



UNIVERSIDAD RICARDO PALMA

ESCUELA DE POSGRADO

MAESTRÍA EN INGENIERÍA INFORMÁTICA CON MENCIÓN EN
INGENIERÍA DE SOFTWARE

CARACTERÍSTICAS DE LA CALIDAD DE SOFTWARE BASADA EN LOS
ESTÁNDARES NTP 12207:2016, ISO 9001 E ISO 9126, PARA LA MEJORA
DEL ÁREA DE DESARROLLO DE SOFTWARE EN LA UNSCH.

AYACUCHO, 2021

TESIS

PARA OPTAR EL GRADO ACADÉMICO DE MAESTRO EN INGENIERÍA
INFORMÁTICA CON MENCIÓN EN INGENIERÍA DE SOFTWARE

AUTOR

PERALTA SOTOMAYOR, KAREL

(ORCID:0000-0000-0000)

ASESOR

NARIO DULANTO, CARLOS FERNANDO

(ORCID:0000-0000-0000)

LIMA, PERÚ

2023

Metadatos Complementarios

Datos de autor

Peralta Sotomayor, Karel

Tipo de documento de identidad del AUTOR: DNI

Número de documento de identidad del AUTOR: 07965285

Datos de asesor

Nario Dulanto Carlos Fernando

Tipo de documento de identidad del ASESOR: DNI

Número de documento de identidad del ASESOR: 16007568

Datos del jurado

JURADO 1: Narvaez Rivero Tessie Nelly, DNI N° 07916766, ORCID 0002-2842-9143

JURADO 2: Aleman Carmona Ana María, DNI N° 40422486, ORCID 0000-0002-9294-5620

JURADO 3: Ricardo Dario Dejo Torres, DNI N° 45810574, ORCID 0000-0003-2529-5581

Datos de la investigación

Campo del conocimiento OCDE: 612357

Código del Programa: 1.02.00



UNIVERSIDAD RICARDO PALMA
UNIVERSIDAD NACIONAL RESOLUCIÓN DEL CONSEJO DIRECTIVO N° 001-2023



Rectorado
Secretaría General

ANEXO N-1

DECLARACIÓN JURADA DE ORIGINALIDAD

Yo, **Karel PERALTA SOTOMAYOR**, con código de estudiante N° **201212624** con DNI N° **07965285**, con domicilio en Av. La Marina N° 735-23 distrito Pueblo Libre, provincia y departamento de Lima, en mi condición de Maestro en Ingeniería Informática con mención en Ingeniería de Software de la Escuela de Posgrado, declaro bajo juramento que:

La presente tesis titulada: "Características de la calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126, para la mejora del área de desarrollo de software en la UNRCPH. Ayacucho, 2021" es de mi única autoría, bajo el asesoramiento del docente Mg. Nario Dulanto, Carlos Fernando, y no existe plagio y/o copia de ninguna naturaleza, en especial de otro documento de investigación presentado por cualquier persona natural o jurídica ante cualquier institución académica o de investigación, universidad, etc; la cual ha sido sometida al antiplagio Turnitin y tiene el 20% de similitud final.

Dejo constancia que las citas de otros autores han sido debidamente identificadas en la tesis el contenido de estas corresponde a las opiniones de ellos, y por las cuales no asumo responsabilidad, ya sean de fuentes encontradas en medios escritos, digitales o de internet.

Asimismo, ratifico plenamente que el contenido íntegro de la tesis es de mi conocimiento y autoría. Por tal motivo, asumo toda la responsabilidad de cualquier error o omisión en la tesis y soy consciente de las connotaciones éticas y legales involucradas.

En caso de falsa declaración, me someto a lo dispuesto en las normas de la Universidad Ricardo Palma y a los dispositivos legales nacionales vigentes.

Surco, 10 de octubre de 2023



Karel Peralta Sotomayor
DNI N° 07965285

* Se debe seleccionar la opción que corresponda, realizar lo mismo en todo el texto del documento.

"Formamos a tres humanos para una cultura de Paz"

CARACTERÍSTICAS DE LA CALIDAD DE SOFTWARE BASADA EN LOS ESTÁNDARES NTP 12207:2016, ISO 9001 E ISO 9126, PARA LA MEJORA DEL ÁREA DE DESARROLLO DE SOFTWARE EN LA UNSCH. AYACUCHO, 2021

INFORME DE ORIGINALIDAD



FUENTES PRIMARIAS

1	docero.tips Fuente de Internet	6%
2	cybertesis.unmsm.edu.pe Fuente de Internet	3%
3	www.scribd.com Fuente de Internet	2%
4	idoc.pub Fuente de Internet	1%
5	documentop.com Fuente de Internet	1%
6	lecasabe.com Fuente de Internet	1%
7	repositorio.unap.edu.pe Fuente de Internet	1%
8	rua.ua.es Fuente de Internet	1%

9	vsip.info Fuente de Internet	<1 %
10	www.grafiati.com Fuente de Internet	<1 %
11	www.coursehero.com Fuente de Internet	<1 %
12	repositorio.ucv.edu.pe Fuente de Internet	<1 %
13	es.scribd.com Fuente de Internet	<1 %
14	oa.upm.es Fuente de Internet	<1 %
15	repositorioinstitucional.uabc.mx Fuente de Internet	<1 %
16	Submitted to Universidad de Jaén Trabajo del estudiante	<1 %
17	moam.info Fuente de Internet	<1 %
18	pt.scribd.com Fuente de Internet	<1 %
19	Submitted to Universidad Ricardo Palma Trabajo del estudiante	<1 %
20	www.monografias.com Fuente de Internet	<1 %

21	repositorio.undar.edu.pe Fuente de Internet	<1 %
22	repositorio.uss.edu.pe Fuente de Internet	<1 %
23	Submitted to Universidad Nacional de Colombia Trabajo del estudiante	<1 %
24	repositorio.uladech.edu.pe Fuente de Internet	<1 %
25	qdoc.tips Fuente de Internet	<1 %
26	repositorio.unsa.edu.pe Fuente de Internet	<1 %

Excluir citas

Activo

Excluir coincidencias < 50 words

Excluir bibliografía

Activo

Dedicatoria:

En primer lugar, a Dios, a mis padres que desde el cielo me protegen, a mis hermanos y hermanas que siempre me motivan a seguir adelante.

A mi familia, Rihanna mi niña adorada e Ivanna mi amada, por su apoyo constante que son la razón de mi constante esfuerzo.

A mis profesores de la Maestría por compartir su sabiduría y conocimientos.

Karel Peralta Sotomayor

Agradecimiento

A la Universidad Ricardo Palma, dónde hice la maestría y me permitió conocer a excelentes catedráticos quienes impartieron sus conocimientos en el transcurso de dos años de posgrado.

Un agradecimiento especial a mí asesor de tesis Mg. Carlos Fernando Nario Dulanto por su amistad y su compromiso de asesoramiento a lo largo del desarrollo de este trabajo de investigación, a los señores jurados gracias por sus críticas y comentarios que ayudaron a perfeccionar este trabajo de tesis.

Un reconocimiento especial al Mg. Carlos García, gracias por su constante motivación a culminar este trabajo. Mis oraciones hasta el cielo.

Índice de Contenido

Introducción	
15 1 Capítulo 1. Planteamiento del problema	17
1.1 Descripción del Problema	17
1.2 Formulación del Problema	19
1.2.1 Problema general	19
1.2.2 Problemas específicos	19
1.3 Importancia y Justificación del estudio.....	20
1.4 Delimitación del estudio	21
1.5 Objetivos de la Investigación	22
1.5.1 Objetivo general	22
1.5.2 Objetivos específicos	22 2
Capítulo 2. Marco Teórico	23
2.1 Marco histórico	23
2.1.1 El progreso histórico de la calidad total	23
2.1.2 Etapas históricas de la evolución del software	24
2.2 Investigaciones relacionadas con el tema	26
2.3 Estructura Teórica y Científica que Sustenta el Estudio	29
2.3.1 CMMI	29
2.3.2 Diseño de CMMI	30
2.3.3 Niveles CMMI	30
2.3.4 Interpretación de cada nivel de madurez.	35

2.3.5	Gestión de la calidad	40
2.3.6	Calidad del software	42
2.3.7	Estándares de software	45
2.3.8	Estándares de ISO 9001	47
2.3.9	Revisiones e inspecciones	51
2.3.10	Medición de software	53
2.3.11	Norma Técnica Peruana - NTP	58
2.3.12	ISO 9001 - Mejora	61
2.3.13	Desarrollo continuo	65
2.4	Definición de Términos Básicos	66
2.4.1	Marco conceptual de la calidad	66
2.4.2	Estándar o especificación	68
2.4.3	Norma ISO 9001	70
2.4.4	La NTP-ISO/IEC 12207	71
2.4.5	La norma ISO/IEC 9126 la calidad interna	71
2.4.6	Modelo de calidad establecido por el estándar ISO 9126	72
2.5	Fundamentos teóricos que sustenta las hipótesis	73
2.6	Hipótesis	76
2.6.1	Hipótesis general	76
2.6.2	Hipótesis específicas	76
2.7	Variables	76 3
Capítulo 3. Marco metodológico		81

3.1	Tipo, Método y Diseño de la Investigación	81
3.2	Población y Muestra	83
3.3	Técnicas e instrumentos de recolección de datos	84
3.4	Descripción de Procedimientos de Análisis de Datos.....	88 4
Capítulo 4. Resultados y análisis de resultados		90
4.1	Resultados	90
4.1.1	Aplicación de la estadística descriptiva	90
4.1.2	Aplicación de la estadística inferencial	108
4.1.3	Contraste de hipótesis de la investigación	114
4.2	Análisis de resultados	122
CONCLUSIONES Y RECOMENDACIONES		130
CONCLUSIONES		130
RECOMENDACIONES		131
5	REFERENCIAS	132
ANEXOS		137
ANEXOS A: Declaración de autenticidad		137
ANEXOS B: Autorización de consentimiento para realizar la investigación		138
ANEXOS C: Matriz de consistencia.....		139

ANEXOS D: Matriz de operacionalización	
140 ANEXOS E: Protocolos o Instrumentos utilizados	
.....	143
ANEXOS F: Formato de instrumentos o protocolos utilizados.....	
144	
ANEXOS G: Tablas de validez y confiabilidad	
154	

Índice de tablas

Tabla 1 Un ejemplo de objetivo cuantitativo	38
Tabla 2 Comparación de los niveles de capacidad y de madurez	39
Tabla 3 Estándares de productos y procesos	46
Tabla 4 Lista de verificación de una inspección	52
Tabla 5 Valor asignado al indicador de madurez, expresado en porcentaje	91
Tabla 6 Valor asignado al indicador de fiabilidad, expresado en porcentaje	93
Tabla 7 Valor asignado al indicador de calidad de servicio, expresado en porcentaje	95
Tabla 8 Valor asignado al indicador de calidad percibida, expresada en porcentaje.....	96
Tabla 9 Valor asignado al indicador de aprender, expresada en porcentaje	99
Tabla 10 Valor asignado al indicador de operar, expresada en porcentaje	100
Tabla 11 Valor asignado al indicador de preparar las entradas e interpretar salidas de un programa, expresada en porcentaje	103
Tabla 12 Valor asignado al indicador de recurso de cómputo, expresada en porcentaje.....	105
Tabla 13 Valor asignado al indicador de códigos requeridos por un programa, expresada en porcentaje	106
Tabla 14 Prueba de hipótesis de la normalidad de datos asociados a las características que definen la calidad de software.....	109
Tabla 15 Prueba de hipótesis de la aleatoriedad de datos asociados a las características que	

definen la calidad de software.....	
111 Tabla 16 Estadísticos de la muestra correspondiente a funcionalidad	
.....	114
Tabla 17 Prueba de hipótesis para puntuación media de la funcionalidad	
115	
Tabla 18 Estadísticos de la Usabilidad	117
Tabla 19 Prueba de hipótesis para puntuación media de la usabilidad	
117	
Tabla 20 Estadísticos de la eficiencia	
119 Tabla 21 Prueba de hipótesis para puntuación media de la eficiencia	
.....	119
Tabla 22 Estadísticos de la calidad de software	
121	
Tabla 23 Prueba de hipótesis para la puntuación media de la calidad de software	
121	

Índice de figuras

Figura 1 100 años de hitos históricos seleccionados en el movimiento de calidad global	24
Figura 2 Esquema de la evolución histórica del desarrollo de software y del concepto de calidad asociado	25
Figura 3 Representación continua del nivel de capacidad.	31
Figura 4 Representaciones por etapas de CMMI	34
Figura 5 Evaluación de la calidad y desarrollo de software	41
Figura 6 Componentes de calidad del software	44
Figura 7 Evaluación basada en el proceso	45
Figura 8 Procesos centrales de ISO 9001	48
Figura 9 ISO 9001 y la administración de la calidad	50
Figura 10 El proceso de revisión de software	51
Figura 11 Medidas de predicción y control	55
Figura 12 Relaciones entre los atributos internos y externos del software	57
Figura 13 Grupos de procesos del ciclo de vida	60
Figura 14 Integración del ciclo PDCA en el sistema de gestión de la calidad.	66
Figura 15 Coste y valor en función de la calidad del diseño.	69

Figura 16 Fases de uso de QFD.	70
Figura 17 Contexto teórico, metodológico y aplicativo de la calidad de software basada en los estándares para el desarrollo de software	75
Figura 18 Tendencia del valor asignado al indicador de madurez, expresado en porcentaje acumulado según modalidad	92
Figura 19 Tendencia del valor asignado al indicador de fiabilidad, expresado en porcentaje acumulado según modalidad	94
Figura 20 Tendencia del valor asignado al indicador de la calidad de servicio expresado en porcentaje acumulado según modalidad.	95
Figura 21 Tendencia del valor asignado al indicador de la calidad percibida, expresado en porcentaje acumulado según modalidad	97
Figura 22 Tendencia del valor asignado al indicador de aprender, expresado en porcentaje acumulado según modalidad	99
Figura 23 Tendencia del valor asignado al indicador de operar, expresado en porcentaje acumulado según modalidad	101
Figura 24 Tendencia del valor asignado al indicador de preparar las entradas e interpretar salidas de un programa, expresado en porcentaje acumulado según modalidad	103
Figura 25 Tendencia del valor asignado al indicador recurso de cómputo, expresado en porcentaje acumulado según modalidad	105
Figura 26 Tendencia del valor asignado al indicador de códigos requeridos por un programa, expresado en porcentaje	107

Resumen

La tesis tiene el propósito, determinar de qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora el desarrollo de software en la UNSCH, parte de la situación actual en que se encuentra la Unidad de Informática de la Universidad, que ha implementado un software que está en prueba para automatizar sus procesos académicos y administrativos. La metodología tiene un enfoque cuantitativo de una investigación aplicada, de nivel explicativa, con diseño de investigación pre-experimental de tipo transversal aplicado sobre una muestra censal a la que se evaluó un cuestionario de 37 ítems con escala de Likert cuya confiabilidad es de nivel alto con un alfa de Cronbach de 0.88, se aplicó la estadística descriptiva para especificar los atributos de la calidad de software y el test T-Student para contrastar las hipótesis. Los resultados revelan que el modelo conceptual definido por los estándares que aportan las normas 12207:2016, ISO 9001 e ISO 9126, ha permitido identificar con un 95% de confianza y 5% de significancia, que en los atributos de calidad los desarrolladores de software lograron características de calidad media que comprende la funcionalidad, usabilidad y eficiencia ($t = -4,993$; $p = 0.004 < \alpha = 0,05$, $\bar{x} - \mu = -32,833$), asimismo, el diagnóstico determina ajustar el software instalado en la UNSCH a los estándares utilizados como parámetros de las normas 12207:2016, ISO 9001 e ISO 9126.

Palabras clave: Calidad de software y desarrollo de Software según normas y estándares.

Abstract

The thesis has the purpose of determining how the implementation of software quality characteristics based on the NTP 12207:2016, ISO 9001 and ISO 9126 standards improves software development at UNSCH, starting from the current situation in which the Computer Science Unit of the University finds itself, that has implemented a software that is being tested to automate its academic and administrative processes. The methodology has a quantitative approach of an applied research, of explanatory level, with a pre-experimental research design of transversal type applied on a census sample which was evaluated with a questionnaire of 37 items with Likert scale whose reliability is of high level with a Cronbach's alpha of 0.88, descriptive statistics was applied to specify the attributes of software quality and the T-Student test to contrast the hypotheses. The results reveal that the conceptual model defined by the standards provided by the 12207:2016, ISO 9001 and ISO 9126 standards, has allowed to identify with 95% confidence and 5% significance, that in the quality attributes software developers achieved medium quality characteristics that include functionality, usability and efficiency ($t = -4.993$; $p = 0.004 < \alpha = 0.05$, $\bar{x} - \mu = -32.833$), likewise, the diagnosis determines to adjust the software installed at UNSCH to the standards used as parameters of the 12207:2016, ISO 9001 and ISO 9126 standards.

Keywords: Software quality and software development according to norms and standards.

Introducción

La incertidumbre de contar con productos de calidad del software se presentó a inicios de los años 1960, los primeros sistemas presentaban características deficientes como la lentitud, poca fiabilidad, un mantenimiento engorroso y sobre todo no tan amigable en su uso ya que no mostraban amigabilidad ni deducción de los procesos. Estos procedimientos de administración de calidad, empleadas con nuevas tecnologías y revisiones más exhaustivas en las pruebas de software, condujeron a mejoras en la implementación de la calidad de software.

El logro de un software con calidad implica la utilización y empleo de estándares en las etapas de desarrollo como el Análisis, Diseño, Desarrollo, Pruebas, Producción y Mantenimiento. que permitirán uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que elevarán la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

La NTP-ISO/IEC 12207 establece la tecnología, los procesos del ciclo de vida del software en un marco de referencia común para los procesos del ciclo de vida del software donde se incluye el monitoreo de desarrollo de software desde la dotación del suministro, desarrollo, operación y mantenimiento del nuevo software implementado.

El estándar ISO-9126 fija que cualquier componente que engloba la implementación de calidad de software puede ser descrito de acuerdo a sus características básicas como: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad.

En base a lo mencionado en los párrafos anteriores sobre el desarrollo de software empleando las normas NTP e ISOS el área de desarrollo de software de la Universidad Nacional de San Cristóbal de Huamanga ejecuta los requerimientos de desarrollo de los nuevos softwares empleando solo las metodologías de desarrollo de software y no sigue los estándares de desarrollo basados en los estándares de desarrollo de la NTP e ISO, para que de esta manera se garantice que los nuevos productos cumplan con los estándares en mención.

De esta manera, para llevar a cabo este trabajo de tesis, ha sido desarrollado en cuatro capítulos y un apartado final. En el primer capítulo, se describe el estado del problema del trabajo de tesis para ello se plantea el problema general, como los problemas específicos. Por otro lado, se desarrolla la importancia, justificación y delimitación del estudio, aislando la última parte con delimitaciones espaciales, temporales y teóricas. Finaliza el capítulo, planteando adecuadamente el objetivo general y los objetivos específicos

En el segundo capítulo comprende el marco teórico. En ella se describirá primero el marco histórico, luego se plasma las investigaciones nacionales e internacionales que guardan relación con el presente trabajo de tesis, se cita la investigación y se desarrolla el análisis y discusión de cada una de las investigaciones relacionadas con esta investigación. Así mismo se desarrollará la estructura científica y teórica que sustenta el estudio y se definen los términos que comprenden la investigación. Finaliza el capítulo con el planteamiento de la hipótesis general y las hipótesis específicas; así mismo se establece la descripción y operacionalización de las variables de estudio.

En el tercer capítulo se desarrollará el marco metodológico. En esta se planteará el enfoque, tipo, método y diseño de la investigación. Por otro lado, se especificará la población y muestra del estudio; y se describen las técnicas e instrumentos de recolección de datos, finalmente se explicará el procedimiento de análisis en base a los resultados obtenidos.

En el cuarto capítulo, se presentará los resultados obtenidos de acuerdo a los instrumentos empleados para ello se empleó el software estadístico SPSS 27. Con estos resultados se construye el análisis correspondiente.

Por último, se desarrollarán las conclusiones y recomendaciones de acuerdo al análisis y obtención de los resultados obtenidos en el cuarto capítulo.

Capítulo 1. Planteamiento del problema

1.1 Descripción del Problema

De acuerdo con Kerlinger y Lee (2002) con respecto a la importancia del planteamiento del problema, afirma lo siguiente:

Al científico le puede resultar difícil de entender, una vaga preocupación por los fenómenos observables y no observables, y una curiosidad acerca de por qué algo es como es. El primer paso y el más importante es generar ideas y expresar el problema de una manera razonable. El problema nunca o rara vez aparece por completo en este punto. (p. 29)

Cuando se desarrollan nuevos sistemas de proyectos de software donde no se cuenta desde la dirección del proyecto el de seguir estándares de calidad basados en normas NTP e ISO es muy probable que se desarrolle soluciones que posteriormente llevará a soluciones que no están acorde al desarrollo de los procesos de la organización, donde no seguir los procesos de la organización nos lleva a la construcción de sistemas que no se utilizará.

Las normas de calidad son importantes en la organización ya que de esta manera se enriquece la confianza de las relaciones transaccionales, mejoran la capacidad de la organización para satisfacer la demanda de la comunidad y se logra mejorar los objetivos de la organización.

El panorama motivo de investigación del trabajo de tesis es la Universidad Nacional de San Cristóbal de Huamanga y específicamente la Oficina de Informática, la implementación de nuevos proyectos de software se siguen metodologías de desarrollo de software, pero no se toma en cuenta las normas NTP e ISO que especifican los estándares de calidad de software. A la fecha se sigue desarrollando proyectos de sistemas bajo el marco de metodologías ágiles.

Hoy en día se habla bastante de los conceptos relacionados con los Sistemas de Gestión de la Calidad y la metodología de los sistemas de Calidad ISO 9001 y, son justamente estos conceptos los que no se aplican en la Unidad Informática de la Universidad Nacional de San Cristóbal de Huamanga, entender que mediante el desarrollo paralelo de las metodologías que se emplean para la construcción de nuevos sistemas y los manuales de

calidad, las políticas de calidad, manual de procedimientos, así como los procedimientos de los sistemas de calidad ISO 9001 logren implantar mejores modelos para los desarrollos de los nuevos sistemas sin que su implementación en la organización supongan un problema de gestión. El objetivo es lograr la satisfacción de los usuarios mediante soluciones óptimas bajo el contexto de lo mencionado.

El propósito de este trabajo de tesis fue desarrollar una metodología para la gestión del desarrollo del software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126, que permita mejorar los procesos de desarrollo de software con la finalidad minimizar errores, asignar responsabilidades, la responsabilidad de los usuarios debe asumirse para que posteriormente no se acepten peticiones de nuevos cambios al final de la implementación del proyecto en desarrollo y de esta manera minimizar la frecuencia de solicitudes de modificaciones expresadas por los usuarios finales en las etapas finales de la producción del nuevo sistema.

1.2 Formulación del Problema

1.2.1 Problema general

¿De qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora el desarrollo de software en la UNSCH? Ayacucho, 2021?

1.2.2 Problemas específicos

PE1. ¿De qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la funcionalidad del desarrollo de software en la UNSCH?

PE2.

¿De qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la usabilidad del desarrollo de software en la UNSCH?

PE3.

¿De qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la eficiencia del desarrollo de software en la UNSCH?

1.3 Importancia y Justificación del estudio

Importancia

El presente trabajo de tesis es importante porque su principal enfoque estará orientado a la implementación de normas NTP y estándares ISO, patrones de diseño de evaluación de software que permitan brindar soporte en la amigabilidad de uso de las herramientas construidas como solución de software para que se cuente en la institución y que se utilice.

Justificación

Este trabajo de tesis justifica su realización porque se buscará mostrar la importancia de contar con una evaluación de la influencia de como la calidad del software permitirá contar con un buen sistema para el desarrollo de las diversas actividades que tiene que hacer la comunidad universitaria, también servirá de sustento para el enriquecimiento como fuente de información y contar con un modelo de construcción de software de calidad en beneficio de la comunidad universitaria.

El presente trabajo de tesis se justifica por las siguientes razones:

Es original porque la implementación de calidad de software basado en las normas técnicas peruanas e ISO son la forma de medir que la implantación de los nuevos proyectos

de desarrollo de software cumpla con los estándares de calidad por ende su uso por los miembros de la organización.

Teórica

El presente trabajo de tesis tiene un valor teórico y beneficio metodológico porque servirá como ejemplo para el desarrollo de los nuevos sistemas de software, por tal razón el desarrollo y uso como referencia favorece en los resultados y desarrollo futuro de las soluciones de desarrollo de software y que estas sean de calidad así también sean modelos de consulta para los estudiantes de la carrera de ingeniería de sistemas en su formación de pregrado.

Metodológica

Como metodología se pretende hacer posible en demostrar que la calidad de software es importante tener bien conceptualizado el flujo de procesos que se desarrollan en las entidades para de acuerdo a ello muestre lo que se quiere contar y tener como herramienta el uso de las Normas Técnicas Peruanas e ISO.

Práctica

La justificación práctica de este estudio es de suma importancia para los investigadores ya que permitirá contar con una investigación que les permita tener claro la idea de cómo influencia la calidad de software en los sistemas construidos para una entidad sean sólidas los conocimientos y tener claro de cómo se relaciona esta influencia entre la calidad de software con los sistemas construidos siguiendo estándares basados en la Normas Técnicas Peruanas e ISO.

Este trabajo de tesis aporta el conocimiento de las Normas Técnicas Peruanas NTP e ISO para establecer el nivel de aplicación de las normas y modelos de calidad de software basadas en ISO que establecen los requerimientos de aplicación de normas y modelos de calidad de software.

1.4 Delimitación del estudio

Espacial

La investigación espacialmente contemplada para el presente trabajo de tesis planteado se desarrollará en la Universidad nacional de San Cristóbal de huamanga específicamente en el área de desarrollo de software. Sin embargo, para la toma de información será necesario trabajar con toda la población, muestra censal.

Temporal

Para el presente trabajo de tesis se hizo la toma de datos durante el año 2021 de la entidad objeto de estudio.

Teórica

La investigación teóricamente queda delimitado a la implementación basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 empleadas de manera conjunta donde se quiere implementar los entandares de las NTP e ISO para mejorar los procesos de desarrollo de software en la Oficina de Informática de la Universidad Nacional de San Cristóbal de Huamanga.

1.5 Objetivos de la Investigación

1.5.1 Objetivo general

Determinar de qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora el desarrollo de software en la UNSCH.

1.5.2 Objetivos específicos

OE1. Determinar de qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la funcionalidad del desarrollo de software en la UNSCH.

OE2.

Determinar de qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la usabilidad del desarrollo de software en la UNSCH.

OE3. Determinar de qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la

eficiencia del desarrollo de software en la UNSCH.

Capítulo 2. Marco Teórico

2.1 Marco histórico

En el marco histórico del presente trabajo de tesis, se tendrá como determinación el de mostrar la evolución histórica para que nos posicione en el entorno y que nos permita identificar el contexto del estudio. En ese sentido, Carrasco (2014) señala que “forma parte del planteamiento del problema de investigación, ya que contiene referencias fácticas sobre las características históricas del problema en estudio” (p.156).

2.1.1 El progreso histórico de la calidad total

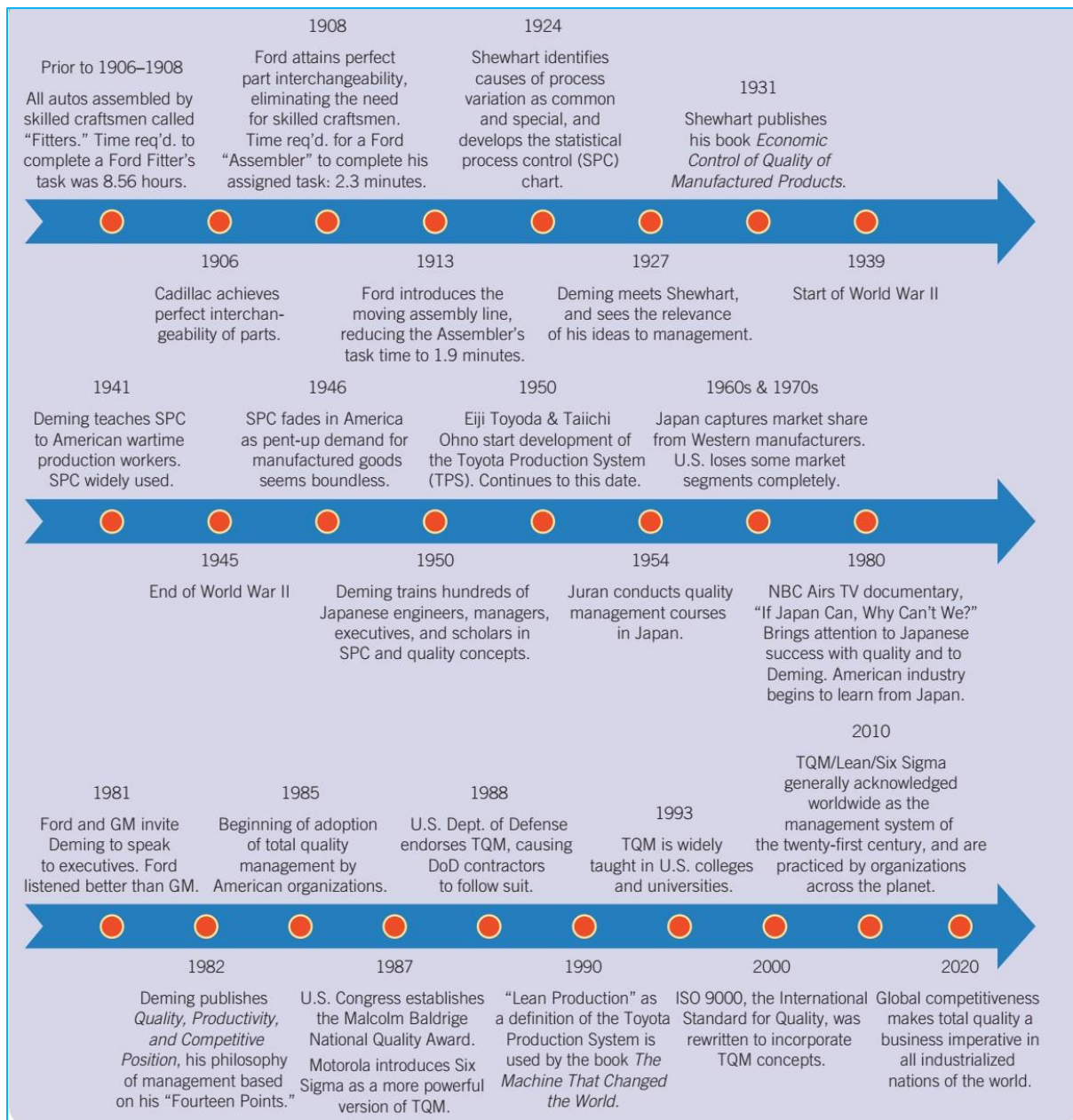
La investigación de Frederick Taylor sobre el tiempo y el movimiento en la década de 1920 sentó las bases del movimiento de la calidad global. La figura 1 muestra algunos de los acontecimientos más significativos en la historia del movimiento de la calidad global desde la época de Taylor. Taylor se ha ganado el apodo de "Padre de la Gestión Científica".

La separación de la planificación y la ejecución es la característica más esencial de la gestión científica. Aunque la división del trabajo dio lugar a aumentos significativos de la producción, abolió efectivamente la práctica anterior de que una persona altamente

competente ejecutara todas las actividades necesarias para hacer un producto de calidad. Esa persona hacía las veces de director general, trabajador de producción y supervisor de control de calidad, todo en uno. Esto fue eliminado por la gestión científica de Taylor, que convirtió la planificación en responsabilidad de la dirección y la producción en responsabilidad del trabajo. Era vital desarrollar un departamento de calidad distinto para evitar que la calidad se perdiera. Estos departamentos tuvieron un comienzo inestable, y no estaba claro quién era responsable de la calidad (Goetsch y Davis, 2016).

Figura 1

100 años de hitos históricos seleccionados en el movimiento de calidad global



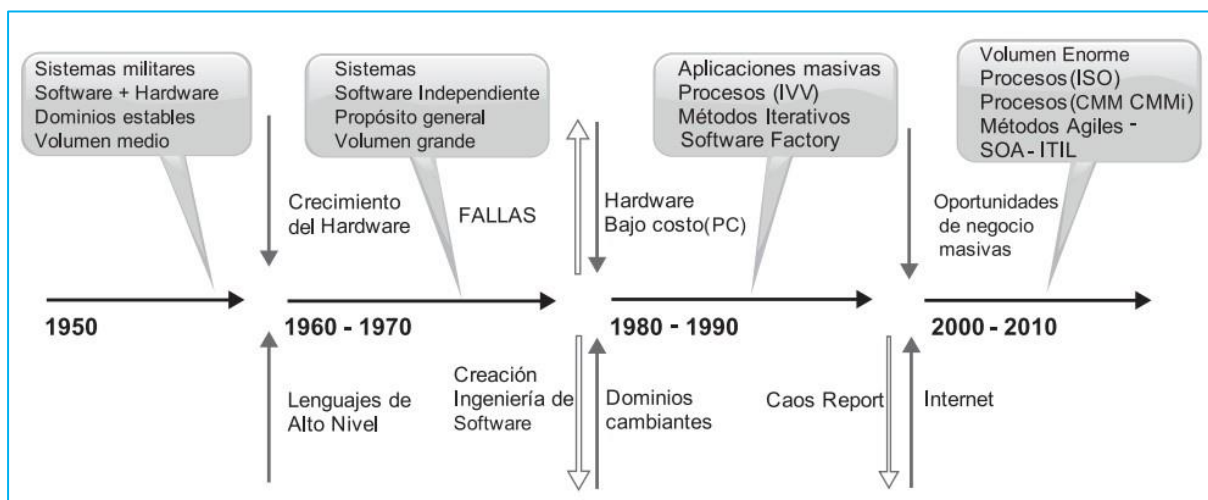
Nota. Tomado de Goetsch y Davis, 2016, p. 5.

2.1.2 Etapas históricas de la evolución del software

Podemos decir que la mayoría del software se desarrollaba y era utilizado por la misma persona u organización donde la persona que lo escribía, hacia las pruebas y ejecutaba y corregía si presentaba fallas. El diseño era un proceso implícito, ejecutado en la mente de alguien.

Haciendo un resumen bajo un modelo simplificado como se muestra en la figura según Pantaleo (2011) las etapas más importantes que enmarcaron en forma secuencial la evolución del desarrollo de software.

Figura 2 Esquema de la evolución histórica del desarrollo de software y del concepto de calidad asociado



Nota. Tomado de Pantaleo, 2011, p. 20.

Bajo esta figura el desarrollo de software se inicia en la década del cincuenta. Los cambios y avances en el desarrollo de software se hicieron en los Estados Unidos de América (EE. UU.) dentro del entorno militar. Desde entonces el desarrollo de las aplicaciones de software fueron implementados para un determinado hardware específico, donde para cada etapa de la evolución del software las aplicaciones eran desarrolladas para un hardware dedicado, sistemas que involucraban al software como una de sus partes. Posteriormente para lograr la (calidad) que es lo que se buscaba desde sus inicios se hacían diversas pruebas y posterior corrección para tener un producto aceptable. A partir de estos acontecimientos varias empresas como Digital Equipment Corporation (DEC) e IBM se involucraron como participantes (Pantaleo, 2011).

Posteriormente ya era inminente el avance de la tecnología electrónica (hardware) así como la aparición de los lenguajes de alto nivel que determino el inicio del desarrollo de sistemas independientes tanto del hardware como del software y fue con la implementación de soluciones comerciales para algunas empresas. A partir de estos acontecimientos nace por ejemplo el sistema de reservación de pasajes aéreos que influyeron para ser catalogados como soluciones de software comerciales (Pantaleo, 2011).

En la década de los sesenta se dio un echo importante que marcó un punto de cambio en esta evolución. Este hecho fue el incremento del presupuesto inicial, también la postergación de la fecha de entrega planeada inicialmente y un proyecto que marco este hecho fue proyecto de desarrollo del sistema operativo OS/360 de IBM. Este proyecto inicio la alerta en el sentido de la necesidad de contar con métodos de desarrollo de software y métodos de análisis para que se garantice la calidad del software construido (Pantaleo, 2011).

2.2 Investigaciones relacionadas con el tema

Tras revisar los diferentes repositorios de tesis relacionadas al tema materia de investigación se hallaron los siguientes trabajos de investigación:

Moreno (2020) en su tesis titulada “Modelo de gestión de calidad basada en los estándares NTP 12207, ISO 9001 E ISO 9126, para los procesos de desarrollo de software: caso RENIEC”, [Tesis de Maestría, Universidad Nacional mayor de San Marcos].

En la Sub Gerencia de Ingeniería de software de RENIEC, existen diversos problemas referentes al desarrollo de los proyectos software, el principal inconveniente es no tener un modelo de calidad, como se evidencia en la cantidad de defectos en los proyectos, insatisfacción de usuarios y un déficit gestión documentaria.

El objetivo de la investigación fue implementar un modelo de gestión de calidad, basada en los estándares NTP 12207, ISO 9001 e ISO 9126, en el desarrollo de software cuyos resultados evidencien que existe disminución en los defectos en los proyectos

desarrollados, mejora en la satisfacción de los usuarios de RENIEC, mejora en la gestión documentaria y mejora en la funcionalidad en la gestión de desarrollo.

Espejo (2016) en su tesis titulado “Modelo de aseguramiento de la calidad en el proceso de desarrollo de software basado en los modelos de madurez de capacidades (CMMi), proceso de software para equipos (TSP) y personas (PSP)”, [Tesis de Maestría, Universidad Nacional mayor de San Marcos].

Hoy en día las empresas que desarrollan software se han reinventado y se han convertido en sus socios estratégicos de sus clientes mediante de la generación de valor, la entrega de productos que soportan sus diversos procesos de negocio, optimizando su competitividad y facilitando su adaptación a los cambios del entorno.

El objetivo de su investigación es plantear y asegurar que un modelo propuesto se basa en tres modelos "Modelos de Madurez de Capacidades Integrado, Proceso de Software para Equipos y Proceso de Software para Personas.

Barrero (2018) en su tesis titulado “Sistema Qsource en la calidad del software desarrollado en RPG”, [Tesis de Maestro, Universidad César Vallejo].

La base teórica de la variable calidad del software se sustentó en las técnicas de revisión y aseguramiento de calidad de software de Pressman, en el modelo de calidad del software ISO/IEC 2510 y en IEEE 1061-1998 - IEEE Standard for a Software Quality Metrics. La base teórica para la variable sistema QSOURCE se sustenta en las técnicas de revisión y aseguramiento de calidad de software de Pressman, en el modelo de calidad de software para RPG.

El objetivo de su investigación es determinar el efecto de la implementación del Sistema QSOURCE en la mejora de la calidad de Software hecho en RPG en una Institución

Bancaria.

Fuertes (2002) en su tesis titulada “Modelo de calidad para el software orientado a objetos”, [Tesis doctoral, Universidad Politécnica de Madrid].

El desarrollo de software en la actualidad ha logrado una gran importancia en todos los ámbitos de la vida cotidiana. Es cierto que la calidad del software tiene un rol fundamental en el desarrollo de los proyectos de software, aunque en ocasiones no se le presta la suficiente atención, razón que fundamenta para el desarrollo de la presente tesis.

En su objetivo de su tesis doctoral plantea la necesidad de un modelo de calidad completo. Este modelo de calidad muestra cómo la calidad del software se descompone en una serie de factores y éstos, a su vez, se descomponen en un conjunto de criterios medibles utilizando medidas.

Portela (2016) en su tesis titulada “El liderazgo transformador en la gestión de la calidad. Un estudio basado en el modelo EFQM”, [Tesis doctoral, Universidad de Alicante - España].

Este dinamismo ha hecho que la calidad fuese percibida de una forma muy distinta a lo largo de la historia. La calidad ha existido desde siempre, pero es a finales del siglo XIX cuando, en las empresas, empieza a hablarse de este término de una manera formal.

En su objetivo manifiesta que se infiere en un desafío para las empresas que tienen que mejorar su competitividad como consecuencia de la desaparición de las barreras entre países y la existencia de productos con altos estándares de calidad, innovación y diseño. En este contexto económico, global y competitivo, las empresas necesitan buscar estrategias que les permitan afrontar con éxito los retos que se les presentan, donde se plantean tres variables básicas en las que descansa la competitividad: la calidad, la innovación tecnológica y el

diseño industrial. Junto a estos aspectos aparecen otros elementos también importantes para muchas empresas como la preocupación por el medio ambiente, la responsabilidad social corporativa y la gestión de los riesgos laborales, elementos que pueden desarrollarse de manera independiente o a partir de la gestión de la calidad.

2.3 Estructura Teórica y Científica que Sustenta el Estudio

2.3.1 CMMI

CMMI trata de recopilar las características de los procesos efectivos y luego utilizar esta información para proporcionar una guía para mejorar los procesos de una organización.

El objetivo final es facilitar que una organización desarrolle productos o soluciones mejorando su capacidad para gestionar el desarrollo, la adquisición y el mantenimiento de sus productos o servicios.

CMMI plantea tres áreas:

- Desarrollo de productos y servicios: CMMI para el desarrollo (CMMI-DEV)
- Establecimiento y gestión de servicios: CMMI para servicios (CMMI-SVC)
- Adquisición de productos y servicios: CMMI para adquisiciones (CMMI-ACQ).

La primera área centrado en prácticas para desarrollar productos o servicios con una calidad estandarizada con el objetivo de satisfacer las necesidades de los consumidores.

La segunda área busca cubrir todas las actividades que requieren gestionar, establecer y entregar servicios.

La tercera área está centrada en actividades de iniciación y manejo de adquisiciones de productos, servicios, herramientas o equipos (Chaudhary y Chopra, 2017)

2.3.2 Diseño de CMMI

El diseño de CMMI, cómo está estructurado el modelo CMMI Dev, cómo interpretarlo y por qué es importante entender el diseño de CMMI.

Para implementar este modelo en la organización, es necesario seguir un camino específico para entender y que sea fácil la conducción al objetivo final.

El tiempo es siempre un factor importante, así que la pregunta sería: "¿Con qué facilidad podemos entender y adaptar este modelo?"

Empecemos por desglosar el diseño del Modelo CMMI DEV. ¿Qué significa CMMI? ¿Y qué es el Desarrollo o DEV?

CMMI es el acrónimo de Capability Maturity Model Integration:

Capacidad: Cuando queremos implantar y conseguir una mejora del proceso en un área de proceso individual.

Madurez: Cuando queremos implantar y conseguir una mejora del proceso en un conjunto de áreas de proceso que están predefinidas.

Modelo: Generado a partir del Marco CMMI.

Integración: Utiliza una combinación de modelos seleccionados (por ejemplo, CMMI para el software, la ingeniería de sistemas y el desarrollo de productos integrados) que se integran en un ambiente único, CMM-I. La "I" significa integración (Chaudhary y Chopra, 2017).

2.3.3 Niveles CMMI

En CMMI, cada nivel de madurez se dispone de áreas de proceso, y estos niveles se logra de forma progresiva.

La capacidad y la madurez están relacionados a los niveles de capacidad y a los niveles de madurez.

Niveles de capacidad

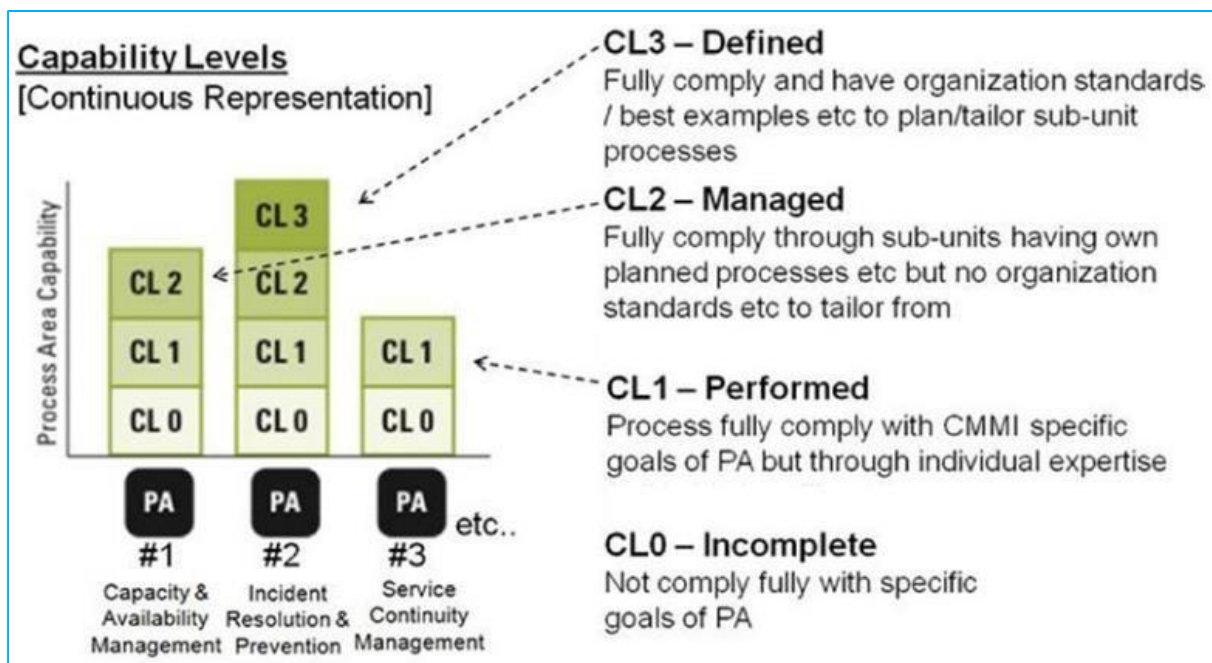
El nivel de capacidad es un camino que garantiza que una organización mejorará un área de proceso personal o un conjunto de zonas de proceso de manera incremental.

Niveles de capacidad:

□ Nivel 0: Incompleto

- Nivel 1: Realizado
- Nivel 2: Gestionado
- Nivel 3: Definido

Figura 3 Representación continua del nivel de capacidad.



Nota. Tomado de Chaudhary y Chopra, 2017, p. 10.

Nivel de capacidad 0: Incompleto

Inconcluso supone que las metas específicas no se satisfacen para la zona de proceso.

Nivel de capacidad 1: Realizado

Llevado a cabo supone que el trabajo primordial se desarrolla y que las metas específicas del área de proceso se satisfacen.

Nivel de capacidad 2: Gestionado

Gestionado supone que su proceso ha sido planificado y ejecutado, tal como lo define la organización y para esto involucra los siguientes aspectos:

- Incluye un equipo de personas calificadas

- Produce resultados de forma controlada
- Involucra a las partes interesadas
- Su proceso se supervisa, controla y revisa
- Incluye una evaluación de la descripción del proceso para comprobar su cumplimiento.

La conexión al proceso en el nivel de capacidad 2 garantiza que la organización conserva las prácticas existentes durante los momentos importantes.

Nivel de capacidad 3: Definido

Definido significa que la adecuación de los procesos a partir de sus procesos definidos se realiza en la organización mediante la adhesión a sus directrices de adaptación.

En el nivel de capacidad 3, las organizaciones también aportan experiencias a los activos de procesos de la organización que se obtienen durante la implementación de los procesos.

Quizá nos preguntemos cómo se distingue entre los niveles de capacidad 2 y 3.

Veamos los factores de distinción:

En el nivel de capacidad 2, el proceso completo o algunas de sus descripciones difieren de un proyecto a otro. En otras palabras, los procesos no son comunes en toda la organización.

En el nivel de capacidad 3, los procesos estándar son comunes en toda la organización. Los procesos pueden adaptarse a un proyecto o a una unidad organizativa. La adecuación se realiza siguiendo las directrices de adaptación definidas para la organización.

En el nivel 3, los procesos se redactan de forma rigurosa y se gestionan de forma proactiva. Los procesos se consideran más coherentes y las únicas diferencias notables las permiten las directrices de adecuación.

Una vez que una organización alcanza el nivel de capacidad 3 en las áreas de proceso que se consideraron para la mejora, puede avanzar más en su camino de mejora de procesos

centrándose en la implementación de las áreas de proceso de alta madurez (es decir, el rendimiento de los procesos organizativos, la gestión cuantitativa de proyectos, el análisis causal y la resolución, y la gestión del rendimiento organizativo).

Definido supone que la adecuación de los procesos desde sus procesos definidos se hace en la organización por medio de la unión a sus directrices de adecuación.

En el grado de capacidad 3, las empresas además aportan vivencias a los activos de procesos de la organización que se obtienen a lo largo de la utilización de los procesos.

- En otras palabras, los procesos no son usuales en toda la organización.
- En el grado de capacidad 3, los procesos estándar son usuales en toda la organización.
- Los procesos se redactan de manera más coherente y las únicas diferencias notables las permiten las directrices de adecuación.

Cuando una organización alcanza el grado de capacidad 3 en las superficies de proceso que se consideraron para la optimización, puede seguir más en su camino de optimización de procesos centrándose en la utilización de las superficies de proceso de alta madurez (es mencionar, el rendimiento de los procesos organizativos, la administración cuantitativa de proyectos, la investigación causal y la resolución, y la administración del rendimiento organizativo) (Chaudhary y Chopra, 2017).

Niveles de madurez

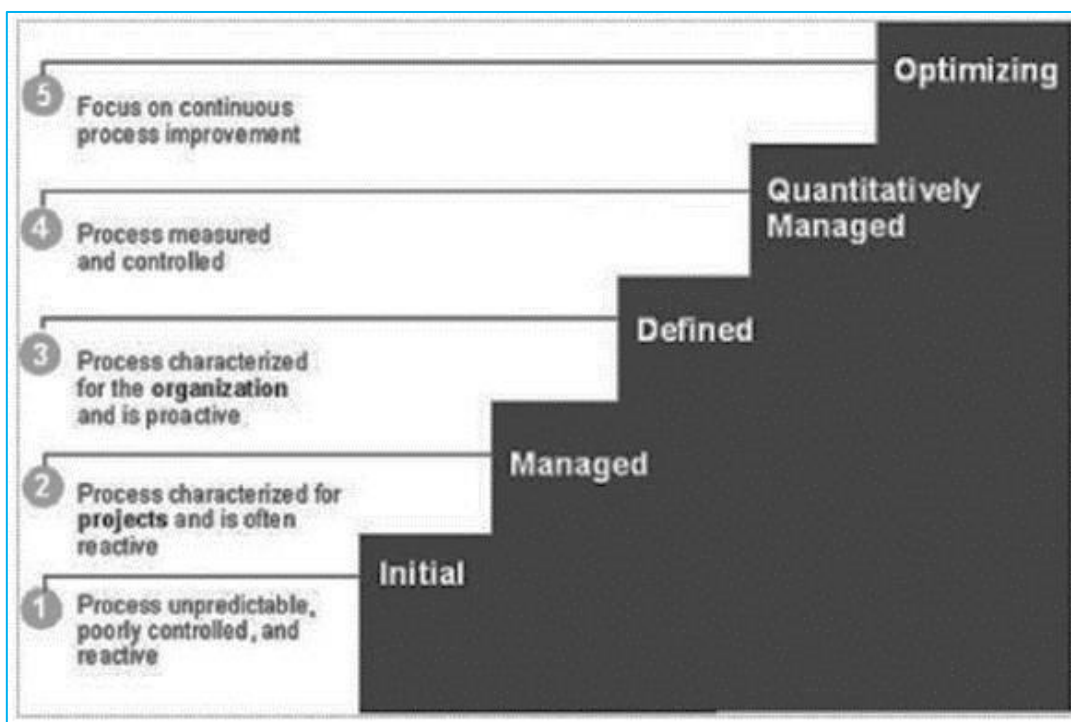
Un nivel de madurez es un camino que garantiza que las organizaciones sean capaces de mejorar sus conjuntos sucesivos de áreas de proceso de forma incremental. Cada nivel de madurez tiene un conjunto de procesos que, si se aplican juntos, le ayudarán a alcanzar un nivel de madurez completo (por ejemplo, pasar del nivel 1 al 2).

Un nivel de madurez es un camino que asegura que las empresas sean capaces de mejorar sus conjuntos sucesivos de áreas de proceso de manera incremental (Chaudhary y Chopra, 2017).

Los cinco niveles de madurez progresan del nivel 1 al 5 en etapas incrementales.

- Nivel 1: Inicial
- Nivel 2: Gestionado
- Nivel 3: Definido
- Nivel 4: Gestionado cuantitativamente
- Nivel 5: Optimización

Figura 4 Representaciones por etapas de CMMI



Nota. Tomado de Chaudhary y Chopra, 2017, p. 11.

2.3.4 Interpretación de cada nivel de madurez.

Nivel de madurez 1: Inicial

Inicial significa que los procesos son preciso. El éxito depende de la capacidad de algunas personas con habilidades formación y motivación de la organización que son protagonistas, involucrados y se fundamenta en los procesos que se pueden utilizar; en un futuro, cuando estos protagonistas dejan la organización, el equipo no podrá repetir el éxito.

En una organización, los productos construidos y los servicios prestados funcionan adecuadamente de manera constante; pese a ello, el presupuesto y el calendario siempre se ven afectados y se desvían de los propósitos que han sido documentados.

Nivel de madurez 2: Gestionado

Gestionado significa que los proyectos se suman a los procesos descritos por la organización. El equipo, así como los recursos que conforman el proyecto poseen los niveles de pericia para producir resultados adecuados:

- Los proyectos y sus productos de trabajo son supervisados y controlados
- Los proyectos y sus productos de trabajo se evalúan para que se adecuen a las descripciones de los procesos
- La dirección conoce el estado de la calendarización de los productos.
- La responsabilidad de las partes del proyecto es visible.

Nivel de madurez 3: Definido

Definido significa que los procesos están bien delimitados, documentados, comprendidos y estos son monitoreados. Los procesos que se siguen según la calendarización de actividades se van mejorando en el nivel de gestión:

El conjunto de procesos estándar de la organización, que es la base del nivel de madurez 3, se establece y se mejora a lo largo del tiempo. Estos procesos estándar se utilizan para establecer la integridad en toda la organización.

Una diferencia crítica entre los niveles de madurez 2 y 3 es el alcance de los estándares, descripciones de proceso y métodos. En el grado de madurez 3, los estándares, descripciones de proceso y métodos para un plan se adaptan desde el grupo de procesos estándar de la organización para adaptarse a un plan especial o unidad organizativa y, por

consiguiente, son más consistentes, exceptuando las diferencias permitidas por las guías de habituación.

Un proceso determinado instituye evidentemente el objetivo, entradas, criterios de ingreso, ocupaciones, papeles, medidas, fases de verificación, salidas y criterios de salida. En el grado de madurez 3, los procesos se gestionan más proactivamente por medio de la comprensión de las colaboraciones de las ocupaciones del proceso, de las medidas detalladas del proceso, de sus productos de trabajo y de sus servicios.

En el grado de madurez 3 la organización busca la optimización, todavía más, sus procesos involucrados con las zonas de proceso del grado de madurez 2 (Chaudhary y Chopra, 2017).

Nivel de madurez 4: Gestionado cuantitativamente

Gestionado cuantitativamente supone que una organización instituye fines cuantitativos para la calidad y el rendimiento de los procesos tanto a grado de organización como de plan.

Las metas cuantitativas se definen evaluando las necesidades del comprador, de los usuarios finales y de los ejecutores de la organización y de los procesos (la tabla 1 muestra una ejemplificación de esto en el cual el tamaño de la desviación del calendario es de -10% a 10%, y el tamaño de la desviación del esfuerzo es de -15% a 15%)

Las metas de calidad y de rendimiento de los procesos se visualizan por medio de medios estadísticos y se gestionan durante la vida de los proyectos

Las líneas de base y los modelos de rendimiento de los procesos se implementan para contribuir a entablar las metas de calidad y rendimiento de los procesos; esto ayudará a la organización a conseguir sus fines empresariales.

En el nivel 4, tenemos la posibilidad de profetizar el rendimiento del proceso.

Tabla 1*Un ejemplo de objetivo cuantitativo*

Objetivos empresariales	Medida	Valor mínimo	Valor máximo	Unidad de medida	Definición del indicador
Para entregar a tiempo	Desviación del programa	-10%	10%	Porcentaje	Grado de variación del número de días de entrega con respecto a la fecha de entrega prevista (con respecto al plan y al calendario revisado)
Para entregar a tiempo	Diferencia de esfuerzo	-15%	15%	Porcentaje	Grado en que el esfuerzo real (medido en horas-persona) se desvía del esfuerzo planificado para el ciclo de lanzamiento/prueba prevista (frente al esfuerzo planificado y revisado)

Nota. Tomado de Chaudhary y Chopra, 2017, p. 14.

Nivel de madurez 5: Optimización

Optimizar supone que una organización se concentra en mejorar sus procesos de manera continua por medio de la comprensión de sus fines empresariales y sus necesidades de rendimiento (en términos cuantitativos).

Las metas de calidad y rendimiento de los procesos se definen a grado organizativo.

En el grado de madurez 4, los conjuntos de la organización y del plan entienden y controlan el rendimiento a grado de subproceso, y usan dichos resultados para gestionar los proyectos.

En el grado de madurez 5, la organización se reúne en la administración y la optimización del rendimiento de la organización; esto se asegura por medio de la colección y el estudio de los datos de diversos proyectos.

Tabla 2 Comparación de los niveles de capacidad y de madurez

Nivel	Representación continua Niveles de capacidad	Representación por etapas Niveles de madurez
Nivel 0	Incompleto	
Nivel 1	Realizado	Inicial
Nivel 2	Gestionado	Gestionado
Nivel 3	Definido	Definido
Nivel 4		Gestionado cuantitativamente
Nivel 5		En optimización

Nota. Los nombres empleados para los niveles de capacidad y madurez en los niveles 2 y 3 son análogos (es decir, administrado y definido). Tomado de Chaudhary y Chopra, 2017, p. 15.

2.3.5 Gestión de la calidad

La utilización de técnicas de administración de la calidad, junto con las novedosas tecnologías de programa y los procedimientos de prueba, permitió mejorar de manera considerable el grado de calidad del programa en los últimos 20 años.

En el área de organización, la administración de la calidad se encarga de implantar un marco de procesos y reglas organizativas que conduzcan a un programa de alta calidad. Los accesorios de administración de la calidad deberían aceptar la responsabilidad de conceptualizar los procesos de desarrollo de programas que se usarán y las reglas que se aplicarán al programa y a la documentación relacionada, integrados los requisitos del sistema, el diseño y el código (Sommerville, 2016).

En el área de proyecto, la administración de la calidad involucra la aplicación de procesos de calidad específicos, la comprobación de que se han seguido dichos procesos planificados y la garantía de que los resultados del plan cumplen las reglas definidas para el mismo. La administración de la calidad del plan además puede involucrar la definición de una estrategia de calidad para un plan. El proyecto de calidad debería entablar las metas de calidad del plan y conceptualizar qué procesos y reglas se van a usar (Sommerville, 2016).

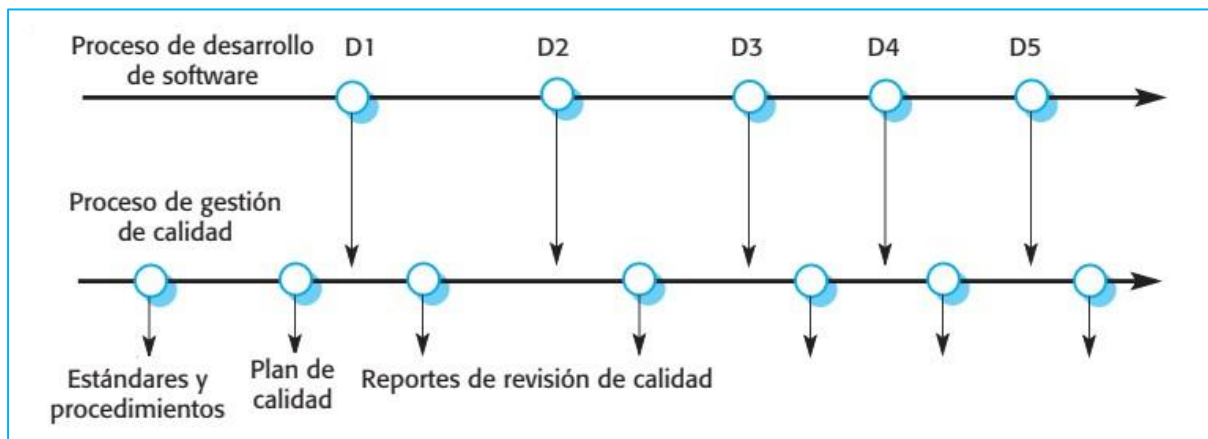
Las técnicas de administración de la calidad de los programas poseen sus raíces en los procedimientos y técnicas que se desarrollaron en las industrias manufactureras, donde el término garantía de calidad y control de calidad se usan extensamente. La garantía de calidad es la definición de los procesos y reglas que tienen que conducir a productos de alta calidad y la introducción de procesos de calidad en el proceso de construcción. El control de calidad es la aplicación de dichos procesos de calidad para remover los productos que no poseen el grado de calidad solicitado. Tanto la garantía como el control de calidad son parte de la administración de la calidad (Sommerville, 2016).

En la industria del software, varias organizaciones ven la garantía de calidad como la definición de métodos, procesos y reglas para asegurar la calidad del programa.

La administración de la calidad da un control sin dependencia del proceso de desarrollo de software (Sommerville, 2016).

Figura 5

Evaluación de la calidad y desarrollo de software



Nota. Tomado de Sommerville, 2016, p. 702.

Los accesorios de gestión de la calidad deberían ser libre y no conformar parte del conjunto de desarrollo de software, de manera que logre tener una perspectiva objetiva de la calidad del software. Tienen la posibilidad de informar sobre la calidad del software sin verse influidos por los inconvenientes de desarrollo del software. Lo ideal es que los accesorios de gestión de la calidad sean responsables de la administración de la calidad en toda la organización.

Un equipo sin dependencia de gestión de la calidad asegura que las metas de la organización de la calidad no se vean influidos por consideraciones de presupuesto y calendario a corto plazo. La administración de la calidad y el desarrollo de programa permanecen inevitablemente entrelazados con personas que poseen responsabilidades tanto de desarrollo como de calidad.

Es el proceso de desarrollo de una estrategia de calidad para un proyecto. El proyecto de calidad debería entablar las cualidades deseadas del software y explicar cómo se van a evaluar estas cualidades.

Humphrey (1989), sugiere un modelo para un plan de calidad:

- **Introducción del producto.** Detalle del producto, la demanda del mercado y las expectativas de calidad para el producto.
- **Planes del producto.** Planea la calendarización del producto, evaluación de la ruta crítica, condiciones reales para atender el servicio del producto.
- **Descripciones de procesos.** Detallan los procesos y estándares de desarrollo y servicio que deben emplearse para diseño y gestión del producto.
- **Metas de calidad.** El objetivo primordial de la calidad del producto, incluyen los atributos primordiales de calidad del producto.
- **Riesgos y gestión del riesgo.** Proceso de identificación y evaluación de riesgos, y a la creación de un plan para disminuirlos o controlarlos

2.3.6 Calidad del software

Las empresas de desarrollo de software establecieron las bases de la administración de la calidad en un esfuerzo por mejorar la calidad de los productos que se fabricaban. Como parte de este esfuerzo, las empresas desarrollaron una definición de calidad basado en la conformidad con una descripción descriptiva del producto. La conjetura subyacente era que los productos podían especificarse del todo y que podían establecerse métodos que permitieran verificar que un producto fabricado se ajustaba a su descripción (Sommerville, 2016).

En la evaluación de la calidad de software la iniciativa de empatía no es aplicable a los sistemas digitales y es erróneo pensar llegar a una culminación satisfactoria del sistema y determinar si cumple o no su descripción, por los siguientes detalles:

- En cuanto a las especificaciones de los requerimientos, es complicado redactar especificaciones de software enteras y sin ambigüedades. Los desarrolladores y consumidores de software tienen la posibilidad de interpretar los requerimientos de diferentes maneras y quizás sea imposible llegar a convenios sobre si el programa se desarrolló acorde a su explicación.
- Generalmente, las especificaciones incorporan requerimientos de algunas clases de competidores. Estos requerimientos son un compromiso inevitable y quizás no incluyan los requerimientos de todos los conjuntos de competidores. Por consiguiente, las partes involucradas quizá perciban que el sistema sea de mala calidad, pese a que se implementa los requerimientos acordados.
- Es erróneo pensar que medir de forma directa ciertas propiedades de calidad (por ejemplo, mantenibilidad) y, por ende, no tienen la posibilidad de especificarse adecuadamente y sin ambigüedades (Sommerville, 2016).

Gracias a dichos inconvenientes, el reconocimiento de calidad del software es un proceso personal en que los accesorios de la administración de calidad tienen que utilizar su juicio para dictaminar si se hizo un grado aceptable de calidad. Los accesorios de administración de calidad deberían tener en cuenta si el software se adapta o no a su objetivo pretendido. Esto involucra contestar cuestiones sobre las propiedades del sistema. Ejemplo:

1. ¿Se desarrollo siguiendo los patrones de programación y acorde a los documentos obtenidos?
2. ¿El software se inspecciono de manera apropiada?
3. ¿El software es confiable para utilizarse adecuadamente en las tareas del día a día?
4. ¿La funcionalidad del software es aceptable para uso normal?
5. ¿El software es útil?

6. ¿Se han seguido las normas de programación y documentación en el proceso de desarrollo?

Figura 6 *Componentes de calidad del software*

Protección	Comprensibilidad	Portabilidad
Seguridad	Probabilidad	Usabilidad
Fiabilidad	Adaptabilidad	Reutilización
Resistencia	Modularidad	Eficiencia
Robustez	Complejidad	Capacidad de aprendizaje

Nota. Tomado de Sommerville, 2016, p. 704.

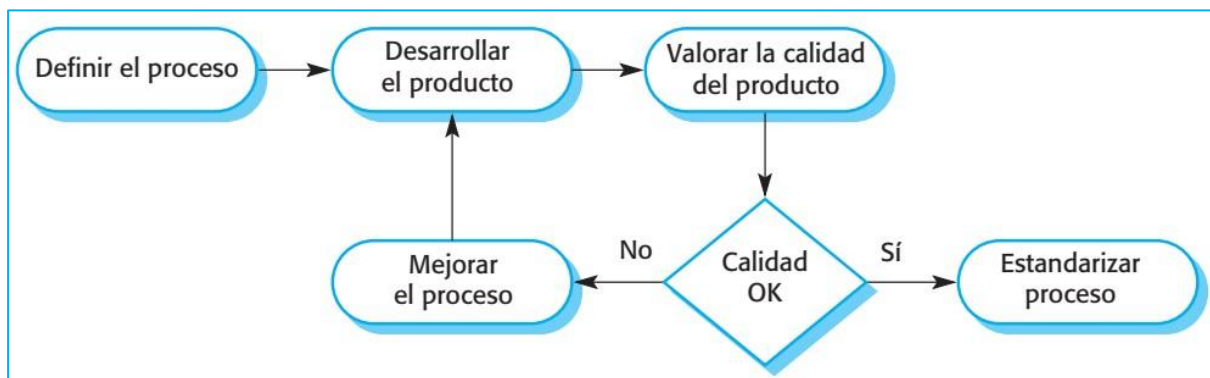
Para la administración de calidad del software es importante ver si el sistema se pondrá a prueba en oposición a sus requerimientos. La elección sobre si dar o no la funcionalidad solicitada debería fundamentarse en los resultados de dichas pruebas. Por consiguiente, los accesorios QA debería comprobar las pruebas que se desarrollaron e inspeccionar los registros de pruebas para comprobar que éstas se hayan llevado a cabo de forma adecuada. Para muchas empresas los accesorios de administración de calidad son responsable de las pruebas del sistema, empero, a veces, un conjunto de pruebas de sistema separado es responsable de esto (Sommerville, 2016).

La calidad subjetiva de un sistema de software se inspira primordialmente en sus propiedades no funcionales. Esto refleja la vivencia práctica del cliente: Si la funcionalidad del software no es lo que se esperaba, entonces los usuarios a menudo solamente le darán la vuelta a este tema y encontrarán otras maneras de hacerlo. No obstante, si el software no es fiable o resulta lento, entonces es básicamente imposible que los usuarios cumplan sus metas (Sommerville, 2016).

La gestión clásica de la calidad del software se fundamenta en la suposición de que la calidad del software está de manera directamente relacionada con la calidad del proceso de desarrollo del mismo. Esta suposición procede de los sistemas de construcción en los cuales

la calidad del producto está íntimamente relacionada con el proceso de producción. Un proceso de construcción involucra la configuración, la puesta en marcha y el manejo de las máquinas que participan en el proceso. Cuando las máquinas funcionan de manera correcta, la calidad del producto es natural. Se mide la calidad del producto y se modifica el proceso hasta conseguir el grado de calidad primordial (Sommerville, 2016).

Figura 7
Evaluación basada en el proceso



Nota. Tomado de Sommerville, 2016, p. 705.

2.3.7 Estándares de software

Las reglas del software desempeñan un papel fundamental en la administración de la calidad del software basada en el proyecto. Como ya he dicho, una gran parte de la garantía de calidad es la definición o selección de las reglas que se tienen que implementar en la construcción de software. Esta implementación tiene además la posibilidad de escoger los instrumentos y los procedimientos para beneficiar la utilización de estas reglas. Cuando se han seleccionado las reglas para su uso, se debe conceptualizar los procesos específicos del plan para supervisar la utilización de las reglas y revisar que se han seguido (Sommerville, 2016).

Las reglas de programa son relevantes por 3 causas:

1. Las reglas recogen conocimientos que son importantes para la organización. Se fundamentan en el razonamiento de la práctica mejor o más correcta para la compañía. Comúnmente, dichos conocimientos se adquieren luego de una enorme proporción de pruebas y errores. Incorporarlo a una regla ayuda a la organización a reutilizar esta vivencia y eludir errores anteriores (Sommerville, 2016).
2. Las reglas otorgan un marco para conceptualizar lo cual significa la calidad en un ámbito concreto. Como ya he dicho, la calidad del software es subjetiva y, al usar reglas, está establecido una base para dictaminar si se alcanzó el grado de calidad solicitado. Obviamente, esto es dependiente del establecimiento de reglas que reflejen las expectativas de los usuarios referente a confiabilidad, usabilidad y rendimiento del programa (Sommerville, 2016).
3. Las reglas ayudan a la continuidad una vez que el trabajo llevado a cabo por una persona es retomado y continuado por otra. Las reglas respaldan que todos los ingenieros de una organización adopten las mismas prácticas. Por consiguiente, se disminuye el esfuerzo de aprendizaje primordial al comenzar un nuevo trabajo (Sommerville, 2016).

Tabla 3 *Estándares de productos y procesos*

Estándares de producto	Estándares de proceso
Formulario de revisión de diseño	Conducta de revisión de diseño
Estructura del documento de requisitos	Presentación de un nuevo código para la construcción de sistemas
Formato de encabezado de método	Proceso de lanzamiento de la versión
Estilo de programación Java	Proceso de aprobación del plan de proyecto
Formato del plan de proyecto	Proceso de control de cambios
Formulario de solicitud de cambio	Proceso de grabación de prueba

Nota. Tomado de Sommerville, 2016, p. 707.

Se tienen la posibilidad de conceptualizar y usar dos tipos involucrados de estándares de ingeniería de software en la gestión de la calidad del software:

1. **Estándares del producto.** Se emplea al resultado de software que se está desarrollando. Integran estándares de documentos, como la composición de los documentos de requisitos, estándares de documentación, como un encabezado de comentario estándar para una definición de clase de objeto, y estándares de codificación, que definen cómo se debería utilizar un lenguaje de programación (Sommerville, 2016).
2. **Estándares de proceso.** Definen las metodologías que tienen que seguirse a lo largo del desarrollo de software. Tienen que resumir las buenas prácticas de desarrollo. Los estándares de proceso tienen la posibilidad de integrar definiciones de procesos de explicación, diseño y validación, herramientas de soporte de procesos y una explicación de los documentos que tienen que redactarse a lo largo de dichos procesos (Sommerville, 2016).

2.3.8 Estándares de ISO 9001

El conjunto de estándares internacionales utilizados para construir sistemas de gestión de la calidad en todas las empresas se conoce como ISO 9000. El estándar ISO 9000 se puede aplicar a varias organizaciones, desde la fabricación hasta los servicios. ISO 9001, la más común de estos estándares, se aplica a las organizaciones que diseñan, desarrollan y mantienen productos, incluido el software. La ISO 9001 se desarrolló originalmente en 1987 (Sommerville, 2016).

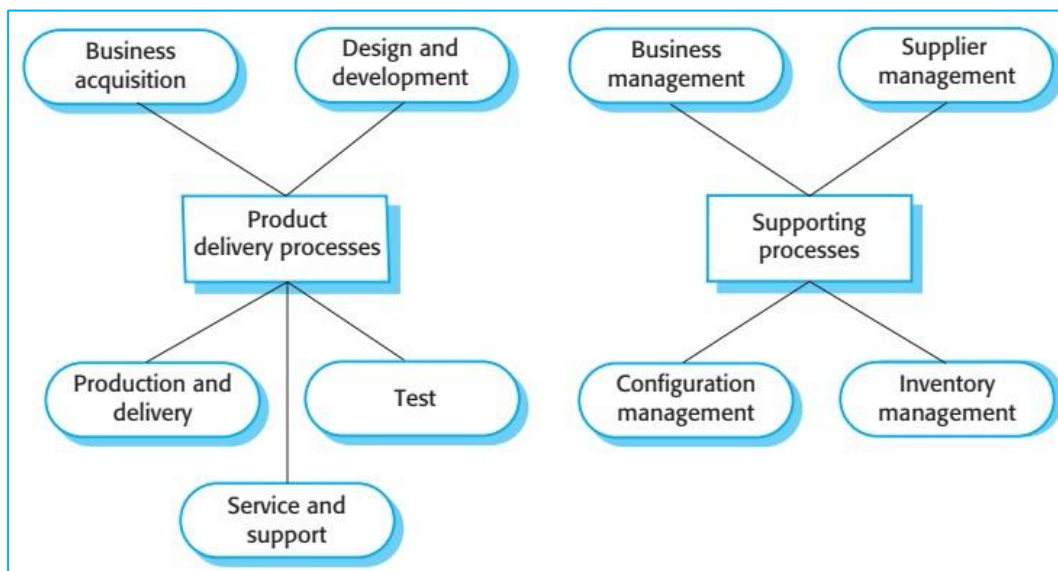
ISO 9001 no es un estándar de desarrollo de software, sino un marco para desarrollar estándares de software. Define los principios generales de la calidad, describe los procesos de la calidad en general y define los estándares y procedimientos que una organización debe

definir. Deben estar documentados en el manual de calidad de la organización (Sommerville, 2016).

En el año 2000, el estándar ISO 9001 sufrió una modificación sustancial que reenforzó el estándar en nueve actividades clave (Figura 8). Para cumplir con la norma ISO 9001, una empresa debe describir cómo sus operaciones se conectan con estos procesos básicos. También debe definir y mantener registros que demuestren que se han seguido los procesos organizativos prescritos. Los procedimientos requeridos, así como los datos de los procesos que deben recogerse y conservarse, deben describirse en el manual de calidad de la empresa (Sommerville, 2016).

Figura 8

Procesos centrales de ISO 9001



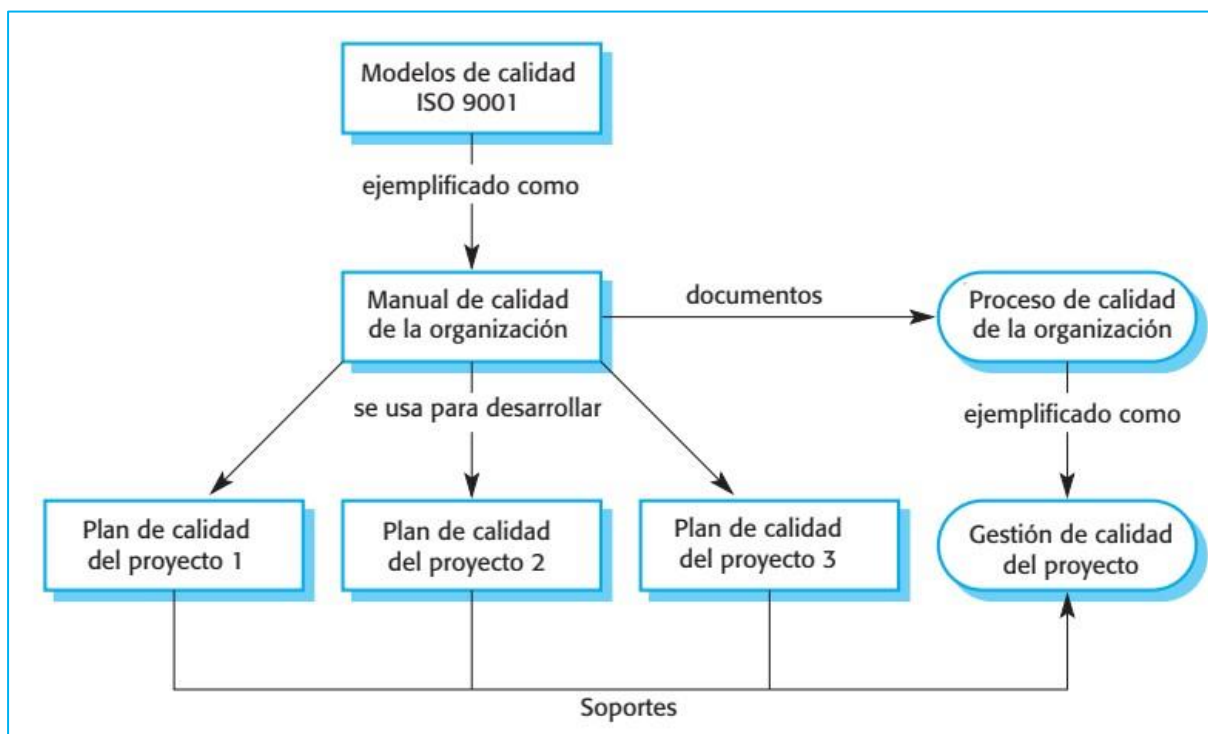
Nota. Tomado de Sommerville, 2016, p. 709.

El estándar ISO 9001 no especifica ni prescribe los métodos de calidad que debe utilizar una empresa. Una empresa debe establecer los tipos de procesos representados y tener procedimientos para demostrar que sus procesos de calidad se están siguiendo para cumplir con el estándar ISO 9001. Esto permite la flexibilidad en una variedad de industrias y tamaños de empresas (Sommerville, 2016).

Se pueden establecer criterios de calidad adecuados al tipo de software que se produce. Las pequeñas empresas pueden cumplir el estándar ISO 9001 incluso si sus procedimientos son básicos y hay poco papeleo. Sin embargo, debido a esta flexibilidad, no se pueden sacar conclusiones sobre la similitud o diferencia de los procesos en las distintas empresas que cumplen el estándar ISO 9001. Algunas empresas tienen procedimientos de control de calidad muy estrictos que conservan registros meticulosos, mientras que otras son mucho menos formales y requieren poco papeleo (Sommerville, 2016).

Las relaciones entre el estándar ISO 9001, los manuales de calidad de la organización y los planes de calidad de los proyectos individuales se muestran en la figura 9.

Figura 9
ISO 9001 y la administración de la calidad



Nota. Tomado de Sommerville, 2016, p. 710.

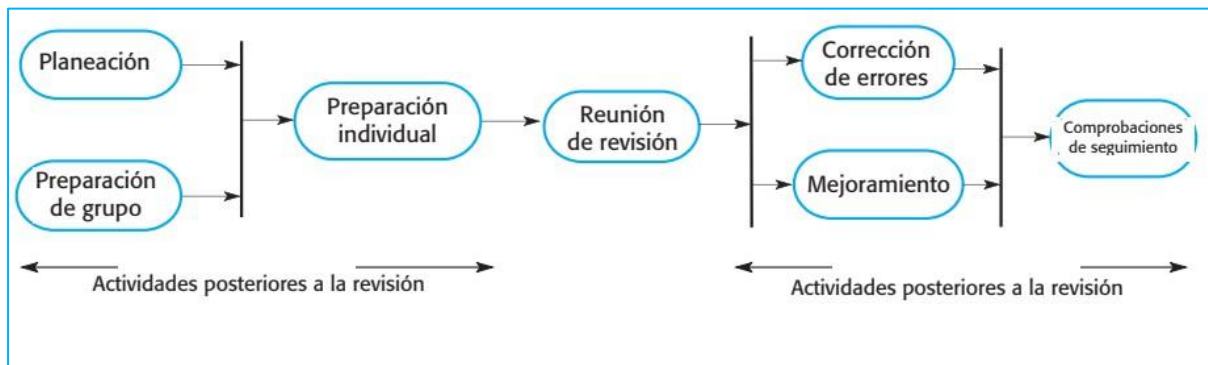
La certificación ISO 9001 es insuficiente, ya que define la calidad como el cumplimiento de las normas. No tiene en cuenta la calidad del software tal y como la

perciben los usuarios. Una empresa puede, por ejemplo, crear criterios de cobertura de pruebas que requieran que todos los métodos de los objetos sean llamados al menos una vez. Desgraciadamente, las pruebas de software insuficientes que excluyen las pruebas con varios parámetros de métodos podrían cumplir estos criterios. La empresa podría obtener la certificación ISO 9001 si se siguen los procesos de prueba requeridos y se mantienen los registros de las pruebas (Sommerville, 2016).

2.3.9 Revisiones e inspecciones

Las operaciones de garantía de calidad, como las revisiones e inspecciones, evalúan la calidad de los resultados del proyecto. Esto implica inspeccionar el programa, su documentación y los registros del proceso en busca de errores y omisiones, así como de infracciones de las normas. Parte del proceso general de verificación y validación del software, las revisiones e inspecciones se utilizan junto con las pruebas del programa (Sommerville, 2016).

Varias personas analizan el software y la documentación que lo acompaña durante una revisión, en busca de posibles problemas e incumplimientos de los estándares. El equipo de revisión toma decisiones fundamentadas sobre la calidad del software o de los documentos del proyecto. Los resultados de estas evaluaciones pueden ser utilizados por los directores de proyecto para tomar decisiones de planificación y asignar recursos al proceso de desarrollo (Sommerville, 2016).

Figura 10*El proceso de revisión de software*

Nota. Tomado de Sommerville, 2016, p. 711.

Inspecciones del programa

Las inspecciones de programas son evaluaciones por pares en las que los miembros de un equipo cooperan para descubrir problemas en un programa en desarrollo. Las inspecciones pueden formar parte de los procedimientos de verificación y validación del software. Son útiles como complemento de las pruebas, ya que no requieren la ejecución de la aplicación. Es posible verificar versiones incompletas del sistema, así como representaciones como los modelos UML. Se pueden examinar las pruebas del programa. Las revisiones de las pruebas suelen descubrir fallos en las mismas, lo que aumenta su capacidad para descubrir defectos en el software.

Las inspecciones de programas implican una evaluación línea por línea del código fuente del programa por parte de un equipo de personas con distintas experiencias. Buscan defectos y fallos y los comunican durante una reunión de inspección. Los errores lógicos, las anomalías en el código que puedan indicar un estado erróneo o las funcionalidades que faltan en el código son ejemplos de defectos. El equipo de revisión examina los modelos de diseño o el código del programa con gran detalle, señalando cualquier anomalía o fallo que deba ser resuelto.

Tabla 4 Lista de verificación de una inspección

Clase de falla	Comprobación de inspección
Fallas de datos	<input type="checkbox"/> ¿Todas las variables del programa se inician antes de usar sus valores? <input type="checkbox"/> ¿Todas las constantes tienen nombre? <input type="checkbox"/> ¿La cota superior de los arreglos es igual al tamaño del arreglo o Valor – 1? <input type="checkbox"/> Si se usan cadenas de caracteres, ¿se asigna explícitamente un delimitador? <input type="checkbox"/> ¿Existe alguna posibilidad de desbordamiento de buffer?
Fallas de control	<input type="checkbox"/> Para cada enunciado condicional, ¿la condición es correcta? <input type="checkbox"/> ¿Hay certeza de que termine cada ciclo?
	<input type="checkbox"/> ¿Los enunciados compuestos están correctamente colocados entre paréntesis? <input type="checkbox"/> En caso de enunciados, ¿se justifican todos los casos posibles? <input type="checkbox"/> Si después de cada caso en los enunciados se requiere un paréntesis, ¿éste se incluyó?
Fallas de entrada/salida	<input type="checkbox"/> ¿Se usan todas las variables de entrada? <input type="checkbox"/> ¿A todas las variables de salida se les asigna un valor antes de que se produzcan? <input type="checkbox"/> ¿Entradas inesperadas pueden causar corrupción?
Fallas de interfaz	<input type="checkbox"/> ¿Todas las llamadas a función y método tienen el número correcto de parámetros? <input type="checkbox"/> ¿Los tipos de parámetro formal y real coinciden? <input type="checkbox"/> ¿Los parámetros están en el orden correcto?

	<input type="checkbox"/> Si los componentes acceden a memoria compartida, ¿tienen el mismo modelo de estructura de memoria compartida?
Fallas de gestión de almacenamiento	<input type="checkbox"/> Si se modifica una estructura vinculada, ¿todos los vínculos se reasignan correctamente? <input type="checkbox"/> Si se usa almacenamiento dinámico, ¿el espacio se asignó correctamente? <input type="checkbox"/> ¿El espacio se cancela explícitamente después de que ya no se requiere?
Fallas de gestión de excepción	<input type="checkbox"/> ¿Se tomaron en cuenta todas las posibles condiciones de error?

Nota. Tomado de Sommerville, 2016, p. 714.

2.3.10 Medición de software

El objetivo de la medición del software es cuantificar algún aspecto de un sistema de software, como su complejidad o fiabilidad. Se pueden hacer inferencias sobre la calidad del software o analizar la eficiencia de los procesos, herramientas y procedimientos de software comparando los valores medidos entre sí y con los estándares que se aplican en toda la empresa. En un mundo ideal, la gestión de la calidad se basaría en medidas de los atributos de calidad del software. Así, se podría evaluar objetivamente las mejoras de los procesos y las herramientas destinadas a mejorar la calidad del software.

Suponga que trabaja en una empresa que está a punto de lanzar una nueva herramienta de prueba de software. Llevas un registro de la cantidad de problemas de software encontrados en un periodo de tiempo específico antes de implementar la herramienta. Esto sirve como punto de referencia para medir la eficacia de la herramienta. Se repite este paso después de utilizar la herramienta durante un tiempo. Si se descubren más fallos en el mismo

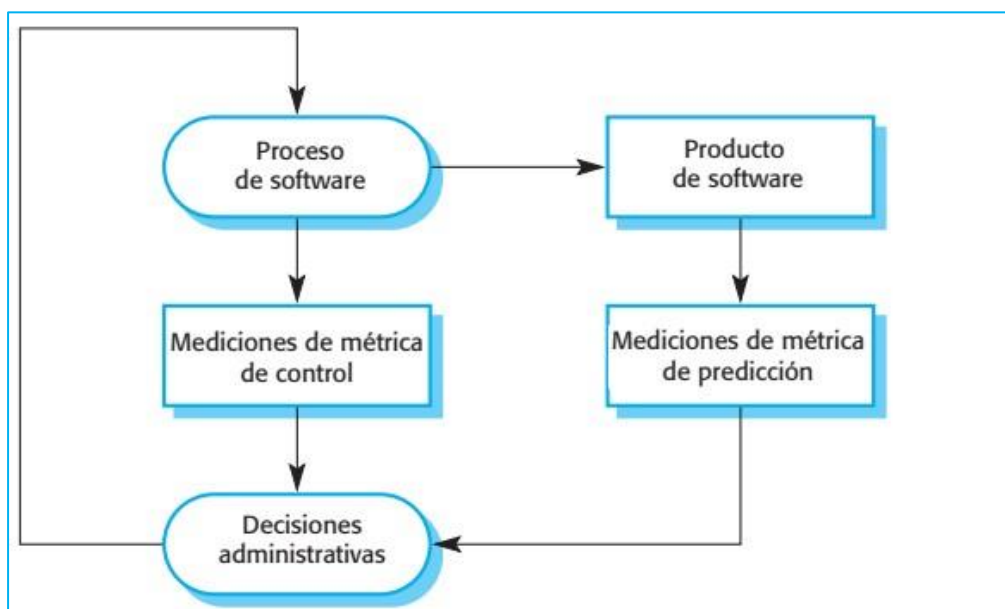
periodo de tiempo después de añadir la herramienta, se puede concluir que es útil en el proceso de validación del software.

El objetivo a largo plazo de la medición del software es utilizarlo para hacer juicios sobre su calidad. En un mundo ideal, las cualidades de un sistema podrían medirse utilizando una variedad de métricas. Se podría calcular un valor para la calidad del sistema basado en las mediciones realizadas. Si el programa cumple un determinado nivel de calidad, puede autorizarse sin necesidad de revisarlo. Las herramientas de medición también pueden revelar secciones del programa que deben ser mejoradas cuando sea apropiado.

Sin embargo, aún estamos muy lejos de esta condición ideal, y es poco probable que la evaluación automática de la calidad se produzca en breve.

Figura 11

Medidas de predicción y control



Nota. Tomado de Sommerville, 2016, p. 717.

Una métrica de software es una característica objetivamente medible de un sistema de software, de la documentación del sistema o del proceso de desarrollo. El índice Fog (Gunning-Fog), que es una medida de la legibilidad del material narrativo, el número de

errores notificados en un producto de software entregado y el número de días-persona necesarios para diseñar un componente del sistema son ejemplos de métricas.

Las métricas de control o de predicción son dos tipos de métricas de software. Las métricas de control ayudan a gestionar los procesos, mientras que las métricas de predicción ayudan a pronosticar las características del software, como su nombre indica. Los procesos de software suelen estar relacionados con las métricas de control. El esfuerzo medio y el tiempo necesario para solucionar los problemas notificados son ejemplos de métricas de control o de proceso. Hay tres tipos de métricas de proceso que pueden utilizarse:

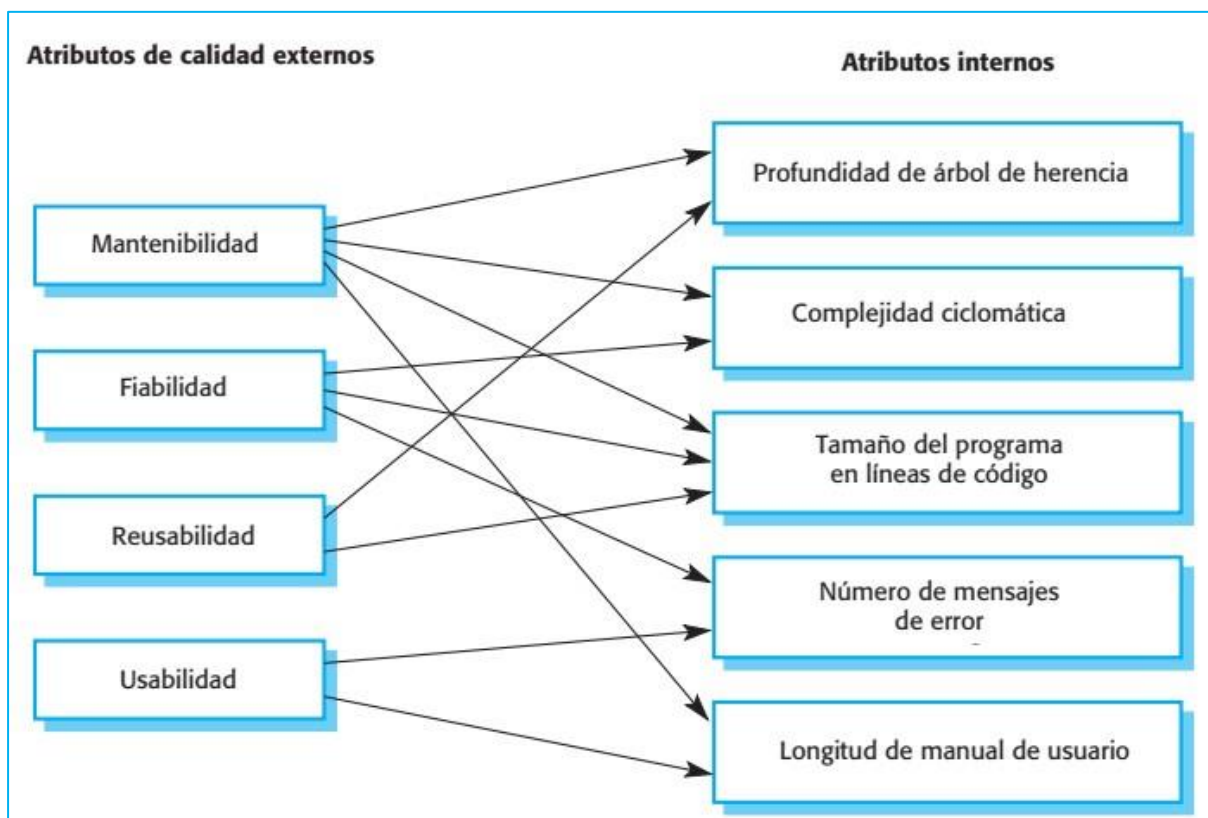
1. La cantidad de tiempo que tarda un procedimiento en finalizar. Puede tratarse de la cantidad global de tiempo empleado en el proceso, el tiempo de calendario, el tiempo empleado en el proceso por determinados ingenieros, etc.
2. La cantidad de recursos necesarios para un procedimiento específico. El esfuerzo total en días-persona, los gastos de viaje y los recursos informáticos son ejemplos de recursos.
3. El número de veces que se ha producido un evento. El número de defectos detectados durante la inspección del código, el número de modificaciones de requisitos solicitadas, el número de informes de errores en un sistema entregado y el número medio de líneas de código actualizadas en respuesta a un cambio de requisitos son ejemplos de eventos que podrían ser monitorizados.

La figura 12 muestra varios atributos de calidad de software externos, así como atributos de calidad de software internos que están intuitivamente vinculados a ellos. El gráfico muestra que las cualidades externas e internas pueden tener vínculos, pero no especifica cómo están conectados estos atributos. Deben cumplir tres requisitos para que la medida del atributo interno sea un predictor fiable de la característica externa del programa:

1. La propiedad interna debe medirse con precisión. Sin embargo, la medición no siempre es sencilla y puede requerir el uso de equipos específicamente especializados.

2. Debe existir un vínculo entre la propiedad medible y el atributo de calidad externo de interés. Es decir, el valor del atributo de calidad debe estar conectado con el valor de la característica evaluable de alguna manera.
3. El vínculo entre las cualidades internas y externas debe ser comprendido, confirmado y enunciado matemáticamente. La formulación del modelo implica el análisis de los datos adquiridos para determinar la forma funcional del modelo (lineal, exponencial, etc.), la determinación de los parámetros que deben incluirse en el modelo y la calibración de estos parámetros con los datos existentes (Sommerville, 2016).

Figura 12 Relaciones entre los atributos internos y externos del software



Nota. Tomado de Sommerville, 2016, p. 718.

Las métricas de producto son métricas de predicción que se utilizan para cuantificar las propiedades intrínsecas de un sistema de software. El tamaño del sistema, medido en líneas de código, o el número de métodos asociados a cada clase de objeto son ejemplos de

métricas de producto. Lamentablemente, como he dicho anteriormente en esta sección, las cualidades del software que se miden fácilmente, como el tamaño y la complejidad ciclométrica, no tienen un vínculo claro y coherente con criterios de calidad como la comprensibilidad y la mantenibilidad. Las correlaciones difieren según el método de desarrollo y la tecnología empleada, así como el tipo de sistema que se construye (Sommerville, 2016).

Hay dos tipos de métricas de producto:

1. Las métricas dinámicas, que se derivan de las mediciones realizadas durante la ejecución de un programa. Estos datos pueden recogerse durante las pruebas del sistema o después de su puesta en marcha. La cantidad de informes de errores o el tiempo que se tarda en terminar un cálculo son dos ejemplos.
2. Las métricas estáticas, que se recogen a través de mediciones de representaciones del sistema como el diseño, el software o la documentación (Sommerville, 2016).

2.3.11 Norma Técnica Peruana - NTP

La presente Norma Técnica Peruana fue desarrollada por el Comité Técnico de Normalización de Ingeniería de Software, Sistemas de Información y gestión de Proyectos - Sub Comité Técnico de Normalización de Ingeniería de Software y Sistemas de Información, utilizando el Sistema 1 o de Adopción, a partir de febrero a noviembre de 2015, con la norma ISO/IEC 12.

El Comité Técnico de Normalización de Ingeniería de Software, Sistemas de Información y gestión de Proyectos - Sub Comité Técnico de Normalización de Ingeniería de Software y Sistemas de Información presentó el PNTP-ISO/IEC 12207:2015 a la Dirección de Normalización -DN-, con fecha 2015-12-31, Norma Técnica Peruana NTP-ISO/IEC

12207:2016 Ingeniería de Software y Sistemas se estableció después de que no se presentaran observaciones. El 16 de julio de 2016, Procesos del ciclo de vida del software, 3a Edición.

Esta Norma Técnica Peruana sustituye a la NTP-ISO/IEC 12207:2006.

TECNOLOGÍA DE LA INFORMACIÓN. Ingeniería de sistemas y software - Procesos del ciclo de vida del software es una adaptación de la norma ISO/IEC 12207:2008 IEEE Std 12207-2008 Systems and software engineering - Software life cycle processes. La presente Norma Técnica Peruana refleja cambios editoriales, principalmente en cuanto a la terminología utilizada sólo en el idioma español, y ha sido estructurada de acuerdo con las Guías peruanas GP 001:1995 y GP 002:1995 (Dirección de Normalización - INACAL, 2016).

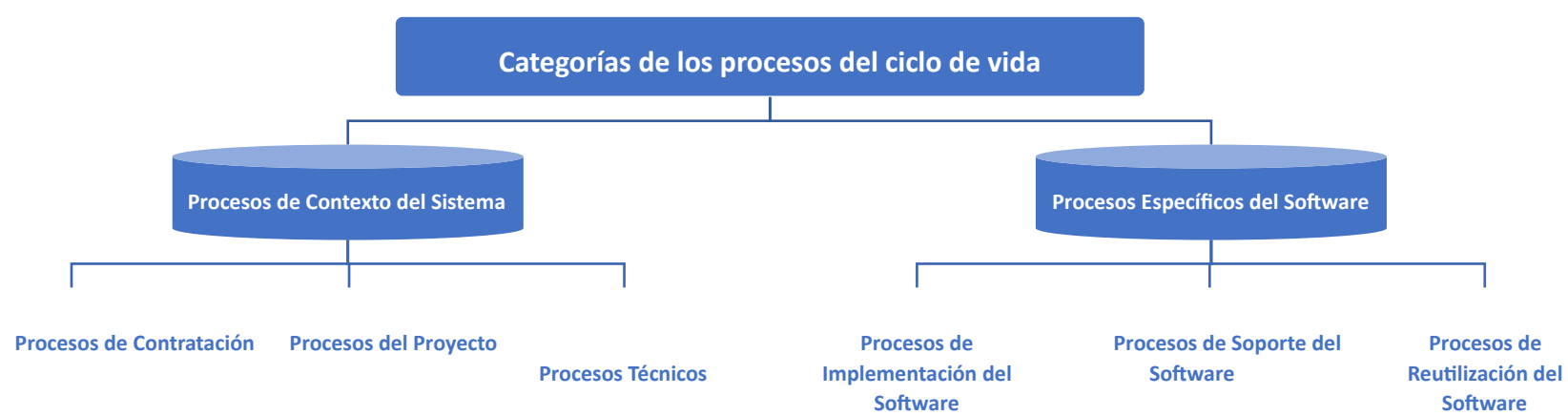
Relación entre sistemas y software

Esta norma establece un fuerte vínculo entre un sistema y su software. Se basa en los principios generales de la ingeniería de sistemas. El software se considera un elemento esencial del sistema global y desempeña determinadas funciones dentro de él. Se aplica extrayendo los requisitos del software de los requisitos del sistema y diseñando el software, para luego producirlo e integrarlo en el sistema. La existencia del software en el contexto de un sistema, aunque éste consista sólo en el procesador en el que se ejecuta el programa, es una premisa clave de esta Norma. En consecuencia, un producto o servicio de software siempre se considera un componente de un sistema.

Como ejemplo podemos mencionar que, la Norma distingue entre el análisis de requisitos del sistema y el análisis de requisitos del software porque, en general, el análisis de requisitos del sistema asigna los requisitos del sistema a varios elementos del sistema, mientras que el análisis de requisitos del software deriva los requisitos de los requisitos del software (Dirección de Normalización - INACAL, 2016).

Figura 13

Grupos de procesos del ciclo de vida





Nota. Tomado de Norma Técnica Peruana NTP-ISO/IEC 12207:2016 2016, p. 41.

2.3.12 ISO 9001 - Mejora

En la perspectiva de la Norma ISO 9001, la mejora es una noción que motivará a la empresa a satisfacer las necesidades del cliente y a mejorar los objetivos de calidad y la felicidad del cliente. En cierto modo, la mejora implica identificar y cambiar los parámetros que influyen en la consecución de los objetivos. El problema es determinar qué procedimientos tienen mayor impacto en la conformidad del producto con la satisfacción del cliente. La norma ISO 9001 obliga a identificar los procesos o resultados de los procesos (productos o servicios) que carecen de rendimiento y eficacia, así como a realizar ajustes controlados para mejorarlos. Como consecuencia, se mejorará la eficacia del sistema de gestión de la calidad (SGC) (Abuhav, 2017).

Generalidades

La Norma ISO 9001 exige que la empresa identifique y priorice las áreas de mejora, así como que tome las medidas necesarias para satisfacer las necesidades del cliente y aumentar su satisfacción. Esto incluye lo siguiente

- Mejora del producto y del servicio para cumplir con las expectativas actuales y futuras de las partes interesadas
- Acciones correctivas
- Medidas preventivas
- Acciones para minimizar las consecuencias indeseables
- Mejoras en el rendimiento del SGC
- Mejoras en la eficacia del SGC

Aumentar la calidad de los productos y servicios

Para satisfacer las expectativas de los clientes y mantener su nivel de satisfacción, diseñamos, ejecutamos y supervisamos las mejoras. Las actividades de mejora deben cumplir dos objetivos al planificar la identificación y ejecución de mejoras en la organización:

1. Cumplir con los requisitos de los clientes. Este objetivo está relacionado con la mejora de los productos y servicios para cumplir con las expectativas existentes de los clientes. Estas mejoras están relacionadas con las expectativas objetivas de los consumidores del producto o servicio, así como con las características que se les comunicaron en el momento de la venta o la entrega. Este objetivo también incluye los requisitos reglamentarios.
2. Mejorar la satisfacción del cliente: estas mejoras se refieren a la opinión subjetiva del cliente sobre el grado de cumplimiento de sus exigencias y expectativas:
 - ¿Qué expectativas de sus clientes no se han cumplido?
 - ¿Qué expectativas de los consumidores no se han reconocido todavía?

Efectos no deseados: corregir, prevenir y reducir

Un impacto negativo inducido por un problema de calidad se denomina consecuencia no deseada. La falta de calificación en una estación de trabajo, por ejemplo, puede resultar en un alto grado de desecho e incluso de disconformidad. Las consecuencias no deseadas tienen el potencial de influir en otros procesos o resultados de procesos más adelante en la cadena de suministro. Durante la evaluación de riesgos y oportunidades, se esperaba que estas consecuencias imprevistas fueran reconocidas y manejadas. El propósito de la mejora es eliminar los impedimentos, interferencias e interrupciones luego de reconocer las ocurrencias que pueden generar el resultado no deseado. La norma exige que se apliquen cambios para solucionarlos, evitarlos o mitigarlos. En caso de evitarlos la organización tendrá que dar los siguientes pasos para solucionar, evitar o mitigar el efecto desfavorable:

- Examinar las causas principales del rechazo relacionadas con el rendimiento de los empleados: ¿qué actividad no se está ejecutando correctamente y por qué?
- Sugerencias para abordar las causas principales: ¿qué tipo de formación puede ayudar a los empleados a realizar mejor su trabajo?

- Mida el rechazo antes y después de la modificación para hacer un seguimiento del progreso (Abuhav, 2017).

Mejorar el rendimiento del sistema de gestión de la calidad

El rendimiento del SGC se mide mediante un conjunto de indicadores de proceso y sus criterios asociados, que indican si los procesos cumplen (o no) sus objetivos. Muchas herramientas y procedimientos de calidad recomendados en la norma se utilizan para supervisar, medir y evaluar el rendimiento del SGC. Las operaciones del SGC pueden ser mejoradas, rediseñadas o rediseñadas como consecuencia de un estudio del rendimiento de los procesos.

La norma exige que la empresa mejore la capacidad de sus procedimientos para lograr los resultados previstos. Las actividades se iniciarán para lograr los objetivos de la empresa basándose en los resultados del seguimiento, la medición, el análisis y la evaluación. La mejora del rendimiento de los procesos se refiere a la capacidad de un proceso para proporcionar los resultados deseados y, en consecuencia, cumplir los objetivos de calidad. Para alcanzar los objetivos, hay que iniciar y promover determinadas actividades. Se puede mejorar una o varias de las siguientes variables del proceso:

- Entradas del proceso que se introducen en el sistema
- Actividades que mantienen el proceso en marcha
- Procesos y recursos
- Resultados del proceso que han sido aceptados

No conformidad y acción correctiva

La acción correctiva es una de las piedras angulares de la gestión de la calidad y es fundamental para el éxito a largo plazo del SGC. La noción básica de acción correctiva fomenta una investigación exhaustiva de los problemas de calidad existentes y la erradicación de

cualquier causa subyacente de las no conformidades mediante el uso de métodos controlados.

La empresa debe crear la capacidad de mostrar la adhesión a los criterios definidos del SGC.

En otras palabras, los clientes adquieren confianza en la capacidad de la empresa para suministrar constantemente productos y servicios que satisfagan sus necesidades y especificaciones. La norma ISO 9001 especifica los siguientes requisitos:

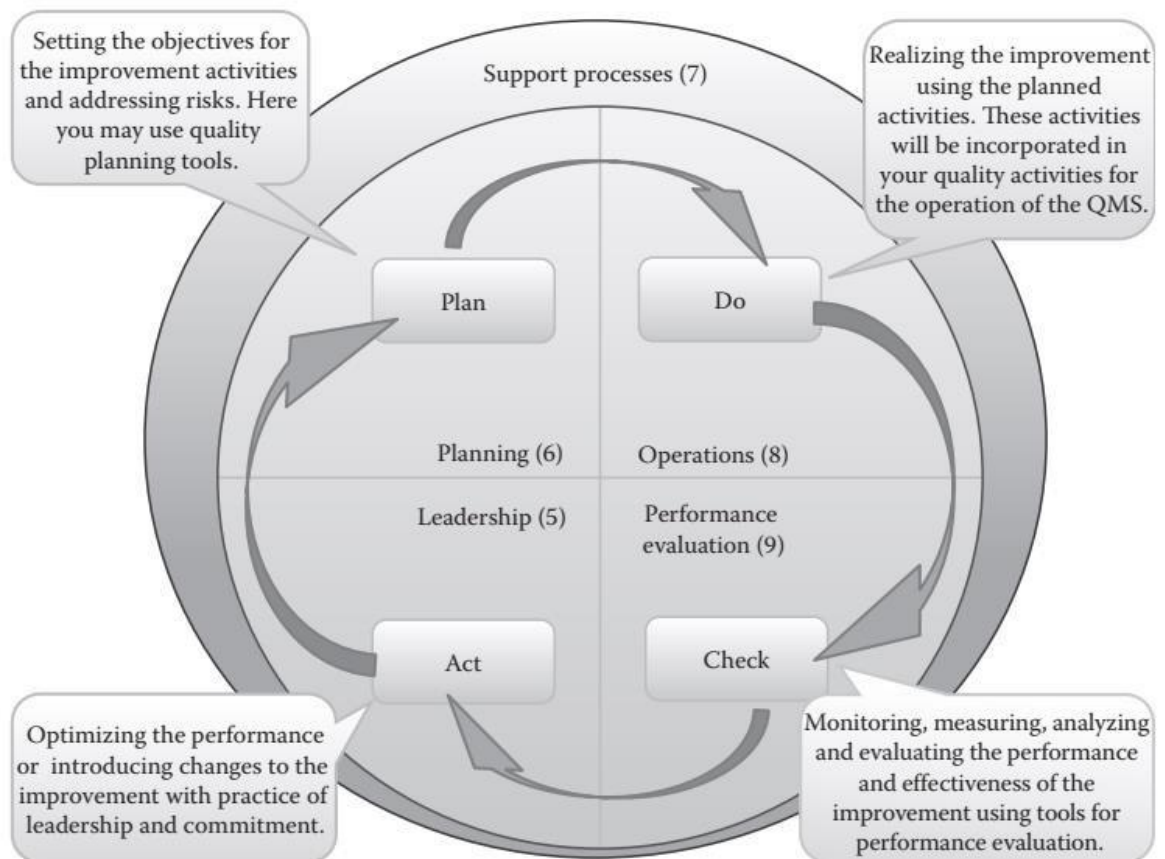
- Cuando surge una disconformidad, incluidas las no conformidades que se revelaron durante una reclamación, la organización debe responder.
- La empresa debe tomar medidas para regular y eliminar la no conformidad.
- La organización debe evaluar y responder a los efectos de la no conformidad.
- La organización debe evaluar la necesidad de actuar para erradicar el origen o las causas de la no conformidad para que no se repita o se produzca en otro lugar.
 - La organización examinará y analizará la no conformidad.
 - La organización debe determinar cuál es la causa de la no conformidad.
 - La empresa debe tomar medidas para abordar el problema fundamental.
 - La organización debe establecer si existen o pueden surgir no conformidades comparables.
- La organización es responsable de iniciar y llevar a cabo los pasos necesarios para abordar la no conformidad.
- La organización evaluará el éxito de las actividades realizadas para abordar la no conformidad.
- Durante el proceso de planificación, la empresa debe actualizar los riesgos y oportunidades asociados a la acción correctiva.
- La actividad correctiva debe ser proporcional al impacto de la no conformidad (Abuhav, 2017).

2.3.13 Desarrollo continuo

La Norma ISO 9001 pretende incorporar los avances de la organización en un proceso que permita controlarla y garantizar la realización de sus objetivos. La mejora continua es un tipo de aprendizaje en el que una organización se examina a sí misma de forma periódica, realiza juicios y toma decisiones basadas en los resultados y adopta medidas para mejorar. La norma promueve el método PDCA (Planificar-Hacer-Verificar-Actuar). La norma ISO 9001 especifica los siguientes requisitos:

- El SGC de la organización debe mejorar constantemente en cuanto a su aplicabilidad, adecuación y eficacia.
- La organización revisará y evaluará las siguientes categorías de resultados:
 - Análisis y resultados de la evaluación
 - El resultado de la revisión por la dirección
- El objetivo es ver si hay posibilidades de mejora que puedan utilizarse como insumos para la mejora continua (Abuhav, 2017).

Figura 14 Integración del ciclo PDCA en el sistema de gestión de la calidad.



Nota. Tomado de Abuhav, p. 418.

2.4 Definición de Términos Básicos

El sustentar teóricamente el estudio conlleva a identificar, distinguir, analizar y evaluar la pertinencia y relevancia en el uso de la tesis.

2.4.1 Marco conceptual de la calidad

Evolución del concepto de calidad

Como consecuencia, para comprender el significado actual del término resulta conveniente analizar el proceso histórico que lo ha desarrollado hasta alcanzar el actual enfoque integral o “sistémico”, distinguiendo cinco etapas claves: 1) edad media-revolución industrial, 2) revolución industrial-finales siglo XIX, 3) administración científica-II Guerra

Mundial, 4) II Guerra Mundial-década de los setenta y 5) década de los ochenta y noventa (Tari, 2000)

Evolución del control de calidad

La calidad de los bienes producidos y de los servicios prestados se controla, directa o indirectamente, desde tiempos inmemoriales. Sin embargo, la utilización de una base cuantitativa con principios estadísticos para controlar la calidad es un concepto moderno.

La calidad de la arquitectura griega del siglo V a.c. era tan envidiada que afectó profundamente a las posteriores construcciones arquitectónicas de Roma. Las ciudades, iglesias, puentes y carreteras construidas por los romanos nos inspiran aún hoy (Mitra, 2021).

Durante la Edad Media y hasta el siglo XIX, la producción de bienes y servicios se limitaba predominantemente a una sola persona o a un pequeño grupo. Esta fase, que comprende el periodo de tiempo hasta 1900, ha sido etiquetada por Feigenbaum como el periodo de control de calidad del operario. Todo el producto era fabricado por una sola persona o por un grupo muy reducido de personas. Por esta razón, la calidad del producto podía ser controlada esencialmente por una persona que también era el operario, y el volumen de producción era limitado (Mitra, 2021).

En Japón, la década de los setenta marcó la expansión del uso de una herramienta gráfica conocida como diagrama de causa-efecto. Ishikawa y a veces se denomina diagrama de Ishikawa. También se le llama diagrama de espina de pescado por su parecido con el esqueleto de un pez. Este diagrama ayuda a identificar las posibles razones por las que un proceso se sale de control, así como los posibles efectos en el proceso (Mitra, 2021).

Calidad

La noción de calidad ha sido definida de diferentes maneras por varios autores.

Además, identifica un marco de ocho atributos que pueden utilizarse para definir la calidad: rendimiento, características, fiabilidad, conformidad, durabilidad, capacidad de servicio, estética y calidad percibida. Una definición más general propuesta por Juran (1974) es la siguiente: “La calidad es la aptitud para el uso” (Mitra, 2021).

Defectos

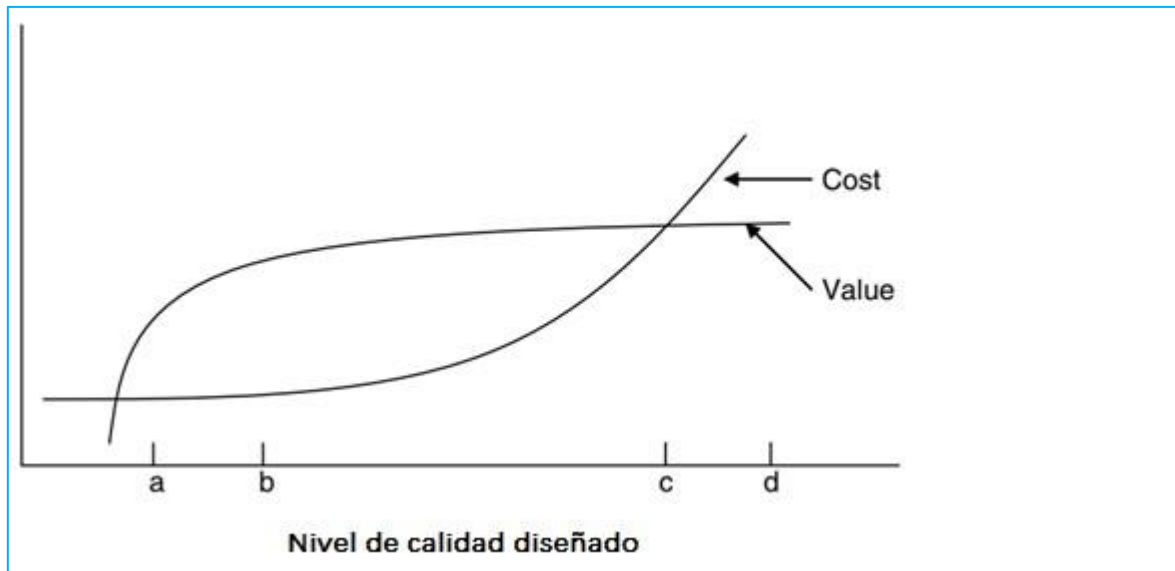
Un defecto está asociado con una característica de calidad que no cumple con ciertos estándares. Además, la gravedad de uno o más defectos en un producto o servicio puede hacer que sea inaceptable (o defectuoso). El término moderno para defecto es no conformidad y el término para defectuoso es artículo no conforme. El American National Standards Institute, la International Organization for Standardization y la American Society for Quality proporcionan una definición de defecto en la norma ANSI / ISO / ASQ A8402 (ASQ 1994) (Mitra, 2021).

2.4.2 Estándar o especificación

Mitra (2021) afirma “La definición de calidad implica cumplir con los requisitos del cliente, estos requisitos deben documentarse. Esto significa que para que un eje sea aceptable para el cliente, cada una de estas dimensiones debe estar dentro de los valores especificados” (p.9).

- **Especificación:** conjunto de condiciones y requisitos, de aplicación específica y limitada, que proporcionan una descripción detallada del procedimiento, proceso, material, producto o servicio para su uso principalmente en adquisiciones y fabricación.
- **Estándar:** conjunto prescrito de condiciones y requisitos, de aplicación general o amplia, establecidos por autoridad o acuerdo, que deben satisfacerse mediante un material, producto, proceso, procedimiento, convención, método de prueba; y/o sus características físicas, funcionales, de rendimiento o de conformidad. Una realización física de una unidad de medida.

Figura 15 Coste y valor en función de la calidad del diseño.



Nota. Tomado de Mitra 2021, 2017, p. 10.

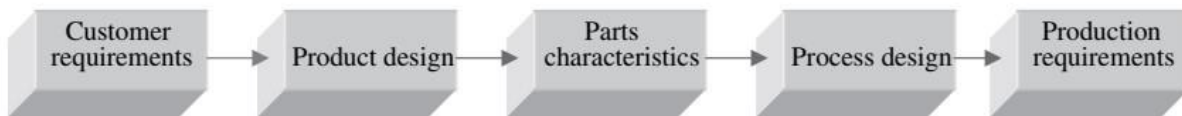
Despliegue de la función de calidad

Este ciclo de calidad comienza con la creación de un diseño que satisfaga las necesidades del cliente y continúa con la realización de análisis detallados de las piezas y componentes para conseguir el producto deseado, la identificación de los procesos necesarios para fabricar el producto, el desarrollo de los requisitos del producto, las pruebas de los prototipos, las pruebas del producto o servicio final y el final de la resolución de problemas posventa (Mitra, 2021).

Mitra (2021) afirma que el ejemplo de ejercicio QFD ilustra la importancia del trabajo en equipo en este proceso. El proceso QFD procede determinando los descriptores técnicos para estos nuevos “qué”. Por lo tanto, podemos considerar la implementación del proceso QFD en diferentes fases. Como muestra la Figura 3, QFD facilita la traducción de los requisitos del cliente en un producto cuyas características cumplen con estos requisitos.

El siguiente nivel puede abordar el diseño de un proceso para fabricar piezas con las características identificadas. Finalmente, el proceso QFD identifica los requisitos de producción para operar el proceso en condiciones específicas (Mitra, 2021).

Figura 16 *Fases de uso de QFD.*



Nota. Tomado de Mitra 2021, 2017, p. 10.

2.4.3 Norma ISO 9001

La cláusula 1 de la Norma ISO 9001 sirve para presentar los propósitos y conceptos de la norma y definir el ámbito de aplicación de la norma a su sistema de gestión de la calidad.

La Norma ISO 9001 es una norma internacional para el establecimiento, diseño e implementación de un sistema de gestión de la calidad en una organización.

La aplicación de los requisitos de la Norma ISO 9001 permite a una organización demostrar su capacidad para proporcionar de forma consistente productos o servicios que cumplan con los requisitos del cliente.

La aplicación de los requisitos de la Norma ISO 9001 permite a una organización demostrar su capacidad para proporcionar productos o servicios que cumplan con los requisitos legales o reglamentarios aplicables.

2.4.4 La NTP-ISO/IEC 12207

Su objetivo principal es proporcionar una estructura común que permita a los compradores, proveedores, desarrolladores, mantenedores, operadores, gerentes y técnicos involucrados en el desarrollo de software hablar un lenguaje común.

Procesos principales

El proceso continúa para preparar y emitir solicitudes de propuestas, seleccionar proveedores y gestionar el proceso de adquisición hasta que se acepta el sistema, el producto de software o el servicio de software. El proceso lo inicia el comprador del producto o servicio de software. Elaborar y actualizar contratos. Aceptación y Culminación.

El proceso continúa definiendo los procedimientos y recursos necesarios para administrar y entregar el proyecto, incluida la preparación y ejecución del plan del proyecto, hasta la entrega del sistema de software, producto o servicio al comprador. Puede incluir operaciones a nivel de sistema si se especifica en el contrato. Las acciones del desarrollador bajo el contrato para realizar o apoyar este proceso.

2.4.5 La norma ISO/IEC 9126 la calidad interna

La calidad interna es medida y evaluada en base a los requerimientos de calidad interna. Es la calidad cuando el software es ejecutado el que determina el éxito o fracaso, la cual es típicamente medida y evaluada mientras se prueba en un ambiente simulado con datos simulados y usando métricas externas. Como es difícil corregir la arquitectura de software u otros aspectos fundamentales del diseño del software, el diseño fundamental permanece sin cambios a través de las pruebas.

Además, la calidad puede analizarse en diferentes perspectivas:

Interna: medible a partir de las características intrínsecas, como el código fuente

Externa: medible en el comportamiento del producto, como en una prueba

En uso: durante la utilización efectiva por parte del usuario

2.4.6 Modelo de calidad establecido por el estándar ISO 9126

Características:

Funcionalidad

Esta característica permite calificar si un producto de software maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado.

Atributos: Adecuación. Exactitud. Interoperabilidad. Conformidad. Seguridad.

Confiabilidad

Se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido.

Sub-características: Nivel de Madurez. Tolerancia a fallas. Recuperación.

Usabilidad

Característica que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema.

Atributos: Comprensibilidad. Facilidad de Aprender. Operabilidad.

Eficiencia

Esta característica permite evaluar la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados.

Aspectos a evaluar: Comportamiento con respecto al Tiempo. Comportamiento con respecto a Recursos.

Mantenibilidad

Aquí permite medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad.

Factores: Capacidad de análisis. Capacidad de modificación. Estabilidad. Facilidad de Prueba.

Portabilidad

Se refiere a la habilidad del software de ser transferido de un ambiente a otro.

Aspectos: Adaptabilidad. Facilidad de Instalación. Conformidad. Capacidad de reemplazo.

2.5 Fundamentos teóricos que sustentan las hipótesis

El objetivo de este fundamento teórico que sustenta las hipótesis es sintetizar los principales conceptos que se han examinado durante este trabajo de tesis.

Se trata de analizar objetivamente la realidad a través de una serie de pasos que comienzan con la formulación de una idea (relacionada con los objetivos de la investigación y el producto final del proyecto) y terminan con la formulación de un problema en el que se identifica de forma clara, precisa y accesible el problema a investigar, que debe incluir variables dependientes e independientes, en este trabajo de tesis sólo se trabajará con una variable por ende no habrá variables dependientes.

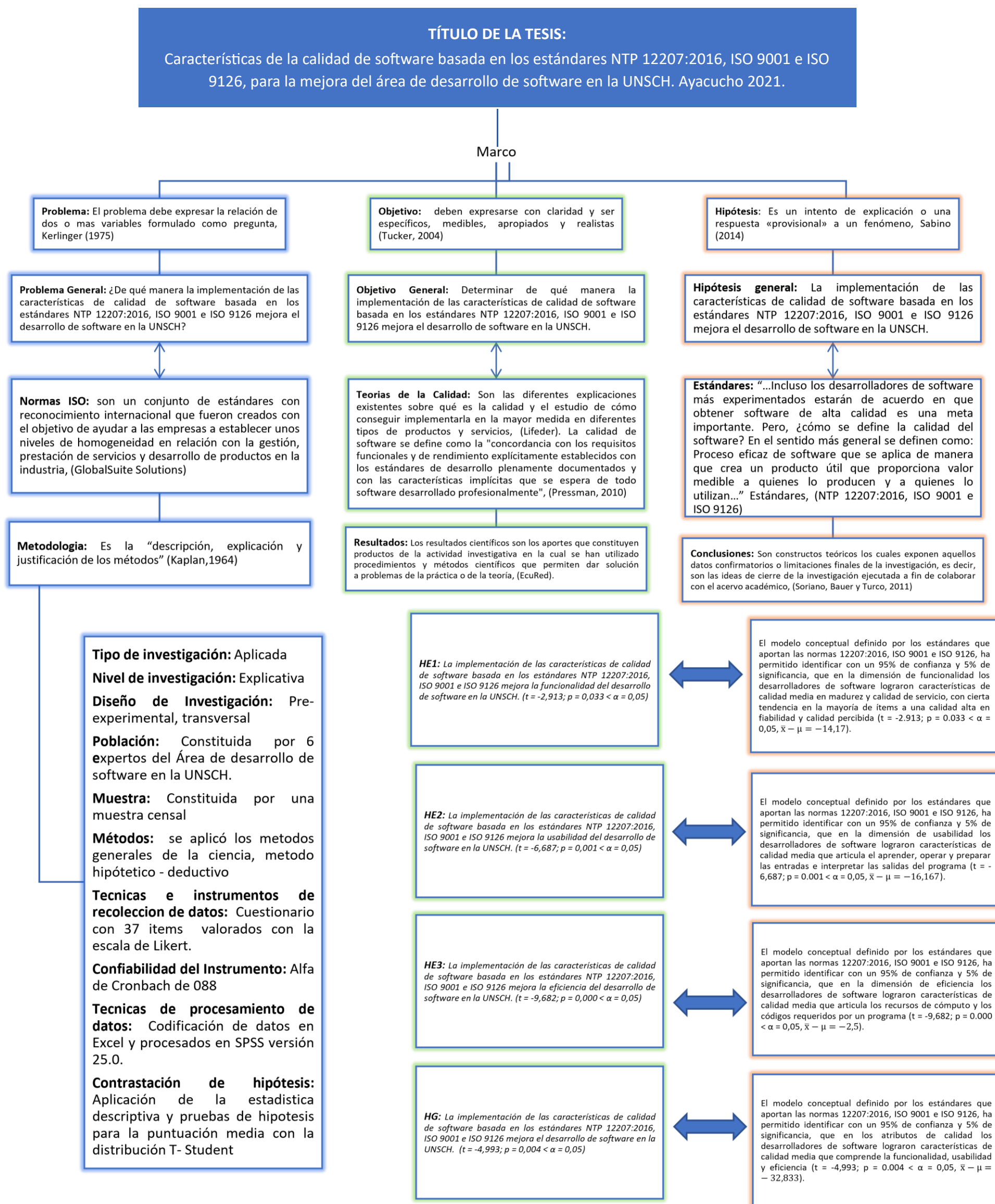
El siguiente paso es elaborar un marco teórico, lo que implica la lectura de libros, artículos científicos actualizados, revistas indexadas y otras fuentes que incluyan conceptos relacionados con el tema de investigación.

Finalizando esta parte del trabajo de tesis se aborda la hipótesis se puede definir como una predicción o explicación provisoria (mientras no sea contrastada) de la relación entre las variables (en esta investigación la variable independiente). Así pues, el problema-pregunta precede a la hipótesis-respuesta que, a su vez, deriva los objetivos de la investigación.

Mapa conceptual: Mediante el mapa conceptual se pretende mostrar los principales procesos desarrollados a lo largo de la presente investigación, detallando de manera objetiva las etapas de desarrollo que se plantean en este trabajo de tesis.

Figura 17

Contexto teórico, metodológico y aplicativo de la calidad de software basada en los estándares para el desarrollo de software



Nota. La figura muestra la relación entre las dimensiones de la variable Calidad de software, así como los instrumentos de medición de las variables.

2.6 Hipótesis

2.6.1 Hipótesis general

La implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora el desarrollo de software en la UNSCH.

2.6.2 Hipótesis específicas

HE1. La implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la funcionalidad del desarrollo de software en la UNSCH.

HE2. La implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la usabilidad del desarrollo de software en la UNSCH.

HE3. La implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la eficiencia del desarrollo de software en la UNSCH.

2.7 Variables

Independiente

□ Estándares NTP 12207:2016, ISO 9001 e ISO 9126

Indicadores

- Niveles de madurez
- Fiabilidad
- Calidad de servicio
- Calidad percibida
- Aprender
- Operar

- Preparar las entradas e interpretar las salidas de un programa.
- Recursos de computo
- Códigos requeridos por un programa

Dependiente

- Calidad de software

Indicadores

- Niveles de madurez
- Fiabilidad
- Calidad de servicio
- Calidad percibida
- Aprender
- Operar
- Preparar las entradas e interpretar las salidas de un programa.
- Recursos de computo
- Códigos requeridos por un programa

Matriz de operacionalización de variables

En el siguiente cuadro se presenta la matriz de operacionalización de variables.

Matriz de operacionalización de variables

Título: Características de la calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126, para la mejora del área de desarrollo de software en la UNSCH. Ayacucho 2021.

VARIABLES	DEFINICION CONCEPTUAL	DEFINICION OPERACIONAL	DIMENSIONES	INDICADORES	ITEM/ESCALA	INSTRUMENTO
Estándares NTP 12207:2016, ISO 9001 e ISO 9126	“...Incluso los desarrolladores de software más experimentados estarán de acuerdo en que obtener software de alta calidad es una meta importante. Pero, ¿cómo se define la calidad del software? En el sentido más general se definen como: Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo	Las características de la calidad de software basadas en estándares para el desarrollo de software desempeño o rendimiento, características, fiabilidad, conformidad al diseño, durabilidad, calidad del servicio, estética y calidad percibida. Permitirá evaluar y determinar los aspectos y criterios de medición individuales.	Criterios de Funcionalidad	Niveles de madurez	1, 3,10,13,	Cuestionario
				Fiabilidad	2,6,11,	
				Calidad de servicio	4,7,14,15,16	
				Calidad percibida	5,8,9,12,33,	
			Criterios de Usabilidad	Aprender	20,22,31	
				Operar	17,18,19,21,23,24,25, 26,30,32	
Preparar las entradas e interpretar las salidas de un programa.	27,28,29,34					

VARIABLES	DEFINICION CONCEPTUAL	DEFINICION OPERACIONAL	DIMENSIONES	INDICADORES	ITEM/ESCALA	INSTRUMENTO
	producen y a quienes lo utilizan...”		Criterios de Eficiencia	Recursos de computo	35,36	
				Códigos requeridos por un programa.	37	
CALIDAD DE SOFTWARE	Son atributos que reúne el software factible de ser medidos cuya valoración determina la calidad del intangible puesto en uso para satisfacer una necesidad del usuario		Valoración de Funcionalidad	Niveles de madurez	Escala de Likert: 1: Nunca 2: Pocas veces 3: A veces 4: Frecuentemente 5: Siempre	Calidad muy baja: [0 – 17> Calidad baja: [17 – 34> Calidad media: [34 – 51> Calidad alta: [51 – 68> Calidad muy alta: [68 – 85>
				Fiabilidad		
				Calidad de servicio		
			Calidad percibida			
			Valoración de Usabilidad	Aprender	Escala de Likert: 1: Nunca 2: Pocas veces	Calidad muy baja: [0 – 17> Calidad baja:
			Operar			

				Preparar las entradas e interpretar las	3: A veces 4: Frecuentemente 5: Siempre	[17 – 34> Calidad media: [34 – 51> Calidad alta: [51 – 68> Calidad muy alta:
VARIABLES	DEFINICION CONCEPTUAL	DEFINICION OPERACIONAL	DIMENSIONES	INDICADORES	ITEM/ESCALA	INSTRUMENTO
				salidas de un programa.		[68 – 85>
			Valoración de Eficiencia	Recursos de computo	Escala de Likert: 1: Nunca	Calidad muy baja: [0 – 3>
				Códigos requeridos por un programa	2: Pocas veces 3: A veces 4: Frecuentemente 5: Siempre	Calidad baja: [3 – 6> Calidad media: [6 – 9> Calidad alta: [9 – 12> Calidad muy alta: [12 – 15>

Capítulo 3. Marco metodológico

En esta parte del trabajo de investigación se implementará todo lo relacionado al tipo de estudio a realizar, la muestra con la que se trabajó, las técnicas, los instrumentos utilizados y el procedimiento seguido para diseñar el modelo propuesto.

3.1 Tipo, Método y Diseño de la Investigación

El significado original del término cuantitativo se vincula a conteos numéricos y métodos matemáticos. Actualmente, representa un conjunto de procesos organizado de manera secuencial para comprobar ciertas suposiciones. Parte de una idea que se delimita y, una vez acotada, se generan objetivos y preguntas de investigación, se revisa la literatura y se construye un marco o perspectiva teórica. (Hernández y Mendoza, 2018).

De acuerdo a Hernández (2014) “enfoque cuantitativo utiliza la recolección de datos para probar hipótesis con base en la medición numérica y el análisis estadístico, con el fin establecer pautas de comportamiento y probar teorías” (p. 4).

El estudio realizado, según las definiciones citadas, está catalogado en el enfoque cuantitativo, porque para desarrollar los objetivos e hipótesis de investigación fue necesario diseñar un instrumento de recolección de datos que incluye los estándares que evalúan los atributos de la calidad de software valorados mediante la escala de Likert, cuyos datos colectados fueron analizados con los criterios de la estadística descriptiva e inferencial y las leyes de la inferencia lógica.

Tipo de Investigación

Según Baudean (2015) “la investigación aplicada está centrado en apoyar la resolución de problemas prácticos o de acción. Lo hace aportando información obtenida de forma sistemática y metódica, así como análisis relevantes de esta información” (p. 2).

La investigación realizada es aplicada, porque estudia una realidad del proceso de automatización de las actividades académicas y administrativas de la UNSCH, tomando como parámetros de evaluación los estándares NTP 12207:2016, ISO 9001 e ISO 9126 para determinar los atributos de calidad de software e interpretados de acuerdo a los lineamientos teóricos.

Método de la investigación

El método analítico es aquel método de investigación que consiste en la desmembración de un todo descomponiéndolo en sus partes o elementos para observar las causas, naturaleza y los efectos. El análisis es la observación y examen de un hecho en particular (Lopera Echavarría, 2010).

El tipo de método elegido para este trabajo de tesis es Analítico ya que se sustenta en hechos comprobables y fundamentos racionales. Requiere verificación empírica y la constatación mediante el uso de instrumentos de medición empírica.

Diseño de la investigación

Para Abreu (2012) el diseño de investigación “explica cómo se realiza el trabajo objeto de investigación, los parámetros que se establecen y los datos estadísticos usados para evaluar la información recolectada. Mediante este enfoque se describe si es un estudio de investigación exploratorio, descriptivo, correlacional o explicativo” (p. 191).

Para responder a las preguntas de investigación, lograr los objetivos de estudio, así mismo someter las hipótesis formuladas a pruebas se empleó el diseño de la investigación cuantitativa en su variante diseño de un grupo con medición antes y después utilizando el diseño pre-experimental transversal, esto origina a que no se puede presumir que los diversos grupos de tratamiento sean inicialmente equivalentes dentro de los límites de error muestral.

Nivel de Investigación

Según su naturaleza o profundidad, el nivel de una investigación se refiere al grado de conocimiento que posee el investigador en relación con el problema, hecho o fenómeno a estudiar (Arias, 2012).

Este trabajo de tesis es de nivel descriptivo-explicativo, porque describe los atributos de la calidad de software teniendo como referencia los estándares NTP 12207:2016, ISO 9001 e ISO 9126, asimismo, busca explicar en qué medida la puesta en marcha del software en la UNSCH ha implicado que los desarrolladores del software tuvieron en consideración los estándares en cuestión para la programación y desarrollo de la sintaxis del lenguaje, para automatizar los procesos académicos y administrativos.

3.2 Población y Muestra

Población

Una población es el conjunto de todos los casos que concuerdan con una serie de especificaciones. Es la totalidad del fenómeno a estudiar, donde las entidades de la población poseen una característica común la cual se estudia y da origen a los datos de la investigación (Hernández y Mendoza, 2018).

La población está formada por 6 profesionales que laboran en la oficina de informática de la Universidad Nacional de San Cristóbal de Huamanga que evalúan los diferentes aspectos del desarrollo de software instalado, teniendo como referencia los estándares NTP 12207:2016, ISO 9001 e ISO 9126 y determinan la calidad de software.

Muestra

Es una sección o pedazo representativo poblacional, cuyas propiedades fundamentales son las de ser objetiva y reflejo de ella, de tal forma que los resultados logrados en la muestra logren generalizarse a todos los recursos que componen esa población (Carrasco, 2014).

“La muestra censal es aquella donde todas las unidades de investigación son consideradas como muestra. De allí, que la población a estudiar se precise como censal por ser simultáneamente universo, población y muestra” (Ramirez, 1997 citado en Vivanco Mena, 2017).

La muestra censal está conformada por 6 profesionales de la unidad de informática de la Universidad Nacional de San Cristóbal de Huamanga quienes evalúan sobre el software instalado en la Universidad Nacional de San Cristóbal de Huamanga la calidad del desarrollo de software teniendo como patrón los estándares NTP 12207:2016, ISO 9001 e ISO 9126.

Criterio de selección de la muestra

Para ser considerado como parte de la muestra, el entrevistado debe ser un profesional con conocimiento en desarrollo de software, activo en la oficina de informática de la Universidad Nacional de San Cristóbal de Huamanga en el año 2022.

3.3 Técnicas e instrumentos de recolección de datos

La recolección de datos significa aplicar una o más herramientas de medición para recolectar información relevante sobre las variables de la encuesta de una muestra o caso seleccionado (individuo, grupo, organización, proceso, evento, etc.). Los datos obtenidos constituyen la base del análisis. Sin ellos no hay datos ni investigación. Esta fase se requiere el establecimiento y definición previa de hipótesis y variables de investigación, tanto conceptual como operativamente, con precisión y claridad. De manera similar, al revisar las referencias, debe identificar las herramientas o métodos para medir o calificar las variables propuestas (Hernández y Mendoza, 2018).

Para la recolección de datos del presente trabajo de tesis, se utilizó la siguiente técnica:

Técnicas:

□ **Encuesta**

Según Anguita (2003) “La técnica de encuesta es ampliamente utilizada como procedimiento de investigación, ya que permite obtener y elaborar datos de modo rápido y eficaz” (p. 527).

Oncins de Frutos (1991) “La técnica de la encuesta puede considerarse dividida en varias fases: (1) Definición de objetivos e hipótesis. (2) Elaboración del cuestionario. (3) Aplicación del cuestionario. (4) Tratamiento estadístico de los resultados. (5) Elaboración del informe” (p. 1).

Una encuesta es una técnica que consiste en aplicar un cuestionario a una muestra de personas. Las encuestas brindan información sobre las opiniones, actitudes y comportamientos de los ciudadanos. (Laza, 2019).

En el presente trabajo de investigación se diseñó un cuestionario que contiene 37 ítems para evaluar el software instalado en la UNSCH, que recoge datos sobre las especificaciones del producto, del desarrollo para conocer si fue realizado de acuerdo los estándares de calidad y objetivos de la parte de planeación del proyecto, además de tomar datos de la especificación de cambios en variables y errores, estos aspectos técnicos fueron clasificados según la funcionalidad del desarrollo de software con 17 ítems, en lo que corresponde a usabilidad con 17 ítems y en el aspecto de eficiencia con 3 ítems, que permiten conocer la calidad de software.

Instrumentos:

□ **Cuestionario**

Para De Paz, (2008) “El instrumento de recolección de datos es imprescindible para la construcción de la Matriz Tripartita de Datos o simplemente la Matriz de Datos; pues mediante el instrumento de recolección de datos se obtiene la información sobre las variables” (p. 13).

El instrumento de recolección de datos es el proceso de “Crear las condiciones para la medición. Los datos son conceptos que expresan una abstracción del mundo real, de lo sensorial, susceptible de ser percibido por los sentidos de manera directa o indirecta, donde todo lo empírico es medible” (Hernández, 2010, p. 217).

De acuerdo con Meneses (2011) “un cuestionario es, por definición, el instrumento estandarizado que utilizamos para la recogida de datos durante el trabajo de campo de algunas investigaciones cuantitativas, fundamentalmente, las que se llevan a cabo con metodologías de encuestas” (p. 9).

Para Meneses (2011) “un instrumento es capaz de generar información cuantitativa, susceptible de ser tratada estadísticamente, a partir de la agregación de la información proporcionada por los participantes” (p. 13).

En el estudio, se ha dado el formato virtual al cuestionario en el formulario del Google Meet, al cual se asoció la escala de Likert, dándole un nombre “Cuestionario de decisiones de implementación de software en la Universidad Nacional de San Cristóbal de Huamanga” (Ver Anexo F) para luego determinar sus propiedades como la validez y la confiabilidad.

□ **Validez**

Para Saavedra (2021) la validez “es la eficacia con que un instrumento mide lo que se pretende”. En tanto que, Hernández y otros (2003), define la validez como el grado en que un instrumento realmente pretende medir la validez (p. 229).

La validez de juicio de experto con conocimientos sólidos en el tema es una opinión informada de personas con trayectoria en el tema, que son reconocidas por otros como expertos cualificados en éste, y que pueden dar información, evidencia, juicios y valoraciones (Garrote, 2015).

En el estudio se aplicó la validación mediante el juicio de expertos, teniendo que consultar a tres especialistas en desarrollo de software con el grado de magister o doctor, a quienes, con su conocimiento y experiencia, se les proporcionó el formato de validación que contenía los siguientes elementos:

Matriz de consistencia.

Definiciones conceptuales de las variables y dimensiones.

Matriz de operacionalización de las variables.

Matriz de validación del instrumento.

Plantilla de validación.

Con el propósito a que evalúen cada uno de los 37 ítems y su escala de medición para que ponderen la claridad en la redacción, la relevancia del ítem en el estudio y la pertinencia del ítem en la investigación.

□ **Confiabilidad**

La confiabilidad del instrumento de recolección de datos es “la fiabilidad se refiere al grado en que un instrumento produce resultados consistentes y coherentes” (Zamorano, 2013, p. 267).

Según Tuapanta (2017) “la confiabilidad tipo consistencia interna se refiere al grado en que los ítems de una escala se correlacionan entre ellos” (p. 39).

Según Castellanos (2017) “la prueba Alfa de Cronbach para medir la fiabilidad del cuestionario nos dice que el instrumento presenta una variabilidad homogénea, por encima del 0.8” (p. 4).

En el presente trabajo de tesis se aplicó la confiabilidad Alfa de Cronbach para determinar la correlación individual del ítem con respecto a la puntuación total del instrumento de recolección de datos.

3.4 Descripción de Procedimientos de Análisis de Datos

En el presente trabajo de tesis se elaboró a través de una muestra censal de seis especialistas que laboran en la Oficina General de Informática y Sistemas de la Universidad Nacional de San Cristóbal de Huamanga, los que fueron entrevistados con el cuestionario elaborado en términos de los estándares de las normas NTP 12207:2016, ISO 9001 e ISO 9126, que definen el modelo contextual de la calidad de software en función de la funcionalidad con 17 ítems, usabilidad con 17 ítems y eficiencia con 3 ítems, contexto donde la variable independiente está definida por el modelo conceptual constituidos por los estándares de las normas de calidad y a nivel operativo se tiene como variable dependiente las características de la calidad valoradas por los expertos en el trabajo que realizaron los desarrolladores de software para automatizar sus procesos académicos y administrativos de la UNSCH, que lógicamente heredan de las normas atributos de calidad según las modalidades de la escala de Likert, las respuestas fueron registradas en el formulario del Google, cuyas puntuaciones sirvieron de base para realizar otras operaciones a través de la propiedad aditiva de los ítems, como construcción de intervalos para identificar la calidad según niveles, puntuaciones totales de dimensión y del cuestionario.

Los datos fueron organizados en una hoja de cálculo Excel permitiendo a través de sus herramientas de filtro determinar a nivel general la tendencia de la calidad valorada a través del juicio de valor de los expertos de Oficina General de Informática y Sistemas, luego para realizar análisis puntuales para el desarrollo de los objetivos y contraste de la hipótesis se migro los datos al Software estadístico SPSS versión 27.0 del que se usó el menú para obtener la distribución de frecuencias de cada ítem y gráficos en tridimensional de porcentajes acumulados y luego aplicar el Test T-Student para implementar las pruebas de hipótesis para la media poblacional, para estimar la puntuación media de la calidad, la misma que fue clasificada según las modalidades de la escala de Likert (1) Calidad muy baja, (2)

Calidad baja, (3) Calidad media, (4) Calidad alta y (5) Calidad muy alta.

Técnicas de análisis de datos

La información producida por los estadísticos descriptivos e inferencial y los recursos que proporciona la lógica proposicional permitieron identificar ciertos atributos de la calidad de software que se exponen con detalle en los resultados y análisis de resultados, que muestran características a nivel operativo del software y a nivel de usuario. Asimismo, los comportamientos fueron analizados desde las teorías de la calidad expuestas en el marco teórico.

Los estadísticos descriptivos

La aplicación de las medidas de resumen estadístico describe un conjunto de atributos puntuales de la calidad de software en lo que corresponde a la funcionalidad, usabilidad y eficiencia del desarrollo de software, asimismo, el análisis conjunto de estos indicadores proporciona un diagnóstico a nivel operativo y de uso del software en cuestión.

Proceso de contraste de hipótesis

En la validación de las hipótesis se aplicó la estadística inferencial mediante el Test TStudent para la puntuación media poblacional, cada hipótesis fue evaluada con una confianza del 95% y 5% de significancia, dichos indicadores fueron interpretados mediante los lineamientos teóricos de la estadística y las teorías expuestas en el marco teórico.

Capítulo 4. Resultados y análisis de resultados

4.1 Resultados

4.1.1 Aplicación de la estadística descriptiva

En el presente acápite se desarrollan los objetivos de investigación a través de los datos recolectados con el instrumento denominado “Características de la calidad de software

basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126” que contiene 37 ítems valorados con la escala de Likert y determinan la calidad de software, dado que, su evaluación de la validez del instrumento mediante el proceso de juicio de expertos tiene una valoración porcentual media del 74% que está contenido en el intervalo de “Bueno”, en tanto que, su alfa de Cronbach es de 0.88 que revela que dicho instrumento tiene consistencia interna para evaluar el modelo conceptual establecido por la norma de calidad y establecer las relaciones formuladas en las hipótesis, asimismo, es importante mencionar que mediante la propiedad aditiva de los ítems se generan en el proceso operaciones aritméticas obteniendo la puntuación del instrumento o de sus partes con el propósito de analizar ciertos comportamientos que subyacen de la población de estudio; por lo que, a continuación se presentan un conjunto de indicadores estadísticos obtenidos con el software SPSS versión 27.0 que permiten conocer patrones de conducta en forma a priori que luego son confirmados en la parte inferencial con el contraste de hipótesis.

Estadísticos descriptivos revelan comportamientos que subyacen de la población de estudio a través del juicio de valor que realizaron los profesionales de la Oficina General de Informática y Sistemas de la Universidad Nacional de San Cristóbal de Huamanga, proporcionando detalles de los problemas cuya operatividad se instrumenta en términos de los objetivos de investigación, permite elaborar los constructos de la calidad, evidenciando algunas bondades y cuestiones que necesitan establecer acciones de la mejora continua, encuadradas en la teoría de calidad total. Los detalles y argumentos se presentan a continuación:

Primer objetivo específico

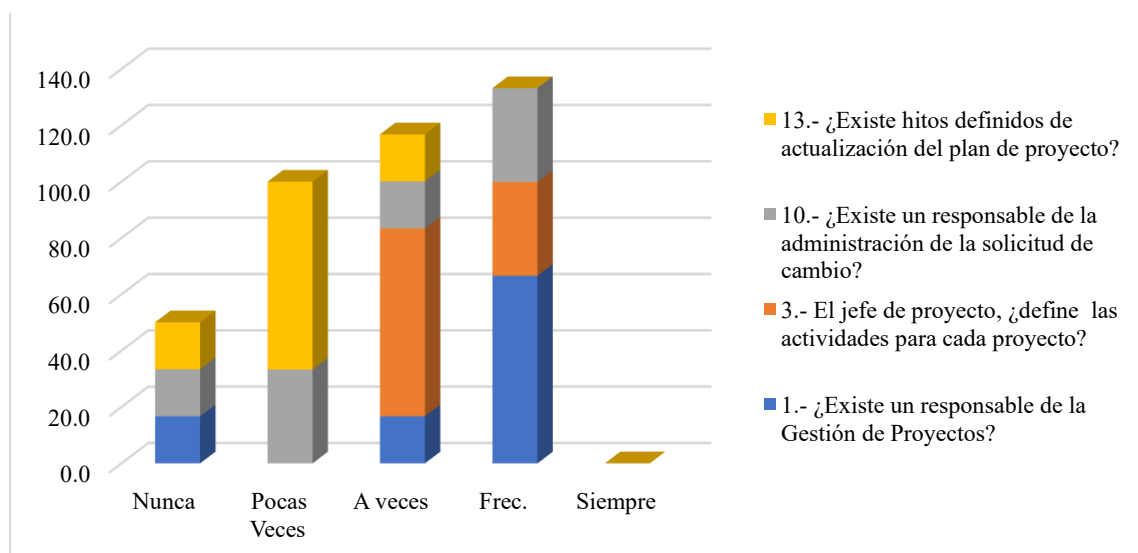
OE1. Determinar de qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la funcionalidad del desarrollo de software en la UNSCH.

Funcionalidad

Tabla 5 Valor asignado al indicador de madurez, expresado en porcentaje

Madurez	Nunca	Pocas Veces	A veces	Frec.	Siempre
1.- ¿Existe un responsable de la Gestión de Proyectos?	16.6	0.0	16.7	66.6	0.0
3.- El jefe de proyecto, ¿define las actividades para cada proyecto?	0.0	0.0	66.7	33.3	0.0
10.- ¿Existe un responsable de la administración de la solicitud de cambio?	16.7	33.3	16.7	33.3	0.0
13.- ¿Existe hitos definidos de actualización del plan de proyecto?	16.7	66.7	16.7	0.0	0.0

Figura 18 Tendencia del valor asignado al indicador de madurez, expresado en porcentaje acumulado según modalidad



Interpretación

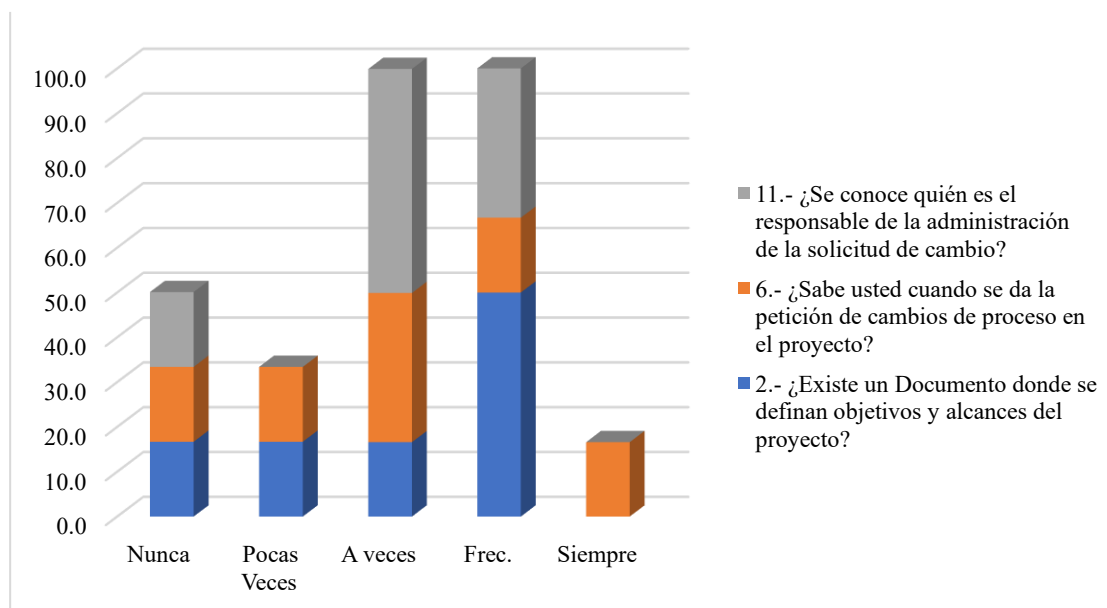
En la tabla 5 y figura 18, basando en el juicio de valor de especialistas y la similitud de puntuación del ítem revela cierto patrón de comportamiento latente que permite establecer

niveles mínimos que debe alcanzar el software para ser considerado de calidad, en este contexto se observa que el 66.7% considera que con frecuencia existe un responsable de la gestión de proyectos, asimismo, se tiene otro tanto similar de 66.7% que manifiesta que a veces el jefe de proyecto define las actividades para cada proyecto, en tanto que, el 50% afirma que a veces o con frecuencia existe un responsable de la administración de la solicitud de cambio y un 66.7% afirma que existen hitos definidos de actualización del plan de proyecto. Además, los indicadores descritos de la madurez de software revelan que las acciones se realizan de forma eventual, probablemente cuando se requiere en esta área de trabajo, aparece el responsable de la gestión del proyecto, para definir las actividades de cada proyecto, para asignar un responsable de la administración de solicitud de cambio y también fija hitos definidos de actualización del plan de proyecto, con el propósito de administrar los procesos, medir y evaluar el crecimiento de su potencial del software; por otro lado, el patrón de comportamiento descrito concuerda con la tendencia de las frecuencias acumuladas de la Figura 18 que determina que las modalidades con mayor apilamiento porcentual son Frecuentemente, seguida de A veces y Pocas veces, sacrificando la modalidad de siempre que exige el modelo conceptual de la calidad que las acciones sigan un protocolo previamente establecido.

Tabla 6 *Valor asignado al indicador de fiabilidad, expresado en porcentaje*

Fiabilidad	Nunca	Pocas Veces	A veces	Frec.	Siempre
2.- ¿Existe un Documento donde se definan objetivos y alcances del proyecto?	16.7	16.70	16.6	50.0	0.0
6.- ¿Sabe usted cuando se da la petición de cambios de proceso en el proyecto?	16.7	16.7	33.3	16.6	16.7
11.- ¿Se conoce quién es el responsable de la administración de la solicitud de cambio?	16.7	0.0	50.0	33.3	0.0

Figura 19 *Tendencia del valor asignado al indicador de fiabilidad, expresado en porcentaje acumulado según modalidad*

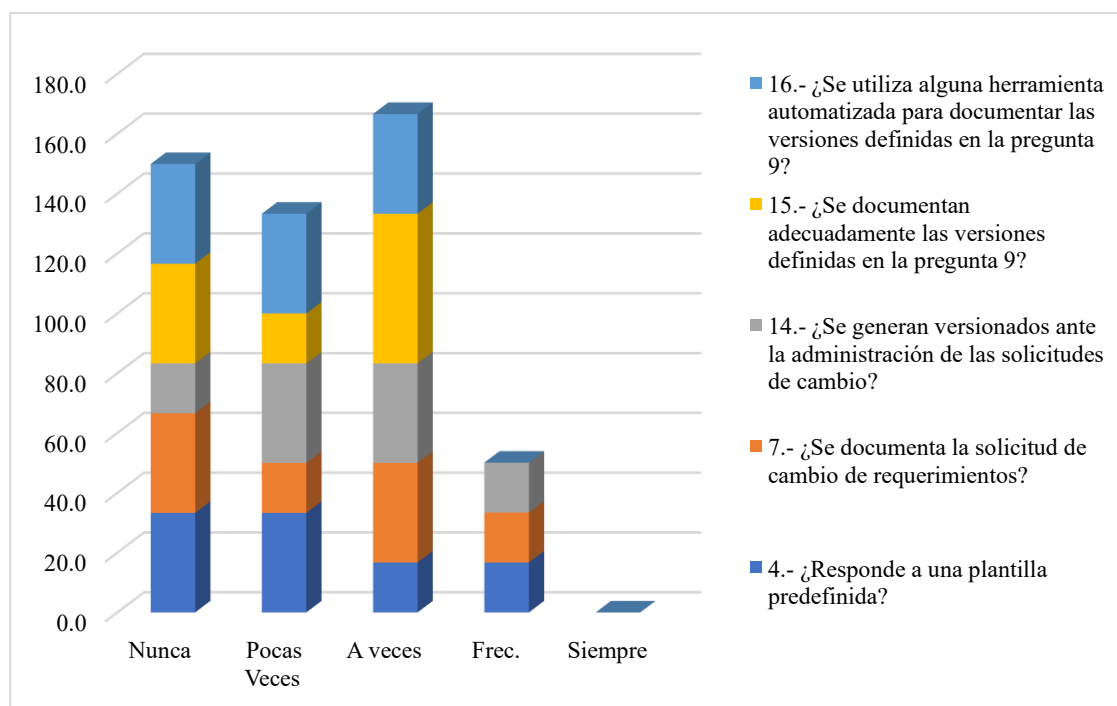


Interpretación

En la tabla 6 y figura 19, se tiene con respecto al indicador de fiabilidad, que un 76.6% menciona que a veces o frecuentemente se observa que existe un documento donde se define los objetivos y alcances del proyecto, asimismo, en forma acumulada un 50% afirma con una regularidad de a veces o frecuentemente se realiza la petición de cambios de proceso en el proyecto y otro tanto de 83.3% menciona que con la observancia de a veces o frecuentemente se conoce quien es el responsable de la administración de la solicitud de cambio. Además, los porcentajes apilados en las modalidades de la escala de Likert de la figura 19, permite entrever que frecuentemente o a veces se tiene en consideración los objetivos y alcances del proyecto para cumplir con las peticiones de cambios de proceso y de la administración de dicha solicitud de cambio en las acciones que determinan la fiabilidad de la puesta en marcha del software utilizado en la UNSCH, denotando que se tiene un horizonte definido para implementar los procesos que requiere el proyecto.

Tabla 7 Valor asignado al indicador de calidad de servicio, expresado en porcentaje

Calidad de servicio	Nunca	Pocas Veces	A veces	Frec.	Siempre
4.- ¿Responde a una plantilla predefinida?	33.3	33.3	16.7	16.7	0.0
7.- ¿Se documenta la solicitud de cambio de requerimientos?	33.3	16.7	33.3	16.7	0.0
14.- ¿Se generan versionados ante la administración de las solicitudes de cambio?	16.7	33.3	33.3	16.7	0.0
15.- ¿Se documentan adecuadamente las versiones definidas en la pregunta 9?	33.3	16.7	50.0	0.0	0.0
16.- ¿Se utiliza alguna herramienta automatizada para documentar las versiones definidas en la pregunta 9?	33.3	33.3	33.3	0.0	0.0

Figura 20 Tendencia del valor asignado al indicador de la calidad de servicio expresado en porcentaje acumulado según modalidad.

Interpretación

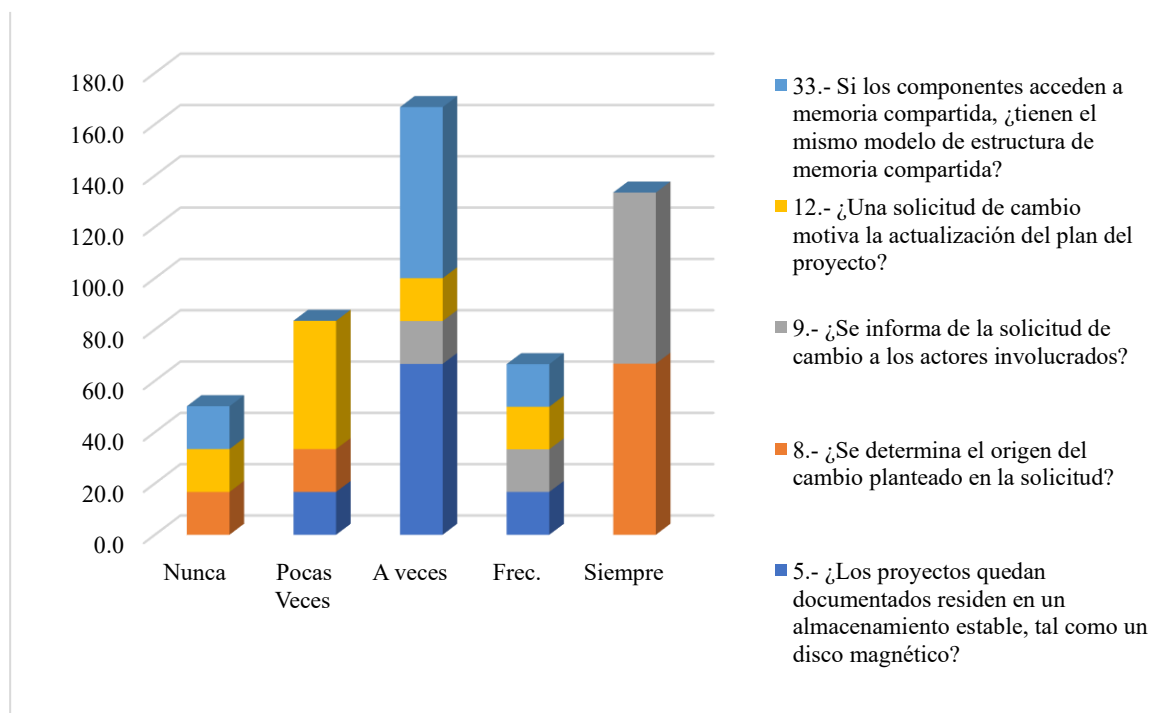
En la tabla 7 y figura 20, se observa en la calidad de servicio, que en un 66.6% los entrevistados consideran que nunca o pocas veces se responde a una plantilla predefinida,

además, un 50% menciona que a veces o frecuentemente se documenta la solicitud de cambio de requerimiento, análogamente, un 50% menciona que a veces o frecuentemente se generan versiones ante la administración de solicitudes de cambio, asimismo, con el mismo porcentaje y frecuencia, un 50% menciona que a veces se documentan adecuadamente las versiones definidas y notificando a los actores involucrados, en tanto que, el 66.6% considera que nunca o pocas veces se utiliza alguna herramienta automatizada para documentar las versiones definidas y notificar a los actores involucrados. Agregando a lo descrito, de los porcentajes apilados se determina que la regularidad de cumplimiento de las acciones y procesos de la calidad de servicio recae en las modalidades de a veces, seguido de nunca y pocas veces, denotando que para la mayoría de situaciones no se tiene en cuenta los criterios de calidad definidos previamente, que son de cumplimiento permanente de roles en una entidad.

Tabla 8 *Valor asignado al indicador de calidad percibida, expresada en porcentaje*

Calidad percibida	Nunca	Pocas	Veces	A veces	Frec.	Siempre
5.- ¿Los proyectos quedan documentados residen en un almacenamiento estable, tal como un disco magnético?	0.0	16.7	66.6	16.7	0.0	
8.- ¿Se determina el origen del cambio planteado en la solicitud?	16.7	16.7	0.0	0.0	66.7	
9.- ¿Se informa de la solicitud de cambio a los actores involucrados?	0.0	0.0	16.7	16.6	66.7	
12.- ¿Una solicitud de cambio motiva la actualización del plan del proyecto?	16.7	50.0	16.7	16.6	0.0	
33.- Si los componentes acceden a memoria compartida, ¿tienen el mismo modelo de estructura de memoria compartida?	16.7	0.0	66.7	16.6	0.0	

Figura 21 *Tendencia del valor asignado al indicador de la calidad percibida, expresado en porcentaje acumulado según modalidad*



Interpretación

En la tabla 8 y figura 21, se tiene en los criterios de calidad percibida, que un 83.3% consideran que a veces o frecuentemente los proyectos quedan documentados y residen en un almacenamiento estable, tal como un disco magnético, más aún, el 66.7% menciona que siempre se determina el origen del cambio planteado en la solicitud, además, el 83.3% afirma que frecuentemente o siempre se informa de la solicitud de cambio a los actores involucrados, en tanto que, un 66.7% menciona que nunca o pocas veces una solicitud de cambio motiva la actualización del plan del proyecto y un 83.3% considera que a veces o frecuentemente los componentes de memoria compartida ¿tienen el mismo modelo de estructura de memoria compartida?. Sin embargo, frente a estos indicadores descritos, en la figura 21 los porcentajes apilados determinan que el patrón de regularidad de cumplimiento de acciones es de a veces, seguido de siempre, esto denota que en algunas acciones u procesos son de cumplimiento obligatorio tal como el origen del cambio planteado en la

solicitud e informar a los actores involucrados para dar trazabilidad al proceso, en tanto que, los demás procesos se cumplen solo cuando se requiere o se necesita tener memoria para otras situaciones de interés, estos comportamientos indican de la calidad percibida, que se cumple en un nivel promedio con ligeros comportamientos que tienden a la mejora de procesos.

El análisis conjunto de las premisas obtenidas del primer objetivo específico, cuya información fue extraída de los ítems en términos de la escala de Likert, determinan que la funcionalidad de software implementado por la UNSCH para el control de sus acciones y procesos cumple características de calidad media con cierta tendencia a la mejora en ciertos criterios como contar con un responsable de la gestión de proyectos, contar con documentos donde se tiene los objetivos y alcances del proyecto, así como, aplicar un protocolo para determinar el origen del cambio planteado en una petición e informan a los actores involucrados de dichos cambios; en los demás aspectos de la funcionalidad, se prioriza si la acción se implementa o no, probablemente sea por lo que, los profesionales encargados desempeñan múltiples funciones obligándose a distribuir su tiempo en el cumplimiento de ciertas acciones prioritarias o porque la acción a implementar carece de relevancia para la institución, que no amerita a documentar en forma virtual y física la trazabilidad de los hechos.

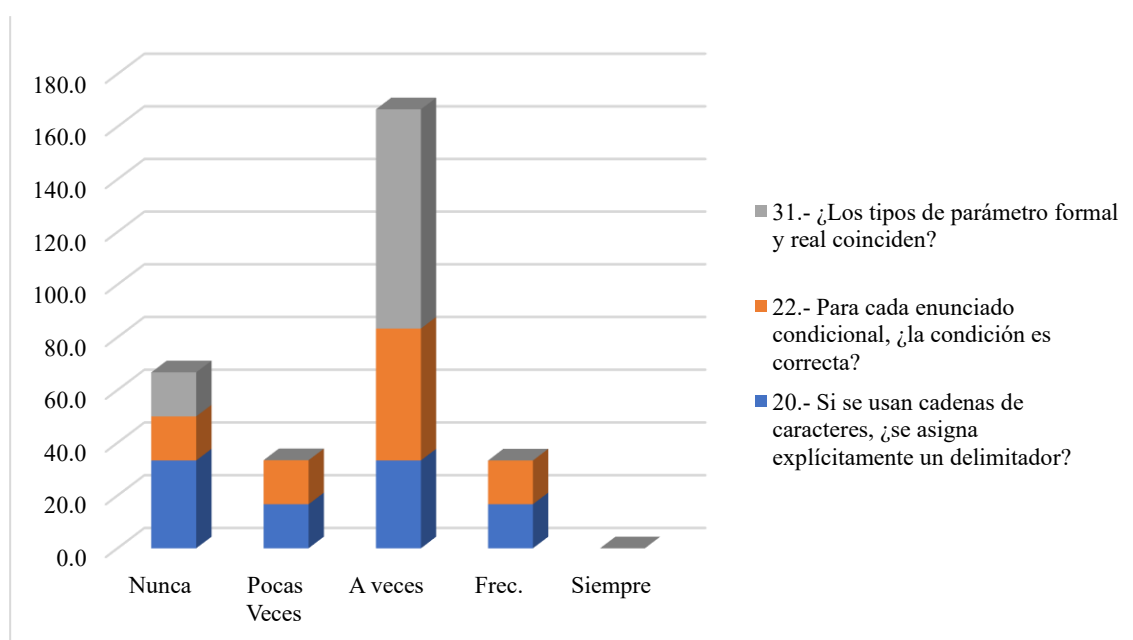
Segundo objetivo específico

OE2. Determinar de qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la usabilidad del desarrollo de software en la UNSCH.

Usabilidad

Tabla 9 Valor asignado al indicador de aprender, expresada en porcentaje

Aprender	Nunca	Pocas Veces	A veces	Frec.	Siempre
20.- Si se usan cadenas de caracteres, ¿se asigna explícitamente un delimitador?	33.3	16.7	33.3	16.7	0.0
22.- Para cada enunciado condicional, ¿la condición es correcta?	16.7	16.7	50.0	16.6	0.0
31.- ¿Los tipos de parámetro formal y real coinciden?	16.7	0.0	83.3	0.0	0.0

Figura 22 Tendencia del valor asignado al indicador de aprender, expresado en porcentaje acumulado según modalidad

Interpretación

En la tabla 9 y figura 22, se tiene que el proceso de aprender en el contexto de la dimensión de usabilidad de software, un 50% considera con una regularidad de a veces o frecuentemente que, si se usan cadenas de caracteres, ¿Se utiliza explícitamente un delimitador?, además, un 66.6% manifiesta que a veces o frecuentemente se procede a

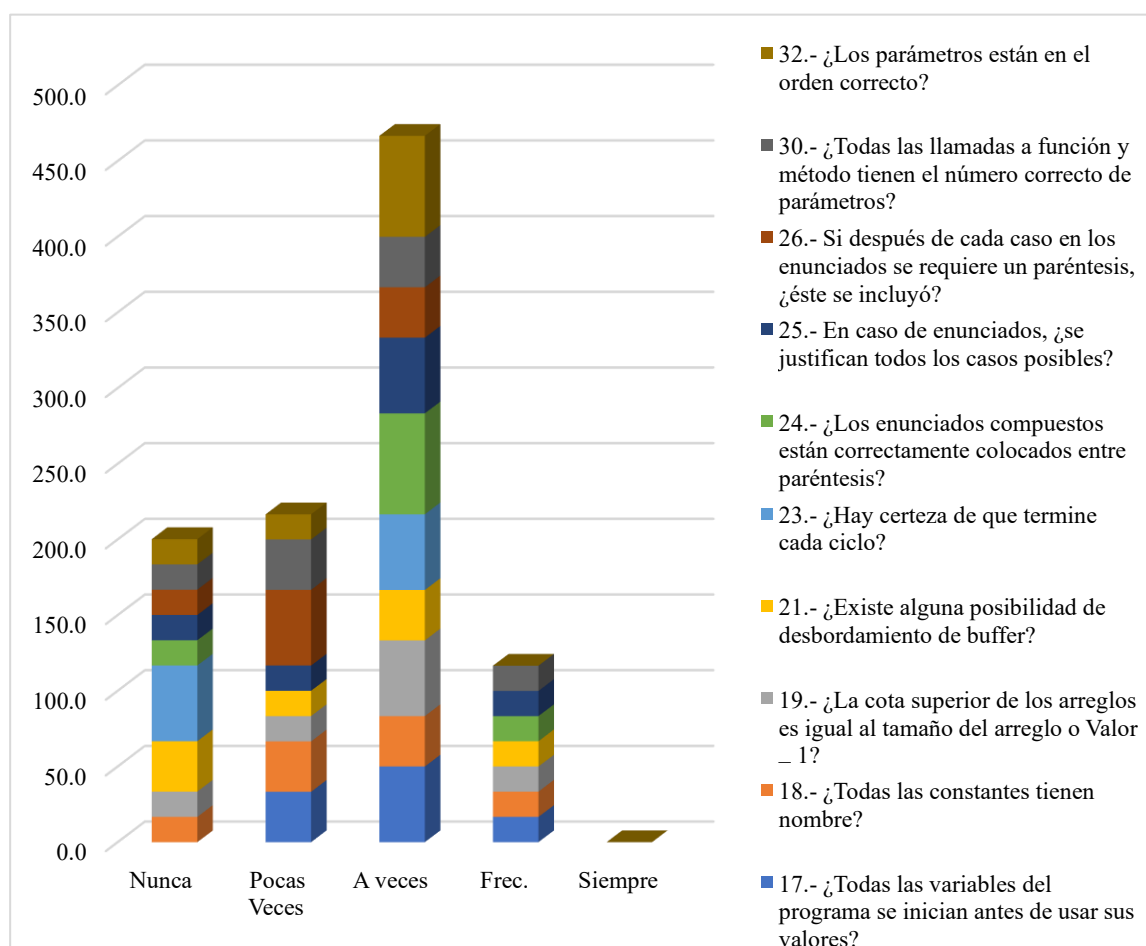
implementar para cada enunciado condicional en su protocolo correspondiente, si ¿la condición es correcta?, y un 83.3% afirma que a veces es necesario conocer que ¿Los tipos de parámetro formal y real coinciden?. Por consiguiente, los indicadores citados revelan que el proceso de aprender necesita de conocer ciertos protocolos para aplicarlos según su naturaleza delimitadora o condicional; según los porcentajes apilados de la figura 22, se establece que la modalidad de a veces concentra la regularidad de respuesta, que denota que no es una política de trabajo tener en cuenta estos conocimientos para resolver una tarea o acción en materia de usabilidad de software.

Tabla 10 Valor asignado al indicador de operar, expresada en porcentaje

Operar	Nunca	Pocas	Veces	A veces	Frec.	Siempre
17.- ¿Todas las variables del programa se inician antes de usar sus valores?	0.0	33.3	50.0	16.7	0.0	
18.- ¿Todas las constantes tienen nombre?	16.7	33.3	33.3	16.7	0.0	
19.- ¿La cota superior de los arreglos es igual al tamaño del arreglo o Valor – 1?	16.7	16.7	50.0	16.6	0.0	
21.- ¿Existe alguna posibilidad de desbordamiento de buffer?	33.3	16.7	33.3	16.7	0.0	
23.- ¿Hay certeza de que termine cada ciclo?	50.0	0.0	50.0	0.0	0.0	
24.- ¿Los enunciados compuestos están correctamente colocados entre paréntesis?	16.7	0.0	66.7	16.6	0.0	
25.- En caso de enunciados, ¿se justifican todos los casos posibles?	16.7	16.7	50.0	16.6	0.0	
26.- Si después de cada caso en los enunciados se requiere un paréntesis, ¿éste se incluyó?	16.7	50.0	33.3	0.0	0.0	
30.- ¿Todas las llamadas a función y método						

tienen el número correcto de parámetros?	16.7	33.3	33.3	16.7	0.0
32.- ¿Los parámetros están en el orden correcto?	16.7	16.7	66.7	0.0	0.0

Figura 23 Tendencia del valor asignado al indicador de operar, expresado en porcentaje acumulado según modalidad



Interpretación

En la tabla 10 y figura 23, se observa que, en el indicador de operar de la dimensión de usabilidad, un 66.7% considera que a veces o frecuentemente todas las variables del programa se inician antes de usar sus valores; asimismo, un 50% manifiesta que nunca o

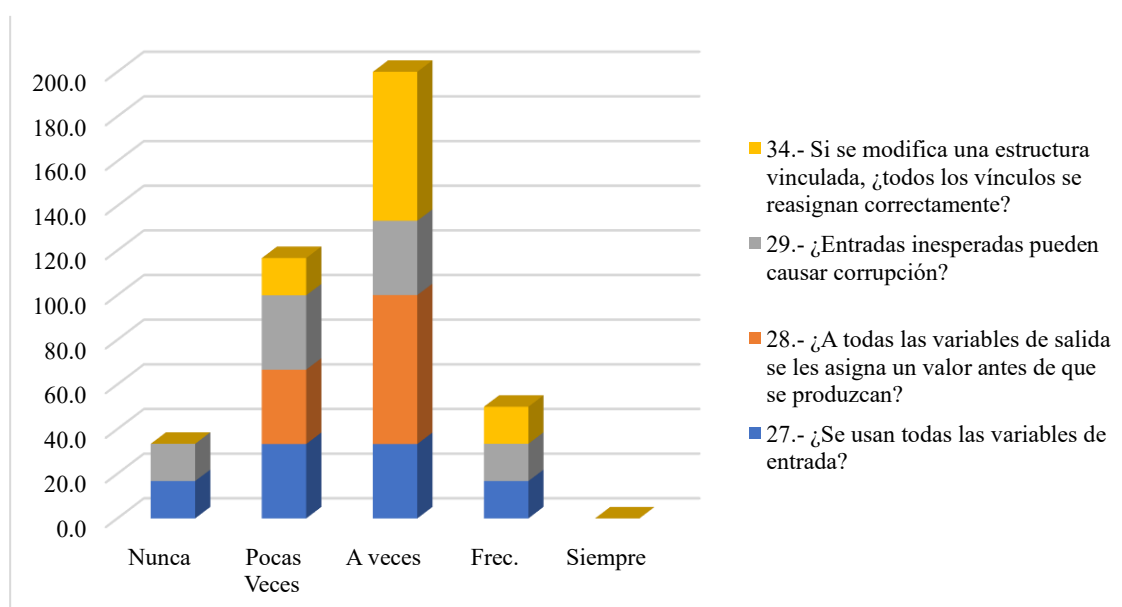
pocas veces las constantes tienen nombre; análogamente, el 66.7% menciona que a veces o frecuentemente la cota superior de los arreglos es igual al tamaño del arreglo o Valor – 1; un 50% afirma que a veces o frecuentemente existe alguna posibilidad de desbordamiento de buffer; otro 50% considera que a veces hay certeza de que termine cada ciclo; un 83.3% menciona que los enunciados compuestos están correctamente colocados entre paréntesis; en tanto que, un 66.7% menciona que a veces o frecuentemente en caso de enunciados, se justifican todos los casos posibles; en tanto que, un 66.7% considera que nunca o pocas veces, si después de cada caso en los enunciados se requiere un paréntesis, ¿éste se incluyó?; un 50% piensa que a veces o frecuentemente todas las llamadas a función y método tienen el número correcto de parámetros, y un 66.7% de los entrevistados afirma que a veces los parámetros están en el orden correcto. Los indicadores expuestos y los porcentajes acumulados de los ítems, revelan que la operabilidad del software usado por la UNSCH funciona a una capacidad media que tiene ciertas situaciones de sintaxis de programación que superar y almacenamiento, por lo que, la mayoría de los entrevistados valoran la operabilidad con un atributo de a veces, pero que en términos de la calidad total la operabilidad debe darse en una regularidad de siempre o que se cumple con todos los criterios que exige la programación y el sistema de almacenamiento.

Tabla 11 *Valor asignado al indicador de preparar las entradas e interpretar salidas de un programa, expresada en porcentaje*

Preparar las entradas e interpretar salidas	Nunca	Pocas Veces	A veces	Frec.	Siempre
27.- ¿Se usan todas las variables de entrada?	16.7	33.3	33.3	16.7	0.0
28.- ¿A todas las variables de salida se les asigna un valor antes de que se produzcan?	0.0	33.3	66.7	0.0	0.0

29.- ¿Entradas inesperadas pueden causar corrupción?	16.7	33.3	33.3	16.7	0.0
34.- Si se modifica una estructura vinculada, ¿todos los vínculos se reasignan correctamente?	0.0	16.7	66.7	16.6	0.0

Figura 24 Tendencia del valor asignado al indicador de preparar las entradas e interpretar salidas de un programa, expresado en porcentaje acumulado según modalidad



Interpretación

En la tabla 11 y figura 24, se presenta las particularidades de preparar las entradas e interpretar las salidas de un programa, el 50% de los entrevistados menciona con una regularidad de a veces o frecuentemente se usan todas las variables de entrada; asimismo, un 66.7% menciona que a veces a todas las variables de salida se les asigna un valor antes de que se produzcan; además, un 50% afirma que a veces o frecuentemente aparecen entradas inesperadas que pueden causar corrupción y un 83.3% considera que a veces o frecuentemente si se modifica una estructura vinculada, todos los vínculos se reasignan correctamente. Los indicadores descritos y los porcentajes acumulados o apilados en la figura

24, permite discriminar que las funciones de entrada – salida del software en cuestión se cumple mayoritariamente en una regularidad de a veces, probablemente hayan observado en los procesos algunas fallas de ejecución inesperadas por violación de alguna especificación en la asignación de parámetros a las variables de entrada / salida.

El análisis conjunto de la información obtenida en el segundo objetivo específico, determinan que los atributos de calidad de la dimensión de usabilidad, permite inferir de forma lógica que las puntuaciones de los especialistas se centran en la regularidad de a veces seguida de pocas veces, este patrón de comportamiento revela que se cumplen ciertas rutinas o protocolos en el proceso de aprender u operar, pero que todavía no alcanzaron su verdadero potencial de uso porque se viene utilizando bajo ciertas condiciones que no son las óptimas; estos atributos según la valoración de los especialistas a través de los ítems establecen un patrón de conducta que corresponde a una calidad media en la dimensión de usabilidad, que supone que existen retos encuadrados en la teoría de calidad total para que los usuarios puedan aprender y operar con facilidad.

Tercer objetivo específico

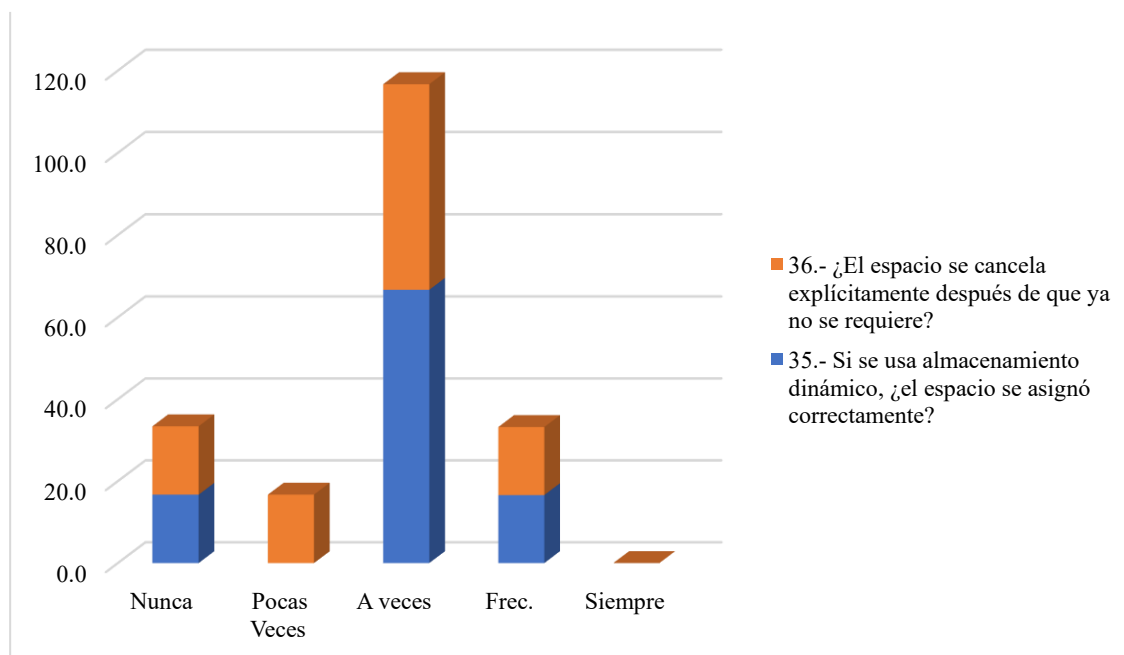
OE3. Determinar de qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la eficiencia del desarrollo de software en la UNSCH.

Eficiencia

Tabla 12 Valor asignado al indicador de recurso de cómputo, expresada en porcentaje

Recursos de computo	Nunca	Pocas Veces	A veces	Frec.	Siempre
35.- Si se usa almacenamiento dinámico, ¿el espacio se asignó correctamente?	16.7	0.0	66.7	16.6	0.0
36.- ¿El espacio se cancela explícitamente después de que ya no se requiere?	16.7	16.7	50.0	16.6	0.0

Figura 25 *Tendencia del valor asignado al indicador recurso de cómputo, expresado en porcentaje acumulado según modalidad*



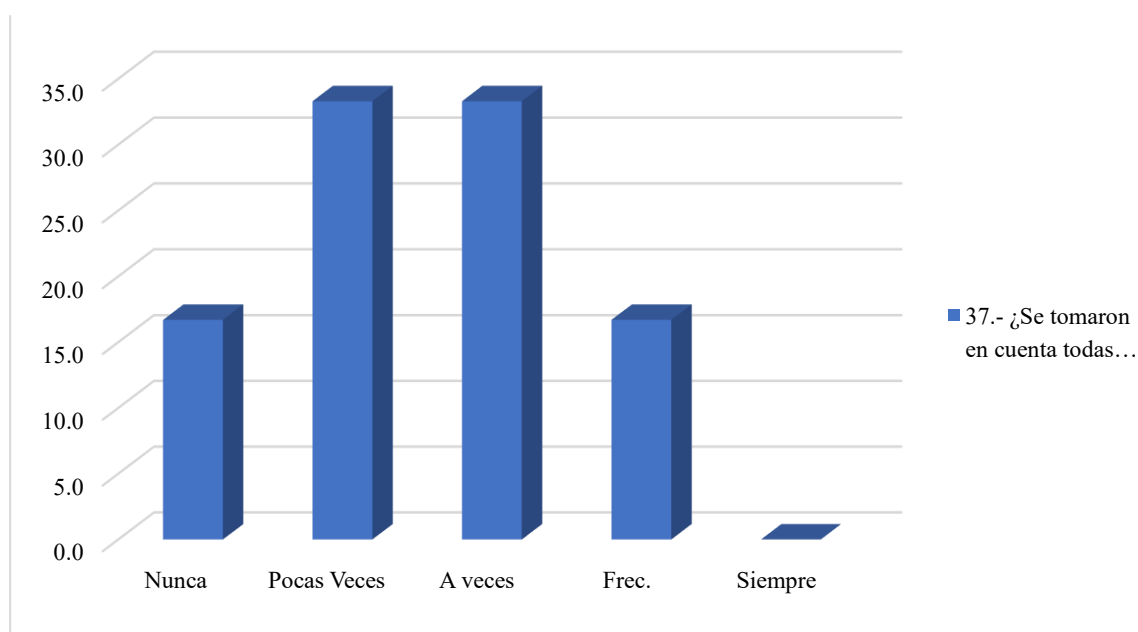
Interpretación

En la tabla 12 y figura 25, se observa en el aspecto de recurso de cómputo, que un 83.3% de los entrevistados menciona que a veces o frecuentemente si se usa el almacenamiento dinámico, ¿el espacio se asignó correctamente?, y un 66.6% considera que a veces o frecuentemente el espacio se cancela explícitamente después de que ya no se requiere. Los indicadores descritos y los porcentajes acumulados en la figura 25, revelan a través de la valoración de los expertos que a veces se usa el almacenamiento dinámico en un espacio asignado correctamente, asimismo, que el espacio se cancela de forma explícitamente cuando ya no se requiere, pero dichos atributos en la actividad de los recursos de cómputo son calificados como de calidad media, probablemente porque el sistema de almacenamiento dinámico no soporta grandes cantidades de datos o información en el espacio asignado.

Tabla 13 Valor asignado al indicador de códigos requeridos por un programa, expresada en porcentaje

Códigos requeridos por un programa	Nunca	Pocas Veces	A veces	Frec.	Siempre
37.- ¿Se tomaron en cuenta todas las posibles condiciones de error?	16.7	33.3	33.3	16.7	0.0

Figura 26 Tendencia del valor asignado al indicador de códigos requeridos por un programa, expresado en porcentaje



Interpretación

En la tabla 13 y figura 26, se tiene en el aspecto de códigos requeridos por un programa, que un 50% de los especialistas considera que a veces o frecuentemente se tomaron en cuenta todas las posibles condiciones de error. La tendencia del indicador revela que el funcionamiento del programa detecta en términos medios los posibles errores, probablemente se necesita implementar o probar el funcionamiento óptimo de algunos códigos de programa, este atributo, según la valoración de los especialistas está en una regularidad de a veces seguido de pocas veces, lo cual indica que el software goza en este aspecto de una calidad media o por debajo de la calidad media.

El análisis de la información producida en el tercer objetivo específico que articula los ítems de eficiencia determina que los atributos de Recursos de cómputo y Códigos requeridos por un programa, son de calidad media y que requieren mejora de ciertos atributos en lo que corresponde al sistema de almacenamiento dinámico y condiciones para detectar error, dado que, son aspectos importantes para documentar las acciones y procesos que permiten verificar la trazabilidad de un hecho.

Objetivo general

Determinar de qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora el desarrollo de software en la UNSCH.

La información proporcionada por los objetivos específicos en los que se concreta que, la funcionalidad identificada en el software utilizado por la UNSCH es de calidad media, asimismo, en la dimensión de usabilidad y eficiencia fueron calificados en forma implícita por los especialistas con atributos de calidad media, frente a estas premisas y usando las reglas de la lógica, permite inferir que el software en cuestión tiene implementado estructura interna y externa características de calidad media, debido a que algunos estándares de las normas NTP 12207:2016, ISO 9001 e ISO 9126 no se visualizan o identifican por los expertos en la regularidad que se necesitan para calificarlos con atributos de calidad muy alta.

4.1.2 Aplicación de la estadística inferencial

En el proceso de contraste de hipótesis es necesario precisar que según el diseño de investigación los criterios de calidad especificados por los estándares NTP 12207:2016, ISO 9001 e ISO 9126 definen un modelo conceptual que sirve para evaluar el software utilizado por la UNSCH para automatizar sus procesos académicos y administrativos correspondientes,

cuyas funciones y procesos valorados en base a los estándares determinan atributos de calidad a través de la asignación de valor que hacen los especialistas que usan diariamente los profesionales de la Institución para realizar sus labores y cuyo juicio de valor es recogido a través del cuestionario denominado “Características de la calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126”, dicho procedimiento, está sustentado por la teoría de los diseños experimentales cuyas variables independientes son controladas por el investigador, que pueden ser de naturaleza cualitativa o cuantitativa cuyo efecto es medido a través de la variable dependiente, en este caso, los estándares son valuados por los especialistas al identificar dichos atributos en el software puesto en marcha en la UNSCH y registrados en el cuestionario, asimismo, mediante la propiedad aditiva de los ítems asociados a una escala de Likert permite hacer operaciones con las puntuaciones asignadas, estas se traducen en puntuación total de dimensión o del test, que permiten identificar patrones de comportamiento que subyacen de la población de estudio.

Otro aspecto importante en el proceso de contraste de hipótesis, es determinar si los datos recogidos a través del cuestionario tienen comportamiento normal y fueron recogidos de forma aleatoria, por tanto, en forma a priori se muestra en la tabla 14 y la tabla 15, respectivamente.

Tabla 14 Prueba de hipótesis de la normalidad de datos asociados a las características que definen la calidad de software

Variables	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig. “p”	Estadístico	gl	Sig. “p”
Calidad_de_Software	,278	6	,162	,854	6	,170
Funcionalidad	,223	6	,200*	,819	6	,087

Usabilidad	,213	6	,200*	,920	6	,505
Eficiencia	,333	6	,036	,827	6	,101

*. Esto es un límite inferior de la significación verdadera.

a. Corrección de significación de Lilliefors

Proceso de contraste de hipótesis para demostrar que los datos de la calidad de software provienen de una población de comportamiento normal.

Formulación de Hipótesis

H₀: Los datos de la puntuación total de la calidad de software se distribuyen en forma normal.

H₁: Los datos de la puntuación total de la calidad de software se distribuyen en forma normal.

Nivel de significancia

El nivel de significancia es $\alpha = 0,05$.

Regla de decisión

Si el $p > \alpha$, Aceptar H_0 . (Normal)

Si el $p < \alpha$, Rechazar H_0 . (No normal)

Estadístico de contraste

El estadístico de contraste por el tamaño de muestra pequeño ($n < 50$) se aplicó el Test de Shapiro – Wilk que se muestra en la tabla 14, para la puntuación total que define la calidad de software.

Interpretación

En la tabla 14 para la puntuación de Calidad de Software se tiene que el valor de $p = 0,170$ es mayor a $\alpha = 0,05$ para el estadístico de contraste de Shapiro-Wilk = 0.854; lo cual indica que se acepta la hipótesis nula que postula que “Los datos de la puntuación total de la

calidad de software se distribuyen en forma normal” con un 95% de confianza y 5% de significancia.

Aplicando este procedimiento de contraste para las otras dimensiones, como la puntuación de funcionalidad, usabilidad y eficiencia, basta con evaluar el p – valor o significancia asintótica ubicado en la última columna de la tabla 14 del Test de Shapiro-Wilk para determinar que en todos los casos $p > \alpha$, por tanto, se infiere que, los datos de la funcionalidad, usabilidad y eficiencia se distribuyen en forma normal.

Tabla 15 Prueba de hipótesis de la aleatoriedad de datos asociados a las características que definen la calidad de software

Estadísticos	Funcionalidad	Usabilidad	Eficiencia	Calidad de software
Valor de prueba ^a	50	43	8	100
Casos < Valor de prueba	3	3	1	3
Casos \geq Valor de prueba	3	3	5	3
Casos totales	6	6	6	6
Número de rachas	2	2	3	2
Z	-1,369	-1,369	,000	-1,369
Sig. asintótica(bilateral) “p”	,171	,171	1,000	,171

Proceso de contraste de hipótesis para demostrar que los datos de la calidad de software fueron seleccionados de forma aleatoria.

Formulación de Hipótesis

H₀: Los datos de la puntuación total de la calidad de software se obtuvieron de forma aleatoria.

H₁: Los datos de la puntuación total de la calidad de software no se obtuvieron de forma aleatoria.

Nivel de significancia

El nivel de significancia es $\alpha = 0,05$.

Regla de decisión

Si el $p > \alpha$, Aceptar H_0 . (Aleatorio)

Si el $p < \alpha$, Rechazar H_0 . (No aleatorio)

Estadístico de contraste

El estadístico de contraste no paramétrico que detecta aleatoriedad de los datos es el Test de Rachas que se muestra la tabla 15 en la última columna, para la puntuación total que define la calidad de software.

Interpretación

En la tabla 15 para la puntuación de Calidad de Software se tiene que el valor de $p = 0,171$ es mayor a $\alpha = 0,05$ para el estadístico de contraste $Z = -1.369$; lo cual indica que se acepta la hipótesis nula que postula que “Los datos de la puntuación total de la calidad de software se obtuvieron de forma aleatoria” con un 95% de confianza y 5% de significancia.

En este orden aplicamos a modo de resumen el proceso de contraste de aleatoriedad a las puntuaciones de las dimensiones de funcionalidad, usabilidad y eficiencia, para tal propósito basta analizar el p – valor de la última fila de la tabla 15 cumple con la regla de decisión que $p > \alpha$, por tanto, se Aceptar H_0 que postula que las puntuaciones de la funcionalidad, usabilidad y eficiencia fueron tomadas en forma aleatoria.

Por consiguiente, de los estadísticos descritos de las tablas 14 y 15, se determina que los datos de la calidad de software y de sus dimensiones funcionalidad, usabilidad y eficiencia, cumplen con el supuesto de normalidad, por tanto, dichas puntuaciones deben ser analizados aplicando las pruebas de la estadística paramétrica para detectar los patrones de comportamiento de la calidad de software puesto en marcha en la UNSCH. Bajo este contexto, se aplica la prueba de hipótesis para la puntuación media para muestras pequeñas

con el T – Student del software SPSS versión 27.0, como estimación de la media poblacional de la calidad, se ha construido intervalos de identificación de la calidad con las puntuaciones de la escala de Likert y el número de ítems que agrupa la dimensión, para asignar el valor del parámetro poblacional en la prueba T- Student que debe administrarse en el ingreso de datos en el menú del SPSS. Por ejemplo:

La dimensión de funcionalidad tiene 17 ítems y la escala de Likert tiene una codificación 1: Nunca; 2: Pocas veces; 3: A veces; 4: Frecuentemente y 5: Siempre, de la que podemos establecer una relación biunívoca con las puntuaciones 1; 2; 3; 4 y 5 que definen atributos asociado a la calidad, en esta secuencia de correspondencia se tiene a 1: Calidad muy baja; 2: Calidad baja; 3: Calidad media, 4: Calidad alta y 5: Calidad muy alta, para cada ítem y extender según el número de ítems de la dimensión de la variable, obteniendo intervalos y su marca de clase, respectivamente: $[0 - 17>$: Calidad muy baja y su marca de clase $8,5 = (0+17)/2$; $[17 - 34>$: Calidad baja y 25,5; $[34 - 51>$ Calidad media y 42,5; $[51 - 68>$: Calidad alta y 59.5; $[68 - 85>$: Calidad muy alta y 76,5. La marca de clase o el punto medio es el valor representativo del intervalo empírico, pero puede fungir de estimación del parámetro poblacional denotado por la media para cada modalidad de la calidad.

La dimensión de usabilidad también tiene 17 ítems, por tanto, la construcción de la escala de la calidad de funcionalidad se adopta para usabilidad.

La dimensión de eficiencia tiene 3 ítems, por tanto, su escala es: $[0 - 3>$: Calidad muy baja y su marca de clase $1,5 = (0+3) /2$; $[3 - 6>$: Calidad baja y 4,5; $[6 - 9>$ Calidad media y 7,5; $[9 - 12>$: Calidad alta y 10,5; $[12 - 15>$: Calidad muy alta y 13,5.

Para la calidad de software o el cuestionario contiene 37 ítems, por tanto, su escala es: $[0 - 37>$: Calidad muy baja y su marca de clase $18,5 = (0+37) /2$; $[37 - 74>$: Calidad baja y 55,5; $[74 - 111>$ Calidad media y 92,5; $[111 - 148>$: Calidad alta y 129,5; $[148 - 185>$: Calidad muy alta y 166,5.

Luego de haber realizado las construcciones basadas en la propiedad aditiva de los ítems, se implementa la prueba estadística T – Student tomando como parámetros poblacionales la marca de clase para la modalidad de Calidad alta que fungirá como media poblacional, es decir para Funcionalidad $\mu = 59,5$; análogamente, para Usabilidad la media poblacional es $\mu = 59,5$; para Eficiencia la media poblacional es $\mu = 10,5$ y para la Calidad de Software $\mu = 129,5$.

En este contexto del constructo de la calidad, haciendo una valuación optimista de la calidad de software puesto en uso en la UNSCH, postulamos que la puntuación media de las dimensiones está en la magnitud de 59,5 para funcionalidad, y también para usabilidad, en tanto que, para eficiencia es 10,5 y para calidad de software es 129,5.

4.1.3 Contraste de hipótesis de la investigación

En el proceso de contraste de las hipótesis de interés se combina los criterios de la estadística inferencial paramétrica, las reglas de la lógica y los lineamientos teóricos proporcionados por los estándares de las normas NTP 12207:2016, ISO 9001 e ISO 9126 que definen la calidad software o proponen un modelo conceptual para determinar los atributos de la calidad y fija un horizonte a seguir por la Institución para estar inmersa en la mejora continua.

Primera hipótesis específica.

HE1: La implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la funcionalidad del desarrollo de software en la UNSCH.

Tabla 16 Estadísticos de la muestra correspondiente a funcionalidad

Variable	N	Media	Desv. Desviación	Desv. Error promedio
Funcionalidad	6	45,33	11,911	4,863

Tabla 17 Prueba de hipótesis para puntuación media de la funcionalidad

Variable	Valor de prueba = 59,5					
	t	gl	Sig. (bilateral)	Diferencia de medias	95% de intervalo de confianza de la diferencia	
					Inferior	Superior
Funcionalidad	-2,913	5	,033	-14,167	-26,67	-1,67

Formulación de Hipótesis

H₀: La puntuación media de la dimensión de funcionalidad es igual a 59,5.

H₁: La puntuación media de la dimensión de funcionalidad es diferente a 59,5.

En forma simbólica se tiene:

$$H_0: \mu = 59,5 \quad \& \quad H_1: \mu \neq 59,5$$

Nivel de significancia

El nivel de significancia teórica es $\alpha = 0,05$.

Regla de decisión

Si el $p > \alpha$, Aceptar H₀.

Si el $p < \alpha$, Rechazar H₀.

Estadístico de contraste

El estadístico de contraste paramétrico T-Student apropiado para estudiar muestras pequeñas, cuyos datos tienen comportamiento normal.

Interpretación

En la tabla 16 y tabla 17 se tiene los estadísticos de resumen para calcular el

$$\text{estadístico } t = \frac{s}{\sqrt{n}} = \frac{11,911}{\sqrt{6}} = -2,913 \quad \bar{x} - \mu \quad 45,33 - 59,5 \quad \text{y su p - valor} = 0,033 < \alpha = 0,05, \text{ que}$$

determinan que se rechaza la hipótesis nula H_0 y se acepta la hipótesis alterna H_1 que afirma que la media poblacional $\mu \neq 59,5$ con un 95% de confianza y 5% de significancia, pero la diferencia aritmética de $\bar{x} - \mu = -14,17$ que denota que la media muestral de 45,33 está por debajo de la media poblacional, estaría próxima a la marca de clase de 42,5 del intervalo que denota calidad media: $[34 - 51>$, que confirmaría lo que se ha detectado a través de los estadísticos descriptivos expuesto en el primer objetivo específico, por tanto, ser valuado el aspecto de funcionalidad como calidad media por los especialistas, indica que tiene que mejorarse en lo que corresponde a madurez en los ítems de ¿Existe un responsable de la administración de la solicitud de cambio?, y ¿Existe hitos definidos de actualización del plan de proyecto?; en calidad de servicio se debe mejorar en los ítems ¿Responde a una plantilla predefinida?, ¿Se documenta la solicitud de cambio de requerimientos?, ¿Se generan versionados ante la administración de las solicitudes de cambio?, ¿Se documentan adecuadamente las versiones definidas en la pregunta 9?, y ¿Se utiliza alguna herramienta automatizada para documentar las versiones definidas en la pregunta 9?; asimismo, en lo que corresponde a calidad percibida se debe mejorar en el ítem de ¿Una solicitud de cambio motiva la actualización del plan del proyecto?; en fiabilidad solo se debe mejorar la regularidad de cumplimiento en una implementación continua de las acciones y procesos (Modalidad de siempre de la escala de calidad).

Aplicando la teoría de los diseños experimentales que el efecto de las variables independientes se mide a través de la variable dependiente; *permite realizar la inferencia que, los desarrolladores del software implementaron algunos estándares NTP 12207:2016, ISO*

9001 e ISO 9126, que determinan características de calidad media en el aspecto de funcionalidad de software con tendencia a una calidad alta.

Segunda hipótesis específica.

HE2: La implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la usabilidad del desarrollo de software en la UNSCH.

Tabla 18 Estadísticos de la Usabilidad

Variable	N	Media	Desv. Desviación	Desv. Error promedio
Usabilidad	6	43,33	5,922	2,418

Tabla 19

Prueba de hipótesis para puntuac

Valor de prueba = 59,5						
Variable	t	gl	Sig. (bilateral)	Diferencia de medias	95% de intervalo de confianza de la diferencia	
					Inferior	Superior
Usabilidad	-6,687	5	,001	-16,167	-22,38	-9,95

Formulación de Hipótesis

H₀: La puntuación media de la dimensión de usabilidad es igual a 59,5.

H₁: La puntuación media de la dimensión de usabilidad es diferente a 59,5.

En forma simbólica se tiene:

$$H_0: \mu = 59,5 \quad \& \quad H_1: \mu \neq 59,5$$

Nivel de significancia

El nivel de significancia teórica es $\alpha = 0,05$.

Regla de decisión

Si el $p > \alpha$, Aceptar H_0 .

Si el $p < \alpha$, Rechazar H_0 .

Estadístico de contraste

El estadístico de contraste paramétrico T-Student apropiado para estudiar muestras pequeñas, cuyos datos tienen comportamiento normal.

Interpretación

En la tabla 18 y tabla 19 se observa los estadísticos de resumen para calcular el

estadístico $t = \frac{s}{\sqrt{n}} = \frac{5,922}{\sqrt{6}} = -6,687$ $\bar{x} - \mu$ $43,33 - 59,5$ y su p – valor = $0,001 < \alpha = 0,05$, que

determinan que se rechaza la hipótesis nula H_0 y se acepta la hipótesis alterna H_1 que afirma que la media poblacional de usabilidad es $\mu \neq 59,5$ con un 95% de confianza y 5% de significancia, pero existe una diferencia aritmética de $\bar{x} - \mu = -16,167$ que expresa que la media muestral de 43,33 está por debajo de la media poblacional y está próxima a la marca de clase de 42,5 del intervalo que denota calidad media: $[34 - 51>$, este comportamiento de la puntuación media muestral confirmaría lo que se ha detectado a través de los estadísticos descriptivos expuesto en el segundo objetivo específico; por consiguiente, el hecho de ser valorado los criterios de usabilidad del software de la UNSCH con un atributo de calidad media por los especialistas, implica que tiene que mejorarse algunos aspectos en lo concerniente a aprender en el ítem de: Si se usan cadenas de caracteres, ¿se asigna explícitamente un delimitador?; en operar en los ítems de: ¿Todas las constantes tienen nombre?; ¿Existe alguna posibilidad de desbordamiento de buffer?; ¿Hay certeza de que

termine cada ciclo?; Si después de cada caso en los enunciados se requiere un paréntesis, ¿éste se incluyó?; ¿Todas las llamadas a función y método tienen el número correcto de parámetros?; ¿Todas las llamadas a función y método tienen el número correcto de parámetros?; en el aspecto de preparar las entradas e interpretar las salidas de un programa, se debe mejorar en los ítems de: ¿Se usan todas las variables de entrada?, y ¿Entradas inesperadas pueden causar corrupción?.

Considerando el mismo orden de ideas del contraste anterior, se inferencia que, *los desarrolladores del software implementaron algunos estándares NTP 12207:2016, ISO 9001 e ISO 9126, que determinan características de calidad media en el aspecto de usabilidad de software.*

Tercera hipótesis específica.

HE3: La implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la eficiencia del desarrollo de software en la UNSCH.

Tabla 20 Estadísticos de la eficiencia

Variable	N	Media	Desv. Desviación	Desv. Error promedio
Eficiencia	6	8,00	,632	,258

Tabla 21

Prueba de hipótesis par

ción media de la eficiencia

Variable	Valor de prueba = 10,5				
	t	gl	Sig. (bilateral)	Diferencia de medias	95% de intervalo de confianza de la diferencia

					Inferior	Superior
Eficiencia	-9,682	5	,000	-2,500	-3,16	-1,84

Formulación de Hipótesis

H₀: La puntuación media de la dimensión de eficiencia es igual a 10,5.

H₁: La puntuación media de la dimensión de eficiencia es diferente a 10,5.

En forma simbólica se tiene:

$$H_0: \mu = 10,5 \quad \& \quad H_1: \mu \neq 10,5$$

Nivel de significancia

El nivel de significancia teórica es $\alpha = 0,05$.

Regla de decisión

Si el $p > \alpha$, Aceptar H₀.

Si el $p < \alpha$, Rechazar H₀.

Estadístico de contraste

El estadístico de contraste paramétrico T-Student apropiado para estudiar muestras pequeñas, cuyos datos tienen comportamiento normal.

Interpretación

En la tabla 20 y tabla 21 se presentan los estadísticos de resumen para calcular el

estadístico $t = \frac{s}{\sqrt{n}} = \frac{0,632}{\sqrt{6}} = -9,682$ $\frac{\bar{x}-\mu}{\sqrt{s}}$ $8-10,5$ y su p – valor = $0,000 < \alpha = 0,05$, que determinan

que se rechaza la hipótesis nula H₀ y se acepta la hipótesis alterna H₁ que afirma que la media poblacional de eficiencia es $\mu \neq 10,5$ con un 95% de confianza y 5% de significancia, pero existe una diferencia aritmética de $\bar{x} - \mu = -2,5$ que expresa que la media muestral de 8,00 está por debajo de la media poblacional de eficiencia y está próxima a la marca de clase de

7,5 del intervalo que denota que la eficiencia es de calidad media: $[6 - 9>$, este comportamiento de la puntuación media muestral confirmaría lo que se ha detectado a través de los estadísticos descriptivos expuesto en el tercer objetivo específico; por consiguiente, se debe mejorar en lo referente a recursos de cómputo y códigos requeridos por un programa.

Análogamente a los contrastes anteriores, se inferencia que, *los desarrolladores del software implementaron algunos estándares NTP 12207:2016, ISO 9001 e ISO 9126, que determinan características de calidad media en el aspecto de eficiencia de software.*

Hipótesis General

HG: La implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora el desarrollo de software en la UNSCH.

Tabla 22 Estadísticos de la calidad de software

Variable	N	Media	Desv. Desviación	Desv. Error promedio
Calidad_de_Software	6	96,67	16,108	6,576

Tabla 23 Prueba de hipótesis para la puntuación media de la calidad de software

Variable t	Valor de prueba = 129,5				
	gl	Sig. (bilateral)	Diferencia de medias	95% de intervalo de confianza de la diferencia	
				Inferior	Superior
Calidad_de_Software -4,993	5	,004	-32,833	-49,74	-15,93

Formulación de Hipótesis

H₀: La puntuación media de la Calidad de software es igual a 129,5.

H₁: La puntuación media de la Calidad de software es diferente a 129,5.

En forma simbólica se tiene:

$$H_0: \mu = 129,5 \quad \& \quad H_1: \mu \neq 129,5$$

Nivel de significancia

El nivel de significancia teórica es $\alpha = 0,05$.

Regla de decisión

Si el $p > \alpha$, Aceptar H₀.

Si el $p < \alpha$, Rechazar H₀.

Estadístico de contraste

El estadístico de contraste paramétrico T-Student apropiado para estudiar muestras pequeñas, cuyos datos tienen comportamiento normal.

Interpretación

En la tabla 22 y tabla 23 se presentan los estadísticos de resumen para calcular el

estadístico $t = \frac{s}{\sqrt{n}} = \frac{16,108}{\sqrt{6}} = -4,993$ $\bar{x} - \mu$ $96,67 - 129,5$ y su p – valor = $0,004 < \alpha = 0,05$, que

determinan que se rechaza la hipótesis nula H₀ y se acepta la hipótesis alterna H₁ que afirma que la media poblacional de la calidad de software es $\mu \neq 129,5$ con un 95% de confianza y 5% de significancia, pero existe una diferencia aritmética de $\bar{x} - \mu = -32,833$ que expresa que la media muestral de 96,67 está por debajo de la media poblacional de Calidad alta de software y está próxima a la marca de clase de 92,5 del intervalo que denota que la calidad de Software es de calidad media: [111 – 148>, este comportamiento de la puntuación media muestral confirmaría lo que se ha detectado a través de los estadísticos descriptivos expuesto a través de los objetivo específicos e inferir que, *los desarrolladores del software*

implementaron algunos estándares NTP 12207:2016, ISO 9001 e ISO 9126, que determinan características de calidad media en el aspecto de Calidad de software.

4.2 Análisis de resultados

Teniendo como referencia los resultados obtenidos, se determina con respecto al **primer objetivo específico**, que la funcionalidad del desarrollo de software en la UNSCH se encuentra en un nivel de madurez intermedio debido a que se tiene un responsable de la gestión de proyectos, que conoce su función de definir los procesos, documentarlos, realizar entrenamiento, controlar, verificar, medir y formular la mejora, pero que sus actividades realizada periódicamente fueron valoradas por los expertos en lo que corresponde a gestión de proyectos y definición de actividades, están con mayor porcentaje en la escala de *a veces y frecuentemente*, pero sin porcentaje para la escala de *siempre* (ver ítem 1 y 3 de Tabla 5), en tanto que, la tendencia de los porcentajes para el responsable de la administración de la solicitud de cambios y para la definición de hitos de actualización del plan de proyecto, están en la escala de *nunca y pocas veces* (ver ítem 10 y 13 de Tabla 5). Estos indicadores permiten inferir que hay algunas actividades del nivel de madurez que se realizan medianamente, pero que las otras se realizan esporádicamente o cuando es urgente. Este comportamiento para que sea valorado con el atributo de calidad tiene que cumplirse en la escala de *siempre* (Ver Tabla 5 y Figura 18).

En lo correspondiente a fiabilidad, se determina que se cumple a nivel medio con cierta tendencia a mejorar, en lo que corresponde, a proyectos que cuentan con un documento donde se definen sus objetivos y alcances, que probablemente se revisen cuando es necesario para no perder el horizonte de la implementación, además, en la dinámica del trabajo se conoce cuándo se da la petición de cambios de procesos en el proyecto y quien es el responsable de la solicitud de cambio; dado que, los criterios aplicados para evaluar la fiabilidad fueron valorados por los expertos con mayor porcentaje en las escalas de *a veces y*

frecuentemente, pero con porcentaje bajo en la escala de *siempre* (Ver ítem 2, 6 y 11 de Tabla 6 y figura 19). En este aspecto de fiabilidad se tiene un propósito claro, pero falta concretarlo, para minimizar los errores y optimizar los recursos disponibles.

En el aspecto de calidad de servicio, en la mayoría de criterios de evaluación aplicados se determina que se cumplen en un nivel bajo con tendencia a un nivel medio, dado que, los especialistas valoraron con mayor porcentaje en la escala de *nunca, pocas veces y a veces* (ver ítem 4, 7, 14, 15 y 16 de Tabla 7 y figura 20). Estos indicadores revelan que se debe mejorar en calidad de servicio en forma urgente para administrar apropiadamente las solicitudes de cambio, para que se cumplan ciertos protocolos previamente definidos, se cumplan los requerimientos, se generen versiones actualizadas de acuerdo a los estándares NTP 12207:2016, ISO 9001 e ISO 9126 y se automatice la documentación de las versiones.

En cuanto a la calidad percibida, se identifica en función a los criterios de evaluación aplicados, que son valorados en un nivel medio con cierta tendencia a la mejora, dado que, los porcentajes están concentrados en la escala de *a veces y siempre* (Ver ítem 5, 8, 9, 12 y 33 de Tabla 8 y figura 21). Estos indicadores muestran que se viene haciendo bien el procedimiento de indagar el origen del cambio planteado en la solicitud e informar a los actores involucrados, pero en los demás indicadores se viene haciendo a un nivel intermedio, en lo correspondiente a los proyectos, quedan documentados en un almacenamiento estable que guarda en un disco magnético, se tiene acceso a los componentes de memoria compartida, y tienen el mismo modelo de estructura de memoria compartida; excepto el criterio que corresponde a la solicitud de cambio, que motiva la actualización del plan del proyecto, que fue calificado en la escala de *nunca y pocas veces*.

En términos de la puntuación total para la funcionalidad del desarrollo de software se obtuvo una media de 45,33 que está contenida en el intervalo de *calidad media o regular* cuyo rango de puntuación varía de 34 a 51 puntos y para la prueba de hipótesis para la media poblacional de la funcionalidad se tomó como parámetro el punto medio de 59.5 (promedio

poblacional) del intervalo de calidad alta que varía de 51 a 68 puntos para efectos de comparación; los resultados de dicha prueba T – Student determinaron que existe diferencia estadística significativa entre el promedio muestral con el promedio poblacional con un 95% de confianza, esto quiere decir, que el patrón de comportamiento de las puntuaciones de la funcionalidad están más concentradas en el intervalo que corresponde a calidad media, que para llegar a posicionarse en el intervalo de calidad alta, la puntuación media de la funcionalidad actual tendría que incrementarse en 14,167 puntos (Ver Tabla 16 y 17).

El análisis articulado de los indicadores de la funcionalidad de desarrollo de software descritos, permiten inferir que los responsables de la unidad de informática de la UNSCH vienen cumpliendo los requerimientos a un nivel intermedio, debido a que están en una fase inicial del proceso de automatización de las actividades académicas y administrativas que requieren programar los algoritmos que realicen las funciones, pero aún no se tiene un conocimiento pleno de la complejidad de procesos que tiene la universidad, pero tiene una funcionalidad aceptable para uso normal, además, se diagnóstica que a medida que los programadores tengan en cuenta los estándares NTP 12207:2016, ISO 9001 e ISO 9126 se insertaran en el proceso de la mejora continua. Estos atributos de la funcionalidad del desarrollo de software tienen estrecha relación con lo que expresa (Sommerville, 2016), que cuando la funcionalidad no proporciona los resultados esperados, los usuarios encontrarán otras maneras de hacerlo; asimismo, los atributos de calidad de la funcionalidad en su vida útil inicial del software en el presente estudio son correspondientes con la investigación de Fuertes (2002) debido a que plantea un modelo de calidad completo del software que se descompone en una serie de factores y éstos, a su vez, se descomponen en un conjunto de criterios medibles utilizando medidas.

Los indicadores del **segundo objetivo específico**, en lo que corresponde a usabilidad en la dimensión de aprender se tiene una valoración media con tendencia a la mejora, puesto que, los especialistas ponderaron en este nivel el uso de cadenas de caracteres y en el proceso de asignación en forma explícita de un delimitador, así como, para cada enunciado

condicional evaluaron que *a veces o frecuentemente* esta condición es correcta, y que a veces los tipos de parámetro formal y real coinciden (Ver ítem 20, 22 y 31 de Tabla 9 y figura 22).

En el aspecto de operar, los expertos valoraron la actividad realizada con porcentajes significativos en la escala de *a veces* seguidas en importancia de las modalidades de *nunca* y *pocas veces*, estos indicadores permiten determinar que la mayoría de criterios de operar se realizan a nivel intermedio, pero otras tantas, que están por debajo de la actividad media, tal es así que, manejan a nivel medio la declaración de las variables del programa y constantes, la cota superior de los arreglos es igual al tamaño del arreglo o valor - 1, cabe la posibilidad de desbordamiento de buffer; la certeza que termine cada ciclo; los enunciados compuestos están correctamente colocados entre paréntesis; los enunciados se justifican en todos los casos posibles y los parámetros están en el orden correcto. En un nivel de nunca o pocas veces, se manejan los criterios: Si después de cada caso en los enunciados se requiere un paréntesis, nunca o pocas veces se incluyó, análogamente se tiene, a todas las llamadas a función y método tienen el número correcto de parámetros (Ver ítem 17, 18, 19, 21, 23, 24, 25, 26, 30 y 32 de Tabla 10 y figura 23). El análisis de estos indicadores en conjunto permite valorar que los criterios de evaluación de la capacidad de operar determinan que la sintaxis de la programación en el desarrollo de software se viene manejando a nivel intermedio, puesto que algunos criterios están formulados en sentido positivo y otros en sentido negativo tal como nunca existe la posibilidad de desbordamiento de buffer.

En el aspecto de preparar las entradas e interpretar las salidas de un programa, los expertos valoraron los criterios en un nivel medio con tendencia por debajo de la media, pues algunas veces a todas las variables de salida se les asigna un valor antes de que se produzcan los resultados y alguna vez se modifica una estructura vinculada; en tanto que, nunca o pocas veces se usan todas las variables de entrada y las entradas inesperadas pueden causar corrupción (ver ítem 27, 28, 29 y 34 de Tabla 11 y figura 24). Las puntuaciones asignadas por

los expertos oscilan alrededor de la puntuación media, lo que indica que los resultados del software instalado o los reportes, son interpretados según la información que se asigna a las variables de entrada, pues los encargados de alimentar al software con datos no realicen el control de calidad del dato y se produzca en algunas ocasiones error en el procesamiento o produzca un resultado incorrecto o reporte incorrecto o error en la programación, por tanto, los datos o la información que ingresa al software debe pasar por un control de calidad para obtener los resultados esperados.

Analizando el contraste de hipótesis del segundo objetivo específico la dimensión de usabilidad tiene una puntuación media de 43.33 que pertenece al intervalo de calidad media que varía de 34 a 51 puntos, para efectos de comparar se toma como parámetro la puntuación promedio del intervalo de 59.5 del intervalo de calidad alta que varía de 51 a 68 puntos, bajo estas premisas de la estadística, los resultados la prueba T – Student establecieron que existe diferencia estadística significativa entre el promedio muestral con el promedio poblacional con un 95% de confianza, es decir, que el comportamiento de las puntuaciones de la usabilidad están más centradas en el intervalo que corresponde a calidad media, en el caso de una mejora, la usabilidad en forma general para llegar a posicionarse en el intervalo de calidad alta, la puntuación media actual tendría que incrementarse en 16,167 puntos (Ver Tabla 18 y 19).

Las premisas descritas, permiten deducir de los indicadores de la usabilidad del software, que lo valoran a nivel intermedio debido a que probablemente observaron que los usuarios de la unidad de informática de la UNSCH tienen algunas dificultades para ingresar los datos a las variables definidas en el software ocasionando algunos errores en los resultados o en los reportes o en algunos aspectos la programación que necesita algún ajuste en la sintaxis de programación, dado que, no están acorde a los estándares NTP 12207:2016, ISO 9001 e ISO 9126; los atributos descritos de usabilidad en lo que corresponde a aprender, operar y preparar las entradas e interpretar salidas, permiten articularlos con el desarrollo continuo propuesto en el estudio de Abuhav (2017), dado que, la calidad media hallada en esta dimensión, supone un reto de

planificar la posibilidad de implementar la mejora continua, porque beneficia a los usuarios finales del software en cuestión, pues los estudiantes, docentes y administrativos, que tendrán instrumentos e insumos que podrán manejarlos de forma amigable con el software.

Los estadísticos estimados del **tercer objetivo específico**, que corresponde a la eficiencia, los especialistas lo valoraron en un nivel medio con tendencia a la mejora, en el aspecto de recurso de cómputo consideran que a veces o frecuentemente se usa el almacenamiento dinámico y que se asignó el espacio o en la bandeja correcta, asimismo, el espacio se cancela en forma explícita después de que la información ya no se requiere (Ver ítem 35 y 36 de Tabla 12 y figura 25).

En el aspecto de códigos requeridos por un programa, los especialistas valoraron con un nivel medio o por debajo del nivel medio, pues consideran que pocas veces o a veces se tomaron en cuenta todas las posibles condiciones de error; probablemente estas condiciones de error no fueron previstas en su totalidad y a medida que el software madure se realizarán las pruebas de verificación y validación para que cumpla las necesidades de la universidad (Ver ítem 37 de Tabla 13 y figura 26).

Las premisas descritas en cuanto a eficiencia del desarrollo de software, están en un nivel medio, con ciertas situaciones que requieren ajuste para tener un funcionamiento de acuerdo a las necesidades de la universidad que debido que sus procesos son complejos se ha abordado solo algunas áreas de gestión importantes y a medida que validen los procesos instalados abarcaran los otros que están pendientes programación.

El análisis de los estadísticos de la prueba de hipótesis para la eficiencia, tiene una puntuación media de 8 que está contenido en el intervalo de calidad media que varía de 6 a 9 puntos y se tomó como parámetro de contraste el punto medio de 10.5 del intervalo de calidad alta que varía de 9 a 12 puntos, la Prueba T- Student evaluada con un 95% de confianza determina que existe diferencia estadística significativa, pero que, la puntuación

actual de la eficiencia igual al parámetro de calidad alta debe incrementarse en 2.5 puntos (Ver Tabla 20 y 21). Probablemente estos resultados de la eficiencia se encuentren relacionados con el proceso de madurez del desarrollo de software porque tiene pocos años de su instalación y puesto en marcha, teniendo en el inicio que priorizar los procesos más importantes de la UNSCH y en el tiempo seguramente tendrá que tener implementado en 100% de los procesos de la institución; estos resultados están encuadrados dentro de los modelos de madurez de capacidades de Espejo Chavarría, A. (2016), dado que, a medida que madura el desarrollo de software y el dominio de las bondades por los usuarios se logrará abarcar la mayoría de procesos de la UNSCH.

Finalmente, en términos de la puntuación total que asignaron los especialistas de Unidad de Informática de la UNSCH a los 37 criterios de evaluación que incorporan los estándares NTP 12207:2016, ISO 9001 e ISO 9126 para determinar la calidad de software, se obtuvo una puntuación promedio de 96.67 (ver tabla 22) que está contenido en el intervalo de calidad media cuya puntuación varía de 74 a 111 puntos y para efectos de comparación se tomó como parámetro poblacional el punto medio de 129.5 del intervalo de calidad alta, la prueba de hipótesis con el T – Student evaluado con un 95% de confianza determina que existe diferencia estadística significativa, por tanto, para que la puntuación de calidad media del software instalado en la UNSCH se posicione en el intervalo de calidad alta debe incrementarse en 32.833 puntos. Estos resultados globales confirman lo ya descrito mediante los estadísticos de resumen y las pruebas de hipótesis implementadas, por tanto, se infiere que

La implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora el desarrollo de software en la UNSCH.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

El proceso de aseguramiento de calidad de software permite mejorar la etapa de desarrollo, la etapa de análisis y diseño. Esto nos permite establecer los siguientes hallazgos:

1. El modelo conceptual definido por los estándares que aportan las normas 12207:2016, ISO 9001 e ISO 9126, ha permitido identificar con un 95% de confianza y 5% de significancia, que en la dimensión de funcionalidad los desarrolladores de software lograron características de calidad media en madurez y calidad de servicio, con cierta tendencia en la mayoría de ítems a una calidad alta en fiabilidad y calidad percibida ($t = -2.913$; $p = 0.033 < \alpha = 0,05$, $\bar{x} - \mu = -14,17$).
2. El modelo conceptual definido por los estándares que aportan las normas 12207:2016, ISO 9001 e ISO 9126, ha permitido identificar con un 95% de confianza y 5% de significancia, que en la dimensión de usabilidad los desarrolladores de software lograron características de calidad media que articula el aprender, operar y preparar las entradas e interpretar las salidas del programa ($t = -6,687$; $p = 0.001 < \alpha = 0,05$, $\bar{x} - \mu = -16,167$).
3. El modelo conceptual definido por los estándares que aportan las normas 12207:2016, ISO 9001 e ISO 9126, ha permitido identificar con un 95% de confianza y 5% de significancia, que en la dimensión de eficiencia los desarrolladores de software lograron características de calidad media que articula los recursos de cómputo y los códigos requeridos por un programa ($t = -9,682$; $p = 0.000 < \alpha = 0,05$, $\bar{x} - \mu = -2,5$).
4. El modelo conceptual definido por los estándares que aportan las normas 12207:2016, ISO 9001 e ISO 9126, ha permitido identificar con un 95% de confianza y 5% de significancia, que en los atributos de calidad los desarrolladores de software lograron

características de calidad media que comprende la funcionalidad, usabilidad y eficiencia ($t = -4,993$; $p =$

$0.004 < \alpha = 0,05$, $\bar{x} - \mu = -32,833$).

RECOMENDACIONES

El conocer los resultados de la calidad de software usado por la UNSCH, permite formular las siguientes recomendaciones:

1. Formular propósitos a lograr en corto plazo en los aspectos de madurez y calidad de servicio de la dimensión de funcionalidad, la implementación de protocolos que permitan en el tiempo observar los atributos deseados de calidad, luego ser valorados por los especialistas con una regularidad que se realizan frecuentemente o siempre.
2. Establecer objetivos a corto plazo para mejorar los aspectos de usabilidad de software, para que luego sean valorados con una regularidad de frecuentemente o siempre, en un proceso de calidad total.
3. Proponer objetivos a corto plazo para mejorar la eficiencia de software para contar con recursos de cómputo y códigos requeridos por un programa confiables para rescatar la trazabilidad de hechos que son importantes en un proceso de mejora continua.
4. Elaborar un diagnóstico general la calidad de software que sirva de referencia para implementar las mejoras de acuerdo a metas alcanzables.

REFERENCIAS

- Abreu, J. (2012). *Hipótesis, método & diseño de investigación (hypothesis, method & research design)*. *Daena: International Journal of Good Conscience*, 7(2), 187-197.
- Abuhav, I. (2017). *ISO 9001:2015 A Complete Guide to Quality Management System*. 6000 Broken Sound Parkway NW, Suite 300 Boca Raton: Taylor & Francis Group, LLC.

- Anguita, J. C. (2003). *La encuesta como técnica de investigación. Elaboración de cuestionarios y tratamiento estadístico de los datos (I). Atención primaria, 31(*
- Arias, F. G. (2012). *El proyecto de investigación. Introducción a la metodología científica. 6ta. Fidiás G. Arias Odón.*
- Barrero, G. (2018). *Sistema Qsource en la calidad del software desarrollado en RPG. (Tesis de maestría). Universidad César Vallejo, Lima.*
- Baudean, M. (2015). *Introducción a la investigación aplicada.* Uruguay: Universidad ORT-FACS.
- Bauer, S. N., & Turco, C. (2011). *Aprender en la Universidad: La formación del estudiante en comprensión y producción académica: entre el conocimiento y el saber hacer.* FACE.
- Carrasco, S. (2014). *Metodología de la investigación científica.* Lima: San Marcos E.I.R.L.
- Castán, Y. (2014). *Introducción al método científico y sus etapas. Metodología en Salud Pública España, 6(3).*
- Castellanos Sánchez, A. S. (2017). *Nuevos modelos tecnopedagógicos. Competencia digital de los alumnos universitarios. Revista electrónica de investigación educativa, 19(1), 1-9.*
- Chaudhary, M., & Chopra, A. (2017). *CMMI for Development Implementation Guide.* Faridabad, Haryana, India: Edición Kindle.
- De Paz, D. C. (2008). *Conceptos y técnicas de recolección de datos en la investigación jurídico social. Recuperado de: <http://www.geocities.ws/jusbaniz/faseI/tesis/tecnicas1.pdf>.*
- Dirección de Normalización - INACAL. (2016). *Norma Técnica Peruana NTP-ISO/IEC 12207:2016.* Lima: Inacal.

- Espejo, A. J. (2016). Modelo de aseguramiento de la calidad en el proceso de desarrollo de software basado en los modelos de madurez de capacidades (CMMi), proceso de software para equipos (TSP) y personas (PSP). (*Tesis de Maestría*). Universidad Nacional Mayor de San Marcos, Lima.
- Fuertes, J. L. (2002). Modelo de calidad para el software orientado a objetos. (*Tesis doctoral*). Universidad Politécnica de Madrid, Madrid.
- Garrote, P. R. (2015). *La validación por juicio de expertos: dos investigaciones cualitativas en Lingüística aplicada. Revista Nebrija de lingüística aplicada a la enseñanza de lenguas, (18), 124-139.*
- GlobalSuite Solutions. (5 de Marzo de 2020). Obtenido de <https://www.globalsuitesolutions.com/es/que-son-normasiso/#:~:text=Las%20normas%20ISO%20son%20un,de%20productos%20en%20la%20industria>
- Goetsch, D. L., & Davis, S. B. (2016). *Quality Management for organizational excellence introduction to total Quality*. United States of America: Pearson Education, Inc. or its affiliates.
- Hernández Sampieri, R. F. (2010). *Metodología de la investigación (5° Ed.)*. México, D.F., México: McGraw Hill Interamericana.
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. (2014). *Metodología de la Investigación*. México: McGraw-Hill / Interamericana Editores, S.A. de C.V.
- Hernández, R., & Mendoza, C. P. (2018). *Metodología de la Investigación: Las Rutas Cuantitativa, Cualitativa y Mixta*. México: McGraw-Hill Interamericana Editores, S.A. de C. V.
- Humphrey, W. S. (1989). *Managing the Software Process*. Pittsburgh: Addison-Wesley

Professional.

Kaplan, A., & Wolf, J. C. (1964). *The Conduct of Inquiry Methodology for Behavioral Science*. New York: Routledge.

Kerlinger, F. N., & Lee, H. B. (2002). *Investigación del comportamiento*. México: McGraw-Hill.

Laza, C. A. (2019). *Investigación y recogida de información de mercados. UF1780. Tutor Formacion*.

Lopera Echavarría, J. D. (2010). *El método analítico como método natural*.

López, N. &. (2016). *Métodos y técnicas de investigación cuantitativa y cualitativa*.

Lozada, J. (2014). *Investigación aplicada: Definición, propiedad intelectual e industria. CienciAmérica: Revista de divulgación científica de la Universidad Tecnológica Indoamérica, 3(1), 47-50*.

Meneses, J. &.-G. (2011). *El cuestionario y la entrevista*.

Mitra, A. (2021). *Fundamentals of Quality Control and Improvement*. John Wiley & Sons, Inc.

Moreno, F. A. (2020). *Modelo de gestión de calidad basada en los estándares NTP 12207, ISO 9001 E ISO 9126, para los procesos de desarrollo de software: caso RENIEC. (Tesis de maestría)*. Universidad Nacional Mayor de San Marcos, Lima.

Oncins de Frutos, M. (1991). *NTP 283: Encuestas: Metodología para su utilización. Nota Técnica de Prevención, Madrid: Instituto Nacional de Seguridad e Higiene en el Trabajo (INSHT)*.

Pantaleo, G. (2011). *Calidad en el desarrollo de software*. Argentina: Alfaomega.

Portela, S. (2016). *El Liderazgo Transformador en la Gestión de la Calidad. Un Estudio Basado en el Modelo EFQM. (Tesis doctoral)*. Universidad de Alicante, España.

Ramírez, A., González, L. J., & Vázquez, P. J. (2003). *Caracterización y diseño de los resultados científicos como aportes de la investigación educativa. EcuRed(40)*.

- Saavedra Benavides, S. J. (2021). *Motivación laboral y Engagement de los colaboradores de la Municipalidad Distrital de Llacanora, Cajamarca 2021*.
- Sabino, C. (1992). *El Proceso de Investigación*. Caracas: Editorial Panapo.
- Sampieri, R. F. (2014). *Definiciones de los enfoques cuantitativo y cualitativo, sus similitudes y diferencias*. RH Sampieri, *Metodología de la Investigación*.
- Sánchez, D. G. (2010). *El compromiso y clima organizacional en la empresa familiar de Rioverde y del Refugio Ciudad Fernández*. *Administración & Desarrollo*, 38(52), 97105.
- Sommerville, I. (2016). *Software Engineering*. Estados Unidos: Pearson Education Limited.
- Tarí, J. J. (2000). *Calidad total: Fuente de ventaja competitiva*. Publicaciones Universidad de Alicante.
- Tuapanta Dacto, J. V. (2017). *Alfa de Cronbach para validar un cuestionario de uso de TIC en Docentes Universitarios*.
- Ulin, P. R. (2005). *Investigación aplicada en salud pública: métodos cualitativos*. Organización Panamericana de la Salud.
- Vivanco Mena, Z. M. (2017). *Gestión de cuentas por cobrar en la Empresa Melkarth Logistic Network SAC, Pueblo Libre-2017*.
- Zamorano, C. O. (2013). *Medición de la satisfacción estudiantil universitaria: Un estudio de caso en una institución mexicana*. *Iberoamerican Journal of Industrial Engineering*, 5(9), 261-274.

ANEXOS**ANEXOS A: Declaración de autenticidad****Escuela de Posgrado****Universidad
Ricardo Palma****DECLARACIÓN DE ATENTICIDAD Y NO PLAGIO****DECLARACIÓN DEL GRADUANDO**

Por el presente el graduando (Apellidos y nombres):

Peralta Sotomayor, Karel

, en condición de egresado del Programa de Posgrado:

Ingeniería Informática con Mención en Ingeniería de Software

deja constancia que ha elaborado el proyecto de tesis titulado:

Características de la calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126, para la mejora del área de desarrollo de software en la UNSCH. Ayacucho, 2021

Declara que el presente trabajo de tesis ha sido integralmente elaborado por el mismo y que no existe plagio de ninguna naturaleza, en especial copia de otro documento de investigación (tesis, revista, texto, congreso, o similar) presentado por cualquier persona natural o jurídica ante cualquier institución académica, de investigación, profesional o similar.

Deja constancia que las citas de otros autores han sido debidamente identificadas en el trabajo de investigación, por lo que no ha asumido como suyas las opiniones vertidas por terceros, ya sea de fuentes encontradas en medios escritos, digitales o de la Internet.

Asimismo, ratifica que es plenamente consciente de todo el contenido de la tesis y asume la responsabilidad de cualquier error u omisión en el documento y es consciente de las connotaciones éticas y legales involucradas.

En caso de incumplimiento de esta declaración, el graduando se somete a lo dispuesto en las normas de la Universidad Ricardo Palma y los dispositivos legales vigentes.

13 de febrero de 2022

ANEXOS

Fecha

Firma del graduando

ANEXOS

B: Autorización de consentimiento para realizar la investigación

Escuela de Posgrado

AUTORIZACIÓN PARA REALIZAR LA INVESTIGACIÓN

**DECLARACIÓN DEL RESPONSABLE DEL ÁREA O DEPENDENCIA
DONDE SE REALIZARÁ LA INVESTIGACIÓN**

Dejo constancia que el área o dependencia que dirijo, ha tomado conocimiento del proyecto de tesis titulado:

Características de la calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126, para la mejora del área de desarrollo de software en la UNSCH. Ayacucho, 2021

el mismo que es realizado por el Sr. Estudiante (Apellidos y nombres):

Peralta Sotomayor, Karel

, en condición de estudiante - investigador del Programa de:

Maestría en Ingeniería Informática con mención en Ingeniería de Software

Así mismo señalamos, que según nuestra normativa interna procederemos con el apoyo al desarrollo del proyecto de investigación, dando las facilidades del caso para aplicación de los instrumentos de recolección de datos.

En razón de lo expresado doy mi consentimiento para el uso de la información y/o la aplicación de los instrumentos de recolección de datos:

Apellidos y nombres del jefe del área o dependencia


 UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
 Ing. Hilda Chacampa Patilla
 ÁREA DE TECNOLOGÍAS DE INFORMACIÓN
 C.A. 122145

Fecha: 15 de enero de 2022

ANEXOS

C: Matriz de consistencia

TITULO: Características de la calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126, para la mejora del área de desarrollo de software en la UNSCH. Ayacucho 2021.

FORMULACIÓN DEL PROBLEMA	OBJETIVOS	HIPÓTESIS	METODOLOGÍA
<p>Problema General ¿De qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora el desarrollo de software en la UNSCH?</p>	<p>Objetivo General Determinar de qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora el desarrollo de software en la UNSCH.</p>	<p>Hipótesis General La implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora el desarrollo de software en la UNSCH.</p>	<p>TIPO DE INVESTIGACIÓN: Aplicada</p> <p>NIVEL DE INVESTIGACIÓN Explicativo.</p> <p>DISEÑO DE INVESTIGACIÓN: Pre-experimental, transversal.</p> <p>POBLACIÓN: El área de desarrollo de software en la UNSCH.</p> <p>MUESTRA:</p>
<p>Problemas Específicos PE1. ¿De qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la funcionalidad del desarrollo de software en la UNSCH?</p>	<p>Objetivos Específicos OE1. Determinar de qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la funcionalidad del desarrollo de software en la UNSCH.</p>	<p>Hipótesis Específicas HE1. La implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la funcionalidad del desarrollo de software en la UNSCH.</p>	
<p>PE2. ¿De qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la usabilidad del desarrollo de software en la UNSCH?</p>	<p>OE2. Determinar de qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la usabilidad del desarrollo de software en la UNSCH.</p>	<p>HE2. La implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la usabilidad del desarrollo de software en la UNSCH.</p>	

PE3. ¿De qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la eficiencia del desarrollo de software en la UNSCH?	OE3. Determinar de qué manera la implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la eficiencia del desarrollo de software en la UNSCH.	HE3. La implementación de las características de calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126 mejora la eficiencia del desarrollo de software en la UNSCH.	Se trabajará para el presente estudio con toda la población, muestra censal
---	---	--	---

D: Matriz de operacionalización

Características de la calidad de software basada en los estándares NTP 12207:2016, ISO 9001 e ISO 9126, para la mejora del área de desarrollo de software en la UNSCH. Ayacucho 2021.

VARIABLES	DEFINICION CONCEPTUAL	DEFINICION OPERACIONAL	DIMENSIONES	INDICADORES	ITEM/ESCALA	INSTRUMENTO
Estándares NTP 12207:2016, ISO 9001 e ISO 9126	“...Incluso los desarrolladores de software más experimentados estarán de acuerdo en que obtener software de alta calidad es una meta importante. Pero, ¿cómo se define la calidad del software? En el sentido más general se definen como: Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo	Las características de la calidad de software basadas en estándares para el desarrollo de software desempeño o rendimiento, características, fiabilidad, conformidad al diseño, durabilidad, calidad del servicio, estética y calidad percibida. Permitirá	Criterios de Funcionalidad	Niveles de madurez	1, 3,10,13,	Cuestionario
				Fiabilidad	2,6,11,	
				Calidad de servicio	4,7,14,15,16	
				Calidad percibida	5,8,9,12,33,	
			Criterios de Usabilidad	Aprender	20,22,31	
				Operar	17,18,19,21,23,24,25,26,30,32	

ANEXOS

	producen y a quienes lo utilizan...”	evaluar y determinar los aspectos y criterios de medición individuales.		Preparar las entradas e interpretar las salidas de un programa.	27,28,29,34	
--	--------------------------------------	---	--	---	-------------	--

VARIABLES	DEFINICION CONCEPTUAL	DEFINICION OPERACIONAL	DIMENSIONES	INDICADORES	ITEM/ESCALA	INSTRUMENTO
			Criterios de Eficiencia	Recursos de computo	35,36	
				Códigos requeridos por un programa	37	
CALIDAD DE SOFTWARE	Son atributos que reúne el software factible de ser medidos cuya valoración determina la calidad del intangible puesto en uso para satisfacer una necesidad del usuario		Valoración de Funcionalidad	Niveles de madurez	Escala de Likert: 1: Nunca 2: Pocas veces 3: A veces 4: Frecuentemente 5: Siempre	Calidad muy baja: [0 – 17> Calidad baja: [17 – 34> Calidad media: [34 – 51> Calidad alta: [51 – 68> Calidad muy alta: [68 – 85>
				Fiabilidad		
				Calidad de servicio		
				Calidad percibida		

		Valoración de Usabilidad	Aprender	Escala de Likert: 1: Nunca 2: Pocas veces 3: A veces 4: Frecuentemente 5: Siempre	Calidad muy baja: [0 – 17>
			Operar		Calidad baja: [17 – 34>
			Preparar las entradas e interpretar las		Calidad media: [34 – 51>
					Calidad alta: [51 – 68>
					Calidad muy alta: [68 – 85>

VARIABLES	DEFINICION CONCEPTUAL	DEFINICION OPERACIONAL	DIMENSIONES	INDICADORES	ITEM/ESCALA	INSTRUMENTO
				salidas de un programa.		
			Valoración de Eficiencia	Recursos de computo	Escala de Likert: 1: Nunca 2: Pocas veces 3: A veces 4: Frecuentemente 5: Siempre	Calidad muy baja: [0 – 3> Calidad baja: [3 – 6> Calidad media: [6 – 9> Calidad alta: [9 – 12> Calidad muy alta: [12 – 15>
				Códigos requeridos por un programa		

Pruebas de normalidad

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Calidad_de_Software	,204	6	,200*	,898	6	,360
Funcionalidad	,327	6	,043	,851	6	,161
Usabilidad	,314	6	,065	,878	6	,259
Eficiencia	,407	6	,002	,640	6	,001

*. Esto es un límite inferior de la significación verdadera.

a. Corrección de significación de Lilliefors

ANEXOS E: Protocolos o Instrumentos utilizados

Nombre	:	Estándares NTP 12207:2016, ISO 9001 e ISO 9126
Autor	:	Tesista
Año	:	2021
Procedencia	:	Ayacucho, Perú.
Objetivo	:	Evaluar el nivel de los estándares para el desarrollo de software.
Administración	:	Individual.
Duración	:	40 minutos
Aplicación	:	Trabajadores de la unidad informática de la Universidad Nacional de San Cristóbal de Huamanga.
Estructuración	:	La prueba tiene 3 factores: Valoración de Funcionalidad, Valoración de Usabilidad, Valoración de Eficiencia
Número de ítems	:	37 ítems
Escala	:	Likert del 1 a 5, donde 1: Nunca, 2: Pocas veces, 3: A veces, 4: Frecuentemente, 5: Siempre.

ANEXOS F: Formato de instrumentos o protocolos utilizados

Cuestionario de desarrollo de decisiones de implementación de software en la Universidad Nacional de San Cristóbal de Huamanga

Mi nombre es Karel Peralta Sotomayor, previo cordial saludo deseo que me regales unos minutos para responder algunas preguntas sobre el desarrollo de software en "La oficina de planeamiento y gestión de CTI" el área de desarrollo de software en la UNSCH al cual tu perteneces. Se busca mejorar los procesos de desarrollo de software.

Likert del 1 a 5, donde 1: Nunca, 2: Pocas veces, 3: A veces, 4: Frecuentemente, 5: Siempre.



kareelperalta@gmail.com (no compartidos) [Cambiar de cuenta](#)



1.- ¿Existe un responsable de la Gestión de Proyectos?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

1.- ¿Existe un responsable de la Gestión de Proyectos?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

2.- ¿Existe un Documento donde se definan objetivos y alcances del proyecto?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

3.- Si no existe el Director de Proyecto, ¿en que se basa para definir las actividades para cada proyecto?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

4.- ¿Responde a una plantilla predefinida?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

5.- ¿En que medio queda documentado (manual/digital)?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

6.- ¿Sabe usted cuando se da la petición de cambios de proceso en el proyecto?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

7.- ¿Se documenta la solicitud de cambio de requerimientos?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

8.- ¿Se determina el origen del cambio planteado en la solicitud?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

9.- ¿Cómo se informa de la solicitud de cambio a los actores involucrados?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

10.- ¿Existe un responsable de la administración de la solicitud de cambio?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

11.- ¿Quién es el responsable de la administración de la solicitud de cambio?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

12.- ¿Una solicitud de cambio motiva la actualización del plan del proyecto?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

13.- ¿Existe hitos definidos de actualización del plan de proyecto?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

14.- ¿Se generan versionados ante la administración de las solicitudes de cambio?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

15.- ¿Se documentan adecuadamente las versiones definidas en la pregunta 9?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

16.- ¿Se utiliza alguna herramienta automatizada para documentar las versiones definidas en la pregunta 9?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

17.- ¿Todas las variables del programa se inician antes de usar sus valores?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

18.- ¿Todas las constantes tienen nombre?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

19.- ¿La cota superior de los arreglos es igual al tamaño del arreglo o Valor - 1?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

20.- Si se usan cadenas de caracteres, ¿se asigna explícitamente un delimitador?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

21.- ¿Existe alguna posibilidad de desbordamiento de buffer?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

22.- Para cada enunciado condicional, ¿la condición es correcta?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

23.- ¿Hay certeza de que termine cada ciclo?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

24.- ¿Los enunciados compuestos están correctamente colocados entre paréntesis?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

25.- En caso de enunciados, ¿se justifican todos los casos posibles?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

26.- Si después de cada caso en los enunciados se requiere un paréntesis, ¿éste se incluyó?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

27.- ¿Se usan todas las variables de entrada?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

28.- ¿A todas las variables de salida se les asigna un valor antes de que se produzcan?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

29.- ¿Entradas inesperadas pueden causar corrupción?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

30.- ¿Todas las llamadas a función y método tienen el número correcto de parámetros?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

31.- ¿Los tipos de parámetro formal y real coinciden?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

32.- ¿Los parámetros están en el orden correcto?

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Siempre

33.- Si los componentes acceden a memoria compartida, ¿tienen el mismo modelo de estructura de memoria compartida?

Nunca 1 2 3 4 5 Siempre

34.- Si se modifica una estructura vinculada, ¿todos los vínculos se reasignan correctamente?

Nunca 1 2 3 4 5 Siempre

35.- Si se usa almacenamiento dinámico, ¿el espacio se asignó correctamente?

Nunca 1 2 3 4 5 Siempre

36.- ¿El espacio se cancela explícitamente después de que ya no se requiere?

Nunca 1 2 3 4 5 Siempre

37.- ¿Se tomaron en cuenta todas las posibles condiciones de error?

Nunca 1 2 3 4 5 Siempre

[Enviar](#) [Borrar formulario](#)

Este contenido no ha sido creado ni aprobado por Google. [Notificar uso inadecuado](#) - [Términos del Servicio](#) - [Política de Privacidad](#)

Google Formularios

ANEXOS G: Tablas de validez y confiabilidad

Pruebas de normalidad

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Calidad_de_Software	,204	6	,200*	,898	6	,360
Funcionalidad	,327	6	,043	,851	6	,161
Usabilidad	,314	6	,065	,878	6	,259
Eficiencia	,407	6	,002	,640	6	,001

*. Esto es un límite inferior de la significación verdadera. a.

Corrección de significación de Lilliefors

Resumen de procesamiento de casos

		N	%
Casos	Válido	6	100,0
	Excluido ^a	0	,0
	Total	6	100,0

a. La eliminación por lista se basa en todas las variables del procedimiento.

Estadísticas de fiabilidad

Alfa de Cronbach	N de elementos
,880	37

Estadísticas de total de elemento

	Media de escala si el elemento se ha suprimido	Varianza de escala si el elemento se ha suprimido	Correlación total de elementos corregida	Alfa de Cronbach si el elemento se ha suprimido
F1.- ¿Existe un responsable de la Gestión de Proyectos?	93,33	227,467	,836	,866
F2.- ¿Existe un Documento donde se definan objetivos y alcances del proyecto?	93,67	221,467	,967	,862
F3.- El jefe de proyecto, ¿define las actividades para cada proyecto?	93,33	257,467	,105	,880

F4.- ¿Responde a una plantilla predefinida?	94,50	249,500	,233	,880
F5.- ¿Los proyectos quedan documentados residen en un almacenamiento estable, tal como un disco magnético?	93,67	244,267	,749	,873
F6.- ¿Sabe usted cuando se da la petición de cambios de proceso en el proyecto?	93,67	229,867	,644	,870
F7.- ¿Se documenta la solicitud de cambio de requerimientos?	94,33	225,867	,883	,865
F8.- ¿Se determina el origen del cambio planteado en la solicitud?	94,00	224,400	,926	,864
F9.- ¿Se informa de la solicitud de cambio a los actores involucrados?	93,67	244,667	,728	,873
F10.- ¿Existe un responsable de la administración de la solicitud de cambio?	94,00	241,200	,447	,875
F11.- ¿Se conoce quién es el responsable de la administración de la solicitud de cambio?	93,67	229,067	,881	,866
F12.- ¿Una solicitud de cambio motiva la actualización del plan del proyecto?	94,33	233,467	,790	,868
F13.- ¿Existe hitos definidos de actualización del plan de proyecto?	94,67	244,267	,749	,873
F14.- ¿Se generan versionados ante la administración de las solicitudes de cambio?	94,17	237,367	,650	,871
F15.- ¿Se documentan adecuadamente las versiones definidas en la pregunta 9?	94,50	247,100	,369	,877

F16.- ¿Se utiliza alguna herramienta automatizada para documentar las versiones definidas en la pregunta 9?	94,67	252,267	,225	,879
U17.- ¿Todas las variables del programa se inician antes de usar sus valores?	93,83	256,567	,097	,881
U18.- ¿Todas las constantes tienen nombre?	94,17	236,167	,689	,870
U19.- ¿La cota superior de los arreglos es igual al tamaño del arreglo o Valor – 1?	94,00	246,800	,357	,877
U20.- Si se usan cadenas de caracteres, ¿se asigna explícitamente un delimitador?	94,33	256,667	,034	,885
U21.- ¿Existe alguna posibilidad de desbordamiento de buffer?	94,33	242,267	,417	,876
U22.- Para cada enunciado condicional, ¿la condición es correcta?	94,00	253,600	,146	,881
U23.- ¿Hay certeza de que termine cada ciclo?	94,67	243,867	,421	,876
U24.- ¿Los enunciados compuestos están correctamente colocados entre paréntesis?	93,83	258,167	,011	,883
U25.- En caso de enunciados, ¿se justifican todos los casos posibles?	94,00	239,600	,588	,873
U26.- Si después de cada caso en los enunciados se requiere un paréntesis, ¿éste se incluyó?	94,50	261,500	-,107	,884
U27.- ¿Se usan todas las variables de entrada?	94,17	262,167	-,112	,886
U28.- ¿A todas las variables de salida se les asigna un valor antes de que se produzcan?	94,00	262,400	-,191	,883

U29.- ¿Entradas inesperadas pueden causar corrupción?	94,17	260,567	-,065	,885
U30.- ¿Todas las llamadas a función y método tienen el número correcto de parámetros?	94,17	269,367	-,320	,890
U31.- ¿Los tipos de parámetro formal y real coinciden?	94,00	239,600	,760	,871
U32.- ¿Los parámetros están en el orden correcto?	94,17	236,567	,863	,869
F33.- Si los componentes acceden a memoria compartida, ¿tienen el mismo modelo de estructura de memoria compartida?	93,83	258,167	,011	,883
U34.- Si se modifica una estructura vinculada, ¿todos los vínculos se reasignan correctamente?	93,67	251,867	,359	,877
E35.- Si se usa almacenamiento dinámico, ¿el espacio se asignó correctamente?	93,83	265,367	-,214	,887
E36.- ¿El espacio se cancela explícitamente después de que ya no se requiere?	94,00	239,600	,588	,873
E37.- ¿Se tomaron en cuenta todas las posibles condiciones de error?	94,17	268,167	-,285	,889