

Modelos y estándares de calidad del software

Calidad en el desarrollo de Software

Tabla de contenido

- 1. Introducción
- 2. Mapa conceptual
- 3. Definiciones
 - 3.1. ¿Qué es calidad?
 - 3.2. ¿Qué es calidad de software?
 - 3.3. ¿Qué es modelo?
 - 3.4. ¿Qué es estándar?
- 4. Modelos y estándares de calidad de software
- 5. Modelos de calidad de software
 - 5.1. A nivel de producto
 - 5.1.1. Modelo McCall
 - 5.1.2. Modelo Boehm
 - 5.2.1. A nivel de proceso
 - 5.2.2. CMMI (Capability Maturity Model Integration)
- 6. Estándares de calidad de software
 - 6.1. A nivel de producto
 - 6.1.1. Estándar ISO 9126
 - 6.1.2. Estándar ISO/IEC 25000:2005
 - 6.1.3. Estándar ISO/IEC 25000:2011
 - 6.2. A nivel de proceso
- 6.2.1. Estándar ISO 15504 - SPICE
- 6.2.2. Estándar ISO 9001-2015
- 7. Glosario
- 8. Referencias bibliográficas

Calidad en el desarrollo de Software

1. Introducción

En esta actividad de aprendizaje se abordarán los modelos y estándares de calidad del software. Inicialmente se realizará un recorrido conceptual de términos básicos, para posteriormente analizar referentes y pautas de calidad a nivel de producto y de proceso.

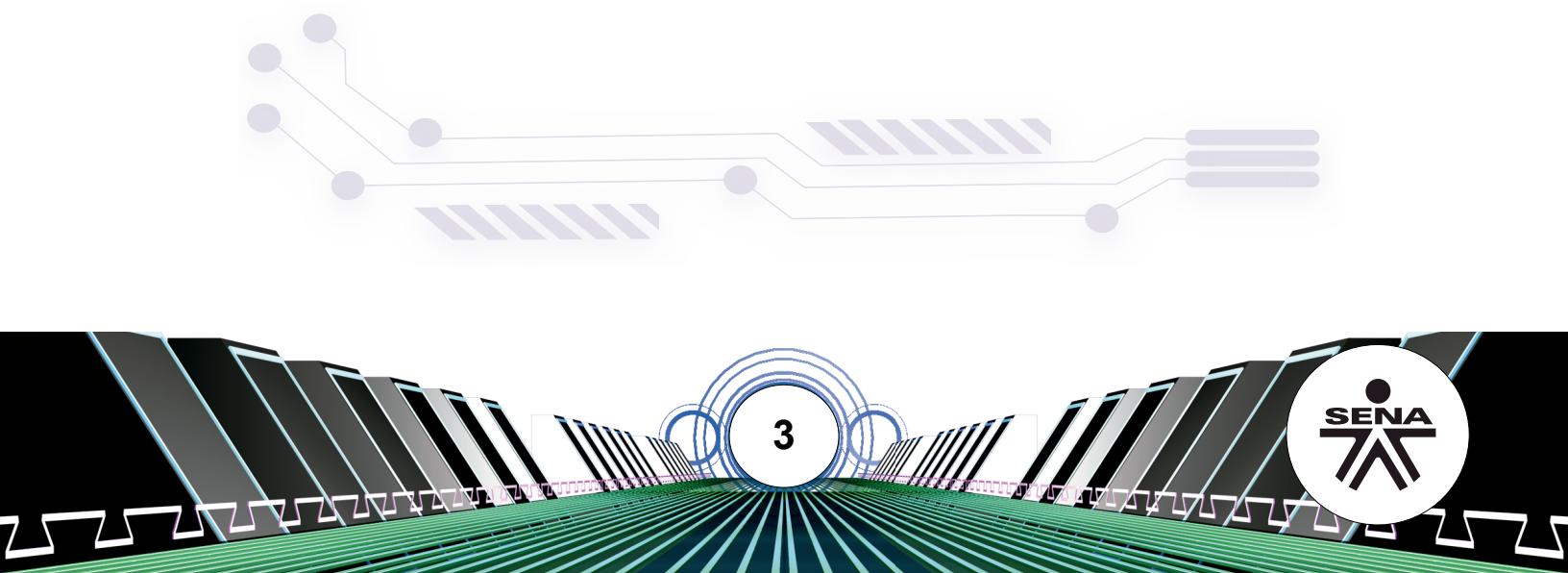
Durante el estudio del presente material se reconocerán las características de los modelos McCall, Boehm y CMMI; y se analizarán estándares de calidad vigentes, como ISO 9126, ISO/IEC 25000:2005, ISO/IEC 25000:2011, ISO 15504 – SPICE y ISO 9001-2015.

De tal manera, se buscará que el Aprendiz logre competencias cognitivas para identificar y clasificar los actuales modelos y estándares que rigen la calidad del desarrollo de software.

La calidad está dada en “La concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente” (Pressman, 2002). En esta definición, la calidad de un producto es directamente proporcional al cumplimiento de los requisitos del mismo.

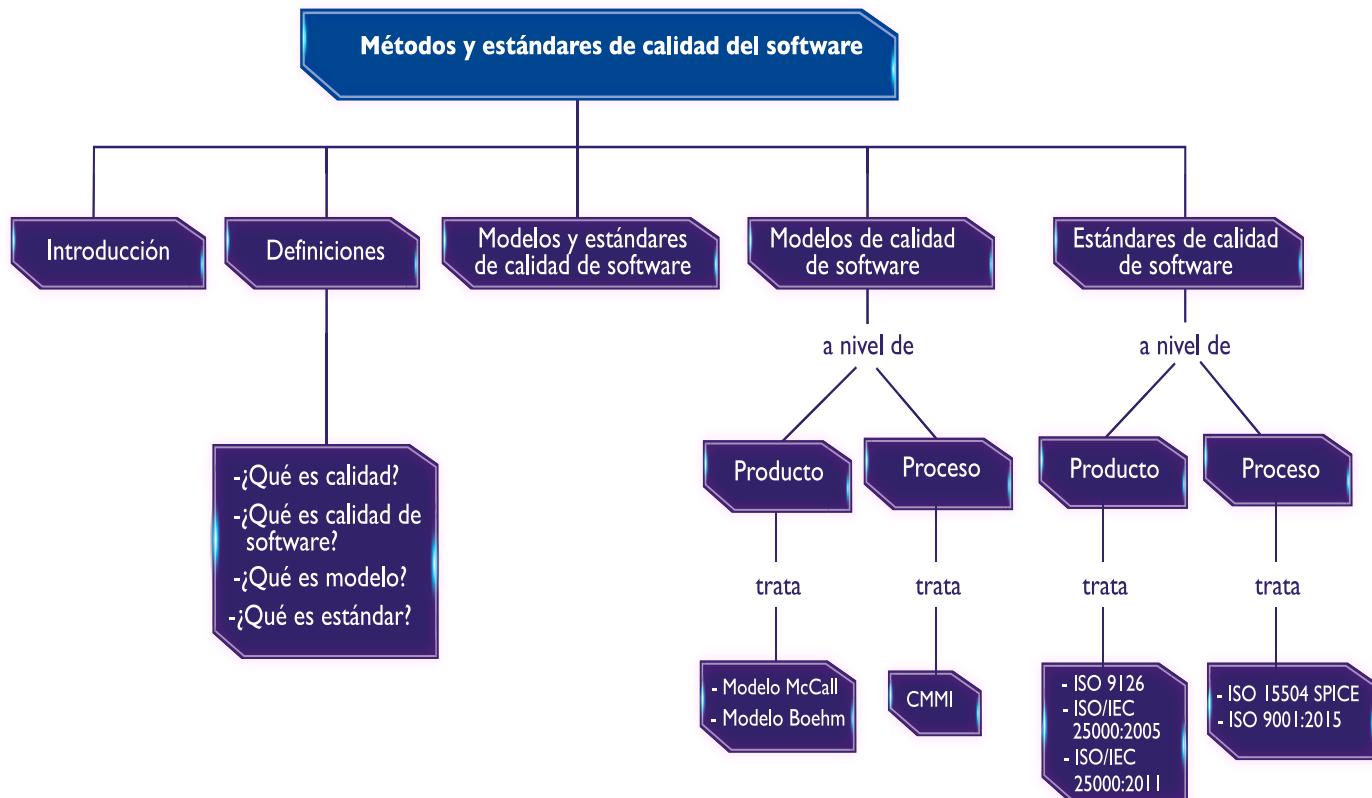
En tal sentido, para lograr la excelencia, los desarrolladores empresariales o independientes deben tener en cuenta varias condiciones, una de ellas, la implementación de buenas prácticas y procedimientos para lograr la totalidad de características que satisfagan las necesidades explícitas o implícitas del cliente, según el estándar ISO/IEC DEC 9126.

Para Vega, Rivera & García (2008), el software tiene calidad si cumple o excede las expectativas del usuario en cuanto a: “Funcionalidad (que sirva a un propósito), ejecución (que sea práctico), confiabilidad (que haga lo que debe), disponibilidad (que funcione bajo cualquier circunstancia), apoyo, a un costo menor o igual al que el usuario está dispuesto a pagar” (p. 133).



Calidad en el desarrollo de Software

2. Mapa conceptual



Fuente: SENA

3. Definiciones

Con el fin de contextualizar las temáticas del presente material de estudio, se ha realizado una explicación de los términos de calidad, calidad de software, modelo y estándares.

3.1. ¿Qué es calidad?

El término puede ser ambiguo e incluso subjetivo porque, como la belleza, la calidad depende de quien la observa. Es necesario definir el concepto con claridad, “Pues si la calidad no puede ser definida, no puede ser medida; y donde la calidad no puede ser medida, entonces no puede ser controlada” (Vega, Rivera & García, 2008, p. 145).

Al revisar el diccionario de la Real Academia Española (RAE), la definición del término “calidad” se refiere al conjunto de propiedades inherentes que permiten juzgar el valor de algo, a la superioridad o búsqueda de la “excelencia”; o a las características especificadas para el control de calidad de un producto.

Calidad en el desarrollo de Software

En este sentido, la calidad de un producto de software, se logra a partir de la aplicación de propiedades, requisitos y características, (especificadas para un producto final y su proceso de desarrollo), contenidas en modelos y estándares de calidad, como referentes.

3.2. ¿Qué es modelo?

La definición del término “modelo” por la RAE, corresponde a arquetipo o punto de referencia para imitar o reproducir. También se refiere a un “esquema teórico”, que explica o expone un sistema o realidad compleja, para facilitar su comprensión y estudio.

También se puede comprender el término modelo como un prototipo que sirve de referencia o ejemplo para quienes diseñan productos. La palabra proviene del italiano *modello*.

3.3. ¿Qué es calidad de software?

Para hablar de calidad de un producto, deben tenerse en cuenta los aspectos que la garantizan y la conducción de procesos que permiten alcanzar resultados idóneos, según indicadores esperados.

“La calidad de software, es el proceso eficaz de software que se aplica para crear un producto útil, proporcionando valor medible a quienes lo producen y a quienes lo utilizan” (Pressman, 2010, p.294).

Según la norma ISO/IEC DEC 91260, la calidad es la totalidad de características de un producto de software, que tienen como habilidad satisfacer necesidades explícitas o implícitas del cliente.

Para al IEEE (Institute of Electrical and Electronics Engineers), calidad es el grado con el que un sistema, componente o proceso cumple con los requisitos especificados y las necesidades o expectativas del cliente o usuario.

En el texto ¿Qué significa calidad de software? (Gómez, 2009), se cita a Mike Bria, con la siguiente definición: “Calidad se refiere a ausencia de defectos en lugar de presencia de valor” (p.22).

3.4. ¿Qué es estándar?

La Real Academia Española (RAE), define el término como: tipo, modelo, norma, patrón o referencia. En términos de calidad de software, el estándar corresponde a un conjunto de guías o patrones a seguir como referente, para lograr el desarrollo idóneo de un producto. Es una construcción colectiva, que se adopta por consenso, a partir de la legitimidad o validez que otorga una o varias entidades acreditadas.

Calidad en el desarrollo de Software

4. Modelos y estándares de calidad del software

A continuación, presenta un cuadro clasificatorio de modelos y estándares, según el proceso y producto.

Tabla 1.

Modelos y estándares de calidad del software

Nivel de calidad	Modelo de calidad del sw	Estándar de calidad del sw
Proceso	CMMI	ISO 90003
	TickIT	ISO 12207
	Bootstrap	ISO 15504 (SPICE)
	Personal SW Process PSP	IEEE / EIS 12207
	Team SW Process TSP	ISO 20000
	Practical SW Measurement PSM	ITIL
Producto	Six Sigma for Software	COBIT 4.0
	Gilb	ISO 9126 – 1
	GQM	ISO 25000 - SQUARE
	Mc Call	IEEE Std 1061 - 1998
	Furps	
	Boehm	
	SATC	
	Dromey	

Fuente: Scalone (2006)

5. Modelos de calidad de software

Los modelos de calidad son aquellos documentos que integran la mayor parte de las mejores prácticas, proponen temas de administración en los que cada organización debe hacer énfasis, integran diferentes prácticas dirigidas a los procesos clave y permiten medir los avances en calidad (Scalone, 2006).

En los últimos años, la industria de desarrollo de software se ha preocupado por mejorar sus procesos, lo que se evidencia en las inversiones que realizan con el fin de abanderar el mercado en este ramo.

Por tal motivo, se han definido varios modelos basados en las experiencias exitosas de la ingeniería de software, que sirven de guía para las mejoras y unifican los criterios de evaluación de las empresas.

Calidad en el desarrollo de Software

Las normas ISO de serie 9000, el modelo estadounidense conocido como CMM (Capability Maturity Model), el BOOTSTRAP (Estándar Europeo para Evaluación y Mejoras de Procesos de Desarrollo de Software) y la norma ISO 15504, conocida como SPICE, (Software Process Improvement and Capability determination) son los ejemplos más reconocidos de estos modelos. (Vega, Rivera & García, 2008).

5.1. A nivel de producto

Con el objetivo de garantizar resultados idóneos en el desarrollo de productos de software, se hace necesario seguir pautas establecidas según modelos de calidad.

5.1.1 Modelo McCall

Desarrollado para la fuerza aérea de los EE.UU en 1977 por Jim McCall, este modelo “Busca reducir la relación entre usuarios y desarrolladores, teniendo en cuenta factores de calidad donde estén presentes las relaciones de ambos” (Estayno, Dapozzo, Cuenca & Greiner, 2009).

Su fin consiste en especificar los requisitos de calidad del producto software al comenzar el proyecto y en cada etapa del ciclo de vida. Se hace necesario evaluar cada factor de calidad estableciendo valores deseables y al finalizar el desarrollo del proyecto, se comprueba si lo establecido inicialmente se cumplió a cabalidad. El modelo de McCall organiza los factores de calidad en tres ejes o puntos de vista, desde los cuales el usuario puede observar la idoneidad del producto. Estos son:

- Operación del producto
- Revisión del producto
- Transición del producto

Cada eje se descompone en once factores y estos a su vez contienen una serie de criterios o propiedades determinantes de la calidad, que pueden ser evaluados mediante un conjunto de métricas. Evaluación que requiere fijar valores máximos y mínimos.

Antes de comenzar a utilizar el modelo de McCall, es necesario seguir las siguientes pautas:

1. Se aceptan los factores, criterios y métricas que propone el modelo.
2. Se aceptan las relaciones entre factores y criterios, y entre criterios y métricas.
3. Se selecciona un subconjunto de factores de calidad sobre los que se aplican los requisitos de calidad establecidos para el proyecto.

Calidad en el desarrollo de Software



Figura 1. Atributos del Modelo McCall

Fuente: Pressman (2010)

De acuerdo con la visión del usuario y sus factores de calidad asociados (visión de la dirección), se puede determinar la visión del desarrollador para cada factor de calidad establecido.



Calidad en el desarrollo de Software

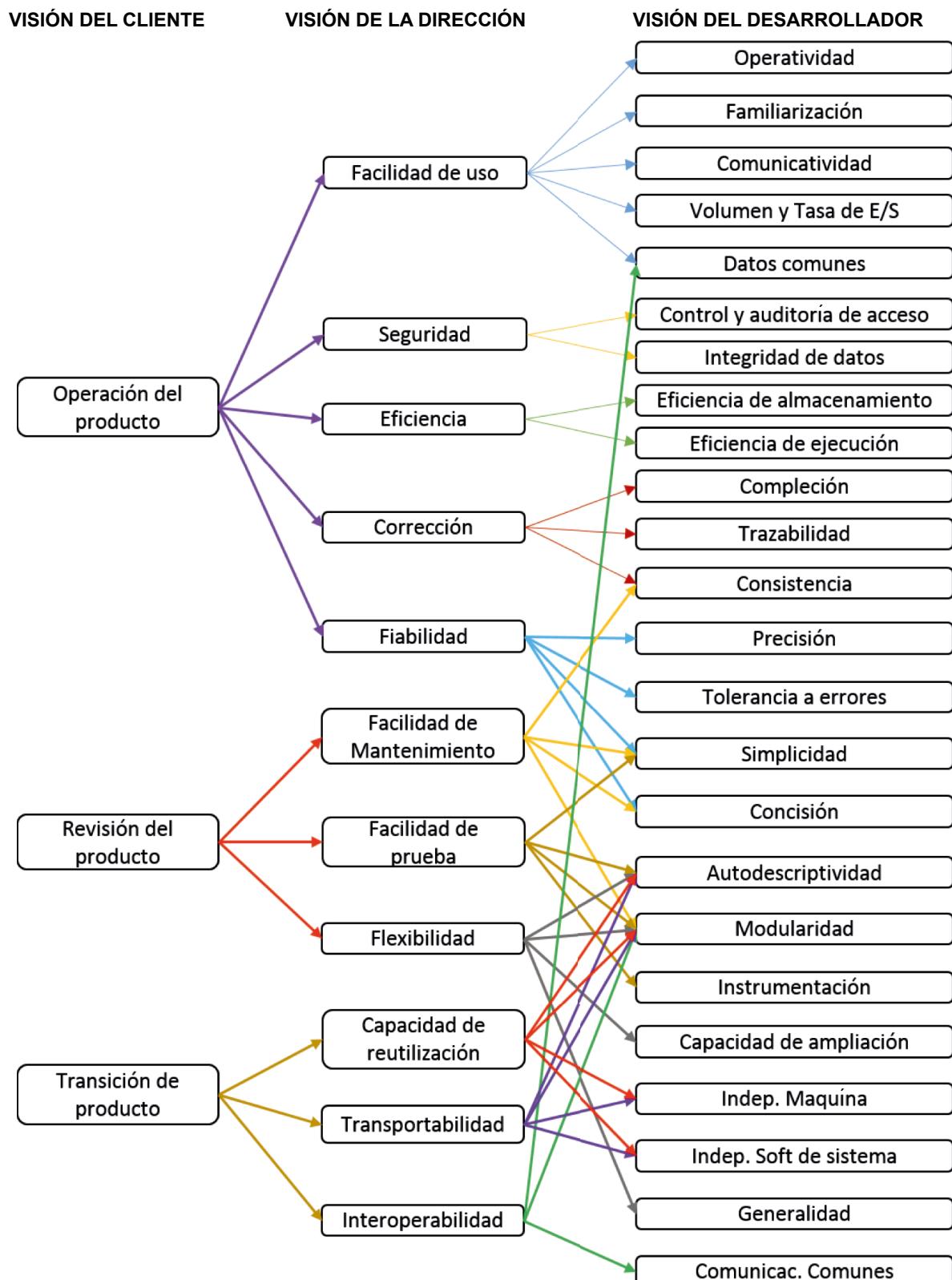


Figura 2. Factores de Calidad – Modelo McCall

Fuente: SENA

Calidad en el desarrollo de Software

5.1.2. Modelo Boehm

Presentado por Barry Boehm en 1978, se considera el segundo más reconocido, después de McCall. Introduce características de alto nivel, nivel intermedio y nivel primitivo.

Características de alto nivel

Representan requerimientos generales de uso, estas pueden ser:

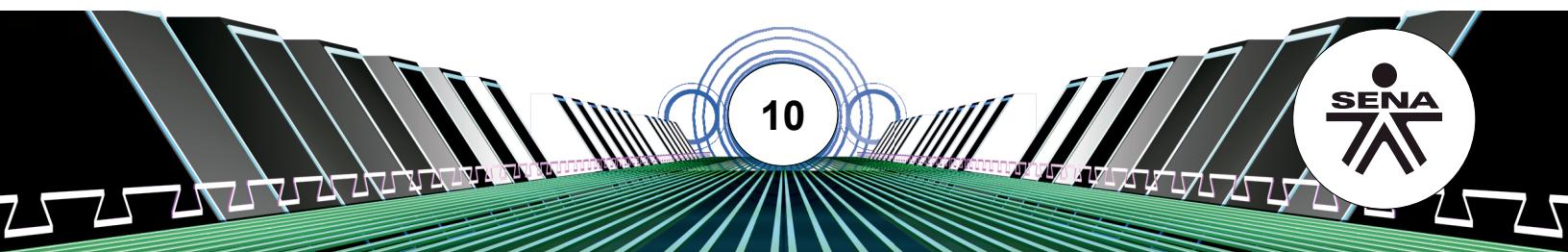
- Utilidad per-se, ¿qué tan usable, confiable, eficiente? es el producto en sí mismo
- Mantenibilidad, ¿qué tan fácil es modificarlo, entenderlo y re-testearlo?
- Utilidad general, si puede seguir usándose, si se cambia el ambiente

Características de nivel intermedio. Representan los factores de calidad:

- Portabilidad (Utilidad General)
- Confiabilidad (Utilidad *Per se*)
- Eficiencia (Utilidad *Per se*)
- Usabilidad (Utilidad *Per se*)
- Testeabilidad (Mantenibilidad)
- Facilidad de entendimiento (Mantenibilidad)
- Modificabilidad o flexibilidad (Mantenibilidad)

Características primitivas

El nivel más bajo corresponde a características directamente asociadas a una o dos métricas de calidad.



Calidad en el desarrollo de Software

Tabla 2.

Características primitivas – modelo Boehm

De portabilidad	Independencia de dispositivos Auto contención
De confiabilidad	Auto contención Exactitud Completitud Consistencia Robustez / Integridad
De eficiencia	Accesibilidad Eficiencia de uso de dispositivos
De usabilidad	Robustez / Integridad Accesibilidad Comunicación
De testeabilidad	Comunicación Auto descripción Estructuración
De entendibilidad	Consistencia Estructuración Concisión Legibilidad
De modificabilidad	Estructuración Automentabilidad

Fuente: SENA

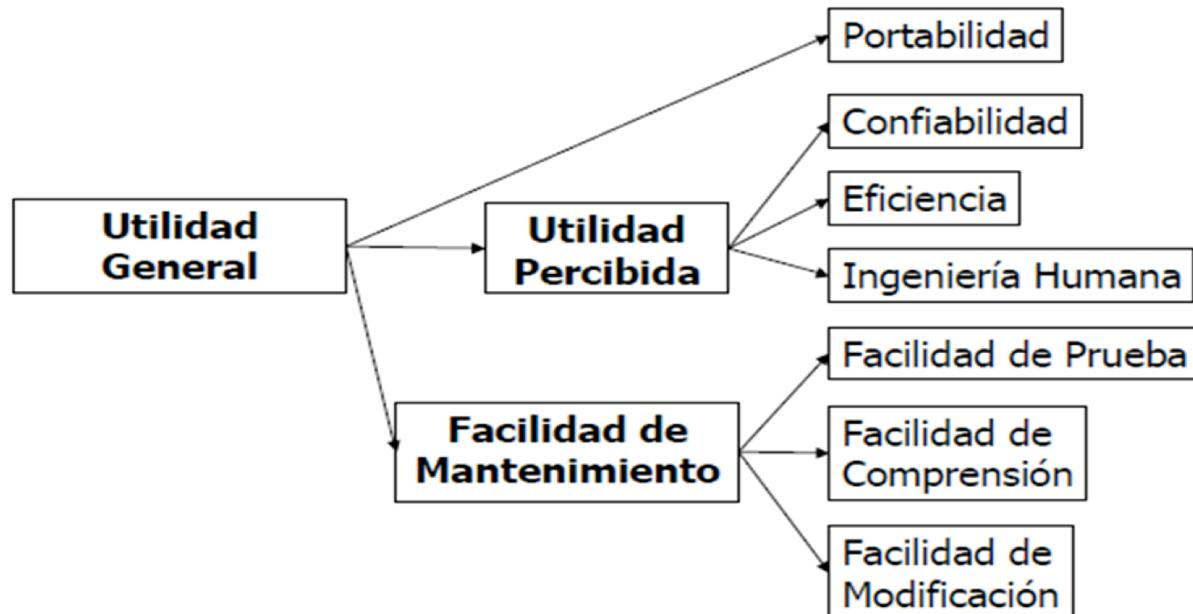


Figura 3. modelo Boehm
Fuente: Carlos Luis Cabascango (2012)

Calidad en el desarrollo de Software

5.2. A nivel de proceso

Existen diferentes modelos que a nivel de proceso orientan la calidad de software. Dentro de ellos se puede mencionar al Modelo CMM padre del modelo CMMI, IRAM-ISO 9001, TICKIT, Modelo Bootstrap, Personal Software Process (PSP), Team Software Process (TSP), Practical Software Measurement (PSM), Six Sigma For Software, entre otros. En el presente material se estudiará el modelo CMMI (Capability Maturity Model Integration), por su gran incidencia en el mercado de desarrollo de software.

CMMI (Capability Maturity Model Integration)

Es un enfoque de mejoramiento de procesos lanzado en el año 2000 por el Instituto de Ingeniería de Software (SEI) de la Universidad Carnegie Mellón, bajo la dirección de Watts Humphrey. Algunos de sus objetivos más importantes son:

- Trazar cinco niveles de madurez certificables, que identifican su nivel de estandarización de procesos, en las áreas evaluadas.
- Permitir a organizaciones que desarrollan software, tener una guía de cómo controlar sus procesos de desarrollo y mantenimiento.
- Determinar la madurez de los procesos actuales de una organización e identificar los elementos críticos en el aseguramiento de calidad y mejoramiento de procesos.
- Identificar prácticas claves requeridas para el mejoramiento e incremento de la madurez de los procesos de desarrollo de software.
- Mientras que su padre CMM está enfocado en el modelo tradicional de cascada, CMMI está pensado en los modelos de desarrollo de software iterativos que se usan en la actualidad.
- CMMI ayuda a integrar funciones organizacionales que estaban tradicionalmente separadas, se enfoca en mejorar las metas y prioridades del proceso, provee guías para procesos de calidad y un punto de referencia para valorar procesos reales.

Los niveles de CMMI son los siguientes:

Nivel de madurez 1. Inicial: procesos caóticos.

Nivel de madurez 2. Gestionando: conciencia de la dirección, gestión de procesos.

Nivel de Madurez 3. Definido: procesos caracterizados y comprendidos.

Nivel de Madurez 4. Gestión cuantitativa: la organización establece objetivos medibles en cuanto a rendimiento de calidad del proceso.

Calidad en el desarrollo de Software

Nivel de Madurez 5. En optimización: mejora continua de procesos basado en una comprensión cuantitativa.

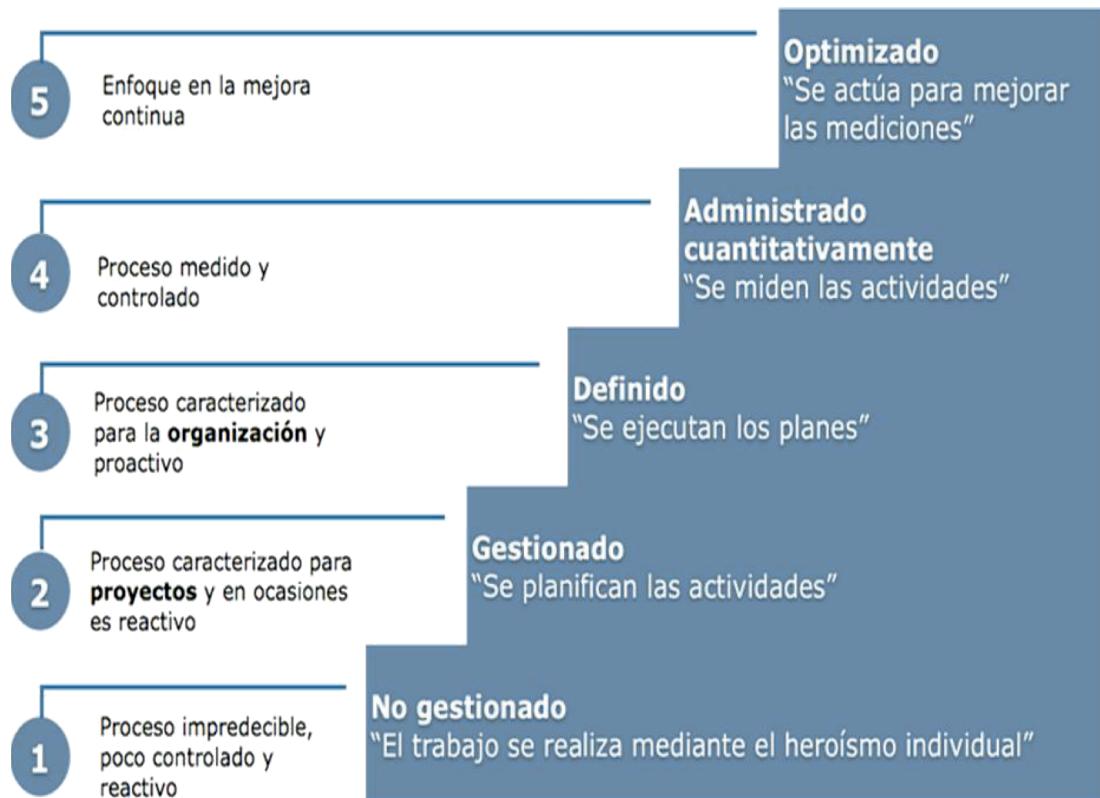


Figura 4. Modelo CMMI

Fuente: CCTI Soluciones tecnológicas (2017)

6. Estándares de calidad del software

Son aquellos que permiten definir un conjunto de criterios de desarrollo que guían la forma en que se aplica la Ingeniería del Software. Los estándares suministran los medios para que todos los procesos se realicen de la misma forma y son una guía para lograr la productividad y la calidad (Scalone, 2006).

6.1. A nivel de producto

Con respecto a los estándares del producto, a nivel internacional existe una organización conocida como ISO (Organización Internacional de Normalización), es una federación mundial de organismos nacionales de normalización (organismos miembros de ISO).

ISO colabora estrechamente con la Comisión Electrotécnica Internacional (IEC) en todas las materias de normalización electrotécnica. La tarea principal de los comités técnicos es preparar Normas Internacionales.

Calidad en el desarrollo de Software

6.1.1. Estándar ISO/IEC 9126

Se originó en 2001 y su última versión revisada es del año 2004. Evalúa la calidad del producto software a partir de seis atributos principales.

En la primera parte, especifica seis características de calidad (interna y externa), divididas en sub categorías, que se manifiestan externamente (cuando el software es utilizado como parte de un sistema).

La calidad interna evalúa el total de atributos que un producto debe satisfacer, teniendo en cuenta condiciones específicas. Esta calidad es medible a partir de las características intrínsecas.

Factores y *subcaracterísticas* de calidad de software que se pueden considerar tanto internos como externos se detallan en la siguiente tabla:

Tabla 3.
Factores y subcaracterísticas de calidad de software

Factores	Sub – características
Funcionalidad: conjunto de atributos que relacionan la existencia de un conjunto de funciones con sus propiedades especificadas. Las funciones satisfacen necesidades especificadas o implícitas.	Adecuación: atributos que determinan si el conjunto de funciones. Son apropiadas para las tareas específicas. Exactitud: atributos que determinan que los efectos sean los correctos o los esperados. Seguridad: atributos que miden la habilidad para prevenir accesos no autorizados, ya sea accidentales o deliberados, tanto a programas como a datos. Interoperabilidad: atributos que miden la habilidad de interactuar con sistemas especificados. Cumplimiento: atributos que hacen que el software adhiera a estándares relacionados con la aplicación, y convenciones o regulaciones legales.
Confiabilidad: conjunto de atributos que se relacionan con la capacidad del software de mantener su nivel de performance bajo las condiciones establecidas por un período de tiempo.	Madurez: atributos que se relacionan con la frecuencia de fallas por defectos en el software. Tolerancia a las fallas: atributos que miden la habilidad de mantener el nivel especificado de performance en caso de fallas del software. Recuperación: atributos que miden la capacidad de reestablecer el nivel de performance y recuperar datos en caso de falla, y el tiempo y esfuerzo necesario para ello. Cumplimiento: atributos que hacen que el software se adhiera a estándares relacionados con la aplicación, y convenciones o regulaciones legales.
Eficiencia: conjunto de atributos que se relacionan con el nivel de performance del software y la cantidad de recursos usados, bajo las condiciones establecidas.	En tiempo: atributos que miden la respuesta y tiempos de procesamiento de las funciones. En recursos: atributos que miden la cantidad de recursos usados y la duración de tal uso en la ejecución de las funciones. Cumplimiento: atributos que hacen que el software se adhiera a estándares relacionados con la aplicación, y convenciones o regulaciones legales.

Calidad en el desarrollo de Software

Usabilidad: conjunto de atributos que se relacionan con el esfuerzo necesario para usar, y en la evaluación individual de tal uso, por parte de un conjunto especificado o implícito de usuarios.	Entendimiento: atributos que miden el esfuerzo del usuario en reconocer el concepto lógico del software y su aplicabilidad. Aprendizaje: atributos que miden el esfuerzo del usuario en aprender la aplicación (control, operación, entrada, salida). Operabilidad: atributos que miden el esfuerzo del usuario en operar y controlar el sistema atractivo Cumplimiento: atributos que hacen que el software se adhiera a estándares relacionados con la aplicación, y convenciones o regulaciones legales.
Mantenibilidad: conjunto de atributos que se relacionan con el esfuerzo en realizar modificaciones.	Analizabilidad: atributos que miden el esfuerzo necesario para el diagnóstico de deficiencias o causas de fallas, o para identificación de las partes que deben ser modificadas. Facilidad para el cambio: atributos que miden el esfuerzo necesario para realizar modificaciones, remoción de fallas o cambios en el contexto. Estabilidad: atributos que se relacionan con el riesgo de efectos no esperados en las modificaciones. Testeabilidad: atributos que miden el esfuerzo necesario para validar el software modificado. Cumplimiento: atributos que hacen que el software se adhiera a estándares relacionados con la aplicación, y convenciones o regulaciones legales.
Portabilidad: conjunto de atributos que se relacionan con la habilidad del software para ser transferido de un ambiente a otro.	Adaptabilidad: atributos que miden la oportunidad de adaptación a diferentes ambientes sin aplicar otras acciones que no sean las provistas para el propósito del software. Instalabilidad: atributos que miden el esfuerzo necesario para instalar el software en el ambiente especificado. Conformidad: atributos que miden si el software se adhiere a estándares o convenciones relacionados con portabilidad. Reemplazo: atributos que se relacionan con la oportunidad y el esfuerzo de usar el software en lugar de otro en su ambiente.

Fuente: Fillottrani (2007)

La calidad externa se encarga de evaluar que el software cumpla con las necesidades del usuario, teniendo en cuenta condiciones previas específicas. Dicha calidad es medible en el comportamiento del producto.

Tanto características como *subcaracterísticas* proveen terminología consistente a la calidad del producto y se pueden aplicar a todos los tipos de software.

Esta norma, permite especificar y evaluar la calidad desde distintas perspectivas, asociadas con la adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad, y auditoría. Puede ser usada por desarrolladores o grupos de aseguramiento de calidad, responsables de especificar y evaluar, evaluadores independientes, entre otros.

Calidad en el desarrollo de Software

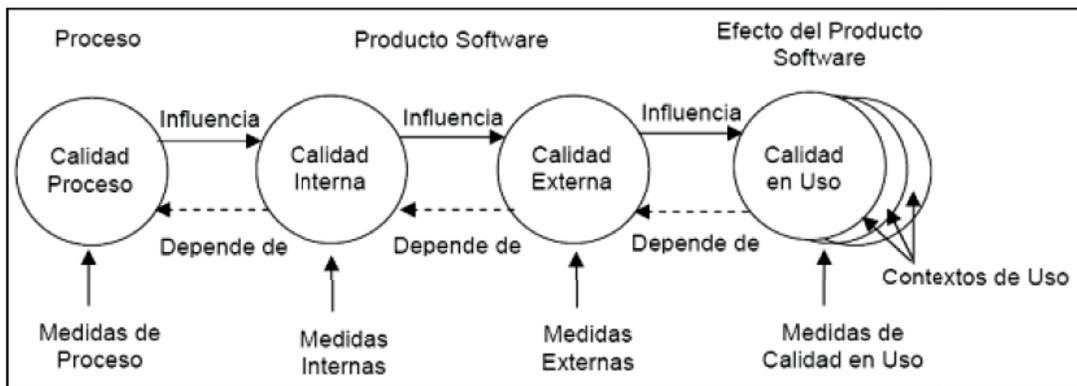


Figura 5. Marco conceptual para el modelo de calidad

Fuente: González (2002)

6.1.2. Estándar ISO/IEC 25000:2005

Los aspectos más importantes en el desarrollo de software son la calidad del producto y del proceso.

ISO/IEC 25000:2005, proporciona una guía para el uso de las nuevas series de estándares internacionales, llamados Requisitos y Evaluación de Calidad de Productos de Software (SQuaRE). Constituyen una serie de normas basadas en la ISO 9126 y en la ISO 14598, y su objetivo principal es guiar el desarrollo de los productos de software con la especificación y evaluación de requisitos de calidad (Portal ISO 25000.com, 2018).

La familia ISO 25000:2005 está orientada al producto software, permitiendo definir el modelo de calidad y el proceso a seguir para evaluarlo. Esta se compone de cinco divisiones:

- 1. ISO 2500n:** gestión de la calidad.
- 2. ISO 2501n:** modelo de calidad.
- 3. ISO 2502n:** medida de la calidad.
- 4. ISO 2503n:** requisitos de calidad.
- 5. ISO 2504n:** evaluación de la calidad.

El estándar ISO/IEC 25000:2005, contiene una explicación sobre el proceso de transición entre el estándar ISO/IEC 9126, las series 14598 y SQuaRE. También presenta información sobre cómo utilizar la norma ISO/IEC 9126 y la serie 14598 en su forma anterior. Ofrece términos y definiciones, modelos referencia, guía general, guías de división individual y los estándares para fines de especificación, planificación y gestión, medición y evaluación (Alfonzo y Mariño, 2013).

Calidad en el desarrollo de Software

6.1.3. Estándar ISO/IEC 25010:2011

El estándar ISO/IEC 25010 (2011), reemplaza y actualiza el estándar ISO9126-1 (2001). Define:

- Un modelo de calidad en uso que se compone de cinco características (algunas de las cuales se subdividen en *subcaracterísticas*). Se relacionan con el resultado de la interacción cuando un producto se emplea en un contexto particular de uso.
- Un modelo de calidad del producto se compone de ocho características (que se subdividen en *subcaracterísticas*). Se refieren a propiedades estáticas de software y las propiedades dinámicas del sistema informático. El modelo es aplicable a los productos de software y sistemas informáticos.

Las características definidas por ambos modelos son relevantes para todos los productos de software y sistemas informáticos. Las características y *subcaracterísticas* proporcionan coherencia terminológica para especificar, medir y evaluar la calidad del producto software y sistemas informáticos.

El modelo de calidad de producto abarca cualidades internas y externas del sistema y está compuesto por ocho características y treinta y una *subcaracterísticas*. El modelo en uso se compone de cinco características y nueve *subcaracterísticas* (Polillo, 2011).

6.2. A nivel de proceso

6.2.1. Estándar ISO 15504 – SPICE

Es un modelo de mejora y evaluación de procesos de desarrollo y mantenimiento de sistemas o productos de software, comúnmente utilizado en Europa y basado en SPICE (Software Process Improvement and Capability) (SEI – CMMI, s.f.). Tiene las siguientes características:

- Enfocado en la evaluación, establece un marco para evaluar, más que ser un método propiamente dicho. Comprende las siguientes áreas:
 - Mejora de procesos
 - Evaluación de procesos
 - Determinación de capacidad
- Está alineado con el estándar ISO 12207 que define los procesos del ciclo de vida de desarrollo y mantenimiento de software.

SPICE provee nueve documentos que guían todo el proceso de implementación del modelo y su correspondiente evaluación:

Calidad en el desarrollo de Software

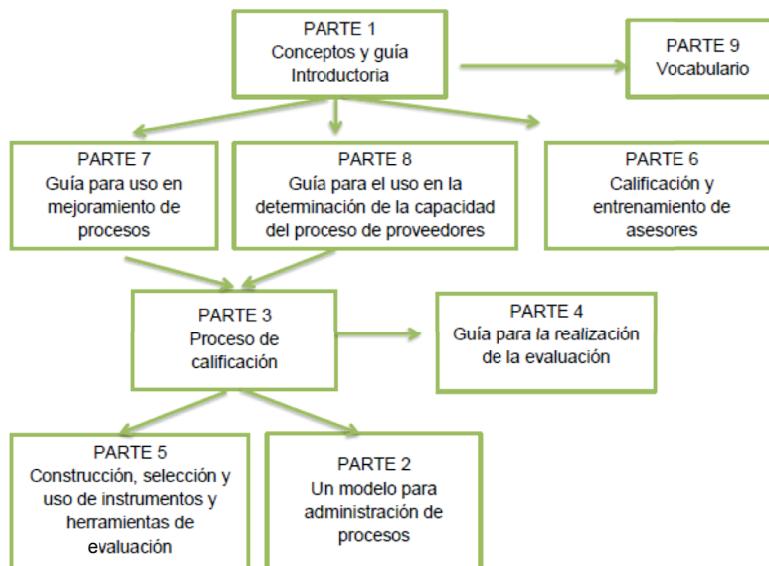


Figura 6. Modelo ISO 15504 – SPICE

Fuente: SENA

6.2.2. Estándar ISO 9001-2015

La norma ISO 9001- 2015 pertenece a la familia ISO 9000 de normas de sistemas de gestión de la calidad (junto con ISO 9004); se basa en la solución de retos empresariales y ayuda a las organizaciones a cumplir con las expectativas y necesidades de sus clientes, entre otros beneficios.



Figura 7. De la norma ISO 9001: 2008 a la ISO 9001:2015

Fuente: CAVALA Gabinete de Asesoría Empresarial (2011)

7. Glosario

Boehm: el desarrollo en espiral es un modelo de ciclo de vida del software definido por primera vez por Barry Boehm en 1986, utilizado generalmente en la Ingeniería de software. Las actividades de este modelo se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades.

Calidad de software: conjunto de características de un producto de software que tienen como habilidad, satisfacer necesidades explícitas o implícitas del cliente.

CMMI: Capability Maturity Model Integration. Metodología para facilitar el control de rendimiento de empresas en el sector de Tecnologías de la Información.

IEC: Comisión Electrotécnica Internacional.

ISO: Organización Internacional de Normalización.

ISO 15504: norma que se encarga de la mejora, evaluación de los procesos de desarrollo, mantenimiento de sistemas de información y productos de software.

ISO 25000:2005: norma conocida como SQuaRE (System and Software Quality Requirements and Evaluation), es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software.

ISO 25010:2011: norma ecuatoriana traducida de la norma internacional ISO/IEC 25010:2011.

ISO 9001:2015: norma internacional 9001, versión 2015 que normaliza la gestión de la calidad.

ISO 9126: es un estándar internacional para la evaluación del Software, fue originalmente desarrollado en 1991 para proporcionar un esquema para la evaluación de calidad del software.

McCall: modelo centrado en tres aspectos de un producto de software: Operación del producto, revisión del producto, transición del producto.

SPICE: acrónimo de Software Process Improvement Capability dEtermination. Se utiliza para identificar a la norma ISO/IEC 15504, la cual se encarga de la mejora, evaluación de los procesos de desarrollo, mantenimiento de sistemas de información y productos de software.

SQUARE: nombre con el que se conoce a la norma ISO 25000:2005.

Calidad en el desarrollo de Software

8. Referencias bibliográficas

BSI GROUP (2015). *Norma ISO 9001 - Gestión de la calidad*. Recuperado de <https://www.bsigroup.com/es-ES/Gestion-de-Calidad-ISO-9001/>

Cavala (2011). *Cambios en la norma ISO 2015*. Recuperado de <http://www.cavala.es/noticias/nueva-iso-9001-2015/>

Cabascango, Carlos L. (2012, octubre 24). *Control de calidad y auditoría informática*. [Web log post]. Recuperado de <http://resumenccai.blogspot.com.co/>

CCTI Soluciones tecnológicas (2017). *¿Sabías qué? (CMMI)*. [Web log post] Recuperado de <https://www.ccti.com.co/index.php/blog/197-sabias-que-cmmi>

Estayno M., Dapozo, G., Cuenca Plethc, L., & Greiner, C. (2009). *Modelos y métricas para evaluar la calidad de software*. Recuperado de <http://sedici.unlp.edu.ar/handle/10915/19762>

Fillottrani, P. (2007). *Calidad en el desarrollo de Software - Modelos de calidad de software*. Recuperado de <http://www.cs.uns.edu.ar/~prf/teaching/SQ07/clase6.pdf>

Gómez, D. (2009). *¿Qué significa calidad de software?* Recuperado de <https://dosideas.com/noticias/reflexiones/564-que-significa-calidad-de-software>

Noriega Quintana, D. (s.f.). *Calidad del Software*. Recuperado de <http://www.monografias.com/trabajos59/calidad-software/calidad-software.shtml>

Normas9000.com. (s.f.). *¿Qué es ISO?* Recuperado de <http://www.normas9000.com/content/que-es-iso.aspx>

Presidencia del Consejo de Ministros. (2013). *Oficina Nacional de Gobierno Electrónico e Informática*. Recuperado de: <http://www.pcm.gob.pe/normaslegales/2013/DS-081-2013-PCM.pdf>

Pressman R. (2010). *Ingeniería del software - un enfoque práctico* (Vol. Calidad de Software). México: Mac Graw Hill.

Revista Venezolana de Información, Tecnología y Conocimiento (2016). *Optimización de estados en la mejora de procesos de software*. Recuperado de <http://www.redalyc.org/pdf/823/82349540001.pdf>

Scalone, F. (2006). *Estudio comparativo de los modelos y estándares de calidad del software*. (Maestría en ingeniería de calidad). Universidad Tecnológica Nacional Buenos Aires, Argentina. Recuperado de <http://laboratorios.fi.uba.ar/lsi/scalone-tesis-maestria-ingenieria-en-calidad.pdf>

Calidad en el desarrollo de Software

Vega, Rivera & García (2008). *Mejores prácticas para el establecimiento y aseguramiento de la calidad de software*. Ciudad de México: Universidad Cristóbal Colón.

UCAM. (2014). *Calidad del Software* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=Hf-47kSvkHc>

Control del documento

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Humberto Amaya Alvear	Experto Técnico	Centro de Gestión de Mercados, Logística y Tecnologías de la Información Distrito Capital - Bogotá – SENA.	Octubre de 2017.

Control de cambios

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)	Gissela del Carmen Alvis Ladino	Asesora pedagógica	Centro de Comercio y Servicios Regional Tolima.	Febrero de 2018	Correcciones y actualización del formato.

Créditos



Equipo de Adecuación
Centro de Comercio y Servicios SENA Regional Tolima
Línea de Producción

Director regional: Félix Ramón Triana Gaitán

Subdirector de centro: Álvaro Fredy Bermúdez Salazar

Coordinadora de formación profesional integral: Gloria Inés Urueña Montes

Senior equipo de adecuación: Jorge Bustos Gómez

Experto temático: Humberto Amaya Alvear

Asesora pedagógica: Gissela del Carmen Alvis Ladino

Guionista: Ofelia Victoria Torres Gómez

Diseño y diagramación: Pedro Nel Cabrera Vanegas
Wilson Alfonso Mahecha Zocadagui
Jefferson Alejandro Fuentes González

Programadores: Davison Gaitán Escobar
Daniel Santiago Bonilla Posada
Oscar Daniel Espitia Marín
Ivan Dario Rivera Guzmán



creative commons



Atribución - No Comercial - Sin Derivar.

(BY-NC-ND)

No se permite un uso comercial de la obra original ni la generación de obras derivadas.

