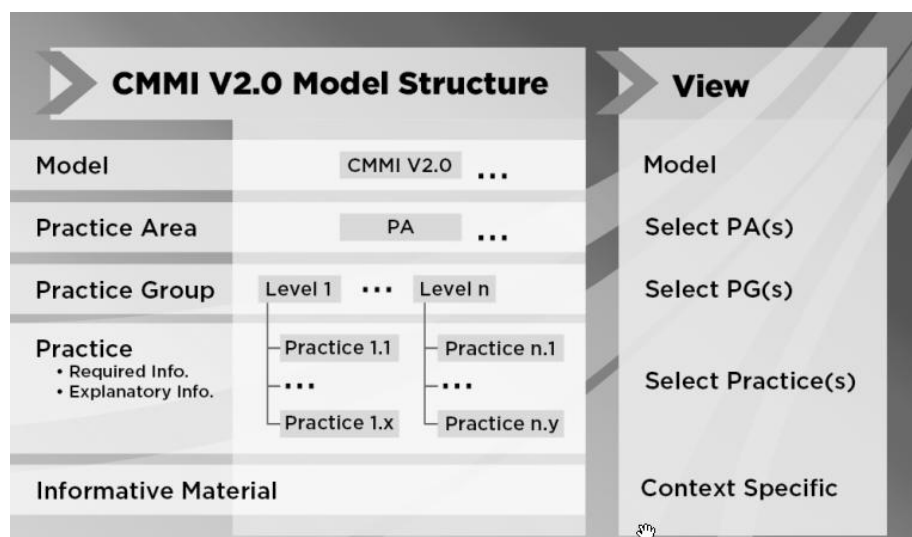


Figura 4. CMMI V2.0 Model Structure. Tomado de ISACA

El CMMI posee una base de datos integrada de sus modelos en un único framework, todas ellas manejan el mismo modo de construcción del modelo, así como de su evaluación, teniendo así uniformidad en adopción de múltiples modelos para una organización interesada en formar capacidades que van más allá del desarrollo, sino que abarcan también la entrega de servicios y el manejo de proveedores de desarrollo, (ver figura 5).

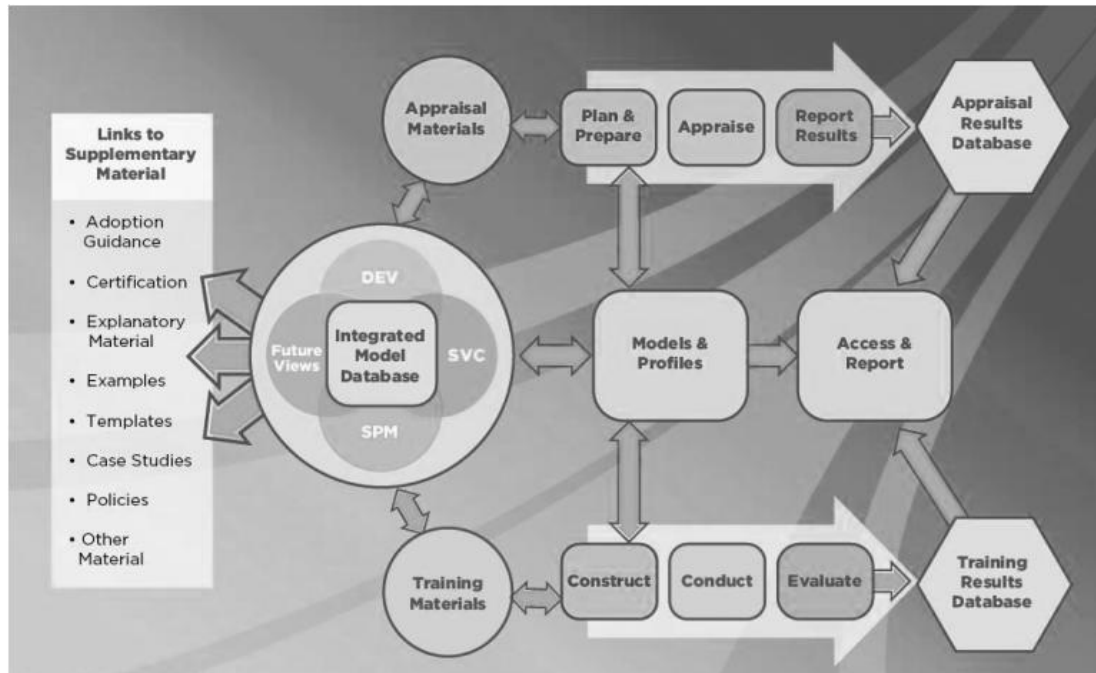


Figura 5: Integrated CMMI Product Suite. Tomado de ISACA

2.3.4. Mejora de Proceso de Pruebas de Software

2.3.4.1. TMMi

Para lograr una optimización de procesos, existen diversos modelos de referencia que definen el conjunto de buenas prácticas que debe seguirse en una organización (ISACA, 2022). CMMI es el modelo de referencia más extendido en la industria del software y proporciona el respaldo necesario para realizar tareas relacionadas con servicios y productos. Igualmente, existen modelos de referencia diseñados específicamente para el procedimiento de pruebas de software. Los que se destacan más son: Test PAI, TPI y TMMi (TMMi Foundation, 2022). Estos modelos ofrecen el marco de referencia para determinar las debilidades y fortalezas en el ámbito de pruebas de software.

CMMI define tres áreas de proceso para el procedimiento de pruebas: Integración, validación y verificación del producto; sin embargo, no cumplen con todos los requisitos del

proceso de pruebas. No están definidas todas las prácticas ideales relacionadas con el testeo de software, ni tampoco todos los productos que se necesitan.

En organizaciones que tienen una estructura de pruebas organizativas y donde los grupos de SQA, los desarrolladores o equipos de pruebas integrados asumen el rol de testers, TestPAI (Sanz et al., 2008) es más eficaz. Para equipos de pruebas independientes o factorías de pruebas, este modelo es demasiado sencillo y no contempla los procedimientos relacionados con la gestión de servicios.

TMMi Foundation (2022) resulta más eficaz si se utiliza en entidades que utilizan una estructura de pruebas organizativas fundamentada en grupos autónomos, equipos de SQA o fábricas de pruebas. Este modelo de referencia no es apropiado si los grupos de pruebas integrados o los desarrolladores son los testers, ya que el tamaño reducido de estos equipos haría que la carga laboral fuera excesiva.

2.3.4.2. TPI

TPI (Sanz et al., 2008) se emplea principalmente en organizaciones que tienen una estructura de pruebas organizativas basada en equipos independientes, grupos de SQA o factorías de pruebas que implementen TMap como metodología. Si los testers o los equipos de pruebas integrados son los desarrolladores, la carga de trabajo sería desmedida en caso de que estos grupos no sean demasiado grandes.

Aún persiste la concepción de que el testing implica un costo adicional; sin embargo, al disminuir los costos de mantenimiento debido a la existencia de menos fallos, se obtiene una recuperación directa de la inversión, así como una recuperación indirecta porque las compañías no ven afectada su productividad ni sufren un impacto negativo en sus negocios.

2.3.5. Estándares de Calidad

2.3.5.1. ISO 9000

La ISO 9000, por la International Organization for Standardization (ISO) refiere un conjunto de estándares de gestión de la calidad que permiten a las organizaciones a cumplir a cabalidad con los requerimientos de calidad del cliente, además de cumplir con requerimientos

regulatorios, todo esto mientras logra la satisfacción del cliente y una mejora continua de su rendimiento.

ISO 9000 es de naturaleza genérica y es aplicable organizaciones del sector público y privado, sin importar el tamaño y tipo, además de ser aplicable a todo tipo de producto o servicio.

2.3.5.2. ISO/IEC 25000

SQuaRE (Software Product Quality Requirements and Evaluation), que corresponde a la serie de normas ISO/IEC 25000, es un marco internacionalmente conocido para el manejo de la calidad del producto software. La Comisión Internacional de Electrotecnia (IEC) y la Organización Internacional de Normalización (ISO) desarrollaron esta serie de estándares con el propósito de definir instrucciones precisas para determinar, medir y evaluar la calidad en los sistemas y productos informáticos. SQuaRE fue creado como una evolución y consolidación de normas anteriores, como ISO/IEC 9126 e ISO/IEC 14598, añadiendo avances en términos conceptuales y metodológicos (ISO/IEC, 2014).

La noción de que la calidad del software no se puede restringir a una evaluación superficial o a controles finales, sino que necesita una perspectiva completa que cubra todo el ciclo de vida, es el principio fundamental de ISO/IEC 25000. Para esto, la serie de normas se divide en cinco partes fundamentales: evaluación, medición, calidad en uso, modelos de calidad y requisitos de calidad. Cada división incluye normas particulares que proporcionan directrices pormenorizadas y sugerencias de carácter práctico para cada elemento de la gestión de calidad (ISO/IEC, 2014).

ISO/IEC 25010:2011 es uno de los elementos fundamentales de este marco; establece el modelo de calidad del sistema y del producto software. Este modelo presenta ocho atributos fundamentales: la adecuación funcional, la portabilidad, la eficacia del rendimiento, la seguridad, la compatibilidad, la usabilidad, el mantenimiento y la fiabilidad. Cada atributo se divide en subatributos más concretos que posibilitan una evaluación detallada de diferentes características del producto (ISO/IEC, 2011). Este modelo se ha implementado en gran medida, tanto en proyectos de desarrollo industrial como en estudios académicos.

El concepto de calidad en uso, que alude a la medida en que un sistema le permite a los usuarios lograr sus metas en una situación real de funcionamiento, también está incluido en la norma ISO/IEC 25010. La efectividad, la satisfacción del usuario, la mitigación de riesgos, la eficiencia y el alcance del contexto son los elementos fundamentales de la calidad en uso. Esta metodología proporciona una visión holística que complementa la evaluación técnica de las funciones (ISO/IEC, 2011).

La adopción de la norma ISO/IEC 25000 permite que las entidades establezcan criterios de aprobación y métricas numéricas desde el comienzo de los proyectos. Esta práctica ayuda a disminuir la vaguedad en los requisitos, hace que la toma de decisiones sea más transparente y promueve el seguimiento de los objetivos de calidad durante todo el ciclo de desarrollo (Pérez-Castillo, García-Rodríguez de Guzmán y Piattini, 2011). La utilización constante de métricas, además, posibilita que se realicen comparaciones entre versiones y se evalúe cómo ha evolucionado la calidad a lo largo del tiempo.

La posibilidad de ser compatible con otros modelos y estándares de referencia, como ISO/IEC 27001, ITIL y CMMI, es otro beneficio importante de este marco. La adopción conjunta de estos marcos posibilita que las organizaciones fortalezcan sus procedimientos de ingeniería de software, administren los riesgos en su totalidad y evidencien que cumplen con lo estipulado en contratos y normativas (Kitchenham et, al., 2008).

La implementación de ISO/IEC 25000 no está libre de retos. La capacitación de los equipos de trabajo, la implementación de herramientas para automatizar el análisis y recolección de datos, así como la definición exacta de métricas adecuadas al contexto del proyecto son algunos ejemplos. Sin embargo, estudios recientes han demostrado que estos esfuerzos tienen resultados concretos, como son la disminución de los defectos, el aumento de la satisfacción del consumidor y el fortalecimiento de la confianza en el producto final (Pérez-Castillo et al., 2018).

2.3.5.3. Gestión de proyectos

2.3.5.4. PMBOK

The Guide to the Project Management Body of Knowledge es un estándar reconocido para la gestión de proyectos (Project Management Institute [PMI], 2021). Proporciona directrices para la gestión de proyectos y describe el ciclo de vida del proyecto mediante procesos y áreas de conocimiento.

PMBOK fue desarrollado mediante consenso voluntario de procesos de desarrollo estándares, coordinados por el Project Management Institute, PMI. Actualmente se encuentra en la versión 7 publicada en el 2021. En comparación a CMMI, el PMBOK hace una clara distinción entre proyectos, los cuales son temporales y tienen fin, y el trabajo operativo, el cual es continuo y sostiene a las organizaciones a través del tiempo.

Según el PMBOK, la gestión de proyectos es entendida como la aplicación de conocimientos, habilidades, herramientas y técnicas, para cumplir con los requerimientos del proyecto. Los principios de la Dirección de Proyectos guían el comportamiento de los Dominios de Desempeño del Proyecto, como se observa en la figura 6.

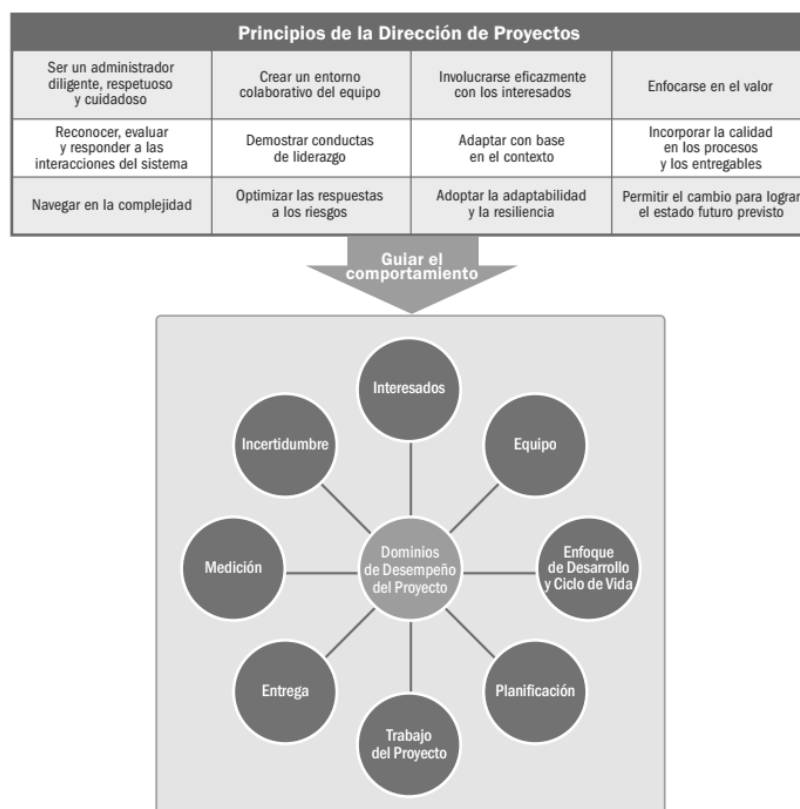


Figura 6. Interacción entre Principios y Dominios de la Dirección de Proyectos.

Estos procesos incluyen las técnicas y herramientas que se utilizan para llevar a cabo habilidades y capacidades especificadas en nueve áreas de conocimiento: gestión de las adquisiciones, gestión del riesgo, gestión de los recursos humanos, gestión de la calidad, gestión del costo, gestión del tiempo, gestión del alcance y gestión de la integración.

2.3.6. Testing de aplicaciones

2.3.6.1. Definición de testing de software

En el ciclo de vida del desarrollo de software, las pruebas o servicios de testing representan un componente esencial para garantizar que los productos cumplan con los requisitos funcionales y no funcionales establecidos por los clientes. Estas pruebas no se limitan a ser una fase aislada, sino que constituyen una estrategia transversal que acompaña desde el diseño hasta la implementación del sistema (Serna y Arango, 2011).

Una prueba de software eficaz va más allá de la simple detección de errores. Involucra una comprensión profunda del contexto operativo del software, los escenarios de uso, las posibles combinaciones de entradas y salidas, y los riesgos potenciales derivados de fallos no detectados (Carrizo y Alfaro, 2018). En este sentido, se ha vuelto imperativo profesionalizar el rol del tester, ya que, como señalan Serna y Arango (2011), aún existe una gran brecha entre el desarrollo de software y la calidad de los procesos de prueba ejecutados.

Las pruebas funcionales, como práctica común, permiten verificar que el sistema cumple con las operaciones y acciones definidas en los requisitos. Estas pruebas se realizan ingresando datos representativos y analizando las salidas resultantes para validar el comportamiento esperado del software (Blanquicett, et, al., 2018). Asimismo, existen pruebas orientadas a la interfaz gráfica, que aseguran la correcta visualización y navegación de los elementos de interacción definidos por los diseñadores.

Otro tipo esencial de servicio de testing es la prueba de regresión, la cual se encarga de reejecutar los casos de prueba previos cuando se han realizado cambios o correcciones en el sistema, con el objetivo de asegurar que nuevas modificaciones no hayan generado efectos colaterales negativos (Blanquicett et al., 2018). Esta práctica es fundamental para garantizar la estabilidad del software en evolución constante.

La automatización de pruebas se ha consolidado como una práctica clave dentro del aseguramiento de la calidad, especialmente en contextos donde los ciclos de entrega son ágiles y recurrentes. Automatizar no solo reduce el esfuerzo manual y los errores humanos, sino que permite ejecutar pruebas con mayor frecuencia, mejorando la cobertura y eficiencia del testing (Carrizo y Alfaro, 2018).

En el estudio de Blanquicett et al. (2018) sobre la plataforma ASISTO, se evidencia cómo un proceso de pruebas bien estructurado puede impactar positivamente en el cumplimiento de los requerimientos funcionales y la satisfacción de los usuarios finales. Esta plataforma, desarrollada para entidades prestadoras de salud, aplicó pruebas de interfaz, funcionalidad y rendimiento, incluyendo tiempos de respuesta, validaciones de seguridad y cumplimiento de normas legales.

Asimismo, la definición de roles dentro del proceso de testing es crucial. La metodología usada por la empresa SYSNET, según Blanquicett et al. (2018), establece claramente el papel del desarrollador, verificador y cliente en el proceso de validación del software, con herramientas específicas de seguimiento y control que facilitan la trazabilidad de los casos de prueba.

Por otra parte, se han desarrollado marcos metodológicos para abordar el testing desde una perspectiva sistemática, como el propuesto por Carrizo y Alfaro (2018), quienes destacan la importancia de incorporar métricas, herramientas y metodologías que fortalezcan el aseguramiento de la calidad a través del monitoreo continuo y la retroalimentación efectiva.

Las pruebas no funcionales también cobran relevancia, ya que permiten evaluar dimensiones como la fiabilidad, la eficiencia, la usabilidad y la seguridad. Estos aspectos son clave para garantizar no solo que el sistema funcione, sino que lo haga de manera óptima y sostenible en distintos contextos de operación (Serna y Arango, 2011).

Además, se reconoce que la planificación del testing debe integrarse desde las fases iniciales del desarrollo. En palabras de Marcano y Benigni (2014), las pruebas deben ser contempladas desde la especificación de requisitos, pasando por el diseño, hasta la validación y evolución del producto, asegurando así una alineación completa con los objetivos del proyecto.

2.3.6.2. *Validación y Verificación*

Según el proceso de testing, la verificación y validación del software y sus elementos habilitan la inclusión de actividades de calidad del software durante los procesos de desarrollo (ISACA, 2022). Esta incorporación puede llevarse a cabo teniendo en cuenta un producto, una iniciativa o un proyecto específicos.

Estas actividades tienen por objeto mejorar y asegurar la calidad del desarrollo durante todo el proceso, y su finalidad es obtener un mayor control sobre el proceso, una identificación temprana de errores y problemas lo que permitirá una reducción de los costos para subsanar errores en los desarrollos.

Estos procedimientos pueden llegar a ser muy costosos y, de hecho, podrían absorber más del 50% del presupuesto de desarrollo de un proyecto de software (Sommerville, 2016). Los procesos de verificación y validación determinan las actividades que se sumarán durante el desarrollo, como las siguientes:

- Diseño e implementación de entornos de pruebas.
- Organización de procedimientos de prueba. Definición de las labores de prueba.
- Evaluaciones formales de documentación.
- Inspecciones de código.
- Pruebas de integración y unitarias.
- Evaluaciones de aceptación y funcionales.
- Métodos estáticos y dinámicos.
- Puesta en marcha de estándares de seguridad y calidad.
- Capacitación del usuario final.

2.3.6.3. *Tipos de Prueba*

Los ensayos de las aplicaciones y/o sistemas informáticos pueden dividirse en (Farrell-Vinay, 2008):

Pruebas unitarias: son las que se dedican a evaluar una clase específica, probando cada uno de sus métodos y comprobando si la salida es la prevista cuando se ingresan algunos parámetros.

Pruebas funcionales: estas son las que ponen a prueba una funcionalidad en su totalidad, y pueden incluir una o más clases además de la interfaz del usuario.

Pruebas de regresión: son aquellas cuyo objetivo es comprobar el funcionamiento conjunto de los nuevos componentes desarrollados y aquellos que ya se encontraban en el ambiente productivo. El objetivo de las pruebas de regresión es no tener que “volver atrás”.

Pruebas de aceptación: son pruebas funcionales, pero observadas directamente desde la perspectiva del cliente. Son las pruebas que demuestran al cliente que la funcionalidad está finalizada y funciona de manera adecuada.

Pruebas de integración: pruebas unitarias, funcionales, de aceptación y/o de regresión realizadas al verificar el software. Además, incluye verificar que lo programado por los diversos desarrolladores no interfiera entre sí y que funcione en un entorno real.

2.3.6.4. Perfil del Tester

Un tester examina un producto de software para recabar información sobre su calidad y el valor que tiene para sus usuarios, y se encarga de identificar la mayor cantidad posible de errores graves (incidentes de alto impacto) con el menor esfuerzo antes de que el software sea puesto en producción (Kaner et al., 1999).

Entre estas destrezas sobresalen:

- Aptitud para abstraer y modelar con el fin de comprender y simular cómo funciona el sistema que se está poniendo a prueba.
- Capacidad para comunicarse de forma oral y escrita con los usuarios y los desarrolladores.
- Capacidad creativa para concebir ideas e imaginar los problemas que podrían surgir.
- Evaluar las ideas, hacer deducciones y asociar lo observado con los estándares de calidad de la compañía requiere pensamiento crítico.
- Pragmatismo para implementar los conceptos y ajustar las técnicas y el trabajo al alcance del proyecto.
- Habilidades para trabajar en equipo, con el fin de interactuar con otros testers y desarrolladores y obtener el máximo provecho de esta interacción.

Estas aptitudes se desarrollan y mejoran conforme se progresa en la carrera de testing, adquiriendo así más conocimiento y experiencia.

2.3.6.5. *Roles en el testing*

Hay varios roles que intervienen en un proyecto de pruebas (Black, 2002; Kaner et al., 1999; Kshirasagar et al., 2008). Una persona puede desempeñar múltiples funciones en un proyecto y ocupar distintos papeles en proyectos diferentes.

Se sostiene que, al asumir diferentes roles, se incrementa la adquisición de experiencia y conocimiento. En un equipo de testing de Software, se distinguen tres categorías principales: Consultor, líder de pruebas y probador de software (Kaner et al., 1999).

- **Tester de Software.** Es la persona que diseña, aplica y comunica el resultado de las pruebas de un producto de software. Los testers forman parte del equipo de pruebas y colaboran con el equipo de desarrollo y la gerencia de proyectos.
- **Líder de testing de software.** Un líder de proyecto de pruebas posee conocimientos y experiencia en la creación, realización e informe de las pruebas sobre un producto de software. Además, tiene la habilidad para calcular, organizar y supervisar un proyecto de pruebas.
- **Consultor.** Es un puesto apropiado para individuos que tienen la capacidad de liderar un equipo de pruebas y guiar acerca de las mejores prácticas en testing.

Los testers podrían ser empleados a su máximo potencial con tareas operativas que colaboran con los desarrolladores en fomentar la creación de soluciones más eficaces y en etapas tempranas.

A la vez, los integrantes del equipo podrían constituir fuentes de información permanente acerca del comportamiento del sistema en construcción, lo que posibilitaría anticiparse a modificaciones en los requerimientos o a requerimientos mal concebidos o desarrollados de manera no deseada.

2.3.6.6. *Estructuras organizativas del testing*

Hay varias maneras de estructurar las pruebas de software, desde estructuras simples hasta organizaciones que ofrecen servicios de prueba (Sanz, 2009; Venkatasubramanian Y Vinoline, 2010). La política y cultura de la organización, así como las capacidades y el conocimiento de los recursos humanos involucrados en el proceso de prueba de software, determinan qué estructura se acomoda mejor a cada organización.

Aquí se presentan las estructuras organizativas más relevantes que son implementadas en las organizaciones de software:

Los desarrolladores son testers. Los responsables de comprobar y validar el producto son los desarrolladores de este. El problema principal es que el creador del código tiene un conocimiento cabal de cómo se ha creado y, sin darse cuenta, puede diseñar casos de prueba para cumplirlos en vez de para detectar errores. Asimismo, no tienen las capacidades y los conocimientos necesarios para llevar a cabo este tipo de trabajos.

Equipos de pruebas integrados: comprobar y validar es responsabilidad del equipo de pruebas. Este está integrado con el equipo de desarrollo mediante el líder del proyecto. El problema principal de esta estructura surge cuando el equipo se encuentra bajo presión. Por lo general, la actitud del jefe de proyecto tiende a respaldar la finalización del producto sacrificando su calidad, o sea, reduciendo las actividades de prueba.

Grupos de pruebas independientes: La organización cuenta con un equipo de pruebas, que no pertenece al equipo de desarrollo, y es responsable de llevar a cabo las actividades de validación y verificación. La comunicación es el mayor inconveniente de esta estructura. El equipo de desarrollo no tiene una buena opinión del equipo de pruebas, lo que crea un muro entre ambos y dificulta la comunicación y el trabajo.

Grupos de SQA (Aseguramiento de la calidad del software): Las empresas tienen al equipo de SQA a cargo de las tareas de validación y verificación. Estos son más valorados por los desarrolladores porque no pertenecen al equipo de pruebas, sino al grupo que se encarga del aseguramiento de la calidad, lo cual permite una comunicación más fluida. No obstante, el equipo de SQA lleva a cabo otras tareas y esta carga extra hace que no puedan realizar su trabajo con eficacia.

Factorías de pruebas: Las entidades delegan la totalidad o parte de las tareas de validación y verificación a empresas que se especializan en pruebas de software: Fábricas de pruebas. La ventaja principal es que las empresas especializadas tienen a su disposición la infraestructura necesaria y personal con experiencia; en cambio, el esfuerzo adicional de gestión que implican estas actividades externalizadas constituye la mayor desventaja.

2.3.7. Outsourcing

El outsourcing es una estrategia de negocios que permite delegar a un agente externo responsabilidades operacionales sobre procesos o actividades previamente realizados por la propia empresa que la contrata (Elmuti y Kathawala, 2000).

Las razones más importantes para la elección del outsourcing como forma de operación son la eficiencia de costos y la reorganización de la producción (Hanafizadeh, P., y Zareravasan, M. (2020), Francheschini, 2006). Mediante esta práctica las compañías pueden enfocar sus esfuerzos en las actividades claves del negocio, objetivos a mediano y largo plazo, así como la diversificación de oportunidades.

Los actores principales de esta práctica son el Cliente y el Proveedor. El primero externaliza sus procesos de negocio, mientras el segundo entrega servicios que efectúan los procesos requeridos por el primero.

Se muestra una evolución en esta práctica de negocio, que va desde el outsourcing tradicional hasta el estratégico. Es tradicional si los procesos externalizados no son considerados críticos para la organización, que es cuando no se requieren competencias específicas por parte del proveedor para llevar a cabo las actividades contratadas. Se convierte en estratégico cuando las compañías externalizan aquellas actividades especiales en la cual podrían lograr ventajas competitivas (Willcocks, 1995)

2.3.7.1. Outsourcing de testing de aplicaciones

En el ámbito del desarrollo de software, estudios indican que la principal fase candidata a externalizar dentro del proceso es la fase de Testing, The Standish Group. (2010). Es así,

debido al ahorro en costes y grado de especialización de los recursos que efectuarán las tareas de prueba.

Comúnmente las empresas que externalizan estas actividades consideran como importante el desempeño de la empresa contratada en:

- Pruebas de regresión manual y automatizada.
- Desarrollo de planes y casos de prueba efectivos, así como scripts de prueba.
- Adaptabilidad al cambio y requerimiento de nuevas tecnologías.
- Equipos de prueba especializados, que van desde los testers hasta los jefes de proyectos de testing.
- Disposición de expertos a un costo menor de si tuviese que contratárseles.

Los beneficios percibidos por las empresas contratantes pueden resumirse en:

- Equipos internos (o externalizados también) enfocados propiamente en el desarrollo del software.
- Flujo de trabajo alternativo en caso de carga laboral.
- Reducción del riesgo de obsolescencia técnica de los ambientes de prueba.
- Mejora en la satisfacción de los usuarios finales de las aplicaciones.
- Enfoque e imparcialidad en las pruebas por ser un equipo independiente, verificador de la calidad del producto.
- Mejora en la flexibilidad de programaciones y disposición de personal experto.
- Reducción del costo del ciclo de desarrollo, al detectar errores en las fases tempranas, lo que incurre en la disminución de la tasa de error con los consecuentes beneficios económicos.

Pero también existen riesgos producto del outsourcing que deben ser tomados en cuenta al momento de efectuar esta práctica (Deloitte, 2008):

- Pobre comunicación entre cliente y proveedor.
- Pobre gestión del conocimiento y transferencia inicial de este lo que genera pobre entendimiento de las aplicaciones y su funcionalidad.
- Gestión de proyecto desfasada entre el desarrollo y el área de testing.

- Falta de métricas, medidas y comparaciones que reporten o no, el éxito de los proyectos.
- Pobre medio de comunicación a través de Internet, que es la vía de coordinación y trabajo de este modelo

CAPÍTULO 3: METODOLOGÍA

3.1. Alcances y limitaciones de la investigación

3.1.1. Alcances de la Investigación

- El desarrollo del modelo de procesos para un servicio de testing de software basado en estándares y marcos de trabajo internacionales.
- Alcance del caso de estudio: PERÚ, Compañía Consultora de Tecnología de Información, Lima-Perú, servicio de testing de software.
- Alcance Temporal: Periodo 2020-2022.

3.1.2. Limitaciones de la Investigación

Limitación de acceso a documentación: algunas consultoras del rubro por motivos de confidencialidad de la información interna no brindan datos acerca de los registros de calidad y ejecución de sus procesos internos.

3.2. Metodología de Investigación

3.2.1. Proceso de Investigación

El presente proyecto utiliza el proceso para investigaciones en el campo de la Ciencia del Diseño propuesto por Offermann (Offermann et al., 2009). El proceso combina métodos de investigación cualitativos y cuantitativos usados en estudios de Sistemas de Información. Este proceso identifica tres áreas mayores compuestas por sus respectivos procesos.

El flujo del proceso de investigación puede apreciarse en la figura 7. La primera fase es la Identificación del Problema, en la cual se hace una primera revisión de la literatura existente sobre el tema de investigación, de esto se desprende y retroalimenta la identificación del problema de investigación. A su vez, mediante encuestas con expertos sobre el tema a investigar, gente con experiencia y conocimientos previos sobre el área de interés, se busca tener una visión global sobre el problema y su contexto. Se termina esta primera parte haciendo una evaluación de la relevancia del problema a investigar y si realmente se agrega conocimiento científico al área de investigación.

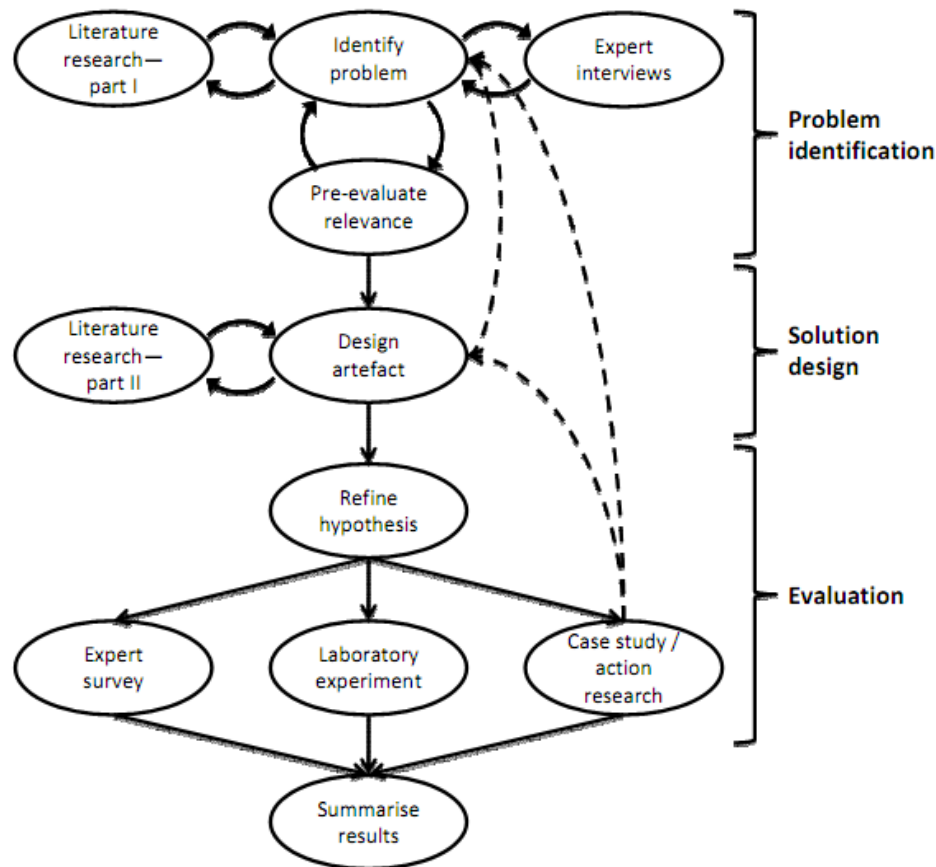


Figura 7. Proceso de Investigación propuesto, por Offermann et al

La segunda fase es el Diseño de la Solución. En esta se hace una segunda Revisión de la Literatura en busca de construcciones similares planteadas como solución al problema de investigación. Estas pueden ser nuevos algoritmos, procesos, o modelos, como el que presenta este estudio. Seguidamente se diseña el artefacto que dará solución al problema y cuya implementación y puesta en práctica buscará conseguir los objetivos planteados en la investigación.

La tercera fase es la de Evaluación, se refina la hipótesis general de la investigación en otras más pequeñas y de alcance más preciso que abarquen la hipótesis general. Luego se elige el método de investigación adecuado para validar la solución planteada. Estos pueden ser: encuestas a Expertos, Experimento de Laboratorio, y Estudio de Caso.

Por último, después de efectuado el método de contrastación de las hipótesis, se exponen y analizan los resultados y se emiten las respectivas conclusiones y líneas futuras de trabajo.

3.2.2. Desarrollo del Método

La primera fase, de Identificación del Problema, se corresponde al Marco Teórico y la primera revisión del Estado del Arte. Aquí se identificó la falta de un modelo de procesos para un servicio de testing de software que utilice las mejores prácticas de los modelos de procesos y de mejora como el problema principal a resolver.

Se complementó la investigación con la aplicación de encuestas estructuradas a 20 expertos en calidad de software y testing de aplicaciones, quienes desempeñan roles de Project Manager, Test Manager, Test Analyst y Team Leader (ver Anexo 1). Estos participantes fueron seleccionados mediante muestreo no probabilístico por criterio, considerando su experiencia profesional en proyectos de aseguramiento de calidad.

El cuestionario estuvo compuesto por ítems con escalas de medición cuantitativas, orientados a recoger datos sobre variables como la existencia de una estructura organizativa definida para el establecimiento de un servicio de testing, el grado de eficiencia percibida, la frecuencia de uso de modelos de calidad y la eficacia en la gestión del cronograma de pruebas.

La recopilación de estos datos cuantitativos permitió analizar de manera objetiva la percepción de los expertos sobre estos aspectos clave, así como identificar tendencias y patrones que fundamentan los resultados de la investigación.

Basado en los resultados de las encuestas y en la revisión de la literatura se evaluó la relevancia de la construcción de un modelo de este tipo, llegando a la conclusión que sería una solución adecuada para la implementación de este tipo de servicio en estas organizaciones.

La segunda fase del proceso de investigación, que corresponde al Diseño de la Solución, acomete la construcción del modelo basado en los resultados obtenidos de estudios previos contenidos en el Estado del Arte y el Marco Teórico.

Esta construcción del modelo tiene como fundamentos de forma la utilización de los principales componentes de marcos de trabajo reconocidos y con resultados reportados en el campo del testing de software, gestión de proyectos, gestión de servicios, y mejora de procesos; así como también los resultados de trabajos previos para solucionar el aspecto organizativo de un servicio de testing de aplicaciones bajo el modelo del outsourcing.

Seguidamente se replantearon las hipótesis de investigación y se definió el método para probarlas. El método elegido fue el de Estudio de Caso puesto que era el que mejor se adaptaba dado el escenario para desarrollarlo. Por lo tanto, se implementa el modelo, utilizando un equipo piloto de la organización en estudio, y se miden sus resultados contra los datos históricos de la organización en búsqueda criterios como:

Calidad funcional del software: La calidad funcional del software se refiere al grado en que un producto satisface las necesidades explícitas e implícitas de sus usuarios, mediante la provisión de funciones que cumplen con los requisitos especificados y que pueden operar correctamente bajo condiciones predefinidas. Esta dimensión incluye aspectos como la adecuación funcional, la corrección y la completitud de las funciones ofrecidas. Según ISO/IEC 25010:2011, la adecuación funcional es una característica clave del modelo de calidad, ya que determina si el software proporciona las capacidades necesarias para alcanzar los objetivos previstos por el usuario (ISO/IEC, 2011).

Rendimiento y eficiencia: El rendimiento y la eficiencia del software se relacionan con la capacidad del sistema para utilizar adecuadamente los recursos, ofreciendo tiempos de respuesta y tasas de procesamiento que satisfagan las expectativas del usuario en diversas condiciones de carga. Este aspecto comprende subcaracterísticas como el comportamiento temporal (latencia y tiempo de respuesta) y la utilización de recursos (memoria, CPU, ancho de banda). ISO/IEC 25010 establece que un software eficiente debe optimizar el consumo de recursos sin sacrificar la calidad funcional, garantizando la sostenibilidad del sistema en producción (ISO/IEC, 2011).

Fiabilidad y disponibilidad: La fiabilidad del software describe el grado en que un sistema mantiene su nivel de desempeño durante un periodo de tiempo bajo condiciones establecidas. Esto incluye atributos como la madurez, la disponibilidad, la tolerancia a fallos y la recuperabilidad. La disponibilidad, en particular, se define como la proporción de tiempo durante la cual un sistema está operativo y accesible cuando se requiere su uso. Según Laprie (1995), la fiabilidad es esencial para aplicaciones críticas, pues incide directamente en la confianza del usuario y en los costes operativos derivados de interrupciones (Laprie, 1995).

Percepción del servicio entregado al cliente: La percepción del servicio entregado al cliente en el contexto de software hace referencia a la valoración que los usuarios finales realizan sobre la experiencia global de uso del sistema, considerando aspectos como la calidad

en uso, la facilidad de aprendizaje, la asistencia técnica recibida y la confianza que genera el producto. Este concepto se vincula estrechamente con la característica de calidad en uso definida por ISO/IEC 25010, que contempla dimensiones como la efectividad, la eficiencia, la satisfacción y la mitigación de riesgos en condiciones reales de operación. Según Bevan (1999), la percepción de la calidad en uso es esencial para evaluar si el software cumple sus objetivos en el entorno del usuario, pues una experiencia deficiente puede afectar la adopción y la fidelización. Además, Kitchenham y Pfleeger (2008) subrayan que la percepción del servicio también se relaciona con la capacidad del proveedor para atender incidencias y responder a las expectativas de soporte, elementos que constituyen un factor diferenciador en mercados altamente competitivos.

3.2.3. Método de Revisión de la Literatura

Referente a la revisión de la literatura se ha seguido la estrategia de búsqueda e identificación de artículos de investigación propuesta por Webster (Webster and Watson 2002). La figura 8 hace referencia al proceso de búsqueda.

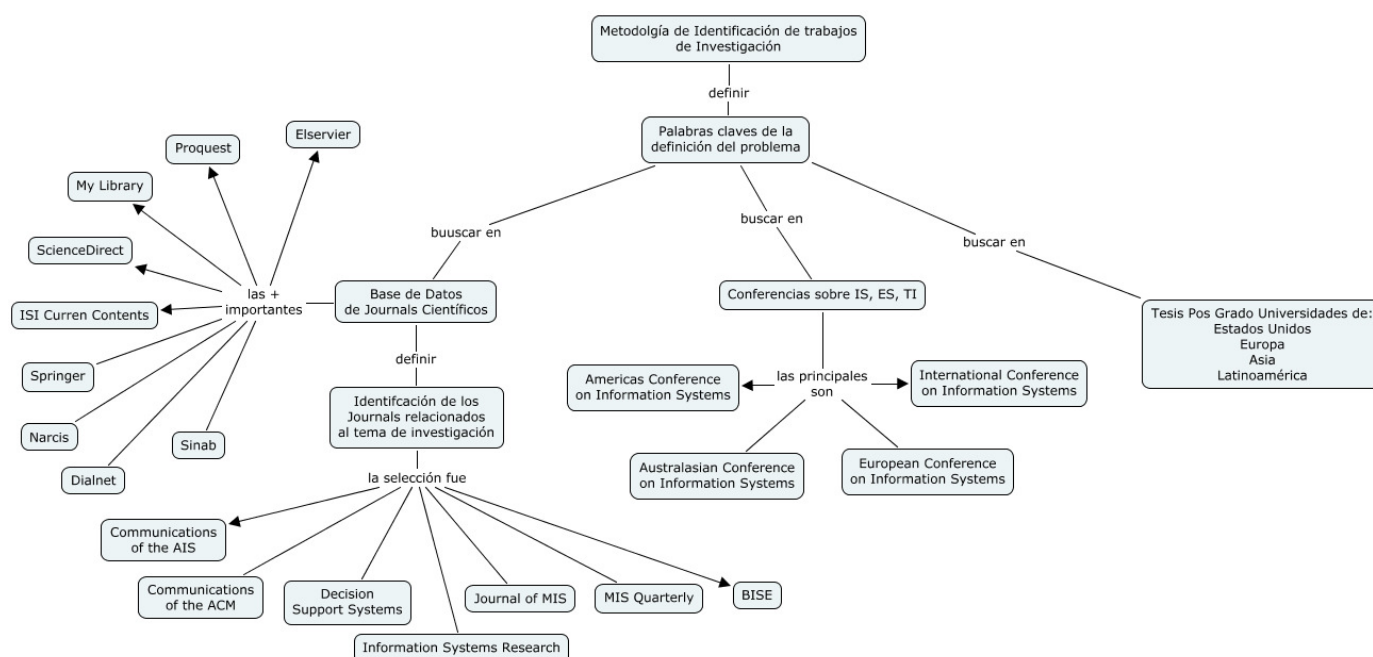


Figura 8. Proceso de revisión de la literatura propuesto por Webster et al. (Adaptación propia)

El proceso se describe a continuación. Primero se identifican las palabras claves que componen la definición del problema, seguidamente se hace las respectivas búsquedas en: Bases de Datos de Journals Científicos reconocidos como: Elsevier, Proquest, ScienceDirect, Springer. Luego se refina la búsqueda identificando los journals referentes al tema de

investigación, como Communications of the AIS, Communications of the ACM, Decision and Support Systems, Information Systems Research, Journal of MIS, MIS Quarterly, etc.

Luego, las palabras claves identificadas se utilizan como criterios de búsqueda en las páginas de las principales conferencias sobre el área de investigación: Americas Conference on Information Systems, Australasian Conference on Information Systems, European Conference on Information Systems, International Conference on Information Systems.

Seguidamente se buscaron las soluciones planteadas al problema por otros investigadores en trabajos de Tesis. De esta manera se utilizaron las palabras de referencia de búsqueda en Universidades y páginas afines que tenían a disposición Tesis de Post Grado en Estados Unidos, Europa, Asia y Latinoamérica.

CAPÍTULO 4: MODELO DE PROCESOS

Después de realizada la revisión de la literatura no se encontró un modelo que integre todos los aspectos necesarios para implementar un servicio de testing de aplicaciones. La estructura organizacional y un modelo de pruebas definido deben ser los componentes básicos para una propuesta de este tipo.

4.1. Diseño del Modelo

El modelo propuesto que se enmarca en la ontología propuesta por Sánchez (Sánchez et al. 2009) es considerado como un modelo de procesos, el cual es “...una representación abstracta de un proceso. Representa una descripción de un proceso desde una perspectiva particular.” (Sommerville 2016)

Así pues, Sánchez considera al modelo de procesos como un “modelo original” y al proceso resultante de su aplicación como “modelo como copia”. La figura 8 muestra la ontología y el lugar que ocupa un modelo de procesos dentro de su clasificación.

Teniendo la premisa que el servicio de testing de Software se regirá bajo el modelo del Outsourcing y que las otras etapas de desarrollo como el análisis, el diseño y la construcción son ejecutadas por el cliente o por un proveedor de desarrollo; las funciones de gestión y monitoreo del proyecto software en su integridad (abarcando todo el ciclo de vida) son realizadas por uno de estos dos actores.

Siendo así, el modelo contempla los límites entre la gestión del cliente/proveedor de desarrollo y el Servicio de testing. La gestión del cliente generalmente abarca todo el ciclo de vida, desde la captura de requerimientos hasta su puesta a producción, contempla todos los aspectos del negocio y es por esto por lo que en un escenario de outsourcing tienen personal, con perfil de gestión generalmente, que permanentemente vigila la ejecución y éxito de los proyectos.

4.2. Construcción del Modelo

En la figura 9 se puede apreciar el modelo de procesos de primer nivel. Cada constructo tiene sus bases en el estudio de estándares y en la revisión de la literatura. Los elementos constituyentes del modelo son luego diagramados y su estructura y flujo son mostrados.

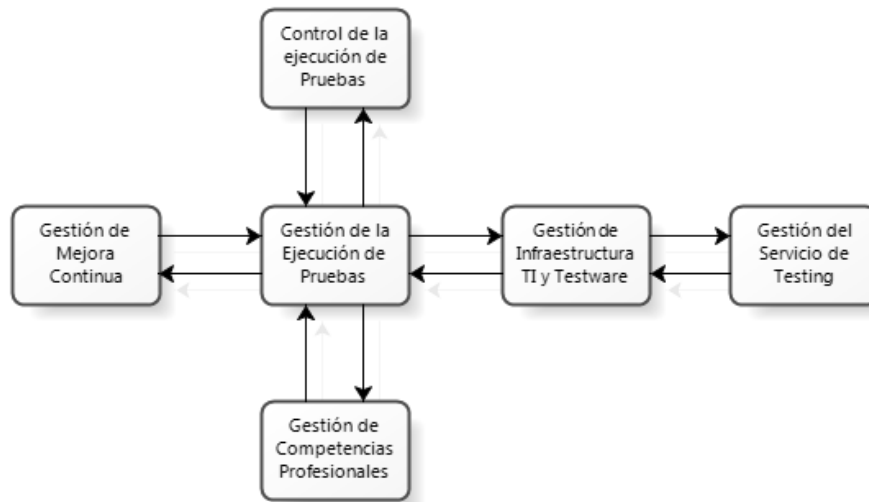


Figura 19. Componentes del modelo de procesos para un servicio de testing de aplicaciones

4.2.1. Gestión de la ejecución de pruebas:

Define cada uno de los procesos necesarios para proporcionar los servicios de pruebas de software. Este constructo tiene sus bases en las mejores prácticas, objetivos generales y específicos de los modelos CMMI V2 y TMMI (ISACA, 2022, TMMi Foundation, 2022), así como los lineamientos de gestión descritos en el PMBOK (Project Management Institute [PMI], 2021) para el tratamiento individual de los proyectos de testing, y de la cartera de proyectos.

En la figura 10 se puede observar el modelo de procesos de segundo nivel para el macroproceso de gestión de la ejecución de pruebas.

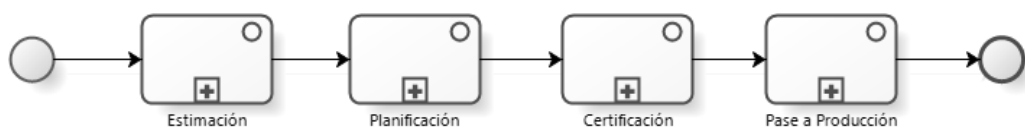


Figura 10. Modelo de procesos de segundo nivel del proceso de gestión de ejecución de pruebas

A continuación, se describe de manera general las actividades comprendidas en cada proceso:

- **Estimación.** Se recibe los entregables del proyecto que pasará por el Servicio de testing. Estos entregables deben comprender el código fuente, así como las especificaciones técnicas y funcionales del software.

Seguidamente los expertos analizan los requerimientos de prueba y estiman el esfuerzo que tomará ejecutar el testing.

- **Planificación.** Basado en la capacidad y cronograma de proyectos en curso, se planifica la ejecución de las pruebas y asignan los recursos adecuados.

El jefe de proyecto analiza el esfuerzo necesario para ejecutar las pruebas y asigna los recursos disponibles y con el conocimiento acerca de las aplicaciones a probar. Luego determina las fechas de inicio y fin para su ejecución, comprometiendo estas fechas hacia el cliente para el lanzamiento del sistema.

- **Certificación.** Abarca las actividades propiamente de la ejecución del testing.

Basado en el análisis de los requerimientos funcionales y no funcionales, y en la documentación asociada al proyecto de software, como son los casos de prueba, los planes técnicos y funcionales, y el mapeo del impacto en componentes de producción, los testers ejecutan las pruebas identificando y reportando defectos y riesgos de las aplicaciones probadas.

Los defectos reportados son subsanados por el equipo de desarrollo y las pruebas son ejecutadas nuevamente (teniendo en consideración el impacto y posibles nuevos errores) sobre los componentes adecuados.

Realizada esta actividad y comprobando el buen funcionamiento del aplicativo, el equipo de pruebas “certifica” la aplicación garantizando su buen funcionamiento en el ambiente de producción.

- **Pase a Producción.** Contempla las actividades implementación de las aplicaciones en el ambiente productivo, asegurando su buen funcionamiento e integración con los demás sistemas.

La gestión de este proceso implica a los principales actores durante el ciclo de vida. Representantes del equipo de desarrollo, de testing y operadores de sistemas del ambiente productivo del cliente se aúnan con el objetivo de implantar el aplicativo en el ambiente de producción garantizando la continuidad de las operaciones de los sistemas.

El flujo y caracterización de actividades de cada proceso constitutivo del macroproceso: Gestión de la ejecución de pruebas, se detallan en el Anexo 3.

4.2.2. Gestión del servicio de testing:

Define cada uno de los procesos necesarios para gestionar los servicios de pruebas de software. Este constructo tiene sus bases en las mejores prácticas relativas a la gestión de servicios de TI de ITIL, y CMMI SVC, identificando el nivel de implantación de este tipo de líneas de actuación para la ejecución del servicio de testing. Siguiendo estos principios se definen los siguientes procesos:

- **Gestión de la demanda.** Se basa en el estudio del histórico de proyectos y estimación por tipología de actividades y tareas a ejecutar.
- **Gestión de cronograma.** Cuantificación de los lapsos de tiempo requeridos para las distintas tareas, referidos a una unidad de esfuerzo.

- **Estudio de la capacidad.** Cálculo del volumen de actividades asumibles para un equipo específico (técnico y humano), desglosado como máximo diario y máximo mensual.
- **Punto de eficiencia.** Dimensionamiento óptimo para una carga de trabajo estimada, fundamentada en los siguientes conceptos: demanda, capacidad, cronograma y tipo de servicio. Asignación y reserva de recursos humanos y técnicos.

4.2.3. Gestión de mejora continua:

Con el propósito de implementar un programa de mejora constante en toda la organización (CMMI V2.0, ITIL 2019, PMBOK (Project Management Institute [PMI], 2021), define procedimientos para la evaluación continua de los procesos y las personas que integran el servicio de testing.

4.2.4. Gestión de infraestructura y testware:

Especifica la tecnología requerida para el funcionamiento del servicio de testing: instrumentos para gestionar el conocimiento y testware (tecnología vinculada a la realización y control del testing), herramientas esenciales para desarrollar los servicios y los procesos de prueba, herramientas que faciliten la valoración de los procesos y las competencias, entre otros.

4.2.5. Control de la ejecución de pruebas:

Define los elementos que se han de tener en cuenta para llevar a cabo el control del rendimiento del Servicio de testing.

Basado en las mejores prácticas sobre control de proyectos de PMBOK (Project Management Institute [PMI], 2021) y el control y monitoreo de la entrega de servicios de ITIL 4 (AXELOS, 2019), se definen:

- **Acuerdos de Niveles de Servicio (ANS):** Se definen los acuerdos de niveles de servicio que serán firmados por ambas partes y los umbrales mínimos y máximos que pueden alcanzar. Son de carácter operativo y se enfocan en el día a día.

- **Tipología de ANS:** Se clasifican los acuerdos para los procesos a ser medidos, como el de Ejecución de Pruebas, cuyo objetivo será controlar la ejecución y efectividad de estas.

Del mismo modo, se definirán los acuerdos para medir la satisfacción del cliente acerca de la gestión de servicio. Son de carácter ejecutivo para la evaluación del servicio.

4.2.6. Gestión de competencias profesionales:

Define la estructura profesional en el área de las pruebas de software. Este modelo permitirá la cualificación del personal del Servicio de testing.

Se delinean aquí los roles y sus funciones para el proceso de Gestión de ejecución de pruebas.

- **Gestor de pruebas:** toma de decisiones, supervisión coordinación, gestión de equipos de trabajo y seguimiento del cumplimiento de hitos y objetivos, elaboración de reportes a nivel ejecutivo.
- **Consultor de calidad:** control de entregables, consultoría, diseño y planificación de líneas de implantación, coordinación y seguimiento operativo.
- **Analista de pruebas:** diseño y planificación de pruebas, asignación y control de equipos de trabajo a nivel operativo, elaboración de entregables, ejecución de líneas de implantación, elaboración de reportes a nivel operativo.
- **Ingeniero de pruebas:** diseño y ejecución de pruebas, registro y seguimiento de defectos, coordinación de líneas de resolución de defectos.
- **Tester:** ejecución de pruebas y registro de defectos.

4.3. Hipótesis de Modelo Propuesto

Los nuevos modelos, procesos o métodos pueden ser evaluados en diversos aspectos. Moody (2003) propone un modelo de evaluación por medio del cual se evalúa si un modelo es exitoso o no por medio de su eficacia actual y su adopción en la práctica.

De esta manera la hipótesis planteada pretende comprobar esos dos elementos. Las hipótesis específicas están constituidas de tal manera que en conjunto deberían comprobar la hipótesis general que es mucho más amplia y abarcadora.

4.3.1. Hipótesis General

Un modelo de procesos para un servicio de testing de software influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones

4.3.2. Hipótesis Específicas

- La ejecución de pruebas influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.
- El servicio de testing influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.
- La mejora continua influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.
- La infraestructura y testware influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.
- El control de ejecución de pruebas influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.
- Las competencias profesionales influyen de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

4.4. Identificación de Variables

4.4.1. Variable Dependiente

X: Modelo de procesos para un servicio de testing de software

4.4.2. Variable Independiente

Y: Calidad del producto final

4.5. Operacionalización de Variables

4.5.1. Operacionalización de la variable modelo de procesos para un servicio de testing de software

Tabla 1. Operacionalización de la variable independiente

Variables	Definición Conceptual	Definición Operacional	Dimensiones	Indicadores	Escala	Valor final
Modelo de procesos para un servicio de testing de software	Un modelo de procesos para un servicio de testing de software es un marco de trabajo estructurado y definido que describe las etapas, actividades, roles y responsabilidades involucrados en la ejecución de pruebas de software. Este modelo proporciona directrices y mejores prácticas para planificar, gestionar y ejecutar pruebas de manera efectiva, con el objetivo de garantizar la calidad del producto final y la satisfacción del cliente.	El modelo se puede medir a partir de las siguientes dimensiones: la gestión de la ejecución de pruebas, la gestión del servicio de testing, la mejora continua, la infraestructura y testware, el control de la ejecución de pruebas, y el desarrollo de competencias profesionales, con el fin de asegurar que las pruebas se realicen de manera eficiente, cumpliendo con estándares y requisitos de calidad.	ejecución de pruebas	Nivel de estimación de esfuerzo Tiempo promedio de certificación de aplicación	1 - Totalmente en desacuerdo 2 - En desacuerdo 3 - Ni de acuerdo ni en desacuerdo 4 - De acuerdo 5 - Totalmente de acuerdo	Bajo Medio Alto
			servicio de testing	Nivel de cumplimiento de la demanda Nivel de eficiencia en la planificación de proyectos		
			mejora continua	Nivel de mejoras implementadas Impacto de las mejoras implementadas		
			infraestructura y testware	Nivel de disponibilidad de las herramientas de testing Nivel de Reutilización de Testware		
			control de ejecución de pruebas	Nivel de Cumplimiento de los ANS Nivel de Desviaciones del Plan de Pruebas		
			competencias profesionales	Nivel de Cumplimiento de Capacitación Nivel de Competencia del Personal de testing		

4.5.2. Operacionalización de la variable Calidad del producto final

Tabla 2. Operacionalización de la variable

Variables	Definición Conceptual	Definición Operacional	Dimensiones	Indicadores	Escala	Valor final
Calidad del producto final	La calidad del producto final se refiere a la medida en que un producto, en este contexto específico, un software que ha pasado por un servicio de testing, cumple con las expectativas y requisitos establecidos. Incluye la evaluación de aspectos funcionales, rendimiento, fiabilidad y otros atributos clave para determinar si el producto satisface las necesidades de los usuarios, opera sin fallos significativos y cumple con los estándares de calidad predefinidos. Una alta calidad del producto final implica que el software es efectivo, eficiente y confiable en su funcionamiento, lo que contribuye a la satisfacción del cliente y al logro de los objetivos del proyecto.	La calidad del producto final se puede medir a partir de las siguientes dimensiones: Requerimientos, Diseño y arquitectura, Desarrollo de software y Documentación	Calidad funcional del software	Nivel de cumplimiento de Requisitos Funcionales Nivel de defectos del producto final	1 - Totalmente en desacuerdo 2 - En desacuerdo 3 - Ni de acuerdo ni en desacuerdo 4 - De acuerdo 5 - Totalmente de acuerdo	Bajo Medio Alto
			Rendimiento y Eficiencia	Nivel de respuesta Nivel de consumo de recursos		
			Fiabilidad y Disponibilidad	Tiempo Promedio Entre Fallos (MTBF) Tasa de Disponibilidad del Sistema		
			Percepción del servicio entregado al cliente	Nivel de percepción		

CAPÍTULO 5: ESTUDIO DE CASO

El caso de estudio es una adaptación de la metodología planteada por Yin (Yin, 2018)

5.1. Diseño de la Investigación (Protocolo)

La organización de los equipos de testing en la empresa en estudio está conformada como sigue:

- Project manager of testing
- Project manager office (PMO)
- Testing team
 - Front End team
 - Facturación y Cobranzas Team
 - Regional Team
 - Aprovisionamiento y Aseguramiento Team
 - Movil Team

Cada equipo es dirigido por un team leader o líder de equipo, que es considerado un experto en temas de testing para la organización.

Las estructuras de los equipos de pruebas, así como las formas de trabajo no estaban delineadas inicialmente. Sus roles y responsabilidades eran definidos y dispuestos por cada equipo según la forma de trabajo que cada líder había aprendido en la ejecución de la práctica profesional. Cabe observar que cada team leader es un consultor senior o experto en materia de testing con más de 15 años de experiencia.

Inicialmente cada equipo seguía las pautas de gestión de cada líder en materia de: Estimaciones, Planificaciones, Ejecución y Automatización de Pruebas, Pases a Producción, Comunicaciones y Negociación.

5.2. Unidad de Análisis

En específico: Proyectos de la cartera del Servicio de Pruebas de Aplicaciones.

La organización en estudio es una empresa española, consultora de TI, especializada que brinda servicios de outsourcing de procesos de negocio, desarrollo de aplicaciones y aseguramiento de la calidad del software.

Fundada el año 2000 en España, cuenta en la actualidad con oficinas en ciudades como Estados Unidos, Francia, Suiza, Ucrania, Reino Unido, México, Brasil, Colombia, República Dominicana y su filial en Lima, Perú.

En Perú se entregan servicios para el mercado local y el extranjero. Tiene clientes en industrias diversas como: Banca, Telecomunicaciones, Gobierno, Retail, Seguros, etc. Los servicios atienden áreas de proceso de TI del cliente en los que este carece de personal experto con la experiencia necesaria para llevarla a cabo de manera exitosa.

El cliente de mayor cobertura y despliegue tanto en equipos de trabajo como en infraestructura tecnológica es una Empresa de Telecomunicaciones, líder en Perú. Para este cliente se realizan las pruebas de todos sus productos software con el objetivo de certificar cada aplicación que pase al ambiente productivo.

Se ha elegido para el caso de estudio y comprobación de la efectividad del modelo propuesto al área de certificación de aplicaciones para el cliente referido, por ser la unidad de negocio de mayor actividad y de la cual derivan los mayores ingresos de la consultora en estudio. Además, por ser la más representativa en extensión e infraestructura desplegada y tener un histórico de mediciones con el cual contrastar los resultados obtenidos de la implementación del modelo.

5.3. Población de Estudio

Empresa de consultoría de TI, específicamente las del rubro de testing de aplicaciones.

5.4. Tamaño de la Muestra

La muestra es el total de proyectos ejecutados durante el periodo de evaluación.

5.5. Selección de la Muestra

Se eligieron los proyectos de la cartera de medianos a grandes, según la clasificación existente; en los cuales se pueda observar los fenómenos del estudio.

5.6. Técnicas de Recolección de Datos

Las técnicas empleadas fueron: observación de los hechos y encuestas.

Cabe indicar que en el desarrollo de las operaciones de testing, cada equipo de pruebas tiene registro de sus labores, así como indicadores de efectividad, productividad entre otros. De la misma manera, existen acuerdos de niveles de servicio cuyo cálculo es regido por pautas establecidas en el contrato de testing. Estos registros de calidad serán contrastados contra los que obtenga del equipo piloto en la implementación del: modelo de procesos para un servicio de testing, propuesto.

Con el objetivo de probar la eficiencia del modelo propuesto se eligió el equipo de Facturación y Cobros como equipo piloto de la implementación. Este equipo cuenta con quince (15) miembros más un team leader.

Se reunió al equipo y se les hizo una presentación del nuevo modelo y sus pautas de implementación se especificó el alcance de este y lo que se pretendía lograr de su adopción. Se escogieron dos (2) testers senior para el apoyo de la implementación y difusión del nuevo modelo entre los miembros del equipo. siempre teniendo en cuenta que el team leader es el responsable del proceso de implantación y prueba de la efectividad del nuevo modelo. Estos dos testers senior, junto a su team leader, son los responsables de las principales actividades de gestión en la ejecución de los proyectos: estimación de esfuerzo y planificación del tiempo.

Para la puesta en marcha del modelo se eligieron seis (6) proyectos de corto, mediano y largo alcance. El criterio para su clasificación viene dado por el esfuerzo para su ejecución. De 5 a 20 Jornadas (corto alcance), de 20 a 50 Jornadas (mediano alcance) y de 50 Jornadas a más (largo alcance).

Además, se consideró para la evaluación, si los proyectos correspondían a nuevos desarrollos o a mantenimiento correctivos. Y también la criticidad y prioridad de estos. Se entiende por criticidad, a si la falla en producción o demora de lanzamiento del sistema afectaría de manera considerable las operaciones de cliente (en términos económicos y de satisfacción del cliente). Y por prioridad, a si el sistema debe ser lanzado con celeridad y si la ruta crítica viene dada por tiempos impostergables en el cronograma.

5.7. Análisis e Interpretación de la Información

La etapa de evaluación consistió en la toma de una encuesta para constatar la validez de las hipótesis. Junto a los resultados comparados (entre los datos históricos y los nuevos, producto de su aplicación a los proyectos piloto) se pretende demostrar facilidad de uso, la utilidad y la intención de uso del modelo (Moody 2003).

La elaboración y adaptación de encuestas a ser aplicada a los equipos de prueba participantes, jefes de proyecto responsables, gerentes del cliente y del servicio de testing, y usuarios finales del sistema está en proceso. En el Anexo 2 se puede apreciar las encuestas aplicadas a los miembros de los equipos participantes en la adopción del nuevo modelo.

Esta encuesta está basada en la encuesta de Moody (2003) que a su vez se basa en la elaborada por Davis et al., (1989) y que ha sido adaptada hacia el dominio de los sistemas de información. Estas encuestas de referencia han sido validadas en investigaciones previas que las utilizan como modelo.

Cada ítem de la encuesta será medido utilizando una escala de Likert de cinco (5) puntos y serán distribuidos de manera aleatoria, para reducir respuestas monótonas a ítems que miden las mismas categorías. Las escalas de los ítems van desde: Muy en desacuerdo / Muy inadecuado –hasta-- Muy de acuerdo / Muy adecuado.

Las técnicas estadísticas que se utilizarán para la evaluación de la encuesta son:

Análisis de correlación: Para evaluar la validez de los constructos y descartar aquellos que tengan poca convergencia en los resultados. Serán desechados o reformulados todos aquellos valores menores a 0.35, los cuales según la literatura no servirán para medir lo que se desea.

Coeficiente de confiabilidad - Cronbach Alpha: Para evaluar la confiabilidad de los resultados de las encuestas. En la literatura un valor mayor a 0.7 es considerado aceptable.

5.8. Resultados

Las tareas para la culminación de la fase de recolección, comparación de los datos históricos de calidad de proyectos de la unidad de análisis (bitácora de indicadores y mediciones realizadas con anterioridad a los equipos de prueba) y análisis de resultados de la encuesta (contra los que se comprobó la eficiencia del nuevo modelo) son las siguientes:

- Fase de recolección de datos
- Comparación de Resultados
- Toma de Cuestionarios
- Análisis de Estadístico
- Interpretación de Resultados

CAPÍTULO 6: RESULTADOS

6.1. Resultados descriptivos

6.1.1. Resultados descriptivos de la variable modelo de procesos para un servicio de testing

Dimensión 1: Gestión de la ejecución de pruebas

Tabla 3. El proceso de Estimación ha mejorado la planificación y el uso de recursos en nuestro servicio de testing.

El proceso de Estimación ha mejorado la planificación y el uso de recursos en nuestro servicio de Testing.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	1	7,1	7,1	7,1
	De acuerdo	4	28,6	28,6	35,7
	Totalmente de acuerdo	9	64,3	64,3	100,0
	Total	14	100,0	100,0	

El proceso de Estimación ha mejorado la planificación y el uso de recursos en nuestro servicio de Testing.

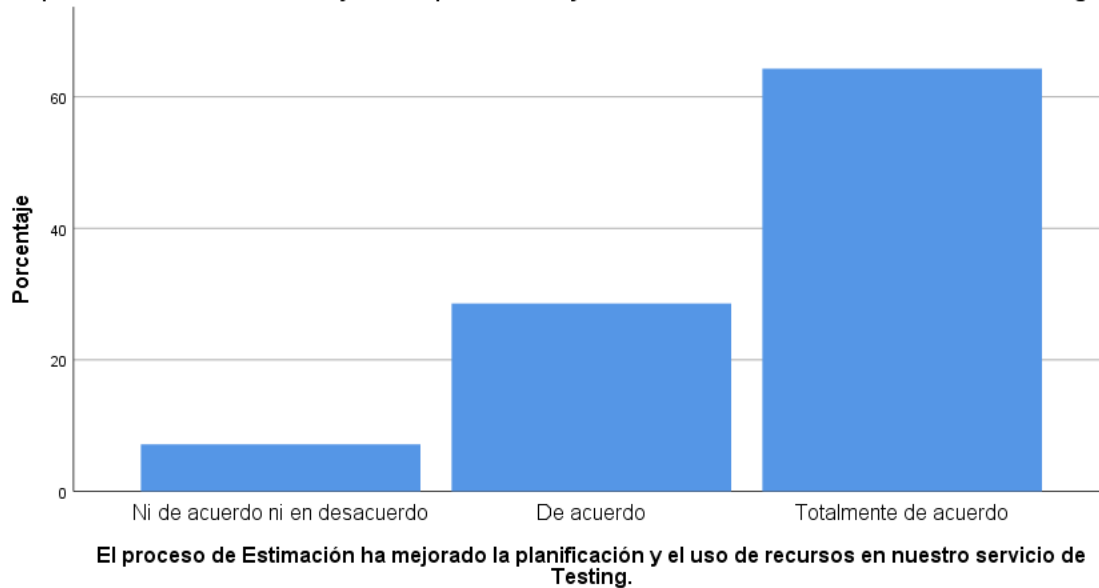


Figura 11. El proceso de estimación ha mejorado la planificación y el uso de recursos en nuestro servicio de testing.

Tabla 4. La Certificación de las aplicaciones probadas garantiza un buen funcionamiento en el ambiente de producción.

		La Certificación de las aplicaciones probadas garantiza un buen funcionamiento en el ambiente de producción.			
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	1	7,1	7,1	7,1
	De acuerdo	4	28,6	28,6	35,7
	Totalmente de acuerdo	9	64,3	64,3	100,0
	Total	14	100,0	100,0	

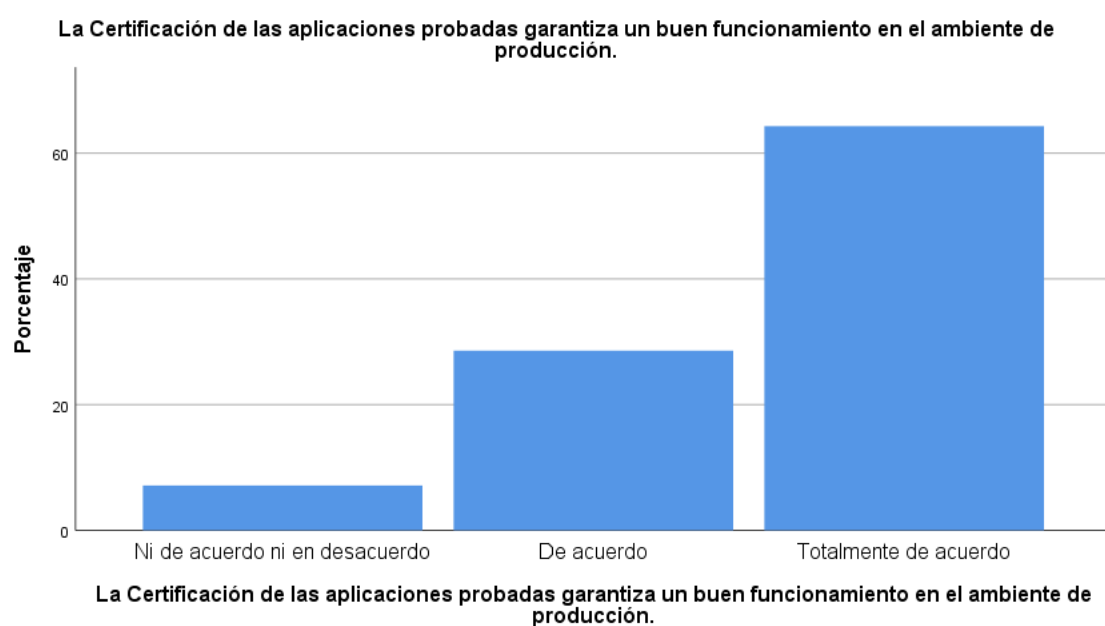


Figura 12. La Certificación de las aplicaciones probadas garantiza un buen funcionamiento en el ambiente de producción.

Tabla 5. La implementación de aplicaciones en el ambiente productivo se realiza de manera efectiva.

La implementación de aplicaciones en el ambiente productivo se realiza de manera efectiva.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	De acuerdo	4	28,6	28,6	28,6
	Totalmente de acuerdo	10	71,4	71,4	100,0
	Total	14	100,0	100,0	

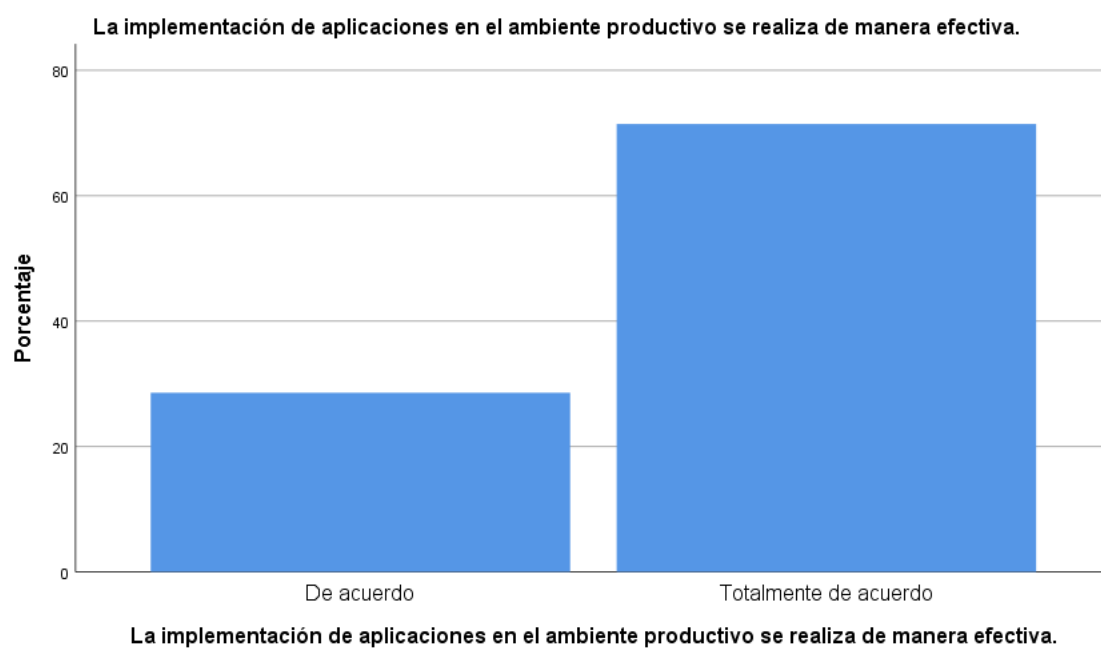


Figura 13. La implementación de aplicaciones en el ambiente productivo se realiza de manera efectiva.

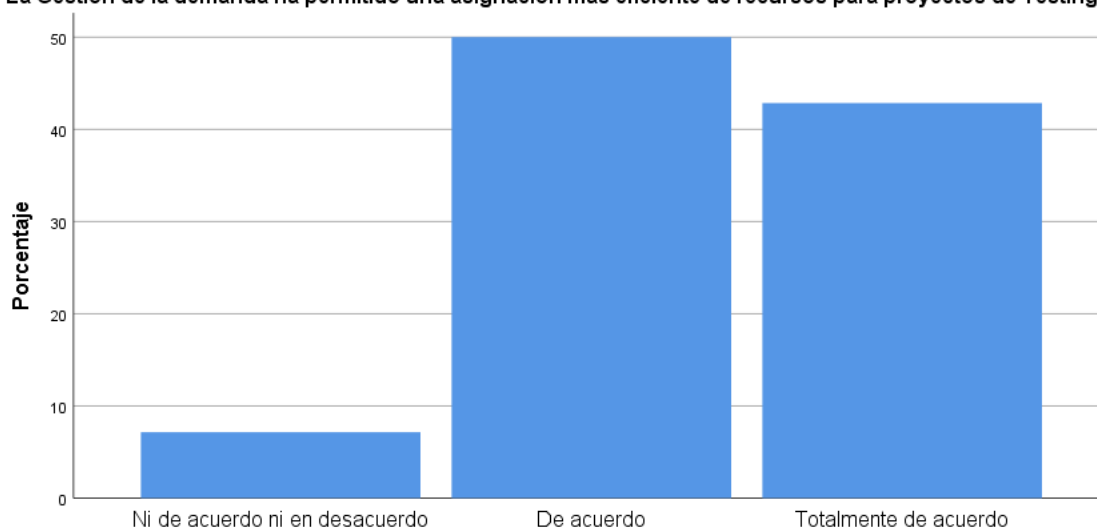
Dimensión 2: Gestión del servicio de testing

Tabla 6. La gestión de la demanda ha permitido una asignación más eficiente de recursos para proyectos de testing.

La Gestión de la demanda ha permitido una asignación más eficiente de recursos para proyectos de Testing.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	1	7,1	7,1	7,1
	De acuerdo	7	50,0	50,0	57,1
	Totalmente de acuerdo	6	42,9	42,9	100,0
	Total	14	100,0	100,0	

La Gestión de la demanda ha permitido una asignación más eficiente de recursos para proyectos de Testing.



La Gestión de la demanda ha permitido una asignación más eficiente de recursos para proyectos de Testing.

Figura 14. La gestión de la demanda ha permitido una asignación más eficiente de recursos para proyectos de testing.

Tabla 7. La gestión de cronograma ha mejorado la planificación temporal de nuestras actividades de testing.

La Gestión de cronograma ha mejorado la planificación temporal de nuestras actividades de Testing.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	De acuerdo	7	50,0	50,0	50,0
	Totalmente de acuerdo	7	50,0	50,0	100,0
	Total	14	100,0	100,0	

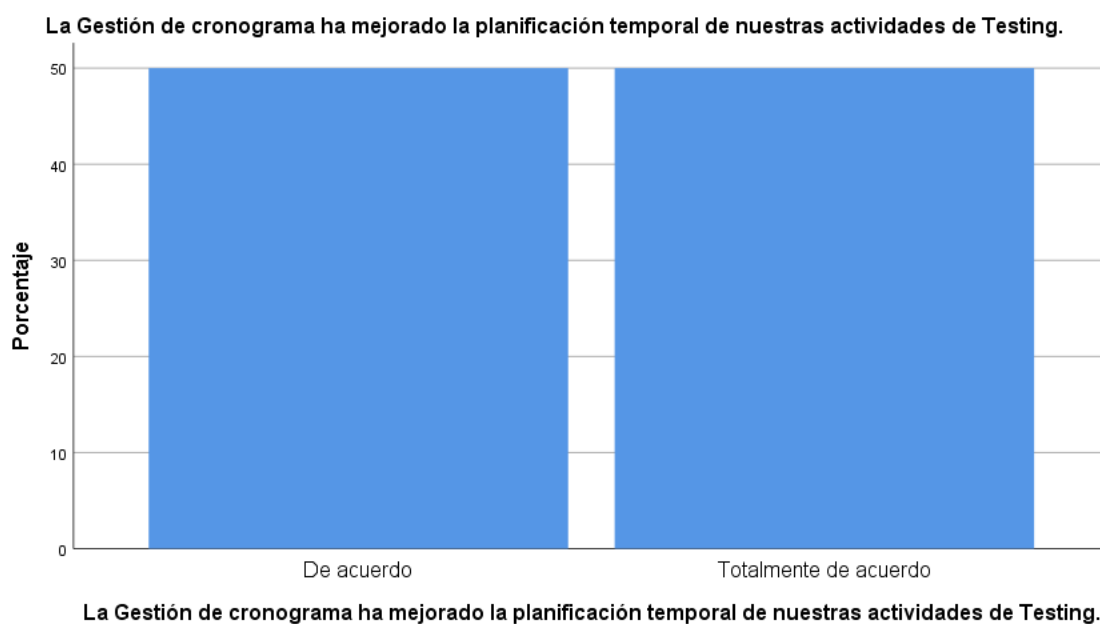


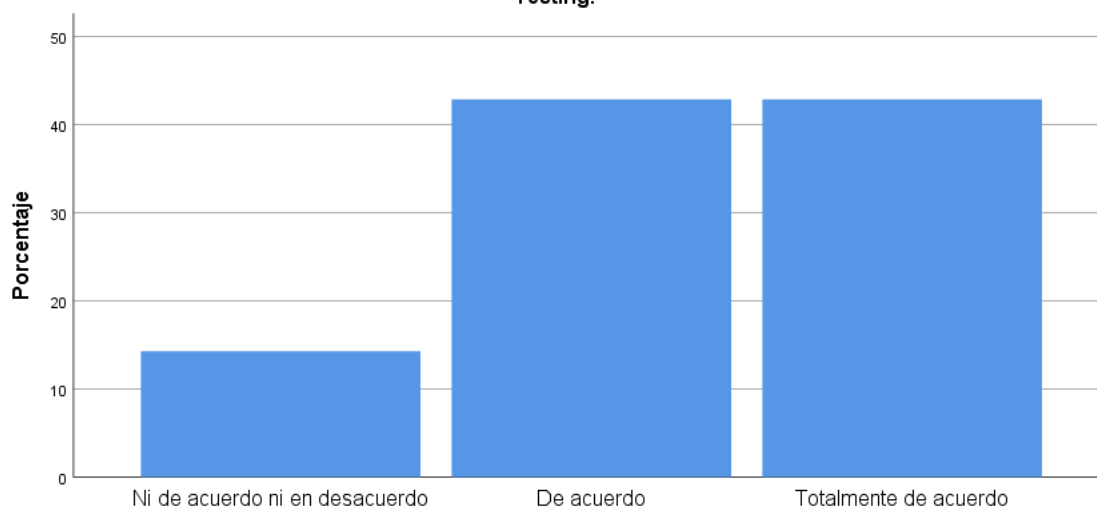
Figura 15. La gestión de cronograma ha mejorado la planificación temporal de nuestras actividades de testing.

Tabla 8. El punto de eficiencia ha ayudado a dimensionar de manera óptima nuestra carga de trabajo en proyectos de testing.

El Punto de eficiencia ha ayudado a dimensionar de manera óptima nuestra carga de trabajo en proyectos de Testing.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	2	14,3	14,3	14,3
	De acuerdo	6	42,9	42,9	57,1
	Totalmente de acuerdo	6	42,9	42,9	100,0
	Total	14	100,0	100,0	

El Punto de eficiencia ha ayudado a dimensionar de manera óptima nuestra carga de trabajo en proyectos de Testing.



El Punto de eficiencia ha ayudado a dimensionar de manera óptima nuestra carga de trabajo en proyectos de Testing.

Figura 16. El punto de eficiencia ha ayudado a dimensionar de manera óptima nuestra carga de trabajo en proyectos de testing.

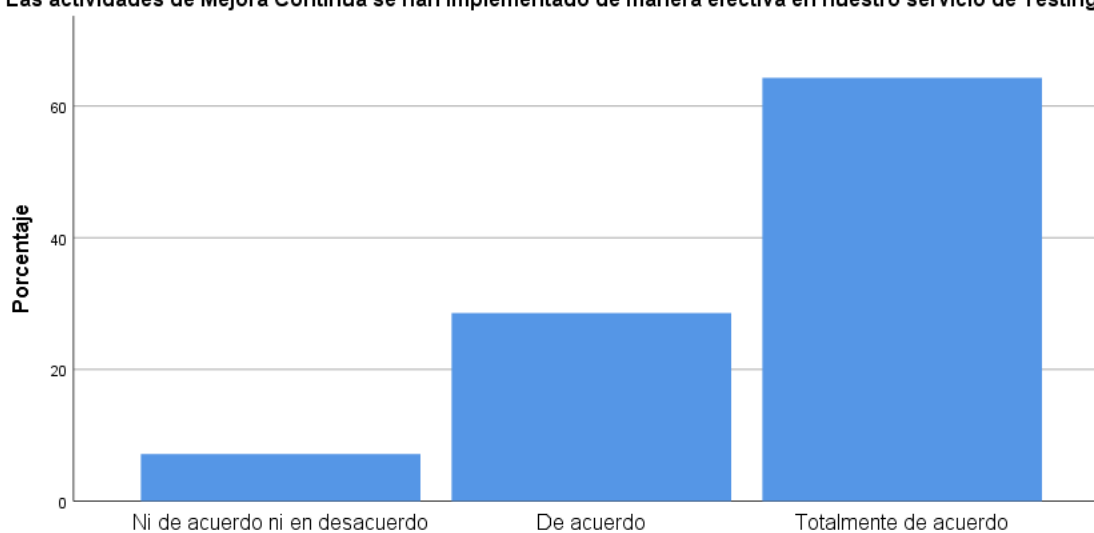
Dimensión 3: Gestión de mejora continua

Tabla 9. Las actividades de Mejora Continua se han implementado de manera efectiva en nuestro servicio de testing.

Las actividades de Mejora Continua se han implementado de manera efectiva en nuestro servicio de Testing.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	1	7,1	7,1	7,1
	De acuerdo	4	28,6	28,6	35,7
	Totalmente de acuerdo	9	64,3	64,3	100,0
	Total	14	100,0	100,0	

Las actividades de Mejora Continua se han implementado de manera efectiva en nuestro servicio de Testing.



Las actividades de Mejora Continua se han implementado de manera efectiva en nuestro servicio de Testing.

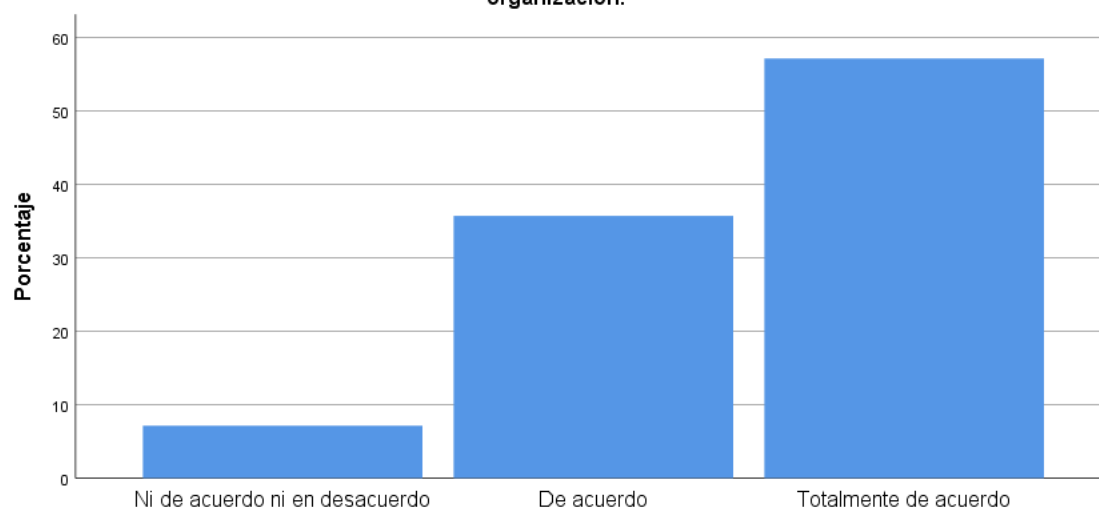
Figura 17. Las actividades de mejora continua se han implementado de manera efectiva en nuestro servicio de testing.

Tabla 10. La evaluación continua de procesos y personas ha contribuido a un programa de mejora continua en nuestra organización.

La evaluación continua de procesos y personas ha contribuido a un programa de mejora continua en nuestra organización.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	1	7,1	7,1	7,1
	De acuerdo	5	35,7	35,7	42,9
	Totalmente de acuerdo	8	57,1	57,1	100,0
	Total	14	100,0	100,0	

La evaluación continua de procesos y personas ha contribuido a un programa de mejora continua en nuestra organización.



La evaluación continua de procesos y personas ha contribuido a un programa de mejora continua en nuestra organización.

Figura 18. La evaluación continua de procesos y personas ha contribuido a un programa de mejora continua en nuestra organización.

Tabla 11. Participa activamente en la implementación de acciones de mejora continua en su área de trabajo desde la adopción del nuevo modelo.

Participa activamente en la implementación de acciones de mejora continua en su área de trabajo desde la adopción del nuevo modelo.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	1	7,1	7,1	7,1
	De acuerdo	10	71,4	71,4	78,6
	Totalmente de acuerdo	3	21,4	21,4	100,0
	Total	14	100,0	100,0	

Participa activamente en la implementación de acciones de mejora continua en su área de trabajo desde la adopción del nuevo modelo.

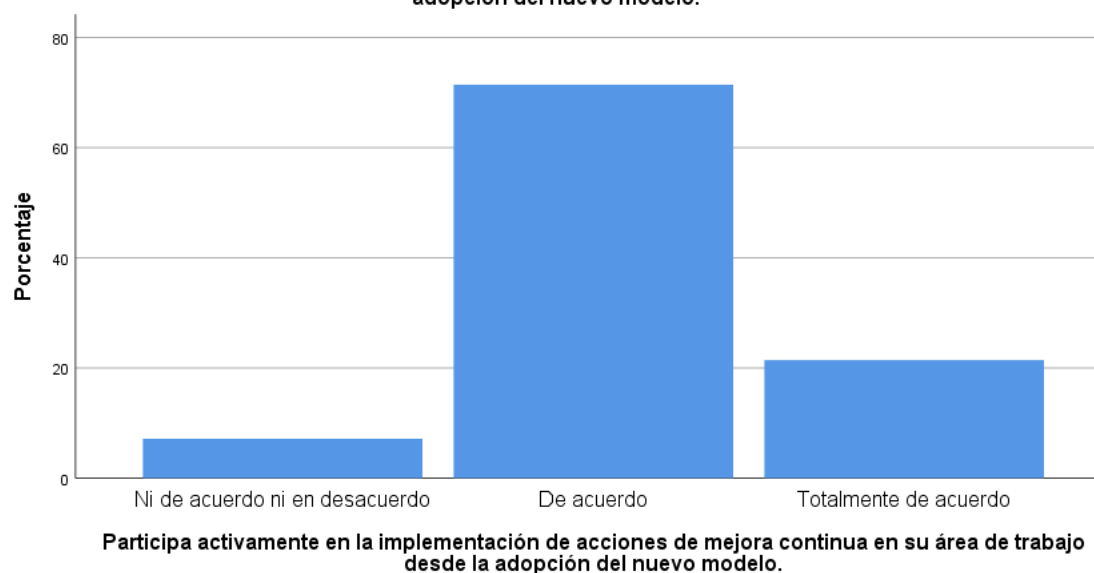


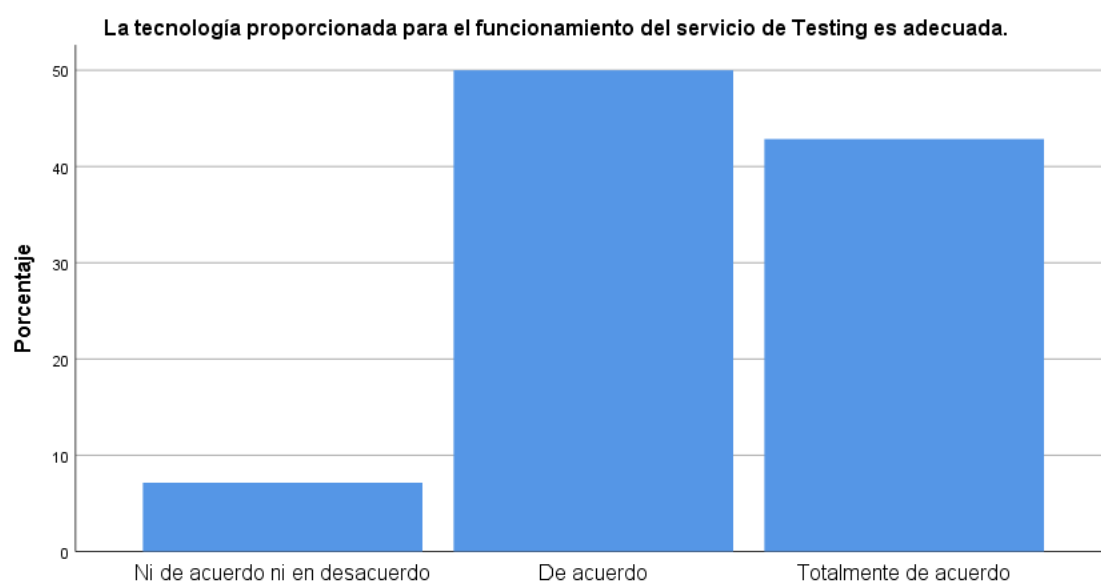
Figura 19. Participa activamente en la implementación de acciones de mejora continua en su área de trabajo desde la adopción del nuevo modelo.

Dimensión 4: Gestión de infraestructura y testware

Tabla 12. La tecnología proporcionada para el funcionamiento del servicio de testing es adecuada.

La tecnología proporcionada para el funcionamiento del servicio de Testing es adecuada.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	1	7,1	7,1	7,1
	De acuerdo	7	50,0	50,0	57,1
	Totalmente de acuerdo	6	42,9	42,9	100,0
	Total	14	100,0	100,0	



La tecnología proporcionada para el funcionamiento del servicio de Testing es adecuada.

Figura 20. La tecnología proporcionada para el funcionamiento del servicio de testing es adecuada.

Tabla 13. Las herramientas de gestión de conocimiento y testware son útiles y eficientes en nuestro servicio de testing.

Las herramientas de gestión de conocimiento y testware son útiles y eficientes en nuestro servicio de Testing.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	3	21,4	21,4	21,4
	De acuerdo	5	35,7	35,7	57,1
	Totalmente de acuerdo	6	42,9	42,9	100,0
	Total	14	100,0	100,0	

Las herramientas de gestión de conocimiento y testware son útiles y eficientes en nuestro servicio de Testing.

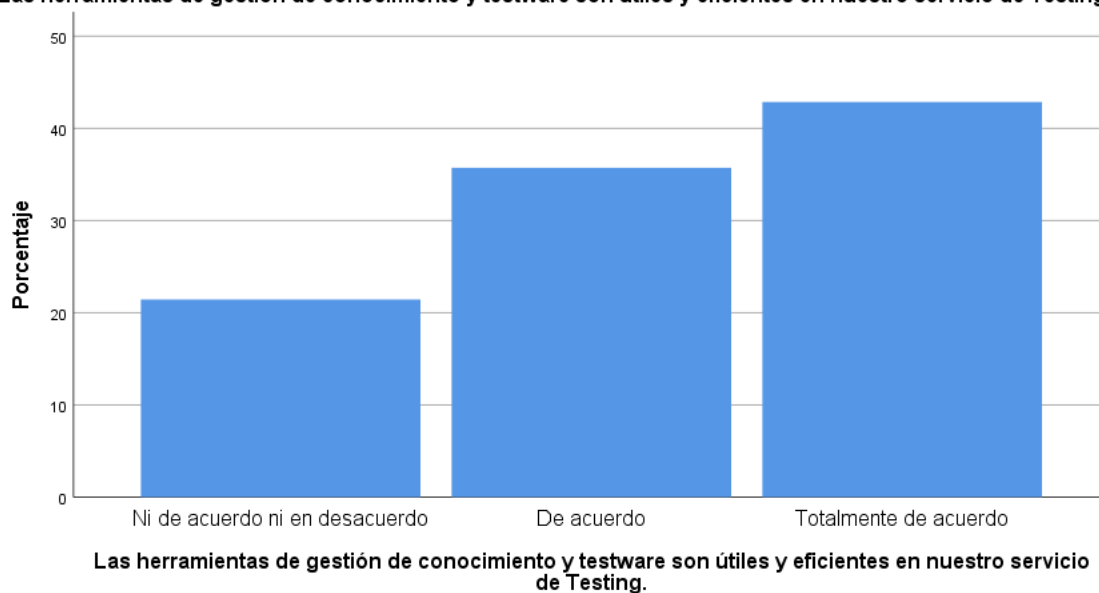


Figura 21. Las herramientas de gestión de conocimiento y testware son útiles y eficientes en nuestro servicio de testing.

Tabla 14. Las herramientas de gestión de conocimiento y testware han mejorado su eficiencia en el proceso de testing.

Las herramientas de gestión de conocimiento y testware han mejorado su eficiencia en el proceso de Testing.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	1	7,1	7,1	7,1
	De acuerdo	7	50,0	50,0	57,1
	Totalmente de acuerdo	6	42,9	42,9	100,0
	Total	14	100,0	100,0	

Las herramientas de gestión de conocimiento y testware han mejorado su eficiencia en el proceso de Testing.

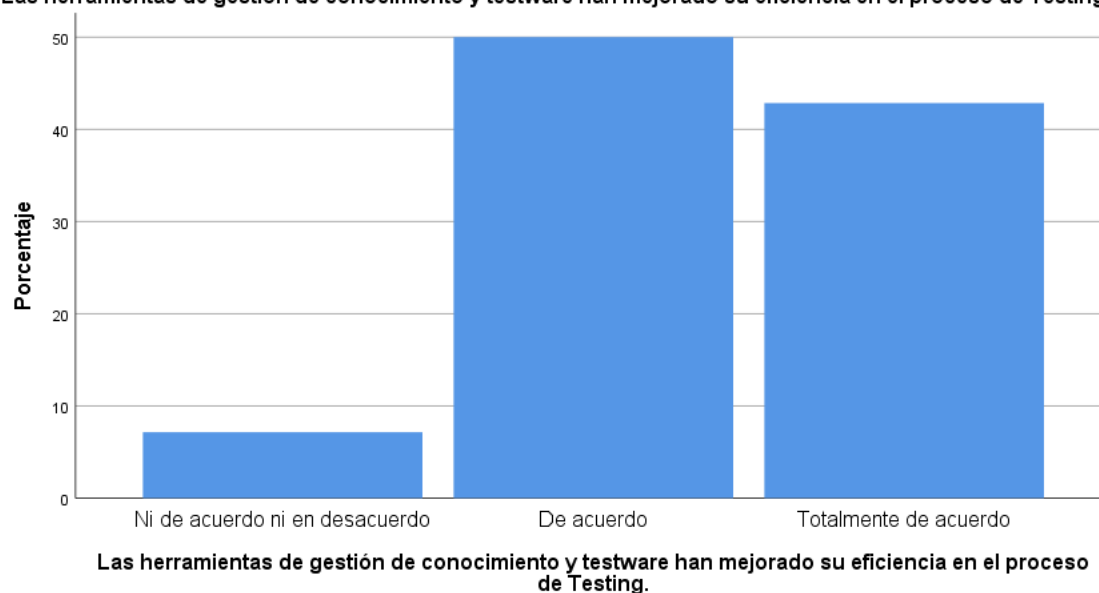


Figura 22. Las herramientas de gestión de conocimiento y testware han mejorado su eficiencia en el proceso de testing.

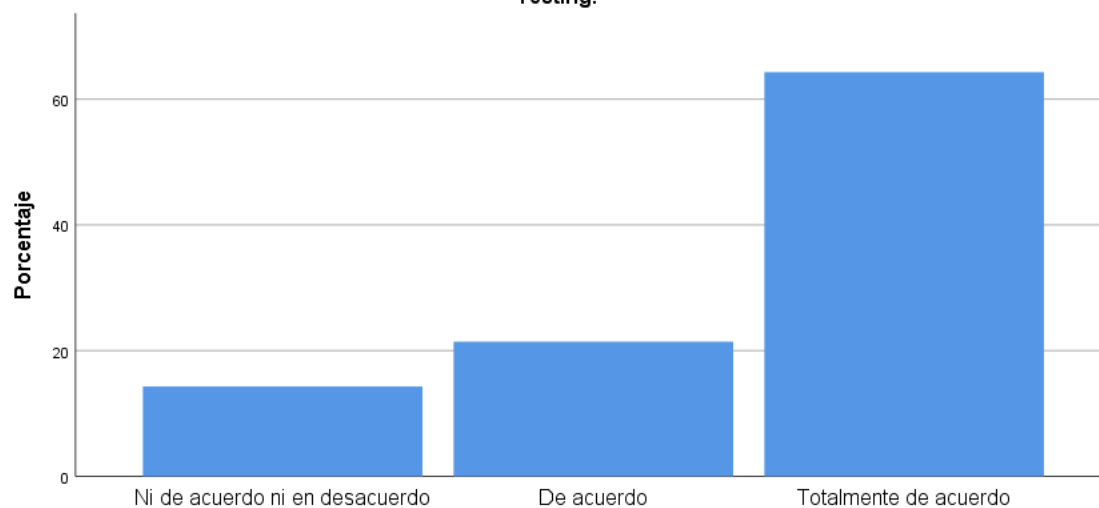
Dimensión 5: Control de la Ejecución de Pruebas

Tabla 15. Los acuerdos de niveles de servicio (ANS) han mejorado el control del rendimiento en nuestro servicio de testing.

Los Acuerdos de Niveles de Servicio (ANS) han mejorado el control del rendimiento en nuestro servicio de Testing.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	2	14,3	14,3	14,3
	De acuerdo	3	21,4	21,4	35,7
	Totalmente de acuerdo	9	64,3	64,3	100,0
	Total	14	100,0	100,0	

Los Acuerdos de Niveles de Servicio (ANS) han mejorado el control del rendimiento en nuestro servicio de Testing.



Los Acuerdos de Niveles de Servicio (ANS) han mejorado el control del rendimiento en nuestro servicio de Testing.

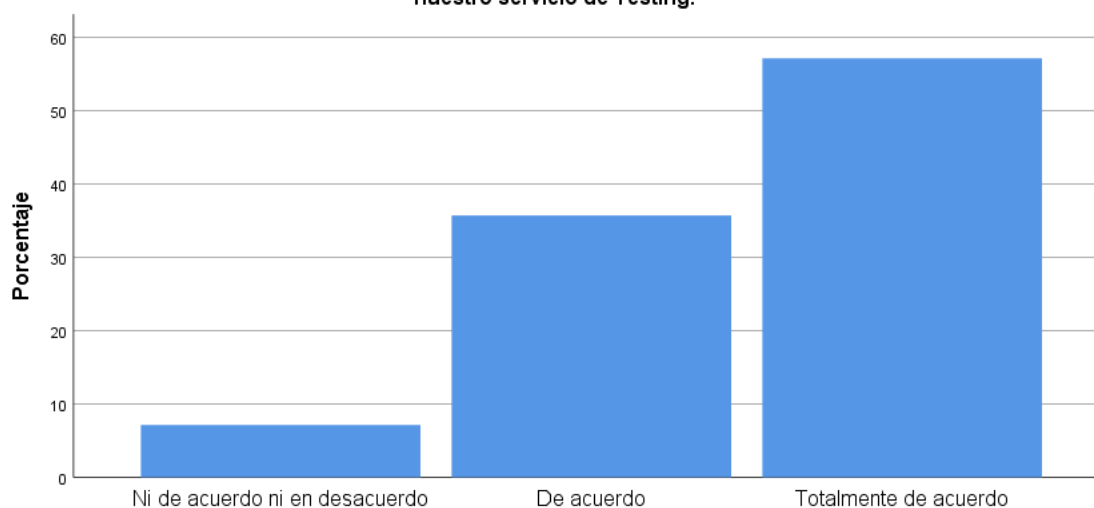
Figura 23. Los acuerdos de niveles de servicio (ANS) han mejorado el control del rendimiento en nuestro servicio de testing.

Tabla 16. Los tipos de ANS establecidos son efectivos para controlar la ejecución y efectividad de las pruebas en nuestro servicio de testing.

Los Tipos de ANS establecidos son efectivos para controlar la ejecución y efectividad de las pruebas en nuestro servicio de Testing.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	1	7,1	7,1	7,1
	De acuerdo	5	35,7	35,7	42,9
	Totalmente de acuerdo	8	57,1	57,1	100,0
	Total	14	100,0	100,0	

Los Tipos de ANS establecidos son efectivos para controlar la ejecución y efectividad de las pruebas en nuestro servicio de Testing.



Los Tipos de ANS establecidos son efectivos para controlar la ejecución y efectividad de las pruebas en nuestro servicio de Testing.

Figura 24. Los tipos de ANS establecidos son efectivos para controlar la ejecución y efectividad de las pruebas en nuestro servicio de testing.

Tabla 17. Los Tipos de ANS han sido efectivos para controlar la ejecución y la efectividad de las pruebas en su proyecto.

Los Tipos de ANS han sido efectivos para controlar la ejecución y la efectividad de las pruebas en su proyecto

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	3	21,4	21,4	21,4
	De acuerdo	5	35,7	35,7	57,1
	Totalmente de acuerdo	6	42,9	42,9	100,0
	Total	14	100,0	100,0	

Los Tipos de ANS han sido efectivos para controlar la ejecución y la efectividad de las pruebas en su proyecto

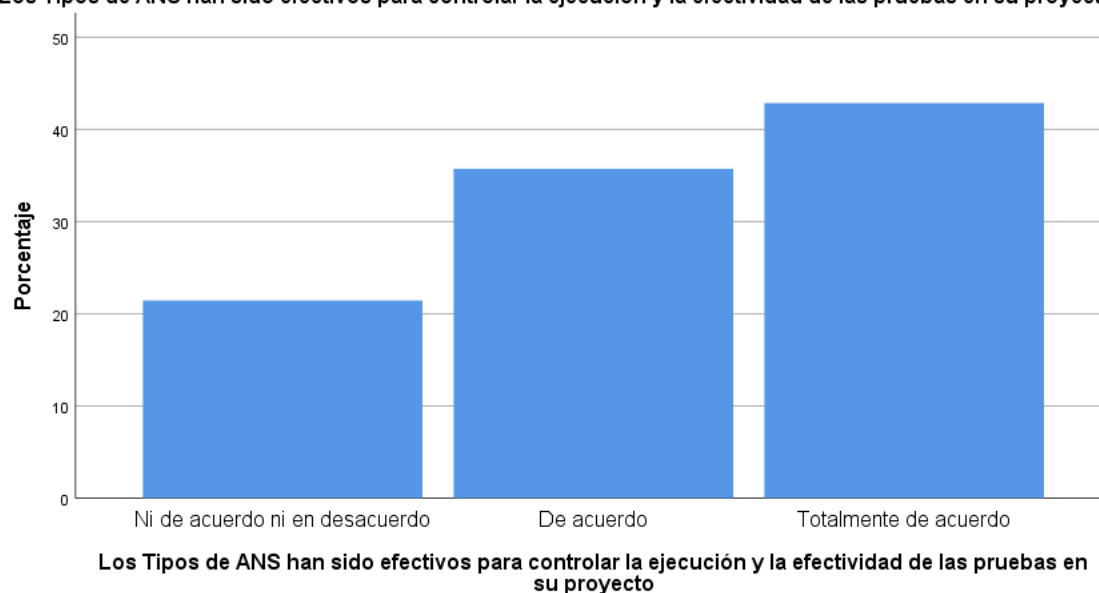


Figura 25. Los Tipos de ANS han sido efectivos para controlar la ejecución y la efectividad de las pruebas en su proyecto.

Dimensión 6: Gestión de Competencias Profesionales

Tabla 18. La estructura profesional definida para el área de pruebas de software ha mejorado la cualificación del personal en nuestro servicio de testing.

La estructura profesional definida para el área de pruebas de software ha mejorado la cualificación del personal en nuestro servicio de Testing.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	1	7,1	7,1	7,1
	De acuerdo	10	71,4	71,4	78,6
	Totalmente de acuerdo	3	21,4	21,4	100,0
	Total	14	100,0	100,0	

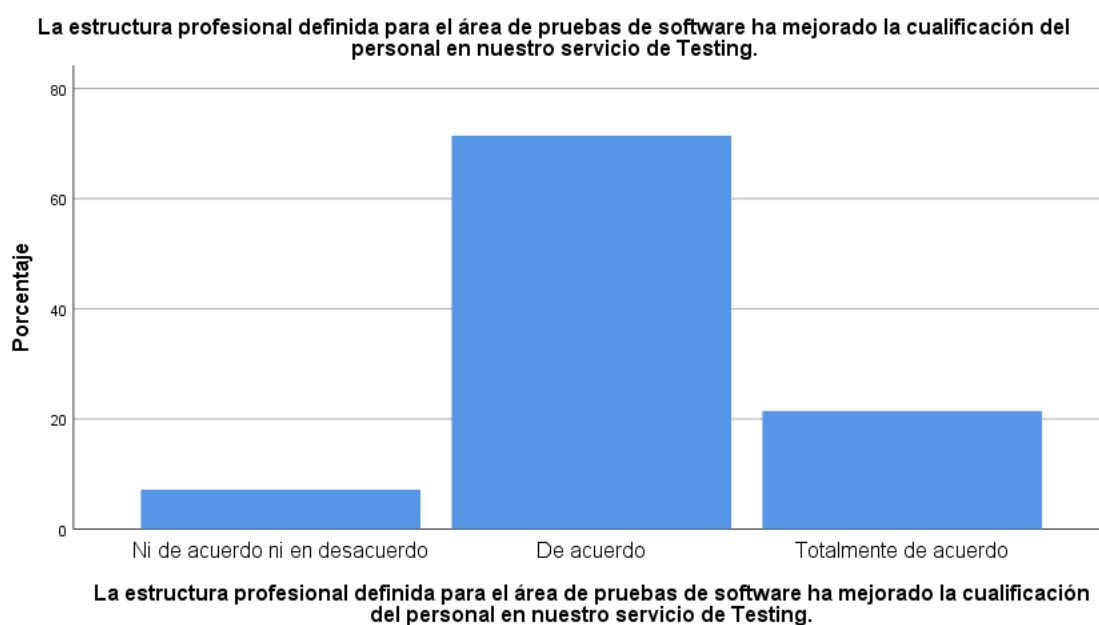


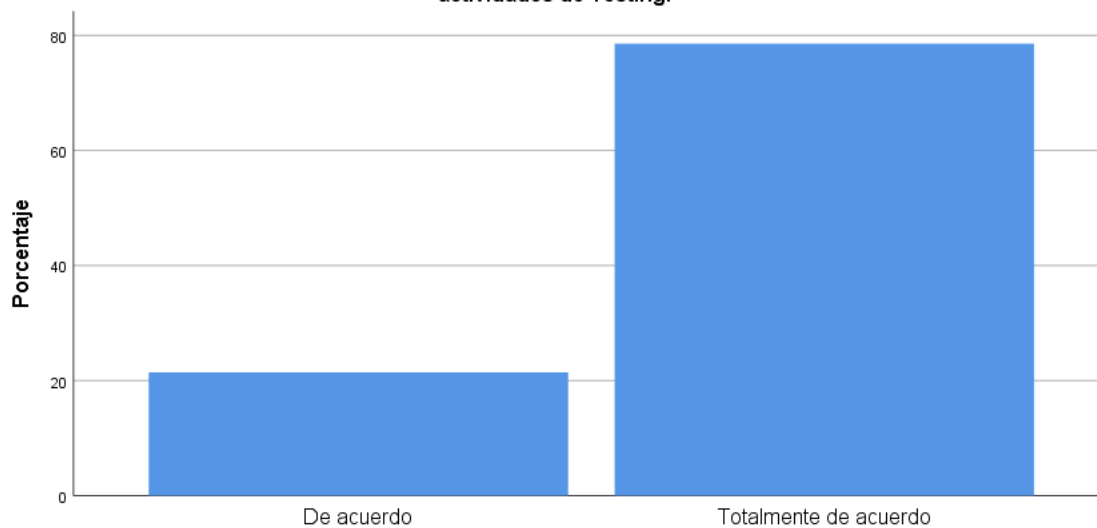
Figura 26. La estructura profesional definida para el área de pruebas de software ha mejorado la cualificación del personal en nuestro servicio de testing.

Tabla 19. La definición de roles y funciones en el proceso de Gestión de ejecución de pruebas ha facilitado nuestras actividades de testing.

La definición de roles y funciones en el proceso de Gestión de Ejecución de Pruebas ha facilitado nuestras actividades de Testing.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	De acuerdo	3	21,4	21,4	21,4
	Totalmente de acuerdo	11	78,6	78,6	100,0
	Total	14	100,0	100,0	

La definición de roles y funciones en el proceso de Gestión de Ejecución de Pruebas ha facilitado nuestras actividades de Testing.



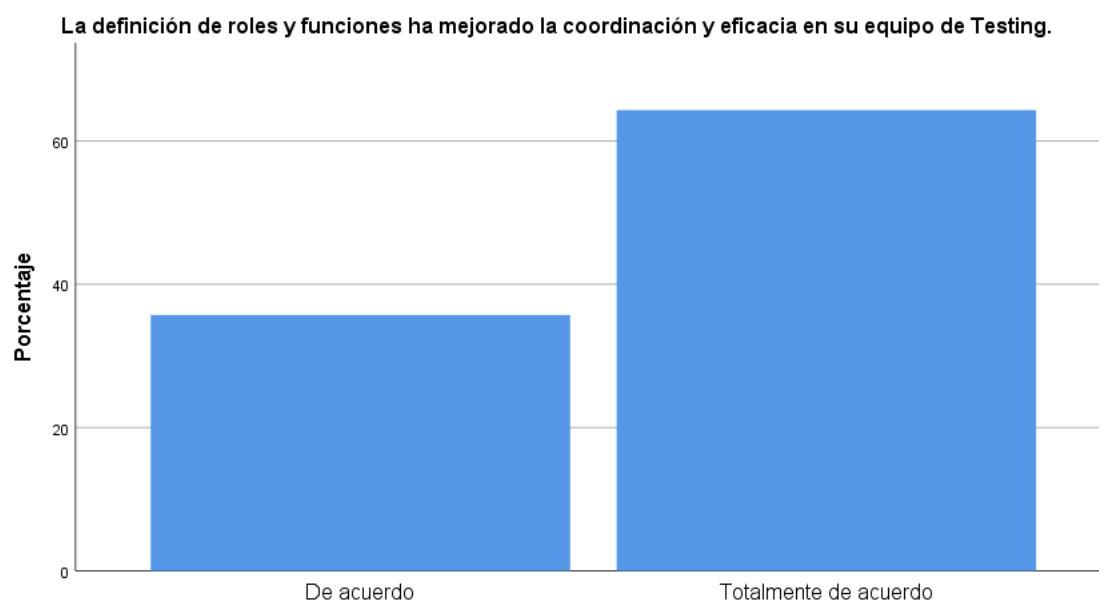
La definición de roles y funciones en el proceso de Gestión de Ejecución de Pruebas ha facilitado nuestras actividades de Testing.

Figura 27. La definición de roles y funciones en el proceso de Gestión de ejecución de pruebas ha facilitado nuestras actividades de testing.

Tabla 20. La definición de roles y funciones ha mejorado la coordinación y eficacia en su equipo de testing.

La definición de roles y funciones ha mejorado la coordinación y eficacia en su equipo de Testing.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	De acuerdo	5	35,7	35,7	35,7
	Totalmente de acuerdo	9	64,3	64,3	100,0
	Total	14	100,0	100,0	



La definición de roles y funciones ha mejorado la coordinación y eficacia en su equipo de Testing.

Figura 28. La definición de roles y funciones ha mejorado la coordinación y eficacia en su equipo de testing.

6.1.2. Resultados descriptivos de la variable calidad del producto final

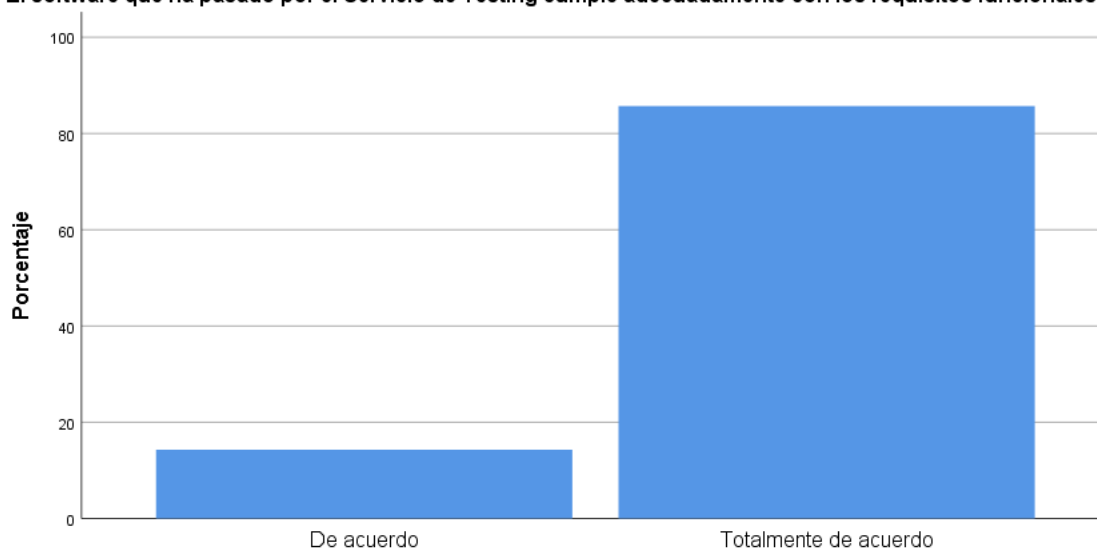
Dimensión 1: Calidad funcional del software

Tabla 21. El software que ha pasado por el Servicio de testing cumple adecuadamente con los requisitos funcionales.

El software que ha pasado por el Servicio de Testing cumple adecuadamente con los requisitos funcionales.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	De acuerdo	2	14,3	14,3	14,3
	Totalmente de acuerdo	12	85,7	85,7	100,0
	Total	14	100,0	100,0	

El software que ha pasado por el Servicio de Testing cumple adecuadamente con los requisitos funcionales.



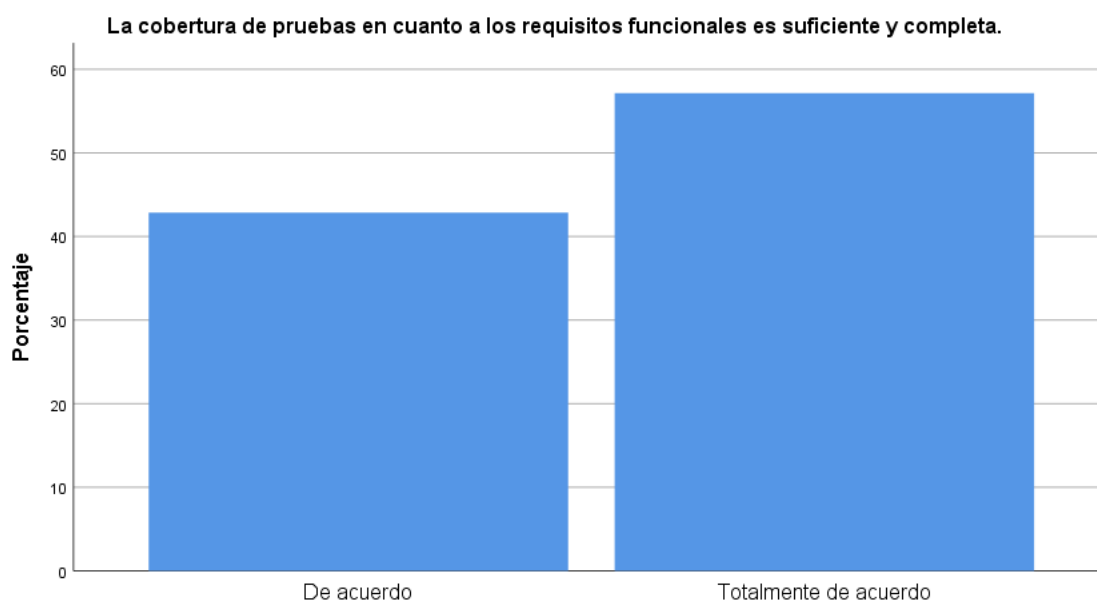
El software que ha pasado por el Servicio de Testing cumple adecuadamente con los requisitos funcionales.

Figura 29. El software que ha pasado por el Servicio de testing cumple adecuadamente con los requisitos funcionales.

Tabla 22. La cobertura de pruebas en cuanto a los requisitos funcionales es suficiente y completa.

La cobertura de pruebas en cuanto a los requisitos funcionales es suficiente y completa.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	De acuerdo	6	42,9	42,9	42,9
	Totalmente de acuerdo	8	57,1	57,1	100,0
	Total	14	100,0	100,0	



La cobertura de pruebas en cuanto a los requisitos funcionales es suficiente y completa.

Figura 30. La cobertura de pruebas en cuanto a los requisitos funcionales es suficiente y completa.

Tabla 23. La tasa de defectos funcionales encontrados es baja, lo que sugiere una alta calidad funcional.

La tasa de defectos funcionales encontrados es baja, lo que sugiere una alta calidad funcional.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	2	14,3	14,3	14,3
	De acuerdo	5	35,7	35,7	50,0
	Totalmente de acuerdo	7	50,0	50,0	100,0
	Total	14	100,0	100,0	



Figura 31. La tasa de defectos funcionales encontrados es baja, lo que sugiere una alta calidad funcional.

Dimensión 2: Rendimiento y eficiencia

Tabla 24. El software muestra un rendimiento rápido y una buena capacidad de respuesta en su funcionamiento.

El software muestra un rendimiento rápido y una buena capacidad de respuesta en su funcionamiento.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	De acuerdo	4	28,6	28,6	28,6
	Totalmente de acuerdo	10	71,4	71,4	100,0
	Total	14	100,0	100,0	

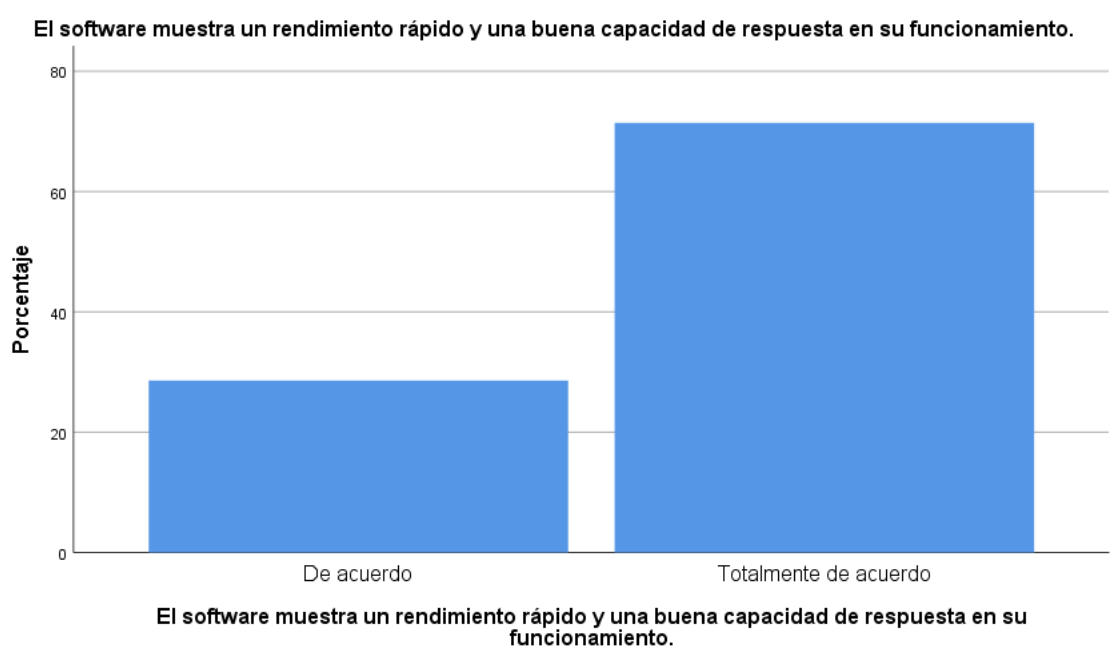


Figura 32. El software muestra un rendimiento rápido y una buena capacidad de respuesta en su funcionamiento.

Tabla 25. Durante la ejecución, el software utiliza eficientemente los recursos de hardware, como memoria y CPU.

Durante la ejecución, el software utiliza eficientemente los recursos de hardware, como memoria y CPU.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	3	21,4	21,4	21,4
	De acuerdo	3	21,4	21,4	42,9
	Totalmente de acuerdo	8	57,1	57,1	100,0
	Total	14	100,0	100,0	

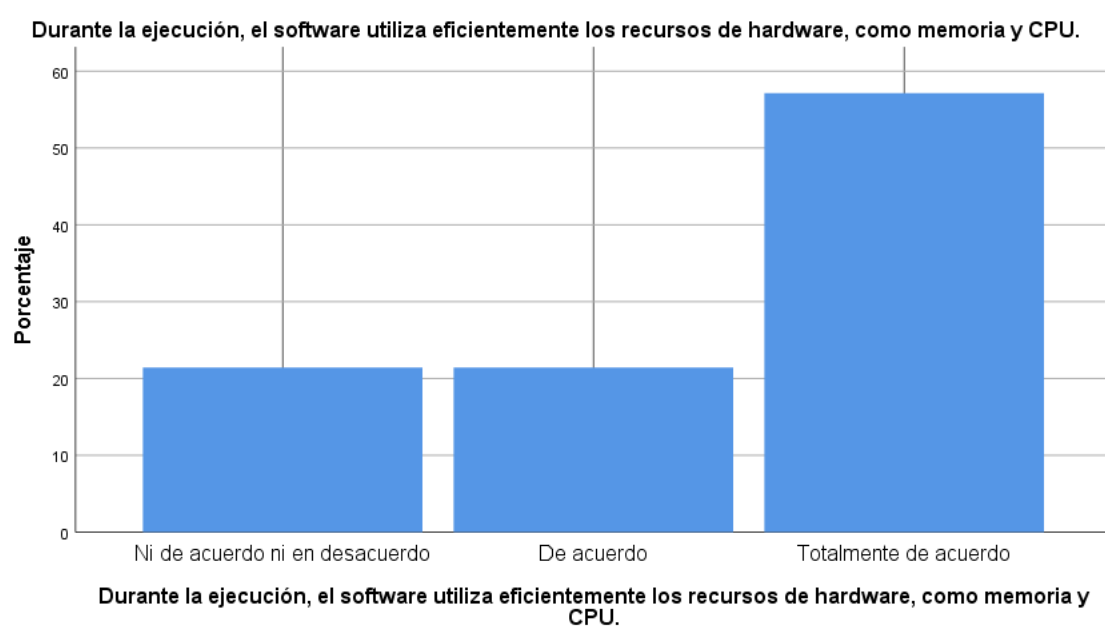


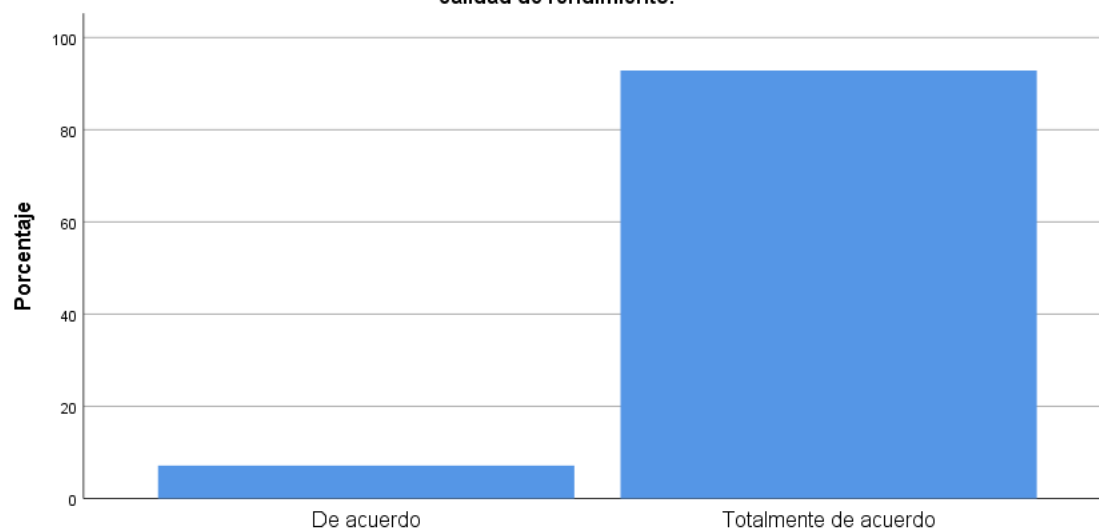
Figura 33. Durante la ejecución, el software utiliza eficientemente los recursos de hardware, como memoria y CPU.

Tabla 26. El software mantiene un buen rendimiento incluso bajo cargas de trabajo significativas, lo que indica una alta calidad de rendimiento.

El software mantiene un buen rendimiento incluso bajo cargas de trabajo significativas, lo que indica una alta calidad de rendimiento.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	De acuerdo	1	7,1	7,1	7,1
	Totalmente de acuerdo	13	92,9	92,9	100,0
	Total	14	100,0	100,0	

El software mantiene un buen rendimiento incluso bajo cargas de trabajo significativas, lo que indica una alta calidad de rendimiento.



El software mantiene un buen rendimiento incluso bajo cargas de trabajo significativas, lo que indica una alta calidad de rendimiento.

Figura 34. El software mantiene un buen rendimiento incluso bajo cargas de trabajo significativas, lo que indica una alta calidad de rendimiento.

Dimensión 3: **Fiabilidad y disponibilidad**

Tabla 27. El software es confiable y rara vez experimenta fallos inesperados.

El software es confiable y rara vez experimenta fallos inesperados.				
		Frecuencia	Porcentaje	Porcentaje válido
Válido	De acuerdo	7	50,0	50,0
	Totalmente de acuerdo	7	50,0	50,0
	Total	14	100,0	100,0

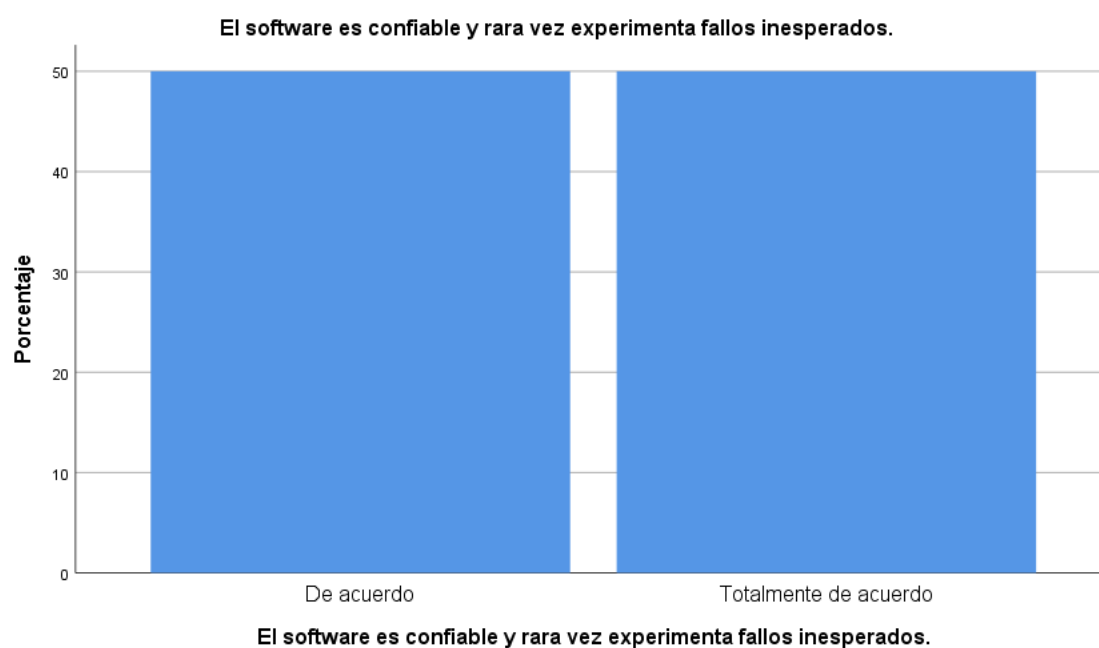


Figura 35. El software es confiable y rara vez experimenta fallos inesperados.

Tabla 28. La tasa de disponibilidad del sistema es alta, lo que significa que el software está disponible y en funcionamiento la mayor parte del tiempo.

La tasa de disponibilidad del sistema es alta, lo que significa que el software está disponible y en funcionamiento la mayor parte del tiempo.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	De acuerdo	2	14,3	14,3	14,3
	Totalmente de acuerdo	12	85,7	85,7	100,0
	Total	14	100,0	100,0	



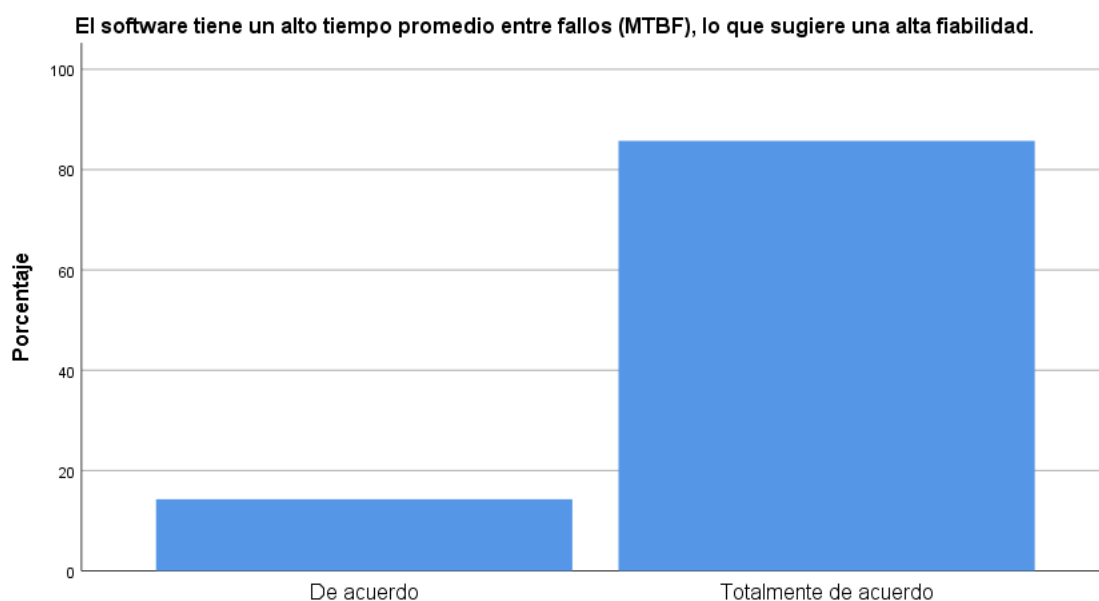
La tasa de disponibilidad del sistema es alta, lo que significa que el software está disponible y en funcionamiento la mayor parte del tiempo.

Figura 36. La tasa de disponibilidad del sistema es alta, lo que significa que el software está disponible y en funcionamiento la mayor parte del tiempo.

Tabla 29. El software tiene un alto tiempo promedio entre fallos (MTBF), lo que sugiere una alta fiabilidad.

El software tiene un alto tiempo promedio entre fallos (MTBF), lo que sugiere una alta fiabilidad.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	De acuerdo	2	14,3	14,3	14,3
	Totalmente de acuerdo	12	85,7	85,7	100,0
	Total	14	100,0	100,0	



El software tiene un alto tiempo promedio entre fallos (MTBF), lo que sugiere una alta fiabilidad.

Figura 37. El software tiene un alto tiempo promedio entre fallos (MTBF), lo que sugiere una alta fiabilidad.

Dimensión 4: **Percepción del servicio entregado al cliente**

Tabla 30. Está satisfecho con la calidad del servicio de testing de Software entregado al cliente.

Está satisfecho con la calidad del servicio de Testing de Software entregado al cliente					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Ni de acuerdo ni en desacuerdo	1	7,1	7,1	7,1
	De acuerdo	5	35,7	35,7	42,9
	Totalmente de acuerdo	8	57,1	57,1	100,0
	Total	14	100,0	100,0	



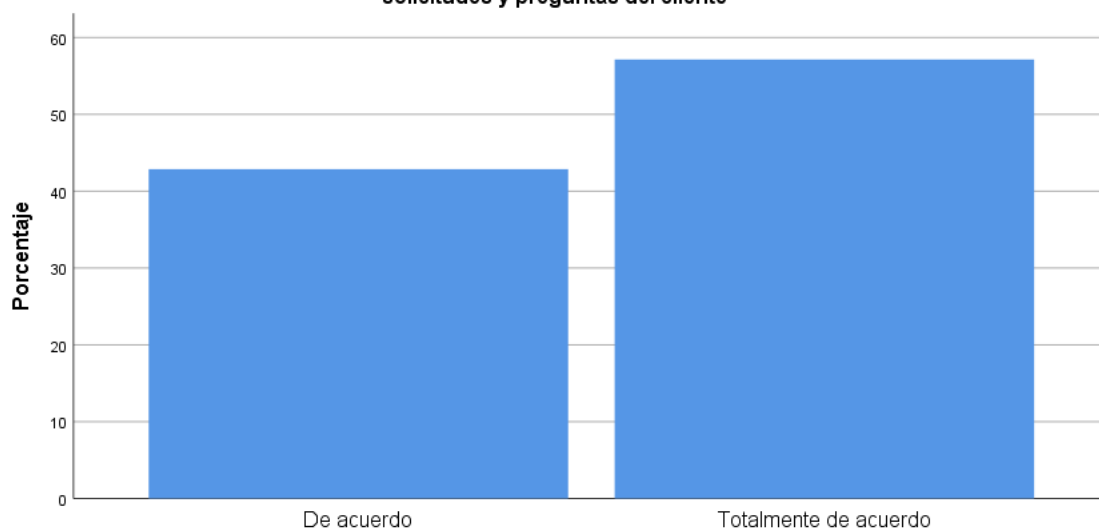
Figura 38. Está satisfecho con la calidad del servicio de testing de software entregado al cliente.

Tabla 31. Está satisfecho con la comunicación y la capacidad de respuesta del equipo de testing de software a las solicitudes y preguntas del cliente.

Está satisfecho con la comunicación y la capacidad de respuesta del equipo de Testing de Software a las solicitudes y preguntas del cliente

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	De acuerdo	6	42,9	42,9	42,9
	Totalmente de acuerdo	8	57,1	57,1	100,0
	Total	14	100,0	100,0	

Está satisfecho con la comunicación y la capacidad de respuesta del equipo de Testing de Software a las solicitudes y preguntas del cliente



Está satisfecho con la comunicación y la capacidad de respuesta del equipo de Testing de Software a las solicitudes y preguntas del cliente

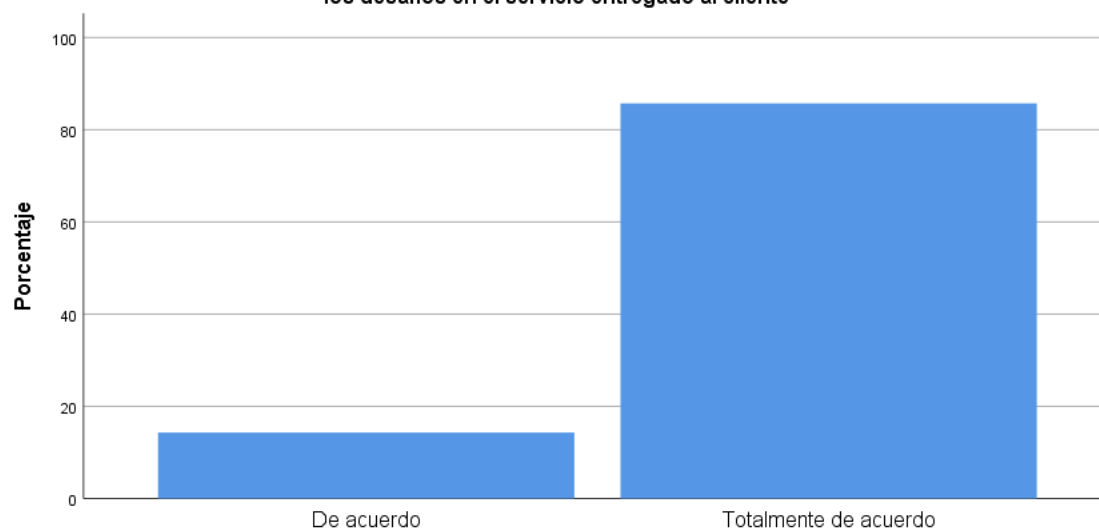
Figura 39. Está satisfecho con la comunicación y la capacidad de respuesta del equipo de testing de software a las solicitudes y preguntas del cliente.

Tabla 32. El equipo de testing de software ha demostrado un compromiso efectivo para resolver problemas y abordar los desafíos en el servicio entregado al cliente.

El equipo de Testing de Software ha demostrado un compromiso efectivo para resolver problemas y abordar los desafíos en el servicio entregado al cliente

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	De acuerdo	2	14,3	14,3	14,3
	Totalmente de acuerdo	12	85,7	85,7	100,0
	Total	14	100,0	100,0	

El equipo de Testing de Software ha demostrado un compromiso efectivo para resolver problemas y abordar los desafíos en el servicio entregado al cliente



El equipo de Testing de Software ha demostrado un compromiso efectivo para resolver problemas y abordar los desafíos en el servicio entregado al cliente

Figura 40. El equipo de testing de software ha demostrado un compromiso efectivo para resolver problemas y abordar los desafíos en el servicio entregado al cliente.

Tabla 33. El producto final cumple con sus expectativas.

El producto final cumple con sus expectativas					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	De acuerdo	8	57,1	57,1	57,1
	Totalmente de acuerdo	6	42,9	42,9	100,0
	Total	14	100,0	100,0	

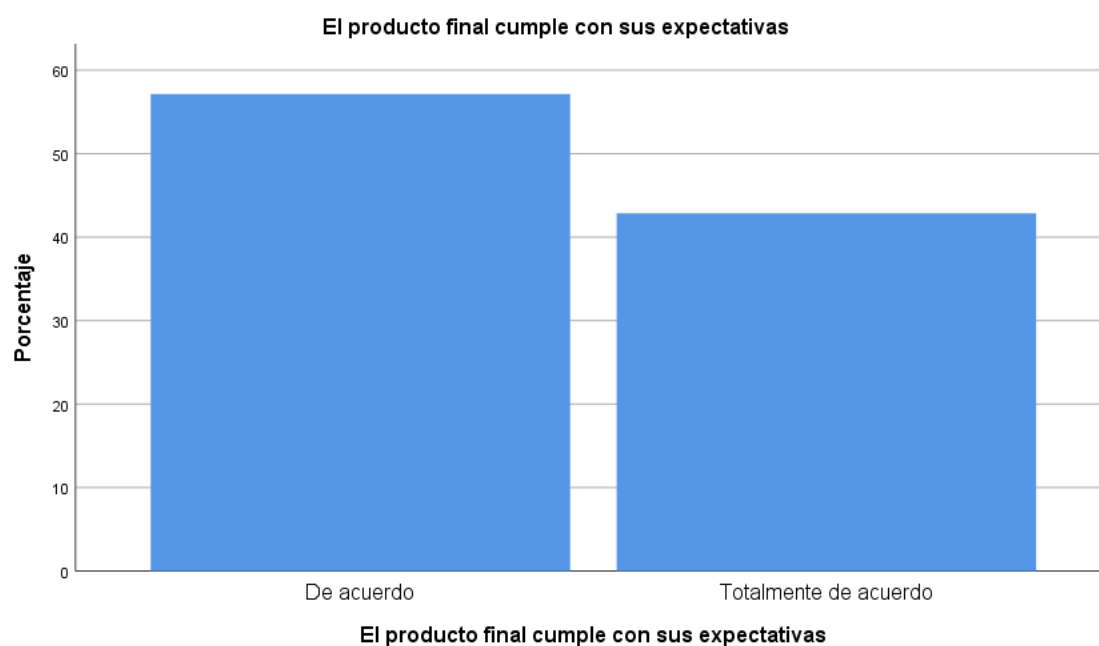


Figura 41. El producto final cumple con sus expectativas.

6.2. Resultados inferenciales

6.2.1. Selección de prueba de normalidad

Con el objetivo de determinar si los datos obtenidos de las encuestas se distribuyen normalmente, se utilizó la prueba de normalidad de Shapiro-Wilk. Dado que la muestra estaba compuesta por 20 expertos de TI, lo que es inferior al umbral sugerido de 50 casos, se utilizó esta prueba. Es relevante señalar que esta información es importante ya que facilitará la elección de una prueba de hipótesis apropiada para ofrecer resultados adecuados. Para esto, se propusieron dos acontecimientos que podrían ocurrir después de la prueba:

- H_0 : El modelo de procesos para un servicio de testing y la calidad del producto final sigue una distribución normal.
- H_a : El modelo de procesos para un servicio de testing y la calidad del producto final no sigue una distribución normal.

Asimismo, el p- valor se escogió como método con el fin de establecer si se rechaza alguno de los supuestos mencionados teniendo en cuenta:

- $\alpha \leq 0.05$ = rechazo H_0 ,
- $\alpha > 0.05$ = se acepta H_0 ,

En donde la significancia está considerada por el valor α , siendo la confiabilidad del 95% y 5% el margen de error. A continuación, se muestra la prueba realizada.

Tabla 42. Prueba de normalidad

	Pruebas de normalidad					
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Modelo de procesos para un servicio de testing de software	,335	20	,000	,641	20	,000
Calidad del producto final	,438	20	,000	,580	20	,000

a. Corrección de significación de Lilliefors

Se obtuvo como resultado de la prueba que las variables modelo de procesos para un servicio de testing y la calidad del producto final, así como sus dimensiones, no seguían una distribución normal. Esto significa que estas variables y dimensiones son no paramétricas.

Como cada variable no tiene una distribución normal, se aplica la estadística no paramétrica en la tesis para examinar las hipótesis. Para ello, se emplea la regresión logística ordinal.

6.2.2. Regresión logística ordinal

En los estudios, a menudo se requiere la medición de fenómenos que pertenecen a categorías cualitativas, las cuales normalmente son limitadas y agrupadas en escalas ordinales o nominales. Se utilizan métodos como la regresión probit y la regresión logística ordinal para tratar este tipo de datos. La principal distinción entre estas metodologías es que no dependen de que las variables obedezcan a una distribución normal.

Morales (2018) señala que, si la normalidad no se cumple, algunos expertos recomiendan utilizar regresión logística ordinal. Esta técnica se expande sobre la regresión logística binaria y emplea un modelo logístico acumulativo para la variable ordinal, que se simboliza con "Y", en sus distintas categorías. El propósito principal de este estudio es entender la conexión entre una variable ordinal y sus categorías, según cada variable independiente.

6.2.3. Decisión estadística

Para decidir si se debe aceptar o rechazar la hipótesis nula (H_0), es necesario formular suposiciones lógicas. Uno de los métodos más comunes para lograr esto es el valor p. Este último muestra la probabilidad de que la hipótesis nula sea cierta y se utiliza para determinar si los fenómenos observados son probablemente consecuencia de sucesos aleatorios. Si α es menor o igual a 0.05, se descarta la hipótesis nula, pero si α supera el 0.05, no se desecha la hipótesis nula. α simboliza el grado de significación en este contexto, donde para un nivel de confianza del 95% (Mendenhall et al., 2010), el margen de error es de 0.05.

6.2.4. Prueba de hipótesis general

H_0 : Un modelo de procesos para un servicio de testing de software no influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

H_i : Un modelo de procesos para un servicio de testing de software influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

En el análisis presentado en la tabla 43, se encontró que el pvalor o significancia fue de 0.006 que resulta ser menor que el margen de error establecido (0.05). Por ello, se procede a la

aceptación de la hipótesis alternativa. En ese sentido, se afirma que un modelo de procesos para un servicio de testing de software influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

Tabla 43. Información de ajuste del modelo de la hipótesis general

Modelo	Información de ajuste de los modelos			
	Logaritmo de la verosimilitud -2	Chi-cuadrado	gl	Sig.
Sólo intersección	9,578			
Final	1,914	7,664	1	,006

Función de enlace: Logit.

En el análisis presentado en la tabla 44, teniendo en cuenta estos valores de la prueba Pseudo R2 de Cox y Snell, la calidad del producto final es explicada por el modelo de procesos para un servicio de testing de software en un 42.2% y según la prueba de Nagelkerke indica que la calidad del producto final es explicada por el modelo de procesos para un servicio de testing de software en un 65.2%; de tal forma que, el nivel de ajuste del modelo con respecto a los datos es alto. El estadístico R2 de Cox y Snell fue utilizado para comparar el modelo nulo con el modelo ajustado que incluye los parámetros del modelo de procesos para un servicio de testing de software, confirmando que la adopción del modelo de procesos es un predictor determinante para la mejora de la calidad en el contexto estudiado.

Tabla 44. Pseudo R cuadrado de la hipótesis general

Pseudo R cuadrado	
Cox y Snell	,422
Nagelkerke	,652
McFadden	,527

Función de enlace: Logit.

6.2.5. Prueba de hipótesis específica 1

H0: La ejecución de pruebas no influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

Hi: La ejecución de pruebas influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

En el análisis presentado en la tabla 45, se encontró que el pvalor o significancia fue de 0.002 que resulta ser menor que el margen de error establecido (0.05). Por ello, se procede a la aceptación de la hipótesis alternativa. En ese sentido, se afirma que la ejecución de pruebas influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

Tabla 45. Información de ajuste del modelo de la hipótesis específica 1

Información de ajuste de los modelos				
Modelo	Logaritmo de la verosimilitud -2	Chi-cuadrado	gl	Sig.
Sólo intersección	11,956			
Final	1,906	10,050	1	,002

Función de enlace: Logit.

En el análisis presentado en la tabla 46, teniendo en cuenta estos valores de la prueba Pseudo R2 de Cox y Snell, la calidad del producto final es explicada por la ejecución de pruebas en un 51.2% y según la prueba de Nagelkerke indica que la calidad del producto final es explicada por la ejecución de pruebas en un 73.4%; de tal forma que, el nivel de ajuste del modelo con respecto a los datos es alto. El estadístico R2 de Cox y Snell fue utilizado para comparar el modelo nulo con el modelo ajustado que incluye la dimensión de gestión de la ejecución de pruebas.

Tabla 46. Pseudo R cuadrado de la hipótesis específica 1

Pseudo R cuadrado	
Cox y Snell	,512
Nagelkerke	,734
McFadden	,600

Función de enlace: Logit.

6.2.6. Prueba de hipótesis específica 2

H0: El servicio de testing no influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

Hi: El servicio de testing influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

En el análisis presentado en la tabla 47, se encontró que el pvalor o significancia fue de 0.015 que resulta ser menor que el margen de error establecido (0.05). Por ello, se procede a la aceptación de la hipótesis alternativa. En ese sentido, se afirma que un el servicio de testing influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

Tabla 47. Información de ajuste del modelo de la hipótesis específica 2

Modelo	Información de ajuste de los modelos			
	Logaritmo de la verosimilitud -2	Chi-cuadrado	gl	Sig.
Sólo intersección	8,372			
Final	2,434	5,938	1	,015

Función de enlace: Logit.

En el análisis presentado en la tabla 48, teniendo en cuenta estos valores de la prueba Pseudo R2 de Cox y Snell, la calidad del producto final es explicada por el servicio de testing en un 34.6% y según la prueba de Nagelkerke indica que la calidad del producto final es explicada por el servicio de testing en un 49.5%; de tal forma que, el nivel de ajuste del modelo con respecto a los datos es moderado. El estadístico R2 de Cox y Snell fue utilizado para comparar el modelo nulo con el modelo que integra la dimensión gestión del servicio de testing, demostrando que los parámetros de gestión del servicio aportan una capacidad explicativa adicional al modelo base.

Tabla 48. Pseudo R cuadrado de la hipótesis específica 2

Pseudo R cuadrado	
Cox y Snell	,346
Nagelkerke	,495
McFadden	,354

Función de enlace: Logit.

6.2.7. Prueba de hipótesis específica 3

H0: La mejora continua no influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

Hi: La mejora continua influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

En el análisis presentado en la tabla 49, se encontró que el pvalor o significancia fue de 0.001 que resulta ser menor que el margen de error establecido (0.05). Por ello, se procede a la aceptación de la hipótesis alternativa. En ese sentido, se afirma que la mejora continua influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

Tabla 49. Información de ajuste del modelo de la hipótesis específica 3

Modelo	Información de ajuste de los modelos			
	Logaritmo de la verosimilitud -2	Chi-cuadrado	gl	Sig.
Sólo intersección	13,533			
Final	1,785	11,748	1	,001

Función de enlace: Logit.

En el análisis presentado en la tabla 50, teniendo en cuenta estos valores de la prueba Pseudo R2 de Cox y Snell, la calidad del producto final es explicada por la mejora continua en un 56.8% y según la prueba de Nagelkerke indica que la calidad del producto final es 81.4%; de tal forma que, el nivel de ajuste del modelo con respecto a los datos es muy alto. El estadístico R2 de Cox y Snell fue utilizado para comparar el modelo nulo con el modelo ajustado por la dimensión gestión de la mejora continua. El incremento en el R2 evidencia que las prácticas de mejora continua son determinantes para explicar la varianza en la calidad del producto.

Tabla 50. Pseudo R cuadrado de la hipótesis específica 3

Pseudo R cuadrado	
Cox y Snell	,568
Nagelkerke	,814
McFadden	,701

Función de enlace: Logit.

6.2.8. Prueba de hipótesis específica 4

H0: La infraestructura y testware no influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

Hi: La infraestructura y testware influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

En el análisis presentado en la tabla 51, se encontró que el pvalor o significancia fue de 0.003 que resulta ser menor que el margen de error establecido (0.05). Por ello, se procede a la aceptación de la hipótesis alternativa. En ese sentido, se afirma que la infraestructura y testware influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

Tabla 51. Información de ajuste del modelo de la hipótesis específica 4

Modelo	Información de ajuste de los modelos			
	Logaritmo de la verosimilitud -2	Chi-cuadrado	gl	Sig.
Sólo intersección	11,335			
Final	2,222	9,113	1	,003

Función de enlace: Logit.

En el análisis presentado en la tabla 52, teniendo en cuenta estos valores de la prueba Pseudo R2 de Cox y Snell, la calidad del producto final es explicada por la infraestructura y testware en un 47.8% y según la prueba de Nagelkerke indica que la calidad del producto final es explicada por la infraestructura y testware en un 68.6%; de tal forma que, el nivel de ajuste del modelo con respecto a los datos es alto. El estadístico R2 de Cox y Snell fue utilizado para comparar el modelo nulo con el modelo que incorpora la dimensión gestión de infraestructura y testware, validando que los recursos tecnológicos y herramientas son factores explicativos significativos frente a un modelo sin intervención.

Tabla 52. Pseudo R cuadrado de la hipótesis específica 4

Pseudo R cuadrado	
Cox y Snell	,478
Nagelkerke	,686
McFadden	,544

Función de enlace: Logit.

6.2.9. Prueba de hipótesis específica 5

H0: El control de ejecución de pruebas no influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

Hi: El control de ejecución de pruebas influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

En el análisis presentado en la tabla 53, se encontró que el pvalor o significancia fue de 0.007 que resulta ser menor que el margen de error establecido (0.05). Por ello, se procede a la aceptación de la hipótesis alternativa. En ese sentido, se afirma el control de ejecución de pruebas influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

Tabla 53. Información de ajuste del modelo de la hipótesis específica 5

Modelo	Información de ajuste de los modelos			
	Logaritmo de la verosimilitud -2	Chi-cuadrado	gl	Sig.
Sólo intersección	9,641			
Final	2,450	7,191	1	,007

Función de enlace: Logit.

En el análisis presentado en la tabla 54, teniendo en cuenta estos valores de la prueba Pseudo R2 de Cox y Snell, la calidad del producto final es explicada por el control de ejecución de pruebas en un 40.2% y según la prueba de Nagelkerke indica que la calidad del producto final es explicada por el control de ejecución en un 57.6%; de tal forma que, el nivel de ajuste del modelo con respecto a los datos es moderado-alto. El estadístico R2 de Cox y Snell fue utilizado para comparar el modelo nulo con el modelo ajustado que incluye la dimensión control de la ejecución de pruebas, confirmando la relevancia estadística de las actividades de monitoreo y control en el resultado final.

Tabla 54. Pseudo R cuadrado de la hipótesis específica 5

Pseudo R cuadrado	
Cox y Snell	,402
Nagelkerke	,576
McFadden	,429

Función de enlace: Logit.

6.2.10. Prueba de hipótesis específica 6

H0: Las competencias profesionales no influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

Hi: Las competencias profesionales influye de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

En el análisis presentado en la tabla 55, se encontró que el pvalor o significancia fue de 0.006 que resulta ser menor que el margen de error establecido (0.05). Por ello, se procede a la aceptación de la hipótesis alternativa. En ese sentido, se afirma que las competencias profesionales influyen de forma positiva en la calidad del producto final en una empresa de telecomunicaciones.

Tabla 55. Información de ajuste del modelo de la hipótesis específica 6

Modelo	Información de ajuste de los modelos			
	Logaritmo de la verosimilitud -2	Chi-cuadrado	gl	Sig.
Sólo intersección	9,578			
Final	1,914	7,664	1	,006

Función de enlace: Logit.

En el análisis presentado en la tabla 56, teniendo en cuenta estos valores de la prueba Pseudo R2 de Cox y Snell, la calidad del producto final es explicada por las competencias profesionales en un 42.2% y según la prueba de Nagelkerke indica que la calidad del producto final es explicada por las competencias profesionales en un 65.2%; de tal forma que, el nivel de ajuste del modelo con respecto a los datos es alto. El estadístico R2 de Cox y Snell fue utilizado para comparar el modelo nulo con el modelo que integra la dimensión gestión de competencias profesionales, lo cual ratifica que la cualificación del capital humano añade un poder predictivo sustancial al modelo.

Tabla 56. Pseudo R cuadrado de la hipótesis específica 6

Pseudo R cuadrado	
Cox y Snell	,422
Nagelkerke	,652
McFadden	,527

Función de enlace: Logit.

6.3. Discusiones

En la presente sección se exponen las discusiones principales del trabajo de investigación:

Se ha evidenciado que la implementación de un modelo de procesos estructurado, como el modelo propuesto, genera un impacto significativo y medible en la calidad del producto final. Los resultados obtenidos mediante regresión logística ordinal confirmaron que el modelo propuesto explica entre el 42.2% (Cox y Snell) y el 65.2% (Nagelkerke), de la variabilidad en la calidad del producto final, lo cual representa una capacidad predictiva alta según los estándares establecidos en la literatura metodológica (Mendenhall et al., 2010). Este hallazgo resulta particularmente relevante al contrastarlo con investigaciones previas, como las de Carrizo y Alfaro (2018) y Serna y Arango (2011), quienes identificaron que la ausencia de marcos metodológicos integrados es una de las principales debilidades en los procesos de aseguramiento de calidad.

También se observa que los seis componentes del modelo propuesto: gestión de ejecución de pruebas, gestión del servicio, mejora continua, infraestructura y testware, control de ejecución, y competencias profesionales, demostraron contribuciones diferenciadas pero complementarias. La ejecución de pruebas ($R^2 = 73.4\%$) y la mejora continua ($R^2 = 81.4\%$) emergieron como los pilares fundamentales, mientras que dimensiones como el servicio de testing ($R^2 = 49.5\%$) y el control de ejecución ($R^2 = 57.6\%$) mostraron efectos moderados pero significativos. Esta distribución de impactos sugiere que el modelo opera mediante una arquitectura de refuerzo mutuo, donde las dimensiones operativas (ejecución y control) se complementan mediante las dimensiones estratégicas (mejora continua y competencias profesionales). Esta sinergia replica los principios de sistemas de gestión integrados propuestos por ISO/IEC 20000 y ITIL 4, donde la efectividad no radica en componentes aislados sino en su articulación.

El modelo propuesto integra componentes de CMMI V2.0, TMMI 1.3, ITIL 4, PMBOK 2021 e ISO/IEC 25000, superando las limitaciones de aplicar estos marcos de manera aislada. La investigación demuestra que la integración de dichos marcos, bajo una contextualización adecuada puede mitigar las barreras de complejidad de adopción, resistencia cultural o sobrecarga en la documentación. Por ello, a diferencia de otros modelos que se enfocan

principalmente en estructuras técnicas de las pruebas, como TestPAI (Sanz et al., 2008), el modelo propuesto incorpora explícitamente la dimensión de gestión del servicio bajo esquemas de outsourcing. Esta diferencia es fundamental, pues según Poston (2007) y Hodosi (2024), dado que la gestión de expectativas, comunicación y acuerdos contractuales constituyen factores críticos de éxito en servicios externalizados que no están suficientemente cubiertos por los modelos tradicionales de madurez de Testing.

CONCLUSIONES

Modelo de procesos para un servicio de testing de software: El modelo tiene una influencia positiva significativa en la calidad del producto final en una empresa de telecomunicaciones. Esto se respalda con un valor de p de 0.006, un Pseudo R^2 de Cox y Snell del 42.2%, y un R^2 de Nagelkerke del 65.2%. Estos resultados sugieren que el modelo de procesos para un servicio de testing de software es esencial para mejorar la calidad del producto final.

Ejecución de Pruebas: La ejecución de pruebas también influye de manera significativa en la calidad del producto final en una empresa de telecomunicaciones. Esto se refleja en un valor de p de 0.002, un Pseudo R^2 de Cox y Snell del 51.2%, y un R^2 de Nagelkerke del 73.4%. Los resultados indican que una ejecución efectiva de pruebas es fundamental en la calidad del producto final.

Servicio de testing: El servicio de testing tiene un impacto positivo significativo en la calidad del producto final en una empresa de telecomunicaciones, como se demuestra por un valor de p de 0.015, un Pseudo R^2 de Cox y Snell del 34.6%, y un R^2 de Nagelkerke del 49.5%. Esto destaca la importancia de un servicio de testing en la calidad del producto final.

Mejora Continua: La mejora continua influye de manera significativa en la calidad del producto final en una empresa de telecomunicaciones, como se confirma con un valor de p de 0.001, un Pseudo R^2 de Cox y Snell del 56.8%, y un R^2 de Nagelkerke del 81.4%. Estos resultados subrayan la necesidad de implementar procesos de mejora continua para asegurar la calidad del producto final.

Infraestructura y Testware: La infraestructura y el testware tienen un impacto positivo significativo en la calidad del producto final en una empresa de telecomunicaciones, como se refleja en un valor de p de 0.003, un Pseudo R^2 de Cox y Snell del 47.8%, y un R^2 de Nagelkerke del 68.6%. Esto destaca la importancia de la infraestructura tecnológica en el proceso de testing.

Control de Ejecución de Pruebas: El control de ejecución de pruebas influye de manera significativa en la calidad del producto final en una empresa de telecomunicaciones, como se

demuestra por un valor de p de 0.007, un Pseudo R^2 de Cox y Snell del 40.2%, y un R^2 de Nagelkerke del 57.6%. Estos resultados resaltan la necesidad de una supervisión efectiva durante la ejecución de pruebas.

Competencias Profesionales: Las competencias profesionales también influyen de manera significativa en la calidad del producto final en una empresa de telecomunicaciones, como se respalda con un valor de p de 0.006, un Pseudo R^2 de Snell del 42.2%, y un R^2 de Nagelkerke del 65.2%. Esto subraya la importancia de contar con personal altamente calificado en el servicio de testing.

RECOMENDACIONES

Se recomienda a la Gerencia General priorizar la adopción formal del modelo de procesos para el servicio de testing de software en toda la organización, estableciendo políticas internas que aseguren su implementación de manera estandarizada. Dado el impacto significativo evidenciado en la calidad del producto final en una empresa de telecomunicaciones, es fundamental consolidar este modelo como un pilar estratégico para garantizar la competitividad y la confianza de los usuarios.

Se recomienda a la Oficina de Gestión de Proyectos (PMO) fortalecer los lineamientos y procedimientos asociados a la ejecución de pruebas, asegurando que los equipos cuenten con guías claras, tiempos bien definidos y métricas objetivas que permitan controlar y evaluar la eficacia de esta fase. Una ejecución rigurosa de pruebas contribuye de forma determinante al éxito de los proyectos y a la reducción de incidencias en producción.

Se recomienda a la Jefatura del Área de Calidad de Software implementar mecanismos de seguimiento y control que garanticen que el servicio de testing mantenga niveles de desempeño consistentes y acordes con las expectativas de los clientes internos y externos. El desarrollo de acuerdos de nivel de servicio (SLA) específicos y la retroalimentación sistemática sobre la calidad percibida fortalecerán esta práctica clave en el ciclo de vida del software.

Se recomienda a la Dirección de Aseguramiento de Calidad instaurar un programa permanente de mejora continua, integrando revisiones periódicas de procesos, lecciones aprendidas y planes de acción orientados a la optimización. Dado su impacto altamente significativo en la calidad y satisfacción del cliente, este enfoque permitirá evolucionar de manera constante y adaptarse a las necesidades cambiantes del negocio y del mercado.

Se recomienda a la Gerencia de Tecnología de la Información garantizar la actualización y disponibilidad de la infraestructura tecnológica y del testware necesarios para soportar adecuadamente los procesos de testing. La inversión en entornos de prueba robustos y herramientas especializadas constituye un factor decisivo para asegurar la fiabilidad de los resultados y la percepción positiva del servicio por parte del cliente.

Se recomienda a la Coordinación de Pruebas y Validación fortalecer los mecanismos de control y supervisión de la ejecución de pruebas, mediante el uso de indicadores de desempeño, auditorías internas y reportes de avance que permitan identificar desviaciones de manera oportuna. Este control efectivo es indispensable para consolidar la calidad del producto y garantizar el cumplimiento de los cronogramas establecidos.

Se recomienda a la Gerencia de Recursos Humanos, en coordinación con la Jefatura del Área de Calidad, diseñar un plan de capacitación continua orientado a reforzar las competencias técnicas y metodológicas del personal involucrado en el servicio de testing. La consolidación de un equipo altamente calificado impacta de manera directa en la calidad final del software y en la satisfacción del cliente, por lo que debe considerarse una inversión estratégica prioritaria.

REFERENCIAS BIBLIOGRÁFICAS

- AXELOS. (2019). *ITIL® Foundation: ITIL 4 edition*. The Stationery Office.
- Baresi, L., & Pezzè, M. (2006). An introduction to software testing. *Electronic Notes in Theoretical Computer Science*, 148(1), 89–111.
<https://doi.org/10.1016/j.entcs.2005.12.014>
- Barthelemy, J. (2003). The seven deadly sins of outsourcing. *Academy of Management Executive*, 17(2), 87-98. <https://doi.org/10.5465/AME.2003.10025>
- Bevan, N. (1999). Quality in use: Meeting user needs for quality. *Journal of Systems and Software*, 49(1), 89–96. [https://doi.org/10.1016/S0164-1212\(99\)00070-9](https://doi.org/10.1016/S0164-1212(99)00070-9)
- Bittinger, S. (2005). ITIL adoption in Asia Pacific - 2005 update, Gartner.
- Black, R. (2002). *Managing the testing process: Practical tools and techniques for managing hardware and software testing* (2.^a ed.). Wiley.
- Blanquicett, L. A., Bonfante, M. C., & Acosta-Solano, J. (2018). Prácticas de pruebas desde la industria de software. La plataforma ASISTO como caso de estudio. *Información Tecnológica*, 29(1), 11-18. <https://doi.org/10.4067/S0718-07642018000100011>
- Campbell, P. L. (2005). A COBIT primer (Nº. SAND2005-3455): Sandia National Laboratories.
- Cappiello, C., & Comuzzi, M. (2009). A utility-based model to define the optimal data quality level in IT service offerings. En 17th European Conference on Information Systems. <https://aisel.aisnet.org/ecis2009/324/>
- Carrizo, D., & Alfaro, A. (2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. *Ingeniare: Revista Chilena de Ingeniería*, 26(1), 114-129. <https://doi.org/10.4067/S0718-33052018000100114>
- Casson, D. (2005). North American ITIL assessment: an in-depth analysis of the current state and readiness of IT organizations to adopt ITIL-based processes. Evergreen Systems.
- Catelani, M., Ciani, L., Scarano, V. L., & Bacioccola, A. (2010). Software automated testing: A solution to maximize the test plan coverage and to increase software reliability and quality in use. *Computer Standards & Interfaces*, 32(5-6), 298–306.
<https://doi.org/10.1016/j.csi.2010.03.004>
- Cater-Steel, A. (2006). Challenge of adopting multiple process improvement frameworks. University of Southern Queensland.

- Cater-Steel, A. P., & Tan, W.-G. (2005, 29 de September). Summary of responses to itSMF Conference. Survey. <http://www.usq.edu.au/users/caterst/>
- The Standish Group. (2010). Outsourced and off-shore quality assurance and testing services. Chaos Report.
- Chen, J.-C., & Huang, S.-J. (2009). An empirical analysis of the impact of software development problem factors on software maintainability. *Journal of Systems and Software*, 82, 981-992. <https://doi.org/10.1016/j.jss.2008.12.036>
- Dávila, A., García, C., & Córdor, S. (2017). Análisis exploratorio en la adopción de prácticas de pruebas de software de la ISO/IEC 29119-2 en organizaciones en Lima, Perú. *RISTI*, (21), 1-17. <https://doi.org/10.17013/risti.21.1-17>.
- Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User acceptance of computer technology: A comparison of two theoretical models. *Management Science*, 35(8), 982–1003. <https://doi.org/10.1287/mnsc.35.8.982>
- De Haes, S., & Van Grembergen, W. (2009). *Enterprise governance of information technology: Achieving strategic alignment and value*. Springer.
- De Haes, S., Van Grembergen, W., & Debreceny, R. S. (2013). COBIT 5 and enterprise governance of information technology: building blocks and research opportunities. *Journal of Information Systems*, 27(1), 307–324. <https://doi.org/10.2308/isis-50422>
- Deloitte Risk Services. (2008) *Bringing the market to you: Offshoring software testing*.
- Early, Marvin M. (2006). *Improving the success rate of software development projects*. Ph.D. dissertation, Northcentral University, United States -- Arizona. Retrieved February 13, 2011, from ABI/INFORM Global. (Publication No. AAT 3200346).
- Elmuti, D., & Kathawala, Y. (2000). The effects of global outsourcing strategies on participants' attitudes and organizational effectiveness. *International Journal of Manpower*, 21(2), 112-128. <https://doi.org/10.1108/01437720010319743>
- Farrell-Vinay, P. (2008). *Manage software testing*. Auerbach Publications.
- Francheschini, F., Galetto, M., & Varetto, M. (2003). Outsourcing: Guidelines for a structured approach. *Benchmarking: An International Journal*, 10(3), 246-260. <https://doi.org/10.1108/14635770310477771>
- Gillenson, M. L., Racer, M. J., Richardson, S. M., & Zhang, X. (2011). Engaging testers early and throughout the software development process: Six models and a simulation study. *En 1st International Research Workshop on Advances and Innovations in Software Testing*.

- Global Growth Insights. (2024). Outsourced software testing market size, share & growth trends report 2025-2033. <https://www.globalgrowthinsights.com/market-reports/outsourced-software-testing-market-104283>
- Hanafizadeh, P., & Zararavasan, M. (2020). A systematic literature review on IT outsourcing decision and success. *Information Systems and e-Business Management*, 18, 639-696. <https://doi.org/10.4018/JGIM.2020040108>
- Hardy, G. (2006). Using IT governance and COBIT to deliver value with IT and respond to legal, regulatory and compliance challenges. *Information Security Technical Report*, 11(1), 55-61. <https://doi.org/10.1016/j.istr.2005.12.004>
- Hochstein, A., Tamm, G., & Brenner, W. (2005). Service-oriented IT management: Benefit, cost, and success factors. University of St. Gallen.
- Hodosi, G. (2024). Influencing factors in information technology outsourcing. *Procedia Computer Science*, 231, 567–572. <https://doi.org/10.1016/j.procs.2023.12.236>
- IEEE Computer Society. (2024). SWEBOK guide version 4: Software engineering body of knowledge. IEEE.
- Illinois Institute of Technology. (1998). Test maturity model (TMM). Illinois Institute of Technology.
- ISACA. (2019). *COBIT 2019 Framework: Introduction and Methodology*. ISACA.
- ISACA. (2022). *CMMI® Development V2.0: Driving Performance Through Capability*. ISACA
- ISO. (2005a, 15 Sept). Latest ISO Survey confirms integration of ISO 9001 and ISO 14001 with world economy. Retrieved 12 Nov, 2005, from <http://www.iso.org/iso/en/commcentre/pressreleases>
- ISO. (2005b, 16 Sept). Understand the basics: ISO 9000 and ISO 14000. Retrieved November 9, 2005
- ISO/IEC. (2016). ISO/IEC 12207:2008 - Standard for information technology - Software life cycle processes (STABILIZED Jun 2016). IEEE/EIA.
- ISO/IEC. (1998). ISO/IEC 15504 - Information technology - Software process assessment. IEEE/EIA.
- ISO/IEC. (2005). ISO/IEC 20000-1:2005 - Information technology - Service management. IEEE/EIA.
- ISO/IEC. (2011). ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software

- quality models. International Organization for Standardization.
<https://www.iso.org/standard/35733.html>
- ISO/IEC. (2014). ISO/IEC 25000:2014 - Systems and software engineering: Systems and software quality requirements and evaluation (SQuaRE): Guide to SQuaRE.
<https://www.iso.org/standard/64764.html>
- Kaner, C., Falk, J., & Nguyen, H. Q. (1999). Testing computer software (2.^a ed.). Wiley.
- Khan, T., Pezeshki, V., Clear, F., & Al-Kaabi, A. (2010). Diverse virtual social networks: Implications for remote software testing teams. En European, Mediterranean & Middle Eastern Conference on Information Systems 2010.
- Kitchenham, B., & Pfleeger, S. L. (2008). Personal opinion surveys. En F. Shull, J. Singer, & D. I. K. Sjøberg (Eds.), Guide to advanced empirical software engineering (pp. 63-92). Springer. https://doi.org/10.1007/978-1-84800-044-5_3
- Kruse, P. M., & Luniak, M. (2010). Automated test case generation using classification trees. Software Quality Journal, 13(1), 4-12. <https://doi.org/10.1007/s11219-009-9081-8>
- Kshirasagar, N., & Tripathy, P. (2008). Software testing and quality assurance: Theory and practice. John Wiley & Sons.
- Laprie, J. C. (1995). Dependable computing and fault tolerance: Concepts and terminology. In 25th International Symposium on Fault-Tolerant Computing (pp. 2–11). IEEE.
<https://doi.org/10.1109/FTCSH.1995.532603>
- Marcano, I., & Benigni, G. (2014). Análisis de alternativas metodológicas para el desarrollo de software educativo. Saber, 26(3), 297-304.
- Market Research Future. (2025). Outsourced software testing market research report: Forecast to 2034. <https://www.marketresearchfuture.com/reports/outsourced-software-testing-market-11653>
- Mendenhall, W., Beaver, R. J., & Beaver, B. M. (2010). *Introduction to probability and statistics* (14th ed.). Brooks/Cole, Cengage Learning.
- Moody, D. L. (2003). The method evaluation model: A theoretical model for validating information systems design methods. European Conference on Information Systems, Paper 79. <http://aisel.aisnet.org/ecis2003/79>
- Morales Vallejo, P. (2018). *Estadística aplicada a las ciencias sociales*. Universidad Pontificia Comillas.
- Myers, G. J. (2004). The art of software testing (2.^a ed.). John Wiley & Sons.

- Norma Técnica Peruana NTP-ISO-IEC 12207 (2006) Procesos del ciclo de vida del software. 2da Edición. INDECOPI.
- Offermann, P., Levina, O., Schönherr, M., & Bub, U. (2009). Outline of a design science research process. En Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology (Artículo 978). ACM. <https://doi.org/10.1145/1555619.1555629>
- Onita, C., & Dhaliwal, J. (2007). A framework for aligning testing and software development. En 1st International Research Workshop on Advances and Innovations in Software Testing.
- Park, J. Y., Im, K. S., & Kim, J. S. (2011). The role of IT human capability in the knowledge transfer process in IT outsourcing context. *Information & Management*, 48(1), 53-61. <https://doi.org/10.1016/j.im.2010.12.001>
- Pérez-Castillo, R., García-Rodríguez de Guzmán, I., & Piattini, M. (2011). Business process archeology using MARBLE. *Information and Software Technology*, 53 (10), 1023-1044. <https://doi.org/10.1016/j.infsof.2011.05.002>
- Project Management Institute. (2021). A guide to the project management body of knowledge (PMBOK guide) (7.^a ed.). Project Management Institute.
- Poston, R. S., Simon, J. C., & Jain, R. (2007). Managing offshore outsourcing of software testing. En 1st International Research Workshop on Advances and Innovations in Software Testing.
- Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9.^a ed.). McGraw-Hill Education.
- Sánchez, D. M., Cavero, J. M., & Marcos, E. (2009). The concepts of model in information systems engineering: A proposal for an ontology of models. *The Knowledge Engineering Review*, 24(1), 5-21. <https://doi.org/10.1017/S0269888908001520>
- Sanz, A., García, J., Saldaña, J., Amescua, A., & Velasco, M. (2009). Definición de un modelo para una factoría de pruebas de software. *Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos*, 3(4).
- Sanz, A., Saldaña, J., García, J., & Gaitero, D. (2008). TestPAI: A proposal for a testing process area integrated with CMMI. En *European Systems and Software Process Improvement and Innovation*. Dublin, Irlanda.
- Serna, E., & Arango, F. (2011). Prueba del software: Más que una fase en el ciclo de vida. *Revista de Ingeniería*, (35), 34-40.

- Serrano, J., et al. (2021). *An IT Service Management literature review: Challenges, benefits, opportunities, and implementation practices*. Information, 12(3), 111.
<https://doi.org/10.3390/info12030111>
- Sommerville, I. (2016). Software engineering (10.^a ed.). Pearson Education.
- The Business Research Company. (2024). Outsourced software testing services global market report 2024. <https://www.thebusinessresearchcompany.com/report/outsourced-software-testing-services-global-market-report>
- TMMi Foundation. (2022). Test Maturity Model Integration (TMMi®) Guidelines for Test Process Improvement Release 1.3.
- Venkatasubramanian, A., & Vinoline, V. (2010). Software test factory: A proposal of a process model to create a test factory. International Journal of Computational Intelligence Techniques, 1(1), 14-19.
- Venkatraman, N., & Prescott, J. E. (1990). Environment-strategy coalignment: An empirical test of its performance implications. Strategic Management Journal, 11(1), 1–23.
<https://doi.org/10.1002/smj.4250110102>
- Wang, F., Sun, X., & Zheng, Y. (2022). How do information strategy and information technology governance affect firm performance? Frontiers in Psychology, 13, 1023169. <https://doi.org/10.3389/fpsyg.2022.1023169>
- Weill, P., & Ross, J. (2004). IT governance: How top performers manage IT decision rights for superior results. Harvard Business School Press.
- Willcocks, L., & Choi, C. J. (1995). Co-operative partnership and total IT outsourcing: From contractual obligation to strategic alliance. European Management Journal, 13(1), 67-78. [https://doi.org/10.1016/0263-2373\(94\)00059-L](https://doi.org/10.1016/0263-2373(94)00059-L)
- Yang, Y., Onita, C., & Dhaliwal, J. (2008). Conceptualizing software testing as a service: The TESTQUAL measurement for testing service quality. En Second International Research Workshop on Advances and Innovations in Software Testing.
- Yin, R. K. (2018). *Case study research and applications: Design and methods* (6th ed.). SAGE Publications.

ANEXOS

ANEXO 1 - Cuestionario “Modelo de Procesos para un Servicio de Testing”

“Modelo de Procesos para un Servicio de Testing de Aplicaciones”

Objetivo: Establecer cómo influye un Modelo de Procesos para un Servicio de Testing de Software en la calidad del producto final en una empresa de telecomunicaciones.

Instrucciones: Los siguientes enunciados se utilizan con fines científicos, por favor conteste de forma cuidadosa y sincera marcando su respuesta con una cruz o aspa.

I. Datos del encuestado:

1. Cargo:

Team Leader ()

Gestor de Pruebas ()

Consultor de calidad ()

Analista de pruebas ()

Ingeniero de pruebas ()

Tester ()

II. En cada enunciado, marcando con una cruz o aspa indicando si está:

- Totalmente en desacuerdo = 1
- En desacuerdo = 2
- Ni de acuerdo ni en desacuerdo = 3
- De acuerdo = 4
- Totalmente de acuerdo = 5

Dimensión 1: Gestión de la Ejecución de Pruebas					
1.	El proceso de Estimación ha mejorado la planificación y el uso de recursos en nuestro servicio de Testing.				
2.	La Certificación de las aplicaciones probadas garantiza un buen funcionamiento en el ambiente de producción.				
3.	La implementación de aplicaciones en el ambiente productivo se realiza de manera efectiva.				
Dimensión 2: Gestión del Servicio de Testing					
4.	La Gestión de la demanda ha permitido una asignación más eficiente de recursos para proyectos de Testing.				
5.	La Gestión de cronograma ha mejorado la planificación temporal de nuestras actividades de Testing.				
6.	El Punto de eficiencia ha ayudado a dimensionar de manera óptima nuestra carga de trabajo en proyectos de Testing.				
Dimensión 3: Gestión de Mejora Continua					
7.	Las actividades de Mejora Continua se han implementado de manera efectiva en nuestro servicio de Testing.				
8.	La evaluación continua de procesos y personas ha contribuido a un programa de mejora continua en nuestra organización.				
9.	Participa activamente en la implementación de acciones de mejora continua en su área de trabajo desde la adopción del nuevo modelo.				
Dimensión 4: Gestión de Infraestructura y Testware					
10.	La tecnología proporcionada para el funcionamiento del servicio de Testing es adecuada.				
11.	Las herramientas de gestión de conocimiento y testware son útiles y eficientes en nuestro servicio de Testing.				
12.	Las herramientas de gestión de conocimiento y testware han mejorado su eficiencia en el proceso de Testing.				
Dimensión 5: Control de la Ejecución de Pruebas					
13.	Los Acuerdos de Niveles de Servicio (ANS) han mejorado el control del rendimiento en nuestro servicio de Testing.				
14.	Los Tipos de ANS establecidos son efectivos para controlar la ejecución y efectividad de las pruebas en nuestro servicio de Testing.				
15.	Los Tipos de ANS han sido efectivos para controlar la ejecución y la efectividad de las pruebas en su proyecto				
Dimensión 6: Gestión de Competencias Profesionales					
16.	La estructura profesional definida para el área de pruebas de software ha mejorado la cualificación del personal en nuestro servicio de Testing.				
17.	La definición de roles y funciones en el proceso de Gestión de Ejecución de Pruebas ha facilitado nuestras actividades de Testing.				
18.	La definición de roles y funciones ha mejorado la coordinación y eficacia en su equipo de Testing.				

ANEXO 2 - Cuestionario “Calidad del Producto Final”

“Modelo de Procesos para un Servicio de Testing de Aplicaciones”

Objetivo: Establecer cómo influye un Modelo de Procesos para un Servicio de Testing de Software en la calidad del producto final en una empresa de telecomunicaciones.

Instrucciones: Los siguientes enunciados se utilizan con fines científicos, por favor conteste de forma cuidadosa y sincera marcando su respuesta con una cruz o aspa.

I. Datos del encuestado:

1. Cargo:

Team Leader ()
Gestor de Pruebas ()
Consultor de calidad ()
Analista de pruebas ()
Ingeniero de pruebas ()
Tester ()

II. En cada enunciado, marcando con una cruz o aspa indicando si está:

- Totalmente en desacuerdo = 1
- En desacuerdo = 2
- Ni de acuerdo ni en desacuerdo = 3
- De acuerdo = 4
- Totalmente de acuerdo = 5

Dimensión 1: Calidad Funcional del Software					
1.	El software que ha pasado por el Servicio de Testing cumple adecuadamente con los requisitos funcionales.				
2.	La cobertura de pruebas en cuanto a los requisitos funcionales es suficiente y completa.				
3.	La tasa de defectos funcionales encontrados es baja, lo que sugiere una alta calidad funcional.				
Dimensión 2: Rendimiento y Eficiencia					
4.	El software muestra un rendimiento rápido y una buena capacidad de respuesta en su funcionamiento.				
5.	Durante la ejecución, el software utiliza eficientemente los recursos de hardware, como memoria y CPU.				
6.	El software mantiene un buen rendimiento incluso bajo cargas de trabajo significativas, lo que indica una alta calidad de rendimiento.				
Dimensión 3: Fiabilidad y Disponibilidad					
7.	El software es confiable y rara vez experimenta fallos inesperados.				
8.	La tasa de disponibilidad del sistema es alta, lo que significa que el software está disponible y en funcionamiento la mayor parte del tiempo.				
9.	El software tiene un alto tiempo promedio entre fallos (MTBF), lo que sugiere una alta fiabilidad.				
Dimensión 4: Percepción del servicio entregado al cliente					
10.	Está satisfecho con la calidad del servicio de Testing de Software entregado al cliente				
11.	Está satisfecho con la comunicación y la capacidad de respuesta del equipo de Testing de Software a las solicitudes y preguntas del cliente				
12.	El equipo de Testing de Software ha demostrado un compromiso efectivo para resolver problemas y abordar los desafíos en el servicio entregado al cliente				
13.	El producto final cumple con sus expectativas				

ANEXO 3 - Flujo y caracterización del Proceso de Gestión de la Ejecución de Pruebas

Objetivo: Diagramar el flujo de actividades del proceso de Gestión de la Ejecución de las Pruebas.

Subproceso de Estimación:

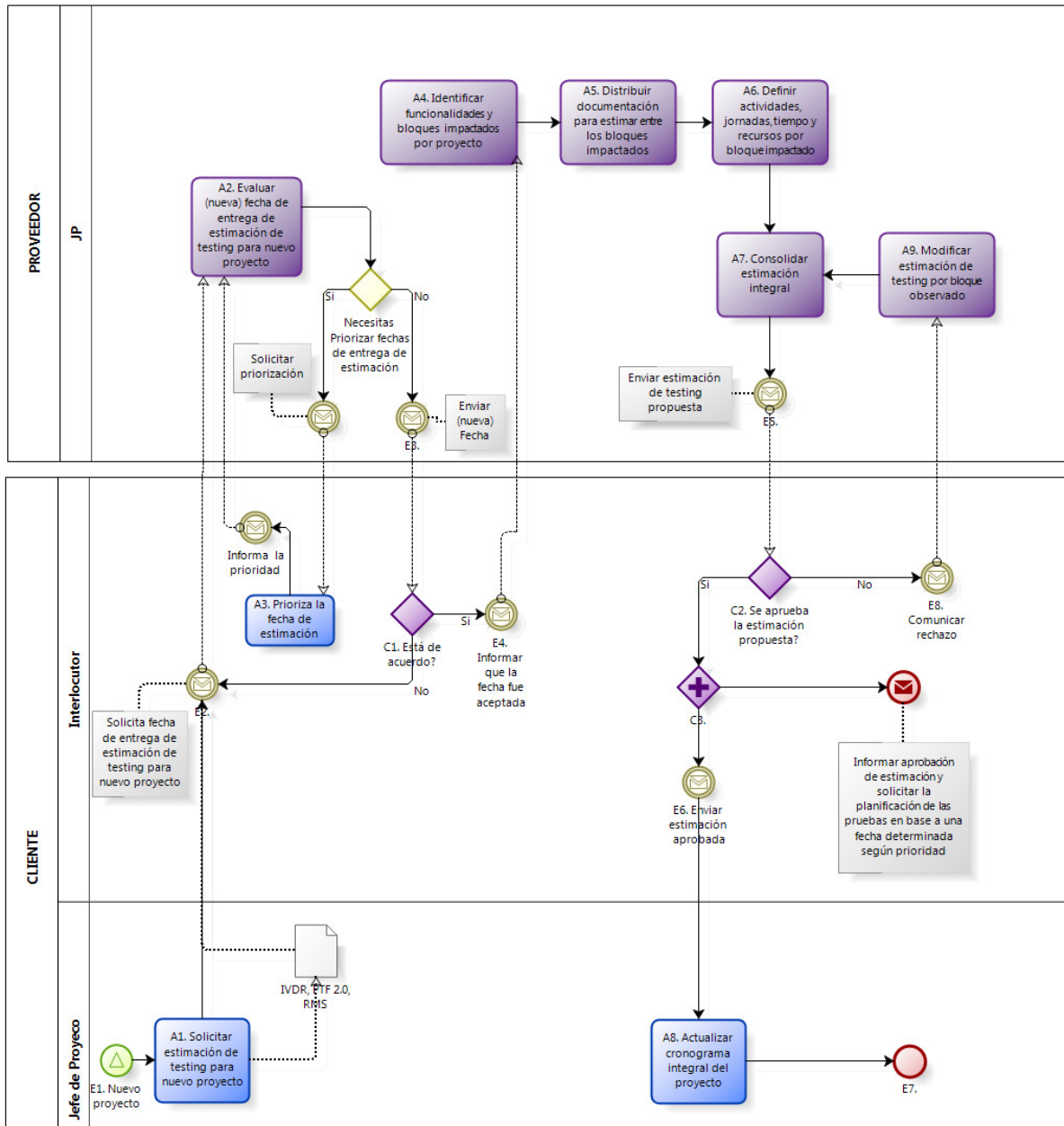


Figura 2: Flujo de Subproceso de Estimación del proceso de Gestión de Ejecución de Pruebas

Caracterización:

Actividad	Entrada	Salida	Descripción	Rol Involucrado	Actividad Previa
E1	-	Nuevo proyecto	El proceso inicia recibiendo el nuevo proyecto a ser testeado	JPI Cliente	
A1	Nuevo proyecto próximo a pasar por Testing	1. Solicitud de estimación para nuevo proyecto. 2. IVDR 3. PTF 2.0 4. RMS	Luego de evaluar el nuevo Proyecto el JPI de Cliente tiene la necesidad de conocer la estimación de Testing para nuevo Proyecto, para esto contacta al Interlocutor de Cliente	JPI Cliente	E1
E2	-	-	Solicita fecha de entrega de estimación de Testing para nuevo proyecto	Interlocutor Cliente	A1 C2 (Si)
A2	1. Correo solicitando fecha de estimación de Testing para nuevo proyecto 2. IVDR 3. PTF 2.0 4. RMS	Fecha de entrega de estimación de Testing para nuevo proyecto	Evaluar fecha de entrega de estimación de Testing para nuevo proyecto	JP / JPI Proveedor	E2 E4
C1	Fecha de entrega de estimación de Testing para nuevo proyecto	Si No	¿Necesitas Priorizar fechas de entrega de estimación?	JP / JPI Proveedor	A2
E3	-	-	Solicitar Priorizar	JP / JPI Proveedor	C1 (Si)
A3	Solicitud de priorización	Fecha de estimación priorizada	Prioriza la fecha de estimación	Interlocutor Cliente	E3
E4	-	Hacia A2	Informa la prioridad	Interlocutor Cliente	A3
E5	-	-	Enviar (nueva) Fecha	JP / JPI Proveedor	C1 (No)
C2	-	Si No (Regresa a E2)	¿Está de acuerdo?	Interlocutor Cliente	E5
E6	-	-	Informar que la fecha fue aceptada	Interlocutor Cliente	C2 (Si)
A4	Correo de aceptación de fecha de estimación de Testing para nuevo proyecto	funcionalidades y bloques impactados identificados	Identificar funcionalidades y bloques impactados por Testing del proyecto	JP / JPI Proveedor	E6
A5	funcionalidades y bloques impactados identificados	Documento de mapeo e identificación de funcionalidades y bloques impactados distribuidos	Distribuir documentación para estimar entre bloques impactados	JP / JPI Proveedor	A4
A6	Documento de mapeo e identificación de funcionalidades y bloques impactados distribuidos	Documentos de estimación de jornadas, tiempo y recursos por bloque impactado	Definir actividades, jornadas, tiempo y recursos por bloque impactado	Jp / Jpi Proveedor	A5
A7	Documentos de estimación de jornadas, tiempo y recursos por bloque impactado	Documento de estimación de jornadas, tiempo y recursos por bloque impactado consolidado	Consolidar estimación integral	Jp / Jpi Proveedor	A6 A9
E7	-	-	Enviar estimación de Testing propuesta	Jp / Jpi Proveedor	A7
C3	Documento de estimación de jornadas, tiempo y recursos por bloque impactado consolidado enviados	Si No	¿Se aprueba la estimación propuesta?	Interlocutor Cliente	E7
E10	-	-	Comunicar rechazo	Interlocutor Cliente	C3 (No)

Actividad	Entrada	Salida	Descripción	Rol Involucrado	Actividad Previa
A9	Rechazo de estimación propuesta por el equipo de Testing enviado	estimación propuesta por el equipo de Testing modificado	Modificar estimación de Testing por bloque observado (Regreso a A6)	Jp / Jpi Proveedor	E10
C4	-	1a: Se envía documento a JPI Cliente para que este pueda actualizar el cronograma integral del proyecto 2a: Informar estimación a Testing	Gateway en paralelo	Interlocutor Cliente	C3 (Si)
E8	-	-	Enviar por correo estimación propuesta por el equipo de Testing aprobada	Interlocutor Cliente	C4 (1a)
A8	Estimación propuesta por el equipo de Testing aprobada	Cronograma integral del proyecto actualizado	Actualizar cronograma integral del proyecto	JPI Cliente	E8
E9	-		No se desarrolla ninguna actividad	JPI Cliente	A8
E11	-	-	Informar aprobación de estimación y solicitar la planificación de las pruebas en base a una fecha determinada según prioridad	Interlocutor Cliente	C4 (2a)

Tabla 1: Caracterización de Subproceso de Estimación del proceso de Gestión de Ejecución de Pruebas

Subproceso de Planificación:

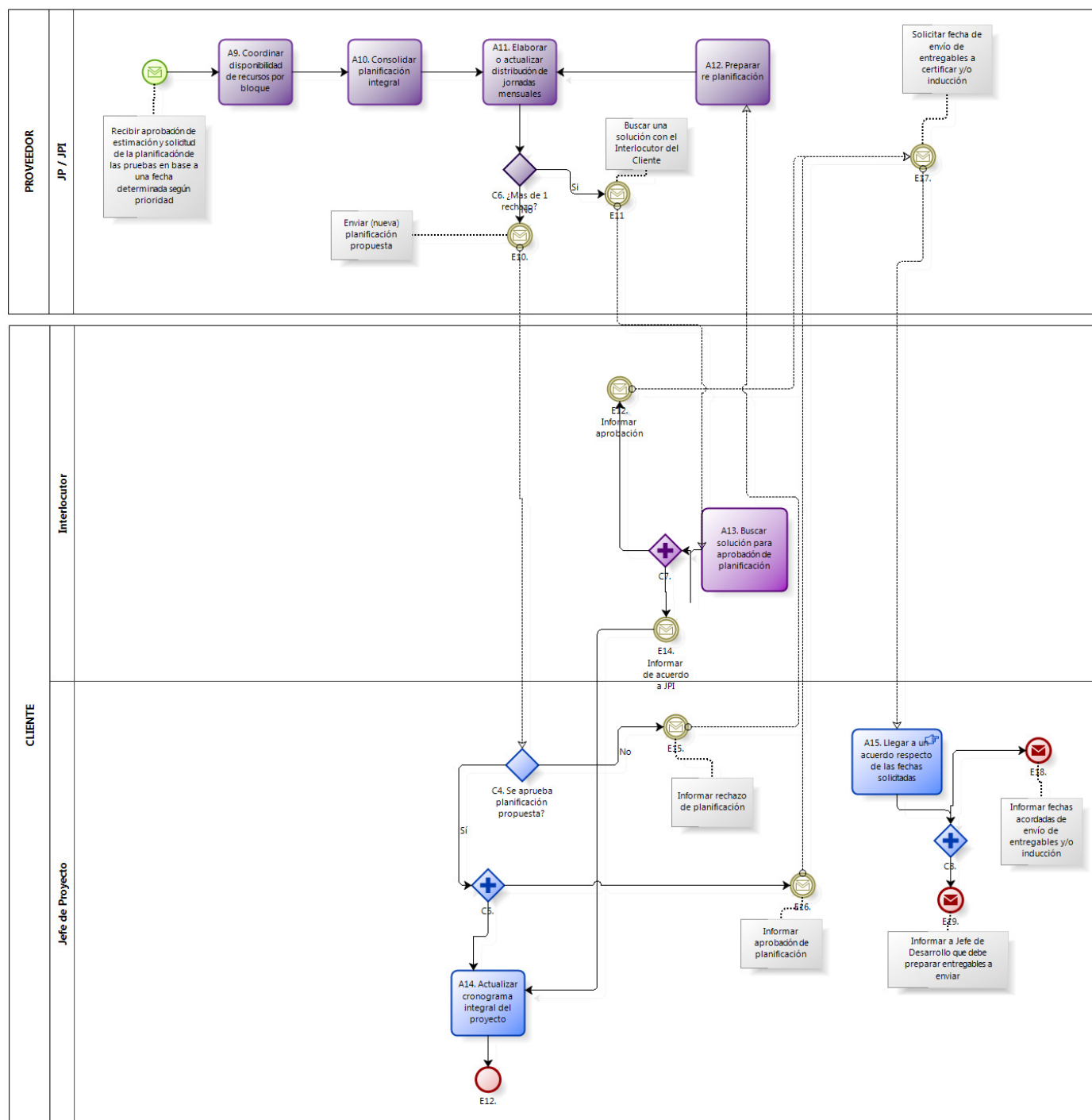


Figura 3: Flujo de Subproceso de Planificación del proceso de Gestión de Ejecución de Pruebas

Caracterización:

Actividad	Entrada	Salida	Descripción	Rol Involucrado	Actividad Previa
E1	-	-	Recibir aprobación de estimación y solicitud de la planificación de las pruebas en base a una fecha determinada según prioridad	JP / JPI Proveedor	-
A1	Solicitud de la planificación de las pruebas en base a una fecha determinada según prioridad	Documento de disponibilidad de recursos por bloque a partir de estimación aprobada por Cliente	Coordinar disponibilidad de recursos por bloque	JP / JPI Proveedor	E1
A2	Documento de disponibilidad de recursos por bloque a partir de estimación aprobada por Cliente	Planificación integral consolidada	Consolidar planificación integral	JP / JPI Proveedor	A1
A3	Planificación integral consolidada	Planificación integral consolidada y documento donde se indica la distribución de jornadas mensuales	Elaborar distribución de jornadas mensuales	JP / JPI Proveedor	A2 A6
C1	-	Si No	¿Se ha recibido más de un rechazo respecto a la aprobación de la planificación presentada a Cliente?	JP / JPI Proveedor	A3
E2	-	-	Enviar por correo planificación propuesta	JP / JPI Proveedor	C1 (No)
C2	-	Si No	¿Se aprueba planificación propuesta?	Interlocutor Cliente	E2
C3	-	1a: Hacia actividad A4 2a: Hacia evento E5	Gateway en paralelo	JPI Cliente	C2 (Si)
A4	Planificación propuesta por el equipo de Testing aprobada	Cronograma integral del proyecto actualizado	Actualizar cronograma integral del proyecto	JPI Cliente	C3 (1a)
E3	-	-	No se desarrolla ninguna actividad	JPI Cliente	A4
E5	-	-	Informar por correo rechazo de planificación	JPI Cliente	C2 (No)
A6	Informe sobre rechazo de planificación propuesta	Nueva planificación propuesta	Preparar replanificación	JP / JPI Proveedor	E5
E6	-	-	Buscar una solución con el Interlocutor de Cliente	JP / JPI Proveedor	C1 (Si)

Actividad	Entrada	Salida	Descripción	Rol Involucrado	Actividad Previa
A5	Solicitud de intermediación para resolver conflicto	Solución basada en repriorización de peticiones	Buscar solución para aprobación de planificación	Interlocutor Cliente	E6
C4	-	1b: Hacia E8 2b: Hacia E9	Gateway en paralelo	Interlocutor Cliente	A5
E7	-	Hacia A4	Informar de acuerdo a JPI	Interlocutor Cliente	C4 (1b)
E8	-	Hacia E8	Informar por correo aprobación a Testing	Interlocutor Cliente	C4 (2b)
E4	-	-	Informar por correo aprobación de planificación	JP / JPI Proveedor	C3 (2a)
E9	-	-	Solicitar por correo fecha de envío de entregables a certificar y/o inducción	JP / JPI Proveedor	E4 E8
A7	Solicitud por correo de fecha de envío de entregables a certificar y/o inducción	Acuerdo respecto de fechas solicitadas por Testing	Llegar a un acuerdo respecto de las fechas solicitadas	JPI Cliente	E9
C5	-	1C: Hacia E10 2C: Hacia E11	Gateway en paralelo	JPI Cliente	A7
E10	-	-	Informar por correo fechas acordadas de envío de entregables y/o inducción	JPI Cliente	C5 (1C)
E11	-	-	Informar por correo a Jefe de Desarrollo que debe preparar entregables a enviar	JPI Cliente	C5 (2C)

Tabla 2: Caracterización de Subproceso de Planificación del proceso de Gestión de Ejecución de Pruebas

Subproceso de Certificación:

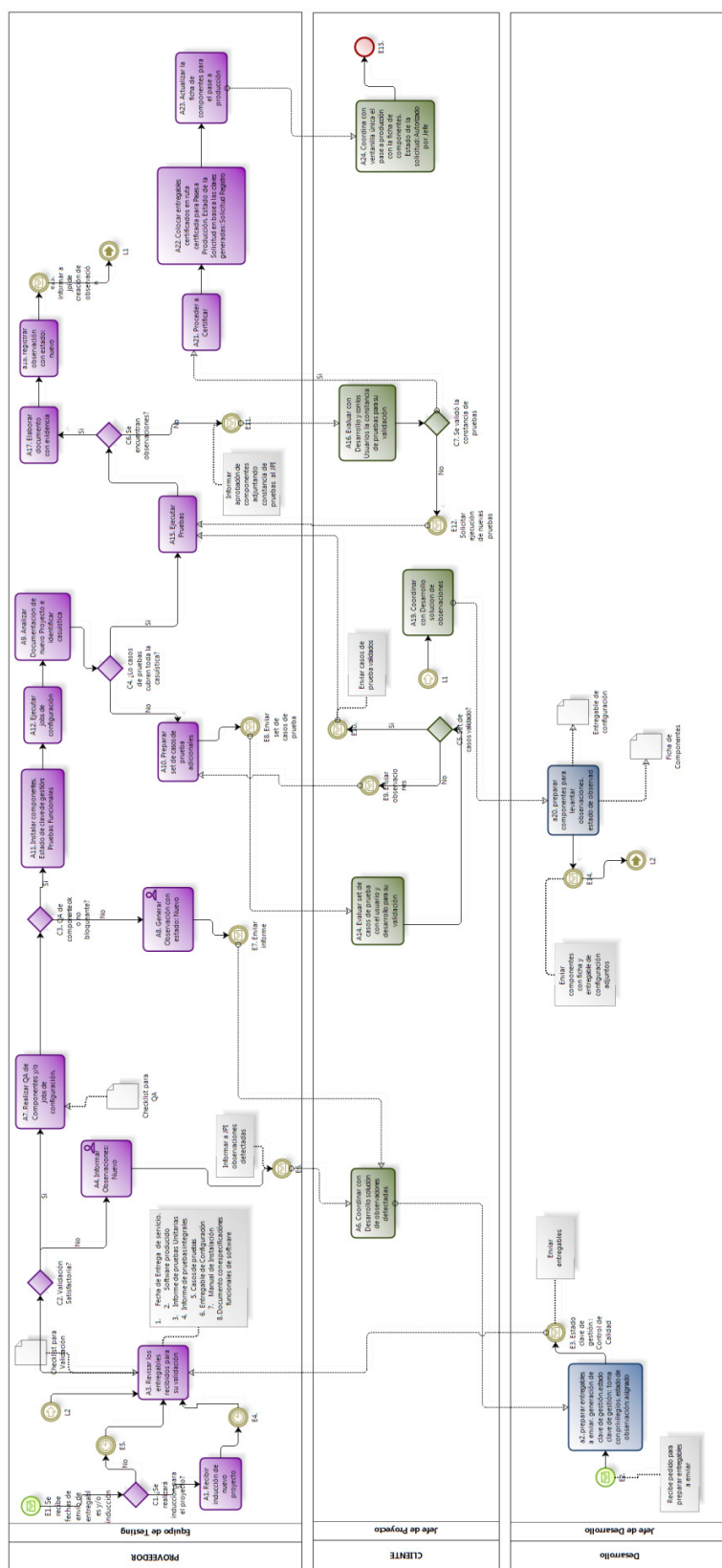


Figura 4: Flujo de Subproceso de Certificación del proceso de Gestión de Ejecución de Pruebas

Caracterización:

Actividad	Entrada	Salida	Descripción	Rol Involucrado	Actividad Previa
E1	-	-	Recibir aprobación de estimación y solicitud de la planificación de las pruebas en base a una fecha determinada según prioridad	JP / JPI Proveedor	-
A1	Solicitud de la planificación de las pruebas en base a una fecha determinada según prioridad	Documento de disponibilidad de recursos por bloque a partir de estimación aprobada por Cliente	Coordinar disponibilidad de recursos por bloque	JP / JPI Proveedor	E1
A2	Documento de disponibilidad de recursos por bloque a partir de estimación aprobada por Cliente	Planificación integral consolidada	Consolidar planificación integral	JP / JPI Proveedor	A1
A3	Planificación integral consolidada	Planificación integral consolidada y documento donde se indica la distribución de jornadas mensuales	Elaborar distribución de jornadas mensuales	JP / JPI Proveedor	A2 A6
C1	-	Si No	¿Se ha recibido más de un rechazo respecto a la aprobación de la planificación presentada a Cliente?	JP / JPI Proveedor	A3
E2	-	-	Enviar por correo planificación propuesta	JP / JPI Proveedor	C1 (No)
C2	-	Si No	¿Se aprueba planificación propuesta?	Interlocutor Cliente	E2
C3	-	1a: Hacia actividad A4 2a: Hacia evento E5	Gateway en paralelo	JPI Cliente	C2 (Si)
A4	Planificación propuesta por el equipo de Testing aprobada	Cronograma integral del proyecto actualizado	Actualizar cronograma integral del proyecto	JPI Cliente	C3 (1a)
E3	-	-	No se desarrolla ninguna actividad	JPI Cliente	A4
E5	-	-	Informar por correo rechazo de planificación	JPI Cliente	C2 (No)
A6	Informe sobre rechazo de planificación propuesta	Nueva planificación propuesta	Preparar replanificación	JP / JPI Proveedor	E5
E6	-	-	Buscar una solución con el Interlocutor de Cliente	JP / JPI Proveedor	C1 (Si)

Actividad	Entrada	Salida	Descripción	Rol Involucrado	Actividad Previa
A5	Solicitud de intermediación para resolver conflicto	Solución basada en repriorización de peticiones	Buscar solución para aprobación de planificación	Interlocutor Cliente	E6
C4	-	1b: Hacia E8 2b: Hacia E9	Gateway en paralelo	Interlocutor Cliente	A5
E7	-	Hacia A4	Informar de acuerdo a JPI	Interlocutor Cliente	C4 (1b)
E8	-	Hacia E8	Informar por correo aprobación a Testing	Interlocutor Cliente	C4 (2b)
E4	-	-	Informar por correo aprobación de planificación	JP / JPI Proveedor	C3 (2a)
E9	-	-	Solicitar por correo fecha de envío de entregables a certificar y/o inducción	JP / JPI Proveedor	E4 E8
A7	Solicitud por correo de fecha de envío de entregables a certificar y/o inducción	Acuerdo respecto de fechas solicitadas por Testing	Llegar a un acuerdo respecto de las fechas solicitadas	JPI Cliente	E9
C5	-	1C: Hacia E10 2C: Hacia E11	Gateway en paralelo	JPI Cliente	A7
E10	-	-	Informar por correo fechas acordadas de envío de entregables y/o inducción	JPI Cliente	C5 (1C)
E11	-	-	Informar por correo a Jefe de Desarrollo que debe preparar entregables a enviar	JPI Cliente	C5 (2C)

Tabla 3: Caracterización de Subproceso de Certificación del proceso de Gestión de Ejecución de Pruebas