

UNIVERSIDAD NACIONAL DE EDUCACIÓN

Enrique Guzmán y Valle

*Alma Máter del Magisterio Nacional*

FACULTAD DE CIENCIAS

Escuela Profesional de Matemática e Informática



**MONOGRAFÍA**

**CONTROL DE CALIDAD DEL SOFTWARE.**

**Conceptos de calidad. Historia y evolución. Normas y estándares ISO.  
Calidad de software. Medición de calidad. Fundamentos de la prueba del  
software. Tipos de prueba. Estudios de la calidad del software educativos.  
Modelo de calidad de software. Garantía de calidad de Software.  
Aplicaciones.**

Examen de Suficiencia Profesional Res. N° 0599-2019-D-FAC

Presentada por:

**Maguiña Pastrana, David Wilmer**

Para optar al Título Profesional de Licenciado en Educación

Especialidad: Informática

Lima, Perú

2019

## MONOGRAFÍA

### CONTROL DE CALIDAD DEL SOFTWARE.

**Conceptos de calidad. Historia y evolución. Normas y estándares ISO.  
Calidad de software. Medición de calidad. Fundamentos de la prueba del  
software. Tipos de prueba. Estudios de la calidad del software educativos.  
Modelo de calidad de software. Garantía de calidad de Software.  
Aplicaciones.**

Designación de Jurado Resolución N° 0599-2019-D-FAC



---

**Dr. Quivio Cuno, Richard Santiago**  
Presidente



---

**Dr. Quispe Andía, Adrián**  
Secretario



---

**Mg. Márquez Beltrán, José Alberto**  
Vocal

Línea de investigación: Tecnología y soportes educativos

**Dedicatoria**

A mis padres y familia, que siempre me apoyaron en todas mis etapas de la vida, manteniéndose pendiente por mi bienestar y salud.

A Dios, que siempre está contigo; cuando lo necesites o no, solo se necesita dar un salto de fe.

## Índice de contenidos

Portada.....	i
Hoja de firmas de jurado .....	ii
Dedicatoria.....	iii
Índice de contenidos .....	iv
Lista de figuras .....	viii
Introducción.....	ix
Capítulo I. Conceptos de calidad.....	10
1.1  Calidad .....	10
1.1.1 Etimología de calidad. ....	10
1.2  Definiciones de calidad .....	10
Capítulo II. Historia y evolución .....	12
2.1  Historia.....	12
2.2  Primeras civilizaciones.....	12
2.3  Revolución industrial .....	13
2.4  Periodo entre guerras.....	14
2.5  Japón después de la segunda guerra mundial.....	15
2.6  Década de los setenta-noventa .....	16
2.7  Década de los noventa-actualidad.....	16
Capítulo III. Normas y estándares ISO.....	18
3.1  Introducción .....	18
3.2  Composición y enlaces.....	19
3.3  Tipos de normas ISO.....	19
3.4  Proceso de elaboración de normas de la ISO y otras cuestiones.....	19

3.5	ISO 9000 .....	20
3.6	ISO 9001:2015 .....	20
3.7	Enfoque en procesos .....	21
3.7.1	Ciclo planificar-hacer-verificar-actuar. ....	22
3.7.2	Pensamiento basado en riesgos.....	22
3.7.3	Capítulos de la norma ISO 9000:2015.....	22
	Capítulo IV. Calidad de software .....	24
4.1	Contextualización de calidad de software.....	24
4.2	Enfoque de los modelos de calidad de software .....	24
4.2.1	Calidad a nivel de proceso.....	25
4.2.2	Calidad a nivel de producto. ....	25
4.2.3	Calidad en uso.....	25
	Capítulo V. Medición de calidad .....	26
5.1	Introducción .....	26
5.2	Factores de medición según McCall .....	27
5.3	Métrica de calidad según modelo Furps (1987).....	27
5.4	Métrica de Bang DeMarco .....	28
5.5	Métrica Modelo Sistemático de Calidad Mosca (Mosca) .....	28
5.6	Métrica del Qualification and Selection of Open Source (QSOS).....	29
	Capítulo VI. Fundamentos de la prueba del software .....	31
6.1	Fundamento.....	31
6.2	Prueba.....	31
6.3	Objetivos de las pruebas de software .....	32
6.4	Alcance de la prueba de software.....	32
	Capítulo VII. Tipos de prueba .....	34

7.1	Prueba de software .....	34
7.1.1	Pruebas funcionales.....	34
7.1.2.	Pruebas no funcionales.....	35
7.1.3.	Pruebas estructurales.....	36
7.2	Técnicas de prueba.....	36
7.2.1.	Técnicas estáticas.....	36
7.2.2.	Técnicas dinámicas.....	37
7.2.2.1.	Prueba de caja blanca.....	37
7.2.2.2.	Prueba de caja negra.....	37
7.2.3.	Técnicas basadas en la experiencia.....	38
7.3	Estrategia de prueba .....	39
7.3.1.	Pruebas unitarias o de componente.....	39
7.3.2.	Pruebas de integración.....	39
7.3.3.	Pruebas de sistema.....	40
7.3.4.	Pruebas de validación o aceptación.....	40
	Capítulo VIII. Estudios de la calidad del software educativo .....	41
8.1	Generalidades.....	41
8.2	Evaluación de la calidad de software educativo .....	41
8.3	Modelo de cuatro dimensiones para software educativo .....	43
8.4	Modelo TUP (Tecnología-facilidad de uso-pedagógicos).....	43
8.5	Evaluación de software educativo con base a la eficiencia pedagógica .....	44
8.6	Evaluación de software educativo a través de Internet .....	44
8.7	Evaluación de factores para el software educativo .....	45
8.8	MECSE (Métricas para evaluar la calidad de software) .....	45
8.9	Propuesta para la evaluación didáctica de software educativo .....	46

Capítulo IX. Modelo de calidad de software .....	47
9.1 Concepto .....	47
9.2 Modelos de calidad de software .....	47
9.3 Modelo a nivel de proceso .....	48
9.3.1 Modelo ITIL. ....	48
9.3.2 Modelo ISO/IEC 15504. ....	49
9.3.3 Modelo ISO/IEC 9126. ....	50
9.3.4 Modelo CMMI. ....	51
9.3.5 PSP. ....	52
9.3.6 Modelo Team Software Process (TSP). ....	53
Capítulo X. Garantía de calidad de software .....	55
10.1 Introducción .....	55
10.2 Actividades de SQA .....	57
10.3 Diferencia de SQA y SQC .....	57
10.4 Funciones generales del SQA .....	58
10.5 Aseguramiento de la calidad de procesos y productos en CMMI. ....	58
Aplicación didáctica .....	60
Síntesis .....	65
Apreciación crítica y sugerencias .....	66
Referencias .....	67

**Lista de figuras**

Figura 1. Línea de tiempo.....	13
Figura 2. Modelo ISO/IEC 15504. ....	49
Figura 3. Ciclo PHVA.. ....	63

## Introducción

La presente monografía, titulada Calidad de software, tiene la finalidad de dar a conocer una de las áreas más importantes de este campo. Cuando hablamos de calidad, muchas veces solemos entenderla como una sola y simple acepción, cuando en realidad es un término que ha ido evolucionando, cambiando a lo largo del tiempo desde la época más remota de la humanidad.

El software es una de las herramientas más útiles para optimizar los procesos de las empresas y organizaciones en el mundo con el propósito de aumentar la eficiencia y satisfacer necesidades. Por esta razón, varias empresas o personas dedicadas a la investigación pueden proponer tácticas, técnicas y normas en software que incluyen modelos y patrones de calidad que apoyan el crecimiento y lucimiento de productos de programas y sistemas, dependiendo de las necesidades de calidad, para que pueda evaluar si el producto realmente tiene un nivel de calidad. De esta manera se nutre un entorno de alta calidad.

Esta monografía se contextualiza, primero, en términos de calidad de software, comenzando con diferentes significados de distintos personajes como Deming e Ishikawa. Mientras tanto, tenemos una historia de calidad a corto plazo y una corta historia de evolución en los mundos este y oeste. El estándar de la Organización Internacional de Normalización (ISO) es un tema esencial cuando se habla de calidad, porque es la organización responsable de proporcionar un estándar a fin de tener un estándar de calidad para todos los artículos de un producto o servicio.

Posterior a esto se realizará una clasificación de los fundamentos de la prueba del software, tipos de prueba, patrones de excelencia y garantía de condición de software.

## **Capítulo I**

### **Conceptos de calidad**

#### **1.1 Calidad**

Existen varios conceptos para la palabra calidad. Haremos lo posible por darle una definición o explicar lo que se entiende de ella, debido a que es un tema muy tratado por diferentes profesionales en el mundo, empresas, organizaciones. A continuación, se presenta a algunos personajes que la definen de una manera determinada.

##### **1.1.1 Etimología de calidad.**

Según la Real Academia Española (RAE, 2019) “etimológicamente, el término calidad procede del latín *qualitas-atis*, que a su vez proviene del griego *poiotes*, y se entiende como las cualidades o atributos de algo” (p.1).

#### **1.2 Definiciones de calidad**

Según menciona Ishikawa (1988) calidad “es fomentar, trazar y mantener un artículo de excelencia, que sea el más austero, beneficioso y totalmente conveniente en la persona que va a hacer uso de ella” (p.42). No debe entenderse solo como calidad del producto.

Por su parte, Deming (1989) expresa que esta se define como el grado de uniformidad y confiabilidad que se puede predecir a un bajo costo, y se ajusta a lo que el mercado requiere.

Crosby (1989) manifiesta que es impecable y en un tiempo razonable para entregarlo al trabajo, productos y servicios de clientes y personas cercanas.

Yamaguchi (1989) la define como un cúmulo de bienes, características que, según su postura, pueden complacer actividades ya determinadas.

Según Feigenbaum (1994) la calidad es un sistema eficiente que integra los empeños para mejorar la administración de varios grupos en una organización y puede ofrecer un nivel de productos y servicios que permiten el deleite de la persona a un costo modesto para la organización.

Para Cuatrecasas (1999) examina la calidad como el cúmulo de detalles que posee un artículo o utilidad obtenidos en un sistema fructífero.

Según la norma ISO 9000 (2005) el conjunto de características únicas es suficiente para cumplir con los requisitos y se expresa como la concreción de estos requerimientos únicos basados en la satisfacción del menester del usuario.

## **Capítulo II**

### **Historia y evolución**

#### **2.1 Historia**

Según Francisco (2007) la historia:

Con el tiempo, a partir de la historia podemos encontrar diferentes expresiones de interés humano en la definición de la calidad, por lo que para comprender el significado actual del término puede ser apropiado analizar su desarrollo histórico, para lo cual debemos dividirlo en cinco etapas o momentos específicos en la historia humana (p.38).

#### **2.2 Primeras civilizaciones**

Miranda, Chamorro y Rubio (2007) conciben que:

El primer rastro de interés en la calidad humana se remonta a Babilonia, la civilización más antigua de la humanidad. Esto se describe en el código Hammurabi. En la cultura mesopotámica se declaró lo siguiente: si un trabajador de obra civil edifica una vivienda para una persona y si el labor no es resistente y la vivienda se viene abajo asesinando al propietario, el trabajador de obra civil será castigado a fenecimiento (p.2).

En la construcción de civilizaciones como la egipcia y las centroamericanas se puede ver la atención que tenían sobre las medidas relacionadas con la agrupación de piedras.

Las civilizaciones antiguas a menudo eran importantes para la igualdad comercial y la conclusión de inconvenientes en la generalidad de sucesos, incluso si le daban a la persona a cargo la mayor penalidad.

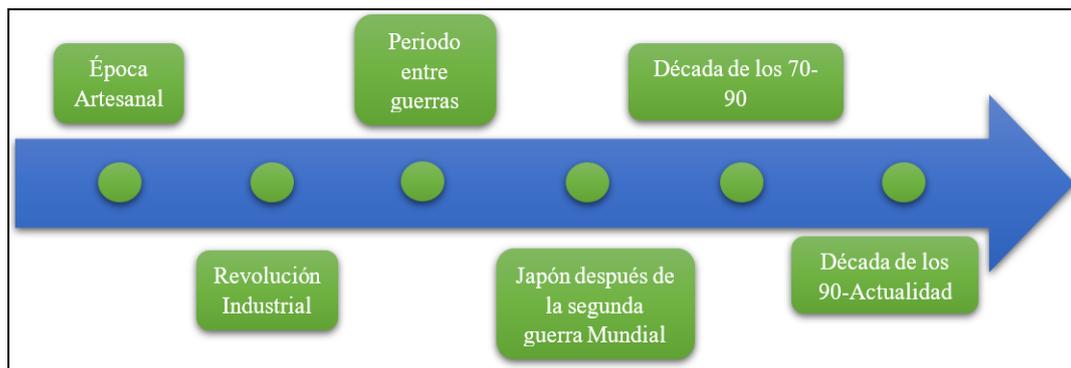


Figura 1. Línea de tiempo. Fuente: Autoría propia.

### 2.3 Revolución industrial

El concepto que se conoce hoy nació en este lapso a partir de mediados del siglo XVIII hasta principios del siglo XIX, cuando las máquinas se integraron en el taller y crearon nuevas formas de trabajo desde la Revolución Industrial. Inicialmente se mantenía una comunicación directa entre el fabricante del producto y el consumidor, por lo que el fabricante era responsable de la calidad durante todo el proceso de fabricación del producto.

Pero con el tiempo comienzan a existir nuevas formas de producción. Se habla sobre la producción en serie. Esto permite que las personas fabriquen y otros controlen la calidad del producto. Aquí es donde nació el concepto de inspectores de calidad de fábrica y hay un departamento a cargo de ello. Esta separación de los roles de ejecución, planificación y control tuvo un impacto negativo en la calidad del producto final.

A pesar de todo esto, en este momento, la baja calidad del producto debido a la demanda del consumidor no es un gran problema; la demanda es excesiva en comparación con la oferta, pues el consumidor ha comprado el producto así esté mal elaborado.

## **2.4 Periodo entre guerras**

Llegando a finales del siglo XIX, en EE.UU. deja de haber una comunicación total entre los fabricantes y los consumidores o clientes, debido a que era imposible que se de esto. No se podía satisfacer las necesidades de un cliente.

El proceso de estandarización de las condiciones y métodos de trabajo comenzó oficialmente, en otras palabras, cuando apareció la teoría de la gestión científica, siendo el personaje principal Frederick Winslow Taylor.

Según Evan y Lindsay (2008) los pilares básicos de la mencionada hipótesis decreta que los planes de trabajo y las actividades de elaboración deben separarse para aumentar la productividad. Este nuevo paradigma fue el primero en producir defectos obvios en la calidad del beneficio. Esto se debe a que los errores humanos han aumentado significativamente porque las pruebas realizadas por cada operador han desaparecido como antes.

La solución fue adaptar la fundación del ejercicio de revisar la fábrica, concentrándose en una persona conciente de determinar si los productos eran satisfactorios o insatisfactorios.

El principal motivo para la utilización de los nuevos enfoques vino con la Segunda Guerra Mundial, al imponer el ejército de Estados Unidos (EE.UU). normas más severas a sus proveedores para garantizar productos fiables en un pequeño lapso de tiempo.

Este es el primer método que aparece en el organigrama creado por la empresa y el departamento de control de calidad. El departamento de control de calidad verifica cada producto a través de la inspección, detecta si hay un defecto, toma contramedidas según sea necesario y brinda una solución. En esta etapa, la calidad se entendía como tener una lucha contra los defectos y no la causa; teniendo un enfoque puramente correctivo, porque el único responsable era el departamento de inspección de calidad.

## **2.5 Japón después de la segunda guerra mundial**

Conforme a Evans y Lindsay (2008):

Luego que finalizó la Segunda Guerra Mundial, la calidad continúa por dos caminos distintos, básicamente por los hemisferios oriental y occidental. El lado occidental mantenía el enfoque basado en inspeccionar el producto, pero en el lado oriental, Japón, se maneja un enfoque totalmente distinto. Durante los años cincuenta, comprendió que para no fabricar productos con defectos y venderlos era necesario elaborar productos buenos desde un comienzo. A este enfoque se le conoce como prevención (p.74).

Deming llegó a Tokio y en 1947, ahora conocido por una serie de reuniones sobre control estadístico de calidad y modelos de gestión que procesan la calidad de una manera más eficiente, conociendo así el ciclo Planificar, Hacer, Verificar y Actuar (PHVA).

Por lo tanto, a pesar de que el control de calidad empezó con una idea de inspección, con los años pasó a ser la prevención la forma sobre cómo controlar los factores del proceso de creación del producto referente a su calidad.

Joseph M. Juran arribó Japón en 1954 y participó en enaltecer el compromiso del área gerencial, pudiéndose así lograr un control de calidad con el modelo PHVA, que luego fue teniendo aceptación en todas partes del mundo.

Mencionando a Armand V. Feigenbaum, quien originó el concepto de gestión de la calidad o gestionar la calidad, insertó el programa de la calidad, también aplicó por primera vez lo que se conoce como control total en EE.UU., que apareció en 1951 en su libro Total Quality Control. Este personaje también visitó Japón y aportó una serie de mejoras en la calidad de los productos de dicho país, consolidados después por Ishikawa a partir de 1955.

En 1962, el emprendedor japonés Ishikawa Ka fundó el primer círculo de control de calidad en Japón para mejorar significativamente la calidad del producto.

## **2.6 Década de los setenta-noventa**

Debido a los inéditos paradigmas económicos mundiales, se manifiesta una siguiente etapa en el proceso de calidad, originando así una expectativa mundial “El proceso de calidad total”

En esta etapa se puede percibir un denuedo por lograr obtener la calidad en todas las características del producto dentro de las organizaciones sin importar el rubro en que se encuentren. Esto implica que los que intervenían en el proyecto, diseño y otras actividades de improvisados productos puedan participar sin ser excluidos.

## **2.7 Década de los noventa-actualidad**

Según Evan y Lindsay (2008) “la tercera etapa culmina con el inicio de la calidad por los años noventa, cuando aparecen inéditas manifestaciones socioeconómicas, como lo es la globalización, que cambia en su totalidad el concepto de una empresa” (p.84).

En esta última etapa, la calidad es conocida como una mejora permanente en el marco de calidad total, el factor humano cumple un rol de suma importancia al iniciar un proceso constante de disminuir los costos de materiales para la elaboración. La empresa

descubre que tiene que emprender su propio crecimiento, pero de una forma ordenada, sistematizada.

Al competir en un mercado globalizado, las empresas se enfrascan en el desafío de producir y vender productos con un estándar alto en calidad al mínimo costo de producción. Los esfuerzos de las personas por lograr una óptima secuencia de procesos se ven reflejada en una reducción de los costos, junto a la reducción de la distancia entre el cliente, lo que se traduce en las ventas y, por tanto, ello refleja una mejora en las ganancias para la empresa.

## **Capítulo III**

### **Normas y estándares ISO**

#### **3.1 Introducción**

Una organización que integra organismos nacionales de normalización con el objetivo de facilitar el desarrollo de normas relacionadas con productos y servicios en sectores e industrias específicas de todo el mundo. ISO no es una organización internacional pública. Algunos organismos de estandarización son parte de la estructura gubernamental del país o son administrados por órdenes gubernamentales que pertenecen a otras instituciones privadas. En el caso del Perú, el organismo público autónomo es el Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual, que se le conoce con el acrónimo de Indecopi.

Muchos de los organismos de normalización se encuentran en una estructura que pertenece al gobierno de cada país y es controlado por órdenes gubernamentales, mientras que el resto pertenece al sector privado. Sin embargo, una de las organizaciones autónomas es el Indecopi. Por otro lado está el Instituto Nacional de Calidad nombrado por el acrónimo Inacal, una organización técnica pública especializada que pertenece al Ministerio de Producción.

### **3.2 Composición y enlaces**

Actualmente, ISO tiene 156 personas y una secretaria central en Ginebra, Suiza. Existe un miembro adicional para cada país registrado y hay tres categorías de miembros.

### **3.3 Tipos de normas ISO**

Hay dos tipos generales de estándares: estándares obligatorios y recomendaciones o pautas. La certificación se puede hacer con los estándares requeridos. ISO es responsable de desarrollar, actualizar y publicar estándares actualizados. Existen organismos de notificación activos en todas partes y son administrados por los mandos del país donde se encuentran.

### **3.4 Proceso de elaboración de normas de la ISO y otras cuestiones**

A finales de 2006, ISO publicó 16.455 normas internacionales sobre una amplia gama de temas, incluidos productos técnicos, servicios financieros y sistemas de gestión. Debido a que los miembros de ISO no son gobiernos, los estándares de ISO no son legalmente vinculantes. Los gobiernos pueden crear leyes usando reglas. O el departamento de actividades puede adoptar voluntariamente las reglas. Las normas ISO pueden ser prácticamente necesarias si son ampliamente aceptadas en el campo de actividad.

El desarrollo de las normas ISO puede llevar de 3 a 5 años, pero si el problema es controvertido, puede llevar más tiempo. Según ISO, la norma es el principio de consenso a nivel de todo el sector de actividad, fabricantes, vendedores, usuarios, grupos de consumidores, laboratorios, gobiernos, expertos técnicos, instituciones de investigación.

Una vez publicado el estándar internacional, el comité técnico que desarrolló el estándar lo actualizará cuando ocurra un caso. Si se descubre un error o ya no se necesita

información, se puede publicar una solución técnica. Si necesita introducir nueva información o cambios en ciertas partes del estándar, puede hacer correcciones.

### **3.5 ISO 9000**

Este estándar internacional está diseñado para ayudarlo a comprender los conceptos básicos requeridos, los principios de calidad y una amplia gama de términos administrativos para que pueda aplicarlos de manera eficiente y efectiva.

Esta norma internacional consta de siete principios de control de calidad que respaldan conceptos básicos. Cada principio tiene una "base racional" que especifica por qué la organización necesita trabajar en ello.

### **3.6 ISO 9001:2015**

Según Noguez (2004) es la regla del método de gestión más popular y certificada del mundo". Inicialmente, el sector industrial era el sector más comúnmente introducido y certificado desde su nacimiento en 1987, pero recientemente se ha vuelto particularmente prominente en sectores como la provisión de tecnología y servicios.

El mayor uso en estas nuevas áreas es consistente con la evolución del estándar y los cambios que presentó ISO 9001: 2015, que fue desarrollado para adaptarse a una nueva era en una situación más dinámica, volverse complejo en el mundo de los negocios y tener las características de insertar nuevas prácticas y tecnologías.

El 23 de septiembre de 2015 se dio a conocer la norma internacional ISO 9001: 2015. Esto requirió mucho trabajo hasta ese día, pero el documento normativo tuvo que seguir un camino aburrido que comenzó en 2012.

Los objetivos principales que ISO persigue con esta originaria versión de la norma ISO 9001 son los siguientes:

- Preservar la aplicabilidad de la regla.
- Proporcionar requisitos básicos estables por más de 10 años.
- Mantener la versatilidad para constituciones de todos los tamaños y tipos, operando en todos los sectores.
- Concéntrese en la administración de procesos ahora para obtener los resultados esperados.
- Tiene en consideración los cambios realizados desde la última revisión importante en 2000.

### **3.7 Enfoque en procesos**

Como se menciona en Organización internacional de Normalización (ISO, 2015) “este estándar internacional facilita la adopción de un enfoque basado en procesos para el desarrollo, implementación y mejora de la efectividad de los sistemas de gestión de calidad y mejora la satisfacción del cliente” (p.101).

Administrar y comprender cada proceso interrelacionado como un sistema puede ayudar a la compañía a mejorar su efectividad para lograr las metas y los resultados esperados. Esta perspectiva le permite a la organización innovar el trabajo general de la organización porque puede controlar entre los procesos del sistema.

El concepto y la gestión sistemática del proceso y sus interacciones se centran internamente en el proceso para lograr los resultados deseados de acuerdo con las normas ISO. La gestión de todo el proceso y del sistema se puede lograr mediante la aplicación de un ciclo de PHVA y un enfoque basado en el proceso.

### **3.7.1 Ciclo planificar-hacer-verificar-actuar.**

Los ciclos de PHVA pueden aplicarse en todos los procesos y sistemas de gestión de calidad y pueden agruparse en relación con los ciclos de PHVA. El ciclo de PHVA se puede describir brevemente de la siguiente manera:

- **Planificar:** Esto significa aclarar el propósito del sistema y sus procesos, las necesidades del cliente y los materiales necesarios para producir y entregar los resultados de acuerdo con las políticas de la organización.
- **Hacer:** Se refiere a implementar, llevar a cabo lo planificado.
- **Verificar:** Realizar el seguimiento y medir los procesos, productos y servicios resultantes para las políticas, metas, requisitos y actividades planificadas.
- **Actuar:** Tomar medidas para mejorar el rendimiento, si es necesario.

### **3.7.2 Pensamiento basado en riesgos.**

Lograr un sistema de control de calidad efectivo es muy importante. Las organizaciones necesitan planificar y ejecutar acciones para llevar riesgos y oportunidades. Abordar el riesgo como una oportunidad aumenta la eficiencia del sistema de gestión de calidad, logra mejores resultados y establece una base para prevenir los efectos adversos.

### **3.7.3 Capítulos de la norma ISO 9000:2015.**

La norma está dividida por capítulos, los cuales están relacionado al ciclo PHVA.

- **Contexto organizacional:** Debe determinar los problemas externos e internos relacionados con el propósito y la dirección estratégica de su organización.
- **Liderazgo:** La alta gerencia necesita demostrar liderazgo y compromiso con el sistema de gestión de calidad.

- **Planificación:** La organización debe considerar los riesgos y las oportunidades que sean necesarias con el fin de asegurar la calidad y lograr la mejora.
- **Apoyo:** Las organizaciones necesitan determinar y proporcionar los recursos necesarios para establecer, implementar, mantener y mejorar continuamente.
- **Operación:** La organización debe planificar, implementar y controlar los procesos necesarios a fin de cumplir los requisitos para la prevención de productos o servicios.
- **Evaluación de desempeño:** Este capítulo cubre las actividades de monitoreo, medición, análisis y evaluación de la idoneidad del sistema de gestión y la satisfacción del cliente.
- **Mejora:** Las organizaciones deben determinar y seleccionar oportunidades de mejora y tomar las medidas necesarias para cumplir con los requisitos del cliente.

## **Capítulo IV**

### **Calidad de software**

#### **4.1 Contextualización de calidad de software**

Según Callejas (2017):

La calidad de software se refiere al nivel en que se desenvuelve las principales características con las que debe contar un sistema de cómputo mientras dure su ciclo de vida, es decir, debe de otorgar la garantía que el cliente cuente con un sistema confiable, lo cual aumenta los requerimientos del cliente y su eficiencia del sistema (p.14).

#### **4.2 Enfoque de los modelos de calidad de software**

Los diseños de calidad de software pueden desarrollarse y tener varios factores de calidad que comparten los criterios evaluados por las métricas, así como las razones para validar de general a específico.

Del mismo modo, los diseños de calidad de software se clasifican de acuerdo con un enfoque de valoración, independientemente del proceso, producto o nivel de calidad usado.

#### **4.2.1 Calidad a nivel de proceso.**

La calidad del software debe programarse desde el principio hasta el final del plan y realizarse en cada nivel de la gestión de desarrollo. Gestione y supervise los aspectos de calidad para minimizar el riesgo y proporcionar soporte continuo. Si queda algún factor y criterio de validación en cualquier etapa, considere que puede haber defectos, los cuales reducen el nivel de calidad del producto final y el proceso.

#### **4.2.2 Calidad a nivel de producto.**

Uno de los objetivos principales es evaluar y especificar el cumplimiento del producto o servicio. Esto aplica medios externos e internos, por lo que se definen tres tipos: externos, internos y de uso. Esta perspectiva tiene como objetivo verificar la realización de todas las características que hacen posible cumplir con las expectativas del cliente sobre las necesidades definidas anteriormente en el proceso de desarrollo.

#### **4.2.3 Calidad en uso.**

Covella (2005) manifiesta que la calidad en uso se define como un conjunto de atributos relacionados con la aceptación del usuario final y se basa en la eficiencia, la productividad, la seguridad y la satisfacción.

## **Capítulo V**

### **Medición de calidad**

#### **5.1 Introducción**

Como expresa Pressman (2010) el objetivo fundamental de la ingeniería de software es crear un producto que pueda ser un programa con la mejor calidad posible, por tal motivo los ingenieros utilizan diversos métodos altamente calificados y herramientas propias del contexto con un proceso de elaboración totalmente seguro.

Por su parte, McCall y Cavano (1991) le otorgan el significado de un grupo de elementos de alta calidad, siendo este la primera pieza angular de las métricas del software. Estas características del software evalúan dentro una perspectiva de tres puntos en consideración. La gente en el campo está explicando cómo estos factores de calidad y otras características se relacionan con la ingeniería de software.

Primero, desde una perspectiva de trabajo, los administradores pueden usar herramientas para identificar lo que creen que necesita el proyecto, como confiabilidad, facilidad de uso y consistencia.

En segundo lugar, proporciona una forma de evaluar la cantidad de progreso realizado durante la fase de desarrollo en relación con los objetivos de calidad previamente establecidos. Proporcionar más funciones al personal de calidad en el trabajo de desarrollo.

Finalmente, el personal de calidad puede usar pautas de calidad.

## 5.2 Factores de medición según McCall

Según McCall (1977) la calidad tiene un significado importante por medio de relaciones de grado entre propiedades de calidad, teniendo como base perspectivas que se centran en tres características importantes de una producción de software: propiedades operativas, capacidad de cambios y adaptabilidad a nuevos entornos. Es necesario identificar una serie de criterios, como lo son la simplicidad, la capacidad de extenderse, entre otros.

Las métricas impuestas tienen una relación con los principios de calidad y la relación que esta tiene se mide en función del grado en que pueda cumplir con los criterios.

## 5.3 Métrica de calidad según modelo Furps (1987)

Por su parte, Pressman (1998) señala que el modelo FURPS fue desarrollado principalmente por uno de los personajes pioneros de la informática, Hewlett-Packard, en el año 1987. En él se desarrollan un conjunto de características de calidad de software. El acrónimo FURPS tiene un significado que consta de cuatro términos: funcionalidad, usabilidad, fiabilidad y rendimiento, a los cuales podemos agregar la propiedad de capacidad de soporte.

- **Funcionalidad:** Se puede observar evaluando en grupos las capacidades y características del programa en desarrollo, los aspectos generales de las funciones entregadas en un entorno de seguridad del sistema global.
- **Usabilidad:** Se centra en los aspectos humanos, estéticos y documentación que esta posee.
- **Fiabilidad:** Se evalúa teniendo en cuenta la frecuencia e intensidad de los fallos que posee, teniendo en cuenta la recuperación de un fallo y la calidad de predicción del sistema o programa.

- Rendimiento: Se calcula por la rapidez de transformación, el tiempo en que se demore en dar una respuesta, el rendimiento efectivo total y eficacia.

#### 5.4 Métrica de Bang DeMarco

Bang De Marco (1991) manifiesta que puede utilizarse para realizar una orden en relación al software a desarrollar, como consecuencia del modelado del análisis. Es una instrucción propia de la manera en cómo se lleva a cabo teniendo en cuenta la magnitud del sistema. Para poder calcular la métrica de Bang, la persona que realiza el proyecto tiene que tener en cuenta unas indicaciones llamadas primitivas.

- Primitivas funcionales (Pfu): Cambios o permutaciones que empiezan en el nivel primario de un gráfico de flujo de datos (DFD).
- Componentes de datos (ED): Características de un objeto de datos, los elementos no compuestos y son nombrados en el vocabulario de datos.
- Objetos (OB): Objeto de datos.
- Relaciones (RE): Las relaciones entre objetos de datos.
- Transiciones (TR): El número de arreglos de estado en el grafico de transición de estado.

#### 5.5 Métrica Modelo Sistemático de Calidad Mosca (Mosca)

Según Mendoza (2002) es una perspectiva que incluye los modelos de calidad mencionados anteriormente, considerados hasta cierto punto submodelos de este. Principalmente, la cualidad del proceso garantiza la calidad del producto y, consecuentemente, no puede desprenderse de estas cualidades. Mantener modelos separados capaces de medir individualmente la calidad de un producto o de un proceso de software no garantiza la conexión sistemática que debe presentarse entre ambos, para que sea visto como un todo.

- Dimensiones: En el nivel 0 las cuatro magnitudes propuestas por Mosca son: eficiencia del proceso, realidad del proceso, eficiencia del producto y efectividad del producto.
- Categorías: En el nivel 1 se contemplan once jerarquías, de las cuales 6 corresponden al producto y 5 al proceso de desarrollo.
- Características: En el nivel 2 cada categoría tiene asociado un conjunto de características, las cuales dotan de las características de cada una de las áreas claves a reparar para lograr afianzar y examinar la calidad del producto y/o del proceso.
- Métricas: En el nivel 3, cada dimensión ofrece un conjunto de métricas utilizadas para medir la calidad sistemática, teniendo en cuenta la cantidad de valores asociados con cada una de las características que componen Mosca.

## **5.6 Métrica del Qualification and Selection of Open Source (QSOS)**

Es una de las formas que permite contar y calificar el software. Open source es el método para calificar y escoger un documento de software libre, está destinado enteramente al producto de software.

Dentro de su metodología, está compuesta por cuatro partes:

- Definir: Progresar los marcos de referencia que son de vital importancia en los pasos que continúan. El principal objetivo es darles una definición a los elementos de la tipología.
- Evaluación: En esta parte tiene el objetivo primordial de conseguir información mediante el código abierto, información disponible.
- Calificación: El propósito de este paso es definir un filtro que traduzca las necesidades y limitaciones asociadas con la selección de software de código abierto en un contexto específico.

- Selección: El propósito de este paso es identificar el software que incluye y cumple con sus requisitos. De manera más general, permite comparaciones de software de la misma familia.

## Capítulo VI

### Fundamentos de la prueba del software

#### 6.1 Fundamento

Según la RAE (2018) una de las acepciones de la palabra concepto significa “Raíz, principio y origen en que estriba y tiene su mayor fuerza algo no material” (p.1).

#### 6.2 Prueba

Como expresa Myers (1979) los desarrolladores siguen las siguientes afirmaciones que llevan a una percepción errónea:

- Las pruebas son el proceso de demostrar que no hay posibles confusiones.
- El asunto de las pruebas es demostrar que un programa opera funciones en óptimas condiciones.

Myers (1979) precisa que estas definiciones:

Tienen un concepto vago en el contexto en que fueron planteadas. Cuando probamos un programa se quiere aportar un valor agregado a lo que estamos evaluando, aumentar la calidad y fiabilidad, y esto nos lleva a tener que encontrar y a continuación eliminar los errores en el programa. Esto se traduce a que no tenemos que evaluar un programa para demostrar que funciona, sino empezar con

la premisa que el programa tendrá errores. La prueba es el transcurso de ejecución de un programa con el designio de encontrar errores (p.10).

Según el International Software Testing Qualifications Board (ISTQB, 2015)

“Consiste en todas las actividades de ciclo de vida estáticas y dinámicas asociadas con el diseño, la preparación y la evaluación de software y productos relacionados con el trabajo, a la vez que cumple con los requisitos especificados” (p.15).

Edsger (2015) menciona que “las evaluaciones de software pueden ser la mejor forma de demostrar la existencia de errores, pero no son del todo adecuadas para demostrar la existencia de errores” (p.33).

### **6.3 Objetivos de las pruebas de software**

Según el ISO 9126 los objetivos fundamentales que se esperan encontrar con la prueba de software son:

- Comprender el nivel de calidad del producto de software para facilitar una correcta administración del tiempo en que estará en venta el producto.
- No costear un producto que no está implementado a su 100%, hasta que no haya sido elaborado como lo desea el cliente.
- Reducir costos de mantenimiento mediante el diagnóstico.
- Conseguir información correcta acerca de errores, que pueda gastar como base en la evolución o perfección del producto.

### **6.4 Alcance de la prueba de software**

La prueba de software tiene limitaciones, tanto en la práctica y la teoría. Desde una perspectiva teórica, la prueba es un problema que suele conocerse como no decidible; esto significa que no podemos escribir un programa que pruebe los programas sin que el

hombre tenga un rol. Sin embargo, como mencionábamos, la prueba sí es automatizable en varias características.

De acuerdo con Edsger (2015)

Desde una perspectiva práctica, la cantidad de posibilidades para probar exhaustivamente un sistema es sencillamente inmanejable; es necesario utilizar técnicas adecuadas para aumentar la cantidad de fallos encontrados en el sistema. Cada forma que se utilice para identificar defectos deja un residuo de defectos más sutiles contra los cuales ese método es ineficaz. La prueba de software involucra la utilización de técnicas y herramientas apropiadas en el marco de un proceso bien determinado (p.27).

## **Capítulo VII**

### **Tipos de prueba**

#### **7.1 Prueba de software**

Existen varios tipos de pruebas de software. Algunos de ellos necesitan probar la funcionalidad del software en sí, otros intentan analizar características no funcionales como la confiabilidad, y otros intentan probar la estructura del software.

Teniendo en cuenta esto, se diferencian tres tipos principales:

##### **7.1.1 Pruebas funcionales.**

Los tipos de prueba del sistema funcional se describen como especificaciones de requisitos. Esa es la función del sistema. Aunque puede no estar documentado, se requiere experiencia avanzada para interpretar este tipo de evidencia.

También se dividen en las siguientes características:

Las propiedades funcionales descritas en ISO 25010 son, entre otras, precisión y seguridad. Esta característica también se especifica para representar la capacidad de los productos de software de proporcionar funcionalidad basada en los estándares del cliente.

- **Integridad funcional:** la medida en que una función cubre todas las tareas y objetivos de un usuario determinado.

- **Modificación funcional:** la capacidad de un producto o sistema para proporcionar el resultado correcto con el nivel de precisión requerido.
- **Afiliación funcional:** la capacidad de un producto de software de proporcionar un conjunto de características apropiadas para una tarea específica y el propósito del usuario.

Las pruebas funcionales generalmente se asocian con métodos de diseño de prueba de recuadro negro para tener en cuenta el comportamiento externo del software.

### **7.1.2 Pruebas no funcionales.**

Este tipo de pruebas a menudo consideran el comportamiento externo del software, es decir, el comportamiento del sistema, y utilizan técnicas de diseño de recuadro negro. Al igual que las características funcionales, las características no funcionales deben definirse en la especificación del producto. ISO 25010 también define las propiedades: confiabilidad, facilidad de uso, eficiencia, compatibilidad y seguridad requerida para estas pruebas.

Conforme menciona Myers (2004) las siguientes características no funcionales deben considerarse en las pruebas:

- **Pruebas de rendimiento:** estas pruebas miden la velocidad de procesamiento y el tiempo de respuesta del sistema.
- **Pruebas de volumen:** mide la capacidad de un sistema para procesar grandes cantidades de datos, como el procesamiento de archivos muy grandes.
- **Pruebas de esfuerzo:** se realiza una prueba para sobrecargar el sistema y analizar la capacidad de recuperación.
- **Pruebas de compatibilidad:** estas son pruebas de comportamiento del sistema utilizando diferentes sistemas operativos y plataformas de hardware.

- Pruebas de usabilidad: la usabilidad, la efectividad y la satisfacción siempre se miden dentro de un grupo de usuarios específico.

### **7.1.3 Pruebas estructurales.**

Según ISTQB (2015) las evaluaciones organizadas le permiten medir toda la prueba evaluando el tipo estructural. En estas evaluaciones se llevan a cabo técnicas de anteproyecto de caja blanca.

## **7.2 Técnicas de prueba**

Para mejorar la calidad del producto, se debe observar varias técnicas que se pueden aplicar a las pruebas. Estos métodos están destinados a condiciones de prueba, recuperación de casos y gestión de datos.

- Técnicas estáticas
- Técnicas dinámicas
- Técnicas basadas en la experiencia

### **7.2.1 Técnicas estáticas.**

Para las pruebas basadas en métodos estáticos, no se necesita ejecutar la aplicación. Esto solo se hace a un nivel de especificación mucho más teórico. No ejecuta código, pero también verifica parte de la documentación del proyecto, como especificaciones, requisitos y casos de prueba para análisis estático.

Su objetivo principal es ejecutar este tipo de herramienta de prueba para identificar errores en el código fuente del programa o software y analizar el código del programa. Este tipo de prueba puede ayudarlo a encontrar fácilmente defectos o errores sospechosos en su código o diseño.

### **7.2.2 Técnicas dinámicas.**

Este tipo de técnicas se realizan ejecutando la aplicación y se utilizan para diseñar casos de prueba.

La mayoría del software se puede probar de dos maneras diferentes. Si conoce la operación interna, puede probar que todos los módulos son compatibles entre sí, es decir, desde una perspectiva interna. Estas son pruebas de caja blanca.

Las técnicas dinámicas son técnicas de ejecución de código que también se utilizan para diseñar casos de prueba.

#### ***7.2.2.1 Prueba de caja blanca.***

El portal EcuRed (2018) define a la caja blanca como: “La evaluación de caja blanca, también llamada "caja de vidrio", es una herramienta de diseño de evaluación que utiliza una estructura de control para capturar casos de prueba” (parr.1)

Dentro de esta estructura de control, puede encontrar la estructura de los componentes de hardware, como las declaraciones de decisión, las diferentes rutas del código y la estructura de las páginas web.

Los métodos de prueba de caja blanca indican los siguientes puntos:

- Asegurar que todas las rutas de código se revisan al menos una vez.
- Chequear las condiciones lógicas.
- Chequear las estructuras de datos.

#### ***7.2.2.2 Prueba de caja negra.***

El portal EcuRed (2018) define la caja negra como:

El método de diseño de caja negra, también llamado prueba de comportamiento, utiliza análisis de especificaciones funcionales y no funcionales sin considerar la

estructura interna del diseño de prueba de caso y prueba de contraste. Cuadro blanco, estas pruebas se pueden realizar en la etapa final de la prueba (parr.1).

Con los métodos de caja negra se intentan hallar los posibles errores:

- Funciones faltantes o con errores
- Errores al empezar y terminar
- Error de interfaz
- Errores en las estructuras

### **7.2.3 Técnicas basadas en la experiencia.**

Según ISTQB (2015) se definen las técnicas basadas en la experiencia y se definen como pruebas que parten de la habilidad del operante, su experiencia con aplicaciones similares.

Estos tipos de pruebas pueden tener poco efecto dependiendo de la experiencia del probador y generalmente se aplican a proyectos con documentación insuficiente o insuficiente. Existen dos técnicas dentro de las técnicas basadas en la experiencia:

- Predicción de error: En este tipo de prueba, los casos de prueba se diseñan en función de la experiencia del probador, se esperan errores y se diseñan pruebas específicas en función de esa experiencia.
- Pruebas exploratorias: Este tipo de prueba enfatiza la libertad y la responsabilidad del individuo de mejorar continuamente el valor del trabajo aprendiendo la prueba, diseñando la prueba, ejecutando la prueba e interpretando los resultados de la prueba como operaciones. Apoyar y ejecutar proyectos del sistema.

### **7.3 Estrategia de prueba**

Una estrategia es poder identificar el tipo de prueba y sus características dentro del ciclo de vida del software. Existen varios tipos de modelos de desarrollo, pero el modelo V es uno de los modelos más utilizados.

#### **7.3.1 Pruebas unitarias o de componente.**

Según Schaefer (2014):

Dentro del primer nivel de las pruebas a un nivel menor dentro de la escala V, son las pruebas de una unidad y están basadas en dar una verificación de cada parte del software por separado, en otras palabras, probar cómo funciona parte por parte el código (p.31).

Las unidades de código en este contexto son las mismas que las unidades del programa. Cada unidad debe ser probada porque es una función o método de la clase requerida desde fuera de la unidad y puede requerir otras unidades. Funciona por separado de la unidad de código.

#### **7.3.2 Pruebas de integración.**

El segundo nivel debajo de la estructura del modelo V tiene una parte de integración. Generalmente se hacen las siguientes preguntas: ¿Por qué necesitamos pruebas de nivel 2? Cada componente funciona bien, pero cuando combina los diversos componentes del sistema de software puede hallar errores que no se encontraron inicialmente.

Según el ISTQB (2015) este tipo de exámenes buscan probar las interfaces entre las partes, como lo hace el sistema operativo u otros sistemas.

Existen cuatro estrategias para llevar a cabo las pruebas de integración:

- Integración descendente: Comience con el aspecto de más alto nivel del sistema, enfocándose primero en aumentar el método de desarrollo de software o programa, llamando a otras partes, pero que no sea lo mismo
- Integración ascendente: El enfoque ascendente comienza con los componentes más simples de un sistema que no requieren componentes adicionales.
- Integración Ad Hoc: Las partes se integran en un orden predefinido y, una vez que el componente pasa la prueba del componente, comienza la prueba de integración, verifica si coincide con otra parte que ya ha sido evaluada y comienza la prueba.
- Integración del esqueleto: Esta es una técnica para evaluar cuándo una pieza de software participa regularmente.

### **7.3.3 Pruebas de sistema.**

Conforme al ISTQB (2015) “cuando se haya comprobado cada parte y la forma como se integran en un todo, se puede empezar el tercer nivel de la prueba, que es donde se verifica si el producto o software cumple con los requisitos necesarios”.

En esta etapa, la prueba puede integrar tipos de pruebas basadas en riesgos, casos de uso, entre otros métodos, y validar la interacción con los sistemas operativos y los recursos del sistema. Una verificación del sistema incluye un examen de los requisitos funcionales y no funcionales dentro del programa, así como las características de calidad. Para hacer esto se aplica el método de prueba de caja negra.

### **7.3.4 Pruebas de validación o aceptación.**

Como señala ISTQB (2015) las pruebas anteriores son responsabilidad de quien ejecuta el programa. Las pruebas de aceptación son responsabilidad del cliente y pueden ser parte de las pruebas relevantes.

## **Capítulo VIII**

### **Estudios de la calidad del software educativo**

#### **8.1 Generalidades**

Como indica Marqués (2000) el software, en su mayor parte o en su mayor utilización en particular, son productos que deben suplir un sector del mercado, el cliente final. Por otro lado, el software educativo debe ser de resguardo al aprendizaje, por lo que este producto debe de tener una alta calidad para beneficiar al aprendizaje de los estudiantes.

Existen diseños de calidad para evaluar el software, los que en su mayoría se centran en modelos propuestos como software para el aprendizaje. La definición de software educativo o software de aprendizaje reemplaza su definición como un programa educativo más amplio que el aprendizaje, teniendo en cuenta los elementos y objetivos específicos de la enseñanza del software educativo y considerando otros aspectos como culturales o puramente técnicos.

#### **8.2 Evaluación de la calidad de software educativo**

La evaluación de la calidad de software es una de las actividades más comunes, no solo por personas especializadas, sino por profesionales de otras afinidades. Estas

actividades pueden desarrollarse en cualquier momento como, por ejemplo, durante el desarrollo o una vez ya construido.

El propósito del software difiere de otros factores porque se enfoca en una variedad de problemas y dominios, como el propósito final, la tarea a realizar, la tecnología utilizada y el tipo de usuario satisfecho. Por lo tanto, considerando estas características, es razonable pensar que es necesario un modelo de evaluación que tenga en cuenta estas diferencias.

En la perspectiva del software para el aprendizaje, la evaluación cumple un rol importante, para el cual debe considerar aspectos comunes a la evaluación.

Según Elissavet (2000) la baja calidad del software, entre otros aspectos, afecta directamente en el aprendizaje del estudiante, por tal motivo es crucial considerar que este tipo de programa forme parte de un entorno educativo donde deben considerarse diversos factores, haciendo su evaluación más compleja.

Por su parte, Lage y Cataldi (2000) explica que el software instructivo implementado con una forma que integra aspectos psicoeducativos en un diseño de período de vida del software brinda la oportunidad de aprender conceptos más que un software similar, pero se desarrolló desde una perspectiva que no tiene estas características.

Marqués (2000) un buen ingrediente didáctico a cada momento tendrá mayor perspectiva didáctica que un material de pauperrima calidad.

Reeves (1998) muestra los 14 aspectos educativos del software de enseñanza CBE (Computer Based Education). Usando estas 14 dimensiones, puede identificar las cualidades clave de un producto de software. En otras palabras, puede encontrar métricas que se centran en el área de educación del software solo educativo.

Para Marqués (2002) la evaluación y elección de un programa educativo es un referente con una orientación específica en la pedagogía, donde tiene que contar con criterios que tienen que ser evaluados.

Si hablamos de la perspectiva pedagógica, se puede señalar la variedad didáctica, incentivos, enfoque pragmático y creativo, informando sobre el autoaprendizaje.

### **8.3 Modelo de cuatro dimensiones para software educativo**

Marqués (2002) establece una perspectiva no jerárquica con dos ejes: tecnológico y pedagógico. La forma en que se evalúa gira en torno de tres características: operativa, racional y holística. La perspectiva operativa considera las características de evaluación de cada uno de los tres ejes por separado.

La dimensión holística incluye características de evaluación con relación a aspectos socioculturales y de axioma. En la última dimensión, la factibilidad considera los criterios relacionados con el análisis teniendo en consideración el costo y beneficio asociado a la implementación del programa.

Las métricas relacionadas con el eje tecnológico consideran los materiales utilizados, legibilidad, flexibilidad de cambio, adaptación a diferentes plataformas, etc.

### **8.4 Modelo TUP (Tecnología-facilidad de uso-pedagógicos)**

El autor Bednarick (2002) cree en modelos que enfatizan tres aspectos: evaluación, tecnología, facilidad de uso y educación. Cada uno tiene una lista de verificación para evaluación.

Desde una perspectiva técnica, se consideran varios criterios, incluidas las interacciones y dependencias de hardware y software, y la privacidad. Finalmente, el

aspecto educativo considera criterios para evaluar la calidad, como el contexto de uso y los escenarios.

### **8.5 Evaluación de software educativo con base a la eficiencia pedagógica**

Según Ramirez, Perez y Gamboa (2016) la evaluación es una perspectiva constructiva del aprendizaje, un aspecto muy importante. Esta propuesta hace que la evaluación no comprometa el producto por sí mismo, sino que está centrado en el aspecto del análisis del software en un caso real de uso con respecto a la enseñanza y el aprendizaje, teniendo una reflexión sobre su contribución en términos de efectividad.

Además del contenido de la aplicación, los aspectos educativos, la interfaz gráfica de usuario, la interactividad, las herramientas de exploración y los criterios de facilidad de uso, la evaluación también considera criterios técnicos. La evaluación incluye docentes, estudiantes, productores y profesionales de las Tecnologías de la Información y Comunicación (TIC).

La evaluación debe diseñarse no solo para proporcionar una única evaluación de calidad, sino también para publicar el potencial del material evaluado y las condiciones de uso eficientes. El enfoque se basa en un conjunto de preguntas, criterios relacionados para alcanzar los indicadores, para determinar los criterios evaluados en cada categoría.

### **8.6 Evaluación de software educativo a través de Internet**

Según Chiarani (2014) es importante realizar evaluaciones de calidad del software educativo a través del sitio web. Los aspectos a evaluar se basan en la propuesta de Márquez y se clasifican en técnicos, educativos y funcionales.

Los aspectos técnicos incluyen el equipo informático necesario, los medios considerados y la facilidad de uso. Los aspectos educativos incluyen contenido, objetivos

educativos, bases psicoeducativas de aprendizaje, comunicación, etc. Finalmente, los aspectos funcionales incluyen el aprendizaje, los tipos de interacción, las fortalezas y debilidades del software, la actividad cognitiva y las capacidades de diseño presentadas por el software.

### **8.7 Evaluación de factores para el software educativo**

Como indica Elissavet, el software educativo debe evaluarse técnica y educativamente, además de la viabilidad rentable. Algunos de los factores educativos incluyen motivos de contenido, estructura, gestión del estudiante, adaptación a las diferencias individuales y aprendizaje colaborativo. El diseño de la interfaz incluye aspectos como la interactividad, la navegación, la retroalimentación y el diseño de la pantalla.

La evaluación del software educativo es complicada por la variedad de medios y la cantidad de información disponible, el control del aprendizaje y el potencial de colaboración entre personas y sistemas.

Finalmente, después de usar el software, debe evaluar el aprendizaje del estudiante. Con este fin, proponemos una evaluación de los resultados y el proceso de aprendizaje. Los resultados, la cantidad y la calidad del aprendizaje pueden juzgarse mediante pruebas de rendimiento. La evaluación del proceso de aprendizaje es la facilidad de aprendizaje, la facilidad de uso, la facilidad de memoria y la facilidad de uso medida por errores bajos.

### **8.8 MECSE (Métricas para evaluar la calidad de software)**

Según Abud (2005) un diseño de calidad ISO 9126 que agrega aspectos de educación y contenido. Cada característica se describe en detalle según las características

de las características, la facilidad de uso, la confiabilidad, la eficiencia y la portabilidad.

Los atributos de calificación se ordenan según su importancia en la calidad general.

Cada evaluación de atributo se realiza en una escala de cuatro valores y se pondera para obtener una evaluación final. El equipo propuesto proporciona una visión general de la calidad del software educativo.

### **8.9 Propuesta para la evaluación didáctica de software educativo**

El autor Gonzales (2002) propone un conjunto de criterios para evaluar el software de difusión educativa bajo un enfoque constructivista.

Para fines educativos, es interesante investigar las discrepancias en la valoración de diferentes diseños de software educativo. Esto se debe a que, para fines de evaluación, la diferencia entre software popular y software educativo no es trivial. La perspectiva de mirar uno u otro tipo de software define varios criterios específicos diferentes.

Desde una perspectiva didáctica, las características que deben ser valoradas son:

- Correlación entre el tipo de empresas y aspectos como el idioma, argumentos, organización y cantidad de información, diseño gráfico y sistema de navegación.
- Aprovechamiento de los medios de manera que se maximice el beneficio del uso del computador.
- Calidad de los datos, se determina que esta sea real, acorde al currículo, moderna y presentada correctamente.

## Capítulo IX

### Modelo de calidad de software

#### 9.1 Concepto

Según Callejas (2017) el software es una de las herramientas más útiles para optimizar los procesos de una organización, y su propósito es contar y proporcionar optimización, eficacia y justificación con las necesidades. Por lo tanto, el software tiene la obligación de tener estándares de garantía de calidad.

Dependiendo de esta necesidad, algunas entidades o investigadores pueden admitir estándares, incluidos estándares que pueden respaldar el desarrollo y/o el uso de estrategias modelo, metodologías, guías, productos de software y si realmente existe un nivel de calidad. Evaluar, proponer ciclo de vida.

#### 9.2 Modelos de calidad de software

Para Moszkowitz (2010) las palabras modelo y metodología distan entre sí en significado, pero la metodología permite a las organizaciones realizar autoevaluaciones o autodiagnósticos a través de una revisión sistemática de estrategias y prácticas de gestión.

Para la calidad del software, el modelo debe centrarse en el monitoreo y la evaluación en cada etapa de la construcción del producto de software. Un modelo de

calidad, por otro lado, es un documento que integra la mayoría de las mejores prácticas, sugiere problemas de gestión que cada organización debe enfatizar, integra diversas prácticas dirigidas a procesos clave y mide el progreso declarado en calidad.

En el campo de la construcción de software, los modelos de calidad deben poder evaluar sistemas cualitativa o cuantitativamente, lo que permite a las organizaciones proponer e implementar estrategias que puedan mejorar los procesos durante la fase de análisis. Diseño, desarrollo, pruebas de software.

### **9.3 Modelo a nivel de proceso**

Conforme a Callejas (2017) “un proceso es un conjunto de actividades y recursos interrelacionados que van a transformar elementos de entrega en elementos de salida, de forma que nos aporten un valor agregado para el cliente o usuario” (p.19).

#### **9.3.1 Modelo ITIL.**

Según la Biblioteca de Infraestructura de Tecnologías de la Información (ITIL, 2008) es un cogruppo de términos y buenas destrezas usadas para la organización de ayuda dentro del ámbito de tecnologías.

No es una procedimiento, no da reglas, ni asigna tareas a los personajes de la empresa; solo propone una distribución organizacional compuesta por cuatro principios o pilares:

- Proceso: Alinear el trafico y la gestión de servicios de tecnologías de la información con enfoque en procesos.
- Calidad: Basar en en transcurso de disposiciones y progreso continuo.
- Cliente: Definir al receptor final de la perfección de los servicios.

### 9.3.2 Modelo ISO/IEC 15504.

También denominado Determinación de la capacidad de mejora del proceso de software acortado de SPICE, un modelo para mejorar y evaluar los procesos de desarrollo y mantenimiento de productos de software y sistemas de información.

Proporciona un marco para evaluar procesos y establece los requisitos mínimos para realizar evaluaciones constantes. El estándar actualmente consta de siete partes, que evalúan el nivel de madurez de la organización.

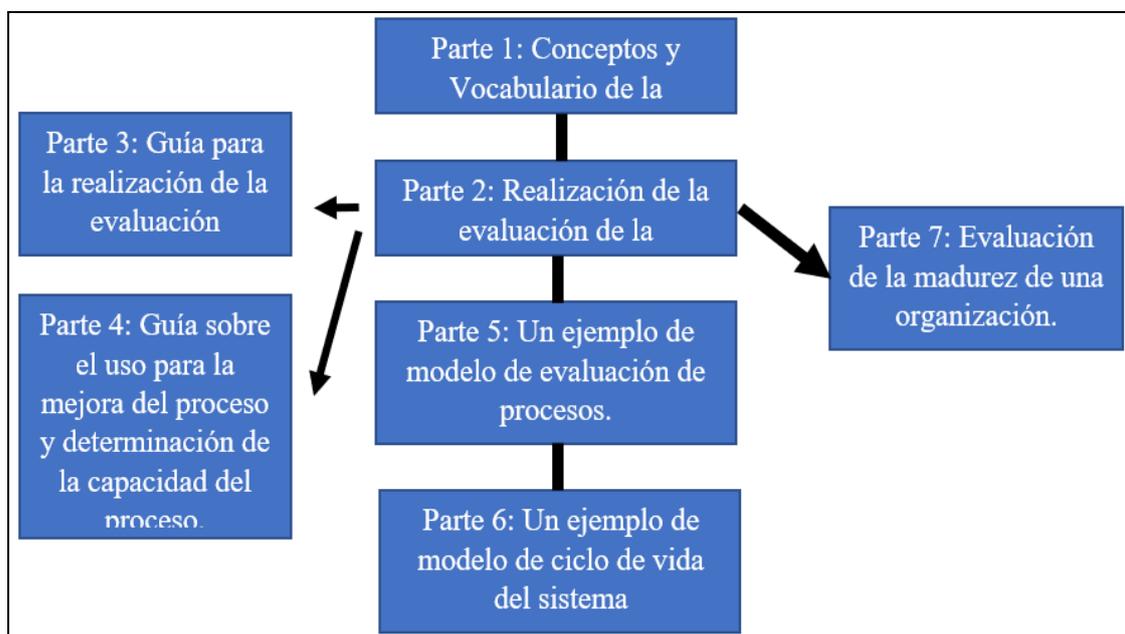


Figura 2. Modelo ISO/IEC 15504. Fuente: Autoría propia.

Existe una variante de ISO 15504-7 que define un marco para determinar la madurez de una organización. Esto le brinda no solo una organización, sino también un puntaje, ya que incorpora la posibilidad de evaluar una organización ISO por nivel de madurez. A nivel de proceso. Establecer cinco diferentes niveles de madurez para catalogar las organizaciones.

- Nivel de madurez 0: La organización no tiene una implementación efectiva del proceso.
- Nivel de madurez 1: El propósito del proceso se logra en un sentido general, y el proceso se realiza según sea necesario, pero no de manera planificada o seguida.

- Nivel de madurez 2: El producto del proceso se adquiere, pero esta vez el seguimiento se realiza de acuerdo con el plan. Estos productos cumplen con las normas y especificaciones prescritas. Los plazos y recursos también están definidos.
- Nivel de madurez 3: Los procesos se ejecutan y gestionan utilizando pasos definidos. Cada implementación del proceso se realiza utilizando procedimientos creados de acuerdo con estándares y documentación.
- Nivel de madurez 4: Se recopilan y analizan medidas detalladas de los niveles de rendimiento del proceso. Esto mantiene el proceso dentro de límites favorables y le brinda un mejor lugar para cuantificar las capacidades del proceso y predecir su comportamiento.
- Nivel de madurez 5: La realización del proceso se optimiza continuamente teniendo en cuenta la contribución al logro de los objetivos comerciales de la organización. Se establecen objetivos cuantitativos para la eficiencia y la efectividad en la realización del proceso, en función de los objetivos comerciales de la organización. Se realiza un monitoreo continuo del proceso y se analizan los datos obtenidos. Esto le permite cambiar dinámicamente los procesos estándar definidos dentro de su organización para adaptarse efectivamente a los objetivos de su empresa y los objetivos futuros.

### **9.3.3 Modelo ISO/IEC 9126.**

También conocido como evaluación de productos de software, proporciona propiedades de calidad y pautas para su uso. Han sido desarrollados para apoyar estas carencias. Menciona seis propiedades de calidad y inculca un modelo de proceso para la evaluación de productos de software.

- Funcionalidad: Dentro de esta característica, tiene que cumplir con una correcta adecuación, un grupo acorde de funciones para actividades y finalidades del usuario

específicos, además de una exactitud teniendo resultados correctos con cierto grado de precisión.

- **Confiabilidad:** El sistema tiene que poseer un grado de madurez, evitando fallos en el software; además, tiene que poseer la capacidad de restaurar un nivel de prestaciones especificadas.
- **Usabilidad:** Debe de tener una exactitud y productividad, permitiendo al usuario desgastar una porción de recursos con finalidad a la realidad alcanzada.

### **9.3.4 Modelo CMMI.**

Según Crosby (1987) el modelo Capability Maturity Model Integration (CMMI) es un conjunto de reglas enfocadas en la calidad en el mundo del software. Este conjunto se aplica a los diferentes procesos dentro del marco de la calidad, es imprescindible mencionar algunas normas como la ISO 90003, debido a que este modelo nos menciona qué hay que hacer y no cómo hacerlo.

Este modelo nos da la posibilidad de mencionar los componentes del diseño y sus relaciones entre los mismos. Está compuesto por cinco niveles:

- En el primer nivel se desarrollan procesos solitarios y no monitoreados; también, no cuenta con un medio estable para dar soporte a otros procesos, haciendo que el futuro éxito depende de otros factores, como podría ser el esfuerzo individual o propiamente del mercado.
- Dentro del nivel 2 multiplicable se programa y se lleva a cabo los procesos relacionados a las políticas establecidas.
- En el nivel tres se describen normas, procedimientos y metodologías que se tengan en cuenta, a la vez que herramientas a utilizar.

- En el nivel cuatro los procesos se gestionan cuantitativamente. Existen criterios para lograr la gestión de procesos y obtener proyectos definibles y medibles de calidad, tanto en el rendimiento del proceso y objetivos medibles que se instruye a las carencias de los clientes.
- Dentro del nivel cinco o nivel de optimización se toma en cuenta los objetivos medibles de mejora dentro de los procesos de la organización, se evalúan constantemente y reflejan cambios dentro de las finalidades del negocio. También se podría utilizar como perspectiva para restablecer el proceso de administración.

### **9.3.5 PSP.**

De acuerdo con Humphrey (1995) el modelo PSP (Process software Personal) fue mencionado inicialmente para uso académico, para los estudiantes, pero tiempo después fue dirigido para profesionales. Este modelo busca dotar un marco de trabajo para el equipo de trabajo involucrado dentro de la creación de un programa de software, se centra también en la administración del tiempo y la calidad eliminada, posibles defectos en un primer momento.

Luego de que Watts S. Humphrey guiara el crecimiento primario de CMM para software, se decidió utilizar los pilares de CMM a los programas de un menor tamaño. Después de eso, un gran sector de gente preguntaba cómo aplicar CMM a las organizaciones pequeñas.

Los principios de CMM se aplicaban a software de gran tamaño. Luego de esto se volvió imprescindible la guía para saber a continuación cómo proceder. En este contexto, Watts S. Humphrey decidió utilizar los principios de CMM para observar si podía enfocar o convencer a los ingenieros de software a que utilizaran tales prácticas. Este modelo consta de 6 principios:

- Cada ingeniero es inherentemente diferente, y los ingenieros deben planificar su trabajo utilizando su información personal.
- Los ingenieros necesitan usar personalmente procesos bien definidos y medidos para mejorar continuamente las operaciones.
- Para desarrollar un producto de alta calidad, los ingenieros deben sentirse personalmente comprometidos con la calidad del producto.
- Encontrar y corregir errores al principio del proyecto es menos costoso que encontrar errores más adelante en el proyecto.

El modelo PSP está diseñado en niveles donde puede ser aplicado en el desarrollo de software, definición de requerimientos, documentación, pruebas de sistema y mantenimiento de sistemas. A continuación, se mostrará un gráfico donde Humphrey plantea la evolución del proceso personal de software.

### **9.3.6 Modelo Team Software Process (TSP).**

Según Humphrey (1996) el modelo TSP es un proceso que se centra en todo lo relacionado con la calidad y las métricas del plan de software. Se desenvuelve una serie de ciclos a través del TSP, cada etapa empieza con un transcurso de elaboración llamado lanzamiento y culmina con un transcurso de terminación.

TSP ayuda a planificar y administrar equipos para proyectos de software y se basa en cuatro principios básicos:

- El aprendizaje será más productivo si sigues ciertos procesos.
- La labor en un conjunto de personas requiere una mezcla de objetivos específicos, un ambiente de trabajo de apoyo y liderazgo y capacitación competentes.
- Evaluar los servicios de las destrezas en desarrollo.
- La capacitación es más efectiva cuando hay información previa disponible.

Este modelo tiene ciertos objetivos, como lo es mejorar o maximizar la calidad de software, incluir equipos solitarios de un gran rendimiento que puedan planificar, registrar y establecer metas.

## **Capítulo X**

### **Garantía de calidad de software**

#### **10.1 Introducción**

Durante la década de los años cincuenta, el software empieza a encontrar su sendero dentro de los sistemas del departamento de defensa de Estados Unidos. Estos proyectos se encontraban indiferentes a una etapa de planificación, se componían básicamente del presupuesto y tenían muchos problemas a nivel técnico.

Normalmente no funcionaban como se suponía y muchos proyectos eran cerrado antes de la etapa final. Durante este periodo las empresas que se contrataban para el desarrollo de software casi siempre hacían estimaciones muy altas sobre la etapa del desarrollo del software.

El software empezó a encontrar la senda dentro de los sistemas del departamento de defensa de Estados Unidos. Para resolver el problema, se ha establecido la validación y validación de interdependencias. Este es un proceso de ingeniería que utiliza una metodología rigurosa para evaluar la precisión y calidad del software a lo largo de su existencia o ciclo de vida.

El primer software que usó esta perspectiva fue el Programa de Misiles de EE.UU. en la década de 1950. A continuación, el proyecto recopiló información que indica que el proyecto que usa IV y V funcionó mejor que el proyecto sin él.

Luego de los años setenta la actividad de desarrollo de software comenzó a agrandarse y las compañías que desarrollaban software fueron investigando otros tipos de mecanismos, pero sin tener resultados de las agencias del gobierno en las primeras décadas. Las compañías tenían problemas para entregar el producto dentro del tiempo pactado, aumentando el presupuesto y la calidad planificada. Varios proyectos desarrollados entre los años 80 y 90 fueron realmente desastrosos, muchos se extendían en las entregas y había otros factores adicionales.

Empezando los años ochenta, esta experiencia se volvió lo que se conoce como la crisis del software, el tiempo que continuó en el mantenimiento se alargaba y se excedía del tiempo pactado en el que se desarrollaba nuevos productos de software.

Cuando terminó esa crisis del Software, finalizando los años ochenta, el Scottish Qualifications Authority (SQA) evolucionó hacia ser una herramienta que las compañías de desarrollo de software necesitaban para identificar de forma inmediata los problemas de calidad en el proceso en el desarrollo.

Según Galin (2003) un conjunto de acciones sistematizadas y planificadas, necesarias para otorgar la evidencia que necesita el proceso de evaluación o mantenimiento de un sistema de software, también debe de cumplir con los requerimientos técnicos.

Para Reifer (2011) la forma como se asegura la calidad de software es teniendo un sistema de métodos usados para que el producto de software alcance los requerimientos. El software involucra ciertas características como planificación, estimación, entre otras.

## **10.2 Actividades de SQA**

El afiancimiento de la calidad del software reside principalmente en una amplia gama de tareas relacionadas con los diversos tipos de ingenieros de software que son responsables de planificar, monitorear y administrar los registros de afiancimiento de la calidad y trabajar juntos técnicamente. Los ingenieros de software enfrentan calidad utilizando diferentes métodos.

El plan de desarrollo a lo largo de la elaboración del proyecto es analizado por las partes relacionadas. Las acciones de garantía de calidad desarrolladas por el equipo de ingeniería de software, el plan se identifica de la siguiente manera: evaluaciones que se podrían desarrollar, conferencias y revisiones a emprender, feedback de la información obtenida y equipos de proyecto para desarrollar un proyecto.

## **10.3 Diferencia de SQA y SQC**

En la mayoría de casos los profesionales del software, cuando conversan de aseguramiento de calidad de software, siempre comienzan a hablar de lo que es netamente la etapa de testeo, algunos incluyen la validación y verificación, luego empiezan a comentar sobre revisiones; en otras palabras, son continuaciones del testing.

Las pruebas y revisiones por sí solas no garantizan la calidad del producto, pero sí garantizan el cumplimiento de las especificaciones de primer plano. Durante el desarrollo de software, las diferencias entre SQC y SQA no están completamente definidas, y estos términos a menudo se confunden.

El papel de SQA es proporcionar una auditoría que tenga en cuenta los diversos equipos de la organización, incluido SQC, siguiendo los procedimientos, estándares y procesos establecidos.

#### **10.4 Funciones generales del SQA**

Una de las responsabilidades del equipo de SQA es asegurar que el proceso de desarrollo continúe con las funciones establecidas. Entre sus funciones, en este rol se encuentran:

- Generar debate sobre los productos del trabajo para localizar carencias.
- Establecer el cumplimiento de la hoja de ruta del proyecto.
- Emitir un juicio de valor del proceso y no del producto.
- Localizar la funcionalidad que al cliente le resulte satisfactorio.
- Ayudar a la empresa a tener conciencia sobre las necesidades del cliente.
- Interactuar como un cliente para evaluar si se obtiene una alta satisfacción.
- Recolectar todos los datos posibles sobre todas las características del producto y del proceso.
- Suministrar información técnica para que la empresa pueda usarla para fines administrativos.

#### **10.5 Aseguramiento de la calidad de procesos y productos en CMMI**

La calidad del producto es una función de la calidad durante el proceso utilizado en la fabricación. Como resultado, las organizaciones deben continuar mejorando la forma en que manejan los proyectos de creación de software. Las fases del proceso incluyen la categoría de soporte CMMI Nivel 2 PPQA (Garantía de calidad de producto y proceso). Este es el sector de proceso inicial o clave para el aseguramiento de la calidad del software.

La razón para garantizar la calidad del proceso y del producto es proporcionar a los equipos y empresas relevantes las perspectivas relevantes del proceso y del producto. El objetivo principal de PPQA es garantizar que la empresa respalde el proceso definido.

Los objetivos de esta área de proceso son:

- Evaluar fehacientemente la forma en cómo se llevan a cabo los procesos, los elementos de los servicios y trabajos.
- Localizar y documentar si existen inconformidades. Evoquemos que definimos una no conformidad con el incumplimiento de una norma de la empresa o del proyecto.
- Solventar información a los integrantes del equipo que están utilizando los procesos y a la gerencia del proyecto.
- Verificar que las inconformidades puedan ser manejadas.

Se presenta un listado de funcionalidades para satisfacer esta área del proceso:

- Terminar las plantillas en base al cumplimiento de los procesos definidos en la empresa dentro del programa de control de documentación.
- Originar la llamada a una auditoria de calidad dentro de las actividades.
- Permitir el registro llevando a cabo una auditoria de calidad.
- Registrar las no conformidades dentro del sistema de gestión de incidencias.

Dependiendo de cómo se implementen estos objetivos y prácticas, hay diferentes formas de desarrollar la madurez de una organización en el desarrollo de software para CMMI. Certificado en procesos y actividades de control de calidad, garantizando niveles de calidad estándar de productos y servicios para los clientes.

## Aplicación didáctica

### Sesión de aprendizaje: “CONOCIENDO EL CICLO PHVA”

#### DATOS INFORMATIVOS

**ÁREA CURRICULAR:** Educación para el trabajo

**TIEMPO:** 90 minutos

**DOCENTE:** Maguiña Pastrana, David

**TEMA TRANSVERSAL:** Practicando y fortaleciendo valores para una mejor calidad de vida.

**EXPECTATIVA DE LOGRO:** Reconoce el ciclo PHVA y su utilidad considerando la práctica de valores para mejorar su calidad de vida.

#### ORGANIZACIÓN DE LOS APRENDIZAJES

Contenido Diversificado	Aprendizajes Esperados	Valores/actitudes	Indicadores
PREVIOS  ¿Qué es calidad?  ¿Qué es un ciclo?	Reconoce las etapas del ciclo PHVA (Planear, Hacer, Verificar y Actuar) para elaborar un plan de acción.	Responsabilidad, respeto, deseo de superación, laboriosidad.  Culmina eficazmente la practica en clase.  Demuestra respeto durante su trabajo en el aula  Se esmera en mantener atención en clase.	Identifica las etapas del ciclo PHVA  Realiza procedimientos para crear un esquema aplicando el ciclo PHVA.

## SECUENCIA DIDÁCTICA

Situación de aprendizaje	Estrategias didácticas	Tiempo	Medios y/o materiales	Instrumento de evaluación
<b>Inicio</b>  <b>Motivación</b>	<ul style="list-style-type: none"> <li>El docente realiza una actividad para organizar los grupos de trabajo dentro del laboratorio de computo.</li> <li>Se motiva para el orden y limpieza en el laboratorio.</li> </ul>	5 min		Observación sistemática
<b>Recuperación de saberes previos</b>	<ul style="list-style-type: none"> <li>Se realiza una secuencia de lluvia de ideas relacionado al termino de calidad y un ciclo, lo cual servirá para tener claramente definido lo que se conoce como ciclo PHVA.</li> <li>¿Qué entiendes por calidad?</li> <li>¿Qué es un ciclo?</li> <li>¿Es necesario tener un orden al momento de realizar un trabajo determinado?</li> </ul>	10 min	PC  Pizarra  Plumones  Material Impreso (guía)	Lista de cotejo
<b>Conflicto cognitivo</b>	<ul style="list-style-type: none"> <li>¿Qué beneficios tendría aplicar una estrategia al realizar un trabajo?</li> <li>¿Es necesario establecer un orden durante el proceso de un trabajo?</li> </ul>	5 min	PC	
<b>Proceso</b>  <b>Procesamiento de la nueva información</b>	<ul style="list-style-type: none"> <li>El docente aplica dinámicas interactivas grupales para promover el conocimiento y la integración en equipo.</li> <li>El docente apertura el dialogo sobre el tema “Ciclo PHVA” y da indicaciones sobre sus etapas, concepto, creación y aplicación.</li> <li>Utilizando la guía de trabajo, se les</li> </ul>	40 min	PC  Pizarra  Plumones  Material Impreso (Guía)	

	<p>explica cada etapa del ciclo PHVA.</p> <ul style="list-style-type: none"> <li>• Se identifica los pasos para su elaboración.</li> <li>• Utilizando el grupo creado, elaboran el ciclo PHVA sobre un producto para el día del logro.</li> </ul>			
<b>Transferencia</b>	<ul style="list-style-type: none"> <li>• Haciendo uso del proceso anterior, realizan el mismo procedimiento para aplicarlo a una actividad cotidiana.</li> </ul>	20 min	Material Impreso (Guía)	
<b>Cierre</b>	<ul style="list-style-type: none"> <li>• ¿Qué hemos aprendido hoy?</li> </ul>	10 min		
<b>Evaluación (metacognición)</b>	<ul style="list-style-type: none"> <li>• ¿Cómo lo hemos aprendido?</li> <li>• ¿Para qué lo he aprendido?</li> </ul>			
<b>Extensión</b>	<ul style="list-style-type: none"> <li>• Averigüe qué otras utilidades tiene el ciclo PHVA.</li> </ul>			

#### Referencias:

Castro, A. (2007). Gerencia de proyectos en un SGSI que se encuentra en producción. Recuperado de

[http://www.escolme.edu.co/almacenamiento/oei/tecnicos/ppios\\_admon/contenido\\_u3\\_2.pdf](http://www.escolme.edu.co/almacenamiento/oei/tecnicos/ppios_admon/contenido_u3_2.pdf)

Peña, L. (2017). propuesta de mejoramiento del sistema de gestión de calidad a partir de la revisión de procesos institucionales del Colegio Andres Escobar, Bogotá, Colombia.

Recuperado de

<http://repository.udistrital.edu.co/bitstream/11349/5652/1/Pe%C3%B1aCaraballoLeidyConstanza2017.pdf>

TOPPUNTO (2015) El ciclo PHVA Ejemplo de aplicación de esta herramienta. Recuperado de

<http://www.blog-top.com/el-ciclo-phva-ejemplo-de-aplicacion-de-esta-herramienta-de-calidad/>

## Guía de trabajo: CICLO PHVA (Planear)

El ciclo PHVA, nombrado como ciclo Deming. En la lengua española sería PHVA (Planificar-Hacer-Verificar-Actuar); además, podemos observarlo de las dos formas sometiéndose del documento que estemos consultando, es una táctica de mejora continua de la calidad de un producto o servicio en cuatro pasos.

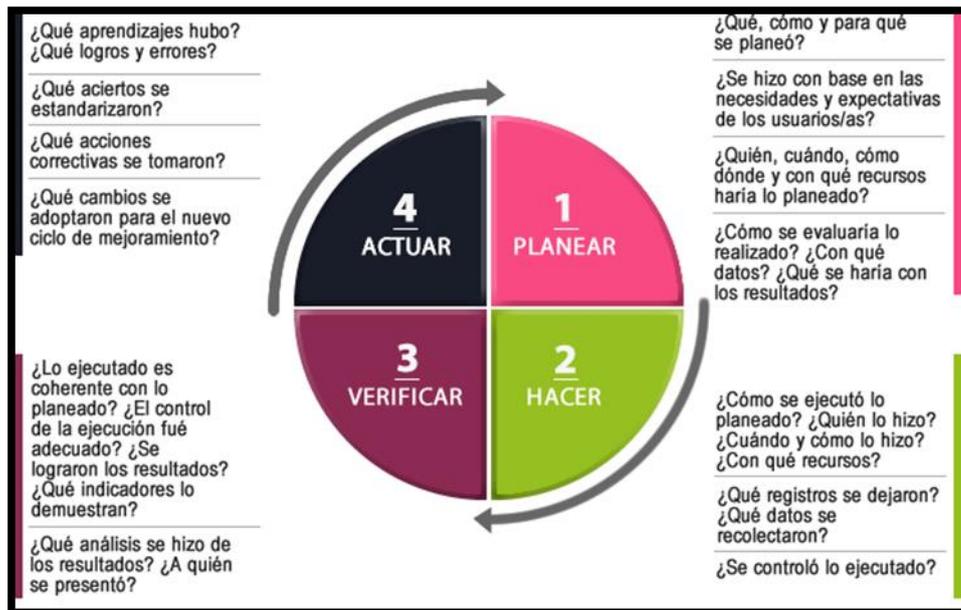


Figura 3. Ciclo PHVA. Fuente: Autoría propia.

- El primer paso (Planear) se establece los objetivos y se identifican los procesos necesarios para lograr los resultados.

Ejm: “Proyecto Chocotejas”

Paso 1: Planear

- Se necesita chocolate bitter 100 g, manjar blanco y pecanas.
- Se elaborará 20 unidades.
- Se utilizará un molde de flores y de corazones
- Se venderá a S/ 1.50 cada unidad
- Se elaborará una encuesta al consumidor.

### Registro Auxiliar de evaluación

LISTA DE COTEJO				UNIDAD		
				SESIÓN		
Título	"CONOCIENDO EL CICLO PHVA"					
Indicador	Identifica las etapas del ciclo PHVA					
Grado:	Sección:		Fecha:			
N°	Estudiantes	Indicadores				Puntaje
		Ingresar correctamente a M.S Word (0-2)	Analiza adecuadamente la guía de trabajo (0-4)	Identifica las etapas del ciclo PHVA (0-5)	Crea en M.S. Word las etapas del proyecto de un producto (0-5)	
1.						
2.						
3.						
4.						
5.						
6.						
7.						
8.						
9.						
10.						
11.						
12.						
13.						
14.						
15.						
16.						
17.						
18.						
19.						
20.						

## Síntesis

Lo que se ha visto en este trabajo monográfico son unas bases simplificadas del concepto de calidad y cómo se aplica al software. De ello nos viene a la mente la idea de evolución del término y la necesidad de diferenciarla en el ámbito de la informática. El estadio actual de la industria del software requiere la aplicación de los estándares y de los modelos y propuestas que se han venido desarrollando en el mundo. Perú, a través de la empresa “Sistemas y Software Consultores”, viene implementando el modelo CMMI, que se ha explicado en este trabajo.

El desarrollo de estándares y modelos de calidad en el mundo lleva varios años de desarrollo, debido a que nada que tenga que ver con informática es estático; todo está en una constante evolutiva de cambio.

### **Apreciación crítica y sugerencias**

Es primordial que, en las instituciones educativas e institutos en el área de educación para el trabajo o Informática, se tome en cuenta para la malla curricular el tema de calidad de software. Debido a su importancia en el contexto en el que vivimos, donde somos los protagonistas del consumismo, compramos una cantidad de productos día a día y no le damos mucha relevancia al producto en cuestión.

En las instituciones educativas se debe de tratar el tema para el desarrollo de los jóvenes del futuro, donde a través de algunas metodologías, como lo es el ciclo PHVA, puedan aplicarlo en su vida diaria, como una especie de estilo de vida para una mejor calidad a través de los años como ser humano.

Dentro del aula e institución deben organizarse actividades que promuevan el uso de insumos de calidad que hayan sido certificados como proceso, y no solo a nivel de Informática sino también alimentario.

Por último, hacer que el estudiante y futuro ciudadano sea consciente de la importancia de la calidad, siendo responsables de lo que consumimos, ya sea un producto o servicio.

## Referencias

- Andersen, A. (1995). *Factores humanos de la calidad.*, Madrid: Prentice Hall.
- Bednarick, R. (2002). *Development of the TUP Model-Evaluating Educational Software.*  
Recuperado de  
[https://www.researchgate.net/publication/4104863\\_Development\\_of\\_the\\_TUP\\_model\\_-\\_Evaluating\\_educational\\_software](https://www.researchgate.net/publication/4104863_Development_of_the_TUP_model_-_Evaluating_educational_software)
- Cavano, J. y McCall, J. (1978). *A Framework for the Measurement of Software Quality.*  
Recuperado de [https://people.dsv.su.se/~joco2917/ft\\_gateway11.cfm.pdf](https://people.dsv.su.se/~joco2917/ft_gateway11.cfm.pdf)
- Covella, G. (2005). *Medición y evaluación de calidad en uso de aplicaciones web.*
- Dijkstra, E. (2015). *¿Qué es la prueba de software?* Recuperado de  
[https://es.wikipedia.org/wiki/Edsger\\_Dijkstra](https://es.wikipedia.org/wiki/Edsger_Dijkstra)
- Figuerola, A. (2005). *Conjunto de Metricas para Evaluar Software Educativo [MECSE].*  
Recuperado de  
<https://www.repositoriodigital.ipn.mx/bitstream/123456789/5329/2/39-2.pdf>
- Galín, D. (2004). *Software Quality Assurance From Theory to Implementation.*  
Recuperado de  
<http://desy.lecturer.pens.ac.id/Manajemen%20Kualitas%20Perangkat%20Lunak/ebook/Software%20Quality%20Assurance%20From%20Theory%20to%20Implementation.pdf>
- González, A. (2002). *¿Qué observar cuando se evalúa el software?* Recuperado de  
<https://dialnet.unirioja.es/servlet/libro?codigo=6825>
- Mauro, A. (2017). *Modelos de calidad de software, un estado de arte.* Recuperado de  
<http://www.scielo.org.co/pdf/entra/v13n1/1900-3803-entra-13-01-00236.pdf>

- Moszkowitz, A. (2010). *Modelos de excelencia en la gestión*. Revista de Antiguos Alumnos del IEEM. Recuperado de <https://dialnet.unirioja.es/servlet/revista?codigo=11592>
- Myers, G. (2004). *El arte del testeo de software* (2da ed.) Recuperado de [http://barbie.uta.edu/~mehra/Book1\\_The%20Art%20of%20Software%20Testing.pdf](http://barbie.uta.edu/~mehra/Book1_The%20Art%20of%20Software%20Testing.pdf)
- Noguez, V. (2004). *ISO 9001:2015 El futuro de la calidad*. Recuperado de <https://www.isotools.org/pdfs-pro/e-book-9001-2015-futuro-calidad.pdf>
- Organización Internacional de Normalización 9001:20015 (2015). *Sistemas de gestión de la calidad – Requisitos*. Recuperado de bps: <https://www.bps.gub.uy/bps/file/13060/1/normativa-internacional-iso-9001.2015.pdf>
- Pérez, I. (2007). *Construcción y Validación del cuestionario: Métrica de Calidad de Credibilidad e Interacción de Cursos de Teleformación*. Recuperado de [https://www.researchgate.net/publication/267832825\\_Construccion\\_y\\_Validacion\\_del\\_cuestionario\\_Metrica\\_de\\_Calidad\\_de\\_Credibilidad\\_e\\_Interaccion\\_de\\_Cursos\\_de\\_Teleformacion](https://www.researchgate.net/publication/267832825_Construccion_y_Validacion_del_cuestionario_Metrica_de_Calidad_de_Credibilidad_e_Interaccion_de_Cursos_de_Teleformacion)
- Pressman, R. (2010). *Ingeniería de software Un enfoque practico* (7ma ed.) Recuperado de <http://cotana.informatica.edu.bo/downloads/ld-Ingenieria.de.software.enfoque.practico.7ed.Pressman.pdf>
- Reifer, D. (2011). *Software Maintance Success Recipes*. Recuperado de [http://www.freebooks.world/getfile.pdf?q=software\\_maintenance\\_success\\_recipes\\_by\\_reifer\\_donald\\_j\\_2011\\_hardcover](http://www.freebooks.world/getfile.pdf?q=software_maintenance_success_recipes_by_reifer_donald_j_2011_hardcover)
- Reeves, T. (1998). *Evaluating What Really Matter in Computer-Based Education*  
Recuperado de

<https://www.eduworks.com/Documents/Workshops/EdMedia1998/docs/reeves.htm>

1

Schaefer, H. (2014). *Software Testing foundations* (4ta ed.) Recuperado de

[http://prof.mau.ac.ir/images/Uploaded\\_files/Software%20Testing%20Foundations%20A%20Study%20Guide%20for%20the%20Certified%20Tester%20Exam\[5309302\].PDF](http://prof.mau.ac.ir/images/Uploaded_files/Software%20Testing%20Foundations%20A%20Study%20Guide%20for%20the%20Certified%20Tester%20Exam[5309302].PDF)

Software Testing Qualifications Board (2015). *Manual-ISTQB Foundation – SSTQB*.

Recuperado de [https://people.dsv.su.se/~joco2917/ft\\_gateway11.cfm.pdf](https://people.dsv.su.se/~joco2917/ft_gateway11.cfm.pdf)

Tari, J. (2000). *Calidad Total: fuente de ventaja competitiva*. Recuperado de

<http://www.biblioteca.org.ar/libros/133000.pdf>

Watts, S. (1990). *Managing the Software Process*. Recuperado de

<http://docshare03.docshare.tips/files/28289/282899986.pdf>