

DOI: [https://doi.org/10.37811/cl\\_rcm.v6i3.2456](https://doi.org/10.37811/cl_rcm.v6i3.2456)

## Componentes de calidad software y su utilización en aplicaciones web

Elizabeth Alexandra Veloz Segura

[evelozs@uteg.edu.ec](mailto:evelozs@uteg.edu.ec)

<https://orcid.org/0000-0003-4562-7619>

Universidad Técnica Estatal de Quevedo.

### RESUMEN

La presente investigación se enmarca en analizar componentes software y su factibilidad de adaptación en las aplicaciones web; para lo cual se utiliza una metodología documental y descriptiva, conjuntamente con una revisión bibliográfica.

Al no ser evaluados dichos componentes, se desconoce qué tan compatibles y adaptables son a los sistemas en los que se utiliza.

Mediante la investigación se analiza los componentes software comúnmente utilizados para el desarrollo de aplicaciones, componentes COTS Commercial Off-The-Shelf

Para el análisis de calidad de dichos componentes se puede evaluar los atributos de calidad de acuerdo con la norma ISO 9126, con herramientas y métodos de selección.

**Palabras Clave:** *calidad; componentes; software; web*

Correspondencia: [evelozs@uteg.edu.ec](mailto:evelozs@uteg.edu.ec)

Artículo recibido: 02 mayo 2022. Aceptado para publicación: 25 mayo 2022.

Conflictos de Interés: Ninguna que declarar

Todo el contenido de **Ciencia Latina Revista Científica Multidisciplinar**, publicados en este sitio están disponibles bajo

Licencia [Creative Commons](https://creativecommons.org/licenses/by/4.0/) 

Como citar: Veloz Segura, E. A. (2022). Componentes de calidad software y su utilización en aplicaciones web. Ciencia Latina Revista Científica Multidisciplinar, 6(3), 3193-3204. DOI: [https://doi.org/10.37811/cl\\_rcm.v6i3.2456](https://doi.org/10.37811/cl_rcm.v6i3.2456)

## Software quality components and their use in web applications

### ABSTRACT

The present investigation is framed in analyzing software components and their feasibility of adaptation in web applications; for which a documentary and descriptive methodology is used, together with a bibliographic review.

Since these components have not been evaluated, it is unknown how compatible and adaptable they are to the systems in which they are used.

Through the investigation, the software components commonly used for the development of applications, COTS Commercial Off-The-Shelf components, are analyzed.

For the quality analysis of these components, the quality attributes can be evaluated in accordance with the ISO 9126 standard, with selection tools and methods.

*Keywords: quality; components; software; Web*

## INTRODUCCIÓN

El aumento de las tecnologías y el desarrollo de las telecomunicaciones han traído consigo que el mercado del software se encuentre actualmente en un gran momento. Teniendo en cuenta este auge, se hace imprescindible que las empresas que desarrollan software prioricen la calidad, ya que paralelo a la gran demanda de software, se encuentran los clientes que exigen que este posea cada vez mayor calidad. (Marcos, 2016)

En los últimos tiempos el software ha tenido una gran evolución, debido a la presencia de la ingeniería del software con estándares, normas y procedimientos que lo regulan, debido a lo expuesto las empresas e industrias empezaron a confiar en el software para automatizar procesos o para utilizarlos como sistemas que les ayuden a gestionar, administrar y controlar sus procesos administrativos. De acuerdo con esta necesidad, diferentes entidades o investigadores han propuesto estrategias modelos, metodologías, guías, incluso normas y estándares de calidad que brindan apoyo al desarrollo y/o uso de un producto software y permiten evaluar si efectivamente tiene un nivel de calidad durante su ciclo de vida, y de esta manera fomentar un ambiente de calidad, con base en la adecuada administración de la información. (Callejas Cuervo, Alarcón Aldana, & Álvarez Carreño, 2017)

Las tecnologías de comunicación logran cambios significativos en los procesos de las instituciones, mejorando su gestión y organización. El software es una de las herramientas de mayor utilidad en la optimización de procesos en las organizaciones, con el propósito de contar y ofrecer optimización, eficiencia y satisfacción de necesidades, razón por la cual el software debe contar con criterios que garanticen su calidad. (Callejas Cuervo, Alarcón Aldana, & Álvarez Carreño, 2017)

La Calidad de software, se asocia a la "concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo plenamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente", con base en los requisitos funcionales y no funcionales identificados en la etapa de análisis del sistema, insumo principal para implementar dichos requisitos con los atributos mínimos de calidad, fomentando la aplicación de procesos estandarizados y criterios necesarios en cada una de sus etapas, así se fomenta

que el avance en el ciclo de vida del software minimice el riesgo de fracaso del proyecto. (PRESSMAN, 2010)

El desarrollo de software basado en componentes es uno de los mecanismos más efectivos para la construcción de grandes sistemas y aplicaciones de software. Una vez que la mayor parte de los aspectos funcionales de esta disciplina comienzan a estar bien definidos, la atención de la comunidad científica comienza a centrarse en los aspectos extra funcionales y de calidad, como un paso hacia una verdadera ingeniería según (Bertoa, 2011).

Los componentes son abstracciones de alto nivel en comparación con los objetos y se definen mediante sus interfaces. (Sommerville, 2011).

Al realizar una correcta selección de componentes, se puede conseguir proyectos software de manera rápida y eficiente, ya que se puede incorporar componentes ya existentes, lo cual permite ahorrar recursos: humanos, tecnológicos, financieros, otros. Según (Sommerville, 2011), que existen dos tipos de estándares de ingeniería de software

relacionados que pueden definirse y usarse en la gestión de calidad del software:

**Estándares del producto:** Se aplican al producto de software a desarrollar. Incluyen estándares de documentos (como la estructura de los documentos de requerimientos), estándares de documentación (como el encabezado de un comentario estándar para una definición de clase de objeto) y estándares de codificación, los cuales definen cómo debe usarse un lenguaje de programación.

**Estándares de proceso:** Establecen los procesos que deben seguirse durante el desarrollo del software. Deben especificar cómo es una buena práctica de desarrollo. Los estándares de proceso pueden incluir definiciones de especificación, procesos de diseño y validación, herramientas de soporte de proceso y una descripción de los documentos que deben escribirse durante dichos procesos para apoyar el uso de dichos estándares.

La tendencia actual se orienta al desarrollo web para diferentes ámbitos como el área empresarial, educativo, otros, por lo tanto, es de mucha relevancia que se lleve un seguimiento de los servicios que brindan, considerando el cumplimiento de la calidad.

Los sistemas y aplicaciones basados en web han evolucionado de simples conjuntos de contenido de información a sistemas sofisticados que presentan una funcionalidad

compleja y contenido en multimedios. Aunque dichas webapps tienen características y requerimientos únicos, son software. (PRESSMAN, 2010)

Una aplicación web (web-based application) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados. (Luján Mora, 2002).

En la ingeniería de software se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. (López Torralba, 2015).

Las utilidades de aplicaciones Web permiten ahorrar tiempo, proporcionan mayor rendimiento, eficiencia, permite una actualización continua, permite ahorro de recursos, facilidad para recuperar información desde cualquier lugar y dispositivo, siempre y cuando se cuente con acceso a Internet.

La investigación se realiza con la finalidad de hacer una evaluación a los componentes software existentes, y determinar su grado de calidad, basándonos en las métricas de calidad como: usabilidad, seguridad, rendimiento, portabilidad, mantenimiento, eficiencia.

Los resultados de dicha investigación permitirán que se realice una correcta selección de componentes, evitando que contengan fallos de seguridad, que puedan poner en riesgo a las aplicaciones o que ocasionen problemas de rendimiento y la factibilidad de su implementación en aplicaciones web.

## **METODOLOGÍA**

La metodología utilizada en el presente trabajo es de carácter documental y descriptiva, con enfoque mixto, realizándose una revisión bibliográfica y un análisis de la información, de tipo descriptivo utilizando publicaciones científicas, búsqueda de información, elección de una serie de datos documentales válidos y herramientas software.

Investigación Cualitativa: Para determinar atributos de calidad.

Investigación bibliográfica: Contribuye al conocimiento científico y teórico en la presente investigación, ya que se revisará información existente en libros, artículos científicos, revistas, tesis, repositorios digitales ya sean en físicos o digitales.

Según (Morales, 2003) la investigación documental tiene la particularidad de utilizar como una fuente primaria de insumos, mas no la única y exclusiva, el documento escrito en sus diferentes formas: documentos impresos, electrónicos y audiovisuales. Los estudios descriptivos “buscan especificar las propiedades, las características y los aspectos importantes del fenómeno que se somete a análisis. (Gómez M. , 2006)

### **RESULTADOS Y DISCUSIÓN**

La calidad en ingeniería del software es el cumplimiento de los requerimientos contractuales por parte del producto software desarrollado, así como durante el proceso de desarrollo. La calidad se obtiene mejorando día a día el proceso de producción, mantenimiento y gestión del software. (Redondo, 2016)

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

Entendiéndose por componente de software según (Szyperski , 2002), como "una unidad de composición de aplicaciones software, que posee un conjunto de interfaces y un conjunto de requisitos, y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros componentes de forma independiente, en tiempo y espacio.

El DSBC(desarrollo de software basado en componentes), no es más que el proceso de desarrollo que utiliza la integración de componentes para crear productos de software, puede verse como una extensión natural de la programación orientada a objetos dentro del ámbito de los sistemas abiertos y distribuidos. (Bertoa, M. F., Troya , J., & Vallecil, A. 2017)

Según (Bertoa, M. F., Troya , J., & Vallecil, A. 2017), una de las principales ventajas del desarrollo de software basado en componentes se basa en la “reutilización” de los mismos. De esta forma, los componentes se diseñan y desarrollan con el objetivo de poder ser reutilizados en otras aplicaciones, reduciendo el tiempo de desarrollo, mejorando la fiabilidad del producto final (al usar componentes ya probados previamente), y siendo más competitivos en costes.

## Calidad de Componentes Software

Según (Bass L. , Clements P. y R. 1999) menciona:

- **Componente:** clase especial de componentes software, normalmente de grano grueso que presentan las siguientes características :
  1. Vendidos o licenciados al público en general
  2. Su código no puede ser modificado por el usuario No hay control sobre su evolución:
  3. Los mantiene y actualiza el propio vendedor, quien conserva los derechos de la propiedad intelectual.
  4. Están disponibles en forma de múltiples copias, todas idénticas entre sí.

La tecnología básica de componentes comienza a estar lo suficientemente madura (a través de plataformas de objetos distribuidos como EJB, CORBA o .NET) como para que numerosas empresas la adopten en sus nuevos desarrollos y sistemas de información.

Una de las tareas en el desarrollo de software basado en componentes se centra en los procesos y herramientas para la evaluación y selección de componentes de calidad COTS. Una vez seleccionados los componentes candidatos, normalmente es preciso realizar una adaptación y/o extensión de los mismos a la hora de integrarlos. En general, al tratar de combinar los componentes suelen aparecer tres tipos de problemas típicos de los ambientes COTS: las lagunas (gaps), los solapamientos (overlappings), y la incompatibilidad de interfaces (mismatches). (Garlan, 1995)

Según (Carvallo, J. P., Franch, X., & Carme , Q. 2006), la selección de componentes COTS (Commercial Off-The-Shelf) se realiza no sólo mediante el análisis de su calidad técnica sino también considerando cómo dichos componentes cumplen con los requisitos no-técnicos considerados relevantes, esta selección de componentes COTS tiene como objetivo localizar en el mercado de componentes COTS aquéllos que mejor satisfacen los requisitos a satisfacer, se debe tomar en cuenta dos clases de requisitos: técnicos y no técnicos.

Para la evaluación de calidad de estos componentes se lo realiza , evaluando los atributos de calidad de acuerdo con la norma ISO 9126, la cual menciona que un producto de se considera de calidad, si presenta los siguientes atributos:

1. Funcionalidad
2. Eficiencia
3. Fiabilidad

4. Mantenibilidad

5. Usabilidad

6. Portabilidad

Norma ISO 9126, está dividido en cuatro partes: modelo de calidad, métricas externas, métricas internas y calidad de métricas en uso; elementos en torno a seis características (funcionalidad, Habilidad, usabilidad, eficiencia, mantenibilidad y portabilidad) y subcaracterísticas asociadas. (ANGO HERRERA, 2014)

**Cuadro 1:** Características de calidad según ISO 9126

Características	Sub-características
<i>Funcionalidad</i>	<i>Adecuación Corrección Interoperatividad Conformidad Seguridad</i>
<i>Fiabilidad</i>	<i>Madurez Recuperabilidad Tolerancia a Fallos</i>
<i>Facilidad de Uso</i>	<i>Facilidad de Aprendizaje Facilidad de Comprensión Operatividad</i>
<i>Eficiencia</i>	<i>Comportamiento Temporal Utilización de Recursos</i>
<i>Mantenibilidad</i>	<i>Estabilidad Analizabilidad Cambiabilidad Facilidad de Prueba</i>
<i>Portabilidad</i>	<i>Facilidad de Instalación Adecuación Remplazabilidad Adaptabilidad</i>

Para componentes COTS, es fundamental distinguir entre métricas internas y externas. Las inter-nas miden los atributos internos del producto final o de los productos intermedios (p.e. la especi-ficación o el código fuente) durante el diseño y la codificación. Las externas son las que realizan las mediciones en función del comportamiento del sistema durante las pruebas y la operación del componente. Por tanto, debido al carácter de caja negra de los componentes COTS, son las métricas externas las que interesan. (Bertoa, M. F. 2003)

**Cuadro 2:** Modelo de calidad para componentes COTS (Bertoa, M. F. 2003)

Características	Sub-características (Tiempo de ejecución)	Sub-características (ciclo de vida)
Funcionalidad	Corrección  Seguridad	Idoneidad  Interoperatividad Conformidad  <i>Compatibilidad</i>
Fiabilidad	Recuperabilidad	Madurez
Facilidad de Uso		<b>Facilidad de Aprendizaje</b> <b>Facilidad de Comprensión</b> <b>Operatividad</b> <i>Complejidad</i>
Eficiencia	Comportamiento Temporal Utilización de Recursos	
Mantenibilidad		Cambiabilidad Facilidad de Prueba

El desarrollo por componentes y la programación orientada a objetos permiten la reutilización del software, permitiendo la creación de aplicativos web que cumplen estándares de calidad.

La estandarización es el único camino de unificación de criterios que permite el manejo de políticas de calidad en el desarrollo web.

Para evaluar la usabilidad y funcionalidad de los componentes se prueba las funcionalidades que debe cumplir de acuerdo a la descripción del mismo; para los atributos de fiabilidad, eficiencia, mantenibilidad se los evalúa mediante la utilización de la herramienta software como SonarQube( plataforma para evaluar código fuente, software libre y usa diversas herramientas de análisis estático de código fuente), entre otras; para determinar si los componentes cumplen con el atributo de portabilidad se los implementa en diferentes navegadores (programas que permite el acceso a la Web) y diferentes plataformas.

## CONCLUSIONES

El estudio de la calidad de los componentes software juega un papel muy importante en el DSBC, los procesos de selección de componentes es necesario conocer con detalle el comportamiento relativo a aquellos criterios que se corresponden con los requisitos del sistema en desarrollo, tanto funcionales como no-funcionales.

Con una evaluación y correcta selección de componentes se desarrolla software de calidad, disminuyendo el tiempo en el desarrollo y los costos, para lo cual se utiliza ISO 9126.

Los componentes de calidad se los puede incluir y adaptarlos con facilidad en las aplicaciones web, siguiendo métricas y parámetros de calidad.

#### REFERENCIAS BIBLIOGRAFICAS

Agüero Atmitim, J. M. (22 de 02 de 2021). *Qué es Ionic*. Obtenido de <https://profile.es/blog/que-es-ionic/>

Alegsa, L. (2010). *Definición de aplicación web*. Obtenido de [https://www.alegsa.com.ar/Dic/aplicacion\\_web.php](https://www.alegsa.com.ar/Dic/aplicacion_web.php)

Alegsa, L. (2016). Obtenido de <http://www.alegsa.com.ar/Dic/sistema%20informatico.php>

ANGO HERRERA, L. F. (2014). *Evaluación de Sistemas", Tesis.Pontifica Universidad Catolica del Ecuador*.

Barzanallana, R. (2012). *Historia del desarrollo de aplicaciones Web*. Obtenido de <http://www.um.es/docencia/barzana/DIVU>

Bass L. , Clements P. y R. (1999). *Software Architecture in Practice*. 6ª edición. Addison-Wesley, 1999

Bertoa, M. F. (2011). *Aspectos de Calidad en el Desarrollo de Software Basado en Componentes*. Obtenido de [https://www.ecured.cu/Calidad\\_en\\_el\\_Desarrollo\\_Software\\_Basado\\_en\\_Componentes](https://www.ecured.cu/Calidad_en_el_Desarrollo_Software_Basado_en_Componentes)

Bertoa, M. F., Troya , J., & Vallecil, A. (2017). *Aspectos de Calidad en el Desarrollo de Software, basado en componentes*. Obtenido de <http://www.lcc.uma.es/~av/Publicaciones/02/CalidadDSBC.pdf>

Bertoa, M. F. (2003). *Atributos de calidad para componentes COTS*. Obtenido de [https://d1wqtxts1xzle7.cloudfront.net/67122774/Atributos\\_de\\_calidad\\_para\\_componentes\\_CO20210505-31717-l7d4kx-with-cover-page-v2.pdf?Expires=1654616505&Signature=buHm2vmxk-TumygUwzyY74tEp~k3gPI5Q7wrb4v7K4~nTv6CctVPO~r6EmjPn5e5ZjHqDONy7ZKAHlaKJk47Re-x-KWA7](https://d1wqtxts1xzle7.cloudfront.net/67122774/Atributos_de_calidad_para_componentes_CO20210505-31717-l7d4kx-with-cover-page-v2.pdf?Expires=1654616505&Signature=buHm2vmxk-TumygUwzyY74tEp~k3gPI5Q7wrb4v7K4~nTv6CctVPO~r6EmjPn5e5ZjHqDONy7ZKAHlaKJk47Re-x-KWA7)

- Blog infranetworking*. (04 de 01 de 2019). Obtenido de <https://blog.infranetworking.com/servidor-web/>
- Borges, S. (04 de Enero de 2019). *blog infranetworking*. Obtenido de <https://blog.infranetworking.com/servidor-web/>
- Carrizo, D., & Alfaro, A. (2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. *Ingeniare. Revista chilena de ingeniería*.
- Callejas Cuervo, M., Alarcón Aldana, A., & Álvarez Carreño, A. (2017). Modelos de calidad del software, un estado del arte. *Entramado*.
- Carvalho, J. P., Franch, X., & Carme, Q. (2006). *Un Catálogo de Factores de Calidad para la Definición de Requisitos No-Técnicos en la Selección de Componentes COTS*. Obtenido de <https://upcommons.upc.edu/bitstream/handle/2117/135310/carvalho.pdf?sequence=1&isAllowed=>
- Chacón Alvarado, L. (2002). *Automatización de la Biblioteca. Universidad Estatal a Distancia*. Obtenido de [https://www.ecured.cu/Lenguaje\\_de\\_marcado](https://www.ecured.cu/Lenguaje_de_marcado)
- Garlan et al., (1995). Garlan, R. Allen and J. Ockerbloom. "Architectural Mismatch: Why Reuse is So Hard". *IEEE Software*. 12(6):17-26, November, 1995
- Gómez, K. (27 de 07 de 2017). *Megapractical*. Obtenido de <https://www.megapractical.com/blog-de-arquitectura-soa-y-desarrollo-de-software/metodologias-de-desarrollo-de-software>
- Gómez, M. (2006). *Introducción a la metodología de la investigación científica*. Argentina.
- LaundrySoft. (27 de 08 de 2015). *Metodología y Ciclo de Vida del Software*. Obtenido de <https://medium.com/@laundrysoft/taller-1-metodolog%C3%ADa-y-ciclo-de-vida-del-software-7f505db08523>
- López Pino, J. (2010). *Servidores Web mas usados*. Obtenido de <http://lopezpino.es/2010/07/30/servidores-web-mas-usados/>
- López Torralba, M. (20 de 01 de 2015). *Mialtoweb*. Obtenido de <http://mialtoweb.es/definicion-de-aplicacion-web/>
- Lujan Mora, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*.

- Luján Mora, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Club Universitario.
- Marcos, S. V. (2016). Pruebas de rendimiento a componentes de software utilizando programación orientada a aspectos. *Ingeniería Industrial vol.37 no.3*.
- Martinez, F. (2013). *Maestros del Web*. Obtenido de <http://www.maestrosdelweb.com/educacion-para-desarrollo-de-un-pais/>
- Morales, O. (2003). *Fundamentos de la investigación documental y la monografía. Facultad de Odontología de la Universidad de Los Andes. Manual para la elaboración y presentación de la monografía*. Mérida, Venezuela.
- Neosoft. (2018). *¿Qué es una aplicación Web?* Obtenido de <https://www.neosoft.es/blog/que-es-una-aplicacion-web/>
- PRESSMAN, R. S. (2010). *Ingeniería de Software "Un Enfoque Práctico"*. México: MCGRAW-HILL INTERAMERICANA.
- Redondo, M. (2016). *Manual de calidad y procedimientos para la gestión del sistema de calidad de una empresa de desarrollo de software*.
- Saldaña, R. (2010). *Ingeniería de software Un enfoque práctico*. México: MCGRAW-HILL.
- Scalone, F. (2006). *Estudio comparativo de los modelos y estándares de calidad del software. Tesis Ingeniería de Calidad*. Buenos Aires: Universidad Tecnológica Nacional Regional de Buenos Aires.
- Szyperski, C. (1998). *Component Software: Beyond Object-Oriented Programming (ACM Press)*.
- Sommerville, L. (2011). *Ingeniería de Software (9na ed.)*. México: PEARSON EDUCACIÓN.
- Szyperski, C. (2002). *Component Software: Beyond Object-Oriented Programming*. . Arizona: United States Addison-Wesley Educational Publishers Inc.
- Yañez, D. (17 de 12 de 2014). *ranscripción de la importancia de la informática en la vida cotidiana, escolar y profesional*. Obtenido de <https://prezi.com/hp5rcmm3w9le/la-importancia-de-la-informatica-en-la-vida-cotidiana-escol/>