

APUNTES TEÓRICOS SOBRE MODELOS DE EVALUACIÓN DE CALIDAD EN PROCESOS DE DESARROLLO DE *SOFTWARE* PARA PERSONAS NO VIDENTES

ÁREA: DESARROLLO DE *SOFTWARE*

Verónica Lucía Sánchez Jaya ¹

Darío Javier Robayo Jácome²

Galo Mauricio López Sevilla³

1. Estudiante de Ingeniería en Sistemas, Pontificia Universidad Católica del Ecuador – Ambato, veronica.l.sanchez.j@pucesa.edu.ec
2. Magister en Gerencia Informática. Pontificia Universidad Católica del Ecuador, Docente de la Escuela de Ingeniería en Sistemas, dario.j.robayo.j@pucesa.edu.ec
3. Magister en Informática. Pontificia Universidad Católica del Ecuador, Docente de la Escuela de Ingeniería en Sistemas, glopez@pucesa.edu.ec .

RESUMEN

El presente trabajo tiene como objetivo plantear aspectos teóricos sobre los distintos modelos de evaluación de calidad para el desarrollo del *software* orientado a personas con discapacidad visual. Para el estudio del mismo se realizó una investigación exhaustiva en documentos de Google académico, en donde se ubicaron 30 documentos referentes al tema que fueron analizados de forma particular. El estudio de los mismos lanzó como resultado que los modelos existentes para evaluar la calidad y el desarrollo del *software* son genéricos, es decir, no son específicamente orientados al tipo de estudio propuesto.

PALABRAS CLAVE

Modelo de calidad, desarrollo de *software*, discapacidad visual, métricas.

ABSTRACT

The present work aims to propose theoretical aspects about the different models of quality assessment and development of software directed to peoples with visual disabilities. An exhaustive investigation was done in Google academic documents, 30 documents was found, they are related to the topic. The result was the existent models to evaluate the quality and the development of the software are generic, it means, they are not oriented to the type of study proposed.

KEYWORDS

Quality model, *software* development, visual disability, metrics.

INTRODUCCIÓN

Coral, Moraga, & Piattini (2010), indican que desde el punto de vista del *software* se pueden distinguir proceso, producto y recursos, donde la calidad de los dos primeros son los que cobran mayor importancia. Asimismo, Peñalva (2014), menciona que el modelo de desarrollo de *software* incluye mejorar la vida del usuario en la intercomunicación con el *software* alcanzando un grado de satisfacción en utilidad, productividad y confianza. Por otra parte, Callejas, Alarcón, & Álvarez (2017), mencionan que un modelo de calidad de *software* se refiere al nivel de ejecución de las principales características con las que debe cumplir un sistema computacional durante su ciclo de vida. De las definiciones mencionadas, se determina que un modelo de calidad en el desarrollo de *software*, se refiere al nivel de cumplimiento de las principales características durante su ciclo de vida, para brindar el mejor servicio a los usuarios.

Por otra parte, Galaz (2014), menciona que el modelo de calidad en forma general sirve también para que las organizaciones puedan realizar sus autoevaluaciones, a través de una revisión integral de sus estrategias y prácticas de gestión. A lo que, Pérez (2018), acota que un modelo de calidad es un conjunto de características y las relaciones entre ellas, que proporciona un marco para la especificación de requisitos de la calidad y su evaluación. Además, la Asociación Española para la Calidad (2018), menciona que los modelos contienen directrices para mejorar el cumplimiento de la calidad de un sistema y no requisitos, así mismo, indica que las mismas organizaciones pueden desarrollar modelos de calidad orientados a la mejora del proyecto.

Scalone (2006), señala que los modelos de calidad son los documentos que conllevan prácticas, organización y administración que ayuda a las instituciones a evaluar la calidad de sus productos. De la misma manera, Callejas, Alarcón, & Álvarez (2017), menciona que el modelo debe ir enfocado en la evaluación en cada etapa del desarrollo del *software* para tener un producto de calidad.

METODOLOGÍA

Se realizó búsquedas bibliográficas en Google Académico para recompilar información. Cada una de las investigaciones realizadas estuvo basada en el análisis profundo de su contenido partiendo desde lo que es un diseño de calidad de *software*, el desarrollo de *software* y *software* orientado a personas con discapacidad visual. Por lo tanto se llegó a analizar 63 documentos para lograr eficacia en la información, contrastando la misma y encontrar aspectos relevantes en común. Finalmente, la selección fue de 30 trabajos que cumplieran con los temas descritos anteriormente y con criterios de calidad en la información que mostraban.

RESULTADOS

Modelo de Calidad

Pérez & Martínez (2011), al hacer mención a la calidad de *software*, sostienen que ésta alude a las diversas cualidades que determinan su beneficio; es el nivel en que el *software* lleva a efecto las condiciones determinadas. Por otra parte, Estévez

(2014), manifiestan que en las últimas décadas, desde que hicieron su aparición las diversas tecnologías del *software*, se ha tratado de hacer una evaluación de la calidad de los propios productos, teniendo como meta el mejoramiento de cada una de sus características con el fin de obtener productos *software* más cualificados que complazcan las exigencias de los clientes versados en la materia.

Galaz (2014), acota que el modelo de calidad sirve también para que las organizaciones puedan realizar sus autoevaluaciones, a través de una revisión integral de sus estrategias y prácticas de gestión. Mientras que, Pérez (2018), menciona que un modelo de calidad es un conjunto de características y las relaciones entre ellas, que proporciona un marco para la especificación de requisitos de la calidad y su evaluación. Además, la Asociación Española para la Calidad (2018), menciona que los modelos contienen directrices para mejorar el cumplimiento de la calidad de un sistema y no requisitos, así mismo, indica que las organizaciones pueden desarrollar modelos de calidad orientados a la mejora del proyecto.

Objetivos de un Modelo de Calidad

ISO/IEC 25000 menciona que uno de los objetivos de un modelo de calidad es:

“Guiar la evaluación de calidad de producto *software* estableciendo criterios para la especificación de requisitos de calidad de *software*, sus métricas y su evaluación”.

Según la ISO/IES (9001:2015),

- Los objetivos de calidad son metas, retos que se definen a partir de la planificación estratégica de la empresa y de su política de calidad.
- Los objetivos de calidad deben ser establecidos por la alta dirección de la organización. Tienen que ser coherentes con la política de calidad y perseguir la mejora continua. Según la norma ISO los objetivos de la calidad deben ser:
 - Similares con la política de la calidad.
 - Ser medibles.
 - Considerar los requisitos aplicables.
 - Ser apropiados para la aprobación de los productos y servicios y para el acrecentamiento del gusto del cliente.
 - Ser ente de seguimiento.
 - Ser comunicados.
 - Actualizarse, según sea necesario.

Calidad en el desarrollo de *Software*

Estayno, Dapozo, Cuenca, & Greiner (2009), manifiestan que los Modelos y Métricas para Evaluar la calidad de software, “son el conjunto de factores de calidad, y de relaciones entre ellos, que proporciona una base para la especificación de requisitos de calidad y para evaluación de la calidad de los componente de *software*” (p. 382-388). Por consiguiente, el definir un modelo de calidad durante el desarrollo de software permitirá que aumente el nivel de satisfacción de los clientes y/o usuarios.

Desarrollo de *Software*

En su texto Ingeniería de *Software*, Pressman (2010), señala que el *software* se encuentra presente en cada instante de nuestra existencia, ya que se emplea en la mayor parte de las tareas cotidianas, así como brinda la posibilidad de elaborar sistemas complejos con la mejor calidad en un tiempo razonable.

Por su parte, Trigo (2010), indica que en 1958 John W. Tukey en uno de sus artículos por primera vez publicó el término *software*, para distinguirlo del hardware. El *software*, según la R.A.E., es el conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

En el transcurso de su evolución, el *software* ha tenido avances por demás significativos, llegando a convertirse en una gran industria, ayudando a quienes lo desarrollan a alcanzar programas de un mantenimiento fácil y escalable. Es así que, Moreno (2017), menciona que es muy importante tener cuidado y mantenimiento especial en el *software*, desde el instante de su elaboración hasta el momento en que los usuarios adquieren finalmente el producto. Así mismo, señala que la Ingeniería de *Software* no abarca solamente procesos técnicos sino también diferentes aspectos tales como la gestión de proyectos de *software* y la creación de teorías, métodos y herramientas de apoyo, al momento de su elaboración.

Criterios de calidad en el desarrollo de *software*

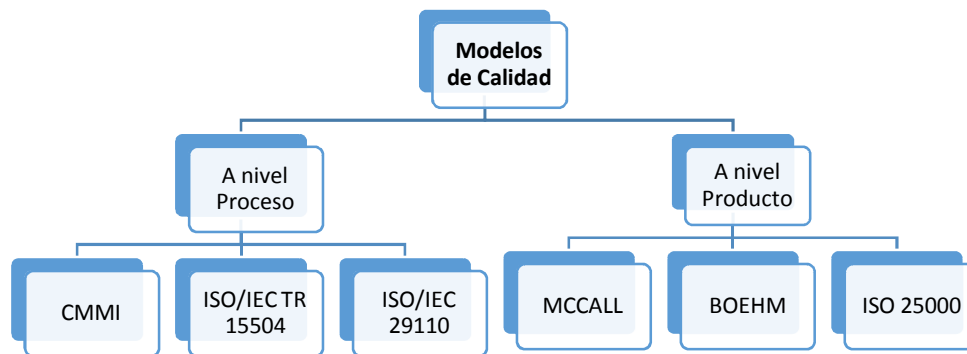
Los criterios de calidad son aspectos o condiciones que un producto debe cumplir para alcanzar un nivel adecuado de eficiencia, funcionalidad, y grado de satisfacción de los usuarios, así, en la ISO 9000:2015, se consideran los siguientes criterios:

- **Funcionalidad:** que el *software* cumpla con los requerimientos.
- **Fiabilidad:** el *software* no tenga fallos.
- **Eficiencia:** optimización de recursos del sistema.
- **Usabilidad:** facilidad en la que el *software* puede ser entendido y usado.
- **Mantenibilidad:** facilidad con la que el *software* puede soportar cambios y/o actualizaciones.
- **Portabilidad:** que el *software* sea adaptable a otras plataformas de hardware y *software*.

Modelos de calidad en el desarrollo de *software*

Estayno, Daposo, Cuenca, & Greiner (2009) y Callejas, Alarcón, & Álvarez (2017), mencionan que la calidad del *software* es una compleja mezcla de componentes y partiendo de las especificaciones de los modelos dados por la ISO, presentan los tipos de modelos de calidad en el desarrollo de *software*:

Figura 1: Tipos de modelos de calidad



Fuente: Estayno, Daposo, Cuenca, & Greiner (2009): Callejas, Alarcón, & Álvarez (2017)

Modelos de Calidad a nivel de Proceso

Callejas, Alarcón, & Álvarez (2017), menciona que la calidad del *software* debe ser programada desde el inicio del proyecto, y posteriormente en cada etapa del proceso de desarrollo, para garantizar un óptimo nivel de cumplimiento de los factores de calidad en la evolución del mismo. Los modelos de calidad a nivel de proceso son:

CMMI (Capability Maturity Model Integration)

Según Tuya, Ramos, & Dolado (2007); (Yepes, César, & Gómez, 2015) coinciden que CMMI es un modelo que integra la arquitectura, enfoque, contenido y aplicación para obtener productos de calidad; además, mencionan que este modelo contiene un enfoque disciplinado para mejorar los procesos de una organización y para que se puedan implantar procesos de calidad. Enfatizando lo antes mencionado, el modelo CMMI ayudará a determinar de mejor manera los procesos esenciales para optimizar el proceso del *software* y obtener productos de calidad.

Madrid (2008), indica que CMMI tiene cinco niveles de madurez, que se mencionan a continuación:

1 Inicial: en este nivel las organizaciones no cuentan con un ambiente estable para el desarrollo y mantenimiento de *software*.

2 Administrado: “En este nivel las organizaciones disponen de prácticas institucionalizadas de gestión de proyectos, existen métricas básicas y un razonable seguimiento de la calidad. La relación con subcontratistas y clientes está gestionada sistemáticamente”.

3 Definido: En este nivel, las organizaciones cuentan con correctos funcionamientos, técnicas de ingeniería más precisa y un nivel más avanzado de métricas en los procesos.

4 Administrado Cuantitativamente: “Las organizaciones disponen de un conjunto de métricas significativas de calidad y productividad, que se usan de modo sistemático para la toma de decisiones y la gestión de riesgos”.

5 Optimizado: “La organización completa está orientada a la mejora continua de los procesos. Se hace uso intensivo de las métricas y se gestiona el proceso de innovación”.

ISO/IEC TR 15504 (Software Process Improvement and Capability Determination)

Loon (2004), menciona que es un modelo que se centra en la realización y cumplimiento de los procesos de desarrollo, mantenimiento de sistemas de información y productos de software; igualmente, menciona que este modelo contiene los requisitos necesarios para garantizar evaluaciones de procesos consistentes; y, que muchas de las organizaciones lo utilizan para evaluar la capacidad de sus proveedores.

De la misma forma, la Asociación Española para la Calidad – QAEC, señala que: está diseñada para facilitar una aproximación común para realizar valoraciones de procesos, haciendo posibles comparaciones entre los resultados de las mismas. Estos resultados se pueden basar en diferentes modelos de valoración (siempre que sean compatibles con el estándar) y métodos de valoración.

Destacando lo mencionado por los autores, es necesario tener en cuenta este modelo ya que permite realizar valoraciones con diferentes modelos y garantiza que las evaluaciones de los procesos sean estables.

ISO /IEC 29110 (*Software Process Improvement and Capability Determination*)

Yepes, César, & Gómez (2015), indican que la ISO/IEC 29110 es un estándar que está encaminado a las pequeñas y medianas organizaciones, con el fin de alcanzar niveles internacionales de competitividad. Además, busca mantener sus productos con atributos de calidad tanto en el proceso como en el producto. Igualmente, indica que es una norma que se desarrolló con el fin de mejorar la calidad del producto y/o servicio de software mejorando el desempeño de la organización.

Madruñero (2018), indica que:

Es una norma ISO que lleva como denominación Ingeniería de Software - Perfiles de Ciclo de Vida en Pequeñas Entidades, esta norma es basada en las siguientes normas: mexicana NMX-I-059-NYCE-2011 (MoProSoft), en la ISO/IEC 12207, la ISO/IEC 15289, la ISO/IEC 15504 entre otras. La propuesta mexicana fue ofrecer sus procesos de la categoría operación como un perfil básico, la categoría de gerencia como perfil intermedio y la categoría alta Dirección como perfil Avanzado (p. 47).

Modelos de Calidad nivel Producto

Bevan (2010), indica que la finalidad del modelo de calidad a nivel de producto es especificar y evaluar el cumplimiento de criterios del producto, para lo cual se aplican medidas internas y/o medidas externas.

McCALL

Medina (2014), indica que este modelo aparece en la Fuerza Aérea Estadounidense, que lo utilizó en los productos de *software* del Departamento de Defensa en 1977.

McCall se basa en once factores de calidad los mismos que están organizados en tres ejes, donde el usuario puede examinar la calidad del producto. De la misma manera, Reinoso, Vivanco, & Coba (2008), señala que este modelo desglosa el concepto de calidad en tres usos importantes para un producto de *software*.

Según ambos autores los ejes o usos son:

Operación del Producto: se refiere a las características de operación y los factores de calidad que integran este apartado son:

- Facilidad de Uso, por parte de los usuarios del sistema
- Integridad, para proteger al programa de accesos que no han sido autorizados
- Eficiencia, en la ejecución del programa y en la utilización de recursos por parte del mismo.
- Corrección o exactitud
- Fiabilidad, que el sistema no falle.

Revisión del producto: es la habilidad para ser cambiado e incluye los siguientes factores:

- Facilidad de prueba, asegurar que el programa esté libre de errores y conoce las especificaciones del usuario.
- Facilidad de Mantenimiento, esfuerzo requerido para encontrar y solucionar errores que se presenten en la operación del sistema.
- Flexibilidad, facilidad de realizar cambios.

Transición del Producto: describe la adaptación al nuevo ambiente e incluye los siguientes factores de calidad:

- Reusabilidad, se puede volver a usar el *software*.
- Portabilidad, capacidad de transferir un programa de un ambiente a otro.
- Interoperabilidad, se puede unir a un sistema con otro.

BOEHM

Este modelo fue desarrollado por Barry Boehm en el año de 1978; el factor principal en el que se centra el modelo es que el *software* realice lo que el usuario desee que realice.

El *software* tiene que utilizar correcta y eficazmente los recursos del computador.

Tiene que ser de fácil de uso y aprendizaje para los usuarios.

Estar bien diseñado, codificado; ser probado y mantenido fácilmente.

En su estructura, el modelo Boehm, presenta tres niveles para las características: de alto nivel, de nivel intermedio y características primitivas. Cada una de estas características contribuye a tener un nivel general de calidad.

Scalone (2006) y Fillottrani (2007), presentan tres niveles de la calidad del *software*:

Características de alto nivel:

- Utilidad per-se: qué tan eficiente, usable y confiable es el producto en sí mismo.
- Mantenibilidad: qué fácil es entender, modificar y retestearlo.
- Utilidad general: en qué grado puede seguir utilizándose si se cambia el entorno.

Características de nivel intermedio:

- 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:

- **de portabilidad:**
 - independencia de dispositivos
 - auto-contención
- **de confiabilidad:**
 - auto-contención
 - exactitud
 - completitud
 - consistencia
 - robustez/integridad

ISO/IEC 25000 (*System and Software Quality Requirements and Evaluation*)

Laporte (2018), indica que el objetivo de esta norma es crear un marco de trabajo para la calidad de productos de *software*, y este estándar es el progreso de la ISO 9126. Está dividida en 5 estándares. Tello (2016), explica los diferentes apartados de esta norma:

1. ISO/IEC 2500n División de Gestión de Calidad

- **ISO/IEC 25000 Guía de SQuaRE:** abarca el modelo de la arquitectura de SQuaRE, la terminología, una síntesis de las partes, los usuarios predichos y las partes asociadas y los modelos de referencia.
- **ISO/IEC 20001 Planificación y Administración:** establece los requerimientos y orientaciones para gestionar la evaluación.

2. ISO/IEC 2501n – División de Modelo de Calidad

Presenta las características específicas internas, externas y en uso del producto.

- **ISO/IEC 25010 – Modelos de Calidad del Software y Sistemas:** detalla el modelo de calidad para el producto *software* y para la calidad en uso.
- **ISO/IEC 25012 – Modelo de Calidad de los Datos:** precisa un modelo general para la calidad de los datos, adaptable a los datos que estén recopilados y formen parte de un Sistema de Información.

3. ISO/IEC 2502n División de Medición de Calidad

Abarcan un modelo de referencia de la medición de la calidad del producto, definiciones de medidas de calidad a nivel interno, externo y en uso, y pautas prácticas para su aplicación.

- **ISO/IEC 25020 Guía y Modelo de Referencia de Mediciones:** muestran una explicación y un modelo de referencia común a los elementos de medición de la calidad.

- **ISO/IEC 25021 Elementos de Métricas de Calidad:** delimita un conjunto de métricas base y derivadas que puedan ser usadas a lo largo de todo el ciclo de vida del *software*.
- **ISO/IEC 25022 – Métricas de Calidad en Uso:** puntualiza las métricas para realizar la medición de la calidad en uso del producto.
- **ISO/IEC 25023 Métricas de Calidad del Producto y Sistema de Software:** establece las métricas para realizar la exactitud de la calidad de productos y sistemas *software*.
- **ISO/IEC 25024 Métricas de Calidad de los Datos:** implanta las métricas para realizar la medición de la calidad de datos que son parte de un Sistema de Información y están almacenados de manera estructurada.
- 4. **ISO/IEC 2503n División de Requerimientos de Calidad estas normas ayudan a** definir requerimientos de calidad que pueden ser utilizados en el proceso de elicitación de requerimientos de calidad del producto de *software* a desarrollar o como entrada del proceso de evaluación.
 - **ISO/IEC 25030 Requerimientos de Calidad:** provee un conjunto de recomendaciones para realizar la determinación de los requerimientos de calidad del producto de *software*.
- 5. **ISO/IEC 2504n División de Evaluación de Calidad**

Contiene normas que facilitan requisitos, recomendaciones y pautas para realizar el proceso de evaluación del producto *software*.

 - **ISO/IEC 25040 Guía y Modelo de Referencia de la Evaluación:** plantea un modelo de referencia general para la evaluación, que considera las entradas al proceso de evaluación, las restricciones y los recursos que son ineludibles para alcanzar salidas oportunas.
 - **ISO/IEC 25041 Guía de Evaluación para los Desarrolladores, Compradores y Evaluadores Independientes:** desde el punto de vista de los desarrolladores detalla los requisitos y recomendaciones para la implementación práctica de la evaluación del producto *software*.
 - **ISO/IEC 25042 Módulos de Evaluación:** define el Módulo de Evaluación y la documentación, estructura y contenido que se debe utilizar a la hora de definir los módulos.
 - **ISO/IEC 25045 – Módulo de Evaluación para la Recuperabilidad:** delimita un módulo para la evaluación de la subcaracterística Recuperabilidad.

Software para personas con discapacidad visual

En Ecuador, como señalan Martínez, Silva & Naranjo (2016), en los últimos años, se ha tratado de incluir a las personas con discapacidad visual en el ámbito de la tecnología; esto ha provocado que, tanto en las instituciones educativas, como a nivel comercial, se generen proyectos de aplicaciones o tecnologías que permitan ayudar a las personas no videntes.

La ayuda de las nuevas tecnologías, como destaca Ortiz (2009), son una herramienta para integrar a las personas invidentes en un mundo moderno; además, manifiesta que de cierta forma ésta inclusión permitirá desaparecer las fronteras entre personas videntes y no videntes, ya que el avance de la tecnología ha hecho que ésta se vuelva un medio de sobrevivencia y competitividad.

Criterios de calidad para *software* orientado a personas no videntes

Según Rojas (2013), los criterios a tomarse en cuenta para el desarrollo de *software* orientado a personas con discapacidad visual pueden ser:

➤ **Accesibilidad**

La asociación SIDAR (Acceso Universal) define a la accesibilidad como “el grado en que toda persona puede acceder a utilizar un objeto, visitar un lugar o acceder a un servicio independiente de sus capacidades técnicas o físicas”.

- **Percepción:** es la información y componentes de la interfaz de usuarios que son presentados y la manera en que los usuarios la pueden percibir.
- **Operatividad:** Tanto los componentes de la interfaz y la navegación deben funcionar correctamente.
- **Entendible:** la información que proporciona el sistema y la interfaz deben ser descifrables.
- **Robustez:** La *Web Content Accessibility Guidelines* (2008) define que el contenido debe ser adecuadamente robusto de tal forma que pueda ser descifrado apropiadamente.

➤ **Usabilidad**

La norma ISO/IEC9126 define que: "La usabilidad se refiere a la cabida de un *software* de ser entendido, asimilado, usado y ser atrayente para el usuario, en condiciones específicas de uso". Además, Bevan (1999), indica que la usabilidad no puede valorarse estudiando un producto de manera aislada. Esta definición hace hincapié en los atributos internos y externos del producto los mismos que favorecen a su eficiencia y funcionalidad.

CONCLUSIONES

Debido a la poca importancia dada al Desarrollo de *Software* orientado a personas con discapacidad visual, los modelos que se utilizan para la evaluación de los mismos son genéricos. Según los estudios realizados se asume que el contar con un modelo específicamente orientado a este tipo de *software*, permitirá alcanzar productos que incorporen características de accesibilidad y usabilidad, de manera que, este sector vulnerable de la sociedad pueda contar con los mejores servicios que brinda la tecnología.

Se han definido aspectos relevantes en el proceso de desarrollo de *software*, referentes a la calidad, confiabilidad, accesibilidad y usabilidad que deben ser tomadas en cuenta durante todo el proceso de desarrollo.

REFERENCIAS BIBLIOGRÁFICAS

- Rojas , Y., & Rojas Galindo, Y. (2013). *repositorio umsa*. Obtenido de <http://repositorio.umsa.bo/xmlui/handle/123456789/7771>
- Alistair, W. (7 de septiembre de 2017). Luxury Brand Marketing – The Experience Is Everything! *Journal of Brand Management*., 43-57.
- Arenas , J. (2015). Obtenido de <https://es.slideshare.net/clauddiaa/factores-de-calidad-segn-mc-call>
- Baker, C. (2011). *Marketing*. España: Paraninfo.
- Bastidas, J. (2010). *Guía de aplicación del marketing sensorial y marketing directo como estrategia de fidelización de clientes*. Ecuador: UDLA.
- Buitrago, N., & Gómez, C. (2010). *Marketing sensorial*. Medellín: EIA.
- Callejas, M., Alarcón, A., & Álvarez, A. M. (2017). Modelos de calidad del *software*, un estado del arte*. *Entramado*, 236-250.
- Cárdenas, S. (2015). *Marketing Sensorial: influencia de los sentidos en el comportamiento y toma de decisiones de los clientes de tiendas de ropa*. Bogotá.
- Carillo, E., & Gómez, B. (2016). *Marketing Sensorial. Influencia y evolución en el sector hotelero*. España: CET.

- Charles, D., Newman, A., & Wright, L. (2016). Enhancing consumer empowerment. *European Journal of Marketing*, 40(9), 925-935.
- Espina, C. (2016). *Arquitectura sensorial: La atmósfera del espacio comercial*. España.
- Estévez, Y. (12 de junio de 2014). Obtenido de www.unihorizonte.edu.co/revistas/index.php/TECKNE/article/view/129
- Fernández, M. (2016). El marketing experiencial: antecedentes y estado actual de la investigación. *Revista Interdisciplinaria*(14), 236-252.
- Galdos, K. (2016). *Influencia del marketing sensorial en la imagen corporativa*. Perú.
- Gavilán, D., Manzano, R., & Serra, T. (2012). *Marketing sensorial: comunicar a través de los sentidos*. España: Havard Deusto Marketing y Ventas.
- Gaviria, H., & Lopera, G. (2015). El marketing sensorial: una lectura acerca de su Implementación en los almacenes Befit, Chevignon y Vélez ubicados en el centro comercial el tesoro de la Ciudad de Medellín. *Revista Facultad de Publicidad*, 2(1), 1-26.
- Gómez, M., & García, C. (2015). Marketing Sensorial. *Comercialización e Investigación de Mercados*, 30-40.
- IEMD. (5 de mayo de 2017). *Instituto Internacional Español de Marketing Digital*. España.
- Madrid, J. (2008). Obtenido de http://www.tesis.uchile.cl/tesis/uchile/2008/madrid_jm/sources/madrid_jm.pdf
- Madruero, E. (abril de 2018). *repositorio utn*. Obtenido de <http://repositorio.utn.edu.ec/bitstream/123456789/8165/1/PG%20638%20TESIS.pdf>
- Martínez. (7 de mayo de 2015). *Marketing olfativo: los aromas que aumentarán tus ventas*. Obtenido de celestinomartinez.com/2011/05/03/marketing-olfatorio-aromas-aumentar-ventas/
- Mejía, J., & Gómez, C. (2015). La gestión del Marketing que conecta con los sentidos. *Revista Escuela de Administración de Negocios*(73), 168-183.
- Merca2.0. (8 de Abril de 2016). *Pasos claves para desarrollar el marketing sensorial*. Obtenido de <https://www.merca20.com/3-puntos-clave-para-desarrollar-estrategias-de-marketing-sensorial-y-experiencias/>
- Moreno, M. (Mayo de 2017). *repositorio pucesa*. Obtenido de <http://repositorio.pucesa.edu.ec/bitstream/123456789/1929/1/76434.pdf>
- Nigel, B. (1999). Meeting User Needs for Quality. *Journal of systems and software*, 89-96.
- Rieunier, S. (2011). *El Marketing Sensorial en el punto de venta*. Paris: Dunod.
- Rodas, J., & Cervantes, J. (2010). El marketing sensorial en el proceso de toma de decisiones de compra: análisis exploratorio. *Contaduría, administración e informática*, 1-22.

Singhal, S., & Khare, K. (2015). Does Sense Reacts For Marketing - Sensory Marketing. *International Journal of Management*, 5(5), 1-18.

Web Content Accessibility Guidelines. (2008). Obtenido de <http://www.w3.org/TR/WCAG20/> Web Content Accessibility Guidelines (WCAG) 2.0

Yepes, J., César, P., & Gómez, O. (Diciembre de 2015). Obtenido de <http://rte.espol.edu.ec/index.php/tecnologica/article/view/454/319>

Zuloaga, L., & Obando, M. (2015). *Marketing Sensorial en Banana Boat*. Bogotá: CESA.